

Textures for Data Storage: Shadows

CSCI 4830/7000

**Advanced Computer Graphics
Spring 2012**

Shadows in Computer Graphics

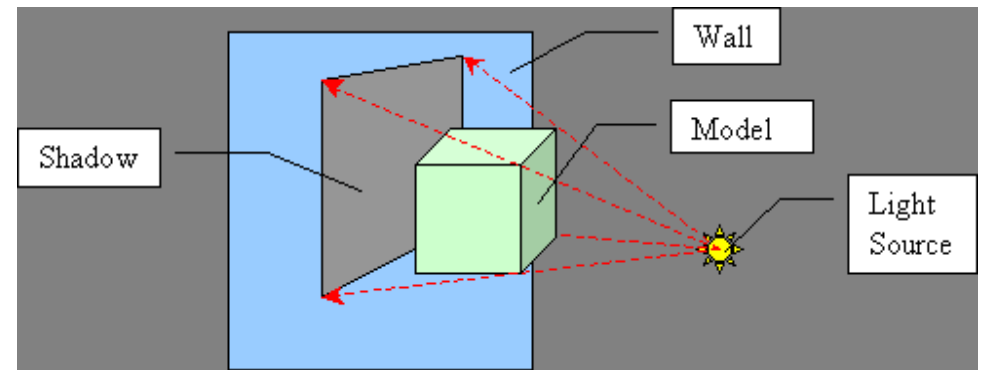
- Shadows are important to realism
 - Depth cues
 - Relative positions of objects
- Doesn't “just happen” when lighting is turned on
 - Nor is there a `glEnable(GL_SHADOWS)`
- Scene must be rendered 2-4 times
- Shader implementation can be efficient
 - Draw once every time the light or scene changes
 - Draw once for every eye position

Shadow Examples

- Planar Shadows (CSCI 5229 ex32)
 - Shadows on the floor only
- Shadow Volumes (CSCI 5229 ex33)
 - True shadow, very hard
- Shadow Maps (CSCI 5229 ex34)
 - True shadows, depth in textures
- Shader Shadow Map (ex18)
 - Fast implementation via shader

Planar Shadows

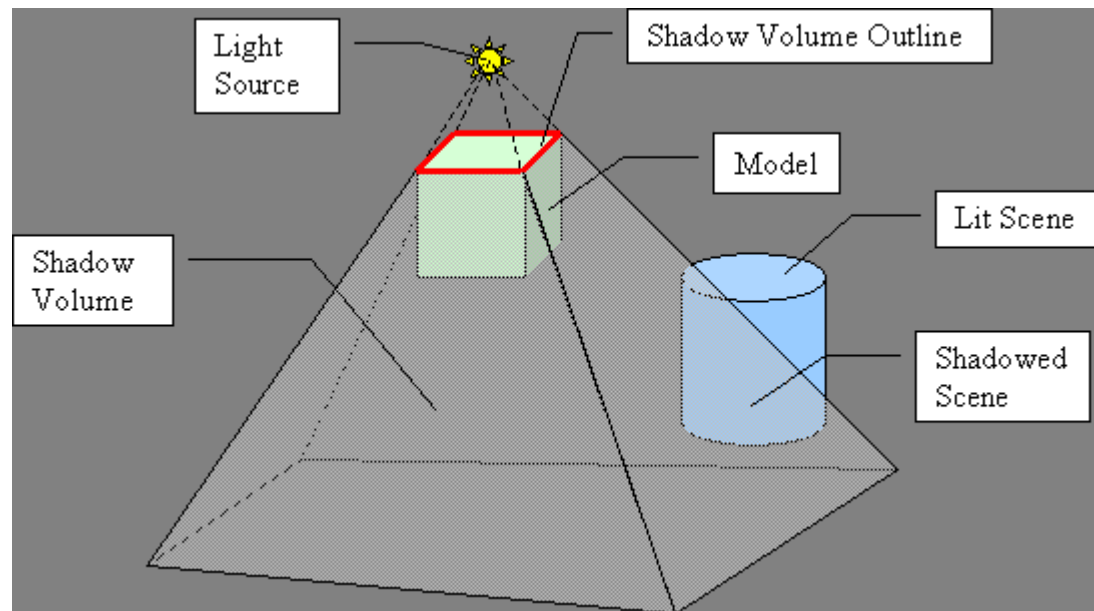
- Projects object on surface
- Simplest shadows
- Fast but very limited
- The problem:



- Surface defined by point E and normal N
- L is the light
- P is on the object
- Find P' the projection of P on the surface

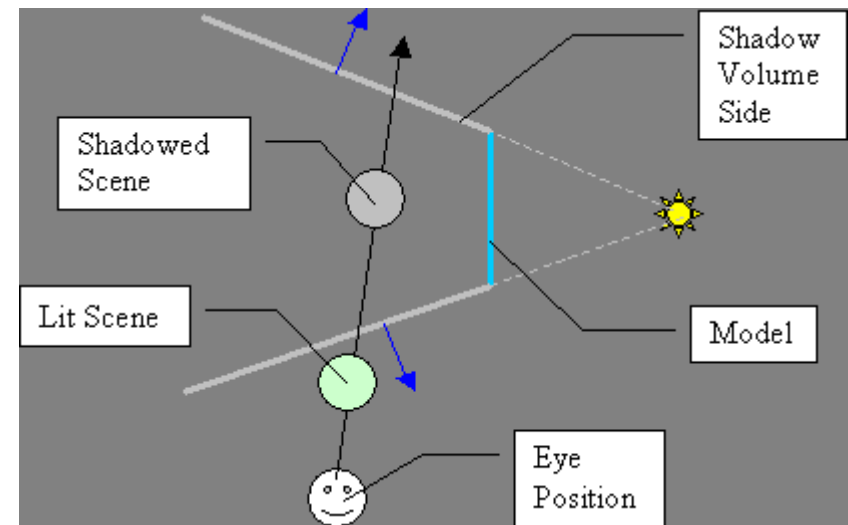
Shadow Volumes

- The volume corresponding to the shadow cast by a facet of each object
 - Potentially multiple shadow volumes per object
 - Shadow of the object is the combination of all shadow volumes for the object



