

하드웨어 텍스처 매핑을 이용한 의료영상의 3 차원 볼륨 렌더링 기법

손재기*, 전준철*

*경기대학교 전자계산학과

e-mail : jgson@kuic.kyonggi.ac.kr, jcchun@kuic.kyonggi.ac.kr

3D Rendering Technique for Medical Images using Hardware Texture Mapping

Jea Gi Son*, JunChul Chun*

*Dept. of Computer Science, Kyong-gi University

요 약

의료영상의 크기가 커짐에 따라 처리해야 할 데이터량이 많아졌다. 또한 점점 더 많은 사람들이 빠른 3 차원 가시화 결과를 기대한다. 하지만 이전에 제시된 많은 방법들은 실시간 응답 결과를 기대하기는 힘들다. 본 논문은 하드웨어를 이용하여 의료영상을 빠른 속도로 3 차원 가시화하는 방법과 그 구현 결과에 대해 기술한다. 구현에 있어 자바를 이용함으로써 플랫폼 독립적이며 OpenGL 을 기반으로 한 자바 3D API 를 사용함으로써 쉽고 빠르게 3 차원 가시화 결과를 디스플레이할 수 있다.

1. 서론

의료 영상(MRI, CT 등)의 3 차원 가시화에 대한 연구는 국내외적으로 실시간 처리에 대해 많은 관심을 가지고 있다. 최근 들어 텍스처 매핑은 기하학적 복잡도를 줄이기 위해 가시화 시뮬레이션 및 컴퓨터 애니메이션 분야에서 많이 이용되고 있다[2]. 이러한 실시간 처리는 병렬 패러다임을 이용한 방법 및 하드웨어 수행능력을 향상시키는 방법, 그리고 소프트웨어적인 최적화를 통해서 이루어진다. 하지만 소프트웨어적인 렌더링 기법은 실시간 처리에 많은 한계점을 가지고 있다. 또한 병렬 패러다임을 이용한 방법은 특별한 하드웨어적 설계와 이러한 병렬성을 높이기 위해 소프트웨어적으로 지원이 되어야 한다.

특별한 하드웨어를 이용한 영상처리에서는 표준화되지 않은 하드웨어 인터페이스를 이용하였다. 이러한 특별한 하드웨어를 이용한 방법은 비용이 많이 들었으며 호환성 문제가 대두되었다[3][12]. 하지만 많은 사람들의 노력으로 수행능력을 향상시킨 여러 가지 방법들이 표준화되어 가고 있다. OpenGL 과 같은 여러 가지 표준 그래픽 라이브러리를 개선

하여 좀 더 확장성 있고 이식하기 쉽도록 이러한 하드웨어를 지원하도록 하고 있다.

또한 최근 들어 컴퓨터 시스템이 발달함에 따라 주변 장치 또한 급속도로 발전을 하였다. 게임 산업의 발달로 성능이 우수한 그래픽 인터페이스 카드가 많이 보급되었다. 이러한 그래픽 인터페이스 카드들은 내부적으로 많은 양의 버퍼를 확보하고 있으며 3 차원 및 2 차원 영상 데이터를 빠르게 디스플레이 할 수 있도록 한다.

본 논문은 이러한 그래픽 하드웨어를 사용하여 하드웨어 텍스처 매핑을 이용한 의료 영상의 3 차원 렌더링 기법과 구현에 관련하여 기술한다.

2. 관련 연구

3 차원 렌더링(rendering)을 위한 많은 방법들이 소개되었다. 예를 들어 마칭 큐브(marching cube)와 같은 방법은 3 차원 렌더링을 위해 표면을 다각형으로 구성하였다. 그러나, 이러한 방법들은 비록 컴퓨터시스템이 빠르게 발전하고 있지만 점차 영상 데이터량의 크기가 증가함에 따라 더 많은 시간을 필요로 한다. 또한 실시간으로 3 차원을 구성하거나 사용자와의 상호동작에는 한계가 있다. 이러한 한

계를 극복하기 위해 전처리 단계를 거쳐 계산시간을 줄이는 방법들이 제시되었다.

