

Detection and Reconstruction of an Implicit Boundary Surface by Adaptively Expanding A Small Surface Patch in a 3D Image

Lisheng Wang, Pai Wang, Liuhan Cheng, Yu Ma, Shenzhi Wu, Yu-Ping Wang, and Zongben Xu

Abstract—In this paper we propose a novel and easy to use 3D reconstruction method. With the method, users only need to specify a small boundary surface patch in a 2D section image, and then an entire continuous implicit boundary surface (CIBS) can be automatically reconstructed from a 3D image. In the method, a hierarchical tracing strategy is used to grow the known boundary surface patch gradually in the 3D image. An adaptive detection technique is applied to detect boundary surface patches from different local regions. The technique is based on both context dependence and adaptive contrast detection as in the human vision system. A recognition technique is used to distinguish true boundary surface patches from the false ones in different cubes. By integrating these different approaches, a high-resolution CIBS model can be automatically reconstructed by adaptively expanding the small boundary surface patch in the 3D image. The effectiveness of our method is demonstrated by its applications to a variety of real 3D images, where the CIBS with complex shapes/branches and with varying gray values/gradient magnitudes can be well reconstructed. Our method is easy to use, which provides a valuable tool for 3D image visualization and analysis as needed in many applications.

Index Terms—Continuous implicit boundary surface (CIBS), zero-crossing surfaces, 3D edge detection, 3D reconstruction, surface/model generation, adaptive contrast detection, surface connectivity, context dependence

1 INTRODUCTION

IN many application areas, users usually explore and understand a 3D image by browsing a series of 2D sectional images. This way they can easily see the structure of interest in 2D sectional images and extract qualitative information. However, the 3D model of the structure, which is very important for understanding and quantitative analysis of the structure [1], [2], cannot be reconstructed. Therefore, it is necessary to develop 3D model reconstruction techniques suitable for the needs of these users. The techniques should request little human interventions so users with little computer skills can easily use it.

Adapting to users' habit of browsing 3D images, such a 3D model reconstruction technique may be described as follows: "it can reconstruct automatically the 3D model of a structure of interest from a 3D image as long as the structure is seen and specified (or designated) in a certain 2D sectional image". The 3D model of a structure is a connected

boundary surface in a 3D image. Thus, when users see a structure of interest in a 2D sectional image, they can easily specify the structure or its boundary surface by drawing a specific small closed region (on the 2D image) containing a salient boundary curve segment of the structure [3], [4]. With this operation, a small piece of boundary surface patch of the structure is actually determined implicitly in a small 3D image region, and the entire boundary surface is specified or designated by the surface connectivity. Therefore, we plan to develop such a 3D model reconstruction technique that can automatically trace and reconstruct the entire boundary surface from the 3D image as long as a small boundary surface patch is known. To our knowledge, little research exists on such technique, which is the focus of this paper.

We mainly focus on boundary surfaces where gray intensities have certain change, instead of blurry or obscure boundary surfaces. Without loss of generality, we assume that gray intensities of the 3D image change slightly or gradually in the structure and background regions. Since a boundary surface in a 3D image can be regarded as a specific zero-crossing surface (or patch) of the Laplacian function of the 3D image [23], [30], it is called a continuous implicit boundary surface (CIBS) in this paper. When a structure of interest and its background have varying intensity values, the corresponding boundary surface patches usually have varying gray values and gradient magnitudes, and the whole boundary surface cannot be well represented by any isosurface [28], [29], [30], [31]. Therefore, boundary surfaces in 3D images are some special continuous implicit surfaces differing from implicit isosurfaces.

Our proposed method is similar to the archaeological analysis in the search for the whole CIBS of an object based

- L. Wang, P. Wang, L. Cheng, Y. Ma, and S. Wu are with the Institute of Image Processing and Pattern Recognition, Department of Automation, Shanghai Jiao Tong University, and with the Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai, 200240, P. R. China. E-mail: lswang@sjtu.edu.cn, gggird@gmail.com, flyc86@126.com, lmayu@163.com, wshen_zhi@sina.com.
- Y.-P. Wang is with the Department of Biomedical Engineering, Tulane University, New Orleans, Louisiana. E-mail: wyp@tulane.edu.
- Z. Xu is with Institute for Information and System Science and Research Center for Applied Mathematics, Xi'an Jiaotong University, Xi'an 710049, P. R. China. E-mail: zbxu@mail.xjtu.edu.cn.

Manuscript received 19 Dec. 2012; revised 20 Feb. 2014; accepted 24 Feb. 2014. Date of publication 13 Mar. 2014; date of current version 1 Nov. 2014.

Recommended for acceptance by L. Kobbelt.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TVCG.2014.2312015

on a known small boundary surface patch. We will extract the whole CIBS by expanding a small boundary surface patch in the 3D image gradually and adaptively. To accomplish this goal, we need to address the following challenges: detecting small local regions (especially all cubes) passed through by the CIBS, distinguishing boundary surface patches from false ones, and expanding the known boundary surface patch in the 3D image. CIBSes in different 3D images may have different gray values and gradient magnitudes. Further, a CIBS in a 3D image may have varying gradient magnitudes and complex shape and branches. Therefore, our proposed method should be adaptive to these variations.

Our method makes use of local, regional and global characteristics of boundary surfaces to recognize, trace and reconstruct a CIBS from a 3D image. An adaptive detection technique is introduced to detect boundaries, by combining the context dependence and the adaptive contrast detection approach motivated by the human vision system. A recognition approach is introduced to distinguish true boundary surface patches from false ones in different cubes. A hierarchical tracing strategy is used to expand the known boundary surface patch gradually in the 3D image. It first detects small regions that are around the known boundary surface patch and passed through by the CIBS, and then detects and traces cubes passed through by the CIBS from each of them. By integrating these different techniques, a high-resolution polygonal boundary surface model can be reconstructed automatically from a 3D image based on a known small boundary surface patch.

The rest of the paper is organized as follows. Section 2 reviews relevant works. Section 3 describes our method for detecting and reconstructing a CIBS in 3D images. Section 4 presents experimental results of our method in many real 3D images. Section 5 discusses some properties of our method, and the comparison with other related methods. Section 6 concludes the paper.

2 RELATED WORKS

In the last two decades, various techniques have been developed to reconstruct 3D models of structures in 3D images. They mainly include isosurface-extraction [5], [6], [7], [8], [9], [10], [11], [12], deformable surface [13], [14], [15], [16], 3D boundary detection [17], [18], [19], [20], [21], [22], [23], [24], [25], and surface reconstruction from contours [26], [27].

Isosurface-extraction technique has been studied widely, since marching-cube algorithm was proposed [5]. Examples include the algorithm for extracting high-quality or topologically-correct isosurface [7], [8], [9], [10], [11], [12], [57], [58], [59], and GPU-based isosurface techniques [50], [51]. The technique uses only one parameter (i.e., an isovalue) and can extract a closed isosurface model from 3D images. Thus, it will be used to reconstruct (or approximate) 3D models of structures in 3D images whenever possible [34], [35], [36], [37]. However, the isosurface-extraction technique has limitations, because boundary surfaces in many 3D images have varying gray levels and cannot be well represented by any isosurface with a fixed isovalue, as shown in [28], [29], [30], [31].

Instead of using an isosurface, different isosurface patches may be used to approximate a boundary surface with varying gray values [30], [31]. However, it is cumbersome to determine different isosurface patches in different local regions. Some approaches such as [52] apply image processing to correct intensity inhomogeneities in 3D images. However, for those structures of interest in a complex background or whose branches or sub-structures have varying gray values (e.g., the nerve cell in a 3D confocal microscopy image), their boundary surfaces can still not be well represented by any isosurface.

Various 3D edge detectors have been proposed to detect discrete digital boundaries (i.e., discrete surfaces formed by many discrete boundary voxels) from 3D images [17], [18], [19], [20], [21], [22], [23], [24], [25]. They can be divided into two classes: gradient-based detectors and zero-crossing-based detectors. The former marks a voxel whose gradient magnitude is larger than a predefined threshold as the boundary voxel [17], [18], [19], [20], [21], [22], while the latter regards a boundary surface as a zero-crossing surface patch in a new 3D image transformed from the original [23], [24], [25]. These two types of detectors usually cannot distinguish effectively the true boundary voxels from the false ones. In particular, zero-crossing fragments introduced by noise, image details and phantom edges are often falsely detected as boundaries [32], [33]. A high gradient threshold is usually used to separate the false boundary voxels from the true ones, but it is difficult to determine [38], [39], [40]. For those boundary surfaces with varying gradient magnitudes, they even cannot be well separated from the background by any threshold. In such cases, a gradient threshold function or local gradient thresholds is required [3], but its computation is a hard task.

Deformable surface technique has been developed to extract a closed boundary surface from a 3D image of poor quality [13], [14], [15], [16], for example, extracting an obscure or fuzzy boundary surface from MRI or Ultrasound images. It starts with an initial surface near a boundary surface, and then adjusts the surface iteratively until a good approximation to the boundary surface is obtained. In the technique, an appropriate initial surface is needed to set in the 3D image, and several appropriate parameters are needed to control the surface's evolution. Both operations are difficult tasks for most users. The technique cannot work if only a small boundary surface patch is known.

The technique for reconstructing boundary surfaces from contours can generate a smoothed surface model from contours [26], [27]. However, it is usually a complex task to extract closed boundary curves from 2D sectional images. In addition, the technique has limitations in reconstructing boundary surfaces with complex branches because of the branching problem of the contours and the difficulty in matching correct points on adjacent contours. The technique cannot work if only a small boundary surface patch is known.

Volume rendering may be used to visualize boundary surfaces in a 3D image by designing an appropriate transfer function that usually includes the gradient value as the parameter [53], [54], [55], [56]. However, it only generates a 2D projection image of a boundary surface instead of an explicit boundary surface model.

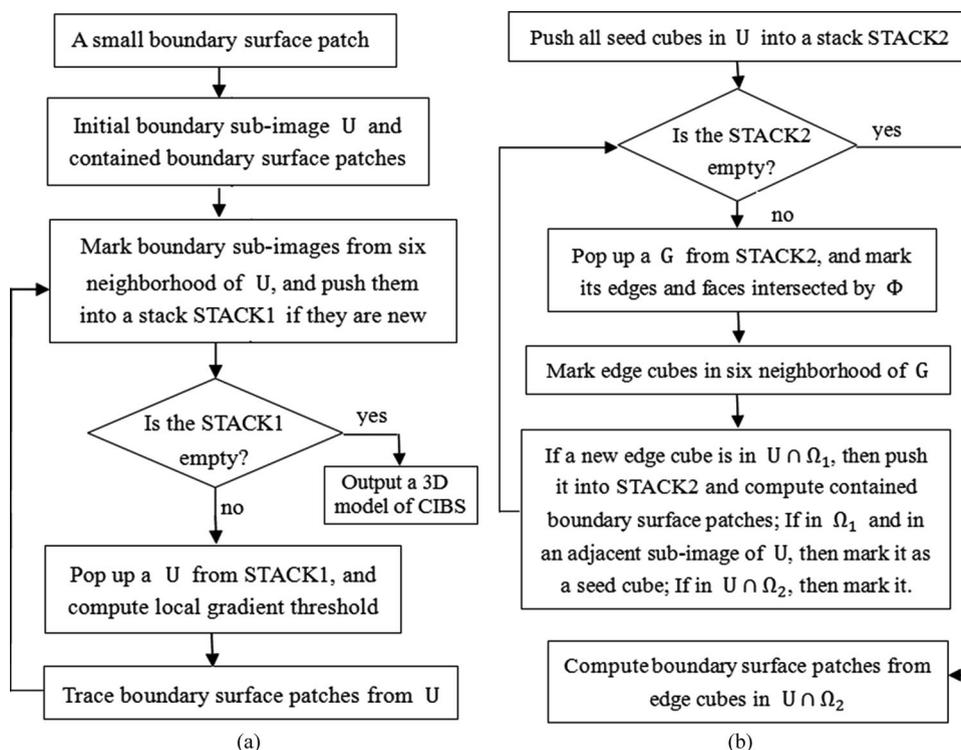


Fig. 1. The workflow of the hierarchical tracing framework. (a) The tracing of boundary sub-images in the 3D image. (b) The tracing of edge-cubes and boundary surface patches in each boundary sub-image U .

