Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

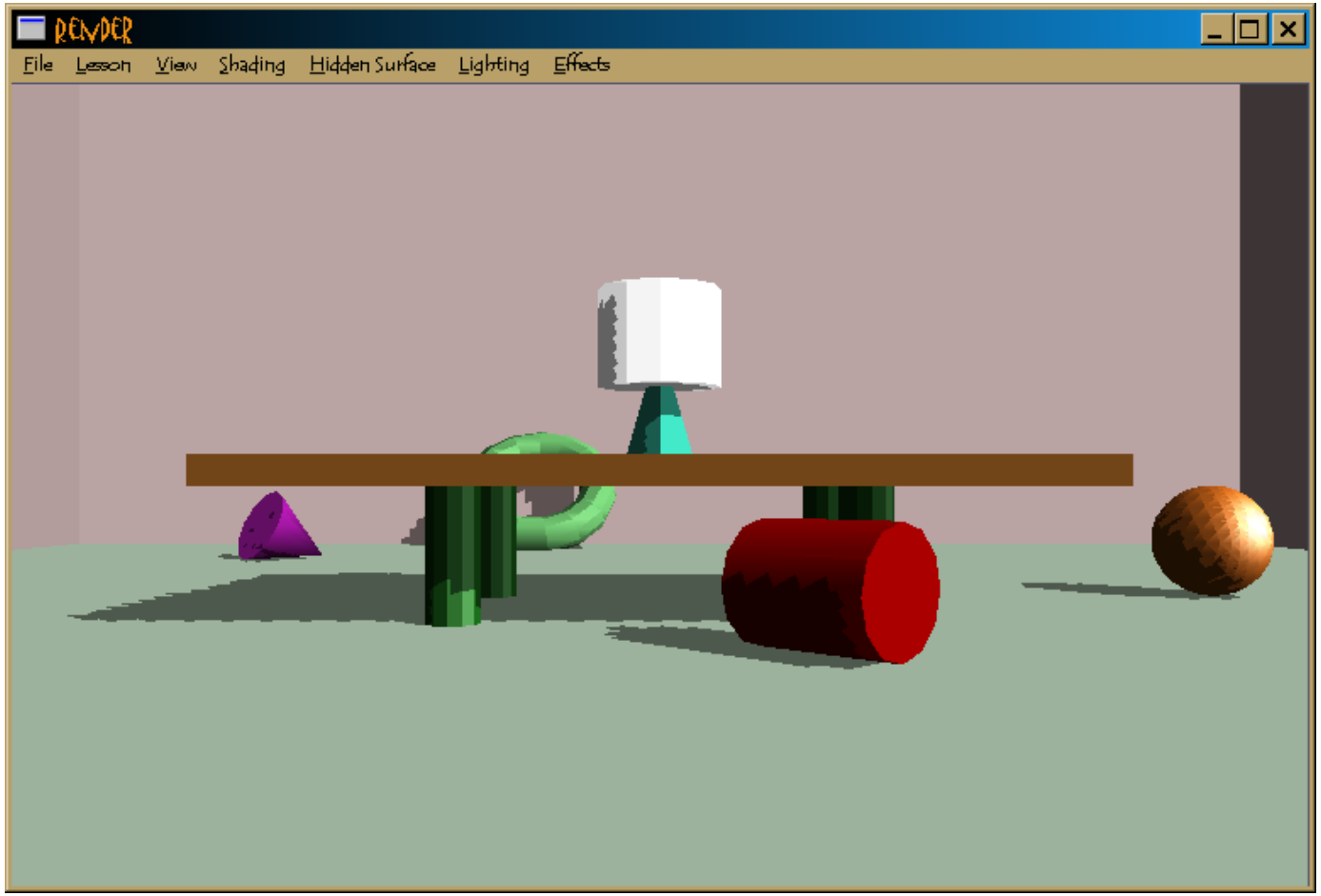
Brian Russin

Module 8A
Shadows

Shadows and Planar Reflections

- Introduction to Shadows
  - Definition of terms
  - Nature of shadows
- Stencil buffer
- Shadow methods
  - Planar, projected shadows
  - Shadow volumes
  - Shadow map / shadow Zbuffer
  - Using textures for shadows
    - Projected textures, light maps, soft shadows
- Planar reflections

