Skeleton Cuts—An Efficient Segmentation Method for Volume Rendering

Dehui Xiang, Jie Tian, *Fellow*, *IEEE*, Fei Yang, Qi Yang,
Xing Zhang, *Student Member*, *IEEE*, Qingde Li, and Xin Liu

Abstract—Volume rendering has long been used as a key technique for volume data visualization, which works by using a transfer function to map color and opacity to each voxel. Many volume rendering approaches proposed so far for voxels classification have been limited in a single global transfer function, which is in general unable to properly visualize interesting structures. In this paper, we propose a localized volume data visualization approach which regards volume visualization as a combination of two mutually related processes: the segmentation of interesting structures and the visualization using a locally designed transfer function for each individual structure of interest. As shown in our work, a new interactive segmentation algorithm is advanced via skeletons to properly categorize interesting structures. In addition, a localized transfer function is subsequently presented to assign visual parameters via interesting information such as intensity, thickness, and distance. As can be seen from the experimental results, the proposed techniques allow us to appropriately visualize interesting structures in highly complex volume medical data sets.

Index Terms—Volume rendering, classification, skeleton cuts, segmentation, localized transfer function.

◆

1 INTRODUCTION

Volume data rendering is one of the key tasks involved in the development of a medical data visualization system. In clinical practice, a properly rendered 3D image allows physicians or radiologists to observe and analyze meaningful and complex structures of human organs and tissues so as to make a quick and accurate diagnosis of cancers, cerebrovascular diseases, cardiovascular diseases, infectious diseases, and so on.

Two crucial issues involved in volume rendering are the classification of volume data and the management of the visual parameters for each individual voxel element. Traditionally, this is achieved by specifying a single global transfer function [1], [2]. However, the specification of such a transfer function to meet the requirements of a volume visualization task can be very difficult. On the one hand, it is not easy to effectively and accurately perform the classification of volume data and the management of visual parameters in low-dimensional transfer function domain. For instance, for the Computed Tomography Angiography (CTA) data set ($256 \times 256 \times 256$) shown in Fig. 1, it is very difficult to differ aorta from spine by just a 1D transfer function. As can be seen from the figure, many uninteresting tissues are also included because the intensity values of these materials are overlapping. The use of 2D transfer functions only improves differentiation fractionally with additional information being used to visualize the boundaries between different materials (see the hepar and stomach in Fig. 1c). On the other hand, a higher dimensional transfer function for classification is difficult to operate for a user; meanwhile, the local features such as scalar value, gradient magnitude etc., are often not intuitive enough to capture desirable structures [3], [4]. Users often have to work in the abstract transfer function domain to accomplish the two assignments simultaneously, which is usually frustrating and time-consuming. Therefore, the volume data classification and the visual parameters management should be separated in the traditional volume rendering pipeline [1].

In this paper, we present a two-step technique for volume visualization to classify volume data, explore interesting information and decrease user interaction. The first step is to classify voxels and recognize structures of interest with skeleton cuts. The new version algorithm is based on euclidean distance transformation and skeleton extraction. The second step of our proposed volume rendering technique is to visualize the resulted structures using a locally designed transfer function. There are several advantages of our proposed volume rendering framework. First and foremost, it is its effectiveness. The interactive segmentation process introduced in the technique can insure the accuracy of volume data classification in volume rendering systems, which produces more believable volume rendered images, since it is more effective to integrate both the internal information of volume data and the user interaction. Second, it is easy user interaction. Users only need to select their

---

- *D. Xiang, F. Yang, and X. Zhang are with the Institute of Automation, Chinese Academy of Sciences and Graduate University of Chinese Academy of Sciences, Beijing 100190, China. E-mail: {dehui.xiang, fei.yang, xing.zhang}@ia.ac.cn, xiangdehui07@mails.gucas.ac.cn.*
- *J. Tian is with the Medical Image Processing Group, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: tian@ieee.org, jie.tian@ia.ac.cn.*
- *Q. Yang is with the Radiology Department, Xuanwu Hospital, Capital Medical University, No. 45 Changchun Street, Beijing 100053, China. E-mail: yangyangqiqi@gmail.com.*
- *Q. Li is with the Department of Computer Science, University of Hull, Hull HU67RX, United Kingdom. E-mail: Q.Li@hull.ac.uk.*
- *X. Liu is with the Paul C. Lauterbur Biomedical Imaging Center, Institute of Biomedical and Health Engineering and Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518067, China. E-mail: xinliu86@gmail.com.*

Manuscript received 6 Dec. 2009; revised 7 Apr. 2010; accepted 31 Aug. 2010; published online 29 Oct. 2010.
Recommended for acceptance by H.-W. Shen.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2009-12-0278.
Digital Object Identifier no. 10.1109/TVCG.2010.239.

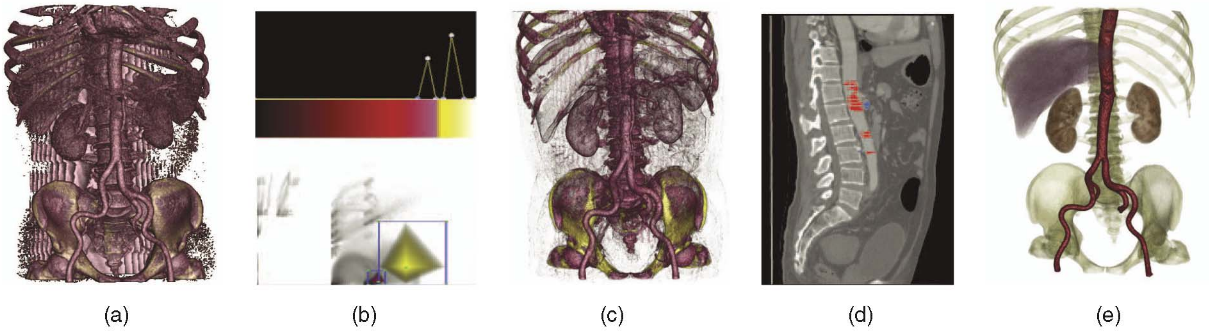


Fig. 1. Volume rendering using different transfer functions. (a) A rendered image with a 1D transfer function; (b) 1D transfer function user interface (top) and 2D transfer function user interface (bottom); (c) A rendered image with a 2D transfer function; (d) Marking interested structures on 2D slices; (e) A rendered image with localized transfer functions.

interesting structures and remove those uninteresting structures (see Fig. 2). Moreover, the design of required transfer functions can now be based on each individually recognized object in the volume data, which correspondingly makes the specification of a required transfer function much easier since the intensity variation in a segment is relatively small. In addition, it is efficiency. Our advanced graph cuts algorithm has significantly improved the performance of the conventional graph cuts algorithm in quickly processing large volume data sets. The running time and memory consumption required for the conventional segmentation algorithm has been greatly reduced.

2 RELATED WORK

Volume rendering has long been an active research area in volume data visualization and tremendous amount of dedicated studies have been conducted to develop techniques for volume data classification and visual parameters management.

Different features. Most of the previous methods have been proposed mainly via designing a single global transfer function in low-dimensional feature space. Levoy [1] introduced the gradient magnitude of volume data. Kindlmann et al. [5] extended to the second derivative to capture the information about the boundaries. Lundström et al. [6] employed local histograms to detect and identify materials with similar intensities. Šereda et al. [7] presented a new transfer function generation method by using the extended LH histograms [8], which can be used more easily to visualize boundaries. Johnson and Huang [9] introduced the local frequency distribution of intensity values in neighborhoods centered around each voxel. Many other features were also introduced such as contour spectrum [10], local intensity of 3D structures [11], curvature [12], and spatial information [13]. More recently, sizes of structures [14], occlusion spectrum [15], patterns and textures [16], and the occurrence in time of the distribution of feature sizes and densities [17] were also taken into account. These kind of methods usually lead to an inaccurate classification of voxels for a complex data set, since it is in general difficult for common users to specify a transfer function globally to classify voxels precisely when the underlying structures in a given data set are rich and complex.

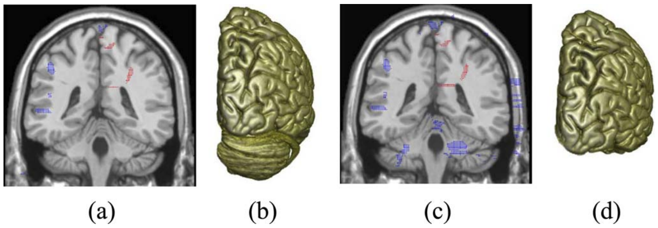


Fig. 2. A user's intention to choose interesting tissues in MR data ($181 \times 217 \times 181$). (a) Marking on 2D slices to get right cerebrum; (b) A rendered right cerebrum (yellow) image with right cerebella; (c) Removing unwanted right cerebella on the 2D slices; (d) Only the right cerebrum is extracted.

