International Journal of Image and Graphics
Vol. 5, No. 4 (2005) 699–714

CONTROL OF OBJECT VISIBILITY IN VOLUME RENDERING — A DISTANCE-BASED APPROACH

JIANLONG ZHOU*, ANDREAS DÖRING† and KLAUS D. TÖNNIES‡

Computer Vision Group, Department of Computer Science
Otto-von-Guericke University Magdeburg
Magdeburg 39106, Germany
**Zhou@isg.cs.uni-magdeburg.de*
†Doering-andreas@web.de
‡Klaus@isg.cs.uni-magdeburg.de

Received 15 May 2004
Revised 19 November 2004
Accepted 27 May 2005

Volume data often have redundant information for clinical uses. The essence of volume rendering can be regarded as a mechanism to determine visibility of redundant information and structures of interest using different approaches. Controlling the visibility of these structures in volume rendering depends on the following factors in existing rendering algorithms: The data value of current voxel and its derivatives (used in transfer function based approaches), and the voxel position (used in volume clipping). This paper introduces the distance which is defined by the user into volume rendering pipeline to control the visibility of structures. The distance based approach, which is named as *distance transfer function*, has the flexibility of transfer functions for depicting data information and the advantages of volume clippings for visualizing inner structures. The results show that the distance based approach is a powerful tool for volume data information depiction.

Keywords: Volume rendering; distance; transfer function; object visibility.

1. Introduction

In volume rendering, we seek to explore the volumetric data using visuals. This exploration process aims to discover and emphasize structures of interest embedded in the data, while de-emphasizing or completely removing away occluding structures that are currently not of interest. Medical volume data often have redundant information. The main problem of volume rendering is to determine visibility of different structures using different approaches. Usually there are two methods to remove the redundant information in volume rendering: *Volume clipping* and *transfer function*.

Volume clipping planes and more general clipping primitives[1,2] provide geometric tools to remove or displace occluding structures in their entirety. They play an important role in understanding volume data because they allow us to cut away selected parts of the data set based on information of the position of voxels in volume. This geometric approach can be regarded as a complementary to the specification of the transfer function, which is based on the data value and its derivatives.[2] Transfer functions can alter the overall look-and-feel of the data set in a continuous fashion.[3] In this process, the redundant information is removed through setting its corresponding opacity to zero. Good transfer functions can depict and reveal the important structures in the data with some degrees of details. But there is not a standard for transfer function specification although much work has been devoted on this topic.[4,5] The shortcoming for the transfer function based method is that objects of interest often have similar scalar values with their surrounding objects. This results in that objects of interest and their surrounding objects with similar scalar values will be rendered in the scene at the same time. These surrounding objects disturb the understandings of objects of interest. The traditional transfer function based method cannot remove this kind of objects. On the other hand, medical data sets often contain a complex combination of structures of different tissues. Features of interest in volume data are visualized depending on their voxel positions. This can be frustrating if the user is interested in isolating one feature of the volume which is spatially localized in terms of voxel positions from the other regions. In summary, the essence of controlling the visibility of objects of interest in volume rendering depends on following factors in existing rendering algorithms: the data value of current voxel and its derivatives, and the voxel position.

This paper aims to demonstrate the importance and power of voxel position dependent transfer function, which is used to control the visibility of structures in volume rendering. In our previous work,[6] we demonstrated that the distance information can be used to control the object enhancement in the context region in focal region based volume rendering.[7] We also presented a method to combine distance into rendering pipeline. The main contributions of this paper are as follows:

- To introduce voxel position information (distance) into transfer function domain and propose a concept of *distance transfer function*;
- To demonstrate the ability of distance based approach for controlling the visibility of structures of interest;
- To provide different distance definition cases in the proposed approach;
- To show its important application fields in volume rendering.

One of the most important contributions of this paper is to provide comprehensive understanding of segmented objects through presenting different surrounding structures of segmented objects based on distance transfer function. The methods presented here allow effectively exploration of different features of the volume data set and allow the user to adjust the complexity of the output by modulating distance parameters. We start from the premise that a user does not need the whole data displayed in details to undertake a particular task.

The paper is organized as follows. Section 2 discusses previous work. Section 3 presents the outlines and principles of the distance transfer function. Section 4 presents two cases for distance calculations. In Sec. 5, we present the applications of the proposed approach. Sec. 6 gives the results and discussions of presented approach before the paper concludes with a note on future work.

