CS559: Computer Graphics

Lecture 27: Texture Mapping

Li Zhang

Spring 2008

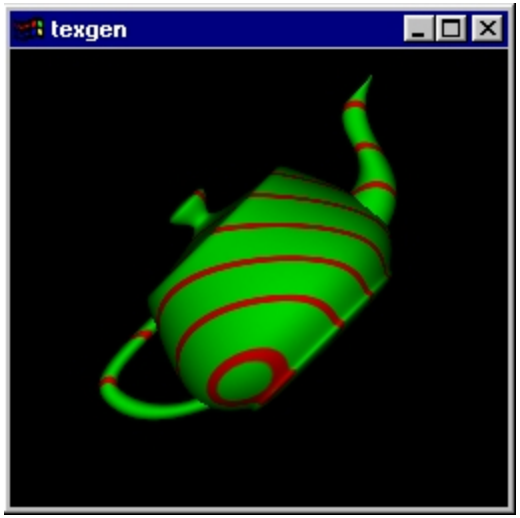
Many slides from Ravi Ramamoorthi, Columbia Univ, Greg Humphreys, UVA and Rosalee Wolfe, DePaul tutorial teaching texture mapping visually, Jingyi Yu, U Kentucky.

Today

- Continue on Texture mapping

- Reading
  - Redbook: Ch 9, Ch 6 (Blending only)
  - (highly recommended) Moller and Haines: *Real-Time Rendering, 3e*, Ch 6
    - Linux: /p/course/cs559-lizhang/public/readings/6_texture.pdf
    - Windows: P:\course\cs559-lizhang\public\readings\6_texture.pdf
  - (optional) Shirley: Ch 11.4 – 11.8

