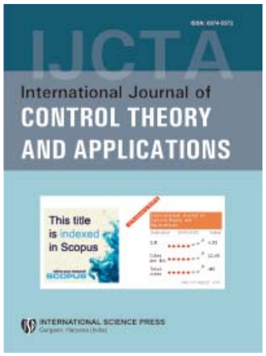

International Journal of Control Theory and Applications

ISSN : 0974–5572

Volume 10 • Number 27 • 2017

Virtual Data Grouping for Volume Visualization using Ray Casting

R. Mehaboobathunnisa[a] A.A. Haseena Thasneem[a] and M. Mohamed Sathik[b]

[a]Research Department of Computer Science Sadakathullah Appa College Tirunelveli, India

E-mail: nisadiq8797@gmail.com, haseenajmal@gmail.com

[b]Principal Sadakathullah Appa College Tirunelveli, India

E-mail: mmdsadiq@gmail.com

Abstract: Visualization of medical data in three dimensions provides an enhanced idea on the objects of focus. The method proposed in this paper renders the 2D CT/MRI slices as a volume using ray casting technique. The classical ray casting technique of volume visualization is combined with the grouping techniques in image processing – clustering based segmentation. Prior to casting the rays, the 2D slices are segmented using Fuzzy Clustering to identify homogeneous regions and a single voxel of individual groups are marked as reference to the other members of the same group. The usefulness of this approach is that it suffices data interpolation for only those unknown samples which serve as reference to other voxels and brings a notable reduction in the computation incurred by the classic ray casting.

Keywords: *Fuzzy clustering, data grouping, ray casting, volume rendering.*

1. INTRODUCTION

The imaging of human internal organs has been evolving with the imaging modalities from normal X-rays and ultrasound scans to CT, MRI, PET, fMRI scans in 2D and 3D. The volumetric visualization of medical data renders a great support for physicians to make clear diagnosis of diseases. The domain of volumetric visualization of medical images encapsulates the conversion algorithms that render the 2D slices in the 3D form. The two broad categories of volume rendering techniques are – Indirect Volume Rendering (IVR) and Direct Volume Rendering (DVR). The IVR algorithms generate an intermediate surface from the 2D data, which is then rendered in 3D. As the rendering is performed from the intermediate surface rather than from the 2D data directly, this technique is referred to as an indirect one. The most common IVR method is Marching cubes [1]. In contrast, the DVR techniques work directly through the input 2D slices to generate the volume and hence called so. Splatting [2], Shear warp [3] and Ray casting [4] are the widely used DVR methods of which the focus of the paper is Ray casting.

Since the proposal, ray casting technique has been revised by many researchers to enhance it in several ways to improve the computational complexity. A review on the improvements in it is discussed in [5]. The general way to visualize individual portions in the data using ray casting is to first segment the input data and then to cast the rays over the segmented regions of interest only [6]. This idea is made use of to bring down the computation.

It is achieved by grouping the data in each slice and then casting rays in a normal manner. In spite of the rays traverse every voxel along its path, only reduced number of unknown voxels are interpolated whereas the remaining are determined by referring to the pre- fixed reference voxels. This visualizes every homogenous segment simultaneously with the specified opacity value with less processing. There is no change in the way of casting the ray; as rays are cast via all pixels. It is the reduction in the number of samples interpolated to determine the accumulative color of the volume that reduces the complexity of the method.

The paper is organized as follows. Section I is introduction; Section II discusses the traditional ray casting technique; Section III reveals the improvement proposed in the classic ray casting; Section IV deals with the results; Section V provides a discussion on the current results and gets concluded with Section VI.

2. CLASSIC RAY CASTING

The ray casting algorithm [4] records the three dimensional information in a two dimensional image plane and visualizes this image plane. The rendering pipeline of the ray casting technique is summarized as follows.

