3 Dimensional Surface Extraction and Visualization

M. M. Shah, *IEEE R-10 TAC*, and Rinki Gupta

***Abstract*---Volume visualization is the process of projecting a multidimensional dataset onto a two-dimensional image plane for the purpose of gaining an understanding of the structure contained within the volumetric data. It aids in the evaluation of engineering designs, the understanding of large-scale simulations, and the viewing of multi-modal medical data, the mining of enormous web databases, and the analysis of homeland security information. The volume modalities sample the volume for some property and produce multiple 2D slices of information. Most often the dataset is defined on a three-dimensional lattice with one or more scalar values at each grid point on the lattice, such as density and water contents in the case of CT and MRI scans respectively. The surface construction is accordingly carried out by processing each cube in the volumetric image. Marching Cubes is an algorithm for rendering isosurfaces in volumetric data. The feasibility of using marching cubes algorithm is presented through experimental investigation.**

***Keywords*—Iso-surfacing, surface rendering, Marching Cubes algorithm, volume visualization.**

I. Introduction

THERE is a wide range of devices and scientific simulation generating volumetric data. Visualizing such data, ranging from regular data sets to scattered data, is a challenging task. Volume visualization is about understanding complex 3D data. That data may be the result of complex simulations, or the result of a medical scan using CT or MRI. A radiologist is trained to look at 2D cross sectional images, called tomograms, and build up a mental picture of the body in three dimensions. With this image in mind, he can describe to his colleagues the nature of the data and any injuries or trauma the patient may have suffered. A key point is that this picture exists only in the radiologist's head and is not available for everybody to see.

Volume visualization can be used to create a physical image of this data that everybody can see and understand, and of course point to and discuss. Hence, visualization is the process of transforming information into a visual form, enabling users to observe the information. The resulting visual display enables the scientist or engineer to perceive visually features which are hidden in the data but nevertheless are needed for data exploration and analysis.

Iso-contouring methods have been in use for a long period of time, but the major breakthrough in the field occurred with the works by Wyvill et al. [8] in 1986 and Lorensen et al. [1] in 1987, called the Marching Cubes 3D Surface Construction Algorithm. The success of this method stems from its divide-and-conquer strategy. These methods convert the complex global isosurface extraction task into many small and simple local triangulations. The visualized data is often generated or acquired from three-dimensional images or as solutions to numerical approximation techniques, such as from finite difference or finite element methods. These acquisition methods represent the data as a set of polyhedral cells where the data points define the vertices. The surface construction is accordingly processed to each cube in the volumetric image. Divide-and-conquer in this context amounts to replacing the global isosurfacing problem with a local contouring of each of the dataset cells.

II. Volume Data

A visualization technique is used to create and manipulate a graphic representation from a set of data. Transforming CT data into CAD systems offers a lot of potential today and rendering 3D-CAD data set from 3D-tomograms for simulation is even more important. Because of this true object geometry can serve as a base for computation instead of theoretical models. Some techniques will be appropriate only for specific applications while others are more generic and can be used in many applications. Animation is very important for a full understanding of 3D volumetric data and so needs to be a straight forward extension to the fundamental techniques. Volume visualization is used to create images from scalar and vector datasets defined on multiple dimensional grids, i.e., it is the process of projecting a multidimensional (usually 3D) dataset onto a 2D image plane to gain an understanding of the structure contained within the data.

Most often the dataset is defined on a three-dimensional lattice with one or more scalar values, and possibly one or more vector values at each gridpoint on the lattice. Methods for visualizing higher dimensional grids and/or irregular grids are relatively unknown. To be useful, volume visualization

Prof. M. M. Shah is with S.P.College of Engineering, Andheri(w),Mumbai, India. He is Chairman, IEEE, Bombay section, Vice Chairman, students' activities, IEEE India Council, FIE (Ind.), SMIEEE(USA), PE(Ind), C. Engg(Ind.) (e-mail: mmshah@yahoo.co.in).

