Segmentation Based Semi-Regular Remeshing of 3D Models Using Curvature Adapted Subdivision Surface Fitting

Abstract

This paper proposes a novel method of semi-regular remeshing for triangulated surfaces to achieve superior triangles lead to advanced visualization of 3D model. It is based on mesh segmentation and subdivision surface fitting which uses curvature adapted polygon patches. Our contribution lies in building a sophisticated system with three stages i.e. curvature-aware mesh segmentation, submesh surface fitting to generate a high quality semi-regular mesh and finally stitching the segments using an efficient algorithm. Our method uses centroidal Voronoi tessellation (CVT) and Lloyd's relaxation to generate curvature adapted site centers. Geodesic distances from site centers are used for labeling segments and indexing corner vertices for each segment boundary. Using information of site centers and corner vertices, feature adapted polygonal patches is generated for each segment. These patches are then subdivided and optimized using squared distance metric to adjust position of the subdivision sampling with segment details and prevent oversampling. At last, an efficient stitching algorithm is introduced to connect regular submeshes together and build the final semi-regular mesh. We have demonstrated the results of our semi-regular remeshing algorithm on meshes with different topology and complexity and compared them with known methods. Superior triangle quality with higher aspect ratio together with acceptable distortion error is achieved according to the experimental results.

Keywords: visualization of triangulated mesh, computer graphics, semi-regular remeshing, centroidal voronoi tessellation (CVT), mesh segmentation, subdivision surface fitting, stitching.

*Corresponding Author: Hossein Ebrahimnezhad

Associate Professor of Electrical and Computer Engineering

Computer Vision Research lab, Electrical Engineering Faculty, Sahand University of Technology, Tabriz, Iran.

Email: ebrahimnezhad@sut.ac.ir

Tel.: (+9841) 33459343, Fax: (+9841) 33444322

1. Introduction

Efficient mesh triangulation of 3D models is an important issue in computer graphics which leads to fine visualization of the model. The models generated by 3D scanning tools are irregular meshes and contain many redundant vertices. Irregular meshes are not suitable for mesh compression and lead to low transmission efficiency. Also, level of details and multi-resolution representation of 3D models requires semi-regular meshes and it is still an interested and active research area especially in mesh compression applications [1]. Semi-regular meshes mostly contain regular vertices. The valance of a regular vertex is 6 for interior vertices and 4 for boundary ones. Semi-regular remeshing algorithms keep the number of non-regular vertices small and constant [2, 3] without any loss of accuracy. It is achieved by higher sampling rate to compromise loss of accuracy. Since, mesh regularization techniques, especially subdivision-based ones, try to increase face quality without paying attention to redundant vertices they cause oversampling during regularization. The oversampling of the mesh is an important factor because it raises the number of vertices in low detail areas of the mesh and causes computational complexity for next processing steps. Also post-simplification of the regulated mesh is not an appropriate solution because it can affect mesh regularity and/or accuracy. To overcome the above mentioned problems, we propose a novel mesh regularization algorithm for which we derive a semi-regular mesh with reduced vertex redundancy. It is done by segmenting a mesh into high detail and low detail segments using curvature feature. Then, adaptive polygonal patches corresponding to each segment are generated as a control mesh. Moreover, adaptively generated polygonal patches control vertex distribution in individual submeshes according to segment details. Then, subdivision surface fitting is applied to every curvature adapted polygonal patch and new regular submeshes are generated. At last, by stitching all the submeshes together, we build the final semi-regular mesh. Our proposed framework produces a high quality semi-regular mesh and at the same time reduces the number of redundant vertices.

