

ASE 파일 파싱 및 모델 데이터베이스 연동을 통한 3D View 웹 서비스 구현

여윤석, 박종구
성균관대학교 정보통신공학부
e-mail:yyunseok@ece.skku.ac.kr

Implementation of 3D View Web Service based on ASE File Parsing and Model Database Linking

Yun-Seok Yeo, Jong-Koo Park
School of Info. & Comm. Eng., SungKyunKwan University

요 약

인터넷 사용자들이 정적인 정보 페이지 검색의 형태를 벗어나 인터넷상에서 프로그램을 실행하여 정보를 해석하고 볼 뿐만 아니라, 변경하고 새롭게 창조할 수 있는 동적인 정보를 제공하기 위해 가장 보편적인 3차원 데이터, 3D MAX-Studio의 텍스트 출력물인 ASE 포맷 파일을 파싱하여 렌더링 해주는 3D Viewer 프로그램 구현 하고, 이를 ActiveX 컴포넌트인 OCX로 만들어 웹 페이지 상에서 실행 가능하게 한다. 그리고 데이터의 효율적 관리와 사용자와의 상호작용을 위하여 ASE 모델들을 위한 데이터베이스를 구축하여 사용자 상호작용적인 3D View Web Service를 실현한다. 이를 통하여 인터넷을 통한 실시간적인 정보 교환이나, 네트워크상의 가상공간 내에서의 공동 업무 작업의 가능성을 내다 보았다.

1. 서론

인터넷은 이제 존재하는 전 세계의 모든 정보를 접할 수 있는 가장 훌륭한 도구가 되었다. 인터넷이 이처럼 대중화된 계기는 무엇보다도 다양한 표현 능력을 가지고 있으면서도, 동시에 사용이 매우 간편한 도구인 월드 와이드 웹(World Wide Web, WWW)의 등장이라고도 할 수 있을 것이다. WWW는 인터넷이라는 거대한 네트워크 자원을 활용하여 정보를 공유하는 분야에 있어 독보적이면서 강력하고도 용이한 지원을 제공하고 있다. 그러나 인터넷 사용자들은 전 세계의 정보를 공유하고자 하는 그들의 요구가 어느 정도 충족되는 듯하자, 좀더 새로운 것을 바라게 되었다. 전 세계를 거미줄같이 연결하는 네트워크 자원을 단순히 정적인 정보를 공유하는 도구로만 사용하기에는 아쉬운 점이 많았기 때문이다. 즉, 인터넷 사용자들과 개발자들은 그간의 WWW의 활용 경험을 통해 네트워크를 통한 실시간적인 정보 교환이나, 네트워크상의 가상공간 내에서의 공동 업무 작업의 가능성을 내다보았으며, 다양한 형태로 2차원적인 정보공유의 한계 극복을 모

색해 왔다. 본 논문에서는 이러한 시도의 하나로서 정적이고 정지되어 있으며 변하지 않는 웹 페이지가 할 수 없는 3차원 정보를 제공하고 사용자와 상호작용 하는 인터넷 서비스를 설계, 구축하고자 하였다. 이를 통해 인터넷 사용자들이 정적인 정보 페이지 검색의 형태를 벗어나 인터넷상에서 프로그램을 실행하여 정보를 해석하고 볼 뿐만 아니라, 변경하고 새롭게 창조할 수 있는 동적인 정보를 제공하고자 하였다.

2. 관련 연구

인터넷 기반 3D 영상기술은 크게 두 가지 범주로 나눌 수 있는데 이미지를 기반으로 하는 파노라마와 3D 폴리곤을 기반으로 하는 Web 3D(VRML, Java3D 등 3차원 기술)가 대표적이며, 파노라마와 Web 3D는 상호간에 보완적인 기술이다.

2.1 이미지 기반 파노라마

파노라마의 경우 사진을 원통 또는 구 형태로 매핑하여 중심점에서 돌려보는 방법인데 실제 사진을

이용하기 때문에 제작 과정이 간단하고 질적인 면에서 우수하다. 의류나 음식처럼 조립, 분해가 불필요한 물건을 보여주는 정도만으로도 효과가 충분할 경우에는 파노라마 기술을 이용하여 3차원 사이트를 꾸미는 것이 효과적이다[5].

2.2 Web 3D

Web 3D는 실제 사진과 3차원 폴리곤 기반으로 구현되므로 사진촬영만 하는 파노라마 방식에 비해 제작이 까다롭다. 그러나 네티즌과 콘텐츠 사이에 상호작용(Interaction)이 가능하므로 쇼핑물, 게임, 전시관, 시뮬레이터 등이 요구되는 사이트에서 뛰어난 효과를 얻을 수 있다.

