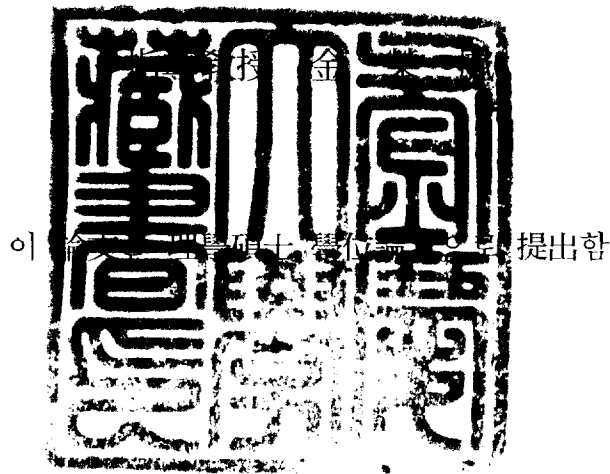


理學碩士 學位論文

웹 기반에서의 상호작용적인
볼륨 가시화를 위한 렌더링
뷰 프로그램의 설계 및 구현



2005年 2月

釜慶大學校 大學院

電子計算學科

尹 要 燮

尹要燮의 理學碩士 學位論文을 認准함

2004年 12月 23日

主 審 工學博士 朴 勝 燮 (印)

委 員 理學博士 朴 興 福 (印)

委 員 理學博士 尹 成 大 (印)

목 차

표 목 차	II
그림 목 차	III
ABSTRACT	IV
1. 서 론	1
2. 관련 연구	3
2.1 웹 애플리케이션 기반 기술	3
2.2 인터넷 기반 3D 영상 기술	8
2.3 ActiveX 컴포넌트 기술	6
2.4 볼륨 렌더링 기법	10
3. 시스템 구현	12
3.1 전체적인 시스템 개요	12
3.2 Rendering Viewer	13
3.3 렌더링 엔진의 DLL 제작	13
3.4 Web ActiveX Component	14
3.5 사용자 웹 Interface와 DB table design	16
3.6 Web Viewer와 DB연동을 통한 렌더링	17
3.7 Rendering 엔진	19
3.7.1 Volume Data 구성 방식	19
3.7.2 Ray Casting 알고리즘	20
3.7.3 볼륨 렌더링 과정	22
3.8 렌더링 웹 서비스 실행 결과	27
4. 결론 및 향후 과제	34
참고문헌	35

표 목 차

표1. 시스템 구현 환경(sever)	27
표2. 시스템 구현 환경(client)	27
표3. 단계별 렌더링 시간	33

그림 목 차

그림1. 웹 기술의 발전 단계	3
그림2. 웹 애플리케이션 구조	4
그림3. 웹 애플리케이션 모델	5
그림4. 픽셀과 복셀의 개념	11
그림5. 구현의 진행 과정	12
그림6. inf 파일 작성	15
그림7. 배치 파일 작성	15
그림8. 웹 인터페이스	16
그림9. DB table design	17
그림10. 데이터 다운로드	18
그림11. Web Viewer 실행 예제	18
그림12. 데이터 구성도	19
그림13. 연산 과정	21
그림14. 3차원 보간	24
그림15. Levoy가 제안한 자동영역분할함수	24
그림16. 광선 추적 과정	26
그림17. 볼륨 관측 예시	27
그림18. 데이터 조작 예시	28
그림19. VRML과 비교	30
그림20. 샘플링 개념	31
그림21. 단계적 렌더링	32

A Design and Implementation of Interactive Volume Rendering Viewer based on Web

Joseph Yoon

*Dept. of Computer Science Graduate School of
Pukyong National University*

Abstract

Since the world wide web, simple and convenient tool, has proposed, the Internet became the most simple network resource which provide many informations of the world. Furthermore, various methodologies are developed to support the dynamic service such as 3D View web service. We will propose the volume rendering view program that interactively visualize the 3D data on the web. The 3D Data is obtained by stacking the 2D images along the z-direction. We also employ the COM based OCX control which is a kind of Active component. This web program will contribute the diagnosis of the diseases through the 3D visualization and image analysis functions at remote places.

1. 서론

질병의 진단을 목적으로 사용하는 CT, MRI, PET 단층사진은 한 두 장의 사진으로 진단하는 X-ray 사진과는 달리 100여장의 사진을 이용한다. 이와 같이 의사가 진료를 할 때 의사는 많은 사진을 한꺼번에 놓고 봐야하고 2차원 단면 사진을 머리 속에서 3차원으로 재구성해야하는 불편함이 있다. 단층 사진을 3차원 영상으로 만들 수 있다면 머릿속에서 재구성할 필요가 없어 의사들은 단층 사진을 진료에 쉽게 이용할 수 있을 것이다. 이러한 연구가 과학적 가시화의 한 분야로 볼륨가시화에서 활발히 진행되고 있다. 그 결과로 Explorer나 AVS, 그리고 VolVis와 같은 다양한 가시화 시스템들이 개발되었고, 실제 able 소프트웨어사의 3d doctor, MediaCybernetics사의 3D Constructor와 같은 우수한 프로그램들이 의료 영상의 분석 및 질병의 진단을 목적으로 상용화되어 쓰여지고 있다. 하지만 이러한 연구들은 주로 자체 컴퓨터나 로컬 네트워크 내에서 3D 볼륨 렌더링 기능을 제공하였다.[1][2][3][4]

또한 최근 몇 년간 많은 병원들이 PACS를 도입하면서 예전과는 달리 슬라이스 영상들을 필름으로 현상해서 일일이 보관하는 것이 아니고 디지털화된 이미지 파일 형태로 저장하고 네트워크를 통해서 빠르게 전송하고 있고, 또한 초고속 인터넷의 발전과 더불어 해당 컴퓨터 뿐만 아니라 원격 진료의 개념에서 멀리 있는 곳에서도 진단이 가능한 그런 시스템의 개발을 요구하고 있는 실정이다.

본 논문에서는 2차원 디지털 이미지를 3차원으로 재구성하여 활용 및 진단에 사용할 수 있는 방법을 제시하고, 더 나아가 병원내 로컬 환경에서만 적용되는 것이 아니라, 어디서나 쉽게 http프로토콜 기반에서의 웹브라우저만 있으면 URL만 입력하여 쉽게 원격지에서도 접근하고, 렌더링 파라미터를 조절함으로써 상호 작용할 수 있는 웹 기반의 볼륨 렌더링 시스템을 개발하고자 한다. 그리고 기존의 PACS와 같은 의료 영상 처리 전문 시스템에서도 쉽게 추가하여 사용할 수 있도록 Com기반의 ActiveX컨트롤로 개발하여 재사용성 및 확장성을 극대화할 것이다.

본 논문의 구성은 다음과 같다. 2장에서 관련연구로서 개발의 배경이 되는 웹 애플리케이션 기반 기술과 의료 영상 가시화를 위한 볼륨렌더링의 특징에 관해 설명한다. 3장에서는 구현된 시스템에 대한 특징을 설명하고, 4장에서 결론을 맺는다.

2. 관련 연구

2.1 웹 애플리케이션 기반 기술

단순히 문서 페이지들에 대한 정보미디어의 한 형태로서 서비스 하던 웹은 빠르고 복잡하게 변하는 웹 기술과 환경으로 인해, 인프라 구조로써 웹을 사용하는 클라이언트/서버 분산 애플리케이션이라는 새로운 종류로써 진화하게 되었다. 웹 기술을 이용한 애플리케이션 구축은 개발이 용이하고, 다양한 멀티미디어 자원으로 그래픽 사용자 인터페이스가 제공되며, 거미줄같이 연결된 네트워크를 통하여 전 세계에 걸친 즉각적인 애플리케이션 전파가 가능하다. 또한 웹 기술은 기능이나 형태에서도(그림1)과 같이 많이 진화되었다[9][10].

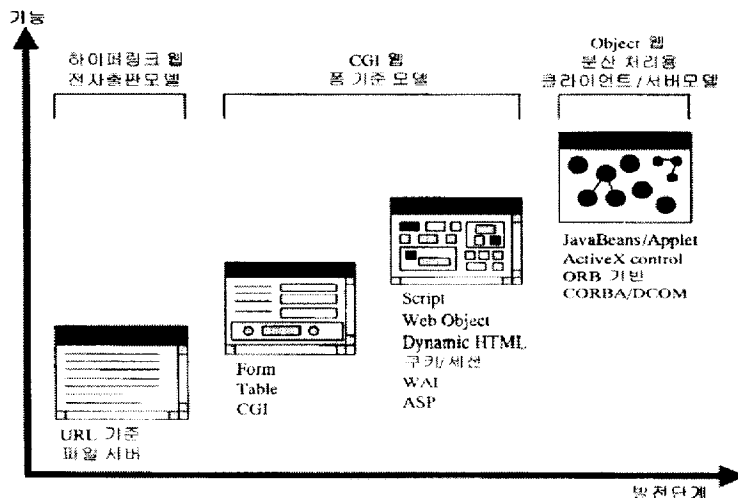


그림1. 웹 기술의 발전 단계

웹 애플리케이션은 올바른 실행을 위해 웹에 의존하는 소프트웨어 애플리케이션이며, 웹에 의존하는 소프트웨어 애플리케이션이며, 웹 전반에 걸쳐 유도되기 위해 설계되어진다. 대부분의 웹 기술은 멀티미디어 파일과 문서의 공유와 분배, 일정한 업무 로직을 가진 웹 애플리케이션으로 구성되었다고 할 수 있으며, 사용자와 상호동작하는 웹 기반의 애플리케이션은 질의, 삽입, 갱신과 정보 처리를 할 수 있다. 웹 애플리케이션에 대한 전체적인 개념 구조는(그림2)와 같이 기술 할 수 있으며[9], (그림3)은 일반적인 웹 애플리케이션의 전체적인 구성요소들을 UML확장 체계를 사용하여 모델링한 것이다[10].