The first goal in high performance semi-regular remeshing is to generate high quality triangles with acceptable error range. In a triangular mesh, the highest quality belongs to the equilateral triangles and loss of quality is measured by deviation of triangles from equilateral form [4]. Distributions of aspect ratio for the higher quality triangles are closer to one in comparison to the lower quality triangles. We are interested in semi-regular meshes with higher quality since they have lower costs for further processing such as storage compression, view dependent representation, multi-resolution analysis and shape editing [5]. Besides, narrow triangular meshes with low shape quality cause further application problems, such as poor finite element matrices, which can compromise the efficiency of convergence and accuracy of solution [6].

1.1. System Overview

The main contribution to improve efficiency is that our new algorithm computes vertex-distribution of the semi-regular mesh based on details of the original 3D model. It is done by allocating curvature-dependent polygonal patches to its corresponding segments. By this way, vertex concentration is controlled in high and low detail areas. Subdivision surface fitting using squared distance metric is then adaptively applied through polygonal patches with respect to segmented sharp and smooth regions and regular submeshes are generated. Finally, in the post processing step, submeshes are stitched together. This method reduces vertex redundancy in the resulting semi-regular mesh for a given approximation error compared to other mesh regularization techniques. The major novelty of the work is to use curvature dependent polygon patches to distribute the vertices adaptively to preserve details of the model. It is achieved by combination of centroidal Voronoi tessellation and mesh segmentation with subdivision surface fitting to generate an adaptive semi-regular surface for the entire mesh. Block diagram of our proposed method is shown in Figure 1.

In this paper, the key contributions are: (1) A segmentation algorithm to separate high and low curvature areas of the mesh using CVT and region labeling (section 2). (2) Building an adaptive polygonal patch for every segmented region and SDM-based subdivision surface fitting to build regular submeshes (section 3). (3) A stitching algorithm to combine submeshes together and construct final semi-regular mesh (section 4). Section 5 shows the experimental results and section 6 is the conclusion and future works. Related works similar to our contributions are introduced as follows.

1.2. Related Work

Semi-regular remeshing received a considerable attention in recent years. It has ever increasing ranges of applications including progressive transmission, multi-resolution analysis and compression of 3D geometric models [7]. For instance, Sun et. al. used semi-regular lunar surface for view dependent progressive transmission and rendering [8]. Different methods to regularize meshes can be put in two major groups including mapping-based (global) [9] and subdivision-based (local) techniques [10].

In mapping-based methods there is a one to one correspondence between original and remeshed triangles [11]. It is known as geometry image (GIM) and mapping distortion is the main challenge of these methods. Zhong et.al. used anisotropic surface meshing with conformal mapping [12]. Choi et al. proposed a robust quasi conformal surface remeshing which is adaptive to area distortion of parameterization [13]. In subdivision oriented techniques all the high and low detail regions of the mesh are uniformly sampled and regulated [14]. Kammoun et.al. generated optimized semi-regular meshes to improve performance of wavelet coder [1]. Lee et. al. used displaced butterfly subdivision surfaces to remesh approximated point cloud [15]. The main challenges of these methods are over sampling in low detail areas and smoothing effect in sharp regions. Our research deals with the second group and aims to compensate the drawbacks of subdivision-based semi-regular remeshing techniques. In the following we explain the related work about the main blocks presented in this study.

Mesh Segmentation: Different methods of mesh segmentation have been comprehensively summarized by Shamir [16]. Basically, the methods can be categorized as one of these groups including region growing [17], watershed methods [18], hierarchical clustering [19], feature points [20], skeleton-based segmentation [21] and feature sensitive segmentation [22].

Our proposed segmentation algorithm uses mesh coarsening based on CVD (centroidal voronoi diagram). Then, mesh is segmented using geodesic distance labeling. CVD is one of the most common techniques for direct remeshing. A comprehensive study of remeshing techniques can be found in [23]. Many recent mesh regularization algorithms use Voronoi diagrams [5] and it is still an effective tool for surface remeshing [24]. Our method benefits from curvature adapted CVD sampling for mesh coarsening. It causes to generate feature adapted segments which will be used for adaptive surface fitting in the next step.

