Caiyun Yang
Institute of Automation,
Chinese Academy of Sciences,
Beijing 100864, China
e-mail: caiyun.yang@ia.ac.cn

Yutaka Ohtake
RCAST (Research Center for Advanced
Science and Technology),
The University of Tokyo,
Tokyo 13-8654, Japan
e-mail: yu-ohtake@den.rcast.u-tokyo.ac.jp

Masaki Moriguchi
Computer Science,
Chuo University,
Tokyo 112-0003, Japan
e-mail: moriguchi@ise.chuo-u.ac.jp

Hiromasa Suzuki
RCAST (Research Center for Advanced
Science and Technology),
The University of Tokyo,
Tokyo 13-8654, Japan
e-mail: suzuki@den.rcast.u-tokyo.ac.jp

Generation of Segmented Triangular Meshes From CT Images Based on Centroidal Voronoi Tessellation and the Graph Cut Method

Mesh generation from X-ray computed tomography (CT) images of mechanical parts is an important consideration in industrial application, and boundary surface meshes in multimaterial parts can be extracted by generating segmented meshes from segmented images. In this paper, the authors outline a new approach for achieving segmented mesh generation. The image is first subjected to centroidal Voronoi tessellation and Delaunay tessellation steered by a density map to create a triangular mesh while maintaining discontinuities between materials. Given an input domain and a number of initial sites, the energy function is minimized automatically by iteratively updating the Voronoi tessellation and relocating sites to produce optimized domain discretization and form the mesh. Thus, the mesh in question is effectively and quickly segmented into different parts via this new graph cut method. The proposed approach is considered more efficient because there are fewer triangles than pixels, which reduces computation time and memory usage.
[DOI: 10.1115/1.4026292]

Keywords: Mesh generation, centroidal Voronoi tessellation, mesh segmentation, graph cut

1 Introduction

X-ray CT scanning is commonly adopted in industry to detect internal voids and cavities in manufactured parts. Thanks to recent advances in the field, the technique can now be used to generate object shapes with sufficient accuracy for industrial application [1]. Against this background, there is a need for a method to generate models of shapes such as boundary surface meshes and finite element meshes to support engineering simulation.

CT scanners output images of objects with pixels, whose CT values are roughly proportional to the actual density of the materials they represent. The top-left image in Fig. 1 shows a typical example of such a CT image with a number of segments indicating different materials. The pixels in each segment have a uniform gray color.

Surface meshes for single-material objects can be extracted using an iso-surfacing method such as Marching Cubes. However, pixels in multimaterial objects need to be classified into material segments, and boundary meshes between different materials must be generated. This process is known as *segmentation* in the field of image processing. Thus, a common way of generating meshes involves the two steps of image segmentation and iso-surfacing [2].

In the novel approach proposed here, the image is first subjected to centroidal Voronoi tessellation (CVT) [3] (see Fig. 1) and Delaunay tessellation to create a mesh of triangles, which is then segmented into different components representing different materials using the graph cut method [4]. Although this method provides powerful segmentation performance, it is time-consuming (approximately $O(n^2 \log n)$ where n is the number of elements such as pixels) and memory-intensive. With this in mind, the graph cut approach is applied not to the image itself but to the tessellated triangles because there are significantly fewer triangles than pixels in the image. The technique is also expected to be beneficial for extending the application of the proposed approach to 3D volumetric CT images in the future.

Based on the proposed method, a nonsegmented CT image with gray values is transformed into a mesh with discontinuities between different materials preserved using CVT. For this purpose, an energy function reflecting the image gradient is introduced. The image in the bottom right of Fig. 1 shows small triangles generated along the material boundaries. The graph cut method is subsequently applied to the mesh to segment the triangles into different components representing different materials with reference to the gray values of the triangles.

2 Paper Overview

In the rest of this paper, Sec. 2 outlines related work, Sec. 3 introduces CVT, Sec. 4 presents the CVT-based mesh generation method, Sec. 5 highlights the new graph cut approach developed to segment generated meshes, Sec. 6 details the results of the study's experiments, and Sec. 7 concludes the work.

3 Related Work

3.1 Mesh Generation. Meshes are extensively adopted to provide solutions to various application problems. Due to the wide range of mesh generation approaches proposed in recent decades, an exhaustive review of all such methods is outside the scope of this paper. In general meshing, Delaunay-based techniques [5–11] are quite popular. For a given set of sample points in a 2D domain, CVT can be used to generate mass center points of corresponding Voronoi regions with respect to a given density

Contributed by the Computers and Information Division of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received August 8, 2013; final manuscript received December 9, 2013; published online January 29, 2014. Assoc. Editor: Xiaoping Qian.

function globally [12]. The dual structure of CVT is the corresponding Delaunay triangulation, which has the canonical property of maximizing the minimum angle of the minimum containment sphere. The quality of meshes created using Delaunay triangulation is often closely related to the point placement generated in CVT, although other factors need to be addressed to produce good meshes. In Ref. [13], CVT was used to optimize the distribution of generated points so that high-quality meshes were achieved. Thus, CVT can be applied to produce good meshes that satisfy the desired specifications in terms of considerations such as local refinement and the sizing field. In other words, it can be used in the research discussed here to provide meshes with edges maintained between different materials in image data. The authors of Ref. [14] discussed segmented mesh generation from an image with presegmentation. In addition, Ref. [15] proposed mesh images labeled ahead of time. Our method is different from them in Refs. [14,15] in that our inputs are "unsegmented" images, which are first subject to centroidal Voronoi tessellation and Delaunay tessellation determined by gray values to generate unsegmented meshes with discontinuities between different materials. Those unsegmented triangular meshes with gray values are then segmented into different parts using a new graph cut.