Rinki Gupta is with Guru Tegh Bahadur Institute of Technology, G-8 Area, Rajouri Garden, New Delhi, India. (Phone: 9511-25998852; e-mail: rgupta_k5@rediffmail.com).

techniques must offer understandable data representations, quick data manipulation, and reasonably fast rendering. Being able to manage a huge volume of data is still a challenge. Volume datasets are often acquired by scanning the material of interest using Magnetic Resonance Imaging (MRI), Computer Aided Tomography (CT), Positron Emission Tomography (PET), and/or Sonogram machines.

Volumes of data are treated as either an array of volume elements (voxels) or an array of cells. These approaches stem from the need to resample the volume between gridpoints during the rendering process [21]. Resampling requiring interpolation occurs in almost every volume visualization algorithm. In the voxel method, the voxel surrounds the central gridpoint and the data value is constant in the voxel. There is no interpolation used and the image may appear chunky. In the cell method, the corners of each cell are the grid points and the values inside the cell are interpolated. Images using this method are smoother than with the voxel method.

III. VOLUME DATA ACQUISITION

Volumetric data can be computed, sampled, or modeled and there are many different areas where volumetric data is available. Medical imaging is one area where volumetric data is frequently generated. Using different scanning techniques, internals of the human body can be acquired using MRI, CT, PET, or ultrasound. Different applications evolved within this area such as cancer detection, visualization of aneurisms, surgical planning, and even real-time monitoring during surgery.

The combination of computerized imaging and medicine has resulted in the use of the techniques of Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), Single Photon Emission Computed Tomography (SPECT) and Ultrasonography, all of which produce volume imaging data. These data may be displayed as acquired, generally as a series of slices oriented in the acquisition plane [21]. This type of display has proven quite useful for diagnostic purposes; it is simple and requires unsophisticated computer resources. Displaying the slices as they were acquired has one serious limitation, the three dimensional relationships between adjacent slices cannot be visualized directly. This limitation is most evident when the information is being used for therapy planning.

A. Computed Tomography

Computed Tomography is probably the most common source of 3 dimensional data. CT scanners are relatively inexpensive, and most hospitals have at least one scanner. CT uses an X-ray radiation source to image the patient. The CT scanner consists of a couch upon which the patient is placed and a circular gantry through which the couch with patient is passed. Within the gantry is a rotating ring with an X-ray source opposed to a linear array of detectors. The X-ray source is collimated so that the X-rays form a flat fan beam with a thickness determined by the user. During the acquisition of a "slice" of data, the source-detector ring is rotated around the patient. The raw output from the detector array is back projected to form an image of the slice of the body. The couch is moved and then another slice is obtained.

The output from a CT scanner is a series of transaxial slices of the patient. Each slice represents a slab of the patients' body with a thickness set by the collimation for the slice (typically 1-10mm)[11]. For most CT scanners each slab has 512 by 512 pixels. The size of a pixel can be varied within certain limits (generally 0.5 to 2 mm). Generally each slice is spaced such that they are either overlapping or contiguous, though some protocols call for gaps between the slices. Each pixel ideally represents the absorption characteristics of the small volume within its bounds. This is measured in Hounsfield units (HU).

Fig. 1. A CT Scanner

Modern CT scanners can generally acquire one slice within 1 to 5 seconds. An entire study of a patient generally represents 30-40 slices, with a study time of 3-15 minutes. The radiation dose from a CT scan is comparable with that of a series of traditional X-rays.

B. Magnetic Resonance Imaging

In Magnetic resonance imaging (MRI, formerly referred to as Nuclear Magnetic Resonance imaging), the study object is placed within a high intensity magnetic field. This causes the magnetic moments of the molecules within the object to become aligned. The object is then irradiated with pulses of low-level microwave radiation (excitation pulses) that cause some of the magnetic moments of the molecules to oscillate and re-emit microwaves after each pulse. These re-emissions are measured and stored digitally. By introducing gradients in the background magnetic field, it is possible to determine the spatial location of a re-emitted microwave. An image representing various characteristics about the molecular emissions at discrete samples throughout the object scanned is then formed. By modifying the frequency and timing characteristics of the excitation pulse, and the delay time before measurement of the emitted energy, it is possible to image particular types of molecules (water for instance), movement (blood flow), and many other characteristics[22].

