

OpenGL을 이용한 한양대학교 안산캠퍼스 가상 캠퍼스 가이드 맵 구현

최태영, 이효국, 김동우, 김재형, 손진현
한양대학교 컴퓨터공학과

e-mail : fighting-taeng@hanmail.net, hyofresh@nate.com, k_sd21@hotmail.com, jhkim@cse.hanyang.ac.kr
jhson@hanyang.ac.kr

Design for Virtual Campus Guide Map for Hanyang University at Ansan using OpenGL

Tae-Young Choi, Hyo-Gook Lee, Dong-Woo Kim, Jae-Hyung Kim, Jin Hyun Son
Department of Computer Science and Engineering, Hanyang University in Ansan

요 약

본 연구는 OpenGL을 이용하여 3차원 그래픽으로 가상캠퍼스 맵을 구현한다. 매핑 효과, 블렌딩 이론등을 적용하여 사실적인 물체를 구현하는 것을 목표로 한다. 매핑 효과는 조금 더 사실적인 캠퍼스 모습을 보여준다. 그리고 가상캠퍼스에 사용될 지형지물은 그래픽 기법과 블렌딩 이론을 적용하여 구현함으로써 보다 수학적인 구조를 보여준다. 본 논문은 데이터의 용량과 렌더링 시간을 줄이고, 전체적인 그래픽의 향상을 위한 알고리즘을 개발하는데 중점을 둔다.

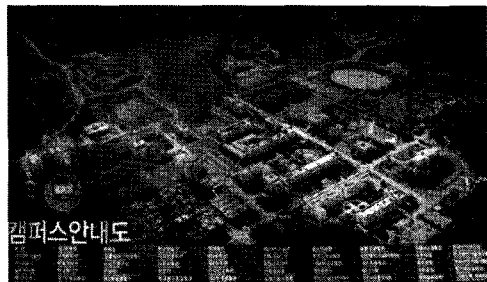
키워드 : OpenGL, Mapping, ASE, Blending

1. 서론

컴퓨터 기술은 갈수록 비약적인 발전을 하고 있으며, 집에서 온라인 쇼핑, 사이버 대학교, 사이버 박물관등 가상체험의 기법은 '컴퓨터 그래픽스'라는 분야에 적용되어 집약적인 발전을 하고 있다. 일반적인 사진조각을 3D물체에 입히는 것을 매핑이라고 한다. 그러나 실제 OpenGL를 이용해서 사이버 캠퍼스를 구현하면 그래픽 구현의 한계에 부딪히게 된다. 매핑 기법이 사실적이긴 하나 캠퍼스의 자연물은 매핑 기법을 적용하여 표현하기에 한계가 있기 때문에 본 논문에서는 블렌딩(Blending)기법을 이용한 텍스처 매핑을 적용하여 자연물의 현실감을 더하였다. 그래픽 라이브러리인 OpenGL을 사용하여 매핑이 된 캠퍼스 건물과 블렌딩 이론을 적용한 지형지물들로 더 사실적인 캠퍼스를 구현하여 가상으로 체험할 수 있게 하는 엔진 개발이 이번 연구의 목표이다.

현재 한양대학교 안산캠퍼스 안의 캠퍼스 안내도 <그림 1>은 인공위성의 사진을 이용한 2D의 평면 지도이다. 그러나 처음 학교를 방문하는 방문자에게는 시각적이지 못하고

글자또한 보기가 어렵다. 그리고 우리학교 건물 구조상 비슷한 구조의 건축물들이 많아서 그 건물을 찾아 갈려고 해도 식별이 불가능하는 경우가 있을 수 있다. 또한 방대한 캠퍼스의 크기로 인해 2D의 평면지도로는 표현의 한계성이 있다. 그래서 우리는 기존의 2D의 지도를 가상의 3D의 가이드 맵으로 구현해 보기로 했다.