3.2 Segmentation. Segmentation involves dividing an input model into a set of meaningful sections, each of which is smaller and simpler than the overall object. Although a variety of algorithms are used for this purpose, a detailed discussion of related literature is outside the scope of this paper, which therefore simply outlines a number of segmentation methods related to the graph cut approach. Combinatorial min-cut algorithms on graphs are widely applied as a useful optimization tool in medical imaging, computer vision and computer graphics [16–18]. Graph cut methods can provide piecewise smoothness for graphs while maintaining relevant sharp discontinuities [18]. The study outlined in Ref. [19] used the graph cut method to segment an image into target and other parts, including the background. The authors of Ref. [20] converted a triangular mesh into a tetrahedral mesh and then segmented it into two components iteratively until no further segmentation was possible. In the final version, each mesh was successfully segmented into functional parts.

4 Centroidal Voronoi Tessellation (CVT)

CVT is a particular type of Voronoi tessellation in which each site becomes the mass center (centroid) of the Voronoi region associated with it [12]. Recently, this approach and a wide range of related applications in computational science engineering have been studied [3]. A brief outline of the basic CVT concept follows.

Given a set of sites (also called generators) $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ belonging to a domain $\Omega \subset R^N$, the Voronoi region Ω_i of site $\boldsymbol{x}_i$ is defined as the set of points whose closest site is $\boldsymbol{x}_i$:

$$\Omega_i = \{\boldsymbol{x} \in \boldsymbol{R}^N, \|\boldsymbol{x} - \boldsymbol{x}_i\| \leq \|\boldsymbol{x} - \boldsymbol{x}_j\|, \quad \forall \boldsymbol{j} \neq \boldsymbol{i}\}. \tag{1}$$

Voronoi tessellation (or a Voronoi diagram) is a set of Voronoi regions $\{\Omega_i\}_{i=1}^{n}$ and involves the decomposition of Ω into n regions. The dual tessellation of the Voronoi diagram generates Delaunay triangulation, and especially, the dual representation of CVT can be used for high-quality mesh generation.

In CVT, each site coincides with the centroid of the Voronoi region associated with it. The mass center $\boldsymbol{x}_i^*$ of the Voronoi region Ω_i is defined as

$$\boldsymbol{x}_i^* = \frac{\int_{\Omega_i} \boldsymbol{x} \cdot \boldsymbol{\rho}(\boldsymbol{x}) d\boldsymbol{\sigma}}{\int_{\Omega_i} \boldsymbol{\rho}(\boldsymbol{x}) d\boldsymbol{\sigma}} \tag{2}$$

where $\rho(\boldsymbol{x})$ is a mass density function defined over Ω. CVT can be computed by considering a particular energy function. Given any set of sites $\{\boldsymbol{x}_i\}_{i=1}^{n}$, this function is defined by

$$\boldsymbol{F}(\{\boldsymbol{x}_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \left(\int_{\Omega_i} \boldsymbol{\rho}(\boldsymbol{x}) \|\boldsymbol{x} - \boldsymbol{x}_i\|^2 d\boldsymbol{\sigma} \right) \tag{3}$$

where Ω_i is the Voronoi region of site $\boldsymbol{x}_i$. CVT can then be applied by minimizing this function, as the tessellation is a local optimal solution of the function [12]. For example, in the Lloyd relaxation method, CVT is applied by alternating the computation of Voronoi tessellation and the optimization of site positions.

For 2D images consisting of pixels, Voronoi tessellation can also be defined in a similar way. In such images, Voronoi regions $\{\Omega_i\}$ are defined as sets of pixels, and the distance to a pixel is measured as the distance to its center. Each site is located in a pixel.

5 CVT-Based Mesh Generation

The triangles in the resulting meshes are smaller in regions rich in details, and their edges need to respect with the discontinuities of the underlying images. CVT is a good approximation of the different segments representing the materials of the original image data. Second, a centroidal distribution of sites is useful as the sites are well-spaced, our density function steered regions with higher values of ρ more sites than regions with lower values of ρ. It is more precise than triangular meshes achieved by algorithms without any relationship to the underlying images. Third, for the centroidal Voronoi regions, it is simpler to implement some integral computations required by our method. At last, it is easy to constrain the corners of the original data to be vertices in the