Projective (slide projector) Texture



GL_OBJECT_LINEAR vs GL_EYE_LINEAR

Bump map vs Displacement map

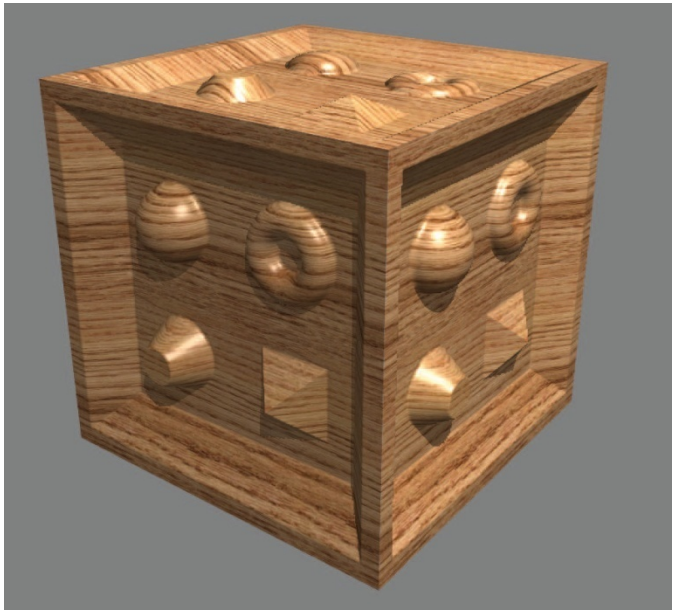
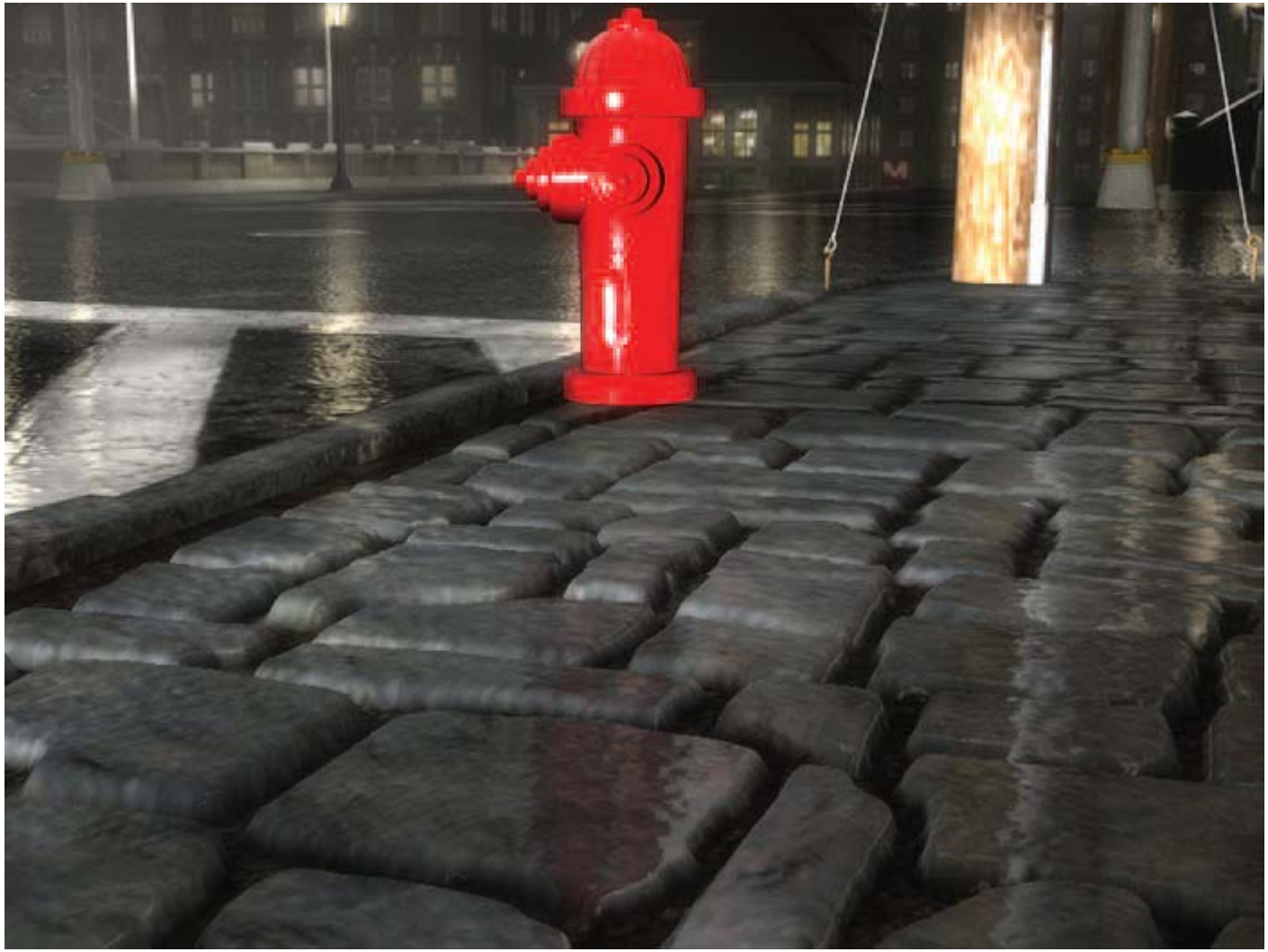
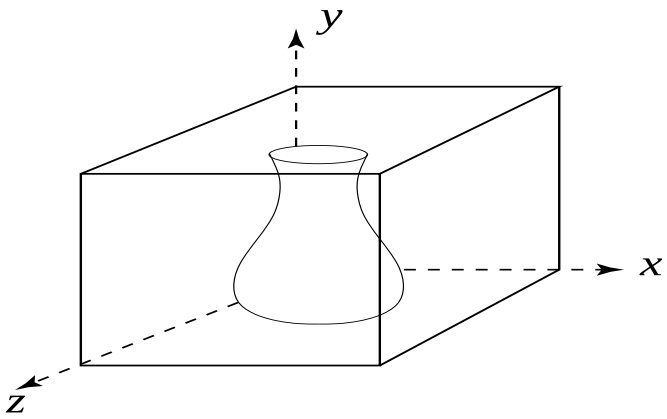


Figure 6.24 and 30 from RTR book

Displacement map in a scene



Solid textures



- Use model-space coordinates to index into a 3D texture
- Like “carving” the object from the material

- One difficulty of solid texturing is coming up with the textures.