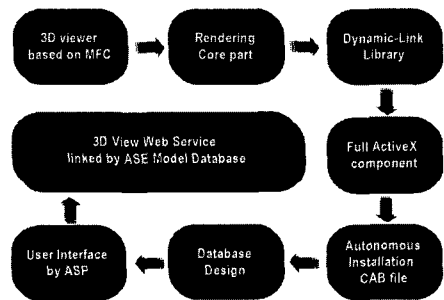
2.3 최근 경향

슈퍼스케이프, VRML, 쉘트3D, 펄스3D 등이 세계의 주요 사이트에서 활용되고 있다. 지금까지 Web 3D 기술이 대중화되는데 있어 가장 큰 걸림돌은 최소 200KB에서 최대 3MB에 이르는 플러그인 소프트웨어를 내려받아야 한다는 것이었다. 그러나 Blaxxun3D, Shout3D, Java3D 등의 기술은 자바 프로그래밍 언어를 기반으로 하기 때문에 현재의 PC 기종은 물론이고 향후에는 자바버추얼머신(JVM)을 탑재한 PC, 셋탑박스, 휴대형 정보 단말기, 워크스테이션 등 다양한 환경에서 3차원 콘텐츠를 접할 수 있게 된다.

3. 구현 과정

먼저 인터랙티브하게 사용자의 마우스 입력에 반응해 적절한 움직임을 보여주는 3D 프로그램을 구현하기 위해서는 물체의 모든 정보를 메모리에서 매 프레임마다 해당 정보를 바탕으로 정점의 변형 및 렌더링을 수행하는 3D 그래픽 렌더링 엔진이 있어야 하고[2, 7, 8, 9, 10, 11], 웹상에서 공유되어지는 정보로서 범용적인 3차원 데이터가 있어야 한다. 그래서 가장 보편적인 3차원 데이터로서는 3D 모델러인 3D MAX-Studio의 텍스트 출력물인 ASE 포맷 데이터[4]를 사용하여 그것을 파싱하여 렌더링 해주는 3D Viewer 프로그램을 구현 하였다. 3D Viewer 프로그램은 3D 그래픽 엔진에 렌더링 부분은 OpenGL 라이브러리를 사용하여 빠르고 안정적인 렌더링을 실현하도록 하였다[1, 3, 12]. OpenGL과 함께 그래픽 라이브러리 분야를 양분하고 있는 DirectX도 이용이 가능하겠으나, DirectX는 그래픽

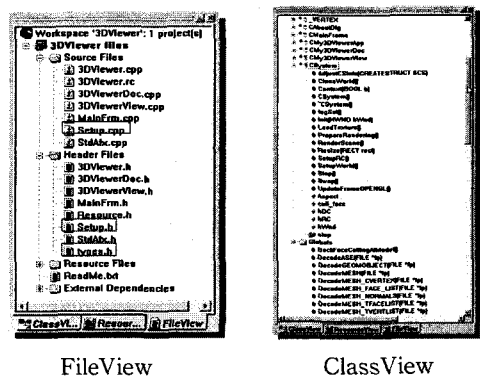
렌더링뿐만 아니라 다양한 멀티미디어 기능 및 네트워크기능 등을 지원하기 적합한 라이브러리이다. 즉, 동일한 하드웨어 가속을 지원받을 수 있고, 사이즈가 더 적으며, 플랫폼 독립적인 프로그램이 되기 위해 OpenGL을 선택하였다. 그리고 MAX의 ASE 모델 파일을 웹상에서 구동하기 위해 3D Viewer 프로그램을 Full ActiveX 컴포넌트인 OCX로 만들었다. ActiveX는 마이크로소프트사에서 인터넷에 상용 비즈니스 프로그램을 접목하기 위해 개발된 넓은 범위의 기술이다[6]. 마지막으로 MAX -Studio의 ASE 모델을 웹상에서 ASP(Active Server Page)와 연동하여 데이터베이스로 관리함으로써 모델 데이터의 효율적 관리와 접근 및 수정의 용이함을 도모하였다. 이 과정을 <그림 1>에 도시하였다.