Surface fitting: Optimized surface fitting can be categorized into three major groups including: point distance minimization (PDM) [25], tangent distance minimization (TDM) [24] and squared distance minimization (SDM) [27]. A complete analysis of optimization methods for subdivision surface fitting is presented in [28]. We have used SDM-based optimization method for its faster convergence and reliability with respect to other techniques. Our implemented algorithm is a modified version of [29] for best fitting to our segmented mesh. Curvature adapted polygonal patches after subdivision act as an initial control mesh for SDM-based surface fitting in this paper.

Stitching: Two different applications of 3D stitching in the literature are: (1) mesh repair [30], (2) connecting segmented parts of a mesh together [31]. We have introduced a stitching algorithm to connect generated submeshes together and build the entire semi-regular mesh.

2. Mesh Segmentation

3D model segmentation in our proposed method has two steps including curvature-based CVD remeshing and distance-based region labeling.

2.1. Curvature-dependent Voronoi diagram remeshing

Voronoi diagrams (or Voronoi Tessellation) are important geometric data structures for remeshing. Let $\mathrm{P} = \{p_1, p_2, ..., p_n\}$ be a set of points (so-called *sites*) in R^d. We associate to each site p_i its *Voronoi region* $V(p_i)$ such that:

$$V(p_i) = \{x \in R^d : \|x - p_i\| \le \|x - p_j\|, \forall\ j \ne i\} \qquad (1)$$

The collection of the non-empty Voronoi regions and their faces, together with their incidence relations, constitutes a cell complex called Voronoi Diagram (VD) of P.

A Centroidal Voronoi diagram (CVD) is a VD where each Voronoi site p_i is also mass centroid of its Voronoi region [32]:

$$p_i = \frac{\int V(p_i) x \rho(x)}{\int V(p_i) \rho(x)} \qquad (2)$$

where, $\rho(x)$ is the density function and controls the size of the Voronoi regions.

In CVDs energy is minimized by

$$E = \sum_{i=1}^{n} \int_{V(p_i)} \rho(x) \|x - p_i\|^2 dx \qquad (3)$$

Recent studies have introduced new investigation techniques using isotropic Voronoi diagrams [33, 34]. We make no assumption on the points p_i, as centers of mass of their respecting region. This generalization is useful when considering non-planar meshes, where the best location of the p_i points is not the cluster centroids. It is achieved by choosing user defined density function in equation (2) for obtaining non-uniform feature-dependent coarsening of the entire mesh. We initialize our coarsening process with randomly distributed set of seed vertices. Then, we iterate the flooding energy minimization in equation (3) and seed repositioning until convergence is achieved. The seed repositioning step moves the seed of each site to the closest vertex to the 3D centroid of the site.

For every vertex v of the triangular mesh, we use the method of [35] to estimate the discrete Gaussian and mean curvatures denoted as $K(v)$ and $H(v)$, respectively. Then, we define a hybrid curvature function as:

$$D(v) = |K(v)| + |H(v)| \qquad (4)$$

$D(v)$ is the sum of absolute values of Gaussian and mean curvatures. For every vertex position $v \in R^d$ of the mesh, we define variable $\rho(v)$ in equation (2) as:

$$\rho(v) = D(v) \qquad (5)$$

The map of curvature distribution of $D(v)$ on the mesh surface is shown in four different models in Figure 2. The color of the top and bottom of color-bar is related to the highest and lowest values of $D(v)$, respectively. Color-bar is normalized according to $D(v)$ values in four models between 0 and 1. It is shown that high curvature areas are extracted by curvature function defined in equation (4).