3차원 텍스처 매핑 하드웨어를 이용한 볼륨 렌더링 기법은 빠른 속도로 3차원 물체를 생성할 수 있다. 이러한 방법은 1994년 실리콘 그래픽스사의 RealityEngine graphics system[1]에서 처음 소개되었고 1998년에 InfiniteReality graphics system[4]에서 소개되었다. 실리콘 그래픽스 사에서는 소개한 시스템에서 OpenGL을 확장하여 3차원 가시화 방법을 제시하였다.

본 논문에서는 Cullip 과 Neumann[5]에 의해 발표되었던 방법을 바탕으로 3차원 렌더링을 하였다. 그들이 제시했던 방법을 자바를 이용하여 쉽고 빠르게 3차원 렌더링 결과를 산출해 낼 수 있다. 또한 인터랙티브하게 렌더링 결과를 디스플레이할 수 있다.

3장에서는 텍스처 매핑을 이용한 볼륨 렌더링 기법과 자바를 이용한 구현에 관하여 기술하며 4장에서는 구현된 3차원 결과물에 대해, 그리고 마지막으로 결론 및 향후 연구과제에 대해 기술한다.

3. 볼륨 렌더링 기법 및 구현

3.1. 2차원 텍스처를 이용한 볼륨 렌더링

볼륨 렌더링은 영상 데이터를 3차원 가시화를 위한 유용한 기술이다. 텍스처 매핑 하드웨어를 이용한 볼륨 렌더링 기법은 2차원 또는 3차원 텍스처 슬라이스를 이용하여 직접 가시화하는 기법이다. 이 방법은 레이-캐스팅(ray-casting)과 비슷한 방법이며 거의 동일한 수준의 3차원 가시화 결과를 생성해 낸다.

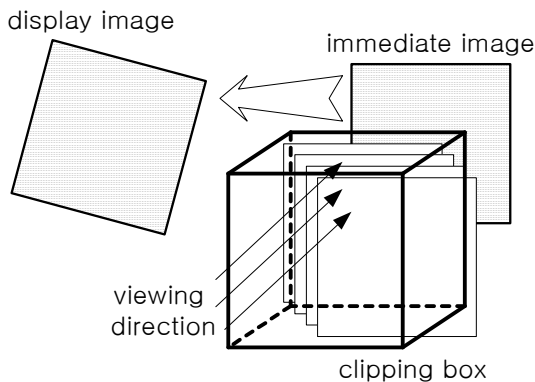


그림 1 2차원 텍스처를 이용한 렌더링

이러한 텍스처 매핑 하드웨어를 이용한 볼륨 렌더링 기법은 3차원 텍스처를 이용하는 방법과 2차원 텍스처를 이용하는 방법으로 나뉘어진다. 3차원 텍스처를 이용한 방법은 2차원 텍스처를 이용한 방법에 비해 좋은 결과를 생성해 내며 더 빠른 방법이다. 하지만 2차원 텍스처를 이용한 방법은 거의 모든 OpenGL 버전에서 구현될 수 있다. 그림 1은 2차원 텍스처를 이용한 렌더링 기법에 대해 도시화 한 것이다. 각 2차원 텍스처 슬라이스를 배

열하여 보는 시각에서 병렬적으로 브렌딩(blending) 연산을 수행한다. 일반적인 브렌딩 연산은 Over, Attenuate, Maximum Intensity Projection, Under가 있다. 이러한 연산들은 이미 OpenGL과 같은 그래픽 라이브러리에 구현되어 있다.