2.1. Data Preparation

1. Casting rays from all pixels of the image plane at the desired angle
2. Taking equidistant samples along all rays
3. Tri - linearly interpolating the samples to get the color

2.2. Visualization

1. Determining gradient normals at each sample and hence the opacity
2. Classifying the voxel samples using the opacity value
3. Compositing all the samples' value of each of the rays to the corresponding pixel in the image plane

The final result of the above procedures is actually a 2D image with all the details of a 3D incorporated into it. In spite of the result being in 2D, it is the optical property embedded which gives the effect of a volume. The opacity of the voxels is calculated using the transfer function in (1).

$$\alpha(x_i) = \begin{cases} 1 & ; |\nabla f(x_i)| = 0 \text{ and } f(x_i) = f_v \\ 1 - \dfrac{1}{r}\left|\dfrac{f_v - f(x_i)}{\nabla f(x_i)}\right| & ; |\nabla f(x_i)| > 0 \text{ and } f(x_i) - r|\nabla f(x_i)| \le f_v \le f(x_i) + r|\nabla f(x_i)| \\ 0 & ; \text{otherwise} \end{cases} \quad (1)$$

The term f_v represents the desired intensity and $f(x_i)$ the current intensity. The opacity calculation is directly dependent on the gradient at each position. For every individual dataset, the opacity of each voxel is predetermined and is stored in a look up table. Hence repeated executions do not require the opacity value to be re-calculated. It costs only a look up to the corresponding entry into the opacity look up table.

The major calculation complexity is due to the interpolation performed for the samples. In general, the trilinear interpolation in (2) is made use of to determine the value of the sample.

$$\begin{aligned} S(i, j, k) = {} & P_{000}(1-i)(1-j)(1-k) + P_{100}\, i(1-j)(1-k) + P_{010}(1-i)j(1-k) + P_{110}\, ij(1-k) \\ & + P_{001}(1-i)(1-j)k + P_{101}\, i(1-j)k + P_{011}(1-i)jk + P_{111}\, ijk \end{aligned} \quad (2)$$

The subscripts i, j, k are the fractional offsets of the sample position which ranges between 0 and 1; and P_{abc} is the voxel whose relative position in a 2 × 2 × 2 neighborhood of voxels is (a,b,c). Whereas the opacity is a readily available value for successive executions rather than for the first time, the sample value is not that case. Every time rays are cast with different sampling distance and at different view angles which necessitates the samples to be interpolated for every execution. As noted in [7], the volume rendering operation requires fairly large number of samples to prevent aliasing effects. Trilinear interpolations employed in ray casting have additional speed penalty to come out with good results. As a step forward to reduce the number of samples being interpolated for distinct datasets, we cluster the data prior to casting the rays. The forthcoming section discusses the procedure to develop a grouped ray casting with the most common and efficient clustering, the fuzzy clustering.

3. VIRTUAL DATA GROUPING FOR RAY CASTING

Grouping, also referred to as clustering is a very common image processing technique whose objective is to identify the homogeneous regions in an image to make processing simple. In general, medical data shall be readily clustered to delineate the organs. Distinct clusters have individual features and intensities and these aspects do not vary tremendously in adjacent slices when the inter - slice distance is very less. This concept is incorporated into the ray casting method to avoid interpolations within homogeneous regions in the interleaving virtual slices which are assumed to have uniform intensities as that of their corresponding segments in their preceding real slice.

Initially, the individual two dimensional data slices are clustered using the Fuzzy C Means (FCM) technique [8]. FCM is a well known clustering algorithm which assigns the data to clusters depending upon the strength of the degree of membership of the data with all available clusters. The degree of membership is determined as given in (3).

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{ik}}\right)^{2/(m-1)}} \tag{3}$$

where u_{ij} is the degree of membership; c is the total number of clusters; d_{ij} represents the distance between the i^{th} data and j^{th} cluster centre; m is the fuzziness factor whose value is generally set to 2.