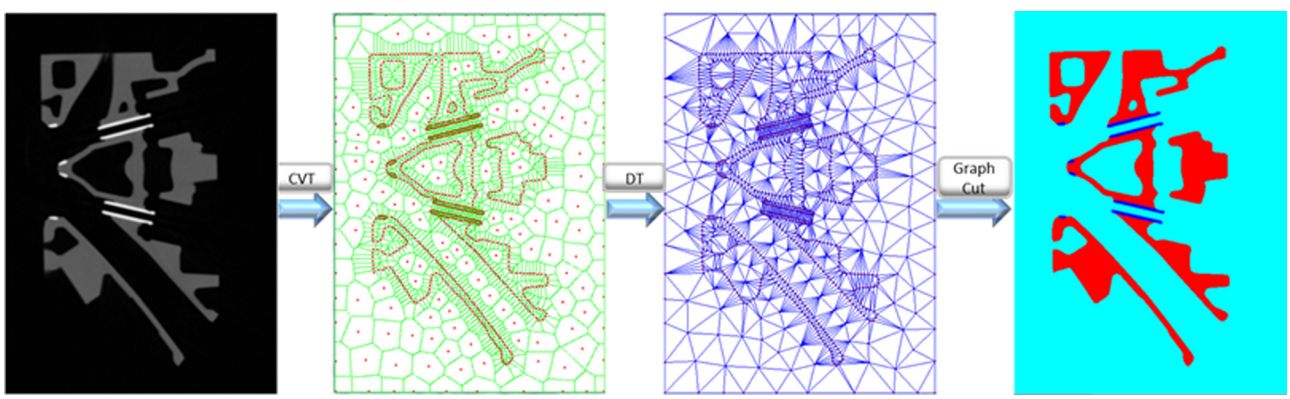

Fig. 1 Outline of the proposed method. CVT, centroidal Voronoi tessellation; DT, Delaunay triangulation

CVT, which ensures they will be kept in the subsequent segmentation.

This section outlines an efficient technique based on CVT for the generation of triangular meshes from images. The steps of the proposed approach can be summarized as follows:

(1) Input the image Ω
(2) Compute the mass density function
(3) Construct initial sites $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ inside Ω
(4) Minimize the energy to implement CVT
(5) Generate the triangular mesh based on CVT

5.1 Mass Density Function. In this research, the input CT image Ω is of a connected compact set with gray values that determines the image weight as well as the density function.

For the purpose of preserving discontinuities between different components in an image on the edges of the resulting triangular meshes, the density function $\rho(\boldsymbol{x})$ is constructed to enable focus on considerably more Voronoi region sites in discontinuity positions, while the sites are updated. To this end, the proposed density function should have a higher value near boundaries and a lower value in other regions. In this study, different density function methods were experimented with, including those involving the Sobel weight and the gradient magnitude. It was found that gradient magnitude consideration results in appropriate adaptive meshes with smaller triangles in higher-gradient regions and larger ones in lower-gradient regions (Fig. 2).

Therefore, CVT steered by the density map can approximate well the different regions of the underlying image data [3,12] and guarantee the accuracy, the boundaries in the original images are aligned with the edges of the obtained triangular meshes.

5.2 Initial Site Sampling. Good initialization of site points for CVT is desirable due to the potential complexity of the input domain and optimized CVT in terms of geometry and topology. To achieve this, Ward's hierarchical clustering method [21] is used for the initial site sampling. This method involves greedy iterative merging of clusters in which the pair of clusters with the minimum energy increase is merged at each step. Energy is measured as the weighted sum of the squared distances between data and cluster centers. Here, the weight is the density of each element proposed above. As this energy is the same as that used in CVT, the adoption of Ward's method can be expected to create high-quality initial samples. Although the initialization method is slower than that with random sampling, it results in the generation of high-quality initial sample points.

5.3 Energy Minimization. For the images used in this research, the energy function of CVT can be defined as follows:

$$F = \sum_{i=1}^{n}\left(\sum_{u_j\in\Omega_i}\int_{u_j}\rho(\boldsymbol{x})\|\boldsymbol{x}-\boldsymbol{x}_i\|^2 d\sigma\right) = \sum_{i=1}^{n}\left[\sum_{u_j\in\Omega_i}\boldsymbol{\rho_j}(\boldsymbol{x_i}-\boldsymbol{c_j})^T(\boldsymbol{x_i}-\boldsymbol{c_j})\right] \tag{4}$$

where $\rho_j = \int_{u_j}\rho(\boldsymbol{x})d\sigma$, and $\boldsymbol{c}_j = (1/\rho_j)\int_{u_j}\boldsymbol{x}\cdot\rho(\boldsymbol{x})d\sigma$ u_j is a pixel in the input image, and $\boldsymbol{x}_i$ is the site of a Voronoi region Ω_i. To simplify the density function computation for each element, the gray value on one element is assumed to be uniform, as this does not affect the results.

The sites of the Voronoi diagram are updated to give their optimal positions with the following three steps:

(1) If no boundary pixels of the domain are included in a Voronoi region Ω_i, the site $\boldsymbol{x}_i$ in the region Ω_i is defined as the region barycenter

$$\boldsymbol{x_i} = \frac{\sum_{\Omega_i}\boldsymbol{\rho_j}\cdot\boldsymbol{c_j}}{\sum_{\Omega_i}\boldsymbol{\rho_j}} \tag{5}$$

(2) Otherwise, if a corner pixel of an image belongs to a Voronoi region Ω_i, the site $\boldsymbol{x}_i$ is constrained to the center of the pixel.
(3) Otherwise, if the boundary curve passes through a Voronoi region Ω_i, the site $\boldsymbol{x}_i$ is set as the barycenter of the boundary curve $\partial\Omega_i$.

$$\boldsymbol{x}_i = \frac{\sum_{\partial\Omega}\rho_j\cdot c_j}{\sum_{\partial\Omega}\rho_j} \tag{6}$$

Energy is minimized by alternately implementing CVT from the current site positions and moving each Voronoi region site to its optimal placement until the convergence is reached.

