SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •
Special Focus

September 2013, Vol. 56 092108:1–092108:9
doi: 10.1007/s11432-013-4987-2

Edge detection based multi-material interface extraction on industrial CT volumes

OHTAKE Yutaka* & SUZUKI Hiromasa*

Research Center for Advanced Science and Technology, The University of Tokyo, Japan

Received May 2, 2013; accepted July 26, 2013

Abstract X-ray CT scanning is recently becoming a popular technology for industrial applications as well as medical ones. Since the geometrical accuracy is often important for industrial applications, more precise methods for processing CT volumes are required. This paper proposes a method for extracting the multi-material interfaces on CT volumes obtained by industrial CT scanners. Instead of extracting isosurfaces we detect *edge-points*, at which the norm of the volume gradient takes a local maximum in the gradient direction, and then interpolate the points as the material interfaces represented by the zero-level of compounded implicit functions. In order to achieve a robust material-identification, multilabel graph-cut is utilized in our method. Using edge-points, we can reduce the inaccuracy caused by CT scanning artifacts.

Keywords CT volume, multi-material interface, segmentation, edge extraction

Citation Ohtake Y, Suzuki H. Edge detection based multi-material interface extraction on industrial CT volumes. Sci China Inf Sci, 2013, 56: 092108(9), doi: 10.1007/s11432-013-4987-2

1 Introduction

X-ray CT (Computed Tomography) scanning is a convenient for capturing the geometry of a real-world object with complex topology, as demonstrated in Figure 1. Because of the convenience, CT scanning is recently becoming a popular technology for industrial applications.

It is generally said that the geometrical accuracy of the 3D shapes obtained by CT scanning is not so high as one of surface scanning, though the accuracy is often important for industrial applications. Thus, more researches on the precise geometry processing of CT scanned data (CT volumes) are now desired. In this paper, we focus on extracting accurate interfaces of multi-materials on industrial CT volumes.

• **Conventional interface extraction.** Conventionally, the interfaces of the different materials on a CT volume are simply extracted as isosurfaces (thresholding of CT values) [1]. However, this simple extraction is not robust to CT scanning artifacts [2]. As shown in Figure 2, a blurring effect prevents a correct extraction of multi-material interfaces with thresholding because intermediate values exist between high and low values. Ideally, CT values are constant for an object with a uniform material, but this is not true in real situations. If CT values have oscillations as shown in Figure 3, then the isosurface has distortion.

*Corresponding author (email: yu-ohtake@den.rcast.u-tokyo.ac.jp, suzuki@den.rcast.u-tokyo.ac.jp)

• **Our approach.** To solve the problem on the conventional interface extraction, we propose to define multi-material interfaces as a set of *edge-points* at which the norm of the volume gradient takes a local maximum along the gradient direction. Mathematically, the following two conditions are satisfied at edge-points:

$$\frac{\partial\,|\boldsymbol{g}(\boldsymbol{x})|}{\partial\,\boldsymbol{e}(\boldsymbol{x})}=0,\quad \frac{\partial^2\,|\boldsymbol{g}(\boldsymbol{x})|}{\partial\,\boldsymbol{e}(\boldsymbol{x})^2}<0, \tag{1}$$

where a scalar field $I(\boldsymbol{x})$ corresponds to the CT values on a volume at point $\boldsymbol{x}$, gradient $\boldsymbol{g}(\boldsymbol{x})=\nabla I(\boldsymbol{x})$, and $\boldsymbol{e}(\boldsymbol{x})$ is the normalized gradient $\boldsymbol{g}(\boldsymbol{x})/|\boldsymbol{g}(\boldsymbol{x})|$. By using a set of edge-points, we can solve the problems of isosurfacing. In Figure 2, extra-interfaces do not appear on the interface between the white and black materials since only single layer of edge-points exist between the interface. In Figure 3, an interface without large distortion can be obtained because small oscillations of CT values do not move the edge-points.

In order to develop a novel method for extracting multi-material interfaces with interpolating edge-points, we combine the existing algorithms as building blocks; edge detection, data segmentation, surface reconstruction, and multi-material polygonization. In Section 2, we mention related work for our method. The detail of our method is described in Section 3. In Section 4, we show the effectiveness and discuss the problems.

2 Related work

In this section, we mention methods which are selected as the building blocks in our method and their related work.

• **Edge detection.** Edges on 2D images (scalar data sampled at a 2D regular grid) has intensively been studied in image processing. As introduced in [3], there are several definitions of image edges. Definition (1) used in our work was proposed by Canny [4], and one of the most popular method. Canny's edge detector is easy to extend to 3D images (volumes) [5]. In this work, we compute sub-voxel size accurate edge-points based on Canny's edge detector, and then interpolate the points after identifying the materials. The detail of our algorithm of edge detection is described in Subsection 3.2.

• **Data segmentation.** Data segmentation based on the graph based approach is widely used in both of image and geometry processing. The binary graph-cut [6] was proposed by Boykov et al., then they extended it for multilabel cases [7]. Because of its global optimization nature, the multilabel graph-cut can robustly segment low-quality data. In our work, CT volumes are typically corrupted by artifacts; thus we select the multilabel graph-cut for the material identifications as explained in Subsection 3.3.