In this case it is necessary that the user must have some prior knowledge about the data, as the number of clusters to be identified is a user defined variable.

The main concept of the paper is to group the unknown data in the virtual slices that are considered between two successive real slices and then to cast rays in a conventional way. The advantage here is that the data value for the unknown samples are retrieved directly from priorly fixed reference voxels maintained for individual segments in all of the virtual slices instead of performing interpolation. The flow of operations is depicted in Fig.1. Initially, a mask is created for every virtual slice to hold the identity of the reference voxels for each segment. The reference voxels of a specific segment in the virtual slices are generally a permuted set of spatial locations within each segment of the virtual slice. The mask value is set to 1 for those reference voxels and to 0 for the rest. Another data structure – a color reference array, to store the calculated color value of the reference voxels is maintained independently for the slices.

Whenever a ray is sampled along its path, the belonging group of that sample and hence the nearest reference voxel of the corresponding group is identified. The first hit on any segment will have an undefined data value for the reference voxel also. In that case, interpolation is performed normally to define the data for that reference voxel which is stored in the color reference array of the corresponding slice. This procedure is performed whenever a reference voxel is hit for the first time within a group. Successive hits anywhere inside the groups need no interpolation; just a look up to the color reference array will render the data value.

An illustration of the grouping of data in three successive slices is shown in Fig. 2. It shall be noted that the allotment of the groups need not be similar in structure. The logic does not depend on the group size, shape or labels. The number of groups is image dependent and need not be necessarily perfect. The method is tolerant to reasonable levels of under segmentation and over segmentation.

Every group is randomly permuted to get a specified percentage of voxels to be fixed as reference voxels. Hence, a sample is assigned data value of the reference voxel of the same group which has the minimum Euclidean distance with the it. This ensures reduction in the interpolation computation with better reliability of data.