Solid textures (cont'd)

- Here's an example for a vase cut from a solid marble texture:

- *Solid marble texture by Ken Perlin*

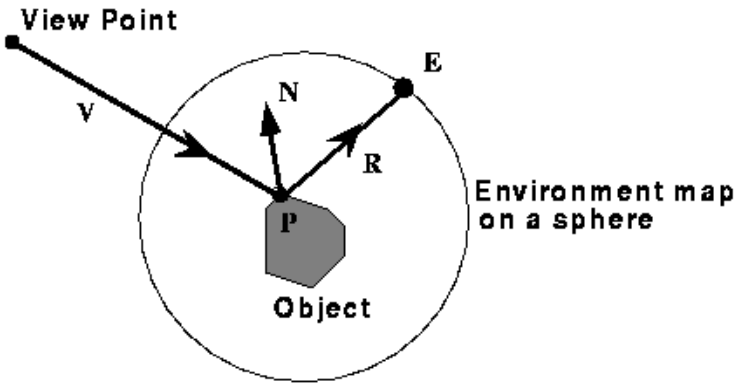
http://legakis.net/justin/MarbleApplet/

Environment Maps

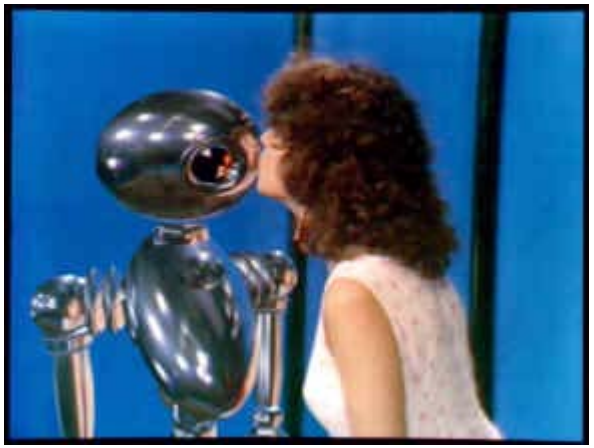
Images from *Illumination and Reflection Maps:*
Simulated Objects in Simulated and Real Environments
Gene Miller and C. Robert Hoffman
SIGGRAPH 1984 "Advanced Computer Graphics Animation" Course Notes

Environment Maps

Instead of using transformed vertices to index the projected texture', we can use transformed *surface normal* s to compute indices into the texture map. These sorts of mapping can be used to simulate reflections, and other shading effects. This approach is not completely accurate. It assumes that all reflected rays begin from the same point, and that all objects in the scene are the same distance from that point.



