Improving Separability of Structures with Similar Attributes in 2D Transfer Function Design

Shouren Lan, Lisheng Wang, Yipeng Song, Yu-ping Wang, Liping Yao, Kun Sun, Bin Xia, and Zongben Xu

Abstract—The 2D transfer function based on scalar value and gradient magnitude (SG-TF) is popularly used in volume rendering. However, it is plagued by the boundary-overlapping problem: different structures with similar attributes have the same region in SG-TF space, and their boundaries are usually connected. The SG-TF thus often fails in separating these structures (or their boundaries) and has limited ability to classify different objects in real-world 3D images. To overcome such a difficulty, we propose a novel method for boundary separation by integrating spatial connectivity computation of the boundaries and set operations on boundary voxels into the SG-TF. Specifically, spatial positions of boundaries and their regions in the SG-TF space are computed, from which boundaries can be well separated and volume rendered in different colors. In the method, the boundaries are divided into three classes and different boundary-separation techniques are applied to them, respectively. The complex task of separating various boundaries in 3D images is then simplified by breaking it into several small separation problems. The method shows good object classification ability in real-world 3D images while avoiding the complexity of high-dimensional transfer functions. Its effectiveness and validation is demonstrated by many experimental results to visualize boundaries of different structures in complex real-world 3D images.

Index Terms—Transfer function, volume rendering, connectivity computation, set operations, boundaries, classification ability

1 Introduction

Volume rendering is an important technique for visualizing structures in a 3D image, by which structures and their sizes, shapes and spatial relationships can be observed. In order to visualize structures in a 3D image, an appropriate transfer function must be designed [1], which assigns different voxels with different optical properties (e.g., opacity, color). The opacity setting determines which structure is visible, and the color setting is used to distinguish different structures.

Transfer function design has been widely studied by many researchers, see [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20] and references therein. But it remains a challenging problem to design an appropriate transfer function for a real-world 3D image containing multiple different structures. Generally, different structures should be visualized by different colors. This means that an appropriate transfer function should be able to well separate different structures of interest. However, some structures may have similar attributes, and some of them are even closely adjacent spatially and thus mistakenly connected. This makes it complicated to separate different structures in a real-world 3D image.

In traditional transfer functions, users try to separate different structures by selecting different regions from a given transfer function space [2], [3], [4], [5], [6], [7], [8]. Here, each transfer function space is formed by a variety of attributes. Possible attributes include scalar value, gradient magnitude [2], [14], second order derivative [2], [13], texture [16], visibility [17], curvature [23], occlusion [26], size [7], etc. According to the number of dimensions, transfer functions may be divided into two classes: low-dimensional transfer functions (LD-TF) and high-dimensional transfer functions (HD-TF).

LD-TFs attempt to separate different structures by fewer attributes [2], [3], [4], [5], [6], [7] [14], [15], [16], [17] [20], [21], [22], [23], [24], [25], [26]. Typical examples are the 2D transfer function (2D-TF) based on scalar value and gradient magnitude (abbreviated to SG-TF) [3] and the one based on the L-H histogram (abbreviated to LH-TF) [5]. Both SG-TF and LH-TF are used to visualize boundaries (i.e., surfaces having certain thickness, formed by many boundary voxels [2]) of structures by selecting certain regions from transfer function spaces. LD-TFs usually can provide user-friendly interfaces to observe and intuitively select or adjust regions in LD-TF spaces [3], [5], [25]. Therefore, they have been widely applied in volume rendering. However, LD-TFs face a common region-overlapping problem: different structures with similar attributes will have an overlapping region in a LD-TF space. This is illustrated in Fig. 1. Fig. 1a shows four different structures with the same attributes. Figs. 1b and 1c show that, either in the SG-TF space or the LH-TF space, boundaries of the four structures correspond to the same region. Hence, no matter which region is selected from the SG-TF space or LH-TF space, boundaries of the four structures cannot be correctly separated. As a result, LD-TFs have the limited ability of

- *L. Wang, S. Lan, Y. Song, and B. Xia 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, P. R. China. E-mail: lswang@sjtu.edu.cn, {lanshouren, song8733133}@163.com, flying.xiabin@gmail.com.*
- *Z. Xu is with the Institute for Information and System Science, School of Mathematics and Statistics, Xi'an Jiaotong University, P. R. China. E-mail: zbxu@mail.xjtu.edu.cn.*

Manuscript received 31 Jan. 2015; revised 20 Jan. 2016; accepted 16 Feb. 2016. Date of publication 2 Mar. 2016; date of current version 29 Mar. 2017. Recommended for acceptance by H. Qu.
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.2016.2537341

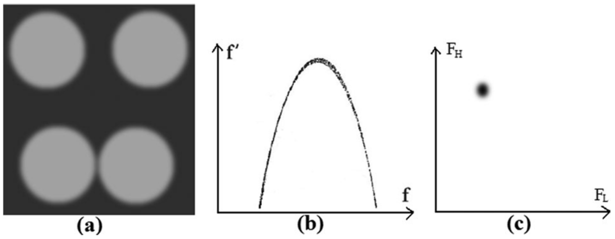


Fig. 1. (a) four different structures with the same attributes. (b)-(c) the corresponding regions of boundaries of the four structures in the SG-TF space and LH-TF space.

object classification when visualizing real-world 3D images with multiple structures.

In the SG-TF space, the similar boundary overlapping problem also exists: boundaries of different structures with similar attributes will have an overlapping region. Furthermore, due to the partial volume effects (PVE) [27], boundaries of two closely adjacent structures (e.g., there is a very thin region between them, see Fig. 7a) with similar attributes are usually regarded as a connected boundary by the SG-TF, and are difficult to separate. This makes the boundary overlapping problem more challenging in the SG-TF space, and thereby reducing the object classification ability. In Fig. 2, the boundary overlapping problem is illustrated via several real-world 3D images. Fig. 2 shows that when a certain region is selected from the SG-TF space, either boundaries of several structures with similar attributes (Figs. 2a, 2d, 2f, 2g) or boundaries together with some small fragments and uninteresting surface patches (Figs. 2b, 2c, 2e) are visualized simultaneously. Here, uninteresting surface patches refer to the surface patches that are not of interest to the user. Small fragments and uninteresting surface patches usually interrupt the visualization of boundaries of interest, but they cannot be separated from boundaries by the SG-TF because of the boundary overlapping problem.

By utilizing more attributes, HD-TFs usually have better structure-separation ability than LD-TFs [8], and useful interfaces in HD-TFs are studied in [8], [9], [36], [37]. However, HD-TFs have several limitations. First, users usually are unclear about which and how many attributes to use for an appropriate HD-TF. Second, the classification in HD-TFs is usually a black-box operation and users cannot manipulate when necessary [8], [9]. Third, the computational time in some HD-TFs can be very long, as shown in [8]. Finally,

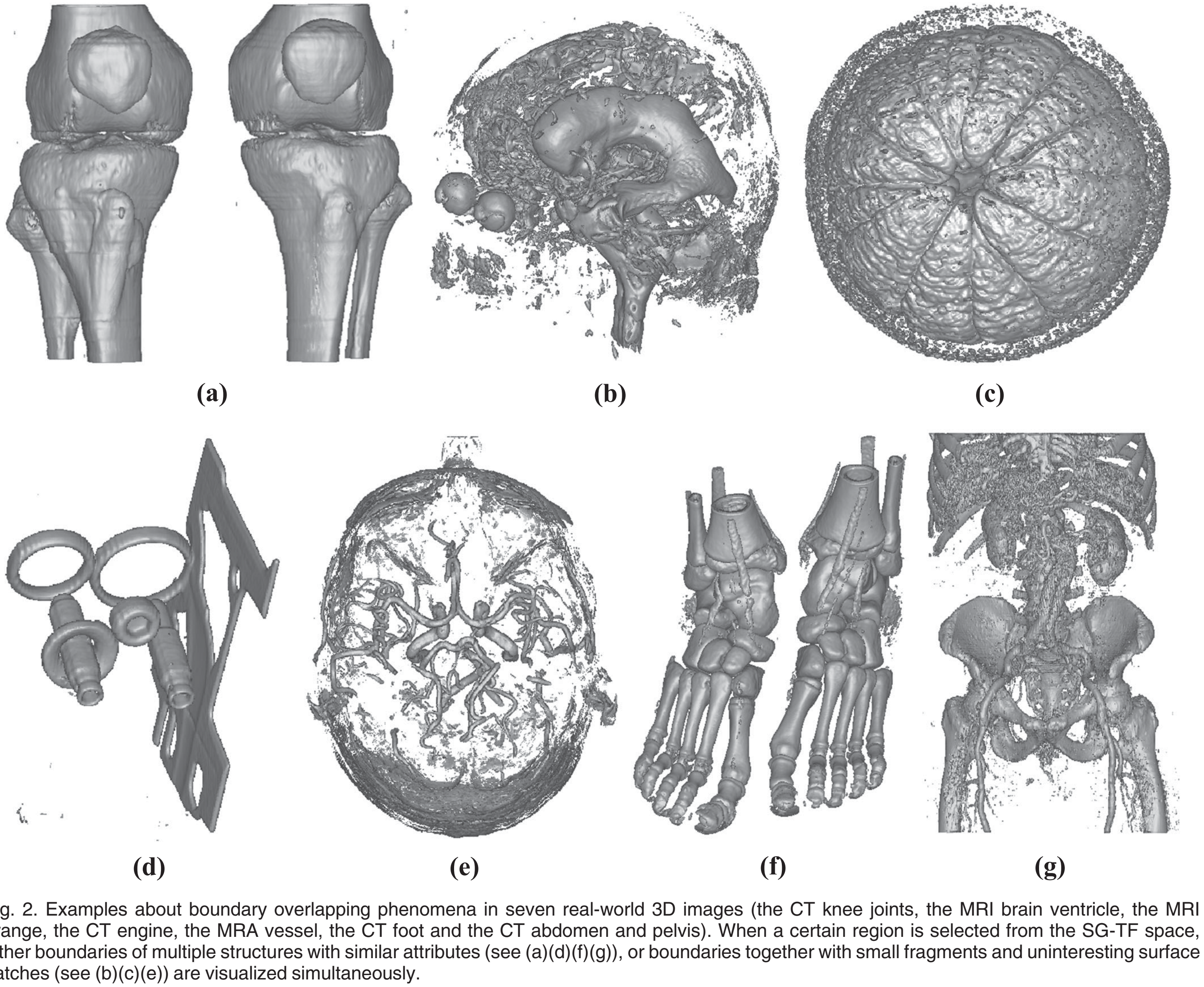


Fig. 2. Examples about boundary overlapping phenomena in seven real-world 3D images (the CT knee joints, the MRI brain ventricle, the MRI orange, the CT engine, the MRA vessel, the CT foot and the CT abdomen and pelvis). When a certain region is selected from the SG-TF space, either boundaries of multiple structures with similar attributes (see (a)(d)(f)(g)), or boundaries together with small fragments and uninteresting surface patches (see (b)(c)(e)) are visualized simultaneously.

HD-TFs have limitations in separating connected boundaries of two closely adjacent structures with similar attributes. Therefore, HD-TFs have limited applications to visualizing complex real-world 3D images.

Currently, the SG-TF is the most widely used transfer function. Therefore, we intend to improve its ability for object classification, and at the same time, avoid the complexity of HD-TFs. For the purpose, a novel framework is proposed to solve the boundary overlapping problem in the SG-TF. In the framework, the spatial connectivity computation of boundaries along with set operations on boundary voxels are integrated into the SG-TF. With the framework, when boundaries of structures with similar attributes are roughly determined by selecting a certain region from the SG-TF space, we can separate different boundaries and exclude small fragments and uninteresting surface patches. Thus, spatial positions of different boundaries, which are of great importance in distinguishing different boundaries, can be determined. Using such information, different boundaries are separated and assigned different colors in the transfer function.

