Extraction of Isosurfaces from Multi-Material CT Volumetric Data of Mechanical Parts

M. Haitham SHAMMAA∗ Hiromasa SUZUKI† Yutaka OHTAKE‡

Research Center for Advanced Science and Technology
The University of Tokyo

Figure 1: Triangulated mesh isosurfaces of mechanical parts generated from CT volumetric data composed of more than one material using our method.(a) Full model of Automobile cylinder head composed of aluminum (yellow) and steel (brown).The size of the volumetric model is 425 × 400 × 530 voxel. The surface of steel contains 408034 triangles, and for the aluminum 444032 triangles.(b) vibration damper in automobile engine composed of steel (brown) and rubber (cyan).The size of the volumetric model is 400 × 360 × 320. The surface of steel contains 1490228 triangles, and for the rubber 640154 triangles.

Abstract

We introduce a method for extracting boundary surfaces from volumetric models of mechanical parts by X-ray CT scanning. When the volumetric model is composed of two materials, one for the object and the other for the background (Air), these boundary surfaces can be extracted as isosurfaces using a contouring method such as Marching Cubes [Lorensen and Cline 1987]. For a volumetric model composed of more than two materials, we need to classify the voxel types into segments by material and use a generalized Marching Cubes algorithm that can deal with both CT values and material types. Here we propose a method for precisely classifying the volumetric model into its component materials using a modified and combined method of two well-known algorithms in image segmentation, region growing and Graph-cut. We then apply the generalized Marching Cubes algorithm to generate triangulated mesh surfaces. In addition, we demonstrate the effectiveness of our method by constructing high-quality triangular mesh models of the segmented parts.

CR Categories: I.3.5 [COMPUTER GRAPHICS]: Computational Geometry and Object Modeling— [I.4.6]: IMAGE PROCESSING AND COMPUTER VISION—Segmentation, Region growing

Keywords: Isosurface, multi-material, volumetric data

1 Introduction

X-ray CT (Computer Tomography) is a powerful non-destructive evaluation technique to generate cross-sectional images of objects from which we can further produce three-dimensional volumetric models of voxels. In our study, we mainly focus on industrial applications for such CT volumetric models to analyze machine parts. In this case, we are particularly interested in parts made of more than one material. Figure 1 show typical examples of a cylinder head and a vibration damper respectively, which are both used in automotive engines. In both cases, a number of mechanical elements made of different materials are assembled to create a functioning mechanical part. Our challenge is to extract the boundary surfaces between these elements of different materials from a volumetric model of the part.

When a part is made of a single material, these boundaries can be easily extracted using an isosurfacing method such as Marching Cubes [Lorensen and Cline 1987]. Figure 2 shows a simple example of two materials, aluminum (Al) and the background (Air). A curve plot (on the right of the figure) shows how the intensity of the CT value at each voxel changes gradually along the yellow scan line. In this case, an isosurface is simply generated at the boundary of the two materials by using the isovalue $iso(Al, Air)$ as a threshold to distinguish the aluminum and the air. As our CT images are generally very clear, an isosurfacing method is useful because it can generate surfaces at a sub-voxel level of accuracy by interpolating

∗e-mail: haitham@den.rcast.u-tokyo.ac.jp
†e-mail: suzuki@den.rcast.u-tokyo.ac.jp
‡e-mail: yu-ohtake@den.rcast.u-tokyo.ac.jp
the CT values of voxels. Such isosurface accuracy is one of the key requirements of industrial applications, so here we stick to an isosurfacing approach, although other contouring methods can be used.

However, this approach cannot be applied to parts made of more than two materials. Let us take the case in Figure 3 as an example. The model consists of three materials, steel (St), aluminum (Al) and air. The targets of generation are the boundary surfaces between the different materials as shown in Figure 3-c. However, since isosurfacing methods can deal with only one threshold value at a time, they can only generate the three bounding surfaces shown in Figure 3-b by isosurfacing three times with the three different isovalue.

Figure 3: Volumetric model containing three materials. (a) Curve plots of CT values along the yellow arrow. (b) Trivial isosurfaces between each pair of materials generate unsatisfactory boundary surfaces. (c) Isosurfaces generated using our method give satisfactory geometrical shapes.