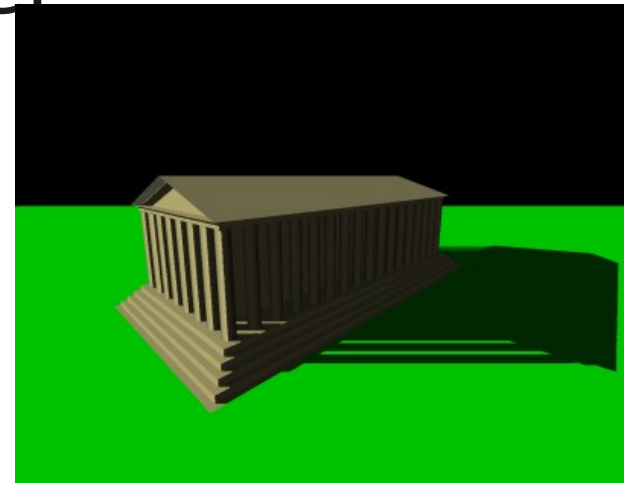
Shadow Volume Algorithm

- Count transitions in and out of shadow volumes
 - Increment of in, decrement for out
 - Similar to polygon winding rule for in/out
- Lit areas has value of zero (initial value)



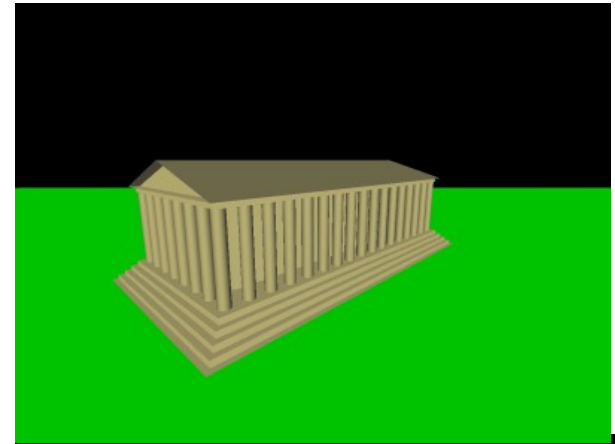
Shadow Mapping

- Project with light as viewpoint
- Depth buffer from light
- Light/shadow determined just like visibility
 - Objects in light foremost in depth buffer
 - Objects in shadow depth obscured
- Requires second depth buffer
 - Store depth to texture
 - Compare R to texture
- In OpenGL extensions
- Used in *Toy Story* etc.

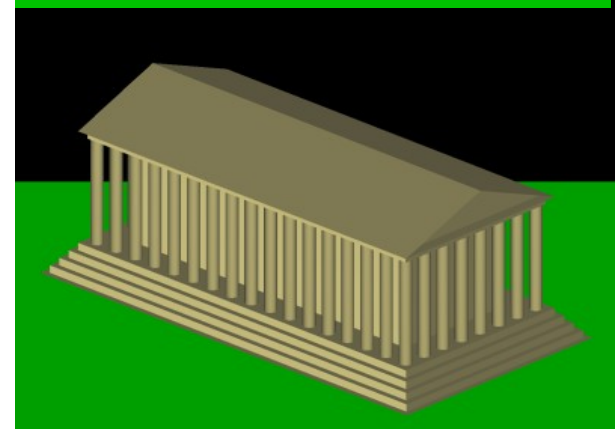


Shadow Map Example

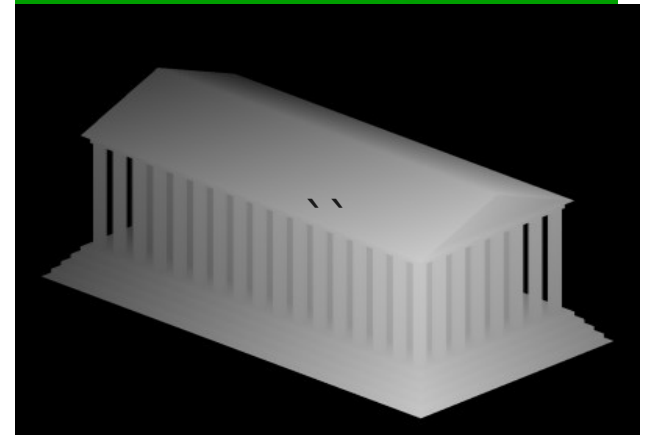
No Shadows



Light View



Light View Depth



Shadow Map Shader

- Draw shadow map
 - Bind framebuffer to depth texture
 - Draw scene with eye at light to generate depths
 - Update if light or scene changes
- Draw scene
 - Generate texture coordinates with light PoV
 - Compare depth (R) with depth texture
 - $R = \text{depth}$ means lit - light as normal
 - $R > \text{depth}$ means shadowed - ambient light only
- Fast, Simple, Realistic