5.4 Mesh Generation. In this method, a dual representation of the final CVT diagram is used to generate the corresponding triangular mesh. The Delaunay tessellation provided in CGAL [22] is applied to the set of sites, and the mesh produced represents material boundaries accurately. In Fig. 2(*a*) shows the original 2D image. The density map in (b) is constructed by computing the gradient of each pixel. The energy function is consistently minimized for repeatedly updating of the Voronoi diagram and the

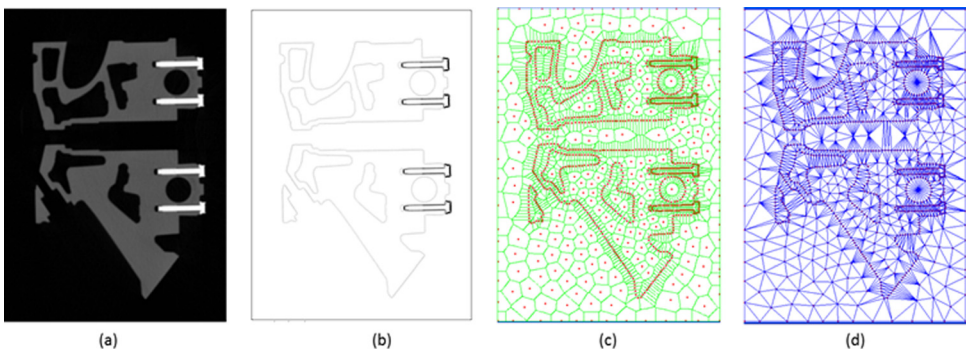

Fig. 2 Mesh generation based on CVT (*a*): the input image with a gray value (500*700 pixels) (*b*): the corresponding density map (*c*): the final CVT diagram with 1200 sites (*d*): the corresponding triangular mesh associated with the final CVT diagram

optimization of its sites, resulting in the final CVT diagram shown in (c). The dual representation of the CVT is used to define the triangular mesh in (d).

6 Graph-Cut-Based Mesh Segmentation

This section outlines a method of segmenting the triangular mesh obtained from an image depending on the material in question using the method presented in the previous section. For instance, there are three materials in Fig. 2(*a*): the background (air) and two metals. Each triangle is classified as representing one of these materials.

To achieve this segmentation, the graph-cut-based method presented in Ref. [17] is extended using a triangular mesh M with its underlying image as input. To simplify the computation of the gray value $f(t_i)$ for all triangles t_i, $f(t_i)$ is defined as the average gray value of all pixels in the triangle t_i. The pixels on the edges of a triangle are subdivided to compute a precise average gray value for the triangle.

As graph cut involves binary segmentation, a recursion strategy is used for triangular mesh segmentation. A triangular mesh is taken as a component. If the difference between the highest and lowest gray values in the component is above a certain threshold (which depends on the image), the component will be segmented into two smaller ones via the proposed graph cut technique. The process is repeated until the difference between the highest and lowest gray values in each component is below the threshold.

6.1 Graph Cut Technique. A triangular mesh $M = \{t_i\}$ can be segmented into two sub triangular meshes using the graph cut method. For this purpose, a graph $G = (V, \Xi)$ is constructed whose node set V includes all triangles $\{t_i\}_{i=0}^{n}$ in the triangular mesh M. These triangles are initially classified into three sets by their gray values. In the case of Fig. 2, they are classified into a set V_0 of those in the background and a set V_1 of those in the metals. The third set V_f contains all other triangles that cannot be easily classified

Two triangles s_0 and s_1, which, respectively, have the lowest and highest gray values in V, are selected. V_0, V_1, and V_f, which are subsets of V, are then defined: $V_0 = \{v \in V | |f(v) - f(s_0)| \leq \theta\}$

$$V_1 = \{v \in V |\ |f(v) - f(s_1)| \leq \theta\} \quad \text{and}$$

$$V_f = V - (V_0 \cup V_1)$$

where θ is a user-defined threshold value. The user can also directly select s_0 and s_1. The two triangles s_0 and s_1 are denoted as the source node and the sink node. The undirected edges Ξ of the graph G consist of two kinds of edges. One set contains edges defined between two neighboring triangles in M, and the other contains those between each triangle and s_0 and s_1.

The goal of triangular mesh segmentation is to give a binary label $\Gamma_m \in \{0, 1\}$to each triangle t_m in V_f. Thus, $\Gamma = \{\Gamma_m | t_m \in V_f\}$ represents a binary labeling of the triangular mesh M. If $\Gamma_m = 0$, t_m is added to V_0, and if $\Gamma_m = 1$, t_m is added to V_1. As a consequence, any one triangle t_m in V_f can be classified into either of V_0 and V_1. Accordingly, the triangular mesh M becomes segmented into two sub meshes.