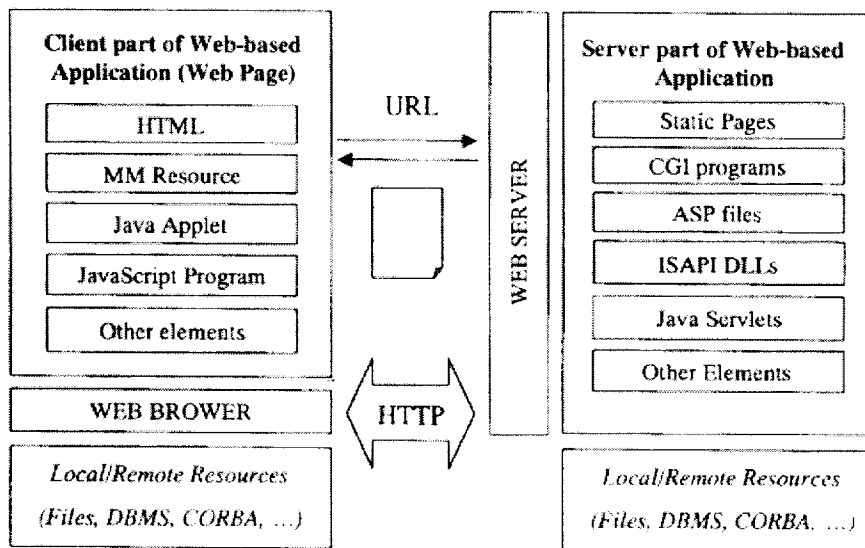


그림2. 웹 애플리케이션 구조

웹 애플리케이션은 웹 브라우저, 웹 서버, 미들웨어, 데이터 베이스 계층, 웹 서버 연동 프로그램으로 구성된다. 웹 브라우저는 사용자 인터페이스의 기능을 향상시키기 위하여 ActiveX컨트롤, 자바 애플릿을 포함 할 수 있으며, 미들웨어는 클라이언트와 서버 애플리케이션을 연결해 주는 역할을 하며, 넓은 의미에서의 미들웨어는 애플리케이션과 애플리케이션 통신을 담당하는 통신 미들웨어를 말한다. 웹 서버 연동 프로그램은 CGI, Sever API, Servlet, ASP 프로그램 등을 의미하며, 이는 주로 웹 서버와 미들웨어를 연동시켜 주거나 웹 서버와 데이터 서비스 계층을 연결시켜 주는 역할을 수행한다.

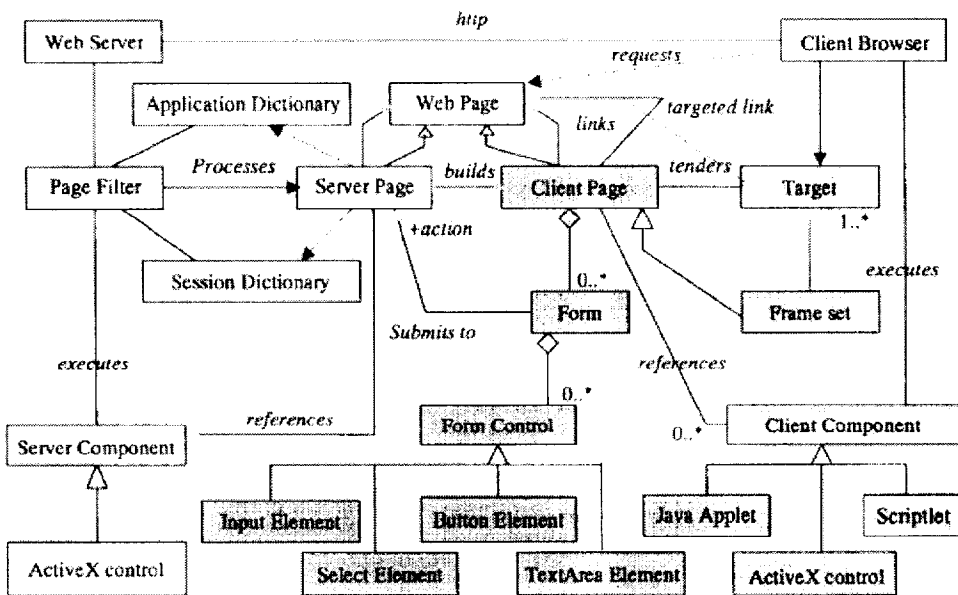


그림3. 웹 애플리케이션 모델

이러한 웹 애플리케이션들의 특징들은 다음과 같다[10].

첫째, 플랫폼에 독립적이다. 웹 페이지들은 클라이언트 측면의 운영시스템에 기반을 두고, 브라우저를 통해 작동하고 접근되어진다. 최근 웹 브라우저는 웹 애플리케이션들의 계층적 네트워크 상에 통일된 플랫폼을 제공하는 모든 종류의 운영체제에서 실행되어지고 있다.

둘째, 단위들의 연속 전개이다. 웹 애플리케이션의 전개 단위인 웹 페이지는 필요할 때 실행시간 동안 즉각적으로 클라이언트 상에서 전개되고, 작업 완료 후에는 또 다른 애플리케이션의 단위가 애플리케이션 유지를 위해 연속적으로 전개된다. 이런 특징은 편리하고 집중화된 방법으로 관리하고, 분산되기 위한 웹 애플리케이션의 능력을 제공한다.

셋째, 다중 엔트리를 제공한다. 웹 애플리케이션은 전통적인 애플리케이션과는 다르게 고정된 시작점을 가지지 않는다. 클라이언트 측에서, 웹 애플리케이션의 서버 부분은 서로 다른 URL들에 의해서 식별되는 많은 엔트리들을 통해서 접근되어 질 수 있고, 대개의 경우 이러한 엔트리들은 서로 독립적으로 접근되어 질 수 있다.

넷째, 느슨한 단위 구조이다. 하나로 된 웹 페이지 대신에 웹 기반의 서버 부분은 정적 페이지 문서, CGI 프로그램, 스크립트 파일, 컴포넌트 등의 서로 다른 형식의 많은 단위들로 일반적으로 구성되어지고, 실질적으로 대부분의 웹 사이트들은 오늘날 그러한 혼합이다. 이것은 다른 것과 간섭 없이 애플리케이션의 일부분을

업그레이드나 재배포 등 갱신을 가능하게 만든다. 더욱이, 이런 특징은 또한 애플리케이션 단위들의 재사용의 잠재성을 가져 올 수 있다.

다섯째, 이벤트 운용방식의 클라이언트와 절차적인 서버이다. 웹 페이지들은 웹 애플리케이션의 사용자 인터페이스이다. 초기 웹 페이지들에서 거의 디스플레이만 되던 것이 DHTML, java applet, java/VB script, 기타 script들로 인해 애플리케이션 단위로 실행이 가능하도록 하였다. 애플리케이션의 부분으로써, 그것들은 최종 사용자와 상호작용동안 이벤트에 응답하는 이벤트-운영의 방법으로 실행된다. 서버 부분은 이와 다르게 전형적으로 절차적인 방법으로 실행한다. 그것은 서버 부분의 작업이 요구에 대한 응답으로 웹 페이지들을 생산하기 때문이다. 이것은 동기식의 이벤트를 생산하여 사용자와 인터페이스를 하지 않고, 요구가 주어진 후에 그것의 실행 전후관계가 고정되므로 실행코스는 순차적이다.

여섯째, 개발은 애플리케이션의 수명이 끝날 때까지 지속적이다. 처음 사용된 날부터 수시로 변화하고, 갱신되고, 업그레이드 될 수 있다. 이것은 변하기 쉽고, 자주 변할 필요가 있는 페이지 유래의 정보 서비스에 그것의 근원이 있기 때문이다.

2.2 인터넷 기반 3D 영상 기술

최근 xDSL과 같은 초고속 인터넷 기반 기술의 발전은 단순히 문자 기반의 문서 정보 전달, 정적인 페이지 검색의 형태에서 벗어나 동적인 3차원 정보를 제공하고 사용자와 상호작용하는 멀티미디어적인 확장을 가능케 하였다. 이와 관련한 인터넷 기반의 3차원 영상 기술을 살펴보면, 단순히 이미지를 원통이나 구 형태로 매핑하여 중심점에서 돌려보는 이미지 기반의 파노라마 기술과 3차원 폴리곤을 기반으로 하는 Web 3D 기술이 있다. Web 3D 기술로는 VRML, 슈퍼스케이프, 쿨트 3D, 펄스 3D 등이 있는데, 이런 기술들은 현재 세계의 주요 사이트에서 활용하고 있다. 그러나 200KB에서 3MB에 이르는 플러그인 소프트웨어를 내려받아야 하는 불편함이 있다. 그 외의 Web 3D 기술로는 자바버추얼머신(JVM)을 탑재한 환경 하에서 실행되는 자바 언어기반의 Blaxxun3D, Shout3D, Java3D 등이 있었다[17].