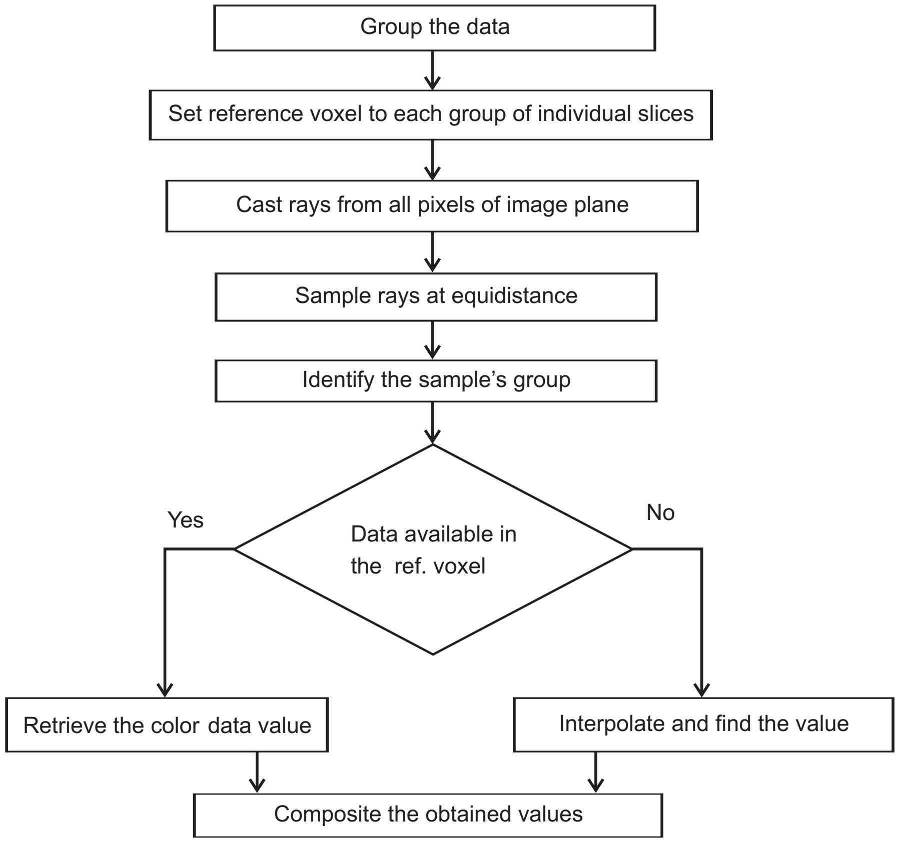

Figure 1: Sequential procedure of the proposed method

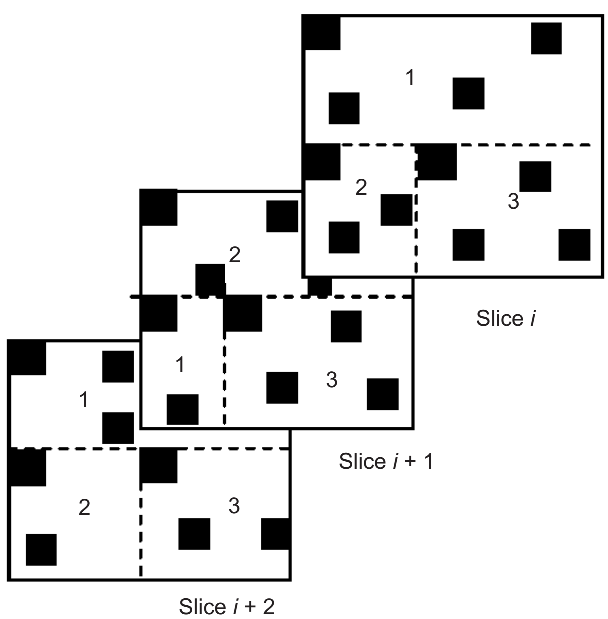

Figure 2: Data groups in 3 successive slices – an illustration

4. EXPERIMENTAL RESULTS AND ANALYSIS

This section shows our rendered volumes using the traditional ray casting and the grouping based ray casting at different sampling distances. We have also shown an analysis chart that depicts the improvement in its computation. The following results depict the visualization of two input data sets each including 10 individual slices and 15 individual slices respectively.

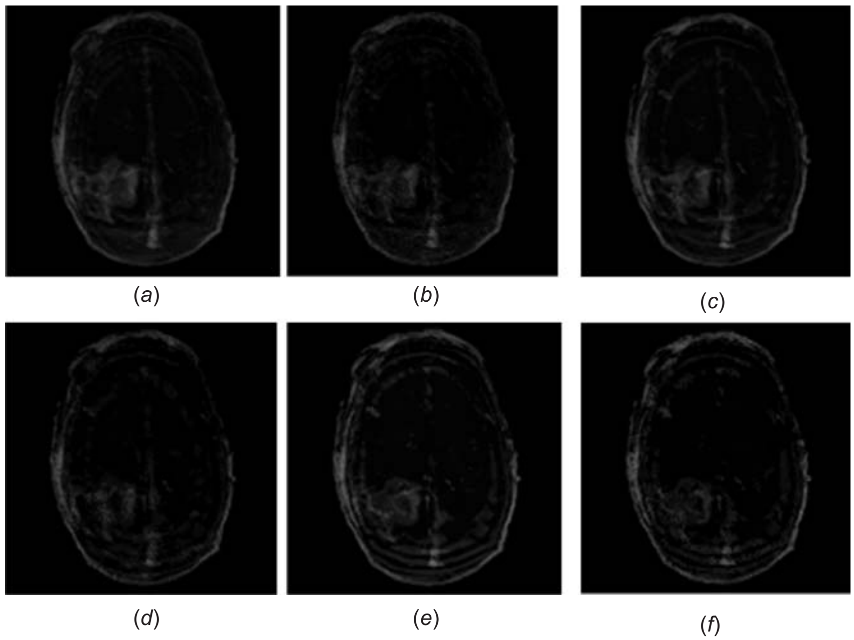

Figure 3: Rendered Volume for dataset 1 using (*a*) Traditional RC; SD = 1, (*b*) Grouped RC ; SD =1 (*c*) Traditional RC; SD = 1.5, (*d*) Grouped RC ; SD = 1.5(*e*) Traditional RC; SD = 2, (*f*) Grouped RC ; SD = 2

