1998 International Computer Symposium
Workshop on Computer Graphics and Virtual Reality
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

# SEGMETTATION AND VISUALIZATION OF ABDOMINAL CT IMAGES

Yu-Tai Ching
Department of Computer and Information Science
National ChiaoTung University
HsinChu, Taiwan, ROC

## Abstract

One of the purposes of abdominal CT images is for the diagnosis of the diseases of colon. We present a framework for visualization of abdominal CT images. The pipe line of this framework in order is preprocessing (boundary detection), classification of the boundary points, and visualization of the area of interest. Boundary detection is carried through out a 3D Sobel filter to find out the set of boundary points. Classification of the boundary points is to find out the connected components of the boundary points. The area of interest is one of the connected components which is generally the wall in the interior of a colon. Visualization is to render the connected component. We present a method to generate a very wide angle perspective projection image which is to simulate the colonscopy for visualization.

**Key Words:** volume rendering, boundary detection, colonic scope.

## 1 Introduction

Abdominal CT scan has many clinical purposes. One of the purposes is for the diagonis of colonic diseases. The quality of the diagonisis is possible to improve by using the state of art computer techniques.

Volume rendering is a technique for visualization of a set of volume data such as a set of CT or MR images[4, 5]. Volume rendering techniques are generally characterized into two categories. The first category is to model a voxel as a light absober or a glowing object. In the first model, the volume is placed between the light source and the view plane. We calculate the light arrives the view plane after asorbtion. In the second model, each voxel is a light source. We calculate the intensity of a pixel in the view plane by accumulating the light emiting from voxels. The second category is to render the isosurfaces for a given threshold. Marching cube method is one of the approaches in this category. This method approximates the isosurfaces for all the voxels of a given threshold. We then render the polygonal patches. Another approach is to renders the first sampled point along a ray whose associated value is greater than or equal to the given threshold. Implementation for a volume rendering algorithm generally set the view point outside the volume and implement the orthogonal projection.

Consider the case when we need to visualize a set of abdominal CT scan for colon. The area of interest is the wall inside the colon, i.e., the view point should be inside the volume. That means we should simulate colonscopy view in the colon. In this case, we should implement a very wide angle perspective projection[3, 2, 6].

Preprocsssing the volume data is an issue that should be considered in a volume visualization process. Typical tasks in preprocessing are boundary detection and area of interest identification. But this issue was rarely discussed in previous volume rendering literatures. For example, in the marching cube algorithm, we find out the isosurfaces for given threshold(s), but we do not know whether a surface belongs the area of interest or not. Another example is to apply the ray casting algorithm to render the isosurfaces for a given threshold. We shoot a ray into the volume (view point is not in the volume) and

1998 International Computer Symposium
Workshop on Computer Graphics and Virtual Reality
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

sample points along the ray. The ray stops when a sampled point whose associated value is greater than or equal to the given threshold. These algorithms work only when all the isosurfaces for a given threshold are on the area of interest. It is fine to apply this assumption to render skull from a set of CT images of head. To visualize a set of MR images or a of abdominal CT images for colon would need preprocessing steps.

Consider the case of visualization of the interior of colon. The isosurfaces we are looking for must be the boundaries between air (very low intensity) and soft tissue. But not all these boundaries are on the wall inside a colon. Another consideration is to simulate the colonscopy. In order to see as much as we can, we need a very large wide-angle perspective projection in the rendering process. Very often, volume rendering algorithms do not implement perspective projection. Some volume rendering algorithm such as the shear warp algorithm even excludes its capability to generate perspective projection images otherwise it sacrifies its computational efficiency. Other volume rendering algorithms, such as direct ray casting method and the marching cube method, can implement perspective projection. However, since we generally assume the view window is in a plane (so that a point in the view window corresponds to a pixel in the computer monitor), it is hard to implement a very large wide-angle perspective projection. In this work, we present a framework for visualization of a set of abdominal CT images. In the preprocessing step, we present a method which automatically identify the colon. To the second problem, we present a modified ray casting method to generate a very large wide-angle endoscopic view from inside of a colon.

# 2 Preprocessing

The preprocessing step includes identification of a set of boundary points and classification the boundary points.