• **Surface reconstruction.** Reconstructing a surface form scattered points in 3D has intensively been studied in geometry processing and still an active research topic [8,9] (see also references therein). The methods are utilized to interpolate the edge-points in our method. However, it is difficult to directly apply most of surface reconstruction methods because they are designed to output 2-manifolds, as while multi-material interfaces have junctions where more than three materials meet. For this reason, we first fit implicit functions [10] for individual materials, and then compound the set of functions to define multi-material interfaces. See Subsection 3.4 for the details.

• **Polygonization of multi-material interfaces.** Mesh generation (polygonization) of multi-material interfaces is an important issue for representing the complex objects, for example inside human bodies, assembled mechanical parts, and so on. Grid-based [11,12] and Delaunay-based [13,14] algorithms are well developed in geometric modeling. By applying these algorithms, our multi-material interface defined by a set of implicit functions can be polygonized. We choose grid-based algorithm [11] by directly using the grid of an input volume.

• **Extremal surfaces.** The interface defined by Equation (1) is a kind of extremal surface [15] which is defined by a pair of scalar and direction fields. A method for direct polygonization of extremal surfaces is proposed in [16]. For our purpose the method is not suited because extremal surfaces represent 2-manifolds with boundaries and junctions are not properly represented. This is the reason we once convert multi-material interfaces into a set of implicit functions.

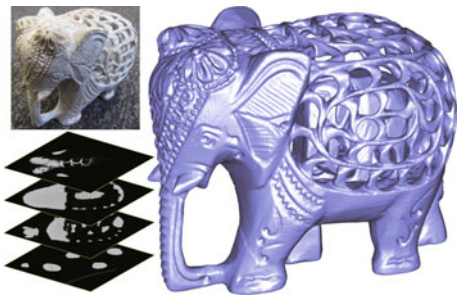

Figure 1 CT volume (left) and the extracted interface (right) of a topologically complex object.

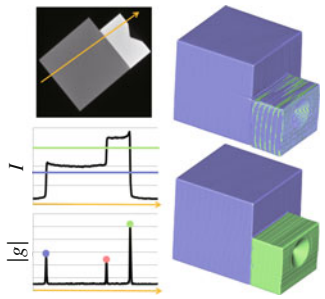

Figure 2 Extraction of multi-material interfaces. A cross-section of a volume (upper-left). Graph of CT value $I(\boldsymbol{x})$ with thresholds (middle-left) and the norm of gradient $|\boldsymbol{g}|$ with edge-points (bottom-left). Extracted interface with isosurfacing (top-right) and with interpolating edge-points (bottom-right).

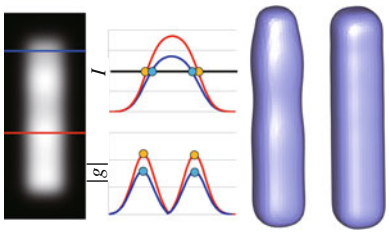

Figure 3 Extraction of an interface on a volume with an oscillation of CT values. Left: the cross-section of an artificial CT volume. Middle: Graphs of $I(\boldsymbol{x})$ and $|\boldsymbol{g}(\boldsymbol{x})|$ along red and blue lines. The circles indicate the points on the isosurface (top) and the edge-points (bottom). Right: an isosurface with distortion (left) and an interpolation of the edge-points (right).

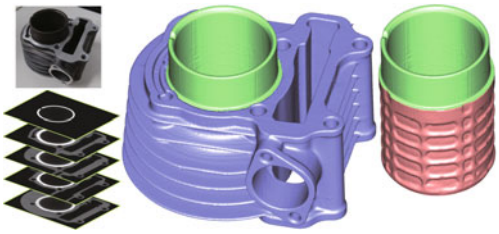

Figure 4 An input CT volume (left) and the output polygon mesh (middle and right) of our algorithm. The colors of the polygons mean the material identifications.

• **Isosurfacing based multi-material interface extraction.** Shammaa et al. [17,18] proposed methods to obtain multi-material interfaces on CT volumes using voxel based binary graph-cuts [6] and active contours [19]. Their problem setting is same as ours; however, their interfaces are located to isosurfaces, not edge-points. Thus, the problem in Figure 3 cannot be solved by their methods. To reduce the inaccuracy caused by CT scanning artifacts is the main motivation of our work.

3 Extraction of multi-material interfaces

The input of our algorithm is a CT volume consisting of scalar values sampled at 3D regular grid. We deal with the volume as a continuous scalar field $I(\boldsymbol{x})$ with a standard trilinear interpolation of the grid points. The left-bottom image of Figure 4 shows an example of input volume (a sequence of 2D images). The volume in this example contains three materials; air, aluminum, and steel, which corresponds to black, gray, and white, respectively.

