CSE 190 Discussion 3

HW2: Level of Immersion

Agenda

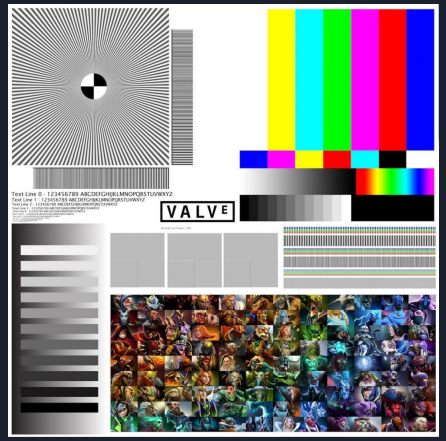
- Homework 2 Overview
- PPM Loading - Portable PixMap format
- Texture mapping

Homework 2 Overview

- Link to the assignment: http://ivl.calit2.net/wiki/index.php/Project2S18
- Due Date: May 4th 2pm
  - If you have scheduling conflicts, let us know
- Features you need to implement:
  - Map a calibration texture to a cube
  - Render a skybox in stereo
  - Change the rendering settings
  - More specifications in the assignment page.

PPM
Portable PixMap format

```
P6
# Created by IrfanView
2048 2048
255
```



PPM Loading - Portable PixMap format

1. Code for loadPPM(): http://ivl.calit2.net/wiki/images/0/09/Loadppm.txt
2. ppm file format:

   Header:

```
1 P6
2 # Created by IrfanView
3 2048 2048
4 255
```

   1. P6: byte format. P3: ASCII text
   2. # comments
   3. The image width & height
   4. Maximum value of colour components for pixels e.g. (0...255) color values here

3. One byte per color in the order of R, G, B, 0=black, 255=white.
4. Standard convention: top to bottom left to right order.

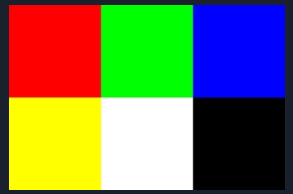
PPM Example

- This is an brief example of a color RGB image stored in PPM format.

```
1   P3
2   3 2
3   255
4   255   0   0     0 255   0     0   0 255
5   255 255   0   255 255 255     0   0   0
```

```
# "P3" means this is a RGB color image in ASCII
# "3 2" is the width and height of the image in pixels
# "255" is the maximum value for each color
# The part below is image data: RGB triplets
```

Rendered result:



PPM Loading - loadPPM() explained

1. Use fgets to read lines and skip the comments
2. Read width and height using sscanf of format “%s %s” and stoi
3. Skip the maximum value of color
4. Use fread to read all the remaining bytes of size N = width *height*3
5. Returns unsigned char* rawData, which is a unsigned byte array of size N. (Do not forget to delete this after passing it to GL)

```
// Read magic number:
retval_fgets = fgets(buf[0], BUFSIZE, fp);

// Read width and height:
do
{
        retval_fgets=fgets(buf[0], BUFSIZE, fp);
} while (buf[0][0] == '#');
retval_sscanf=sscanf(buf[0], "%s %s", buf[1], buf[2]);
width  = atoi(buf[1]);
height = atoi(buf[2]);

// Read maxval:
do
{
retval_fgets=fgets(buf[0], BUFSIZE, fp);
} while (buf[0][0] == '#');
```

