

Assignment 3: Texture Mapping and Off-screen Rendering

20131218

Park KeeHun

1. Introduction

In this assignment, I implement an environmental mapping to render mirror-like surfaces using OpenGL. I get 3 tasks in this assignment, first is implementing 2D texture mapping, second is environment mapping using static cube mapping, and last is environment mapping using dynamic cube mapping. And I loaded the BMP images, such as mob.bmp, by using the I/O functions in bmploader.cpp.

2. Method

2-1. 2D texture mapping

```
void cube(GLfloat s)          // s = size of cube
{
    glEnable(GL_TEXTURE_2D);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
    glBindTexture(GL_TEXTURE_2D, color_tex);

    glBegin(GL_QUADS);
    // Front
    glTexCoord2f(0.34f, 0.25f); glVertex3f(-s, -s, s);
    glTexCoord2f(0.66f, 0.25f); glVertex3f(s, -s, s);
    glTexCoord2f(0.66f, 0.5f); glVertex3f(s, s, s);
    glTexCoord2f(0.34f, 0.5f); glVertex3f(-s, s, s);

    // Top
    glTexCoord2f(0.34f, 0.5f); glVertex3f(-s, s, s);
```

Through GL_TEXTURE_2D, I implement 2D texture mapping. Function cube() is making texture cube, and in this function I match all points of each face from 2D to 3D. And in display(), I just call this cube() function. Virtual rotation and zooming function is realized in display() by using glPushMatrix() and glPopMatrix().

2-2. Environment mapping using static cube mapping

```
#ifdef STATIC_CUBE_MAP
// create static cubemap from synthetic data
glGenTextures(1, &cube_tex);
glBindTexture(GL_TEXTURE_CUBE_MAP, cube_tex);

// ToDo...
glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_WRAP_T, GL_REPEAT);
glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_WRAP_R, GL_REPEAT);
glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_MAG_FILTER, GL_NEAREST);

glTexImage2D(GL_TEXTURE_CUBE_MAP_POSITIVE_X, 0, GL_RGBA, 4, 4, 0, GL_RGBA, GL_UNSIGNED_BYTE, image1);
glTexImage2D(GL_TEXTURE_CUBE_MAP_NEGATIVE_X, 0, GL_RGBA, 4, 4, 0, GL_RGBA, GL_UNSIGNED_BYTE, image2);
glTexImage2D(GL_TEXTURE_CUBE_MAP_POSITIVE_Y, 0, GL_RGBA, 4, 4, 0, GL_RGBA, GL_UNSIGNED_BYTE, image3);
glTexImage2D(GL_TEXTURE_CUBE_MAP_NEGATIVE_Y, 0, GL_RGBA, 4, 4, 0, GL_RGBA, GL_UNSIGNED_BYTE, image4);
glTexImage2D(GL_TEXTURE_CUBE_MAP_POSITIVE_Z, 0, GL_RGBA, 4, 4, 0, GL_RGBA, GL_UNSIGNED_BYTE, image5);
glTexImage2D(GL_TEXTURE_CUBE_MAP_NEGATIVE_Z, 0, GL_RGBA, 4, 4, 0, GL_RGBA, GL_UNSIGNED_BYTE, image6);

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);
#endif
```

First of all, in `init()` function, I implement cube mapping at `&cube_tex` through `GL_TEXTURE_CUBE_MAP`, which has 6 faces. And by using `glTexGeni(GL_S,T,R)`, realizing reflection mapping. Above figure, each image is check board of diverse color.

```
glGenTextures(6, &box_tex[0]);

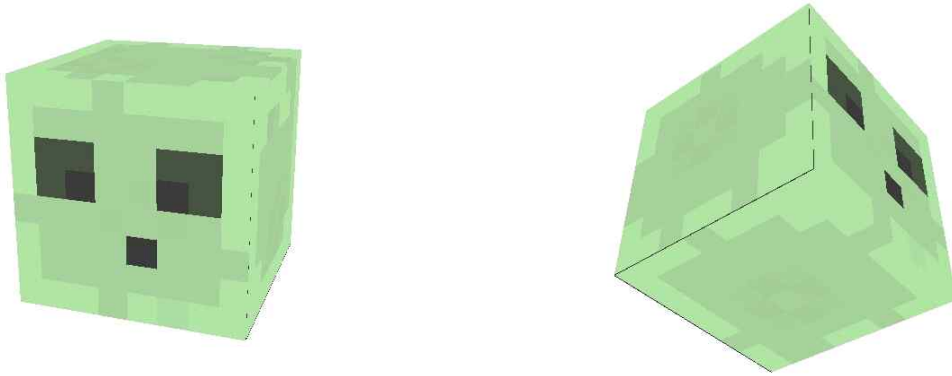
tex_binding(0, &image1[0][0][0]);
tex_binding(1, &image2[0][0][0]);
tex_binding(2, &image3[0][0][0]);
tex_binding(3, &image4[0][0][0]);
tex_binding(4, &image5[0][0][0]);
tex_binding(5, &image6[0][0][0]);

e_cube(1.2);
```

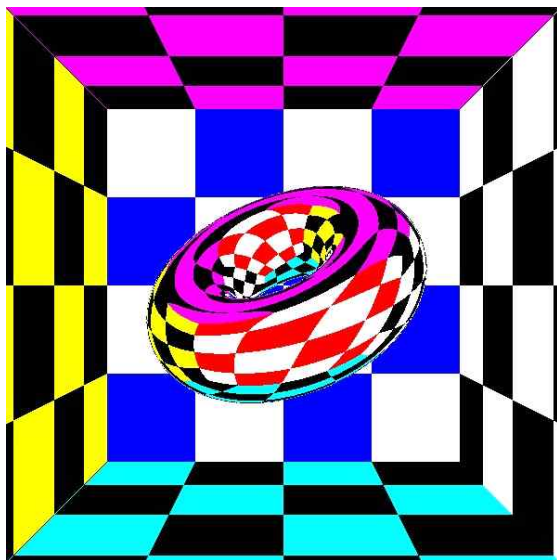
In `display()` function, I make 6 textures for static environment cube, which is `box_tex[6]` array. `tex_binding` is my function which binds each image with each texture, and I use these bound textures in `e_cube()` function to make static environment cube.

3. Result

3-1. 2D texture mapping (mob.bmp) with virtual rotation



3-2. Environment mapping using static cube mapping



Torus Reflection



Teapot Reflection

4. Conclusion

Through this assignment implementation, I can learn and understand about how to use texture mapping and off-screen rendering techniques. However, I did only 2 tasks in this assignment, and I fail to implement 3rd task, which is environment mapping using dynamic cube mapping. So, I regrets very much about that. Next time, I will invest more time and effort to achieve all of assignment, and although I fail to achieve all of this assignment, it is very nice that I become able to deal 2D texture mapping and environment mapping.