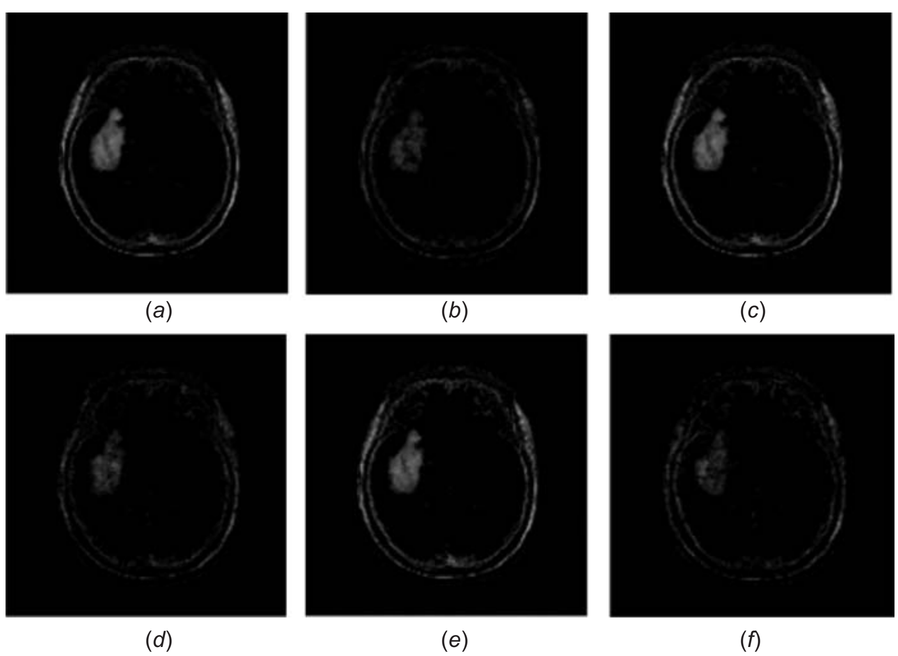

Figure 4: Rendered Volume for dataset 2 using (*a*)Traditional RC; SD = 1, (*b*) Grouped RC ; SD = 1 (c) Traditional RC; SD = 1.5, (*d*) Grouped RC ; SD = 1.5 (*e*) Traditional RC; SD = 2, (*f*) Grouped RC ; SD = 2

Table 1
Analysis of image quality for Traditional RC vs Grouped RC

	PSNR for dataset 1	*PSNR for dataset 2*
SD = 1	35.3109	39.52
SD = 1.5	34..199	38.08
SD = 2	33.98	37.97

The analysis of computational complexity of the discussed methods is as follows. The major computation in the traditional ray casting algorithm is the interpolation performed for each sample. As per the formula, to determine the data value for a sample using (2), there are 17 additions and 16 multiplications [9] involved in a single interpolation. The general complexity of N additions is O(N) and that of N multiplications is $O(Nn^2)$ where *n* is the number of digits involved [11]. Therefore a single sample needs about 161 computations with an assumption that the multiplication is made over 3 digit operands.

The proposed improvement which groups the input data prior to ray casting results in a significant reduction in the number of interpolations. But, it incurs additional computations on FCM in spite of which it sounds efficient. The complexity of FCM is $O(ndc^2i)$ where *n* is the number of data points, *d* is the number of dimensions, *c* is the number of clusters and *i* is the number of times FCM is applied over the entire data [10]. The data used for discussion is of 150 x 150 pixels resolution. Dataset 1 includes 10 slices grouped into 4 clusters each and dataset 2 includes 15 slices grouped into 5 clusters each. Using the above mentioned details, the computational complexity for the two input datasets using traditional ray casting and the proposed ray casting are tabulated in Table.2. and Table.3.

