Portal Techniques for Sega's® Next-Generation Console

Jayeson Lee-Steere

GDC - March 18, 1999

NEC

www.powervr.com

VideoLogic

Topics to be covered

- **PowerVR2 Architecture**

  Very short overview of relevant hardware features

- **Part 1: Rendering Portal Special Effects**

  Specifics on how to render special effects such as mirrors, “magic” windows and TV screens

- **Part 2: Portals for Visibility Determination**

  How to use portals as an efficient visibility determination method

Overview of PowerVR2 Architecture

- Hardware renders an entire scene at once
  - Not one triangle at a time
- Final screen pixels are calculated before being written to frame buffer.
  - First, opaque polygons are processed.
    - Only front-most ones are rendered.
  - Second, translucent polygons are processed.
    - Visible ones are rendered.
    - Those occluded by opaque polygons are not.
- Result
  - Very efficient use of available fill rate
  - Overdraw is “free”

Part 1: Rendering Portal Special Effects

- A portal is a window looking into another world.
- Terminology
  - Viewer World: world viewer is in
  - Portal World: world visible through portal
  - Portal Polygon: polygon representing window
    - Can actually be multiple non-coplanar polygons
- For mirrors, Portal World is just a different view of Viewer

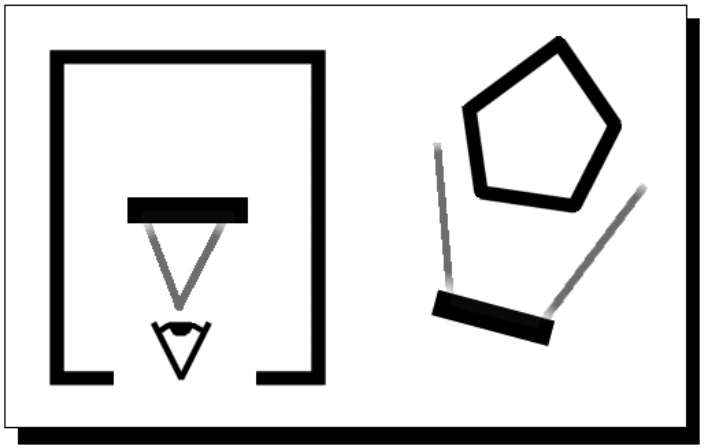

Figure 1. “Magic” Window

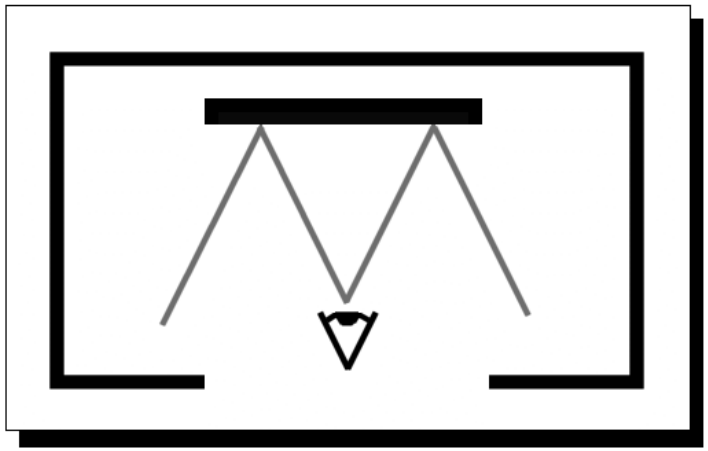

Figure 2. Mirror

Common Technique: Depth Masking

- **Steps**
  1. Render portal world. Do not render objects behind portal polygon(s)
  2. Clear depth buffer.
  3. Render portal polygon(s) fully or partially translucent.
  4. Render viewer world.

- **How It Works**
  - Portal polygon(s) set values in the depth buffer.
  - This prevents parts of viewer world behind portal polygon(s) from being drawn.

Common Technique: Depth Masking (cont.)

- But, on PowerVR2, polygons will be processed in incorrect order:
  1. Portal world opaque polygons
  2. Viewer world opaque polygons
  3. Portal world translucent polygons
  4. Depth clear (large translucent quad)
  5. Portal polygon(s)
  6. Viewer world translucent polygons

- Step 2 needs to come after step 5.
- We need to find a way to enforce rendering order.

PowerVR2 Technique: Multiple Passes

- Hardware can do multiple rendering passes.
- Each pass has the following:
  - Optional depth clear (free)
  - Opaque polygons
  - Translucent polygons
- Enforces rendering order:
  - In other words pass 2 is always processed after pass 1.
  - … but some overdraw is no longer free.
- Rendering technique becomes:
  1. Render portal world.
  2. Start new pass with depth clear enabled.
  3. Render portal polygon(s).
  4. Render viewer world.