In the proposed framework, boundaries in 3D images are divided into three different classes and different boundary separation techniques are applied to them, respectively. This way the complex task of separating different boundaries in 3D images is simplified by dividing it into several small problems easy to solve. Furthermore, based on the framework, a hierarchical processing strategy is applied to explore and visualize various boundaries in a complex 3D image. All boundaries are first grouped into several groups based on whether they have similar attributes (i.e., the overlapping region in the SG-TF space). Subsequently, all groups are processed in the order, and different boundaries in each group are separated by the proposed framework. This strategy can be used to visualize boundaries of different structures in many complex 3D images. To some extent, it can be regarded as a compromise between the SG-TF and HD-TFs. While it can have advantages of the SG-TF and overcome the boundary overlapping problem in the SG-TF, it can also avoid the complexity of HD-TFs. Its effectiveness has been demonstrated by experimental results from many complex real-world 3D images.

The rest of the paper is organized as follows. Section 2 reviews relevant works. Section 3 describes a framework for solving the boundary overlapping problem in the SG-TF. Section 4 presents a hierarchical strategy to separate various boundaries in complex 3D images. Section 5 shows experimental results in many 3D images. Section 6 discusses some properties of our method and Section 7 concludes the paper.

2 Related Works

The popular LD-TFs mainly include 1D transfer functions (1D-TFs) and 2D-TFs. They are based on a special attribute or two, respectively. 1D-TFs mainly use scalar values to classify 3D images. Automatic detection of salient isosurface was discussed in [20], [21] and [28] by the contour spectra or other features. The contour tree in [12] can separate individual structures as ones in Fig. 1 if only a single scalar is used for classification. However, 1D-TFs have limitations and multi-dimensional transfer functions are suggested [1], [2], [3].

In addition to the SG-TF and LH-TF, many other 2D-TFs have been studied. For example, mean value and standard deviation in the local region of each voxel were used to form a new 2D attribute space [22]. The 2D-TF space formed by the ambient occlusion was used to separate structures of similar scalar values [26]. The size-based transfer function was used to explore complex 3D images by the size [7], [24]. Lundstrom et al. proposed to form a 2D-TF by the local histogram [6]. Different 2D-TFs have their own advantages and are effective on some data sets. Particularly, 1D-TFs and 2D-TFs have a useful merit: they can provide interactive interfaces for users to do intuitive operations, such as the region selection and region adjustment in LD-TF spaces. A graphical user interface of a 2D-TF was used in [25], and the point probe was introduced in [3] for separating from structures around. However, 1D-TFs and 2D-TFs usually suffer the region-overlapping problem in LD-TF spaces, and have limitations in classifying objects [8]. The LH-TF can avoid certain ambiguity existed in the SG-TF space [5], but it cannot process the boundary overlapping problem as shown in Fig. 1.

Automatic segmentation of the SG-TF space and the LH-TF space was discussed in [4], [29], [30], [31], [32], [43], [44]. Its purpose is to break a 2D-TF space into different unoverlapping regions so that each structure can correspond to a region. However, a segmented region may correspond to multiple different boundaries, and a boundary might be broken into several surface patches because too many regions are segmented from a LD-TF space [4], [31], [32]. The segmentation of a LD-TF space may be improved by incorporating the spatial information of voxel positions into the LD-TF space [4], [31], [32], [33]. The number of segmented regions may be reduced by merging spatially adjacent regions [31], [32]. However, the parameters for generating a good segmentation result are not easy to determine. Additionally, connected boundaries of two adjacent structures with similar attributes cannot be separated by clustering attributes in a LD-TF space, or by clustering spatial positions of voxels.

Some researchers tried to overcome the limitations of LD-TFs by combining or fusing different LD-TFs [10], [11]. In such techniques, different LD-TFs were combined to separate a structure that otherwise cannot be done by each LD-TF [11], or a new transfer function was generated by fusing two different LD-TFs that correspond to two different structures respectively [10]. These techniques may improve previous LD-TFs by using more attributes. However, in the new transfer function space, the spatial information is not used, and the region overlapping problem still exists.

LD-TFs usually fail in many 3D images, because they utilize fewer attributes for object classification. Thus, HD-TFs were studied [8], [9], [36], [37], [38], [39]. The modified dendrogram was introduced in [8] as a useful 2D user interface. By the interface, users can design a HD-TF in an intuitive and informative manner. However, it takes long computational time for this method to classify large 3D images. An intelligent system was used for high-dimensional classification of 3D images [9], where a 2D interface was provided to interactively paint the region of interest from many 2D slices [36], [37]. Then, machine learning was used for classifying the object of interest from the 3D image. In such intelligent systems, different training data are needed when the object of interest is different. HD-TFs usually have better object classification ability than that of LD-TFs, but they have several limitations mentioned in Section 1.

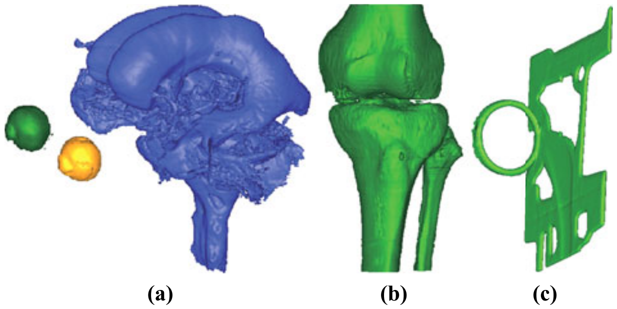


Fig. 3. Typical examples of boundaries of structures in $\Omega_1$or Ω_2 or Ω_3. (a) boundaries of three far away structures in Ω_1. (b) boundaries of two closely adjacent structures in Ω_2, being connected due to the PVE. (c) boundaries of two touched structures in Ω_3.

3 A Framework for Solving the Boundary Overlapping Problem in the SG-TF

In this section, we first describe the boundary overlapping problem in the SG-TF space, and then propose a framework to solve the problem.

3.1 The Boundary Overlapping Problem in the SG-TF

When a region U is selected from the SG-TF space of a 3D image $f(x, y, z)$ for a structure G, a set of voxels (denoted by Φ_U) is determined from $f(x, y, z)$. Φ_U contains not only boundary voxels of G, but also boundary voxels of the structures with similar attributes as G. Additionally, many small fragments and uninteresting surface patches are contained in Φ_U as well, as shown in Fig. 2. However, users do not know how the voxels in Φ_U are classified into each boundary, each fragment and each uninteresting patch. In such case, the boundary overlapping problem can be described as follows:

(P) Separate or extract different boundaries (or their boundary voxels) from Φ_U, and exclude small fragments and uninteresting surface patches from Φ_U.

In Φ_U, suppose that boundaries of several structures with similar attributes are represented by $S_i, i = 1, 2, \ldots, m$, respectively. Fig. 2 illustrates that S_i (or their corresponding structures) may have very complex spatial relationships. For example, two structures with similar attributes might be far from each other, or closely adjacent or touched. Particularly, because of the PVE, boundaries of two closely adjacent structures are often mistakenly regarded as a connected boundary. This makes the problem (P) become more complicated. Instead of processing all boundaries in Φ_U by the same technique, in this paper we will divide boundaries in Φ_U into different classes, and target with different processing techniques, respectively. Such processing mode can help to simplify the problem (P).

According to different spatial relationships, the structures with similar attributes in the 3D image are divided into the following three classes:

(i) A structure belongs to Ω_1 if its boundary does not connect spatially with boundaries of any other structure with similar attributes. See three such structures shown in Figs. 3a and 2b.

(ii) A structure belongs to Ω_2 if it is closely adjacent to (but not touched with) certain structures with similar attributes, and its boundary is (mistakenly) connected with boundaries of these structures due to the PVE. Two bone structures of the CT knee joint shown in Figs. 3b and 2a belong to Ω_2.

(iii) A structure belongs to Ω_3 if it is spatially touched with some structures with similar attributes. Its boundary is connected with boundaries of these structures. See two such structures in Fig. 3c.

Correspondingly, boundaries in Φ_U can also be classified into the same three classes, namely belonging to Ω_1 or Ω_2 or Ω_3. Most human organs in 3D medical images should be far from each other or closely adjacent. Some parts of 3D industrial CT images may be spatially touched, as illustrated in Fig. 3c. Because of the limitation of the imaging resolution, however, sometimes two structures which should be closely adjacent in 3D space are possibly touched in the sampled 3D image.

Based on the classification above, the boundary overlapping problem in Φ_U can be roughly formulated into the following small problems: how different boundaries in Ω_1 or Ω_2 or Ω_3 are separated from each other? In real-world 3D images, structures belonging to Ω_1 and Ω_2 are usually seen. Thus, we will focus on the techniques for separating boundaries of structures in Ω_1 and Ω_2. The effective algorithms for the Ω_3 case are data dependant, we will present a general idea.

3.2 The Framework for Solving the Boundary Overlapping Problem in Φ_U

In this section, a framework is proposed to solve the problem (P). It integrates those techniques that are used to separate boundaries of different structures in Ω_1 or Ω_2 or Ω_3 and to exclude small fragments and uninteresting surface patches from Φ_U. It mainly includes the following steps:

(i) Extract all connected sets from Φ_U by the spatial connectivity. Small fragments are excluded from Φ_U as small connected sets.

(ii) Classify visually the remaining large connected sets in Φ_U into four types: boundaries of structures in Ω_1 or Ω_2 or Ω_3 and uninteresting surface patches. By the user's observation, uninteresting surface patches are excluded from Φ_U and boundary of each structure in Ω_1 is separated from Φ_U.

(iii) Separate boundary of each structure in Ω_2 or Ω_3 or $\Omega_2 \cup \Omega_3$ (e.g., the combination of structures in Ω_2 and Ω_3) by the particularly developed techniques.

Finally, all boundaries are extracted from Φ_U, separately. The flowchart of the framework is shown in Fig. 4. Below, we will introduce each step in detail.

3.2.1 *Computation of Spatial Connectivity of Boundaries in Φ_U*

Each boundary in 3D images is a spatially connected surface [46]. When a suitable U is selected from the SG-TF space, each boundary formed by lots of boundary voxels in Φ_U will keep the connectivity as well. Thus, boundaries of different structures in Ω_1 are different connected sets in Φ_U. Boundaries of several closely adjacent structures in Ω_2 are usually classified incorrectly as a connected set. The boundaries of several touched structures in Ω_3 will also be a connected set. In addition, even small fragments and

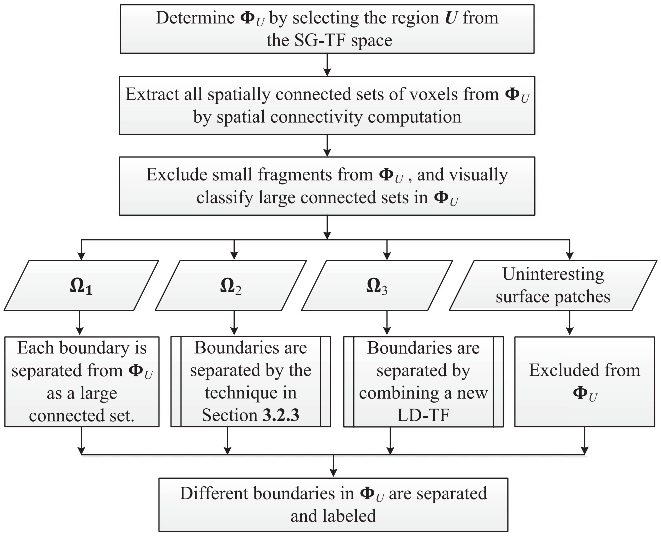

Determine $\mathbf{\Phi}_U$ by selecting the region U from the SG-TF space
Extract all spatially connected sets of voxels from $\mathbf{\Phi}_U$ by spatial connectivity computation
Exclude small fragments from $\mathbf{\Phi}_U$, and visually classify large connected sets in $\mathbf{\Phi}_U$
$\mathbf{\Omega}_1$
$\mathbf{\Omega}_2$
$\mathbf{\Omega}_3$
Uninteresting surface patches
Each boundary is separated from $\mathbf{\Phi}_U$ as a large connected set.
Boundaries are separated by the technique in Section **3.2.3**
Boundaries are separated by combining a new LD-TF
Excluded from $\mathbf{\Phi}_U$
Different boundaries in $\mathbf{\Phi}_U$ are separated and labeled

Fig. 4. The flowchart of the framework for solving the boundary overlapping problem in the SG-TF space.

uninteresting surface patches are also different connected sets in Φ_U. These different connected sets may be extracted from Φ_U by the following tracing algorithm:

Step 1 Mark all voxels in Φ_U as not-visited ones.

