

Intuitive Transfer Function Design for Photographic Volumes

Bin Zhang · Yubo Tao · Hai Lin · Feng Dong · Gordon Clapworthy

Received: date / Accepted: date

Abstract Photographic volumes have been increasingly used in medical and biological researches in recent years. The original colors kept in photographic volumes present great opportunities to capture a rich set of information within the dataset for a wide variety of data analysis and visualization applications. Despite years of research, an interactive and user-friendly transfer function is still lacking for photographic volume visualization. The difficulty lies in how to map colors to a space that is convenient and intuitive for users to interactively classify features, i.e. specifying opacities for voxels. In this paper, we propose a color-based transfer function for intuitive opacity specification of photographic volumes. The color-based transfer function intelligently maps the colors from 3D to 1D, resulting in 256 representative colors which preserve the original colors to the maximum extent. Users can directly classify voxels based on these representative colors similar to the conventional 1D transfer function. Experiments are performed to evaluate the effectiveness of the proposed method, and also demonstrate the intuitiveness and flexibility of the proposed method.

Keywords Photographic volume · Transfer function · Color mapping

1 Introduction

With the recent advance of cryosection techniques, photographic volumes have been increasingly used in medical and biological researches. Modern cryo-imaging systems have allowed us to capture ultra-high resolution realistic color images, for example, 55GB volume data of a whole mouse (Roy et al. 2009), the Visible Human Project (VHP) at the National Library of Medicine (Spitzer et al. 1996) and the Chinese Visible Human Project (CVHP) (Zhang et al. 2006). In contrast to the volumes captured by traditional instruments, such as CT, PET and MRI, photographic volumes preserve original colors of the subject. With the availability of photographic volumes, highly realistic volume visualization can be generated by sampling the colors in the volume without additional color transfer functions. However, specifying opacity for photographic volumes becomes harder due to 3D colors, instead of 1D scalar values in scalar volumes.

Different kinds of transfer functions are designed to simplify the tedious data classification and visual property mapping process. Generally, a graphic design interface is provided together with the distribution of various useful properties, and this is much more intuitive and flexible for users than just tuning several abstract and non-linear parameters. However, it is not easy to create an interactive opacity specification approach for photographic volumes due to the interaction difficulty of the 3D color space. Although several excellent transfer function design methods have been proposed for dealing with photographic volumes, they face the limited classification number or the lack of interactivity. Most of these are global opacity specification approaches, i.e. the opacities of all voxels can only be changed simultaneously (Ebert et al. 2002; Gargesha et al. 2009). Researchers

Bin Zhang · Yubo Tao · Hai Lin (✉)
State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
E-mail: ieasterman@live.com, taoyubo@cad.zju.edu.cn, lin@cad.zju.edu.cn

Feng Dong · Gordon Clapworthy
Centre for Computer Graphics & Visualisation, University of Bedfordshire, UK
E-mail: feng.dong@beds.ac.uk, gordon.clapworthy@beds.ac.uk

have also proposed an interactive method to show different features with different opacities (Takanashi et al. 2002). But the interaction is performed in a derived 3D space, which is not convenient and intuitive for general users due to its abstract meaning. Thus, an intuitive and easy-to-use interactive approach is still absent for photographic volumes.

We propose a color-based 1D opacity transfer function for photographic volumes. As the dimension goes lower, the degree of freedom and the complexity of user interaction decrease. 1D transfer function is easy to understand and convenient for user interaction. In order to develop a 1D transfer function for photographic volumes, the difficulty lies in how to create the axis of the transfer function. So we propose a color mapping approach to map colors from 3D to 1D while maximizing the color preservation. 256 representative colors, which approximate the original colors to their best, are selected to form a color bar that can serve as the axis of the proposed 1D transfer function. Similar to the conventional 1D transfer function for scalar volume visualization, users can directly classify voxels based on the color bar. By taking the original colors of features as reference, the representative colors make opacity manipulation much more intuitive.

2 Related Work

Transfer function design has drawn a lot of attention. Besides the commonly used 1D transfer function based on scalar values, a large number of meaningful transfer functions have been proposed for scalar volumes. Kindlmann et al. proposed a semi-automatic generation of both 1D and 2D transfer functions based on scalar value and gradient magnitude (Kindlmann and Durkin 1998; Pfister et al. 2001). Roettger et al. (2005) clustered the 2D histogram by considering the spatial connectivity of the histogram entries and created spatialized transfer function. Sereda et al. (2006) proposed the LH transfer function to detect feature boundaries based on a histogram generated by following the gradient directions. Correa and Ma (2008, 2011) applied size and visibility to classify features of interest. Ruiz et al. (2011) provided a method to generate automatic transfer functions based on information divergences. Guo et al. (2011) automatically computed transfer functions by analyzing the user interaction on the visualization result.

In photographic volume visualization, most of the researches have also been devoted to transfer function design. Ebert et al. proposed color distance gradient and used it in their transfer functions (Ebert et al. 2002). They set opacity to one of the color components, color distance gradient magnitude or color distance gradient dot product. The opacity setting is fixed once a transfer function is selected, so that users can hardly interact with it. Gargesha et al. (2009) combined color and gradient feature detectors to generate the opacity transfer function, providing a user interface for selecting different feature detectors. However, users have to figure out the composition of the RGB components in their mind according to features of interest, making the interaction not intuitive enough. Takanashi et al. introduced an interactive classification technique called ISpace by using Independent Component Analysis (ICA) to transform the original data into a new space (Takanashi et al. 2002). Users can classify data by clipping data histogram in the ICA space or by specifying several 1D transfer functions, one for each ICA axis. But interactive clipping in 3D space is not an easy task and the derived axes are not as intuitive as those of the original color space. So we create an axis with colors selected from the original data set and establish a color-based 1D transfer function.