<그림 1> 기존의 캠퍼스 가이드 맵

II. 관련 연구

2. 1 OpenGL(Open Graphics Library)

OpenGL은 그래픽 하드웨어에 대한 인터페이스를 제공한다. OpenGL은 강력한 저수준 렌더링 및 모델링 소프트웨어 라이브러리로, 거의 모든 주요 플랫폼들과 다양한 하드웨어들이 지원한다. OpenGL은 게임에서부터 모델링, CAD등에 이르는 다양한 응용프로그램들에서 쓰인다. OpenGL은 오직 저수준 렌더링 루틴만을 제공하는데, 이는 프로그래머에게 좀 더 세밀한 제어와 유연성을 제공하기 위한 것이다. OpenGL이 제공하는 루틴들을 이용해서 고수준 렌더링을 하기 위해서는 모델링 라이브러리(OpenGL Utility Library, GLU)를 이용한다. GLU는 대부분의 OpenGL 배포판에 포함되어 있다. 그리고 DirectX와 달리 OpenGL은 순수한 그래픽 라이브러리이다. 사운드나 입력, 네트워킹 등 그래픽과 직접 관련되어 있지 않은 것들은 OpenGL에 들어 있지 않다.[1]

OpenGL은 그래픽 하드웨어가 제공하는 모든 기능들에 접근할 수 있는 수백개의 함수들을 모아 놓은 것이다. 내부적으로 OpenGL은 하나의 상태 기계(state machine)처럼 작동한다. 여기서 상태 기계란 OpenGL에게 무엇을 할 것인지를 알려주는 상태들을 모아 둔(그리고 그것들을 관리하는) 것으로 생각 할 수 있다. 개발자는 API를 이용해서 상태기계의 여러 측면들(현재 색상, 조명, 혼합 등)을 조사하거나 설정할 수 있다. 만약에 하나 또는 몇개의 상태들이 잘못 설정되어 있으면 최종적인 결과가 전혀 예상과는 다르게 나오는 경우가 많으므로, OpenGL을 사용할 때에는 여러 상태들의 의미나 효과에 대해 잘 알고 있어야 한다.

(1)OpenGL 유틸리티 라이브러리(GLU)

OpenGL 유틸리티 라이브러리(OpenGL Utility, GLU)는 고수준 함수들을 제공하는 OpenGL에 대한 일종의 보안물이다. GLU는 OpenGL 함수들에 대한 간단한 래퍼(wrapper)들에서부터 고급 렌더링 기법들을 지원하는 복잡한 요소들에 이르는 다양한 기능들을 제공한다. GLU가 제공하는 기능들은 다음과 같다. 2D이미지 확대/축소, 구, 원통, 원반 등을 포함한 3D 객체들의 렌더링, 하나의 이미지로부터 자동적 mip맵(mipmap) 생성, 곡면, 비볼록 다각형, 오브젝트의 변환과 행렬등을 지원한다.[2]

(2)OpenGL 유틸리티 툴킷(GLUT)

GLUT은 OpenGL Utility Toolkit의 약자로 대부분의 주요 플랫폼들에서 사용할 수 있는 일련의 보조 지원 라이브러리의 집합이다. OpenGL자체에는 창관리, 메뉴, 이력같은 기능들이 존재하지 않는데 그런 부분을 보충하는 것이 바로 GLUT이다. GLUT은 그런 부분들에 대한 기본적인 기능을 제공하며, 그러면서도 플랫폼 독립성을 유지한다. 플랫폼 독립성 덕분에 예를 들어 GLUT 기반 프로그램을 거의 수

정하지 않고도 윈도우즈에서 유닉스로 이식하는 것이 가능하다. GLUT는 사용하기와 쉽고 배우기도 쉽지만, 운영 체제가 제공하는 모든 기능들을 제공하는 것은 아니기 때문에 주로 데모 프로그램이나 간단한 프로그램에 적합하다.

2. 2 텍스처 매핑(Texture Mapping)

3D로 구현된 가상 캠퍼스의 사실적인 표현을 위해 지형과 건물에 매핑을 한다. 정확한 좌표 매핑을 위해서 3D MAX에서 매핑 기능을 이용해서 필요한 X, Y, Z 값에 대한 매핑 정보를 나타내는 3D MAX의 ASCII Scene Export(ASE) 기능을 통해서 좌표를 추출한다.