3 METHOD

A 3D image can be regarded as a discrete sampling of a three-dimensional function $h(x, y, z)$ in a 3D grid [49]. In the grid, eight adjacent grid points form a cube and all such cubes constitute the 3D sampling region. A boundary surface in the 3D image is a continuous implicit surface in the sampling region. For boundary detection, the 3D image $h(x, y, z)$ is usually smoothed by a Gaussian function $G(x, y, z, \sigma)$ with the scale σ to reduce noise and details in the image [23], [41], [42], [43]. Denote $f(x, y, z) = h(x, y, z) * G(x, y, z, \sigma)$, where $*$ represents the convolution between the two functions. Let $\|\nabla f(x, y, z)\|$ and $\nabla^2 f(x, y, z)$ be the gradient magnitude and the Laplacian of $f(x, y, z)$ [23], respectively. In this paper, unless otherwise specified, “isosurface” refers to the isosurface in $f(x, y, z)$.

Suppose that Φ is the CIBS of a structure of interest in $f(x, y, z)$. Then Φ is contained in some cubes of the 3D image—these cubes are called edge-cubes of Φ [30]. Φ has the following important local characteristics [23], [30], [43]:

(P1) Each point $(x, y, z) \in \Phi$ is a zero-crossing point of $\nabla^2 f(x, y, z)$, namely, $\nabla^2 f(x, y, z) = 0$. This means that Φ is a zero-crossing surface patch (the surface patch satisfying $\nabla^2 f(x, y, z) = 0$).

(P2) Each point in Φ has a relatively high gradient magnitude in its local neighborhood, and each small surface patch of Φ can be separated from the background with a certain local gradient threshold.

In addition to Φ , there are many other zero-crossing surface patches satisfying (P1) and (P2), such as ones induced by noise, image details or phantom edges [32], [33], and boundary surface patches of other uninteresting structures.

In this paper, zero-crossing surface patches satisfying (P1) and (P2) are classified into four types:

(VB1) Φ .

(VB2) Surface patches that are neither connected with Φ , nor pass through any edge-cubes of Φ (i.e., far from Φ).

(VB3) Surface patches or fragments that are not connected with Φ , but pass through certain edge-cubes of Φ (i.e., much close to Φ).

(VB4) uninteresting surface patches or fragments connected with Φ .

In general, surface patches belonging to (VB2)-(VB4) cannot be distinguished correctly from Φ based merely on the local characteristics (P1)-(P2). Therefore, we will find proper rules to distinguish them.

Φ might be of varying gradient magnitudes, and in such a case, the surface patches in different local regions need to be detected adaptively using different local gradient thresholds. For this purpose, the 3D image is divided into a set of non-overlapping small sub-images with equal volume, and gradient magnitudes of Φ should change slightly in each sub-image. Each sub-image is adjacent to six ones. The sub-images containing boundary surface patches of Φ are called boundary sub-images. A boundary sub-image usually contains many edge-cubes of Φ .

In this paper, a hierarchical tracing framework is built to expand a known small surface patch of Φ in the 3D image. The pipeline including several steps is put in Appendix, and the workflow is illustrated in Fig. 1. Fig. 1 shows that, while new boundary sub-images are traced from the known boundary surface patch that are closely surrounded, edge-cubes in each of them will be detected adaptively by the local gradient threshold and traced by the surface

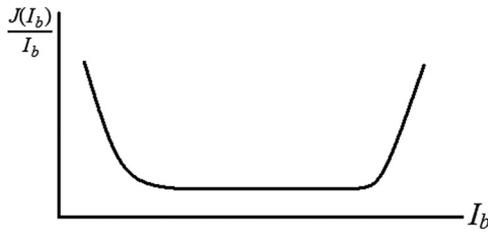


Fig. 2. The distribution of $\frac{J(I_b)}{I_b}$ along the luminance intensity I_b .

connectivity. The boundary surface patches in each new boundary sub-image are reconstructed by first tracing edge-cubes and then computing boundary surface patch in them (Fig. 1b). By repeating the procedure, the known boundary surface patch of Φ will be expanded gradually in the 3D image. Section 3.1 describes how to compute local gradient thresholds. In Section 3.2, edge-cubes of Φ are classified into two classes (Ω_1 and Ω_2) in order to recognize correctly boundary surface patches in each edge-cube. Section 3.3 shows how boundary surface patches of Φ are traced and reconstructed from a boundary sub-image when the local gradient threshold and seed cubes are known. Section 3.4 describes how the approaches in Sections 3.1, 3.2, 3.3 are integrated into a hierarchical framework.

3.1 Computation of Local Gradient Thresholds Adaptively in Different Boundary Sub-Images

The human vision system can detect boundaries with varying gradient magnitudes adaptively from various 2D images. In this paper, we will compute local gradient thresholds by mimicking the adaptive contrast detection mechanisms of the human visual system.

3.1.1 Visible Boundaries and the Adaptive Contrast Detection Mechanism

Generally, the human visual system is more sensitive to luminance difference than absolute luminance. The luminance difference forms a contrast pattern that conveys most of the visual information to the human vision, and the luminance contrast makes an object distinguishable from the background. Due to the visual adaptation of human, the same luminance difference might create different perceived brightness under different backgrounds. In other words, a luminance difference in a dim background may produce a large brightness difference, while produce a smaller brightness difference in a brighter background.

The just noticeable luminance difference (JNLD) refers to the smallest luminance difference possible for the human visual system to perceive. The JNLD varies with different background luminance. Let I_b be the average luminance of the background, then JNLD is a function of I_b , which is denoted by $J(I_b)$. In the case where small features are present in a large uniform background, the curve of $\frac{J(I_b)}{I_b}$ with respect to I_b may be plotted as in Fig. 2 [45], [46], [47]. This shows that the ratio $\frac{J(I_b)}{I_b}$ (e.g., weber contrast) can rapidly decrease/increase or can be nearly a constant when I_b has low/high luminance values or is in between. The curve of $\frac{J(I_b)}{I_b}$ can become more complex if the background varies differently. However, in this paper we only follow the adaptive contrast detection property described by $J(I_b)$ in Fig. 2.

By regarding gray intensities of 3D images as luminance, the boundaries in 3D images correspond to the locations where luminance differences exist. Therefore, we can detect all weak and strong visible boundaries from any 3D image by $J(I_b)$. Here, $J(I_b)$ is computed adaptively in different 3D images and in different sub-images of the same 3D image. $J(I_b)$ in Fig. 2 was defined in [48] as the following piecewise function of the gray intensity x , approximately:

$$J(x) = \begin{cases} \alpha \cdot d & 0 \leq x \leq d \\ \alpha \cdot x & d < x \leq e \\ \alpha \cdot e \cdot 10^{\frac{10}{256-x} - \frac{10}{256-e}} & e < x < 256, \end{cases} \quad (1)$$

where the gray regions $[0, d]$ and $[e, 255]$ are mapped to the low luminance and high luminance, respectively. When $x \in [0, d]$, the brightness is very low and $J(I_b)$ changes very slightly with x . So, $J(x)$ is approximately seen as a constant $C = \alpha \cdot d$. Here, α is selected as a constant around 0.25 rather than the weber contrast 0.02 because 0.02 is so small that many uninteresting weak boundaries will be detected from 3D images. 0.25 can facilitate the detection of boundaries of interest from 3D images while exclude many uninterested weak boundaries. When $x \in [e, 255]$, the brightness is very high and $J(x)$ is represented approximately by an exponential function. In this paper, we will use Equation (1) to compute $J(x)$ in the 3D image.

Let $N(x, y, z)$ be the average gray intensity in a local neighborhood of a point (x, y, z) . Then $V(x, y, z) = J(N(x, y, z))$ defines a 3D threshold function. Various visible boundaries (weak and strong) can be detected adaptively from different sub-images with the following rule:

$$\|\nabla f(x, y, z)\| \geq V(x, y, z). \quad (2)$$

In each edge-cube, the contained boundary surface patches have similar gray values, which fall in between gray values of the local structure and background in the cube. So, in each edge-cube, $N(x, y, z)$ is approximated by the mean of gray values of the contained boundary surface patch. In an edge-cube Q , let $q_i, i = 1, 2, \dots, l$ denote the points of intersection between boundary surface patches and all different edges of Q , and let $E(Q)$ be the average value of $f(q_i), i = 1, 2, \dots, l$. Then $J(E(Q))$ is used to approximate $V(x, y, z)$ in Q . This indicates that $V(x, y, z)$ can be approximated by a piece-wise constant function in the region closely enclosing Φ .

Visible boundaries characterize various image contents, including not only Φ but also surface patches belonging to (VB2)-(VB4). Therefore, we need to identify Φ from these visible boundaries detected, i.e., excluding those in (VB2)-(VB4).

3.1.2 Computation of Local Gradient Thresholds by Combining $V(x, y, z)$ and Context Dependence of Φ

In this section, $V(x, y, z)$ will be further adjusted by the context dependence of Φ , to exclude the surface patches or fragments in (VB4). Since gray intensities of the 3D image change gradually in the background and object regions, Φ and surface patches in (VB4) have the following properties:

(P3) Gradient magnitudes of Φ change slowly and gradually over adjacent small surface patches of Φ .

(P4) Gradient magnitudes of visible boundaries in (VB4) are usually much smaller than ones of local surface patches of Φ to which they are connected.

(P3) and (P4) in fact show the certain context dependence of Φ , by which surface patches or fragments in (VB4) can be excluded. So, we will compute local gradient thresholds by combining $V(x, y, z)$ with (P3) and (P4).

Suppose that U is a boundary sub-image whose boundary surface patch will be computed, while $U_i, i = 1, 2, \dots, m$ are boundary sub-images with known boundary surface patches and are adjacent to U . Below, we show how the local gradient threshold in U is computed. Let $T_U = \min\{\|\nabla f(p)\| : p \in U \cap \Phi\}$, $T_{U_i} = \min\{\|\nabla f(p)\| : p \in U_i \cap \Phi\}$, $i = 1, 2, \dots, m$ and $M_U = \frac{1}{m} \sum_{i=1}^m T_{U_i}$. Then, T_U and T_{U_i} are the infimums (e.g., the largest low bounds) of gradient magnitudes of boundary surface patches in U and U_i , respectively. By (P3), one may think that T_U has a small change compared to M_U . Thus, T_U may be estimated roughly as follows:

$$T_U \approx \kappa \cdot M_U, \quad (3)$$

where $\kappa \in (0, 1)$ is a constant describing the slow change of gradient magnitudes over Φ .

Gradient magnitudes of visible boundaries in U are larger than $V(x, y, z)$. Hence, gradient magnitudes of boundary surface patches in $\Phi \cap U$ are larger than the following function:

$$G_U(x, y, z) = \tau \cdot V(x, y, z) + (1 - \tau) \cdot \kappa \cdot M_U, (x, y, z) \in U, \quad (4)$$

where $\tau \in (0, 1)$ is a constant making a balance between $V(x, y, z)$ and M_U in $G_U(x, y, z)$. From our experiences, in many cases, κ can be selected as a value around 0.8 and τ can be selected as a value around 0.6. $G_U(x, y, z)$ will be regarded as the local gradient threshold in U . In a given edge-cube $Q \in U$, $G_U(x, y, z)$ can be written as follows:

$$G_Q = \tau \cdot J(E(Q)) + (1 - \tau) \cdot \kappa \cdot M_U. \quad (5)$$

Hence, $G_U(x, y, z)$ can be approximated by a piece-wise constant function in U . It will be used for the detection of the edges intersected by Φ from each cube in U . We only need to compute local gradient thresholds in boundary sub-images of Φ rather than the whole 3D image.

3.2 Classification of Edge-Cubes

The surface patches in (VB3) will pass through partial edge-cubes of Φ . So, the boundary surface patches in such edge-cubes should be recognized by a proper algorithm. To this end, edge-cubes of Φ will be classified into two classes with different recognition rules.

The boundary surface patches in each edge-cube are the zero-crossing surface patches. According to the marching-cube algorithm [5], when $\nabla^2 f(x, y, z) = 0$ intersects with an edge-cube, there are 14 different intersection cases, as shown in Fig. 3. In each case, either one connected or several disconnected zero-crossing surface patches are contained in

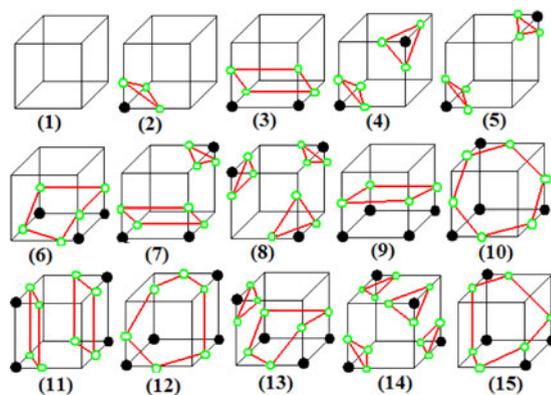


Fig. 3. Fourteen different intersection cases between the zero-crossing surface $\nabla^2 f(x, y, z) = 0$ and an edge-cube.

the edge-cube. Therefore, all edge-cubes of Φ can be divided into the following two classes:

Class Ω_1 : edge-cubes that contain only one connected zero-crossing patch (e.g., (2)(3)(6)(9)(10)(12)(15) in Fig. 3).

