Global Analysis and Discrete Mathematics
Volume 6, Issue 1, pp. 83–98
ISSN: 2476-5341

Ray casting based volume rendering of medical images

Maryam Ghobadi · Arash Azimzadeh Irani

Received: 6 March 2021 / Accepted: 24 April 2021

Abstract The goal of 3-D visualization is to provide the user with an intuitive interface which enables him to explore the $3-D$ data. The rapid development in information technology has immensely contributed to the use of modern approaches for visualizing volumetric data. Consequently, medical volume visualization is increasingly attracting attension towards achieving an effective visualization algorithm for medical diagnosis and pre-treatment planning. Previously, research has been addressing implementation of algorithm that can visualize 2-D images into 3-D. Meanwhile, in medical diagnosis, finding the exact diseases location is an important step of surgery / disease management. For 3-D Medical Data, Magnetic Resonance Images (MRI) have been used to create the 3D model, we used the Direct Volume Rendering technique. This paper proposes a ray casting algorithm for accurate allocation and localization of human abdomen abnormalities using magnetic resonance images (Abdomen MRI) of normal and abnormal patients.

Keywords 3-D visualization · Direct Volume Rendering · Ray casting · Abdomen MRI

Mathematics Subject Classification (2010) 97R60

1 Introduction

In medical field, visualizing the internal structure of body is very important for proper medical diagnosis [1]. To aid doctors in visualizing the internal

M. Ghobadi
Department Mathematics and computer science, Damghan University, Damghan, Iran
E-mail: maryam.ghobadi2774@gmail.com

A. Azimzadeh Irani
School of Mathematics and Computer Sciences, Damghan University, Damghan, Iran.
E-mail: a.azimzadeh@du.ac.ir

body parts, scanning methods like MRI (Magnetic Resonance Imaging), X-ray and CT (Computed Tomography) scan are used. Output of such methods are generally one or more grayscale 2D images [2]. It is difficult to visualize the internal structure with the help of these 2D images. In reality, all the tissues and organs of human body are 3D, so it is almost impossible to determine the accurate spatial location and shape/size of ROI only with 2D slices produced by MRI. To solve this problem, we propose a 3D volume rendering and visualization from the data obtained using MRI scan as shown in Fig. Reconstructing 3D medical images with these 2D slices using volume rendering methods IS essential to computer-aided diagnosis (CAD). This has been achieved using Volume Rendering [3]. Volume rendering is an important graphics and visualization technique. A rendered In order to achieve our goal, we chose Direct Volume Rendering (DVR) technique for 3D visualization. DVR methods generate images of a 3D volumetric data set without explicitly extracting geometric surfaces from the data. Medical image volume rendering can be achieved using three kinds of techniques: object-order techniques, image-order techniques and hybrid techniques. The classic algorithms of these techniques are ray casting (also known as ray tracing), 3D texture mapping and shear-warp algorithm. In ray casting, a ray of light is made to pass through the volume data. The interaction of each voxel with this ray is used to assign RGB and alpha values for every voxel in the volume. As a result, we are able to generate the 3D model of the region of interest using the 3D data.

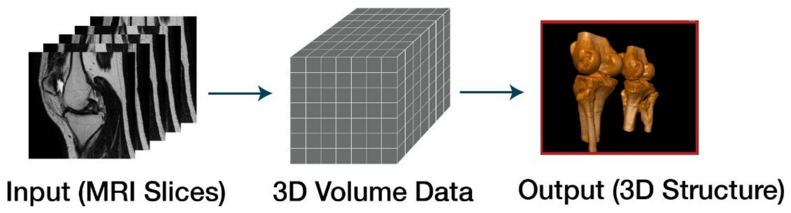

Fig. 1: Extracting $3D$ bone from MRI of knee.

2 Direct Volume Rendering

Direct Volume Rendering achieves rendering by directly obtaining 2-D images of the original object from the 3-D volumetric data without any generation of intermediates. This approach actually overcomes identified drawbacks of surface rendering but with an increase in algorithm complexity and rendering time.
The processing order of the data in DVR technique is divided into three categories:

(i) Object-Order DVR directly obtains 2-D images of the original object from 3-D volumetric data using forward mapping order.

