Conservative Multi-focal Visibility

Greg Marsden

gmarsden@cs.stanford.edu

April 3, 2002
Introduction

- New method for conservatively computing visible geometry for a volume of viewpoints.

- Allows amortization of cost over multiple frames, running asynchronously to the graphics pipeline.

- Uses existing graphics hardware to accelerate visibility computation.

*In this method, occluded polygons are subjected to simplification using an error metric based on their ability to intercept visible rays, and not on usual geometric proximity measures*
Visible surface algorithms

- Efficient visible surface algorithms reduce load on graphics pipeline
  - Z Buffer algorithm
  - Depth sorting
  - View frustum culling
  - BSP tree (occluder fusion)

- Determining what objects are occluded by a set of disconnected polygons for a single viewpoint is a computationally hard problem.
Volume visibility computation

- Notice that many viewpoints have high **spatial** and **temporal** locality: i.e. many objects persist from one scene into the next.
  - Scene voxelization (imprecise)
  - View shafts
  - Cells and portals
Viewpoint correspondence

Want to take the intuition of volume visibility (locality based optimization) and make it into a technique.

Define Viewpoint perspectivity as the coherence between unique viewpoints in viewing volume $V$.

Fix a projection plane $\Pi$ in space. The set of all rays originating from $V$ and passing through $\Pi$ at a given point $\pi$ define a vector bundle. The collection of these bundles defines the interaction between $V$ and $\Pi$.

Up to this point, similar to other techniques.
Correspondence

A sample "vector bundle."
Creating a multifocal $\approx$-buffer

Because we want to get an upper bound for the distance between the occluder and point $\pi$, the occlusion information can be conservatively stored by saving the shortest distance along a vector originating in $V$ and passing through $\pi$.

For the purposes of this exploration, we can get away with using the euclidean distance between occluding simplex $R$ and $\pi$ as a conservative estimate of this value (it is an interesting and untackled problem to determine the shortest distance from $R$ to $\pi$ passing through volume $V$).
Using the multifocal $\mathcal{Z}$-buffer

To use the multifocal $\mathcal{Z}$-buffer and ensure that the technique is conservative, calculate the maximum distance between occludee $E$ and $\pi$ along any ray in the bundle belonging to $\pi$.

Because this is a conservative test, we can get away with using the eight corners of $V$, and use the conventional $\mathcal{Z}$-buffer for the computation.

In practice this can be further accelerated by testing cells of occlusion hierarchies (octrees/etc) instead of actual polygons.
Sample output: voronoi maps

Gray levels indicate closest triangle feature.
Sample output: distance fields

Sample multifocal $z$-buffers for 1 and 2 triangles.
Future work

- Integration with graphics pipeline $z$-buffer
- Further geometric optimization (simplifying $V \rightarrow \Pi$ projection.
- Interaction with dynamic models
- Acceleration of complex rendering effects