Handling complexity

- Level of detail variation
- Progressive refinement
- Handling very complex scenes
- Frameless rendering
Toward Uniform Frame Times

- Interactive/Real-time limits frame time
- Smooth motion etc. requires uniform frame time
- Polygon-bound systems can use multiple LODs
  - LOD = Level of Detail
- Flight simulator systems use LODs heavily
  - Mostly hand-modeled
  - Predictable scene complexity
- Cell-based visibility for indoor scenes (from ‘60s)
- Additional methods needed where complexity varies from frame to frame
Funkhouser and Sequin ‘93

- Visibility culling (view frustum and cells)
- Static level-of-detail thresholds (distance, size)
  - Blend between LODs to avoid pops (chair in paper)
- Dynamic LOD using previous frame time - reactive
- Predictive frame time cost heuristic
  - Based on primitive count and pixel estimate
  - Measurements give rough values (chair example)
  - rendering style also variable
- Benefit heuristic
  - Object importance, position in frame
  - Motion, and LOD hysteresis
More Funkhouser and Sequin ‘93

- Path through 87k polygon database (in paper)
- Sudden changes in scene complexity at turns
- Static LOD - high frame time variance
- Feedback - muffs sudden complexity changes
- Optimization - does best (0.13 - 0.075s)
- Video from Siggraph ‘93
Automatic Object Simplification

• Merge polygons with small dihedral angles
• Join vertices when edge gets short
• Drop polygons when degenerate
• Drop vertices in middle of small shallow clusters
• Error criteria for surfaces
• Issues for spindly objects screen doors, etc.
• Combining parts, filling holes, etc.
Progressive Meshes

- Hoppe ‘96, ‘97
- Retain simplification schema
- Replay to build out from simplified object
- Smooth morph between levels (geomorph)
- Retain color variation using texture (Certain et al ‘96)
Adding view dependence

• Hoppe ‘97
  – View frustum determines refinement
  – Angle to viewer varies refinement

• Luebke and Erikson ‘97
  – Progressive refinement in large database
  – Space division polygon collapse scheme
Appearance-Preserving Simplification

- Cohen, Olano, Manocha ‘98
- Use error-bounded geometry simplification
- Add error-bounded texture distortion
- Use with bump-mapped texture (Pixelflow)
- What do you do when bumps on bump maps get too small?
For Very High Depth Complexity

- Hierarchical depth buffer (Greene ‘93)
  - Octree spatial subdivision
  - Pyramid multiresolution depth buffer
  - Temporal coherence by saving visible octree nodes
- Antialiasing, etc. with Coverage masks
  - Greene ‘96 ( -7:30 on ‘96 tape)
  - Adds “triage” coverage masks, to subpixels
  - 8x8 hierarchy, 512 to 64 to 8 to 1/8 (4096 grid)
Frameless rendering

- Bishop ‘94
  - Update pixels at random with latest info
- Shirley et al ‘99
  - Interactive ray tracing on 64-processor SGI
  - Frameless technique improves interaction
References


More References