2.3 ActiveX 컴포넌트 기술

ActiveX는 컴포넌트의 재사용을 위해 마이크로소프트사가 Component Object Model(COM)에 기반을 두고 개발한 것이다. COM은 개발자들이 자신이 선택한 언어에서 각기 컴포넌트를 만들 수 있는 인프라 구조로서, 더 크고 더 복잡한 시스템을 만들기 위해 상호간에 이러한 객체들을 공유한다. ActiveX는 단지 이러한

기술들의 부분집합과 연관된 브랜드 이름이다[11][12].

ActiveX는 자바 애플릿이 하는 것과 매우 유사한 방식으로 클라이언트들의 기능성을 확장하는 데 사용되어진다. ActiveX 컨트롤은 HTML 페이지 상에서 <object> 태그로써 위치되어진다. ActiveX 컨트롤의 표준 기능은 OLE컨트롤과 마찬가지로 Property, Method, Event를 가지고 있으며, 웹 상에서 사용되어 질 수 있다. 웹 애플리케이션에서, ActiveX 컴포넌트들은 사용자 인터페이스를 강화시키고 비즈니스 로직을 구현한다. 별로 놀랍지는 않겠지만, 인터넷상에서 ActiveX의 가장 일반적인 사용은 사용자 인터페이스 컨트롤이다. ActiveX컴포넌트들 혹은 컨트롤들은 표준 텍스트 박스들이나 선택 리스트에서 하기 어려운 사용자 입력을 획득 할 수 있다.

ActiveX가 웹에서 적합하도록 만드는 주요한 특징 중의 하나는 컴포넌트들이 필요할 때 자동적으로 다운로드 되어 질 수 있다는 것이다. 전형적으로 클라이언트에서 ActiveX가 사용될 때, 컨트롤은 캐비닛(.cab)파일에 묶여진다. 이것은 Java의 JAR file과도 유사하고, 디지털 서명되어질 수도 있다[11].

웹 환경의 특징을 고려해 본다면, 보통 웹에서 브라우저를 통해 우리가 보는 것들은 기본적으로 페이지이다. 모든 것이 페이지 단위로 읽혀지고 화면에 표시가 된다. 그런데, 만일 기존의 클라이언트/서버 방식처럼 많은 데이터를 읽어서 화면에 복잡한 많은 정보를 보여준다면, 한 페이지가 넘어갈 때마다 DB에 접속하여 데이터를 읽어와서 상당한 시간이 걸릴 것이다. 하지만 ActiveX컨트롤은

DB에 접속하여 한번에 데이터를 읽어 들여서 사용자의 요구에 따라 즉시 반응하여 내용을 보여 줄수가 있다. 물론 처음 DB에 접속하여 데이터를 가져올 때 당장은 시간이 좀 걸리는 단점이 있지만, 그 이후로는 상당히 빠른 응답시간을 제공한다. 최근에는 DirectX를 ActiveX컨트롤로 만들어 웹 상에서 3D 그래픽을 구현하기도 하고 화상회의 애플리케이션들에 응용되고 있다.

2.4 볼륨 렌더링 기법

기존의 폴리곤 방식의 그래픽 렌더링 방법은 점과 선으로 연결하여 오브젝트를 생성하기 때문에 surface부분만 렌더링 되어서 내부 영역을 가시화 할 수 없었다. 하지만 모든 물체의 요소를 복셀 단위로 취급하여 각각 점으로 표현하는 볼륨렌더링은 복잡한 영역을 표현 가능케 할뿐 아니라 내부영역의 구조도 표현할 수 있다. 2차원 영상들을 재구성 할 때 픽셀들을 z축으로 쌓은 후 viewing ray에 따라서 샘플링 과정을 거친 복셀 기반으로 렌더링 하기 때문에, 이때 픽셀에 대응하는 복셀이 가지는 값이 오브젝트의 내부 정보가 되는 것이다. (그림4)는 픽셀과 복셀의 개념을 그림으로 표현한 것이다.

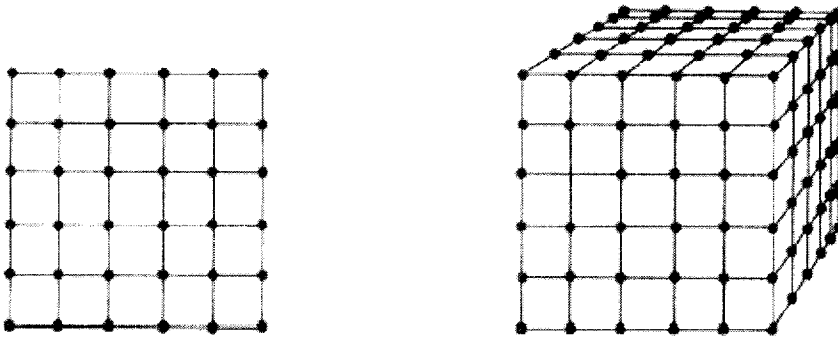


그림4. 픽셀과 복셀의 개념

2차원 데이터 셋을 기본으로 3차원 영상으로 렌더링하는 방법으로, 볼륨데이터의 외곽 표면을 다각형 요소로 나눈 후 렌더링을 하는 간접렌더링 방법과, 각 복셀의 범선 벡터를 구해서 직접렌더링을 하는 방법으로 나눌 수 있고, 또한 직접 렌더링 방법은 투영 공간과 오브젝트의 투영 방향에 따라 영상 공간 렌더링 방법과 물체 공간 렌더링 방법으로 나누어 진다. 간접 볼륨 렌더링 방법으로는 marching cube[13], contour-connecting[14]와 같은 방법들이 있고, 직접 볼륨 렌더링 알고리즘으로는 ray-casting[6], splatting[21], Shear-Warp[16]등의 방법이 있다.

3. 시스템 구현

3.1 전체적인 시스템 개요

웹상에서 사용자의 파라미터에 따라 3차원 볼륨 데이터를 인터랙티브하게 렌더링하여 보여주는 프로그램을 구현하기 위해서는 그래픽 렌더링 엔진이 있어야 하고, 표현되는 정보로서 3차원 볼륨 데이터가 있어야 한다. 실제 웹상에서 볼륨데이터 파일을 구동하기 위해 렌더링 프로그램을 Full ActiveX 컴포넌트인 OCX로 만들었다. 그리고 데이터 파일은 웹상에서 ASP(Active Sever Page)와 연동하여 데이터베이스로 관리함으로써 효율적인 관리와 접근 및 수정의 용이함을 도모 하였다. 시스템 구현의 전체적인 진행과정은 아래 (그림5)와 같다.

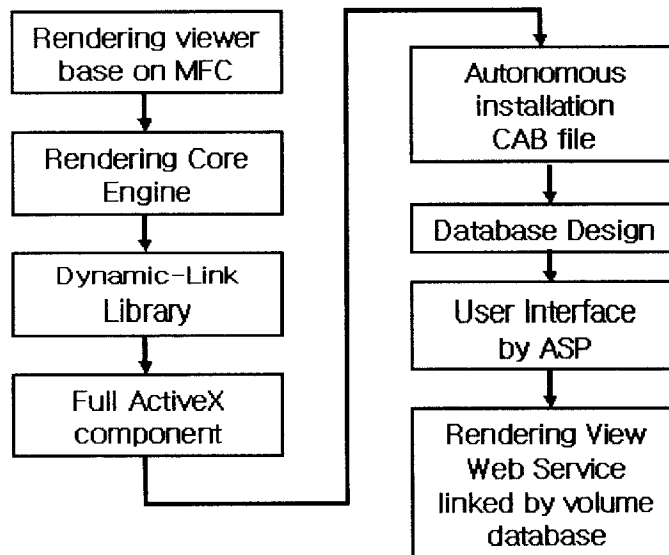


그림5. 구현의 진행 과정

3.2 Rendering Viewer

렌더링 엔진의 독립 모듈화를 위하여 Rendering Engine을 클래스화 및 정리 구성하였다. 여기서 Rendering Viewer란 MFC를 기반으로 제작된 일반 애플리케이션 개념의 프로그램을 의미한다. 이에 Rendering Viewer는 렌더링 엔진의 DLL 제작과 웹에서 실행 가능한 ActiveX component인 OCX를 제작하기 위한 프로토타입이라 할 수 있다. 참고로 ActiveX를 작성하는 방식에는 MFC를 이용하는 방식과 ATL을 이용한 방식이 있는데 ATL을 이용하여 ActiveX Control을 제작하면 MFC를 이용하는 방식보다 다소 어렵기는 하나 작고 빠른 코드를 생산할 수 있는 장점이 있어 더 많이 사용되고 있다. 하지만 컴퓨터와 네트워크의 발전으로 작고 빠른 코드를 생산할 수 있다는 장점은 크게 중요하다고 생각하지 않고, 그보다는 많은 사람들이 사용하는 방식이라는 점이 더 큰 장점이라고 생각된다. 그러나 MFC로 제작된 프로그램을 ActiveX로 수정 작성할 경우에는 ATL을 이용하는 방식보다 특히 간편하고 용이하므로 여기서는 MFC를 이용한 방식으로 제작하였다.