Energy minimization is done according to equation (3) by using Lloyd's relaxation method [36] and our curvature-based density function in equation (5) to obtain a non-uniform feature-dependent segmentation. Segmentation reduces the required memory space in the following processing stages. Also, feature-based remeshing inversely relates segmented areas to mesh details. It means that regions with high curvature details will have smaller areas compared with low detail regions. Figure 3 shows the results of CVT and Lloyd relaxation for 60 randomly distributed sites on two dimensions with non-constant density function $\rho(v(x, y, z))_{z=0} = e^{-(x^2+y^2)}$, where origin is placed on the center of rectangle. Distribution of curvature-based density function is high at the center of the rectangle and decreases when closes to the borders. Seeds are selected randomly. After relaxation, more seeds are assigned to high curvature areas around center and less seeds are distributed near borders with low curvature. Here, our feature-based site centers are generated, but in this step we don't use them for remeshing. Instead, they are used for labeling vertices in the original mesh as described in the next part. Moreover, they are considered as centers of polygonal patches.

2.2. Geodesic-based region labeling

To have independent control over vertices inside each segment, we need to determine which group of connected vertices in the original mesh is related to which one of site centers. We create this relationship by using geodesic distance. In this work, we use the “fast marching on the triangulated domains” utilized in [37]. It computes approximate geodesic paths between two vertices in $O(n\log n)$ time per path (n is the number of vertices in the mesh). We define each arbitrary vertex v_k , $k = 1,...,n$ of the original mesh has label index $Label(i)_{v_k}$ if $\forall p_i$, $i = 1,2,...,m$ (m is the total number of CVD site centers) we have

$$Label(i)_{v_k} = \arg\min_i \left(d_{geo}(p_i, v_k) \right) \qquad (6)$$

$d_{geo}(p_i, v_k)$ is geodesic distance between points p_i and v_k, computed by fast marching algorithm and p_i is the coordinate of a CVD site center which has the minimum geodesic distance from vertex.

2.3. Corner detection and striping segment borders

Output of the previous stage generates *m* groups of labeled vertices, in which, *m* is the total number of user defined CVD site centers and at the same time equals to the number of segments. According to geodesic distance metric in equation (6), each group of vertices belongs to a segment in an isolated island. We define three types of faces generated from labeling procedure as follows:

Type I face: A face that all of its three vertices are only members of one segment (interior-face)

Type II face: A face that has vertices belongs to exactly two different segments (strip-face)

Type III face: A face that all its three vertices belong to more than two different segments (corner-face)

By the above definition, corner faces are generated and the gap between two segments creates strips. Each strip is a collection of strip faces which starts from a corner face and ends to another corner face.

Now, we are ready to define two types of boundary vertices for every segment according to the above mentioned type III face. The first one is the “corner vertex". We call all three vertices of every corner-face as “corner-vertex". Each corner-vertex is in common with one corner-face. The second one is the "strip-vertex". It is a vertex placed on the segment boundary but is not “corner-vertex”. We have used the term “corner-vertex“ because it contains the most valuable information about the connectivity of each segment with its neighbors. Using the above mentioned definition of face types with label information of equation (6), we identify the type of every vertex. Also three types of faces on the labeled mesh are determined. Figure 4 shows the results of vertex and face type definition applied for Nefertiti model with four regions.

3. Surface fitting

In this section previously found corner vertices for each segmented region together with its site center are used to generate a polygonal patch. The sizes of these patches are adapted to segment details according to curvature based segmentation procedure described in section 2. These polygonal patches are used as building blocks to generate control meshes used for optimized SDM-based surface fitting in the next step. After surface fitting a regular submesh is generated for every segment in which distribution of vertices are trimmed to be adapted to segment details. It is because of feature-dependent patches have been defined for every segment. The detailed description is as follows:

3.1 polygonal patch conversion

In this section we introduce an algorithm to generate a polygonal patch for every segmented region in the original mesh. In section 2, a set of segments containing site center, labeled boundaries and corner vertices were generated. Our goal is to build a triangulated polygonal patch which will be subdivided later to build the control mesh in the next surface fitting step. In the proposed method a polygonal patch is generated by connecting succeeding neighbor corner vertices in one segment to create boundary edges. Then site center is connected to every corner vertex to complete polygonal patch conversion. The number of created corners in every patch depends on the number of pre-defined segments and geometric properties of the model and there is always at least three corner vertices. Also, patch center inside the polygon is the same as the position of site center. In order to maximize polygon regularity, we need to generate a hexagonal patch when the number of corner vertices is less than six. Corner adding algorithm is introduced in this step to achieve this task (see Figure 5).