(ii) Image-Order DVR does the rendering the same way as Object-Oder approach above but through backward mapping. This is further separated into Maximum Intensity Projection (MIP), X-Ray Rendering and Full DVR. While MIP only writes the interpolated sample of the largest value along each ray and to each pixel, X-Ray Rendering sums all the interpolated samples along each ray and Full DVR uses sample composition of simulated light effect. Meanwhile, whenever DVR is discussed, image-order full DVR is actually referred.

(iii) Domain-based DVR maps volumetric data to corresponding domain (e.g. frequency domain) before direct generation.

3 RELATED WORKS

3-D Visualization of volumetric medical data is an important aspect of image processing and it has shown as a promising research area due to its significance in the medical domain. The main phases in the direct volume rendering are the classification and the rendering stages. This section will discuss some of the very recent materials published in this domain.

3.1 Classification

Wong et al. [4] design transfer functions based on morphing factor function. The work inputs the start and the end transfer function by user for automatic generation of the intermediate transfer function.

Recently, Chu et al. [5] design an effective transfer function for direct volume rendering. The design of transfer function has a greater impact in direct volume rendering processes towards assigning color and opacity to each sample based on a measured property in the data. The work proposed a transfer function-design method based on feature variation curve. With such good design transfer function coupled with volume rendering development on GPU, the research records desirable speed and even shows that segmentation exercises might not be compulsory for volume rendering processes.

Towards the classification of data for an effective direct volume rendering, Correa & Ma [6] propose ambient occlusion. Since transfer function does the classification in DVR, the proposed approach uses occlusion spectrum, the distribution of ambient occlusion of a given intensity value in a 3-D volumetric data, which provides better two-dimensional transfer functions for complex data classification. Xujia et al. [7] proposes multi-dimensional transfer function based on boundaries and present the scalar-gradient magnitude histogram.

3.2 Rendering

Shihao et al. [8] propose a direct volume rendering with 3-D texture using hardware-assisted texture mapping. As the outstanding challenges in medical domain is how to render a 3-D image fast, the implementation uses trilinear interpolation to accelerate rendering speed. Some samples of engine CT scan, a human MRI head data set, beetle CT scan and hand CT scan data set were exploited.

Visualization Tool Kit from Kitware Inc. has been a great research development tool for research. Ling et al. [9] employ the use of VTK for context-preserving volume rendering. The idea was to contribute to the improvement of Maximum Intensity Projection (MIP) technique. Local Maximum Intensity Projection (LMIP) is an extended version of MIP introduced to overcome the shortcoming of MIP which is its inability to adequately depict the spatial relationships of overlapping tissues. The work presents a better and improved Local Maximum Intensity Projection that computes threshold and shading for LMIP.

With the faster and more stable Graphics Processing Units (GPUs) from Intel, Chen & Hao [10] implemented a GPU-based volume ray casting for resampling and representation of 3-D texture onto a sampling surface. Fragment shaders were performed with the ray casting algorithm. Visual human male CT, human head MRI and pet chest data scan were used for the experiment. The research proved an interactive speed with the algorithm for high image data.

Ray casting technique has been noted to produce high-quality images in direct volume rendering. Kim & aja [11] implemented cell processor architecture for broadband engine with regular datasets for direct volume rendering. Similarly, Cox et al. [12] propose parallel cell architecture broadband engine processor for speeding up the ray casting of irregular datasets. The work still requires optimization of the approach.

Direct Volume Rendering involves sampling and resampling of data. The resampling stage is carried out with the use of filters, typically trilinear, in order to ensure efficient and quality images. In some cases, quadratic or cubic filters are used for higher image quality but they are expensive to evaluate even with GPU acceleration. In addition, Csébfalvi & Domonkos [13] propose a frequency-domain upsampling on an optimal Body-Centered Cubic (BCC) lattice, which was demonstrated to have similar quality as cubic filters.

