Applying Textures I

(Based on E Angel's slides)

Three steps

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

Applying Textures II

- specify textures in texture objects
- set texture filter
- set texture function
- set texture wrap mode
- set optional perspective correction hint
- bind texture object
- enable texturing
- supply texture coordinates for vertex
  - coordinates can also be generated

Texture Objects

Like display lists for texture images

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

Generate texture names

```
glGenTextures( n, *texIds );
```

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

Converting A Texture Image

If dimensions of image are not power of 2

```
gluScaleImage( format, w_in, h_in,
    type_in, *data_in, w_out, h_out,
        type_out, *data_out );
```

- ***_in** *is for source image*
- ***_out** *is for destination image*

Image interpolated and filtered during scaling

Specifying a Texture: Other Methods

Use frame buffer as source of texture image

- **uses current buffer as source image**

```
glCopyTexImage2D(...)
glCopyTexImage1D(...)
```

Modify part of a defined texture

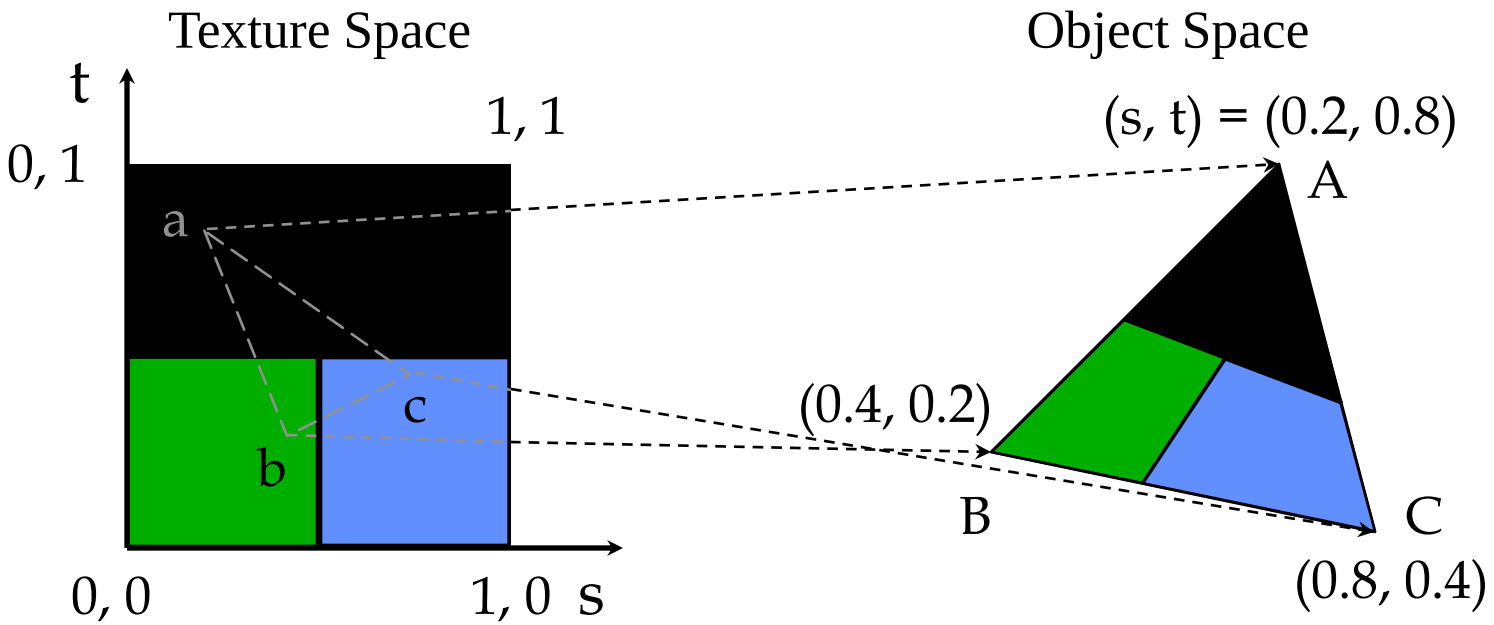
```
glTexSubImage2D(...)
glTexSubImage1D(...)
```

Do both with *`glCopyTexSubImage2D(...)`*, etc.