Step 2 Perform the following operations (Step 2.1-Step 2.2) circularly until all voxels in Φ_U are marked as the voxels visited.

Step 2.1 Select a not-visited voxel from Φ_U as a seed, mark it as a visited voxel and push it into a stack denoted by H.

Step 2.2 Perform the following operation (Steps 2.2.1-2.2.2) circularly until H is empty:

Step 2.2.1 Pop up a voxel from H.

Step 2.2.2 Consider all voxels in the 18-neighborhood of the voxel, respectively. If a voxel is not in Φ_U or is visited before, then skip it. If a voxel is in Φ_U and not visited before, then push it into H, and mark it as the visited voxel.

Finally, all connected sets in Φ_U are extracted. These connected sets are ranked by the voxel number in each connected set and numbered. Usually, small fragments correspond to small connected sets (containing less voxels). Thus, they can be easily recognized and excluded from Φ_U by the property. This is illustrated in Fig. 5, where Φ_U with or without small fragments are visualized. Remaining large connected sets (containing more voxels) correspond to boundaries of different structures in Ω_1, or boundaries of closely adjacent structures in Ω_2 , or boundaries of touched structures in Ω_3, or uninteresting surface patches. They will be further classified.

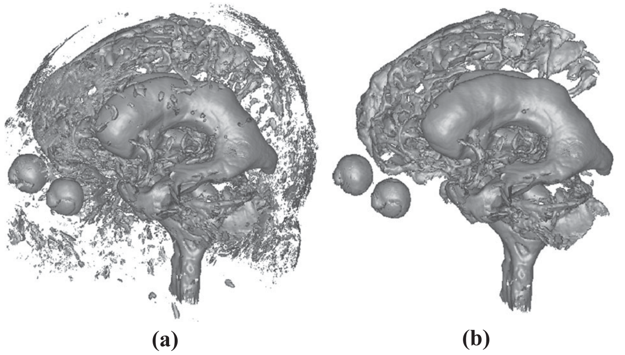


Fig. 5. Visualization Φ_U with and without small fragments.

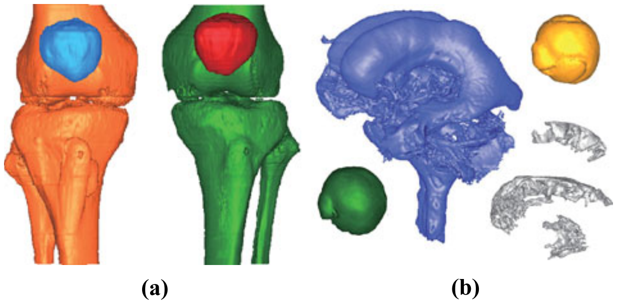


Fig. 6. Visualization of large connected sets in Φ_U. (a) four large connected sets in the CT knee joint are visualized simultaneously with different colors. (b) 6 large connected sets in the MRI brain ventricle are visualized, individually.

3.2.2 *Visual Classification of Large Connected Sets in Φ_U*

A visualization-based technique is used to classify visually large connected sets in Φ_U. With the technique, all large connected sets in Φ_U can be visualized simultaneously with different colors, as in Fig. 6a. Alternatively, each large connected set can also be individually visualized, as in Fig. 6b. By observing these visualization results, users can easily distinguish and classify large connected sets in Φ_U into the four classes: boundaries of structures in Ω_1, or boundaries of closely adjacent structures in Ω_2, or boundaries of touched structures in Ω_3, or uninteresting surface patches. For example, in Fig. 6b, 6 large connected sets in Φ_U can be visually classified into three boundaries in Ω_1 and three uninteresting surface patches. In Fig. 6a, four large connected sets can be visually divided into two boundaries in Ω_1 and two groups of boundaries in Ω_2.

While large connected sets corresponding to uninteresting surface patches are excluded from Φ_U, large connected sets corresponding to boundaries of structures in Ω_1 are marked and separated from Φ_U. The remaining large connected sets, which correspond to boundaries of several connected structures in Ω_2 or Ω_3 or $\Omega_2\cup$ Ω_3, will be further processed. We only need to study the problem (P) in the former two cases, and the $\Omega_2\cup\Omega_3$ case can actually be decomposed into the former two cases.

3.2.3 *Separation of Different Boundaries in Ω_2*

This section discusses how different boundaries in a large connected set $\Phi \in \Omega_2$ are separated from Φ. Without loss of generality, we assume that Φ contains only two boundaries S_1, S_2 of two closely adjacent structures (denoted by D_1, D_2).

Let D denote the thin region in between D_1 and D_2, as in Fig. 7a. Because of the PVE, scalar values of the voxels in D are actually generated by mixing the scalar values of multiple voxels (some voxels in D and some in D_1 and/or D_2) [27]. Consequently, the region D usually has scalar values that are smaller than those of two adjacent ends of D_1 and D_2, and overlaps partially with scalar values of D_1 and/or D_2, as in Fig. 7b. When the region U is selected from the SG-TF space, many voxels of D (denote by Λ) are classified

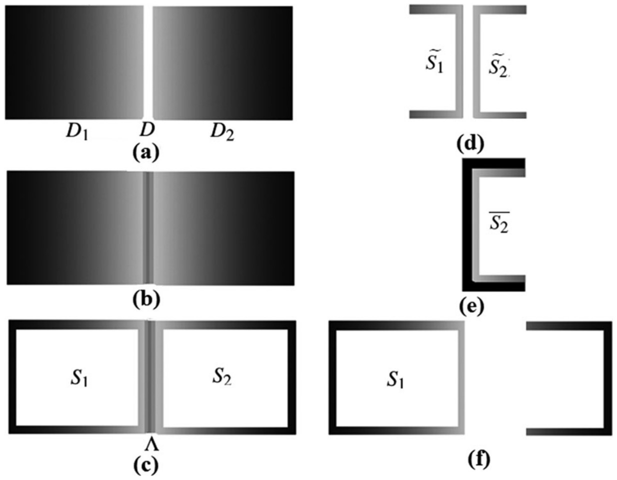


Fig. 7. Illustrations on how boundaries of two closely adjacent structures in Ω_2 are separated. (a) the thin region D in between two closely adjacent structures D_1 and D_2. (b) due to the PVE, D will have scalar values less than ones of two adjacent ends of D_1 and D_2. (c) boundaries $\Phi = S_1 \cup S_2 \cup \Lambda$ and S_1 and S_2 (boundaries of D_1 and D_2) are connected by Λ (boundary voxels in D). (d) S_1 and S_2 are eroded as two incomplete boundaries $\widetilde{S_1}, \widetilde{S_2}$ when Λ is eroded from Φ. (e) the dilated set $\overline{S_2}$ of $\widetilde{S_2}$, $\overline{S_2}$=the dark region + $\widetilde{S_2}$. (f) the set $\Phi - \overline{S_2}$, where S_1 is isolated as a connected set.

incorrectly as boundary voxels in Φ_U, and thus boundaries S_1 and S_2 are incorrectly connected by these voxels, as in Fig. 7c. We have $\Phi = S_1 \cup S_2 \cup \Lambda$. This fact suggests that we can disconnect S_1 and S_2 by removing Λ from Φ. Motivated by this, a technique is proposed to separate S_1 and S_2 from Φ, which is described as follows:

First, a special erosion operation is used to erode Λ *from* Φ**.** Generally, Λ cannot be eroded from Φ by the conventional spatial erosion operation. So, we will introduce a new erosion operation. Assume that V_Φ is the region selected in the SG-TF space for Φ (in this paper we assume $V_\Phi = U$). Denote the scalar range of V_Φ by $[V_a, V_b]$. If we gradually shrink V_Φ from left (or right) by gradually increasing V_a (or decreasing V_b), then some voxels will be gradually removed from Φ. We call such shrinking-operation of V_Φ as the erosion operation of Φ. Λ can be eroded from Φ by using such an erosion operation. The erosion operation can be detailed as follows:

Step 1 Mark S_1, S_2 in Φ by interactively drawing two small seed regions on S_1, S_2 (as in Fig. 8a), respectively. Let $n = 1, \Phi_0 = \Phi$.

Step 2 Perform the following operations (step2.1 - step2.3) circularly until two seed regions belong to two different connected sets:

Step 2.1 $\Phi_n = \Phi_{n-1} - \{(x, y, z) : f(x, y, z) = V_a + n\}$. Let $n{+}{+}$.

Step 2.2 Compute connected sets from Φ_n.

Step 2.3 Judge whether two seed regions are in the same connected set in Φ_n.

After marking S_1, S_2, Λ will be automatically eroded from Φ, and S_1 and S_2 are disconnected but they are eroded as two incomplete boundaries (denoted by $\widetilde{S_1}, \widetilde{S_2}$, respectively), as in Figs. 7d and 8b. $\widetilde{S_1}$ and $\widetilde{S_2}$ are two connected sets in Φ_n and each one contains a seed region.

Secondly, S_1 *and* S_2 *are separated automatically from* Φ**.** This is implemented by performing automatically the following operations:

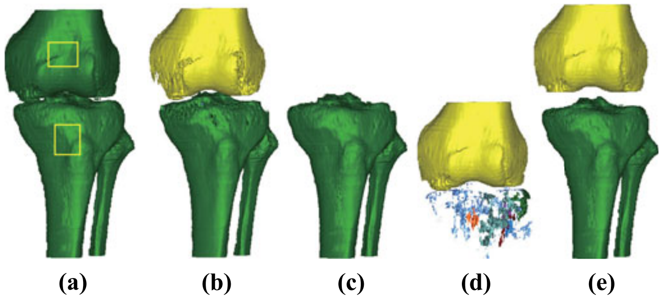


Fig. 8. An example about the separation of boundaries of two adjacent structures in Ω_2. (a) connected boundaries of two adjacent bone structures in Fig. 3b. S_1 and S_2 are marked by interactively drawing two small seed regions on them. (b) S_1 and S_2 are disconnected and eroded as $\widetilde{S_1}$ and $\widetilde{S_2}$ by the erosion algorithm. (c) $\overline{S_2}$, the dilated set of $\widetilde{S_2}$ in the 3D image. (d) different connected sets in $\Phi - \overline{S_2}$. (e) separated boundaries of two bone structures.

Step 1 Dilate spatially $\widetilde{S_2}$ in the 3D image with a given radius r ($r = 3$ in this paper) so that Λ is included in the dilated $\widetilde{S_2}$. Denote the dilated $\widetilde{S_2}$ by $\overline{S_2}$.

