Computerized Medical Imaging and Graphics

Automatic transfer function design for medical visualization using visibility distributions and projective color mapping

Lile Cai [a,*], Wei-Liang Tay [a], Binh P. Nguyen [b], Chee-Kong Chui [b], Sim-Heng Ong [a,c]

[a] Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore
[b] Department of Mechanical Engineering, National University of Singapore, Singapore, Singapore
[c] Department of Bioengineering, National University of Singapore, Singapore, Singapore

ARTICLE INFO

Keywords:
Transfer function design
Volume visualization
Visibility distribution
Color mapping

ABSTRACT

Transfer functions play a key role in volume rendering of medical data, but transfer function manipulation is unintuitive and can be time-consuming; achieving an optimal visualization of patient anatomy or pathology is difficult. To overcome this problem, we present a system for automatic transfer function design based on visibility distribution and projective color mapping. Instead of assigning opacity directly based on voxel intensity and gradient magnitude, the opacity transfer function is automatically derived by matching the observed visibility distribution to a target visibility distribution. An automatic color assignment scheme based on projective mapping is proposed to assign colors that allow for the visual discrimination of different structures, while also reflecting the degree of similarity between them. When our method was tested on several medical volumetric datasets, the key structures within the volume were clearly visualized with minimal user intervention.

1. Introduction

In medical visualization, volume rendering projects 3-D volumetric data to 2-D screen, which addresses the lack of spatial perception present in traditional 2-D slice-by-slice viewing. While volume rendering is a powerful technique in medical visualization, a key challenge is the design of transfer functions (TFs), which assign optical properties, such as opacity and color to voxels, to voxels comprising the volume. A variety of attributes have been developed to construct the TF space. To assist user exploration, clustering and segmentation techniques are often applied to reveal the pattern of the TF space, and various manipulation widgets have been designed in order to facilitate user interactions. However, the numerous degrees of freedom in TF specification and the non-intuitive, position-independent nature of the TF space [1] make the design of a desirable TF a laborious trial-and-error process. Furthermore, most interaction widgets require domain knowledge of TF space for effective manipulation, but clinicians may not possess such knowledge. An automated transfer function design technique to reveal as much information of the volume to the viewer, or an intuitive framework for transferring their medical domain knowledge to manipulate TFs, would increase the ease of generating good and relevant visualizations.

In this paper, we introduce a system for obtaining meaningful volume renderings. The proposed method automatically assigns higher target visibility values to more important regions, which ensures that they are not occluded in the resulting image and thus presents a more informative visualization result to the user. The volume is colored by an automatic color assignment scheme based on projective mapping, which allows users to better distinguish different structures in the volume. The visibility distribution based opacity design method, together with our projective mapping based color assignment scheme, are shown to produce satisfactory visualization results on various volumetric datasets. Lastly, we describe an intuitive framework for applying domain knowledge to manipulate the visual parameters, allowing clinicians to more easily obtain their desired visualizations. Compared with state-of-the-art methods, our method requires minimal user intervention, while also providing clinicians with a more intuitive system for manipulating the TF space.

2. Related work

Existing methods for transfer function design can be divided into *data-driven* and *image-driven* methods. Data-driven methods extract various features from the volume data and assign optical properties based on them. A transfer function can be simply decided by voxel's scalar value, but a 1-D transfer function has inherent difficulties in visualizing datasets where one data value is associated with multiple boundaries. To facilitate the visualization of material

* Corresponding author. Tel.: +65 96375825.
E-mail address: Lile.Cai@nus.edu.sg (L. Cai).

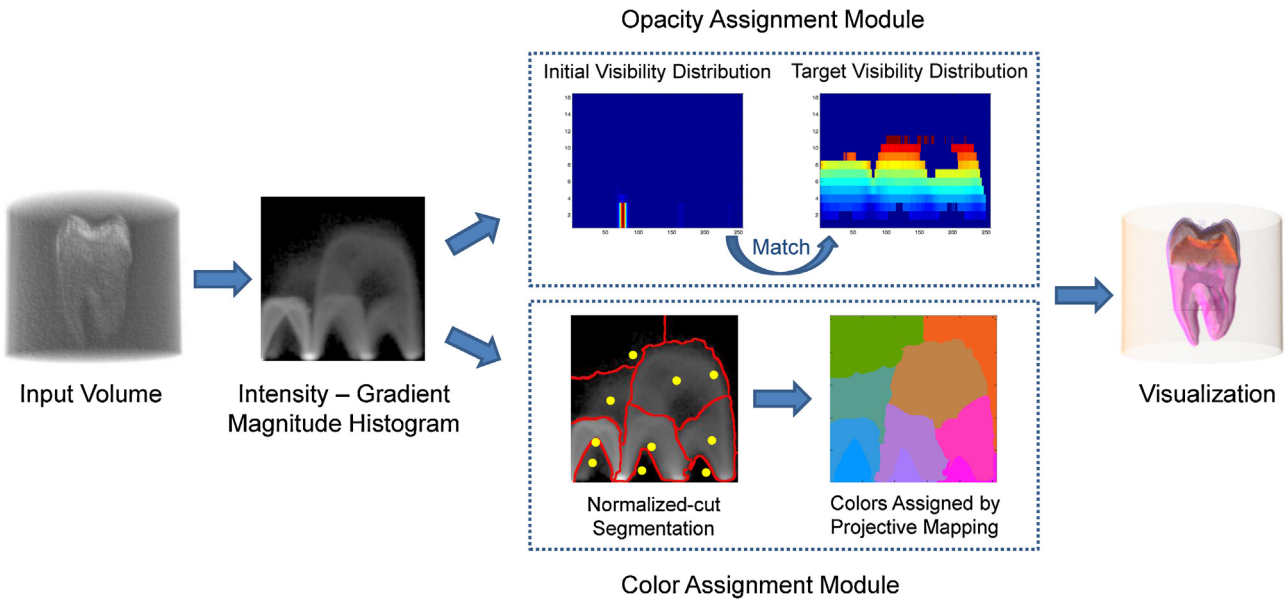

Fig. 1. System diagram of the proposed transfer function design method.

boundaries, Kindlmann and Durkin [2] proposed to include the derived attributes of scalar value, such as the first and second-order derivatives, into the domain of transfer function. Instead of using the derivatives of scalar value, Haidacher et al. [3] constructed the so-called statistical transfer function space using statistical properties, i.e., the mean and standard deviation of the scalar value. Another example is the LH histogram introduced by Šereda et al. [4]. LH histogram is a 2-D transfer function space that uses the lower and higher intensity of the two materials that form a boundary as the two axes. In the LH space, each boundary appears as a blob instead of an arch, which allows for easier detection of boundaries than the intensity-gradient magnitude (IGM) space. Higher-level features, such as feature size [5,6] and ambient occlusion [7], have also been used in transfer function design, which provide different separation abilities than the above-mentioned low-level properties.