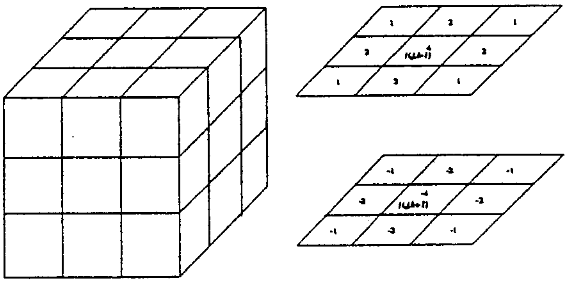

Figure 1: A structure element for 3D Sobel filter which compute the gradient in $z$ direction for voxel $(i, j, k)$

## 2.1 Boundary Detection

CT scan images generally has significant difference in intensity between the boundary of air and soft tissue and the boundary between soft tissue and bone. That is, a simple 3D Sobel filter can determine the boundary points[1].

A Sobel filter calculates gradient at each pixel in an image. Pixels have high gradient is generally a boundary point. The gradient of a voxel $(i, j, k)$ is obtained by summing the weighted intensities of voxels around $(i, j, k)$. For example, the $z$-component of the gradient is obtained by the following equation where the weights are shown in Figure 1.

$$\begin{aligned}
\Delta z \;=\; & I_{(i-1,j-1,k-1)} + 2I_{(i-1,j,k-1)} + \\
& I_{(i-1,j+1,k-1)} + \\
& 2I_{(i,j-1,k-1)} + 4I_{(i,j,k-1)} + \\
& 2I_{(i,j+1,k-1)} + \\
& I_{(i+1,j-1,k-1)} + 2I_{(i+1,j,k-1)} + \\
& I_{(i+1,j+1,k-1)} - \\
& I_{(i-1,j-1,k+1)} - 2I_{(i-1,j,k+1)} - \\
& I_{(i-1,j+1,k+1)} - \\
& 2I_{(i,j-1,k+1)} - 4I_{(i,j,k+1)} - \\
& 2I_{(i,j+1,k+1)} - \\
& I_{(i+1,j-1,k+1)} - 2I_{(i+1,j,k+1)} - \\
& I_{(i+1,j+1,k+1)}.
\end{aligned}$$

The other two components, $\Delta x$ and $\Delta y$, are calculated in a similar way. We generally need the scalar of the gradient which is obtained by summing the absolute values of the three components

$$|\Delta x| + |\Delta y| + |\Delta z|.$$

Taking a threshold of the calculated gradients, we identify a set of boundary points.

### 2.2 Classification

Classification is to characterize the boundary points into clusters. For example, we might want to identify the wall in the interior of colon. The wall in the interior of a colon should be a set of connected boundary points. Two voxels are connected if these voxels are neighboring (one of the differences between their indices is 1). This observation suggests to find find out all the connected components of the boundary points. The colon should be one of the connected components. Note that, this neighboring relationship is symmetric, reflexive, and transitive. The Union-Find operation can efficiently find out all the connected components.

The result of the classified boundary points are stored in an array named **Tag** that

1. each entry in the array contains 0 if the entry is not a boundary point,
2. if the entry is a boundary point, the entry contains an integer $i$ representing that the boundary point is a point in the $i$th largest cluster.

Note that, Tag can be stored in a much more compact manner. However, this is not the main purpose in this paper.

Very often the colon is the second largest cluster, i.e., **Tag**$[i][j][k]$ =2, $(i, j, k)$ is a boundary point of the colon.

## 3 Rendering Process

In this section, we discuss the rendering algorithm. We render the area of interest which is the wall in the interior of a colon. Two cases depending on the view point being inside or outside the colon are discussed in this section. We implement the direct ray casting algorithm to render the surface of a given threshold. Although the direct ray casting rendering method is the least efficient algorithm, it produces the most accurate image comparing to the other volume rendering algorithms.