Table 2
Analysis of computations for Grouped RC vs Traditional RC for Dataset 1

	Grouped RC (10% of group voxels as reference)	*Traditional RC*
SD = 1	36.71144×10^5	36.225×10^6
SD = 1.5	25.84233×10^5	25.3375×10^6
SD = 2	18.59733×10^5	18.1125×10^6

Table 3
Analysis of computations for Grouped RC vs Traditional RC for Dataset 2

	Grouped RC (12% of group voxels as reference)	*Traditional RC*
SD = 1	65.92661×10^5	54.3375×10^6
SD = 1.5	44.19×10^5	36.225×10^6
SD = 2	35.50083×10^5	28.98×10^6

5. DISCUSSION

The ray casting technique based on data grouping is found to assure the reduction in computational complexity with the advantage of preserving the image quality. In spite of the availability of more time efficient K means algorithm for clustering, the paper has used FCM because of the possibility of empty clusters in the former which may leave data unassigned to any of the groups. The speed of the proposed method is directly proportional to the clustering speed; which in turn depends on the cluster number. So it is important to work with an optimum number of clusters. The other features that have impact on the performance of the algorithm include the sampling distance and the number of input slices. The sampling distance is purely a user defined value which when small results in smooth views. An another factor that has an impact on the raise in computational efficiency is the size of the input data.

6. CONCLUSION

Though the grouping of data incurs loss of intensity details, the main focus is visualization but not any kind of reconstruction and hence the technique is adoptable for the purpose. Our future work shall be to incorporate additional features based tuning of groups to maintain even more details in the resultant volume image.

REFERENCES

[1] W. Lorenson, L. Cline , "Marching Cubes:A High Resolution 3D Surface Construction Algorithm", Computer Graphics, Vol.21 no.4, pp.163-169, 1987.

[2] Zwicker, Matthias, Hanspeter Pfister, Jeroen van Baar, and Markus Gross, "Surface Splatting", Proceedings of the 28th annual conference on Computer graphics and interactive techniques, August 2001.

[3] P. Lacroute, M. Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation, *Proc. SIGGRAPH '94*, Orlando, Florida, July, 1994.

[4] Marc Levoy, " Display of surfaces from volume data", Journal on Computer graphics and applications, IEEE 1988.

[5] M. Mohamed Sathik, R.Mehaboobathunnisa, A.A.Haseena Thasneem, S.Arumugam, "Ray Casting for 3D Rendering – A Review", Intl. Journal of Innovations in Engineering and Technology, Vol. 5, Feb 2015.

[6] Bernhard Preim, Dark Bartz, Visualization in Medicine - Theory, Algorithms and Applications, The Morgan Kauffman series in Computer Graphics, pp 138 – 140, Elsevier.

[7] Issac Bankman, Handbook of Medical Image Processing Analysis, Elsevier, pp. 733 – 734, 2008.

[8] James C Bezdeck, Robert Ehrlich and William Full, "FCM – The Fuzzy c-means Clustering Algorithm", *Computers and Geoscience*, Vol 10, No. 2 – 3, 1984.

[9] Lili Lin, Shengyong Chen, Yan Shao and Zichun Gu, "Plane-Based Sampling for Ray Casting Algorithm in Sequential Medical Images", Computational and Mathematical Methods in Medicine, Hindawi, 2013.

[10] Vuda Srinivasa Rao, Vidyavathi, "Comparative Investigations and Performance Analysis of FCM MFPCM Algorithms on Iris Data", Indian Journal of Computer Science and Engineering, Vol.1, pp 145 – 151.

[11] https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations