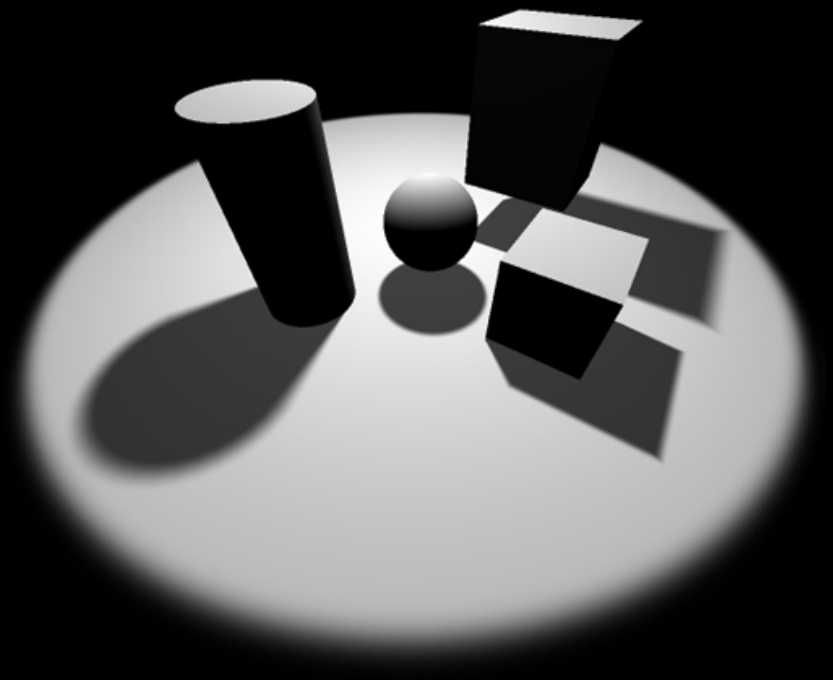
Rendering Fake Soft Shadows with Smoothies



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

NVIDIA

Applications:

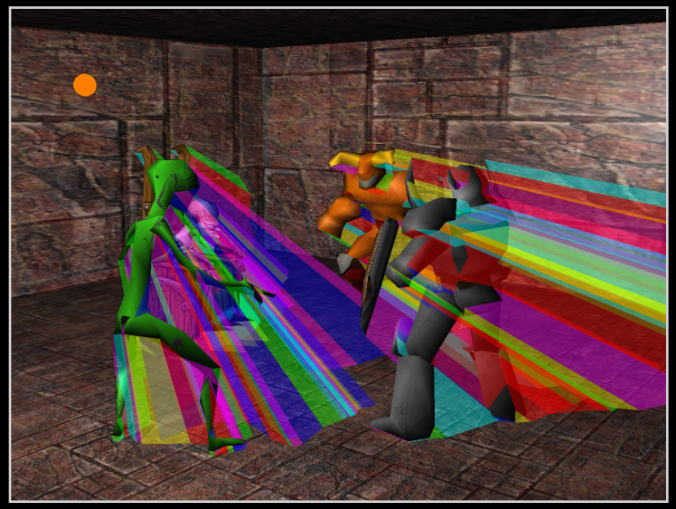
- Game engines (e.g. Doom 3)
- Interactive walkthroughs

Challenge: balancing quality and performance

Two Algorithms from the 1970's

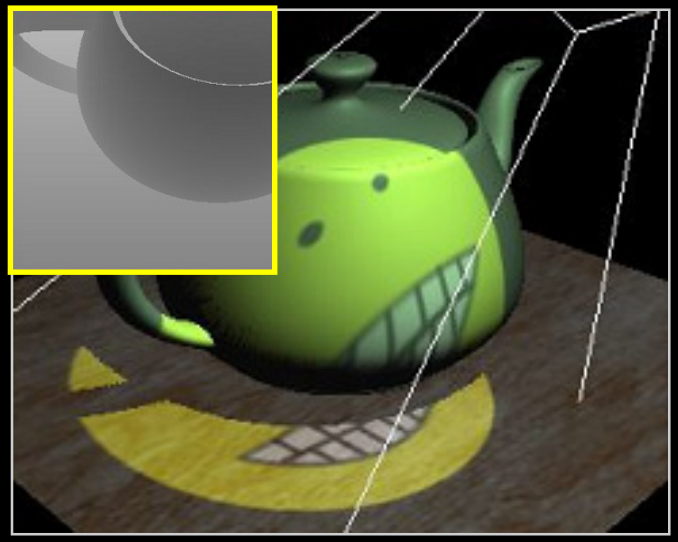
Shadow volumes (Crow 1977)

- Object-space
- Accelerated by hardware stencil buffer
- Large fillrate consumption



Shadow maps (Williams 1978)

- Image-space
- Fast and simple
- Supported in hardware
- Undersampling artifacts



NVIDIA

Soft Shadow Volumes

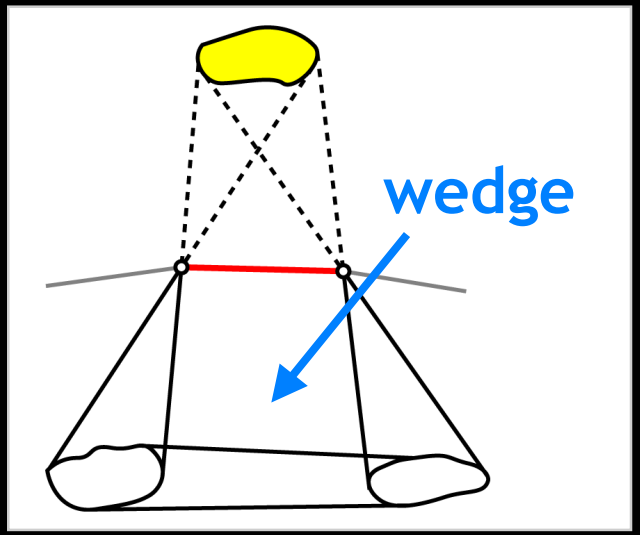
Penumbra wedges:

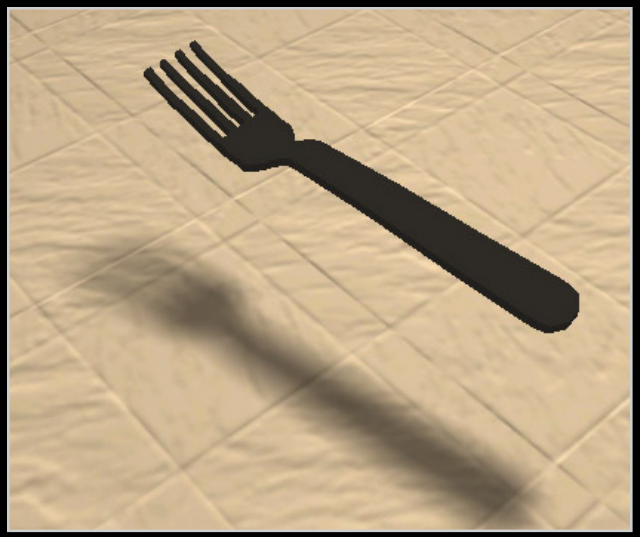
- Shadow polygons → wedges
- Compute penumbra with pixel shaders
- Accurate approximation

Papers:

- Assarsson et al. (EGRW 2002, SIGGRAPH 2003, HWWS 2003)

But: much higher fillrate needed



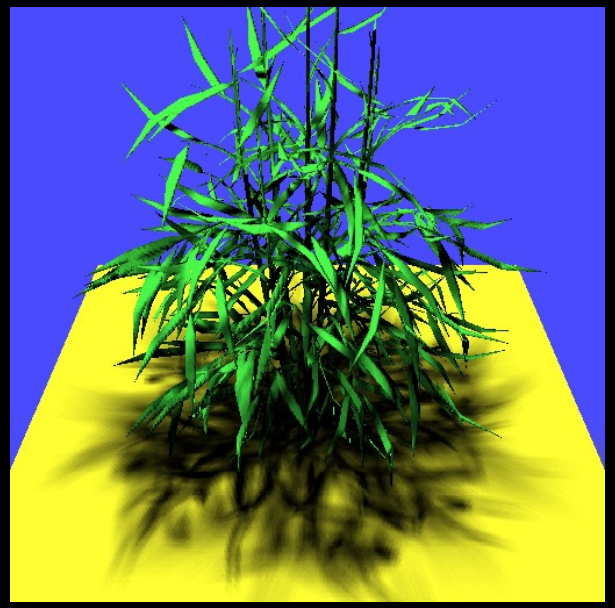


Assarsson and Akenine-Möller

Soft Shadow Maps

Ideas:

- Filtering
- Stochastic sampling
- Image warping



Agrawala et al.

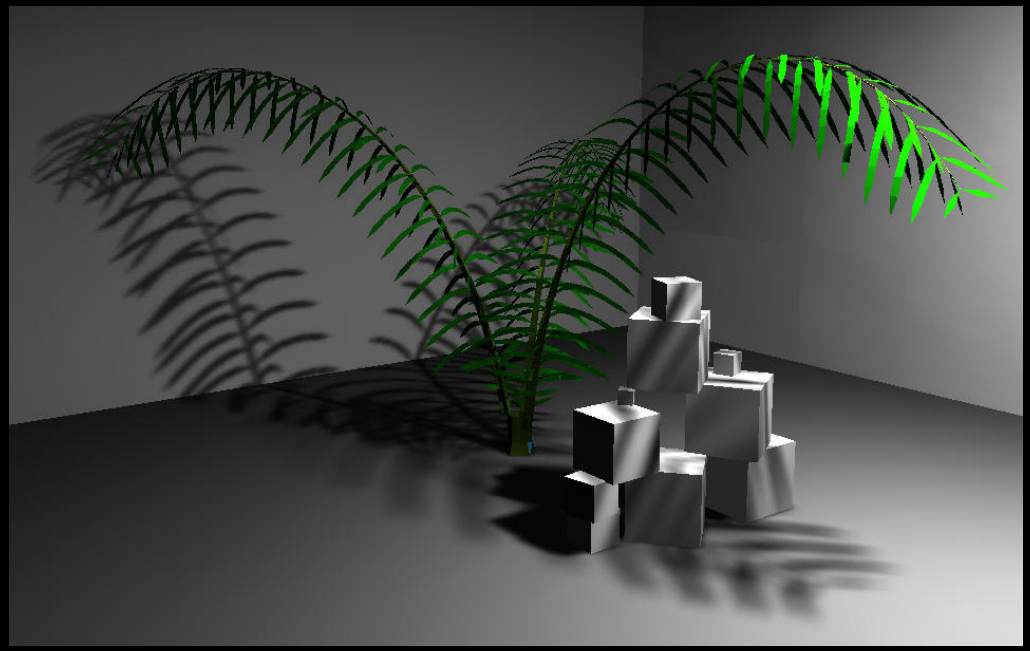
Examples:

- 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



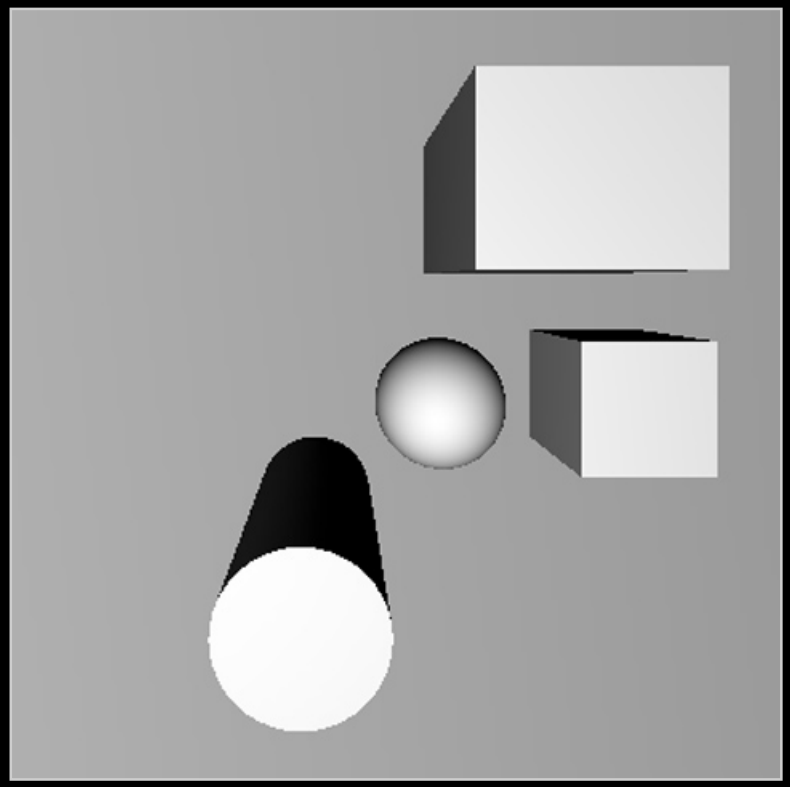
Soler and Sillion

Examples:

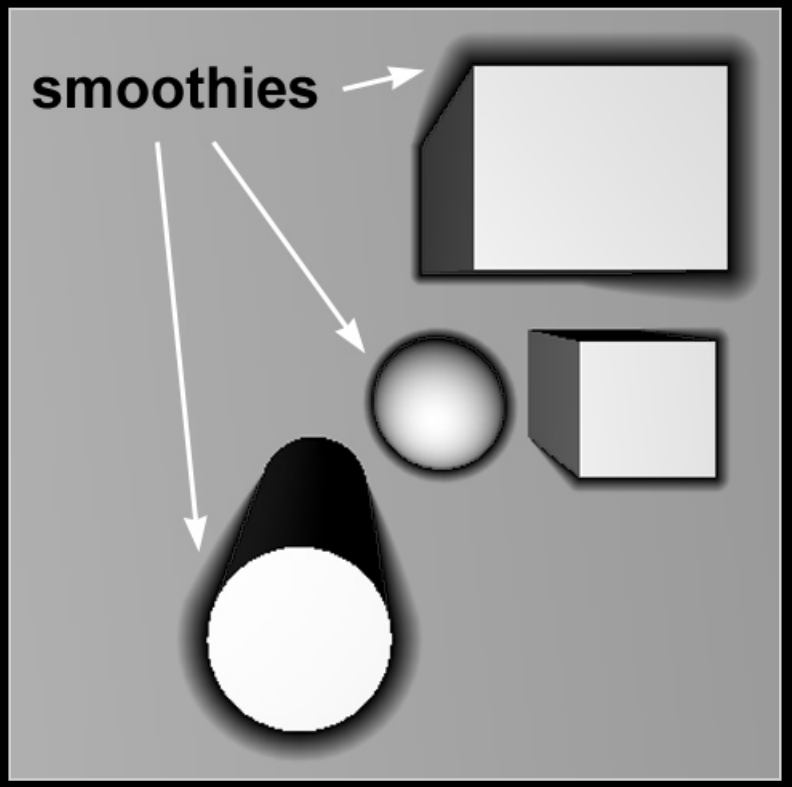
- Convolution (Soler and Sillion, SIGGRAPH 1998)
- Linear lights (Heidrich et al., EGRW 2000)
- Outer surfaces (Parker et al., TR 1998)
- Plateaus (Haines, JGT 2001)
- Penumbra maps (Wyman and Hansen, EGSR 2003)

Overview

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



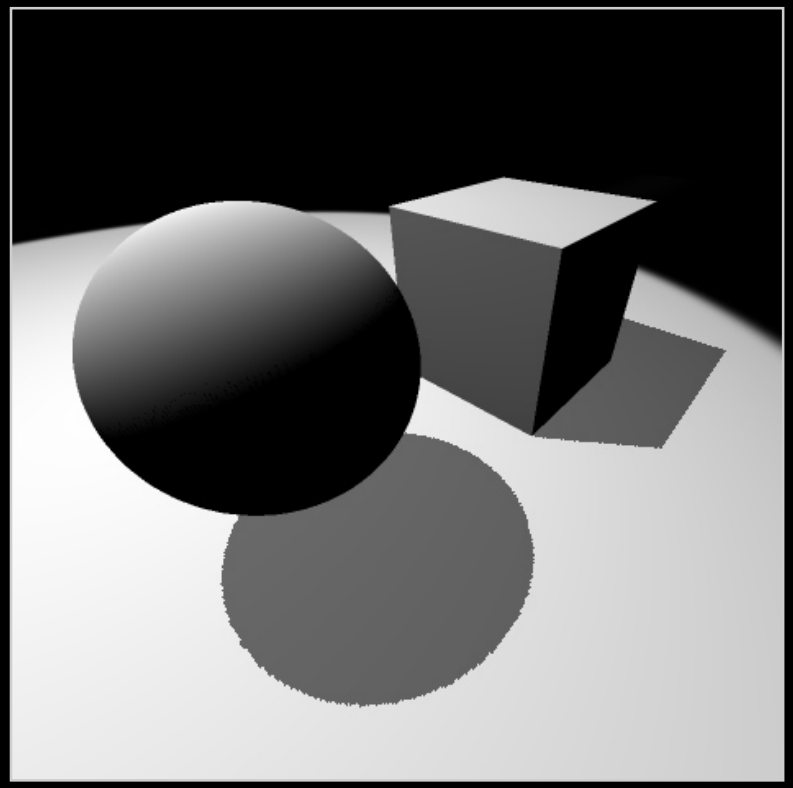
light's view (blockers only)



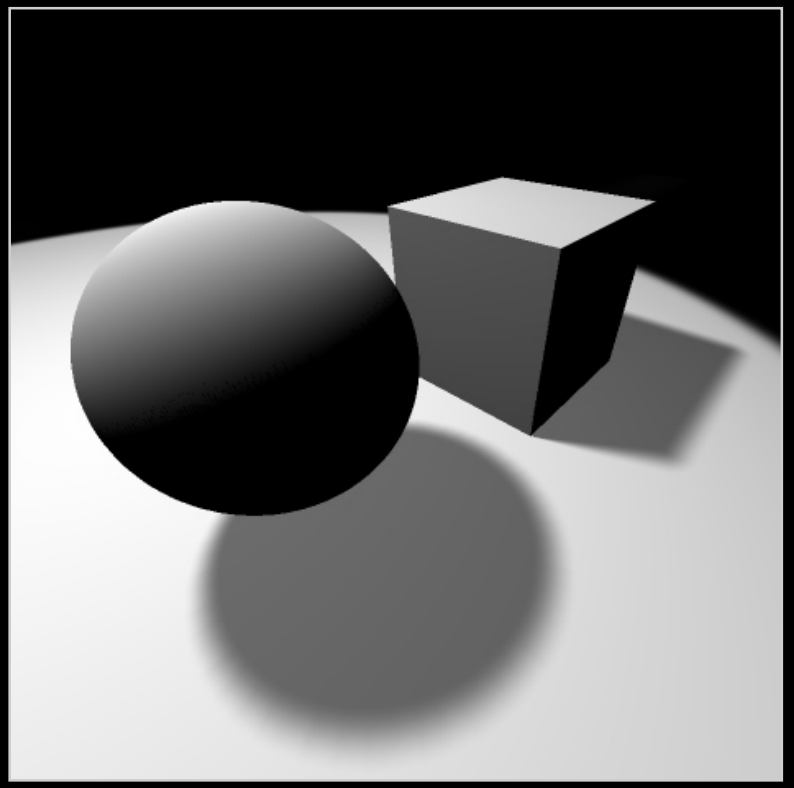
light's view (blockers + smoothies)

Fake Soft Shadows

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



Hard shadows

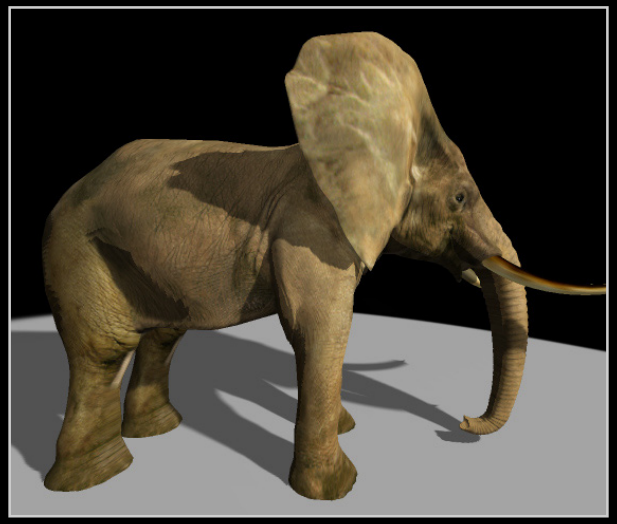


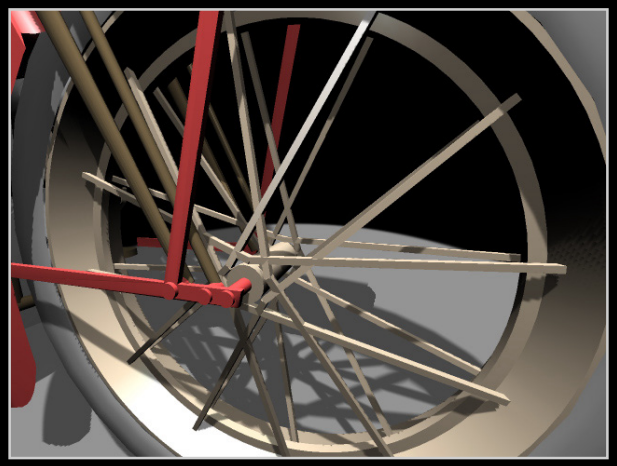
Fake soft shadows

Contributions

Smoothie shadow algorithm:

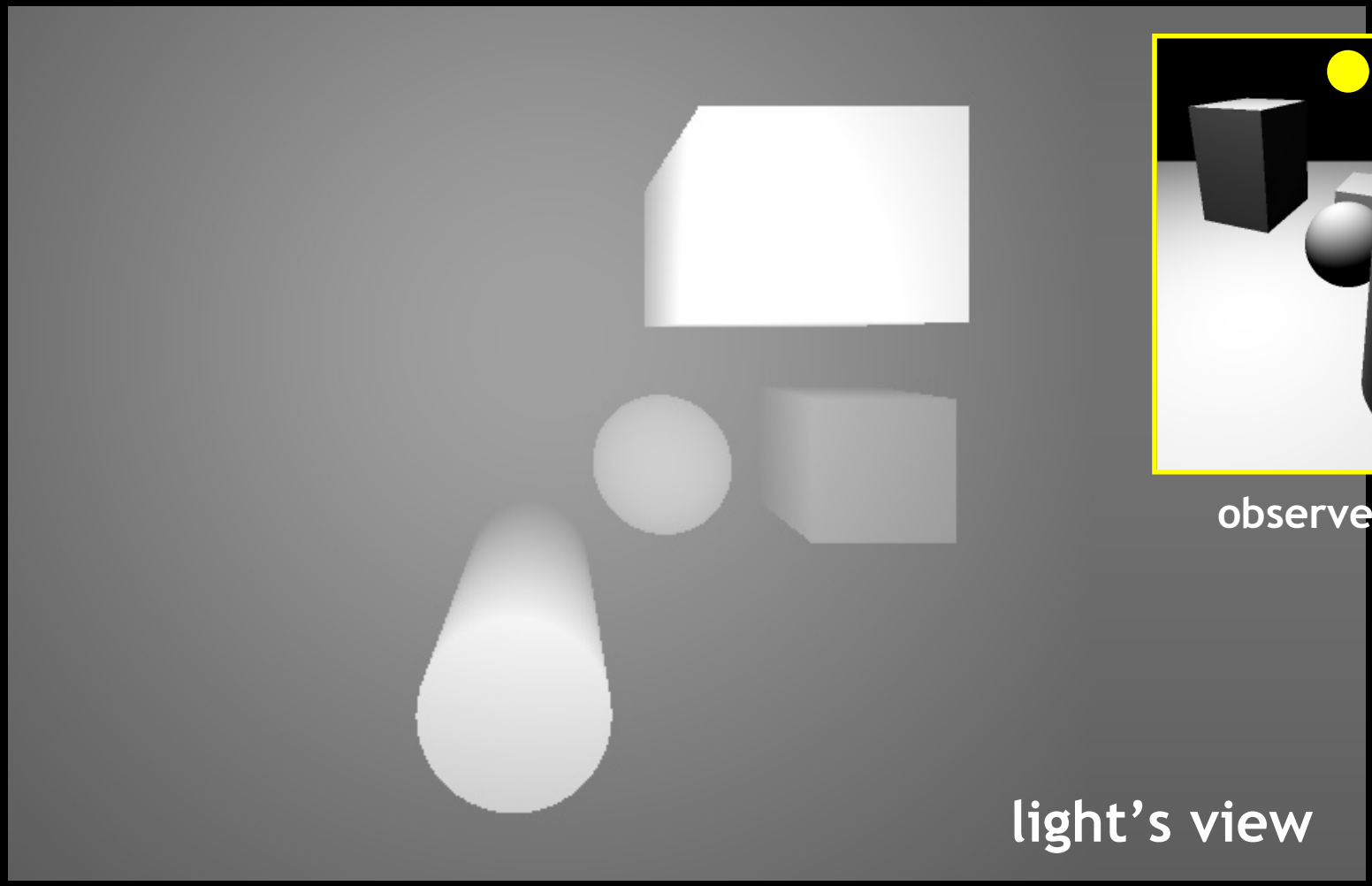
- Creates soft shadow edges
- Hides aliasing artifacts
- Efficient (object / image space)
- Hardware-accelerated
- Supports dynamic scenes