3.3 렌더링 엔진의 DLL 제작

코드는 항상 수정 및 개선되기 마련이며 렌더링 엔진 코드 때문에 분리될 수 있는 Active 코드 부분까지 다시 제작한다면 이 컴포넌트는 확장성이 떨어질 수밖에 없다. 이에 렌더링 엔진 코드

부분을 독립적 모듈화를 통해 최종 개발될 프로그램의 수정 및 업데이트의 편의를 위해 DLL로 만들어 다른 곳에도 유용하게 사용할 수 있도록 할 것이다.

3.4 Web ActiveX Component

클라이언트 웹 브라우저를 통해 CAB파일로 제작된 ActiveX Web Rendering Viewer Component는 사용자 쪽에 자동 설치되고 후에 있을 수 있는 그래픽 엔진 모듈의 수정 및 보완은 최신 버전의 ActiveX Component로 제작되어 다시 CAB 파일을 통한 버전 관리가 이루어지도록 하였다. 이 CAB파일은 렌더링 엔진을 내장한 OCX파일과 컨트롤 및 지원 파일 설치를 제어하는 inf파일로 구성되며, cabinet software development kit을 사용하여 패키지로 만들어진다. 이렇게 패키지를 생성하면, 종속 DLL의 이름을 지정하고 위치를 제공할 수 있으며, 코드 서명을 사용할 수 있고, 더 빠르게 다운로드할 수 있도록 압축할 수 있게 된다. 다음으로 생성된 CAB파일을 웹페이지에 Object태그의 codebase명령을 사용하여 컨트롤의 서버상의 위치를 지정하여 포함시킨다. 이렇게 함으로 웹에서 실행 가능한 ActiveX component로 제작하였다.

(그림6)은 설치 제어를 위한 inf파일을 작성한 것이고, (그림7)은 패키지 배포의 편의를 위해 코드 서명들을 포함한 자동 배치 파일을 제작한 내용을 보여 주고 있다.

3.5 사용자 웹 Interface와 DB table design

보고자하는 볼륨 데이터 파일은 서버에서 제공하는 게시판형 인터페이스를 통해 클라이언트측에 제공하도록 이루어지고, 관련정보, 데이터는 서버측 데이터베이스에 구축 관리된다. (그림8)은 ASP로 제작되어진 클라이언트 인터페이스이고, (그림9)는 My-Sql 데이터베이스 테이블을 이다.

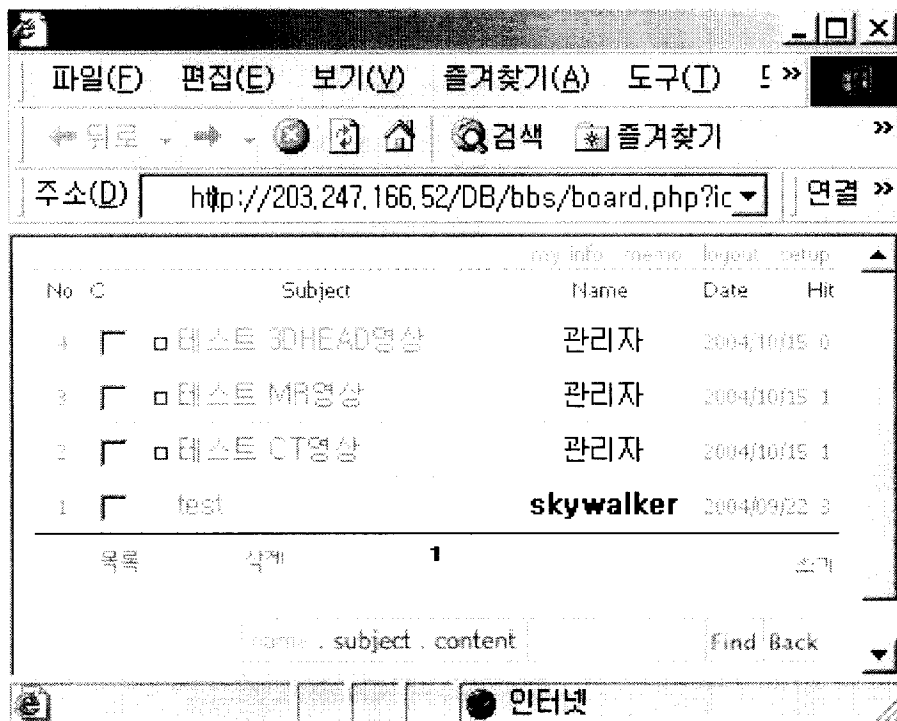


그림8. 웹 인터페이스

DB : rainforest / zetyx_board_bbs_joseph

	필드명	종류	Key	기본값/NULL/추가
C	no	int(20) unsigned	PRi	-/-/- auto_increment
C	division	int(10)	MUL	1/-/-
C	headnum	int(20)		-/-/-
C	arrangenum	int(20)		-/-/-
C	depth	int(10) unsigned	MUL	-/-/-
C	prev_no	int(20)	MUL	-/-/-
C	next_no	int(20)	MUL	-/-/-
C	father	int(20)	MUL	-/-/-
C	child	int(20)		-/-/-

그림9. DB table design

3.6 Web Viewer와 DB연동을 통한 렌더링

렌더링 웹 프로그램을 실행하기 위해서 우선 사용되어질 볼륨 데이터를 구축되어진 데이터베이스를 통해서 (그림10)에서와 같이 전송을 받고, 간단히 브라우저상에 URL만을 입력하여 ActiveX Component Web Page에 접근함으로써 렌더링을 수행할 수가 있다. 각종 데이터의 조작은 클라이언트 웹 페이지에서 VB스크립트로 구성된 폼에 입력 파라미터를 컨트롤에 넘겨줌으로서 인터랙티브하게 볼륨데이터를 가시화 할 수가 있다. (그림10)은 클라이언트의 데이터 다운로드 예시를 나타내며, (그림11)은 실제 Web Viewer를 실행시킨 예제를 나타낸 것이다.

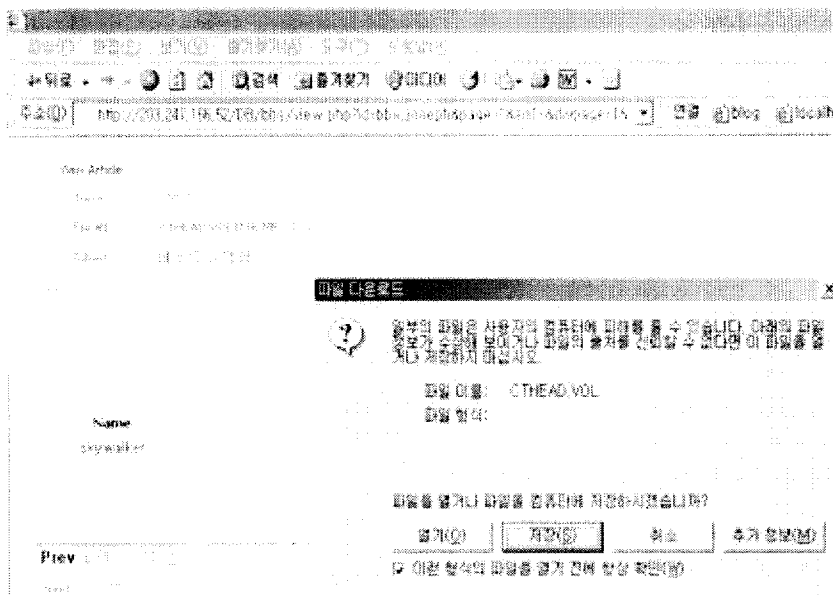


그림10. 데이터 다운로드

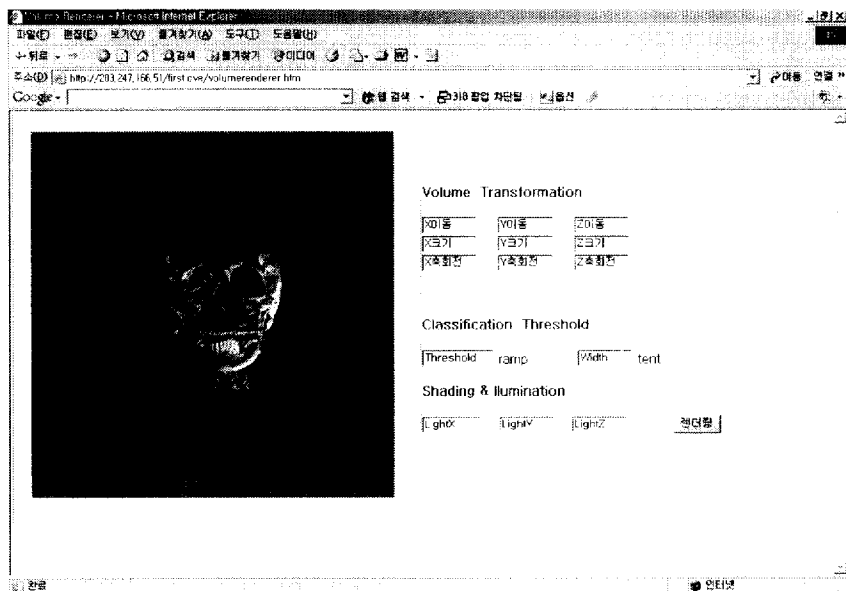


그림11. Web Viewer 실행 예제

3.7 Rendering 엔진

3.7.1 Volume Data 구성 방식

(그림12)와 같이 연속적인 2차원 슬라이스 영상을 z방향으로 쌓아서, 렌더링 웹 프로그램에서 실행되어지는 볼륨 데이터로 만들어 사용하였다. PACS와 같은 시스템에서 이용되어 지는 디지털 정보를 파싱해서 사용한다면, 보다 지능적인 시스템이 될 것이다.

