World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:6, 2016

3D Mesh Coarsening via Uniform Clustering

Shuhua Lai, Kairui Chen

Abstract—In this paper, we present a fast and efficient mesh coarsening algorithm for 3D triangular meshes. Theis approach can be applied to very complex 3D meshes of arbitrary topology and with millions of vertices. The algorithm is based on the clustering of the input mesh elements, which divides the faces of an input mesh into a given number of clusters for clustering purpose by approximating the Centroidal Voronoi Tessellation of the input mesh. Once a clustering is achieved, it provides us an efficient way to construct uniform tessellations, and therefore leads to good coarsening of polygonal meshes. With proliferation of 3D scanners, this coarsening algorithm is particularly useful for reverse engineering applications of 3D models, which in many cases are dense, non-uniform, irregular and arbitrary topology. Examples demonstrating effectiveness of the new algorithm are also included in the paper.

Keywords—Coarsening, mesh clustering, shape approximation, mesh simplification.

I. Introduction

3D mesh models are used in many important areas such as geometric modeling, computer animation, and CAD. With the availability of powerful laser scanners, large and dense meshes are easily acquired from physical world. However, since the full complexity of such models is not always required, coarsening a dense mesh, i.e., replacing the original mesh with a simpler but close enough mesh, is a necessary pre-processing step in many applications. Proposed in this paper is a novel surface mesh coarsening algorithm, which resamples the surface to a uniform mesh with many fewer elements than the original mesh. Our approach is based on a clustering of the original mesh cells, mimicking a Centroidal Voronoi Diagram (CVD) construction, which is theoretically the optimal strategy for resampling [2].

Given a 3D mesh, coarsening is to find a better discrete representation of the underlying surface. What is a good mesh so that the coarsening is of a better quality? In this paper, we present a coarsening algorithm for altering the mesh geometry and connectivity of a 3D model to improve its quality such that the resulting mesh meets the following requirements as much as possible:

1) All triangles are of similar shape (equilateral triangles)
2) All triangles are of similar size (equal edge lengths)
3) All vertices have valence close to 6 (uniformality)
4) Resulting mesh approximates the given mesh well (accuracy)

For a given number n, the new algorithm replaces an arbitrarily structured 3D mesh with a uniformly structured one, which has exactly n vertices. This is done through an elegant and efficient mesh clustering algorithm, which divides the faces of an input mesh into n clusters of similar size and shape. The n clusters then can be used for coarsening purpose by approximating the Centroidal Voronoi Tessellation of the input mesh.

II. Related Work

Coarsening is a key component of many geometric algorithms, including modeling, animation, visualization and simulation. As such, the rapidly developing field of geometry processing has produced a profusion of new Coarsening techniques over the past few years [8]-[13]. Coarsening a mesh consisting resampling the original surface with a lower number of vertices. Many mesh coarsening techniques have been presented, including the global optimization method [7], [5] and re-meshing for mesh coarsening [6], [3]. Mesh clustering is to partition the faces or vertices of the mesh into different regions. Generally, these regions are required to be non-overlapping and connected. One major application of the clustering technique is for mesh coarsening. Such a method builds the approximating mesh based on the clustering of the dense mesh. In mesh coarsening, clustering may not be explicitly required in a greedy clustering technique, like mesh decimation. A decimation method creates implicit partitions of the mesh through greedy and repeatedly collapsing mesh faces or vertices [11]. The resulting mesh is always sub-optimal [1]. The other clustering method for mesh coarsening is to construct the mesh clusters explicitly. The new mesh clustering technique, also designed for mesh coarsening, falls into this category.

