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 visiblility computation

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

Viewpoint correspondence

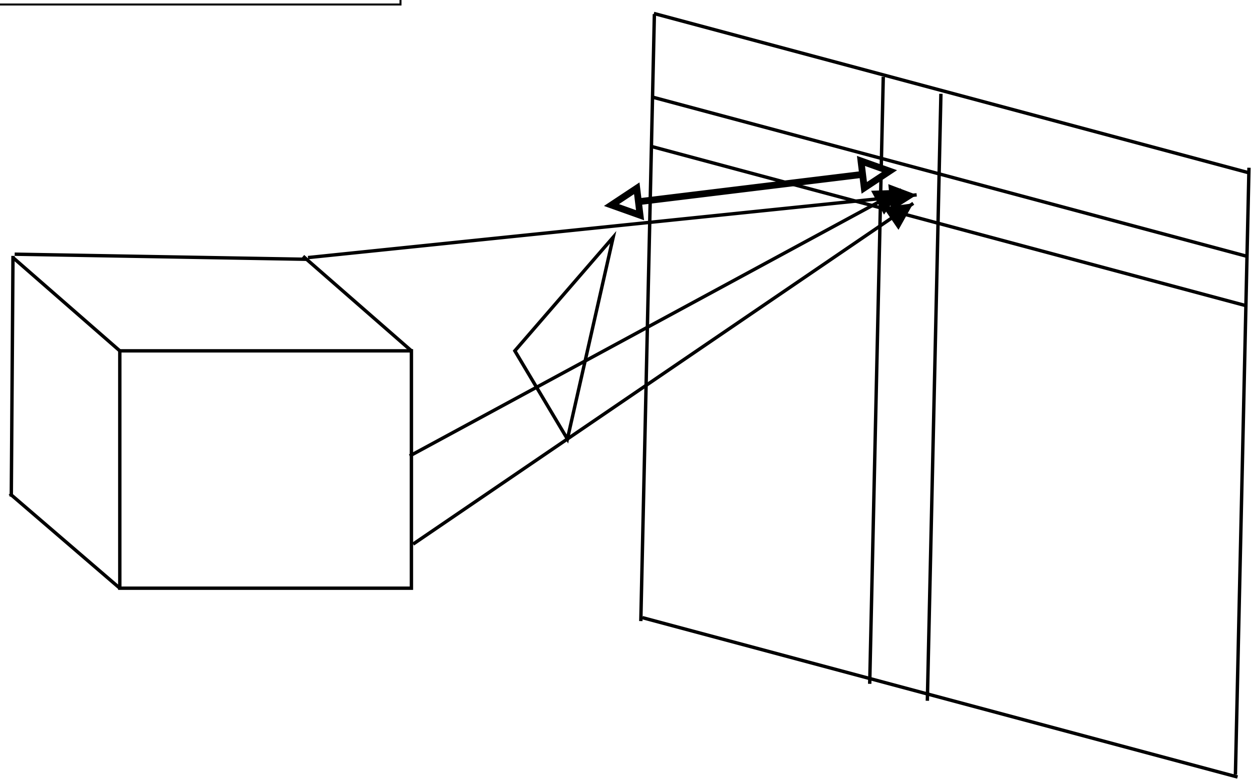
Want to take the intuition of volume visibility (locality based optimization) and make it into a technique.

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

Fix a projection plane Π in space. The set of all rays originating fom V and passing through Π at a given point π define a *vector bundle*. The collection of these bundles defines the interaction between V and Π.

Up to this point, similar to other techniques.

Correspondence



A sample “vector bundle.”