Mapping colors from 3D to 1D is much like the decolorization problem. In general, decolorization can be performed either locally or globally. Local methods apply different mapping functions in different local regions, trying to preserve as much contrast as possible in the result grayscale image (Gooch et al. 2005). This is not suitable for our requirement, as different mapping functions would map the same colors from different local regions to different results. As for global methods, Lu et al. (2012) proposed a very fast yet effective decolorization approach. They reached a robust solution with a simple linear parametric grayscale model. Song et al. (2013) extended this work and proposed a multi-scale contrast preservation strategy, which chooses channel weights depending on specific images to maximally preserve the original color contrast. However, in our case, this kind of simple linear parametric grayscale model may result in an uneven histogram distribution. On the other word, the result 1D space may not be efficiently used due to the empty regions in the result histogram. So we propose a global non-linear color mapping method.

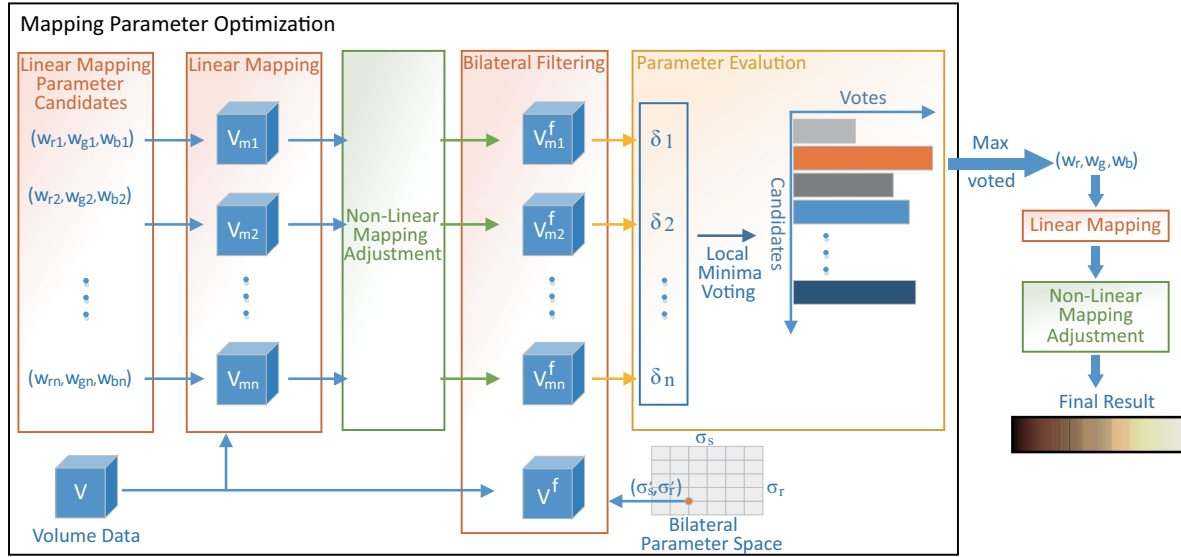


Fig. 1 The pipeline of our color mapping approach.

3 Color-Based Transfer Function

In order to develop an interactive transfer function for photographic volumes, it is necessary to map the voxel values to a space that is convenient for user interaction. The reduction in the dimension of interaction space can reduce the degree of freedom and the complexity of user interaction. So the 1D transfer function is easy to use and widely used in volume visualization. We try to map colors to 1D and pick out a sequence of representative colors that can serve as an axis of the 1D transfer function. With our color mapping method, same colors should map to the same value, ensuring not to separate voxels of a same feature. The representative color sequence should satisfy the color perceptual continuity. Gathering similar colors together can simplify the work spent on feature classification. But through a 3D-to-1D dimensional reduction, the information loss of the original data is unavoidable. We focus our work on how to adjust the color mapping strategy to reduce the information loss and achieve a better color preservation. Obviously, the color mapping problem is similar to decolorization in some degree. While a decolorization method maps colors to grayscale values and focuses on contrast preservation, we need our color mapping method to produce a sequence of representative colors which preserves the color of the original data to its best. The pipeline of our color mapping method is shown in Fig. 1 and each step will be described in detail in this section.

3.1 Non-linear Color Mapping

To construct a 1D transfer function, we need to select an ordinal property as the axis. For photographic volumes, color is one of the most intuitive attribute to serve as the axis. User can easily establish a connection between colors and features. But there are too many colors to be listed along the axis in the user interface. So we should map the colors to 1D first and then select some representative colors for the color-based transfer function.