Compared to data-driven methods, image-driven methods are usually more intuitive. Users do not have to go through the complex and tedious trial-and-error process of manipulating the transfer function space. Instead, they directly search for the most appealing rendered image in the space of renderings. Typically, stochastic search techniques, such as the genetic algorithm, are used in image-driven methods in order to find the optimal transfer function [8] or to disperse the rendering space as much as possible [9]. Wu and Qu [10] circumvented the time-consuming and non-intuitive TF design process by allowing users to fuse, blend or delete features in different rendered images without knowledge of the TF.

Clustering and segmentation are commonly used techniques in transfer function design, which divide the volume into a set of clusters or segments. Users assign optical properties to each cluster or segment and interaction with the clusters seems to be more intuitive than a direct interaction with the TF space. The clustering and segmentation can be conducted either in 3-D volume space or in 2-D transfer function space. Tzeng and Ma [11] applied ISODATA clustering algorithm in the spatial domain, which generated a set of material classes for user manipulation. Šereda et al. [12] introduced a framework in which the users interacted with a hierarchy of clusters generated in LH space. Maciejewski et al. [13] applied a non-parametric kernel density estimation method onto the IGM feature space in order to extract feature patterns. Röettger et al. [14] incorporated spatial information into the IGM histogram and performed clustering based on the barycenter and variance of each bin. Nguyen et al. [15] applied mean shift clustering to over-segment volume boundaries and then used hierarchical clustering to group similar voxels. Wang et al. [16] represented the IGM transfer function space with Gaussian mixture model and used the resulting ellipses to compose transfer functions. Cheuk et al. [17] introduced a hierarchical volume exploration scheme based on Normalized-cut algorithm. Wang et al. [18] adopted the Morse theory to automatically decompose the IGM feature space into a set of valley cells.

The *visibility histogram* introduced by Correa and Ma [19,20] describes the distribution of visibility in a volume-rendered image. It can act as immediate feedback on the quality of the rendered image as users explore the transfer function space. It also proved to be useful in facilitating automatic transfer function design. Ruiz et al. [21] proposed to generate a transfer function automatically by minimizing the Kullback–Leibler divergence between the observed visibility distribution and a target visibility distribution. Wang et al. [22] extended the idea of *visibility histogram* to *feature visibility*. They introduced an interaction scheme where the opacity of each feature was generated automatically based on a user-defined visibility value.

3. System overview

Fig. 1 presents an overview of our automatic transfer function design method for medical visualization. There are two major modules in our system, responsible for opacity assignment and color assignment respectively. Given a volume dataset, a preprocessing module first computes the 2-D IGM histogram. The opacity of each bin is decided by an optimization process which matches the observed visibility distribution to the target visibility distribution; the target visibility distribution can be automatically set based on the IGM histogram of the volume, or set based on predetermined profiles relevant for the medical task. The IGM histogram is then segmented into several regions, each approximating a spatially connected structure, by the Normalized-cut algorithm [23]. Subsequently, the center of each segment is mapped to the CIELAB color space in order to assign a unique color to each region. The resulting opacity TF and color TF are applied to render the volume and generate the visualization result.

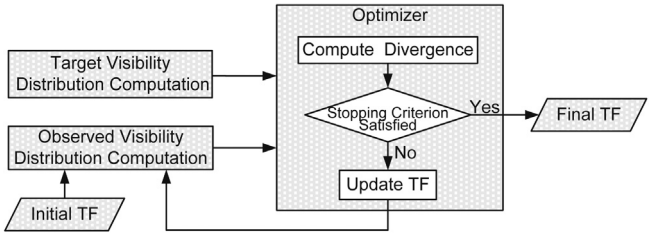

Fig. 2. Flowchart of the proposed opacity definition scheme.

4. Opacity assignment based on matching of visibility distributions

Opacity is the most important visual property in volume rendering. It decides the transparency of each voxel in the rendered image. In the realm of automatic design methods, where user specification and modification are not required, opacity can be simply decided by intensity and modulated by gradient magnitude. The drawback of this method is that although the intention is to highlight high-intensity and high-gradient regions, such regions would still be invisible in the rendered image due to the presence of occlusions within the volume. We propose to account for these occlusions by deciding opacity in such a way that important regions are ensured to achieve a predefined amount of visibility in the final image. Even if interesting regions are located at the very inner part of a volume and greatly occluded, our method can still make them readily visible in the rendered image, as our method would modify the opacity so that these regions achieve the predefined amount of visibility.

Fig. 2 displays the flowchart of our opacity definition scheme. Given a volume, the target visibility distribution is defined based on each bin's importance, which is quantitatively measured by its intensity, gradient magnitude and self-information [24]. Users can modify the target visibility distribution using their prior knowledge. An initial transfer function, which is decided by intensity linearly, is applied to assign opacity to each voxel. The observed visibility distribution is computed from a set of sampled viewpoints. The optimizer then iteratively modifies the TF until the stopping criterion is satisfied. In the following section, we discuss how to compute the observed visibility distribution of a volume, how we define the target visibility distribution and how to measure the divergence between the observed visibility distribution and the target visibility distribution. Finally, we provide the mathematical details on how to minimize the divergence by Newton's method.

4.1. Observed visibility distribution

The visibility of a sample represents the contribution of the sample to the final image in terms of opacity. In the front-to-back composition of ray-casting algorithm, the visibility $\mathrm{vis}(z|v)$ of a voxel z from viewpoint v can be computed by the following equation:

$$\mathrm{vis}(z|v) = A(X(z))(1 - alpha_{v\to z}), \tag{1}$$

where $alpha_{v\to z}$ is the accumulated opacity before voxel z along the ray that is projected from viewpoint v and passes voxel z; $X(z)$ is a classification value of voxel z, typically the scalar value or gradient value, and $A(X)$ is the opacity transfer function. Eq. (1) implies that the visibility of a voxel depends on its own opacity (i.e., $A(X(z))$) as well as the remaining transparency before it (i.e., $(1 - alpha_{v\to z})$).

Summing the visibility values of all voxels belonging to the same histogram bin b from viewpoint v obtains the visibility $\mathrm{vis}(b|v)$:

$$\mathrm{vis}(b|v) = \sum_{z\in bin_b} \mathrm{vis}(z|v). \tag{2}$$

The total visibility $\mathrm{vis}(v)$ of viewpoint v is the sum of $\mathrm{vis}(b|v)$ over all bins:

$$\mathrm{vis}(v) = \sum_{b\in\mathcal{B}} \mathrm{vis}(b|v). \tag{3}$$

The visual probability $p(b|v)$ is defined as the normalized visibility of bin b from viewpoint v:

$$p(b|v) = \frac{\mathrm{vis}(b|v)}{\mathrm{vis}(v)}. \tag{4}$$

The visual probability $p(b|v)$ can be regarded as the probability that the voxels in bin b are visible in the rendered image. The higher the probability is, the more visible bin b is in the rendered image. The visual probability $p(b)$ is computed as the weighted average of bin b's visual probabilities from all sampled viewpoints:

$$p(b) = \sum_{v\in\mathcal{V}} p(v)p(b|v), \tag{5}$$

where $p(v) = \mathrm{vis}(v)/\sum_{i\in\mathcal{V}}\mathrm{vis}(i)$. The volume's observed visibility distribution $p(B)$ is formed from $p(b)$ by computing all the bins' visual probabilities. From the visibility distribution of a volume, we can identify which feature is highly visible on the rendered image and which is not. By modifying the visibility distribution to approach the desired one, we can generate an informative transfer function that assigns more visibility to important features.

