microCAD 2011, 25th International Scientific Conference, University of Miskolc Hungary.
pp 7-12.

PREPROCESSING AND REAL TIME VISUALIZATION OF CT AND MRI DATA

***Csaba Barcsák*[1], *Péter Mileff*[2]**
[1]Bsc Student, [2]PhD,
Department of Information Technology, University of Miskolc

Abstract

The visualization of CT and MRI data has become a large part of scientific computer graphics during the last twenty years. These growing technologies are essential to gain insight to enormous amounts of information, to get better diagnosis. During the last years several different visualization approaches have been developed. In this paper we summarize the benefits and drawbacks of widely used methods, surface rendering and various types of volume rendering. Our aim is to determinate which method is the most effective in different type of situations.

Keywords: Volume rendering, Surface rendering, Marching cubes

1. INTRODUCTION

In the recent years the medical diagnostics have been evolved a lot. During this time the medical image processing has become a dynamically evolving area. New algorithms and methods have been developed. By the development of the computer hardware we can create detailed three dimensional models from CT and MRI slices, and animations from these. These models can help medical doctors to make better diagnosis. They can use these in surgical planning, in surgical simulator programs and in three dimensional anatomy atlases which makes learning of anatomy easier. The more and more detailed visualization of medical images is an important topic nowadays, because making fast diagnosis can save human lives. The continuous development of computer hardware gives new opportunities for the researchers of this area to create more detailed images faster.

In the literature we can meet a wide scale of visualization methods. There can be categorized as surface based and volumetric solutions. Different types of volume rendering methods exist: object order, image order, and hybrid techniques. In this paper we compare a polygon based method with two different volume rendering solutions: an object and an image order methods. The large amount of data can cause long rendering time, therefore we implemented the image order technique using the OpenCL.

2. PREPROCESSING OF THE DATA

The image data from the CT and MRI devices usually needs preprocessing before the visualization. Preprocessing is useful because the quality of the input data can be not satisfying, so it usually needs to improve. The most important topics are contrast enhancement, noise reduction and segmentation. In the literature several different type of methods can be found for these.

For contrast enhancement we can use a histogram based method called histogram equalization [1][2]. This method is useful for images with backgrounds and foregrounds that are both bright and both dark. The contrast enhanced image can be seen in Figure 1. During the digitalization noise can appear in the data. There are different type of noises and different type of noise filtering methods [3]. Commonly used methods in spatial domain are averaging filters, different types of median filters which are good for salt and pepper noise. Before the visualization we can divide the data into multiple segments. This process is called segmentation. By the help of this method we can separate soft tissues, bones and the background.

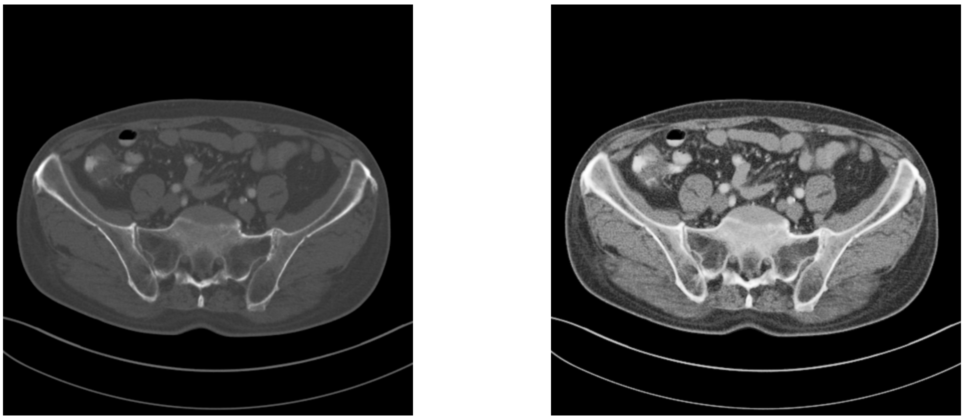

Figure 1. Image with contrast error (left), contrast enhanced image (right)

3. VOLUME RENDERING

In volume rendering [4] objects are typically represented by volume elements called voxels. A voxel can be viewed as a higher dimensional analogy of pixel. In contrast to surface rendering, volume rendering aims to display the data directly as a transparent cloudy object, and the object interiors are always exist.