The labeling Γ for the triangles of V_f can be made by minimizing the following energy function [17]

$$E(\Gamma) = \sum_{t_m \in V} E_1(\Gamma_m) + \lambda \sum_{t_m, t_n \in \Xi} E_2(\Gamma_m, \Gamma_n) \tag{6}$$

which includes the following cost functions for the edges of E_1:

$$E_1(\Gamma_m = 0) = \begin{cases} \infty & t_m \in V_0 \\ 0 & t_m \in V_1 \\ 1 - \dfrac{|f(t_m) - f(s_0)|}{|f(t_m) - f(s_0)| + |f(t_m) - f(s_1)|} & t_m \in V_f \end{cases} \tag{7}$$

$$E_1(\Gamma_m = 1) = \begin{cases} 0 & t_m \in V_0 \\ \infty & t_m \in V_1 \\ 1 - \dfrac{|f(t_m) - f(s_1)|}{|f(t_m) - f(s_1)| + |f(t_m) - f(s_0)|} & t_m \in V_f \end{cases} \tag{8}$$

as well as for those of E_2

$$E_2(\Gamma_m, \Gamma_n) = \frac{1}{|f(t_m) - f(t_n)|} |\Gamma_m - \Gamma_n| + \beta \frac{d(t_m, t_n)}{|d(t_m, t_n) + \overline{d}|} |\Gamma_m - \Gamma_n| \tag{9}$$

where $d(t_m, t_n)$ represents the distance between the barycenter of the triangle t_m and that of the triangle t_n. The triangle t_m is adjacent to the triangle t_n. $\overline{d}$ is the average value of the sum of the distance between the barycenters of any two neighboring triangles.

While the coefficient $\lambda \geq 0$ provides the relationship between the two costs E_1 and E_2, the coefficient $\beta \geq 0$ represents the relative importance of the distance and the gray values of triangles. In this experiment, the models had values of $\lambda = 70$ and $\beta = 0.286$.

This problem can be solved using the graph-cut technique [4] by finding the minimum cut C consisting of a set of edges that separate the graph G into two sub graphs whose triangles represent the relevant segments (see Fig. 3). Finally, the required minimum cut is generated using the max-flow/min-cut algorithm provided in Ref. [4].

An example of graph cut technique application is shown in Fig. 4. The picture on the left presents a triangular mesh from a CT image. The average gray value for all the triangles is computed and shown in the middle picture. The triangular mesh is successfully classified into different materials as shown in the picture on the right.

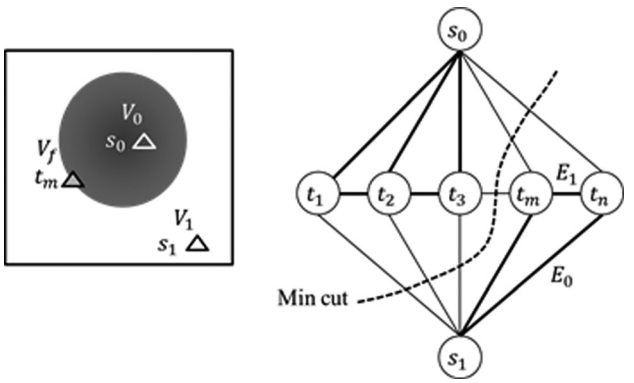

Fig. 3 Graph of $G = (V, \Xi)$. V represents all triangles $t_1, t_2, t_3, \ldots, t_m, t_n$ in one model and the source node s_0/the sink node s_1. Ξ denotes the edges between the centroids of any two adjacent triangles and the edges linking triangles and the source node/sink node. E_1 and E_2 are, respectively, the cost of the edges linking triangles and the source node/sink node and the cost of the edges between the centroids of any two adjacent triangles. The graph is segmented into two parts by a min cut consisting of edges.

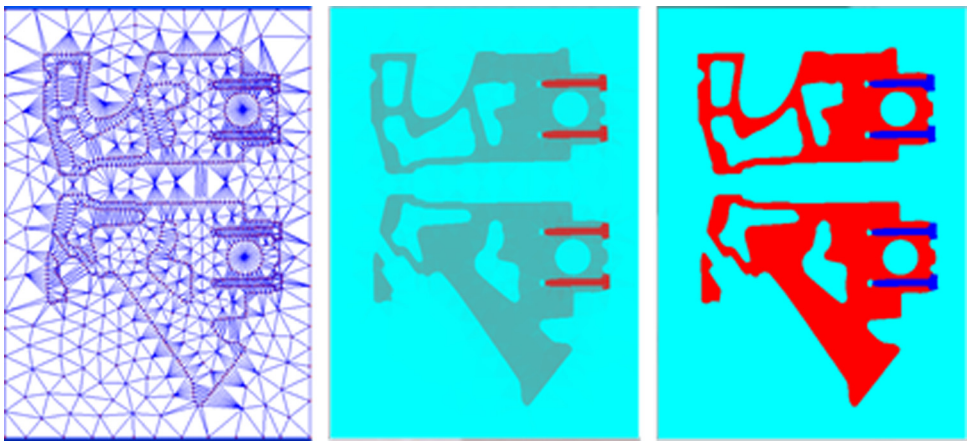

Fig. 4 Mesh segmentation. The image on the left shows the triangular mesh, the one in the middle presents the average gray value for all triangles, and the one on the right illustrates the segmented mesh model with individual materials represented in different colors.

