Rendering Fake Soft Shadows with Smoothies

Eric Chan Frédo Durand

Laboratory for Computer Science
Massachusetts Institute of Technology
Real-Time Shadows

Goals:
- Interactive framerates
- Hardware-accelerated
- Good image quality
- Dynamic environments

Applications:
- Game engines (e.g. Doom 3)
- Interactive walkthroughs

Challenge: balancing quality and performance
Two Algorithms from the 1970’s

Shadow volumes (Crow 1977)
- Object-space
- Accelerated by hardware stencil buffer
- Large fillrate consumption

Shadow maps (Williams 1978)
- Image-space
- Fast and simple
- Supported in hardware
- Undersampling artifacts
Soft Shadow Volumes

Penumbra wedges:
- Shadow polygons → wedges
- Compute penumbra with pixel shaders
- Accurate approximation

Papers:

But: much higher fillrate needed
Soft Shadow Maps

Ideas:
- Filtering
- Stochastic sampling
- Image warping

Examples:
- Percentage closer filtering (Reeves et al., SIGGRAPH 1987)
- Deep shadow maps (Lokovic and Veach, SIGGRAPH 2000)
- Image-based soft shadows (Agrawala et al., SIGGRAPH 2000)
- Multisampling hard shadows (Heckbert and Herf, TR 1997)

But: need dense sampling to minimize artifacts
Soft Shadow Maps (cont.)

Approximations

Examples:
- Convolution ([Soler and Sillion, SIGGRAPH 1998])
- Linear lights ([Heidrich et al., EGRW 2000])
- Outer surfaces ([Parker et al., TR 1998])
- Plateaus ([Haines, JGT 2001])
- Penumbra maps ([Wyman and Hansen, EGSR 2003])
Overview

- Extend basic shadow map approach
- Use extra primitives (smoothies) to soften shadows
Fake Soft Shadows

- Shadows not geometrically correct
- Shadows appear qualitatively like soft shadows
Contributions

Smoothie shadow algorithm:
- Creates soft shadow edges
- Hides aliasing artifacts
- Efficient (object / image space)
- Hardware-accelerated
- Supports dynamic scenes
1. Create Shadow Map

Render blockers into depth map
2. Identify Silhouette Edges

Find blockers’ silhouette edges in object space

object-space silhouettes

observer’s view

light’s view
3. Construct Smoothies

Blocker only:

- silhouette vertex
- silhouette edges
- blocker exterior
3. Construct Smoothies (cont.)

Blocker + smoothies:

- silhouette vertex
- silhouette edges
- smoothie edge
- smoothie corner
- blocker exterior
3. Construct Smoothies (cont.)

- **Smoothie edges** are rectangles in screen space with a fixed width.
- **Smoothie corners** connect adjacent smoothie edges.

![Diagram of geometry and shading](image-url)
4. Render Smoothies

Store depth and alpha values into **smoothie buffer**
5. Compute Shadows

Compute intensity using depth comparisons

Diagram:
- Light source
- Blocker
- Smoothie
- Receiver
5. Compute Shadows

Image sample behind blocker (intensity = 0)
5. Compute Shadows

Image sample behind smoothie (intensity = $\alpha$)
5. Compute Shadows

Image sample illuminated (intensity = 1)
Computing Alpha Values

Intuition:
- Alpha defines penumbra shape
- Should vary with ratio $b/r$
Computing Alpha Values (cont.)

1. Linearly interpolate alpha
2. Remap alpha at each pixel using ratio $b/r$:

$$\alpha' = \frac{\alpha}{1 - \frac{b}{r}}$$
Multiple Blockers and Receivers
Multiple Receivers

Smoothie buffer (linearly-interpolated $\alpha$)

same thickness

light’s view
Multiple Receivers (cont.)

Smoothie buffer (remapped $\alpha$)

different thickness

light’s view
Multiple Receivers (cont.)

Final image

different thickness

observer’s view
Multiple Blockers

What happens when smoothies overlap?
Multiple Blockers (cont.)

**Minimum blending:** just keep minimum of alpha values

smoothie  ray tracer
Comparison to Penumbra Maps

Penumbra maps (Wyman and Hansen, EGSR 2003)

- Same idea, different details

<table>
<thead>
<tr>
<th>Geometry:</th>
<th>Penumbra Maps</th>
<th>Smoothies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cones and sheets</td>
<td>quads</td>
</tr>
</tbody>
</table>

| Store depth:   | blockers only | blockers + smoothies |

Smoothie depth:

- Extra storage + comparison
- Handles surfaces that act only as receivers
Results

System information:
- 2.6 GHz Intel Pentium 4
- NVIDIA Geforce FX 5800 Ultra
Video

Ordinary Shadow Map

Triangles: 2324
Average FPS: 100.0
Hiding Aliasing (256 x 256)

shadow map

16 ms

bicubic filter

129 ms

smoothie (t = 0.02)

19 ms

smoothie (t = 0.08)

19 ms
Hiding Aliasing (1024 x 1024)

- 17 ms shadow map
- 142 ms bicubic filter
- 22 ms smoothie (t = 0.02)
- 24 ms smoothie (t = 0.08)
Comparison to Ray Tracer

increasing size of light source

smoothie  ray tracer
Video

original md2shader demo courtesy of Mark Kilgard
Discussion

Shadow maps:
- Assumes directional light or spotlight
- Discrete buffer samples

Shadow volumes:
- Assumes blockers are closed triangle meshes
- Silhouettes identified in object space

Smoothies:
- Rendered from light’s viewpoint
- Occupy small screen area ➔ inexpensive
Summary

Contribution:
- Simple extension to shadow maps
- Shadows edges are fake, but look like soft shadows
- Fast, maps well to graphics hardware
Trends in Real-Time Shadows

Architectures and algorithms go together

Currently, architectures → algorithms:
- Store per-pixel data at full precision

But also, algorithms → architectures:
- Shadow maps
- Shadow volume depth bounds
- Aggressive early z and stencil reject
Acknowledgments

Hardware, drivers, and bug fixes
- Mark Kilgard, Cass Everitt, David Kirk, Matt Papakipos (NVIDIA)
- Michael Doggett, Evan Hart, James Percy (ATI)

Writing and code
- Sylvain Lefebvre, George Drettakis, Janet Chen, Bill Mark
- Xavier Décoret, Henrik Wann Jensen

Funding
- ASEE NDSEG Fellowship