III. Methodology

A. Constrained Voronoi Tessellation

Voronoi tessellation are essential structures in computational geometry and have been widely used in many important computer graphics applications [9]. Given a domain Ω in R^3 and a set of points $\{z_i\}_{i=1}^k$, the corresponding Voronoi tessellation $\{V_i\}_{i=1}^k$ is a partition of Ω such that:

(1) $V_i \bigcap V_j = \emptyset$ and $\bigcup_{i=1}^k V_i = \Omega$, and

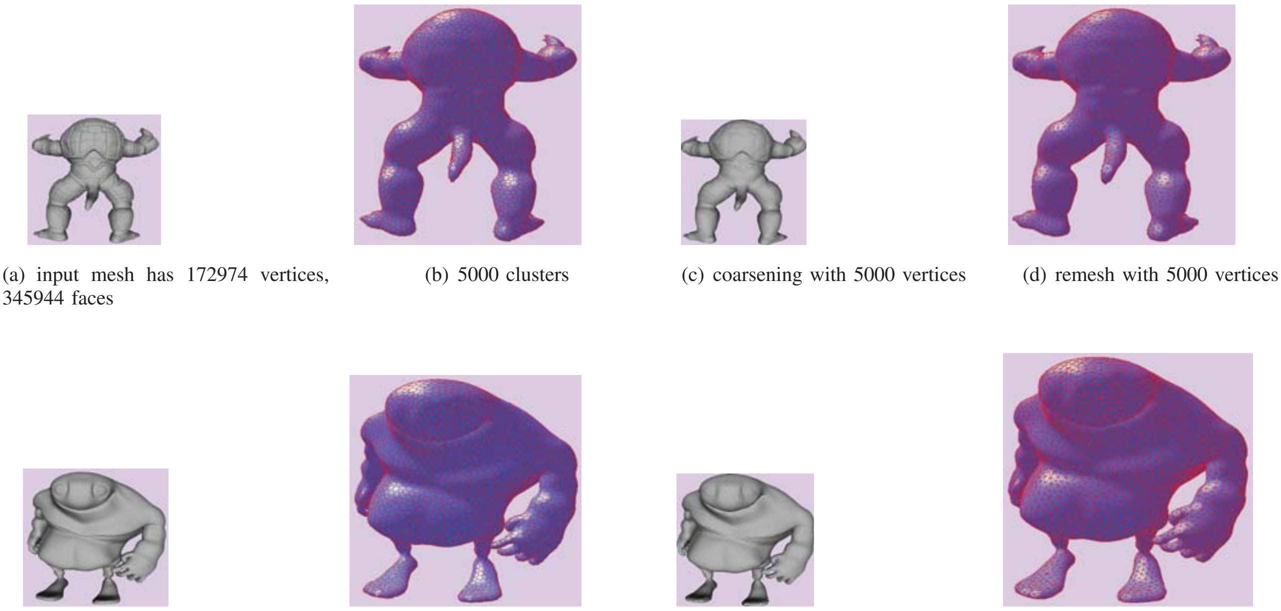
(2) $V_i = \{x \in \Omega || x - z_i| < |x - z_j| \text{ for } j = 1, 2, .., k, j \neq i\}$

$\{z_i\}_{i=1}^k$ are called the generator of V_i and $\{V_i\}_{i=1}^k$ the Voronoi cluster. Centroidal Voronoi Tessellation (CVT) is an extension of Voronoi Tessellation by requiring that the generators are also the mass centroids of the Voronoi regions. Given a density function $\rho(x)$ on Ω , CVT of Ω is a minimizer of the energy functional

$$F(z) = \sum_{i=1}^{n} \int_{V_i} \rho(x)|x - z_i|^2 dx \qquad (1)$$

where $z_i \in \Omega$. Constrained Centroidal Voronoi Tessellation [7] is the restriction of CVT to a given surface.

S. Lai and K. Chen are with the School of Science and Technology, Georgia Gwinnett College, Lawrenceville, Georgia 30043, USA (e-mail: slai, kchen@ggc.edu).