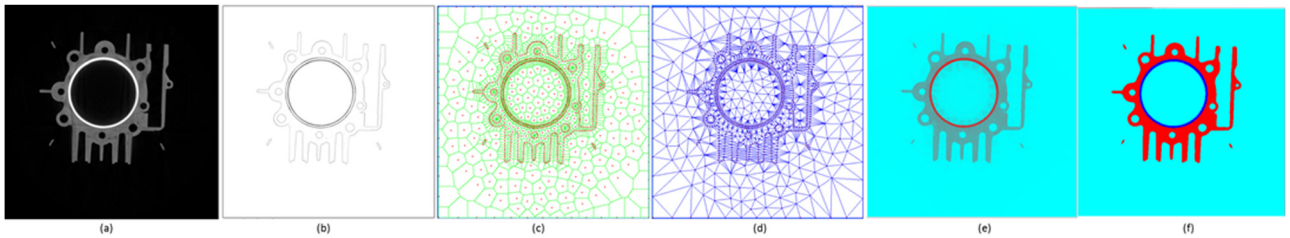

Fig. 5 2D example: Cs1 consisting of 512 * 512 pixels with 1000 sites. (*a*) Shows the input image with the gray value. In line with this value, the density map shown in (*b*) is generated. (*c*) Presents the final optimized CVT, (*d*) gives the corresponding triangular mesh associated with CVT, (*e*) shows the average gray value for all triangles, and (*f*) indicates the segmented meshes created using the novel graph cut method.

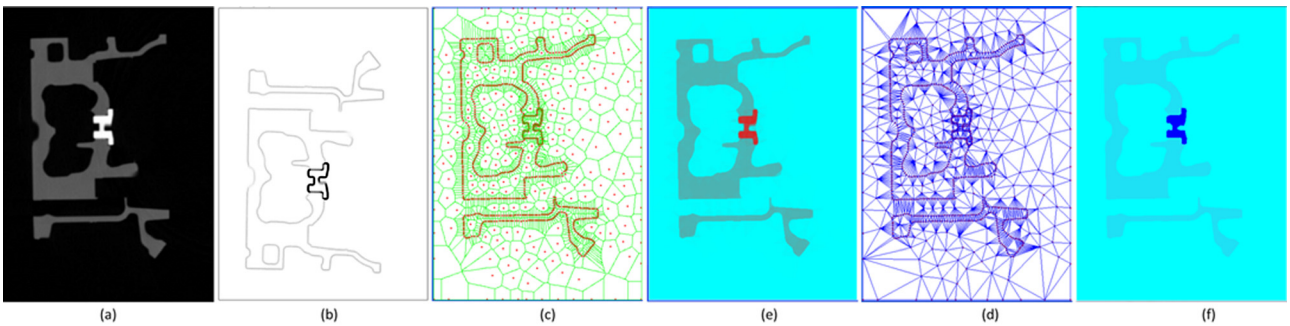

Fig. 6 Engine head 2 consisting of 500 * 700 pixels with 1000 sites. (*a*) Shows the input image with the gray value. In line with this value, the density map shown in (*b*) is generated. (*c*) Presents the final optimized CVT, (*d*) gives the corresponding triangular mesh, (*e*) shows the average gray value for all triangles, and (*f*) indicates the segmented mesh created using the novel graph cut method.

7 Experiment Results and Discussion

The gray values of pixels are almost the same in identical materials but vary among different ones. Those on the interface between different materials change abruptly. Based on these CT image characteristics, a suitable density map is generated that can be used to minimize the energy function and produce a triangular mesh with respect to such boundaries. The graph cut method can then be applied to successfully segment the triangular mesh into different materials based on the gray values of the CT image and the boundaries conforming to the edges of the triangular mesh.

Several examples are shown in Figs. 5–7, where the 2D image shown in (a) is input. In line with the relevant gray values, the density map shown in (b) is generated. The Voronoi regions are iteratively optimized and their sites are relocated by minimizing the energy function until convergence is reached, which results in the final CVT shown in (c). The dual graph is the triangular mesh given in (d), (e) shows the average gray value of each triangle, and (f) shows the segmented triangular mesh created using the graph cut method. Even if the number of sites is selected differently, the desired triangular meshes and their segmented meshes are also obtained (see Figs. 5 and 8). In Fig. 5, the number of sites was set as 1200 for a cylinder head image consisting of 500 * 700 pixels. Here, (a) shows the input image with the gray value, (b) illustrates the corresponding CVT, (c) introduces the corresponding triangular mesh, (d) shows the corresponding triangular mesh segmented into differently colored patches using the graph cut method, (e) presents the average gray values for all triangles, and (f) indicates the segmented meshes created using the novel graph cut method. In Fig. 8, the number of sites is 1000 for a 2D image