To add corner vertices, a prioritizing procedure is defined here for boundary vertices of the segment. To generate non-planar hexagonal patch that best fits segment details, additional corner vertices must be selected from the segment boundary. Between every two neighboring corner vertices, there is a number of boundary vertices which are counted and indexed from 1 to k. Corner adding algorithm selects median index of k boundary vertices as the candidate vertex between two succeeding corners. It means that if k is odd, index is $(k+1)/2$ and if k is even, candidate vertex index is $(k/2)+1$. After determining candidate vertices, selection is done among them to add them to corner vertices. First selected candidate vertex has the maximum k and the next selection has the second maximum k and so on. The algorithm stops to add candidate corner vertex when the sum of corner vertices plus selected candidate vertices to be equal to six. The implementation result of the algorithm is shown in Figure 6.

3.2. SDM-based Subdivision

The basic idea of subdivision surface fitting is to construct a smooth model using iterative subdivisions of a coarse control mesh. Through different global methods of surface fitting, effectiveness of squared distance metric (SDM) is proved in [28]. Unfortunately, in global subdivision, the number of triangles increases exponentially at every subdivision level. For example in the regular loop scheme, the number of triangles is multiplied by 4 after one subdivision. Moreover, subdivision of the entire model creates oversampling in low detail areas of the mesh. We have used Loop subdivision for local surface fitting of submeshes. Independent subdivision of every sub segment instead of the whole mesh is faster and also reduces smoothing effect of subdivision and prevents oversampling in low detail areas of the mesh.

3.2.1. Algorithm flow

We have used modified version of subdivision surface fitting using SDM method in [29] for every segment without local refinement. Since, our method is intrinsically local by feature-aware segmentation and polygonal patch conversion there is no need to local refinement. Our proposed local subdivision surface fitting has the following main steps:

1. Normalization of the target segment by scaling all data points in the cube $[0,1]^3$.
2. Pre-computation of distance and curvature at all vertices of the original target segment.
3. Generate control mesh by Loop subdivision of the polygonal patch in section 3.1.
4. Calculate subdivision limit surface positions of the control mesh in step 3.
5. On the original target segment, find closest data point for every limit surface position in step 4 and generate linear combination of control points in step 3.
6. Adjust control points in step 3 by solving equation generated in step 5.
7. Compute error criteria, if acceptable go to step 8, otherwise go to step 4 with updated control mesh.
8. Get the subdivision surface from optimally updated control mesh and generate optimized submesh.

The results of implementation algorithm for one level of Loop subdivision is illustrated in Figure 7.

4. Stitching

After surface fitting for individual submeshes, we need to merge them together and build the entire semi-regular model. In this section we propose a new method for stitching optimized submesh boundaries to complete their connection with border triangles in neighboring submesh strips. During segmentation, all the vertices of the original mesh are included inside segments and only striped-connected faces are removed as shown in Figure (8-a). For this reason proposed stitching algorithm recovers only new faces in new strips using neighborhood indexing and corner information of the original mesh. It is done by creating duality between segment border information in the original mesh in Figure (8-a) and the optimized submesh in Figure (8-b) using SDM-based metric. Figure (8-c, 8-d) shows stitching steps for the entire mesh. For better illustration of the process, our stitching algorithm is implemented on a coarse mesh. Figure 8-a is the original labeled mesh with detected corner and border vertices. Red and black circles in Figure 8-b are dual corresponding corner and border vertices in Figure 8-a, respectively. For every corner vertex in the original segment, one and only one dual corner is defined in the optimized submesh. A dual corner on the optimized submesh is a boundary vertex on the submesh which has the minimum Euclidean distance from its corresponding dual corner vertex in the original segment. A sample strip is coded as $S_1S_2S_3S_4$ and is shown in Figure 8-c. After stitching all the coded strips of the mesh, oriented corner faces are created by three triangles which are illustrated with red circles on their vertices in Figure 8-d.