In general there are three different techniques that create two dimensional image from volumetric data. The first technique considers rays that are cast from each pixel in the image plane into the volume data. Since the rays are casted in order of the pixels, these techniques are detonated as image order techniques. In contrast the object order techniques map the intensities of the volume data to the image plane. There are hybrid techniques which combine image order and object order methods.

3.1 Image order volume rendering

Image order rendering techniques produce images by casting rays through the volume for each pixel and integrating the color and opacity along the ray.

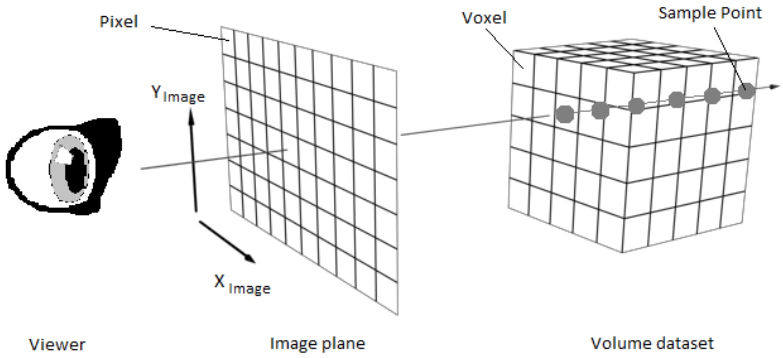

Figure 2. Image order rendering

Figure 2. illustrates the technique. The algorithm iterates over the pixels of the image plane where a parameter iterates over the ray. In every iteration step a sample is taken from the volumetric data. The sample value is determinated using interpolation inside the volume. The final intensity of a pixel is obtained by integrating through the ray using numerical integration.

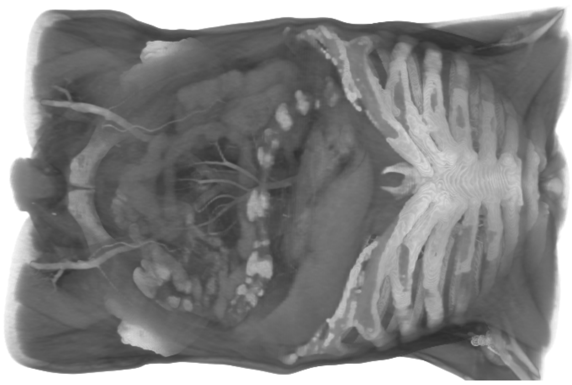

Figure 3. Image rendered from CT data using a GPU-based image order technique

In Figure 3. you can see a rendered image from the upper body of a man. The bones are colored with lighter colors, the soft tissues are colored with darker colors.

The advantage of image order technique that we can create high quality photorealistic images by using this method, moreover it considers those voxels that contribute to the final image and it can be easily turned to parallel execution using the GPU where the entire volume data is stored in a single 3D texture. Furthermore rendering optimizations are possible, for example sparse voxel octree [5], bounding geometry [6], or we can exploit the pixel and voxel space coherency [7]. The drawback of the image order techniques that the calculation of the image is computationally expensive if we don't use the GPU.

3.2 Object order volume rendering

Object order algorithms process the voxels in their memory order by projecting them onto the image plane.

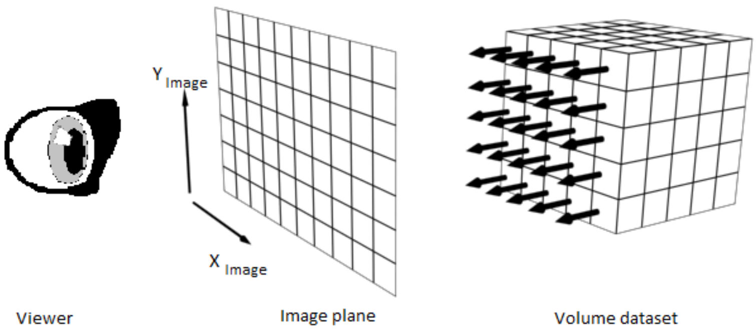

Figure 4. Object order rendering

We have implemented this solution in OpenGL using cubes and point-sprites for representing voxels. Voxels can be represented as different types of geometric objects for example cubes, tetrahedrons, squares. Representing the voxels with cubes is a very simple solution, but we have to calculate the vertices of the cubes and store it, and it takes a lot of memory for big datasets.

