

EECE 478

Multi-Pass Rendering

Multi-Pass Rendering

- Use buffers as processing stages
  - Front/back buffer – *color/compositing*
  - Depth buffer – *distance/occlusion*
  - Stencil buffer – *masking/logical ops*
  - Accumulation buffer – *general calculation*

Traditional MPR

- Render first-pass scene
- Use results to control next pass
  - Copy into texture
    - Done using OpenGL glCopyTexImage
  - Use depth/stencil buffers

Modern MPR

- Vertex and Fragment Programming
  - Use GL_ARB_vertex_program & GL_ARB_fragment_program
  - Assembly language for pipeline
    - Similar to RenderMan backend
    - Closer to techniques used for movies
    - Still real-time
  - Equivalent DirectX functionality
- High-level languages *Cg* or *Sh*

Example: Shadows

Possible approaches:

- Projection shadows (simple, floors)
- Shadow volumes (moderate, general)
- Shadow maps (complex, v-general)

Good online tutorials:

http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=27
http://www.sgi.com/software/opengl/advanced97/notes/node99.html
http://www.3ddrome.com/articles/shadowvolumes.php
http://www.csee.umbc.edu/~olano/s2002c36/

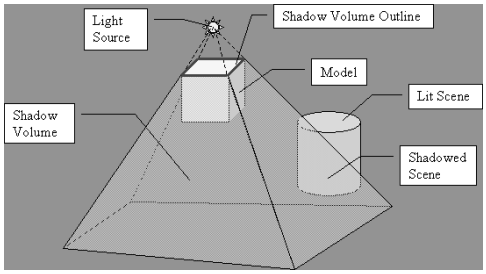
Shadow Volumes (Z-Pass)

Render scene with all objects and lights

For every light source and object casting shadow:

- Turn off depth update; render only to stencil buf
- Render visible object faces facing toward light
  - Incrementing stencil
- Render visible object faces facing away from light
  - Decrementing stencil
- Enable stencil test != 0
- Draw translucent black rect over entire scene!

Simple Shadow Volume



ShadowVols in Cg

http://www.gamedev.net/reference/articles/article1990.asp

Use a vertex program

- Applies to every vertex in pipeline
- Replaces T&L pass (must simulate if needed)
- Cannot create new vertices or faces!
- Can do all mat-vec math on GPU!

Cel Shading

Use a 1D texture map for shade colors

Render object with shade map enabled and lighting disabled (cel shading)

- Per-vertex texture coordinate $s = (n \bullet l)$

Re-render object with textures off

- Cull *front* faces
- Render as wireframe with linewidth ≥ 2