The output is a polygon mesh approximating the material interfaces on the input volume. Since our algorithm can extract the interfaces of multi-materials, material identifications are assigned on the both side of each polygon. As shown in the example in the middle and right images of Figure 4, the coloring of the polygons means the material identifications. In this example, blue corresponds to the interface of air–aluminum, green steel–air, and red aluminum–steel.

Our algorithm requires a user to set the number of the materials N and their representative CT values with possible amount of oscillations. The ranges are denoted by $\{m_1 \pm \epsilon_1, m_2 \pm \epsilon_2, \ldots, m_N \pm \epsilon_N\}$. These values are typically known for industrial CT scanners because CT values are approximately proportional to the material densities, and scanned materials are limited to metals, parasitic, air, water, and so on. If such values are unknown, the user can easily find them by investigating the CT values on the cross-

sectional images of the volume.

In this section, we start with showing an overview of our algorithm in Subsection 3.1, and then the details of the algorithm are described in the following three subsections.

3.1 Algorithm overview

As shown in Figure 5, our algorithm consists of several steps. The followings summarize these steps.

Step 1: 2nd column of Figure 5. The edge-points are detected by shifting the edge-voxels which is obtained by the standard image processing method. The unoriented normals on the interfaces are also assigned to the edge-points.

Step 2: 3rd column of Figure 5. For each edge-point, two offset-points are computed in both directions of the normal. We then create a graph by setting the offset-points as the graph-nodes and the voxel connectivity as the graph-edges. With the graph, the multilabel graph-cut algorithm is applied in order to robustly identify which materials exist along the normal directions.

Step 3: 4th and 5th column of Figure 5. With the material identification along the normals, the oriented point-set of each material is obtained. For each material, we fit an implicit function with positive inside and negative outside. Compounding the set of the implicit functions gives us the material identifications at arbitrary points near the interface. We then polygonize the interface using a grid-based polygonization algorithm.

The implementation details for the above three steps are explained in Subsections 3.2, 3.3, and 3.4, respectively.

3.2 Detection of edge-points

• **Edge-voxels.** Canny's approach [4] is applied to the input volume in order to detect the edge-voxels. As shown in the left image of Figure 6, for each voxel $\boldsymbol{v}_i$ we compute the two intersection points $\boldsymbol{u}_i^+$ and $\boldsymbol{u}_i^-$ at which the line directed to the gradient direction $\boldsymbol{e}(\boldsymbol{v}_i)$ intersects with the boundary faces of $3 \times 3 \times 3$ local grid centered at $\boldsymbol{v}_i$. If the following conditions are satisfied, then the voxel $\boldsymbol{v}_i$ is regarded as an edge-voxel. $|\boldsymbol{g}(\boldsymbol{v}_i)| > |\boldsymbol{g}(\boldsymbol{u}_i^+)|$ and $|\boldsymbol{g}(\boldsymbol{v}_i)| > |\boldsymbol{g}(\boldsymbol{u}_i^-)|$. Similar to the scalar field $I(\boldsymbol{x})$, the gradient $\boldsymbol{g}(\boldsymbol{x})$ is treated as a continuous vector field with component-wise trilinear interpolation on the grid. At each grid point, we approximate the gradient operator $\nabla(\cdot)$ using the central difference formula [20].

To eliminate insignificant edge-voxels, we use hysteresis thresholding [4], which uses two thresholds; high threshold $T_{\boldsymbol{g}}$ and low threshold $\widetilde{T_{\boldsymbol{g}}}$. Starting from edge-voxels with $|\boldsymbol{g}(\boldsymbol{x})| > T_{\boldsymbol{g}}$ we trace the voxels $|\boldsymbol{g}(\boldsymbol{x})| > \widetilde{T_{\boldsymbol{g}}}$ using voxel connectivity. Then the traced edge-voxels are remained and other edge-voxels are eliminated. According to our experiments, it is possible to automatically set the two thresholds using the values $\{m_j\}$, as following equations. $T_{\boldsymbol{g}} = \min_{j_1,j_2;j_1 \neq j_2} |m_{j_1} - m_{j_2}|/(2\,d)$, $\widetilde{T_{\boldsymbol{g}}} = T_{\boldsymbol{g}}/10$, where d is the grid interval of the input volume.

• **Edge-points.** To obtain a sub-voxel size accurate edge-point $\boldsymbol{p}_i$, we use a quadratic polynomial interpolation to find the local maximum of $|\boldsymbol{g}(\boldsymbol{x})|$ on the line segment between $\boldsymbol{u}_i^-$ and $\boldsymbol{u}_i^-$. See the right image of Figure 6. With a simple calculation, the edge-point $\boldsymbol{p}_i$ is given by

$$\boldsymbol{p}_i = \boldsymbol{v}_i + \frac{\left(|\boldsymbol{g}(\boldsymbol{u}_i^+)| - |\boldsymbol{g}(\boldsymbol{u}_i^-)|\right)\,|\boldsymbol{u}_i^+ - \boldsymbol{u}_i^-|}{4\,\left(|\boldsymbol{g}(\boldsymbol{u}_i^+)| + |\boldsymbol{g}(\boldsymbol{u}_i^-)| - 2|\boldsymbol{g}(\boldsymbol{v}_i)|\right)}\,\boldsymbol{e}(\boldsymbol{v}_i).$$