Linear mapping preserves color consistency, which means same colors in different voxels are mapped to the same position in the 1D axis. So we choose to use the general linear composition of colors for color mapping, as shown in Equation 1,

$$\begin{aligned}
 C &= w_r * r + w_g * g + w_b * b, \\
 w_r + w_g + w_b &= 1, \\
 w_r \geq 0, w_g \geq 0, w_b \geq 0,
 \end{aligned} \tag{1}$$

where w_r , w_g and w_b are weights of the three components (r, g, b) of a color and how to select these weights will be discussed in the next subsection. With the linear mapping in Equation 1, each color that appears in the volume data can be mapped onto the 1D axis. After the mapping is performed for all colors, we divide the axis

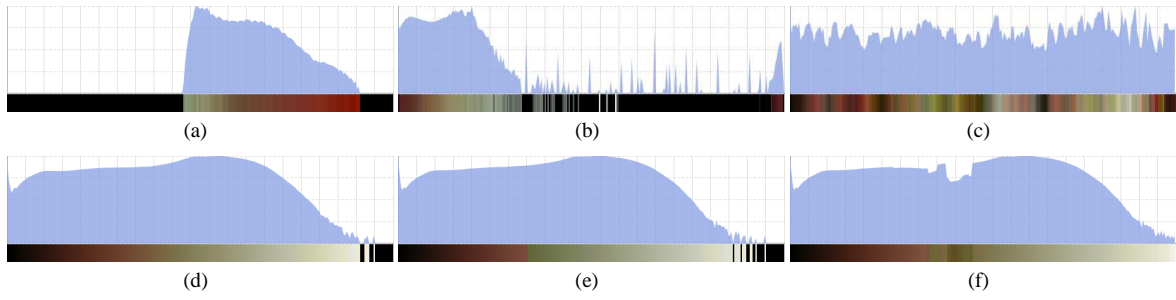


Fig. 2 Histograms generated by different color mapping methods. The associated color bar is painted at the bottom of each figure. (a) U component in CIELUV color space. (b) Hue component in HSV color space. (c) K-Means clustering. (d) Conventional rgb2gray. (e) The linear mapping with weights selected by the proposed method. (f) The proposed non-linear mapping method.

into 256 segments. In another word, the colors are separated into 256 sequential bins uniformly according to their C values. For each bin, one color is selected from all the colors in it and serves as the representative color of the bin. The representative color should minimize the color loss when it is used to substitute for others. The number of bins here is chosen to be 256, as the index of these representative colors can be represented by a byte (8 bits).

Obviously, this simple mapping may not achieve a satisfactory result. The colors in the photographic volume may be clustered in one region or several regions and do not cover the entire color space, so there may be some empty bins after the linear mapping process. The histograms generated by several different color mapping methods are shown in Fig. 2. Fig. 2(a), (b), (d) and (e) are results of linear mapping methods. By comparing these histograms, we can see that the histograms generated by linear methods have a large part of empty bins. Obviously the mapped 1D space is not effectively used. On the other hand, many quite different colors may be mapped to the same bin. Apparently, we should take out part of the quite different colors from one bin and put them into an empty bin. In this way, the bins that are still empty after the linear mapping can be made better use of and then more colors can be preserved. So after performing the linear mapping, we introduce a non-linear mapping adjustment strategy to further make use of empty bins.

Empty bins are removed from the bin sequence. For each non-empty bin, considering colors in it may be greatly different, the variance of the colors is calculated. It should be noted especially that, to calculate the variance of a color set, we need to measure the difference between colors. It is known that the RGB color space is not perceptually uniform but the CIELUV color space is thought to be perceptually uniform. It means that equal distances in the CIELUV color space correspond to equal perceptual differences. So the Euclidean distance in this color space can be used to measure the perceptual difference between colors. Let $\Delta E(c1, c2)$ denote the Euclidean distance between color $c1$ and color $c2$ in the CIELUV color space. Then the variance of the bin C_i can be calculated with

$$Variance_i = \frac{1}{n_i} \sum_{k=1}^{n_i} \Delta E(c_k, \mu), \quad (2)$$

where n_i is the number of the colors in C_i . The mean color μ is calculated by the arithmetic mean of all the color vectors in C_i . The variance of C_i is calculated as the average Euclidean distance from each color in C_i to μ , which can reflect the degree of variability of the colors in the bin.

The bin C_m with the largest variance is singled out to split as it has the maximum diversity of the colors. K-Means clustering is applied to the colors in C_m to partition them into two clusters. We separate the two clusters of colors into two new bins. Then for each of the two new bins, we find the bin which has the most similar representative color to that of the new bin in the bin sequence and insert the new bin to its left or right according to the representative color distances to both sides. This process is repeated until we get a sequence of 256 non-empty bins. Fig. 2(f) shows the result generated by performing the non-linear adjustment on Fig. 2(e). More colors are filled into the color bar and no empty bins remain. This makes a better color preservation. As to Fig. 2(c), which is generated by clustering colors into 256 clusters in the CIELUV color space with K-Means clustering, although it can also make full use of the mapped 1D space, colors selected by K-Means method cannot be easily sorted into a sequence as good as ours. The color bar generated by our non-linear mapping is more consecutive in visual perception and thus it can better facilitate the classification work.

3.2 Parameter Optimization

Having the color mapping approach, we focus on how to optimize it. The best strategy in our opinion is to select the most suitable parameters for the color mapping according to the processed data set, so that the color loss during the mapping process can be reduced as much as possible. In the proposed non-linear color mapping, the parameters are three weights w_r , w_g and w_b used for the linear combination in Equation 1. We can discretize w_r , w_g and w_b in the range of $[0,1]$ and select a best combination of the three weights. Song et al. (2013) found that the results would not change too much when slightly varying the weights w_r , w_g and w_b . It is enough to set the discretization interval to 0.1. When the three parameters are discretized in the range $[0, 1]$ with the interval 0.1, due to the constraint in Equation 1, we actually get 66 parameter candidates that are uniformly sampled from a plane in the w_r - w_g - w_b space. Finer discretization is still considerable when accurate results are needed.

