Texture Mapping in Practice

Dinesh K. Pai

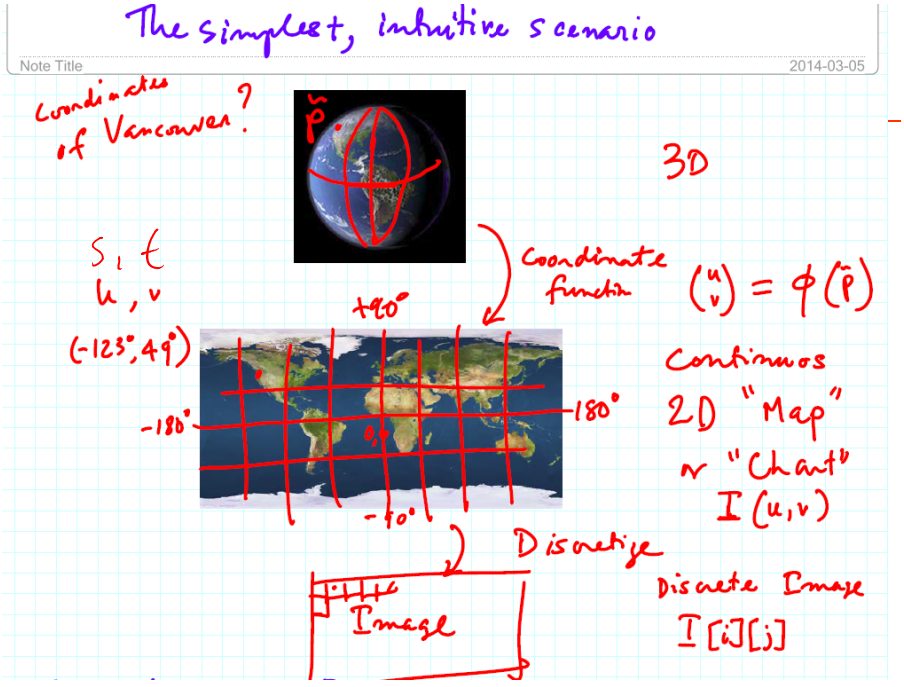
Textbook Appendix A4, Chapter 15

Some slides courtesy of M. Kim, KAIST

1

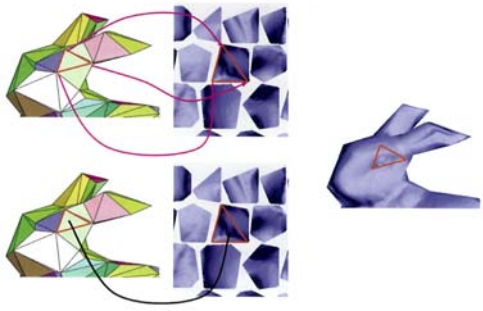
The simplest, intuitive scenario

Note Title 2014-03-05



Texture mapping

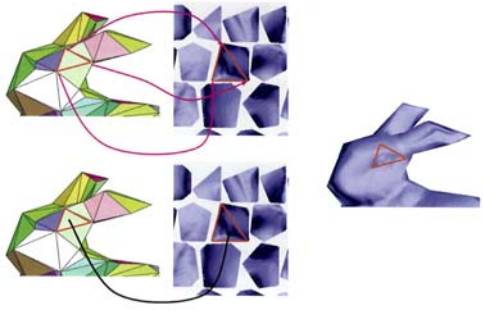
- In basic texturing, we simply ‘glue’ part of an image onto a triangle by specifying texture coordinates at the three vertices.



3

Texture mapping

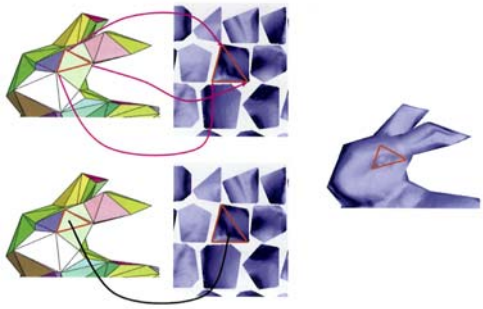
- Bunch of OpenGL functions to load a texture and set various parameters (lin/const, mipmap, wrapping rules).
- A uniform variable is used to point to the desired texture unit



4

Texture mapping

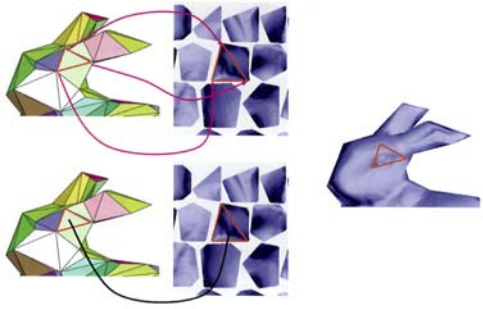
- Varying variables are used to store texture coordinates.
- In this simplest incarnation, we just fetch r,g,b values from the texture and send them directly to the frame buffer.