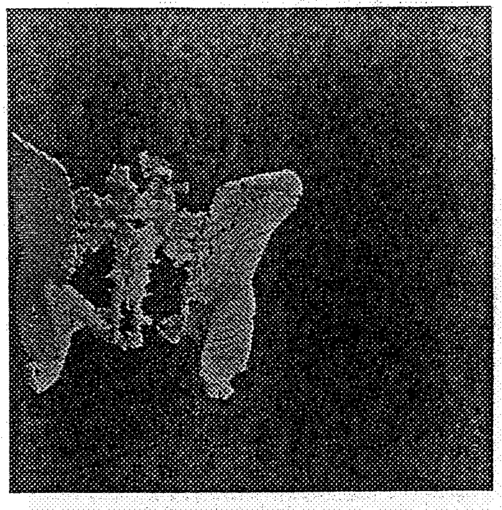

Figure 2: An image is obtained from rendering the largest cluster. We did not invert the intensity since this cluster is a connected component of bone.

### 3.1 Case 1

This case is an ordinary volume rendering process. The only difference is that we suggest to invert the intensity associated with each voxel. Direct ray casting shoots rays into a volume. The condition that a ray stop is generally when the ray encounter a sample point having value greater than or equal to a given threshold. If we would render the wall interior to a colon and the view point is outside the volume, it will be much easier to invert the intensity. Since we have an array Tag containing the information for classified boundary points. The conditions for a ray to stop is

1. the sampled point has value greater than or equal to the given threshold, and
2. the sample point in **Tag** is a boundary point in the cluster representing wall of colon, i.e., **Tag**$[i][j][k]$ = 2 if the colon is the second largest connected component.

Figures (2), (3), and (4) are the images obtained by rendering the three largest clusters.

### 3.2 Case 2

If the view point is inside the colon, the rendering process is to simulate the view of an

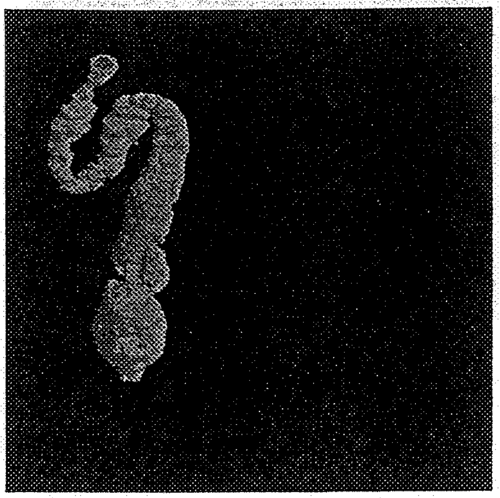

Figure 3: Generally speaking, colon is the second largest cluster.

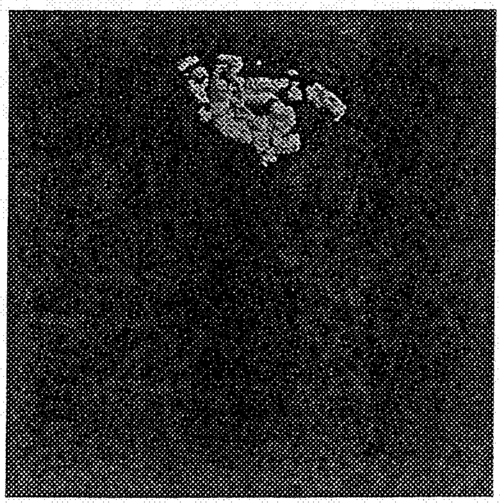

Figure 4: An image for the third largest cluster.

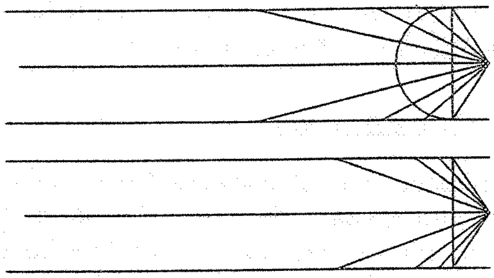

Figure 5: A flat view window and spherical view window.

endoscopic camera. In order to achieve the best visual effect, we need a very wide angle perspective projection. To implement a very wide view angle projection, we propose to use a spherical view window. Figure 5 shows a 2D flat view window and a 2D spherical view window of same wide angle. A flat view window can be thought of as a window that intersections of rays with the window are uniformly distributed over the flat window. A spherical window is a window that a sphere encloses a flat window. The intersections between rays and the sphere are uniformly distributed over the sphere.