Classification or segmentation. Many higher dimensional classification or segmentation approaches have also been developed for volume rendering. Tzeng et al. [3] introduced a machine learning method for classification via considering multidimensional information, such as scalar value, scalar gradient magnitude, location, and neighboring values. The process needs to train a long time to provide high-quality classification in the entire volume. Weber et al. [18] segmented a volume via contour tree into regions to explore the topology of interesting structures [19]. Huang and Ma [20], Owada et al. [21], and Hadwiger et al. [17] integrated the region growing method into volume visualization systems. This technique is unstable and not appropriate for isolating homologous structures.

User interaction. The Design Gallery used an automatic approach to generate sparse and possible transfer functions based on automated analysis [22]. Though much less effort is required from the users in using this kind of volume rendering system, the automatic process may miss some important features. The volume cutout approach allows a user to directly interact with rendered images to achieve the task of segmentation [23], however, it can be time-consuming for visualizing a large and complex volume data set. Salama and Kohlmann presented a high-level semantic model, together with an easy-to-use interface for radiologists and physicians to design the required transfer function [4]. This method needs to create a transfer function template and is limited in visualizing pathological or abnormal tissues. Wu and Qu [24] introduced a genetic algorithm, which allows a user to directly edit volume rendered images, but it needs to be appropriately initialized by the classification widgets in 2D transfer function domain.

3 OVERVIEW OF OUR FRAMEWORK

In this paper, a new framework shown in Fig. 3 is proposed for volume rendering. The novelty of our work lies in the

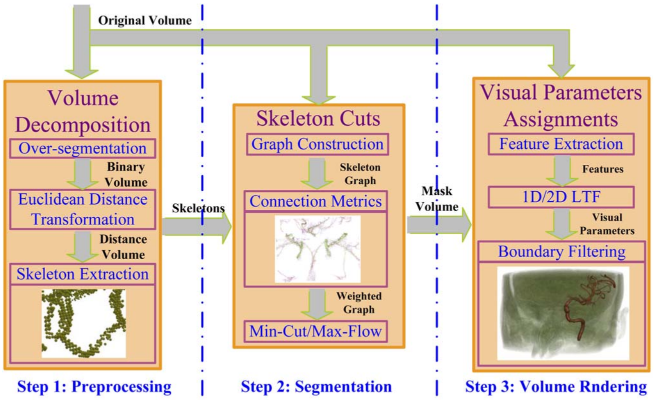


Fig. 3. An overview of our volume rendering framework.

introduction of skeleton-based graph cuts algorithm for fast and high-quality classification, and the localized transfer function designed separately for each individual segmented object. Our approach regards classification for volume visualization as a two-step process: object segmentation and local visual parameters assignments. In the preprocessing stage, a coarse boundary is first extracted from the original volume data to generate a binary volume. For each voxel in the foreground, its euclidean distance to the background is computed by euclidean distance transformation, then skeletons are extracted based on the euclidean distance field to represent the coarse object. In the segmentation stage, skeletons are linked to their neighbors to construct an unweighted skeleton graph, then the weights between two skeletons (nodes) are computed based on the intersected area of two subvolumes (called cells), the intensity means and radii of inscribed spheres. The segmentation is subsequently achieved using the min-cut/max-flow algorithm. In the volume rendering stage, volume rendering is performed based on a set of localized transfer functions for users to analyze isolated structures. A lookup table is constructed locally for each individual structure of interest to map those features, such as thickness and distance, to color and/or opacity. In addition, a new boundary filtering method implemented in CUDA is developed to compute the object boundaries at pixel resolution for properly rendering segmented structures.

4 VOLUME DECOMPOSITION

In computer graphics and computer vision, object recognition, and representation plays an important role in image understanding. For instance, image compression uses the representation of objects in digital images for storing original data in reduced space [25]. This is because when objects in a given image are properly represented, a data compression operation would be able to not only effectively simplify its internal complex structures into some simple components, but also recover efficiently the original images.

The skeleton is a reversible descriptor for precise reconstruction of a shape in image representation. Skeletons provide meaningful cues for the description of discrete objects and enriched information for visualization and analysis, such as feature extraction, pattern recognition, geometric modeling, or registration. Skeletons are also ideal shape representations with efficient memory usage while preserving topological structures. Skeletonization [26] is an effective and intuitive feature detecting technique for data abstraction due to its linearity and simplicity. In our approach, the decomposing an oversegmented volume is used as a preprocessing step to obtain a set of informative cells and describe features in the regions, which is then followed by the search for interesting skeletons to visualize an object in the volume data with visual parameter settings.

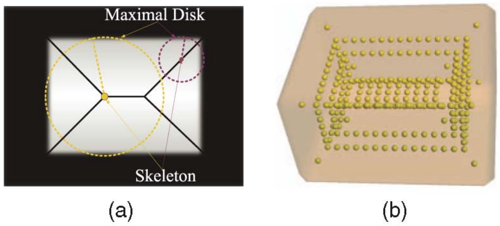


Fig. 4. Skeletons of a 2D and 3D object. (a) Curves in a rectangle are formed by numerous skeletons; (b) Skeletons (yellow balls) of a cuboid (light salmon) in a volume data set ($80 \times 48 \times 48$).

4.1 Skeletons

In the discrete space, skeletons of volumetric objects, on the one hand, are often defined by topology-preserving thinning algorithms which iteratively remove deletable object points until a thin, skeletal structure is left [27]. The elimination of deletable points is disallowed when the deletion results in a change to the object's topology. On the other hand, skeletons are also defined based on distance transformation when they are not necessarily connected with each other. In this paper, skeleton detection is carried out by distance transformation and skeletal points extraction leads to an advanced segmentation algorithm - skeleton cuts.

The skeletons of a shape are the locus of points that are minimally equidistant from at least two boundary points. In other words, it is the set of centers of maximal disks (or spheres) that inscribe against the object and touch the boundary at contact points. A skeleton of a regular 2D object (e.g., rectangle) can consist of 1D curve Fig. 4a, however, the skeleton of a regular 3D object (e.g., cuboid) may consist of some curves and surfaces Fig. 4b. Skeleton calculation of a region can be performed by either euclidean spheres, or spheres defined by other metrics like Manhattan metric. The mapping from a given region into a set of centers of maximal disks (or spheres), labeled by their corresponding radii, is called distance transformation. A maximal disk (or sphere), e.g., the brown disk and the purple disk in the rectangle, is defined to be a disk (or sphere) contained in the shape not entirely covered by another one contained in the shape. The center of a maximal disk (or sphere) is a skeletal point in the region. The skeleton and the associated radius are therefore, used to represent the original object boundary.

4.2 Euclidean Distance Transformation

The distance transformation is a powerful tool for skeletonization. We first introduce an algorithm to compute euclidean distance transformation of d-dimensional images in linear time [28]. This algorithm is time efficient and the results are exact for skeleton extraction. Taking 3D images, for example, let $\mathbf{B}$ denote the background region of an image $\mathbf{F}$ with size $n \times m \times l$. The problem is to assign to every grid point $p(i,j,k)$ the distance to the nearest point in $\mathbf{B}$. The squared distance transformation of $t(i,j,k)$ can be calculated by

$$\begin{aligned} t(i,j,k) = \min_{x,y,z}\{&(i-x)^2 + (j-y)^2 + (k-z)^2 \mid \\ &0 \leq x \leq m, 0 \leq y \leq n, 0 \leq z \leq l, p(x,y,z) \in \mathbf{B}\}. \end{aligned} \tag{1}$$

4.3 Skeleton Extraction

Skeletons are extracted from a given distance transformation image and required to restore the d-dimensional original one. For instance, a voxel $p\ (x,y,z)$ in a 3D image, can be called a skeletal point after acquiring the result $t\ (x,y,z)$ of euclidean distance transformation if there exists an element $p\ (i,j,k)$ in the image satisfying [29]:

1) $(x-i)^2 + (y-j)^2 + (z-k)^2 < t(x,y,z)$;
2) $\max_{u,v,w}\{t(u,v,w) - (u-x)^2 - (v-y)^2 - (w-z)^2, t(u,v,w) \in \mathbf{F}\} = t\ (x,y,z) - (i-x)^2 - (j-y)^2 - (k-z)^2$.