4.2. Target visibility distribution

4.2.1. Automatic approach

In our system, target visibility distribution serves as an automatic guide in assigning opacity. Generally, regions of high intensity and gradient magnitude are important and should be assigned higher visibility. Apart from intensity and gradient magnitude, we also measure the importance of a region by its information amount. In the following, we present an information-based approach to define target visibility distribution.

Information can be measured based on the quality or quantity of data [25]. Quality assessments are subjective and user input is necessary to specify which region is important and which is not. When such prior knowledge is unavailable, we need to rely on the quantity, i.e., frequency, of the data to measure the information. According to information theory, events with low occurrence carry more information than events with high occurrence because people have more uncertainty about them. The *self-information*, which is used to capture this property of event occurrence, is defined as:

$$I(e) = -\log p(e), \tag{6}$$

where $p(e)$ is the probability of event e. If we regard the whole volume as a random variable and each histogram bin as one of its possible values, the probability of each bin can be estimated from its occurrence in the histogram. With the probability, the self-information contained in each bin can be measured by Eq. (6). In the context of medical data, bins containing higher self-information correspond to features with smaller size (e.g., border areas or tissue segments), which are always of user's interest, and bins containing lower self-information can be interpreted as larger regions with the same value (e.g., the background or air), which do not provide much useful information about the volume to the user. Apparently, bins carrying more information are more important and should be

highlighted in the rendered image. This motivates the following target visibility distribution definition strategies:

$$\begin{aligned}(a)q(b) &= \frac{I(b)\cdot \mathrm{Int}(b)}{\sum_{i\in\mathcal{B}} I(i)\cdot \mathrm{Int}(i)},\\ (b)q(b) &= \frac{I(b)\cdot \mathrm{Gm}(b)}{\sum_{i\in\mathcal{B}} I(i)\cdot \mathrm{Gm}(i)},\end{aligned} \tag{7}$$

where $q(b)$ is the target visibility value of bin b, $I(b)$ is the self-information of bin b, $\mathrm{Int}(b)$ is the intensity of bin b and $\mathrm{Gm}(b)$ is the gradient magnitude of bin b. Users can decide which target visibility distribution to use based on their visualization interests. For example, if the user wants to highlight high-intensity region, the self-information can be multiplied by intensity; if the user is more interested in boundaries, the self-information can be multiplied by gradient magnitude.

Compared to the target distribution strategies proposed by Ruiz et al. [21], where the target visibility of each bin is proportional to intensity or gradient weighted by bin occurrence, our scheme favors bins with higher information, i.e., lower occurrence. Our rational is that, if the target visibility is weighted by bin occurrence, large-sized features, which usually correspond to background or unimportant structures (like the material between the tooth and the container in the tooth dataset), would obtain high visibility in the rendered image, making it difficult to clearly observe inner structures. On the contrary, if we weight the target distribution by self-information, a large feature would be assigned a low target visibility value and thus be rendered with a very small opacity, making it possible to reveal more important features in the volume. We will illustrate the influence of target visibility distributions in Section 6.

It should be noted that, if the number of voxels falling into a bin is very low, the self-information of this bin would be very high by definition and would be assigned a large target visibility value. To balance the visibility distribution among structures of different sizes, we set the target visibility value of bins with occurrence below certain threshold to be zero.

4.2.2. Incorporation of prior knowledge

In medical applications, clinicians usually have domain knowledge on the corresponding value range for a particular tissue. This prior knowledge can be incorporated in defining the target visibility distribution such that the tissue of interest can be highlighted in the rendered image. The prior knowledge is represented with an intensity level that the user would like to highlight, and a deviation value that controls how much context around the selected intensity should be provided in the visualization. A Gaussian function is defined based on these two values:

$$g(b) = \exp\left(-\frac{(\mathrm{Int}(b)-\mathrm{Int}_u)^2}{2\sigma^2}\right), \tag{8}$$

where $\mathrm{Int}(b)$ is the intensity of bin b, Int_u is the user-selected intensity level, and σ is the deviation. The target visibility defined in Eq. (7) is then multiplied by $g(b)$ (and normalized again) in order to assign a weightage to the automatic generated target visibility distribution. This allows the creation of visualization results that better meet users' needs.

4.3. Jensen–Shannon divergence

After obtaining the observed visibility distribution and target visibility distribution, we need to find a measure to quantify the distance between them. A natural choice is the Kullback–Leibler (KL) divergence, also called informational divergence, which is defined by

$$D_{KL}(P\|Q) = \sum_{i=1}^{n} p_i \log\frac{p_i}{q_i}, \tag{9}$$

where $P=\{p_1, p_2, \ldots, p_n\}$ is the observed visibility distribution and $Q=\{q_1, q_2, \ldots, q_n\}$ is the target visibility distribution. However, there are some issues with KL divergence measure that make it less than ideal [26,27]. First, it is not a true metric as it is asymmetric and does not obey the triangle inequality. Second, KL divergence is undefined if $q_i=0$ and $p_i \neq 0$. Third, KL divergence does not offer any nice upper bounds. In matching two visibility distributions, we may frequently need to set some target visibility values to be zero, so that uninteresting regions can be rendered totally invisible in the final image. The symmetric Jensen–Shannon (JS) divergence is a more suitable candidate to match two visibility distributions, which is defined by:

$$D_{JS}(P\|Q) = D_{JS}(Q\|P) = \frac{1}{2}(D_{KL}(P\|M) + D_{KL}(Q\|M)), \tag{10}$$

where $M=(P+Q)/2$. JS divergence can also be expressed in terms of entropy:

$$D_{JS}(P\|Q) = H\left(\frac{1}{2}P + \frac{1}{2}Q\right) - \frac{1}{2}\left(H(P)+H(Q)\right). \tag{11}$$

The JS measure does not have the zero target visibility problem and is bounded by $0 \le D_{JS}(P\,\|\,Q) \le 1$ (using the base 2 logarithm). In [26], JS divergence is used to measure the similarity of two viewpoints.

4.4. Visibility distribution matching method

In this section, we discuss how to employ Newton's method to match the observed visibility distribution to the target visibility distribution.

Newton's method is a second-order optimization method that achieves a faster converging rate than gradient descent. Given an objective function $f(\mathbf{x})$, Newton's method finds the solution to minimize $f(\mathbf{x})$ iteratively by applying the following updating rule:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - sH^{-1}(f(\mathbf{x}_t))\nabla f(\mathbf{x}_t), \tag{12}$$

where s is the step size, H is the Hessian matrix and ∇ is the gradient operator.