3D MAX에서 생성된 오브젝트 데이터 .ASE 파일을 프로그램에 필요한 오브젝트의 점(Vertex) 정보, Face의 매핑 소스에 대한 점 정보, Face 정보를 추출하여 생성된 .cg파일의 인덱스(Index)정보를 [표 1]의 함수를 통해 렌더링을 하고 텍스처 매핑을 한다.

〈표 1〉 OpenGL 텍스처 매핑

매핑 순서	소 스
BMP파일 Load	AUX_RGBImageRec *texRec(0) =LoadBMPFile("file.bmp")
객체 생성	glGentextures(1, &tex(0))
이름, 데이터 연결	glBindTexture(GL_TEXTURE_2D, tex(0))
매핑 적용	glEnable(GL_TEXTURE_2D)
매핑 적용 해제	glDisable(GL_TEXTURE_2D)

(1)ASE 파일 포맷

3D MAX로 디자인된 모델을 OpenGL기반으로 컴퓨터 화면에 보여주기 위해서는 3D MAX모델의 모든 오브젝트 정보를 담고 있는 ASE(ASCII Scene Exporter)포맷을 이해해야 한다. 물론, ASE외에 다양한 포맷이 있지만 이해하기 쉽고 사용이 편리한 ASE로 시작하는 것이 유리하다.[3] ASE 파일은 크게 3개의 단위로 이루어져 있다. 각 단위는 씬 정보(*SCENE), 재질 정보(*MATERIAL_LIST), 오브젝트 정보(*GEOMOBJECT)이며, 오브젝트 정보의 경우 추가로 몇 종이 더 있는데 주로 *HELPEROBJECT를 들 수 있다. 그중에서도 오브젝트 정보 단위 중에 *MESH 정보(실제 오브젝트를 이루는 값)의 *MESH_NUMVERTEX, *MESH_NUMFACES, *MESH_VERTEX_LIST, *MESH_FACE_LIST 값만 비어있는 .CG 파일에 저장을 하고 OpenGL를 이용해서 불러들인다.[4]

(2)ASE 파일에서 .CG 파일로의 파싱

그중에 OpenGL 매핑을 위해서는 Vertex, Face, TVertex, TFace값만 따로 추출한 후 임의의 파일 .CG에 저장을 하고 OpenGL를 이용해서 불러 들어야 한다. 여기서

1.Vertex: 오브젝트의 점들의 집합, 2.Face: 폴리곤의 구성정보, 3.TVertex: Vertex의 텍스처 좌표, 4.TFace: Face의 텍스처 좌표가 있어야 OpenGL 에서 오브젝트를 불러올 수가 있다. 덧붙이자면 하나의 Vertex가 몇 개의 TVertex를 가질 수도 있다는 이야기이다.(이는 Vertex를 TVertex만큼 늘려줘야 한다는 이야기와도 같다.)

*MESH는 실제 오브젝트의 이루는 Vertex집단이다. *MESH_NUMVERTEX는 MESH를 이루는 Vertex의 개수, *MESH_NUMFACES는 MESH를 이루는 페이스의 개수를 뜻한다. *MESH_VERTEX는 순번과 X, Y, Z의 좌표값을 뜻한다. 당연히 *MESH_NUMVERTEX 개수만큼 있다.[5]

Face는 표면을 재질을 나타내는 리스트로 Vertex 3개로 이루어지는데 각 숫자가 Vertex리스트의 해당 Vertex 인덱스이다. 또한 *MESH_MTLID는 서브 재질의 인덱스이다.

