# Texture Coordinates

Dinesh K. Pai

Textbook Chapter 15

Some slides courtesy of M. Kim, KAIST

1

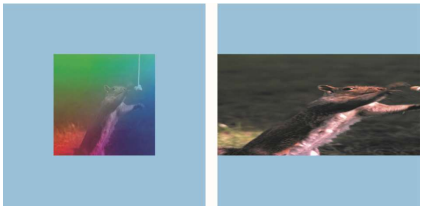
## Today

- Texturing in Three.js
- Bump mapping
- Quiz 2 discussion

2

# Second basic example: Texture mapping a square

- See Appendix A.4



3

# Texture mapping in OpenGL (WebGL is very similar)

- 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);
...
```

4

# Texture mapping

- initShaders()

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

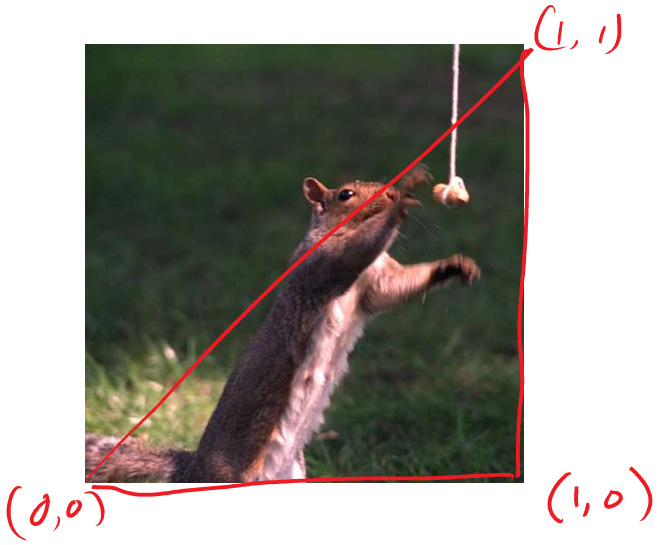
- display()

```
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
};
```

5




6

# 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;
}
```

7

# 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;
}
```

8

## Texture Mapping in Three.js

- Demo

9

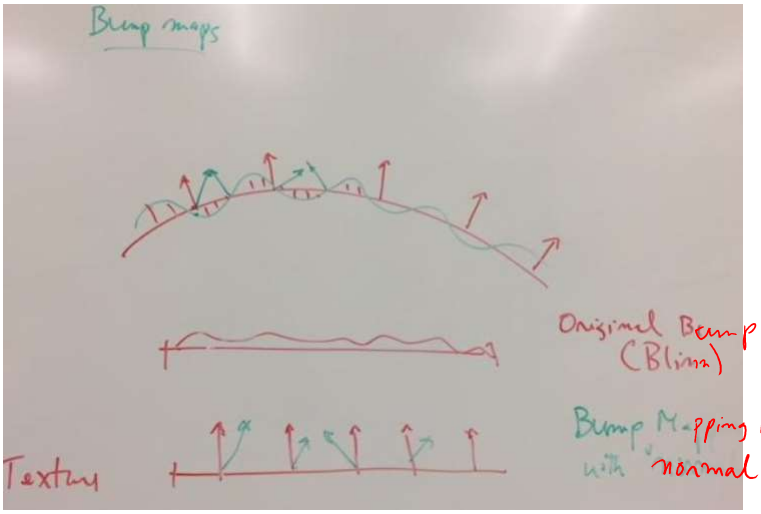
## Texture mapping in Three.js

- Interactive demo, Earth texture mapping

```
var earthColorTexture =
        new THREE.ImageUtils.loadTexture('images/earthmap1k.jpg');

var earthBumpTexture =
        new THREE.ImageUtils.loadTexture('images/earthbump1k.jpg');

var earthMaterial = new THREE.MeshPhongMaterial(
    {
        map: earthColorTexture,
        bumpMap: earthBumpTexture
    } );

var gem = new THREE.Mesh(gemGeometry, earthMaterial);
```

10

## Bump mapping (whiteboard)



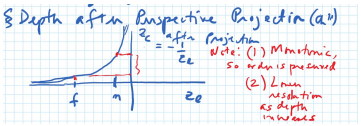

11

## Generating your own Texture Coordinates

- Can be done in Maya, 3DS Max, Blender and other 3D modeling software. This is how it's done in production applications
- Legacy OpenGL had a function (glTexGen) to do this, removed from current versions
- In production, coordinates are designed with model (or "painted" on 3D model)
- Useful texture coordinates can often be computed in shaders (e.g., projection, environment maps)

# Quiz 2 solutions

- Q1. Key: 3,9,5,15,10,17 (1.5 marks each) ;
  F, T, F (1 mark each)
- Q2
  a. L11 p. 4
  b. matrices are written with rows separated by semicolons, as in Matlab $Rot(Z) = \begin{bmatrix} \cos\theta & -\sin\theta & \\ \sin\theta & \cos\theta & \\ & & 1 \end{bmatrix}$ $Rot(Y) = \begin{bmatrix} \cos\theta & & \sin\theta \\ & 1 & \\ -\sin\theta & & \cos\theta \end{bmatrix}$
  c. Most had difficulty with this question. Meant to test your understanding of L15 (ch.11)
  (i) no. divide by zero
  (ii) no. low resolution of far.
  Generous partial credits since question wasn't clear.

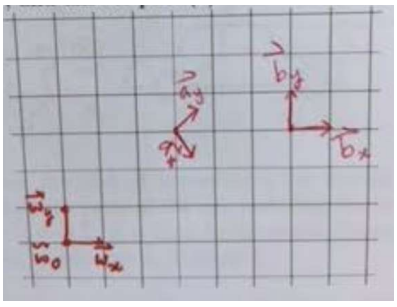



13

- Q3 People did surprisingly poorly (avg. 40%), even though it's straight from the lectures
  a. See L14, p.4
  b. See L13, p.3
- Q4 Generally did well

(b) $\begin{bmatrix} 1 & & 5 \\ & 1 & 3 \\ & & 1 \end{bmatrix}$

(c) $\begin{pmatrix} 0 \\ -\sqrt{3}/2 \\ 1 \end{pmatrix}$

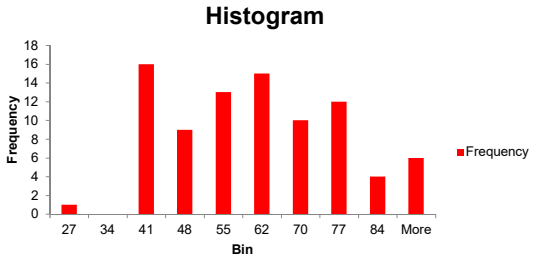


14

- Q5. This is an instance of transformations about auxiliary frames (L11, pages 2 and 3)
  - (a) $p(s) = CMC^{-1} p\_c$
  - (b) $p(s) = M C T R T^{-1} p\_c$

15

# Quiz 2 raw percentage




Mean 57.57 %

16

# Summary

- Performance not as good as in previous quiz
- Q1 and Q3 indicates that straightforward questions were missed by many
- Since some parts may have been a little ambiguous, will add 5 points (~11%) for everyone.

17