Step 2 Subtract $\overline{S_2}$ from the original Φ (namely, $\Phi - \overline{S_2}$).

Step 3 Extract the largest connected set from $\Phi - \overline{S_2}$, which is S_1.

Step 4 Separate S_2 from Φ by the same steps 1-3 as above based on $\widetilde{S_1}$, or alternatively, by extracting the largest connected set from $\Phi - \widehat{S_1}$, where $\widehat{S_1}$ is a spatially dilated set of S_1 in the 3D image.

In Step 1, if $r = 1$, then the voxels that are in 26-neighborhood of voxels in $\widetilde{S_2}$ will all be marked as belonging to $\overline{S_2}$. By performing such spatial dilation operation several times, $\widetilde{S_2}$ may be dilated with a radius $r > 1$. $\overline{S_2}$ includes most voxels of Λ and an expanded $\widetilde{S_2}$, as illustrated in Figs. 7e and 8c. The set $\Phi - \overline{S_2}$ includes all boundary voxels of S_1, partial boundary voxels of S_2 (e.g., $S_2 - \overline{S_2}$) and some other small fragments, see Figs. 7f and 8d. In $\Phi - \overline{S_2}$, S_1 is isolated as a connected set. Therefore, S_1 can be automatically separated from Φ by computing the largest connected set from $\Phi - \overline{S_2}$.

Fig. 8 illustrates the procedure to separate boundaries of two closely adjacent structures by the proposed technique.

3.2.4 Separation of Different Boundaries in Ω_3

This section discusses how different boundaries in a large connected set $\Phi \in \Omega_3$ are separated from Φ. To simplify the discussion, we assume that Φ contains only boundaries of two touched structures in Ω_3, namely, $S_1, S_2 \subset \Phi$.

Generally, S_1 and S_2 cannot be disconnected by the erosion operation. It is necessary to seek other attributes (such as shape, curvature, topological attributes [45], spatial distribution of X or Y or Z coordinate, etc.) to separate S_1 and S_2. This means that S_1 and S_2 will be disconnected by combining a new LD-TF [11]. When a new and suitable LD-TF is determined, the attribute distribution of S_1 and S_2 in the new LD-TF space can be displayed. Based on the observation, users may disconnect S_1 and S_2 by interactively selecting two different regions in the new LD-TF space.

Fig. 9 shows an example to illustrate how boundaries of two touched structures are separated by the proposed technique. In Fig. 9b, the histogram of z-coordinates of all boundary voxels of two touched structures is shown. By

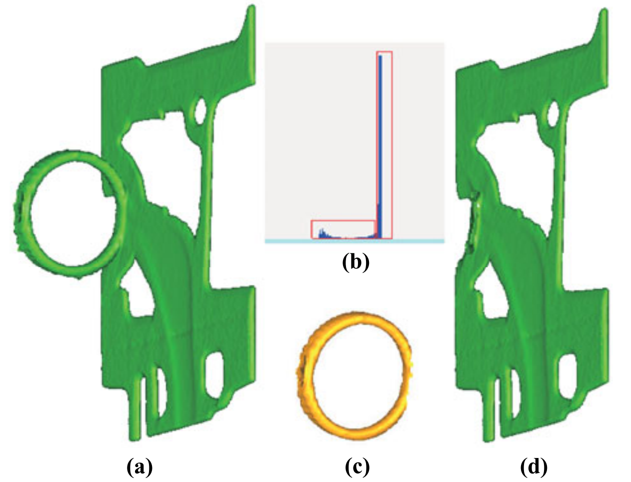


Fig. 9. An example about the separation of two touched structures in Ω_3. (a) a panel is directly touched with a ring, see Fig. 3c. (b) a new LD-TF: the histogram of z-coordinates of all boundary voxels of the two structures. (c)-(d) The ring is separated from the panel by selecting two regions in the LD-TF, as shown in (b).

selecting two different regions from the histogram, the two touched structures are separated, as shown in Figs. 9c and 9d. We observe that, many structures belonging to Ω_3 are contained in 3D industrial CT images, and these structures are usually artificial and have regular shapes. In many cases, boundaries of such touched structures can be separated by a special plane. Therefore, the idea in Fig. 9 can be applied to such case of Ω_3. However, in most cases, to select a suitable new LD-TF is application or data-dependent.

3.3 An Example to Illustrate the Framework

In Fig. 10, an example is provided to illustrate the framework in Section 3.2. First, boundaries of seven different structures with similar attributes are determined by selecting a region from the SG-TF space of the CT engine. After the computation of spatial connectivity, small fragments are excluded from Φ_U, and four different large connected sets (containing seven boundaries) are visualized in different colors and visually classified (two in Ω_1, one in Ω_2 and one in $\Omega_2 \cup \Omega_3$). The large connected set in Ω_2 are disconnected into two different boundaries using the techniques in Section 3.2.3. The large connected set in $\Omega_2 \cup \Omega_3$ is first disconnected into two different connected subsets (one in Ω_3) by the techniques in Section 3.2.3, and then the connected subset in Ω_3 is further disconnected into two different boundaries by combining a new LD-TF as in Fig. 9. Fig. 10 shows how different boundaries are separated from Φ_U.

4 A Hierarchical Processing Strategy for Separating Various Boundaries in Complex 3D Images

A complex 3D image usually contains multiple structures, which can be categorized into several different groups based on their similar attributes. With this fact, in this section we propose a hierarchical processing strategy to explore and visualize different structures in complex 3D images. Below, we first discuss three related problems. Then we will introduce the hierarchical processing strategy in detail.

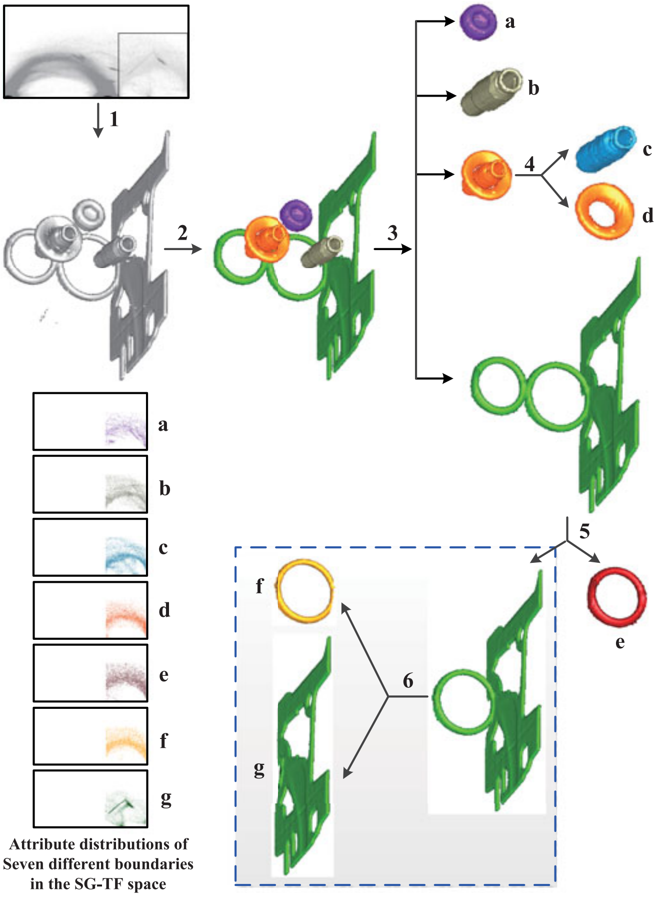


Fig. 10. An example to illustrate the proposed framework, in which seven different boundaries (a-g) are separated from Φ_U and 6 main operations (1-6) are done in the separation. 1: roughly determining Ω_U by selecting a region in the SG-TF space and visualize Ω_U; 2: exclude small fragments and visualize four large connected sets in Ω_U with different colors. 3: visually classify four large connected sets: two in Ω_1, one in Ω_2 and one in $\Omega_2 \cup \Omega_3$. 4: separate boundaries of two structures in Ω_2. 5: separate boundaries of three structures into two connected subsets by the technique in Section 3.2.3. One subset is in Ω_3. 6: separate boundaries of two touched structures ($\in \Omega_3$) by combining a new LD-TF, as in Fig. 9. Attribute distributions of seven different boundaries (a-g) in the SG-TF space are also shown respectively.

4.1 Two Useful Skills for Improving Boundary-Separation in Different Groups

In real-world 3D images, the SG-TF space usually has a much complex distribution and is not easy to understand. In fact, the SG-TF space can be regarded as the superposition of attribute distributions of different boundaries, small fragments and uninteresting surface patches. By the property, whenever a boundary is separated, its attribute distribution will no longer be counted in the SG-TF space. In this way, the distribution of the SG-TF space can be simplified and become easier to understand. So, users can more easily select, adjust and refine the regions for the structures remaining to be separated. In Fig. 14, the original SG-TF space is simplified twice after two groups of boundaries are separated.

When a region U is selected from the SG-TF space for boundaries in a group, Φ_U might also include surface patches of boundaries in other groups. This greatly affects the separation of boundaries in different groups. To overcome such a limitation, when some boundaries are separated, their boundary voxels are marked in the 3D image and are fixed. Here, "fixed" means that these voxels will not

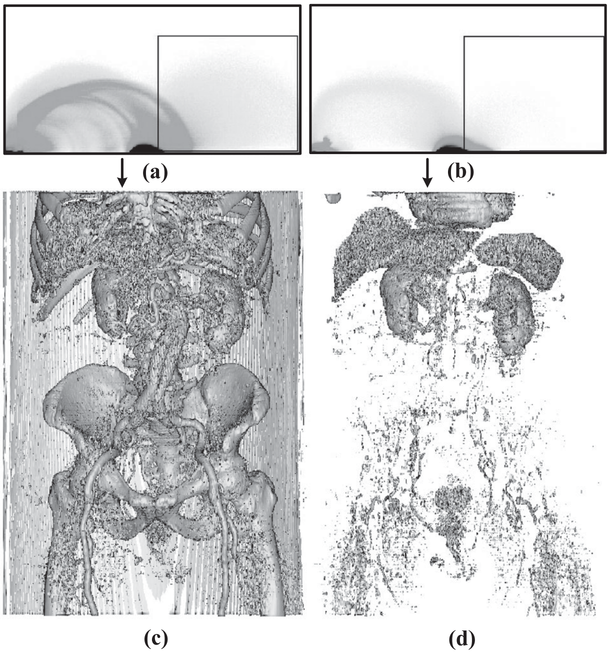


Fig. 11. An example for illustrating two skills in Section 4.1. (a) the original SG-TF space of the CT abdomen and pelvis. (b) the simplified SG-TF space after skin surface, lung boundaries, bone surfaces, large blood vessels all are separated from Φ_U and their boundary voxels are fixed. (c)-(d) different boundary voxels determined by selecting the same region from the two different SG-TF spaces in (a)-(b).

be included in Φ_U when a new region U is selected from the SG-TF space, even if these voxels belong to Φ_U. This way we can reduce the complexity of Φ_U of a new group, and possibly disconnect boundaries that are connected by the fixed voxels.

An example is shown in Fig. 11 for illustrating two skills above. Fig. 11a shows the original SG-TF space of the CT abdomen and pelvis. After having separated skin surface, lung boundaries, bone surfaces, large blood vessels, their boundary voxels are fixed and a simplified SG-TF space is generated in Fig. 11b. When selecting the same region from the two SG-TF spaces, two sets of greatly different boundary voxels are determined, as in Figs. 11c and 11d. Boundaries in Fig. 11d are easy to separate visually.