(a) input mesh has 172974 vertices, 345944 faces

(b) 5000 clusters

(c) coarsening with 5000 vertices

(d) remesh with 5000 vertices

(e) input mesh has 23202 vertices, 46400 faces

(f) 5000 clusters

(g) coarsening with 5000 vertices

(h) remesh with 5000 vertices

Fig. 1 Test examples - 1

B. Discrete Constrained Centroidal Voronoi Tessellation on a Polygonal Mesh

Given a polygonal mesh M and a cluster number n, we will try to divide the faces of M into n connected sets of faces $V_i(i = 1, 2, ..., n)$ by constructing a CCVT on M. These clusters $\{V_i\}$ form a discrete CCVT on the mesh M. Although discrete CCVT can be defined for any polygonal mesh, we will concentrate on triangular meshes in this paper. In the continuous setting, CCVT is the minimizer of an energy functional similar to the one defined in (1). For the discrete version of CCVT on a triangular mesh M, the region V_i is a connected collection of triangles. Assuming $\rho(x) = 1$, we can rewrite the energy functional as

$$F(z) = \sum_{i=1}^{n}(\sum_{T_k\in V_i}\int_{T_k}|x - z_i|^2 dx)$$

where T_k is a triangle in V_i. In fact, the following equation holds

$$\int_{T_k}|x - z_i|^2 dx = |x_k - z_i|^2|T_k| + \delta_k$$

where $|T_k|$ is the area of triangle T_k with vertices $x_j^k(j = 1, 2, 3)$ and x_k is the centroid of T_k. Note that δ_k is a constant for each triangle T_k, hence can be ignored when calculating the energy function. As a result, we have

$$F(z) = \sum_{i=1}^{n}(\sum_{T_k\in V_i}|x_k - z_i|^2|T_k|). \tag{2}$$

The constrained mass centroid centroid z_i of cluster V_i on a continuous surface S is defined as a solution to the minimization problem defined in (2) with V replaced with V_i. For discrete CCVT on M, we can use the same argument as in reformulating $F(z)$ to rewrite the minimization problem as:

$$\min_{z\in M}(\sum_{T_k\in V_i}|x_k - \bar{z}_i|^2|T_k| + \sum_{T_k\in V_i}|\bar{z}_i - z|^2|T_k|)$$

where $\bar{z}_i$ is the mass centroid of cluster V_i of mesh M. Thus, the constrained mass centroid of V_i is the point on M that is closest to its mass centroid $\bar{z}_i$. Before we describe our clustering algorithm, two important properties have to be highlighted first.

Property 1. Let $\{(V_i, z_i)\}$ be the current cluster configuration where z_i is the constrained mass centroid of V_i, and for each triangle $T_r \in V_i$, let x_r be its centroid. If $|x_k - z_q|^2 < |x_k - z_p|^2$ for some triangle $T_k \in V_p$ and V_q adjacent to V_p, then $F(z') < F(z)$, where $z\prime$ is the constrained mass center of V_i' and

$$V_i' = \begin{cases} V_i & i \neq p, q \\ V_p - T_k & i = p \\ V_q \bigcup \{T_k\} & i = q \end{cases}$$

Property 2. Let $\{(Vi, zi)\}$ be the current cluster configuration, and triangles $T_k \in V_p$ and $T_s \in V_q$ with centroids x_k and x_s, respectively, share a common edge. If $|x_k - z_p|^2 > |x_k - z_q|^2$, $|x_s - z_q|^2 > |x_s - z_p|^2$, and $(|x_k - z_p|^2 - |x_k - z_q|^2)|T_k| < (|x_s - z_q|^2 - |x_s - z_p|^2)|T_s|$ are all satisfied, then we have $F'(z) < F(z)$, where $F'(z)$ is defined in (7).

C. Energy Minimization

Recall that a discrete CCVT of a triangular mesh M is a minimizer of the discrete energy functional equation (2). In

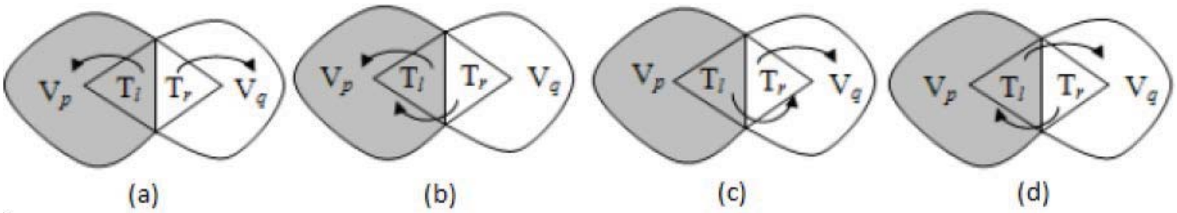


Fig. 2 Illustration of 4 cases in distance comparison. The presence of an arrow indicates direction of the swapping after the comparison. Figs. 2 (a), (b), (c) and (d) correspond to cases 1, 2, 3 and 4 respectively