Mapping a Texture

Based on parametric texture coordinates

`glTexCoord*()` specified at each vertex



Generating Texture Coordinates

Automatically generate texture coords

```
glTexGen{ifd}[v]()
```

specify a plane

- **generate texture coordinates based upon distance from plane**

generation modes

$$Ax + By + Cz + D = 0$$

- `GL_OBJECT_LINEAR`
- `GL_EYE_LINEAR`
- `GL_SPHERE_MAP`

Texture Application Methods

Filter Modes

- **minification or magnification**
- **special mipmap minification filters**

Wrap Modes

- **clamping or repeating**

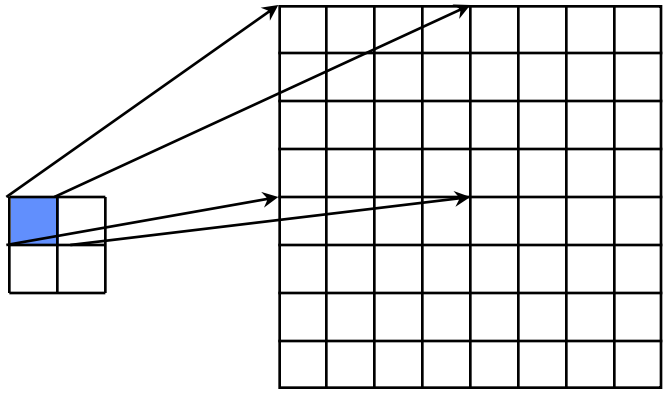
Texture Functions

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

Filter Modes