2. Related Work

Previous work on controlling the visibility of structures in direct volume rendering is mainly focused on volume clipping and transfer function based approach. Weiskopf *et al.*[2,8] described a method that is capable of using complex geometries for volume clipping. Multiple and arbitrary clipping geometries can be set through stencil test in OpenGL.[1,9] Roettger *et al.*[10] combined pre-integration[11] with volume clipping to get more accurate volume clipping.

The transfer function is a critical component for the volume rendering process that specifies the relation between scalar value, as well as derivatives, and optical properties. Finding good transfer functions has been listed among the top ten problems in volume visualization.[12] Consequently, much effort has been spent on improving this situation.[5,13] Existing schemes range from fully manual to semi-automatic techniques for finding transfer functions. These transfer function approaches use volume scalar values and their derivatives as the basic parameters to design transfer function specification methods. The differences between these methods are that they create different widgets and use these parameters in a different way (data-centric or image-centric).

Tzeng *et al.*[14] described an interface for specifying the transfer functions that takes into account additional data properties such as texture and position besides the scalar value and its derivatives. In this approach, the user first paints a region he wants to see with one color and another region he does not want to see with another color in a few slices, then the painted regions are used as the initial information for an artificial neural network to classify the volume data. This technique can be regarded as a region growing based segmentation method[15] used in volume rendering. The painted regions are the seeds needed in region growing for segmentation. One of the limitations of this method is that the user must see the object he wants to render in 2D slices. The user must know in advance which object he wants to see in the 2D slices. If the object of interest does not appear in the 2D slices, it is not possible to separate and render it. Another limitation is that only the painted objects in the slices are rendered in 3D rendition. If multiple objects need to be rendered at the same time, the user must paint each object separately using different colors in the 2D slices. In some cases, it is not possible to precisely locate several objects in the slices at the same time. For example, it is often difficult to locate the vessels in the 2D slices.

Intensity depth-cuing is a well known technique for enhancing the perception of depth in a scene.[16] Rheingans *et al.*[17] adapted this technique for volume rendering, dimming volume sample colors as they recede from the viewer. Lu *et al.*[18] introduced a method to use distance as one of the factors to create the nonphotorealistic volume stippling.

Our approach differs from these prior methods in that it uses distance in rendering in the sense that previous techniques considering depth or distance information mainly focused on creating some kinds of NPR effects (e.g. halo, volume stippling effects) which have been widely developed in computer graphics. In our approach, we pay more attention to the power of distance for analyzing volume data to provide additional information — the importance and visibility of the current voxel. Different from Tzeng *et al.*'s work, our approach uses the voxel position information to create a distance field, and then uses this distance field to control the visibility of structures.

3. Distance Transfer Function

3.1. *Outlines*

The location of different volume elements within the overall volume is challenging. It plays an important role in information extraction for depicting the importance of current volume samples. In order to integrate the voxel position information into the rendering pipeline, a criterion needs to be set up to measure the effects of voxel positions on the rendering. The relative position information of a voxel can show the importance for a specific organ. In this paper, we introduce the distance, which depicts the voxel position information, into rendering pipeline. Distance could be a quite general parameter of space and it can be defined depending on the specific applications (e.g. the distance can be defined between current voxel position and an object boundary). We can control what to show in volume rendering depending on the distance. The emphasis of this approach is focused on how to enhance rendering and control the visibility of structures using distance (voxel position information), but not on how to generate good transfer functions as other traditional transfer function approaches do.[4,5]

In this paper the distance between the current voxel position and a specific point or object specified by the user is introduced into the volume rendering pipeline in order to depict the importance of the current volume sample for the whole rendering. It is an important factor that helps users to understand the relationships between elements within the volume. This is often neglected in the previous rendering approaches. Distance based viewing corresponds to the real human vision. In order to integrate the distance into rendering pipeline, we use the distance d_i to modulate the optical properties of each volume sample position and give different patterns for the modulation. Similar to the traditional transfer function, a mapping function between distance and optical properties can be set up to reveal the distance influence on the rendering. We name this mapping function as *distance transfer function* (*DTF*). This is similar to traditional transfer functions. Although analytic continuous functions are thinkable, in practice the distance transfer function is realized as a lookup table of fixed size. Distance transfer function is interesting because it addresses the problem of integrating voxel position information into the rendering pipeline.