the following we present an algorithm to iteratively reduce the value of $F(z)$ until a limit point is reached. The main idea of the algorithm is to update the clusters by comparing distances from triangle centroids of a cluster to mass centroids of adjacent clusters. The triangles that have to be considered are just boundary triangles, i.e., triangles sharing a common cluster edge. A mesh edge is called a cluster edge if it is shared by two triangle faces of different clusters. The distance comparing procedure is stated below. Let edge e_{lr} be a cluster edge in the current cluster configuration $\{(V_i, z_i)\}$. e_{lr} is shared by triangles T_l and T_r, where $T_l \in V_p$ and $T_r \in V_q$ are in different clusters. Let x_l and x_r be the centroids of T_l and T_r, respectively. Denote $|x_l - z_p|^2$, $|x_l - z_q|^2$, $|x_r - z_p|^2$ and $|x_r - z_q|^2$ with d_{lp}, d_{lq}, d_{rp} and d_{rq}, respectively. We need to compare d_{lp} with d_{lq}, and d_{rp} with d_{rq}, totally in four possible cases. Fig. 3 illustrates these 4 cases:

1. $d_{lp} \leq d_{lq}$ and $d_{rp} \geq d_{rq}$. Do nothing. This is exactly what the convergent state should be.

2. $d_{lp} \leq d_{lq}$ and $d_{rp} < d_{rq}$. Move T_r to V_p. According to property 1, this movement lowers the value of the energy functional $F(z)$.

3. $d_{lp} > d_{lq}$ and $d_{rp} \geq d_{rq}$. Move T_l to V_q. The new value of the energy functional $F(z)$ will be lower, according to property 1.

4. $d_{lp} > d_{lq}$ and $d_{rp} < d_{rq}$. In this case, one more test is needed to decide which triangle should be moved.

- If $d_{lp}|T_l| + d_{rp}|T_r| < d_{lq}|T_l| + d_{rq}|T_r|$, move T_r to V_p,
- Otherwise, move T_l to V_q.

In case 4, the value of the energy functional $F(z)$ will be lower after the movement, according to property 2. Based on this distance comparison process for a single cluster edge, we can easily derive an algorithm which updates the mass centroids of the clusters immediately after finishing the above comparison process for each cluster edge. This algorithm should work because the energy functional decreases after the distance comparison process for each cluster edge. With a valid initial cluster configuration, we can perform distance comparison for each cluster edge and record the centroid variations of adjacent clusters at the same time. After completing the distance comparison process for all cluster edges, we update the mass centroids of clusters and update the cluster edge set. This process is iterated until the cluster edge set no longer changes. It is obvious that the energy functional $F(z)$ has a global minimum on the triangular mesh M. As $F(z)$ decreases strictly after each configuration-wise updating, it is guaranteed to converge to a limit point.