The output values at each image element are not calibrated to any particular scale. Generally they are 10 bit data samples. The values will vary depending upon the scan parameters, and the patient's size and magnetic characteristics. Additionally, the values are not constant over the entire scan space as inhomogeneity in the magnetic field causes pixels that may represent the same tissue, but located some distance apart to give different signals. This lack of an absolute scale for a dataset is a cause of much consternation to the researcher attempting to segment the MRI data. Modern MRI scanners can generally acquire a set of slices (32 or so) within five minutes. An entire study of a patient generally represents two to three sets of slices, with a study time of 30-45 minutes. Each slice usually represents a thickness of 2-10mm and contains 256 by 256 pixels. Each pixel represents from 1-5mm. The radiation dose from an MRI scan is negligible.

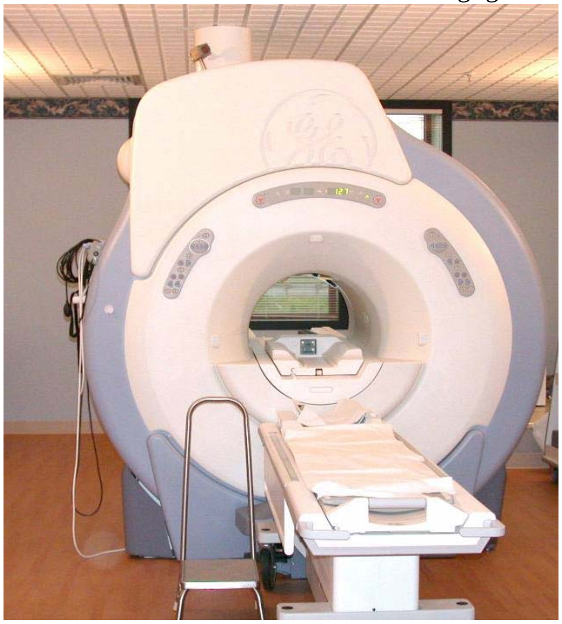


Fig. 2. A MRI Scanner

IV. Volume Visualization Methods

The fundamental volume visualization algorithms [4] are of two types: Direct Volume Rendering (DVR) and Surface-Fitting (SF) Algorithms, as summarized in table 1. DVR algorithms include approaches such as ray-casting, integration methods, splatting, and V-buffer rendering. The two latter methods are sometimes called projection methods. DVR methods are characterized by mapping elements directly into screen space without using geometric primitives as an intermediate representation. DVR methods are especially appropriate for creating images from datasets containing amorphous features like clouds, fluids, and gases. One disadvantage of using DVR methods is that the entire dataset must be traversed each time an image is rendered. A low resolution pass or random sampling of the data is sometimes used to create low-quality images quickly for parameter checking. The process of successively increasing the resolution and quality of a DVR image over time is called progressive refinement.

SF algorithms (sometimes called feature-extraction or iso-surfacing) typically fit (usually planar) surface primitives such as polygons or patches to constant-value contour surfaces in volumetric datasets. The user begins by choosing a threshold value and then geometric primitives are automatically fit to the high contrast contours in the volume that match the threshold. Cells whose comer-values are all above the chosen threshold (cell is inside) or all below the threshold (cell is outside) are discarded and have no effect on the final image. Showing just the cells falling on the threshold is sometimes useful, but can be a problem. Another consideration is the huge number of surface primitives generated for large volumetric data sets. SF methods are typically faster than DVR methods as SF methods only traverse the volume once to extract surfaces.

TABLE I
Volume Visualization Methods

Surface fitting	**Direct Volume Rendering**	
	Projection methods	Image-order methods
Opaque Cubes	V- buffer	Ray-Casting
Contour connecting	Splatting	Cell integration
Marching cubes	Pixar Slice Shearing	Sabella method
Dividing cubes		
Marching tetrahedra		