3.2. *Principles*

To use distance transfer function in a rendering process, the user should first define the distance and compute the distance for each voxel position. In this way, we get the distance volume in which the data value for each voxel position is the distance of corresponding voxels in original volume data. After getting the distance volume, a mapping function between optical properties and distance d_i can be set up to reveal the contribution of the distance d_i to the final rendering result. In practice, the user assigns optical properties (opacities and colors) to voxel position information (i.e. distance d_i) in an arbitrary way through a widget. This assignment process can be described by means of a transfer function:

$$O_{distance} = T_{distance}\,(d_i), \tag{1}$$

where $O_{distance}$ is the optical properties (include opacities and colors) contributed by distance information, $T_{distance}$ is the transfer function that is used to map distance d_i to optical properties.

To combine the optical information to which the original volume data values are mapped using transfer functions and distance based volume optical information together, the following function should be used:

$$O_{final} = f(O_{distance}, O_{original}). \tag{2}$$

One of the typical combination function is:

$$O_{final} = O_{distance} \cdot O_{original}, \tag{3}$$

where O_{final} is the final optical properties of the volume and $O_{original}$ is the optical properties of the original volume data to which the data values are mapped using traditional transfer functions.

Another method to combine the distance into rendering pipeline is to modulate original opacity with distance coefficients and can be realized from the following equation:

$$\alpha_d = \alpha_0 \left(k_{do} + k_{de} \left(1 - \frac{d_i}{d} \right)^{k_{dn}} \right), \tag{4}$$

where α_d is the distance enhanced opacity, α_0 is the original opacity, k_{do} is used to control the contribution of original opacity for the final result, k_{de} controls the amount of distance enhancement, k_{dn} allows the user to adjust the slope of the opacity curve to best enhance the data set, d_i is the current distance value and d is the largest distance of all distance values.

Using distance to modulate opacity of the volume data gives the user another means to render structures of interest. We name this *distance transparency* for its distance coefficient. This distinguishes it from scalar value based transparency. Using this distance transparency, we can emphasize and de-emphasize different regions based on distance. When distance transparency is set to 1.0, we can get a volume clipping effect. When distance transparency is set to zero, the distance loses its influence on the final rendering result.

Furthermore, we can generalize the distance to be an anisotropic factor to be used in the rendering pipeline, e.g. to generalize it to be arbitrarily oriented quadrics.

4. Two Distance Definition Cases

The distance definition is the core for distance transfer function specifications. Different distance definitions can create different results and show the effectiveness of DTF for information depiction. In this paper, two cases of the distance definition are provided:

- Distance between the current voxel position and a user predefined point. We name this distance the *point-to-point distance*. The user predefined point determines what the user will see in the final result. The focal region center in focal region based volume rendering is a good example for this case.[7]
- Distance between the current voxel position and the nearest object boundary. We name this distance as *point-to-object distance*. Based on this distance, the user has the ability to enhance the structures adjacent to the specified object using DTF.

4.1. *Point-to-point distance*

In this kind of distance definition, a user predefined point should be given at first. This point can be specified interactively through a user interface. Then the distance d_i between the predefined point and the current sample position can be calculated from Eq. (5). The distances for every volume sample positions will be pre-computed and used in the rendering pipeline.

$$d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2}, \tag{5}$$

where (x_i, y_i, z_i) is the current volume sample position and (x_c, y_c, z_c) is the position of user predefined point. Other distance metrics can be used for the distance computation, e.g. city block and chess board. The speed of these distance metrics may be better than Euclidean metrics, but Euclidean metrics provides more precise results. During interaction with the volume, when the position of the predefined point is changed, d_i has to be recomputed and then re-put it into the rendering pipeline. In this paper, we use the user predefined point as the focal region center in focal region based volume rendering to enhance volume information and control the visibility of structures in focal region based volume rendering.[7] In this approach, the voxel position plays an important role in context region rendering. The volume data near the focal region should be rendered with detailed information and the voxels far away from focal region should be rendered with less details and faded out. Voxel positions are considered during this rendering pass to control the visibility of structures in context region.

4.2. *Point-to-object distance*

Point-to-object distance defines the distance between the current voxel position and the nearest object boundary. This distance computation process can be seen as a distance transform[19] for volume data. The distance transform is an operator normally only applied to binary images. The result of the distance transform is a graylevel image that looks similar to the input image except that the graylevel intensity of points inside foreground regions are changed to show the distance to the closest boundary from each point. To get the distance transform for volume data, we first segment structures of interest explicitly and then perform the distance transform for the segmented data set.

