Real-Time Volume Rendering of Four-Dimensional Images Based on Three-Dimensional Texture Mapping

Jinwoo Hwang, June Sic Kim, Jae Seok Kim, In Young Kim, and Sun I. Kim

A four-dimensional (4-D) image consists of three-dimensional (3-D) volume data that varies with time. It is used to express a deforming or moving object in virtual surgery or 4-D ultrasound. It is difficult to obtain 4-D images by conventional ray-casting or shear-warp factorization methods because of their time-consuming rendering process and the pre-processing stage necessary whenever the volume data are changed. Even when 3-D texture mapping is used, repeated volume loading is time-consuming in 4-D image rendering. In this study, we propose a method to reduce data loading time using coherence between currently loaded volume and previously loaded volume in order to achieve real-time rendering based on 3-D texture mapping. Volume data are divided into small bricks and each brick being loaded is tested for similarity to one that was already loaded in memory. If the brick passes the test, it is defined as 3-D texture by OpenGL functions. Later, the texture slices of the brick are mapped into polygons and blended by OpenGL blending functions. All bricks undergo this test. Using continuous deforming, 50 volumes are rendered in interactive time with SGI ONYX. Real-time volume rendering based on 3-D texture mapping is currently available for personal computers.

Copyright © 2001 by W.B. Saunders Company

A FOUR-DIMENSIONAL (4-D) image consists of three-dimensional (3-D) volume data that varies with time. With the advance of medical imaging scanners, 4-D images, such as 4-D cardiac images, are quite common in medical imaging.¹ Volume rendering is a powerful technique to visualize 3-D data. However, it takes too much time to render. Recent advances in graphic hardware and rendering algorithm make it possible to visualize 3-D data in real time. However, when it comes to the 4-D image, difficulties remain in visualizing

Table 1. Rendering Time Measured Under the Various Conditions

	Sampling Rate (voxel/voxel)	
	1/125	1/1,000
Deformation		
Locally	15.5 s (3.2 frame/s)	10.9 s (4.6 frame/s)
Globally	167.2 s (0.3 frame/s)	125.6 s (0.4 frame/s)

the 4-D image because of the enormous size of data. Therefore, 4-D rendering algorithms should be capable of processing a very large amount of data while preserving image quality.

The ray-casting method, the representative volume rendering method, can produce high-quality images. However, it is still time-consuming to render the 4-D image, even when various acceleration algorithms is used. The shear-warp factorization method can produce relatively good-quality images at high speed. It does require a time-consuming pre-processing process whenever the volume data is changed. To render 4-D images in real time, we used the volume rendering method using 3-D texture mapping supported by OpenGL1.2 (Silicon Graphics, CA). This method produces good-quality images in real time without special pre-processing.

VOLUME RENDERING USING 3-D TEXTURE MAPPING

The 3-D texture map is defined and rendered by mapping it to a stack of polygon slices. This 3-D texture mapping provides high-quality as well as high-speed volume rendering with hardware acceleration.

Building 3-D Texture

To map 3-D texture data to polygons orthogonal to the viewing direction, the 3-D texture must be defined before rendering. The OpenGL function defines volumetric data as a 3-D texture map. Normal volumetric data is too large to be loaded into the texture memory at once. Therefore, volumetric data is divided into multiple bricks that are small enough fit into the texture memory and each small brick is defined as a 3-D texture map.²

From the Department of Biomedical Engineering, Hanyang University, Seoul, Korea.

Supported by Ministry of Commerce, Industry and Energy, Korea.

Address reprint requests to Sun I. Kim, Department of Biomedical Engineering, College of Medicine, Hanyang University, Sungdong PO Box 55, Seoul, 133-605, Korea. E-mail: sunkim@email.hanyang.ac.kr.

Copyright © 2001 by W.B. Saunders Company

0897-1889/01/1402-1056\$35.00/0

doi:10.1053/jdim.2001.23873

Fig 1. Continuously deforming 3-D volume data with time.

Compositing the Textured Slices

After defining the texture bricks, each 3-D texture is mapped to a stack of polygon slices perpendicular to the viewing direction and then the textured polygon slices are composited by various blending functions such as front-to-back, back-to-front, and maximum intensity projection (MIP).^{2,3} Among the various blending functions, front-to-back is frequently used.

SIMILARITY TEST

Sampling

To test similarity between the volume to be loaded and that already loaded, the entire volume data is uniformly under-sampled. The Intensity value of under-sampled voxels is used to determine whether a brick has been changed. Two sampling rates were used for our experiment: at intervals of 5 voxels (sampling rate: 1/125) and 10 voxels (sampling rate: 1/1,000).

Similarity Test

In order to load only changed part of the volume, the similarity between the previously loaded volume and the volume to be loaded is measured. Each brick is tested for the similarity. Only the bricks that pass the test are loaded into texture memory and rendered.

Fig 2. Deformation from 20th volume to 30th volume.

To quantify the similarity, cross-correlation between the under-sampled values of the two volumes is used. The correlation result is then normalized by the value given by:

$$\frac{1}{N} \left[\sum_{n=0}^{N-1} x_1^2(n) \sum_{n=0}^{N-1} x_2^2(n) \right]^{1/2}$$

to standardize the results. If the resulted value of one brick is smaller than the threshold value, the brick passes the test. The threshold value is always less than 1.0; the value 1.0 means two bricks are exactly the same.

RESULTS

Continuously deforming 50-volume sets were created from one 256*256*256 magnetic resonance image to simulate a 4-D image ($n = 1, 2, \dots, 50$, n : sequence of the volume data) (Fig 1). The rendering time was measured in the various conditions (Table 1, Figs 2 and 3). SGI ONYX (Silicon Graphics) with two 180-MHz R10000 microprocessors; 128 MB memory was used for this simulation.

CONCLUSION

With this method, 4-D volume data was rendered in interactive time with high image quality. The results show that this method provides better performance when the volume data varies locally with time. Globally changing of other data sets was also tested under the same conditions. When the volume data varies globally, the rendering performance is decreased remarkably. The sampling rate also had a significant influence on rendering speed and quality.

Volume rendering based on 3-D texture mapping is currently available for the personal computer. With the proposed method, a real-time 4-D rendering system can be implemented on personal computer.

Fig 3. Cross-correlation results between 20th and 30th volume under the condition sampling rate 1/125 and 1/1,000. The threshold was set up to 0.95. In this case, the 16th and 17th bricks are newly loaded and rendered.

REFERENCES

1. Thomas CF, White JG: Four-dimensional imaging—The exploration of space and time. *Trends Biotechnol* 16:4.1, 1998
2. Van Gelder A, Kim K: Direct volume rendering with shading via three-dimensional textures. *Proceedings of IEEE Volume Visualization '96*. IEEE, October 1996, pp 23-30, 98
3. Dachille F, Kreeger K, Chen B, et al: High-quality volume rendering using texture mapping hardware. *Workshop on Graphics Hardware*, Lisbon, Portugal, August 1998, pp 69-76