D. Coarsening

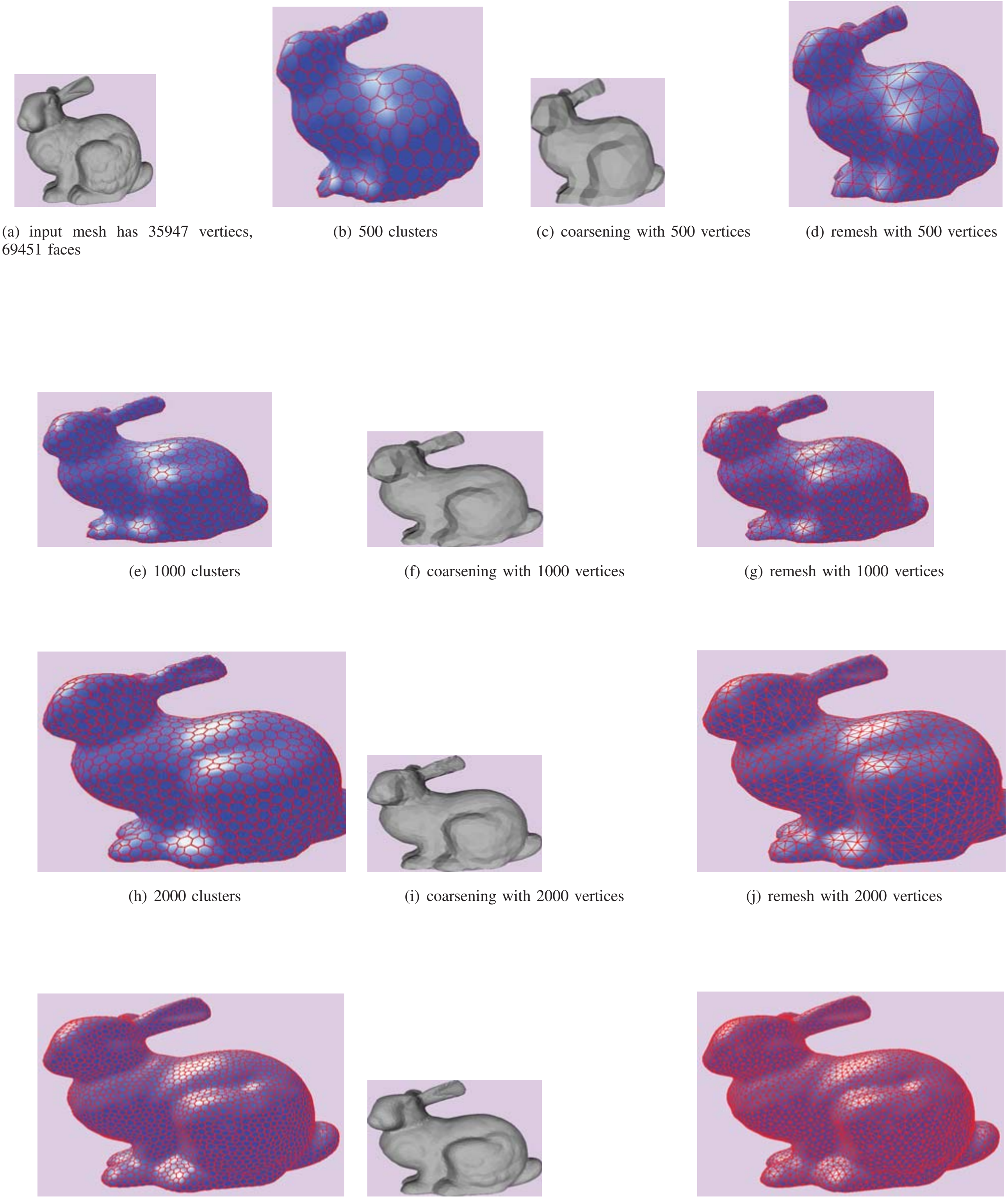
For any given mesh M, and any given number n, we first apply the above CCVT clustering algorithm to obtain n clusters.

Once clustering of a mesh M by approximating CCVT is done, the following work of getting an approximately uniform coarsening triangular mesh is relatively simple. The process is like a Delaunay triangulation, with each cluster treated as a logical vertex. The process is illustrated below. First, a vertex is created for each cluster. Several techniques can be used here. One approach is, for each cluster, take the vertex that is closest to its mass centroid [6]. A second choice is to use the Quadric Error Metric to compute a vertex [8]. The first approach is used here since it conforms more with the spirit of CCVT on a mesh. The second step is to triangulate the vertices created in the above step. Delaunay triangulation is used here. Two vertices are connected with an edge in the new mesh if the corresponding clusters of these vertices are adjacent to each other. Thus a vertex point that is shared by three clusters corresponds to a triangle in the new mesh. Degenerate cases, however, would arise if a vertex point is shared by more than 3 clusters. The solution to such degenerate cases is quite simple. If a vertex point is shared by $m \geq 4$ clusters, then there would be an m-side polygon in the new mesh corresponding to this vertex point. We simply triangulate this polygon to get $m - 2$ triangles. Hence the output remesh is always a triangular mesh. Because CCVT guarantees each cluster has similar size and shape, the resulting triangles in the remesh would be approximately uniformly similar as well. Experimental Results are discussed in the next section.