The most straightforward method of material segmentation is based on thresholding the CT values of voxels, because they are basically proportional to material density. However, even though our CT images are very clear, this method does not work well as shown by the boundary between the steel part and the background in Figure 3-b. As the CT value gradually increases from the steel region (around 10,000) to the background (around 0), some voxels in this transition area have CT values corresponding to aluminum (around 2,600), so simple thresholding misclassifies these voxels as such. This problem becomes critical when the volumetric model includes materials with very high and very low CT values. Such an example is shown in Figure 4, which contains rubber and steel. A more powerful segmentation method is therefore needed.

In this paper, we present a method for generating high-quality isosurfaces as well as satisfactory geometrical shapes. To this end, we first apply a restricted region growing algorithm to define a rough location for each material part, which enables correct classification of the majority of voxels in this step. The remaining unclassified voxels (referred to as doubt voxels) located mostly at the boundaries between adjacent materials are fed into a Graph-cut algorithm [Boykov and Funka-Lea 2006; Boykov and Jolly 2001; Kolmogorov and Zabih 2004] after defining a proper weight for the graph edges to satisfy the prescribed conditions. This Graph-cut process should be repeated \( n - 1 \) times in order to segment the number \( n \) of materials in the volumetric model. Ultimately, a fully classified set of voxels will be ready for the surface generation step using a variant of the generalized Marching Cubes method [Ju et al. 2002; Suzuki et al. 2007].

This paper is organized into six sections. In Section 2 we survey important related work, while Section 3 presents our method of classifying the voxels in a volumetric model, including restricted region growing and the sequenced Graph-cut segmentations and their results. Section 4 discusses application of the existing contouring method, and in Section 5 we present some results for our proposed method. Section 6 discusses the results and future work.

2 Related Work

In the field of volumetric modeling and visualization a lot of work have been done in modeling [Kumar and Dutta 1997] and volume rendering [Kniss et al. 2002] of multi-material objects. In addition many works have been done in the field of reverse engineering [Azerimkov and Fischer 2006] for generating isosurfaces in volumes [Azerimkov and Fischer 2005] and reconstruction of material interface [Bonnell et al. 2003]. Most of isocontouring methods that generate surfaces on the boundaries of multi-materials require classification of data elements into segments each of which represent one material. A vast number of algorithms came from the computer vision community for detecting object boundaries in images have been introduced in the last two decades. Snakes algorithm [Kass et al. 1988], active contours [Caselles et al. 1997], intelligent scissors [Mortensen and Barrett 1998] and many other methods were introduced to segment an image into two partitions, “object” and
3 Classification of voxels in a volumetric model

When a volumetric model contains $n$ materials $i = 0, 1, \ldots, n - 1$, we denote their average CT values by $m_0, m_1, m_2, \ldots, m_{n-1}$. Value $m_i$ is obtained by averaging the CT values of a region filled with material $i$. A threshold value is also denoted to extract the isosurface between materials $x$ and $y$ using $iso(x, y)$. As the CT value of an arbitrary voxel is blurred by a noise model that has a Gaussian-like kernel, the threshold value $iso(x, y) = (m_x + m_y)/2$ will be a good approximation.

3.1 Voxel Classification via Graph-cut

Graph-cut for N-dimensional image segmentation [Boykov and Funka-Lea 2006] was introduced as a global segmentation method. This is done by minimizing a cost function over binary variables whose values only indicate whether the pixel is inside or outside the object of interest. As introduced in [Boykov and Jolly 2001], the user needs to interactively define certain voxels as object or background to provide hard constraints, and additional soft constraints incorporating both boundary and region information are automatically set. The characterization of the energy functions to be minimized by the Graph-cut is detailed in [Kolmogorov and Zabih 2004]. Applying the Graph-cut to our classification problem, the graph nodes are the voxels of the target volumetric model, and the edges are induced from the connectivity of these voxels defined by a 6-, 18-, or 26-neighborhood system. Their cost function can be defined based on the similarity of voxel CT values, and the resulting cut represents the boundary between segments of different materials. However, many problems were encountered when trying to apply Graph-cut segmentation directly to our problem, as described in [Boykov and Funka-Lea 2006]. First, since the volumetric model is very large in general cases, constructing and cutting a full graph will take up large amounts of time and resources. Moreover, the cut provided by the Graph-cut is a global minimum solution of the defined cost function, and the resulting boundaries may not therefore be guaranteed to have the same topological characteristics as the real parts. In addition, since our ultimate target is to construct the isosurfaces between different materials, having the user interactively define the hard constraint voxels to restrict the cut of the graph may not guarantee that the cut will pass through the isovalue threshold if the voxels are wrongly defined. As our solution requires a delicate segmentation that must preserve the topology of the extracted objects as well as a cut that is consistent with isovalue thresholds, here we propose a fully automatic segmentation method. This involves first performing a restricted region growing, followed by $n - 1$ performances of the Graph-cut iteration in order to classify a volumetric model consisting of $n$ materials.

