

Volume Visualization

Intern - Landon Woolley

Mentor - James Money

Objective

During an MRI scan of a human brain, doctors are only able to view data slice by slice with limited three dimensional capability [1].



Figure 1

Implementing volumetric rendering can solve this problem. By rendering the data in real time and combining each slice, doctors can study their data in a three dimensional environment [2]. Viewing the data this way produces a more natural feel and can lead to additional discoveries that would be far more difficult to achieve using a traditional two dimensional plane. Unfortunately, manipulating large amounts of data can be quite taxing for a computers CPU and GPU causing the frames per second to drop dramatically while the processor attempts to handle the vast data. With that being said there are various ways to improve the frame rate and optimize performance.

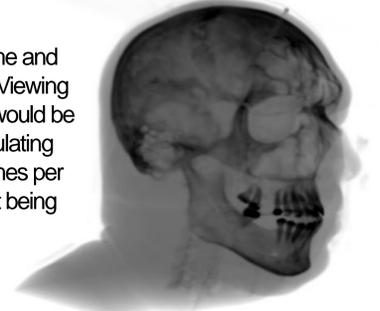


Figure 2

Ray Implementation

Ray casting is the corner stone for volume visualization. It works by creating a ray for every pixel on the users display. Each ray has a point of origin, direction, and length. For volume visualization the ray is not limited to a specific distance in case the data size exceeds expectations. A ray cast will only stop once it has reached an object [3]. However, to achieve a three dimensional volume with varying transparencies the ray must continue through the object once it has successfully made contact. To solve this problem the fragment shader implements a method called ray marching.

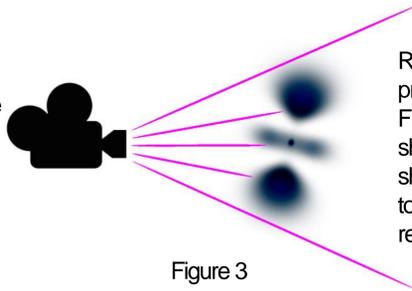


Figure 3

Ray marching allows the ray to hit an object and proceed through it, and continuously return data. From the data that is returned our fragment shader is able to determine the color each pixel should be. The ray cast and ray march work together to gather information. The ray march is represented by the dotted lines in figure 4.

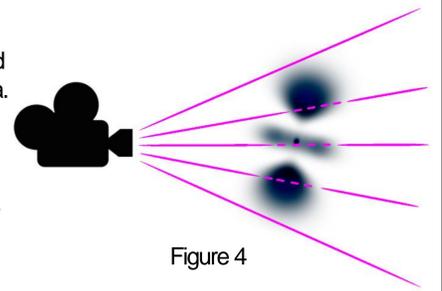


Figure 4

Efficiency Optimization

Detail Restriction

The most detail this software should ever be processing is what the users display can show. By restricting the scale of the data to the scale of the display it is possible to stop unnecessary processing from occurring and therefore increase the performance. Improving the frame rate is a key aspect of the volumetric rendering software. The processing power of a laptop or virtual reality headset can be very limited and any unnecessary computing would be inefficient. Dynamic memory allocation plays an essential role in rendering volumetric data. It permits the software to dump any and all useless data which considerably improves the time it takes to render each individual frame.

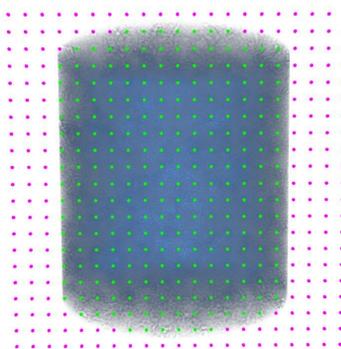


Figure 5

Empty Space Skipping

There is no need to raymarch through the empty space around the volumetric data, otherwise it would be wasting precious computing power by marching through empty space. Fortunately, the raycast has been programmed to ignore the empty space. That, however, does not address the concern of a ray passing through an area of the object where there is no color to render. A noticeable amount of processing power can be saved by telling the ray cast shader to cease ray marching through a ray that is making contact with a part of the object that does not display color. The purple dots in figure 5 represent empty space that the ray cast shader does not ray march through.

Analytical Tools

To promote further discoveries from the data, features such as a transfer function and a clipping plane tool have been added to the volume renderer. These tools allow for more comprehensive and thorough research. Below is an example of the transfer function and clipping planes in action. The transfer function allows the user to see the different layers by assigning them different colors and the transfer function allows the user to slice the data in half.

Clipping plane

Clipping planes allow for a cross-sectional view of the data volume. By modifying the axis that the plane is cutting through the user is able to achieve an unobstructed view of the inside of the volumetric data [7]. It works by stopping all rendering until a certain distance.

Transfer function

The transfer function uses the density data received from the ray march to determine the color and alpha values of each pixel. The ability to adjust the alpha levels of the volumetric data allows the user to see through specific portions of the volumetric data and receive a clearer view of the information that the user is intent on studying [6].

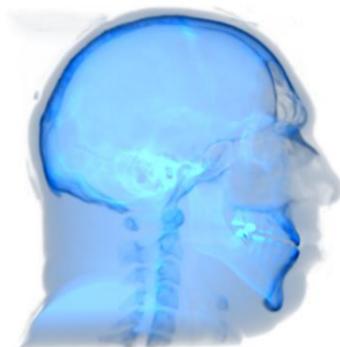


Figure 6



Figure 7

Conclusion

By implementing these methods the software is able to render volumetric data in real time on common hardware without drastic drops in frame rate. The use of ray casting and ray marching allow for pixel perfect rendering and doubles the amount of frames per second the software can render.

