Texture Mapping in OpenGL

Excerpted from

An Interactive Introduction to OpenGL Programming

Dave Shreiner

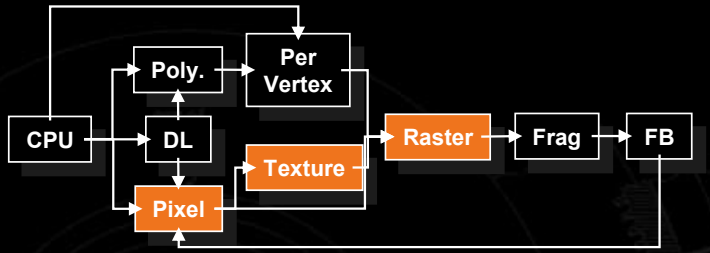
Ed Angel

Vicki Shreiner

Texture Mapping

Ed Angel

Texture Mapping

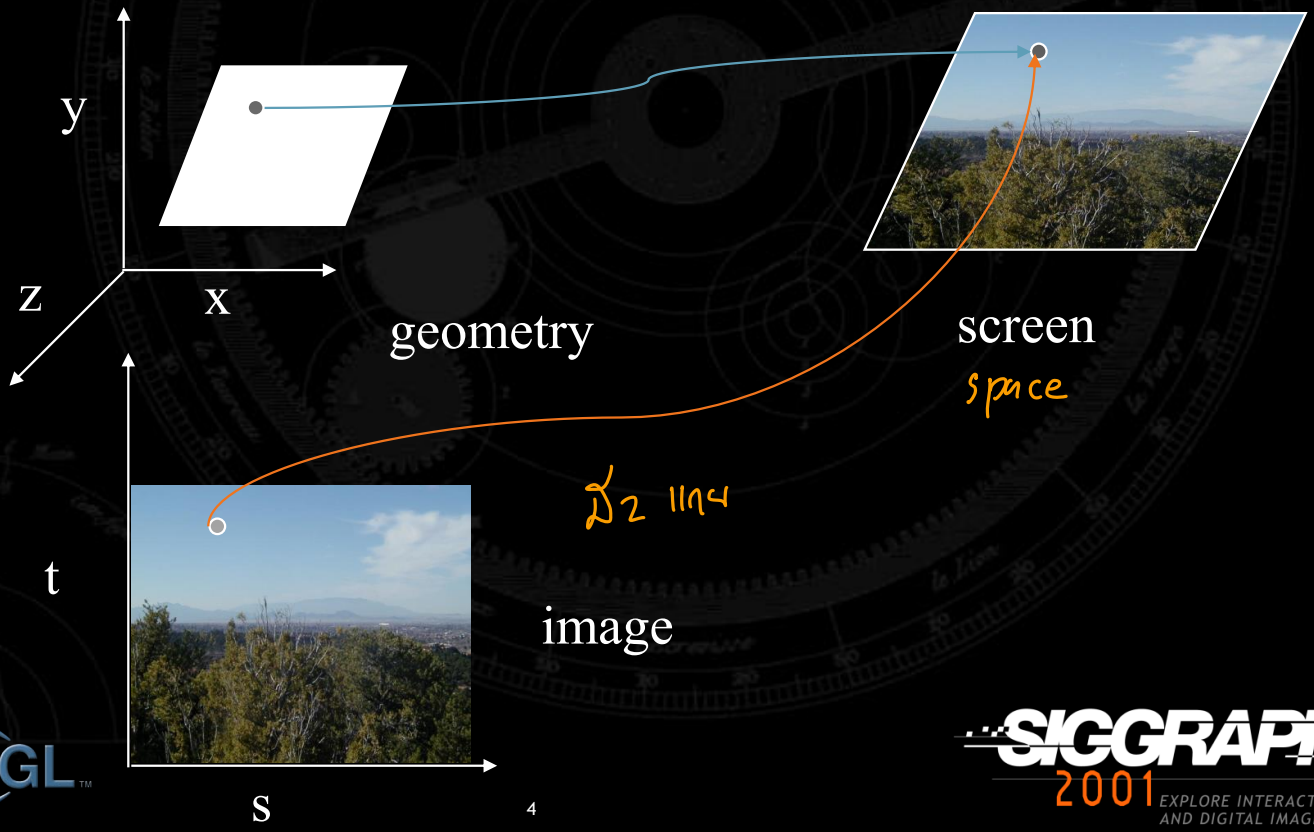


Apply a 1D, 2D, or 3D image to geometric primitives

Uses of Texturing

- simulating materials การทำของพื้นผิว materials
- reducing geometric complexity
- image warping
- reflections วัตถุสะท้อน