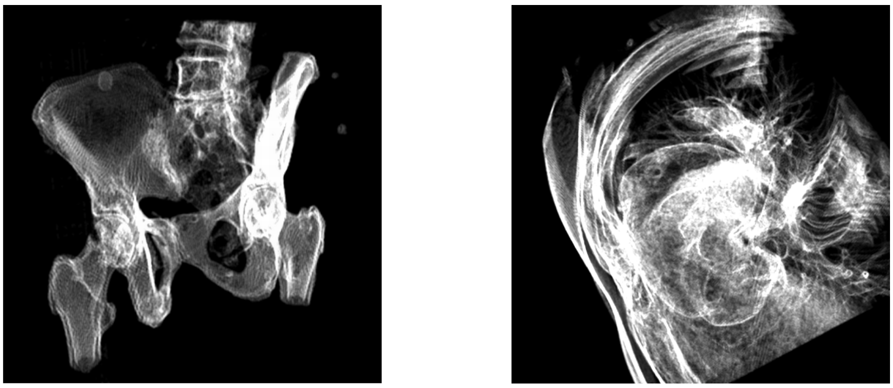

Figure 5. Images created with point-sprite based method, hipbone (left), heart (right).

This is the reason why we have used point-sprites. Point-sprites are two dimensional graphical objects which are facing towards the camera. The point-sprite based rendering is fast, because we only have to store one intensity value per voxel. This method is beneficial in applications where a lot of information has to be visualized, for example in cosmological simulations. The drawback of this method is that the size of the point-sprites does not change automatically when we are moving towards the volumetric model with the camera, so we have to recalculate it if we want better quality of images, which is computationally expensive.

4. SURFACE RENDERING

Surface graphics represents the data with collections of surfaces. Surfaces can be defined by parametric patches, implicit functions or polygons. Using the marching cubes algorithm [8] we can create polygon model from the volumetric data. In the first step the algorithm decides for every voxel if the current voxel is inside the surface or not, in the second step the algorithm finds the boundary of the surface. In order to identify which voxel is inside or outside the surface, we have to define a scalar value.

The algorithm divides the space using cubes, and assigns scalar values to the vertices of the cubes. Taking into account the scalar values at the vertices there are 256 different cube configurations which are come to existence by rotating and reflecting the 15 unique cube configurations.

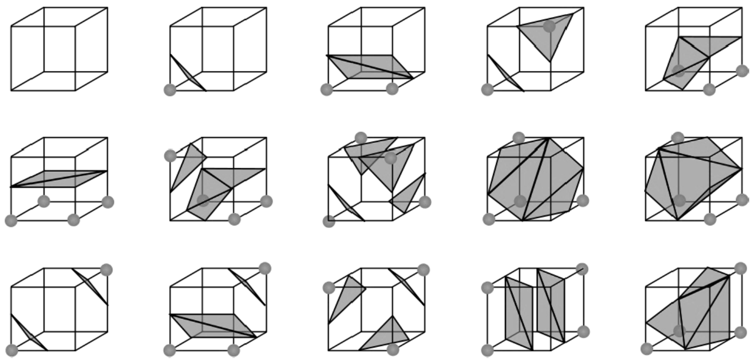

Figure 6. 15 unique cube configurations

The Marching Cubes algorithm calculates the coordinates of the triangle vertices by linear interpolation using the scalar values at the vertices of the cubes. By using the interpolation the algorithm places the vertices of the triangles far from the vertices of the cube which have higher scalar values. Using this algorithm for creating polygon model from CT and MRI data we can only use the color for the interpolation, because we don't store other data from the volumetric model. In this case the algorithm places the vertices of the triangles far if the vertex of the cube has lighter color which don't gives us more information from the dataset, but it's computationally expensive. We have optimized the algorithm by removing this interpolation.