V. Marching Cubes Algorithm

In table-based cell surface-fitting is performed using polygons that are eventually tesselated into triangles. In a table-based surface-fitting procedure is described that fits up to four triangles to each cell. The algorithm goes by the name *marching cubes. The* marching cubes algorithm has been widely implemented and proceeds by reading four data slices into memory, finding the gradients at all of the interior grid points, marching through all of the interior cells, and then fitting small triangles within each cell through which a threshold-value surface passes. These triangles are then passed to a rendering program that maps them to image space. Before the algorithm begins, the user specifies a threshold value. The algorithm then loops on each successive group of four adjacent data slices. The slices are read into memory, gradients are calculated, and each cell between the middle two slices is scanned to find if its comer-values straddle the threshold value. The eight comers of the cube are numbered one through eight and valued "1" if they are above the threshold and "0" if they are below the threshold. The eight values are then put in eight consecutive bit locations (0-7) to form an eight bit byte. This byte is treated as an index into a precomputed edge intersection table, indicating which of the 12 edges of the cell are intersected by the iso-surface. Interpolation is used to find where the edge is intersected by the iso-surface. There are exactly 256 ways that four or less triangles can be fit to a cell, and the number of cases can be reduced to 15 by reflection and rotation. Groups of three cell-edge intersection points are grouped to form triangles [3].

The gradient is the first derivative of the volume. If I(i,j,k) represents the intensity at the voxel point (i,j,k), then the gradient at that point, G(i,j,k) is given by:

$$G_x(i,j,k) = \frac{I(i+1,j,k) - D(i-1,j,k)}{\Delta x}$$
$$G_y(i,j,k) = \frac{I(i,j+1,k) - D(i,j-1,k)}{\Delta y} \qquad (1)$$
$$G_z(i,j,k) = \frac{I(i,j,k+1) - D(i,j,k-1)}{\Delta z}$$

where, $\Delta x, \Delta y, \Delta z$ are the lengths of the cube edges and D (i, j, k) is the density at pixel (i,j) in slice k.

Dividing the gradient by its length produces the unit normal at the vertex required for rendering. To find the density at the triangle vertices, first the gradient at the two voxel vertices closest to the triangle vertex is calculated and then they are linearly interpolated to find the gradient at the triangle vertex located on the edge connecting the two. To find the gradient at the voxel vertices, tri-linear interpolation is used. The interpolated gradients are stored with the triangles and later used for shading.

For the iso-surfaces, a histogram is generated [3] using the values of the voxels closest to the zero-crossing points (since a zero-crossing point is generally not at a grid point, but obtained through interpolation on a voxel edge). A sequence of 1D Gaussian filters with increasing σ values is applied to the histogram curve to generate the iso-values in a scale space.

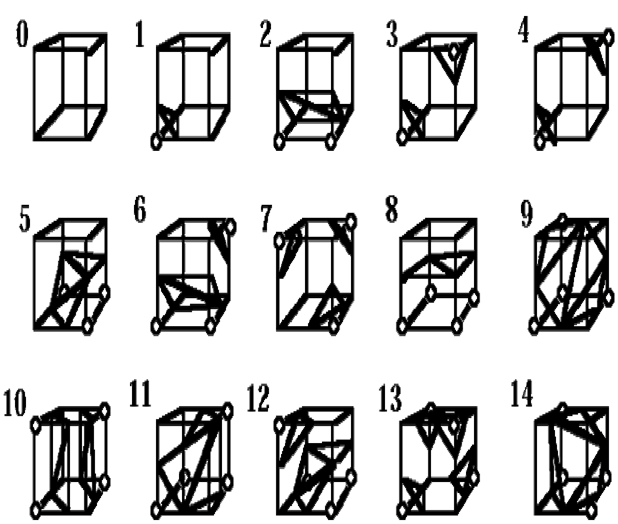


Fig. 3. 256 Cases Reduce to 15

VI. RESULTS

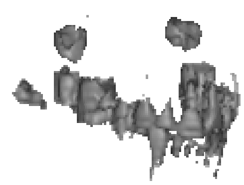
The technique described above has been carried out using C++ and OpenGL under MS Visual Studio. The datasets were structured as regular grids with orthogonal axes and grid spacing that do not change along the axes. Hence straight forward application of a uniform Laplacian of Gaussian was possible as the consecutive points of the data and that of the filter had direct correspondence. For each dataset, a zero-crossing field is first generated, which will then be used to extract the iso-surfaces as unstructured meshes. Users can selectively render a portion of this polygon mesh surface with any given intensity ranges. The volume datasets tested using the above described approaches are