Shadows



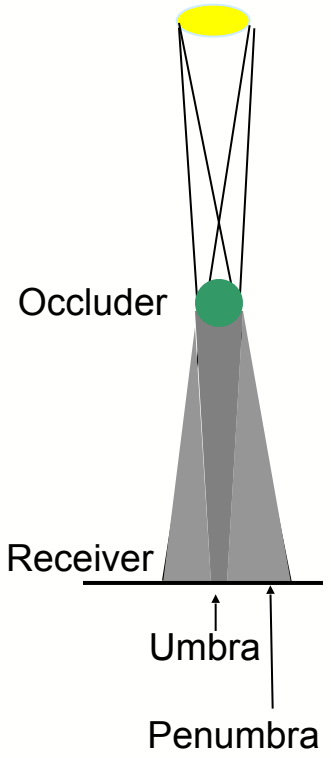
Example of Shadow Mapping with Software Lab (1999)
Note the aliasing and self shadowing issues

Shadows

- Shadows play a vital, yet subtle, role in visual perception
  - Provide information as to how objects relate to each other in space
    - Anchor objects in relation to each other
  - Help eliminate the 'floating' effect of 3D objects
  - Aid depth perception
  - Emphasize changing direction of the light source
- Provide an important visual cue
  - Lack of shadows makes the scene not look real
    - “Real” world scenes have shadows
    - Clue that an image is computer generated
- Can provide artistic effect
  - Soft vs. hard feel
  - Dramatic or spooky feel

Shadow Definitions

- **Occluders** are objects that cast shadows
- **Receivers** are objects onto which the shadow is cast
  - Shadows vary as a function of lighting environment
  - **Umbra** - part of the shadow that is completely cut off from the light source
  - **Penumbra** - part of the shadow receiving some light from the source
    - Penumbra surrounds the umbra
    - Gradual intensity change from the umbra to penumbra
  - **Hard** edged shadow has minimal penumbra
  - **Soft** edged shadows has wider penumbra
- Size and shape of the umbra and penumbra depend on size and shape of the light source, distance from the object, and distance which the shadow is cast



Aspects of Shadows

- Research shows it is better to have an inaccurate shadow than none
  - Eye is forgiving about the shape of the shadow and presence of penumbra
- Graphics algorithms exploit the following aspects of shadows
  - Point light sources produce shadows with no penumbra
    - Hard edge
  - No shadows are seen if the view point and (single) light source are coincident
    - Shadows are areas hidden from the light source
      - Implies: hidden surface algorithms can be used to calculate shadow shapes
  - Shadows are fixed in a static scene
    - Moving the viewpoint will not change shadow positions
    - Must recalculate shadow shapes if light source or object positions change
    - “Pre-calculated” shadows work well in static scenes

Shadows in Computer Graphics

- Need to determine two essential aspects of shadows
  - The **shape** of the shadow
    - Easy to compute if shadows are cast onto a plane
    - More complex if shadows are cast onto other objects
  - The **intensity** of the light reflected from within the shadow area
    - Does not in general reduce to 0 due to secondary reflections
      - Accurate determination is generally in the domain of global illumination
- Does not fit naturally into the rendering pipeline
  - Requires knowledge of other surfaces in addition to that being rendered
  - Often requires **multi-pass** rendering techniques
    - Geometry is rendered multiple times

Shadows and Lighting

- Illumination model in standard graphics pipeline is a **local** model
  - Only consider light's direct path to the surface
  - Local lighting models ignore occlusion of light sources that create shadows
    - Except trivial self-shadowing ($L \bullet N < 0$) for a single polygon
- Beyond simple cases shadows need to be treated as part of the **global** illumination problem
  - Reflecting surfaces become secondary light sources
  - Shadow methods built into the solution
  - Radiosity and ray tracing
    - Generally too expensive for interactive use
- **Interactive shadow algorithms** do not impact local lighting models
  - Most are based on object geometry (e.g., planar projection)
    - Not really modeling underlying physics of light

Introduction to Shadow Algorithms

- Empirical shadow algorithms have been developed
  - Deal with the geometric aspect of shadows
  - Developed as embellishments to most rendering systems
    - Used because of the expense of global illumination techniques
    - Not so neatly incorporated into reflection / illumination model
      - Except in ray-tracing and radiosity models
    - No simple software API
  - Exploit hardware features such as stencils, texture mapping
- Shadow algorithms can be considered a special case of visible surface algorithms
  - Determine which objects can be seen from a light source
  - Occlusion from light's point-of-view instead of viewer's
- Several algorithms have limitations
  - Generally hard to get everything shadowing everything
  - Make simplifying assumptions
    - Treat area light sources as points