4.2 Boundary Repairing by the Dilation Operation

In volume rendering, a boundary is a thick surface without holes. Due to the compact selection of U or other reasons, however, a boundary with small holes or without outer boundary voxels might be included in Φ_U, as shown in Fig. 12a. In this case, it is necessary to repair the boundary. Since the holes are very small and the lost outer boundary is usually thin, we will repair a boundary by spatially dilating it in the 3D image with a small radius. By the dilation, lost boundary voxels may be recovered.

After the dilation operation, other voxels that are not in boundary might also be marked incorrectly as boundary voxels. If they are assigned the same opacity values as true boundary voxels, the visualization results of the repaired boundaries might be greatly distorted, as shown in Fig. 12b. These voxels usually have very small gradient values and comparatively large distances to the original boundaries. So, we adjust opacity values of these voxels by their gradient magnitudes and distances. This can greatly improve the visualization results of restored boundaries, as illustrated in Fig. 12c.

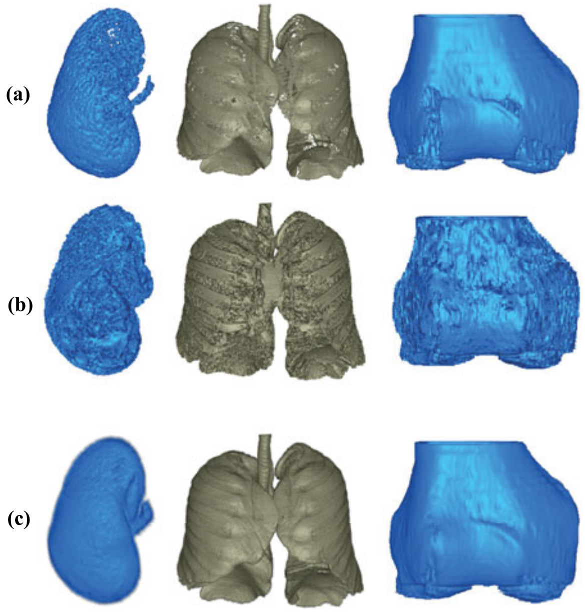


Fig. 12. Examples for illustrating the usefulness of the semi-transparent visualization in the repaired boundaries. (a) three boundaries before repaired. (b)-(c) visualization of repaired boundaries when semi-transparent opacity values are not set (see (b)) or set (see (c)) to expanded voxels.

4.3 Region Selection in the SG-TF Space

In the SG-TF space, the exact region corresponding to a boundary is generally unknown. Users usually roughly determine boundary of interest by interactively selecting a region from the SG-TF space [3], [5]. However, the visualization is usually disturbed by small fragments and other surface patches, as shown in Fig. 2. In this paper, the region U is acceptable as long as it satisfies the following two conditions:

(i) Φ_U includes at least most boundary voxels of boundaries of interest.
(ii) Boundaries of interest in Φ_U can be cut apart from other boundaries, or do not link to other boundaries.

When such a U is selected, Φ_U might include small fragments and uninteresting surface patches, and the contained boundary might have small holes. However, the method in this paper can well handle them. This relaxes the selection condition of the region U.

4.4 Hierarchical Processing Strategy

The flowchart of the hierarchical processing strategy is described in Fig. 13. By the strategy, all boundaries in a 3D image are first grouped into several groups by selecting different regions from the SG-TF space. Next, different groups are processed in the order, and boundaries in each group are separated by the framework in Section 3.2. Finally, several groups of boundaries with different attributes are separated.

Each separated boundary has two kinds of information: the spatial position in the 3D image and the corresponding region in the SG-TF space. The transfer function can be designed based on such information, and different boundaries are assigned with different colors, respectively. By

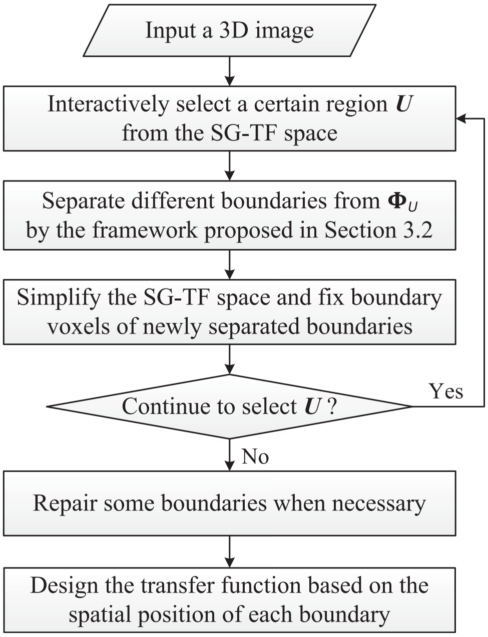

Input a 3D image
Interactively select a certain region U from the SG-TF space
Separate different boundaries from Φ_U by the framework proposed in Section 3.2
Simplify the SG-TF space and fix boundary voxels of newly separated boundaries
Continue to select U ?
Yes
No
Repair some boundaries when necessary
Design the transfer function based on the spatial position of each boundary

Fig. 13. The flowchart of the hierarchical processing strategy for separating various boundaries from a complex 3D image.

assigning low opacity values to some boundaries, the occlusion problem of these boundaries can be avoided [40].

In Fig. 14, the hierarchical processing strategy is illustrated by an example from the CT abdomen and pelvis. All boundaries in the 3D image are grouped into three groups, and 10 different boundaries are separated from the three groups. Fig. 14 demonstrates how different boundaries are hierarchically separated in complex 3D images.

5 Experimental Results

The proposed method has been applied to different 3D images from real-world. For the comparison with visualization results in Fig. 2, experimental results from the same seven 3D images as in Fig. 2 are provided in Fig. 15. The seven images include the CT knee joints, MRI brain ventricle, MRI orange, CT engine, MRA vessel, CT foot, CT abdomen and pelvis. In Fig. 2, due to the boundary-overlapping problem in the SG-TF space, either boundaries of multiple structures or boundaries along with many small fragments and uninteresting surface patches are volume rendered in one color. By the proposed method in this paper, not only small fragments and uninteresting surface patches can be excluded from boundaries of interest (see Figs. 15b, 15c, 15e), but also different boundaries of interest can be separated and visualized with different colors (see Figs. 15a, 15d, 15f, 15g). Figs. 2 and 15 demonstrate that the proposed method has a very good ability in object classification comparing with the SG-TF.

In Figs. 10 and 14, attribute distributions of 7 and 10 different boundaries in the SG-TF space are displayed, respectively. It can be seen that they are really overlapped in the SG-TF space. Particularly, in the CT abdomen and pelvis, CT engine, CT knee joints and CT foot, boundaries of many structures with similar attributes are connected, as shown in Figs. 2g, 2d, 2a, 2f. Aorta and spine in Fig. 2g belong to Ω_2, and nearly all bone structures in Fig. 2f belong to Ω_2. Figs. 10 and 14 show several groups of structures belonging to Ω_2 and/or Ω_3 . Such boundary-overlapping problems are challenging for separating different boundaries from these 3D images with existing approaches. Our proposed method, however, can well handle them, as shown in Figs. 15 and 16.

3D CT images of human abdomen, pelvis and chest contain many anatomical structures with different attributes and complex spatial relationships. Thus, boundary-overlapping is prevalent in them. This makes it very complex to classify different anatomical structures from them. The SG-TF usually cannot well classify such CT images. With the proposed method, however, boundaries of many anatomical structures can be better separated. This is illustrated in Figs. 15g, 16a, 16b, 16c, 16d, where at least seven different anatomical boundaries are separated from each 3D image and visualized with different colors. Each individual anatomical boundary can also be visualized, as in Fig. 17. The CT abdomen and pelvis in Fig. 2g is from a patient with a stent, which is located between two large blood vessels. Figs. 14 and 15g show that the stent can be well separated from vessels. Figs. 14, 15, 16, 17 demonstrate that the proposed method has a strong ability to classify objects from complex 3D images.

In Table 1, we provide the statistics on the size of each 3D image in Figs. 15 and 16, the number and classification of separated boundaries from each 3D image, the experts' evaluation on the visualization result of each 3D image, the time used to separate all boundaries from each 3D image. Here, the time is counted from the selection of U to complete the visualization of all separated boundaries. All experiments are run on a PC with Intel Core i7-4470k@3.50 GHz, 8 GB RAM and Nvidia GeGorce GTX 760. The scores of experts' evaluation were from four independent clinical experts, as shown in Section 6.2, which further validated the practicability and acceptability of the proposed approach.

The 3D images in Figs. 15 and 16 were freely downloaded from http://lgdv.cs.fau.de/External/vollib/ (Figs. 15a, 15c, 15d), http://www.osirix-viewer.com/datasets/ (Figs. 15f, 16a, 16b, 16c, 16d) and www.volvis.org (Figs. 15b, 15e, 15g,). Thanks for sharing data.

6 Discussions

6.1 The Ability for Classifying Different Objects

The boundary overlapping problem often exists and it limits the ability of the SG-TF for classifying different objects from 3D images. With the proposed framework in Section 3.2, however, different boundaries, small fragments and uninteresting surface patches can now be well separated even if they have an overlapping region in the SG-TF space. For example, different boundaries that cannot be separated by the SG-TF previously (see Fig. 2) can now be better separated (see Fig. 15).

Generally, it is a difficult task to classify all different boundaries from a complex 3D image [8], because they may have complex spatial relationships and different attributes. Using the hierarchical processing strategy proposed in this paper, we separate different boundaries from each group. The strategy is very effective in many complex real-world 3D images, as demonstrated in Figs. 14, 15f, 15g, 16a, 16b, 16c, 16d. 47 bone structures are separated from the CT foot

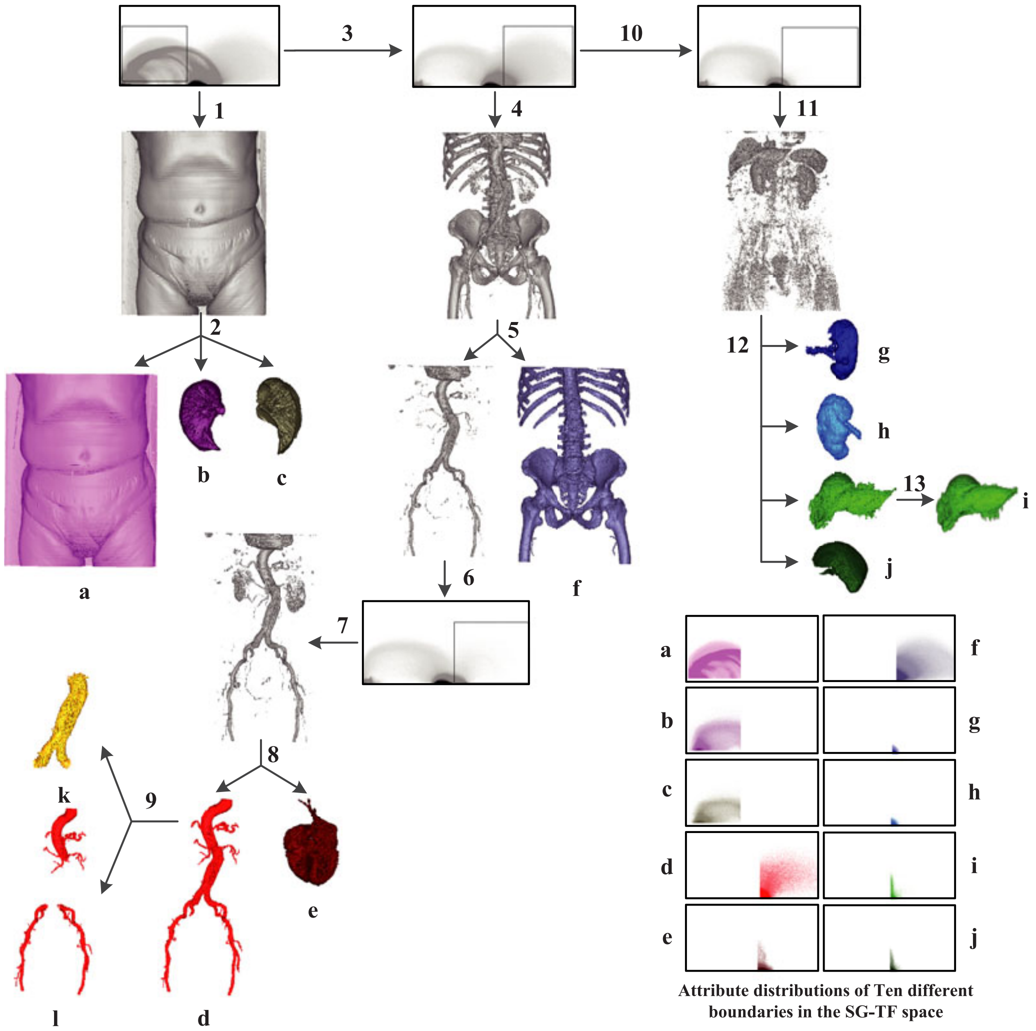

3
10
1
4
11
2
5
12
g
h
13
i
j
a
b
c
f
6
7
8
9
k
l
d
e
a
b
c
d
e
f
g
h
i
j
Attribute distributions of Ten different boundaries in the SG-TF space