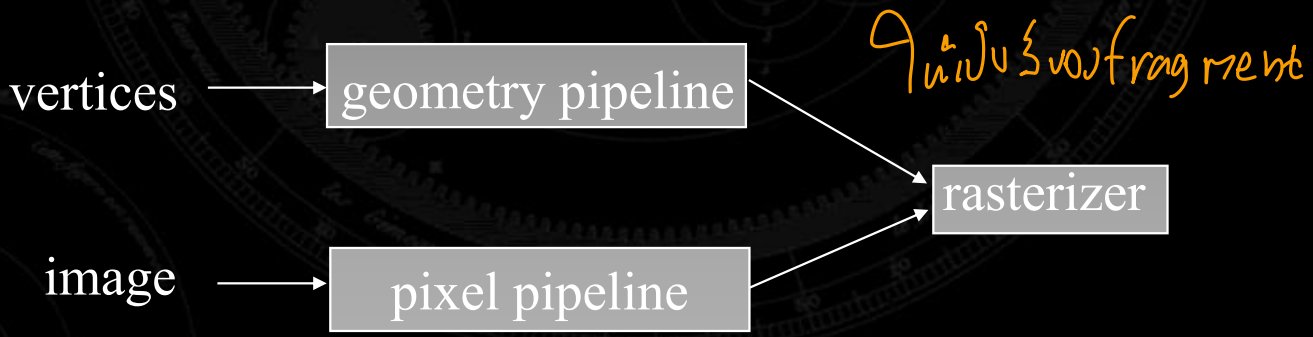
Texture Mapping



Texture Mapping and the OpenGL Pipeline

Images and geometry flow through separate pipelines that join at the rasterizer

- "complex" textures do not affect geometric complexity



Texture Example

The texture (below) is a 256 x 256 image that has been mapped to a rectangular polygon which is viewed in perspective

Applying Textures

Three steps

1. specify texture
   - read or generate image
   - assign to texture
   - enable texturing
2. assign texture coordinates to vertices
3. specify texture parameters
   - wrapping, filtering

Texture Objects ตัวแทนของ texture

Like display lists for texture images

- one image per texture object
- may be shared by several graphics contexts

Generate texture names

```
texIds = glGenTextures( n )
```

Texture Objects (cont.)

Create texture objects with texture data and state

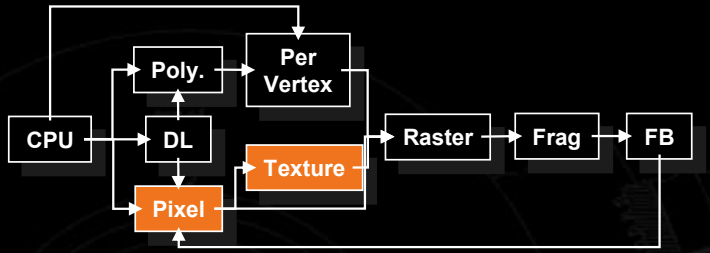
```
glBindTexture( target, id )
```

ใช้ texture ในรูปแบบใหน

Bind textures before using

```
glBindTexture( target, id )
```

Specify Texture Image



Define a texture image from an array of texels in CPU memory

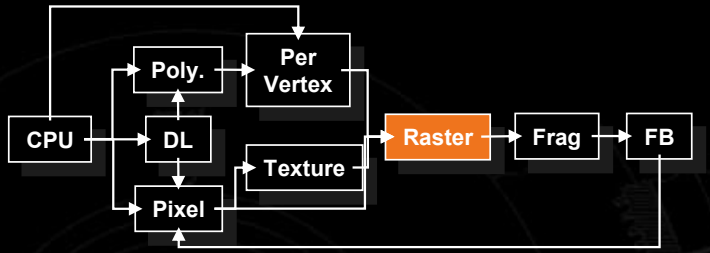
```
glTexImage2D( target, level, components,
    w, h, border, format, type, texels );
```