Applications of Environment Map



"Interface", courtesy of
Lance Williams, 1985

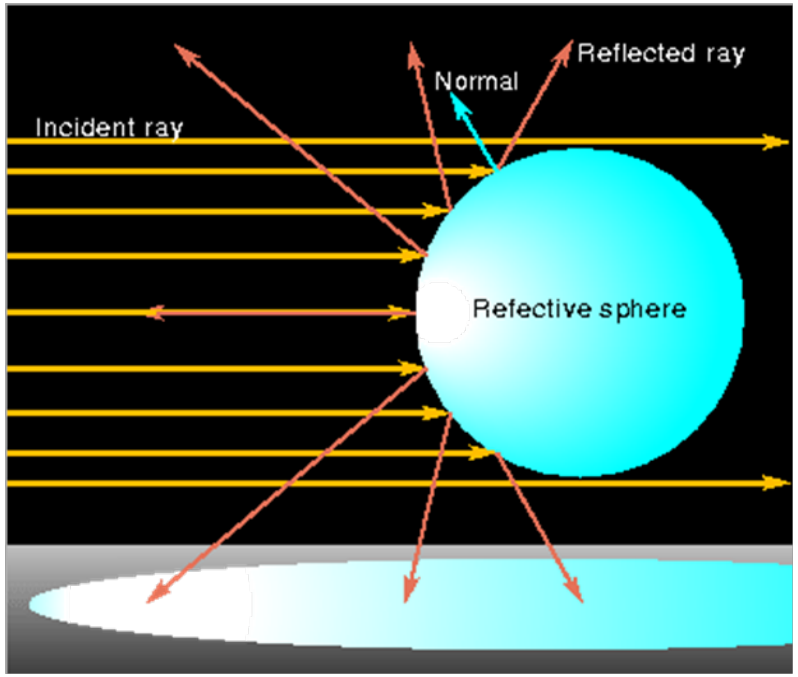
williams_robot4.mov

Terminator 2, 1991

More history info on Reflectance map
http://www.debevec.org/ReflectionMapping/

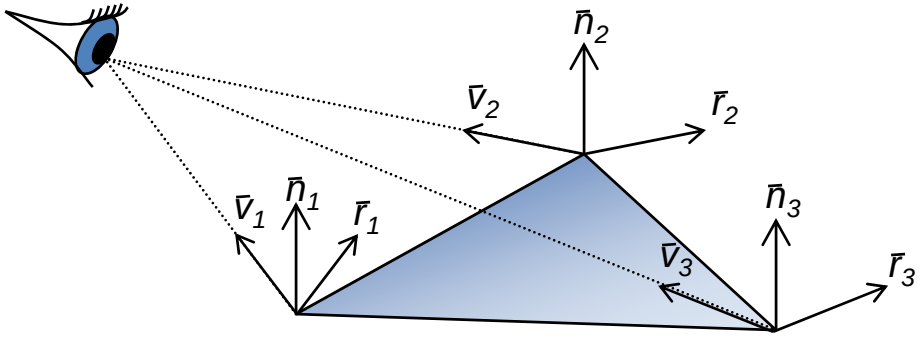
Sphere Mapping Basics

- OpenGL provides special support for a particular form of Normal mapping called sphere mapping. It maps the normals of the object to the corresponding normal of a sphere. It uses a texture map of a sphere viewed from infinity to establish the color for the normal.



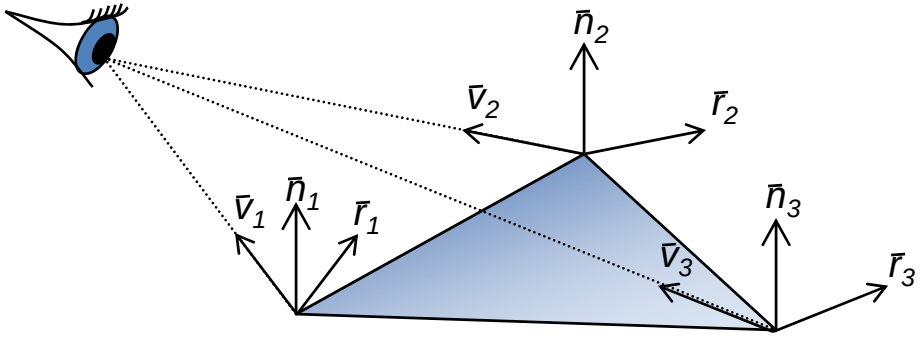
Sphere Mapping

- Mapping the normal to a point on the sphere



Sphere Mapping

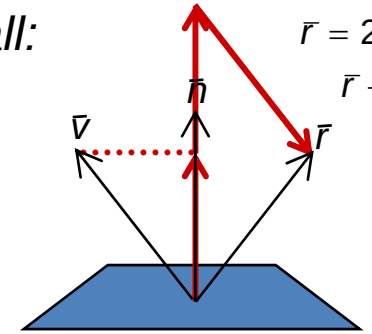
- Mapping the normal to a point on the sphere



Recall:

$\bar{r} = 2(\bar{n} \cdot \bar{v})\bar{n} - \bar{v}$

$\bar{r} + \bar{v} = \alpha \bar{n}$

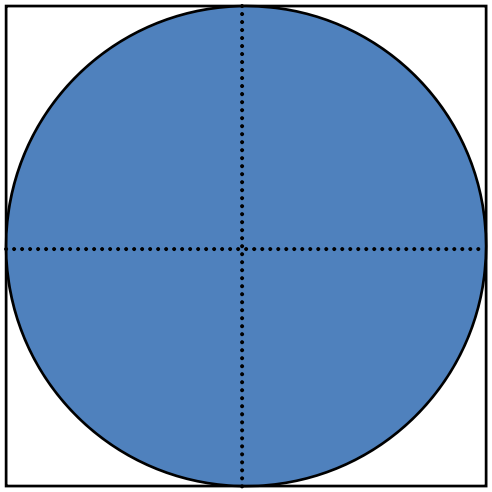


$$\alpha\bar{n} = \bar{r} + \bar{v} = \begin{bmatrix} r_x \\ r_y \\ r_z \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\frac{\alpha\bar{n}}{\|\alpha\bar{n}\|} = \begin{bmatrix} \frac{r_x}{p} \\ \frac{r_y}{p} \\ \frac{r_z+1}{p} \\ 0 \end{bmatrix}$$

$$p = \sqrt{r_x^2 + r_y^2 + (r_z + 1)^2}$$

(1,1)