Class Ω_2 : edge-cubes that contain more than one connected zero-crossing patches (e.g., (4)(5)(7)(8)(11)(13)(14) in Fig. 3).

If an edge-cube Q belongs to Ω_1 , then the contained unique zero-crossing patch is a boundary surface patch and $E(Q)$ can be computed as the average of gray values of the intersection points between the zero-crossing surface patch and all different edges of Q . For each edge-cube in Ω_2 , topology-correct zero-crossing patches may be further constructed by [12]. Here, each zero-crossing surface patch is represented by a polygon that is generated by linking the points of intersection between the zero-crossing surface patch and different edges. If Q belongs to Ω_2 , some zero-crossing patches are boundary surface patches but some might be not. Surface patches belonging to (VB3) pass through certain edge-cubes in Ω_2 .

In the edge-cube Q , if an edge with two vertices p_1 and p_2 is intersected by Φ , then it is a zero-crossing edge or its two vertices p_1 and p_2 are a pair of zero-crossing points [30], [44], namely:

$$\nabla^2 f(p_1) \cdot \nabla^2 f(p_2) < 0. \quad (6)$$

In addition, the two vertices p_1 and p_2 have high gradient magnitudes in their local neighborhoods. Suppose that G_Q is the local gradient threshold in Q computed via Equation (5), then the two vertices p_1 and p_2 satisfy:

$$\|\nabla f(p_1)\| + \|\nabla f(p_2)\| \geq 2 \cdot G_Q. \quad (7)$$

Equations (6) and (7) give two important properties of the edge intersected by Φ .

In each edge-cube passed through by Φ , there are at least three edges that are intersected by Φ or satisfy Equations (6) and (7). If an edge-cube belongs to Ω_1 , then all edges satisfying Equations (6) and (7) are marked as the edges intersected by Φ . However, if an edge-cube belongs to Ω_2 , an edge satisfying Equations (6) and (7) might not be the edge intersected by Φ ; it might be the edge intersected by the surface patch in (VB3). In this case, detecting the intersected edges and computing boundary surface patches from each

edge-cube in Ω_2 become more complicated, which will be discussed in Section 3.3.

3.3 Tracing and Reconstruction of Boundary Surface Patches in a Boundary Sub-Image U

This section describes a technique to trace connected edge-cubes and reconstruct boundary surface patches from the boundary sub-image U , when $G_U(x, y, z)$ and seed cubes of U are known. By this surface tracing technique, surface patches in (VB2) can be excluded. Φ has an important property:

(P5) Φ is a connected surface in the 3D image and its edge-cubes (or boundary sub-images) are also connected.

Here, two edge-cubes (or boundary sub-images) are called connected if both of them are passed through by Φ . In this paper, edge-cubes in each boundary sub-image and boundary sub-images in the 3D image are detected and traced by (P5).

3.3.1 Different Tracing Rules for Different Edge-Cubes in U

Each edge-cube has 12 edges, six faces and six adjacent cubes. Each edge is shared by two faces, and each face is shared by two adjacent cubes. Here, the two cubes sharing a common face are called adjacent. According to the surface connectivity, if an edge in the edge-cube Q is intersected by Φ , then the two faces (denoted by b_1 and b_2) sharing this edge must also be intersected by Φ . Furthermore, the two cubes sharing b_1 and b_2 with Q respectively must be intersected by Φ . Thus, by marking all edges intersected by Φ in Q , we are able to recognize all faces intersected by Φ in Q , and recognize new edge-cubes from six adjacent cubes of Q . By detecting edge-cubes from the six adjacent cubes of each new edge-cube in the same manner, all edge-cubes connected to Q can be traced from U . This paper will trace edge-cubes in U with the above strategy.

For a given edge-cube $Q \in U$, a new edge-cube (denoted by G) adjacent to Q may belong to Ω_1 or Ω_2 , may be in U or not (possibly in adjacent sub-images of U as Q is at the border of U), or may contain at most two or at least three edges satisfying Equations (6) and (7) (e.g., intersected edges). Therefore, the following different tracing rules will be applied to these different cases accordingly:

If (G has at most two intersected edges, or $G \in \Omega_2$ but $G \notin U$)

then {stop surface tracing from G } //rule 1

else if ($G \in \Omega_1$ but $G \notin U$)

then {mark G as a seed cube, stop tracing from G } //rule 2

else if ($G \in \Omega_2$ and $G \in U$)

then {mark G as the cube in Ω_2 , stop tracing from G } //rule 3

else {mark the intersected edges and faces in G and trace new edge-cubes from the six-neighborhood of G } //rule 4.

When G has at most two intersected edges, the contained zero-crossing patches have low gradient magnitudes and are most likely to be surface fragments or patches in (VB4). By the rule 1 above, we can avoid tracing surface fragments or patches in (VB4). By the rule 2, seed cubes can be marked for the sub-images adjacent to U . They will be used later to

detect new boundary sub-images from the six neighborhood of U . By the rules 1 and 2, we can limit the surface tracing in U . By the rule 3, we avoid tracing edge-cubes from the edge-cubes in Ω_2 , and thus avoid tracing surface fragments or patches in (VB3). The surface tracing is done in the edge-cubes that are in U , belong to Ω_1 and have at least three intersected edges. $G_U(x, y, z)$ is used for the detection of the edges intersected by Φ in the rules 1 and 4.

The workflow of tracing edge-cubes in U is shown in Fig. 1b with each detailed step described in Appendix. The following summarizes the basic procedures. First, an edge-cube in $\Omega_1 \cap U$ is selected as the seed cube. The edges and faces intersected by Φ can be detected from the seed cube, and new edge-cubes can be determined from the six neighborhood of the seed cube. Next, different tracing rules 1-4 are applied to each new edge-cube. By repeating the same procedure on each newly determined edge-cube, all connected edge-cubes in $U \cap \Omega_1$ are detected from U , and the edge-cubes in $U \cap \Omega_2$ and the seed cubes in adjacent sub-images of U are also marked.

3.3.2 Reconstruction of Boundary Surface Patches in U

After having traced edge-cubes from U , we will reconstruct boundary surface patches from U . First, boundary surface patches in all edge-cubes belonging to $U \cap \Omega_1$ are computed. If an edge-cube belongs to $U \cap \Omega_1$, then the boundary surface patch in it can be computed by extracting the zero-crossing surface patch from it using the marching-cubes algorithm [5].

Second, boundary surface patches in the edge-cubes belonging to $U \cap \Omega_2$ are computed. If an edge-cube (denoted by B) is in $U \cap \Omega_2$, then B possibly contains both boundary surface patches and zero-crossing patches in (VB3). Denote Ψ_B as the set of all disconnected zero-crossing patches (or disconnected polygons) in B . The topology-correct zero-crossing patches in Ψ_B can be determined according to [12]. It is necessary to recognize correctly boundary surface patches from Ψ_B . This will be done by the surface connectivity of Φ , e.g., according to known boundary surface patches in adjacent edge-cubes. Among the six neighborhood of B , some edge-cubes belonging to $U \cap \Omega_1$ have been marked and the edges intersected by Φ have been detected in them as well. Some of these intersected edges actually are the edges of B as well. In this way, the edges intersected by Φ are determined in B . If an edge (denoted by L) of B is intersected by Φ , by the surface connectivity of Φ , the zero-crossing patch (or the polygon) intersecting at L is recognized as a boundary surface patch. Hence, boundary surface patches are recognized from Ψ_B by detecting the intersected edges of B from adjacent edge-cubes in $U \cap \Omega_1$.

Boundary surface patches contained in each edge-cube are represented by at least one and at most five triangles [5], [10], [11], [12]. Triangles in all edge-cubes of U form polygonal boundary surface patches of Φ in U . T_U , which is roughly estimated in Equation (3) before, is now updated as follows:

$$T_U = \min\{\|\nabla f(x, y, z)\| : (x, y, z) \in \Omega_U \subset (U \cap \Phi)\}, \quad (8)$$

where Ω_U denotes the set of vertices of the polygonal boundary surface patches in U . The updated T_U will be

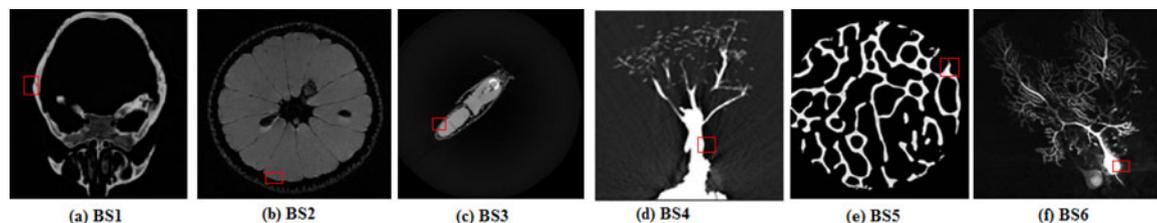


Fig. 4. Slices of six different 3D images (CT head, MRI orange, CT lobster, CT bonsai, scanned bovine proximal tibial bone, Confocal microscopy neuron). Rectangles drawn on different slices designate different boundary surfaces of interest.

used for estimating local gradient thresholds in adjacent boundary sub-images of U ; see Equations (3)-(5).

3.4 Hierarchical Tracing of Φ in the 3D Image

For the given 3D image, Φ , the CIBS of interest, can be designated or specified from a certain 2D sectional image by the interactive operation in [3]. When users browse a series of 2D section images on the computer screen, they can easily see the structure of interest, and select a specific 2D section image in which a salient boundary curve segment of the structure is visible. On the 2D image, users can draw interactively a small closed region (denoted by R), in which only the salient boundary curve passes through. Then the closed regions that have the same position as R but distribute on five consecutive slices form a small 3D region (denoted by W) in the 3D image, which contains a small salient boundary surface patch of Φ by the surface continuity. In this way, a small salient surface patch of Φ is determined implicitly, and Φ is designated in the 3D image.

By W , the initial boundary sub-image will be determined from the 3D image. In W , a cube G will be marked as an edge-cube if it satisfies: (i) it has at least three zero-crossing edges; (ii) the mean of gradient magnitudes of the contained zero-crossing surface patch reaches the maximum value of those in the cubes in W . Without loss of generality, suppose that $G \in \Omega_1$ and in a sub-image U_0 , T_0 is the mean of gradient magnitudes of the zero-crossing surface patches contained in G . U_0 has a small size, and contains a salient surface patch of Φ . We may detect, trace and reconstruct the salient surface patch of Φ from U_0 by the tracing rules in Section 3.3.1, where $G_{U_0}(x, y, z)$ is assumed to be a constant $\frac{T_0}{2}$. Finally, the boundary surface patch in U_0 and T_{U_0} are computed, and seed cubes in the sub-images adjacent to U_0 are marked.

Whenever U_0 , T_{U_0} and seed cubes in the adjacent sub-images of U_0 are known, we can trace new boundary sub-images from the six neighborhood of U_0 by (P5). Each adjacent sub-image (denoted by U) containing seed cubes is marked as a boundary sub-image, and its $G_U(x, y, z)$ can be estimated by Equations (3)-(5). In U , one can trace edge-cubes from each seed cube by the tracing technique in Section 3.3. Finally, one or several connected boundary surface patch of Φ can be computed in U . Meanwhile, seed cubes in the adjacent sub-images of U are marked, which can be further used to determine new boundary sub-images. We can repeat the same operation as above in each newly detected boundary sub-image. Eventually, all connected boundary sub-images and the boundary surface patches contained in them are computed. All these patches

form a polygonal surface model of Φ with a sub-voxel accuracy. The workflow and steps of tracing boundary sub-images in the 3D image are shown in Fig. 1a and Appendix, respectively. Since boundary surface patches in different boundary sub-images (or different edge-cubes) all belong to the same isosurface $\nabla^2 f(x, y, z) = 0$, boundary surface patches extracted from adjacent boundary sub-images (or adjacent edge-cubes) can be pieced together [5], [10], [11], [12].