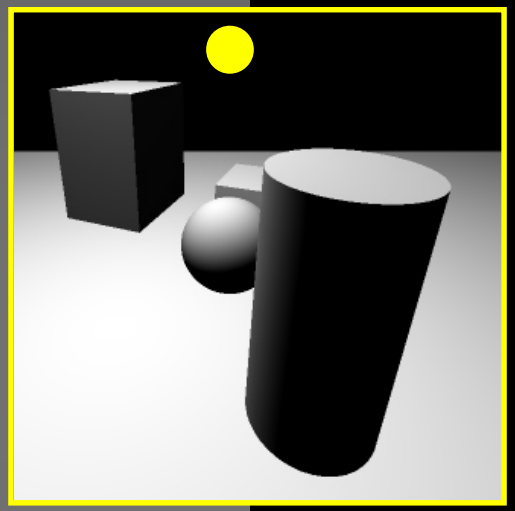




1. Create Shadow Map

Render blockers into depth map

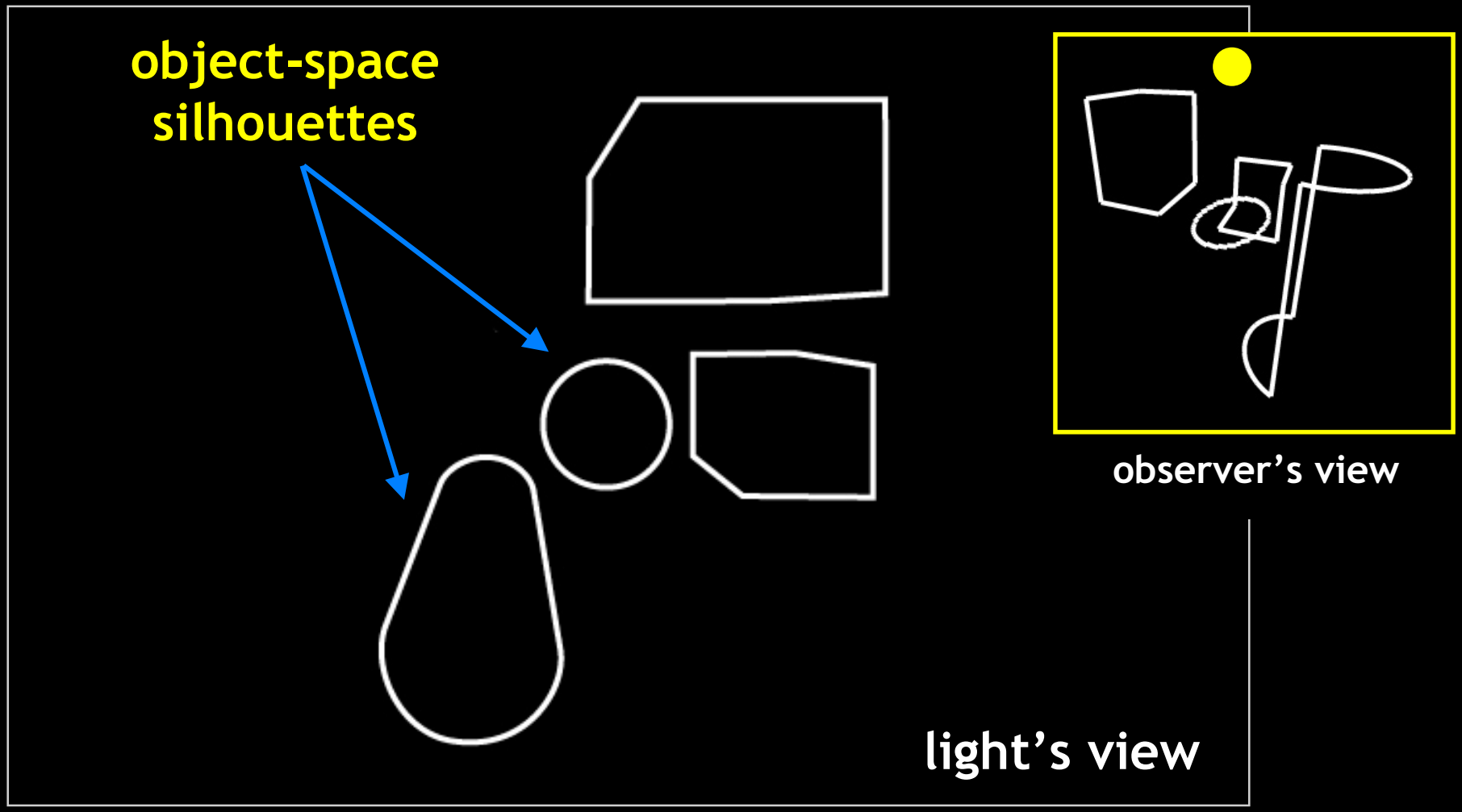




observer's view

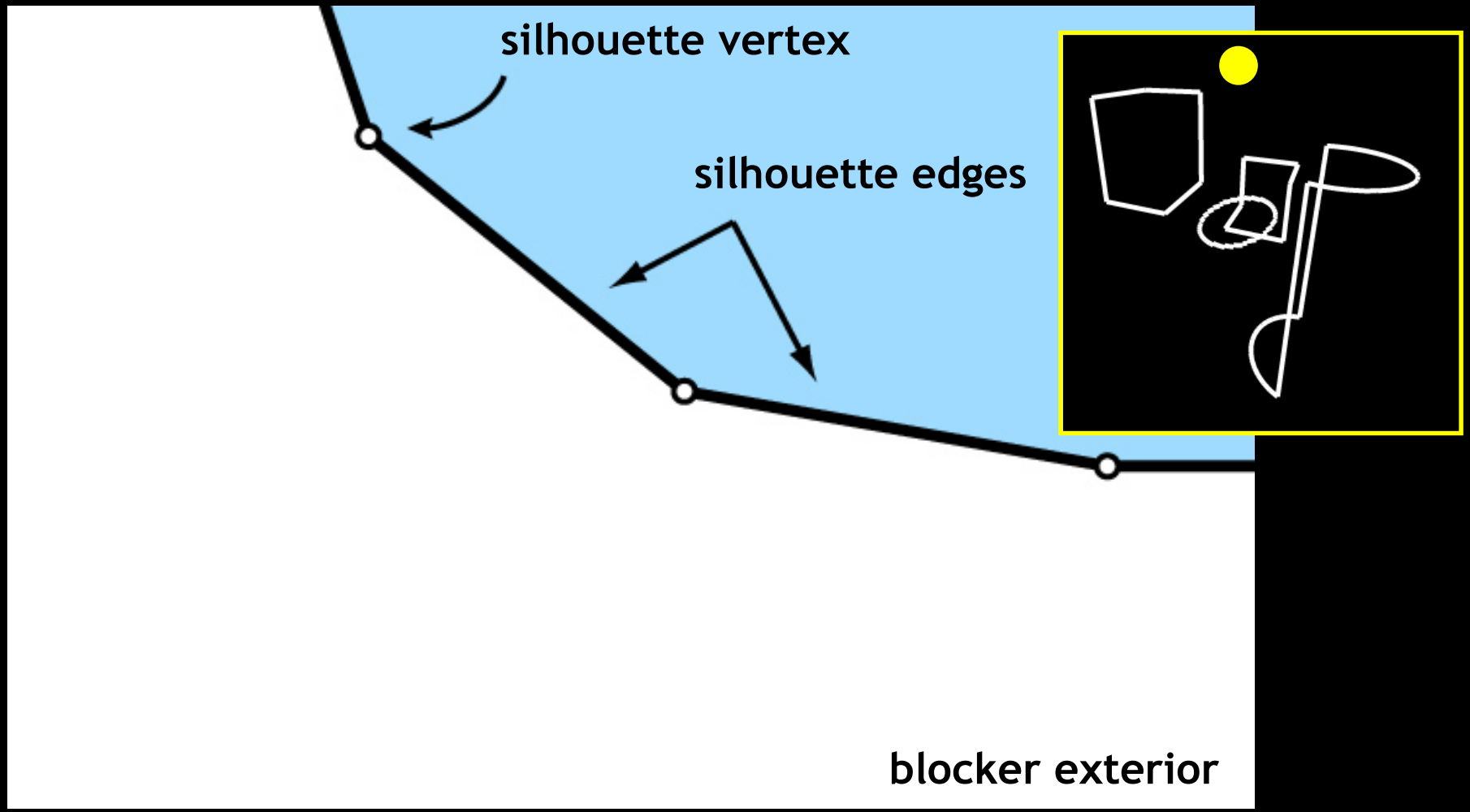
2. Identify Silhouette Edges

Find blockers' silhouette edges in object space



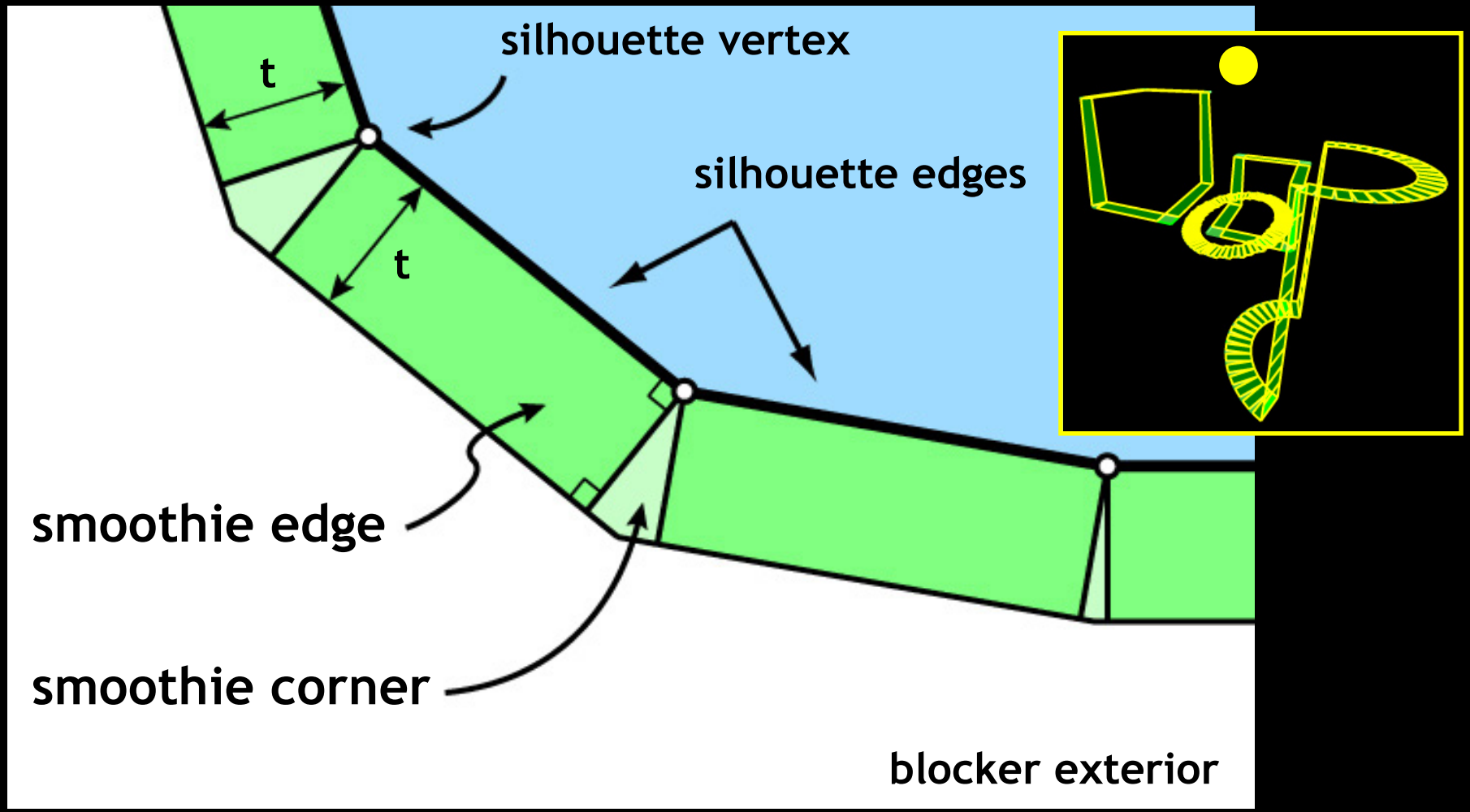
3. Construct Smoothies

Blocker only:



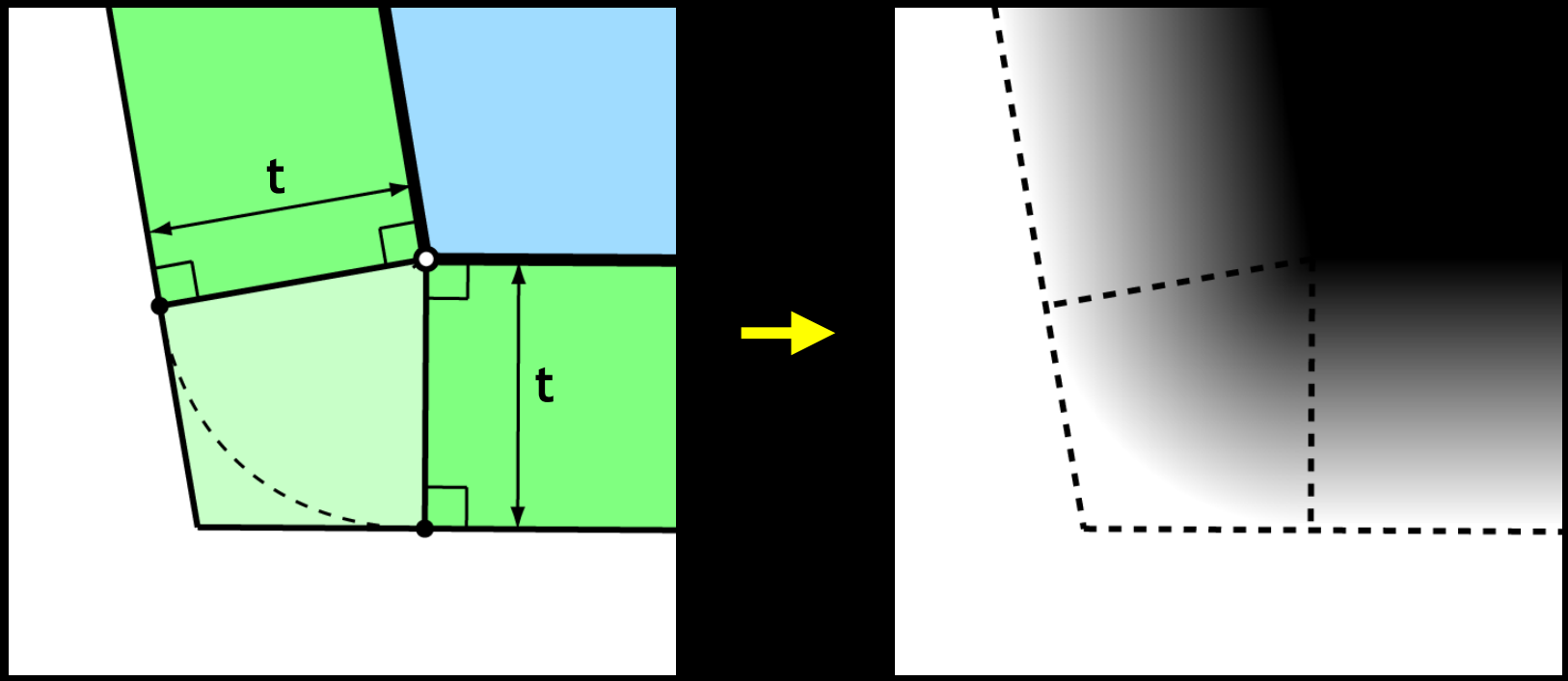
3. Construct Smoothies (cont.)