(-1,-1)

$$\bar{n} = \begin{bmatrix} s \\ t \\ \sqrt{1 - s^2 - t^2} \\ 0 \end{bmatrix}$$

$$s = \frac{r_x}{p} \quad t = \frac{r_y}{p}$$

$$s' = \frac{s}{2} + \frac{1}{2} \quad t' = \frac{t}{2} + \frac{1}{2}$$

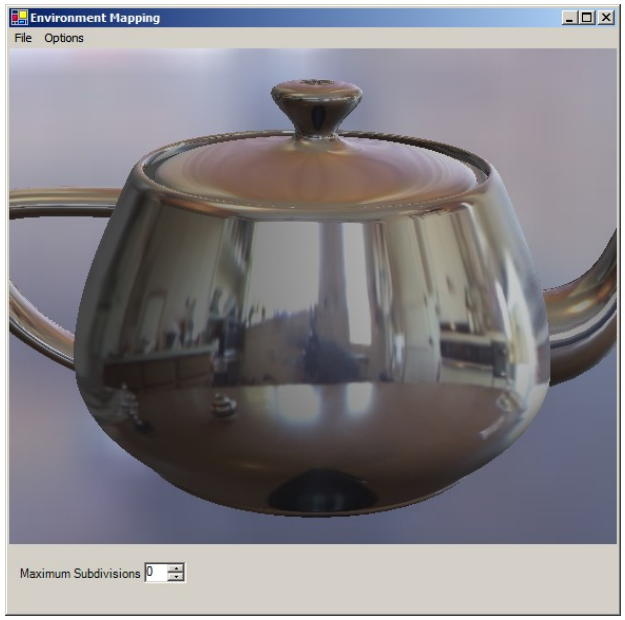
$$s' = \frac{r_x}{2p} + \frac{1}{2} \quad t' = \frac{r_y}{2p} + \frac{1}{2}$$

OpenGL code Example

```
// this gets inserted where the texture is created

glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, (int) GL_SPHERE_MAP);
glTexGeni(GL_T, GL_TEXTURE_GEN_MODE, (int) GL_SPHERE_MAP);

glEnable(GL_TEXTURE_2D);
glEnable(GL_TEXTURE_GEN_S);
glEnable(GL_TEXTURE_GEN_T);
```



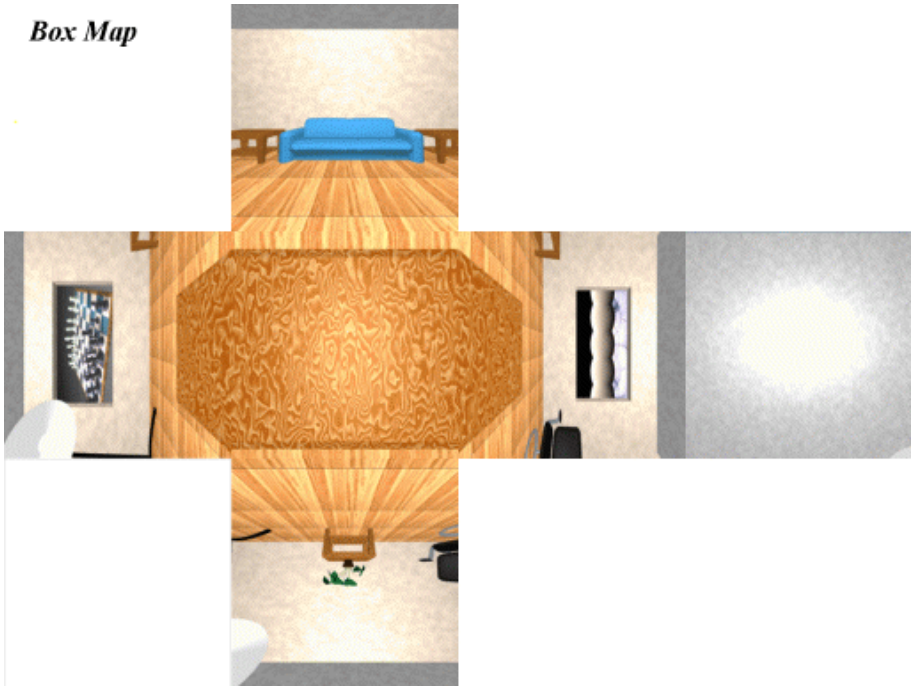
Without a mirror ball, where to get environment map images?
http://www.debevec.org/probes/

When doesn't EM work well?

