



A fast approach to multi-volume display of medical images

Meysam Rahimi

Department of Electrical Engineering ,Urmia Branch ,Islamic Azad University, Urmia ,Iran

Abstract

We required a process called rendering in order to display three-dimensional models, including the volume of organs and tissues in medical images. In computer graphics, rendering operations are divided into two general categories: volume rendering and surface rendering. Volume rendering converts all 3D data to a 2D image, while surface rendering first converts surfaces to basic shapes such as dots, lines, and triangles, and then maps the volume level to a 2D image, and through this operation a large part of the data is deleted. In this research, by changing the process of sampling and composition phases, as well as the changes that we apply, in many cases, we improve the performance of ray casting method of VTK by several times, which is also more practical for the visualization of medical images. Since speed addition of multi-volume display of medical images reduces the speed and accuracy of diagnosis by physicians, this study slightly improved the speed of multi-volume rendering. Previous methods of ray casting have slowed down rendering due to the simultaneity of the sampling phase, the composition and lack of proper memory efficiency in cases with a lot of free space between volumes. However, in this study, in order to increase the rendering speed of multi-volume images of medical images, by defining subspace as a method for separating the sampling phase and the composition phase, we have been able to obtain better results for rendering speeds where there is empty space between volumes. Thus, volume visualization of medical images can be provided at a higher frame rate, which gives a better interactive experience to the end user and enhances the application of volume visualization in the medical sciences. A smart selector can be used in future research to decide based on 3D environmental conditions which of the methods will be used to render the 3D environment. As well as, some volumes can be considered as a multi-volume and some volumes as a single-volume on an instantaneous basis; and the rendering speed can be improved far more by sorting it based on depth with two methods of multi-volume and single-volume rendering.

Keywords:

Multi volume display images, Medical images, VTK

1. Introduction

We required a process called rendering in order to display three-dimensional models, including the volume of organs and tissues in medical images. In computer graphics, rendering operations are divided into two general categories: volume rendering and surface rendering. Volume rendering [Drebin et al. \(1988\)](#) converts all 3D data to a 2D image, while surface rendering first converts surfaces to basic shapes such as dots, lines, and triangles, and then maps the volume level to a 2D image, and through this operation a large part of the data is deleted. With this description, volume rendering in medical image visualization has received more attention due to its accuracy. Several methods are used in volume rendering, which are

* Corresponding author: Meysam Rahimi, Department of Electrical Engineering ,Urmia Branch ,Islamic Azad University, Urmia, Iran
E-mail: roz_e_sefid_67@yahoo.com

classified into image-based, object-based, domain-based and hybrid groups. Ray Casting is one of the image-centric methods that produces a better quality than other methods, but does not have a very good efficiency in terms of rendering speed (Zhang et al., 2011; Shaul et al., 2020). Because of the large volume of data, volume rendering, especially the ray casting technique, is slower than surface rendering. This is exacerbated by the addition of multiple volumes. In single volume, various algorithms are used for early termination ray and space empty skipping to improve the rendering speed (Levoy, 1990; Duran et al., 2019), which by themselves do not solve the rendering speed problem in several volumes. We used the idea of these methods to increase the multi-volume rendering speed in the proposed method through this study. Volumes in full volume rendering are usually observed as cloud-like and semi-transparent volumes. In surface rendering, the problem of simultaneous multi-volume semi-transparent rendering is solved by depth sorting (Amor et al., 2005) or depth peeling of surfaces (Bavoil and Myers, 2008), which of course increases the computational cost. In volume rendering, it is impossible to actually sort or terminate the depth between the two voxels, and combined methods must be used. Several studies (Brecheisen et al., 2008; Bozorgi and Lindseth, 2015; Rößler et al., 2006) have provided methods for multi-volume rendering. The Visualization Toolkit (VTK) (Schroeder et al., 2006; Elshafei et al., 2019), as a comprehensive scientific visualization tool, has used the idea presented in the research (Bozorgi and Lindseth, 2015) to add multi-volume rendering feature. This method is simple and in many cases efficient, but when the volumes are placed at greater distances from each other, the efficiency is greatly reduced. However, research (Brecheisen et al., 2008) used depth peeling, which is a costly operation, especially for most cases of medical imaging, and (Rößler et al., 2006; Binder et al., 2019) used volume layering, which is also costly. Other methods for ray casting in several volumes are compared in (Schubert and Scholl, 2011; Binder et al., 2021). In the mentioned researches, sampling and composition phases are performed and repeated consecutively, but in this research, by changing the process of sampling and composition phases, as well as the changes that we apply, in many cases, we improve the performance of ray casting method of VTK by several times, which is also more practical for the visualization of medical images.