Blocker + smoothies:



3. Construct Smoothies (cont.)

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

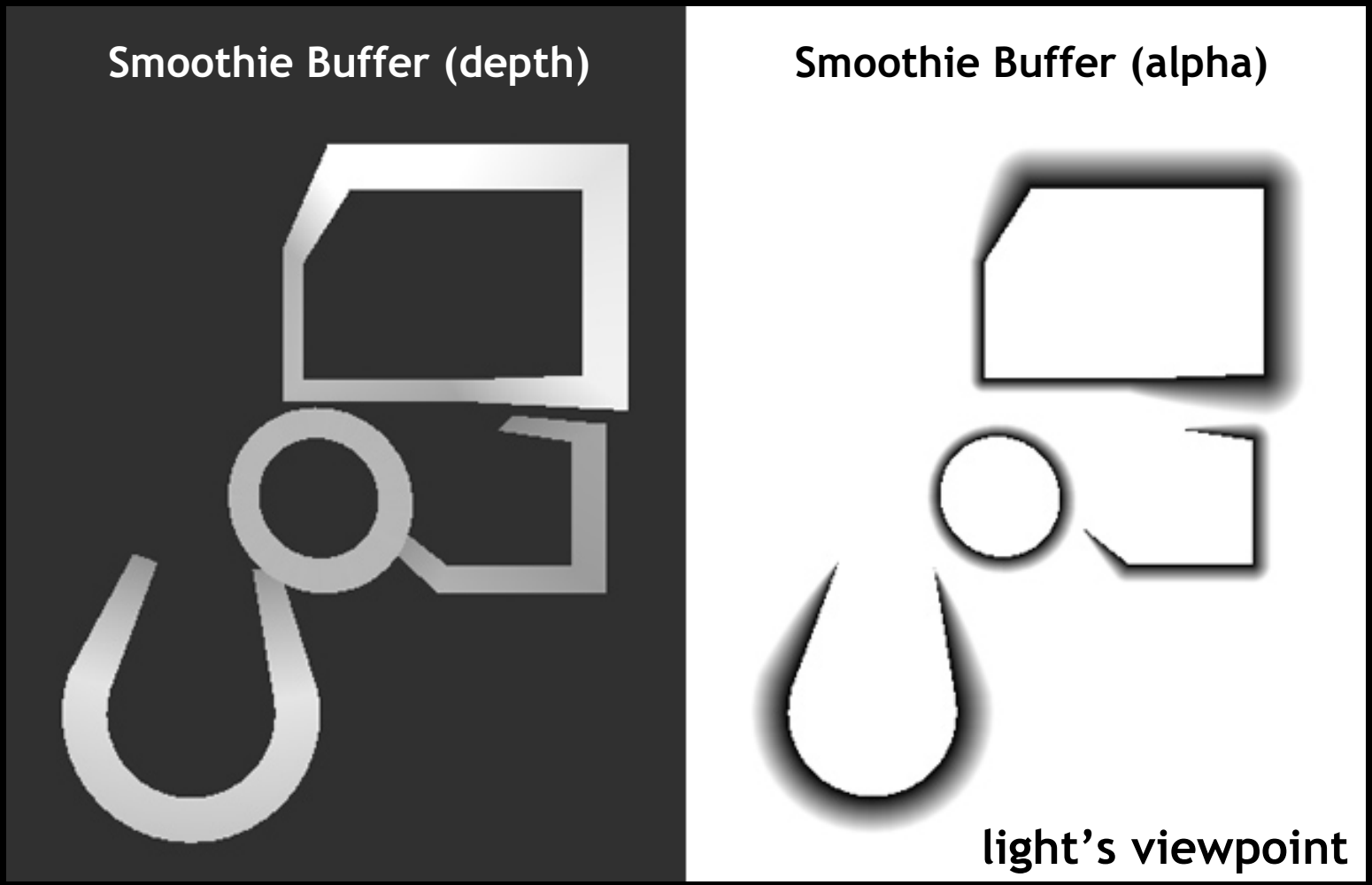


geometry

shading

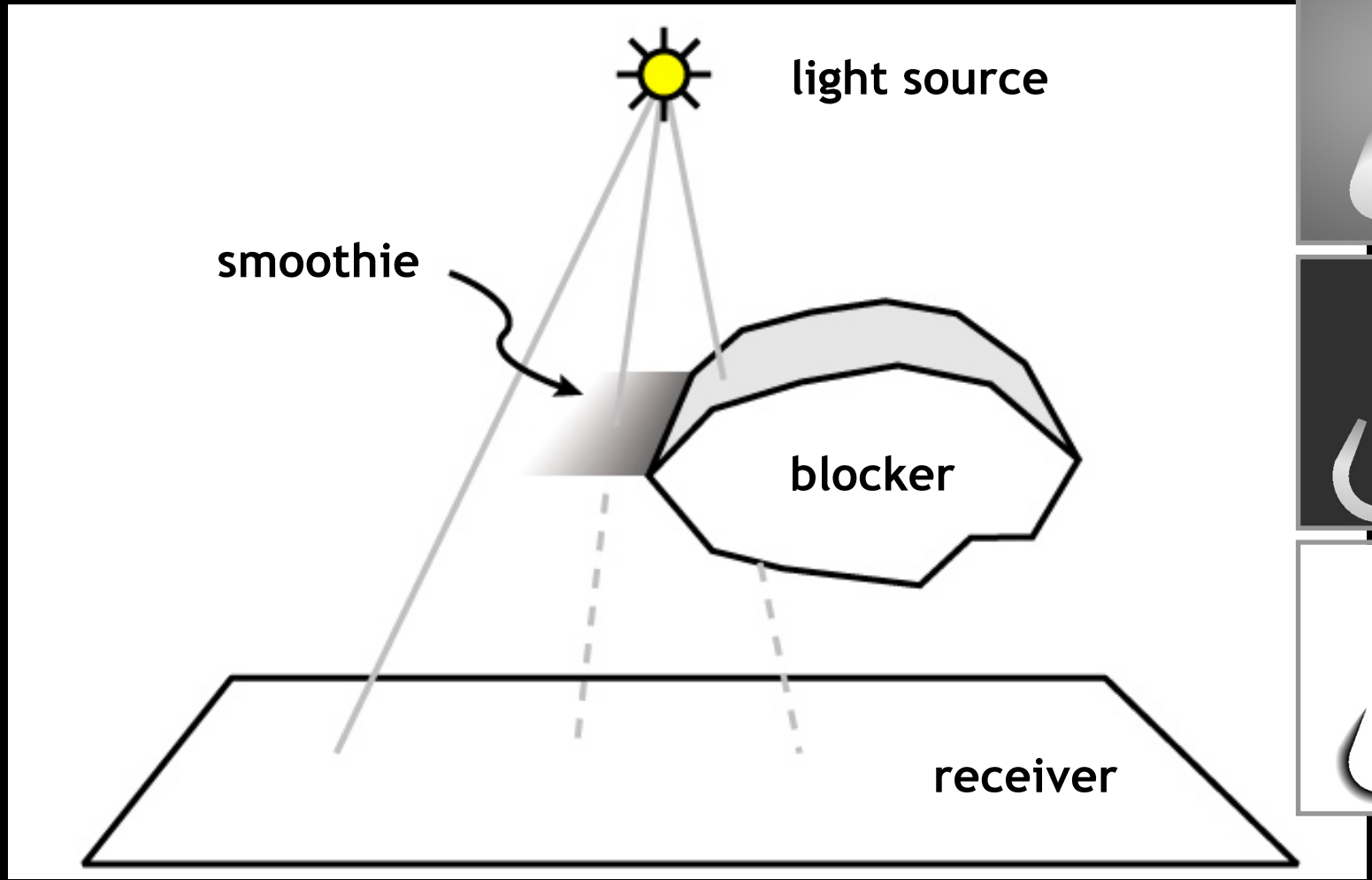
4. Render Smoothies

Store depth and alpha values into smoothie buffer



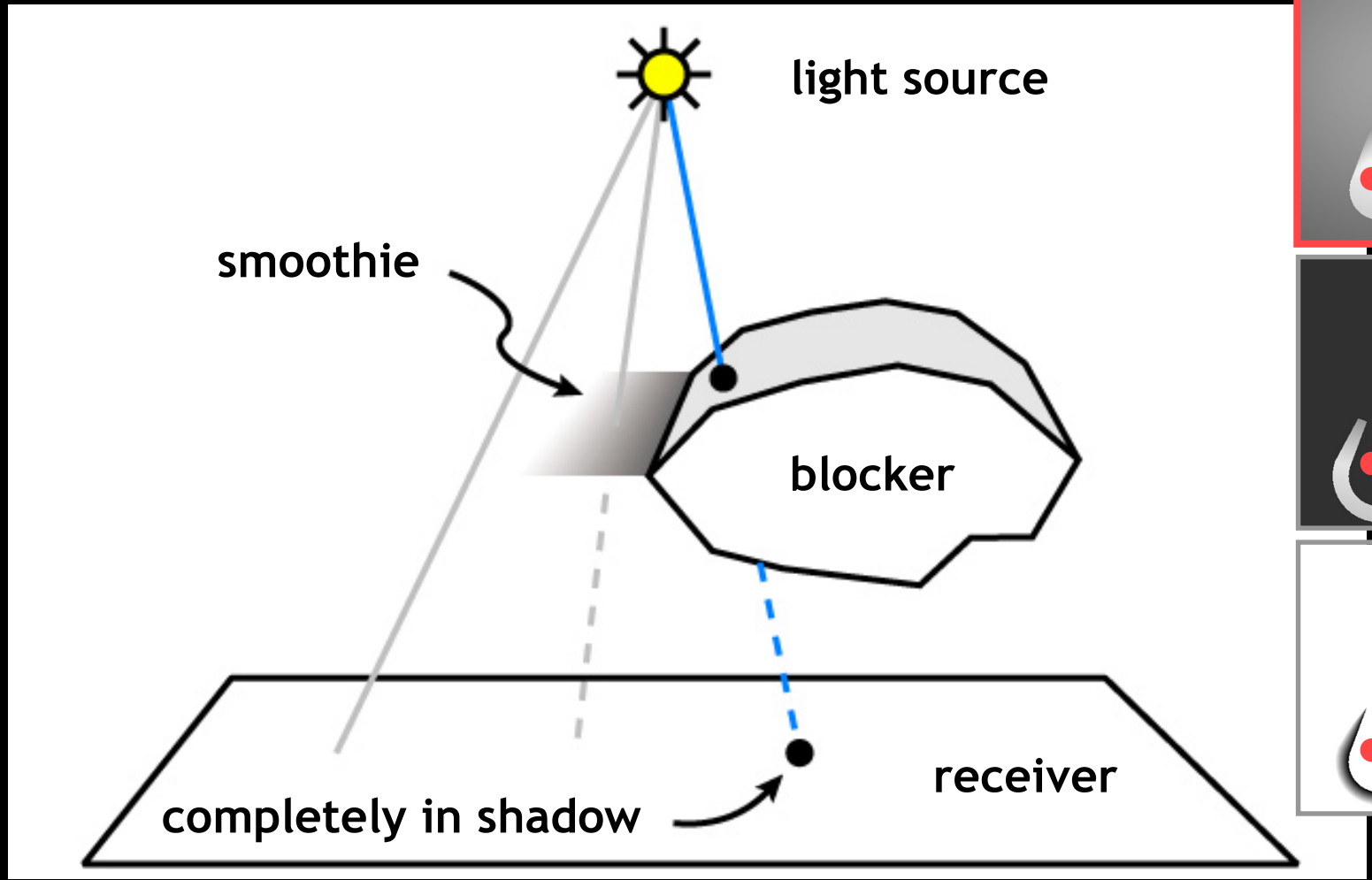
5. Compute Shadows

Compute intensity using depth comparisons



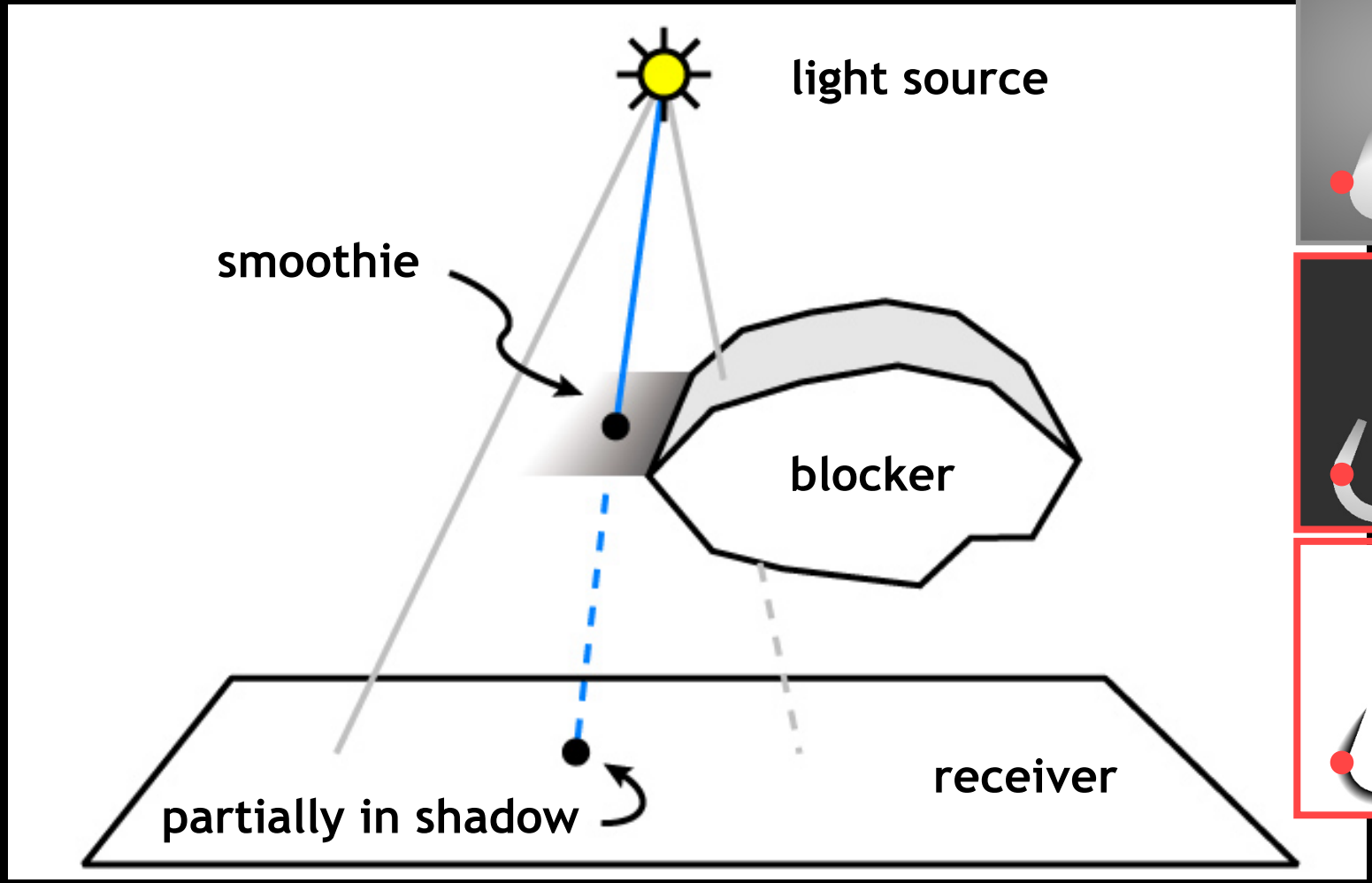
5. Compute Shadows

Image sample behind blocker (intensity = 0)