4.1. Stitching algorithm flow

After submesh optimization, corner and neighborhood information which is extracted in section 2.3, is transferred to optimized submesh using Euclidean distance metric. We indicate every corner vertex i in submesh k , with *Segment(k).Corner(i)*, *k=1 ,..., n_seg* and *i=1, ...,n_corner* where *n_seg* is the total number of segments and *n_corner* is the total number of corners in *Segment(k)*. Also, we define a structure for corners as follows (see Table 1):

- *coord* : is vertex coordinate of i^{th} corner in the k^{th} segment.
- *neighbor_ind* : is the index of neighborhood segments connected to i^{th} corner in the k^{th} segment.
- *neighbor_coord*: is the neighborhood vertex coordinates of i^{th} corner in the k^{th} segment.
- *Segment(k)*: is the k^{th} segment in the original model.
- *Segment_opt(k)*: is the k^{th} submesh in the optimized model.
- *Corner(i)* : is a structure relating three fields to its connected k^{th} segment.
- *Segment_opt(k).Border(j)* : is the coordinate of the j^{th} boundary vertex of the k^{th} submesh in the optimized model where *j=1, ..., n_border* and *n_border* is the total number of border vertices of the k^{th} submesh.

With the above notation, our proposed stitching steps are:

Step 1: For every corner vertex in segment *k*, find minimum distance of border vertices in the optimized submesh and mark it as the corner vertex in the optimized submesh:

$$\begin{aligned} &Segment\_opt(k).Corner(i).coord = \\ &\min_j \left(dist(Segment\_opt(k).Border(j).coord\ , Segment(k).Corner(i).coord) \right) \end{aligned} \tag{8}$$

Where *dist(A,B)* is the Euclidean distance between points *A* and *B*. These detected corners are illustrated with black and red circles in Figure (8-b).

Step 2: By tracing on the border of every submesh *k*, for every two succeeding corners on the submesh as shown with S_1S_2 in Figure (8-c) find their corresponding corners S_3S_4 on the neighbor submesh.

Step 3: In the created oriented strip $S_1S_2S3S_4$ find the length of $L_1=S_1S_2$ and $L_2=S_3S_4$ (for example $L_1=6$ vertices and $L_2=5$ vertices as shown in Figure 8-c), then calculate:

$$Q = floor\left(\frac{\max(L_1,L_2)}{\min(L_1,L_2)}\right) \tag{9}$$

$$R = \text{remainder}\left(\frac{\max(L_1,L_2)}{\min(L_1,L_2)}\right) \tag{10}$$

Where, $floor(\frac{A}{B})$ and $\text{remainder}(\frac{A}{B})$ are integer part and remainder of $\frac{A}{B}$ division result, respectively.

Step 4: Assuming $L_2 < L_1$, start from S_4 and create Q triangles whose heads are common with S4 and their bases are connected to Q succeeding edges starting from S_1. We call these triangles, "direct faces". Then move to the next vertex neighbor to the S_4 and create the same "direct faces" for that vertex till finishing in the last vertex, S_3. Here, "direct faces" have been created. Then build single triangular holes available between "direct faces", and we call them "between faces". Finally, create R triangles which have common head with S_3 and also have R neighbor bases starting from last "direct face" and ending in S_2.We call them, "remained faces". In case of $L_1 < L_2$, do the same procedure by replacing the role of L_1 with L_2.