In order to compare different candidate combinations of the three parameters, we need to find a way to evaluate a specified group of parameters. Given a group of parameters, each color can be mapped to one of the 256 representative colors that are selected by the non-linear color mapping method mentioned above. A new color volume data can be established by mapping the color of each voxel in the original volume data to its corresponding representative color. Suppose that the input volume data is \mathbf{V} and the mapped volume data is \mathbf{V}_m , the difference between them can be obtained by a voxel wise comparison. Taking the multi-scale color-preservation into consideration, we introduce the joint bilateral filtering (Petschnigg et al. 2004), which is also known as the cross bilateral filter (Eisemann and Durand 2004). By varying the spatial parameter and the range parameter used in bilateral filter, we can evaluate the color mapping parameters in different scales.

The joint bilateral filtering can be defined as follows. Let $\mathbf{V}(\mathbf{p})$ be the color of the voxel at position \mathbf{p} , $\mathbf{c}_f(\mathbf{p})$ be the filtered color and \mathbf{V}_g be the volume data used to compute weights in the range kernel, which is also mentioned as guidance volume data in the following, then we have,

$$\begin{aligned} \omega(\mathbf{p}, \mathbf{q}) &= G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)G_{\sigma_r}(\Delta E(\mathbf{V}_g(\mathbf{p}), \mathbf{V}_g(\mathbf{q}))), \\ \mathbf{c}_f(\mathbf{p}) &= \frac{\sum_{\mathbf{q} \in \Omega_{\mathbf{p}}} \omega(\mathbf{p}, \mathbf{q}) \mathbf{V}(\mathbf{q})}{\sum_{\mathbf{q} \in \Omega_{\mathbf{p}}} \omega(\mathbf{p}, \mathbf{q})}, \end{aligned} \quad (3)$$

where \mathbf{q} is a position in the neighborhood $\Omega_{\mathbf{p}}$ of \mathbf{p} , G_{σ_s} is the spatial filter kernel for measuring the spatial similarity and G_{σ_r} is the range filter kernel measuring the color similarity. The bilateral filtering is applied to the original volume \mathbf{V} and generates the filtered volume \mathbf{V}^f . Meanwhile, the filter is applied to \mathbf{V} with the mapped volume \mathbf{V}_m as the guidance, and generates the filtered volume \mathbf{V}_m^f . By summing up the color difference between the colors of each pair of corresponding voxels at \mathbf{p} , we can get the matching cost between the two filtered volumes,

$$\delta = \sum_{\mathbf{p} \in \mathbf{V}} \Delta E(\mathbf{V}^f(\mathbf{p}), \mathbf{V}_m^f(\mathbf{p})). \quad (4)$$

The matching cost stands for the color preservation quality. It can be used as a metric of the validation of a color mapping method. By quantizing the parameter space of bilateral filtering, which involves σ_s and σ_r , we can measure the color preservation in different spatial and range scales. According to Song et al. (2013), we quantize the parameters σ_s as $[0.1, 0.2, 0.3, \dots, 1.0]$ and σ_r as $[0.1, 0.5, 1.0, 2.0]$. For each pair of (σ_s, σ_r) , matching costs are calculated for the 66 parameter candidates, and the candidate with the local minimum value in its parameter space is voted. After all pairs of (σ_s, σ_r) are traversed, the parameter candidate that has the most votes is picked out as the selected parameter for the final color mapping.

3.3 Color-Based Transfer Function

Once the color mapping has been completed, we can construct a graphic interface which is similar to the widely used scalar based 1D transfer function widget. The representative colors produced by the non-linear color mapping is listed in the horizontal axis and the vertical axis stands for the opacity value. For a better perception of each value listed in the horizontal axis, we draw a color bar using the representative colors at the corresponding location. Because the colors lying in the horizontal axis is related to the colors in the visualization result, with the present of the color bar, it becomes more intuitive for user interaction. Users do not need to think much about the correspondence between the structures in the result image and the feature regions in the transfer function. All they need to do is finding the features of interest in the visualization result and adjust

the opacity curve of the regions with similar colors in our transfer function. A human chest rendered with the proposed color-based transfer function is shown in Fig. 3. The associated transfer functions are listed below the visualization results. In this data set, the colors of muscle tend to be dark and red while the colors of humerus and part of the ribs seem brighter. Fig. 3(a) gives a hybrid visualization of these tissues. By decreasing the opacities of the dark and red areas, which are used to visualize muscles in Fig. 3(c), the humerus and part of the ribs can be recognized with the help of our transfer function as shown in Fig. 3(b).

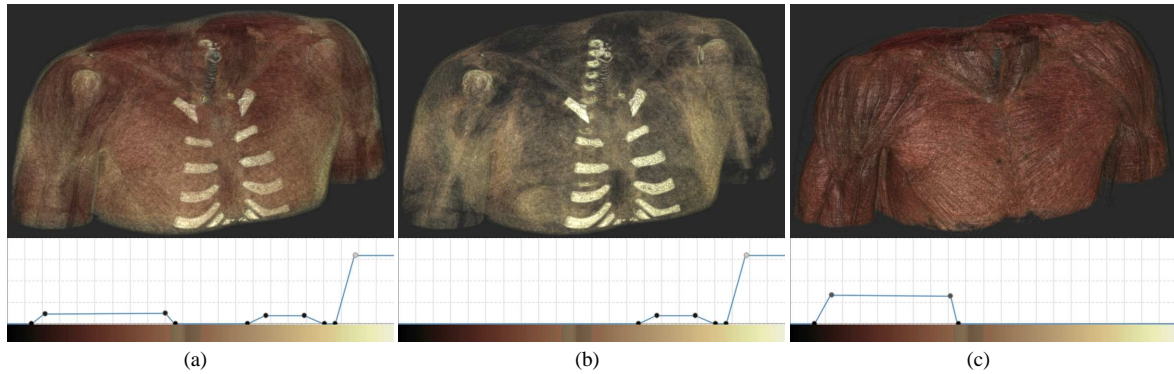


Fig. 3 Human chest rendered with our color-based transfer function.