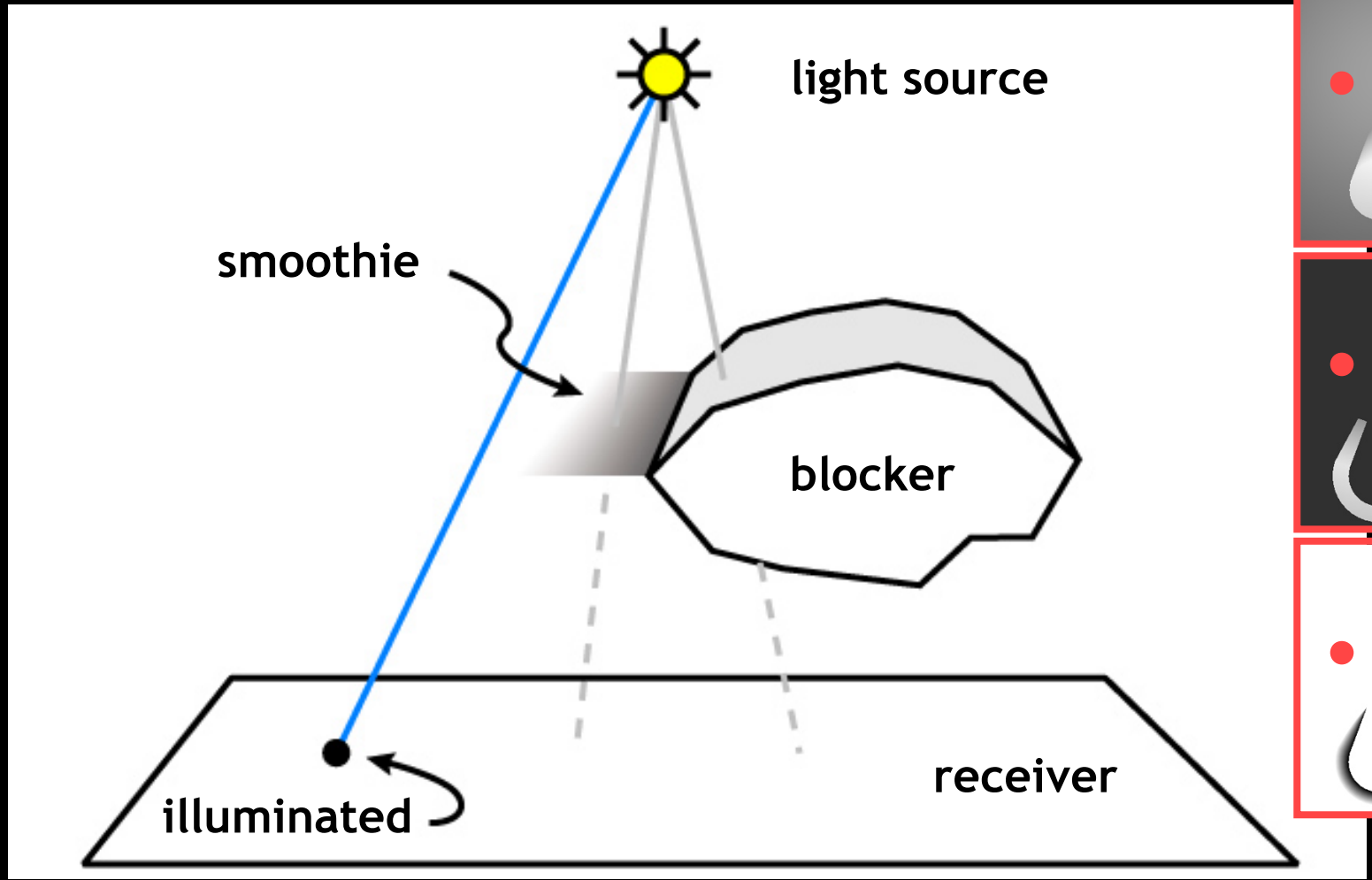
5. Compute Shadows

Image sample behind smoothie (intensity = α)



5. Compute Shadows

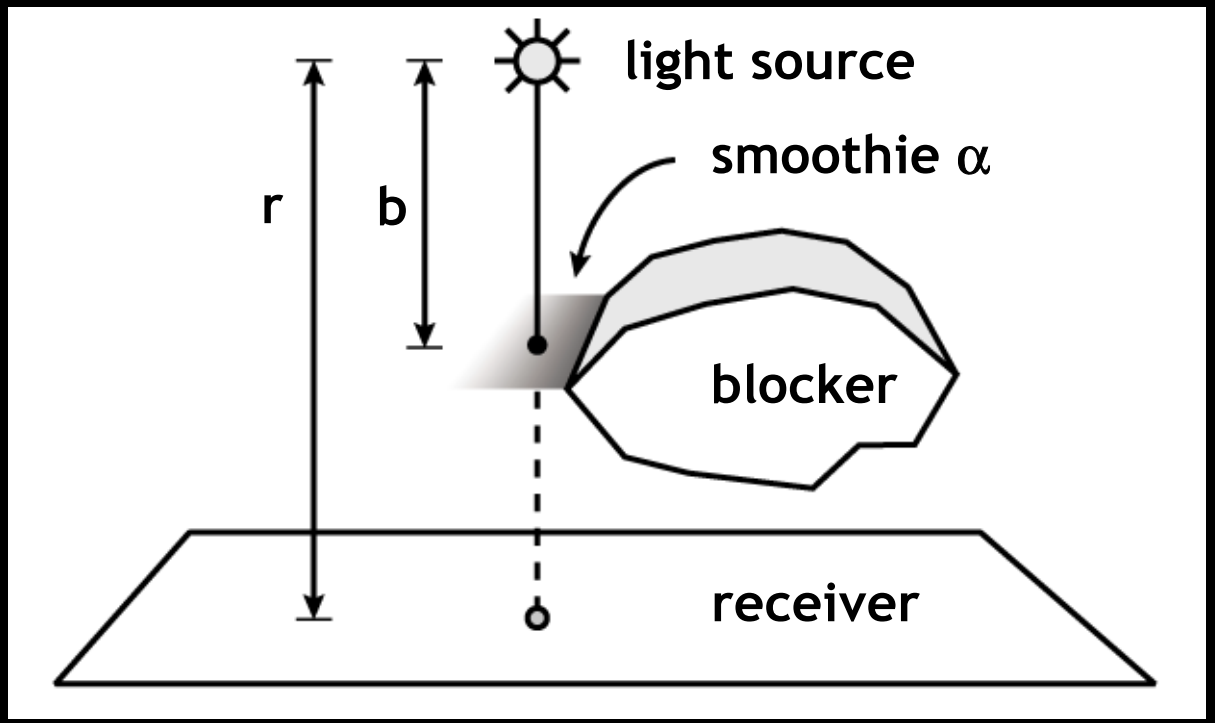
Image sample illuminated (intensity = 1)



Computing Alpha Values

Intuition:

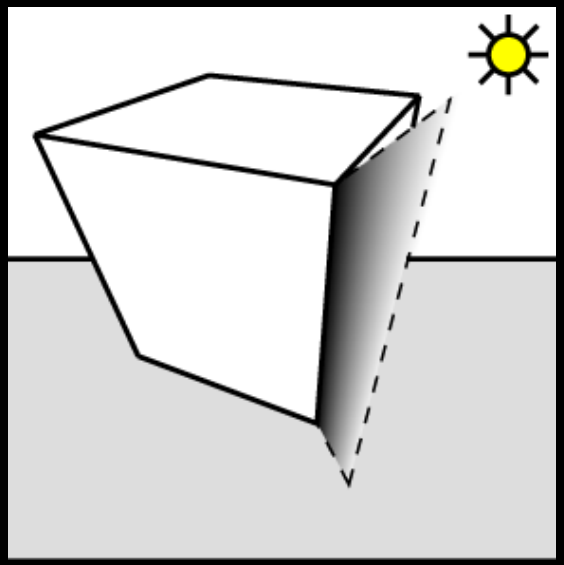
- Alpha defines penumbra shape
- Should vary with ratio b/r



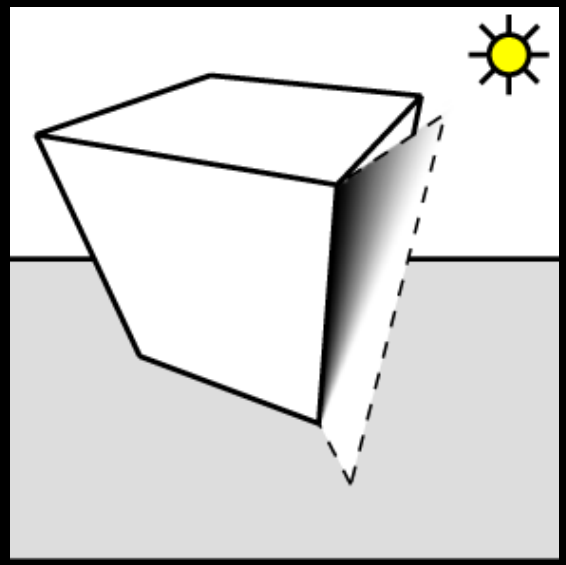
Computing Alpha Values (cont.)