- CT scan of human male, head and neck, of size 205×233×251
- CT Scan of a Two Cylinder Engine, of size 256×256×110

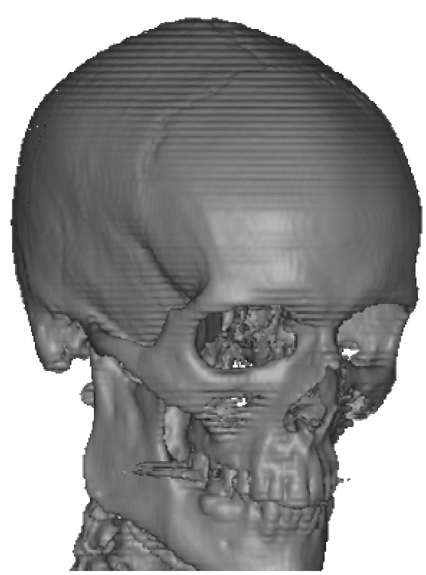
TABLE II
SCALE SPACE OF THE ISO-VALUES FOR CT SCAN OF HUMAN HEAD

σ	Peaks Detected at following Intensity Values
0.1	56 59 65 97 106 111 203 212 216 219 234 237 240
0.7	56 65 98 104 106 199 203 210 216 219 225 240
1.3	65 98 106 204 213 232 241 254
1.7	65 98 106 200 202 213 226 250 254
2.1	65 106 196 226 240 254
2.3	**65 106** 193 242 248 254 Frequency of occurrence: **76332, 1466**, 21, 4, 4, 16

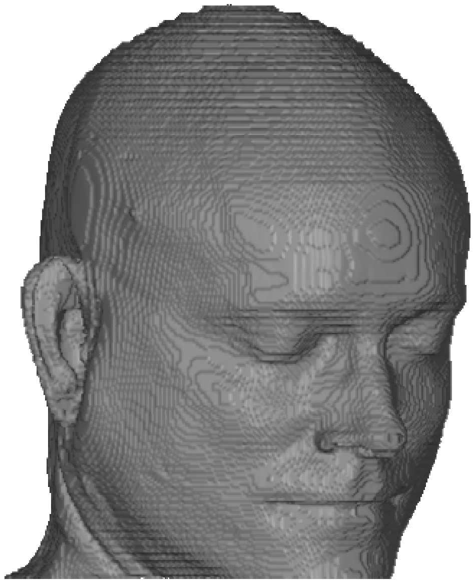
When looking at the scale-space generated by the histogram analysis technique in table 2, one can notice that it does exhibit consistent hierarchical structure as expected, with iso-values gradually disappearing as σ increases.



(a)
Isovalue=186



(b)
Isovalue=105



(c)

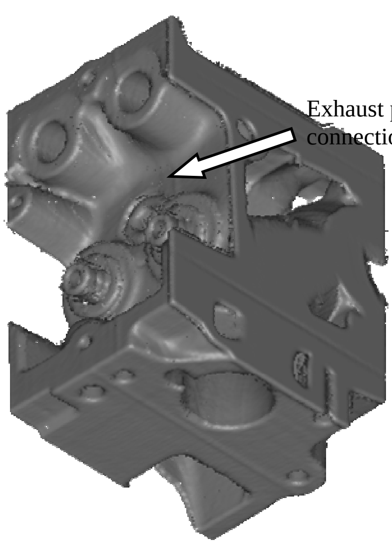
Isovalue=65

Fig. 4. Iso-surfaces of Human CT head dataset with respective isovalues

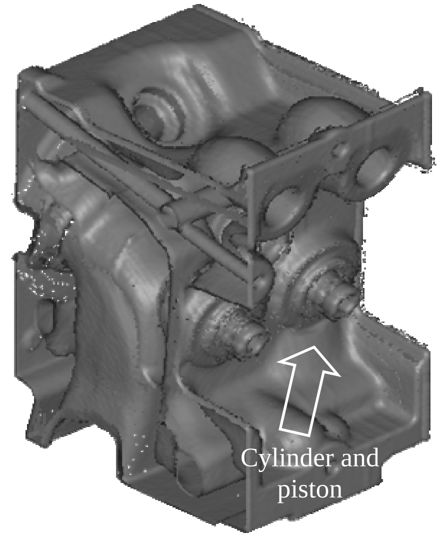
While not all detected iso-values in the scale-space correspond clearly to boundary surfaces; major boundary surfaces, such as the skin, skull and teeth, are indeed detected by this process.