4 Result

Experiments are performed to evaluate the proposed color mapping method. Given a color mapping method, we generate a mapped volume by replacing each voxel with its mapped color value. Then the average matching cost is calculated by per-voxel averaging the sum of the matching costs in different scales,

$$\bar{\delta} = \sum_{\sigma_s} \sum_{\sigma_r} \delta(\sigma_s, \sigma_r) / N, \quad (5)$$

where N is the number of voxels.

By measuring the average matching cost, we can evaluate the variation caused by color mapping. The lower the average matching cost is, the better the mapping method preserves the original colors. In our experiments, the average matching cost is calculated with 7 color mapping methods on 4 data sets. The results are shown in Table 1. The CIELUV(U) method is performed by mapping colors to 1D according to their U components in CIELUV color space. Similarly, the HSV(H) method maps colors according to their hue values in HSV color space. The rgb2gray method applies the linear mapping used in Matlab. The K-Means method is performed by clustering the colors into 256 clusters in the CIELUV color space. We also conduct a group of experiments by adding the proposed non-linear adjustment to rgb2gray (noted with +NL in Table 1) and a group of experiments by removing the non-linear adjustment from the proposed method (noted with -NL in Table 1). By doing this we can see the validity of the mapping parameters selected by our method. As we can see from Table 1, for each of the four data sets, the average matching cost of the proposed method is minimum among these mapping

Table 1 The Average Matching Costs of Different Mapping Methods

Method	Mouse	Chest	Leg	Head
CIELUV(U)	0.0912675	0.1263410	0.0808967	0.0485085
HSV(H)	0.0849311	0.0759441	0.0666765	0.0316832
rgb2gray	0.0131045	0.0128178	0.0129036	0.0059713
rgb2gray(+NL)	0.0134504	0.0125809	0.0128129	0.0059282
K-Means	0.0518242	0.0391509	0.0377042	0.0170200
Proposed method(-NL)	0.0114841	0.0117594	0.0106860	0.0095955
Proposed method	0.0107069	0.0109380	0.0101340	0.0050037

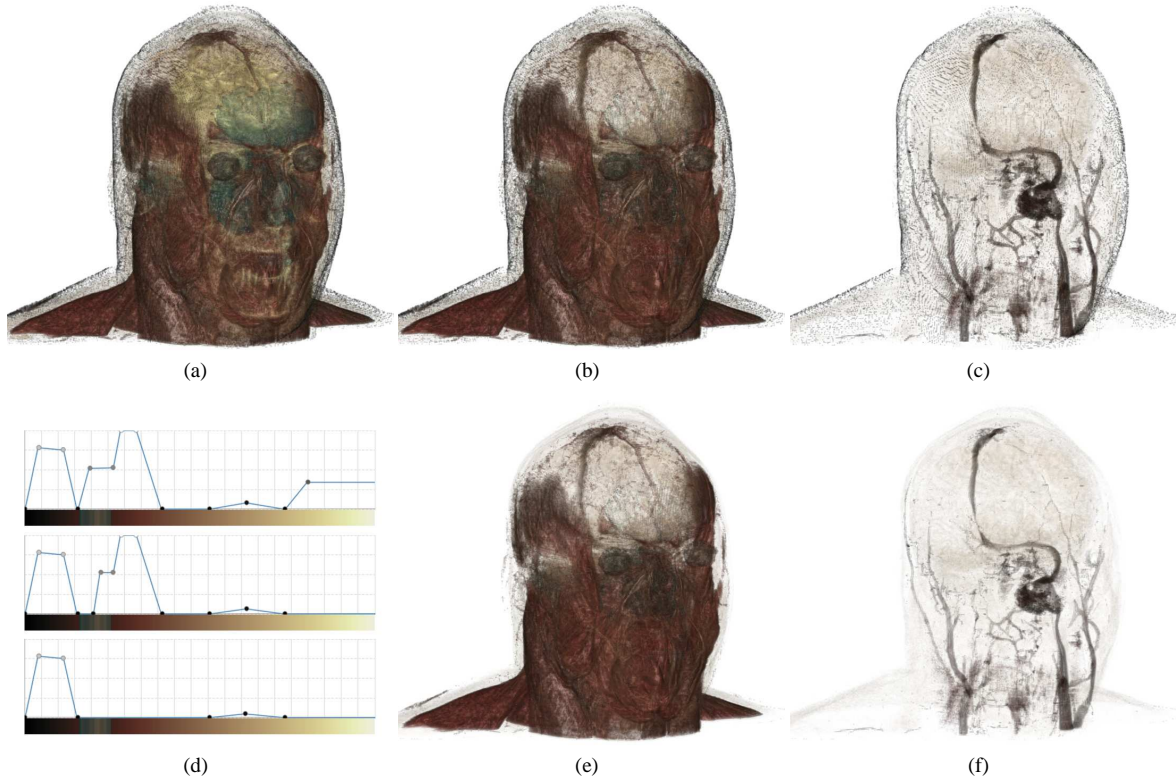


Fig. 4 Visualization results of human head. (a), (b) and (c) are rendered with the three color-based transfer functions in (d) respectively. (e) and (f) are rendered by combining the bottom two color-based transfer functions with a simple gradient-magnitude-based transfer function.

methods. It means that the color loss caused by the color mapping process is minimized with our method. So the representative colors that we choose can represent the original data values better. On the other hand, the color bar generated by the proposed method is continuous in color perception, as shown in Fig. 2(f). This ensures that similar colors will not be divided to disperse areas and improves the efficiency of opacity specification, making the proposed transfer function more user-friendly.