In order to display the polygonal surface model of Φ , it is necessary to calculate unit normals at vertices of polygons or triangles. The conventional rendering algorithm takes the normals to produce Gouraud-shaded images. The gradient vector at a boundary point is just the direction of the normal at this point ([22], p. 40). So, one can obtain the unit normals at vertices of triangles by interpolating gradient vectors of vertices of intersected edges.

4 EXPERIMENTAL RESULTS

The proposed method has been applied to many different 3D images. Here, we show experimental results from six different 3D images, including CT images of a human skull, a lobster and a bonsai (Figs. 4a, 4c, 4d), an MRI image of an orange (Fig. 4b), a confocal microscopy image of a nerve cell (Fig. 4f) and an image of a bovine proximal tibial bone (Fig. 4e). For each 3D image, a 2D slice image is shown in Fig. 4, where a salient boundary curve segment of interest can be seen. At each slice, a small closed region (rectangle) is drawn to show where a salient boundary curve passes through. From these small regions, six different boundary surfaces of interest are selected respectively, and six small salient boundary surface patches are determined implicitly. These boundary surfaces are named as BS1, BS2, ..., BS6, respectively. Some of these boundary surfaces have varying gray values and gradient values, and have complex branches, as can be seen from Figs. 4d, 4e, 4f.

After small closed regions are specified on slices as in Fig. 4, the six boundary surfaces of interest can be automatically detected and reconstructed from the 3D images by the proposed method. These boundary surfaces are displayed in Fig. 5. In order to evaluate experimental results in Fig. 5, volume rendering results of the boundary surfaces BS1-BS6 are provided in Fig. 6. Figs. 6a, 6b, 6d, 6e are generated by the ray casting technique, Fig. 4f by the maximum intensity projection (MIP), and Fig. 4c by semi-transparent surface display, respectively. A comparison between Figs. 5 and 6 demonstrates that the proposed method can correctly reconstruct boundary surfaces from

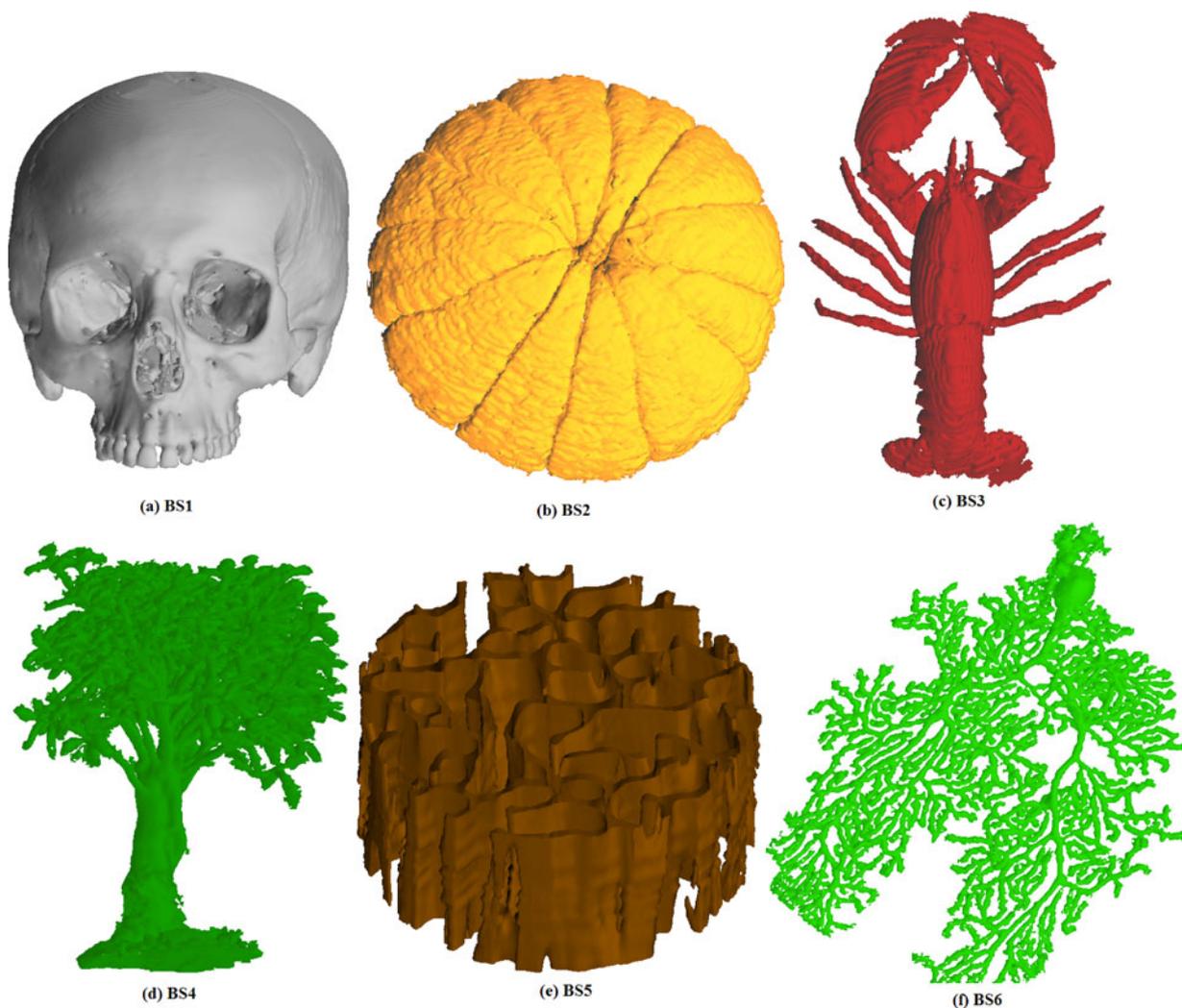


Fig. 5. 3D models of the boundary surfaces designated in Fig. 4, which are reconstructed automatically from 3D images by the proposed method when rectangles are drawn as that in Fig. 4.

3D images, even if some of them have varying gray values and gradient magnitudes, and complex branches. We also display the distributions of boundary surfaces BS1-BS3 and BS5 on some 2D sectional images, as shown in Fig. 7. Fig. 7 shows clearly that the boundaries detected by the proposed method are consistent with the boundaries observed by human vision, indicating the effectiveness of the method. In this paper, a hierarchical tracing framework is used to gradually expand a known small boundary surface patch until the entire boundary surface is reconstructed. So, we show examples to illustrate such procedures in Fig. 8. These experimental results show that the proposed method is easy to use, facilitating the exploration and understanding of the structure of interest in 3D images.

In Table 1, for each selected boundary surface, we show the number of its edge-cubes in Ω_1 , edge-cubes in Ω_2 and boundary surface patches (e.g., triangles) computed from all edge-cubes. The computational time used for tracing, reconstructing and visualizing each boundary surface of interest is also provided. These experiments were run on a PC with Intel Core i5 760 CPU 2.80 GHz, 4 GB RAM. Table 1 shows that the computational time is mainly affected by the

numbers of edge-cubes and boundary surface patches. The use of GPU technique or the parallelization technique can further reduce the computational time.

5 DISCUSSIONS

5.1 Boundary Surface Patches, Zero-Crossing Surface Patches and Isosurfaces

5.1.1 Isosurfaces in 3D Images

Although the isosurface-extraction technique is widely used, in many cases Φ cannot be well approximated by any isosurface in $f(x, y, z)$. For example, the CIBS with varying gray values and the CIBS of the structure whose sub-structures have different gray values usually cannot be well approximated by any isosurface. Let's see two examples in Figs. 9 and 10.

In Figs. 9a, 9b, 9c, 9d, four isosurfaces with different iso-values of 20, 50, 80 and 110 are reconstructed from the 3D confocal microscopic image of the nerve cell, respectively. One can see that the isosurfaces with high iso-values can well extract the surfaces of the cell body and axon, but cannot extract the surfaces of many dendrites. On the other hand, the isosurfaces with low iso-values can extract

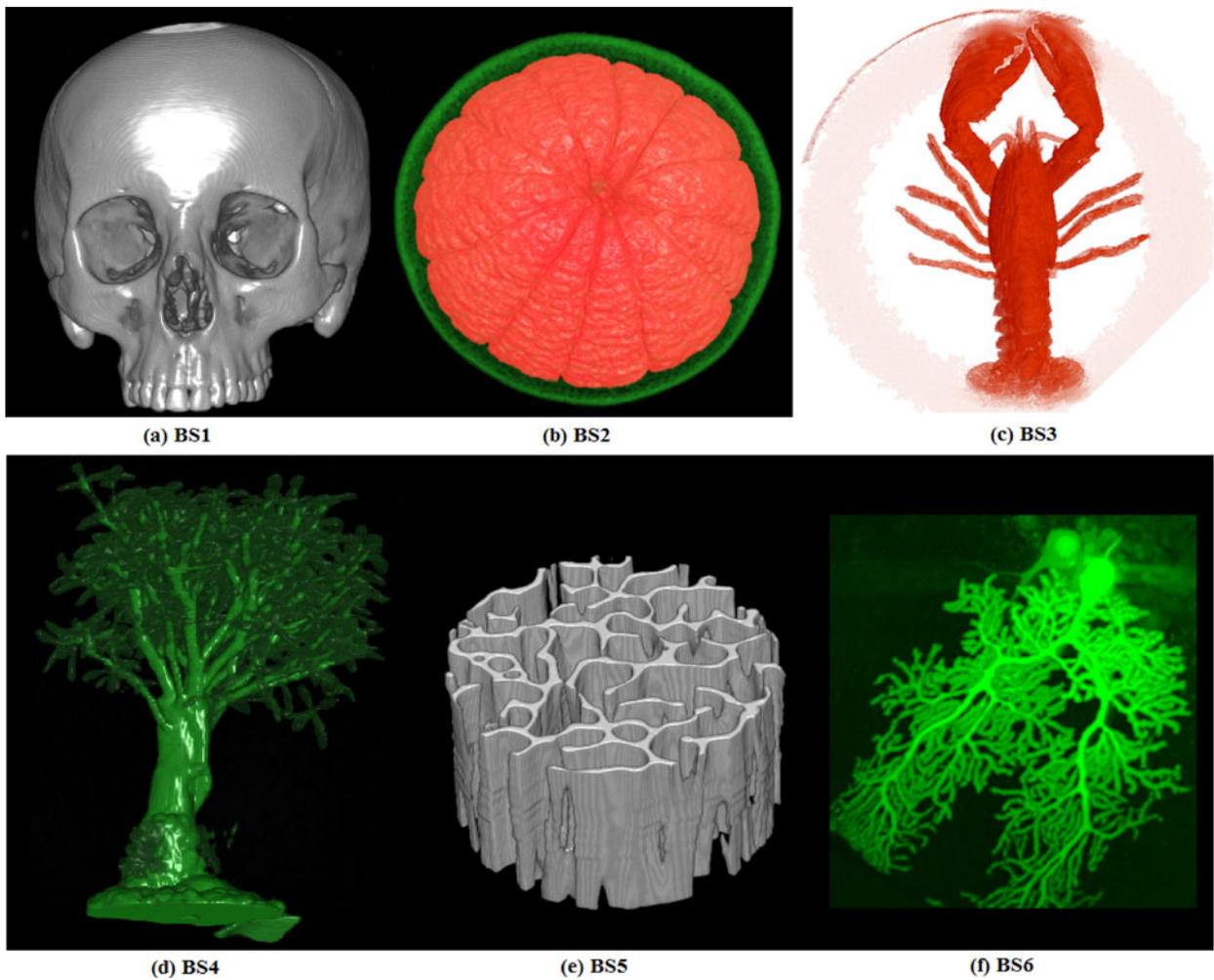


Fig. 6. Volume rendering results of the boundary surfaces designated in Fig. 4. They are rendered by different techniques. (a)(b)(d)(e): the ray-casting. (c): semi-transparent surface display. (f): MIP.

surfaces of most dendrites, but cannot extract correctly the surfaces of the cell body and axon. Furthermore, in these isosurfaces, surfaces of adjacent dendrites cannot be separated correctly. In contrast to these isosurfaces, the whole boundary surface of the nerve cell (including surfaces of cell body, axon and dendrites) can be well reconstructed by the proposed method, see Fig. 9f. Additionally, the surfaces of adjacent dendrites can also be well separated.