3.2 Restricted Region Growing

Region growing is well known as one of the simplest segmentation techniques, and was first presented in [Haralick and Shapiro 1985] followed by [Adams and Bischof 1994] as a robust, rapid and free of tuning parameters method for the segmentation of intensity images. The algorithm needs a small initial set of seed pixels (voxels), from which the region is grown by adding neighboring pixels (voxels) whose CT values are similar to those of the seeds. Unfortunately, this simple method is not robust in practice because it is prone to a range of problems such as “leaking” through weak spots in object boundaries. As a result of this problem, we present the idea of restricted region growing. In order to segment a model of $n$ materials, we define $n(n - 1)/2$ threshold values for all pairs of different materials, i.e. $iso(0, 1), iso(0, 2), \ldots, iso(n - 2, n - 1)$. Consider a set of voxels $P$ of the whole volumetric model including objects and the background and a set $N$ of all unordered pairs $(p, q)$ of neighboring voxels of $P$. The neighborhood of the voxels is defined by an 18-neighborhood system. The initial seed voxels of a material $i$, $D_i \subset P$ of the restricted region growing are defined as

$$D_i = \{p \in P, L_i < I_p < U_i \forall (p, q) \in N, \text{ and } |I_p - I_q| < \alpha\}$$

for $i = 0, 1, \ldots, n - 1$

where

$$L_i = \max_{j=0,1,\ldots,n-1} \{iso(i, j) : iso(i, j) < m_i\}$$

and

$$U_i = \min_{j=0,1,\ldots,n-1} \{iso(i, j) : iso(i, j) > m_i\}.$$ 

$I_p$ denotes the CT value of voxel $p$, and $\alpha$ is a variable represents the distribution of noise among neighboring voxels in the CT data.

This will ensure that the selected initial seeds have a neighborhood occupied by voxels of similar CT values with a low gradient so that selection of seeds in boundary areas is prevented. The growing of regions will follow conditions similar to those of the seed selection, so that newly added voxels should be in the neighborhood of the initial seeds and are not in high-gradient regions. The results of restricted region growing are given in Figure 4, which shows a slice of a volumetric model consisting of three materials (steel, rubber and air). A comparison is also shown for the results of general region growing and those of our proposed restricted region growing method.

Restricted region growing will generate a set of segments $\{R_i\}$ each of which represents the main bodies of material $i$. Note that each $R_i$ can be disconnected. Voxels that could not be classified using restricted region growing (mostly those located at boundary areas between different materials) are referred to as doubt voxels,
and will be classified in the next step using a sequence of Graph-cut operations. Let \( X \) be the set of doubt voxels. We also need to define the closure voxels \( C_i \) of \( R_i \) as follows:

\[
C_i = \{ p \in R_i | \exists (p, q) \in N, q \in X \}.
\]

This restricted region growing operation is very fast, and has \( O(s.n) \) complexity where \( s \) is the number of voxels in the volumetric data, \( n \) is the number of materials. The only voxels fed to the Graph-cut in the next step are the doubt voxels \( X \) and the set of all closure voxels \( \{ C_i \} \) of the classified regions. Restricted region growing is therefore very useful as a pre-process to the expensive Graph-cut in drastically reducing the size of the graph. Moreover, it will ensure the correct geometrical shape of the segmented parts, this is because the closure voxels \( C_i \) will be hard-constrained to the terminal nodes of the graph, and this will force the cut to pass only through the doubt voxels at the boundary area of material \( i \).

