Real-Time Shadows

André Offringa
Timo Laman
Real-Time rendering

- Radiosity/Raytracing not feasible in real-time applications
- Scene rendered using projection & scan conversion of polygons
- Advantage: significant speedup
- Disadvantage: no more “natural” solutions to
  - visibility
  - lighting
  - view volume visibility
Rendering pipeline

Model Transformation → Lighting → View Transformation → Projection Transformation → Clipping → Scan Conversion
Lighting model

- After modeling transformation light contribution to polygon vertices determined
- During scan conversion light contribution to interior polygon pixels calculated by interpolating
  - intensities: Gouraud shading
  - surface normals: Phong shading
No more shadows!

- Lighting model has no provisions for shadows
- Shadows add significant amount of detail to scenes
- Therefore, methods for adding back shadows to real-time rendering model desirable
Kinds of shadows

- **Umbra**
  - Point light source
  - Directional light source
- **Penumbra**
  - Area light source
Kinds of Shadows (2)
Computation of Umbras

- Image precision – calculated per pixel in the resulting image
- Object precision – calculated viewpoint-independently
- Hybrid – combination of the above
Shadow z-buffer method

- For each light, render the scene into the z-buffer
- When rendering the view, determine shadow of scan-converted pixel by checking the z-buffers (shadow buffers) of each light:
  - Transform from view space to light space
  - Compare z-value of transformed pixel to shadow buffer
  - Equal z-value implies pixel lit, otherwise in shadow
Shadow z-buffer (2)
Shadow volumes

- Shadow planes constructed in object space
- For each point count the number of intersections with front-facing and back-facing shadow planes
- Difference = 0: point lit
- Otherwise: point in shadow
- Bergeron (1986): determine shadow when scanconverting (hence, hybrid method)
Shadow volumes (2)
Shadow Volumes (3)

- OpenGL allows for rendering shadow volumes in hardware using z-buffer and stencil buffer (Heidmann, 1991)
- This technique is used in Doom 3 (id Software, 2004)
Shadow Volume BSP

• Idea: use binary space partition to represent shadow volumes

• Steps
  – Make depth-ordering of scene polygons w.r.t. light
  – Treat all polygons in front-to-back order:
    • If in shadow volume, mark polygon as shadow
    • If outside shadow volume, mark polygon as lit and \textit{add shadow planes to SVBSP}
    • If straddling shadow plane, split polygon
SVBSP (2)
SVBSP (3)
“Fake” shadows

- Create umbras by projecting objects to one or more selected planes
- The more planes considered to receive shadows, the less practical this becomes
Computation of Penumbras

- Two approaches
- Analytical determination: trace surfaces forming umbra/penumbra boundaries into the scene
- Sampling: approximation of shadows, source is (usually) treated as a collection of point sources
Extremal shadow boundaries

- Analytical determination of bounding surfaces of shadow
Extremal shadow boundaries (2)

- Nishita and Nakamae (1983):
- Compute extremal boundaries in object space
- Transform boundaries to image space and use in illumination
- Determine visible parts of lightsource from penumbra points
Illumination calculation

- Intensity at $p$ is approximated by

$$I_p = \frac{I_s}{2} \sum_{\nu=1}^{n} \theta_\nu \cos(\phi_\nu)$$
Aspect Graphs

- Aspect graph theory from Computer Vision can aid in determining the discontinuities in the illumination of surfaces
- Discontinuity surfaces bound regions with continuous illumination functions
Critical Surfaces

(a) EV surface

(b) EEE surface
Discontinuity Edges

(a) $D^2$ discontinuity edges

(b) $D^1$ discontinuity edge
Discontinuity Meshes

- Lishinski et al. (1992):
  - Trace discontinuity curves of discontinuity surfaces on receiver polygons
  - Construct meshes on the receivers
  - Calculate illumination on mesh vertices
  - Merge meshes (when multiple light sources)
Discontinuity Meshes (2)
Sampling

- Sampling is generally less accurate, but easier to implement
- Uses umbra computation techniques for different points on the area of the light source
- Results are averaged
- Hardware support makes this especially attractive
Conclusion

- Real-time shadows, although often yielding only an approximation of the results possible by ray-tracing/global illumination methods, adds significant realism.
- Umbra generally more easily calculated but less “real” than Penumbra.