In Figs. 10a, 10b, 10c, three isosurfaces with isovalues of 30, 40, 50 are reconstructed from the 3D CT image of the lobster, respectively. Fig. 10f shows the semi-transparent visualization result of the lobster, which displays clearly different sub-structures of the lobster, such as pincher claw, crusher claw, walking legs, antenna and tail fan (they are

enclosed by four different blue circles). It is easy to see that the isosurfaces with high isovalues can correctly extract surfaces of the pincher claw and crusher claw, but not the

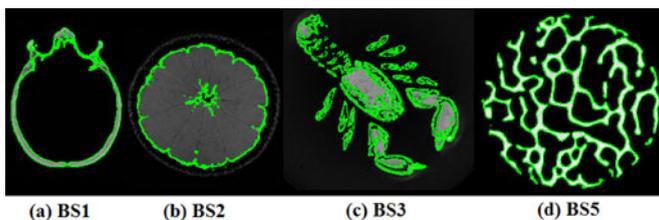


Fig. 7. The distributions of the boundary surfaces BS1-BS3 and BS5 on certain 2D sectional images.

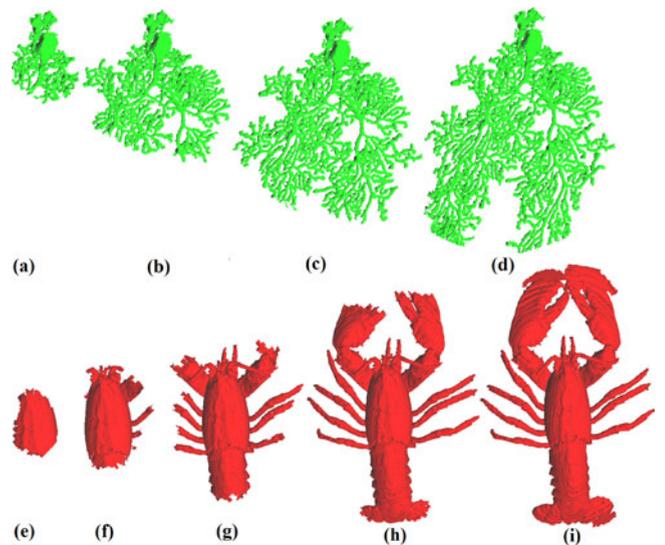


Fig. 8. Gradually expanded procedure of a known small boundary surface patch in 3D images. (a)-(d): BS6; (e)-(i): BS3.

TABLE 1
 Ω_1, Ω_2 : The Numbers of Edge-Cubes Belonging to Two Different Classes. BSP: The Number of Boundary Surface Patches Recognized from all Edge-Cubes

BS	image sizes	Ω_1	Ω_2	BSP	times(s)
BS1	218×218×142	296764	15445	659460	56.0
BS2	256×256×64	119999	26994	341308	19.8
BS3	301×324×56	199496	12075	452726	36.1
BS4	256×256×113	271768	17202	621021	89.4
BS5	320×240×29	173820	2844	357973	11.2
BS6	180×206×35	120206	19042	156360	3.08

Times: computational time (unit: second) used for detecting and reconstructing each boundary surface of interest.

surfaces of walking legs, antenna and tail fan (see Fig. 10c). On the other hand, the isosurfaces with low isovalues can correctly extract surfaces of walking legs and antenna, but not the surfaces of the pincher claw, the crusher claw and tail fan (see Fig. 10a). This implies that the whole surface of the lobster cannot be well approximated by any isosurface, because different structures of the lobster have different gray values. On the contrary, the whole boundary surface of the lobster can be well reconstructed by the proposed method, as shown in Fig. 10e.

Figs. 9, and 10 illustrate the difference between the CIBS and the isosurface. It is trivial to mark cubes intersected by an isosurface from 3D images, but it is more complicated to detect edge-cubes from 3D images and to distinguish boundary surface patches from the uninteresting zero-crossing surface patches.

Instead of using the isosurface with a fixed isovalue, we have discussed the adaptive approximation of the boundary surface with varying gray values by different isosurface patches (in different local regions) [30]. The 3D models of

the nerve cell and the lobster generated by such a method are shown in Figs. 9e and 10d, respectively. Although these models can improve the approximation of various isosurfaces, they are still inferior to the proposed method. In particular, the surfaces constructed with this method usually contain some uninteresting surface patches.

5.1.2 Zero-Crossing Surface Patches in 3D Images

A CIBS Φ is a specific zero-crossing surface patch, but not all zero-crossing surface patches belong to Φ . In some 3D images, many uninteresting zero-crossing surface patches are linked with Φ , and the gradient magnitudes of some patches are possibly larger than those of boundary surface patches of Φ . This is illustrated in Fig. 11.

In Fig. 11, zero-crossing surface patches linked with BS3, BS4 and BS6 and the 3D model of a nerve cell are shown. In addition, the gradient magnitudes over these zero-crossing surface patches and BS3, BS4, BS6 and the nerve cell model are shown (the color mapping table is the same as in Fig. 14). Fig. 11 indicates that zero-crossing surface patches in (VB4) can have negative impact on the understanding and visualization of the boundary surfaces of interest. Further, by comparing Fig. 6f to 11a, it is easy to see that some zero-crossing surface patches in (VB4), such as the surface of another uninteresting cell body, may have gradient magnitudes larger than those of the dendrites. Fig. 11 demonstrates that the boundary surfaces of interest usually cannot be reconstructed correctly by directly extracting the zero-crossing surface $\nabla^2 f(x, y, z) = 0$.

5.1.3 Boundary Surface Patches in 3D Images

In addition to the local characteristics (P1)-(P2), other global or regional characteristics of Φ (such as (P3)-(P5)) are used

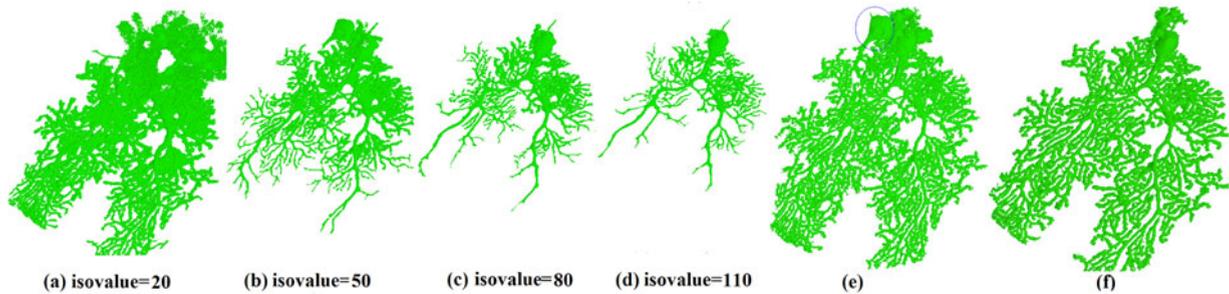


Fig. 9. Visualization results of the confocal microscopic nerve cell by different methods. (a)-(d): isosurfaces with four different isovalues from low to high: 20, 50, 80, 110. (e): the surface formed by different isosurface patches. (f): the boundary surface reconstructed by the proposed method.

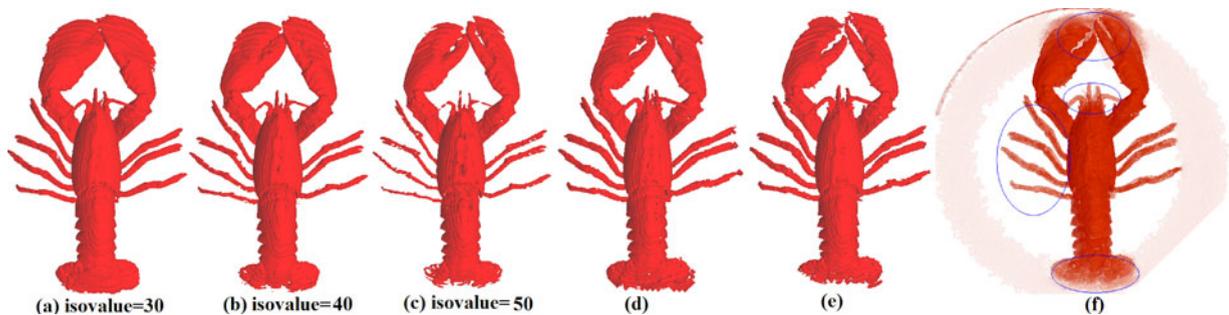


Fig. 10. Visualization results of the CT lobster by different methods. (a)-(c): isosurfaces with three different isovalues from low to high: 30, 40, 50. (d): the surface formed by different isosurface patches. (e): the boundary surface by the proposed method. (f): the semi-transparent surface as that in Fig. 6c.

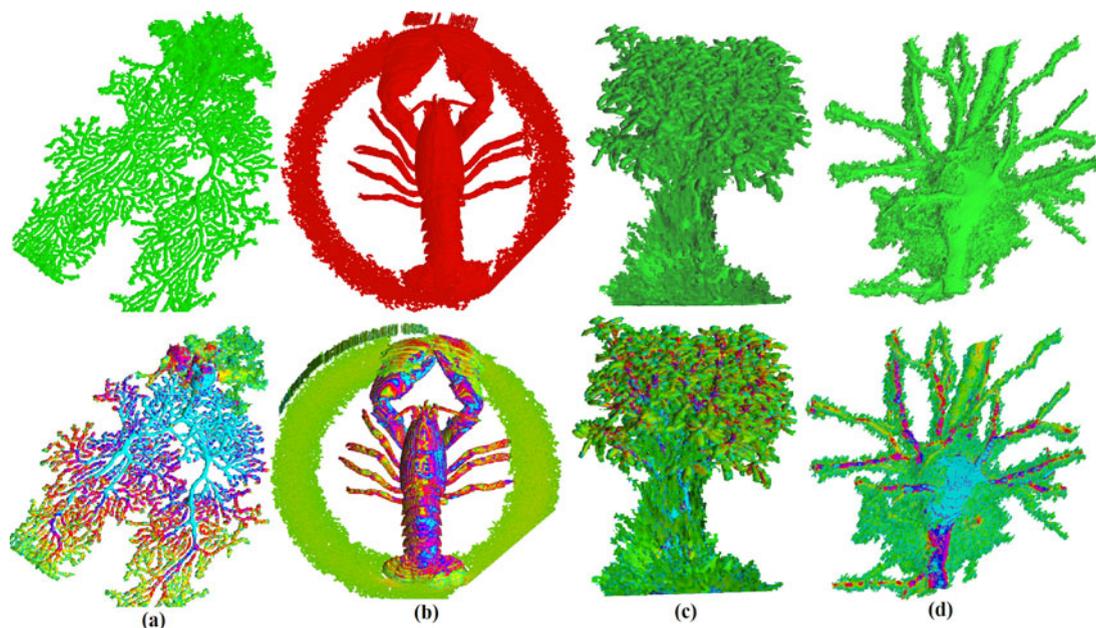


Fig. 11. Visualizations of various zero-crossing surface patches linked with BS6 or BS3 or BS4 or the 3D model of another nerve cell in Fig. 12b (the upper part), and visualizations of distributions of gradient magnitudes over these zero-crossing surface patches (the lower part). CIBS of interest cannot be correctly displayed.

to trace and recognize boundary surface patches in the 3D image. Particularly, they are used to exclude zero-crossing surface patches in (VB2)-(VB4).

Surface patches in (VB2) do not pass through edge-cubes of Φ . Thus, our work excludes such patches by only tracing edge-cubes of Φ in the 3D image, e.g., by (P5). In this way, we can avoid detecting uninteresting boundary surface patches of other structures and many other surface fragments, which are not close to Φ .

Surface patches in (VB4) are linked with Φ . Hence, they cannot be excluded by the surface connectivity. By (P4), each surface patch in (VB4) usually has gradient magnitudes lower than ones of the boundary surface patch it attaches to. Therefore, we compute local gradient thresholds $G_U(x, y, z)$ to exclude surface patches in (VB4). By the tracing rule 1 in Section 3.3.1, if an edge-cube has at most two zero-crossing edges satisfying Equations (6) and (7), then surface tracing will stop from it. See Figs. 5f and 11a, and Figs. 5d and 11c for illustration.

