SVGPU
Real Time 3D Rendering to Vector Graphics Formats

Apollo I. Ellis  University of Illinois (Presenting)
Warren Hunt  Oculus Research
John C. Hart  University of Illinois
SVGPU (Scalable Vector Graphics on the GPU)

- Renders vector images from 3D scenes, fast
- Applications in client server graphics domain
Hidden Surfaces

- Roberts 1963
  Tiled Binned

- Appel 1967
  Quantitative Invisibility

- Sutherland’s
  Taxonomy 1974

- Devai Optimal
  Hidden Line
  1986

- McKenna Optimal
  Hidden Surface
  1987

- Rasterization...

- Auzinger et al.
  Analytic Visibility
  on the GPU

- SVGPU
Pipeline

- Vertex shade and bin to screen tiles
- Hash edges and extract silhouettes
- Clip triangles to silhouette edges
- Check for occlusion
Silhouette Edge Extraction

- Hash all triangles by each edge
- Sweep the hash buckets
- Check collisions for front-back pairs
- Bin silhouette edges by screen tile
Clip Setup

- Dynamic parallelism parent kernel
- One thread per bin.. Say 64..
- Each thread runs a bin’s MxN clipping kernel
- Each thread runs a bin’s N’xN occlusion kernel
Trivial Rejection

- If AB lies outside 12, 23, or 31
  - Reject.
- If 1,2 and 3 lie outside AB
  - Reject.
- Gather all accepted pairs (AB,123)
- Construct adjacency list for clipper
Clipping

• For all triangles in adjacency
  • Sutherland-Hodgman [BF09]
  • Walk the vertices in turn
  • Classify vertices as In, Out, or On
  • 3 Vertices for ambiguous cases
  • LUT specifies behavior for each edge

Triangle A
  Edge 1  >  Edge 2
Triangle B
  Edge 3  >  Edge 4
Triangle C
  Edge 5  >  Edge 1  >  Edge 2  >  Edge 3

projection plane
window coords.
(including depth)
Clipping

- While(round < longest list)
  - Clip all triangles to next edge
  - Never reuse 4 or 5 for clipping
Clipping

- While(round < longest list)
  - Clip all triangles to next edge
  - Never reuse 4 or 5 for clipping
  - Consider polygon 1245

Triangle A
  - Edge 1
  - Edge 2

Triangle B
  - Edge 3
  - Edge 4

Triangle C
  - Edge 5
  - Edge 1
  - Edge 2
  - Edge 3

projection plane

window coords. (including depth)
Clipping

- While(round < longest list)
  - Clip all triangles to next edge
  - Never reuse 4 or 5 for clipping
  - Consider polygon 1245
  - Clipped by edge 2

Triangle A
- Edge 1
  - Edge 2

Triangle B
- Edge 3
  - Edge 4

Triangle C
- Edge 5
  - Edge 1
    - Edge 2
    - Edge 3

Projection plane

Window coords. (including depth)
Clipping

- While(round < longest list)
  - Clip all triangles to next edge
  - Never reuse 4 or 5 for clipping
  - Consider polygon 1245
  - Clipped by edge 2
  - Must use original edge 23 not 24
Clipping

- While(round < longest list)
  - Clip all triangles to next edge
  - Never reuse 4 or 5 for clipping
  - Consider polygon 1245
  - Clipped by edge 2
  - Must use original edge 23 not 24
  - LUT diverges here
Occlusion

- Triangles now fully occluded *or* fully visible
- One point occluded? Every point occluded.
- Centroid provides least ambiguity
Rasterized Planar Maps
Rasterized Planar Maps
Perf Scaling: Triangle Count

- Bunny: 65 triangles
- Armadillo: 20 triangles
- Dragon: 10 triangles
- Buddha: 5 triangles
Perf Scaling: Silhouettes per Bin

![Graph showing Ave. Silhouettes/Bin and FPS over different bins.]

- **Ave. Silhouettes/Bin**
- **FPS**
## Phase Breakdown

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sil. Hash</td>
<td>1.2</td>
<td>3.8</td>
<td>24</td>
<td>35</td>
<td>3</td>
<td>66</td>
<td>.4</td>
<td>.19</td>
</tr>
<tr>
<td>Sil. Clip</td>
<td>12</td>
<td>30</td>
<td>42</td>
<td>64</td>
<td>175</td>
<td>249</td>
<td>177</td>
<td>22</td>
</tr>
<tr>
<td>Occlusion</td>
<td>2.3</td>
<td>18</td>
<td>38</td>
<td>78</td>
<td>39</td>
<td>179</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>15.5</td>
<td>51.8</td>
<td>105</td>
<td>205</td>
<td>217</td>
<td>527</td>
<td>185.4</td>
<td>24.19</td>
</tr>
</tbody>
</table>
Motivation for Re-binning
Pathological Clipping Case

- Many silhouettes one poly
- Adjacency list very long
- After one clip many edges invalid
- Need to re check trivial reject
- Need to re check in parallel
Attribute Interpolation
Summary

• We can generate planar maps fast
• ~5X previous approaches
• We’ve evaluated binning and scaling considerations
• 1024 (32x32) bins performs best in most cases
• Highlighted pathological issues needing mitigation
• Robert’s based approach performs reasonably well
Future Work

- Precision issues need filtering
- Uniform binning precluding teapot in stadium
- Adaptive binning i.e. quad/kd trees are attractive
- In progress work on cloud gaming
- Theorizing about adaptive sampling and shading
- So called “Free” effects need validation and POC
References

- [Dév86] F Dévai, Quadratic bounds for hidden line elimination, Proceedings of the second annual symposium on Computational geometry, p.269-275, June 02-04, 1986, Yorktown Heights, New York, USA
Contacts

• aiellis2 at illinois dot edu
• warren dot hunt at gmail dot com
• jch at illinois dot edu
• http://graphics.cs.illinois.edu/