<그림 1> 구현의 전체 진행 과정

3.1 3D Viewer와 렌더링 엔진의 DLL 제작

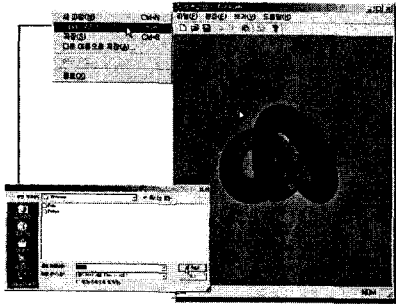
렌더링 엔진의 독립 모듈화를 위하여 Setup.h, Setup.cpp 및 types.h만으로 Rendering Engine을 클래스화 및 정리 구성 하였다. (<그림 2> 참조)



<그림 2> Visual Studio 6.0 Workspace

3D Viewer는 렌더링 엔진의 DLL 제작과 웹에서 실행 가능한 ActiveX component인 OCX를 제작하기

위한 프로토타입이라 할 수 있다. (<그림 3> 참조)



<그림 3> 3D Viewer 실행화면

코드는 항상 수정 및 개선되기 마련이며 3D 그래픽 엔진 코드 때문에 분리될 수 있는 ActiveX 코드 부분까지 다시 제작한다면 이 컴포넌트는 확장성이 떨어질 수밖에 없다. 이에 3D 그래픽 엔진 코드 부분을 독립적 모듈화를 통해 최종 개발될 프로그램의 수정 및 업데이트의 편의를 위해 DLL로 만든다면 앞으로 다른 곳에도 유용하게 사용할 수 있을 것이다. DLL 제작 시 필요한 매크로와 익스포트할 API 정의는 <그림 4>와 같다.

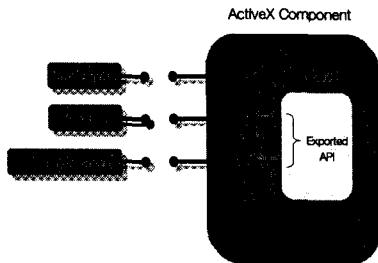
```

#define CORE_EXPORTS
#define CORE_API __declspec(dllexport)
#else
#define CORE_API __declspec(dllimport)
#endif
#include "src/Setup.h"
#include "src/types.h"
core.h

class CORE_API CSystem
POINT3D CORE_API ModelAngle
POINT3D CORE_API ViewCoord
void CORE_API ReadASE()
    
```

<그림 4> 매크로와 익스포트할 API

3.2 ActiveX Component 제작

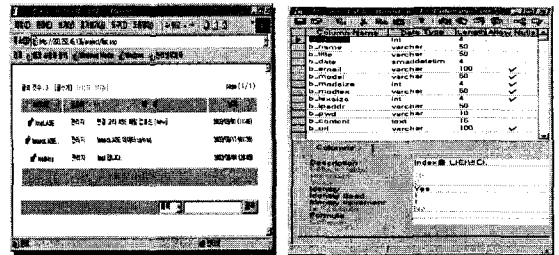


<그림 5> 제작된 ActiveX Component 개념도

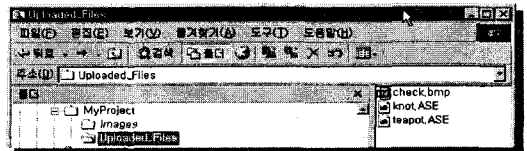
클라이언트 웹 브라우저를 통해 CAB 파일로 제작된 ActiveX 3D Web Viewer Component는 사용자 쪽에 자동 설치가 되고 후에 있을 수 있는 그래픽 엔진 모듈의 수정 및 보완은 최신 버전의 ActiveX Component로 제작되어 다시 CAB 파일을 통한 버전관리가 이루어지도록 하였다. 3D View Web Service가 이루어지기 위해서는 모델 데이터와 모델 데이터가 사용하는 텍스처 파일이 있어야 한다. 사용자가 자기가 제작한 3D MAX-Studio의 텍스트 출력파일인 ASE를 가지고 있다면 모델 데이터의 서버로 업로드 없이 ActiveX Component Web page의 접근만으로 자동설치가 이루어져 해당 모델 데이터 및 텍스처의 하드디스크상의 위치를 ActiveX Component와 동일하게 함으로써 실행 가능하다. 다른 사람과의 공동작업이 필요한 경우에는 웹에 모델 데이터를 업로드하고 다수의 사용자가 다운로드를 하여 동일한 모델을 볼 수 있다. 이러한 일련의 구현을 위해서 <그림 5>와 같은 인터페이스를 ActiveX 3D Web Viewer가 가져야만 했다.

3.3 DB Design 및 사용자 Web Interface

사용자가 원하는 모델 데이터 및 사용에 필요한 텍스처는 서버에서 제공하는 게시판형 인터페이스를 통해 클라이언트 측의 업로드, 다운로드가 이루어지고, 관련정보 및 모델 데이터, 텍스처는 서버 측 데이터베이스에 구축 관리된다. (<그림 6, 7> 참조)



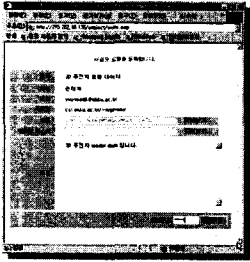
(a) 모델 등록 인터페이스 (b) DB Table Design
<그림 6> 데이터 모델 등록 인터페이스 (a) 및 DB Table Design (b)