Figure 4 show iso-surfaces of a human CT. Prominent surfaces in the datasets have been extracted, like the skin surface, the skull surface and the teeth surface, by using the respective iso-values obtained from analysis of the scale-space map.

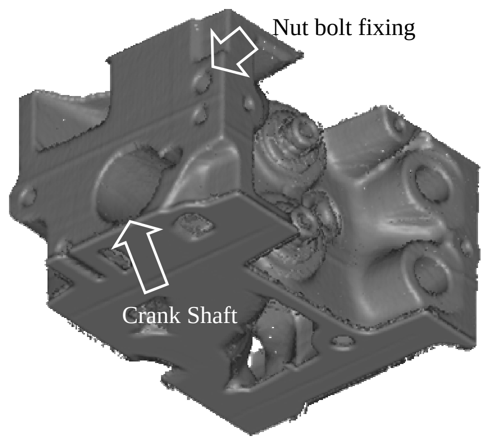
Figure 5 gives the iso-surfaces of the same dataset. The dataset was treated with a Gaussian of $\sigma = 0.7$ of size 5×5×5. Figure 5 show iso-surfaces of a two cylinder engine. The OpenGL [19] libraries used aid in making the implemented system highly interactive.



(a)



(b)



(c)

Fig. 5. Different views of surface of a two cylinder engine casting as extracted from the 3D CT Scan data

Table 3 shows scale space of the iso-values for the two cylinder CT dataset.

TABLE III
SCALE SPACE OF
THE ISO-VALUES FOR CT SCAN OF TWO CYLINDER ENGINE

σ	Peaks Detected at following Intensity Values
0.1	6 53 58 61 79 87 91 97 101 111 114 142 174 183 192 198 204 216 228 234 245 251
0.7	6 58 61 63 79 91 101 142 174 183 190 198 216 234
1.3	6 62 74 79 91 142 175 184 190 198 216 254
1.7	6 62 74 142 175 190 198 215 254
2.1	6 62 75 142 175 184 191 198 216 254
2.3	6 62 74 91 **142** 176 191 198 216 254 Frequency: 4103, 434, 418, 440, **5479**, 77, 70, 326, 202, 242

VII. Conclusion

A general technique for extracting and visualizing boundary surfaces in volumetric image data is presented. Volume rendering is a key technology with increasing importance for the visualization of 3D sampled, computed, or modeled datasets. This paper gives a practical method to visualize boundary surfaces in polygon mesh form. The surfaces extracted and visualized are more natural representations of object boundaries, hence are often more interesting for applications as opposed to the two dimensional contour surfaces or the engineering drawing layouts, which give information that only trained eyes can decipher.

The dataset can be studied from different viewpoints. Such visualization of the information aids in better understanding of structure, eases rectification of faults as it can be shared with personnel belonging to areas other than design engineering. The technique presented here can be applied to any dataset that has layers of various materials with clear transitions from one material to the next and of course there should be the need to extract and study the boundaries. So it holds tremendous potential to bring a revolution in the medical imaging field where it may be used to visualize surfaces in medical three dimensional data such as CT scan and MRI scan of human body.

VIII. References

Periodicals:

[1] Lorensen, W. E. and Cline, H. E., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", Computer Graphics, vol. 21, no. 3, pp. 163-169, (Proc. of SIGGRAPH), July 1987

[2] Schroeder, W. J., Zarge, J., and Lorensen, W. E., "Decimation of Triangle Meshes", Computer Graphics, vol. 26, no. 2, pp. 65-70, August 1992

[3] Shiaofen Fang and Marwan Adada, "Multi-Scale Iso-Surface Extraction for Volume Visualization", ACM, 2004

[4] B. Dogdas, D. Shattuck, and R. M. Leahy, "Segmentation of Skull in 3D Human MR Images Using Mathematical Morphology"