2. Methodology

Unlike the methods presented in the articles mentioned in the introduction, the proposed method in this research, to improve the rendering speed using the ray casting method, separates the sampling phase and the composition phase and leaves empty spaces more efficiently by optimally using memory. If sufficient memory is available and each volume can be sampled along each ray and in places where there is volume, and the combination is performed after sampling from all volumes, the best performance is expected in theory. However, there is usually not enough memory on the GPU for every fragment shader. Therefore, we divide the three-dimensional space into smaller subspaces and perform this operation in the subspaces through this method.

2.1. Importance and necessity

2.1 Sampling phase

We first obtain the boundary of each volume in each subspace to sample each volume in each subspace, or once, first calculate the 3D environment rendering process and save it in an array. We first transfer the ray to the coordinate space of the desired volume, in which the volume has a unit dimension to reduce the calculation of ray collision with the volumes in 3D space, instead of calculating the ray (line) collision with the plates. Using Equation (1), we transfer the initial and final points of the ray to the desired volume space:

$$P' = T_v M_v \cdot M_0^{-1} T_0^{-1} \cdot P \quad (1)$$

In Equation (1), P is the point of interest and P' is its transfer. Mv matrix is the transfer from the total volume to the desired volume v and Tv is the transfer from the desired volume to the texture space of the

volume that has single dimensions. Equation (2) shows the parametric relation of the ray line. Using this Equation and Equation (3), we obtain the variable t point of collision:

$$P = (1 - T)O + tE, 0 \leq t \leq 1 \tag{2}$$

$$t_i = \frac{O_i - P_i}{O_i - E_i} \tag{3}$$

In these Equations, O is the starting point and E is the end point of ray. Due to the plate symmetry in the coordinate space of the desired volume (plates $x, y, z = 0, x, y, z = 1$), we perform the operation of all three pages with the ability to calculate the GLSL programming language vector at one time. Then, we first determine the presence or absence of the desired volume in each subspace using the obtained coordinates, and if the desired volume does not exist, we test the subspace for other volumes in terms of their presence or absence. If the volume is in this subspace, it skips to the initial step and, as shown in Figure 1, the samples are placed naive in the sample array elements and, unlike previous methods, are not combined. Therefore, each volume is sampled once in each subspace, and thus there is no need to examine other volumes several times. If there is a value in the array element, there is a need to integrate the sample, which we will describe in Section 2.2.

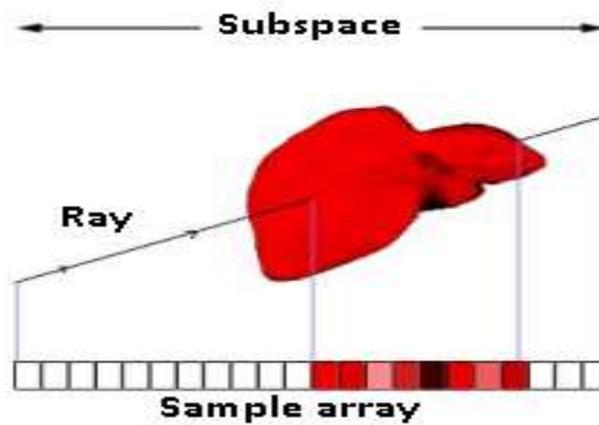


Fig. 1. Subspace for each ray and corresponding sample array

2.2 Combination phase

After sampling all volumes in a subspace, using Equation (4), the common discrete estimation equation in full volume rendering (Max, 1995) is used to combine array samples:

$$I(D) = \sum_{k=1}^D C_k \alpha_k \prod_{i=1}^k (1 - \alpha_i) \tag{4}$$

In this equation, C_k is the color of k th sample, α_k is the cloudy of k th sample, and $I(D)$ is the final value (color) in step D . To reduce the computations in the composition of the samples, we use Equation (5) to calculate the reversal ray in a forward to back direction (Max, 1995):

$$C = C(1 - \alpha_k) + C_k \tag{5}$$

$$\alpha = \alpha(1 - \alpha_k) + \alpha_k \tag{6}$$

Parameters c and α are the color and the amount of cumulative cloudy up to step k , respectively. While we have a lot of memory available, the length of the arrays can be increased, in which case it is required to create a link list in the array to efficiently leave non-sampled spaces in the composition phase. However, the length of these arrays cannot be very large at present due to hardware limitations, and in the arrays selected in this study, not using the link list has advantages such as using the data locality for cache memory and instead the cost of examining each element of the array is the only zero opposite condition. If there is already a value in an array element when sampling, they can be combined by two approaches. The first approach is to use Equations (5) and (6) to combine point samples with the same coordinates. In this case, the sampling sequence will theoretically be effective in the formation of the output color. In practice, if the volumes are not cloudy, the effect will not be obvious. However, if we want to have a theoretically correct result for cloudy volumes, the second approach is to use Equation (7) to calculate color and Equation (6) to calculate cloudy:

$$C = \frac{\sum_{v=1}^N C_v \alpha_v}{\sum_{v=1}^N \alpha_v} \quad (7)$$

In this Equation, C is the color value of the array element for the volume 1 to N in the same coordinate. Equation (7) for color and Equation (6) for cloudy are independent of the sampling order and establish a multi-volume rendering target, which is independent of the order of the volumes to display correctly. However, this Equation will be associated with memory overhead or computational overhead for embedding information in sample array elements to maintain the relationship fraction.

