

Texture Mapping

321190
2007년 봄학기
5/25/2007
박경신

OpenGL Texturing

- OpenGL에서 텍스처 맵핑(texture mapping)을 위한 3 단계
 - 텍스처 활성화
`glEnable(GL_TEXTURE_2D)`
 - 텍스처 맵핑방법 (랩핑, 필터 등) 정의
`glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT)`
`glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT)`
`glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR)`
`glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR)`
`glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_RGB, GL_UNSIGNED_BYTE, imageData)`
 - 텍스처 좌표 (texture coordinates) 지정
`glTexCoord2f(0, 0);`
`glVertex3f(-1.0, -1.0, 0.0);`
`glTexCoord2f(1, 0);`
`glVertex3f(1.0, -1.0, 0.0);`

OpenGL Texture Names

- 텍스처의 이름(name) 지정하기 (textureID)
`GLuint textureID;`
`glGenTextures(1, &textureID);`
`glBindTexture(GL_TEXTURE_2D, textureID);`
`glTexImage2D(...);`
`glBindTexture(GL_TEXTURE_2D, 0);`
...
`glBindTexture(GL_TEXTURE_2D, textureID);`

glTexImage2D

- `glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid *pixels);`
 - target: `GL_TEXTURE_2D`
 - level: 텍스처 맵의 다양한 해상도를 지원하기 위해 설정.
 - 1개의 해상도를 지정하려면 1로 설정.
 - mip맵(mipmapping)에 사용
 - 각 텍스처를 위한 다수의 크기를 가지고 있는 이미지를 사용하지 않는다면 0으로 지정
 - internalFormat: 일반적으로 format과 같음. RGB라면 3, RGBA라면 4로 설정
 - width, height: 텍스처 이미지의 너비와 높이는 2의 자승으로 되어야 함 (즉, 2, 4, 8, 16, 32, 64, 128, 256, 512, etc)
 - border: 텍스처의 경계선 너비를 지정. 보통 0이고 이미지 데이터가 border를 가지고 있으면 1로 지정.
 - type: 텍스처 이미지 데이터의 형식을 설정.

glTexImage1D

- `glTexImage1D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid *pixels);`
- `glTexImage2D()` 함수는 2차원 텍스처 이미지를 정의하고 `glTexImage1D()` 함수는 1차원 텍스처 이미지를 정의한다. `glTexImage1D()` 함수와 `glTexImage2D()` 함수의 사용방법과 인자의 의미는 거의 동일하며 `glTexImage2D()` 함수에만 이미지 텍스처의 `height`(높이) 인자가 추가된다.

glTexSubImage2D

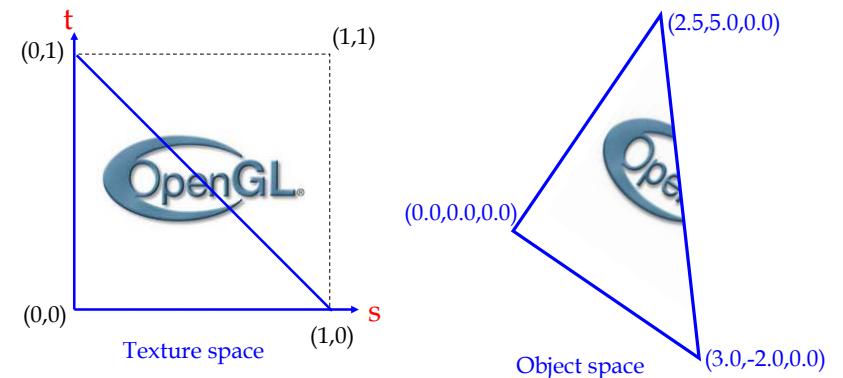
- 텍스처 크기가 2의 승수(e.g., 64x64, 128x256, ..)가 아닌 경우 텍스처 이미지의 일부분만 읽어 들일 때 사용함
- `glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid *pixels);`

OpenGL Texture Coordinates

- 텍스처 맵핑이 사용되려면 객체에 텍스처 좌표를 정의해야 한다.
- GLUT teapot은 텍스처 좌표를 포함하고 있다.
- GLU quadrics도 텍스처 좌표를 옵션으로 정의할 수 있다.
 - `gluQuadricTexture(quadric, GL_TRUE)` 를 사용하여 텍스처맵핑 활성화

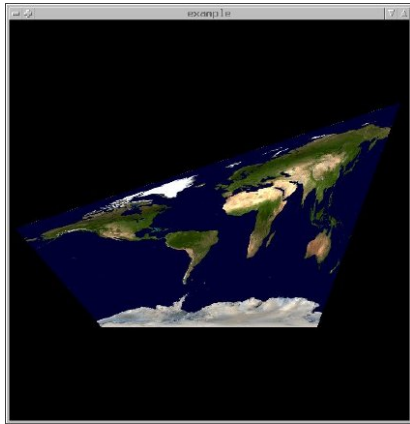
OpenGL Texture Coordinates

- OpenGL에서 텍스처 좌표는 텍스처 이미지의 각 방향 (S, T)의 0부터 1의 영역으로 형성됨
- OpenGL에서 각 정점별로 텍스처 좌표를 지정해야 함.
- 정점의 텍스처 좌표가 객체의 표면에 보간되어 나타남.