Some 3D images might contain many edge-cubes in Ω_2 and these edge-cubes might contain lots of surface fragments in (VB3). This is illustrated in Table 2 and Fig. 12. In Table 2, for each boundary surface in BS1-BS6, the ratio

between edge-cubes in Ω_2 and in Ω_1 , the number of zero-crossing surface patches in all edge-cubes in Ω_1 and Ω_2 (denoted by ZCP), the number of boundary surface patches recognized from all edge-cubes (denoted by BSP), and the ratio between $ZCP - BSP$ and BSP are provided. In Fig. 12, boundary surface models with and without surface fragments in (VB3) are shown respectively, where the red triangles or polygons are the boundary surface patches

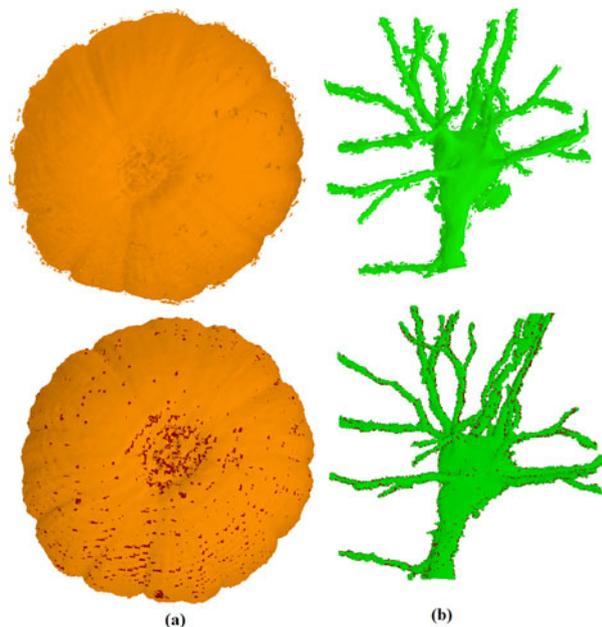


Fig. 12. The upper part is the visualization of zero-crossing surface patches contained in all edge-cubes in Ω_1 and Ω_2 . Many surface fragments in (VB3) can be seen. The lower part is the visualization of boundary surface patches contained in the same edge-cubes. Red polygons represent boundary surface patches computed from the edge-cubes in Ω_2 .

TABLE 2
 Ω_2/Ω_1 : The Ratio of Two Classes of Edge-Cubes

BS	$\frac{\Omega_2}{\Omega_1}$	ZCP	BSP	$\frac{ZCP-BSP}{BSP}$
BS1	5.2%	664579	659460	0.77%
BS2	5.8%	357063	341308	4.61%
BS3	6.0%	456179	452726	0.76%
BS4	6.3%	632704	621021	1.88%
BS5	1.63%	359259	357973	0.35%
BS6	15.8%	167441	156360	7.09%

ZCP: the number of zero-crossing surface patches in all edge-cubes;
BSP: the number of boundary surface patches recognized from all edge-cubes.

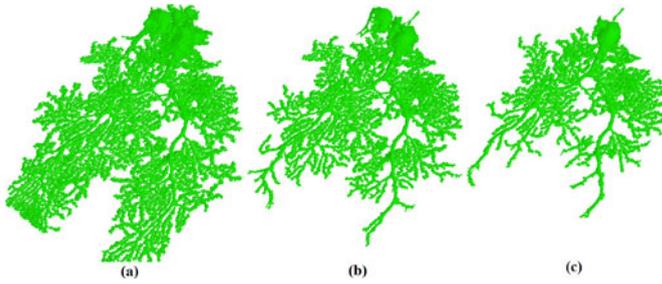


Fig. 13. Boundary surfaces extracted from the 3D confocal microscopic images of the nerve cell by three different global gradient thresholds from low to high. In these boundary surfaces, either uninteresting zero-crossing surface patches in (VB4) cannot be excluded ((a)-(b)), or surfaces of many dendrites are lost ((b)-(c)).

computed from the edge-cubes in Ω_2 . Table 2 and Fig. 12 show that it is necessary to correctly recognize boundary surface patches from the edge-cubes in Ω_2 and treat edge-cubes in Ω_1 and Ω_2 differently. Otherwise, surface fragments in (VB3) and zero-crossing surface patches connected with them will be incorrectly traced and recognized as boundary surface patches. By the proposed method, surface fragments in (VB3) can be well excluded.

5.2 Adaptive Computation of Local Gradient Thresholds

If only a small boundary surface patch is known, it is not clear whether or not a global gradient threshold is effective for expanding the boundary surface patch in the 3D image and how to determine an appropriate global threshold. If some surface patches in (VB4) have gradient magnitudes

larger than those of boundary surface patches of Φ , Φ usually cannot be separated from the 3D image by any global gradient threshold. This is illustrated in Fig. 13. It shows that high global gradient thresholds may exclude surface patches in (VB4) (see Fig. 13c), but the surfaces of many dendrites are lost. On the other hand, low global gradient thresholds may keep surfaces of dendrites, but the surface patches in (VB4) cannot be excluded (see Figs. 13a, 13b). It is necessary to compute local gradient thresholds to separate Φ from such patches in the surface tracing. Therefore, this paper detects and expands the CIBS of interest by the gradient threshold function $G_U(x, y, z)$.

$G_U(x, y, z)$ is computed by combining $V(x, y, z)$ and gradient magnitudes of boundary surface patches in certain adjacent sub-images of U (e.g., the context dependence of Φ). It characterizes the boundary surface patches in U that are visible according to $V(x, y, z)$ and their gradient magnitudes change slightly in contrast to those of adjacent boundary surface patches. Thus, $G_U(x, y, z)$ can adjust itself adaptively according to image contents (e.g., $V(x, y, z)$) in different 3D images or in different sub-images, and hence can be utilized to detect boundary surface patches in different sub-images adaptively. In other words, $G_U(x, y, z)$ can change adaptively with gradient magnitudes of boundary surface patches, which is illustrated in Fig. 14. In Fig. 14, the gradient magnitudes over four different boundary surfaces (BS1, BS6, BS4, BS3) are shown, where gradient magnitudes are mapped to different colors by a color table. In addition, the plots of $G_U(x, y, z)$ over the same four boundary surfaces are shown respectively, where they are mapped to different colors by the same color table. Fig. 14 illustrates that

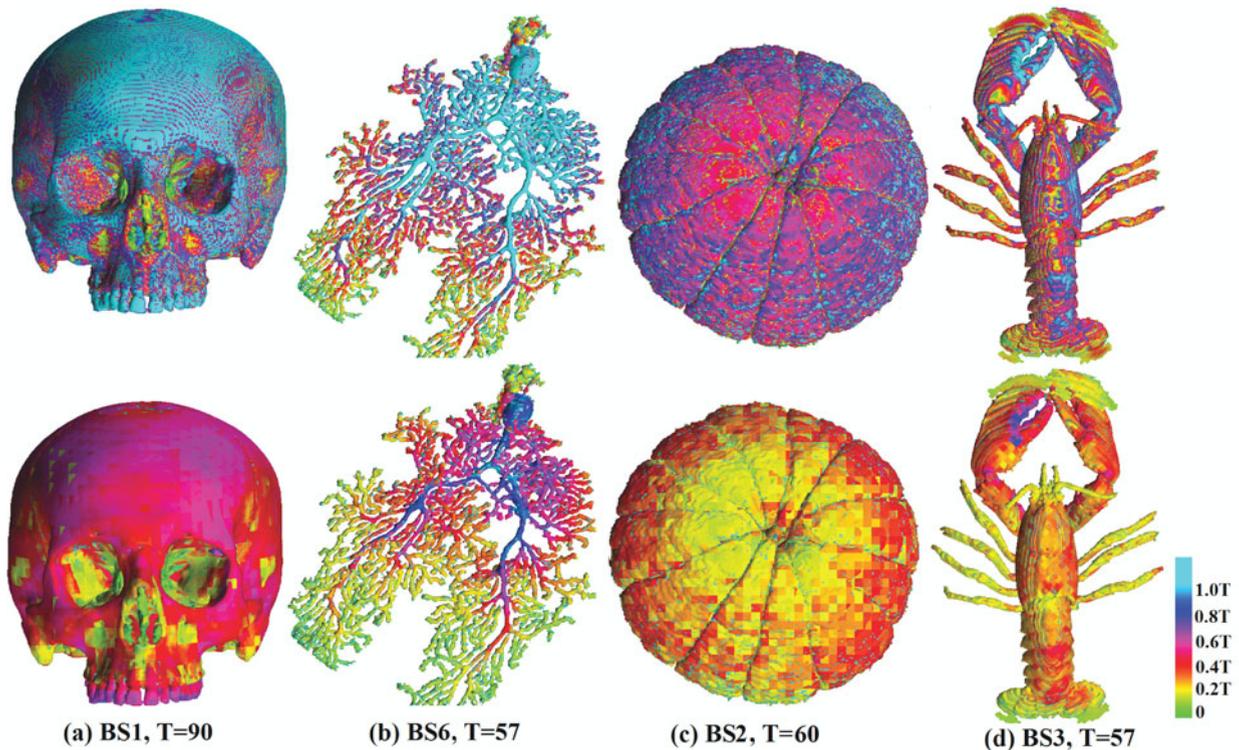


Fig. 14. Visualization of distributions of gradient magnitudes over four different boundary surfaces BS1, BS6, BS2 and BS3 (the upper part), and visualization of distributions of local threshold function $G_U(x, y, z)$ over the same four boundary surfaces (the lower part). It is easy to see that local gradient thresholds are adaptively changed with gradient magnitudes.

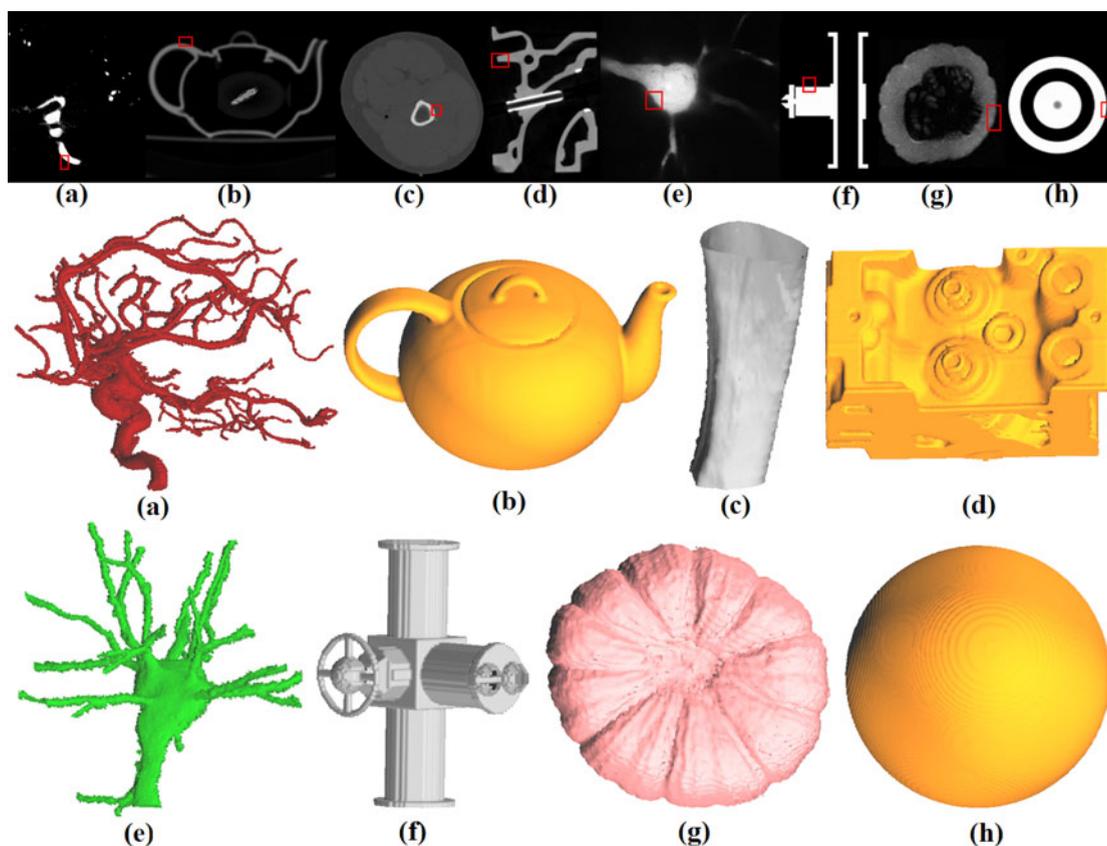


Fig. 15. Boundary surfaces reconstructed from different 3D images by the proposed method when the five parameters $d, e, \alpha, \kappa, \tau$ have the same fixed values: $(e, \kappa, \tau, d, \alpha) = (210, 0.8, 0.6, 45, 0.25)$. CIBSes of interest are designated in slices (a)-(h).

$G_U(x, y, z) \leq \|\nabla f(x, y, z)\|$ holds over Φ and $G_U(x, y, z)$ changes adaptively on a boundary surface patch, where $\|\nabla f(x, y, z)\|$ changes. This explains why the proposed method can adaptively detect and trace the CIBS with varying gradient magnitudes, such as BS6 and BS4. If we detect edge-cubes from the 3D image based merely on $V(x, y, z)$ rather than $G_U(x, y, z)$, then some surface patches in (VB4) and with high gradient magnitudes cannot be excluded. By (P4), $G_U(x, y, z)$ can exclude such surface patches.

