OpenGL을 이용한 3D 뷰어 개발

김병수, 강병익
건양대학교 IT 학부

A Development of 3D Viewer Using OpenGL

Byungsoo Kim, Byungik Kang
Dept. of IT, Konyang University

요 약

본 논문에서는 OpenGL을 이용한 삼차원 뷰어(Viewer)를 개발한다. 3D Max와 같은 3D 개발 툴에서 침대, 소파 등 3D 모델 라이브러리를 제작한 후 일어들이는 렌더링 모듈을 개발한다. 3D 모델 라이브러리에 직물 이미지 라이브러리를 매핑하는 모듈도 개발하여 직물 이미지와 유사한 효과를 낼 수 있도록 한다. 3D max에서 광원과 질감을 포함하여 렌더링한 후 개발되는 프로그램으로 데이터 손실없이 읽어들이는 것을 목표로 한다. 3차원 모델에 설계된 직물 이미지를 사용하여 3D 매핑된 후, 사용자가 확대, 축소, 이동, 회전 등의 효과를 줄 수 있게 하는 사용자 인터페이스 제공 및 3D 애니메이션 기능을 구현한다.

1. 서론

3D 기술은 불과 2~3년 사이에 엄청난 발전을 해왔다. 3D는 여러 분야에서 사용되고 그에 대한 구현도 활발하게 일어나고 있다. 가상현실은 이러한 3D 컴퓨터 그래픽 기술을 이용하여 사용자와 온라인으로 상호대화를 하는 기법이라고 정의할 수 있다[1,2]. 3D 게임과 애니메이션 시장의 급성장은 3차원 시각화 기능을 제공하는 렌더링 및 애니메이션 기술의 발전과 궤를 같이 하고 있으며, 여기에 3D 프로그래밍을 위한 OpenGL등의 게임엔진 언어가 나오는 등 기존 2D에서 애니메이션 및 게임을 한단계 넘어서는 3D Technology의 등장이 2D 게임 및 애니메이션 제작업체들을 속속 3D 시장으로 유인하고 있다. 3D 세계는 실제로 인간이 보는 세계와 유사하기 때문에 이를 표현하기 위해서는 다양한 기술들이 필요한데, 이 기술들은 화면에 3D 그래픽을 출력하는 렌더링 엔진, 객체를 움직이는 애니메이션 처리, 실제에 더욱 가깝게 만드는 텍스처 처리등이다. 본 논문에서는 3D로 제작된 가상의 방에서 소파나 침대 같은 3D 물체를 불러와 화면에 보여주는 3D 뷰어(viewer)를 OpenGL을 이용하여 개발한다. 각 객체에 필요한 직물 이미지 라이브러리를 3D MAX와 같은 툴에서 제작한 후 3D 뷰어로 매핑하는 모듈을 개발하여 직물 이미지와 유사한 효과를 낼 수 있도록 한다. 매핑된 후에 사용자가 확대, 축소, 이동, 회전 등의 효과를 줄 수 있게 하는 편리한 사용자 인터페이스 제공 및 3D 애니메이션 기능을 구현한다.

2. OpenGL 및 3D 모델

OpenGL은 강력한 저수준 렌더링 및 소프트웨어 라이브러리로, 거의 모든 주요 플랫폼과 다양한 하드웨어들이 지원한다. OpenGL은 게임서부터 모델링, CAD 등에 이르는 다양한 응용 프로그램들에서 쓰인다. 많은 게임들이 OpenGL을 자신의 핵심 그래픽 렌더링 엔진에 사용한다. OpenGL은 오직 저수준 렌더링 루틴만을 제공하는데, 이는 프로그래머에게 좀 더 세밀한 제어와 유연성을 제공하기 위한 것이다. OpenGL이 제공하는 루틴들을 이용해서 고수준 렌더링 및 모델링 라이브러리를 구축할 수 있도록 glu, glut, aux등의 다양한 유틸리티 라이브러리를 제공한다[3,4]

본 논문에서는 3D 제작툴로 작업된 침대, 쇼파, 커튼, 벽지, 테이블보, 자동차용시트, 방석 등의 3차원 모델을 Visual C++(MFC)와 OpenGL을 이용하여 화면에 보여지게끔 하는 3D 뷰어를 제작한다. 그리고 단순히 3차원 모델을 화면에 보여지게 하는 것뿐만

아니라, 화면에 보여지는 3차원 모델에 이미지를 매핑 시키고, 이동, 회전, 확대, 축소 등의 연산들을 지원한다. 이 연산들을 3차원 가상공간에 불러들여진 3차원 모델을 배치하게 된다. 그리고 다양한 조명들과 재질들을 적용하여 좀 더 사실적으로 3차원 모델들을 렌더링하게 된다. 이 뷰어는 직물을 제작하거나 가공하여 완제품을 만들지 않고도 화면상에서 시뮬레이션 시켜볼 수 있도록하는 장점이 있다. 지원하는 3D 모델은 *.3ds, *.ase, *.obj 이고, 지원하는 텍스처 이미지는 *.bmp, *.jpg, *.gif이다.