Fig. 14. The hierarchical processing procedure for separating 10 different boundaries (a-j) from the CT abdomen and pelvis. All boundaries are grouped into three groups by the operations 1,4 and 11, and 13 operations (1 - 13) are done. Different groups are processed in order, and boundaries in each group are separated by the proposed framework (see 1-2, 4-9 and 11-12, respectively). 3, 6, 10: simplify the SG-TF space after certain boundaries are separated, and fix boundary voxels of separated boundaries. 1, 4, 7 and 11: roughly determining boundary voxels Φ_U by selecting certain regions from SG-TF spaces; 2, 8, 12: exclude small fragments and visually classify different large connected sets. Uninteresting surface patches are excluded and boundaries in Ω_1 are separated; 5 and 9: disconnecting boundaries of structures in Ω_2; 13: the boundaries i is further repaired by the technique in Section 4.2. Attribute distributions of 10 different boundaries (a-j) in the SG-TF space are shown, respectively. They are really shown as three different overlapping regions

(Fig. 15f), and 7 to 10 anatomical structures from 5 CT images of human abdomen, pelvis and chest (Figs. 15g, 16a, 16b, 16c, 16d).

6.2 Clinical Evaluations of Classification Results

Four clinical experts were asked to evaluate the visualization results in Figs. 15, 16, 17, independently. They evaluated the correctness and acceptability of each visualization result, and gave respective scores. Here, the scores reflect the subjective evaluation of these experts with ranks as follows: very poor-1, poor-2, fair-3, good-4, very good-5, perfect-6. For the CT knee joints, MRI brain ventricle, MRA vessel and CT foot, the visualization results of all separated boundaries were evaluated and given a score by every expert. However, CT images of human abdomen, pelvis and chest contain many different organs, such as spine (or bone structure), two lungs, heart, two livers, large blood vessel, two kidneys, stomach, spleen, etc. So, each organ was evaluated by every expert individually. Let $S_{k,i,n}$ represent the score given by the nth expert to the ith organ separated from the kth 3D image, $k = 1, 2, \ldots, p$, $i = 1, 2, \ldots, m$, $n = 1, 2, 3.4$. Denote $\Psi_{k,i} = \sum_{n=1}^{4} S_{k,i,n}$, $L_{k,i} = \frac{\Psi_{k,i}}{4}$, $\Gamma_k = \sum_{i=1}^{m} L_{k,i}$, $\Xi_i = \sum_{k=1}^{p} L_{k,i}$. Then $L_{k,i}$ is the score of the ith organ in the kth 3D image. The mean $\frac{\Gamma_k}{m}$ roughly indicates whether different organs in the kth 3D image are well classified, and $\frac{\Xi_i}{p}$ indicates whether the ith organ is well classified from different 3D images. The scores (i.e., $L_{k,i}$) of different organs in five different 3D images (see Figs. 15g, 16a, 16b, 16c, 16d) are shown in Table 2, respectively. $\frac{\Gamma_k}{m}$ is the average of each column, and $\frac{\Xi_i}{p}$ the average of each row. Table 2 shows that the visualization results of two lungs, spine (bone structure), aorta, two kidneys, spleen and heart from different 3D CT images are highly recognized by the clinical experts. Even liver, stomach and intestine were also given good evaluations. This evaluation further validated the practicability and acceptability of the proposed method.

We also display the distributions of boundary voxels of four separated boundaries (Figs. 15g, 15a, 15c, 15d) on some

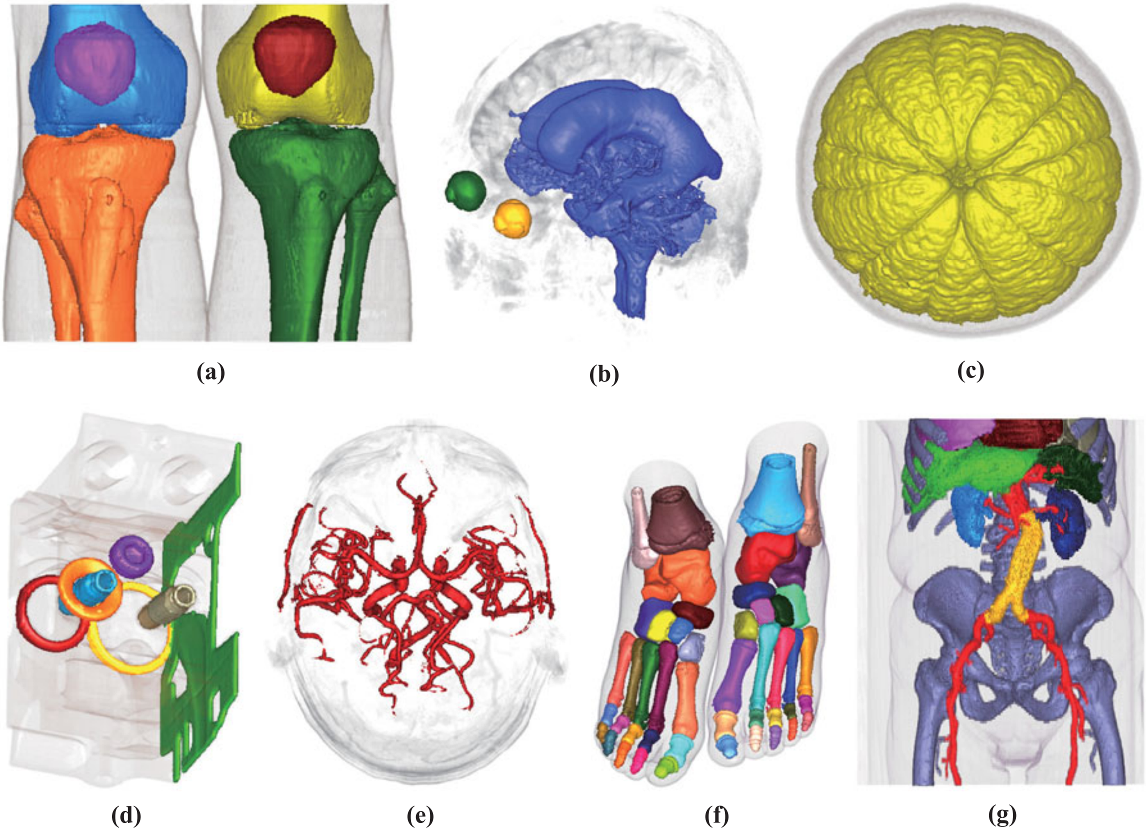


Fig. 15. Visualization of boundaries contained in the same seven 3D images as ones in Fig. 2. Different boundaries in each 3D image are separated by the proposed method and visualized with different colors. Other surface patches, fragments and skin surfaces are set very low opacity values. (a) 6 boundaries in the CT knee joints; (b) three boundaries in the MRI brain ventricle; (c) two boundaries in the MRI orange; (d) seven boundaries in the CT engine; (e) a boundary in the MRA vessel; (f) 47 boundaries in the CT foot; (g) 10 boundaries in the CT abdomen and pelvis.

2D sectional images, as in Fig. 18. Fig. 18 shows that the thick boundaries separated by the proposed method are consistent with those observed by human vision, indicating the effectiveness of the method.

In Fig. 19, the orange flesh and blood vessel structure are visualized by two different methods, respectively. In Figs. 19a and 19c, they are separated from 3D images and visualized by the proposed method. In Figs. 19b and 19d, their boundary surfaces are reconstructed by the method in [46]. The visualization results by two different methods have little difference.

6.3 Comparisons with Some Related Works

A HD-TF with 10 attributes was applied in [8] to classify the structures in the CT engine. While most structures were correctly classified, two closely adjacent rings ($\in \Omega_2$) were incorrectly classified as one structure. In this paper, these two rings are well separated, as shown in Fig. 15d. In [8], it takes about 15 hours to classify structures in the CT engine, while our method uses much less time, as shown in Table 1. The CT engine was also volume rendered in [18], [32], [38], [41], but many different structures were classified into one class.

The intelligent system in [9] can divide a CT knee joint into three pieces of bones, but the visualized bone surfaces have certain distortion, and are not as smooth as our processed ones in Fig. 15a. In [9], the structure separation was

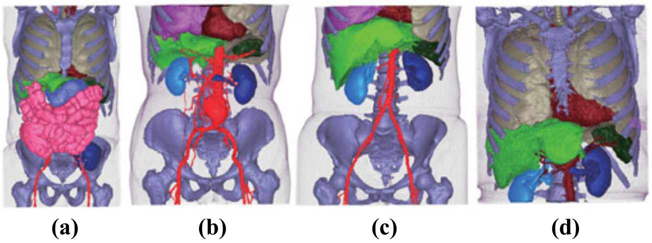


Fig. 16. Visualization of anatomical boundaries separated from four different complex 3D CT medical images by the proposed method.

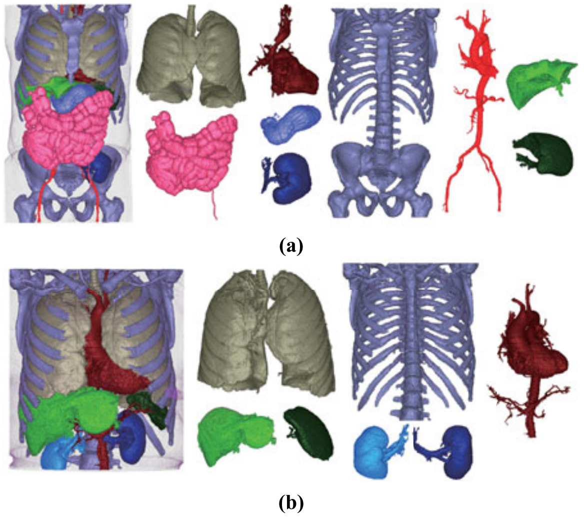


Fig. 17. Sixteen different boundaries in Figs. 16a and 16d are visualized, individually.

