

Texturing in GLSL



Oregon State
University

Mike Bailey

mjb@cs.oregonstate.edu



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/)



Oregon State
University
Computer Graphics

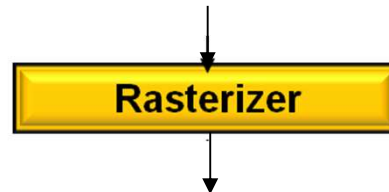
2D Texturing

2

Vertex shader:

```
#version 330 compatibility
out vec2 vST;

void
main( )
{
    vST = gl_MultiTexCoord0.st;
    gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```



Fragment shader:

```
#version 330 compatibility
in vec2 vST;
uniform sampler2D uTexUnit;

void
main( )
{
    vec3 newcolor = texture(uTexUnit, vST ).rgb;
    gl_FragColor = vec4( newcolor, 1. );
}
```

On Macs, this function should be called **texture2D**

Both-named functions return a vec4: rgba

Fragment shader:

```
#version 330 compatibility
in vec2 vST;
uniform sampler2D uTexUnit;

void
main( )
{
    vec3 newcolor = texture( uTexUnit, vST ).rgb;
    gl_FragColor = vec4( newcolor, 1. );
}
```

```
GLuint TexName;                // a global
...
glGenTextures( 1, &TexName );   // in InitGraphics( )
int nums, numt;
unsigned char *texture = BmpToTexture( "filename.bmp", &nums, &numt );
glBindTexture( GL_TEXTURE_2D, TexName );
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT );
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT );
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR );
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR );
glTexImage2D( GL_TEXTURE_2D, 0, 3, nums, numt, 0, GL_RGB, GL_UNSIGNED_BYTE, texture );
...
Pattern.Use( );                // in Display( ):
glActiveTexture( GL_TEXTURE6 ); // use texture unit 6
glBindTexture( GL_TEXTURE_2D, TexName );
Pattern.SetUniformVariable( "uTexUnit", 6 );
```

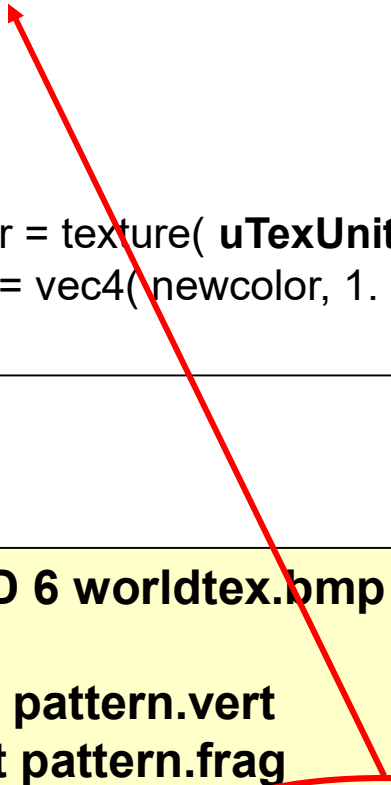
2D Texturing in glman

Fragment shader:

```
#version 330 compatibility
in vec2 vST;
uniform sampler2D uTexUnit;

void
main( )
{
    vec3 newcolor = texture( uTexUnit, vST ).rgb;
    gl_FragColor = vec4( newcolor, 1. );
}
```

```
Texture2D 6 worldtex.bmp
...
Vertex pattern.vert
Fragment pattern.frag
Program World uTexUnit 6
```



The OsuSphere Has Sensible s,t Coordinates Assigned



The GLUT Teapot Doesn't



What if You Want to Use Two Textures in One Shader?

7

C++ Program:

```
// In Display():
Pattern.Use();
glActiveTexture( GL_TEXTURE5 );
glBindTexture( GL_TEXTURE_2D, TexName0 );

glActiveTexture( GL_TEXTURE6 );
glBindTexture( GL_TEXTURE_2D, TexName1 );

Pattern.SetUniformVariable( "uTexUnit0", 5 );
Pattern.SetUniformVariable( "uTexUnit1", 6 );

<< draw something >>
Pattern.UnUse();
```

Fragment shader:

```
#version 330 compatibility
in vec2 vST;
uniform sampler2D uTexUnit0;
uniform sampler2D uTexUnit1;

void
main( )
{
    vec3 newColor0 = texture( uTexUnit0, vST );
    vec3 newColor1 = texture( uTexUnit1, vST );
    gl_FragColor = ...
}
```

Why Might You Want to Use Two Textures in One Shader?

Once the RGBs have been read from a texture, they are just numbers. You can do any arithmetic you want with the texture RGBs, other colors, lighting, etc. Here is an example of blending two textures at once:



Using Two Textures in One Shader, I

9

