IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 2, February 2015.
www.ijiset.com
ISSN 2348 – 7968

3D Visualization of Medical Image

Pravin P Kalyankar, [1], **Dr.S S Apte** [2]

Assistant Professor, Department of CSE, TPCT COE Osmanabad , India [1]
Professor & Head, Department of CSE, WIT Splapur,, India [2]

Abstract: Two techniques have been chosen to produce the 3D models. The first technique is known as Volume Rendering and the second technique is known as the Marching Cubes Algorithm. Both techniques use voxels (3D square pixels) to determine the 3D area to be constructed. The Marching Cubes (MC) algorithm by Lorensen and Cline is most popular algorithm for extraction of isosurface out of volume data. Several drawbacks of MC algorithm are solved by using new improved version of MC algorithm. We proposed an improved version of the Marching Cubes algorithm which gives a topologically correct triangular approximation of the isosurface for any cube configuration. Unlike the past work on Marching Cube algorithm, a robust triangulation strategy, complementary and rotation operations is presented. Our algorithm is adaptive to the small the changes of data or the small changes of the threshold, and obtains more reasonable result of triangulation of isosurface than those produced by standard MC algorithm. Marching Cubes Algorithm is considered a thresholding technique because it uses only the pixel information at the eight corners of the voxel

Keywords: Marching Cube, topological correct, isosurafce

I. Introduction

The Marching Cubes (MC) algorithm by Lorensen and Cline [1] is the most popular algorithm for the extraction of isosurface out of volume data. Figure 1 shows the Marching cube. Several problems were apparent after the MC algorithm first developed, and many adaptations have been proposed to solve these imperfections. MC algorithm might produce holes in the isosurface. These holes have their origin in what is referred to as the ambiguity problem. Nielson [2] proposed an asymptotic decider over the ambiguous face. Natarajan [3] determines the topology of an isosurface in a cell. We propose a robust and intelligent triangulation strategy and performing complementary and rotational symmetry operation. This is a significant practical improvement compared to the former scheme of isosurface tilings. The 3D images are developed from 2D CT/MRI scans of a particular object of interest. The core of this algorithm is the creation of 3D CT/MRI rendering software. The software will produce a 3D interpretation of 2D CT/MRI data. Initially, the software stacks the 2D data to form a large cube of data. Statistical classification is used to create and label voxels(3D pixels). Finally, the Marching Cubes algorithm is used to extract and create polygon surfaces from the volume Some limited user interaction is permitted such as rotating, scaling and moving the volume.

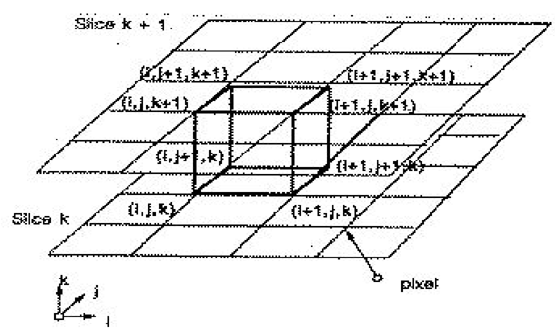

Figure 1: Marching Cube

II. Marching Cubes algorithm

Marching Cubes algorithm uses information at the eight corners of the voxel to determine how a surface cuts through the voxel. Using a threshold (RGB value) each corner of the voxel is determined to be inside or outside of the surface. If the value at a corner is greater than or equal to threshold it is considered to be inside the surface, if it is less than threshold it is considered to be outside the surface. This information is used to create a 3D polygon surface that approximates the original surface. To create a 3D model based on CT/MRI scans we need access to this type of data.

III. . SURFACE CONSTRUCTION

For every voxel a logical cube is formed. This is done by reading in eight neighboring voxels into the eight corners of the cube .Using a threshold determined by the user each corner is determined to be inside or outside of the surface. A surface intersects those cube edges where one vertex is outside the surface and one vertex is inside the surface. With this assumption the topology of the cube can be determined (Lorensen and Cline, 1987). Since there are two states and eight corners there are $2^8 = 256$ ways a surface can intersect a cube. Using complementary and rotational symmetries

Lorensen and Cline(1987) have reduced the intersections to fifteen possible patterns. Figure 2 shows the triangulation for 15 patterns.

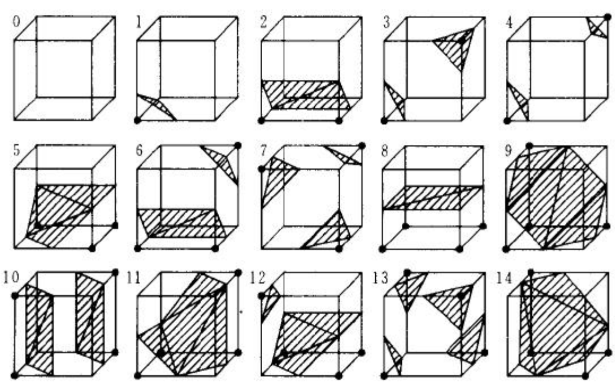

Fig. 1. Triangulated Cubes

The triangles vertices are created by finding intersections between the inside and outside points (corners). Gradient Calculations are used to estimate the normals of the triangle vertices. This is done by estimating the gradient vectors at the point of intersection. Given a voxel at (i,j,k) the gradient is estimated using central differences along the 3 co-ordinates axes by:

Gx(i,j,k) = D(i+1,j,k) - D(i-1,j,k)/ length(x)

Gy(i,j,k) = D(i,j+1,k) - D(i,j-1,k)/ length(y)

Gz(i,j,k) = D(i,j,k+1) - D(i-1,j,k-1)/ length(z)

where D(i,j,k) is the density function (RGB value) at pixel (i,j) in slice k and length(x),(y),(z) are the lengths of the cube edges. However, these algorithms have the following problems. The Marching Cube algorithm might produce holes in the isosurface; these holes have their origin in what is referred to as 'the ambiguity' problem.