$\boxed{\text{Creating a multifocal } z\text{-buffer}}$

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

For the purposes of this exploration, we can get away with using the euclidean distance between occluding simplex R and π as a conservative estimate of this value (it is an interesting and untackled problem to determine the shortest distance from R to π passing through volume V)

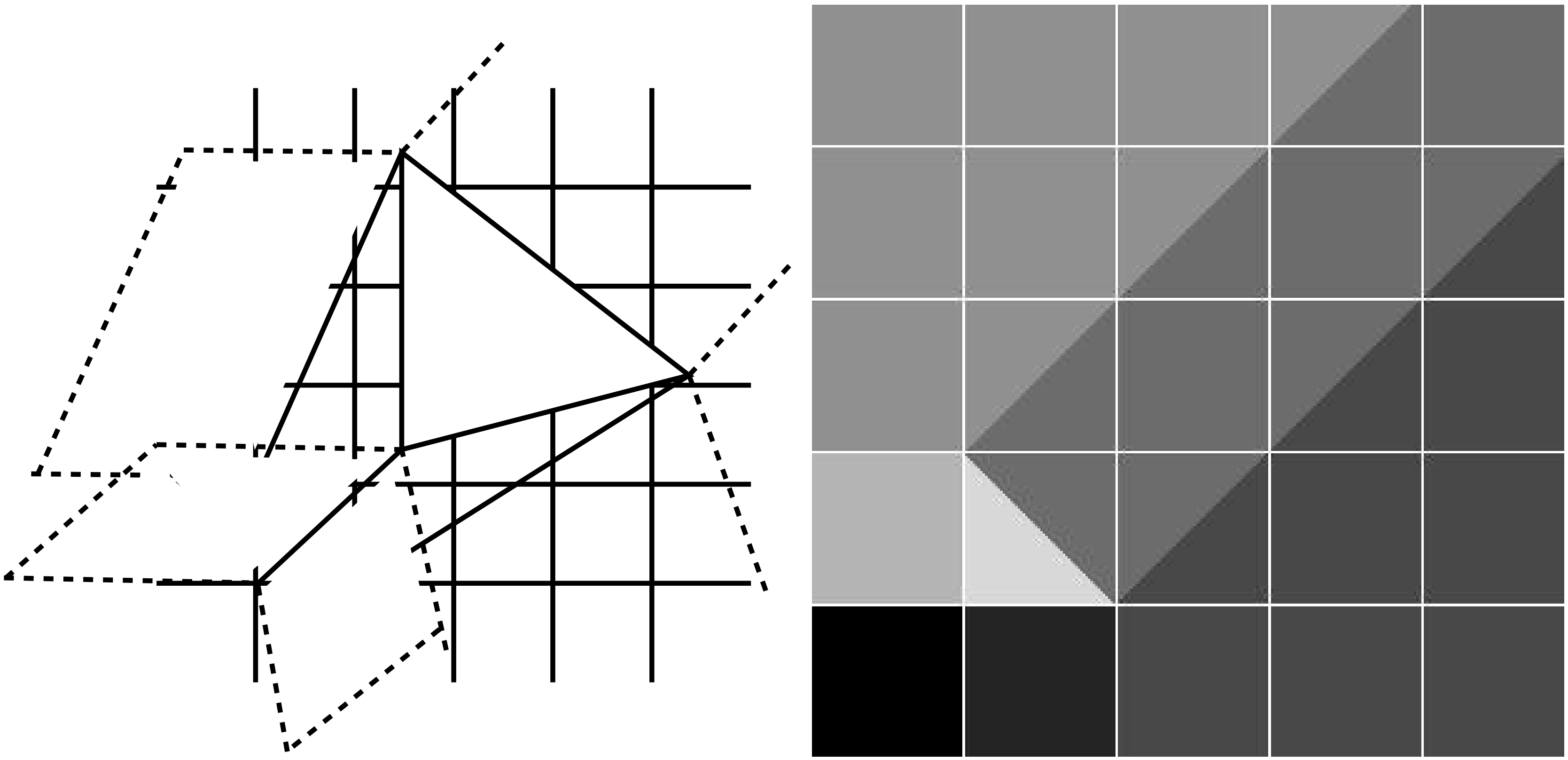
Using the multifocal z-buffer

To use the multifocal z-buffer and ensure that the technique is conservative, calculate the maximum distance between occludee E and π along any ray in the bundle belonging to π.

Because this is a conservative test, we can get away with using the eight corners of V, and use the conventional z-buffer for the computation.

