3-D IMAGING USING NORMALIZED GRADIENT SHADING

C. R. Crawford, E. Santos, I. D. Weinstein,
B. C. Teeter, H. E. Cline and W. E. Lorensen

GE Medical Systems, P.O. Box 414, W-875, Milwaukee, WI 53201

ABSTRACT

In previous work [3-4] two algorithms were developed called Marching Cubes and Dividing Cubes for the three-dimensional display of objects contained in CT and MRI images. A key to these algorithms is that surface normals are derived from the normalized gradient of the original tomographic images. The resulting images have been subjectively judged superior to images generated with the Cuberille and Ray-casting algorithms. This paper shows how the Cuberille and Ray-casting algorithms can be extended to use the normalized gradient. The image quality attainable using the four existing and the two extended algorithms will be demonstrated and the implementation trade-offs for the algorithms will be discussed.

INTRODUCTION

This paper presents the comparison of six methods for three-dimensional (3D) surface reconstruction with respect to image quality and ease of implementation. The methods of Ray-casting [1], Cuberille [2], Marching Cubes [3-4], Dividing Cubes [4], extended Ray-casting [5], and extended Cuberille [6] have been implemented to view complex anatomy contained in a set of contiguous CT or MRI tomograms. The Ray-casting algorithm generates images based on the distance from an imaginary observation point to a patch on the surface. The Cuberille method renders images based on the surface normal of the patch obtained using the relationship of the orientation of the patch to its neighboring patches. The methods of Marching Cubes and Dividing Cubes produce images using surface normals obtained from the normalized gradient of the original values in the CT or MRI tomograms. It will be shown how the Ray-casting and Cuberille methods can be extended to utilize the normalized gradient. The image quality attainable using the four existing and the two extended algorithms will be demonstrated and the implementation trade-offs for the algorithms will be discussed.

DESCRIPTION OF ALGORITHMS

The basic principles of Marching Cubes will now be explained. The input to a 3D reconstruction algorithm is a series of contiguous cross-sectional images. Consider four square-adjacent pixels on a slice and the four parallel square-adjacent pixels on an adjacent slice. The pixels are said to be cubically adjacent and can be placed on the vertices of a cube yielding a marching cube. For each of the eight vertices, a check is made to see if the vertex is inside or outside the object. Because the decision is binary, there are 256 unique cases. For each case, from zero to four triangles are fitted into the cubical region. Using symmetry considerations, it can be shown that there are 15 unique cases for the placement of the triangles. Images are rendered from the triangles using conventional computer graphics methods. Computer graphics requires surface normals at each triangle vertex. We have found that the normalized gradient of the original tomographic data provides an excellent estimate of the surface normal. The components of the gradient can be found using difference equations.

In a typical application of Marching Cubes, each triangle projects to a small number of pixels. In these cases, the pre-processing time for scan-conversion of the triangles makes the image rendering step computationally expensive. The Dividing Cubes algorithm overcomes this limitation. The key to Dividing Cubes is that if the triangles produced with Marching Cubes are small enough, then they will project to exactly one pixel. Thus, the triangles can be represented by single points; no scan conversion is required; and image rendering is computationally tolerable. The condition of small triangles can be met if the original 2D slices are sufficiently inter- and intra-slice interpolated into sub-cubes. Surface normals are obtained from the normalized gradient of the tomographic data. For subcubes through which the surface passes, Dividing Cubes will output the location of the cube along with the value of the normalized gradient.

With the Cuberille method of 3D image generation voxels that are part of the object are identified. Then the visible faces of these voxels are extracted to define the surface. A surface normal, based on the context of a face with respect to its neighbors, is then determined. Finally, images are rendered using conventional computer graphics. We have found that improved image quality can be obtained if the surface normal is replaced with the normalized gradient of the tomographic data. The modified Cuberille algorithm is denoted the Cuberille' method.

With the Ray-casting method of 3D image generation, rays are cast from a 2D raster into the set of slices. The rays are typically perpendicular to two of the three major axes on which the images are sampled. Casting along a ray is terminated when an object voxel is found. The raster pixel is assigned the distance from it to the object point. We have found that improved image quality can be obtained if the raster pixel is replaced with the component of the normalized gradient of the tomographic data parallel to the viewing direction. The modified Ray-casting algorithm is denoted the Ray-casting' method.

ALGORITHM COMPARISON

The final goal of 3D image generation is a synthesized image that is indistinguishable from a photograph of the object of interest. If the goal is not met, then measurement of 3D image quality is very subjective. Using our own subjective criteria, we evaluated the image quality of the six algorithms. In general, the methods of Marching Cubes, Dividing Cubes and Cuberille' produce the best images.

Ray-casting and Ray-casting' have the advantage that there is no intermediate surface required. Among the three polygon based methods, Cuberille and Cuberille' are the best because only three polygons have to be scan-converted. Dividing Cubes does not require scan-conversion. Efficient implementations of Ray-casting and Ray-casting' are limited to viewing directions perpendicular to two of the three orthogonal axes of the patient. With Marching Cubes, scan conversion is expensive. Dividing Cubes does not allow object zoom.

IMPLEMENTATION

After examination of the advantages and disadvantages of the six algorithms we determined that Dividing Cubes was the best algorithm. We found that an interactive version of Dividing Cubes could be implemented using the image reconstructor found in the GE Medical System's CT 9800 Quick. Typical surface extraction times are between three and five minutes. The time for image rendering is typically five seconds.

CONCLUSION

We presented two new 3D reconstruction algorithms called Marching and Dividing Cubes. The key to these methods is the use of the normalized gradient of the original tomograms as an estimate of surface normals. We showed how the Cuberille and Ray-casting methods could be extended to use gradient normals for improved image quality. Finally, we discussed the implementation advantages and disadvantages of the various algorithms.

REFERENCES

[1] M. W. Vannier, J. L. Marsh and J. O. Warren, "Three-dimensional graphics for craniofacial surgical planning and evaluation," Computer Graphics, Vol. 17, pp. 263-273, July, 1983.

[2] G. T. Herman and H. K. Liu, "Three-dimensional display of human organs from computed tomograms," Computer graphics and image processing, Vol. 9, pp. 1-21, January, 1979.

[3] W. E. Lorensen, and H. E. Cline, "Marching Cubes: a high resolution 3D surface construction algorithm," Computer Graphics, Vol. 21, pp. 163-169, July, 1987.

[4] H. E. Cline, W. E. Lorensen, S. Ludke, C. R. Crawford and B. C. Teeter, "High Resolution 3D Reconstruction of Tomograms," accepted for publication in Medical Physics.

[5] I. D. Weinstein, "Algorithms for the Display of Surfaces in Three Dimensional Medical Computerized Tomography Data," Electrical Engineering Thesis, Marquette University, May, 1986.

[6] E. Santos, "Implementation of three-dimensional display algorithms for the display of three-dimensional computerized tomographic images on a large minicomputer," Electrical Engineering Thesis, Marquette University, July, 1986.