```
#version 330 compatibility

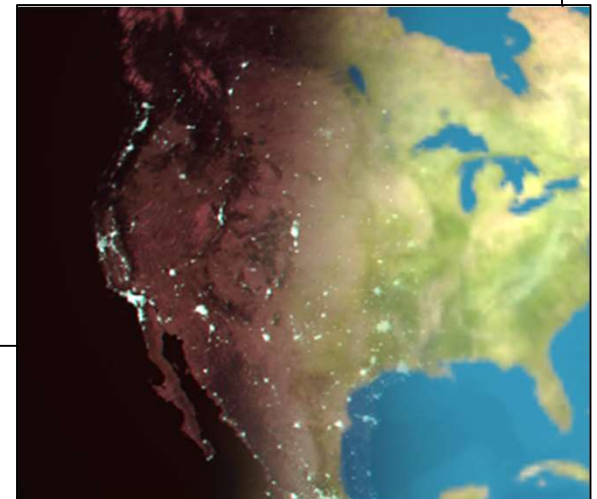
uniform float uBlend;
uniform float uSunLng;
uniform sampler2D uTexUnitDay, uTexUnitNight;

in float vLightIntensity;
in vec2 vST;

const float OFFSETS = 0.0;           // how much the images are off horizontally

void
main( )
{
    float sunLng = uSunLng; // -180. to +180.
    float sunLngp = sunLng - 180.;
    if( sunLngp < -180. ) sunLngp += 360.; // -180. to +180.

    vec2 nightST = vST;
    vec2 dayST = vST;
    dayST.s += OFFSETS;
    if( dayST.s < 0. ) dayST.s += 1.;
```



Using Two Textures in One Shader, II

10

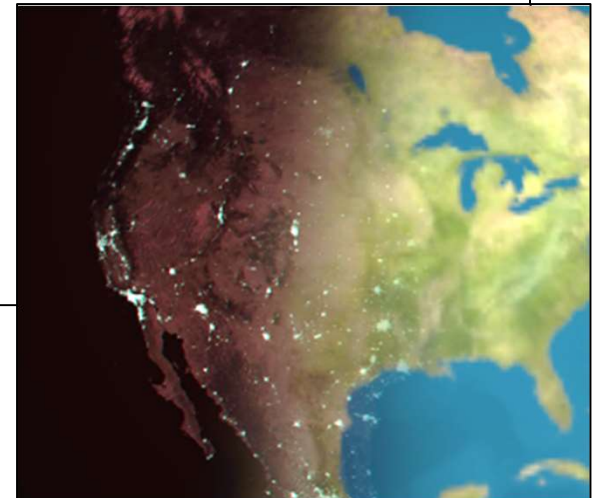
```
float earthing = 360. * ( dayST.s - 0.5 );           // -180. to +180.

float delta0 = earthing - sunIng;
if( delta0 < -180. ) delta0 += 360.;
if( delta0 > 180. ) delta0 -= 360.;

float delta1 = earthing - sunIngp;
if( delta1 < -180. ) delta1 += 360.;
if( delta1 > 180. ) delta1 -= 360.;
float delta = delta0;
if( abs(delta1) < abs(delta0) )
    delta = -delta1;

vec3 rgbday = texture( uTexUnitDay, dayST).rgb;
vec3 rgnight = texture( uTexUnitNight, nightST).rgb;