Step 5: repeat step 4, in all the strips of the entire mesh.

Step 6: create corner faces as shown in Figure (9-d) with three red vertices.

Step 7: merge created faces of the strips with corner faces and submeshes together

All the process is illustrated in Figure 9.

5. Results

In this section, we present experimental results of our semi-regular remeshing framework to demonstrate the quality improvement and efficiency of the proposed method. All the experiments were achieved on a Notebook with 2.1 GHz Intel Core 2 Duo and 4GB of RAM. Algorithms have been implemented in MATLAB programming language installed in windows 7 operating system.

Many models have been tested that a number of them are demonstrated. We calculate performance of our semi-regular remeshing algorithm with two parameters including the number of Voronoi regions (segments), the number of iterations used for Lloyd's relaxation. The number of iterations used for Lloyd's relaxation is selected to be equal to 15. Since, we aim to generate a high quality semi-regular mesh with reduced number of vertices; only one level of subdivision is used for every segment to check the mesh regularity for all the models. In regular Loop scheme, the number of faces in the mesh is multiplied by 4 after every subdivision. The number of vertices in the semi-regular mesh is selected from 10 to 70 percent of vertices in the original model used in Table 2. It is controlled by selecting user defined number of site centers. The tolerance on vertex number depends on the resolution of the original mesh. There is another motivation to generate simpler meshes, too. It is because small meshes have lower triangle quality than larger ones. In practice, regularity gets worse during simplification. Experiments show that our method preserves regularity better than the other remeshing techniques even in the coarsened meshes.

Quality measurement of triangular mesh is traditionally achieved by measuring the geometric properties of the resulting triangles. We use the criteria in [4] to measure the semi-regular remeshing quality (see Table 2). The quality of triangle t *is* measured by $Q_t = \alpha \cdot \frac{R_t}{h_t}$ where R_t is the in-radius of t, h_t the longest edge length of t and α is the normalizing coefficient equal to $\frac{6}{\sqrt{3}}$. Q_{min} is the quality of a triangle and Q_{avg} is the average triangle quality. Q_{min} and Q_{avg} can be applied for the whole triangles of a 3D model. θ_{min} is the smallest angle of the minimal angles of all triangles and $\theta_{min,avg}$ is the average of the minimal angles of all triangles. Obviously, θ_{min} is between 0^0 and 60^0, anywhere . For a high quality mesh, the minimum of these values should not be less than 45^0 according to [38]. $\theta < 30^0$ is the percentage of triangles with its minimal angle smaller than 30^0. The error d is the Hausdorff distance between the original model and the remeshed model with respect to the bounding box diagonal [40].

In the experiments, our proposed method is tested for various models with different geometric properties. Comparison of our coarsened mesh with CVT-based vertex clustering [38] and QEM (Quadric Error Metrics) simplification [39] is shown quantitatively in Table 2. Also, results are illustrated qualitatively in Figure 10.

For precise investigation of generated semi-regular mesh, distribution of quality histogram for Table 2 is calculated. Visual results of semi-regular remeshing for CAD models (turbine and rocker-arm) and graphical models (dino and egea) are illustrated by triangle quality aspect ratio histogram in Figure 10 and results are compared with respect to other remeshing techniques in [38] and [39]. Regarding mesh quality with respect to average aspect ratio and average minimum triangle degree, compared with [38] and [39], our method shows better quality performance for all the models. Also, in Egea model about the same distortion error occurs compared with [38] and [39]. But in Dino, Rocker-arm and Turbine models we have increased distortion with better performance quality. Increased distortion error appears in the models with large curvature diversity in small areas, especially in CAD models. It is the reason that adapting curvature in remeshing of CAD surfaces is still a challenging problem [41]. Also, it should be noted that mesh coarsening automatically reduces mesh quality but histogram distribution in all the models of Figures 10 shows quality improvement regarding to aspect ratio distribution in the simplified meshes. In fact, distribution of aspect ratio for surface with higher quality triangles is closer to one in comparison to the surface with lower quality triangles. Numerical Examples in Table 2 show improvements in quality measure (increase of Q_{min} and Q_{avg} and decrease of $\angle 30^o$ percentage) with respect to [38] and [39].