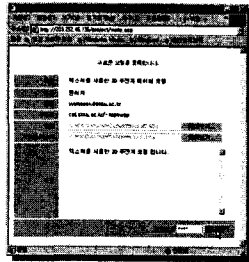
<그림 7> 서버 측 저장된 모델 데이터와 텍스처

4. 3D View Web Service 구현

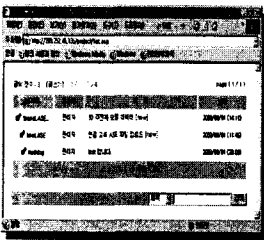
텍스처를 사용하지 않은 "teapot.ase"과 텍스처를 사용한 "teapot2.ase" 모델 데이터의 Web Service의 실행결과는 <그림 8>과 같다.



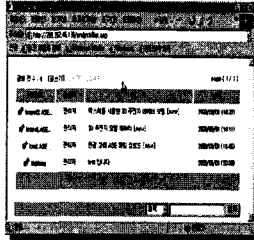
(a) 데이터 upload



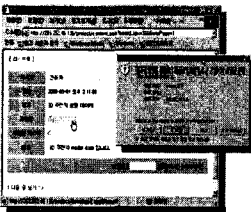
(a') 데이터 upload



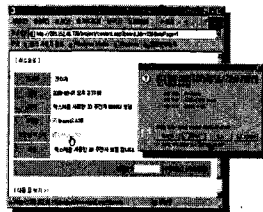
(b) 모델 데이터 등록



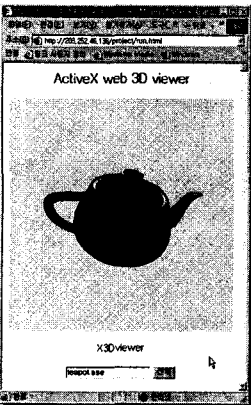
(b') 모델 데이터 등록



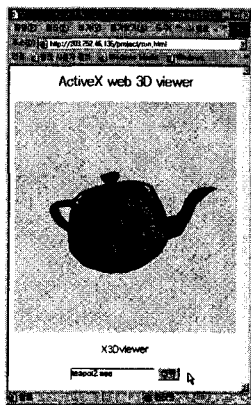
(c) 데이터 download



(c') 데이터 download



(d) 결과화면(teapot)



(d') 결과화면(teapot2)

<그림 8> 3D View Web Service 화면

5. 결론 및 향후 연구 방향

범용 모델 데이터와 인터넷을 이용하여 사용자 위주의 편의와 상호작용적인 Web Service를 구현하고자 하였다. 그러나 여기에 쓰인 ActiveX Component 제작 기술이 대다수의 마이크로소프트 운영체제를 사용하는 클라이언트만을 위한 플랫폼 종속적 기술이라는 점에서 이기종간의 플랫폼 독립적인 Web Service가 이루어지지 못하였다. 이 문제를 해결하기 위해서는 Java기반으로 구현해 나간다면 가능할 수 있겠다. 또한 렌더링 엔진에서의 여러 가지 최적화 알고리즘의 적용 및 기능 개선 또한 차후에 이루어져야 할 연구과제이다.

참고문헌

- [1] R. S. Wright Jr. and M. Sweet, *OpenGL SuperBible*, 2nd Ed., Macmillan Computer, 1999.
- [2] Foley, Van Dam, Feiner, Hughes and Phillips, *Introduction to Computer Graphics*, Addison Wesley, 1993.
- [3] Hill, Francis J., *Computer Graphics Using OpenGL*, 2nd Ed., Prentice Hall, 2000.
- [4] A. Bicalho and S. Feltman, *MAXScript & the SDK for 3D Studio MAX*, Sybex, 2000.
- [5] A. E. Walsh, M. Bourges-Sevenier, *Core Web 3D*, Prentice Hall, 2000.
- [6] D. Appleman, *Developing COM/ActiveX Components with Visual Basic 6*, SAMS, 1998.
- [7] Eberly, David H., *3D Game Engine Design*, Morgan Kaufmann, pp. 7-390, 2001.
- [8] M. Deloura, *Game Programming Gems*, Charles River Media, 2000.
- [9] A. Watt and W. Policarpo, *3D Games Realtime Rendering and Software Technology*, Addison Wesley, 2000.
- [10] T. Möller and E. Haines, *Real-Time Rendering*, AK Peters, pp. 7-143, 1999.
- [11] D. Hearn and M. P. Baker, *Computer Graphics*, 2nd Ed., Prentice Hall, pp. 271-468, 1994.
- [12] K. Hawkins and D. Astle, *OpenGL Game Programming*, Course Technology, pp. 28-266, 2001.