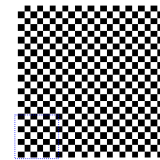
OpenGL Texture Coordinates

- 텍스처는 다각형에 고르게 입힐 필요는 없음.
- 기하학적 모델이나 텍스처 좌표 사용에 따라서, 이미지가 때론 왜곡되게 나타날 수도 있음.



OpenGL Texture Filtering

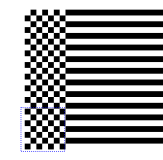
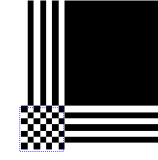
- OpenGL에서 텍스처 좌표가 $[0, 1]$ 영역을 벗어날 경우에 texture wrapping 방법으로 정의함: Repeat, Clamp
- 사각형에 4개의 텍스처좌표를 (0,0), (0,3.5), (3.5,0), (3.5,3.5) 로 정의한 예



Repeat



Clamp



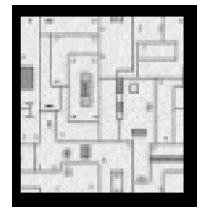
Repeat & Clamp

OpenGL Texture Filtering

- Mipmap은 이전 mipmap 너비와 폭의 절반 크기임.
 - 텍스처가 작아질 수록, 보다 많은 텍셀이 한 픽셀에 적용되어야하므로 GL_NEAREST나 GL_LINEAR 필터가 정확한 계산결과를 만들지 않을 수 있음. 따라서 객체가 움직일 때 텍스처가 flickering하게 나타날 수 있음.
 - mip맵은 이런 flickering문제를 줄여줄 수 있음. 그러나 일반적으로 희미하게 보임.



GL_LINEAR



GL_LINEAR_MIPMAP_LINEAR

OpenGL Texture Filtering

- 텍스처 맵핑을 위한 필터링 방법
 - 최근점 필터 (nearest neighbor filter)
 - GL_NEAREST
 - 이선형 필터 (bilinear interpolation filter)
 - GL_LINEAR
 - 삼선형 필터 (trilinear interpolation filter) - mipmap filter
 - GL_LINEAR_MIPMAP_LINEAR
 - 혼합 필터 (hybrid filter)
 - GL_NEAREST_MIPMAP_LINEAR
 - GL_LINEAR_MIPMAP_NEAREST
 - GL_NEAREST_MIPMAP_NEAREST

OpenGL Texture Filtering

- `glTexParameter{if}v(GLenum target, GLenum pname, TYPE *param);`
 - `GL_TEXTURE_WRAP_S, GL_TEXTURE_WRAP_T`
 - `GL_CLAMP, GL_REPEAT`
 - `GL_TEXTURE_MAG_FILTER`
 - `GL_NEAREST, GL_LINEAR`
 - `GL_TEXTURE_MIN_FILTER`
 - `GL_NEAREST, GL_LINEAR` (Mipmap을 사용하지 않는 경우)
 - `GL_NEAREST_MIPMAP_NEAREST, GL_NEAREST_MIPMAP_LINEAR, GL_LINEAR_MIPMAP_NEAREST, GL_NEAREST_MIPMAP_NEAREST`
 - `GL_TEXTURE_BORDER_COLOR`
 - [0.0, 1.0]영역의 값
 - `GL_TEXTURE_PRIORITY`
 - 0 또는 1

OpenGL Texture Transformations

- 텍스처 좌표의 변환
 - 기하물체의 정점에 변환하듯이 텍스처 좌표에 이동 (translation), 회전 (rotate), 크기변환 (scaling)을 적용함
 - `glMatrixMode(GL_TEXTURE)`를 사용하여 정점이 아닌 텍스처 좌표에 변환을 적용함을 지시함

```
glMatrixMode(GL_TEXTURE);
glLoadIdentity();
glTranslatef(0.1, 0.05, 0);
glRotatef(30.0, 0, 0, 1);
glMatrixMode(GL_MODELVIEW);
// geometry with texture coordinates
```