PowerVR2 Technique: Render to Texture

- Render as 2 separate 3D scenes:
- Scene 1
  - Portal World only
  - Rendered to a texture, not frame buffer
- Scene 2
  - Render portal polygon(s).
    - Textured, using scene 1 as texture
  - Render viewer world.
- Technique useful for TV screens
  - TVs need perspective mapping of portal world.
- Also useful for holographic displays
  - Image of portal world needs to be translucent.

PowerVR2 Technique: “Simple” Portals

- A simple portal is where no part of the Viewer World is ever drawn behind the portal polygon(s).

Fig 3. Not Simple

Fig 4. Simple

- Dependence on rendering order can now be avoided:
  1. Render portal world
  2. Render portal polygon(s), if not completely translucent.
  3. Render viewer world.
- Steps 2 and 3 need to rendered in front of step 1
  - Achieved by scaling depth values
- Result is very efficient
  - Takes full advantage of “free” overdraw

Example: Scene from Unreal

- Sky box is rendered as simple portal.
- Reflective floor is rendered with multiple passes.

Part 2: Portals for Visibility Determination

- For visibility determination, portals are convex windows placed where mostly convex pieces of world geometry join.

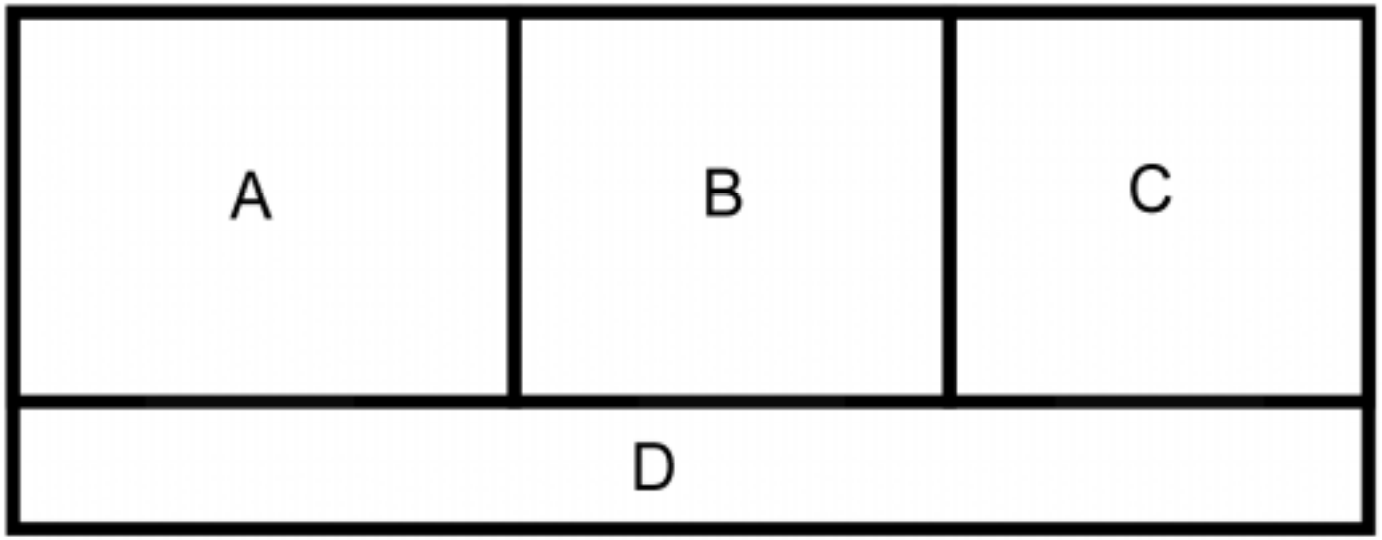

Fig 5. Top View of World With Portals

- Best suited for:
  - Indoor environments
  - Closed outdoor environments (valleys, canyons)
  - NOT for open outdoor environments (flightsims)

The Good and the Bad

Bad

- Overdraw can often be high.
  - Caused by concave geometry
- Floating-point calculations can be intensive.

Good

- PowerVR2 has “free” overdraw.
- Calculations are mostly inner products.
  - SH-4 CPU is super fast at these

* So the console is well suited for this technique! *

How it Works

- Calculate normalized bounding planes around portal to viewer.