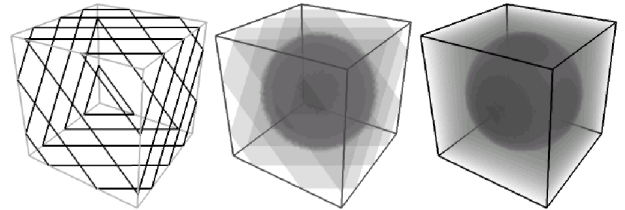


그림 2 텍스처를 이용한 3D 가시화[8]

그림 2는 텍스처를 이용하여 3차원 가시화를 하는 방법을 그림으로 나타낸 것이다. 박스는 클리핑 박스를 나타낸 것이다.

본 논문에서는 2차원 텍스처를 이용한 볼륨 렌더링 기법을 사용하여 의료 영상을 3차원 가시화를 하였다.

텍스처 매핑 하드웨어를 이용한 3차원 볼륨 렌더링 기법은 메모리의 속도, 시스템 아키텍처의 버스 속도, 프로세서의 처리 속도, 캐쉬의 속도, 운영체제의 오버헤드 등에 의해 렌더링 속도가 좌우된다.

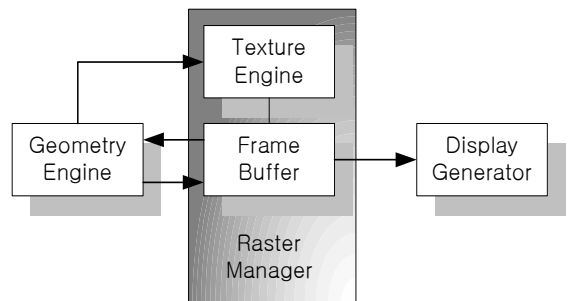


그림 3 그래픽 하드웨어 아키텍처

3.2. 시스템 구조 및 구현

● 시스템 구조

구현된 시스템의 구조는 아래의 그림 4과 같이 나타낼 수 있다.

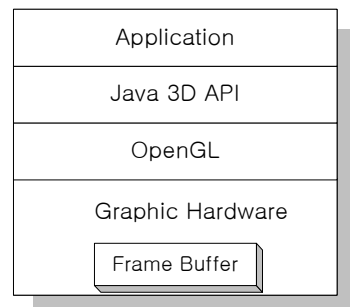


그림 4 시스템 구조

가장 하부에 3차원 텍스처 매핑을 지원하는 고속 그래픽 하드웨어, 그 상위 계층에 OpenGL, 그리고 자바 3D API가 존재한다. 자바 3D API는 OpenGL을 바탕으로 작성되었다.

● 구현

텍스처 매핑 하드웨어를 이용하여 의료 영상을 3차원으로 가시화하기 위해 본 논문에서는 자바를 사용하여 구현을 하였다. 자바는 플랫폼 투명성을 가진다. 다시 말해 현재 많은 시스템(윈도우, 유닉스 등)에서 실행이 가능하다. 자바를 사용함으로써 확장성, 다양한 하드웨어를 지원할 수 있으며 좀더 유연성 있는 어플리케이션을 작성할 수 있다. 추가적으로 구현에 있어서 자바 3D API를 사용하였다. 자바 3D API는 OpenGL을 바탕으로 작성되었으며 3차원 그래픽을 위한 고성능의 API이다[6]. 씬 그래프(scene graph) 구조를 사용하여 3차원 물체를 간단하면서도 효과적으로 표현할 수 있다.

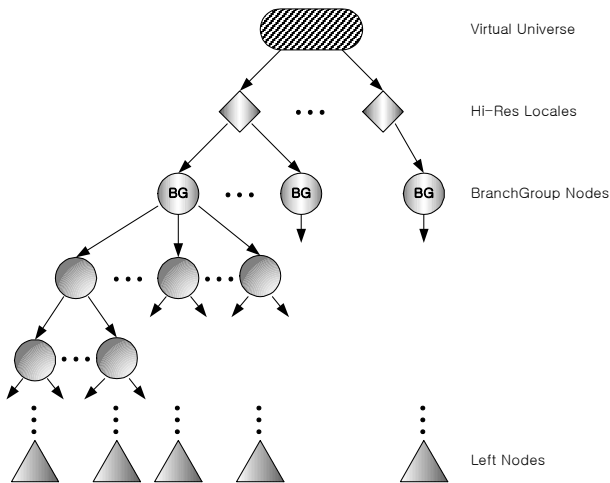


그림 5 자바 3D 씬 그래프 구조[6]