- dimensions of image must be powers of 2

Texel colors are processed by pixel pipeline

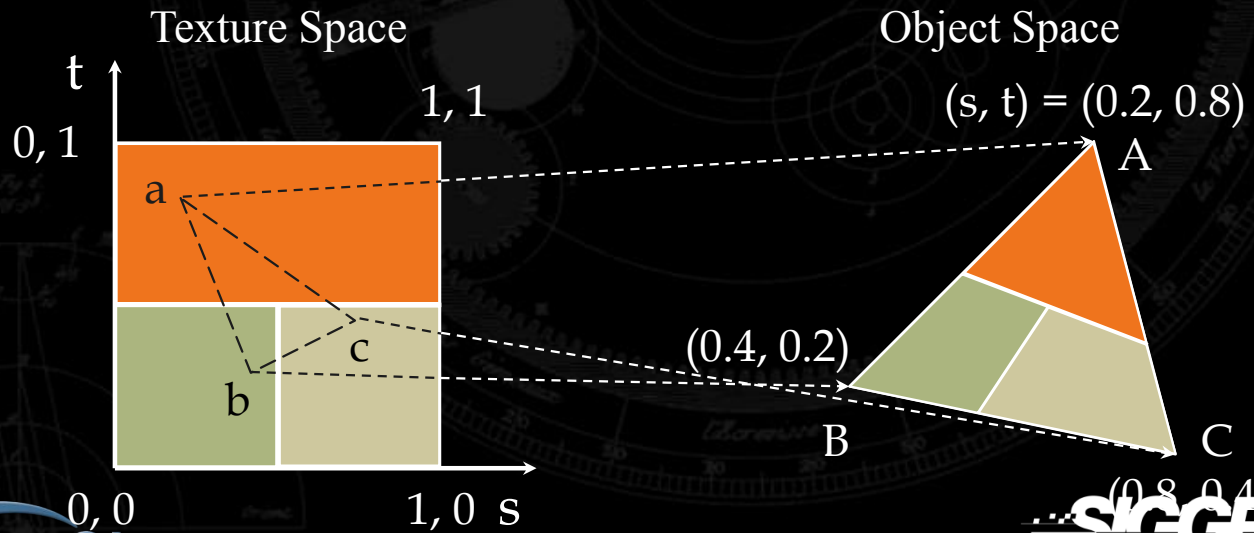
- pixel scales, biases and lookups can be done

Mapping a Texture



Based on parametric texture coordinates

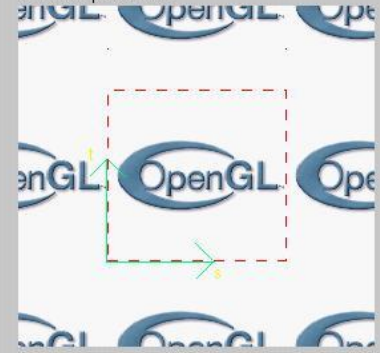
`glTexCoord*()` specified at each vertex



Tutorial: Texture

Screen-space view

Texture-space view



Command manipulation window

```
GLfloat border_color[ ] = {  1.00 ,  0.00 ,  0.00 ,  1.00 };
GLfloat env_color[ ] = {     0.00 ,  1.00 ,  0.00 ,  1.00 };

glTexParameterfv(GL_TEXTURE_2D, GL_TEXTURE_BORDER_COLOR, border_color);
glTexEnvfv(GL_TEXTURE_ENV, GL_TEXTURE_ENV_COLOR, env_color);

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,  GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,  GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S,  GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T,  GL_REPEAT);
glTexEnvi(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE,  GL_MODULATE);

glEnable(GL_TEXTURE_2D);
gluBuild2DMipmaps(GL_TEXTURE_2D, 3, w, h, GL_RGB, GL_UNSIGNED_BYTE, image);
glColor4f(   0.60  , 0.60  , 0.60  , 1.00  );
glBegin(GL_POLYGON);
glTexCoord2f( 0.0  , 0.0  ); glVertex3f( -1.0 , -1.0 , 0.0  );
glTexCoord2f( 1.0  , 0.0  ); glVertex3f( 1.0  , -1.0 , 0.0  );
glTexCoord2f( 1.0  , 1.0  ); glVertex3f( 1.0  , 1.0  , 0.0  );
glTexCoord2f( 0.0  , 1.0  ); glVertex3f( -1.0 , 1.0  , 0.0  );
glEnd();
```

Click on the arguments and move the mouse to modify values.