5

Texture mapping

- Alternatively, the texture data could be interpreted as, say, the diffuse material color of the surface point, which would then be followed by the diffuse material computation described earlier.



6

Steps for Texture Mapping

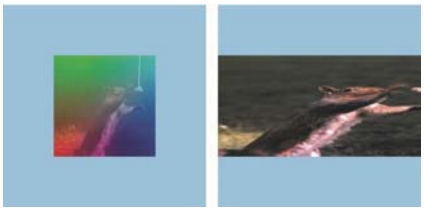
1. Create a *texture object* and load texels into it
2. Include *texture coordinates* with your vertices
3. Associate a *texture sampler* with each texture map used in shader
4. Retrieve texel values

(Reference: Red Book)

7

Second basic example: Texture mapping a square

- See Appendix A.4



8

Texture mapping

- initGLState()

```
…
glActiveTexture(GL_TEXTURE0);
glGenTextures(1, &h_texture);
glBindTexture(GL_TEXTURE_2D, h_texture);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
int twidth, theight;
packed_pixel_t * pixdata = ppmread("reachup.ppm", &twidth, &theight);
assert(pixdata);
glTexImage2D(GL_TEXTURE_2D, 0, GL_SRGB, twidth, theight, 0, GL_RGB,
GL_UNSIGNED_BYTE, pixdata);
free(pixdata);
...
```

texture unit to use

dimension & type of texture

Lots of options open source eg. SOIL

9

- See Nehe texture tutor

10

Texture mapping

- initShaders()

```
h_texUnit0 = safe_glGetUniformLocation(h_program, "texUnit0");
h_aTexCoord = safe_glGetAttribLocation(h_program, "aTexCoord");
```

- display()

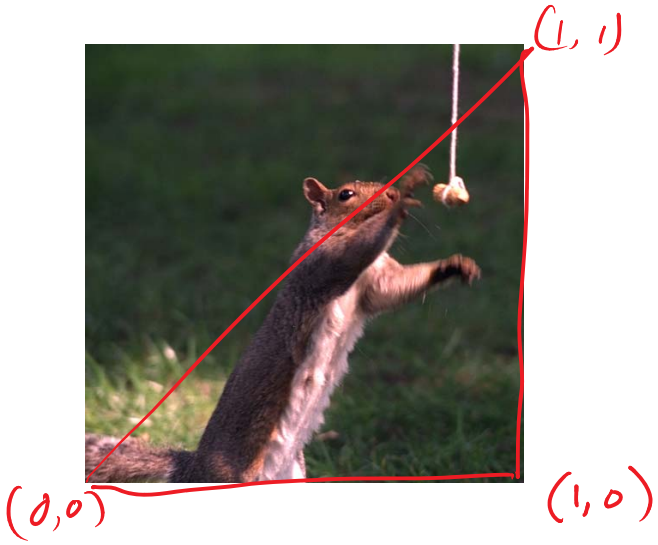
```
safe_glUniform1i(h_texUnit0, 0);
```

- Texture location (0,0) → lower left, (1,1) → upper right

```
GLfloat sqTex[12] =
{
  0, 0,
  1, 1,
  1, 0,

  0, 0,
  0, 1,
  1, 1
};
```

11



12

Texture mapping

- Vertex shader, just “pass through”

```
#version 330
uniform float uVertexScale;
uniform mat4 uProjMatrix;
uniform mat4 uModelViewMatrix;
in vec2 aVertex;
in vec2 aTexCoord;
in vec3 aColor;
out vec3 vColor;
out vec2 vTexCoord;

void main()
{
  gl_Position = vec4(uProjMatrix * uModelViewMatrix * aVertex);
  vColor = aColor;
  vTexCoord = aTexCoord;
}
```

13

Texture mapping

- Fragment shader changes

```
#version 330

uniform sampler2D texUnit0;
in vec2 vTexCoord;
out vec4 fragColor;

void main() {
  vec4 texColor0 = texture2D(texUnit0, vTexCoord);
  fragColor = texColor0;
}
```

Every Texture Object has a default Sampler Object

floating pt values

14