TABLE 1
The Statistics of the Image Size, the Number and Type of Separated Boundaries (B-number, B-type), the Evaluation of the Volume Rendering Result by Experts (Score: Very Poor-1; Poor-2; Fair-3; Good-4; Very Good-5; Perfect-6), the Time Used to Separate all Boundaries from Each 3D Image (TOH, TOC, TOS, TOI: the Times Used for Computing 2D Histogram in the SG-TF Space, for Computing Connected Sets, for Set Operations, and for Interactive Operations and others, Respectively. Time: Total Time. Unit: Second)

3D images	Image size	B-number	B-type	Score	TOH(s)	TOC(s)	TOS(s)	TOI(s)	Time(s)
in Fig 15g	512 × 512 × 174	10	Ω_1, Ω_2	4.84	2.98	5.27	13.93	183.82	206
CT Knee joint	379 × 229 × 305	6	Ω_1, Ω_2	5.8	0.78	1.04	25.46	12.72	40
MRI ventricle	256 × 256 × 124	3	Ω_1	5.5	0.25	0.76	-	50.99	52
CT Engine	256 × 256 × 256	7	$\Omega_1, \Omega_2, \Omega_3$	-	0.47	0.22	30.47	41.85	73
MRA blood	256 × 320 × 128	1	Ω_1	5.8	0.33	0.51	-	37.16	38
MRI Orange	256 × 256 × 64	2	Ω_2	-	0.09	0.20	7.72	12.98	21
CT foot	512 × 512 × 250	47	Ω_2	5.8	5.15	7.46	495.28	528.11	1036
in Fig 16a	512 × 512 × 593	9	Ω_1, Ω_2	5.07	10.94	40.42	109.17	476.47	637
in Fig 16b	512 × 512 × 288	9	Ω_1, Ω_2	5.3	6.99	30.59	13.51	375.91	427
in Fig 16c	512 × 512 × 394	8	Ω_1, Ω_2	5.32	5.76	20.65	25.19	233.40	285
in Fig 16d	512 × 512 × 347	7	Ω_1, Ω_2	5.17	4.24	14.24	60.78	222.74	302

done by painting different pieces of bones into different classes in the training data. Hence, in the cases where a 3D image contains more structures (such as the CT foot in Fig. 15f) and/or the structures with complex shapes, it is a time-consuming task to manually generate training data.

The size-based transfer function may highlight large blood vessels in MRA images [7], but small fragments and some uninteresting surface patches cannot be excluded. In this paper, they are all separated from vessels of interest (refer to Fig. 15e). In [15], the MRI cerebral ventricle was volume rendered, where small fragments and uninteresting surface patches were gradually excluded by an iterative topological smoothing technique. In this paper, such small fragments and uninteresting surface patches can be simply excluded by the spatial connectivity.

Region-growing method was mainly used to segment a structure with homogeneous attribute from 3D images [35], [36]. In many cases, it is a complex task to design an appropriate rule to stop the growing of the region. If two closely adjacent structures with similar attributes have inhomogeneous scalar values (see Fig. 7a), due to the PVE, the region-growing method usually cannot correctly distinguish such two structures.

TABLE 2
Experts' Evaluation of Boundary of each Anatomical Structure shown in Figs. 15g, 16a, 16b, 16c, 16d

3D images	15g	16a	16b	16c	16d	R-average
Left lung	5.5	5.5	5.5	5.5	5.8	5.56
Right lung	5.5	5.5	5.5	5.5	5.8	5.56
Spine	5	5.8	5.5	5.8	5.5	5.52
Heart	4.8	5.3	5.3	5.5	4.8	5.14
Left kidney	4.8	5.5	5.8	5.5	5	5.32
Right kidney	5	-	5.8	5.5	5	5.325
Aorta	5.5	5.5	5.8	5.8	5	5.52
Liver	3	4.3	4	4	4.3	3.92
Spleen	4.5	5	4.5	4.8	5.3	4.82
Stomach	-	4	-	-	-	4
Intestine	-	4.3	-	-	-	4.3
C-average	4.84	5.07	5.3	5.32	5.17	-

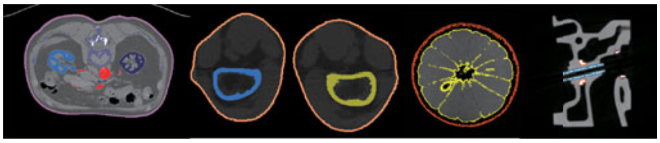
Grades and scores: very poor-1; poor-2; fair-3; good-4; very good-5; perfect-6. C(R)-average: the average of data in each column or row.

To some extent, the proposed method can be regarded as a compromise between the SG-TF and HD-TFs. While it can greatly improve the ability of the SG-TF for object classification, it can also avoid the complexity of HD-TFs. However, because of the erosion operation and interactive operation used in the proposed method, much time still might be taken if many structures are separated. Additionally, when grey between a structure and closely adjacent components (small fragments or sub-structures) are slowly and gradually changed, the boundary of the structure sometimes cannot be well visualized.

6.4 User Interface, Interactions and Usability

The user interface used in this paper is simple. It mainly consists of four parts: the conventional user interface of the SG-TF space in Fig. 20a, a pop-up context menu in Fig. 20b, two buttons in Fig. 20c and a default color table in Fig. 20d with numbers and colors of different connected sets. Fig. 20a is used for selecting and adjusting the region in the SG-TF space. Fig. 20b will pop up when right-clicking, and is used in the separation of different structures from Φ_U. Fig. 20c is for computing connected sets from Φ_U and for displaying separated structures in Φ_U. Fig. 20d is used for visually excluding several uninteresting surface patches and for merging multiple connected sets (such as different ribs) by their numbers and colors.

In volume rendering, it is useful and sometimes necessary, to incorporate users' intelligence into the visualization framework. Based on the user interface and the method in this paper, user interactions for separating structures in Φ_U may become very intuitive and simple. For example, by visualizing all different large connected sets in Φ_U with different colors, users can easily classify these connected sets



(a) (b) (c) (d)

Fig. 18. Distributions of boundary voxels of four separated boundaries (Figs. 15g, 15a, 15c, 15d) on some 2D sectional slices.

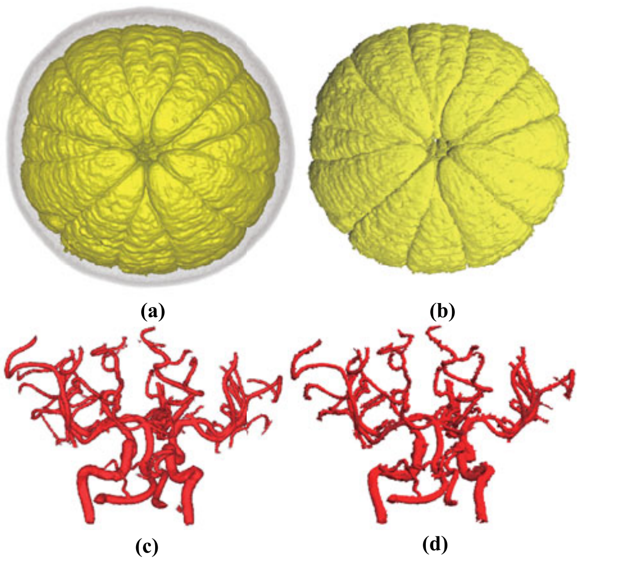


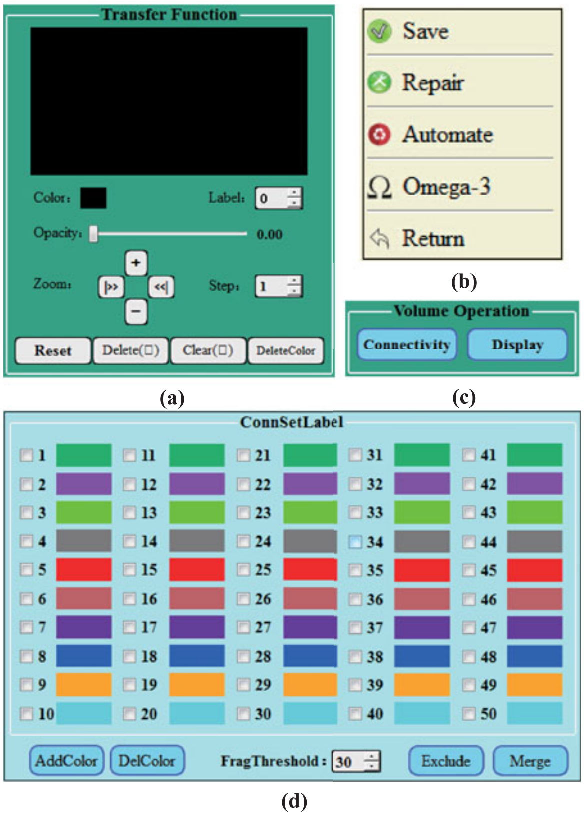
Fig. 19. Visualizations of the same structures by two different methods. (a)(c) volume rendering of the boundaries separated from 3D images by the proposed method. (b)(d) surface rendering of the boundary surfaces reconstructed by the method in [46].

into three classes: $\Omega_1, \Omega_2, \Omega_3$. By right-clicking a connected set belonging to Ω_1, and selecting the "Save" item of the pop-up context menu, the connected set can be automatically separated and saved. By first marking two boundaries S_1, S_2 in the connected set $\Phi \in \Omega_2$, and then right-clicking Φ and selecting the "Automate" item of the pop-up context menu, S_1, S_2 can be automatically separated from Φ and saved. Furthermore, by right-clicking Φ and selecting the "Return" item of the pop-up context menu, users can return to process other connected sets in Φ_U. By right-clicking a connected set $\Phi \in \Omega_3$, and selecting the "Ω_3" item of the pop-up context menu, a dialogue box containing the interface for processing Φ will pop up. In this paper, the processing results at each step can be visualized, observed, and modified when necessary. Users may select and adjust the region U in the SG-TF space according to the visualization results of Φ_U. A boundary whose outer boundary voxels are lost can be observed and repaired by right-clicking the boundary and selecting the "Repair" item of the pop-up context menu. Therefore, the method in this paper can be easily manipulated by users via the visual feedback mode.

Users can visually recognize different structures from the visualization of Φ_U based on the smoothed or regular shapes of these structures. Both the user interface and interactive mode in this paper are intuitive and simple, and easy to operate. This enables users to easily separate different structures from Φ_U, even by domain experts.

7 Conclusion

The SG-TF is a popular 2D transfer function used in volume rendering. However, the boundary-overlapping problem in the SG-TF space seriously limits its ability for object classification in many complex 3D images. In this paper we propose a hierarchical framework to solve the problem. As a result, the ability of the SG-TF for object classification can be greatly improved. A variety of experimental results demonstrate the validation and effectiveness of our proposed approach for visualizing and classifying different structures in complex 3D images, while avoiding the difficulty and complexity of HD-TFs.



□ 1	🟩	□ 11	🟩	□ 21	🟩	□ 31	🟩	□ 41	🟩
□ 2	🟪	□ 12	🟪	□ 22	🟪	□ 32	🟪	□ 42	🟪
□ 3	🟩	□ 13	🟩	□ 23	🟩	□ 33	🟩	□ 43	🟩
□ 4	⬛	□ 14	⬛	□ 24	⬛	□ 34	⬛	□ 44	⬛
□ 5	🟥	□ 15	🟥	□ 25	🟥	□ 35	🟥	□ 45	🟥
□ 6	🟥	□ 16	🟥	□ 26	🟥	□ 36	🟥	□ 46	🟥
□ 7	🟪	□ 17	🟪	□ 27	🟪	□ 37	🟪	□ 47	🟪
□ 8	🟦	□ 18	🟦	□ 28	🟦	□ 38	🟦	□ 48	🟦
□ 9	🟧	□ 19	🟧	□ 29	🟧	□ 39	🟧	□ 49	🟧
□ 10	🟦	□ 20	🟦	□ 30	🟦	□ 40	🟦	□ 50	🟦

Fig. 20. The user interface used in this paper. (a) the conventional user interface of the SG-TF space. (b) a pop-up context menu. (c) two buttons for computing connected sets from Φ_U and for displaying separated boundaries in Φ_U. (d) a default color table.