4 RESEARCH METHODOLOGY

4.1 Ray casting

Ray casting is an image-order volume rendering technique firstly proposed by Levoy in 1988 (Gong & Wang 2010) [14]. In ray casting, rays are cast into the dataset. Each ray originates from the viewing point, and penetrates a pixel in the image screen, and passes through the dataset. At evenly spaced intervals along the ray, sample values are computed using interpolation [ref: Figure 2]. The sample values are mapped to display properties such as opacity and color. Final pixel values are found by compositing the color and opacity values along the ray. The composition models the physical reflection and absorption of light [15]. Composite ray casting is a flexible approach for visualizing several semi-transparent surfaces contained in the data and produces high quality images. However, the alpha blending evaluation of viewing rays is computationally expensive, especially when super sampling is performed tri linearly interpolating each single sample.

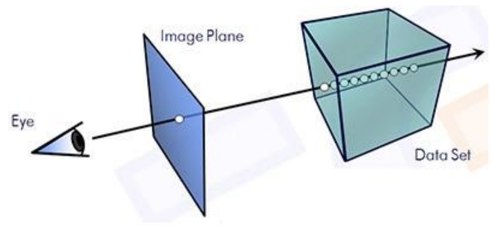

Fig. 2: Ray casting scheme

4.2 Algorithm Overview

For each pixel:

- Calculate ray from viewer point through pixel
- Find intersection points with scene objects (e.g., a sphere)
- Calculate the color at the intersection point near to viewer (e.g., Phong illumination model)

Equation of the ray passing through a pixel:

$$x(t) = o + tv \tag{1}$$

Where; o is the camera (eye) position; v is the vector that stands for the direction of the ray starting at o and passing through pixel (i, j):

$$v = \frac{x - o}{\|x - 0\|} \tag{2}$$

Where x is the float-point location of the window corresponding to the pixel (i, j) of a discrete view screen (in the view plane) of resolution (W, H):

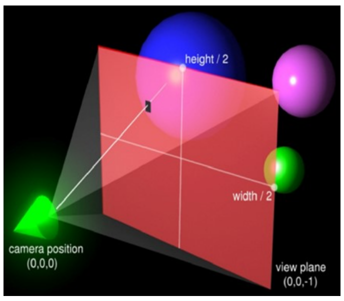

Fig. 3: Constructing a ray through each pixel

Side view of camera at o:

- Position of the i-th pixel $x[i]$?
- Let us first agree that:

$$\begin{aligned} P_0 &= \bar{o} + dl - d\tan(\theta)u \\ P_1 &= \bar{o} + dl + d\tan(\theta)u \\ h &= 2d\tan(\theta) \end{aligned} \tag{3}$$

- Also:

$$x[i] = \frac{i+0.5}{H}(P_1 - P_0) \tag{4}$$

So

$$x[i] = \frac{i+0.5}{H}hu \tag{5}$$

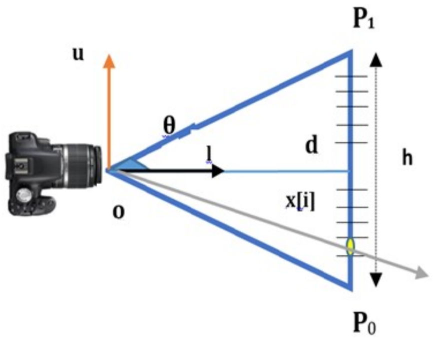

Fig. 4: Side view of camera at o

Where o origin of camera (pinhole), l look vector, u up vector, H discrete height of screen (in pixels), h height of screen, d distance to screen, θ field of view (FOV) or frustum halfangle.