In our system, each histogram bin is assigned the same opacity value. Thus, the visual probability of bin i has the following form according to Eqs. (1)–(5):

$$p_i = \frac{e_i\alpha_i}{E}, \tag{13}$$

where α_i is the opacity value of bin i, e_i is the sum of remaining transparency of voxels in bin i, and $E=\sum_{j=1}^{n} e_j\alpha_j$ can be interpreted as the total absorbed energy of voxels from all n bins in a volume. In order to apply Newton's method to minimize the JS divergence, the analytic forms of the first and second order derivatives of $D_{JS}(P\,\|\,Q)$, i.e., $\frac{\partial D_{JS}(P\|Q)}{\partial\alpha_i}$ and $\frac{\partial^2 D_{JS}(P\|Q)}{\partial\alpha_i\partial\alpha_j}$, $i, j = 1, 2, \cdots, n$, are needed. We use the entropy summation form of JS divergence (i.e., Eq. (11)) to deduce the derivatives. Here, $H(Q)$ is independent of α_i, hence its derivative $(\partial H(Q))/(\partial\alpha_i)=0$. Therefore,

$$\frac{\partial D_{JS}(P\|Q)}{\partial\alpha_i} = \frac{\partial H\left(\frac{1}{2}P+\frac{1}{2}Q\right)}{\partial\alpha_i} - \frac{1}{2}\frac{\partial H(P)}{\partial\alpha_i}. \tag{14}$$

For the first term in Eq. (14):

$$\frac{\partial H\left(\frac{1}{2}P+\frac{1}{2}Q\right)}{\partial\alpha_i} = -\frac{\partial\sum_{j=1}^{n}\left(\frac{1}{2}p_j+\frac{1}{2}q_j\right)\log\left(\frac{1}{2}p_j+\frac{1}{2}q_j\right)}{\partial\alpha_i} = -\sum_{j=1}^{n}\left(\frac{1}{2}\frac{\partial p_j}{\partial\alpha_i}\log\left(\frac{1}{2}p_j+\frac{1}{2}q_j\right)+\frac{1}{2}\frac{\partial p_j}{\partial\alpha_i}\right). \tag{15}$$

We assume that e_i and E are independent of α_i, thus:

$$\frac{\partial p_j}{\partial\alpha_i} \approx \begin{cases} e_i/E, & i=j \\ 0, & \text{otherwise} \end{cases}. \tag{16}$$

Therefore,

$$\frac{\partial H\left(\frac{1}{2}P+\frac{1}{2}Q\right)}{\partial\alpha_i} \approx -\frac{1}{2}\frac{e_i}{E}\left(\log\left(\frac{1}{2}p_i+\frac{1}{2}q_i\right)+1\right). \tag{17}$$

The derivative of the entropy term $H(P)$ in Eq. (14) is:

$$\frac{\partial H(P)}{\partial\alpha_i} = -\frac{\partial\sum_{j=1}^{n}p_j\log p_j}{\partial\alpha_i} = -\sum_{j=1}^{n}\left(\frac{\partial p_j}{\partial\alpha_i}\log p_j+\frac{\partial p_j}{\partial\alpha_i}\right) \approx -\frac{e_i}{E}(\log p_i+1). \tag{18}$$

Combining Eqs. (17) and (18), we obtain the first derivative of $D_{JS}(P\,\|\,Q)$:

$$\frac{\partial D_{JS}(P\|Q)}{\partial\alpha_i} \approx \frac{1}{2}\frac{e_i}{E}\left(\log p_i-\log\left(\frac{1}{2}p_i+\frac{1}{2}q_i\right)\right) = \frac{1}{2}\frac{p_i}{\alpha_i}\left(\log p_i-\log\left(\frac{1}{2}p_i+\frac{1}{2}q_i\right)\right). \tag{19}$$

For Newton's method, we use Eqs. (16) and (19) to obtain the second derivative of $D_{JS}(P\,\|\,Q)$:

$$\frac{\partial^2 D_{JS}(P\|Q)}{\partial\alpha_i\partial\alpha_j} \approx \begin{cases} \frac{1}{2}\left(\frac{p_i}{\alpha_i}\right)^2\left(\frac{1}{p_i}-\frac{1}{p_i+q_i}\right), & i=j \\ 0, & \text{otherwise} \end{cases}. \tag{20}$$

With these derivatives, we can apply Newton's method to solve the optimization problem. Note that in Eq. (16), e_i and E are assumed to be independent of α_i in order to obtain a simplified close-form solution. Our experiments showed that the Newton's method can still converge quickly despite this assumption.

5. Color assignment based on projective mapping

The original method described by Correa and Ma [19,20] does not include any scheme for the assignment of color, but color is an important property for volume rendering. A good color transfer function allows for the visual discrimination of dissimilar structures, and is particularly important for distinguishing the boundaries between materials. We propose a color assignment scheme based on CIELAB color space projection to address this problem. Our scheme consists of the following three steps, as illustrated in Fig. 3:

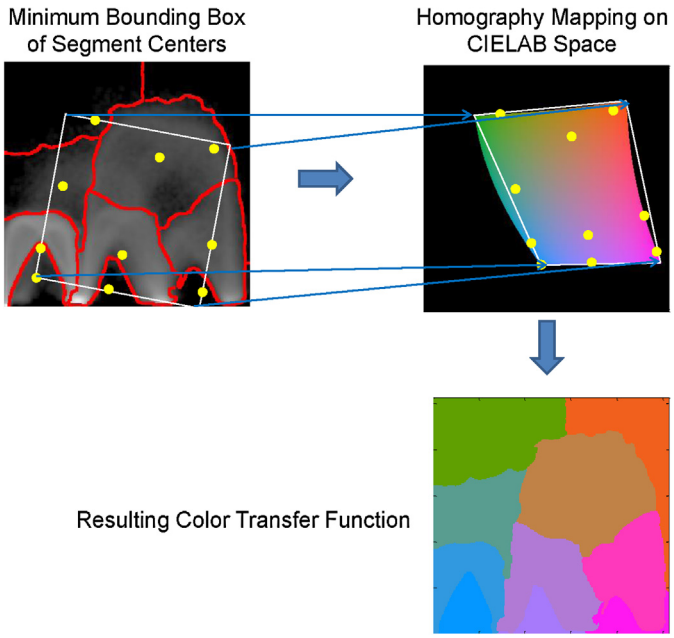

Fig. 3. Homography mapping of minimum bounding box onto CIELAB color space.

1. Histogram segmentation using Normalized-cut algorithm on the log-scale IGM histogram.
2. Computation of the minimum bounding box enclosing the center points of all segments.
3. Mapping the minimum bounding box to the CIELAB space using a homography projection.

CIELAB is a perceptually uniform color space where the perceptual color difference is proportional to Euclidean distance between two colors in the color space. Therefore, a color transfer function using CIELAB can directly reflect the similarity of two structures in the transfer function space. Also, in order to maximize the color separation between structures, a homography transformation is used to map the IGM space to the CIELAB space such that the color space is fully utilized.