위의 그림 5은 자바 3D의 씬 그래프 구조를 도식화한 것이다. 씬 그래프 구조는 그림 5에서 보는 것과 같이 사이클(cycle)을 포함하지 않는 DAG(directed acyclic graph) 구조이다. 이러한 씬 그래프 구조는 노드객체로 구성되는데 노드 객체는 그룹노드 객체와 리프노드 객체(leaf node objects)로 나뉘어진다. 그룹노드 객체는 하나 이상의 자노드들로 구성되어지며 단 하나의 부모노드를 갖는다. 리프노드 객체는 실제 모양이나 빛 등과 같은 것을 정의한다. 자바 3D API는 구성되어진 씬 그래프를 바탕으로 3차원 객체를 생성한다. 아래의 코드는 그룹객체를 생성하기 위해 본 구현에서 사용한 일부 코드이다.

```
public BranchGroup createSceneGraph(String[] args) {
    BranchGroup objRoot = new BranchGroup();

    centerGroup = new TransformGroup();

    centerGroup.setCapability(TransformGroup.ALLOW_CHILDREN_WRITE);

    Transform3D centerXform = new Transform3D();

    centerGroup.setTransform(centerXform);

    objectGroup = new TransformGroup();
    ...
}
```

4. 구현결과

다음은 의료 영상을 구현된 프로그램으로 실행한 결과에 대해 기술한다. 구현된 결과는 아래의 그림과 같다[그림 6].

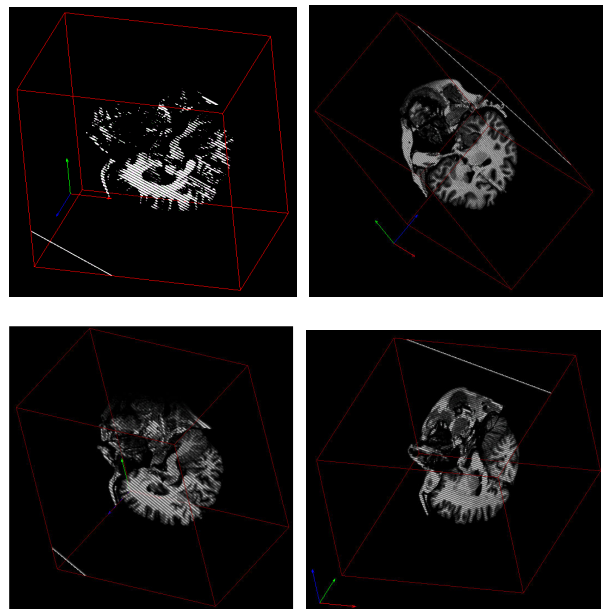


그림 6 구현 결과

실험 데이터는 256*256*109 크기를 사용하였다. 그림 6에서 좌상단은 좌측에서 투시한 것이고, 우상단은 위에서 투시한 것이다. 실험 환경은 다음과 같다. 펜티엄 II 300Mhz 듀얼 프로세서, 256메모리, 그리고 그래픽 인터페이스 카드는 하드웨어 가속능력이 있는 밀레니움 G400 환경 하에서 수행하였다.

5. 결론 및 향후 과제

의료 영상의 크기가 방대해지고 3차원 렌더링 결과를 실시간으로 보고 싶어하는 요구가 많아졌다. 그리하여 영상 데이터를 빠르게 3차원으로 구성할 수 있는 많은 방법들이 제시되었다. 마칭큐브나 레이 캐스팅과 같은 소프트웨어적인 방법들은 실시간으로 3차원 구성 결과를 디스플레이 하기에는 여

러 가지 문제점이 있다. 또한 병렬 컴퓨팅 환경을 이용한 방법들이 제시되고는 있지만 특정 하드웨어나 소프트웨어에 종속적이다. 이러한 병렬성을 위해 특별히 제작된 소프트웨어 알고리즘이 필수적이다. 컴퓨터시스템이 발전하여 비교적 빠른 속도로 3차원 가시화 결과를 볼 수 있으나 실시간으로 사용자가 보기에는 많은 제약을 가지고 있다.