[5] National Library of Medicine. The Visible Human Project, 1986

[6] T. Todd Elvins, Advanced Scientific Visualization Laboratory, San Diego Supercomputer Center , " A Survey of Algorithms for Volume Visualization", Computer Graphics, Volume 26, Number 3, Page 194, August 1992

[7] Benassarou, E. Bittar, N. W. John, L. Lucas, "MC Slicing for Volume Rendering Applications", School of Informatics, University of Wales, Bangor, United Kingdom

[8] Feng Dong, Gordon J. Clapworthy and Mel Krokos, Department of Computer & Information Sciences, De Montfort University, U.K., "Volume Rendering of Fine Details Within Medical Data", Presented at IEEE Visualization 2001, October 21 - October 26, 2001, San Diego Paradise Point Resort, San Diego

[9] Gregory M. Nielson, Senior Member, IEEE, "On Marching Cubes", IEEE Transactions on Visualization and Computer Graphics, VOL. 9, NO. 3, page 283, JULY-SEPTEMBER 2003

[10] Bill Lorensen, "Marching Through the Visible Man", GE Corporate Research and Development

[11] M. R. Stytz, G. Frieder, 0. Frieder, "Three-Dimensional Medical Imaging: Algorithms and Computer Systems", ACM Computing Surveys, Vol 23, No. 4, December 1991

[12] H. Shu , L. Luo , D. Boulmier, "A 3D Static Heart Model From a MSCT Data Set", Proceedings of the 2005 IEEE, Engineering in Medicine and Biology 27th Annual Conference Shanghai, China, September 1-4, 2005

[13] Xia Renbo, Liu Weijun, Wang Yuechao, Shanghai, China, September 1-4, 2005, "A Robust and Topological Correct Marching Cube Algorithm Without Look-Up Table", Proceedings of the 2005 The Fifth International Conference on Computer and Information Technology (CIT'05), 2005 IEEE

[14] Jianping Li Dr. Pan Agathoklis, "A Case Study of Isosurface Generation in 3D Visualization", IEEE, 1993

[15] S. Parker, P. Shirley, Y. Livnat, C. Hanson, P. Sloan. Interactive Ray Tracing for Isosurface Extraction IEEE Proceesings Visualization '98, pp. 233-238, Oct 1998

[16] Yarden Livnat, "NOISE, WISE and SAGE: Algorithms for Rapid Isosurface Generation", Scientific Computing and Imaging Institute, University of Utah, December, 1999

[17] Wyvill, G., Mcpheeters, C., and Wyvill, B., "Data structure for soft objects", The Visual Computer 2, 1986, 227–234

[18] Yarden Livnat, Xavier Tricoche, "Interactive Point-Based Isosurface Extraction", Scientific Computing and Imaging Institute, University of Utah

Books:

[19] The Redbook, OpenGL Programming Guide, Second Edition, OpenGL Architecture Review Board, Addison- Wesley Publishing Company

[20] F. S. Hill, Jr., "Computer Graphics using OpenGL", Second Edition, Pearson Education

[21] R. S. Khandpur, "Handbook of Biomedical Instrumentation", Second Edition, Tata McGraw- Hill Publishing company Ltd.

[22] John G. Webster, "Medical instrumentation- Application and design", Third Edition, John Wiley & Sons, INC.

[23] Gonzalez C., Woods E., "Digital Image Processing", Second edition, Pearson Education

Papers Presented at Conferences (Unpublished):

[24] Gupta Rinki, "Volume Visualization by Iso-Surfacing", The Institution of Engineers (India), 23rd National Convention, 2007.

IX. Biographies

[1]M.MShah is currently Academic Adivisor in S. P. College of Engineering (unaided), Andheri(w) Mumbai-58. He is Chairman of IEEE Mumbai section and Vice Chairman of IEEE India Council. His areas of interest include Multimedia, Machine Vision and VLSI. He has written three books in Electronics Engineering besides having many national and international papers at his credit.

[2]Rinki Gupta is currently Lecturer in Guru Tegh Bahadur Institute of Technology, G-8 Area, Rajouri Garden, New Delhi, India. She is associate member of IETE. Her areas of interest include biomedical electronics digital image processing and volume visualization. She has presented three papers in National Conferences.