In order to assign different colors to different features, classification or segmentation can be performed on the 3-D volume to extract the different regions of the volume; however, doing so is both complex and time-consuming. A simplification commonly adopted in volume rendering literature is to segment the 2-D histogram. It has been shown that continuity in the IGM transfer function space reasonably approximates spatial continuity in the 3-D volume. Cheuk et al. [17] recursively applied Normalized-cut onto the IGM histogram, and the resulting hierarchy of segments was used to guide the user's exploration of the volume. In the first step of our color assignment scheme, We apply Normalized-cut to segment the log-scale IGM histogram into a predefined number of segments. Each segment will then be assigned a color based on a mapping to the CIELAB color space.

In the second step, each segment i is first represented by the segment centroid (xc_i, yc_i):

$$(xc_i, yc_i) = \frac{\sum_{j\in segment_i}(x_j, y_j)\log occ_j}{\sum_{j\in segment_i}\log occ_j}, \tag{21}$$

where occ_j is the occurrence of bin j, and (x_j, y_j) is the position of bin j. From the segment centroids, a minimum bounding box containing all the center points is computed.

Lastly, the four corners of the minimum bounding box are mapped onto the corners of the CIELAB space using a homography

transformation. The four corners of the CIELAB space (with Lightness set to 60) are chosen as the target for the mapping to maximize the color separation in the final visualization. A homography can be estimated from these four pairs of matching points using Direct Linear Transformation algorithm [28]. Since the projected quadrilateral is slightly larger than the feasible CIELAB space, points that are projected outside feasible CIELAB space are relocated to the nearest point in the feasible region.

Colorization is a natural way for people to distinguish different structures. Our color assignment method increases the visual discrimination of dissimilar structures and helps users to better understand the volume. The visualization results generated by our system can serve as the basis for further user manipulation. For example, if the user is not satisfied with the automatic-assigned colors, he or she can modify the original color transfer function and assign a segment the color he or she prefers.

6. Results

This section introduces our experimental setups and demonstrates our automatic transfer function design method on several medical volumetric datasets.

6.1. Implementation and experimental setup

Our algorithm was implemented in CUDA C to use the GPU for the expensive visibility computation and volume rendering operations. For Normalized-cut algorithm, we used the Matlab code provided by [29]. The experiment platform was an Nvidia GTX680 graphics card and Intel i7 2600 CPU.

6.2. Parameter settings

There are several parameters involved in our system which control the tradeoff between the rendering quality and the processing time required. These parameters have been preset and fixed in our experiments.

Number of viewpoints: Computing over all possible viewpoints is computationally expensive, and [21] showed that increasing the number of viewpoints (beyond a small number) resulted in little visual improvement. Therefore, only a small number of viewpoints (i.e., 6), uniformly distributed on each face of a unit cube, were used to compute the observed visibility distribution.

Bin configurations: Different bin configurations were used for opacity specification and color assignment. For opacity specification, we used 256 intensity bins and 16 gradient bins. The number of histogram bins directly determines the resolution of visibility distribution. To balance histogram resolution and computational cost, more bins were assigned to quantities with a larger dynamic range (i.e., intensity) and fewer bins to quantities with a smaller dynamic range (i.e., gradient magnitude, whose values are typically distributed in the low end). For color assignment, the intensity and gradient magnitude were quantized into 256 bins each; the IGM histogram is thus represented as a 256×256 image for Normalized-cut segmentation.

Stopping criterion for Newton's method: In our experiments, early termination of the optimization process was performed after 10 iterations. Typically, the JS divergence was below 0.1 after 10 iterations. Further iterations reduced the divergence slowly without significant improvement in visualization quality. Fig. 4 demonstrates the convergence of Newton's method on the Tooth dataset (16-bit CT). It can be seen that the JS divergence dropped quickly in the first several iterations, but did not decrease much in the following iterations; regarding the visual quality, a small number of iterations sufficed to produce reasonably good results. Therefore, considering the tradeoff between visual quality and computational cost, an early termination strategy was adopted.

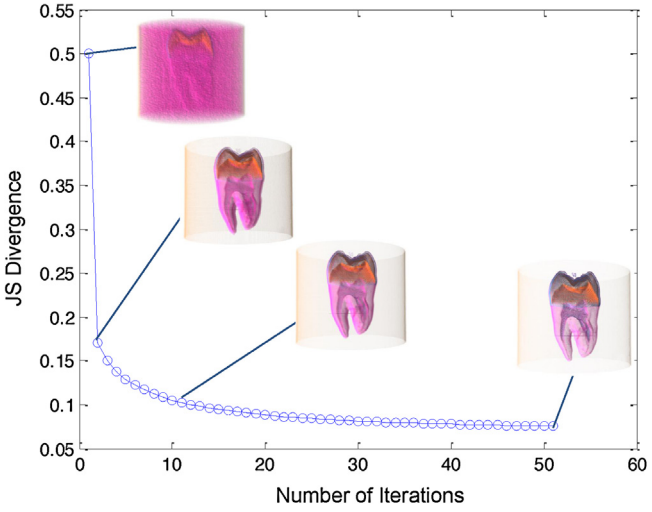

Fig. 4. Convergence of Newton's method on the Tooth dataset.

Step size for Newton's method: For the step size of Newton's method, a tradeoff between the convergence rate and accuracy needs to be considered. As Newton's method is a second-order optimization method and usually converges quickly, a small step size 0.05 was chosen. A small step size allows the algorithm to find a minimum more accurately while retaining an acceptable converging speed, as illustrated by Fig. 4.

Number of segments of Normalized-cut: In general, the more complex the volume is, the more segments we should use. In our experiment, we did not spend efforts in tuning this parameter and fixed it to 10.

6.3. Experimental results

We applied our transfer function design method on a variety of volumetric datasets. The influence of several important parameters on the visualization outcomes is described in this section.

Influence of target visibility distribution: The top row of Fig. 5 displays the rendering results on the DTI Scan dataset (8-bit Diffusion Tensor MRI) with the proposed information-based target visibility distributions. It can be seen that different target distributions resulted in different visualizations: when the target visibility is proportional to $I(b) \cdot \mathrm{Int}(b)$, regions with high intensity are rendered with higher opacity; when the target visibility is proportional to $I(b) \cdot \mathrm{Gm}(b)$, boundaries are emphasized.

The middle row of Fig. 5 displays the visualization results rendered with target visibility weighted by bin occurrence. It can be seen that under this strategy, large-size features (e.g., the background), were assigned high target visibility values and thus were still clearly visible in the rendered image, which made it difficult to observe more important features (e.g., the brain).

The bottom row of Fig. 5 displays the visualization results rendered with opacity linearly decided by intensity and gradient magnitude. Comparing with the results in the top row, it can be seen that the visibility distribution-based method made the inner structures more visible and produced clearer contrast between the cortex and the surrounding material.