Fig. 4 shows a group of visualization results of the human head data set. Different features are blended with different opacities in Fig. 4(a). We can see the brain tissue, the teeth, the sinus, the muscles and part of the skin. The color-based transfer function can reduce the blindness during data exploration. By decreasing the opacities of the dark green area and the light yellow area, we can remove the sinus, the brain and the teeth, getting the result in Fig. 4(b). Further removing the red area, only the blood vessels and part of the skin, whose colors are close to black, can be seen. Due to the data acquisition process, skins cover a large range of colors. It cannot be easily removed with the color-based transfer function only. Fortunately, the proposed transfer function can be combined with other transfer functions. By applying the color-based transfer functions used in Fig. 4(b) and Fig. 4(c) in multiplicative combination with a transfer function based on gradient magnitude, which simply sets the opacity of high gradient magnitude area to zero, we can remove the skin and get the results in Fig. 4(e) and (f). As a 1D transfer function, the proposed method is easy to understand and convenient for user interaction. With the help of the color bar, feature classification becomes much more intuitive due to the correspondence between color and feature.

We make a compare between the visualization results rendered with the proposed color-based transfer functions and those rendered with the method of Ebert et al. (2002). The opacity transfer function used in their method is $renderedopacity = (voxel_opacity * scalar)^{exponent}$, where $scalar$ is a coefficient to control the overall opaqueness of the volume and is set to a constant value in our experiments. In Fig. 5(a) and (b), $voxel_opacity$ is set to color distance gradient magnitude. In Fig. 5(a), when the exponent is set to 0.6, different features can be well blended with the opacity determined by the color distance gradient. But when the exponent increases to 0.7, the opacities of all features decreases simultaneously, making the result image pale and dim, as shown in Fig. 5(b). The transfer functions they proposed are not able to highlight a specified feature. With the help of our color-based transfer function, we can get a result that is almost the same as theirs. At the same time,

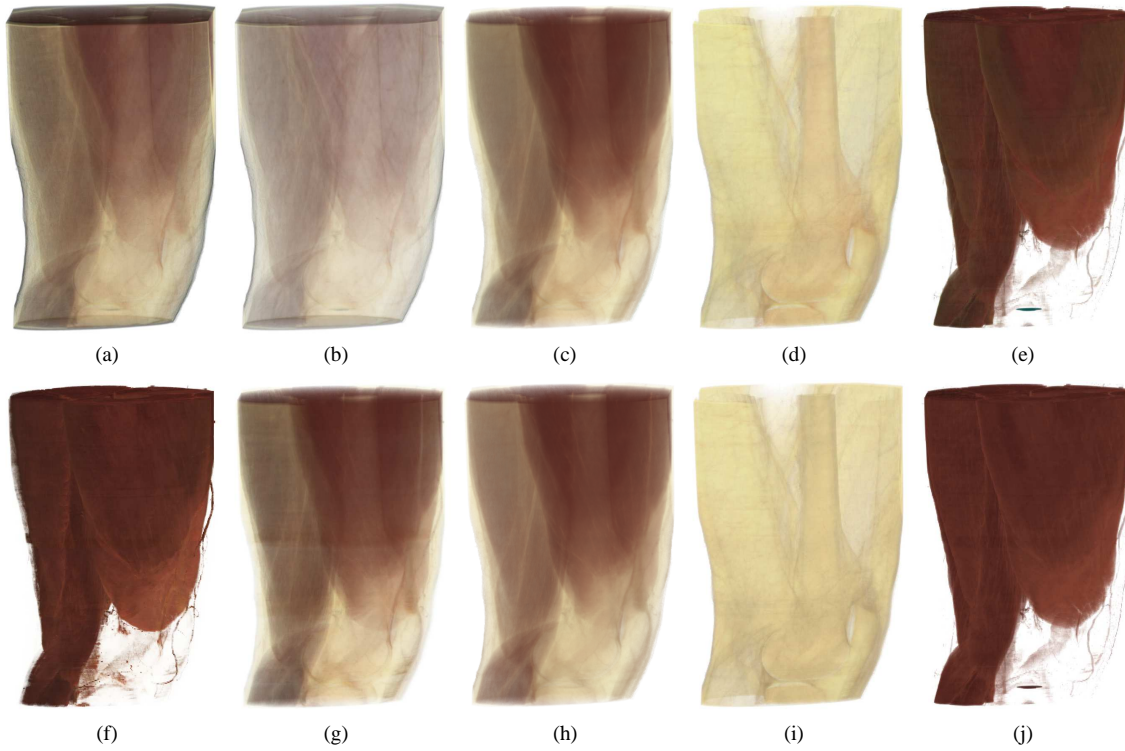


Fig. 5 Volume rendering of the human leg data set. Opacities in (a) and (b) are determined by the transfer function based on the color distance gradient, with the exponent equal to 0.6 and 0.7 respectively. Opacities in (c), (d) and (e) are specified by the proposed color-based transfer function. (f) and (g) are rendered with transfer functions based on color and gradient feature detectors. (h), (i) and (j) are rendered with representative colors instead of original colors with the proposed transfer function.