OpenGL Texture Transformations

- 텍스처 좌표의 변환
 - 텍스처 좌표에 크기변환 (scaling)을 적용할 때 물체도 또한 같은 크기변환을 해야함

```
glMatrixMode(GL_TEXTURE);
glLoadIdentity();
glScalef(size, 1, 1);
glMatrixMode(GL_MODELVIEW);
glScalef(size, 1, 1);
// geometry with texture coordinates
```

OpenGL Texture Movies

- 텍스처 이미지 sequence를 이용하여 flipbook 애니메이션 제작
 - `initTexture()` 함수에서 전체 텍스처 이미지를 읽어 들임
 - `idle()` 함수에서 `currentTextureID`를 update함
 - `drawTexture()` 함수에서는 동일한 정점좌표와 텍스처 좌표에 `glBindTexture(GL_TEXTURE_2D, currentTextureID);`를 사용하여 프레임당 한 텍스처를 binding함 - 애니메이션 효과를 줌



OpenGL Compressed Textures

- `glCompressedTexImage2DARB`를 사용하여 압축한 텍스처를 생성할 수 있음.
 - 너비와 높이의 RGB 값을 가진 일반 텍스처보다 압축한 텍스처는 메모리 사용량을 줄이고 빨리 그릴 수 있음
 - `glCompressedTexImage2DARB(GL_TEXTURE_2D, 0, format, width, height, 0, size, imageBuffer);`

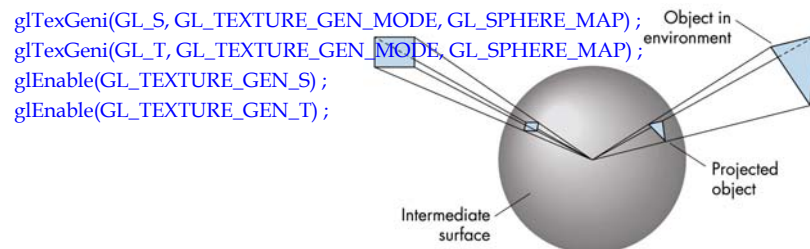
OpenGL Texture Coordinate Generation

- OpenGL에서는 텍스처 좌표를 자동적으로 생성할 수 있음
 - S, T 방향으로 텍스처 좌표 자동생성을 활성화해야 함
 - `glEnable(GL_TEXTURE_GEN_S), glEnable(GL_TEXTURE_GEN_T)`
 - `GL_TEXTURE_GEN_MODE` 모드:
 - `GL_OBJECT_LINEAR, GL_EYE_LINEAR, GL_SPHERE_MAP`
 - 평면 (plane)을 지정해야 함 - 평면으로부터의 거리에 바탕을 둔 텍스처 좌표를 생성
 - `glTexGenfv(GL_S, GL_OBJECT_PLANE, planeCoefficients)`

```
planeCoefficients = { 1, 0, 0, 0 };
glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_OBJECT_LINEAR);
glTexGenfv(GL_S, GL_OBJECT_PLANE, planeCoefficients);
glEnable(GL_TEXTURE_GEN_S);
glBegin(GL_QUADS);
glVertex3f(-3.25, -1, 0); glVertex3f(-1.25, -1, 0);
glVertex3f(-1.25, 1, 0); glVertex3f(-3.25, 1, 0);
glEnd();
```

OpenGL Sphere Mapping

- OpenGL에서는 구형 맵핑 (sphere mapping) 지원
 - 구형 맵핑 텍스처 좌표는 view 벡터가 구 표면의 법선 벡터에 반사된 reflection 벡터로 계산됨.
 - 반사벡터를 2차원 텍스처 좌표로 맵핑하는 것이 간단하고 하드웨어, 소프트웨어로도 구현이 가능.
 - 그러나 원형 이미지를 구하는 것이 어려움 (360도의 주변환경을 담은 이미지여야 함). 아주 넓은 광학 렌즈에 의한 원근 투영을 구하거나, 입방체 투영을 이용하여 근사한 값을 얻음.



OpenGL Box Mapping

- OpenGL에서는 입방체 맵핑 (box mapping) 지원
 - 입방체맵은 반사 맵핑 (reflection mapping)의 하나임
 - 그러나 입방체맵은 3차원 텍스처 좌표를 사용해야 함
 - 반사 텍스처는 환경을 둘러 싸고 있는 입방체의 6면 2차원 텍스처

```
glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_REFLECTION_MAP);
glTexGeni(GL_T, GL_TEXTURE_GEN_MODE, GL_REFLECTION_MAP);
glTexGeni(GL_R, GL_TEXTURE_GEN_MODE, GL_REFLECTION_MAP);
glEnable(GL_TEXTURE_GEN_S);
glEnable(GL_TEXTURE_GEN_T);
glEnable(GL_TEXTURE_GEN_R);
glEnable(GL_TEXTURE_CUBE_MAP);
```