Top view of camera at o:

- Position of the j-th pixel $x[j]$?
- Analogously, we have:

$$x[j] = \frac{i + 0.5}{W} wv \tag{6}$$

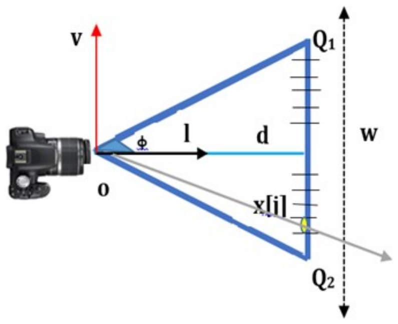

Fig. 5: Top view of camera at o

Where o origin of camera (pinhole), l look vector, u up/side vector, W discrete width of screen (in pixels), w width of screen, d distance to screen, φ field of view (FOV) or frustum halfangle.

In conclusion The equation of the ray through each pixel (i, j) is given by:

$$x[i, j] = o + \frac{i + 0.5}{H} hu + \frac{j + 0.5}{W} wv \tag{7}$$

4.3 Proposed system

A flowchart may be of great importance towards better understanding of the Ray casting Method presented in this section. Figure 2 is therefore provided prior to any further elaboration

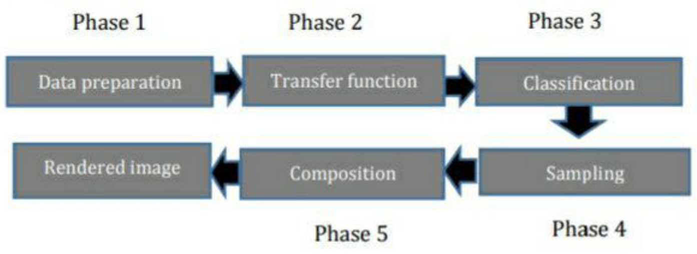

Fig. 6: Ray casting method Flowchart

4.3.1 Phase 1- Data preparation

The first step is data preparation, which could be any processing from checking that voxels are aligned correctly to interpolation of grids to increase the number of voxels on one axis.

4.3.2 Phase 2-Transferfunction

Ray casting requires classification and shading stages. Transfer functions are used to perform classification and shading in DVR by assigning color and opacity to each sample in the data based on a measured property. The ability to differentiate between different tissues is of great importance to medical images. Color Transfer Function and Opacity Transfer Function are for color and opacity assignment respectively.

Data Classification, CT will identify fat, soft tissue and bone and Each will have known absorption levels, say $f_{fat}, f_{soft-tissue}, f_{bone}$

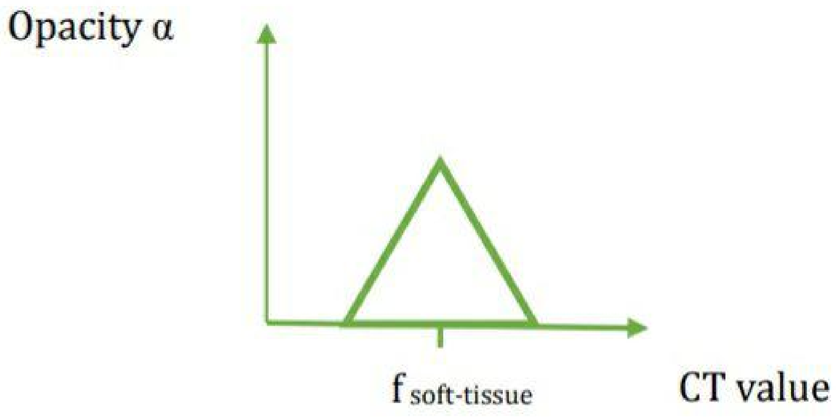

Fig. 7: This transfer function will highlight soft tissue

To show all types of tissue, we assign opacitiesto each type and linearly interpolate between them