In order to show the relations between distance transform and the segmented data set, Fig. 1 shows an example of distance transform for segmented MR brain data. The segmented MS lesions are shown in the first image, and in the second image the distance transform is computed. We combine segmented objects into distance transform and get the third image to show the relations between original objects and the distance transform.

4.3. *Texture mapping-based distance transfer function*

Distance transfer function is implemented based on 3D texture mapping. Three 3D texture maps are needed in this pipeline: *Texture Unit 0* stores the pixel maps of original volume, *Texture Unit 1* stores the pixel maps of original volume for focal region (for point-to-point distance case) or the pixel maps of segmented data set (for point-to-object distance case), and *Texture Unit 2* stores the distance based pixel

Fig. 1. Distance transform for segmented MR brain data. The first image shows the segmented MS lesions, and in the second image the distance transform is computed. The third image combines the segmented lesions and the distance transform together.

maps (distance to colors and distance to opacity). The distance transfer function is realized through a pixel transfer function.

In practice, it is useful for volume rendering to combine voxel position-dependent rendering with traditional transfer function method together to make full use of their respective advantages during information depiction. In this way, we can render the volume like this: Traditional transfer function based method is used to control object display in the focal region or segmented objects, distance based method is used to control the object visibility and details in the context region or surrounding structures.

For point-to-point distance calculation case, we use distance transfer function in focal region based volume rendering as an example to show how distance transfer function is used in a rendering pipeline. The volume rendering steps using a distance transfer function are as follows:

(i) **Set up the texture units as described before.**

(ii) **Set up the stencil buffer for each region.** There are three regions to be separated: The context region, the region in front of the focal region and the focal region (see Fig. 2). The stencil buffer is prepared for each region separately. Firstly the radius of the intersection circle of the focal region with current texture slice is calculated. Then three different values are set for three different regions in the current slice in the stencil buffer.

(iii) **Render the context region.** In this stage, *Texture Unit 1* is disabled and *Texture Unit 2* is used to modulate the values in *Texture Unit 0*. The data is rendered twice — first for the context region and in the second step for the region in front of the focal region. The difference between two rendering steps is to set up different main colors for each step — white (1.0, 1.0, 1.0, 1.0) for the

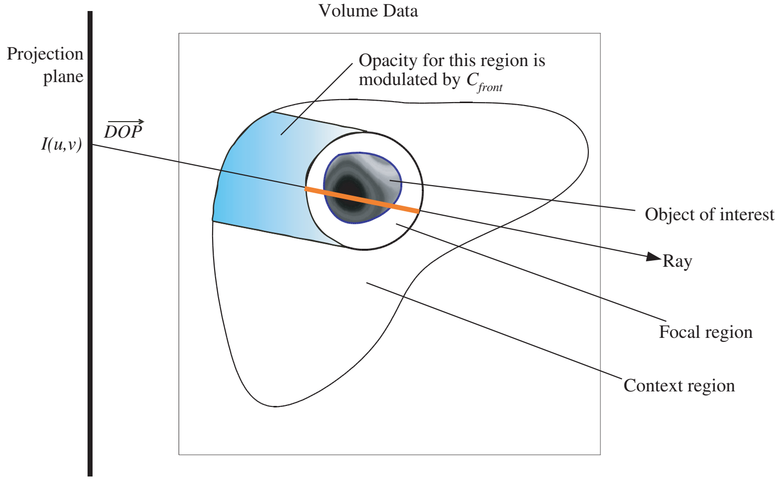


Fig. 2. Three different regions in focal region based volume rendering.

context region and $(1.0, 1.0, 1.0, \alpha_{front})$ for the region in front of focal region. α_{front} is specified by the user to control the visibility of structures in front of the focal region.

(iv) **Render the focal region.** During this stage, *Texture Unit 0* and *Texture Unit 2* are disabled and only the *Texture Unit 1* is enabled. The intersection polygon is now rendered again to get the focal region.

During distance transfer function based rendering, the stencil test is modified to represent only the region that will be rendered. All other parts of the current slice are not enabled. By doing so it is possible to select and render the specified region. When combining distance transfer function into focal region based volume rendering, the distances need to be calculated every time when the focal region position or the radius of the focal region is changed. Afterwards the distance texture needs to be reloaded back into the texture memory. All these operations take much time. Furthermore, setting up the stencil buffer takes up some calculation time. This explains the overall low performance for this implementation. However, this implementation is more efficient than software based approach because of the use of texture mapping based volume rendering.