- Flat/planar surface
  - In particular under orthographic projection

What's the Best Map?

A sphere map is not the only representation choice for environment maps. There are alternatives, with more uniform sampling properties, but they require different normal-to-texture mapping functions.



Multi-Texturing

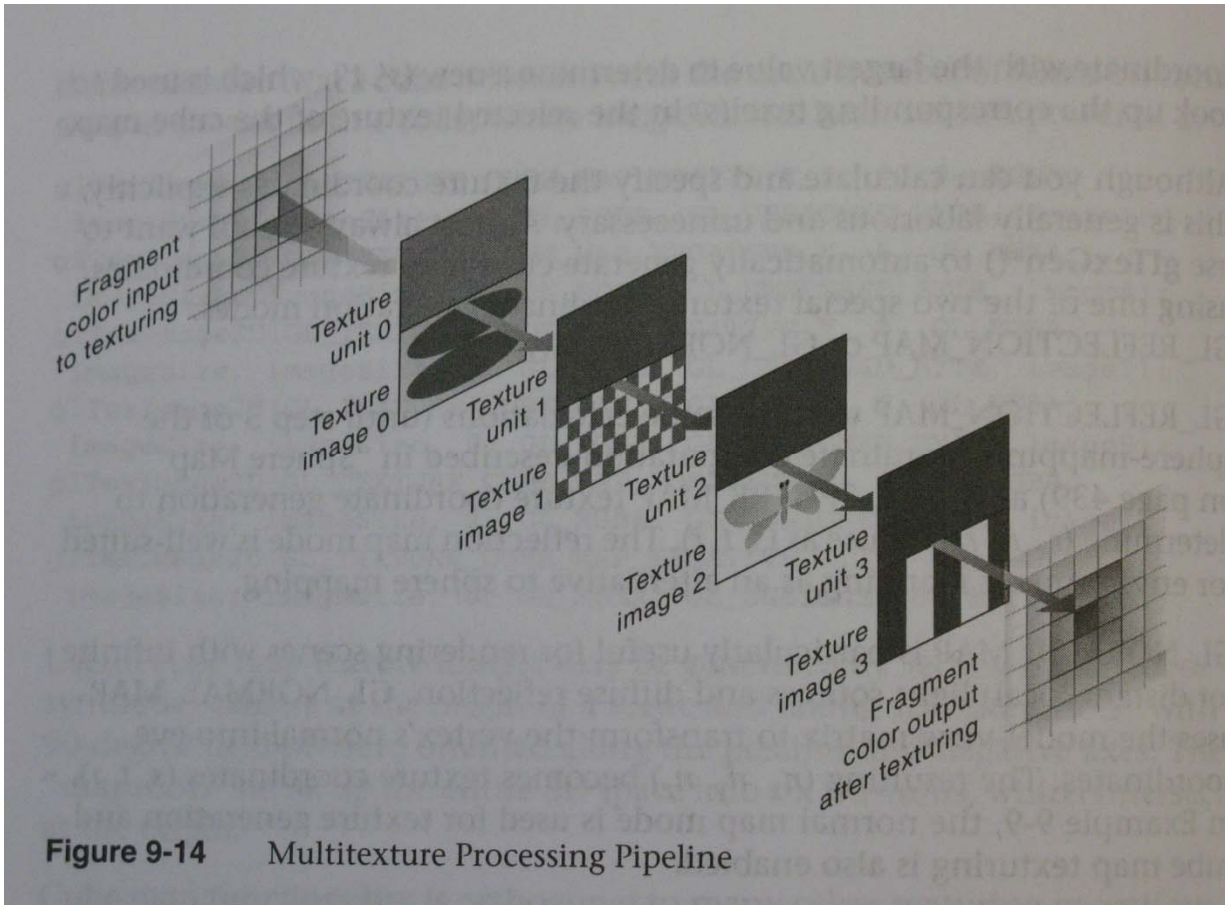


Figure 9-14 Multitexture Processing Pipeline

Why Multi-texturing

- Interpolating Textures
  - From day to night

- Implementing Bump maps
  - Create texture for diffuse coefficient, light vector, and normal maps: kd * max(L·n, 0)