그러나 본 논문에서는 이러한 여러 가지 제약을 벗어나고자 성능이 우수한 그래픽 하드웨어를 바탕으로 의료 영상을 3차원으로 구성하였다. 자바를 사용함으로써 플랫폼 독립적으로 수행될 수 있는 어플리케이션을 작성하였다. 또한 자바 3D API를 사용하여 3차원 가시화를 하였으며 좀 더 유연하고 쉽게 3차원 가시화 결과를 보여 줄 수 있다.

텍스처 매핑 하드웨어를 이용한 방법은 마칭큐브와 같은 소프트웨어적인 렌더링 기법에 비해 빠른 속도로 3차원 렌더링 결과를 얻어 낼 수 있으며 거의 동등한 수준의 가시화 결과를 보인다. 이러한 장점을 이용하여 의료영상 저장 및 전송 시스템과 연동하여 빠른 속도의 3차원 가시화 결과를 사용자에게 제공할 수 있는 시스템으로 구성될 수 있다.

향후 연구과제로는 인터넷을 통한 공동작업 환경을 구축할 수 있다. 공동작업 환경은 실시간 디스플레이가 필수이며 본 연구에서 사용된 자바는 인터넷에서 동작하도록 작성하기 쉽다. 이러한 공동작업 환경은 원격진료와 같은 환경을 제공할 수 있다. 또한 렌더링 결과를 개선할 수 있는 방법에 대한 연구가 또 다른 연구 과제이다.

참고문헌

- [1] K. Akeley. "RealityEngine graphics", Computer Graphics, vol. 27, no. Annual Conference Series, pp. 109-116, 1993.
- [2] M. Teschner, C. Henn. Texture Mapping in Technical, Scientific and Engineering Visualization. <http://www-europe.sgi.com/chembio/resources/texture>
- [3] B. Cabral, N. Cam, and J. Foran. Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware. *ACM Symp. on Vol. Vis.*, pages 91-98, 1994.
- [4] J. S. Montrym, D. R. Baum, D. L. Dignam, and C. J. Migdal, "Infinite Reality: a Real-Time Graphics System", in Proceedings of Siggraph 97, pp. 293-302, ACM, Aug. 1997.
- [5] T. J. Cullip and U. Neumann. Accelerated Volume Reconstruction with 3D Texture Mapping Hardware. Technical Report TR93-027, Department of Computer Science at the University of North Carolina, Chapel Hill, 1993.
- [6] Sun Microsystems. *Java 3D™ API Specification version 1.1*, 1998.
- [7] J. K. Udupa and G. T. Herman. *3D Imaging in Medicine*, CRC Press, Inc., 1991.
- [8] K. Engel, T. Ertl. Texture-based Volume Visualization for Multiple Users on the World Wide Web. Technical Report 16/1998, Universität Erlangen-Nürnberg, 1998.
- [9] Michael Bailey, Cherilyn Michaels. VizWiz: A Java Applet for Interactive 3D Scientific Visualization on the Web. In *Proceeding IEEE Visualization '97*, pages 261-267, 1997.
- [10] Hans-Georg Pagendarm Jens Trapp. A Prototype for a WWW-based Visualization Service. In *Proceedings Eurographics '97*, pp 23-30, 1997.
- [11] Richard S. Gallagher. *Computer Visualization: Graphics techniques for Scientific and Engineering Analysis*, CRC Press, Inc., 1995.
- [12] Robert Fraser. Interactive Volume Rendering Using Advanced Graphics Architectures. SGI Computer Systems. <http://www.sgi.com/Technology/volume/VolumeRendering.html>, 1998