Influence of threshold for zero target visibility: A small threshold for setting a bin's target visibility to zero would make small structures being assigned a large target visibility value. As a result, less visibility could be assigned to other structures in the volume. This would generate a transparent result with only a few small features overly highlighted. Fig. 6 demonstrates the influence of threshold

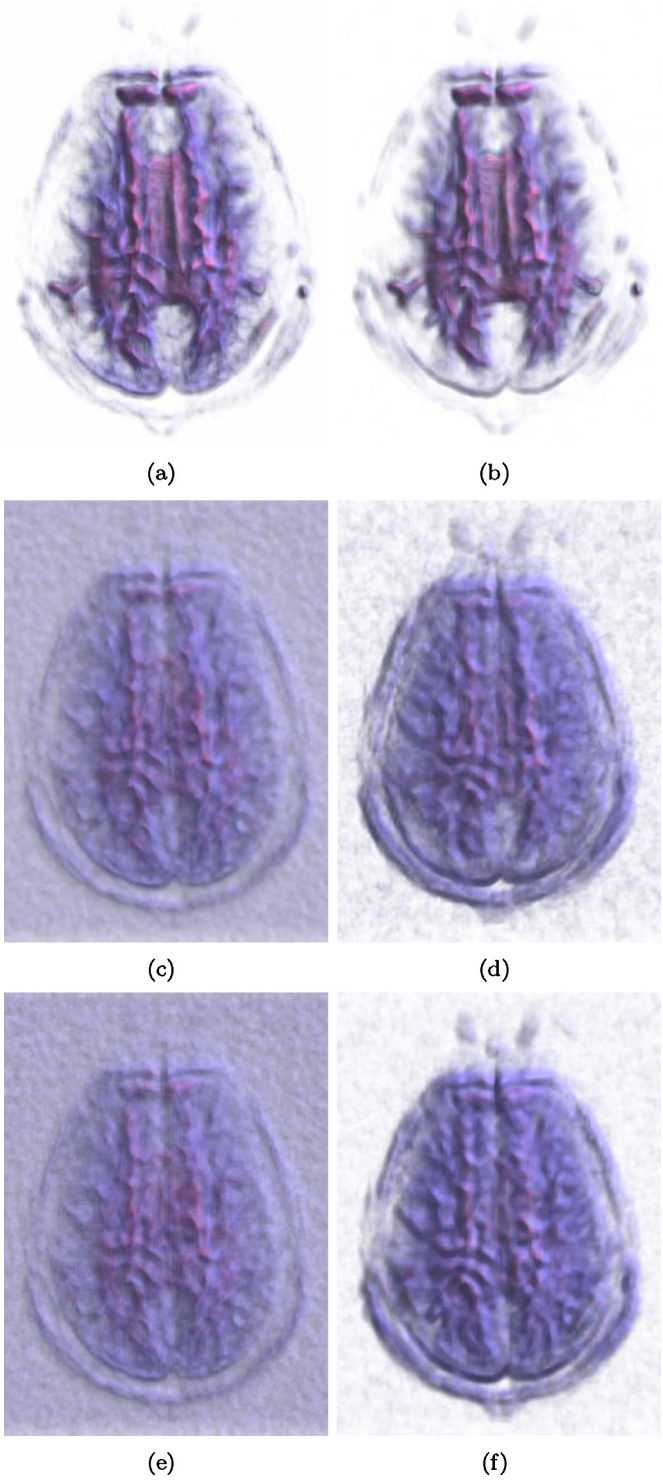

Fig. 5. Volume rendering results on the DTI Scan dataset. (a) Rendered with target visibility distribution $q(b) \propto I(b) \cdot \mathrm{Int}(b)$; (b) rendered with target visibility distribution $q(b) \propto I(b) \cdot \mathrm{Gm}(b)$; (c) rendered with target visibility distribution $q(b) \propto \mathrm{Occurence}(b) \cdot \mathrm{Int}(b)$; (d) rendered with target visibility distribution $q(b) \propto \mathrm{Occurence}(b) \cdot \mathrm{Gm}(b)$; (e) rendered with opacity linearly decided by intensity; (f) rendered with opacity linearly decided by gradient magnitude.

for zero target visibility on the Visible Male Head dataset (16-bit CT). It can be seen that when the threshold is small (Fig. 6(a) and (b)), the target visibility assigned to small structures (i.e., teeth, otic bones) is high, resulting in insufficient target visibility left for other structures, such as skull and backbone. By increasing the threshold, we can achieve a more balancing visibility distribution

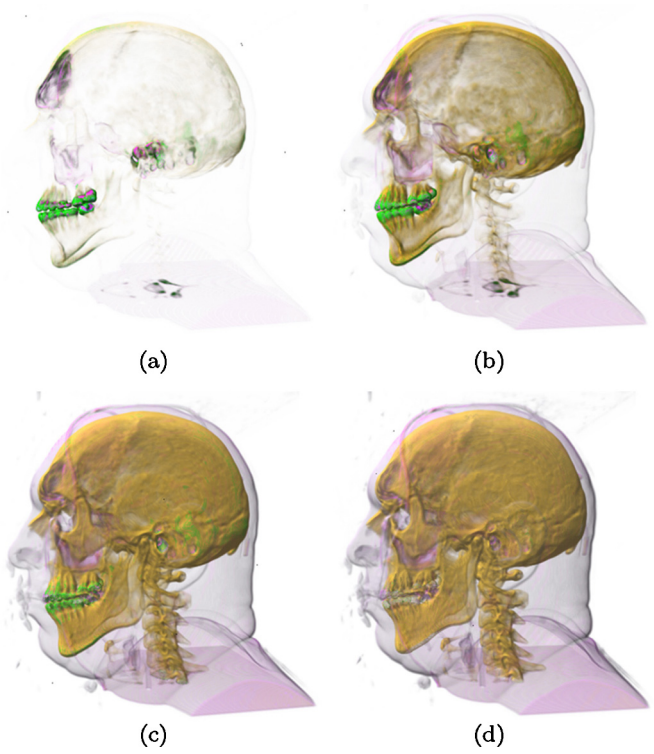

Fig. 6. Influence of threshold for zero target visibility on the Visible Male Head dataset. (a) Threshold = $10^{-7} \times$ volumesize; (b) threshold = $10^{-6} \times$ volumesize; (c) threshold = $10^{-5} \times$ volumesize; (d) threshold = $10^{-4} \times$ volumesize.

(Fig. 6(c)). However, this threshold can not be too large, otherwise some detailed structures would be forced to be invisible (i.e., the teeth tip in Fig. 6(d)), which is not desired. In our experiment, we found that a value between $10^{-6} \times$ volumesize (i.e., the total number of voxels in the volume) and $10^{-5} \times$ volumesize allowed a balancing visibility distribution for most datasets.

Specification of target visibility distribution using prior knowledge: Using the method described in Section 4.2.2, medical domain knowledge can be applied to define target visibility distributions to achieve different visualization outcomes. Fig. 7 shows three visualizations for the Pig dataset (16-bit CT) generated using different target visibility distributions. By modulating the contributions of different bins in the target visibility distribution, the visualization can be focused on soft tissue (Fig. 7(b)), or on bone and regions with contrast agent (Fig. 7(c)). Fig. 8 shows four visualizations for the CT Head dataset (16-bit CT). It can be seen that different user-selected values would result in visualizations emphasizing skin (Fig. 8(b)), brain (Fig. 8(c)) and bone (Fig. 8(d)). In our experiment the intensity is quantized into 256 levels, i.e., $\mathrm{Int}_u = 0, 1, 2, \ldots, 255$.