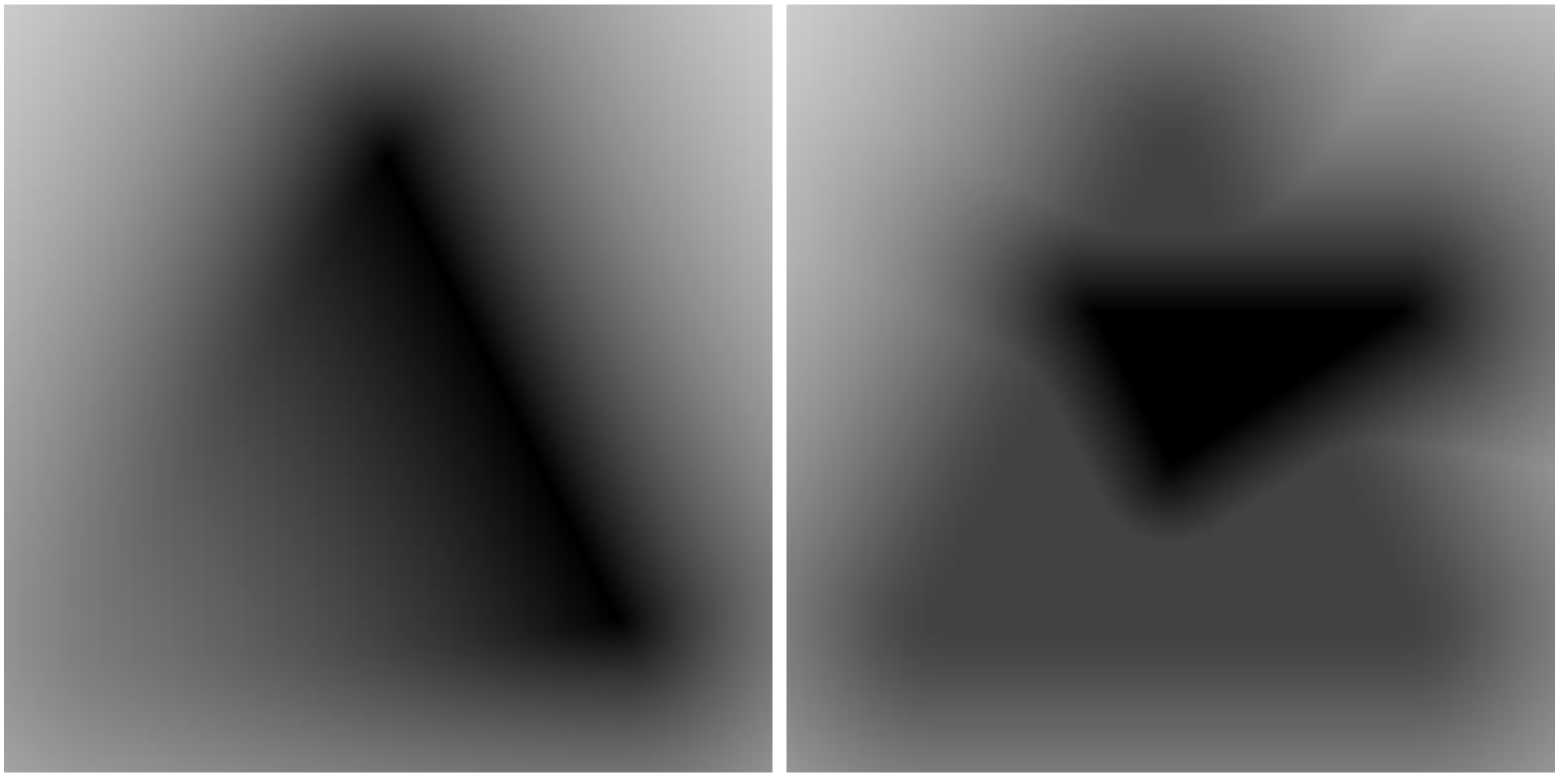
In practice this can be further accelerated by testing cells of occlusion hierarchies (octrees/etc) instead of actual polygons.

Sample output: voronoi maps



Gray levels indicate closest triangle feature.

Sample output: distance fields



Sample multifocal z-buffers for 1 and 2 triangles.

$\boxed{\textbf{Future work}}$

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