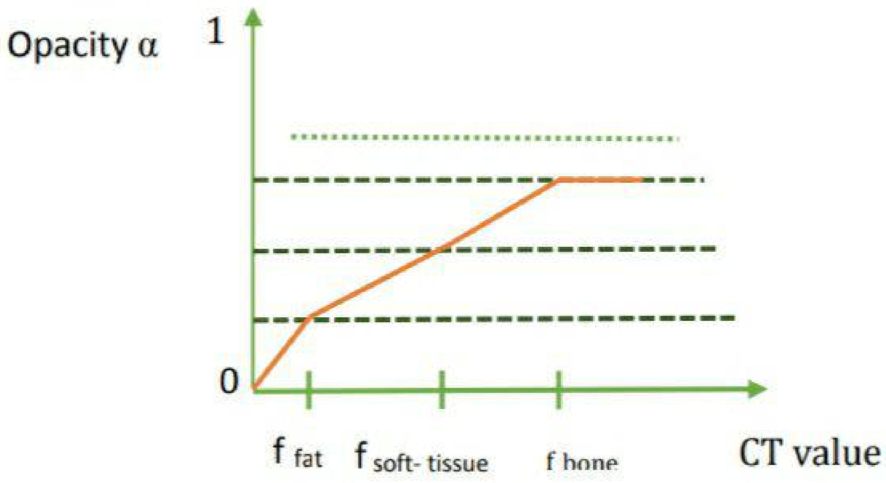

Fig. 8: This is known as opacity transfer function

Color classification is done similarly:

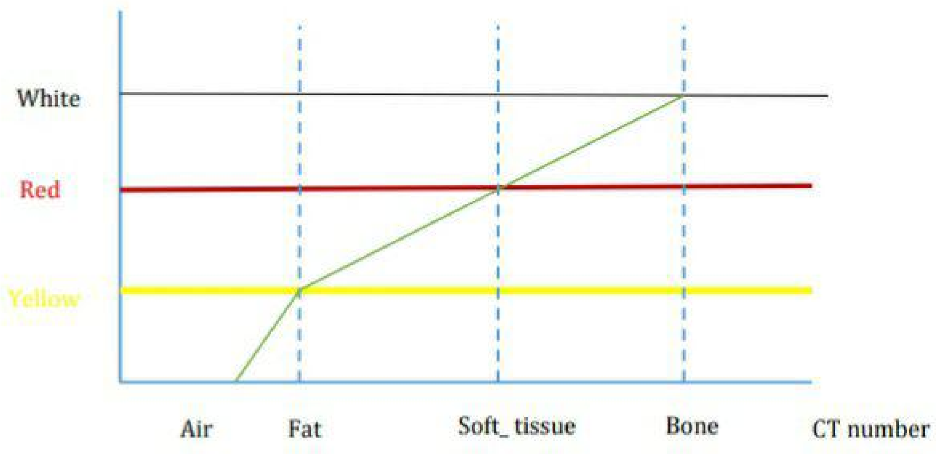

Fig. 9: This is known as color transfer function

4.3.3 Phase3-CLASSIFICATION

Classification for volume rendering is the assignment of color and opacity for a discrete step defined by two samples through the volume. Color and opacity are assigned to a scalar within the volume through a user specified mapping called a transfer function. The shading is done to yield voxel colors usually using Phong's shading model, while the classification is to determine voxel opacities with two different methods: iso-value contour surfaces, which is for pure volume rendering with the possibility of showing several semitransparent surfaces and region boundary surfaces, which is concentrated on rendering medical data and biological tissues in particular.
The shading and classification transfer functions were set using *vtkColorTransferFunction* and *vtkPiecewiseFunction* respectively.

4.3.4 Phase 4-Resampling

In the third step, the volume is set up for the ray casting stage. Rays are cast into the volume from each pixel on screen. Each ray is sampled at equidistant intervals, every sample is determined using tri-linear interpolation of the eight closest voxels.

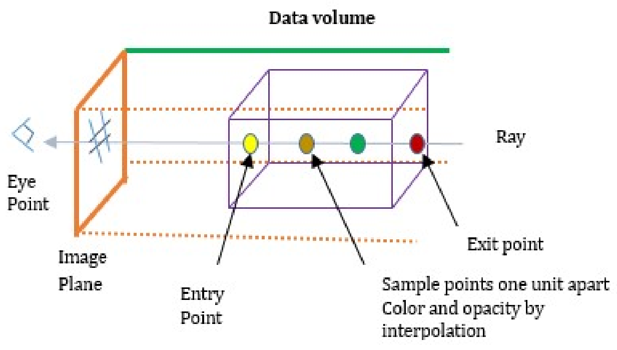
Data volume
Ray
Eye Point
Image Plane
Entry Point
Exit point
Sample points one unit apart
Color and opacity by interpolation