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

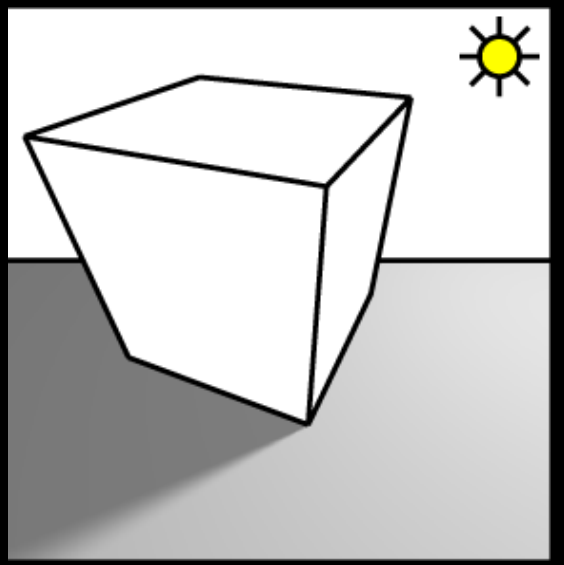
$$\alpha' = \alpha \,/\, (1 - b/r)$$



original α

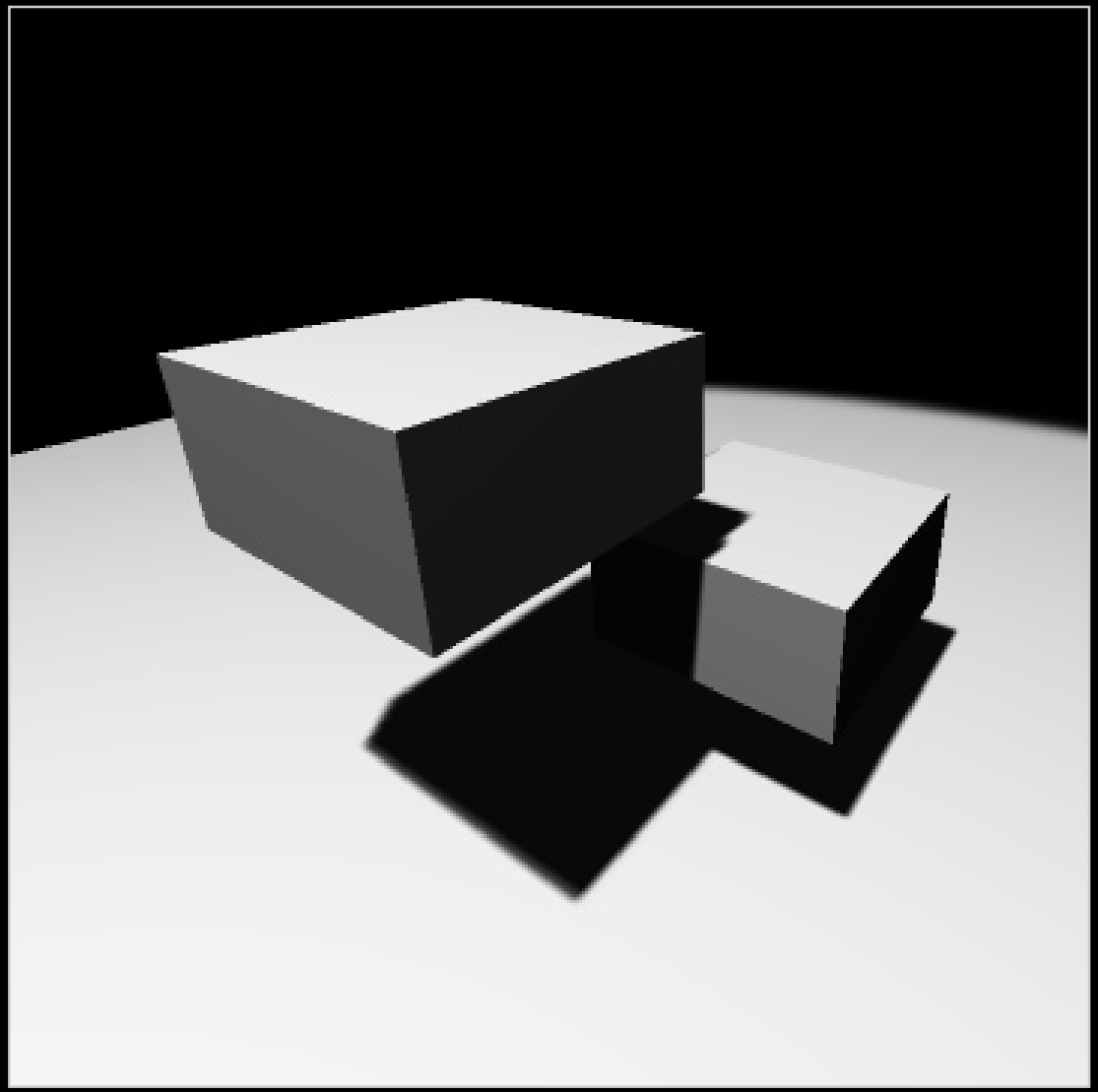


remapped α

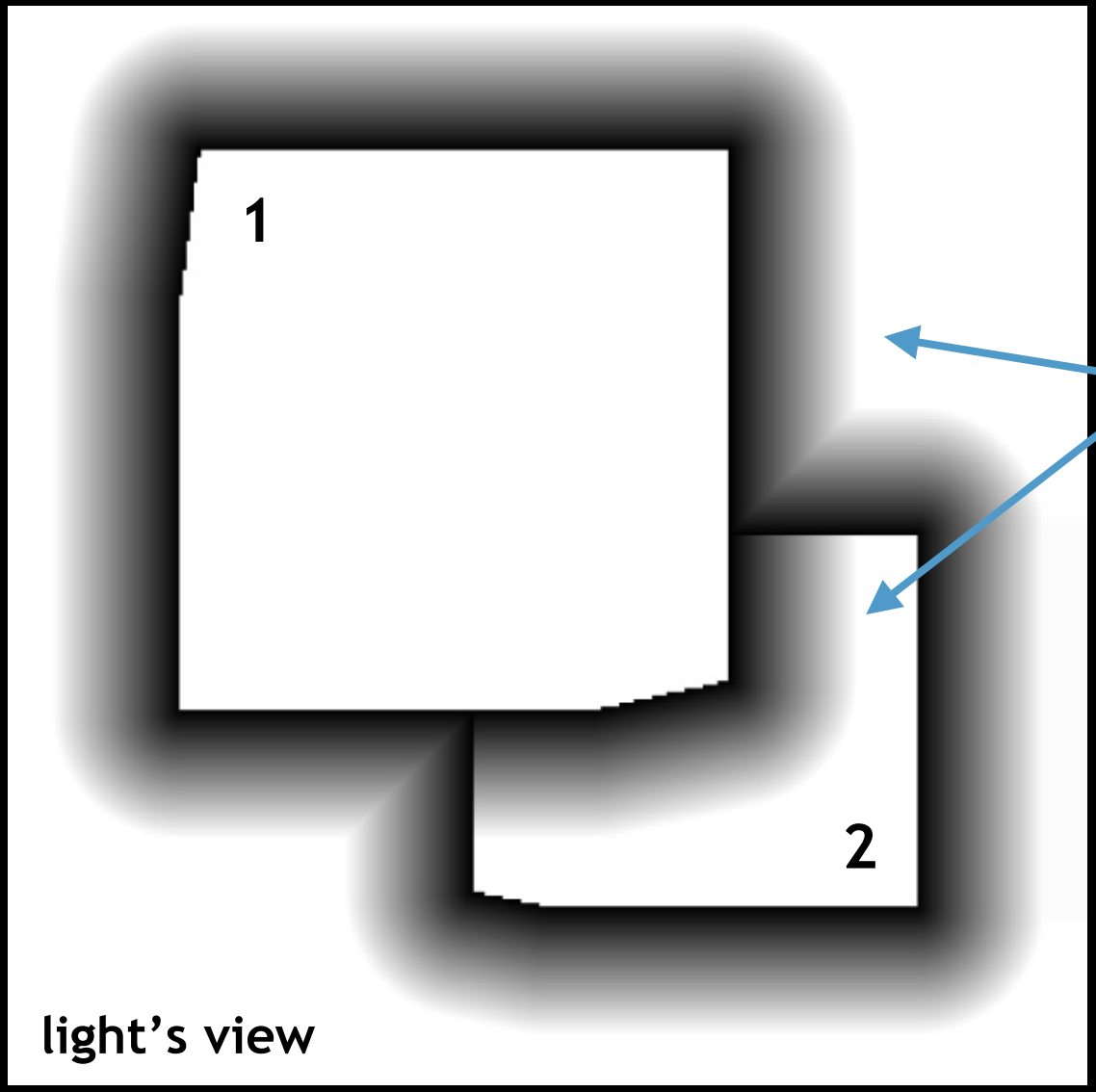


result

Multiple Blockers and Receivers



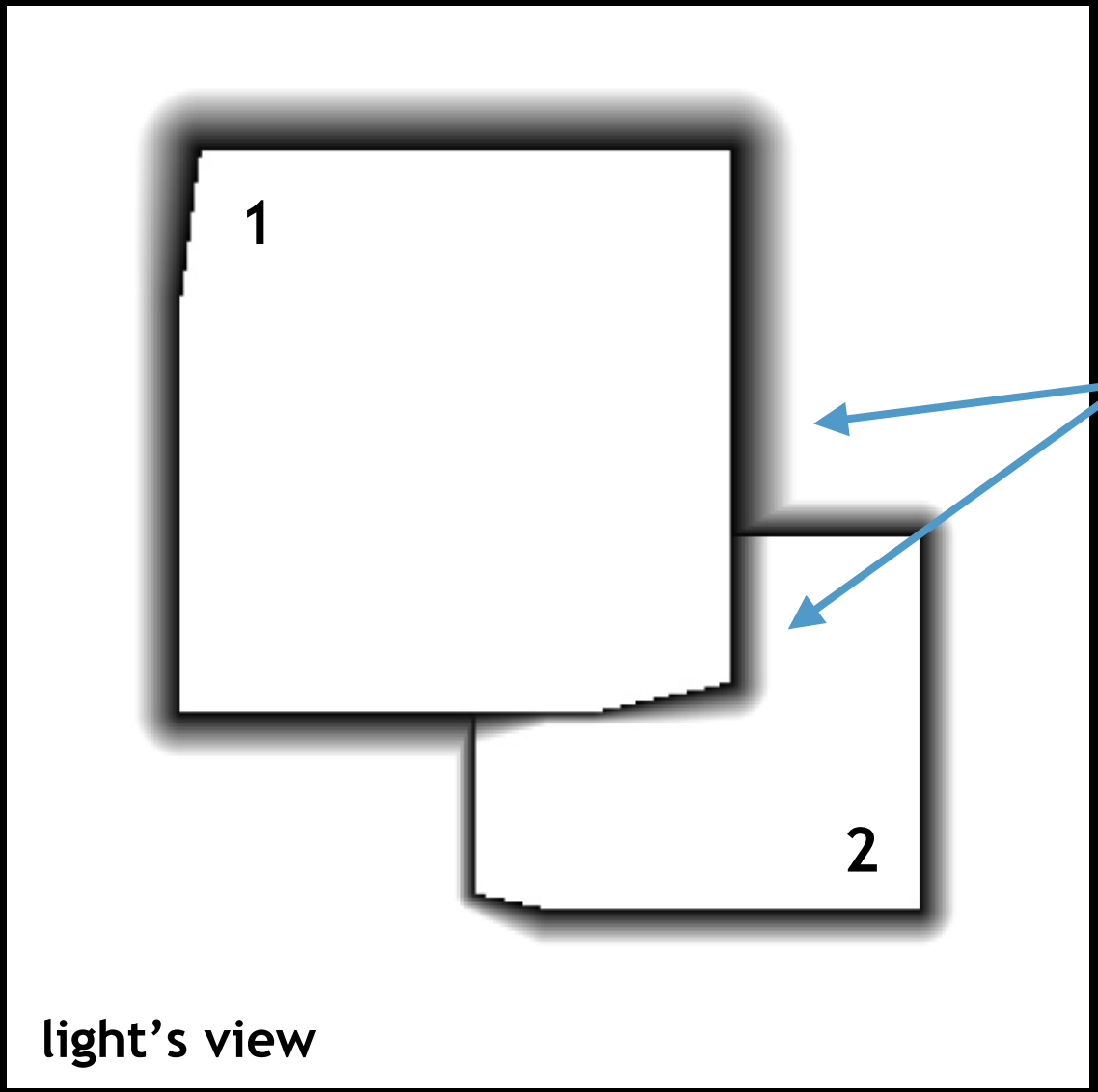
Multiple Receivers



Smoothie buffer
(linearly-interpolated α)

same thickness

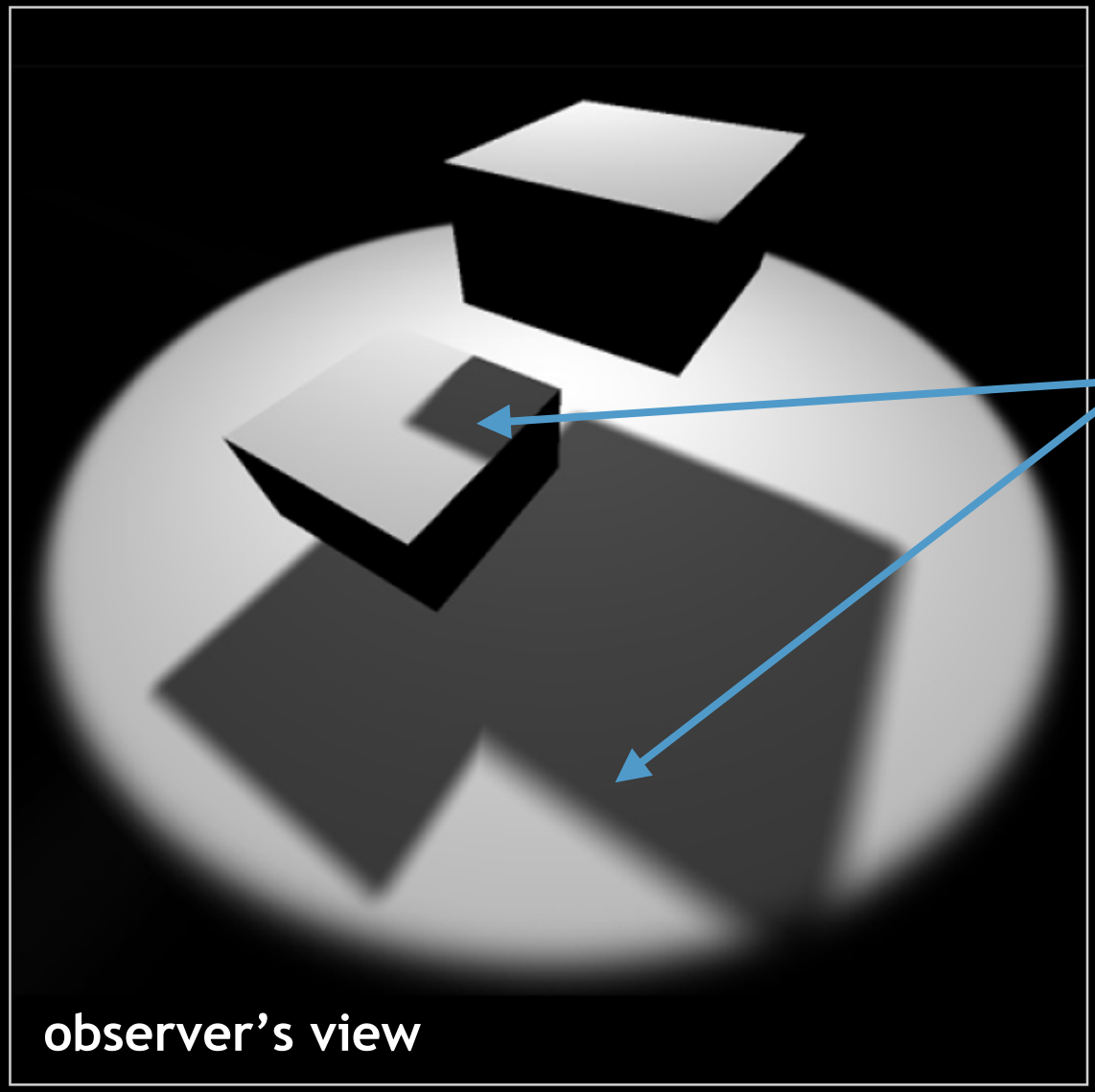
Multiple Receivers (cont.)



Smoothie buffer
(remapped α)

different thickness

Multiple Receivers (cont.)