### 3.3 Sequential Graph-cut Operations

After performing restricted region growing, we need to perform sophisticated classification to define the material segment to which each doubt voxel belongs using the Graph-cut. However, as the Graph-cut is a binary method, it cannot perform multiple cuts using the Graph-cut on any arbitrary graph generating more than two subgraphs in polynomial time, as the cutting of a graph into more than two subgraphs will create an NP-hard problem [Dahlhaus et al. 1992]. We therefore need to repeat the classification using the Graph-cut \((n-1)\) times in order to classify the doubt voxels into \( n \) segments of different materials. In each iteration of the Graph-cut, voxels that belong to one of the materials are completely classified out from the doubt voxels. Let \( M = \{0, 1, 2, \cdots, n-1\} \) to form an ordered set of all the \( n \) materials, where \( m_0 < m_1 < m_2 < \cdots < m_{n-1} \). Considering \( m_0 \) is the CT value of the background, we select the material \( i \neq 0 \) starting from material 1 increasingly to be classified in the \( i^{th} \) iteration of the Graph-cut sequence, and define \( M \) as the set of unclassified materials excluding the material \( i \), \( M = \{ i+1, i+2, \cdots, n-1 \} \). In Figure 5, by assuming \( m_0 < m_1 < m_2 \), the first material to be segmented must be material 1 because it is the one that will most probably be misclassified at the boundaries of material 2 with a higher CT value. We repeat the following process while \( M \) is not empty.

**Figure 5:** (a) A simple illustrative example of a set of 2D pixels to be classified into three materials \( \{m_0, m_1, m_2\} \). (b) Results of restricted region growing; doubt pixels are shown in yellow. (c) The boundary pixels of restricted region growing are marked with an X. (d) The nodes of the graph \( \times \)-marked blue nodes are hard-constrained to one terminal, while the \( \times \)-marked red nodes are hard-constrained to the other terminal of the graph.

### 3.3.1 Graph construction

We define an undirected graph \( G = (V, E) \) as a set of nodes \( V \) and a set of edges \( E \). \( V \) consists of nodes corresponding to voxels of \( X \) (doubt voxels), \( \{ C_j \} \), \( j \in M \) (closure voxels), and two specially designated terminal nodes \( s \) (source) and \( t \) (sink). \( s \) represents material 1, while \( t \) represents the sum of materials 0 and 2.

We then define edges between those nodes. First, the edges between terminal nodes and those of closure voxels are defined as follows:

\[
E_s = \{(p, s) | p \in C_i \} \quad \text{and} \quad E_t = \{(p, t) | p \in C \}.
\]

where \( C_i \) corresponds to closure voxels of material \( i \) to be classified (blue \( \times \)-marked voxels in Figure 5), and \( C = \bigcup_{j \in X} C_j \), which is a set of nodes corresponding to closure voxels of the materials of \( M \) (red \( \times \)-marked voxels in Figure 5). Next, we define edges \( (p, q) \) between nodes \( p, q \) for doubt and closure voxels that are 6-connected:

\[
E_{dc} = \{(p, q) | p, q \in V, p \text{ and } q \text{ are 6-connected} \}.
\]

### 3.3.2 Weight definition

It is the most important for the segmentation to define proper weights for these edges. Weights for edges \( E_s \) and \( E_t \) are set to infinity as hard constraints. This will ensure that no cut will happen to exclude any closure voxel belonging to its corresponding terminal.

**Figure 6:** Edges of some nodes of the graph, \( w_{\text{cut}} \) (red) and \( w_{\text{join}} \) (blue). Each hard-constrained node tries to pull the doubt voxel toward one of the terminals using different edge weights.

For the edges \( (p, q) \) of \( E_{dc} \), the weights \( w(p, q) \) are defined to be proportional to \( |I_p - I_q| \) which is the absolute difference of \( I_p \) and \( I_q \) in many segmentation applications of Graph-cut [Boykov and Funka-Lea 2006; Kolmogorov and Zabih 2002; Roy and Cox 1998]. However, this would not be a proper definition of weights for our case since our target is not only to segment the data, but also to define the boundaries of the segments to be consistent with the isosurface. Consequentially we define \( w(p, q) \) using the following rule:

\[
w(p, q) = \begin{cases} 
  w_{\text{cut}} & \text{if } I_p \leq v \leq I_q \text{ or } I_p \geq v \geq I_q \text{ for some } v = \text{iso}(m_i, m_j), j = 0, 1, \cdots, n-1 \\
  w_{\text{join}} & \text{otherwise}
\end{cases}
\]

\( w_{\text{cut}} \) and \( w_{\text{join}} \) are both positive constants. \( w_{\text{cut}} \) is assigned when \( I_p \) is over some isovalue threshold and \( I_q \) is below it, or vice versa. While \( w_{\text{join}} \) is assigned in all other cases where \( I_p \) and \( I_q \) belong to the same range of material inside the isosurface. Since the edges of \( E_s \) and \( E_t \) will not be cut due to the hard constraints, the Graph-cut separates \( G \) into the two subgraphs of \( S \) and \( T \) so as to minimize the following cost function:

\[
E = \sum_{p \in S, q \in T} w(p, q)
\]

\( S \) includes a terminal node \( s \) and other nodes to be classified as material \( i \), while \( T \) includes \( t \) and all the other nodes. In this summation of Eq. 1, let \( N_{\text{cut}} \) be the number of edges assigned \( w_{\text{cut}} \).
and \( N_{\text{join}} \), the number of edges assigned \( w_{\text{join}} \). Eq. 1 can then be rewritten as:

\[
E = \sum_{p \in S, q \in T} w(p, q) = N_{\text{cut}} \cdot w_{\text{cut}} + N_{\text{join}} \cdot w_{\text{join}}
\]

(2)

Eq. 2 is further rewritten to:

\[
E = w_{\text{join}} \left( N_{\text{cut}} \cdot w_{\text{cut}}/w_{\text{join}} + N_{\text{join}} \right) = w_{\text{join}} \left( \tau \cdot N_{\text{cut}} + N_{\text{join}} \right)
\]

(3)

using the cut-ratio parameter \( \tau = w_{\text{cut}}/w_{\text{join}} \), which will specify the relative importance of these two weights, where for a reasonable cut the cut-ratio varies in the range \( 0 < \tau < 1 \) depending on how the final mesh is reflecting more similarity to the original shape as shown in Figure 11.

Defining this form of cost function will find the optimal cut by simply pulling doubt voxels using different weights \( \{w_{\text{cut}}, w_{\text{join}}\} \) toward both graph terminals. This will prevent voxel misclassification at boundary areas as shown in Figure 6, because nodes that are linked to only a few nodes by \( w_{\text{join}} \), but also to enough nodes by \( w_{\text{cut}} \) will be pulled toward the cut direction even if \( w_{\text{cut}} < w_{\text{join}} \). This is akin to many small children beating a few strong men in a tug-of-war. The concept is presented in Figure 7, which shows illustrative results of applying the Graph-cut using different values of cut-ratio \( \tau \).

In Figure 7, we can see that the isoline will pass through or near the isovalue threshold for any value of the cut-ratio \( \tau \), however by defining \( \tau \) we can control the shape smoothness of isolines (in bold) to give such a satisfactory shape that is similar to the original part.

\[\text{Figure 7: Graph-cut illustrative results in 2D. An 8-neighborhood system is used to build the graph. Blue pixels belong to one subgraph, and red ones belong to the other subgraph (a) } \tau = 0.4, \text{ (b) } \tau = 0, \text{ (c) } \tau = 3.\]

3.3.3 Graph-cut

After building the graph as described above and performing the Graph-cut to minimize the cost function of Eq. 3, all the doubt voxels for nodes of subgraph \( S \) are classified as material \( i \) (material 1 in this example), while other doubt voxels for \( T \) will still be considered doubt voxels. By denoting these classified voxels as \( M_i \), we subtract classified voxels in \( S \) \( (S \cap X) \) from \( X \) and also remove material \( i \) from the list of materials \( M \) by \( M = M - M_i \). And we repeat the above process. We denote \( W_j \) as the set of voxels which are completely classified to be material \( i \) by both the restricted region growing and the Graph-cut segmentation.

The material selection order for the Graph-cut segmentation sequence is determined by giving priority to segmenting materials with lower CT value (excluding the background) of the volumetric data histogram. Choosing to perform the Graph-cut sequence in this order will give the priority of classification to voxels of materials of lower CT value those most likely to be misclassified on the boundary of higher CT value materials as shown in Figure 8, the correct classification result is obtained by performing the Graph-cut for the rubber in the first iteration, where rubber has lower CT value than steel voxels.

\[\text{Figure 8: Effect of the order of material selection on the sequential Graph-cut (a) } \text{CT volumetric model for three materials (steel - rubber - air). (b) Results of classification of doubt voxels using sequential Graph-cut starting with steel then rubber. (c) Results of classification of doubt voxels using sequential Graph-cut starting with rubber then steel.}\]

The series of Graph-cut operations will continue until we have the final result of the volumetric model’s classification into segments of \( n \) materials as shown in Figure 9.

\[\text{Figure 9: Second iteration of Graph-cut. (a) Result of applying the first Graph-cut. (b) The nodes of the graph (c) Result of the second Graph-cut (d) Final classification result.}\]

4 Multi-material Countouring

Once the material types of all voxels are classified, we can generate boundary surfaces between different materials using an extended version of the Marching Cubes method. The standard technique generally processes a cubic cell whose corners are voxels of a volumetric model. It inserts a vertex to a cell edge if the isovalue is between the CT values of the two end points of the edge as shown in Figure 10-a.

\[\text{Figure 10: Examples of the generalized marching cubes table. Colors of voxels represent material types.}\]
Additionally, an extended version of the Marching Cubes method can deal with voxels that have material types and CT values. As shown in Figure 10-b, it inserts vertices on edges whose two end points are of different materials. The positions of these vertices are computed by interpolation using their CT values. We used our own generalized Marching Cubes method [Suzuki et al. 2007] that can generate non-manifold surface meshes. We also apply a dual contouring method [Ju et al. 2002]. Non-manifold junctions occur where more than two materials meet, however in our case each isosurface generated of each part is a manifold surface.

5 Results

We implemented the proposed methods described in this paper on a standard workstation equipped with a Pentium 4 3.00 GHz CPU and 3.00 GB of main memory, our prototype program uses MinCut/Max-Flow algorithms provided by [Boykov and Kolmogorov 2004] for the Graph-cut. The calculation time shown in each example includes the total time of classification of voxels and triangulated surface generation. However, before introducing the results we first need to define a quality measure for the extracted isosurfaces. Consider an isosurface passing at isosurface threshold $iso(x, y)$ between two different materials $x$ and $y$. By assuming $m_x < m_y$ for any voxel $p$ of the volumetric model in an unordered pair of neighboring voxels $(p, q)$ with CT values $(I_p, I_q)$, the isosurface quality measure $Q(x, y)$ of the isosurface is defined as follows:

$$Q(x, y) = \frac{\# \{ (p, q) \mid I_p \leq iso(x, y) \leq I_q, p \in W_x, q \in W_y \}}{\# \{ (p, q) \mid p \in W_x, q \in W_y \}}$$

where $p$ and $q$ are 6-, 18- or 26-connected, $\# \{ \bullet \}$ denotes the number of elements of a set.

First of all we show a simple example of a volumetric model that includes two blocks consisting of two materials, aluminum (Al) and rubber (Ru), in addition to the background (Air). The size of this volumetric model is $256 \times 256 \times 142$ voxels, with a voxel size of $0.4 \times 0.4 \times 0.5$ mm. The three materials have average CT values of $m_{Al} = 2,700$, $m_{Ru} = 1,000$ and $m_{Air} = 0$. Figure 11 shows how the smoothness of the extracted isosurfaces changes with variations in the value of the cut-ratio $\tau$. We can see that although the isosurface quality $Q$ of the isosurfaces has dropped slightly in Figure 11-b compared to Figure 11-a, the shape smoothness of the isosurfaces is more satisfactory in the former.

![Figure 11: Extracting isosurfaces from blocks of rubber and aluminum. (a) At cut-ratio $\tau = 0.01$ (b) At cut-ratio $\tau = 0.3$. Calculation time is 25 sec, Q shows classification qualities.](image1)

Figure 12 shows the isosurfaces extracted from a volumetric model of an automotive engine cylinder head consisting of steel (St), aluminum (Al) and an air background (Air). The average CT values of the three materials are $m_{St} = 10,000$, $m_{Al} = 3,600$ and $m_{Air} = 0$. Figure 12 shows a satisfactory shape with high-quality isosurfaces. Since this CT volumetric data is very large of $500 \times 700 \times 231$ voxels, using the primer classification of restricted region growing helped to reduce the graph to 14.3% of the total volumetric model size.

![Figure 12: Isosurfaces of a car engine part. The cut-ratio $\tau = 0.2$. The surface of the steel contains 331,928 triangles, and the surface of the aluminum contains 4,160,075 triangles. Calculation time is 560 sec, Q shows classification qualities.](image2)

Figure 13 shows the extracted isosurfaces of a mechanical part consisting of steel (St), rubber (Ru) and an air background (Air), their CT values are $m_{St} = 11,000$, $m_{Ru} = 1,600$ and $m_{Air} = 0$. This part functions as a damper to reduce vibration in automotive engines. Note that $m_{Ru}$ is noticeably low compared to $m_{St}$, which will cause misclassification of rubber voxels as background voxels of air using a former low-sensitivity classification method in [Fujimori and Suzuki 2005].

![Figure 13: Isosurface of vibration damper. (a) Extracted isosurfaces at cut-ratio $\tau = 0.05$ using our classification method. The surface of the steel contains 836,716 triangles, and the surface of the rubber contains 856,939 triangles. Calculation time is 200 sec, Q shows classification qualities. (b) Extracted surface using voxel-based classification method in [Fujimori and Suzuki 2005].](image3)
Figure 14 presents the results of extracting isosurfaces in the form of a volumetric model made of four materials, steel (St), aluminum (Al), rubber (Ru) and background (Air). The CT values are $m_{St} = 10,000$, $m_{Al} = 3,900$, $m_{Ru} = 2,100$ and $m_{Air} = 0$. This example shows the validity of our method to classify an n-material volumetric model and generate high-quality isosurfaces.

In general, we consider the isosurface quality is very good for values of $Q$ over 95%. However in the presented examples we can notice that although the quality is rather low for some surfaces, the geometric shape of the object is satisfactory. This trade-off between the isosurface quality and its smoothness can be controlled by defining a proper cut-ratio $\tau$ as shown in Figure 11.

![Figure 14: Isosurface of a block with three materials (steel - aluminum - rubber). The surface of the steel contains 51,508 triangles, the surface of the aluminum contains 54,304 triangles, and that of the rubber contains 54,698 triangles. Calculation time is 59 sec, Q shows classification qualities.](image)

$Q(Air, Ru) = 89.00\%$

$Q(Air, Al) = 98.45\%$

$Q(Air, St) = 100\%$

$Q(Ru, Al) = 91.41\%$

$Q(Ru, St) = 95.55\%$

$Q(Al, St) = 96.17\%$

In Figure 15 we show the effect of the noise in the CT model on the result of our algorithm. We define the CT data noise as the coefficient of variation $c_v$ which measures the dispersion of probability distribution of a homogeneous sample of the CT data, and it is defined as the ratio of the standard deviation $\sigma$ to the mean $\mu$ follows:

$$c_v = \frac{\sigma}{\mu}.$$  

The coefficient of variation is reported in percentage (%) by multiplying the above calculation by 100. Figure 15 shows a scaled portion of the mesh shown in Figure 12. Notice that increasing the noise does not affect much the result of material classification as the classification quality does not change much for the three cases, also CPU time is almost the same for the classification stage; however the smoothness of the final surface has less smoothness as the noise increase in the CT data, and CPU time for isosurfacing increase directly to the noise due to having more triangles in the noisy surface.

<table>
<thead>
<tr>
<th>Case</th>
<th>CPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>526</td>
</tr>
<tr>
<td>(b)</td>
<td>534</td>
</tr>
<tr>
<td>(c)</td>
<td>550</td>
</tr>
</tbody>
</table>

Figure 15: The effect of noise on the result of classification ($\tau = 0.2$) and isosurfacing ($a$) $c_v = 1.7\%$ ($b$) $c_v = 5\%$ ($c$) $c_v = 12\%$.

Our method works for extracting isosurfaces of materials that have distinct CT values, however it may fail to classify voxels belonging to different materials of rather similar CT values especially if the data is very noisy. Although the CT value of each material does not affect the result of the classification, however by increasing the number of materials $n$ in the volumetric data, the quality of the extracted meshes may be slightly degraded due to the accumulative error in voxel classification which will affect materials to be classified in later iterations of Graph-cut. We also plan to investigate improvement of our method to enable automatic prediction of the value of the cut-ratio $\tau$ through geometrical analysis of the shape at the junction areas between more than two materials. This will be done by defining an adaptive cut-ratio that changes according to the shape of each junction area.

### Acknowledgments

Authors would like to thank for anonymous reviewers for valuable comments and suggestions, Michikawa Takashi for valuable discussions, and Fujimori Tomoyuki for providing the program of mesh isosurface generation. CT data is courtesy of Toyota Motor Corporation.

### References