Figure 10: Casting the Rays and Taking Sample

Fig. 10

Estimating the sample value requires evaluation of the tri linear interpolation equation:

$$\begin{aligned}C(u,v,w) = {} & C_{000}(1-u)(1-v)(1-w) \\ & +C_{100}u(1-v)(1-w) + C_{010}(1-u)v(1-w) \\ & +C_{110}uv(1-w) + C_{001}(1-u)(1-v)w \\ & +C_{101}u(1-v)w + C_{011}(1-u)vw + C_{111}uvw\end{aligned} \tag{8}$$

Where:

$$u = \frac{x - x_0}{x_1 - x_0} \tag{9}$$

$$v = \frac{y - y_0}{y_1 - y_0} \tag{10}$$

$$w = \frac{z - z_0}{z_1 - z_0} \tag{11}$$

$u, v,$ and w are fractional off sets of the sample position in the $x, y,$ and z directions, respectively. These variables are between 0 and 1. Cabc is a voxel whose relative position in a $2\times2\times2$ neighborhood of voxels is (a, b, c). $a, b,$ and c are the least significant bit of the $x, y,$ and z sample position, respectively.

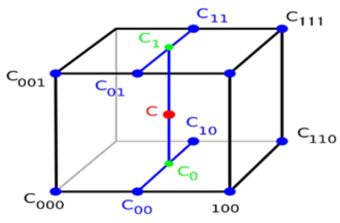

Fig. 11: Resampling using tri linear interpolation

4.3.5 Phase 5-COMPOSITING

After a sample has been classified and shaded, the last step before moving to the next sample is to blend it with the previous samples using alpha compositing. Just as rearranging plates of different colored glass will change the color of objects seen through the plates, the order in which the data is composited will change the results of the volume rendering. The compositing algorithm for back-to-front slicing is computed as a function of RGB color C and opacity α [16]:

$$C_{out} = C_{in}(1 - \alpha_{new}) + C_{new}\alpha_{new} \tag{12}$$

And for front-to-back order:

$$C_{out} = C_{new}\alpha_{new}(1 - \alpha_{in}) + C_{in}\alpha_{in} \tag{13}$$

$$\alpha_{out} = (1 - \alpha_{in})\alpha_{new} + \alpha_{in} \tag{14}$$

C_{new} and α_{new} are the color and opacity values of the newly calculated voxel contributing to a particular pixel, while, $C_{out,in}$, and $\alpha_{out,in}$ are the accumulated color and opacity values at the pixel from which the ray is emanating after and before the ray passes through the new voxel [17].
In this paper, we applied front-to-back compositing because it facilitates acceleration techniques such as early-ray termination, which prevents compositing after a pre-defined opacity has been reached (e.g., 80% opaque). This method avoided unnecessary computation by skipping regions of the volume that are obscured in the current view. This step was done using the VTK class *vtk* VolumeRayCastCompositeFunction.

5 EXPERIMENTS AND RESULTS

Based on the algorithm and technique introduced above, we developed a medical image rendering system using Tcl and VTK. Figure6 illustrates the system rendering pipeline. Some useful VTK classes are applied to complete the main work such as ray casting, volume properties' setting and rendering. We have experimented MRI volume (MRI : $512x512x119$ voxels) with our system and converting it to a three-dimensional model. And the results are located in the form of the following images in this paper.