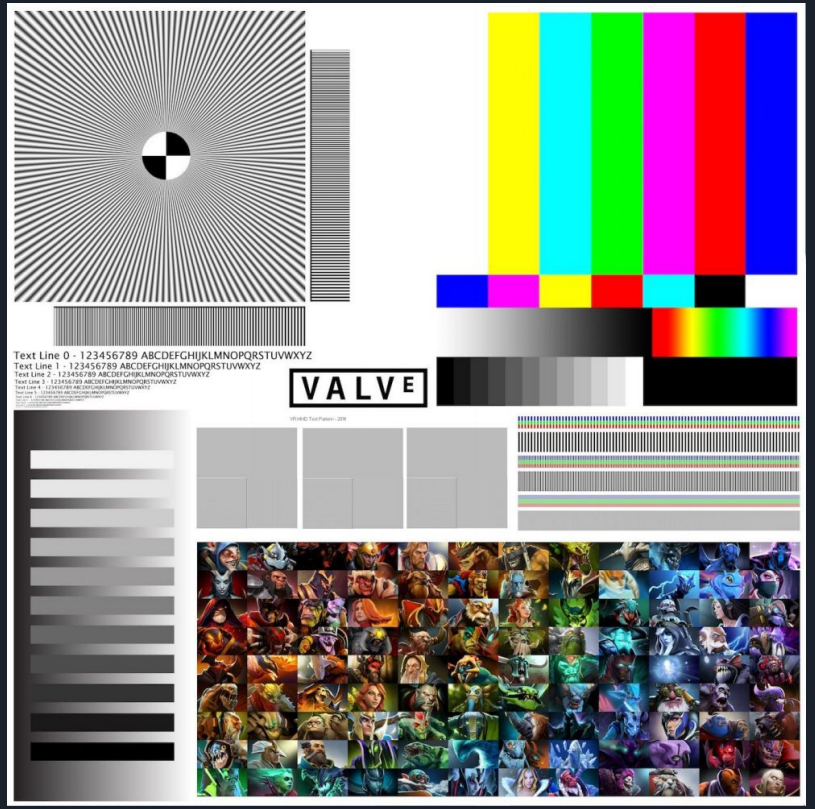
```
// Read image data:
rawData = new unsigned char[width * height * 3];
read = fread(rawData, width * height * 3, 1, fp);
fclose(fp);
if (read != 1)
{
        std::cerr << "error parsing ppm file, incomplete data" << std::endl;
        delete[] rawData;
        width = 0;
        height = 0;

        return 0;
}

return rawData;
```

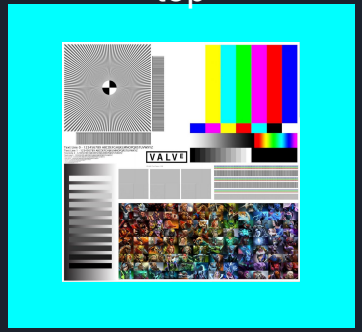
vr_test_pattern.ppm

Rendered image result:



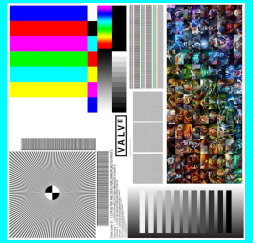
Texture Rendering Overview

top



Overview

bottom



front

left

back

right

Texture mapping

- An example is posted on the assignment page.
  - http://ivl.calit2.net/wiki/images/8/88/Hellovr_opengl_main.cpp
- Feel free to look at other tutorials as well
  - https://learnopengl.com/Getting-started/Textures
  - http://www.opengl-tutorial.org/beginners-tutorials/tutorial-5-a-textured-cube/

HelloVR example

```
SetupTexturemaps()
...
glGenTextures(1, &m_iTexture );
glBindTexture( GL_TEXTURE_2D, m_iTexture );
glTexImage2D( GL_TEXTURE_2D, 0, GL_RGBA, nImageWidth, nImageHeight,
      0, GL_RGBA, GL_UNSIGNED_BYTE, &imageRGBA[0] );
glGenerateMipmap(GL_TEXTURE_2D);
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR );
GLfloat fLargest;
glGetFloatv(GL_MAX_TEXTURE_MAX_ANISOTROPY_EXT, &fLargest);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAX_ANISOTROPY_EXT, fLargest);

glBindTexture( GL_TEXTURE_2D, 0 );
```

Texture Mapping Setup

```
//Creates the texture.
glGenTextures(1, &m_iTexture );
glBindTexture( GL_TEXTURE_2D, m_iTexture );

//Pass in the data from the ppm you loaded
glTexImage2D( GL_TEXTURE_2D, 0, GL_RGBA, nImageWidth,
nImageHeight,
    0, GL_RGBA, GL_UNSIGNED_BYTE, &imageRGBA[0] );
```

Texture Parameters

```
//Mipmapping
glGenerateMipmap(GL_TEXTURE_2D);

//Texture wrapping settings
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE );

//Texture filtering settings - play around with these settings to see the effect
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR );
GLfloat fLargest;
glGetFloatv(GL_MAX_TEXTURE_MAX_ANISOTROPY_EXT, &fLargest);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAX_ANISOTROPY_EXT, fLargest);
```

Using the texture

```
//In the Shader

in vec2 TexCoord;
uniform sampler2D calibrationTex;
...
color = texture(calibrationTex, TexCoord);
```

Remember to bind the texture before you draw!

Multiple Textures

- But wait! I didn’t need to set the uniform in the shader and it worked.