Five parameters $d, e, \alpha, \kappa, \tau$ are used in the computation of $G_U(x, y, z)$. According to, Figs. 5, 15, and 16, $e = 210$ can be applied to various cases. In particular, Fig. 16 shows that the final results are unchanged for different $e \in (180, 240)$. So, we only need to consider d, α, κ, τ . The parameter κ describes the slow change of gradient magnitudes over Φ . τ is used to make a balance between $V(x, y, z)$ and M_U in $G_U(x, y, z)$. α determines how many weak boundaries are detected, where more (or less) will be detected if α become smaller (or larger). d is small if the structure of interest (or its sub-structure) has small gray values. From our experiences, in most cases, the values around 0.8, 0.6, 45 and 0.25 are acceptable for κ, τ, d and α . For example, $(\kappa, \tau, d, \alpha) = (0.8, 0.6, 45, 0.25)$ are appropriate for reconstructing BS1, BS2, BS3 and BS5, respectively. In Fig. 15, by the same values, we also well reconstruct many different CIBSes from different 3D images. In some cases, these values may be adjusted to exclude uninteresting surface patches from Φ . For example, in Figs. 5f and 12b, d is adjusted as 35 so as to adapt to low gray values of dendrite structures of nerve cells. In Fig. 10e, d and α are

adjusted as 55 and 0.3 so as to exclude some weak boundaries. In Fig. 5d, κ and α are adjusted as 0.95 and 0.3 to exclude surface patches in (VB4) from Φ .

Usually, the reconstructed Φ is quite robust to the change of the parameter κ, τ, d, α . This is illustrated in Fig. 16. We show how the triangle number of the reconstructed Φ is changed when a parameter is gradually changed while other three parameters are fixed. When a parameter is fixed, it has the fix value as follows: $e = 210, \kappa = 0.8, \tau = 0.6, d = 45, \alpha = 0.25$. Each curve in Fig. 16 indicates how the triangle number of the reconstructed Φ is changed when a certain parameter is changed. Here, the changing trend of triangle number may describe the changing trend of the reconstructed Φ . Fig. 16 demonstrates that the reconstructed Φ has little change when a certain parameter (such as d or α or κ or τ) has a small variation. The change of e almost has no effect on the reconstructed Φ .

In most 3D images, the size of $4 \times 4 \times 4$ is acceptable for sub-images, and the final result has little change if the size has a slight difference. In this paper, $4 \times 4 \times 4$ and $5 \times 5 \times 5$ are used in Figs. 5 and 15. If gradient magnitudes changed very slowly over Φ , then the size can be larger.

5.3 Comparison with Other Related Methods

The isosurfaces usually fail to represent boundary surfaces with varying gray values, as shown in Figs. 9 and 10. Especially, for certain boundary surfaces linking with surface patches in (VB4), they even cannot be well

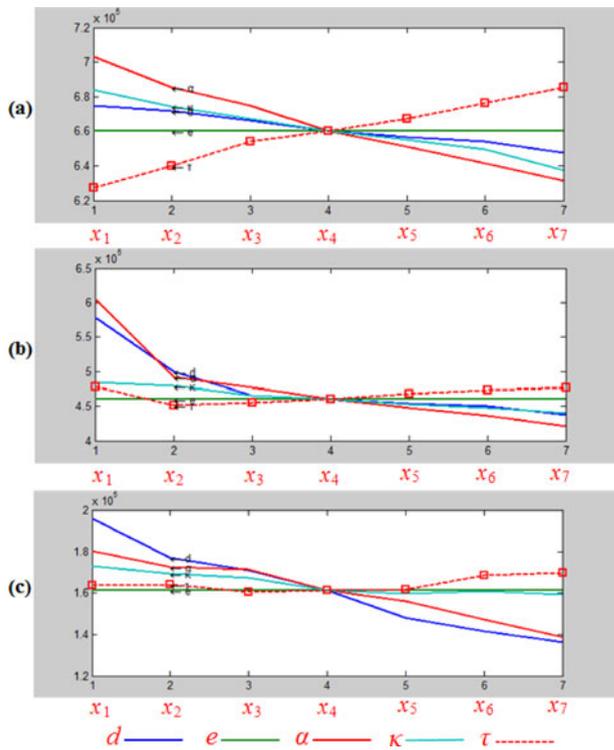


Fig. 16. Illustration of the changing trend of the triangle number of the reconstructed Φ when a parameter (d or e or α or κ or τ) is gradually changed. In (a)-(c), each curve shows how the triangle number of a reconstructed boundary surface model ((a): BS1; (b): BS3; (c): BS6) is changed when a parameter (d or e or α or κ or τ) is gradually changed while other four parameters are fixed. Several sampling values of each parameter are as follows separately: In (a)-(c): $e - (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (180, 190, 200, 210, 220, 230, 240)$; $\alpha - (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (0.175, 0.20, 0.225, 0.250, 0.275, 0.30, 0.325)$; $\kappa - (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95)$. In (c): $d - (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (20, 25, 30, 35, 40, 45, 50)$; $\tau - (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75)$. In (a)-(b): $d - (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (30, 35, 40, 45, 50, 55, 60)$; $\tau - (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90)$.

approximated by different isosurface patches in different local regions; see Figs. 9e and 10d. The proposed method can overcome this limitation.

In existing literatures, all zero-crossing surface patches in an edge-cube are usually regarded as boundary surface patches [3], [4], [30], [31], [44]. However, Table 2 and Fig. 12 show that some zero-crossing surface patches contained within some edge-cubes in Ω_2 are actually not boundary surface patches. The proposed method overcomes the drawback. In [3], [30], [31] and [44], edge-cubes are detected from 3D images based on a global gradient threshold. On one hand, it is a difficult task to select appropriate global gradient thresholds for different boundary surfaces in different 3D images. On the other hand, some uninteresting surface patches as shown in Fig. 13 cannot be excluded by using any global gradient threshold. In [3], we suggested detecting edge-cubes from 3D images by local gradient thresholds, but the technique of computing local gradient thresholds is not effective in many 3D images. The proposed method can be applied to many different 3D images.

3D zero-crossing based edge detectors usually regard a zero-crossing surface or a zero-crossing surface with high gradient magnitudes as a boundary surface. However,

such boundary surface might include many surface patches in (VB2)-(VB4), as shown in Figs. 11, 12, 13. By the proposed method, these unwanted surface patches in (VB2)-(VB4) can be excluded from the true CIBS. 3D gradient based edge detectors usually detect boundary voxels based on a high global gradient threshold. However, a fixed gradient threshold usually cannot work for the surfaces with varying gradient magnitudes, as shown in Fig. 13. The proposed method can detect and reconstruct the CIBS with varying gradient magnitudes, as shown in Figs. 5f and 11a, Figs. 5d and 11c. Reference [17] suggested generating a polygonal boundary surface model by computing a boundary surface patch in each boundary voxel. However, boundary surface patches from adjacent boundary voxels usually cannot be pieced together. The proposed method overcomes this difficulty.

The deformable surface technique, 3D edge detector and the approach for reconstructing a 3D model from contours are all important 3D model reconstruction techniques. Each of them has its own advantage and can work for some special cases. However, they cannot work if only a small piece of boundary surface of interest is known. The proposed method usually can reconstruct a CIBS automatically from a small boundary surface even if without prior knowledge about the CIBS. It can also reconstruct correctly the CIBS with complex shapes or branches, or with varying gray values or gradient values. Particularly, it is easy to use by researchers with little training.

6 CONCLUSION

This paper proposes a novel 3D surface reconstruction technique to facilitate users to explore and understand 3D image in an efficient way. With the technique, a 3D model of the structure of interest can be reconstructed easily, once a small region of interest is specified or designated from a 2D section image. This technique is based on the context dependence and the connectivity of the CIBS, and uses the adaptive contrast detection approach motivated by the human visual system. It can detect adaptively and trace the CIBS with varying gray values or gradient values from a 3D image. It can still detect surfaces with complex shapes or branches while eliminating wrong surface patches introduced by noise and other variations in the 3D image. The effectiveness of the technique is illustrated by various experimental results of real 3D images. Its advantages are also demonstrated by comparing with several existing methods such as the isosurface technique.

APPENDIX

The procedure for the proposed hierarchical tracing framework can be described briefly as follows:

Step 1. Based on a known small boundary surface patch of Φ , determine a seed cube and a local gradient threshold in an initial boundary sub-image U_0 . Furthermore, by the tracing rules and reconstruction technique in Section 3.3, compute boundary surface patches in U_0 and T_{U_0} (i.e., the infimum of gradient magnitudes of boundary surface patches in U_0), and mark seed cubes belonging to adjacent sub-images of U_0 .

Step 2. Trace boundary sub-images from six adjacent sub-images of U_0 based on whether or not a sub-image contains at least one seed cube. Push all new boundary sub-images into a stack (STACK-1) that is created to store boundary sub-images.

Step 3. Perform the following operations (Step 3.1-Step 3.4) in the 3D image circularly until STACK-1 is empty:

Step 3.1. Pop up a boundary sub-image U from STACK-1 and compute $G_U(x, y, z)$ (i.e., the local gradient threshold in U) by Equations (3)-(5). Create a new stack (STACK-2) to store edge-cubes. Push the seed cubes of U into STACK-2.

Step 3.2. Perform the following operations (Step 3.2.1-Step 3.2.4) in U circularly until STACK-2 is empty.

Step 3.2.1. Pop up an edge-cube G from STACK-2.

Step 3.2.2. Detect the edges and faces intersected by Φ from G by Equations (6)-(7).

Step 3.2.3. Trace edge-cubes from the six neighborhood of G . Each cube that shares an intersected face with G is marked as a new edge-cube.

Step 3.2.4. Treat new edge-cubes by the tracing rules in Section 3.3.1, respectively. If a new edge-cube has at most two edges satisfying Equations (6) and (7), then skip it. If a new edge-cube belongs to Ω_2 and is in U , then mark it. If a new edge-cube belongs to Ω_1 but is in an adjacent sub-image of U , then mark it as a seed cube. If a new edge-cube belongs to Ω_1 and is in U , then compute boundary surface patch in it and push it into STACK-2.

Step 3.3. Compute boundary surface patches from those edge-cubes that are in U and belong to Ω_2 by the technique in Section 3.3.2. Then boundary surface patches of Φ are reconstructed from U . Compute T_U by Equation (8).

Step 3.4. Trace boundary sub-images from the six neighborhood of U based on whether or not a sub-image contains at least one seed cube. Push all new boundary sub-images into STACK-1.

ACKNOWLEDGMENTS

The authors wish to thank the reviewers and the editor for their valuable comments and helpful suggestions that greatly improved the paper's quality. This work was supported in part by the National Natural Science Foundation of China (61375020), National Basic Research Program of China (2013CB329401) and Cross Research Fund of Biomedical Engineering of Shanghai Jiaotong University (YG2013ZD02, YG2012MS19).