Table 1 lists the corresponding parameters for our test datasets. Our system took around 10 seconds to visualize a $256 \times 256 \times 128$ dataset. The time cost of our method can be greatly reduced by more efficient implementation of the involved algorithms, such as the Normalized-cut segmentation and histogram computation. Considering a user may take several minutes to get a satisfactory result by manually masking regions of interest in the transfer function space, our method effectively visualized a volume and saved a lot of efforts for the user.

6.4. Intuitive parameters for medical visualization

In this section, we discuss how our framework can be applied to obtain visualizations fulfilling different medical visualization objectives, and how the visualization interface can be designed to

Table 1
Parameters of the evaluation datasets.

Dataset	Resolution	Target visibility distribution	Threshold for zero target visibility	Optimization time	Clustering time
Tooth	$256 \times 256 \times 161$	$q(b) \propto I(b) \cdot \mathrm{Gm}(b)$	$10^{-5} \times$ volume size	2.557 s	6.170 s
		$q(b) \propto I(b) \cdot \mathrm{Int}(b)$		0.481 s	
DTI Scan	$128 \times 128 \times 58$	$q(b) \propto I(b) \cdot \mathrm{Gm}(b)$	$10^{-5} \times$ volume size	0.493 s	5.552 s
Visible Male Head	$128 \times 256 \times 256$	$q(b) \propto I(b) \cdot \mathrm{Gm}(b)$	$10^{-5} \times$ volume size	3.220 s	8.142 s
Pig	$256 \times 256 \times 128$	$q(b) \propto I(b) \cdot \mathrm{Int}(b)$	$10^{-6} \times$ volume size	1.844 s	8.037 s
CT Head	$256 \times 256 \times 113$	$q(b) \propto I(b) \cdot \mathrm{Gm}(b)$	$10^{-5} \times$ volume size	4.155 s	8.167 s

present visualization parameters in terms of concepts intuitive to clinicians.

Scale of visualized structures: Smaller structures in the volume may be structures of medical interest (tumors or blood clots), but small structures may also be the result of imaging noise. A parameter that allows the clinician to select the scale of visualized structures will be useful for medical visualization. As shown in Fig. 6, the threshold for zero target visibility controls the minimum size of structures that are visible. Lowering the threshold assigns a higher target visibility to small regions, thus increasing their visibility relative to larger structures. For medical visualization interfaces, this can be intuitively presented as a slider which controls the scale of visualized structures.

Type of tissues: Different biological tissues have different imaging characteristics. For example, bone has a much higher CT value than soft tissues. The target visibility distributions described in Section 4.2.1 may result in a visualization dominated by bone, while deemphasizing other soft tissues in the volume. If the desired visualization target is not bone, the automatic-generated target visibility distribution can be weighted by a Gaussian function centered at the user-selected value, as demonstrated in Figs. 7 and 8. One way to include this in a simplified interface is a slider which determines the position of the user-selected intensity level; this allows the visual contributions to be varied between soft tissues (which tend to have low CT values) and hard tissues (which tend to have high CT values).

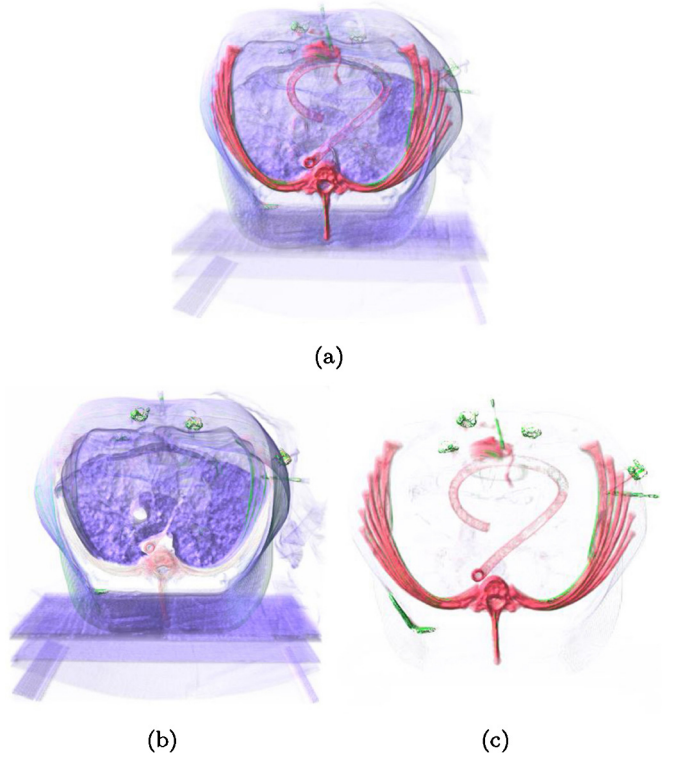

Fig. 7. Volume rendering results on the Pig dataset. (a) Rendered with target visibility distribution $q(b) \propto I(b) \cdot \mathrm{Int}(b)$; (b) rendered with prior knowledge $\mathrm{Int}_u = 20$, $\sigma = 20$; (c) rendered with prior knowledge $\mathrm{Int}_u = 100$, $\sigma = 20$.

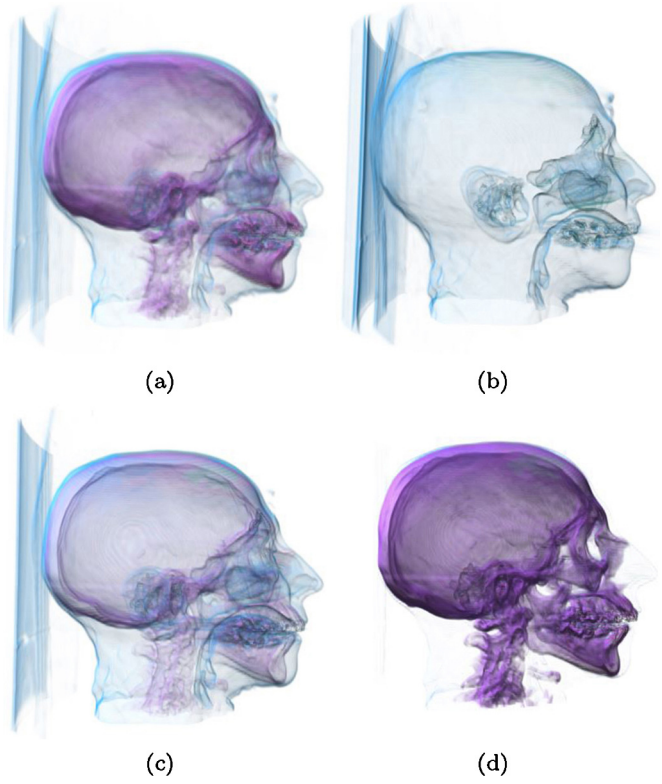

Fig. 8. Volume rendering results on the CT Head dataset. (a) Rendered with target visibility distribution $q(b) \propto I(b) \cdot \mathrm{Gm}(b)$; (b) rendered with prior knowledge $Int_u = 30$, $\sigma = 20$; (c) rendered with prior knowledge $\mathrm{Int}_u = 80$, $\sigma = 20$; (d) rendered with prior knowledge $\mathrm{Int}_u = 120$, $\sigma = 20$.