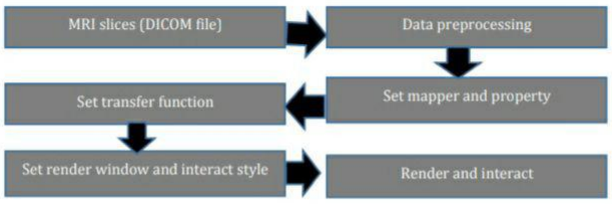

Fig. 12: Medical image rendering system pipeline

5.1 Demonstration

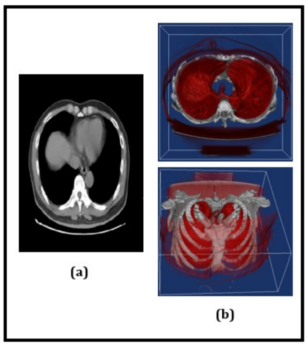

Fig. 13: : (a) *MRI* data of chest, (b) 3D volume data of chest

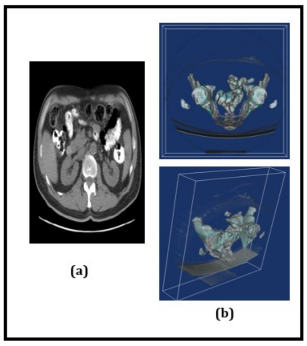

Fig. 14: :(a) *MRI* data of liver and Kidney, (b) 3D volume data liver and kidney

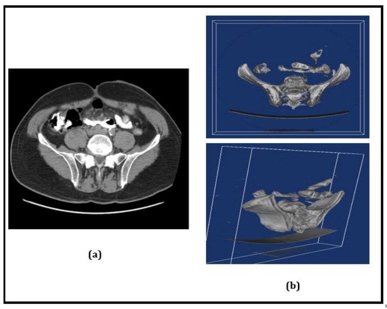

Fig. 15: (a) *MRI* data of pelvis, (b) 3-D volume data of pelvis

5.2 Analysis

This section elaborates on the figures presented in section 5.1. As may be noticed, figures 7, 8 and 9 merely present the ray casting methodology of this paper. According to the above figures, it is easier to diagnose the disease in 3-D images and stained by ray casting method. In Figure 7, the distinction between lung tissue and rib bones is clearly marked by discoloration and it is making easy to diagnose pneumothorax, emphysema, and rib fracture. In Figure 8, shows the diagnosis of kidney and gallstones and trauma that causes the kidneys and liver to rupture, leading to internal bleeding in the peritoneal region. Figure 9 clearly shows the diagnosis of fluid accumulation inside the pelvis, ovarian problems, bladder stones, and hip fractures in 3D images.

6 VTK CLASSES DESIGN

This paper focuses on volume rendering. We create a vtkPiecewiseFunction object to map the scalar values in the volume dataset to opacity, and a vtk-ColorTransferFunction object to map the scalar values to color. These two transfer functions are referenced from the vtkVolumeProperty object. In addition, we use the ShadeOn() method of vtkVolumeProperty to enable shading for this volume, and the SetInterpolationTypeToLinear() method to request trilinear interpolation. Since we are using a ray casting approach, we need to create a ray function. In this example we use vtkVolumeRayCastCompositeFunction object for this purpose. The output of the reader is given to the vtkVolumeRayCastMapper as the scalar input, and the SetVolumeRayCast-Function() method is used to assign the ray function. The vtkVolume object is quite similar to a vtkActor , and the SetVolumeMapper() and SetVolumeProperty() methods are used just like the SetMapper() and SetProperty() methods of vtkActor . Finally, we add this volume to the renderer. Fig. 7 shows typical vtk classes for volume rendering application.

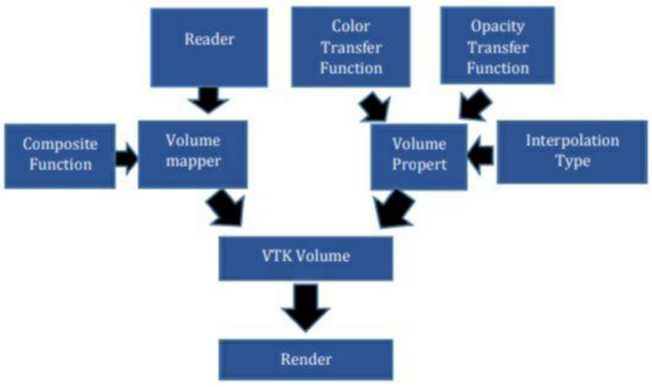

Fig. 16: Typical VTK Classes for Volume Rendering

7 CONCLUSION

The existence of standard surface modeling has only assisted in defining the opaque surface of objects, this hinders us from seeing the inner part of the object. With volume visualization, we can make the boundaries of the object transparent and hence the inner part would become visible. Making the 2-D image structure of Abdomen MRI visible by converting it to 3-D model will facilitate in medical analysis. Hence, Ray casting for direct volume rendering of abdomen MRI will be a meaningful contribution to the application of MRI data in human Abdomen diagnosis and treatment.