• **Normals on interfaces.** To each edge-point $\boldsymbol{p}_i$, the unoriented normal $\boldsymbol{n}_i$ is assigned for the next steps in our algorithm. From Equation (1) the interfaces are defined as the zero-level set of $e(\boldsymbol{x})^t\,\nabla|\boldsymbol{g}(\boldsymbol{x})|$. Thus the unit normal $\boldsymbol{n}_i$ is obtained by $\boldsymbol{n}_i = \nabla\left(\boldsymbol{e}(\boldsymbol{p}_i)^t\,H(\boldsymbol{p}_i)\,\boldsymbol{e}(\boldsymbol{p}_i)\right)/|\nabla\left(\boldsymbol{e}(\boldsymbol{p}_i)^t\,H(\boldsymbol{p}_i)\,\boldsymbol{e}(\boldsymbol{p}_i)\right)|$, where $H(\boldsymbol{x})$ is a Hessian matrix $\nabla\boldsymbol{g}(\boldsymbol{x})$. To evaluate the above formula at arbitrary points, trilinear interpolation with the central differences is also used.

3.3 Identification of materials

• **Offset-points.** At an edge-point the CT value takes an intermediate value between the representative values for materials, which is caused by the blurring effect of CT scanning. Thus, it is difficult to identify

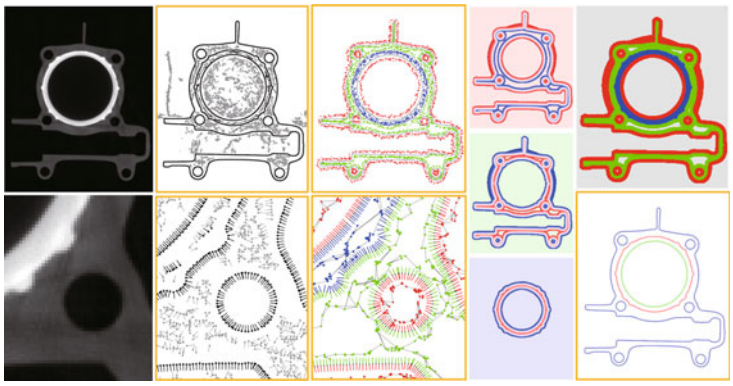

Figure 5 Algorithm overview. Algorithm steps are shown on the cross-section of a processed volume. 1st column: the CT value $I(\boldsymbol{x})$ (top) and its zoomed part (bottom) on a cross-section of the volume in Figure 4. 2nd column: edge-points $\{\boldsymbol{p}_i\}$ (top) and their normals $\{\boldsymbol{n}_i\}$ (bottom). 3rd column: offset points $\{\boldsymbol{q}_k\}$ colored with their material labels $\{l_k\}$ (top). We identify the labels using a graph structure (bottom). 4th column: the set of implicit functions $\{f_j(\boldsymbol{x})\}$ (red inside and blue outside) obtained from the oriented point-sets. The background colors correspond to the materials. 5th column: segmented grid-points according to the label function $l(\boldsymbol{x})$ (top). The polygonal approximation of the multi-material interfaces (bottom). Notice that grid-free data is enclosed by the orange rectangles. Others are data sampled at regular grids.

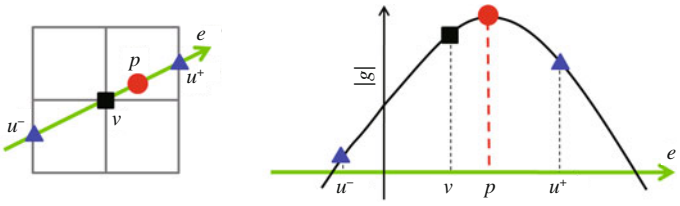

Figure 6 Computation of an edge-point. Left: Canny's edge detector. Right: quadratic polynomial interpolation of the norm of gradient.

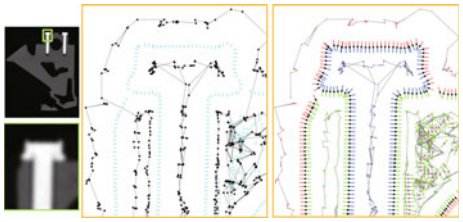

Figure 7 Material identification with multilabel graph-cut. Left: the cross-section of a CT volume. Middle and right: the graphs before and after applying the graph-cut.

the materials using the values close to edge-points. To solve this problem, we apply the following procedure for sampling stationary CT values.

We firstly compute two offset-points $\boldsymbol{q}_{(2i)}$ and $\boldsymbol{q}_{(2i+1)}$ for each edge-point $\boldsymbol{p}_i$. $\boldsymbol{q}_{(2i)} = \boldsymbol{p}_{(2i)} + dt\,\boldsymbol{n}_{(2i)}$, $\boldsymbol{q}_{(2i+1)} = \boldsymbol{p}_{(2i+1)} - dt\,\boldsymbol{n}_{(2i+1)}$, where dt is the length of displacement. We set dt to the half of grid interval $d/2$.