The first condition ensures that all elements $p\ (i,j,k)$ are located on the disk centered at $p\ (x,y,z)$ with radius $\sqrt{t(i,j,k)} > 0$; and the second ensures that the radius is maximal. That is, the elements of a skeleton set are the pixels corresponding to all apexes of the elliptic paraboloids which constitute the upper envelope with heights given by the euclidean distance transformation.

5 SKELETON CUTS

Our choice to graph cuts algorithm lies in its global optimization and its incorporation of users' expertise. First, global solutions to segmentation are generally preferable because of their stability. Min-cut/max-flow algorithms guarantee the minimization of energy function when computing minimum cuts on graphs [30]. Second, general purpose classification is a highly ambiguous problem. User guidance can be provided to reduce its ambiguities. Furthermore, the integration of physician's professional knowledge is conducive to perform more accurate tissue identification (see Fig. 2). In addition, the modification of the resulted tissues is advisable and required to improve the accuracy if necessary. However, the running time and the memory consumption of the conventional graph cuts limit its feasibility in many applications, such as the 3D segmentation for volume rendering.

5.1 Introduction to Graph Cuts

In this section, we give a brief review on Boykov and Kolmogorov's graph cuts algorithm in computer vision [30]. A directed and weighted graph $G = \{V, E, \Omega\}$ is constituted by a set of nodes $v \in V$ and a set of directed edges $e \in E$ with non-negative weights $\omega \in \Omega$. The nodes represent pixels or voxels in regular 2D or 3D grid data. Two special nodes are called terminals, respectively, *source* $\mathbb{S}$ and *sink* $\mathbb{T}$, which are abstracted from users' labeled nodes and added into G to connect to elements in V. Every edge e in the graph is assigned some weight ω. The weight represents the similarity between the connecting nodes. Edges are sorted into two types: *n-links* and *t-links*. *n-links* connect pixels or voxels with their several neighbors (e.g., 4-8 neighbors in 2D, and 6-26 neighbors in 3D). *t-links* connect nodes with terminals. A graph cut for an image $\mathbf{F}$ defined as a subset of edges in E is a separation of the nodes V into two disjoint sets $\mathbf{O}$ ($\mathbb{S} \in \mathbf{O}$) and $\mathbf{B}$ ($\mathbb{T} \in \mathbf{B}$). Recognizing and partitioning an object with boundaries from an image is equivalent to find an appropriate cut in the graph G. In the literature, this problem is regarded as an optimization of energy function [31] to find the minimum cost $\hat{C}$ among all feasible cuts $\mathcal{C}$:

$$\hat{C} = \arg\min_{C \in \mathcal{C}} \sum_{e\ (p,q) \in C} \omega(p,q), \tag{2}$$

where $\omega(p,q)$ denotes the weight assigned to the edge $e(p,q)$.

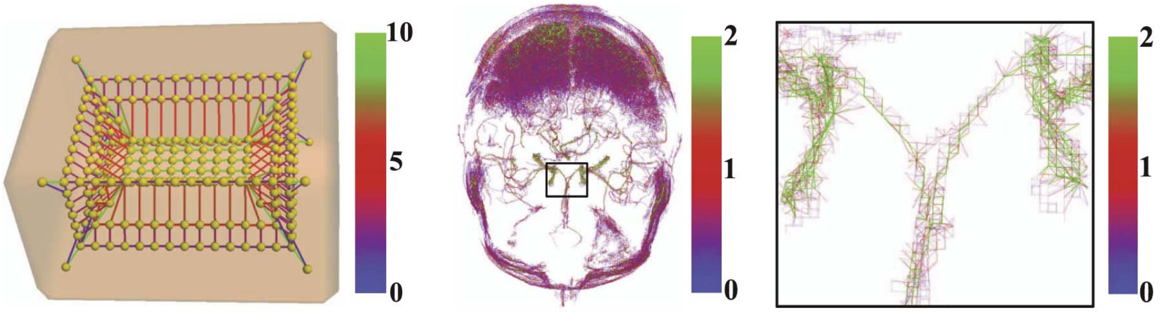
5.2 Graph Construction

In this section, we provide algorithmic details about our skeleton network construction. We decompose an over-segmented volume into a set of cells represented by skeletons which are detected by the algorithms described in Section 4. Indeed, given a set of voxels in the foreground, the euclidean distance transformation value at a voxel $p(i,j,k)$ corresponds to the radius of the largest sphere centered at $p(i,j,k)$ contained in the foreground. A maximal sphere is a sphere contained in the foreground not entirely covered by another sphere contained in the foreground. Skeletons are the set of central voxels of maximal spheres covering the foreground. Once skeletons are extracted from the euclidean distance field, each element $p(i,j,k)$ in the foreground is scanned to find its skeleton $p(x,y,z)$, so that these elements constitute a set called a skeleton cell $SC(x,y,z)$. The new version of graph incorporates skeletons as nodes $p(x,y,z)$ (see the yellow balls in Fig. 5a) and the intersection between two skeleton cells $SC(x_1,y_1,z_1)$ and $SC(x_2,y_2,z_2)$ as edges (see the links in Fig. 5a). The number of neighbors of a skeletal point depends on those skeleton cells which intersect it in this irregular system. Two special nodes *source* $\mathbb{S}$ and *sink* $\mathbb{T}$ are also designated as the labels of foreground and background.

5.3 Connection Metrics

In this section, we concentrate on analyzing the internal features in the regions where skeletons have been extracted. The conventional graph cuts algorithm optimize energy function (6) in the pixel or voxel level, and assign weights to *n-links* and *t-links* simply using their scalar values [32], local intensity gradient, Laplacian zero crossing etc. In addition, the traditional graph cuts algorithm for categorization in volume data is limited both by its intense memory requirements and nonlinear time complexity. In order to keep the representation as simple as possible, the area that neighbored skeleton cells touch each other, radii of maximal inscribed spheres and intensity means are taken into account in our framework.

In our system, the edges of pairs of nodes are bidirectional except the edges with terminals. Each pair is connected by two directed edges $e(p,q)$ and $e(q,p)$ with homologic weight ω. If two connected nodes p and q are



(a) (b) (c)

Fig. 5. Some examples of skeleton graph. (a) The skeletons (yellow balls) of a cuboid (light salmon) and their connections (colored sticks) of the same data in Fig. 4b described by a ball-and-stick model, where the connection values range from 0 to 10; (b) The connection between skeletons of a coarse result ($256 \times 320 \times 128$) segmented by threshold value 87.9 described by the colored lines, where the connection values range from 0 to 2; (c) Zoomed in connection between skeletons of (b).

disjoined by an optimal cut so that p is categorized as a part of object while q is categorized as a part of background, then one single weight ω is added up to the energy function in the (6) while the other is neglected, and vice versa. As a matter of fact, one of the most important issues, which should be tackled to achieve an excellent cut, is to assign proper weights to edges. Hence, we exploit the features contained in the skeletons and propose a novel metrics M to compute the connectivity of neighboring nodes p and q:

$$M(p,q) = A(p,q) \cdot S(p,q), \tag{3}$$

where nodes p and q represent two intersected cells in the foreground; $A\,(p,q)$ is the intersected area of p and q; and $S\,(p,q)$ is the similarity between p and q. Given n measurements of features $f_1, f_2 \cdots f_n$, then we can compute the similarity of two nodes:

$$S(p,q) = \exp\left(-\sum_{j=1}^{n} k_j \frac{(f_{jp} - f_{jq})^2}{E[(f_{jp} - f_{jq})^2]}\right), \tag{4}$$

where k_j is the coefficient of the jth feature; f_{jp}, f_{jq} are respectively, the measurements of the jth feature of nodes p and q. $E\,[\cdot]$ is the expectation of the measurement of the jth feature. In this paper, we use two features to compute $S(p,q)$: the intensity mean i of voxels in a node (a skeleton cell) and the radius r of its maximal inscribed sphere. That's why, we have

$$S\,(p,q) = \exp\left(-k_i \cdot \frac{(i_p - i_q)^2}{E[(i_p - i_q)^2]} - k_r \cdot \frac{(\log r_p - \log r_q)^2}{E[(\log r_p - \log r_q)^2]}\right). \tag{5}$$

Given that i and r are independent random variables, we can get $E\,[(\log r_p - \log r_q)^2] = 2D\,[\log r]$ and $E\,[(i_p - i_q)^2] = (\frac{1}{N_p} + \frac{1}{N_q})D\,[i]$, where $D\,[\cdot]$ is variance, N_p and N_q are, respectively, the numbers of voxels in nodes p and q.