IV. . PROPOSED EXPERIMENTAL WORK

The proposed an improved version of the Marching Cube algorithm which gives a topologically correct triangular approximation of the isosurface for any cube configuration. Then, unlike the past work on Marching Cube algorithm, a robust triangulation strategy and complementary and rotation operations is presented. This algorithm is adaptive to the small changes of data or the small changes of the threshold, and obtains more reasonable result of triangulation of isosurface than those produced by standard MC algorithm

GUI (Graphical User Interface) presents the application using graphical icons, visual indicators and many other graphical elements through the help of "widgets". Widgets are interface elements of GUI such as windows, text box, slider, push button, and menu bar and so on. These widgets serve as the basic element of visual blocks, eventually built up as a complete GUI environment. There are several widget toolkits available in various platforms to ease the GUI design and production. Light GUI toolkits for example are OpenGL Utility Toolkit (GLUT) toolkit, OpenGL User Interface Library (GLUI) and Fast Light Toolkit (FLTK). Large GUI toolkits which focus on more professional based software development are WxWidgets and Qt. A program based on OpenGL is successfully implemented. The GUI enables the user to upload the dataset using the file browser as shown in Figure 3, set the cube size for the Marching Cube, on and off the transparency and lighting, also to set the drawing mode. Marching Cube is an algorithm that transforms volumetric 3D object representation to 3D surface representation. The 3D surface is represented in a triangular form. The Marching Cubes module initially reads the volumetric data the user has provided and creates a surface model of a 3D object. The user has ability to read the MRI/CT scan images of various file formats as well as directly shown it on screen. Marching Cubes is a single document interface (SDI) windows application .It is coded in JAVA Programming language. All operations are executed through the menu commands and dialogs which extend the user friendliness of the JAVA software. Each 3D model is displayed in an OpenGL platform inside the applet workspace.

Experimenting with Marching Cubes is the best way to clearly understand how it works and why it's important to use ambiguous case resolution. In this way, the applet gives you the opportunity to view the algorithm running on different cases. Run the applet, here is a description of the controls and the supported functionalities of the applet: the purpose of which is to extract the isosurface from the 3D data-set and the use of 3D mesh reconstruction by applying lighting, transparency and shading using Open Graphics Library (OpenGL) Application Programming Interface (API). Read the input MRI/CT images then the modules for cube generation & isosurface extraction to get the output model which is 3D object. Eventually, a Graphical User Interface (GUI) environment is implemented. In addition to that, user interface such as rotation of the 3D model is shown to provide the basic needs in virtual reality environment. The source code is written in JAVA language. Results are then demonstrated by showing the final snapshots of the application. The development of full interactive 3D virtual reality for use in medical diagnosis and treatment has reached a very productive state. This GUI enables the user to upload the dataset and set the cube size.

V. Conclusion

The software developed works successfully for MC Algorithm. The software uses OpenGL API for GUI environment. Following conclusions are made from the result. The GUI was successfully implemented in the present work. The model constructed provides a very clear visualization which could be enhanced by Lighting, vertex normal shading and transparency. The software is user friendly which enables the user to perform rotation and moving operations of the model using mouse and the keyboard. The MC Algorithm extracts isosurface for volume data and for any case of cube configuration this algorithms do generate topologically exact approximation to the isosurface and improves accuracy. This algorithm resolves the ambiguity which might creates holes in the isosurface & hence this algorithm shows a great potential in medical application.

References

[1] W. Lorensen and H. Cline, "Marching Cubes: A High resolution 3D Surface Reconstruction Algorithm," Computer Graphics, 1987, vol. 21, no. 4, pp: 163-169.

[2] G. M. Nielson, B. Hamann, "The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes". Proc. IEEE Visualization '92, 1992, pp 83-91.

[3] B. Natarajan, "On generating topologically consistent isosurfaces from uniform samples" .The Visual Computer, 1994, 11(1) p: 52–62.

[4] E. Chernyaev, " Marching Cubes 33: Construction of Topologically Correct Isosurfaces." Technical Report CN/95-17, CERN, 1995.

[5] P. Cignoni, F. Ganovelli, C. Montani, R. Scopigno, "Reconstruction of Topologically Correct and Adaptive Trilinear Surfaces". Computers and Graphics, 2000, 24(3) p: 399-418.

[6] G. M. Nielson," On Marching Cube". IEEE Transactions on visualization and computer graphics, 2003, 9(3) p: 283-297.

[7] [7] A. Lopes, K. Brodlie," Improving the Robustness and Accuracy of the Marching Cubes Algorithm for Isosurfacing". IEEE Transactions on Visualization and Computer Graphics, 2003, 9(1) p: 16-29.