Texture Application Methods

Filter Modes

- minification or magnification
- special mipmap minification filters

Wrap Modes

- clamping or repeating

Texture Functions

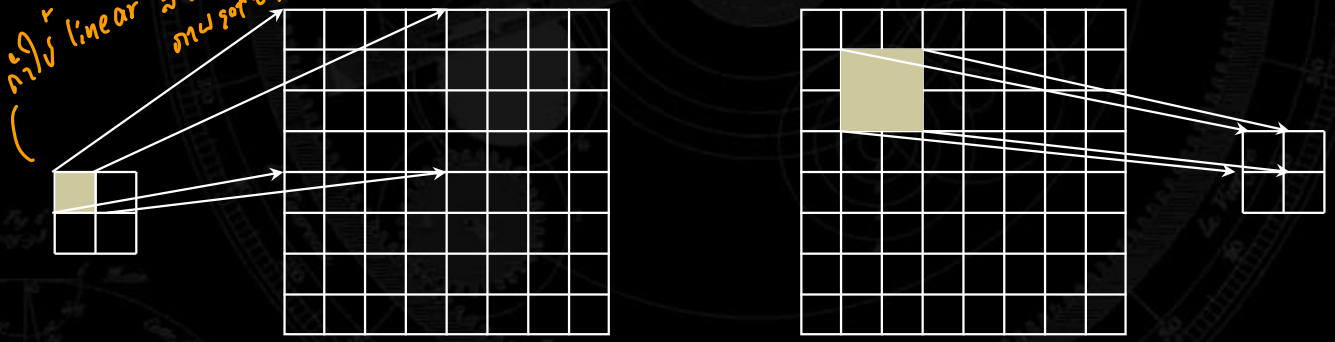
- how to mix primitive's color with texture's color
  - blend, modulate or replace texels

Filter Modes

```
GL_TEXTURE_MIN_FILTER
GL_TEXTURE_MAG_FILTER
GL_NEAREST, GL_LINEAR,
GL_NEAREST_MIPMAP_NEAREST
GL_NEAREST_MIPMAP_LINEAR
GL_LINEAR_MIPMAP_NEAREST
GL_LINEAR_MIPMAP_LINEAR
```

Example:

```
glTexParameteri( target, type, mode );
```



Texture Polygon

Magnification

Texture Polygon

Minification

Mipmapped Textures

Mipmap allows for prefiltered texture maps of decreasing resolutions

Lessens interpolation errors for smaller textured objects

Declare mipmap level during texture definition

```
glTexImage*D( GL_TEXTURE_*D, level, … )
```

GLU mipmap builder routines

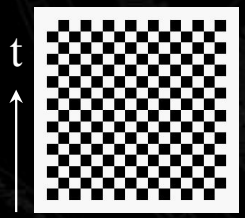
```
gluBuild*DMipmaps( … )
```

OpenGL 1.2 introduces advanced LOD controls

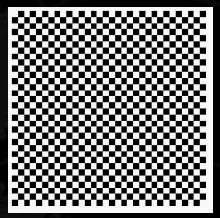
Wrapping Mode

Example:

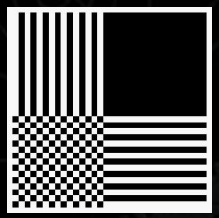
```
glTexParameteri( GL_TEXTURE_2D,
     GL_TEXTURE_WRAP_S, GL_CLAMP )
glTexParameteri( GL_TEXTURE_2D,
     GL_TEXTURE_WRAP_T, GL_REPEAT )
```



texture



GL_REPEAT wrapping



GL_CLAMP wrapping

Texture Functions

Controls how texture is applied

`glTexEnv{fi}[v]( GL_TEXTURE_ENV, prop, param )`

GL_TEXTURE_ENV_MODE modes

- GL_MODULATE
- GL_BLEND
- GL_REPLACE

Set blend color with *GL_TEXTURE_ENV_COLOR*

Perspective Correction Hint

Texture coordinate and color interpolation

- either linearly in screen space
- or using depth/perspective values (slower)

Noticeable for polygons “on edge”

```
glHint( GL_PERSPECTIVE_CORRECTION_HINT, hint )
```

where *hint* is one of

- *GL_DONT_CARE*
- *GL_NICEST*
- *GL_FASTEST*

On-Line Resources

- `http://www.opengl.org`
  - start here; up to date specification and lots of sample code
- `news:comp.graphics.api.opengl`
- `http://www.sgi.com/software/opengl`
- `http://www.mesa3d.org/`
  - Brian Paul's Mesa 3D
- `http://www.cs.utah.edu/~narobins/opengl.html`
  - very special thanks to Nate Robins for the OpenGL Tutors
  - source code for tutors available here!

Books

OpenGL Programming Guide, 3rd Edition

OpenGL Reference Manual, 3rd Edition

OpenGL Programming for the X Window System

- includes many GLUT examples

Interactive Computer Graphics: A top-down approach with OpenGL, 2nd Edition

Authors

Dave Shreiner	shreiner@sgi.com
Ed Angel	angel@cs.unm.edu
Vicki Shreiner	vshreiner@sgi.com