Here, *n-links* are defined in the irregular neighboring system which are varied greatly from the traditional graph cuts algorithm. They are expressed as the connection of adjacent nodes with more features. Geometrically, the intersected area denotes the spatial relationship of the intersected pairs. Intensity mean represents intensity information since similar features should occur approximately at the adjacent regions; meanwhile, the geometric modality describes the similarity of two neighboring regions in structural conformation. The property of intersected area, intensity mean, and radius can be explained in Fig. 5. Fig. 5a shows the connection of adjacent skeletons of a regular 3D object (cuboid). Fig. 5b shows the connection between skeletons of a coarse result segmented by threshold value. Fig. 5c reveals the zoomed in connection. It shows that the connection value of an object decreases from the center to boundary.

6 VOLUME RENDERING

6.1 Localized Transfer Function

In this section, we detail an interactive transfer function design method. To decrease user interaction [4], only a few visual parameters should be managed to perform the operation of transfer functions. For a given volume of interest (VOI) isolated by skeleton cuts, it can be displayed realistically by a localized transfer function and special features can be coded via visual parameters. A localized transfer function can be defined as a mapping $\mathcal{F} : f_1, f_2, \ldots, f_n \rightarrow \overrightarrow{V}$, where, $f_1, f_2, \ldots, f_n$ are n types of features like *intensity*, *radius* of maximal spheres, *gradient magnitude*, and *curvature* etc. $\overrightarrow{V}$ denotes visual parameters such as color $\vec{c}$ or opacity o.

6.1.1 1D Localized Transfer Function

A basic 1D localized transfer function can be defined with only one feature *intensity* in the following form:

$$\mathcal{F}\,(i,j) = P\{I < i | voxel \in SV_j\}, \quad j = 1, 2, \ldots, N, \tag{6}$$

which is the intensity distribution function $P\,\{\cdot\}$ of the jth subvolume SV_j with the random variable I and intensity value i. N is the total number of segmented structures. We can get the final 1D localized color (opacity) transfer function $\overrightarrow{c}_{1d}(i,j)(o_{1d}(i,j))$ with only two control points $\overrightarrow{c_1}(j)$ and $\overrightarrow{c_2}(j)$ (similarly $o_1(j)$ and $o_2(j)$) on the transfer function editor (see Fig. 10).

$$\overrightarrow{c}_{1d}(i,j) = \mathcal{F}(i,j)\overrightarrow{c_1}(j) + (1 - \mathcal{F}(i,j))\overrightarrow{c_2}(j). \tag{7}$$

6.1.2 2D Localized Transfer Function

- VOI Highlighting: We define a local neighborhood $B\,(p,r)$ centered at the point p with the radius r in an object. For every point inside the region $B(p,r)$, we define some filtering operations, called low passing, band passing and high passing, to map color or opacity to another feature $radius$ (r). The feature seems to be similar to the size proposed by Correa and Ma [14], but the radius r proposed here is an euclidean scale to highlight some regions of an object. For an accurately recognized object, euclidean distance transformation and skeleton extraction described in Section 4 are employed again to calculate the radii of the inscribed spheres and skeletal points. Low-pass, band-pass, and high-pass widgets are created for operating by users (see Fig. 12). With a given color $\overrightarrow{c_0}(j)$ for highlighting the jth subvolume SV_j, we then can get the final 2D localized color transfer function:

$$\overrightarrow{c}_{2d}(i,j,r) = k_{rad}\overrightarrow{c_0}(j) + (1-k_{rad})\overrightarrow{c_{1d}}(i,j), \qquad (8)$$

where k_{rad} is the highlight weight given by the curve widget corresponding to the radii r.

- Non-VOI Diminishment: For a simple point pos in the complement of interesting structures, its euclidean distance d to VOI is computed by the euclidean distance transformation. The 2D localized opacity transfer function $o_{2d}(i,j,d)$ can be implemented by multiplying the 1D localized opacity transfer function $o_{1d}(i,j)$ with the following function:

$$k_d = b_0 - b_1 \cdot d, \qquad (9)$$

where b_0 and b_1 are set by the user with a curve widget (see Fig. 13).

6.2 Boundary Filtering

The results of segmentation are stored in an integer volume called a mask volume. Each integer value is an object ID corresponding to a segmented object. It has been well known that for rendering a segmented volume, a basic task is to calculate the object boundaries at pixel resolution. One solution was given by Hadwiger et al. [33] which considered a mask volume as multiple binary volumes, and a sample belongs to an object if the value interpolated from the corresponding binary volume is bigger than 0.5. The implementation in that paper utilized vector operations and was very efficient on vector GPUs. However, some most recent GPUs use scalar architecture for better programmability. On such platforms, vector operations have little advantages. Moreover, the time required for the boundary filtering is linearly related to the number of objects, hence an upper bound of object number is usually predefined and not scalable during execution. To overcome these limits, we develop our own boundary filtering scheme.

The purpose of the boundary filtering operation is to decide which localized transfer functions are to be used and how the transferred values are to be blended for a scalar value sampled along the viewing ray. This is because a viewing ray may intersect with different objects. As a result, different localized transfer functions may be involved in the calculation of the color corresponding to the viewing ray.

One solution to the boundary filtering problem is through trilinear interpolation, which can be achieved by calculating the weights for each vertex according to the sampling location, when a voxel is considered as a cuboid. To take the advantage of hardware interpolation functionality, we build a $2 \times 2 \times 2$ texture with four channels as shown in Fig. 6. For each sample, we first fetch the texture using the fraction part of the sample's voxel coordinate, then with the z-coordinate flipped, we fetch it again. So, we get eight values in total, which are exactly the weights for the vertices. Note that due to platform specification, it may be required to add a 0.5 shift to the texture coordinates to make it fall into the linear part of the texture.

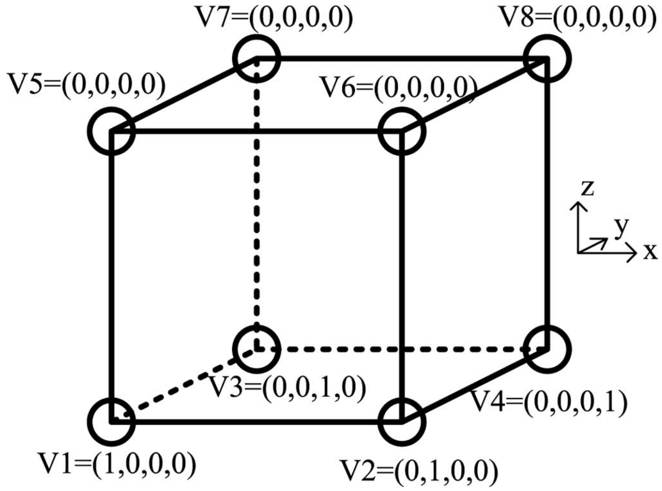


Fig. 6. A texture used for weight calculation in boundary interpolation.

Algorithm 1. Calculate Object Weights

```
Require: $id[0] \cdots id[7], w_0 \cdots w_7$
  BYTE: $binMask \Leftarrow 11111111b$
  while $binMask > 0$ do
    $curID \Leftarrow id[\mathbf{ffs(binMask) - 1}]$
    $weight \Leftarrow 0$
    if $id[0] = curID$ then
      $weight \Leftarrow weight + w_0$
      $binMask \Leftarrow binMask \& 11111110b$
    end if
    if $id[1] = curID$ then
      $weight \Leftarrow weight + w_1$
      $binMask \Leftarrow binMask \& 11111101b$
    end if
    ⋮
    if $id[7] = curID$ then
      $weight \Leftarrow weight + w_7$
      $binMask \Leftarrow binMask \& 01111111b$
    end if
    Process $(curID, weight)$
  end while
```

Another crucial problem in the boundary filtering is to calculate the sum of vertex weights for each related objects. The default approach is to use an array of object weights to store the weights for all possible object IDs. However, performance will drop when a large number of objects are defined, since the more objects are there, the bigger is the array required and the more time is consumed to traverse the array. We notice that the eight vertices can refer to no more than eight objects, so it is possible to design an algorithm whose complexity is not related to the object number. The idea of our algorithm is to enumerate the object IDs one by one. The first object ID can be achieved from any of the vertices, then all the vertices with the same object ID are checked. If necessary, the second object ID is achieved from the first unchecked vertex, and all the vertices with the same object ID are checked, and so on. To efficiently find "the first uncheck vertex," we encode the checked status of the vertices into an 8 bit integer, and a map from the 8 bit integer to an "unchecked" bit index is needed. In some most recent graphics hardware, an integer operation called "find first set bit" is provided, which returns the index of the first bit set beginning with the least significant bit, and bits are numbered starting at one. In the CUDA C environment [34], such operation is implemented as an intrinsic function called "_ffs." Since, what we need is an unchecked vertex index, we should encode the checked status inversely to the bit mask. When "_ffs" is not present, an array with 256 elements can be built, which can serve as a lookup table. Note that the lookup table can be shared between multiple threads, so it can still be a very cheap technique if there are many threads executed concurrently on GPU. In the worst case, the eight vertices belong to eight different objects, and then, eight rounds are needed. However, such a case is unlikely to happen in terms of probability. In most cases, all vertices belong to only one object, which is the case when a ray is traveling inside an object. The pseudocode of the algorithm is given by Algorithm 6.2, assuming a "ffs" function is present. $id\,[0]\cdots id\,[7]$ denote the object IDs of the vertices, $w_0 \cdots w_7$ denote the weights of the vertices. Object IDs and corresponding weights are successively generated, which can be processed.