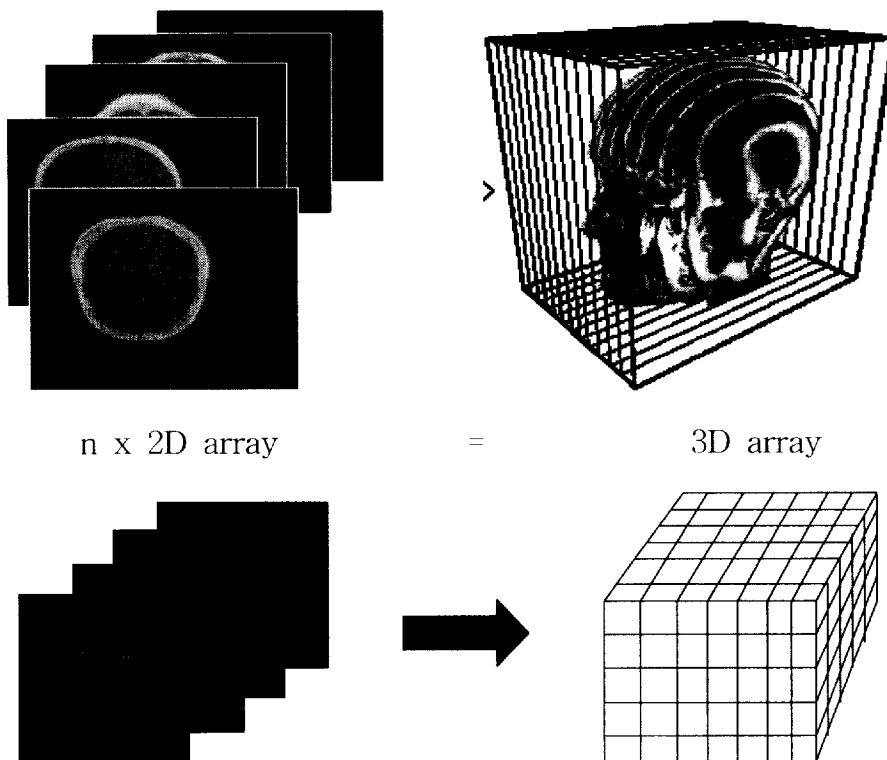


그림12. 데이터 구성도

3.7.2 Ray Casting 알고리즘

2차원 평면에 3차원 물체를 표현하는데 사용되는 알고리즘이다. 3차원 공간상의 한 점을 복셀이라고 하고 3차원 공간상의 물체를 투사하였을때 2차원 투사평면에 투사된 2차원 공간상의 한 점을 픽셀이라고 한다. 샘플링된 좌표의 위치를 결정하기 위해서는 아래와 같은 변환 행렬을 사용하여 ray가 지나가는 위치를 결정한다.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a & e & i & m \\ b & f & j & n \\ c & g & k & o \end{pmatrix} X \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (1)$$

여기서, (x',y',z')는 고정된 z'값을 가지는 2차원 투사 평면이며 (x,y,z)는 실제 물체가 놓여 있는 3차원 공간상의 좌표이다.

Ray casting 알고리즘은 거꾸로 2차원 투사 평면에 맺힌 화소로부터 3차원 공간상의 복셀들의 위치를 추적하는 방법이므로 위의 변환 행렬식을 역변환해야 된다. 역변환 행렬을 아래와 같고 역변환식은 다음과 같이 표현할 수 있다

$$\begin{pmatrix} al & el & il & ml \\ bl & fl & jl & nl \\ cl & gl & kl & ol \\ 0 & 0 & 0 & 1 \end{pmatrix} = \text{inverse } X \begin{pmatrix} a & e & i & m \\ b & f & j & n \\ c & g & k & o \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

역변환 행렬

$$\begin{pmatrix} al & el & il & ml \\ bl & fl & jl & nl \\ cl & gl & kl & ol \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3)$$

역변환식

또 아래와 같이 시선 방향으로 z' 값을 증가시키면서 2차원 평면과 3차원 물체를 만나게 하여 3차원 좌표를 구한다.

$$\begin{pmatrix} x + il \\ y + jl \\ z + kl \\ 1 \end{pmatrix} = \begin{pmatrix} al & el & il & ml \\ bl & fl & jl & nl \\ cl & gl & kl & ol \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x' \\ y' \\ z' + dz' \\ 1 \end{pmatrix} \quad (4)$$

Ray Casting은 이런 복셀의 값을 시선 방향으로 모두 누적시켜 최종으로 2차원 공간상에 투사되는 3차원 영상을 표현할 수 있다.

3.7.3 볼륨 렌더링 과정

볼륨 렌더링을 위한 렌더링 결과를 얻기 위해 (그림13)과 같이 그라디언트 연산, 보간 연산, 클래스파이, 셰이딩, 콤포지팅의 순으로 계산 결과가 만들어져야 한다.

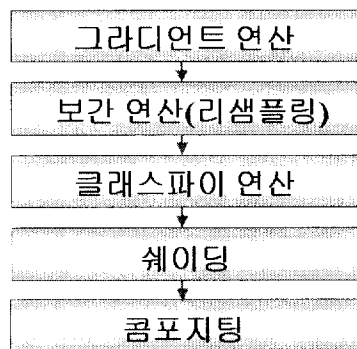


그림13. 연산 과정

그라디언트 연산

여기서 그라디언트는 곧 복셀 표면법선 벡터를 의미한다. 셰이딩과 클래스파이 과정에서 그라디언트의 크기를 이용하여 빛과 관계된 요소와 불투명도를 얻게 되어 있으므로 그라디언트의 크기와 방향을 구할 필요성이 있다. 그라디언트 (D_x , D_y , D_z)는 해당 복셀의 x , y , z , 방향의 전후 복셀의 값의 차이를 이용하여 다음과 같이 구한다.

$$\begin{aligned}
D_x &= f(\mathbf{x}-\mathbf{1}, y, z) - f(\mathbf{x}+\mathbf{1}, y, z) \\
D_y &= f(x, \mathbf{y}-\mathbf{1}, z) - f(x, \mathbf{y}+\mathbf{1}, z) \\
D_z &= f(x, y, \mathbf{z}-\mathbf{1}) - f(x, y, \mathbf{z}+\mathbf{1})
\end{aligned} \tag{5}$$

그라디언트의 벡터 크기는 다음 식과 같이 구할 수 있다.

$$|\bar{D}| = \sqrt{D_x^2 + D_y^2 + D_z^2} \tag{6}$$

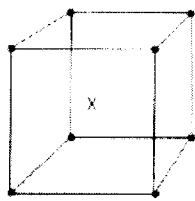
보간 연산

앞의 변환식에 의해 구한 3차원 좌표는 실제 3차원물체의 복셀 좌표와 일치하지 않는 경우가 발생하므로 복셀간 보간 연산이 필요하게 된다. 복셀간 보간 연산으로는 현재 구해진 좌표에 0.5씩을 더해준 다음 소숫점 이하는 버리는 근접연산을 사용한다.

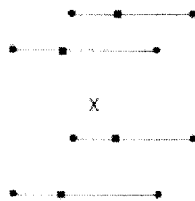
$$\begin{aligned}
x &= [x+0.5] \\
y &= [y+0.5] \\
z &= [z+0.5]
\end{aligned}$$

여기서 $[x]$ 는 x 보다 작은 최대 정수를 취하는 연산자이다.