3. 3차원 로더(Loader)

3.1 3DS 파일 로더

3DS 로더를 제작하기 위해서는 3DS 파일 포맷 형식에 대한 정보가 있어야 한다. 3DS 파일 포맷 형식은 아직까지 공식적으로 발표된 적은 없으나 인터넷을 통해 대강의 정보를 구할 수 있다[5,6]. 3DS에 대해 공개된 정보를 살펴보면, 3DS 파일은 Chunk라고 불리는 블록으로 구성됨을 확인할 수 있다. 기본적인 Chunk 구조는 Chunk ID와 Chunk ID의 위치부터 다음 Chunk까지의 상대적 포인터로 구성되어 있다.

위치	크기	이 름
0~1	2	Chunk ID
2~5	4	Chunk ID부터 다음 Chunk까지 상대적 포인터(Chunk의 길이)

간단한 Chunk ID 구조와는 달리 3DS Chunk의 계층적 구조는 복잡한 편이다. 계층적 구조는 각각의 ID에 의해 형성된다. 최상위 Chunk ID는 4D4D로 언제나 파일의 첫 Chunk이다. 그리고 최상위 Chunk에 메인 Chunk가 존재한다.

3DS Loader를 제작하기 위해서는 Chunk 구조를 숙지하고 있어야 한다. 왜냐하면 Chunk 구조를 이용하여 3D 오브젝트 정보를 추출할 수 있기 때문이다. 이 Chunk를 이용하여 3D 오브젝트가 가지고 있는 재질에 대한 정보나 컬러 정보, 메시 데이터, 면 정보 등을 추출할 수 있다[5,6].

다음은 3DS 파일 로더 알고리즘이다.

```
// P_OBJECT, P_MATERIAL, P_CAMERA는 각각
//장면을 구성하고 있는 3D 오브젝트와 재질, 카메라
//의 속성을 기술하기 위한 구조체 포인터
// load3ds 함수를 통해 연결 리스트가 구성되면,
// 드로잉을 위한 루틴에서 초기화되어 전달된다.
// maxx, minx, maxy, miny, maxz, minz는 전체 3D
//오브젝트의 좌표값중 x, y, z 축에 대한 최대,
// 최소 값을 반환
int Load3ds(char *filename, P_OBJECT p_obj_head,
P_MATERIALp_mat_head,P_CAMERA p_cam_head,
P_LIGHT p_light_head, float *maxx, float *minx,
float *maxy, float *miny, float *maxz, float *minz,
float *r, float *g, float *b) {
   . . .
   bin3ds=fopen (filename,"rb"); // 이진3DS파일open
   fseek(bin3ds, 12L, SEEK_SET);
   fread(&version,sizeof(unsigned short),1,bin3ds);
   버전이 3 이하인  함수 종료하고, 파일 포인터를
   파일의 처음으로 이동시킨다.
   int continue = 0;
   while(continue == 0) {
      if(현재 Chunk ID를 읽는다. == MAIN3DS) {
         fseek(bind3ds, 2, SEEK_SET);
         ReadMainChunk();
      } else coutinue = 1;  // 진행할 것이 없다
      continue = 0;   // 진행할 것이 있다.
   }
   ....
   전체장면을 구성하는 좌표값의 최대,최소값을 반환
}
unsigned long ReadMainChunk() {
   청크의 시작옵셋을 찾고, 청크의 바이트를 읽는다.
   재질의 헤더를 초기화한다.
   while (end_found==FALSE) {
      temp_int = 하위 청크의 ID를 읽는다.
      switch (temp_int) {
         case KEYF3DS :
            스튜디오에서 애니메이션에서 필요한
            정보를 가지고 온다.
            break;
         case EDIT3DS :
            오브젝트가지고 정보를 추출한다.
            재질 정보 추출, 컬러 정보 추출,
            메시 데이터 추출, 면 정보 추출 등
            break;
      }
      읽어온 바이트의 개수를 업데이트.
   }
   새로운 청크로 포인터를 이동시키고, 현 청크의
   크기를 리턴.
}
```

3.2 Ase 및 OBJ 파일 로더

Ase 파일 포맷도 3DS와 마찬가지로 블록 구조로

되어 있지만, 복잡한 Chunk 구조를 가진 3DS와 다르게 간단한 구조로 되어 있다. 그리고 일반 텍스트 편집기로 로드하여도 그 내용을 쉽게 알아볼 수 있게 되어있다. ASE 파일의 구조는 일반적으로 크게 SCENE, MATERIAL_LIST, GEOMOBJECT, LIGHTOBJECT의 블록으로 되어있다[7].