Here, we actually have multiple options for the operations on an object ID sorted out with corresponding weight value. One option, as with [33], is to compare the weight with 0.5. If it is bigger than 0.5, the loop is terminated, and the localized transfer function corresponding to the object ID can be exclusively used. Alternatively, we can use accumulators to calculate the weighted sum of the values (opacities and opacity weighted colors) transferred using multiple localized transfer functions corresponding to the IDs sorted out.

In this algorithm, the number of objects does not need to be known in advance. Consequently, the limit of object number is removed entirely and no additional time is required when more objects are defined.

7 EXPERIMENTS AND EVALUATION

Volume rendering can assist physicians in diseases diagnosis and pathologies analysis. However, due to the diversity in imaging modality and the amount of data generated by modern 3D imaging devices, it can take hours to perform classification or segmentation manually to insure the accuracy of results. Automatic and semiautomatic techniques aim at creating a user friendly environment to ease and speed up the process. However, the development of an automatic or semiautomatic segmentation technique to achieve a satisfactory result has proved to be a tough task. This is because, first, the organs and tissues usually exhibit high variability in size and shape. In addition, their appearance and geometry can be distorted and deformed by various diseases such as aneurysms, stenoses, and traumas. All these factors make it difficult to analyze interesting structures in a medical volume data set. Accordingly, a great deal of experiments have been carried out on medical data sets to evaluate the performance of our approach. All the algorithms involved in the technique are implemented in C++ as part of our medical imaging processing tool kit (MITK) and application system 3DMed [35]. All the experiments were done using a PC with an Intel Core 2 Quad8200 2.33 GHz CPU with 2 GB of RAM and an NVidia GeForce GTX260 GPU with 896 MB memory, and a workstation with an Intel Xeon 3.2 GHz CPU with 16 G RAM. Volume rendering is achieved by ray casting and implemented in CUDA . All data sets were collected by our partners within clinical practice at the Radiology Department of Xuanwu Hospital, Capital Medical University, and Segmentation Challenge MICCAI 2009.

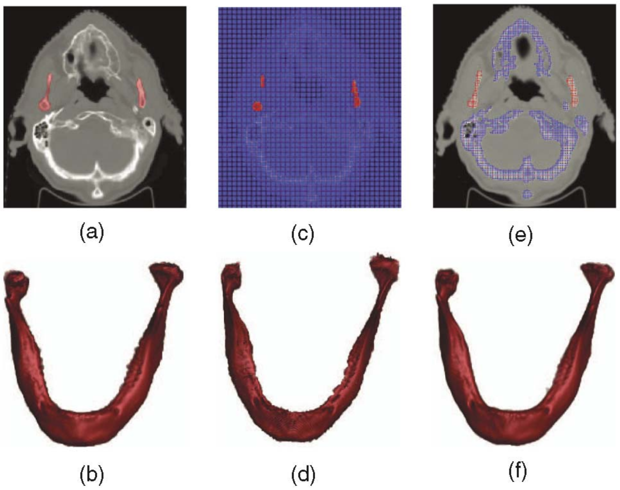


Fig. 7. Comparisons of mandible segmentation accuracy. (a), (b) A manual method; (c), (d) Conventional graph cuts; (e), (f) Skeleton cuts.

7.1 Skeleton Cuts

7.1.1 Accuracy and Complexity

This section provides a quantitative analysis of the accuracy and complexity of Boykov and Funka-Lea's conventional graph cuts (CGC) [32] and skeleton cuts (SC). The data sets used in our experiment are the head and neck data obtained from Segmentation Challenge MICCAI 2009 (http://grand-challenge2009.bigr.nl/). The goal of the experiment is to segment mandibles out of the data, which is an extremely challenge segmentation problem. Dice similarity coefficient [36] was used to test the results with CGC and SC (see Figs. 7b and 7c) against the expert manual segmentation (see Fig. 7a). This criterion is valued from 0 and 1, and is a measure of the total volumetric overlap $\kappa(tv)$ computed as

$$\kappa = 2 \times \frac{|X \cap Y|}{|X| + |Y|},$$

where $|\cdot|$ is the number of voxels. This criterion is also evaluated in axial slices where the manual delineations are present, and the average $\kappa(as)$, and median $\kappa(ms)$ are used to produce the slice overlap scores. The produced statistics are given in Table 1. The number of skeletons generated in the process is shown in the ninth row. The ratio was computed between the number of skeletons and the number of voxels. These results were obtained via only

TABLE 1
The Accuracy and Complexity of CGC versus SC

Datasets		1	2	3	4	5
Size		180×220×100	200×240×80	200×230×80	170×210×80	190×240×80
CGC	$\kappa(as)$	0.8979	0.8813	0.886	0.8649	0.8703
	$\kappa(ms)$	1	1	1	1	0.8998
	$\kappa(tv)$	0.8482	0.844	0.8484	0.8322	0.8057
SC	$\kappa(as)$	0.9505	0.9081	0.9163	0.8951	0.9444
	$\kappa(ms)$	1	1	1	1	0.9399
	$\kappa(tv)$	0.9123	0.871	0.841	0.8815	0.9101
No.		74188	60522	57414	54447	51057
Ratio		1:53.4	1:63.4	1:64.1	1:52.5	1:71.4

placing the same seeds at the joints between mandible and skull. These experimental results have demonstrated that skeleton cuts can improve the segmentation accuracy and achieve a satisfactory segmentation; meanwhile, it can reduce the segmentation complexity and enable the CGC method to work on less nodes. In addition, we have also studied how the parameters of connect metrics affect segmentation results. Figs. 5b and 5c describe the connection between skeletons of an oversegmentation result. Its value ranges from 0 to 2. The same seeds were then painted on only one slice in this experiment. Intersected area was used to segment the arteries as a standard ($k_i = 0$, $k_r = 0$). Figs. 8 and 9 show the changes of Ratio, $\kappa(as)$, $\kappa(ms)$, and $\kappa(tv)$ when additional features, namely, the intensity mean and radius, were taken into account. With these parameters, less users' inputs are needed to remove unwanted structures. Further, it is difficult to achieve a perfect segmentation of a given medical image in most practical circumstances. Therefore, with the implementation of an improved version of the CGC in our interactive volume rendering framework, it is very convenient for end users to modify flexibly the result of segmentation when necessary, and the result of segmentation can be further refined by placing additional seeds.

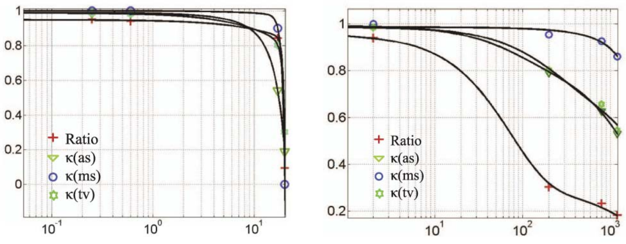
7.1.2 Performance in Speed and Memory Usage

The performance in speed and memory usage of our advanced segmentation algorithm for various volume data sets has also been tested. Fig. 10 shows segmentation of two data sets: heart and hepar. The results in the left column were segmented with CGC, and those in the right column were segmented by the new version. As shown in the figures, the segmented organs obtained by the two algorithms appear to be similar. However, the experiments show that it is relatively easier to extract a complete coronary artery and eliminate heart from hepar via SC than CGC (see the blue arrows in Figs. 10a and 10b). Additionally, we tested the algorithm with another three more data sets: pulmo, kidneys, and artery. The performance of our newly proposed algorithm exceeds the CGC method both in time and memory usage (shown in Table 2, S1: Step 1 and S2: Step 2 in Fig. 3). The number of skeletons and the ratio between skeletons and voxels were also given in this table. We have also carried out some experiments to test the performance in different volume sizes. We have further applied our approach to five heart data sets by downsampling the CTA heart data ($256 \times 256 \times 128$) at the factor of 2, 4, 8, 16, and 32. Fig. 11a shows the running time of CGC and SC. The performance of our proposed algorithm is much better than the conventional version. Moreover, as shown in Fig. 11b, the expense of memory has also been significantly reduced with our advanced graph cuts algorithm.