For point-to-object distance calculation case, the rendering process using a distance transfer function are as follows: First, we composite the *Texture Unit 0* which stores original volume texture with the *Texture Unit 2* which stores distance transform texture using OpenGL function GL_MODULATE. The result is composited with the *Texture Unit 1* which stores segmented volume texture using OpenGL function GL_ADD.

5. Applications

One of the important application fields for distance transfer function is to use it to control the visibility of structures in the context region in focal region based volume rendering. Usually the voxels near the focal region are rendered with detailed information and the voxels far away from the focal region are faded out. This kind of effects can be achieved using the distance transfer function presented in Fig. 3. In this distance transfer function, low distance values are mapped to high opacities and high distance values are mapped to low opacities. So multi-resolution rendering can be performed for context region.

Through changing the shape of distance transfer function, the different volume clipping effects can be performed to reveal the inner structures of the volume data. Figure 4 is a typical distance transfer function for volume clipping. The volume is clipped when the opacity of the corresponding distance is set to zero. This is an easier and more flexible way to perform volume clipping compared to traditional geometric plane based volume clipping. In this method, we just modulate distance transfer function to get volume clippings but not use additional geometrical primitives.

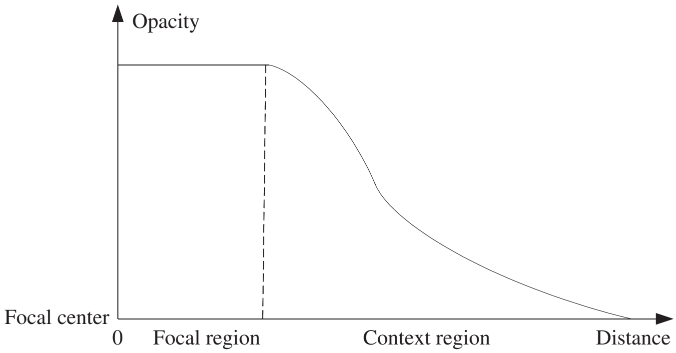


Fig. 3. A typical distance transfer function which emphasize the focal region and de-emphasize the context region.

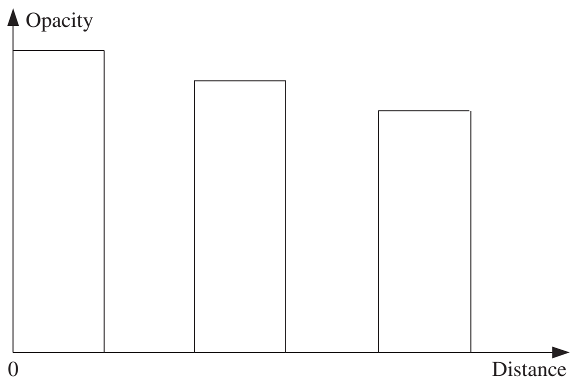


Fig. 4. A typical distance function used for volume clipping. The volume is clipped when the opacity of the corresponding distance is set to zero.

Furthermore, when combining the color into the distance transfer function, the more non-photorealistic rendering oriented results can be estimated. A typical result is the NPR halo effects, which have been widely used in computer graphics.

Using distance transfer function to control the visibility of structures in segmented volume, the user can get a closer vision about the segmented object and its adjacent structures at the same time through combining the segmented volume and the original volume data together. This leads the user to a good understanding of structures of interest.

In summary, the distance transfer function based rendering can be used in several areas:

- To remove unnecessary objects in context region in order to make user concentrate more on the focal region in focal region based volume rendering;
- To perform volume clipping in any position and clip into any parts (depending on different distance functions);

- To perform multi-resolution rendering through changing distance function values to high value for high resolution and low value for low resolution;
- To realize a halo effect, which has been widely investigated in computer graphics when combine distance-color function into the rendering;
- To provide adjacent structures of segmented objects in order to get better understanding of segmented objects.

6. Results and Discussions

The proposed method is first applied to the liver data to show its usefulness for removing the redundant information in the rendition in order to enhance the visibility of structures of interest. In the liver data set, because the data value of the liver is similar with its surrounding structures, if the user only wants to visualize the liver, other structures which have similar data values with the liver will also be shown in the rendition using traditional transfer function based approaches. The distance-based approach allows the user to remove the position-dependent redundant information through using different distance transfer functions. In Fig. 5, the left image is rendered with focal region based volume rendering without distance transfer function applied to it, and the right image is rendered with focal region based volume rendering with distance transfer function applied to it. From these two images, we get two important conclusions. First, with the use of focal region based volume rendering, the user can peel inside the volume data and get the data information which cannot be explored using traditional volume rendering methods. Secondly, because the user is only interested in the liver structure, the distance transfer function is applied to the context region to remove the unnecessary structures which have similar values with the liver. If the traditional transfer function based volume rendering methods are used to render the data, these unnecessary structures cannot be removed because of similar data values. This makes the user concentrate more on structures of interest. The distance transfer function can be used to solve this problem.