Then, the following gradient descent method is applied to converge the offset-points $\{\boldsymbol{q}_k\}$ $(k = 1, 2, \ldots, 2\,i, 2\,i+1, \ldots)$ to the local minima of $|\boldsymbol{g}(\boldsymbol{x})|$. $\boldsymbol{q}_k \leftarrow \boldsymbol{q}_k - dt\,\nabla|\boldsymbol{g}(\boldsymbol{q}_k)|/|\nabla|\boldsymbol{g}(\boldsymbol{q}_k)||$. The above update rule is terminated if the dot product of the current and previous displacement vectors is negative.

• **Multilabel graph-cut.** At almost offset-points in $\{\boldsymbol{q}_k\}$, we can sample stationary CT values to identify the materials. However, there still exist unconfident points caused by CT scanning artifacts. For robust identification of the materials at such points, we adapt multilabel graph-cut method [7].

Our graph-nodes are $\{\boldsymbol{q}_k\}$, and a set of graph-edges G are reflecting voxel connectivity of the edge-points $\{\boldsymbol{p}_i\}$. For each of two edge-points $\boldsymbol{p}_{i_1}$ and $\boldsymbol{p}_{i_2}$ whose corresponding voxels are connected with 26-neighbors, we add the following two graph-edges into G.

$$G \subset \begin{cases} \{(2i_1, 2i_2), (2i_1+1, 2i_2+1)\}, & \text{if } \boldsymbol{n}_{i_1}^t\,\boldsymbol{n}_{i_2} > 0, \\ \{(2i_1, 2i_2+1), (2i_1+1, 2i_2)\}, & \text{if } \boldsymbol{n}_{i_1}^t\,\boldsymbol{n}_{i_2} < 0. \end{cases}$$

The above rule means two pairs of offset-points are connected by taking into account the offset directions. An example of the graph is shown in the middle image of Figure 7.

By setting the unknown variables as labels $\{l_k\}$ $(l_k \in \{1, \ldots, N\})$, our multilabel graph-cut is formulated as a minimization problem of the following cost function $E(\{l_k\})$. $E(\{l_k\}) = \sum_k D_{l_k}(\boldsymbol{q}_k) +$

$\lambda \sum_{(k_1,k_2)\in G l_{k_1}\neq l_{k_2}} S(\boldsymbol{q}_{k_1}, \boldsymbol{q}_{k_2})$, where D and S are the node-cost and edge-cost, respectively. The parameter λ adjusts the balance of the two cost functions.

The node-cost $D_j(\boldsymbol{q}_k)$ should be small in the case where the graph-node $\boldsymbol{q}_k$ tends to be j-th material. If the CT value $I(\boldsymbol{q}_k)$ is in the range for j-th material and the norm of gradient $|\boldsymbol{g}(\boldsymbol{q}_k)|$ is low, we set the cost as a small value. Otherwise the cost is set to a large value.

$$D_j(\boldsymbol{q}_k) = \begin{cases} \min\left(|\boldsymbol{g}(\boldsymbol{q}_k)|, T_{\boldsymbol{g}}\right), & \text{if } I(\boldsymbol{q}_k) \in [m_j \pm \epsilon_j]. \\ T_{\boldsymbol{g}}, & \text{otherwise.} \end{cases}$$

The edge-cost $S(\boldsymbol{q}_{k_1}, \boldsymbol{q}_{k_2})$ is set to a large value if the two CT values $I(\boldsymbol{q}_{k_1})$ and $I(\boldsymbol{q}_{k_2})$ are close. To make S as the same dimension with D, we divide the CT value difference by the distance of the two nodes. $S(\boldsymbol{q}_{k_1}, \boldsymbol{q}_{k_2}) = \max\left(T_{\boldsymbol{g}} - |I(\boldsymbol{q}_{k_1}) - I(\boldsymbol{q}_{k_2})|/|\boldsymbol{q}_{k_1} - \boldsymbol{q}_{k_2}|, 0\right)$.

As shown in the right image of Figure 7, the labels $\{l_k\}$ are decided after minimizing E. All results in this paper are obtained by setting $\lambda = 1$. To minimize the cost E, we use the publicly available software *gco* [21].

3.4 Polygonization of multi-material interfaces

• **Oriented point-sets.** The label $l_{(2i)}$ (or $l_{(2i+1)}$) indicates which material exists toward the diction $+\boldsymbol{n}_i$ (or $-\boldsymbol{n}_i$) from the edge-point $\boldsymbol{p}_i$. Using the labels, we can find a point-set equipped with the oriented normals for each materials. Let us denote by P_j the oriented point-set for the jth material.

$$P_j \in \begin{cases} (\boldsymbol{p}_i, +\boldsymbol{n}_i), & \text{if } j = l_{(2i)} \text{ and } j \neq l_{(2i+1)}, \\ (\boldsymbol{p}_i, -\boldsymbol{n}_i), & \text{if } j \neq l_{(2i)} \text{ and } j = l_{(2i+1)}. \end{cases}$$

Note that we do not use the edge-point $\boldsymbol{p}_i$ with the same labels in both sides $l_{(2i)} = l_{(2i+1)}$, which is shown as the points colored with magenta in the right image of Figure 7.