We should compromise between triangle quality and Hausdorff error especially in high rate mesh coarsening. Mesh coarsening techniques mentioned in [38] and [39] are not subdivision oriented and do not generate semi-regular meshes. To illustrate performance of the proposed semi-regular remeshing more clearly, we compare our method with a related semi-regular remeshing algorithm called displaced subdivision surface in Ref [42]. Comparison results is listed in Table 2 and visually illustrated in Fig.11. Results show that our method not only preserves sharp features with greater high quality triangles but also has lower distortion error in graphical models of bunny, horse and dragon models in comparison to the method of [42] which is also a subdivision based method and generate semi-regular mesh. Our algorithm can produce semi-regular meshes adapted to the local curvature of the model through the proposed remeshing framework.

6. Conclusion

We have presented an effective framework to generate semi-regular meshes. Our semi-regular remeshing algorithm is based on segmentation of an input mesh with a curvature adapted CVT function. This algorithm depends on the concepts of segmentation, subdivision surface fitting and stitching. We stitch the global property in curvature-based density function of CVT with local subdivision to generate feature-adapted semi-regular meshes. Iterative process of optimized surface fitting leads to high quality semi-regular meshes with the same topology as the original ones. Future work will aim at remeshing models having small holes with respect to segment dimensions. As a post processing step, designing other feature dependant criteria during submesh optimization process will be considered to increase accuracy for coarsened CAD models.

7. References

[1] Kammoun A, Payan F, Antonini M. Sparsity-based optimization of two lifting-based wavelet transforms for semi-regular mesh compression. J Comput Graph 2012; 36: 272-282.

[2] Eck M,DeRose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution Analysis of Arbitrary Meshes. In Proc. ACM SIGGRAPH ; 1995; New York, ACM, NY, USA: ACM. pp. 173-182.

[3] Lee A W F, Sweldens W, Schroder P, Cowsar L, Dobkin D. MAPS: Multiresolution Adaptive Parameterization of Surfaces. In Proc. ACM SIGGRAPH ; 1998; New York, ACM, NY, USA: ACM. pp. 95-104.

[4] Valette S, Chassery J, Prost R. Generic remeshing of 3D triangular meshes with metric-dependent discrete voronoi diagrams. J IEEE Trans. Vis Comput Graph 2008; 14(2) March/April: 369-381.

[5] Chiang C, Jong B, Lin T, A robust feature-preserving semi-regular remeshing method for triangular meshes. J Visual Comput 2011; 27: 811-825.

[6] Ciarlet, P.G.: Basic error estimates for elliptic problems. In: Ciarlet, P.G., Lions, J.L. (eds.) Handbook of Numerical Analysis, vol. II, pp. 17–351. North-Holland, Amsterdam (1991). Finite Element Methods (Part 1)

[7] Hoppe H, Praun E. Shape compression using spherical geometry images. Math Visual 2005; 27-46. In: Dodgson N, Floater M, Sabin M, editors, Advances in Multi resolution for Geometric Modelling, Springer-Verlag, 2005, pp. 27-46.

[8] Yankui Sun , Yafeng Dong , Kaimin Maoa, Zesheng Tang, View-dependent progressive transmission and rendering for lunar model based on bicubic subdivision-surface wavelet, Advances in Space Research 53 (2014) 1848–1857.

[9] Gu X, Gortler SJ, Hoppe H. Geometry images. In Proc. ACM SIGGRAPH ; 2002; pp. 355-361.

[10] Loop C. Smooth subdivision surfaces based on triangles. MSc, Department of Mathematics, University of Utah, Utah, United States, 1987.