2. 3 BMP 파일 읽기

BMP파일을 메모리에 읽어 들일 때에는 우선 비트맵 파일 헤더를 읽어서 주어진 파일이 실제로 BMP 파일인지 확인 한 후, 비트맵 정보 헤더를 읽어서 그 뒤에 나오는 이미지 데이터의 크기를 알아낸다. 그리고 그 크기에 맞게 메모리 블록을 할당한다. 실제 이미지 데이터를 읽어 메모리 블록에 넣는다. 그런데 주의 할 점은, 24비트 BMP 파일에는 RGB가 아니라 BGR 형태로 데이터가 들어있다는 점이다. 따라서 OpenGL에서 제대로 된 색상을 나타내기 위해서는 각 픽셀 데이터의 R과 B를 서로 바꿔 줘야한다.

2. 4 블렌딩(Blending) 기법

OpenGL에서 대부분의 특수효과는 Blending를 이용한다. 블렌딩은 이미 화면상에 그려진 픽셀의 색과 이제 바로 같은 위치에 그려질 픽셀의 색의 조합하는 방식이다. 어떤 식으로 색상을 조합하는 지는 알파값과 블렌딩 함수에 의해 정해진다. 알파 값이란 보통 색상을 지정할 때 4가지 구성요소 중 마지막 네 번째 값이다. 지금까지 색상을 지정하는 방식으로 GL_RGB를 사용했었는데 여기에는 알파값이 없다. 알파값의 추가를 위해 GL_RGBA를 사용할 수 있다. 그리고 알파값을 포함한 색상을 지정하기 위해 glColor3f대신 glColor4f를 사용할 수 있다. 알파값 0.0은 완전한 투명이고 1.0은 완전한 불투명하다.

블렌딩 요소인 원본(Source)픽셀값과 대상(Destination) 픽셀값이 블렌딩 방식을 결정하는데, OpenGL에서 두 픽셀 간의 블렌딩 결과를 계산하기 위해 <표 2>를 이용한다.

<표 2> 블렌딩 결과 계산

$$\begin{matrix} RsSr + RdDr, & GsSg + GdDg, \\ BsSb + BdDb, & AsSa + AdDa \end{matrix}$$

s와 d요소는 블렌딩 요소이다. 이러한 값들이 어떤 방식으로 블렌딩 할 것인지를 지정한다. 또한 r, g, b, a의 첨자는 색의 3요소(빨강, 초록, 파랑)과 알파값이다.

원본 픽셀값으로는 원본 알파값(As, As, As, As)이며 대상 픽셀값으로는 1 - 원본 알파값((1, 1, 1, 1) - (As, As, As, As))이다. <표 3>를 이용해 투명, 반투명 효과를 나타낸다.

<표 3> 반투명 효과를 나타내는 식

$$\begin{matrix} RsAs + Rd(1 - As), & GsAs + Gd(1 - As), \\ BsAs + Bs(1 - As), & AsAs + Ad(1 - As) \end{matrix}$$

OpenGL에서는 <표 4>에 있는 함수들을 이용하여 원본 이미지와 대상 이미지 간의 블렌딩 기법을 적용할 수 있다. 첫 번째 이미지의 원본픽셀에 대상 픽셀을 곱하고 대상 픽셀을 0으로 하는 (GL_DST_COLOR, GL_ZERO)로 블렌딩 하여 텍스처 매핑을 하고, 두 번째 이미지에 원본픽셀과 대상픽셀을 그대로 하는(GL_ONE, GL_ONE)로 블렌딩 하여 텍스처 매핑을 함으로써, 이미지의 원하는 부분을 제외한 부분에 투명한 효과를 주어 선택이미지를 추출한다.

<표 4> OpenGL 블렌딩 함수

블렌딩 함수 인자	의 미
GL_ZERO	0, 0, 0, 0으로 셋팅
GL_ONE	현 색상을 그대로 사용
GL_DST_COLOR	배경색과 컬러를 혼합
GL_SRC_ALPHA	알파값을 혼합
GL_DST_ALPHA	배경의 알파값을 혼합

2. 5 시점 변환

1차원 시점을 3차원 시점으로 변환해 할때 점(x, y, z)을 <표 5>에 적용하여 x, y, z방향의 이동거리만큼 이동시켜 새로운 점(x', y', z')을 얻어 시점 변환을 해 준다. 여기서 tx, ty, tz 는 x, y, z축에 대한 이동거리를 나타낸다.