As studied by Boykov and Kolmogorov [30], the time complexity of graph cuts is $O(|V||E|^2|\hat{C}|)$, hence those methods are advisable to improve the performance of standard graph cuts if they can reduce the number of nodes $|V|$, the number of edges $|E|$, and the cost of the minimum cut $|\hat{C}|$. With our skeleton cuts, the link values along the contour produced by the threshold are set to zero using oversegmentation approaches like a threshold. Also, most of the


(a) (b) (c)

Fig. 8. Examples of artery segmentation with different parameters k_i and k_r. (a) $k_i = 0$, $k_r = 0$; (b) $k_i = 17$, $k_r = 0$; (c) $k_i = 0$, $k_r = 800$.



(a) (b)

Fig. 9. The plot of Ratio, $\kappa(as)$, $\kappa(ms)$, and $\kappa(tv)$ against the change of k_i and k_r. (a) The variation of k_i when $k_r = 0$; (b) The variation of k_r when $k_i = 0$.

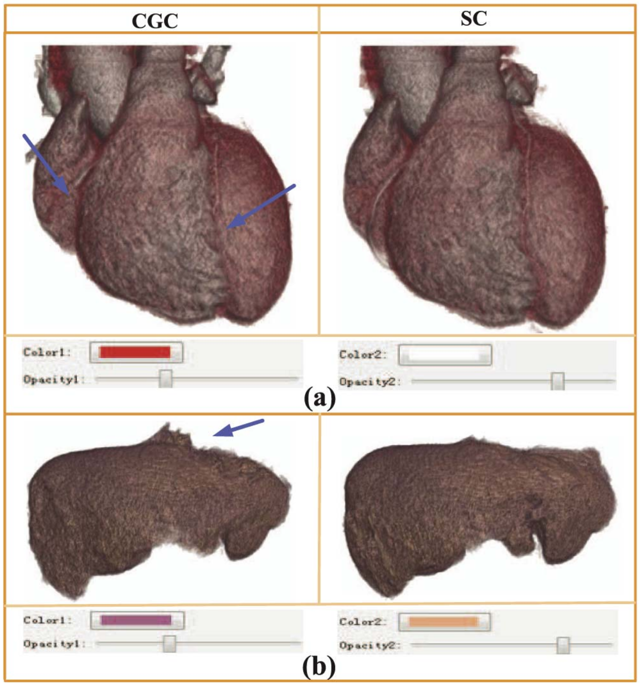


Fig. 10. Comparison between CGC and SC in segmenting organs: (a) heart, (b) hepar. The threshold values were 1,015 for heart and 1,020 for hepar. The segmented results were rendered with the same 1D localized transfer functions.

TABLE 2
Comparison of Time and Memory Usage between CGC and SC

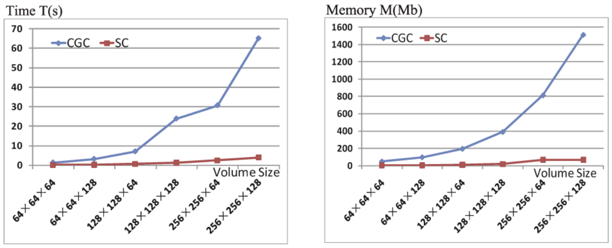
		CGC		SC			
	Size	T(s)	M(Mb)	T(s)(S1+S2)	M(Mb)(S1+S2)	No.	Ratio
Heart	$256 \times 256 \times 128$	65.1	1512.6	3.9(3.7+0.2)	64.4(32.4+32.0)	55103	1:152.2
Hepar	$256 \times 256 \times 128$	168.2	1560.3	5.4(4.4+1.0)	69.9(32.6+36.3)	281269	1:29.8
Pulmo	$256 \times 256 \times 128$	113.2	1614.3	4.8(4.1+0.8)	62.5(33.2+29.3)	361299	1:23.2
Kidneys	$512 \times 512 \times 314$	\	\	33.3(30.8+2.5)	586.0(368.2+217.8)	1456066	1:56.5
Artery	$512 \times 512 \times 957$	\	\	186.7(184.2+2.5)	2082.0(1978.5+103.5)	822137	1:305.1

segmentation result will be co-ordinated to the contour, so that the min-cut cost can be made very small. Additionally, the decomposition of the volume using skeleton extraction can dramatically reduce the number of nodes and edges.

7.2 Volume Rendering

7.2.1 VOI Highlighting

In many clinical scenarios, VOI highlighting is an important illustration technique to analyze the shape and geometry of interesting tissues. Users only need to visually choose color and move a curve widget to design a required transfer function (see the bottom of images in Fig. 12). With the construction of the euclidean distance and the skeletons for gray level structure, we were able to find the radii of inscribed maximal spheres and their corresponding locations in a rendered image. In practice, this can be done easily by users to just adapt the parameter r using the curve widget. Then, the structures will be automatically be highlighted by a specific color. The proposed technique has also been applied to qualitatively display blood vessel anomalies like stenoses and aneurysms [37]. In this case, a low-pass technique for stenosis highlight in CTA data $(512 \times 512 \times 330)$ was used (see Fig. 12a). The curve widget was applied at upper band scales in order to detect stenose. This is done by adjusting the parameter r in the low-pass curve widget from *min* to *max*. Another application of our technique was the visualization of CT pelvis data $(512 \times 512 \times 276)$ (see Fig. 12b), which illustrated the proposed method can obtain the geometrical information. A third high-pass technique was also applied to visualize intracranial aneurysms. The experiment was carried out with a DSA data set $(512 \times 512 \times 256)$ shown in Fig. 12c. The high-pass curve widget allows users to know the shape and rough size of the aneurysms, including their *min* radius and *max* radius of inscribed maximal spheres, as well as the corresponding locations.

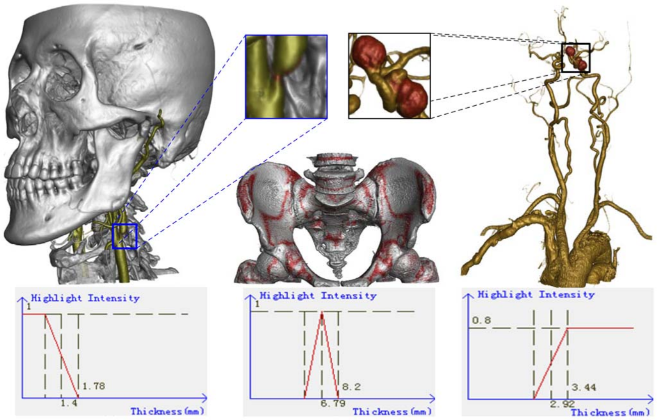


(a) (b)

Fig. 11. Comparison of time and memory usage between the conventional graph cuts and our proposed version. (a)The processing time was tested in different volume sizes; (b)The memory usage was also tested in different volume sizes. These experiments demonstrate that our approach significantly reduces the processing time and memory consumption.