SCENE : 장면에 대한 정보가 들어 있다.
LIGHTOBJECT : 조명의 위치, 밝기 등 조명에 관한 정보들이 들어있다.
MATERIAL_LSIT : 재질에 관한 정보들이 들어 있는 블록이다.

OBJ 파일 포맷은 텍스트 파일 포맷이다. 그래서 OBJ 파일은 간단한 텍스트 에디터로도 쉽게 수정할 수도 있다. OBJ 파일은 각 라인마다 첫 번째 문자에 따라 명령 형식이 다르다. 만약 첫 문자가 '#'이거나, 공백 라인이면 그 라인은 무시가 된다. 다음 몇가지 OBJ 파일 포맷에서 사용되는 명령 형식이다[8].

#a comment line
이 라인은 항상 무시된다. 통례적으로 OBJ 파일의 제일 첫 머리는 외부 프로그램에 의해 쓰여진다.

v x y z
정점 명령이다. 이것은 3D-좌표계에 표현되는 정점을 기술한 것이다.

vt u v [w]
텍스처 정점 명령은 UV 매핑을 기술한다. 0~1 사이의 값을 가지고 텍스처 매핑을 한다.

vn x y z
정점 법선 명령은 법선 벡터를 기술한다. 정점 법선 명령은 많이 사용되지는 않는다.

f v1[/vt1][/vn1] v2[/vt2][/vn2] v3[/vt3][/vn3]
면 명령은 정점 리스트로부터 다각형을 기술한다.

4. 3차원 뷰어 만들기

지금 만들려는 3D 뷰어는 위에서 말 했듯 3가의 3D 모델을 지원하고, 3가지의 텍스처 이미지 파일 포맷을 지원한다. 뷰어는 위에서 만들어진 DLL을 조합하여 만들어진다.

bmp 파일은 BITMAPFILEHEADER와BITMAP INFOHEADER 그리고 RGB 값으로 구성된다. BITMAPFILEHEADER와 BITMAPINFOHEADER는 bmp가 가지고 있는 기타 정보(bmp 파일 형식, 이미지 너비와 높이, 압축방식, 컬러 비트수 등)들을 가지고 있다. 그러나 텍스처 맵핑을 시키기 위해서는 기타 정보들은 필요 없고, 이미지의 너비와 높이, RGB 값만을 필요로 한다.

아래는 bmp 파일에서 RGB 값을 가져오는 알고리즘이다[9,10].

```
// bmp RGB 값을 저장할 구조체를  정의
struct rgb_format {
        float *rgb;
        int  xsize;
        int  ysize;
};
// 제대로 데이터를 읽어오면 0을 반환, 그렇지 않으면 1을 반환한다.
int load_BMP(char *f, GL_RGB_FORMAT* image){
   BITMAPFILEHEADER   BMFH; // 파일 헤더
   BITMAPINFOHEADER  BMIH; //파일 정보헤더

   파일을 열고 파일 헤더의 파일 형식 비트맵 파일의
   크기 등의 내용을 읽는다.

   파일이 BMP 형식이 아닐 경우 1을 반환

   파일 정보 헤더의 구조체의 크기, 비트맵의 너비,
   비트맵의 높이, 색상 평면이 수, 픽셀당 비트수
   (1, 4, 8, 16, 24, 32 비트) 등의 내용을 읽는다.

   bmp의 RGB 값을 읽어온다.
   if(BMIH.biBitCount == 24) { // 24bit일 경우
        RGB 값을 rgb_format의 구조체 멤버 변수
        rgb에 저장한다.
   }
   파일을 닫고 0을 반환
}
```

먼저 불러온 3D 모델들을 관리할 수 있도록 하나의 ModelList라는 리스트를 구성한다. 그리고 불러온 3D 모델들의 정보를 ModelList로 Node에 추가한다.

그리고 불러온 모델들을 화면에 보여주기 위해 화면을 다시 그리게 될 때는 이 ModelList 리스트를 순회하면서 그동안 불러온 모든 3D 모델을 화면에 보여지게 한다.