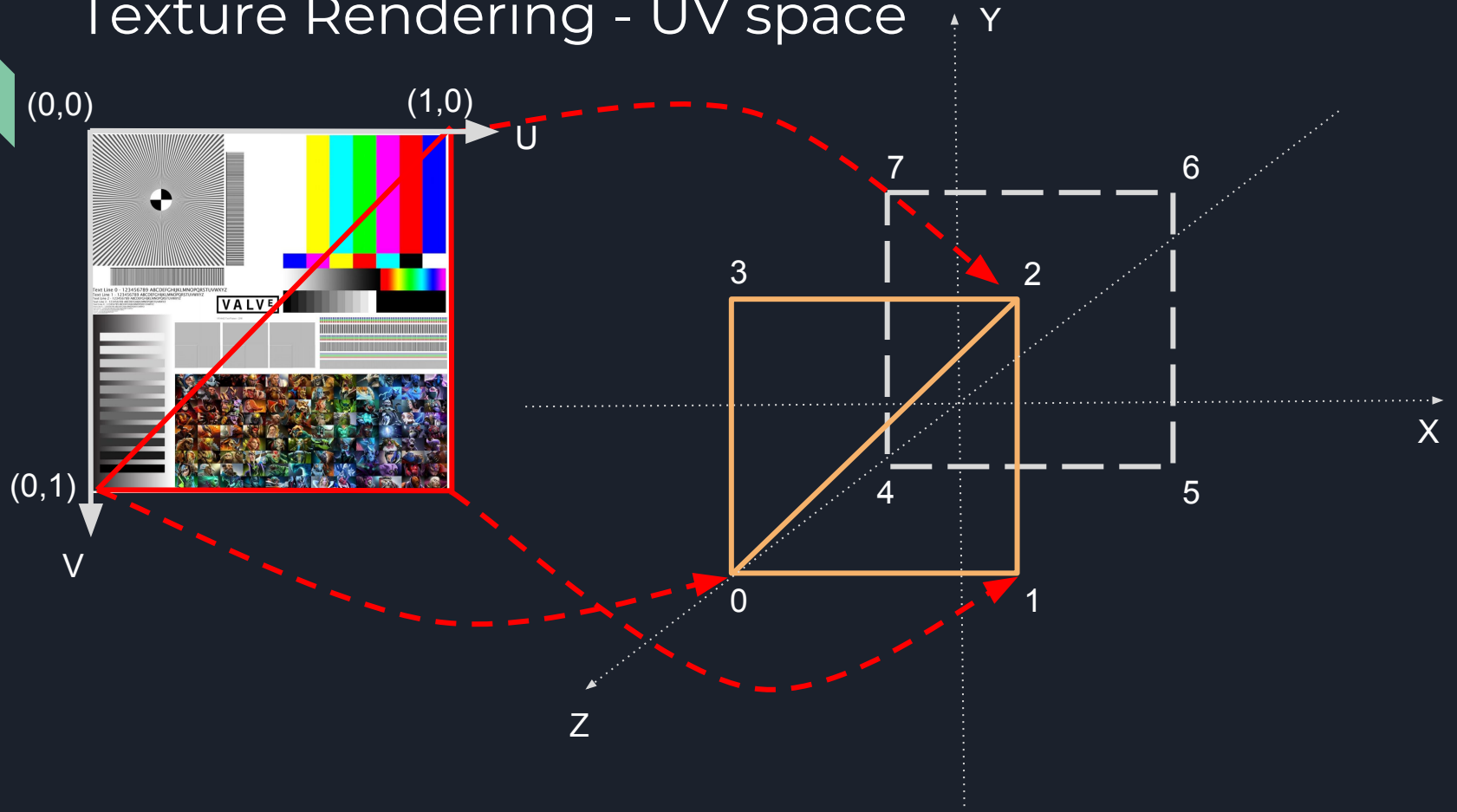
Multiple Textures

- But wait! I didn't need to set the uniform in the shader and it worked.
- OpenGL will default to a texture location of 0
- You can use `glActiveTexture(GLTEXTURE0 + i)` to set which texture location you want to work with.
- Follow this with `glBindTexture(GL_TEXTURE_2D, texture)` to bind a texture to that location.
- Then you can set the uniform to specify which location to work with.

Creating the cube

- You will probably need to make your own cube vertex data
- Each corner will need need three different vertices, each with different texture coordinates

Texture Rendering - UV space



QUESTIONS?