IV. Experimental Results

The algorithms presented in this paper has been implemented. Test results are given in Figs. 1 and 3. As we can see, for any given mesh, our algorithm can coarsen to any given number of vertices using our clustering algorithm. Most of the triangles in the new mesh are close to equilaterals and most of the vertices have a valance of 6. Hence the remesh is almost uniform everywhere.

Our iterative algorithm always begins with a valid initial cluster configuration, i.e., the clusters are connected and non-overlapping. A good initialization can reduce the clustering time significantly. We apply the hierarchical face clustering idea in [4] to design our cluster initialization. Hierarchical face clustering respects the connected requirement of clusters strictly.



(a) input mesh has 35947 vertiecs, 69451 faces

(b) 500 clusters

(c) coarsening with 500 vertices

(d) remesh with 500 vertices

(e) 1000 clusters

(f) coarsening with 1000 vertices

(g) remesh with 1000 vertices

(h) 2000 clusters

(i) coarsening with 2000 vertices

(j) remesh with 2000 vertices

(k) 5000 clusters

(l) coarsening with 5000 vertices

(m) remesh with 5000 vertices

Fig. 3 Test examples - 2

V. Conclusion

In this paper we propose a novel clustering algorithm for coarsening an input polygonal mesh M by approximating CCVT on M. The new coarsening algorithm can alter the mesh geometry and connectivity of a 3D model to improve its mesh quality from the perspective of mesh uniformality. The new coarsening algorithm replaces an irregularly structured 3D mesh with a uniformly structured one, which has given number of vertices, meanwhile approximates the original input mesh well.

References

[1] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Levy, and Mathieu Desbrun. Anisotropic polygonal remeshing. ACM.Transactions on Graphics, 22:485493, 2003.

[2] Pierre Alliez, Eric Colin de Verdiere, Olivier Devillers, and Martin Isenburg. Isotropic surface remeshing. In M.S. Kim, editor, SMI03: Proceedings of Shape Modeling International 2003, pages 4958, Los Alamitos, 2003. IEEE Computer Society.

[3] P. Alliez, M. Meyer, and M. Desbrun. Interactive geometry remeshing. Acm Transactions on Graphics, 21(3):347354, 2002.

[4] Mario Botsch and Leif Kobbelt. A remeshing approach to multiresolution modeling. In R. Scopigno and D. Zorin, editors, Proceedings of 2nd Eurographics Symposium on Geometry Processing, pages 189196. Eurographics, 2004.

[5] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3d models. Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings, 2004.

[6] L. Kobbelt, J. Vorsatz, U. Labsik, and H.P. Seidel. A shrink wrapping approach to remeshing polygonal surfaces. Computer Graphics Forum, 18:119130, 1999.

[7] Emil Praun and Hugues Hoppe. Spherical parametrization and remeshing. Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings, pages 340349, 2003.

[8] F. Fan, S. Lai, J.Wang, C. Huang, Y. Li, Mesh Clustering by Approximating Centroidal Voronoi Tessellation. Proc. 2009 SIAM/ACM Joint Conference on Geometric & Physical Modeling, 2009, pp. 301-306.

[9] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. SIAM Rev., 41(4):637676, 1999.

[10] Q. Du, M. D. Gunzburger, and L. Ju. Constrained centroidal voronoi tessellations for surfaces. SIAM J. Sci. Comput., 24(5):14881506, 2002.

[11] A. W. M. Garland and P. S. Heckberty. Hierarchical face clustering on polygonal surfaces. In Proceedings of the Symposium on Interactive 3D Graphics, 2001.

[12] S. Valette and J.-M. Chassery. Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. 23(3):381389, 2004. (Proc. Eurographics04).

[13] S. Valette, J.-M. Chassery, and R. Prost. Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams. IEEE Transactions on Visualization and Computer Graphics, 10(2):369381, 2008.