그리고 화면에 보여지는 모델중 원하는 모델을 선택할 수도 있고, 선택된 모델에 이동, 회전, 확대, 축소를 할수 있게 키 이벤트 및 마우스 이벤트의 구현을 하고, 텍스처 맵핑도 할 수 있게 메뉴를 구성한다. 그리고 배경에 보이는 3D 모델은 3ds 파일 포맷 형식

만을 지원하게 된다.

그림 1은 3ds 파일에서 와이어프레임을 불러와 뷰어로 출력한 화면이다.

원하는 객체를 클릭한 후 여러 종류의 bmp 파일을 그림 2에서 보여주고 있다.

그림 3은 방 안에서 침대, 소파등의 인테리어를 구성한 그림이다. 각 객체를 클릭한 후 오른쪽 버튼으로 회전 이동 텍스처 매핑을 할 수 있도록 구성하였다.

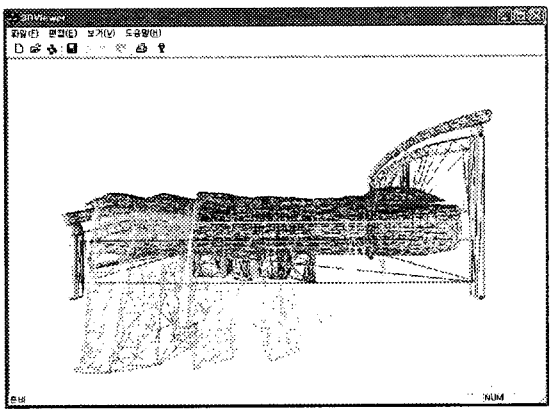

그림 1. 와이어프레임

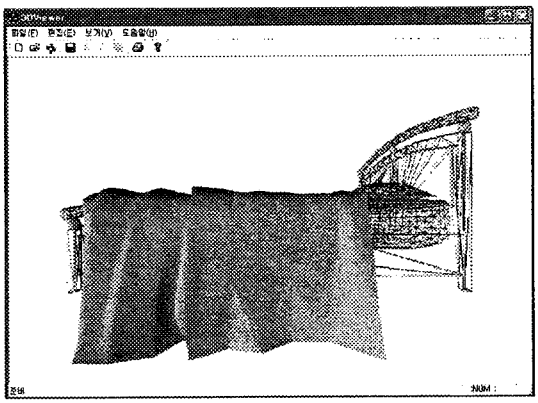

그림 2. bmp 파일을 원하는 객체에 추가

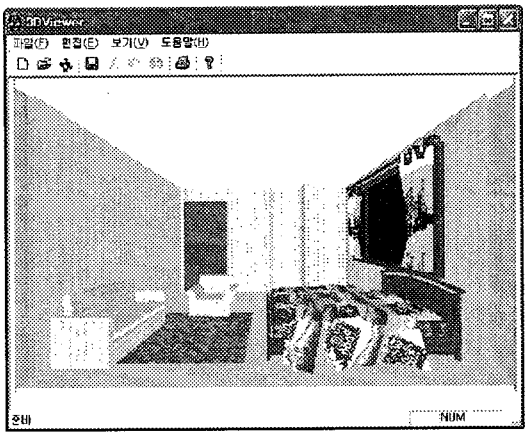

그림 3. 침대, 소파등의 인테리어 구성

5. 결론

본 논문에서는 3차원 뷰어를 일상 생활에 필요한 침대, 소파와 같은 가구들을 중심으로 하여 구성하였다. 각 객체마다 원하는 텍스처를 매핑하여 렌더링 한 후 회전 이동 확대 축소와 같은 여러가지 기능을 할 수 있도록 옵션을 추가하였으며, 조명과 재질을 넣어 보다 현실감있는 뷰어가 될 수 있도록 하였다.

실제 직물을 표현할 수 있는 Bump 같은 기능이나 3D MAX 상에서 표현될 수 있는 조명은 아직 본 논문에서 구현된 뷰어에서는 제한적이다. 따라서 앞으로는 이와 같은 보다 실제감있는 렌더링을 할 수 있도록 보완할 예정이다.

[참고문헌]

[1] www.dreamscape.co.kr

[2] www.web3d.co.kr

[3] http://www.opengl.org/users/about/index.html

[4] http://www.sgi.com/software/opengl /overview.html

[5]http://www.the-labs.com/Blender/3DS-details.html

[6] http://astronomy.swin.edu.au/~pbourke /geoformats/3ds

[7] http://astronomy.swin.edu.au/~pbourke /geoformats/ase

[8] http://astronomy.swin.edu.au/~pbourke /geoformats/obj

[9] Kevin Hawkins and Dave Astle, OpenGL Game Programming, Prima Inc.

[10] Visual C++ 6.0 Studio MSDN