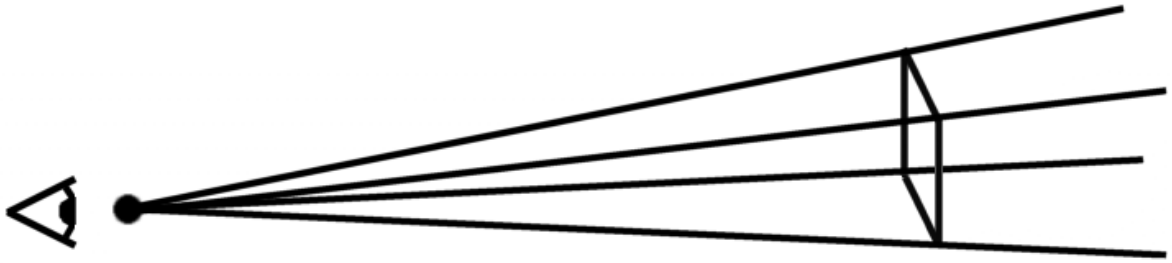

Fig 6. Portal Planes

- Planes should be in world coordinate system.
- Visible objects are those that intersect the volume created by the planes.

How it Works (cont.)

- Portals should be processed recursively, starting with those closest to the viewer.
- View volume can be handled like other portal volumes.
  - It should be processed first
  - Takes care of trivial screen clipping
- Each portal can be clipped against the last volume.

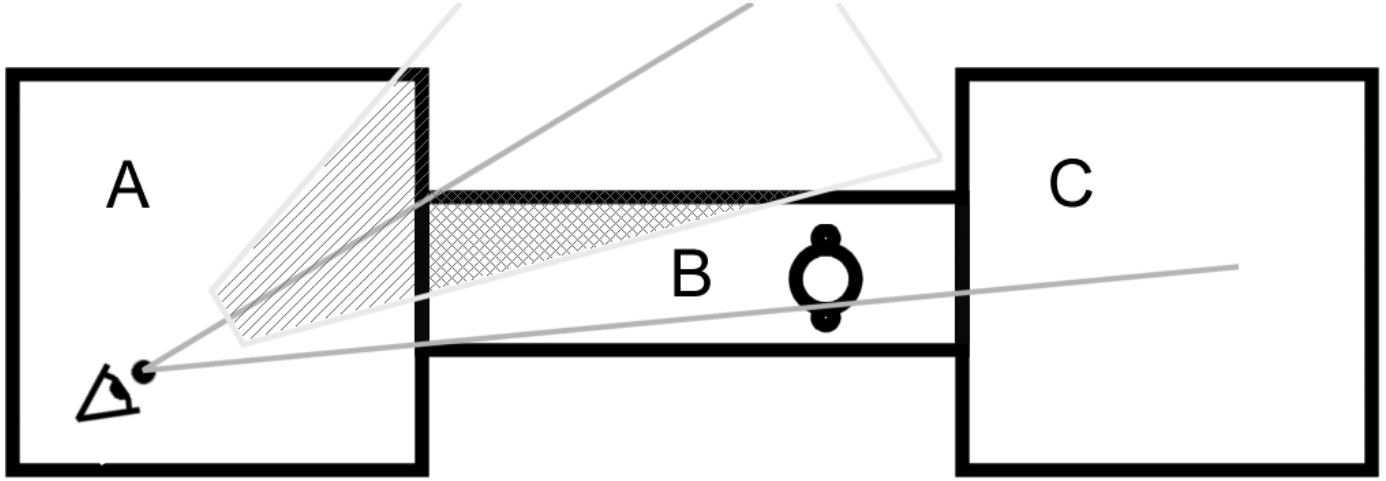

Fig 7. Processing Portals

Determining if a Polygon Intersects a Volume

- For each volume plane:
  - See if all vertices are on the “outside” of the plane.
  - If so, polygon is not visible.
  - If not (for all planes), treat polygon as visible.
- Math: Inner product between vertex and plane
  - Xv*Xp + Yv*Yp + Zv*Zp + 1*Wp
  - Yields distance of vertex from plane
  - +ve distance means “outside”
- Testing a quad against 5 planes will require 20 inner product calculations.
- Oring results from each calculation will yield clip flags.
  - So clip testing is done *before* transformation for little cost
- Only need to clip against front Z plane
  - Hardware will handle X/Y clipping.

Determining if a Bounding Sphere Intersects a Volume

- For each volume plane:
  - Calculate inner product between sphere center and plane
  - Compare distance of center with radius of sphere
  - if (Distance > Radius) sphere is outside volume
- Very simple and fast

Determining if an Axis-Aligned Bounding Box Intersects a Volume

- Not so simple
  - See Graphics Gems IV, Section 1.7 for a detailed description
- Math
  - Several simple comparisons
  - Two inner products per volume plane
  - Several simple 2D trivial clip test calculations
- … so still fast

References

- Geometry Formulas and Facts
  - http://www.geom.umn.edu/docs/reference/CRC-formulas/
- Graphics Gems IV
  - Edited by Paul S. Heckbert, AP Professional

Questions

POWERVR