Table 1 advantages and disadvantages of Ray casting

Advantages of Ray casting	**Disadvantages of Ray casting**
Realistic simulation of lighting, better than ray casting.	Performance is very poor.
Effects such as reflections and shadows, which are difficult to simulate using other algorithms,are a natural result of the ray casting algorithm.	Scan line algorithms and other algorithms use data coherence to share computations between pixels, while ray casting normally starts the process anew, treating each eye ray separately.
Relatively simple to implement yet yielding impressive visual results.	Other methods, including photon mapping, are based upon ray casting for certain parts of the algorithm, yet give far better results

References

1. F.H. Post, G.M. Nielson, G.P. Bonneau, Data Visualization: The State of the Art, Kluwer Academic. boston., (2002).
2. R.A. Robb, C. Barillo, Interactive display and analysis of 3-D medical images, IEEE. Trans. Med. Imag., 8(3), 217-226 (1989).
3. J. T. Kajiya, The rendering equation, Conf. SIGGRAPH. Proc., 20(4), (1986).
4. H.C. Wong, U.H. Wong, Z.S. Tang, Direct volume rendering by transfer function morphine, Conf. Info. Commun. Signal. Proc ., 1-4 (2009).
5. H. Chu, L. Chen, J. Yong, Feature variation curve guided transfer function design for 3D medical image visualization, Conf. Biomed. Eng. Informatics., 1, 1-445 (2010).
6. C. Correa, k. Liu Ma, The occlusion spectrum for volume visualization and classification, IEEE. Trans. Vis. Comput. Graph., 15(6), (2009).
7. X. Qin, S. Zhu, X. Chen, J. Han, Research and implementation of multi-dimensional transfer function based on boundaries, Conf. Biomed. Eng. Informatics., (2009).
8. C. Shihao, H. Guiqing, H. Chongyang, Rapid texture-based volume rendering, Conf. Envir. Sci. Info. App. Tech., (2009).
9. F. Ling, L. yang, Z.K Wang, Improvement on direct volume rendering, Image. Signal. Proc., (2009).
10. S. Chen, C. Hao, Interactive GPU-based volume rendering for medical image, Conf. Biomed. Eng. Informatics., (2009).
11. J. Kim, J. Jaja, Streaming model based volume ray casting implementation for cell broadband engine, Sci. Program., 17, 173-184 (2009).
12. G. Cox, A. Maximo, C. Bentes, R. Farias, Irregular grid ray casting implementation on the cell broadband engine, proc. IEEE. Comput. Archite. High. Perform. Comput., 93-100 (2009).
13. B. Csébfalvi, B. Domonkos, Frequency-Domain upsampling on a Body-Centered Cubic Lattice for Efficient and High-Quality Volume Rendering, Vision. Model. Vis., 225-232 (2009).
14. F. Gong, H. Wang, An Accelerative ray casting algorithm based on crossing-area technique, Conf. Mach. Vision. Human. mach. Inter., (2010).
15. H. Ray, H. Pfister, D. Silver, T.A. Cook, Ray Casting Architectures for Volume Visualization, IEEE. Trans. Vis. Comput. Graph., 5(3), 210-233 (1999).
16. J. Kruger, R. Westermann, Acceleration Techniques for GPU-based Volume Rendering, IEEE. Vis., 19-24, (2003).
17. C. Barrilot, Surface and Volume Rendering Techniques to Display 3D Data, Proc. IEEE. Eng. Med. Biol. Soc., 12(1), 111-119 (1993).