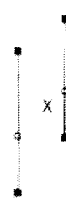
(그림14)는 주변의 복셀값을 고려해서 ray의 일직선상에 복셀이 놓이도록 하기 위하여 근접 보간을 하는 방법을 나타낸 것이다



8복셀



x보간



y보간



z보간

그림14. 3차원 보간

클래스파이

물체의 불투명도를 계산하는 단계이다. 데이터를 구성하는 여러 물질이 섞여 있을 때 각각의 물질들의 경계면을 뚜렷하게 표현함으로서 영역을 구분해 내는 것이다. 해당 복셀에서 물질과 얼마나 가까운가를 나타내는 기준이 되는 값인 그라디언트의 크기와 복셀의 밀도값으로서 불투명도를 결정하는 르보이가 제안한 방법을 사용하였다.(그림15)는 이 함수에 대한 개념을 나타낸 것이다

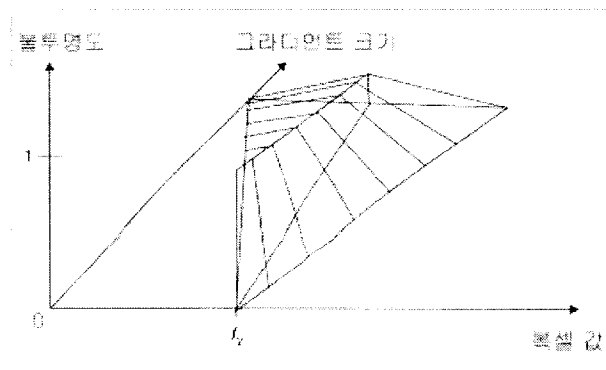


그림15. Levoy가 제안한 자동영역분할함수

그라디언트의 크기와 복셀값에 의해서 해당 복셀에 대한 불투명도를 얻는다. 복셀값 I_i 를 갖는 복셀 I 의 표면 $|\vec{D_i}|$ 일 때 불투명도 $(\alpha_i(r, f_v))$ 를 구하는 식은 다음과 같다.

$$(\alpha_i(r, f_v)) = \begin{cases} 1 - \frac{1}{r|\vec{D_i}|}|fv - I|, \\ \text{(if } |\vec{D_i}| > 0 \text{ and } |fv - I| \leq r|\vec{D_i}|) \\ 1, \\ 0, \end{cases} \quad (7)$$

여기서, f_v 는 최대 불투명도를 갖게 되는 복셀값을 의미하며, r 는 불투명도가 0이 되지 않도록 f_v 에서 유도된 복셀의 최대값이다.

쉐이딩

그라디언트 연산에서 구한 복셀 표면 법선벡터를 이용하여 렌더링하는 방법에는 constant(flat) shading, image-based contextual shading, normal-based contextual shading, gouraud shading, phong shading[19]등이 있다. 이중에서 이미지의 질이 가장 우수한 영상을 생성해 내는 Phong 쉐이딩을 적용하여 구현한다.

아래에 주어진 풍의 조명 모델을 이용하여 각 복셀에서의 쉐이딩 계산을 수행한다.

$$C_i = C_a \cdot K_a \cdot O_d + C_p \cdot K_d \cdot O_d \cdot (\bar{A} \cdot \bar{L}) \quad (8)$$

여기서, C_i 는 쉐이딩 된 값, C_a 는 물체의 고유 밝기값, K_a 는 표면

반사값, O_d 는 물체의 빛에 의한 확산치, C_p 는 빛의 칼라값, K_d 는 물체의 확산반사값, $\bar{N}$ 는 표면법선벡터, $\bar{L}$ 은 빛의 방향 벡터이다

컴포지팅

Front-To-Back 방식을 통해서, 위의 과정에서 불투명도와 셰이딩에 의해 결정된 값을 이용하여 시선방향으로 누적 시키는 과정이다. 식으로 표현하면

$$I(a,b) = \sum_{i=0}^n C_i \prod_{j=0}^{i-1} (1 - \alpha_j) = C_0 + C_1(1 - \alpha_0)(1 - \alpha_1) + \dots \quad (9)$$

$I(a,b)$ 는 복셀누적화 된 값이고, a 는 시선방향으로 입사시 처음 시작하는 위치고 b 는 마지막 위치이다.

$$I_{out} = I_{in} + (1 - \alpha_{in}) \alpha_i C_i$$

$$\alpha_{out} = \alpha_{in} + \alpha_i (1 - \alpha_{in})$$

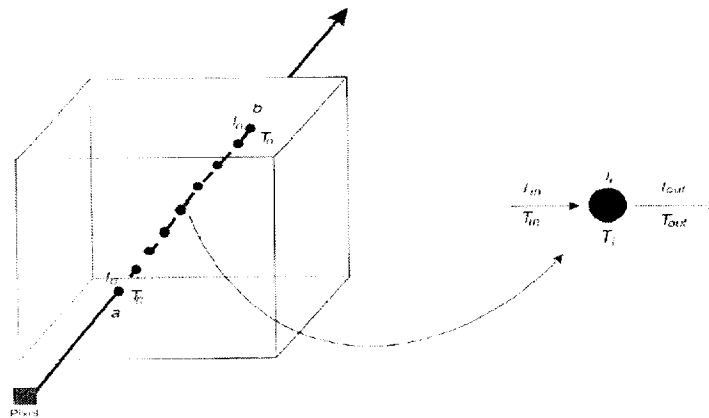
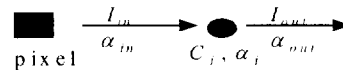


그림16. 광선 추적 과정

3.8 렌더링 웹 서비스 실행 결과

실험환경

내 용	시뮬레이션 환경(Server)
운영체제	Windows2000 Asvanced Server
CPU	AMD athlon(tm) MP 1800+
주기억 용량	2GB DDR RAM
웹 서버	IIS 5.0
DBMS	MS-SQL 2000 Server
프로그래밍 툴	ASP 3.0

<표1> sever 시뮬레이션 환경

내 용	시뮬레이션 환경(Client)
운영체제	Windows2000 Professional
CPU	Intel Pentium-4 866 MHz
주기억 용량	256MB DDR RAM
그래픽카드	ATI Radeon 7200(3D 가속지원)
프로그래밍 툴	Visual C++ 6.0
웹 브라우저	Windows Internet Explorer 6.0

<표2> client 시뮬레이션 환경

3차원 볼륨 데이터의 여러 가지 분석과 관찰을 위해서 볼륨의 회전, 확대/축소, 이동, 불투명도 조절, 광원의 위치 조절 등의 기능들로 사용자 환경을 구성하였다.

CT영상 렌더링

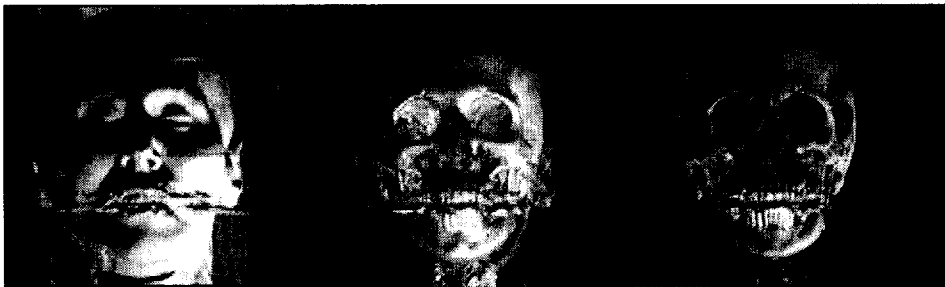


그림17. 볼륨 관측 예시

(그림17)은 본 연구에서 구현한 웹에서 표면을 구하는 밀도값의 범위를 변화시킴에 따라 다르게 보여지는 모습을 표현한 것이다.

MR영상 렌더링

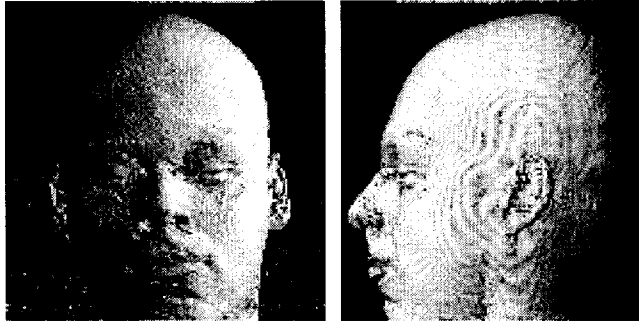


그림18. 데이터 조작 예시

사용자가 물체를 3차원 공간상에서 회전, 확대, 축소하기 위하여 3차원 변환 행렬에 회전, 확대, 축소행렬을 추가로 곱해줌으로써 회전, 확대, 축소 등의 효과를 구현하였다

$$\begin{pmatrix} al & el & il & ml \\ bl & fl & jl & nl \\ cl & gl & kl & ol \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (10)$$

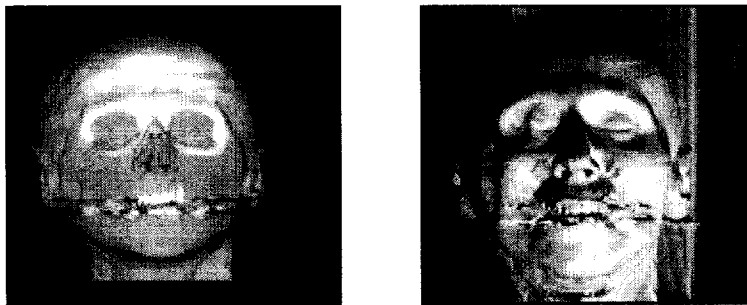
확대,축소

$$\begin{pmatrix} al & el & il & ml \\ bl & fl & jl & nl \\ cl & gl & kl & ol \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (11)$$

