TEXTURE MAPPING 2

RECAP:
Object / Triangle
Texture Coordinate
Linear in Space
Interpolation Perspective on Screen
For each pixel do lookup for color
Bilinear interpolation since u,v is continuous
TRI-LINEAR interpolation in MIPMAP to get areas

Small GOTHCA
Lighting computed at VERTEX
Color at PIXEL

1. do GOURAUD Shading
Texture Modulates ← multiplies color
\[
\text{color} = C_0 \ (N \cdot L) + C_s \ (H \cdot N)^p + C_A \ L_a
\]
\[
\text{color} = C_T \ (---)
\]
→ Make objects white / mult over color
→ Doesn't allow control of specular color separately
GL has a workaround
→ Modulate color
other choices (replace, subtract)
Opportunity to Combine multiple color sources
Combine Multiple Textures = Multi-Texture
(more on that later)
Additional Texture Map Tricks

Objects look too flat

**Bump Mapping** ← use texture to change normal
need to compute lighting per-pixel (since normal changes)

Surface is flat, but reflects as if bumpy.

**AKA Normal Maps**

Note: offsets to "real" normal are what's stored
for triangle have "u" and "v" vector

\[ \mathbf{n} = \mathbf{n} + \alpha \mathbf{u} + \beta \mathbf{v} \]

vectors from triangle

Only changes lighting
no self-shadowing (hacks to fix)
doesn't change silhouette
per-pixel lighting required

**Displacement Mapping**

Actually move point (which changes normal)
Very hard to do (pixel might move to other pixel)
Gots realistic effects
How to fake this without per-pixel computation

View Dependent Texture Mapping
- determine what object looks like under different view directions
- Blend between different textures $T_1$, $T_2$, $T_3$, $\frac{T_2 + T_3}{2}$, $T_3$

Use of Multi-Texture
1. Multipass w/ multiply or blend
2. Texture Combining

- Use texture over whole triangle per-pixel effects pre-computed (need some way to compute them)
- Can vary based on light direction as well

More Mapping:

Environment mapping
- make mirror reflections
- assume object is infinitesimal sphere $R \rightarrow N$ $R$ depends on $E, N$

Assume $E$ is constant

Use $R$ to look up into Map of "Environment"

Cylindrical environment map

Spherical environment map

Cubic environment map
Lighting w/ Texture

Environment Map is mirror reflection (specularity) put lights in it to get realistic lights very bright spots in texture (use of HDR)

"Paint" Lighting Details on to objects

→ Multi-Texture to add layers of lights

→ Slide Projector Mapping Light point

→ u, v on light's "image" plane

Shadow Maps - something different

1. render scene from light's point of view
2. Visible objects are lit, occluded are in shadow render and keep the z-buffer (the shadow map)
3. draw from the camera's viewpoint for any pixel, see its distance from light check in map to see if occluded
Hack Shadows / Hack Lights
draw black or light splotches

How to control where they go?
How to avoid overdrawing

Stencil Buffer
A buffer you can do anything with
Write values w/ drawing
Test values w/ drawing

Example
1) clear to zero
2) draw ground set stencil bit
3) draw shadows
   - only draw when stencil bit set
   - reset stencil bit when drawing