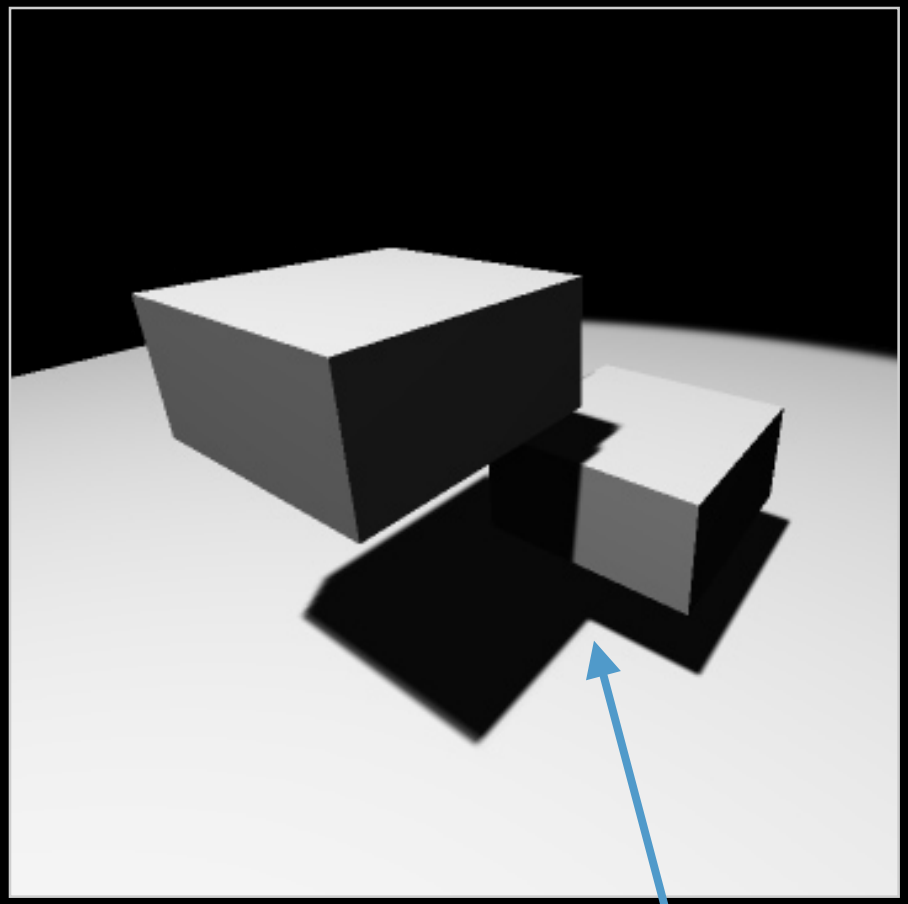


Final image

different thickness

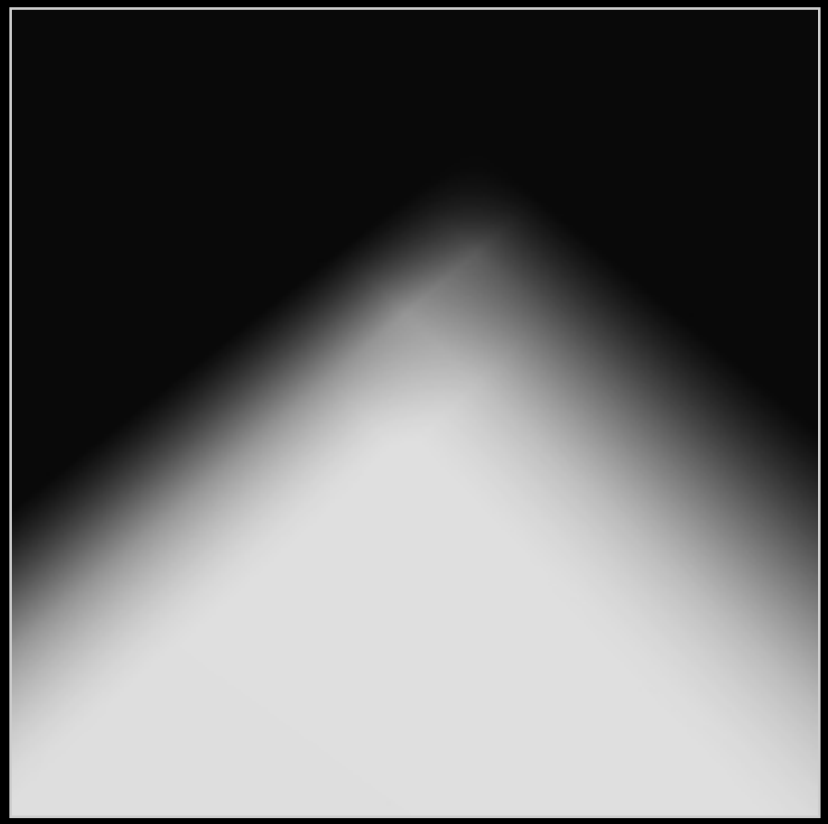
Multiple Blockers

What happens when smoothies overlap?

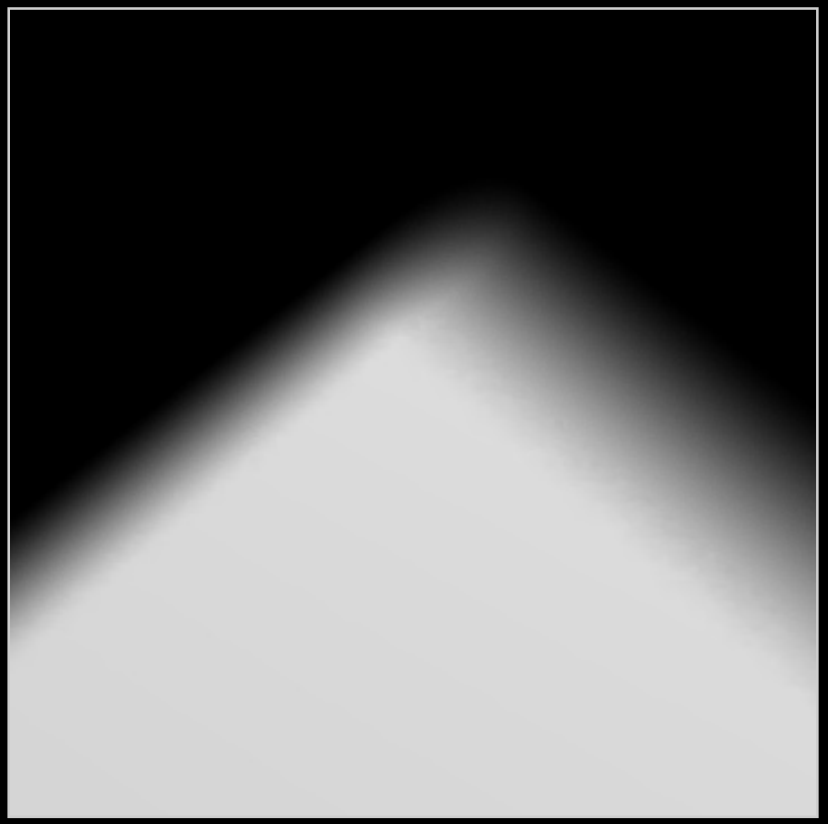


Multiple Blockers (cont.)

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

	Penumbra Maps	Smoothies
Geometry:	cones and sheets	quads
Store depth:	blockers only	blockers + smoothies

Smoothie depth:

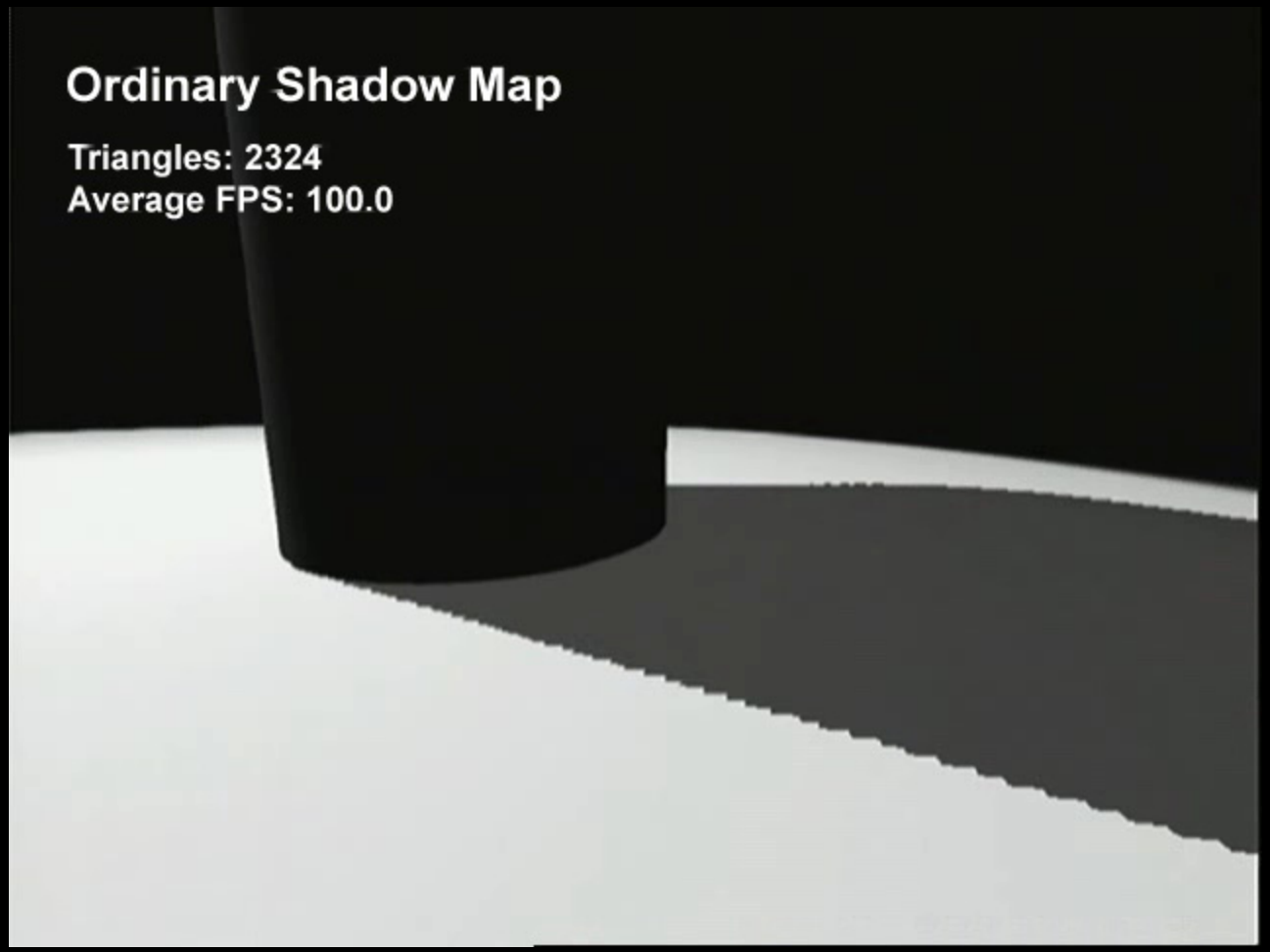
- Extra storage + comparison
- Handles surfaces that act only as receivers

Results

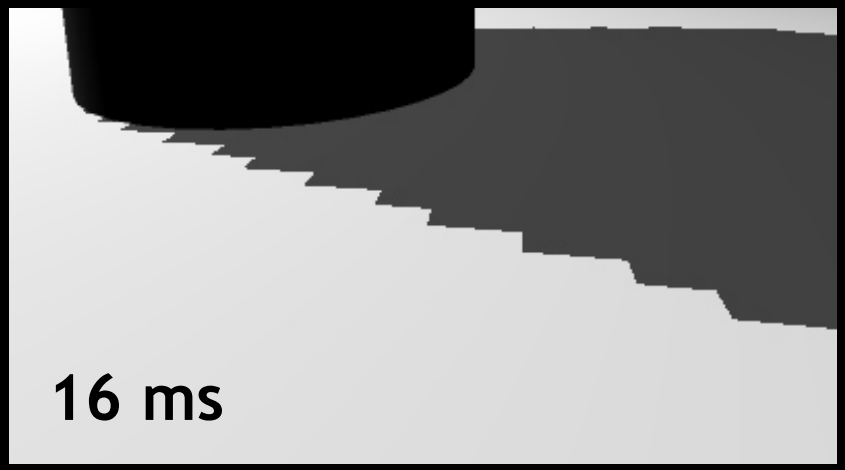
System information:

- 2.6 GHz Intel Pentium 4
- NVIDIA Geforce FX 5800 Ultra

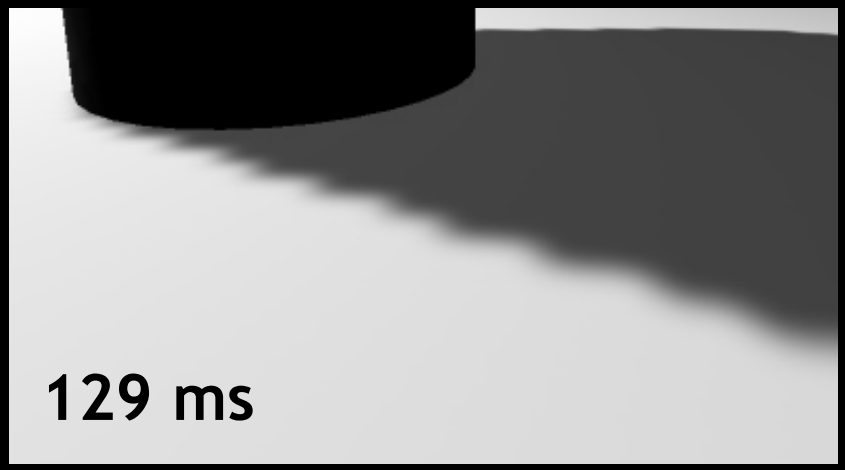
Video



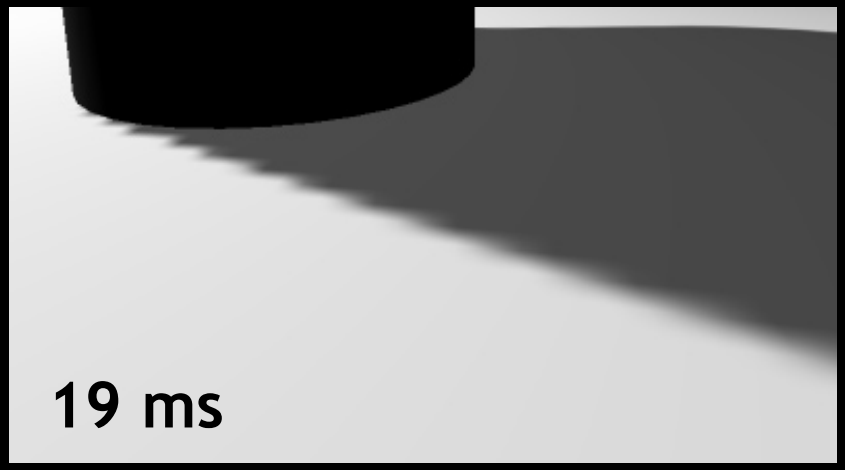
Hiding Aliasing (256 x 256)