3. Results

In this research, the proposed method has been implemented by customizing the latest version of the Visualization Toolkit (VTK) library version 8.1. The applied system is a regular desktop computer with a dedicated Nvidia GeForce 525m graphics card, 600 MHz frequency, and 2 GB of memory, a 3.2 GHz Core i5 CPU and 6 GB of main memory. Two and three square cubes with dimensions of 100 pixel are used to test the rendering speed, which include a uniform cloudy from 0 to 1, so that the early termination ray algorithm also affects the rendering speed.

Figure 3 shows the rendering of the proposed method and the visualization method for two volumes spaced apart on the diameter of a cube, and the size of the sample array in this experiment is 40. We have also obtained a similar diagram for the three volumes as shown in Figure 3. The results show that by distancing the volumes and creating empty space, the proposed method uses the advantage of empty space to improve the rendering speed and in some cases achieve a 6-fold rendering speed. However, at short distances and when the volumes overlap, due to the overhead that is created, the frame rate cannot be equal to the Visualization Toolkit (VTK) method.

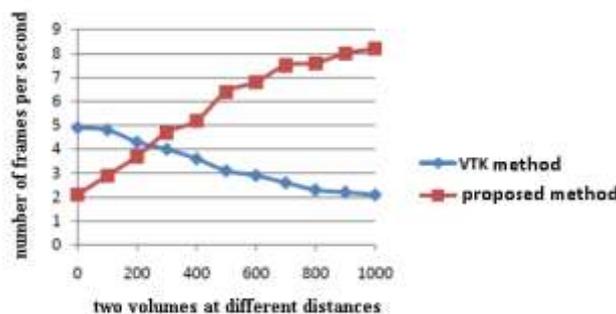


Fig. 2. Comparison of rendering speed in terms of number of frames per second of Visualization toolkit method and the proposed method for two volumes at different distances

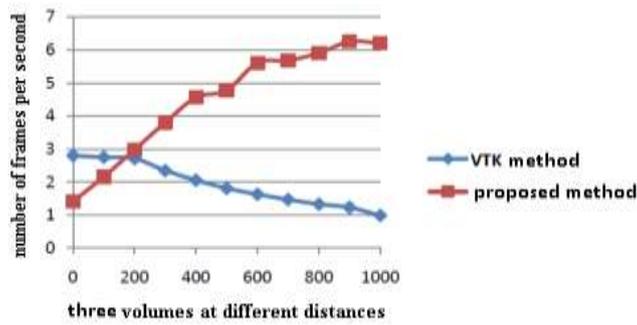


Fig. 3. Comparison of rendering speed in terms of number of frames per second of Visualization toolkit method and the proposed method for three volumes at different distances

We have also examined the size of the sample array, which is representative of the length of the subspace (Fig. 3). The results show that computational overhead and memory access do not have an advantage in small arrays. This improves with increasing the size of the array, but it cannot be considered a significant advantage over the size of the array 30-50 and increasing the size too much is not very effective. As a result, by selecting the size of the array 30, in addition to optimizing performance, more memory can be left for other variables. It can also be observed this diagram after the ascending trend decreases with increasing distance, which is due to the increase of subspaces by increasing the distance and not reducing the number of rays involved in the formation of the final image. However, after a certain distance, the proposed method is quite superior to the visualization toolkit method. According to Figure 5, the muscles, bones, and other organs are visualized from a set of CT scan images of the thigh and pelvis (Zhang et al., 2011) using the proposed method. By placing these three volumes with overlap, the proposed method visualizes the three volumes as shown in Figure 6, which has no quality reduction and is desirable.