different features can be separately rendered by interactively setting a large opacity value to the corresponding area and decreasing the opacity of the rest. For example, once we want to see the muscle tissue, the region in red in the color-based transfer function is selected and set to a large opacity value, as shown in Fig. 5(e). By removing the muscle tissue from Fig. 5(c), we can get the volume rendering of the fat and the bone whose colors are close to light yellow, as demonstrated in Fig. 5(d). As for the method based on color and gradient feature detectors proposed by Gargasha et al. (2009), similar results can be got with careful parameter tuning, as shown in Fig. 5(f) and (g). But the color feature detector in their method is not as flexible as the proposed transfer function. For example, although fat and bone tissues can be detected with the yellow color detector $1.0 * R + 1.0 * G + 0.0 * B$, features in red are also detected by this detector, as shown in Fig. 5(g). Result such as Fig. 5(d) cannot be easily got with their method. The proposed color-based transfer function surpasses previous methods in flexibility and ease of use. On the other hand, in our implementation, mapped volume data can also be visualized with the proposed transfer function, as shown in Fig. 5(h), (i) and (j). Although the results are closed to those of the original data in Fig. 5(c), (d) and (e), by comparing to Fig. 5(e) we can see that some detail colors are missing in Fig. 5(j). However, the size of mapped volume data is only one third of the original data. Therefore we can use mapped volume to get a quick overview when memory is not enough, but not for serious situations.

Another group of comparison is performed on the digimouse data set (Dogdas et al. 2007). The results are shown in Fig. 6. The results in the top two rows are rendered with the method of Ebert et al. (2002) and the results in the last row is created with the proposed method. Fig. 6(a), (b) and (c) set *voxel_opacity* to color distance gradient magnitude. Fig. 6(d), (e) and (f) set *voxel_opacity* to the value of U component in CIELUV color space. When increasing the exponent, features in the results of both of these two transfer functions fade out gradually. Although meaningful results can be got, the opacity of a specified feature cannot be solely set due to the global transfer function. Thus different features cannot be separated effectively. The proposed color-based transfer function can get a result that is similar to theirs, such as Fig. 6(g). Furthermore, we can distinguish features that cannot be easily selected with their method, such as the organs in Fig. 6(h) and the heart and the blood vessels in Fig. 6(i). In the last two results, we set a low opacity to the mouse body designed to make

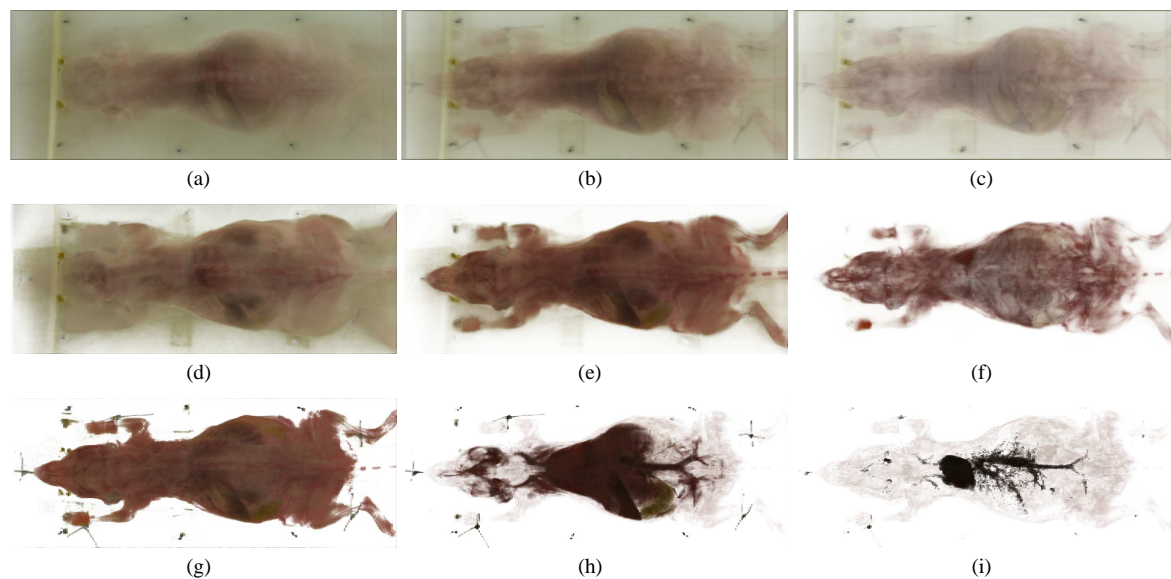


Fig. 6 Volume rendering of the digimouse data set. Opacities in (a), (b) and (c) are determined by the transfer function based on the color distance gradient, with the exponents equal to 0.5, 0.6 and 0.7 respectively. Opacities in (d), (e) and (f) are determined by the transfer function based on CIELUV U color component. Exponents equal to 0.5, 1.0 and 1.5 respectively. Opacities in (g), (h) and (i) are set with the proposed color-based transfer function.

it available as background. Our method is flexible enough to set different opacities to different features intentionally, making it more useful in data exploration. Aided by the color bar user can classify features according to their original colors. This makes the exploration process much more intuitive and reduces the work spend on trial-and-error.