Fig. 5. Liver data rendering with focal region applied to it (left), and with distance transfer function presented in Fig. 3 applied to it (right).

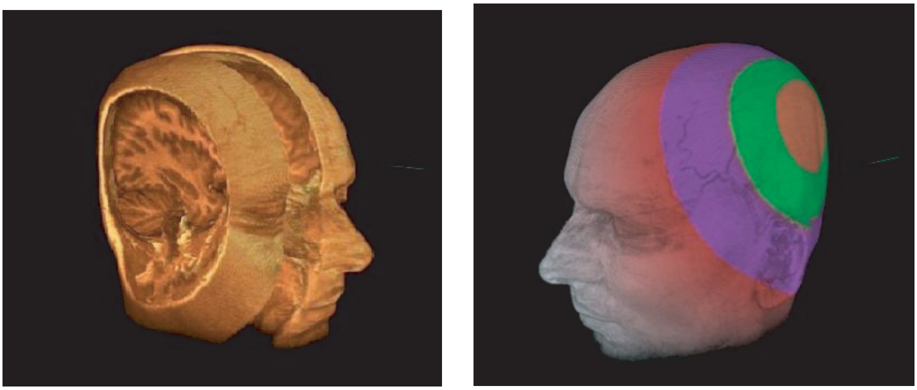


Fig. 6. Distance based volume clipping (left) and distance based volume halo effect (right).

Furthermore, when distance transfer function is applied to the MR brain data set, the different volume clipping effects can be achieved through changing the distance transfer function to the shape presented in Fig. 4. The left image in Fig. 6 shows the result of volume clipping using distance transfer function based approach. From the result we can see that distance transfer function based volume clipping has advantages compared with traditional geometric primitives based volume clipping: It does not need geometric objects to perform volume clipping, and it is more flexible to directly modulate distance transfer function to perform volume clipping and visualize inner structures. When distance is used to modulate colors, the volume halo effects[20] can be gotten as shown in the right image of Fig. 6. This is an easier and more flexible way to realize halo effects for volume data. Volume halo effects can be used in volume graphics to show data features.

The distance-based approach can be used to increase the visibility of relevant information of structures of interest in order to get better understanding of structures of interest. In a segmented data set, the user often has difficulty in understanding segmented objects isolated from their surrounding structures. He often wants to see adjacent structure information of segmented objects in addition to visualizing segmented objects themselves. This is useful for the user to better understand segmented objects and their adjacent tissues for diagnosis and treatment planning. This is a voxel position dependent problem. When the traditional rendering methods (e.g. direct volume rendering and surface rendering) are used to visualize segmented objects, it is difficult to get adjacent structure information and segmented objects at the same time. The distance-based approach allows the user to get adjacent structure information and segmented objects at the same time flexibly through using different distance transfer functions. In Fig. 7, the MS lesions of a MR brain data are first segmented. The distance is defined as the distance between current voxel position and the nearest segmented object boundary. In the first image of

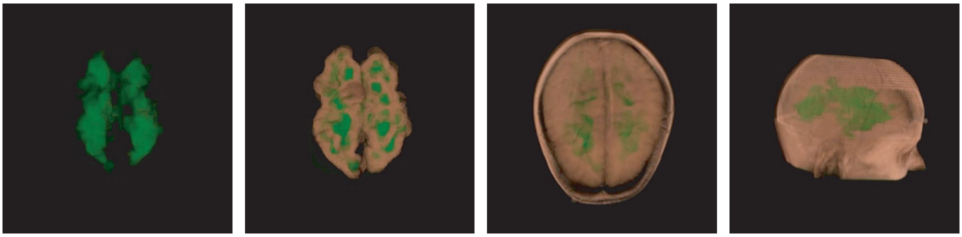


Fig. 7. Using different distance transfer functions to depict segmented objects and their adjacent structures in a MR brain data.