회전

웹 렌더링 비교 및 고찰

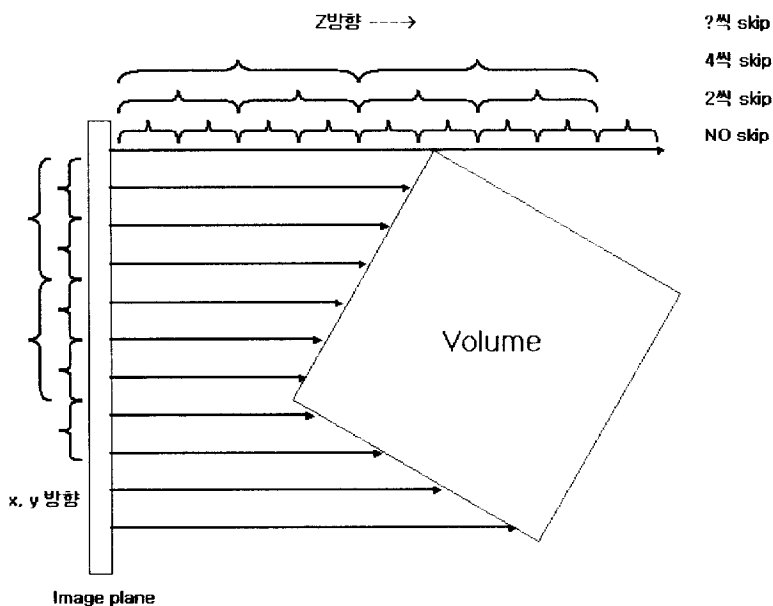
(그림 19)는 볼륨 렌더링의 웹서비스를 위해 VRML로 구현되어진 것[18]과 본 논문에서 구현되어진 결과를 비교한 것이다. 같은 CT head data(256*256*93)을 가지고 IE 6.0 웹 브라우저를 통해 렌더링한 결과이다.



<그림19> VRML VS 본 논문

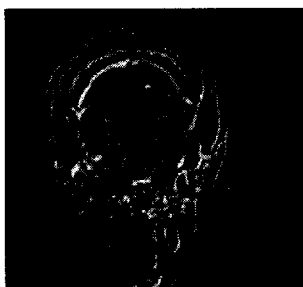
그림과 같이 렌더링된 이미지의 질적인 면에서 본 논문의 결과가 각 특징을 가지는 눈, 코, 입등의 부분에서 좀 더 원 볼륨데이터의 특징을 잘 살려 표현되어진 것을 볼 수가 있다. 하지만 단점으로서 렌더링 된 시간을 개선해야 될 필요가 있다. 사실적인 표현을 위해서 보다 많은 연산량을 가지므로 그만큼 렌더링 시간이 길어졌다. VRML 뷰어에서는 6.77 초 정도 걸린것에 비해 본 논문의 뷰어에서는 29.252초가 걸렸다. 이러한 문제점을 해결하기 위해서 다음과 같은 방법을 제시하였다. 브라우저상에서 렌더링 시간이 길어지게 되면, 상당히 지루한 지연 시간을 갖기 때문에 데이

터를 단계적으로 보내는 것이다. 한꺼번에 바로 영상을 렌더링하는 시간보다는 상대적으로 렌더링 시간이 더 길어진다. 하지만 중간 중간 단계에서 지속적으로 화면에 뿌려 줌으로서 기다리는 시간의 지루함을 해결하고 자 하였다. (그림 20)은 각 단계별로 렌더링 하기 위해서 z 방향으로 볼륨 데이터를 건너뛰어 연산하는 개념을 그림으로 나타낸 것이다.

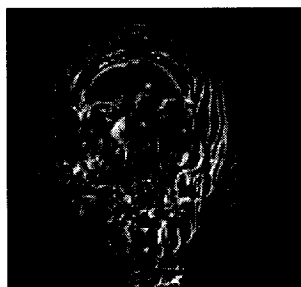


<그림20> z 축 샘플링

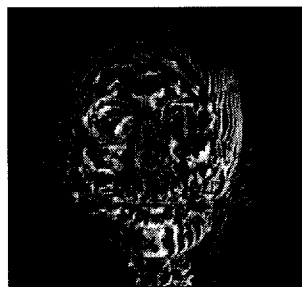
본 논문에서는 z 축 샘플링만을 고려하였다. (그림20)과 같이 x, y 축으로도 샘플링이 가능하고, 속도면에서도 더 빨라지겠지만, 그렇게 할 경우 이미지가 너무 작게 나오고, 단순하게 디스플레이 되어, 단계적으로 제대로 렌더링되지 않는 특징이 있어서, z, y 축 샘플링은 배제시켰다. (그림21) 각 단계별로 렌더링 된 결과이다.



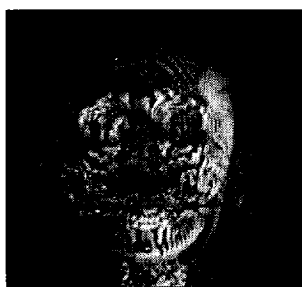
1단계 z방향 30



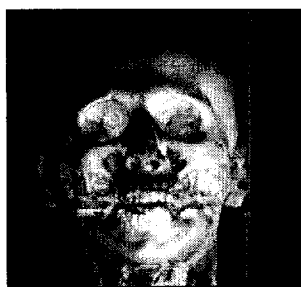
2단계 z방향 15



3단계 z방향 8



4단계 z방향 4



5단계 최종영상

<그림21> 각 단계별 렌더링

한 번에 렌더링 시(sec)	단계적 렌더링시(sec)	
29.292	1단계	3.926
	2단계	5.067
	3단계	6.840
	4단계	10.189
	5단계	29.292

<표3> 단계별 렌더링 시간

의료 영상을 렌더링한다는 관점에서 볼 때 원 볼륨의 사실성, 즉 데이터의 신뢰성에 중점을 두고 구현을 하였으며, 웹 서비스라는 특징에서 길어진 지연 시간을 개선하기 단계적으로 표현하였다.

4. 결론 및 향후 과제

최근 MRI와 CT와 같은 단층 촬영기기의 발달과 병원의 PACS구축으로 디지털화가 가속화되고 있고, 또한 초고속 인터넷의 발전과 더불어 디지털 데이터를 사용한 원격 진료의 개념이 부각되고 있는 실정이다. 이에 본 논문에서는 인터넷이 사용 가능한 웹브라우저만 있으면 CT나 MRI로 구성된 의료영상을 볼륨 가시화하는 편리한 방법을 제시하였다. 향후 과제로 렌더링 엔진에서의 여러 가지 최적화 알고리즘의 적용 및 기능 개선에 대한 연구를 수행 할 것이다.

[참 고 문 헌]

- [1] <http://www.avs.com>
- [2] http://www.cs.sunysb.edu/~vislab/volvis_home.html
- [3] <http://www.ablesw.com/3d-doctor>
- [4] <http://www.mediacy.com/3d/3dconstructor.htm>
- [5] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan, "Volume Rendering", Computer Graphics(Proceedings of SIGGRAPH 88)
- [6] Marc Levoy"Efficient Ray Tracing of Volume Data",ACM Transactions on Graphics(july 1990)
- [7] Marc levoy"Display of surface from volume data", IEEE Computer Graphics and Applications,1988
- [8] D.Ney, E.Fishman, D.Magid, and Drebin "volumen rendering of compute tomography data:Principles and techniques" IEEE Computer Graphics and Applications
- [9] Weiquan Zhao, Jian Chen, "CoOWA: A Component Oriented Web Application Model", Proceedings of the Technology of Object-Oriented Languages and Systems - TOOLs 31, pp.191-199, February,1999
- [10] Hans-W. Gellersen and Martin Gædke, "Object-Oriented Web Application Development", IEEE Internet Computing Vol. 3, No.1, pp.60-68, January, 1999
- [11] Jim Conallen, "Building Web Applications with UML", Addison Wesley, 1999

- [12] Gilda Pour, "Understanding Software Component Technology: JavaBeans and ActiveX", proc. of the Technology of Object Oriented Lanfuages and Systems - TOOLs 29, pp.398, June, 1999
- [13] William c. Lorensen and Harvey E.Cline, "Marching Cube:A High Resolution 3D surface Construction Algorithm", Computer Graphics, Vol.21, No.4, pp.163-169, July 1987
- [14] Z.M. Kedem H.Fuchs and S.P. Uselton, "Optimal surface reconstruction from planar contours", ACM, Inc Vol.20, No.10, October 1977
- [15] Philippe Lacroute, Marc Levoy, "fast Volume Rendering Using a Shear-Warp Factorization of Viewing Transformation", Computer Graphics Processing, Annual Conference Series, 1994
- [16] Lih-Shyang Chen, Gabor T, Herman R, Anthony Reynolds, Jayaram K.Usups."Surface Shading in the Cuberille Environment", ACM, Inc, Vol.20, No.10 October 1977.
- [17] A.E.Walsh, M.Bourges-sevenier, "Core Web 3D", Prentice Hill,2000
- [18] Jinrui Liao, Irene Gargantini, "Interactive Volume Rendering in a VRML Environment", The University of Western Ontario, 2003
- [19] 천보경, "컴포넌트 기반 웹 애플리케이션의 개발 프로세스에 관한 연구", 2000
- [20] 박상민, 윤성현, 김태윤, "밀도값 보간을 이용한 볼륨 렌더링" 한국정보처리학회 추계 Vol3, NO2, 1996
- [21] 임인성, 이래경,"On Enchancing the Speed of splatting with Indexing" In Proxeedings of Visualization 95, 1995

- [22] 박재영, 이병일, 최홍국, "3차원 샘플링에 기반을 둔 볼륨렌더링 프로그램의 설계 및 구현"멀티미디어학회논문지 Vol.5, NO.5, oct., 2002
- [23] 김보선, 최우영, "영역분할된 의료영상의 볼륨 렌더링" Journal of Research Institute of Industrial Technology Vol 20. 2001. MyongJi University
- [24] 윤성의, 신영길, "3D 텍스처 매핑 하드웨어 하에서 법선 벡터 블렌딩을 이용한 가속화된 볼륨 렌더링", 정보과학회 논문지:시스템 및 이론 28권 3호, pp. 166-172, 2001.4.
- [25] 여윤석, 박종구 "ASE 파일 파싱 및 모델 데이터베이스 연동을 통한 3D view 웹 서비스 구현" 한국정보처리학회,Vol.10, No.2. 2003.11.