The wide-angle depends on the distance between the view point and the flat window. As view point getting closer to the view window, the wide-angle is increasing.

When a flat window is applied, many rays hit objects (wall in the interior of a colon) close to the view plane. This has the zoom effect for the region close to the view plane since we use many pixels to display a small region. Such artifact is more significant if the wide angle is larger. If the spherical view window is applied, the rays hit object in a much uniform manner so that the artifacts are reduced. Figures (6), (7), and (8) are obtained from the different wide-angle (90 degree, 130 degree and 170 degree) but the same view point and view direction. We can see more region as the wide angle increase.

## 4 Conclusion

We present a framework for visualization of a set of CT abdominal volume data. This process includes boundary detection and visual-

Figure 6: Image is obtained when the wide-angle is 90 degree.

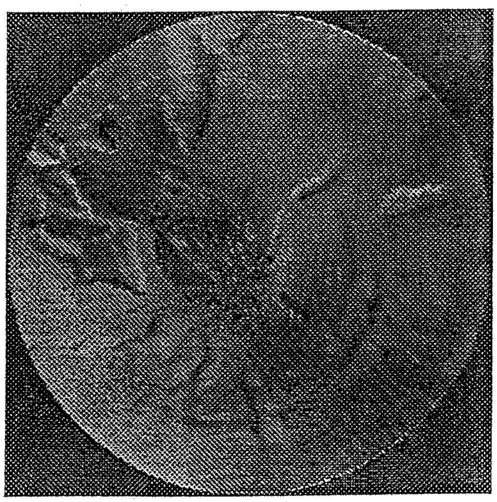

Figure 7: Image is obtained when the wide-angle is 130 degree.

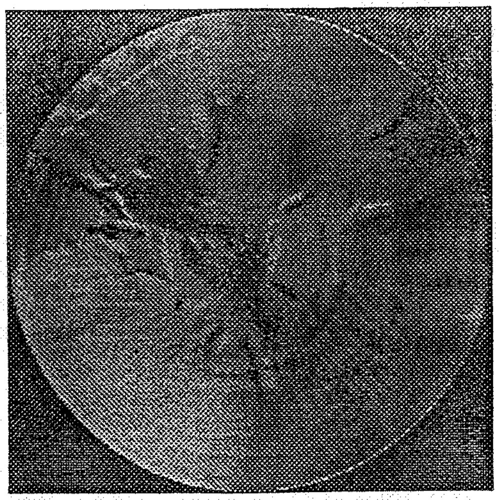

Figure 8: Image is obtained when the wide-angle is 170 degree.

ization of the set of boundary points. We also present a way to produce a very wide angle perspective projection images. This method can be applied to other places where endoscopic views are needed. The spherical view window approach can simulate the colonic scope very well.

Since direct ray casting algorithm is applied, it takes about 15 seconds to generate a 300 by 300 image. (The run time is record from a Pentium II running at clock 266. The operating system is Linux. ) To improve the rendering speed is a future work.

## References

[1] R. Gonzalez, and R. Woods, Digital Image Processing, reading, Addison-Wesley Publishing Company, 1992.

[2] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He, "Vitural Voyage: Interactive Navigation in the Human Colon," *Computer Graphics*. SIGGRAPH 97, Los Angeles, pp. 27-34, 1997.

[3] L. Hong, A. Kauffman, Y. Wei, A. Viswambharan, M. Wax, and Z. Liang, "3D Virtual Colonscopy," *IEEE Symposium on Biomedical Visualization*, pp. 26-32, IEEE. October 1995.

[4] C. Lin and Y. Ching, "An Efficient Volume-Rendering Algorithm with an Analytic Approach," *The Visual Computer*, Vol. 12, pp. 515-526.

[5] W.cLorensen, "Marching-Cube: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, Vol. 21, pp. 163-169.

[6] G. Rubin, C. Beaulieu, V. Argiro, H. Ringl, A. Norbash, J. Feller, M. Dake, R. Jeffey, and S. Napel, "Perspective Volume Rendering of CT and MR Images: Applications for Endoscopic Imaging," *Radiology*, Vol. 199, May 1996, pp. 321-330.