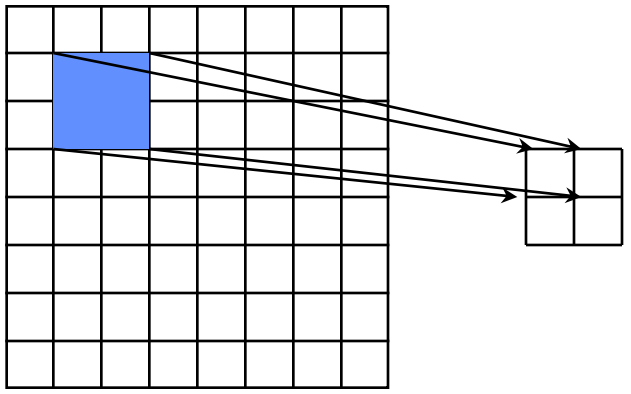
Example:

```
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

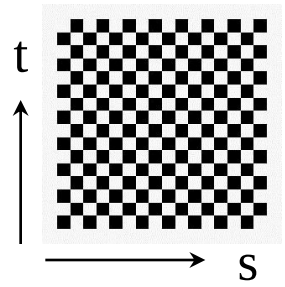
```
gluBuild*DMipmaps( … )
```

OpenGL 1.2 introduces advanced LOD controls

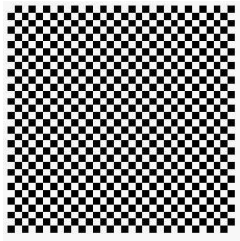
Wrapping Mode

Example:

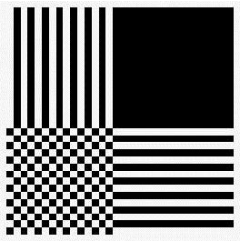
```
glTexParameteri( GL_TEXTURE_2D,
     GL_TEXTURE_WRAP_S, GL_CLAMP )
glTexParameteri( GL_TEXTURE_2D,
     GL_TEXTURE_WRAP_T, GL_REPEAT )
```



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)**

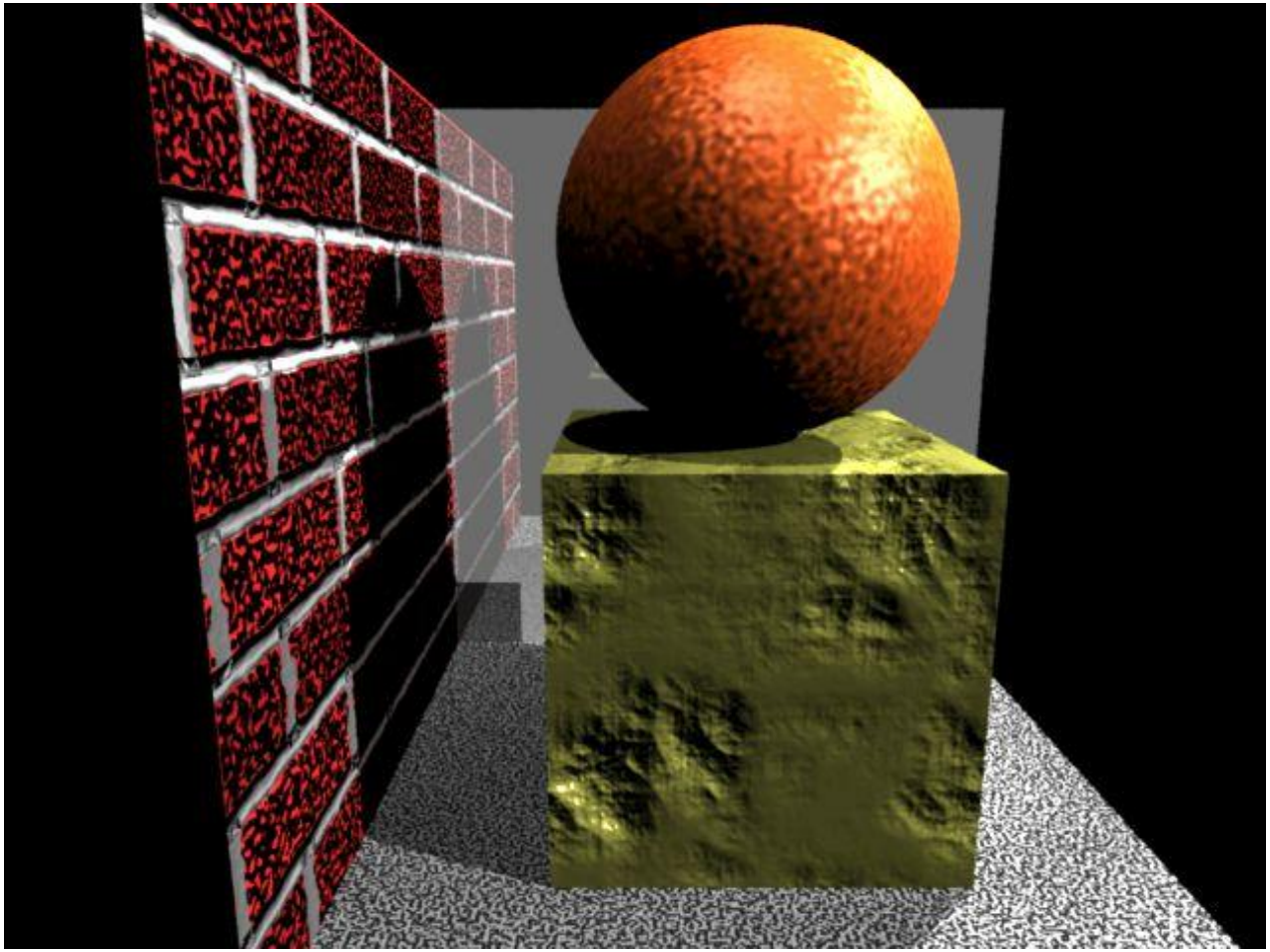
Noticeable for polygons "on edge"

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

