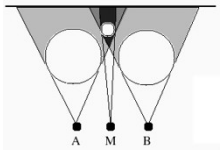
The "Occlusion Problem"

Timothy S. Milliron
CS 598d, Princeton University

What is the "Occlusion Problem?"

- Between two images, occlusion relationships might change unpredictably
  - Background might be exposed
  - Holes in surfaces might result.



- IBR needs more information to avoid artifacts
- Chen & Williams: Associate a depth-image with each RGB image.

Another Approach

- Use depth information to generate something more like a traditional mesh
- Perform a 3-D warp on the texture-mapped mesh (which looks just like the image).
- Traditional Z-buffering will (almost) solve the occlusion problem.

Mark, McMillan, & Bishop

- Application:
  - Increase frame-rates from 5 FPS to 60 FPS for expensive scenes
  - Obtain good remote display performance
- Approach:
  - Render a few frames traditionally ("reference" frames) per second, interpolate the rest.

Solving the Occlusion Problem

- Treat the image as a texture-mapped triangular mesh with depth encoded in the Z coordinate.
- To interpolate, perform a standard 3-D warp using triangulation and geometry
- Depth-buffering should suffice to solve occlusion and ensure no holes in surfaces
- But …

Rubber Sheets

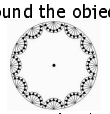
- At object edges, triangles can be perpendicular to image plane, between foreground and background.
  - Cause problems for occlusion
- Solution: Tag triangles as "connected:" connected triangles are part of a surface
  - Complicated calculation
- Okay, but …

Undersampling

- If a surface is oblique to the camera in one reference frame, and the derived frame shows the surface head-on, the surface is "undersampled"
  - Holes in the surface result
- Solution: "confidence" value for each pixel
  - Measures how well a given view can "see" a surface

View-Based Rendering

- Application:
  - Viewing an arbitrary real-world object in 3-D from a concentric sphere around the object.
    - Spherical lumigraph:



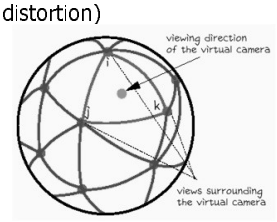
- Approach:
  - Use range-images to generate view-dependent triangular meshes to approximate the surface.
  - Texture-map the surface with the acquired image
  - Composite multiple views to form novel viewpoints and solve occlusion.

"Soft" Z-Buffering

- To composite the meshes:
  - If the z-values are not within some threshold, use the closest
  - If they are, blend them, using three weights

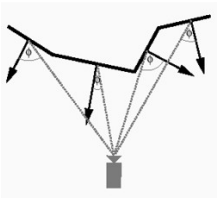
Weights (1)

- First Weight: "Distance" from reference image (like Gouraud triangle interpolation) (corrects for image distortion)



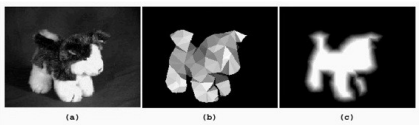
Weights (2)

- Obliqueness to camera (corrects for holes)
  - Prefer "head-on" images to "oblique" images



Weights (3)

- Distance from mesh boundary (less confident at the boundary) (corrects for polygonal error)



(a) A color image of a toy dog. (b) Weight w_ϕ is applied to each face of the triangular mesh. (c) Weight w_γ smoothly decreases towards the mesh boundary.

Summary

- All IBR technologies that incorporate translation must solve the occlusion problem.
- There are various approaches
  - Purely image-based (RGB and Z-buffer)
  - Partially geometry-reconstructing (meshes and 3-D warps)
- Special considerations
  - Using existing graphics hardware.