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 by NSFC of China (61375020, 61572317), 973 program of China (2013CB329401), Cross Research Fund of BME of SJTU (YG2013ZD02) and fund of SCST of China (12JC1406600).

References

[1] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L.S.Avila, K. Martin, R. Machiraju, and J. Lee,"The transfer function bake-off," *IEEE Comput. Graph. Appl.*, vol. 21, no. 3, pp. 16–22, May 2001.

[2] G. Kindlmann and J. W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," in *Proc. IEEE Symp. Vol. Vis.*, 1998, pp. 79–86.

[3] 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, Jul. 2002.

[4] S. Rottger, M. Bauer, and M. Stamminger, "Spatialized transfer functions," in *Proc. IEEE/Eurograph. Symp. Vis.*, 2005, pp. 271–278.

[5] P. Sereda, A. Vilanova, I. W. O. 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.

[6] C. Lundstrom, P. Ljung, and A. Ynnerman, "Local histograms for design of transfer functions in direct volume rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 6, pp. 1570–1579, Nov. 2006.

[7] C. Correa and K. Ma, "Size-based transfer functions: A new volume exploration technique," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1380–1387, Nov./Dec. 2008.

[8] L. Wang, X. Zhao, and A. Kaufman, "Modified dendrogram of attribute space for multidimensional transfer function design," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 1, pp. 121–131, Jan. 2012.

[9] F. Tzeng, E. B. Lum, and K. Ma, "An intelligent system approach to higher-dimensional classification of volume data," *IEEE Trans. Vis. Comput. Graph.*, vol. 11, no. 3, pp. 273–284, May 2005.

[10] Y. Wu and H. Qu, "Interactive transfer function design based on editing direct volume rendered images," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 5, pp. 1027–1040, Sep./Oct. 2007.

[11] L. Zhou, M. Schott and C. Hansen, "Transfer function combinations," *Comput. Graph.*, vol. 36, pp. 596–606, 2012.

[12] G. H. Weber, S. E. Dillard, H. Carr, V. Pascucci, and B. Hamann, "Topology-controlled volume rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 2, pp. 330–341, Mar. 2007.

[13] Y. Sato, C. F. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis, "Tissue classification based on 3D local intensity structure for volume rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 6, no. 2, pp. 160–180, Apr.-Jun. 2000.

[14] M. Levoy, "Display of surfaces from volume data," *IEEE Comput. Graph. Appl.*, vol. 8, no. 3, pp. 29–37, May 1988.

[15] S. Shafii, S. E. Dillard, M. Hlawitschka, and B. Hamann, "The topological effects of smoothing," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 1, pp. 160–172, Jan. 2012.

[16] J. J. Caban and P. Rheingans, "Texture-based transfer functions for direct volume rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1364–1371, Nov.-Dec. 2008.

[17] C. D. Correa and K. Ma, "Visibility histograms and visibility-driven transfer functions," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 2, pp. 192–204, Feb. 2011.

[18] R. Maciejewski, Y. Jang, I. Woo, H. Janicke, K. P. Gaither, and D. S. Ebert, "Abstracting attribute space for transfer function exploration and design," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 1, pp. 94–107, Jan. 2013.

[19] S. Lindholm, P. Ljung, C. Lundstrom, A. Persson, and A. Ynnerman, "Spatial conditioning of transfer functions using local material distributions," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 1301–1310, Nov./Dec. 2010.

[20] C. L. Bajaj, V. Pascucci, and D. R. Schikore, "The contour spectrum," in *Proc. IEEE Vis.*, 1997, pp. 167–173.

[21] V. Pekar, R. Wiemker, and D. Hempel, "Fast detection of meaningful isosurfaces for volume data visualization," in *Proc. IEEE Vis.*, 2001, pp. 223–230.

[22] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M. E. Groller, "Volume visualization based on statistical transfer-function space," in *Proc. Pacific Vis. Symp.*, 2010, pp. 17–24.

[23] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller, "Curvature-based transfer functions for direct volume rendering: Methods and applications," in *Proc. IEEE Vis.*, 2003, pp. 513–520.

[24] S. Wesarg, M. Kirschner, and M. F. Khan, "2D histogram based volume visualization: combining intensity and size of anatomical structures," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 5, no. 6, pp. 655–666, 2010.

[25] H. Shin, B King, M. Galanski, and H. K Matthies, "Use of 2D histograms for volume rendering of multidetector CT data: Development of a graphical user interface," *Acad. Radiol.*, vol 11, no. 5, pp. 544–550, 2004.

[26] C. D. Correa and K. Ma, "The occlusion spectrum for volume classification and visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1465–1472, Nov./Dec. 2009.

[27] D. H. Laidlaw, K. W. Fleischer, and A. H. Barr, "Partial-volume Bayesian classification of material mixtures in MR volume data using voxel histograms," *IEEE Trans. Med. Imaging*, vol. 17, no. 1, pp. 74–86, Feb. 1998.

[28] 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. Technol. Biomed.*, vol. 11, no. 6, pp. 668–682, Dec. 2007.

[29] R. Maciejewski, I. Woo, W. Chen, and D. S. Ebert, "Structuring feature space: A non-parametric method for volumetric transfer function generation," *IEEE Trans. Visual. Comput. Graph.*, vol. 15, no. 6, pp. 1473–1480, Nov. 2009.

[30] Y. Wang, W. Chen, J. Zhang, T. Dong, G. Shan, and X. Chi, "Efficient volume exploration using the gaussian mixture model ,"*IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 11, pp. 1560–1573, Nov. 2011.

[31] B. P. Nguyen, W. L Tay, C. K. Chui, and S. H. Ong, "A clustering-based system to automate transfer function design for medical image visualization," *Vis. Comput.*, vol. 28, pp. 181–191, 2012

[32] P. Sereda, A. Vilanova, and F. A. Gerritsen, "Automating transfer function design for volume rendering using hierarchical clustering of material boundaries," in *Proc. IEEE/Eurograph. Symp. Vis.*, 2006, pp. 243–250.

[33] J. S. Prabni, T. Ropinski, and K. H. Hinrichs, "Efficient boundary detection and transfer function generation in direct volume rendering," in *Proc. 14th Int. Fall Workshop Vis., Model. Vis.*, 2009, pp. 285–294.

[34] M. Teistler, R. S. Breiman, S. M. Liong, L. Y. Ho, A. Shahab, and W. L. Nowinski, "Interactive definition of transfer functions in volume rendering based on image markers," *Int. J. Comput. Assisted Radiol. Surg.*, vol. 2, no. 1, pp. 55–64, 2007

[35] R. Huang and K. Ma, "RGVis: Region growing based techniques for volume visualization," in *Proc. 11th Pacific Conf. Comput. Graph. Appli.*, 2003, pp. 355–363.

[36] F. Tzeng and K. Ma, "A cluster-space visual interface for arbitrary dimensional classification of volume data," in *Proc. Symp. Data Vis.*, 2004, pp. 17–24.

[37] F. Y. Tzeng, E. B. Lum, and K. Ma, "A novel interface for higher-dimensional classification of volume data," in *Proc. IEEE Vis.*, 2003, pp. 505–512.

[38] X. Zhao and A. Kaufman, "Multi-dimensional reduction and transfer function design using parallel coordinates," in *Proc. IEEE/EG Int. Symp. Vol. Graph.*, 2010, pp. 69–76.

[39] J. E. Nam, M. Maurer, K. Mueller, "Knowledge assisted visualization: A high-dimensional feature clustering approach to support knowledge-assisted visualization," *Comput. Graph.*, vol. 33, no. 5, pp. 607–615, 2009

[40] H. Hauser, L. Mroz, G. I. Bischi, and M. E. Groller, "Two-level volume rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 7, no. 3, pp. 242–252, 2001.

[41] Y. Kim and A. Varshney, "Saliency-guided enhancement for volume visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 925–932, Jul./Sep. 2006.

[42] S. Park and C. Bajaj, "Feature selection of 3D volume data through multi-dimensional transfer functions," *Pattern Recog. Lett.*, vol. 28, no. 3, pp. 367–374, 2007.

[43] Y. Wang, J. Zhang, D. J. Lehmann, H. Theisel, and X. Chi,"Automating transfer function design with valley cell-based clustering of 2D density plots," *Comput. Graph. Forum*, vol. 31, no. 2, pp. 1295–1304, 2012.

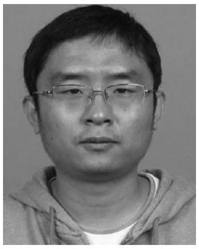
[44] C. Y. Ip, A. Varshney, and J. JaJa, "Hierarchical exploration of volumes using multilevel segmentation of the intensity-gradient histograms," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 2, pp. 2355–2363, Dec. 2012.

[45] Y. Takeshima, S. Takahashi, I. Fujishiro, and G. M. Nielson, "Introducing topological attributes for objective-based visualization of simulated datasets," in *Proc. 4th Int. Workshop Vol. Graph.*, Jun. 2005, pp. 137–236.

[46] L. Wang, P. Wang, L. Cheng, Y. Ma, S. Wu, Y. Wang, and Z. Xu, "Detection and reconstruction of an implicit boundary surface by adaptively expanding a small surface patch in a 3D image," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 11, pp. 1490–1506, Nov. 2014.

Shouren Lan received the BS degree in computer science from Jinan University in 2009 and the MS degree in computer vision from the Kunming University of Science and Technology in 2012. He is currently working toward the PhD degree in image processing and pattern recognition in Shanghai Jiao Tong University. His main interests include volume rendering and medical image analysis.

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. In 2003, he joined the Department of Automation, Shanghai Jiao Tong University, China, and now is a professor. His research interests include analysis and visualization of 3D images, computer-aided diagnosis.

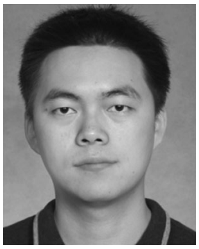


Yipeng Song received the MS degree in communication and information system from Shanghai Maritime University in 2010. Currently, he is working toward the PhD degree in image processing and pattern recognition at Shanghai Jiao Tong University. His research interests include feature recognition in the medical domain and volume rendering.

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. Now, he is an associate professor in the Department of Biomedical Engineering, Tulane University. His research interests mainly include multiscale bioimaging and Bioinformatics.

Liping Yao received BS and MS degrees in clinical medicine from the Second Medical University of Shanghai, China, in 2000 and 2003, respectively. She is currently working toward the PhD degree in medical imaging analysis in Shanghai Jiao Tong University, and is an attending physician in Xinhua Hospital, China. Her current research interest is Imaging in the diagnosis of congenital heart disease.

Kun Sun received the MS and PhD degrees in pediatrics from the Second Medical University of Shanghai, China, in 1990 and 1998, respectively. He is now the professor and chief of pediatric cardiology, Xinhua Hospital Affiliated to Shanghai Jiao Tong University, China. His research interests mainly include Ultrasonic diagnosis of congenital heart disease and Interventional treatment of congenital heart disease.



Bin Xia received the BS degree in control engineering from the Southeast University of China in 2010, and the MS degree in image processing from Shanghai Jiao Tong University in 2013. His main interests include 3D image analysis and visualization.

Zongben Xu received the MS and PhD degrees in mathematics from Xi'an Jiaotong University, China, in 1981 and 1987, respectively. He has been with Xi'an Jiaotong University since 1982, and was promoted to a professor in 1991. He is the academician of the Chinese Academy of Sciences. His current research interests include intelligent information processing and computational vision.