where *hint* **is one of**

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

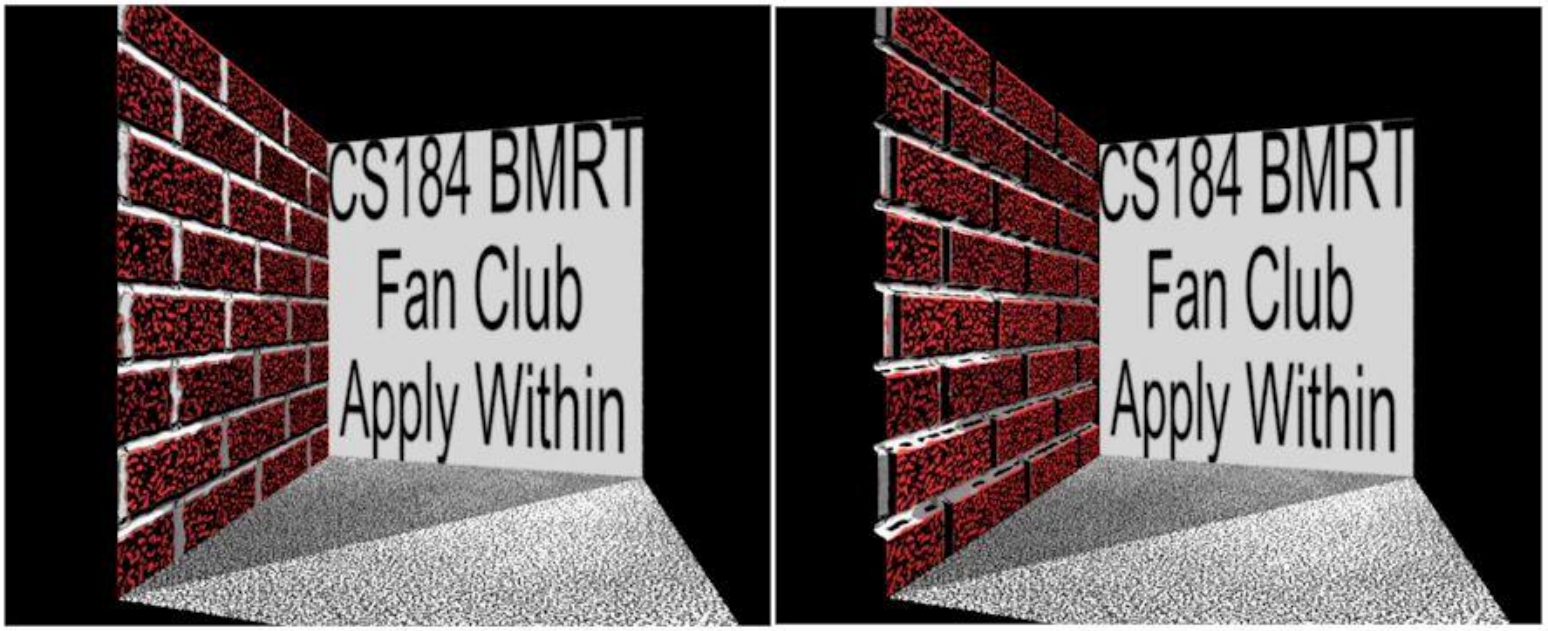
Bump Mapping



Displacement Mapping

Bump mapped normals are inconsistent with actual geometry. Problems arise (shadows).

Displacement mapping actually affects the surface geometry



Mipmaps

multum in parvo **-- many things in a small place**

A texture LOD technique

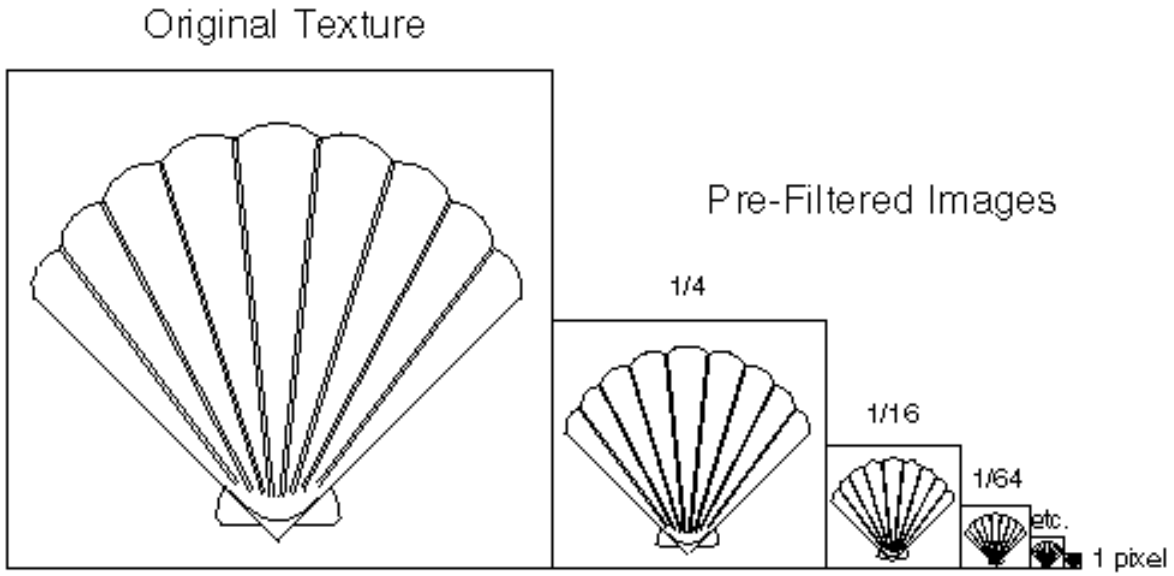
Prespecify a series of prefiltered texture maps of decreasing resolutions

Requires more texture storage

Eliminates shimmering and flashing as objects move

MIPMAPS

Arrange different versions into one block of memory



MIPMAPS

With versus without MIPMAP