• **Fitting implicit functions.** As shown in Figure 8, implicit functions with positive inside are computed for all materials, and then the functions are compounded to define the multi-material interfaces.

We use the partition of unity for implicit functions [10] to compute the function $f_j(\boldsymbol{x})$ for the j-th material. $f_j(\boldsymbol{x}) = \sum_{(\boldsymbol{p}_i,\widetilde{\boldsymbol{n}_i})\in P_j} w(|\boldsymbol{x} - \boldsymbol{p}_i|)\,(\widetilde{\boldsymbol{n}_i}^t(\boldsymbol{x} - \boldsymbol{p}_i))/\sum_{(\boldsymbol{p}_i,\widetilde{\boldsymbol{n}_i})\in P_j} w(|\boldsymbol{x} - \boldsymbol{p}_i|)$, where the weight function $w(r)$ is selected as the singular weight function [22] ($\lim_{r\to 0} w(r) = \infty$) to achieve interpolation of the points in P_j. The local support size of $w(r)$ is fixed at $3\,d$.

As introduced in the multi-material level-set methods [20], we simply compound the implicit functions by taking the maximum. The material identification at an arbitrary point $\boldsymbol{x}$ is given by $l(\boldsymbol{x}) = \mathrm{argmax}_j\, f_j(\boldsymbol{x})$.

• **Polygonization.** Since we already have the 3D grid as the input volume, we polygonize the interface using a grid-based polygonization algorithm called Dual Contouring [11]. To compute the intersection point of a grid-edge with the interfaces, trilinear interpolation is used as proposed in [23]. We denote two grid-points connected with a grid-edge by $\boldsymbol{v}_{i_1}$ and $\boldsymbol{v}_{i_2}$, and the two different materials by $j_1 = l(\boldsymbol{v}_{i_1})$, $j_2 = l(\boldsymbol{v}_{i_2})$. The intersection point is computed as

$$\frac{(f_{j_2}(\boldsymbol{v}_{i_2}) - f_{j_1}(\boldsymbol{v}_{i_2}))\ \boldsymbol{v}_{i_1} + (f_{j_1}(\boldsymbol{v}_{i_1}) - f_{j_2}(\boldsymbol{v}_{i_1}))\ \boldsymbol{v}_{i_2}}{(f_{j_1}(\boldsymbol{v}_{i_1}) - f_{j_2}(\boldsymbol{v}_{i_1})) + (f_{j_2}(\boldsymbol{v}_{i_2}) - f_{j_1}(\boldsymbol{v}_{i_2}))}.$$

Dual Contouring gives us a topologically correct polygonal approximation at the point meeting with three materials, as demonstrated in the bottom-right image of Figure 8.

4 Results and discussion

In this section, we demonstrate the effectiveness of our method by comparing our results with isosurfaces. Then we clarify the limitations of the current method, and possible future directions to solve the problems.

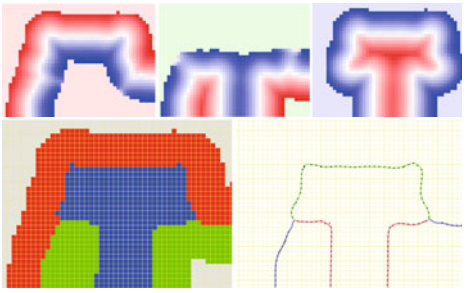

Figure 8 Top: implicit functions. Bottom: material identification (left) and polygonal approximation (right).

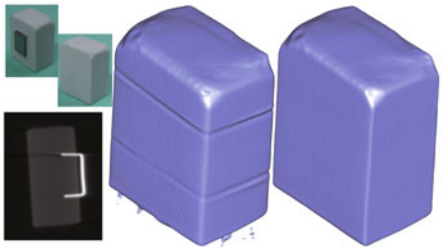

Figure 9 A demonstration of robustness of our method to CT scanning artifacts. Left: photos of front- and back-side of an eraser with a staple (top) and a cross-section of the CT scanning (bottom). Middle: the back-side face of the eraser obtained with isosurfacing. Right: the face obtained with interpolating the edge-points.

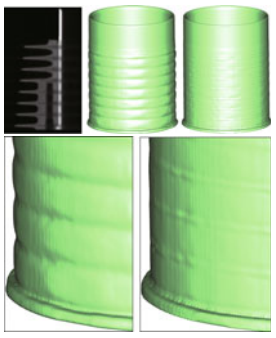

Figure 10 A demonstration of robustness of our method to CT scanning artifacts. Top-left: the cross-section of a CT volume with an artifacts on a liner (white region). Top-middle and bottom-left: the liner obtained by isosurfacing. Top-right and bottom-right: the liner obtained by our method.