REFERENCES

- [1] A. A. Goshtasby, M. Sonka, and J. K. Udupa, "Analysis of volumetric image," *Comput. Vis. Image Understanding*, vol. 77, pp. 79–83, 2000.
- [2] J. K. Udupa, "Three-dimensional visualization and analysis methodologies: A current perspective," *RadioGraphics*, vol. 19, pp. 783–806, 1999.
- [3] P. Wang, L. Cheng, B. Lin, and L. Wang, "Boundary surface detection in 3D biomedical images based on local gradient thresholds," in *Proc. IEEE Conf. Biomed. Eng. Inform.*, 2011, pp. 175–179.
- [4] L. Cheng, L. Wang, and Y. Ma, "A new strategy for boundary surface detection in 3D biomedical images," in *Proc. IEEE Conf. Biomed. Eng. Inform.*, 2010, vol. 1, pp. 51–55.
- [5] W. E. Lorensen and H. E. Cline, "Marching Cubes: A high resolution 3D surface construction algorithm," in *Proc. Annu. Int. Conf. Comput. Graph. Interactive Techn.*, 1987, pp. 163–169.
- [6] J. Bloomenthal, "Polygonization of implicit surfaces," *Comput. Aided Geometric Des.*, vol. 5, no. 4, pp. 341–355, 1988.
- [7] G. M. Treece, R. W. Prager, and A. H. Gee, "Regularised marching tetrahedra: Improved iso-surface extraction," *Comput. Graph.*, vol. 23, no. 4, pp. 583–598, 1999.
- [8] J. Schreiner, C. Scheidegger, and C. Silva, "High-quality extraction of isosurfaces from regular and irregular grids," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 1205–1212, Sept.-Oct. 2006.
- [9] A. Lopes and K. Brodlie, "Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 1, pp. 16–29, Jan.-Mar. 2003.
- [10] S. Raman and R. Wenger, "Quality isosurface mesh generation using an extended marching cubes lookup table," *Comput. Graph. Forum*, vol. 27, no. 3, pp. 791–798, 2008.
- [11] G. M. Nielson and B. Hamann, "The asymptotic decider: removing the ambiguity in marching cubes," in *Proc. IEEE Vis.*, 1991, pp. 83–91.
- [12] E. V. Chernyaev, "Marching cubes 33: Construction of topologically correct Isosurfaces," Tech. Rep. CN-95-17, CERN, Geneva, Switzerland, 1995.
- [13] J. S. Suri, K. Liu, S. Singh, S. N. Laxminarayan, X. Zeng, and L. Reden, "Shape recovery algorithms using level sets in 2D/3D medical imagery: A state of the art review," *IEEE Trans. Inform. Technol. Biomed.*, vol. 6, no. 1, pp. 8–28, Mar. 2002.
- [14] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Med. Image Anal.*, vol. 1, no. 2, pp. 91–108, 1996.
- [15] T. McInerney and D. Terzopoulos, "Topology adaptive deformable surfaces for medical image volume segmentation," *IEEE Trans. Med. Imag.*, vol. 18, no. 10, pp. 840–851, Oct. 1999.
- [16] S. J. Osher and R. P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. New York, NY, USA: Springer, 2002.
- [17] S. W. Zucker and R. A. Hummel, "A three-dimensional edge operator," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. PAMI-3, no. 3, pp. 324–331, May 1981.
- [18] L. M. Luo, C. Hamitouché, J. L. Dillenseger, and J. L. Coatrieux, "A moment-based three dimensional edge operator," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 3, pp. 693–703, July 1993.
- [19] C. Pudney, M. Robins, B. Robbins, and P. Kovsi, "Surface detection in 3D confocal microscope image via local energy and ridge tracing," *J. Comput.-Assist. Microsc.*, vol. 8, no. 1, pp. 5–20, 1996.
- [20] M. Brejl and M. Sonka, "Directional 3D edge detection in anisotropic data: Detector design and performance assessment," *Comput. Vis. Image Understanding*, vol. 77, pp. 84–110, 2000.
- [21] Y. Q. Guan, Y. Y. Cai, Y. T. Lee and M. Opas, "An automatic method for identifying appropriate gradient magnitude for 3D boundary detection of confocal image stacks," *J. Microsc.*, vol. 223, pp. 66–72, July 2006.
- [22] J. Udupa and G. Herman, *3-D Imaging in Medicine*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2000.
- [23] M. Bomans, K. H. Hohne, U. Tiede, and M. Riemer, "3-D segmentation of MR images of the head for 3D display," *IEEE Trans. Med. Imag.*, vol. 9, no. 2, pp. 177–183, Jun. 1990.
- [24] S. Zhan, "A zero-crossing-based optimal three-dimensional edge detector," *CVGIP: Image Understanding*, vol. 59, no. 2, pp. 242–253, 1994.
- [25] L. Hu, M. H. Loew, and J. Ostuni, "Automated registration of brain images using edge and surface features," *IEEE Eng. Med. Biol.*, vol. 18, no. 6, pp. 40–47, Nov. 1999.
- [26] L. G. Nonato, A. J. Cuadros Vargas, R. Minshim, and M. C. F. de Oliveira, "Beta-connection: Generating a family of models from planar cross sections," *ACM Trans. Graph.*, vol. 24, no. 4, pp. 1239–1258, 2005.
- [27] W. Koh and B. H. McCormick, "Topology-graph directed separating boundary surfaces approximation of nonmanifold neuroanatomical structures: Application to mouse brain olfactory bulb," *IEEE Trans. Med. Imag.*, vol. 28, no. 4, pp. 555–563, Apr. 2009.
- [28] C. F. Westin, A. Bhalerao, H. Knutsson, and R. Kikinis, "Using local 3D structure for segmentation of bone from computer tomography images," in *Proc. IEEE Comput. Soc. Comput. Vis. Pattern Recog.*, San Juan, PR, USA, Jun. 1997, pp. 794–800.
- [29] M. Styner, G. Gerig, C. Brechbuehler, and G. Szekely, "Parametric estimate of intensity inhomogeneities applied to MRI," *IEEE Trans. Med. Imaging*, vol. 19, no. 3, pp. 153–165, Mar. 2000.

- [30] L. Wang, J. Bai, P. He, P. A. Heng, and X. Yang, "A computational framework for approximating boundary surfaces in 3D biomedical images," *IEEE Trans. Inform. Tech. Biomed.*, vol. 11, no. 6, pp. 668–682, Dec. 2007.
- [31] L. Wang, J. Bai, and K. Ying, "Adaptive approximation of the boundary surface of a neuron in confocal microscopy volumetric images," *Med. Biol. Eng. Comput.*, vol. 41, no. 4, pp. 601–607, 2003.
- [32] J. J. Clark, "Authenticating edges produced by zero-crossing algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 1, pp. 43–57, Feb. 1989.
- [33] A. A. Masoud and M. M. Bayoumi, "Using local structure for the reliable removal of noise from the output of the LOG edge detector," *IEEE Trans. Systems, Man, Cybern.*, vol. 25, no. 2, pp. 328–337, Feb. 1995.
- [34] K. Rathnayaka, T. Sahama, M. A. Schuetz, and B. Schmutz, "Effects of CT image segmentation methods on the accuracy of long bone 3D reconstructions," *Med. Eng. Phys.*, vol. 33, pp. 226–233, 2011.
- [35] J. O. Lachaud, "Continuous analogs of digital boundaries: A topological approach to iso-surfaces," *Graph. Models*, vol. 62, pp. 129–164, 2002.
- [36] S. Fang and M. Adada, "Multi-scale iso-surface extraction for volume visualization," *Int. J. Image Graph.*, vol. 6, no. 2, pp. 173–185, 2006.
- [37] L. Wang, J. Bai, P. A. Heng, and T. T. Wong, "Isosurface computation for approximating boundary surfaces in 3D images," *J. Electron. Imag.*, vol. 16, no. 1, pp. 1–12, 2007.
- [38] R. Koren and Y. Yitzhaky, "Automatic selection of edge detector parameters based on spatial and statistical measures," *Comput. Vis. Image Understanding*, vol. 102, pp. 204–213, 2006.
- [39] R. Medina-Carnicer and F. J. Madrid Cuevas, "Unimodal thresholding for edge detection," *Pattern Recog.*, vol. 41, no. 7, pp. 2337–2346, 2008.
- [40] R. R. Rakesh, P. Chaudhuri, and C. A. Murthy, "Thresholding in edge detection: A statistical approach," *IEEE Trans. Image Process.*, vol. 13, no. 7, pp. 927–936, July 2004.
- [41] D. Marr and E. Hildreth, "Theory of edge detection," in *Proc. Roy. Soc. Lond. B29*, 1980, vol. 207, pp. 187–217.
- [42] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [43] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd ed. Beijing, China: People's Posts and Communication Publishing House, 2002.
- [44] P. A. Heng, L. Wang, T. T. Wong, K. S. Leung, and J. C. Cheng, "Edge surfaces extraction from 3D images," in *Proc. Med. Imag.: Image Process.*, 2001, pp. 407–416.
- [45] S. Drew, *Anatomy and physiology of the eye, contrast, contrast sensitivity, luminance perception and psychophysics*, Draft Rep. Griffith Univ., Australia, 2005, [online]. Available: <http://www.ict.griffith.edu.au/teaching/1008ICT/docs/>
- [46] S. Winkler, "Vision models and quality metrics for image processing applications" Ph.D. dissertation Inst. Elect. Electron. Eng., Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2000, p. 17.
- [47] W. B. Thompson, R. W. Fleming, S. H. Creem Regehr, and J. K. Stefanucci, *Visual Perception From a Computer Graphics Perspective*, New York, NY, USA: CRC Press, 2011 p. 19.
- [48] Z. Hou, C. Han, L. Zheng, and L. Xiao, "Threshold selection tactics for an edge detection based on vision model," *Opto-Electron. Eng.*, vol. 31, no. 2, pp. 59–63, 2004.
- [49] T. T. Elvins, "A survey of algorithms for volume visualization," *Comput. Graph.*, vol. 26, no. 3, pp. 194–199, 1992.
- [50] N. Tatarchuk, J. Shopf, and C. DeCoro, "Real-Time isosurface extraction using the gpu programmable geometry pipeline," in *Proc. ACM SIGGRAPH courses*, 2007, pp. 122–137.
- [51] S. Martin, H. Shen, and P. McCormick, "Load-balanced isosurfacing on multi-GPU clusters," in *Proc. Eurograph. Workshop Parallel Graph. Vis.*, 2010, pp. 91–100.
- [52] U. Vovk, F. Pernus, and B. Likar, "A review of methods for correction of intensity inhomogeneity in MRI," *IEEE Trans. Med. Imaging*, vol. 26, no. 3, pp. 405–421, Mar. 2007.
- [53] M. Levoy, "Display of surfaces from volume data," *IEEE Comput. Graph. Appl.*, vol. 8, no. 5, pp. 29–37, May 1988.
- [54] S. Fang, T. Biddlecome, and M. Tuceryan, "Image-based transfer function design for data exploration in volume visualization," in *Proc. IEEE Vis*, 1998, pp. 319–326.
- [55] P. Sereda, A. V. Bartroli, I. W. Serlie and F. A. Gerritsen, "Visualization of boundaries in volumetric data sets using LH histograms," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 2, pp. 208–218, Mar.-Apr. 2006.
- [56] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional transfer functions for interactive volume rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 8, no. 3, pp. 270–285, July 2002.
- [57] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 339–346, 2002.
- [58] G. M. Nielson, "On Marching Cubes," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 3, pp. 283–297, July-Sept. 2003.
- [59] L. Custodio, T. Etienne, S. Pesco, and C. Silva, "Practical considerations on Marching Cubes 33 topological correctness," *Comput. Graph.*, vol. 37, pp. 840–850, 2013.



Lisheng Wang received the MS degree in mathematics and the PhD degree in electronic and information engineering from Xi'an Jiaotong University, China, in 1993 and 1999, respectively. From October 1999 to October 2000, he was a research associate in the Department of Computer Science and Engineering, Chinese University of Hong Kong. From May 2001 to July 2003, he was a postdoctoral researcher in the Department of Biomedical Engineering, Tsinghua University, China. In August 2003, he joined Department of Automation, Shanghai Jiao Tong University, China, where he is currently a professor. His research interests mainly include analysis and visualization of 3D images, computer-aided diagnosis, and surgery planning.



Pai Wang received the BS degree in control theory and engineering and the MS degree in image processing and pattern recognition, both from Shanghai Jiao Tong University in 2010 and 2013, respectively. His main interests include 3D image analysis and visualization.



Liuhan Chen received the BS degree in mathematics from Sichuan University in 2008, and the MS degree in image processing and pattern recognition from Shanghai Jiao Tong University in 2011. His main interests include 3D image analysis and visualization.



Yu Ma received the MS degree in image processing and pattern recognition from Shanghai Jiao Tong University in 2007, and the PhD degree in computer vision from Northwestern Polytechnical University in 2013. His main interests include 3D reconstruction and visualization, 3D image analysis.



Shenzhi Wu received the BS degree in automatic control from Dalian Maritime University in 2006, and MS degree in image processing and pattern recognition from Shanghai Jiao Tong University in 2009. Her main interests include visualization and 3D reconstruction, 3D image analysis.



Zongben Xu received the MS degree and PhD degree in mathematics from Xi'an Jiaotong University, China, in 1981 and 1987, respectively. He has been with the Faculty of Science, Xi'an Jiaotong University, since 1982, where he was promoted to full professor in 1991, and now serves as director of the Institute for Information and System Sciences. He is the academician of the Chinese Academy of Sciences. His current research interests include intelligent information processing and computational intelligence.



Yu-Ping Wang received the MS degree in mathematics and the PhD degree in communications and electronic systems from Xi'an Jiaotong University, Xi'an, China, in 1993 and 1996, respectively. From 1999 to 2000, he was a post-doctor in the School of Medicine, Washington University. Now, he is an associate professor in the Department of Biomedical Engineering, Tulane University. His research interests mainly include multiscale bioimaging and bioinformatics.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**