We have mentioned that in the first step the algorithm decides for every voxel if the current voxel is inside the surface or not using a scalar value. If we use segmentation preprocessing technique on the dataset this step can be removed from the algorithm too.

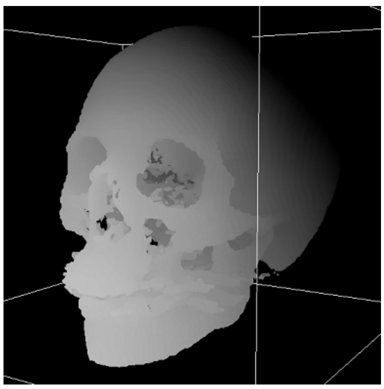

Figure 7. Image rendered from a skull CT using the optimized Marching Cubes algorithm.

The benefit of this algorithm is simplicity, the drawback is that it creates too much triangles, and there are special cases where it gives not satisfying results. Dürst [9] noticed that Marching Cubes creates incorrect geometry when the shared face of two adjacent cells has exactly two outside vertices lying diagonally opposite one another. Payne and Toga [10] have created a marching cubes like algorithm which using tetrahedrons instead of cubes, and don't create incorrect geometry, but more triangles are typically produced. There are other algorithms which can create polygon model from volumetric data for example the ball pivoting algorithm or the Delaunay triangulation.

The benefits of surface based methods are that after creating the model from the volumetric data, the rendering time is very fast, and it takes less memory to create an image than the volumetric solutions. The major limitation of the surface based methods that it can only represent objects using collection of surfaces with zero thickness. In the most cases surface based objects are tessellated with primitives so their interior is completely hidden from the camera. In case of semi-transparent objects the interior is defined homogenously or left undefined.

5. CONCLUSION

In this paper we have summarized the benefits and drawbacks of widely used methods, surface rendering, and two types of volume rendering. The surface rendering is beneficial when we only want to display a surface and we don't want or we don't need to show the interior. Volume rendering is beneficial when we have to show the interior of a graphical object. If we are satisfied with lower image quality than the image order techniques can be used, and we need fast rendering time we should use object order volume rendering solutions. When photorealistic images should be rendered, image order volume rendering methods are sufficient. They are computationally expensive without the GPU. Using the GPU the limitations of image order techniques are starting to disappear nowadays.

6. REFERENCES

[1] Bernd Jahne: **Practical handbook on image processing for scientific and technical applications,** ISBN: 0-8493-1900-5, Crcpress 2005.

[2] Jirí Jan: **Medical image processing, reconstructing and restoration**, ISBN: 978-0-8247-5849-3, Taylor&Francis 2006.

[3] Bernhard Preim, Dirk Bartz: **Visualization in medicine: Theory, Algorithms and Applications**, ISBN: 978-0-12-370596-9, Elsevier 2007.

[4] Klaus Engel, Markus Hadwiger, Joe Kniss, Christof Rezk-Salama, Daniel Weiskopf: **Real-Time volume graphics**, 978-1568812663, A K Peters 2006.

[5] Kristof Römisch: **Sparse voxel octree ray tracing on the GPU**, Master's thesis, Department of Computer Science Aarhus University, Sept 2009.

[6] Markus Hadwiger, Patric Ljung, Christof Rezk-Salama, Timo Ropinski: **Advanced illumination techniques for GPU-based volume raycasting**, SIGGRAPH Asia Course notes, 2009.

[7] André Neubauer: **Cell-based first hit ray casting**, Diplomarbeit, Institut für Computergraphik und Algorithmen Technischen Universität Wien, 2001.

[8] William E. Lorensen, Harvey E. Cline: **Marching cubes: A high resolution 3D surface construction algorithm**, ACM SIGGRAPH Computer Graphics Volume 21, Issue 4, 1987.

[9] M.J. Dürst: **Letters: Additional reference to "marching cubes"**, Computer Graphics, 22(2), pages 72-73, Apr 1988.

[10] B.A. Payne, A.W. Toga: **Surface mapping brain function on 3D models**, Computer Graphics & Applications, Sept 1990.