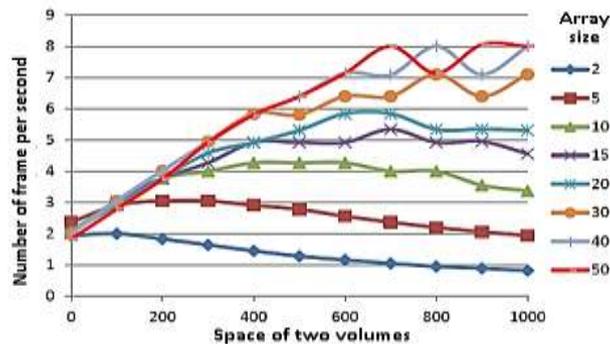


Fig. 4. The effect of sample array size on the frame rate of the proposed multi-volume rendering method



Fig. 5. Three separate volumes of muscles, bones and other organs of the thigh and pelvis with the proposed rendering method



Fig. 6. Simultaneous display of three volumes of muscles, bones and other limbs of the thigh and pelvis with overlapping by the proposed rendering method

4. Conclusion

Since speed addition of multi-volume display of medical images reduces the speed and accuracy of diagnosis by physicians, this study slightly improved the speed of multi-volume rendering. Previous methods of ray casting have slowed down rendering due to the simultaneity of the sampling phase, the composition and lack of proper memory efficiency in cases with a lot of free space between volumes. However, in this study, in order to increase the rendering speed of multi-volume images of medical images, by defining subspace as a method for separating the sampling phase and the composition phase, we have been able to obtain better results for rendering speeds where there is empty space between volumes. Thus, volume visualization of medical images can be provided at a higher frame rate, which gives a better interactive experience to the end user and enhances the application of volume visualization in the medical sciences. A smart selector can be used in future research to decide based on 3D environmental conditions which of the methods will be used to render the 3D environment. As well as, some volumes can be considered as a multi-volume and some volumes as a single-volume on an instantaneous basis; and the rendering speed can be improved far more by sorting it based on depth with two methods of multi-volume and single-volume rendering.

References

- Amor, M., Bóo, M., Padrón, E. J., & Bartz, D. (2006). Hardware oriented algorithms for rendering order-independent transparency. *The Computer Journal*, 49(2), 201-210.
- Bavoil, L., & Myers, K. (2008). Order independent transparency with dual depth peeling. *NVIDIA OpenGL SDK*, 1, 12.
- Binder, J. S., Scholz, M., Ellmann, S., Uder, M., Grützmann, R., Weber, G. F., & Krautz, C. (2021). Cinematic rendering in anatomy: A crossover study comparing a novel 3D reconstruction technique to conventional computed tomography. *Anatomical Sciences Education*, 14(1), 22-31.
- Binder, J., Krautz, C., Engel, K., Grützmann, R., Fellner, F. A., Burger, P. H., & Scholz, M. (2019). Leveraging medical imaging for medical education—A cinematic rendering-featured lecture. *Annals of Anatomy-Anatomischer Anzeiger*, 222, 159-165.
- Bozorgi, M., & Lindseth, F. (2015). GPU-based multi-volume ray casting within VTK for medical applications. *International journal of computer assisted radiology and surgery*, 10(3), 293-300.
- Brecheisen, R., Vilanova, A., Platel, B., & ter Haar Romeny, B. M. (2008). Flexible GPU-based multi-volume ray-casting. In *VMV* (pp. 303-312).
- Drebin, R. A., Carpenter, L., & Hanrahan, P. (1988). Volume rendering. *ACM Siggraph Computer Graphics*, 22(4), 65-74.
- Duran, A. H., Duran, M. N., Masood, I., Maciolek, L. M., & Hussain, H. (2019). The additional diagnostic value of the three-dimensional volume rendering imaging in routine radiology practice. *Cureus*, 11(9).
- Elshafei, M., Binder, J., Baecker, J., Brunner, M., Uder, M., Weber, G. F., ... & Krautz, C. (2019). Comparison of cinematic rendering and computed tomography for speed and comprehension of surgical anatomy. *JAMA surgery*, 154(8), 738-744.
- M. S. Chung. (2000). Visible Korean Available: <http://vkh3.kisti.re.kr>
- Max, N. (1995). Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2), 99-108.
- Rößler, F., Tejada, E., Fangmeier, T., Ertl, T., & Knauff, M. (2006, January). GPU-based multi-volume rendering for the visualization of functional brain images. In *SimVis* (Vol. 2006, pp. 305-18).
- Schroeder, W., Martin, K. M., & Lorensen, W. E. (1998). *The visualization toolkit an object-oriented approach to 3D graphics*. Prentice-Hall, Inc..
- Schubert, N., & Scholl, I. (2011). Comparing GPU-based multi-volume ray casting techniques. *Computer Science-Research and Development*, 26(1), 39-50.
- Shaul, R., David, I., Shitrit, O., & Raviv, T. R. (2020). Subsampled brain MRI reconstruction by generative adversarial neural networks. *Medical Image Analysis*, 65, 101747.
- Zhang, Q., Eagleson, R., & Peters, T. M. (2011). Volume visualization: a technical overview with a focus on medical applications. *Journal of digital imaging*, 24(4), 640-664.
- M. Levoy. "Efficient ray tracing of volume data." *ACM Transactions on Graphics (TOG)*, vol 9. pp. 245-261, 1990.