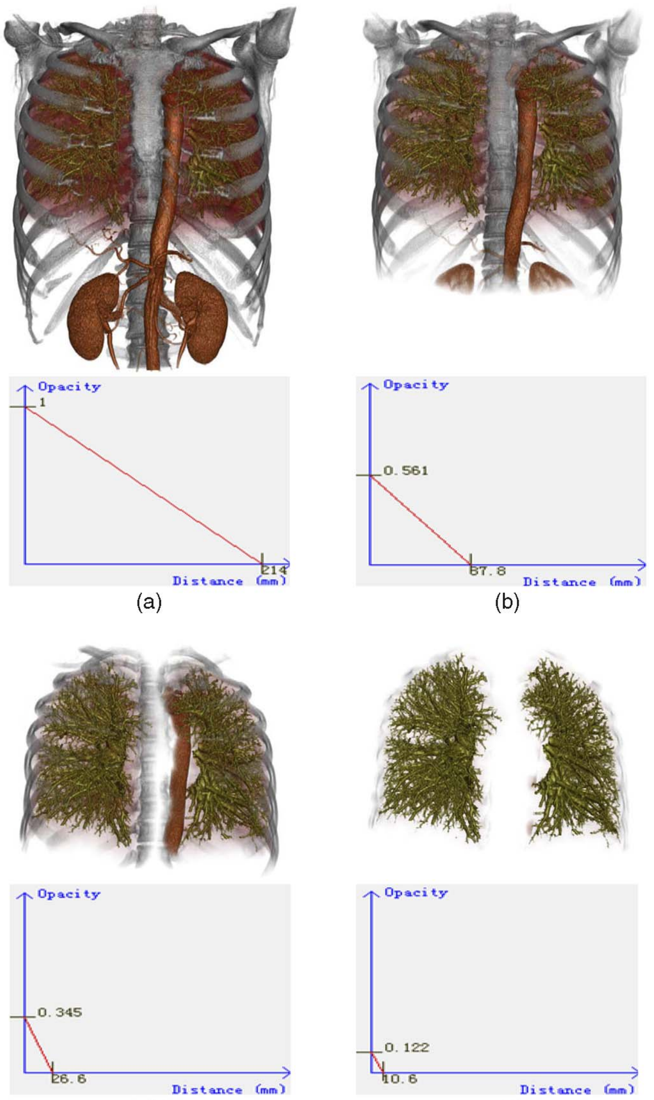
7.2.2 Non-VOI Diminishment

Our framework can be used not only as VOI highlighting for intuitive observation, but also as an interactive method for object positioning. In some practical situations such as the diagnosis of pulmonary cancer, the identification of spatial relationship between the lesion and pulmonary vessels is crucial for the differential diagnosis of pulmonary malignant and benign tumor, as well as surgery planning. However, some conventional approaches are mainly designed to let users subjectively select a view to perform the measurements in 2D or 3D space. By just using the original CT images or following the traditional way of quantification, radiologists find it difficult to determine in a selected view if the pulmonary vessels and pleura are pulled to the lesion, an important sign which strongly suggests the possibility of pulmonary cancer. With non-VOI diminishment in volume rendering, radiologists can intuitively and quickly assess the spatial relationship between the lesion and its circumjacent structures. As a kind of computer animation, this process of interaction can be easily understood in clinical practice by common users. Since segmentation is embedded in our volume rendering framework, a lookup table can be constructed for expressing the euclidean distance between two tissues in volume visualization. At first, euclidean distance transformation is performed to compute the euclidean distance between the focused tissue and every voxel in the context. Then, the shortest distance between tissue of interest and other organs can be easily acquired in the localized transfer function domain. As a scale tool, our system provides the specialized transfer function to end



(a) (b) (c)

Fig. 12. Some examples of VOI highlighting. (a) Carotid artery stenosis is highlighted by the low-pass curve widget; (b) Pelvis is highlighted with the band-pass curve widget. (c) Intracranial aneurysms are highlighted with the high-pass curve widget.



(a) (b)

(c) (d)

Fig. 13. VOI positioning by contextual tissues diminishments using the proposed transfer function. The upper rendered images demonstrate that the pulmonary vessels can be positioned by eliminating the surrounded organs. (a) The original rendered image; (b),(d) Volume rendering effects by scaling the euclidean distance.

users who can use it to position the objects of interest in a volume. Another distinctive feature of our approach is its simplification and intuitiveness to use.

Fig. 13 shows an application to position pulmonary vessels in a chest CTA data set $(512 \times 512 \times 256)$. A user can select pulmonary vessels at first, and then, their euclidean distances to kidneys, aorta, and ribs are easily revealed in the proposed transfer function domain. In this application, the opacity is used to diminish contextual tissues and estimate their distance to VOI. We can also use color as an alternative means to code the distance.

8 DISCUSSION AND FUTURE WORK

In this paper, we presented a novel framework based on advanced graph cuts segmentation algorithm and localized transfer function. The proposed approach allows us to fast classify volume data with high quality, extract important information of those interesting structures, and decrease user interaction. However, there are some limitations of our approaches, which are mainly associated with the limitations of the relevant segmentation techniques used in our framework for generating a coarse contour of interesting structures.

As have been shown by a large number of experiments, our approach works well for most real clinical data, but it may not be able to produce desirable visualizations when the volume data sets are extremely noisy. This is because in this situation, it is difficult to obtain required segmentations of the volume data sets. However, we are not concerned much with this extreme situation as the inputs of our medical visualization system are mainly real medical data acquired with different kinds of scanners. In fact, when the noise exceeds certain level, some noise filtering techniques can be applied to the data set to filter out certain noise. As a way to obtain a desirable result and avoid such problem, we can use the anisotropic diffusion filtering method proposed by Perona and Malik [38]. The algorithm can make the anisotropic diffusion preserve edges while reduce noise in the internal region. Then, we can obtain a well-segmented mask volume and fuse it into the original volume. In this way, we can visualize the original volume realistically via a localized transfer function. Another limitation of our approach is its applicability to scientific data. Our approach works well when structures of interest have comparatively clear boundaries, which is the case in general for most volume medical data of different types. In some cases, such as in climatology simulation, the values in the volume data may vary smoothly across the entire domain, i.e., boundaries between regions are fuzzy. It is difficult to determine a coarse contour of interesting regions via thresholding. This can be solved by a fuzzy clustering approach [39], which categorizes voxels into a number of subgroups based on a certain degree of closeness or similarity in the fuzzy environment.

As shown in our work, our approach can be easily integrated with previous features like gradient magnitude and curvature, to obtain different information. In addition, in the future, it is possible to search for more feature information to describe pathologies, such as blood vessels anomalies, in order to assist physicians and radiologists to diagnose diseases more intuitively and accurately via specialized localized transfer functions in our segmentation-based volume rendering framework.

ACKNOWLEDGMENTS

This paper is supported by the National Basic Research Program of China (973 Program) under Grant No. 2011CB707700, the Knowledge Innovation Project of the Chinese Academy of Sciences under Grant No. KSCX2-YW-R-262, the National Natural Science Foundation of China under Grant Nos. 81042002, 81071218, 30873462, and 60910006. The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] M. Levoy, "Display of Surfaces from Volume Data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29-37, May 1988.

[2] G. Kindlmann and J.W. Durkin, "Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering," *Proc. IEEE Symp. Volume Visualization (VVS '98)*, pp. 79-86, 1998.

[3] F.-Y. Tzeng, E.B. Lum, and K.-L. Ma, "An Intelligent System Approach to Higher-Dimensional Classification of Volume Data," *IEEE Trans. Visualization and Computer Graphics,* vol. 11, no. 3, pp. 273-284, May/June 2005.

[4] C.R. Salama, M. Keller, and P. Kohlmann, "High-Level User Interfaces for Transfer Function Design with Semantics," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 1021-1028, Sept. 2006.

[5] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional Transfer Functions for Interactive Volume Rendering," *IEEE Trans. Visualization and Computer Graphics,* vol. 8, no. 3, pp. 270-285, July-Sept. 2002.

[6] C. Lundström, P. Ljung, and A. Ynnerman, "Local Histograms for Design of Transfer Functions in Direct Volume Rendering," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 6, pp. 1570-1579, Nov./Dec. 2006.

[7] P. Šereda, A.V. Bartrolí, I.W.O. Serlie, and F.A. Gerritsen, "Visualization of Boundaries in Volumetric Data Sets Using lh Histograms," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 2, pp. 208-218, Mar./Apr. 2006.

[8] I. Serlie, R. Truyen, J. Florie, F. Post, L. van Vliet, and F. Vos, "Computed Cleansing for Virtual Colonoscopy Using a Three-Material Transition Model," *Proc. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI),* pp. 175-183, 2003.

[9] C.R. Johnson and J. Huang, "Distribution-Driven Visualization of Volume Data," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 5, pp. 734-746, Sept. 2009.

[10] C.L. Bajaj, V. Pascucci, and D.R. Schikore, "The Contour Spectrum," *Proc. IEEE Visualization Conf. '97,* pp. 167-173, 1997.

[11] Y. Sato, C.-F. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis, "Tissue Classification Based on 3d Local Intensity Structures for Volume Rendering," *IEEE Trans. Visualization and Computer Graphics,* vol. 6, no. 2, pp. 160-180, Apr.-June 2000.

[12] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller, "Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications," *Proc. 14th IEEE Visualization (VIS '03) Conf.,* pp. 513-520, 2003.

[13] S. Roettger, M. Bauerand, and M. Stamminger, "Spatialized Transfer Functions," *Proc. EUROGRAPHICS '05/IEEE VGTC Symp. Visualization,* pp. 271-278, 2005.