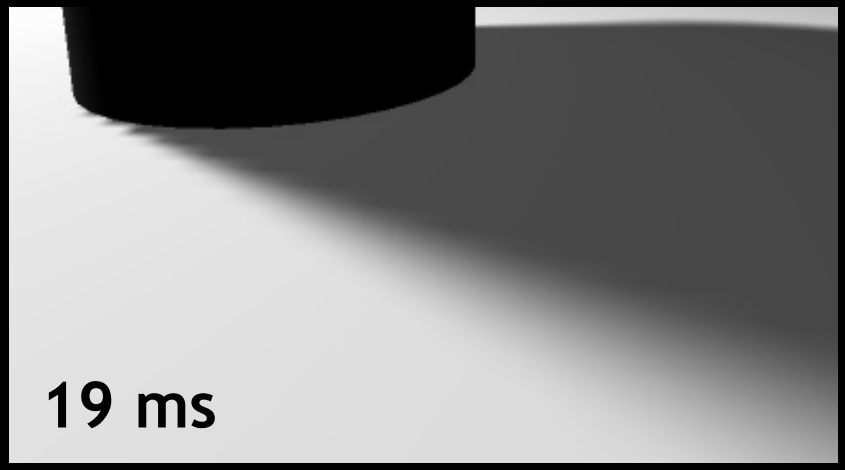
shadow map



bicubic filter

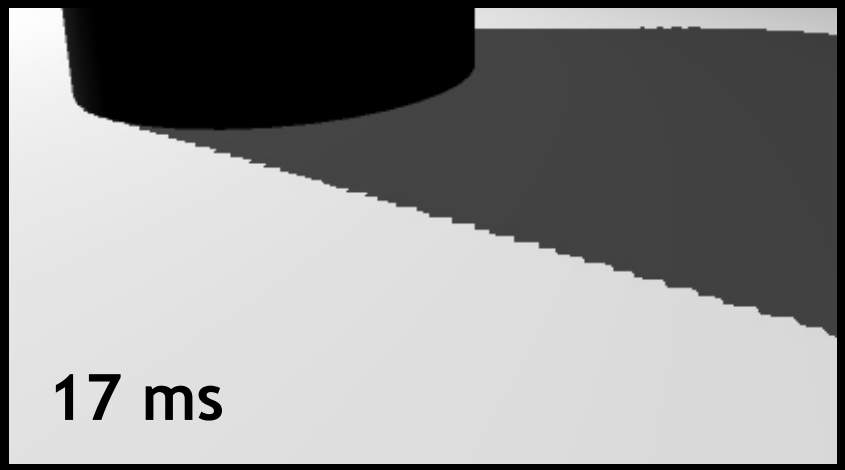


smoothie (t = 0.02)

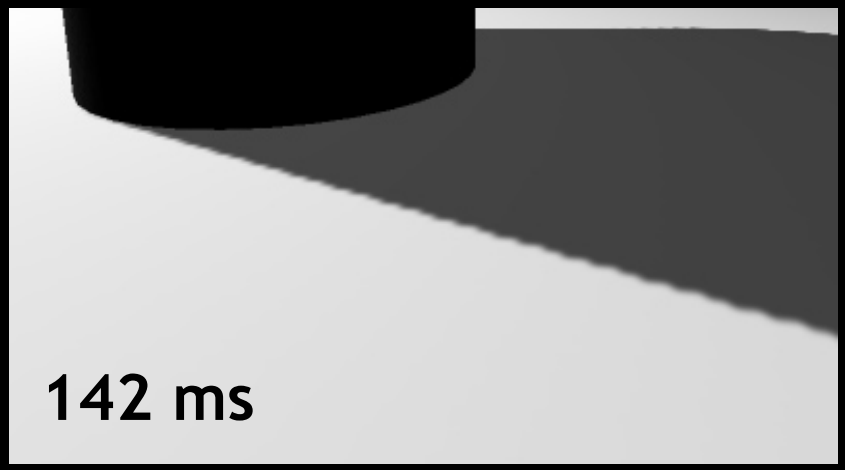


smoothie (t = 0.08)

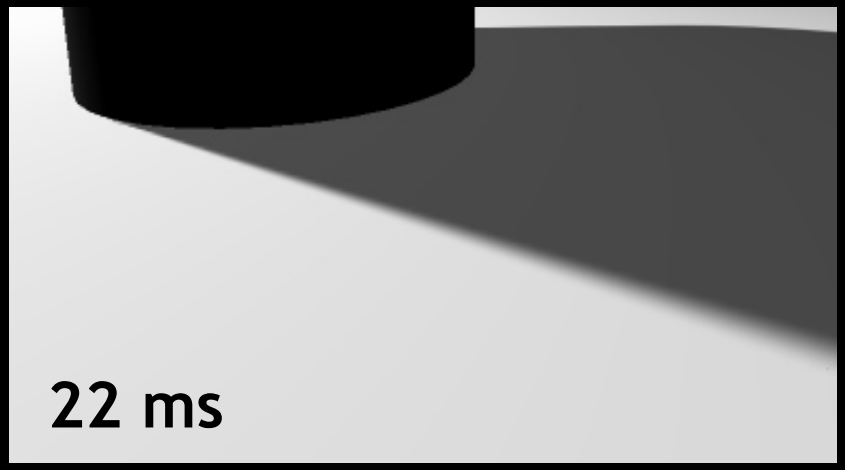
Hiding Aliasing (1024 x 1024)



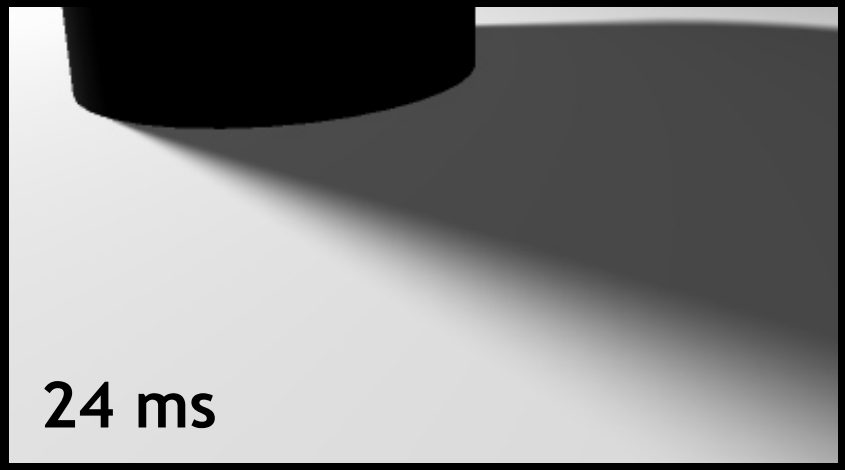
shadow map



bicubic filter

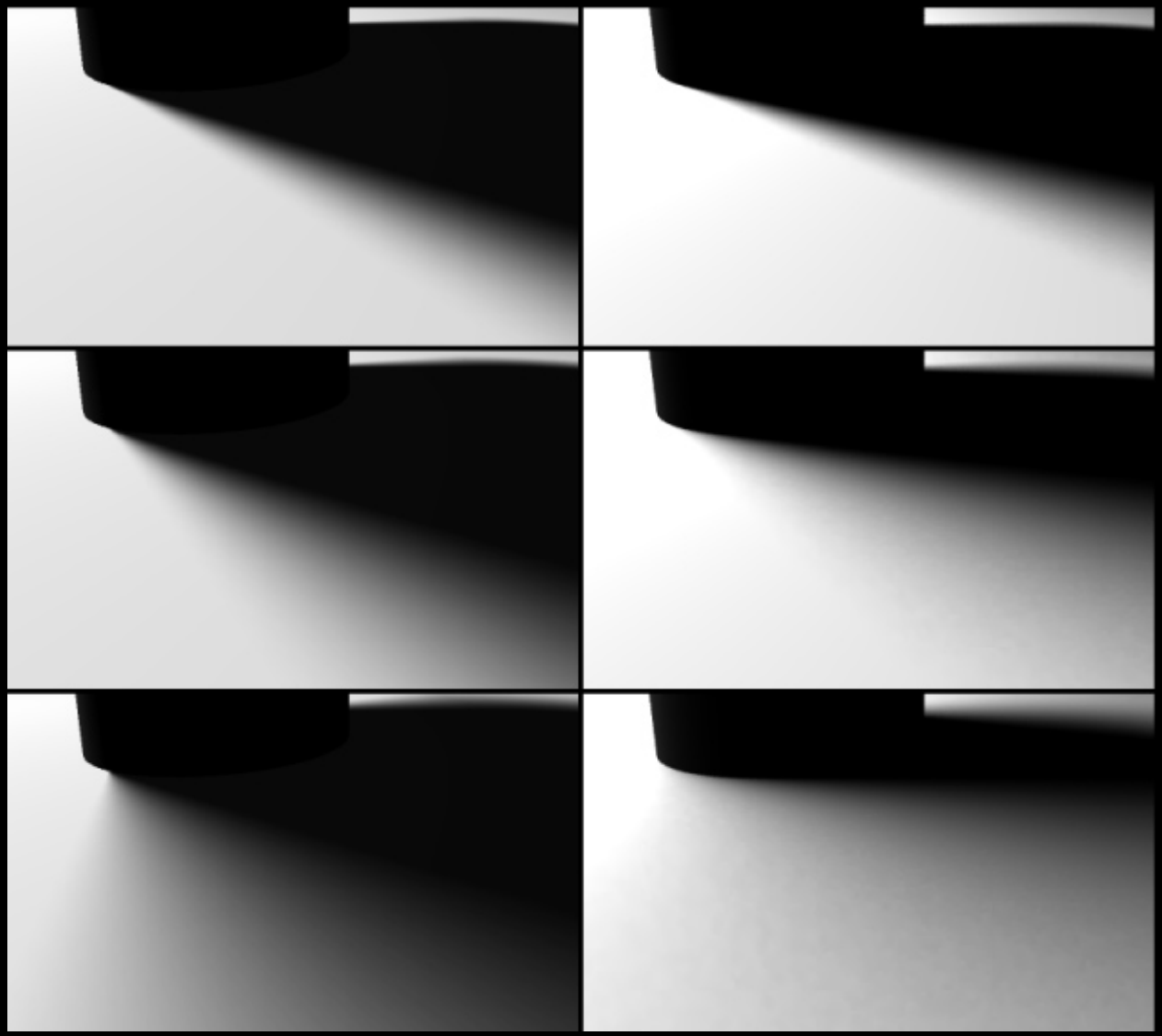


smoothie (t = 0.02)



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

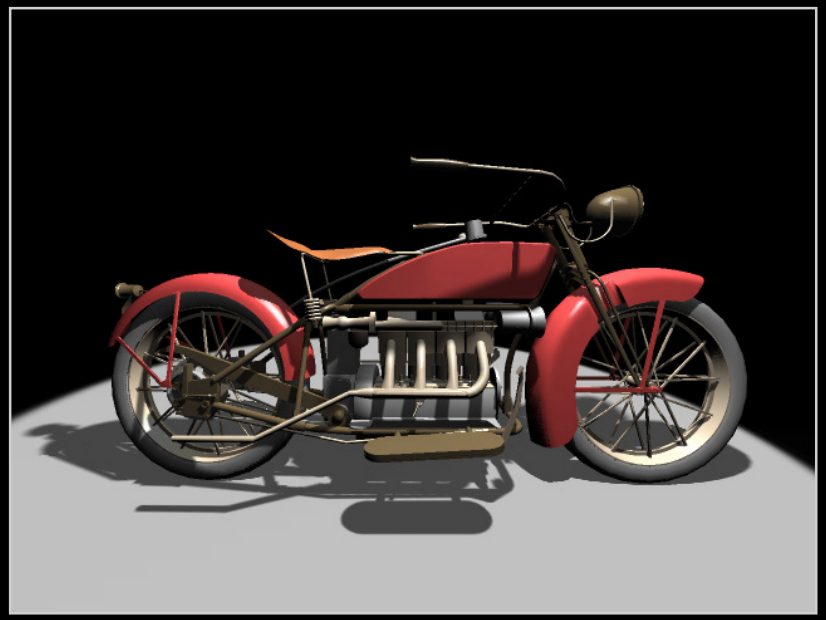
Smoothies:

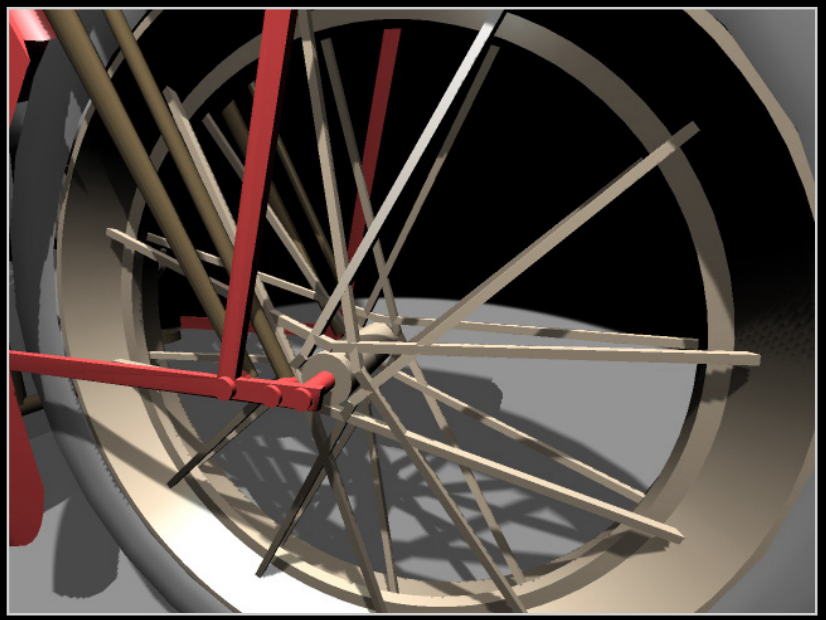
- Rendered from light's viewpoint
- Occupy small screen area → inexpensive

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