

Handouts: Smooth Shading

Advanced Illumination



- Bump Mapping
- Transparency

Bump Mapping



- Adds "texture" using lighting
- Perturb surface normal
 - Displace point on surface
 - Compute new normal

Handouts: Smooth Shading

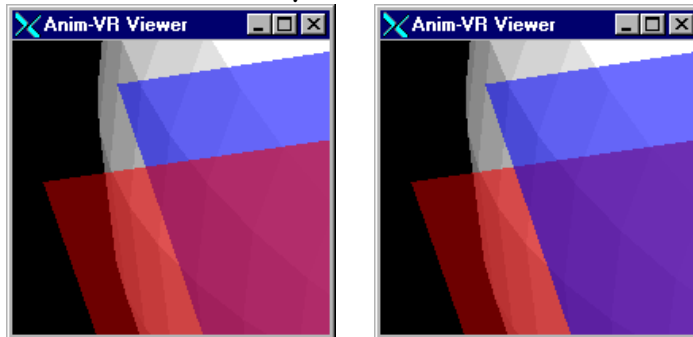
Non-refractive Transparency

- Interpolated transparency
 - ie. OpenGL alpha values
 - $I = (1 - k_t) I_A + k_t I_B$
- Screen door transparency



Simple Rendering of Transparent Polygons Fails

- Even if rendered after opaque
- If more than one, order matters!



Handouts: Smooth Shading

OpenGL Texture Mapping



- `glEnable(GL_TEXTURE_2D)` or 1D
- Pick names for texture objects
 - `glGenTextures` to get free name
 - (analogous to `glGenLists`)
- `glBindTexture` creates, uses tex. objs.
 - Sets "current texture", create on first use

Texture Parameters



- specify 2D (or 1D) images
 - 2^n dimensions (plus borders, mipmaps)
- Specify how to apply it
 - DECAL, REPLACE, MODULATE, BLEND
- Wrap or Clamp
- Filtering
 - Min, Mag

Handouts: Smooth Shading

Supply Texture Coords (when drawing vertices)

- `glTexCoord*` (analogous to `glVertex*`)
 - `(s,t,r,q)` instead of `(u,v)`
 - go from 0..1
- Possible to have them generated
 - `glTexGen`

Supplying the Image

- `glTexImage2D`