5 Conclusion

In this paper, we have proposed a novel transfer function design method for photographic volumes. An intuitive color-based 1D transfer function is developed based on a non-linear color mapping strategy. This transfer function can be used alone or can be combined with other transfer functions to generate desired visualization results. Experiment results show the effectiveness of our transfer function. The main limitation of our proposed method is that the parameter optimization costs a lot of time. Most of the time is consumed by the bilateral filtering. For example, while the bilateral filtering was implemented with OpenCL, it took almost 3 hours to perform the entire color mapping for the head data set on a computer with Intel i5-3450 CPU and NVIDIA GTX 660 GPU. Fortunately, for a given volume data, the nonlinear color mapping only need to be performed once in the preprocessing stage, and then we can store the result for later use. To deal with the color loss caused by color mapping, we can change the number of representative colors. The more representative colors we use, the more accurately they represent the original data. But for a synthesized volume dataset which contains a great many of colors, more representative colors are not helpful. The color bar generated will be less smooth and thus it will be less friendly for user interaction. Another problem is that color itself may not be a sufficiently distinct attribute to distinguish between features. To further enhance the classification ability of our transfer function, we will attempt to extend it with other attributes such as texture properties (Caban and Rheingans 2008). We hope more intuitive and user-friendly transfer functions would be developed for photographic volumes in future.

Acknowledgements This work was partially supported by 863 Program Project 2012AA12A404, National Natural Science Foundation of China No. 61472354 and the Fundamental Research Funds for the Central Universities (2013QNA5010).

References

Caban JJ, Rheingans P (2008) Texture-based transfer functions for direct volume rendering. *Visualization and Computer Graphics*, IEEE Transactions on 14(6):1364–1371

- Correa C, Ma KL (2008) Size-based transfer functions: A new volume exploration technique. *Visualization and Computer Graphics, IEEE Transactions on* 14(6):1380–1387
- Correa C, Ma KL (2011) Visibility histograms and visibility-driven transfer functions. *Visualization and Computer Graphics, IEEE Transactions on* 17(2):192–204
- Dogdas B, Stout D, Chatziioannou AF, Leahy RM (2007) Digimouse: a 3d whole body mouse atlas from ct and cryosection data. *Physics in medicine and biology* 52(3):577
- Ebert DS, Morris CJ, Rheingans P, Yoo TS (2002) Designing effective transfer functions for volume rendering from photographic volumes. *Visualization and Computer Graphics, IEEE Transactions on* 8(2):183–197
- Eisemann E, Durand F (2004) Flash photography enhancement via intrinsic relighting. *ACM transactions on graphics (TOG)* 23(3):673–678
- Gargsha M, Qutaish M, Roy D, Steyer G, Bartsch H, Wilson DL (2009) Enhanced volume rendering techniques for high-resolution color cryo-imaging data. In: *SPIE Medical Imaging, International Society for Optics and Photonics*, pp 72,622V–72,622V
- Gooch AA, Olsen SC, Tumblin J, Gooch B (2005) Color2gray: salience-preserving color removal. *ACM Transactions on Graphics (TOG)* 24(3):634–639
- Guo H, Mao N, Yuan X (2011) Wysiwyg (what you see is what you get) volume visualization. *Visualization and Computer Graphics, IEEE Transactions on* 17(12):2106–2114
- Kindlmann G, Durkin JW (1998) Semi-automatic generation of transfer functions for direct volume rendering. In: *Proceedings of the 1998 IEEE symposium on Volume visualization*, ACM, pp 79–86
- Lu C, Xu L, Jia J (2012) Real-time contrast preserving decolorization. In: *SIGGRAPH Asia 2012 Technical Briefs*, ACM, New York, NY, USA, SA '12, pp 34:1–34:4
- Petschnigg G, Szeliski R, Agrawala M, Cohen M, Hoppe H, Toyama K (2004) Digital photography with flash and no-flash image pairs. *ACM transactions on graphics (TOG)* 23(3):664–672
- Pfister H, Lorensen B, Bajaj C, Kindlmann G, Schroeder W, Avila LS, Raghu K, Machiraju R, Lee J (2001) The transfer function bake-off. *Computer Graphics and Applications, IEEE* 21(3):16–22
- Roettger S, Bauer M, Stamminger M (2005) Spatialized transfer functions. In: *Proceedings of the 7th Joint Eurographics/IEEE VGTC conference on Visualization*, pp 271–278
- Roy D, Steyer GJ, Gargsha M, Stone ME, Wilson DL (2009) 3d cryo-imaging: A very high-resolution view of the whole mouse. *The anatomical record* 292(3):342–351
- Ruiz M, Bardera A, Boada I, Viola I, Feixas M, Sbert M (2011) Automatic transfer functions based on informational divergence. *Visualization and Computer Graphics, IEEE Transactions on* 17(12):1932–1941
- Sereda P, Bartoli AV, Serlie IW, Gerritsen FA (2006) Visualization of boundaries in volumetric data sets using lh histograms. *Visualization and Computer Graphics, IEEE Transactions on* 12(2):208–218
- Song Y, Bao L, Xu X, Yang Q (2013) Decolorization: Is rgb2gray() out? In: *SIGGRAPH Asia 2013 Technical Briefs*, ACM, New York, NY, USA, SA '13, pp 15:1–15:4
- Spitzer V, Ackerman MJ, Scherzinger AL, Whitlock D (1996) The visible human male: a technical report. *Journal of the American Medical Informatics Association* 3(2):118–130
- Takanashi I, Lum EB, Ma KL, Muraki S (2002) Ispace: Interactive volume data classification techniques using independent component analysis. In: *Proceedings of Pacific Graphics 2002 Conference*, pp 366–374
- Zhang SX, Heng PA, Liu ZJ (2006) Chinese visible human project. *Clinical Anatomy* 19(3):204–215