• **Robustness to CT scanning artifacts.** As mentioned in Section 1 our main purpose is to increase geometrical accuracy of the interfaces on a CT volume with scanning artifacts. Figures 9, 10, and 11 demonstrate the robustness to CT scanning artifacts.

In Figure 9, an eraser with a staple was used for a test object. We can observe an artifact caused by a high-density of the staple made of metal. See the bottom-left image for a cross-section. Because of the artifact, we failed to obtain the back-side of eraser as a flat surface with isosurfacing as shown in the middle image. In contrast, our proposed method give us almost flat surface as shown in the left image.

Figure 10 shows the case for an object with more complex geometry. Too long paths of X-rays along the engine fins cause an artifact on the liner (white region on the CT volume). The liner should be a cylindrical shape without bumps, but the result of isosurfacing in the top-middle and bottom-left has a large bumps. On our result shown in the right images, the size of the dumps are much smaller. See the left silhouette lines on the bottom images.

We also compare our method and isosurfacing in terms of robustness to blurring effects in Figure 11. In order to simulate the blurring on CT volumes, we applied an intensive smoothing with Laplacian (iterative averaging of values at neighboring voxels) to the volume shown in Figure 1, then the interface is extracted. With isosurfacing, thin-parts on the back of the elephant are fragmented as shown in the left images. On our result shown in the right images, such a shrinking problem is not observed.

• **Performance.** Figures 12 and 13 show the results for multi-material interfaces with complex mechanical objects. Similar to the previous methods [18], we succeed in obtaining multi-material interfaces.

Our current experimental implementation is far from optimized; however, the computation is reasonably fast. For example, with a standard laptop PC, it takes about 2 minutes for the volume in Figure 1 consisting of $280 \times 260 \times 215$ voxels, and 5 minutes for the volume in Figure 13 consisting of $1024 \times 1024 \times 240$ voxels.

• **Limitations and future work.** We achieve accurate interface extraction which is robust to CT scanning artifacts. However, our method cannot deal with highly corrupted volumes as shown in Figure 14.

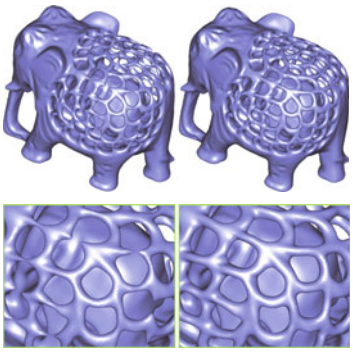

Figure 11 A test of robustness to blurring effects using a smoothed volume. Left: an intensive smoothing leads to shrinking of isosurfaces. Right: our method can keep thin-parts without shrinking.

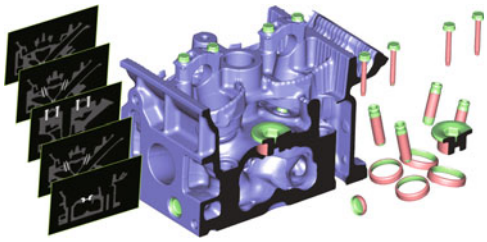

Figure 12 The multi-material interfaces on an engine-head obtained with our method.

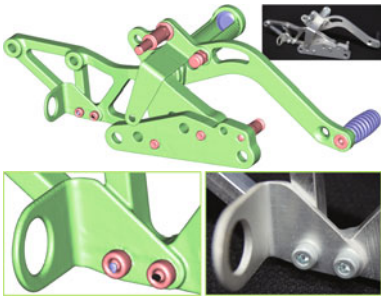

Figure 13 The multi-material interfaces on a pedal parts of motorbike obtained with our method.

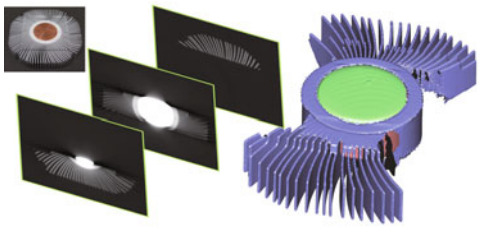

Figure 14 A difficult case for the proposed method because of the metal-artifact caused by the copper part in the center of a CPU heatsink.

In this example, X-rays passing through the copper part of CPU heatsink are almost vanished; thus the aluminum fins located around the center planes are not clearly scanned, and are failed to be extracted by the proposed method. To deal with such a difficult case, artifact reduction techniques [24,25] should be applied to sets of X-ray projection images before reconstructing CT volumes.

Since we use high-order derivatives, our methods is sensitive to small oscillations of CT values (noise) which are different from CT artifacts. In some results, for example Figure 9, we applied pre-smoothing of CT values for eliminating the noise. As demonstrated in Figure 12, smoothing CT values do not affect so much the accuracy of the extracted shape.

Our method is computationally inexpensive than the voxel-based method since we mainly process the edge-points on interfaces. However, processing only points on interfaces cannot guarantee the water-tightness to the output polygon meshes. For example, as shown in the bottom image of Figure 13, small topological problems might happen. One possible solution is to replace more robust surface reconstruction algorithm in the step of implicit surface fitting. Poisson surface reconstruction [26] is a good candidate. Further, fitting implicit functions for individual materials is obviously not a good idea. Considering all materials together in fitting functions is an interesting research topic. Recently a sophisticated fitting algorithm for multi-material implicit functions is proposed in [27].