감사의 글

에벤에셀!

열정적인 꿈에 녹아 올라가 버릴 것 같은 기분으로 학업을 시작한 게 엊그제 같은데 벌써 이렇게 논문을 마무리를 짓고 석사과정에 마침표를 찍으려니 여러 가지로 만감이 교차합니다. 처음이라는 설레임의 감정에서, 공부를 하면서 겪은 수많은 좌절과 고난의 시간, 절망의 마음, 문제를 해결했을 때의 포만감, 아직 끝내지 못한 학문에 대한 미련과 아쉬움까지... 그리고 무엇보다도 더 잘할 수 있었을 텐데 하는 후회가 가장 크게 느껴집니다. 그래도 제가 지금 이 자리에서 있기에 누구보다도 기쁜 마음을 감출 수가 없습니다. 이 모두가 날 아껴주신 고마운 분들이 있기에 가능했다고 생각합니다. 모두들 너무 고맙습니다. 마지막 이 아니라 지금부터가 시작입니다. 이제 겨우 제 머리의 돌을 깨는 방법을 알 것 같고, 이제 겨우 걸음마를 할 수 있을 것 같습니다. 아직 못다한 저의 학문적인 열정은 미련으로 남아 박사과정에 진학을 하게 되었고, 지금은 더욱더 열심히 하는 것만이 나를 사랑해주시고 아껴주신 분들에 대한 보답이라 생각하고 최선을 다하겠습니다. 이제 시작이지만 그래도 한 고비를 넘어가기에, 이렇게 힘든 한 걸음을 떼어 놓기까지 사랑해주시고 격려해주신 많은 지인 분들께 미약하지만 짧을 글로 감사의 마음을 전하고자 합니다.

먼저 저의 지도 교수님이신 김영봉 교수님께 감사의 마음을 전하고 싶습니다. 외국에 나가 계시면서도 제자의 바른 길로 갈수 있도록 참된 학문의 진리를 깨닫게 해주시고, 언제나 현실에 안주해서 도태되지 않도록 채찍질도 마다하지 않으셨던 나의 스승님, 해답은 가까이 있는데 멀리서 고민하고 있을 때 항상 멋진 조언과 가르침으로서 그 문제에 근접할 수 있도록, 해결할 수 있도록 도와주신 교수님 너무나도 감사 감사드립니다. 그리고 너무 보고 싶습니다.

또한 연구와 강의로써 바쁘신 와중에서도 저의 논문 심사를 맡아 신경 써주시고 지도해주신 박승섭 교수님, 박홍복 교수님, 윤성대 교수님께 머리숙여 감사드립니다. 학부와 대학원 생활동안 생활적으로나 학문적으로 많은 도움을 주시고 가르침을 주신 전산과의 교수님, 이경현 교수님, 김창수 교수님, 여정모 교수님, 정순호 교수님, 박만곤 교수님께 감사의 마음을 전합니다. 그리고 앞으로도 많이 부족하고 모자란 저의 지도 편달을 부탁드립니다. 존경의 마음을 전합니다. 언제나 교수님의 진심어린 충고를 되새기며 더욱더 최선을 다해서 학업에 열중하도록 하겠습니다.

그래픽스 연구실원께 감사의 마음을 전합니다. 바른 생활로서 연구에 임할 수 있도록 친동생처럼 아껴주고 챙겨주신 그래픽스 방의 아리따우신 누나들, 이영숙, 박경미, 홍기진, 이운정, 이수정 누님들에게 감사드립니다. 김은실 선배님께도 감사의 마음을 전합니다. 그리고 같이 졸업하는 가람이, 연구실의 막내로서 굵은 일은 마다않고 도와준 태우에게도 감사의 말과 함께 잘되기를 바라며 축복의 메시지를 전합니다. 그리고 유익한 생활을 할 수 있도록 많은 배려와 도움의 말을 아껴주신 형들, 상화형, 승우형 께도 감사의 마음을 전합니다. 그리고 먼저 졸업하신, 저를 아끼고 동생처럼 해주신 건호형, 희준형께도 감사하다는 말을 전하고 싶습니다. 그리고 멀리 외국에 계시면서도 종종 후배가 잘되기 위해, 많은 가르침과 배려를 해주신 우석이 형에게도 감사의 마음을 전합니다. 그리고 선생님의 꿈을 안고 교대원에 입학한 영숙이, 영희 열심히 해서 무사히 졸업하고, 꼭 꿈을 이루기를 기원합니다. 앞으로 더욱 그래픽스 연구실이라는 한 울타리에서 가족과 같이 열심히 연구하여 함께 전진하고 싶습니다.

또한 같이 2년간을 석사과정을 마치며 졸업하는 강진수, 김성대, 한승우, 서병만 동기님들께도 감사의 마음을 전하며, 앞으로 좋은 직장에서, 사회에서 더욱 멋지게 건승하기를 기원합니다. 유일한 1년차이신 천희정 후배님도 열심히 하셔서 멋지게 졸업하길 바랍니다.

그리고 친구처럼 친동생처럼 대해주시고 힘들 때 이야기하면 지냈던 5오빠 식구들 박성석, 정태갑, 이석철에게도 감사하며, 앞으로 열심히 해서 좋은 결실을 맺기를 바랍니다. 그리고 시간 날 때마다 같이 운동도 같이하며, 시간 내기를 마다하지 않았던 친구 태오, 석보에게도 감사의 마음을 전합니다. 그리고 재밌는 이야기와 좋은 대화들로 삶을 유익하게 해준 두승이 에게도 고맙다는 말을 전합니다.

그리고 대학생활의 긴 여정동안 같이해주고 앞으로도 같이 나아갈 친구, 별말이 필요 없는 내 친구 상국이 고맙습니다. 그리고 자기를 준비하기에도 바쁜 시간에 그래도 친구라고 갖은 농담과 장난과 험담을 받아 주며 함께 해주었던 우리 태규나, 창무 이자리를 빌어서 고맙다는 말을 하고 싶습니다. 또한 대학 시절의 많은 추억을 함께 했던 우리 막내 잘생긴 형민이에게도, 형을 잘 따르고 인내해 주어서 고맙다는 말을 전합니다. 그리고 바쁜 생활 속에서도 생각해주시고 도와주려고 애써 주신 최진우에게도 감사의 맘을 전합니다.

언제나 친구처럼 동생을 대해준 형들, 직장 생활하느라 바쁜 와중에도 많은 도움을 아끼지 않고 조언해 주신 원경씨, 창울씨 너무나 고맙습니다.

또한 내가 뭘 하는지도 정확히 몰라도, 언제나 날 믿고 생각해주는 친구들, 지칠 때마다 힘이 되어준 친구들, 창준이, 원석이, 민환이, 남구, 수희, 위성이 고맙다는 말과 함께 파이팅을 전하고 싶습니다. 그리고 친구들도 다 잘 될 수 있기를 간절히 소망합니다. 친구들이여 영원하라

그리고 공부로만으로는 무료해질 것 같던 생활을 더욱 유익하게 해준 동아리 "올림피아"의 식구에게도 감사의 마음을 전합니다.

또한 저를 위해 항상 기도로서 지켜주시고 살피주신, 교회의 교우님들께도 감사의 마음을 전합니다.

그리고 일일이 되새기지 못한, 언급하지 못한 분들에게도 미안한 마음과 함께 감사의 마음을 전합니다.

생각만 하면 매마른 나의 눈이 젖어 들게 하는 나의 영원한 팬에게, 이 못난 나를 항상 사랑보다 아름다운 마음으로 날 아껴주고 지켜주신 분께 미안하다는 말과 함께 사랑한다고 말하고 싶습니다.

마지막으로 이 세상에서 가장 소중한, 가장 든든한 후원자이신, 내가 태어나서 지금까지 있게 해주신 나의 뿌리가 되는 사랑하는 부모님께 감사의 마음을 전합니다. 아버지, 그리고 어머니 당신들의 기도가 있기에 작지만 결실을 맺을 수 있었습니다. 항상 감사하게 느끼며 앞으로도 변함없는 마음으로 사랑하며 존경하며 살겠습니다. 다시 한번 감사드립니다. 그리고 사랑합니다

그리고 끝으로 이 모든 일을 주관하시는 하나님께 이 모든 감사와 영광을 돌리며 마지막을 글을 맺고자 합니다.