[14] C. Correa and K.-L. Ma, "Size-Based Transfer Functions: A New Volume Exploration Technique," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 6, pp. 1380-1387, Nov. 2008.

[15] C. Correa and K.-L. Ma, "The Occlusion Spectrum for Volume Classification and Visualization," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 6, pp. 1465-1472, Nov. 2009.

[16] J.J. Caban and P. Rheingans, "Texture-Based Transfer Functions for Direct Volume Rendering," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 6, pp. 1364-1371, Nov./Dec. 2008.

[17] M. Hadwiger, F. Laura, C.R. Salama, T. Höllt, G. Geier, and T. Pabel, "Interactive Volume Exploration for Feature Detection and Quantification in Industrial Ct Data," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 6, pp. 1507-1514, Nov./Dec. 2008.

[18] G.H. Weber, S.E. Dillard, H. Carr, V. Pascucci, and B. Hamann, "Topology-Controlled Volume Rendering," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 2, pp. 330-341, Mar./Apr. 2007.

[19] I. Fujishiro, T. Azuma, and Y. Takeshima, "Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3d Field Topology Analysis (Case Study)," *Proc. Conf. Visualization (VIS '99),* pp. 467-470, 1999.

[20] R. Huang and K.-L. Ma, "RGVis: Region Growing Based Techniques for Volume Visualization," *Proc. 11th Pacific Conf. Computer Graphics and Applications,* pp. 355-363, 2003.

[21] S. Owada, F. Nielsen, and T. Igarashi, "Volume Catcher," *Proc. Symp. Interactive 3D Graphics and Games,* pp. 111-116, 2005.

[22] J. Marks et al., "Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation," *ACM Computer Graphics (Proc. ACM SIGGRAPH '97),* pp. 389-400, Aug. 1997.

[23] X. Yuan, N. Zhang, M. Nguyen, and B. Chen, "Volume Cutout," *The Visual Computer,* vol. 21, no. 8, pp. 745-754, 2005.

[24] Y. Wu and H. Qu, "Interactive Transfer Function Design Based on Editing Direct Volume Rendered Images," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 5, pp. 1027-1040, Sept./Oct. 2007.

[25] G. Klette, "Topologic, Geometric, or Graph-Theoretic Properties of Skeletal Curves," PhD dissertation, Groningen Univ., 2007.

[26] D. Reniers, J. van Wijk, and A. Telea, "Computing Multiscale Curve and Surface Skeletons of Genus 0 Shapes Using a Global Importance Measure," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 2, pp. 355-368, Mar./Apr. 2008.

[27] R.L. Blanding, G.M. Turkiyyah, D.W. Storti, and M.A. Ganter, "Skeleton-Based Three-Dimensional Geometric Morphing," *Computational Geometry: Theory and Applications,* vol. 15, nos. 1-3, pp. 129-148, 2000.

[28] A. Meijster, J.B.T.M. Roerdink, and W.H. Hesselink, "A General Algorithm for Computing Distance Transforms in Linear Time," *Mathematical Morphology and Its Application to Image and Signal Processing,* pp. 331-340, Springer, 2000.

[29] D. Coeurjolly and A. Montanvert, "Optimal Separable Algorithms to Compute the Reverse Euclidean Distance Transformation and Discrete Medial Axis in Arbitrary Dimension," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 3, pp. 437-448, Mar. 2007.

[30] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1124-1137, Sept. 2004.

[31] H. Lombaert, Y. Sun, L. Grady, and C. Xu, "A Multilevel Banded Graph Cuts Method for Fast Image Segmentation," *Proc. 10th IEEE Int'l Conf. Computer Vision (ICCV '05),* pp. 259-265, 2005.

[32] Y. Boykov and G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation," *Int'l J. Computer Vision,* vol. 70, no. 2, pp. 109-131, 2006.

[33] M. Hadwiger, C. Berger, and H. Hauser, "High-Quality Two-Level Volume Rendering of Segmented Data Sets on Consumer Graphics Hardware," *Proc. 14th IEEE Visualization (VIS '03) Conf.,* pp. 301-308, 2003.

[34] NVIDA, *Nvidia CUDA Programming Guide Version 2.3.* NVIDIA Corporation, July 2009.

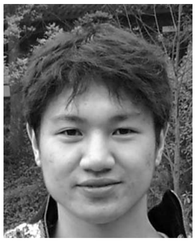
[35] J. Tian, J. Xue, Y. Dai, J. Chen, and J. Zheng, "A Novel Software Platform for Medical Image Processing and Analyzing," *IEEE Trans. Information Technology in Biomedicine,* vol. 12, no. 6, pp. 800-812, Nov. 2008.

[36] L.R. Dice, "Measures of the Amount of Ecologic Association between Species," *Ecology,* vol. 26, no. 3, pp. 297-302, 1945.

[37] D. Lesage, E.D. Angelini, I. Bloch, and G. Funka-Lea, "A Review of 3D Vessel Lumen Segmentation Techniques—Models, Features and Extraction Schemes," *Medical Image Analysis,* vol. 13, no. 6, pp. 819-845, 2009.

[38] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 7, pp. 629-639, July 1990.

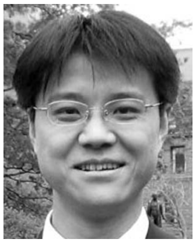
[39] I. Gath and A. Geva, "Unsupervised Optimal Fuzzy Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 773-780, July 1989.



Dehui Xiang received the BE degree in automation from Sichuan University, China, in 2007. He is currently working toward the PhD degree in the Medical Image Processing Group at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include volume rendering, medical image analysis, computer vision, and pattern recognition.

Jie Tian (M'02-SM'06-F'10) received the PhD degree (with honors) in artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1992. From 1995 to 1996, he was a postdoctoral fellow at the Medical Image Processing Group, University of Pennsylvania, Philadelphia. Since 1997, he has been a professor at the Institute of Automation, Chinese Academy of Sciences, where he has been involved in the research in Medical Image Processing Group. His main research interests include the medical image process and analysis and pattern recognition. He is currently the Beijing Chapter chair of the Engineering in Medicine and Biology Society of the IEEE. He is a fellow of the IEEE.

Fei Yang received the BE degree in biomedical engineering from Beijing Jiaotong University, China, in 2009. He is currently working toward the PhD degree in the Medical Image Processing Group at the Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include volume rendering, medical image analysis, and GPU computing.



Qi Yang received the MS degree in medical imaging from Harbin Medical University, Heilongjiang, in 2006, and the PhD degree in medical imaging from Xuanwu Hospital, Capital Medical University, Beijing, in 2009. The objective of his current research is the development and clinical application of fast MR imaging techniques for the evaluation of the cardiovascular system. His research interests include to develop fast MRI techniques to acquire high-resolution images of coronary arteries, to characterize the composition of atherosclerosis using MRI, and to evaluate the utility of MR contrast agents in imaging the anatomy and function of the heart.

Xing Zhang received the BE degree in biomedical engineering from Beijing Jiaotong University, China. She is currently working toward the PhD degree at the Institute of Automation, Chinese Academy of Sciences. Her research interests include medical image processing, including medical image segmentation using graph cut method, and 3D medical image segmentation based on statistical shape models. She is a student member of the IEEE.

Qingde Li received the BS degree in mathematics from Beijing Normal University, China, in 1982, and the PhD degree in computer science from the University of Hull, United Kingdom, in 2002. From 1998 to 2002, he was a professor in the School of Mathematics and Computer Science, Anhui Normal University, China. He was a visiting scholar in the Department of Applied Statistics, University of Leeds, United Kingdom, from October 1990 to May 1992, and in the Department of Computing, University of Bradford, United Kingdom, from September 1996 to August 1997. He is currently working as a lecturer in the Department of Computer Science, University of Hull, United Kingdom. His current research interests include GPU-based volume data visualization, implicit geometric modeling, and computer graphics.

Xin Liu received the MD degree from the Medical School of Wuhan University, China, in 1988, and the PhD degree in diagnostic radiology from PLA Postgraduate Medical School (PLA General Hospital), Beijing, China, in July 2006. He worked as a research associate in the Department of Radiology of Northwestern University, Chicago, from August 2006 to October 2008. He is currently a full-time professor at the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. His research interests include cardiovascular magnetic resonance imaging, noninvasive coronary angiography and vessel wall imaging, carotid atherosclerotic plaque characterization, and early detection of malignant tumor using MRI. He has published more than 20 peer-reviewed journal papers and international conference proceedings and one book chapter in *Novel Techniques for Imaging the Heart* published by the American Heart Association.