By comparing the real object and our polygon mesh shown in the bottom image of Figure 13, the sharp features is clearly smoothed out, which is caused by blurring effects on CT scanning process. Solving this sharp feature problem is highly desired in industrial applications.

Acknowledgements

We are indebted to Masaki Moriguchi for a fruitful discussion. The CT volumes are courtesy of Toyota Motor Corporation (Figure 12) and RIKEN (Figures 4 and 10).

References

1 Lorensen W E, Cline H E. Marching cubes: a high resolution 3D surface construction algorithm. Comput Graph, 1987, 21: 163–169
2 Hsieh J. Computed Tomography Principles, Design, Artifacts, and Recent Advances. SPIE Press, 2003
3 Ziou D, Tabbone S. Edge detection techniques—an overview. Int J Pattern Recognit Image Anal, 1998, 8: 537–559
4 Canny J. A computational approach to edge detection. IEEE Trans Patt Anal Mach Intell, 1986, 8: 679–714
5 Lohmann G. Volumetric Image Analysis. Wiley, 1998
6 Boykov Y, Jolly M P. Interactive graph cuts for optimal boundary & region segmentation of objects in n-D images. In: International Conference on Computer Vision, Vancouver, 2001. 105–112
7 Delong A, Osokin A, Isack H N, et al. Fast approximate energy minimization with label costs. Int J Comput Vis, 2012, 96: 1–27
8 Mullen P, Goes F, Desbrun M, et al. Signing the unsigned: robust surface reconstruction from raw pointsets. Comput Graph Forum, 2010, 29: 1733–1741
9 Berger M, Levine J A, Nonato L G, et al. An End-to-End Framework for Evaluating Surface Reconstruction. SCI Technical Report UUSCI-2011-001. SCI Institute, University of Utah, 2011
10 Ohtake Y, Belyaev A, Alexa M, et al. Multi-level partition of unity implicits. ACM Trans Graph, 2003, 22: 463–470
11 Ju T, Losasso F, Schaefer S, et al. Dual contouring of Hermite data. ACM Trans Graph, 2002, 21: 339–346
12 Anderson J C, Garth C, Duchaineau M A, et al. Smooth, volume-accurate material interface reconstruction. IEEE Trans Vis Comput Graph, 2010, 16: 802–814
13 Boltcheva D, Yvinec M, Boissonnat J-D. Feature preserving delaunay mesh generation from 3D multi-material images. In: Proceedings of the Symposium on Geometry Processing. Switzerland: Eurographics Association Aire-la-Ville, 2009. 1455–1464
14 Dey T K, Janoos F, Levine J A. Meshing interfaces of multi-label data with delaunay refinement. Eng Comput, 2012, 28: 71–82
15 Amenta N, Kil Y J. Defining point-set surfaces. ACM Trans Graph, 2004, 23: 264–270
16 Li R S, Liu L, Phan L, et al. Polygonizing extremal surfaces with manifold guarantees. In: Proceedings of 14th ACM Symposium on Solid and Physical Modeling. New York: ACM, 2010. 189–194
17 Shammaa H M, Ohtake Y, Suzuki H. Segmentation of multi-material ct data of mechanical parts for extracting boundary surfaces. Comput-Aided Des, 2010, 42: 118–128
18 Shammaa H, Suzuki H, Ohtake Y. Creeping contours: a multilabel image segmentation method for extracting boundary surfaces of parts in volumetric images. J Comput Inf Sci Eng, 2011, 11: 011007
19 Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. Int J Comput Vis, 1987, 1: 321–331
20 Sethian J. Level Set Methods and Fast Marching Methods. Cambridge: Cambridge University Press, 1999
21 Veksler O, Delong A. Gcoptimization—software for energy minimization with graph cuts. http://vision.csd.uwo.ca/code/#Multi-label_optimization.
22 Renka R J. Multivariate interpolation of large sets of scattered data. ACM Trans Math Softw, 1988, 14: 139–148
23 Feng P P, Ju T, Warren J. Piecewise tri-linear contouring for multi-material volumes. In: Proceedings of 6th International Conference on Advances in Geometric Modeling and Processing. Berlin/Heidelberg: Springer-Verlag, 2010. 43–56
24 Krumm M, Kasperl S, Franz M. Reducing non-linear artifacts of multi-materialobjects in industrial 3D computed tomography. NDT E Int, 2008, 41: 242–252
25 Amirkhanov A, Heinzl C, Reiter M, et al. Projection-based metal-artifact reduction for industrial 3D x-ray computed tomography. IEEE Trans Vis Comput Graph, 2011 17: 2193–2202
26 Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: Proceedings of 4th Eurographics Symposium on Geometry Processing. Switzerland: Eurographics Association Aire-la-Ville, 2006. 61–70
27 Yuan Z, Yu Y, Wang W. Object-space multiphase implicit functions. ACM Trans Graph, 2012, 31: 114