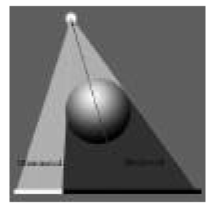
CS 4731: Computer Graphics
Lecture 19: Shadows

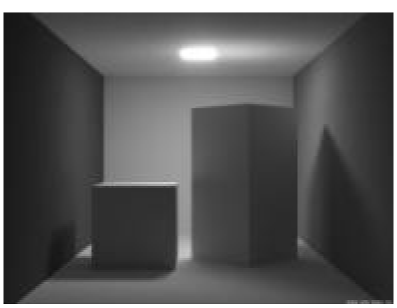
Emmanuel Agu

Introduction to Shadows

Basic idea:



Introduction to Shadows



Introduction to Shadows

- Shadows:
  - Make image more realistic
  - Important visual cues on relative positions of objects in scene
- Rendering shadows:
  - Points in shadow: use only ambient component
  - Points NOT in shadow: use all lighting components
  - Simple illumination models == simple shadows
- Two methods:
  - Shadow buffer
  - Shadows as texture (projection)
- Third method used in ray-tracing (in advanced graphics class)

Shadow Buffer Approach

- Uses second depth buffer called shadow buffer
- Pros: not limited to plane surfaces
- Cons: needs lots of memory
- Theory:
  - Establish object-light path
  - Other objects in object-light path = object in shadow
  - Otherwise, not in shadow

Shadow Buffer Approach

- Shadow buffer records object distances from light source
- Shadow buffer element = distance of closest object in a direction
- Rendering in two stages:
  - Loading shadow buffer
  - Rendering the scene

Loading Shadow Buffer

- Initialize each element to 1.0
- Position a camera at light source
- Rasterize each face in scene updating pseudo-depth
- Shadow buffer tracks smallest pseudo-depth so far

Loading Shadow Buffer

- Shadow buffer calculation is independent of eye position
- In animations, shadow buffer loaded once
- If eye moves, no need for recalculation
- If objects move, recalculation required

Shadow Buffer (Rendering Scene)

- Render scene using camera as usual
- While rendering a pixel find:
  - pseudo-depth D from light source to P
  - Index location [i][j] in shadow buffer, to be tested
  - Value d[i][j] stored in shadow buffer
- If d[i][j] < D (other object on this path closer to light)
  - point P is in shadow
  - set lighting using only ambient
- Otherwise, not in shadow

Shadows as Texture

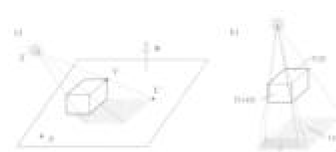
- Paint shadows as a texture
- Works for flat surfaces illuminated by point light source
- Problem: compute shape of shadow

Shadows as Texture

- Project light-object edges onto plane
- Want shadow of entire object
- Theory: union of projections of individual faces = projection of entire object
- Algorithm:
  - First, draw plane using specular-diffuse-ambient components
  - Then, draw shadow projections (face by face) using only ambient component

Shadows as Texture

- **Problem:** find outline of shadow by calculating projections of object vertices onto plane
- Example: want to project vertex *V* to find *V'*
- Plane passes through point A and has normal, **n**



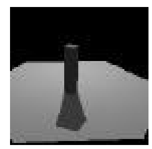
Shadows as Texture

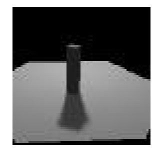
$$V' = S + (V - S)\frac{\mathbf{n}\cdot(A - S)}{\mathbf{n}\cdot(V - S)}$$

Note: can express projection in homogeneous coordinates and use matrices

Other Issues

- Point light sources => simple but a little unrealistic
- Extended light sources => more realistic
- Shadow has two parts:
  - Umbra (Inner part) => no light
  - Penumbra (outer part) => some light





References

- Hill, 8.6

Preview of Projects

- Project 4 due today (Friday)
- Still to be done:
  - Project 5: on class website later today
  - Final Exam
- Project 5
  - Write portions of graphics pipeline
  - Example:
    - used calls like glTranslate, glrotate
    - Learnt matrices and math with numerical problems
    - Project 5: apply these in graphics pipeline to build your own glTranslate, glrotate

Preview of Projects

- Project 5
  - Previously ran application using pure -openGL switch
  - E.g **cs4731app –openGL –hw02**
  - mgl.mglTranslate simply called glTranslate
  - Now run using –cs4731GL switch
  - E.g **cs4731app – cs4731GL – hw02**
  - Program now calls your glTranslate, emmanuel_glTranslate
  - Project 5 goal:
    - to give taste of what goes into building a language like openGL
    - Application of theory, matrix and vector math

Preview of Projects

- Project 5: final words
  - Some people view this as hardest project
  - Start early, you will have problems
  - Check calculations frequently
  - Will organize help session on Tuesday/Wednesday next week

Final Exam

- Similar to midterm
- Non-cumulative, covers lectures 13-24
- Posted powerpoint slides on website
- Most similar to midterm, last year's final
- Same rules:
  - In-class: Thursday, October 16
  - Review session: Tuesday, October 14
  - 1 cheat sheet, 1 calculator
- Less mathy, more algorithmic, conceptual
- For some reasons students find it harder to describe things
- Also comes in finals week, so less time to prepare