<표 5> 시점이동 및 변환

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

2. 6 충돌 처리

충돌을 처리하는 이론은 여러 가지가 있지만 그중에서도 바운딩 박스 이론이 가장 간단하고 많이 응용되어 쓰이고 있다. 물체를 둘러싼 박스를 생성하고 그 박스 안에 다른 물체가 들어오면 서로 충돌했다고 판단한다. 우리는 AABB(Axis Alien Bounding Box)기법으로 물체의 축을 따라서 박스를 만드는 방법을 이용했다.

2. 7 LOD(Level of Detail)기법

카메라가 있는 지역을 중심으로 거리 값에 따른 메쉬값을 생성 한 후 간략화해서 레벨이 다른 지형의 상, 하, 좌, 우를 분리한 후 각 부분을 따로 처리한다. 각 부분 경계면의 메쉬 값을 변화시켜서 왜곡되는 현상을 제거했다. 생성된 메쉬값을 이용하여 0 ~ 5단계로 나눈 뒤 단계별로 픽셀 값을 조정 해주면 기존 보다 향상된 속도를 얻을 수 있다.

<표 6> LOD 지형 적용

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

0 : 카메라가 위치 하는 면
 1 : 1단계 LOD 적용 되는 지형
 2 : 2단계 LOD 적용 되는 지형
 최 소 : 0단계
 최 대 : 5단계

III. 프로그램 구동

[표 7]에 나타난 키를 이용하여 기능을 이용할 수 있으며 기타 움직임은 키보드의 상, 하, 좌, 우 방향키를 이용하여 조작할 수 있다.

<표 7> 프로그램 조작

키 보 드	기 능
숫자 1	정면 시점
숫자 2	비행 시점
Q, A	위, 아래 사용자 정의 시점
Z	걷기 / 달리기
F1	전체 / 부분 화면

IV. 결론

현재 정문의 학교 캠퍼스 안내도(Campus Guide Map)를 대신하여 대형 LCD 모니터나 터치스크린을 이용한 캠퍼스 가이드로써의 활용을 기대할 수 있다. 또한 학교홈페이지의 캠퍼스 안내부분에 3D 가상 캠퍼스 메뉴를 추가해서 홈페이지를 방문하는 사용자들에게 더욱 효과적으로 정보의 전달

을 이끌어낼 수 있다. 이외에도 학교를 배경으로 한 게임, 영화 등 다양한 3차원그래픽 분야에 활용할 수 있을 것이며 또한 자동차 네비게이션, 휴대폰 기기등에 적용해 나갈 수 있을 것이다.

기존 구현한 프로그램에 음성기능을 추가한다면 좀 더 효율적으로 사용자에게 정보전달 효과를 기대할 수가 있을 것이다. 게다가 장애인 사용자를 위해서도 활용을 할 수 있게 된다. 현재 학교 내의 편의시설 중 장애인들을 위한 시설은 극히 일부이다. 심지어는 시각장애인을 위한 안내 시설도 없는 것이 현실이다. 이 시스템을 발전시켜 적용한다면 장애인을 위한 음성인식 안내도로써의 활용도 기대할 수 있을 것이다.

참고문헌

- [1] OpenGL 프로그래밍 가이드 제3판 OpenGL 공식 학습 가이드 Woo, Mason 외 공저, 남기혁 외 공역
- [2] OpenGL SUPERBIBLE Richard S. Wright, Michael Sweet 저, 남기혁 역 인포북
- [3] 디자이너를 위한 3DS MAX + PHOTOSHOP 남현우 저, 웰 북
- [4] 따라 하기 쉬운 3DS MAX 전현철 저, 일진사
- [5] ASE 파일 파싱과 모델 데이터베이스 연동을 통한 3D 웹 서비스 설계 및 구현(정보처리학회 논문지 제11-D권 제 6호 통권 제 95호(2004. 10) p.1327-1334)