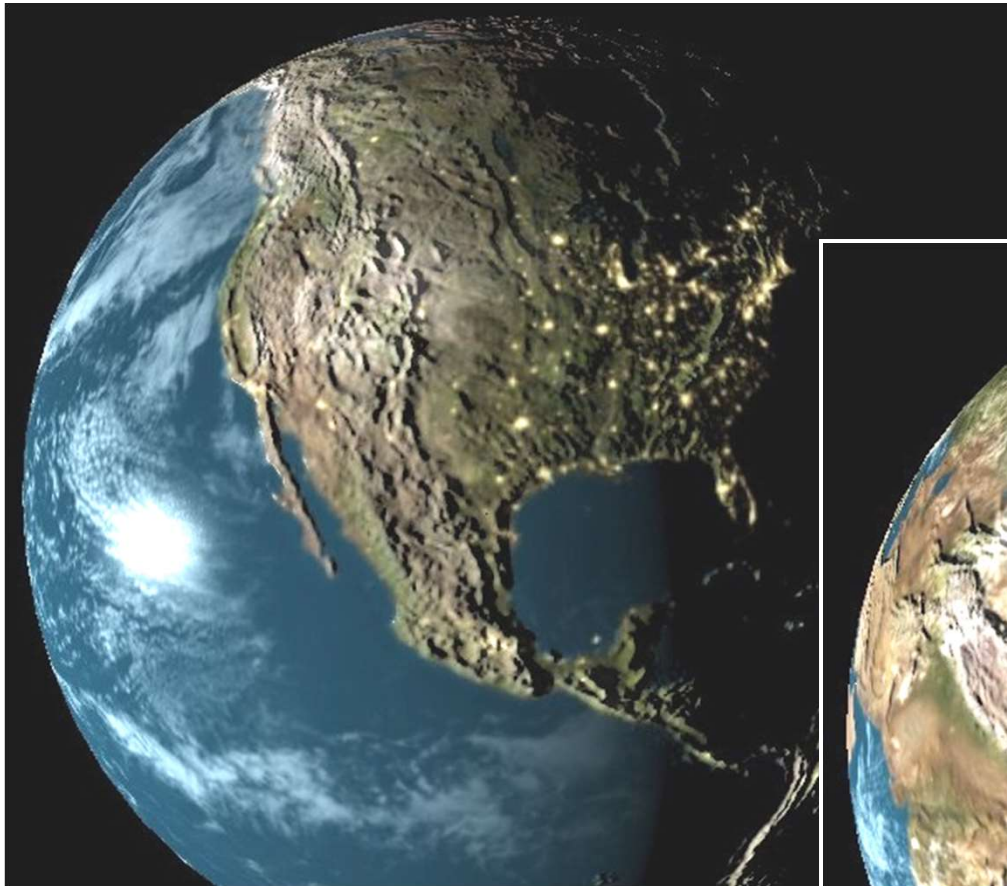
float t = smoothstep( -uBlend, uBlend, delta );
vec3 newcolor = mix( rgbday, rgnight, t );
gl_FragColor = vec4( vLightIntensity*newcolor, 1. );
}
```



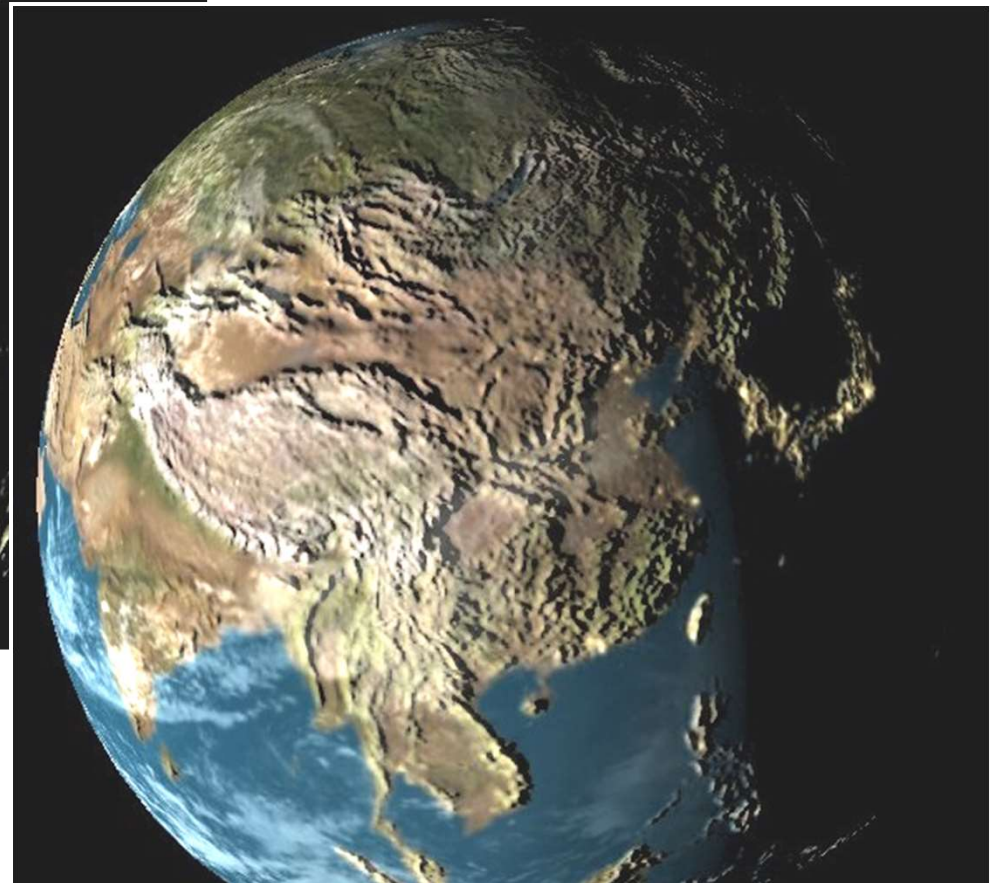
Oregon State
University

Computer Graphics

Shaders Can Combine More than Two Textures



Just assign each texture a different texture unit



- Textures used here:
- Day
 - Night
 - Heights (bump-mapping)
 - Clouds
 - Specular highlights