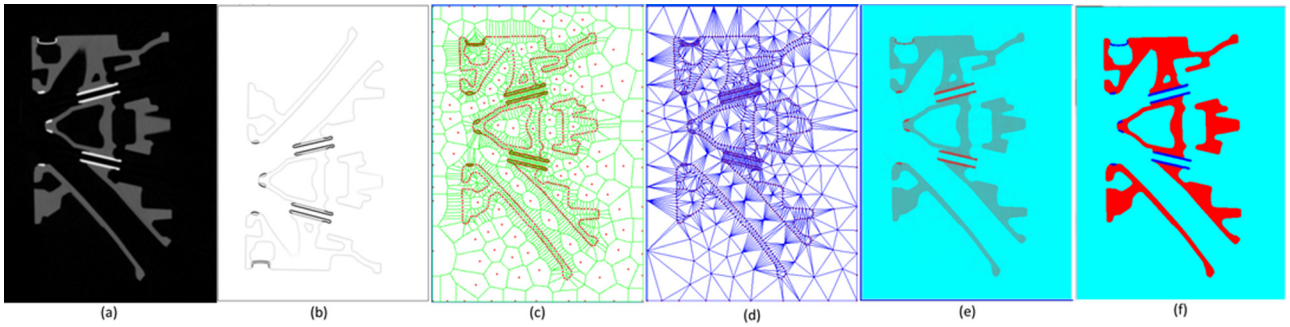

Fig. 7 Engine head 3 consisting of 500 * 700 pixels with 1000 sites. (*a*) Shows the input image with the gray value. In line with this value, the density map described in (*b*) is generated. (*c*) Presents the final optimized CVT, (*d*) gives the corresponding triangular mesh, (*e*) shows the average gray value for all triangles, and (*f*) indicates the segmented meshes created using the novel graph cut method.

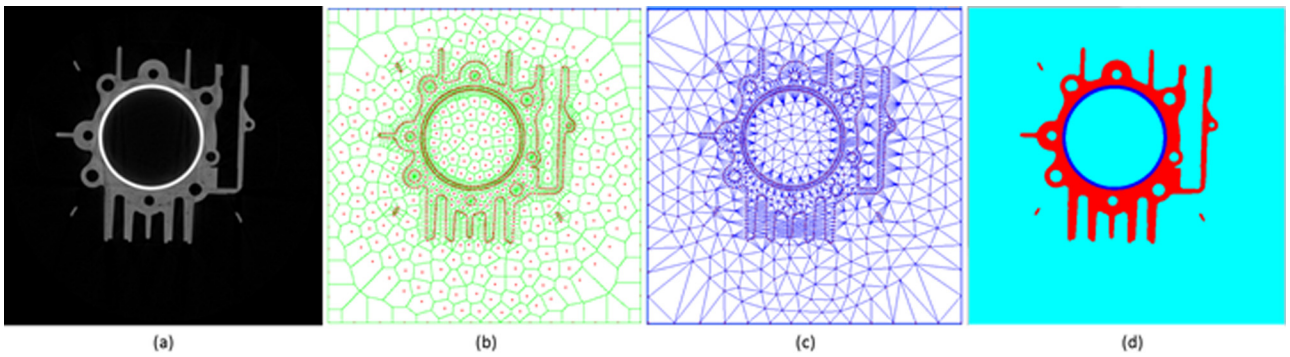

Fig. 8 2D example: Cs2 consisting of 512 * 512 pixels with 1200 sites. (*a*) Shows the input image with the gray value, (*b*) presents the final optimized CVT, (*c*) gives the corresponding triangular mesh, and (*d*) illustrates the segmented meshes created using the novel graph cut method.

Table 1 Time statistics

Models	#input pixels	#output vertices	Mesh generation time (s)	Mesh segmentation (s)
Cs1	512*512	1000	11.12	1.16
Cs2	512*512	1200	10.01	1.36
EngineHeadl	500*700	1000	9.56	1.18
EngineHead2	500*700	1000	10.21	1.16
EngineHead3	500*700	1000	10.16	1.19
EngineHead4	500*700	1000	9.43	1.12
EngineHead5	500*700	1000	10.31	1.17
EngineHead6	500*700	1000	10.29	1.15

with 512 * 512 pixels. Here, (a) shows the input image with the gray value, (b) presents the final optimized CVT, (c) gives the corresponding triangular mesh associated with CVT, and (d) illustrates segmented meshes created using the novel graph cut method. Table 1 shows the running time and other statistics for some of the outcomes detailed in this paper. The results were generated using a desktop computer with an Intel(R) Core(TM) i7-2600S CPU @ 2.80 GHz. In this paper, the edges of the triangles in the resulting meshes can be aligned with the boundaries of the original image data to guarantee the accuracy. CVT can approximate the different regions representing the materials of the underlying images. Second, CVT steered by the density map can ensure the Voronoi sites well-spaced as the density map decide segments with higer values of ρ more sites than segments with lower values of ρ, which lead to more accurate triangular meshes than those gained by methods without any relationship to the underlying images. Third, with CVT, some integral computations required by our method can be simple done. At last, the corner pixels of the underlying images can be easily constrained in the triangular as vertices, which guarantees those sites will be maintained in the triangular meshes. On the other hand, in the novel approach proposed here, the computation cost is significantly reduced. The image is first subjected to CVT [3] (see Fig. 1) and Delaunay tessellation to make a triangular mesh, which is then clustered into different segments expressing different materials using the graph cut method [4]. Although this approach gives powerful clustering performance, it is time-consuming (approximately $O(n^2 \log n)$, where n is the number of elements such as pixels and triangles) and memory-intensive. Thus, the graph cut method is used not to the pixels but to the tessellated triangles as there are greatly fewer triangles than pixels in the same image.

Judging the quality of these experimental results is a difficult task, as a variety of quality evaluation methods have been proposed in recent decades. However, visual inspection enables good assessment of the results.