Fig. 7, the segmented objects are rendered using direct volume rendering. It is difficult for the user to understand the segmented objects in this image because of the lack of relevant adjacent structure information. From the second to the fourth image, the segmented objects are rendered together with their surrounding structures. The different surrounding structures of the segmented objects are rendered using different distance transfer functions in different images, which increase the relevant information for the segmented objects. This kind of presentation of the segmented objects gives the structure information adjacent to the segmented objects and lets the user get a better understanding of the structures of interest.

In summary, the distance-based approach is mainly used to control the visibility of structure information in following ways in order to improve visualization efficiency compared with traditional approaches: To remove the redundant information in the rendition in order to enhance the visibility of structures of interest, and to increase the visibility of relevant information of structures of interest in order to get better understanding of structures of interest. First, the distance based approach allows the user to remove the position-dependent redundant information, which traditional methods cannot remove, and makes the user concentrate more on the structures of interest through representing structures near them in more details. Structures far away for structures of interest are faded out using different distance transfer functions. This corresponds to the characteristics of human vision system and can improve visualization efficiency. It is difficult to get this kind of visualization effect using traditional transfer function based approaches. Secondly, the proposed approach can achieve volume clippings through setting the corresponding distance opacity as 1 in a distance transfer function widget. Distance transfer function based approach does not use geometric primitives to create volume clippings as traditional approaches do. It can create different volume clipping effects easily through using different distance transfer functions. This helps the user to investigate volume information more easily and effectively. Thirdly, direct volume rendering and surface rendering methods are often used to render segmented objects in volume data. One of the obvious disadvantages of these methods is that they only render segmented objects in the rendition and the user often has difficulty to understand segmented objects because of the lack of surrounding structures, e.g. to understand segmented MS lesions in MR brain data. It is better to show segmented objects and the original volume information surrounding segmented objects at the same time in order to get a better understanding of the segmented objects. Using the distance-based approach, the user can get different surrounding structures of segmented objects and the related position of segmented objects in the whole volume through setting different distance transfer functions for the original volume surrounding segmented objects. Rendition of segmented objects improves the perception of the user, who gets better and comprehensive understanding of the segmented objects. It also improves the visualization efficiency as compared to the traditional approaches.

7. Conclusions

We proposed a new approach for volume rendering to control the visibility of structures. Through analyzing the current approaches for controlling the visibility of structures, a conclusion is drawn that the voxel position information plays an important role during volume exploration. This voxel position information can be combined into the transfer function domain. In this way, the distance transfer function is obtained through adding another dimension of voxel position information into the transfer function space. The proposed method is applied to the liver and MR brain data set to investigate volume information. The results show that the distance transfer function is a useful and flexible approach for volume information interpretation and information reduction. Compared with the traditional methods of transfer function and volume clipping for controlling the visibility of structures, the distance based approach is more flexible and effective in depicting objects in volume rendering. Future work will focus on the extension of distance definition and improve rendering performance. Possible distance fields can be created to let distance transfer function show quantitative information like distances.

Acknowledgments

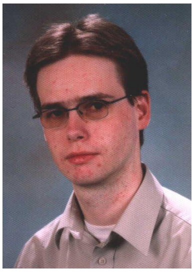
The authors express their thanks to Dr. Hoen-oh Shin (Klinik für Diagnostische Radiologie, Medizinische Hochschule Hannover, Germany), who both had fruitful discussions with the authors and provided the authors with excellent liver data set. They also thank their colleague Fitsum Admasu for providing concertation regarding automatical MS lesions segmentation techniques for MR brain data set, and the reviewers for their comments and suggestions. The MR brain data set and the segmented MR brain data set were provided by Department of Neurology, Otto-von-Guericke University Magdeburg, Germany.

References

1. R. Westermann and T. Ertl, "Efficiently using graphics hardware in volume rendering applications," In *Proceedings of ACM SIGGRAPH 98*, pp. 169–177 (1998).