7. Conclusions

In this work, we proposed to derive an opacity transfer function by minimizing the Jensen–Shannon divergence between the target visibility distribution and the observed visibility distribution. The target visibility distribution was defined based on each bin's importance, which was measured by the intensity, gradient magnitude and self-information of the bin. Our method allowed users to employ their prior knowledge to define the target visibility distribution using a Gaussian function weighting. The essence of the algorithm was to use visibility distribution to guide the design of a transfer function and assign an opacity value to a bin based on the bin's target visibility, rather than directly based on its intensity and gradient magnitude. This allowed the problem of occlusion to be alleviated in the transfer function design process. For the color transfer function, we conducted Normalized-cut segmentation algorithm on the IGM histogram to divide the histogram into

several regions. The segment centers were computed and mapped using a homography projection to the CIELAB color space, and each segment was assigned a color determined by its projected coordinate. The Normalized-cut algorithm was shown to produce meaningful segments on the IGM histogram, with each segment roughly representing a structure in the 3-D volume.

The visualization results generated by the proposed method are comparable to existing state-of-the-art techniques. In future work, we hope to extend our algorithm to operate in other TF spaces, which would allow similar regions in difficult datasets to be more easily discriminated. In order to provide empirical evaluation for the proposed method, we could conduct user studies to compare the visual quality of rendered images generated by our method and other methods. Apart from visual quality, we could take into account the effectiveness and efficiency of the whole visualization system in the evaluation.

References

[1] Kniss J, Kindlmann G, Hansen C. Multi-dimensional transfer functions for interactive volume rendering. IEEE Trans Visual Comput Graph 2002;8(3):270–85.

[2] Kindlmann G, Durkin JW. Semi-automatic generation of transfer functions for direct volume rendering. In: Proceedings of IEEE symposium on volume visualization. 1998. p. 79–86.

[3] Haidacher M, Patel D, Bruckner S, Kanitsar A, Gröller M. Volume visualization based on statistical transfer-function spaces. In: Pacific visualization symposium (PacificVis), 2010 IEEE. 2010. p. 17–24.

[4] Šereda P, Bartroli AV, Serlie IWO, Gerritsen FA. Visualization of boundaries in volumetric data sets using LH histograms. IEEE Trans Visual Comput Graph 2006;12(2):208–18.

[5] Correa CD, Ma K-L. Size-based transfer functions: a new volume exploration technique. IEEE Trans Visual Comput Graph 2008;14(6):1380–7.

[6] Wesarg S, Kirschner M. 3D visualization of medical image data employing 2D histograms. In: Second international conference in visualisation, 2009. VIZ'09. 2009. p. 153–8.

[7] Correa CD, Ma K-L. The occlusion spectrum for volume visualization and classification. IEEE Trans Visual Comput Graph 2009;15(6):1465–72.

[8] He T, Hong L, Kaufman A, Pfister H. Generation of transfer functions with stochastic search techniques. In: Proceedings of the 7th conference on visualization'96, VIS'96. 1996. p. 227–34.

[9] Marks J, Andalman B, Beardsley PA, Freeman W, Gibson S, Hodgins J, et al. Design galleries: a general approach to setting parameters for computer graphics and animation. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques, SIGGRAPH'97. 1997. p. 389–400.

[10] Wu Y, Qu H. Interactive transfer function design based on editing direct volume rendered images. IEEE Trans Visual Comput Graph 2007;13(5):1027–40.

[11] Tzeng F-Y, Ma K-L. A cluster-space visual interface for arbitrary dimensional classification of volume data. In: Proceedings of IEEE/eurographics symposium on visualization. 2004. p. 17–24.

[12] Šereda P, Vilanova A, Gerritsen FA. Automating transfer function design for volume rendering using hierarchical clustering of material boundaries. In: Proceedings of IEEE/eurographics symposium on visualization. 2006. p. 243–50.

[13] Maciejewski R, Chen W, Woo I, Ebert DS. Structuring feature space—a non-parametric method for volumetric transfer function generation. IEEE Trans Visual Comput Graph 2009;15(6):1473–80.

[14] Röettger S, Bauer M, Stamminger M. Spatialized transfer functions. In: EUROGRAPHICS–IEEE VGTC symposium on visualization. 2005. p. 271–8.

[15] Nguyen BP, Tay W-L, Chui C-K, Ong S-H. A clustering-based system to automate transfer function design for medical image visualization. Vis Comput 2012;28(2):181–91.

[16] Wang Y, Chen W, Zhang J, Dong T, Shan G, Chi X. Efficient volume exploration using the gaussian mixture model. IEEE Trans Visual Comput Graph 2011;17(11):1560–73.

[17] Ip CY, Varshney A, JaJa J. Hierarchical exploration of volumes using multilevel segmentation of the intensity-gradient histograms. IEEE Trans Visual Comput Graph 2012;18:2355–63.

[18] Wang Y, Zhang J, Lehmann Dirk J, Theisel H, Chi X. Automating transfer function design with valley cell-based clustering of 2D density plots. In: Eurographics conference on visualization. 2012. p. 1295–304.

[19] Correa CD, Ma K-L. Visibility driven transfer functions. In: IEEE pacific visualization symposium. 2009. p. 177–84.

[20] Correa CD, Ma K-L. Visibility histograms and visibility-driven transfer functions. IEEE Trans Visual Comput Graph 2010;17(2):1077–2626.

[21] Ruiz M, Bardera A, Boada I, Viola I. Automatic transfer functions based on informational divergence. IEEE Trans Visual Comput Graph 2011;17(12):1932–41.

[22] Wang Y, Zhang J, Chen W, Zhang H, Chi X. Efficient opacity specification based on feature visibilities in direct volume rendering. Comput Graph Forum 2011;30(7):2117–26.

[23] Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 2000;22:888–905.

[24] Shannon CE. A mathematical theory of communication. Bell Syst Tech J 1948;27, 379–423, 623–656.

[25] Haidacher M, Bruckner S, Kanitsar A, Gröller ME. Information-based transfer functions for multimodal visualization. In: VCBM, Eurographics Association. 2008. p. 101–8.

[26] Bordoloi U, Shen H. View selection for volume rendering. In: Visualization, 2005. VIS 05. IEEE, 2005. 2005. p. 487–94.

[27] Wang C, Shen H. Information theory in scientific visualization. Entropy 2011;13(1):254–73.

[28] Hartley R, Zisserman A. Multiple view geometry in computer vision. Vol. 2. Cambridge; 2000.

[29] Cour T, Yu S, Shi J. Normalized cut segmentation code. Copyright 2004 University of Pennsylvania, Computer and Information Science Department; 2004.