8 Conclusion

This paper introduces a new meshing approach for X-ray CT images with different materials. Rather than being initially

segmented, the image is converted into a triangular mesh using CVT with more vertices focusing on the boundaries between different materials. This mesh is then segmented into components made of different materials using a graph cut method. Here, a minimum cut is used to segment the mesh into two components including seed triangles with the highest and lowest gray values. This step is repeated recursively until the gray values of all triangles in every part are almost the same. The experiment results show that the proposed approach can be adopted to successfully generate segmented triangular meshes from CT images. The authors plan to extend the method to 3D application in future work. In addition, as the quality of the triangular meshes generated needs improvement, this will be investigated based on a new method.

Acknowledgment

Author Caiyun Yang was supported through the "Global Center of Excellence for Mechanical Systems Innovation," Global COE Program under the auspices of Japan's Ministry of Education, Culture, Sports, Science and Technology.

References

[1] Kruth, J. P., Bartscher, M., Carmignato, S., Schmitt, R., De Chiffre, L., and Weckenmann, A., 2011, "Computed Tomography for Dimensional Metrology," CIRP Ann., **60** (2), pp. 821–842.
[2] Shammaa, H., Suzuki, H., and Ohtake, Y., 2011, "Creeping Contours: A Multi-label Image Segmentation Method for Extracting Boundary Surfaces of Parts in Volumetric Images," ASME J. Comput. Information Science in Engineering **11**(1), pp. 603–610.
[3] Du, Q., Gunzburger, M., and Ju, L. L., 2010, "Advances in Studies and Applications of Centroidal Voronoi Tessellations," Numerical Mathematics: Theory, Methods and Applications **3**(2), pp. 119–142.
[4] Boykov, Y., and Kolmogorov, V., 2004, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," IEEE Tran. on Pattern Analysis and MachineInterlligence **26**(9), pp. 1124–1137.
[5] Du, Q., and Wang, D. S., 2003, "Tetrahedral Mesh Generation and Optimization Based on Centroidal Voronoi Tessellations," International Journal for Numerical Methods in Engineering **56**(9), pp. 1355–1373.
[6] Chen, L., and Xu, J. C., 2004, "Optimal Delaunay Triangulations," J. Comput. Math., **22**(2), pp. 299–308.
[7] Du, Q., Gunzburger, M. D., and Ju, L. L., 2003, "Constrained Centroidal Voronoi Tesselations for Surfaces," SIAM J. Sci. Comput. **24**(5), pp. 1488–1506.
[8] Du, Q., and Wang, D. S., 2005, "Anisotropic Centroidal Voronoi Tessellations and Their Applications," SIAM J. Sci. Comput. **26**(3), pp. 737–761.
[9] Valette, S., and Chassery, J. M., 2004, "Approximated Centroidal Voronoi Diagrams for Uniform Polygonal Mesh Coarsening," Computer Graphics Forum (Eurographics 2004 proceedings) **23**(3), pp. 381–389.
[10] Paul Chew, L.,1989, "Guaranteed-Quality Triangular Meshes," Department of Computer Science, Tech Report No. 89-983, Cornell University.
[11] Ruppert, J., 1993, "A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation," The fourth annual ACM SIAM symposium on Discrete algorithms, pp. 548–585.
[12] Du, Q., Faber, V., and Gunzburger, M., 1999, "Centroidal Voronoi Tessellations: Applications and Algorithms," SIAM Rev. **41**, pp. 637–676.
[13] Du, Q., and Gunzburger, M., 2002, "Grid Generation and Optimization Based on Centroidal Voronoi Tessellations," Appl. Math. Comput., **133**(2–3), pp. 591–607.
[14] Dardenne, J., Valette, S., Siauve, N., Burais, N., and Prost, R., 2009, "Variational Tetrahedral Mesh Generation From Discrete Volume Data," Visual Comput., **25**(5), pp. 401–410.
[15] Zhang, Y. J., Hughes, T. J. R., and Bajaj, C. L., 2010, "An Automatic 3D Mesh Generation Method for Domains With Multiple Materials," Comput. Methods Appl. Mech. Engrg. **199**, pp. 405–415.
[16] Veksler, O., 1999, "Efficient Graph-Based Energy Minization Methods in Computer Vision," Ph.D thesis.
[17] Li, Y., Sun, J., Tang, C. K., Shum, I. Y., 2004, "Lazy Snapping," ACM Trans. Graphics **23**(3), pp. 303–308.
[18] Boykov, Y. Y., and Veksler, O., 2006, "Graph Cuts in Vision and Graphics Theories and Applications," *Handbook of Mathematical Models in Computer Vision*, Springer, New York, pp. 79–96.
[19] Boykov, Y. Y., and Jolly, M. P., 2001, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," Eighth IEEE International Conference on Computer Vision 1, pp. 105–112.
[20] Yang, C., Ohtake, Y., and Suzuki, H., 2011, "Sealed Decomposition of a Triangular Mesh With Tetrahedral Mesh Segmentation," International Journal of Computer-Aided Design and Applications **8**(3), pp. 421–433.
[21] Ward, J. H., 1963, "Hierarchical Grouping to Optimize an Objective Function," Journal of the American Statistical Association **58**, pp. 236–244.
[22] Fabri, A., 2001, "CGAL—The Computational Geometry Algorithm Library," Proceedings of 10th International Meshing Roundtable, pp. 137–142.