2. D. Weiskopf, K. Engel and T. Ertl, "Volume clipping via per-fragment operations in texture-based volume visualization," In *Proceedings of IEEE Visualization02*, pp. 93–100 (2002).
3. A. Kaufman and K. Mueller, "Volume visualization and volume graphics," *Technical Report*, Computer Science Department, Stony Brook University (2003).
4. M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications* **8**(5), 29–37 (May 1988).
5. J. Kniss, G. Kindlmann and C. Hansen, "Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets," In *Proceedings of IEEE Visualization 2001*, pp. 255–262 (October 2001).
6. J. Zhou, A. Döring and K. D. Tönnies, "Distance based enhancement for focal region based volume rendering," In *Proceedings of Bildverarbeitung für die Medizin 2004*, pp. 199–203, Berlin (2004).
7. J. Zhou, M. Hinz and K. D. Tönnies, "Focal region-guided feature-based volume rendering," In *Proceedings 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT)*, pp. 87–90, Padova, Italy (June 2002).
8. D. Weiskopf, K. Engel and T. Ertl, "Interactive clipping techniques for texture-based volume visualization and volume shading," *IEEE Transactions on Visualization and Computer Graphics* **9**(3), 298–312 (2003).
9. M. Meißner, U. Hoffmann and W. Straßer, "Enabling classification and shading for 3D texture mapping based volume rendering using opengl and extension," In *Proceedings of IEEE Visualization99* (1999).
10. S. Roettger, S. Guthe, D. Weiskopf, T. Ertl and W. Straßer, "Smart hardware-accelerated volume rendering," In *Proceedings of the Symposium on Data Visualisation 2003*, pp. 231–238, Grenoble, France (2003).
11. K. Engel, M. Kraus and T. Ertl, "High-quality pre-integrated volume rendering using hardware-accelerated pixel shading," In *Proceedings of Europhocs/SIGGRAPH Workshop on Graphics Hardware 2001*, Saarbrucken, Germany (August 2001).
12. H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. Avila, K. Raghu, R. Machiraju and J. Lee, "The transfer function bake-off," *IEEE Computer Graphics and Applications* **21**, 16–22 (May–June 2001).
13. C. Bajaj, V. Pascucci and D. Schikore, "The contour spectrum," In *IEEE Visualization97*, pp. 167–173 (1997).
14. F.-Y. Tzeng, E. B. Lum and K.-L. Ma, "A novel interface for higher-dimensional classification of volume data," In *Proceedings of IEEE Visualization03* (October 2003).
15. R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(6), 641–647 (1994).
16. J. Foley, A. van Dam, S. Feiner and J. Hughes, *Computer Graphics: Principles and Practice*, Addison Wesley, second edition (1996).
17. P. Rheingans and D. Ebert, "Volume illustration: Nonphotorealistic rendering of volume models," *IEEE Transactions on Visualization and Computer Graphics* **7**(3), 253–264 (2001).
18. A. Lu, C. J. Morris, D. S. Ebert, P. Rheingans and C. Hansen, "Non-photorealistic volume rendering using stippling techniques," In *IEEE Visualization 2002* (2002).
19. T. Saito and J. I. Toriwaki, "New algorithms for euclidean distance transformations of an n-dimensional digitised picture with applications," *Pattern Recognition* **27**(11), 1551–1565 (1994).
20. J. C. Gonzato and S. Marchand, "Efficient simulation of halos for computer graphics," In *Proc. of European Congress of Stereology and Image Analysis*, Bordeaux, France (2001).

Jianlong Zhou received the ME degree from Chongqing University, China, in 1999. From 1999 to 2001, he worked as a research assistant and specialized in image and video processing at Institute for Information Engineering, School of Electronic and Information Engineering, in Xi'an Jiaotong University, China. He is currently doing research in computer science at Computer Vision Group, Department of Computer Science, in Otto-von-Guericke University Magdeburg, Germany, since 2001. His research interests include image processing and volume visualization.



Andreas Döring received the Dipl.-Ing. degree in Computational Visualistics from Otto-von-Guericke University Magdeburg, Germany, in 2003. During his study he was mainly interested in image processing and visualization techniques especially volume rendering and non-photorealistic rendering techniques. In his diploma thesis he analyzed several ways of hardware accelerated volume rendering techniques using NVidia GeForce4 extensions. Since 2003 he is working at BaumerOptronic GmbH, Radeberg, Germany, and is working on image processing techniques based on FPGA hardware.

Klaus D. Tönnies received the PhD degree from Technical University Berlin, Germany, in 1987. From 1987 to 1989, he worked as a research assistant professor at MIPG, Department of Radiology, Medical School, in University of Pennsylvania, USA. He worked as an assistant professor at Computer Graphics, Department of Computer Science, in Technical University Berlin, Germany, from 1989 to 1995. From 1996 to 1998, he was the head of Image Processing Group, Department of Radiology, Inselspital, in University of Berne, Switzerland. He is currently a professor and the head of Computer Vision Group, Department of Computer Science, in Otto-von-Guericke University Magdeburg, Germany, since 1998. His research interests include image processing, computer vision, and volume visualization.