Multi-Texture: OpenGL

Gluint texture0, texture1;
//in Initi() Load in images, Initialize textures as before
//in Display() do the following

```
// bind and enable texture unit 0
glActiveTexture (GL_TEXTURE0);
glBindTexture (GL_TEXTURE_2D, texture0);
glEnable (GL_TEXTURE_2D);

// bind and enable texture unit 1
glActiveTexture (GL_TEXTURE1);
glBindTexture (GL_TEXTURE_2D, texture1);
glEnable (GL_TEXTURE_2D);

// specify two sets of texture coordinates for each vertex
glBegin (GL_TRIANGLES);
    glMultiTexCoord2f (GL_TEXTURE0, 0.0, 1.0);
    glMultiTexCoord2f (GL_TEXTURE1, 1.0, 0.0);
    glVertex3f (...)
    ...
glEnd ();
```

Combining Textures

```
glActiveTexture( GL_TEXTURE0 );
glEnable( GL_TEXTURE_2D );
glBindTexture(GL_TEXTURE_2D, texName0);
glTexEnvi( GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE );

glActiveTexture( GL_TEXTURE1 );
glEnable( GL_TEXTURE_2D );
glBindTexture(GL_TEXTURE_2D, texName1);
int choice = 0;
if (choice == 0) //modulate the two textures
{
   glTexEnvi( GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE );
}
else if (choice == 1) //interpolate the two textures using the alpha of current primary color
{
   glTexEnvi( GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_COMBINE );
   glTexEnvi( GL_TEXTURE_ENV, GL_COMBINE_RGB, GL_INTERPOLATE );
   glTexEnvi( GL_TEXTURE_ENV, GL_SRC0_RGB, GL_TEXTURE1);
   glTexEnvi( GL_TEXTURE_ENV, GL_SRC1_RGB, GL_PREVIOUS);
   glTexEnvi( GL_TEXTURE_ENV, GL_SRC2_RGB, GL_PRIMARY_COLOR);
   glTexEnvi( GL_TEXTURE_ENV, GL_OPERAND0_RGB, GL_SRC_COLOR);
   glTexEnvi( GL_TEXTURE_ENV, GL_OPERAND1_RGB, GL_SRC_COLOR);
   glTexEnvi( GL_TEXTURE_ENV, GL_OPERAND2_RGB, GL_SRC_ALPHA);
}
else                    //disable the second texture
 {
   glActiveTexture(GL_TEXTURE1);
   glDisable(GL_TEXTURE_2D);
}
// continue on the right
```

```
glColor4f(0,0,0,0.2);

glBegin(GL_QUADS);
glMultiTexCoord2f(GL_TEXTURE0, 0.0, 0.0);
glMultiTexCoord2f(GL_TEXTURE1, 0.0, 0.0);
glVertex3f(-2.0, -1.0, 0.0);

glMultiTexCoord2f(GL_TEXTURE0, 0.0, 1.0);
glMultiTexCoord2f(GL_TEXTURE1, 0.0, 1.0);
glVertex3f(-2.0, 1.0, 0.0);

glMultiTexCoord2f(GL_TEXTURE0, 1.0, 1.0);
glMultiTexCoord2f(GL_TEXTURE1, 1.0, 1.0);
glVertex3f(0.0, 1.0, 0.0);

glMultiTexCoord2f(GL_TEXTURE0, 1.0, 0.0);
glMultiTexCoord2f(GL_TEXTURE1, 1.0, 0.0);
glVertex3f(0.0, -1.0, 0.0);

glEnd();
```

Different Combinations for different apps

- Interpolate for fade-in-fade-out between textures
  - Night and day

- Dot product followed by modulation for bump map
  - http://www.paulsprojects.net/tutorials/simplebump/simplebump.html

Multi-Texture: OpenGL

- Each texture unit (GL_TEXTURE*) has its own state:
  - Texture image
  - Environment mode
  - Filtering parameters
  - Texture matrix stack
  - Automatic texture coordinate generation
  - …
- Create visual effect using your imagination.

GL Extensions

- Multi-texturing is not supported in visual studio.
- Need to download GL extension managers:
  - **GLEE** - http://elf-stone.com/glee.php - The "Easy" Gl Extension manager.
  - **GLEW** - http://glew.sourceforge.net/ - The Open GL Extension Wrangler.
  - More info: http://pages.cs.wisc.edu/~cs559-1/GlExtensions.htm

Texture animation

- Moving water texture to simulate flow
- zoom, rotation, and shearing image on a surface
- Fading in and fading out (Blending marble to skin) with multi-texturing