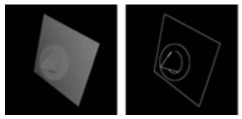
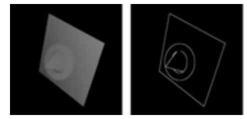
A Hybrid Approach

One more shadow algorithm which deserves mention is McCool's clever idea shadow volume reconstruction from depth maps [McCool 2000].

This algorithm is a hybrid of the shadow map and shadow volume algorithms and does not require a polygonal representation of the shadow volumes.



A Hybrid Approach



Instead of finding the silhouette edges via a dot product per model edge (shadow volumes),

1. A depth map of the scene from the light's point of view is acquired (shadow map)
2. From which the silhouette edges are extracted using computer vision techniques (edge detection).
3. From these edges the shadow volumes are constructed: silhouette edges are extruded

An Efficient Hybrid Shadow Rendering Algorithm

Eric Chan
Frédo Durand

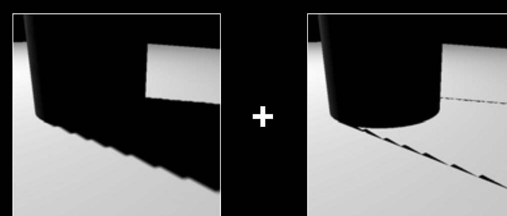
Massachusetts Institute of Technology

CSAIL

Not Another Talk on Shadows?!

Main ideas:

- combination of shadow maps + shadow volumes
- computation masks



Classic Shadow Algorithms

Shadow maps (Williams 1978)

- fast and simple
- undersampling artifacts
- lots of recent research!

Shadow volumes (Crow 1977)

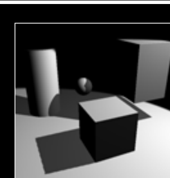
- object-space
- accurate
- accelerated by stencil buffer
- high fillrate consumption!

NVIDIA

Fillrate Problem

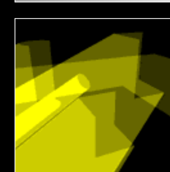
Lots and lots of fillrate!

- rasterization
- stencil updates



Why?

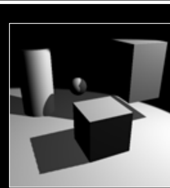
- polygons have large screen area
- polygons overlap



Fillrate Problem

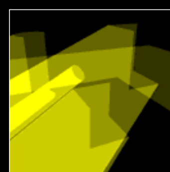
Lots and lots of fillrate!

- rasterization
- stencil updates



Why?

- polygons have large screen area
- polygons overlap



But is this really a problem?

But Is This *Really* A Problem?

Case study: Doom 3 engine (id Software)

- bump mapping
- per-pixel surface shading
- dynamic and projected lights
- atmospheric effects
- particle effects
- shadow volumes

But Is This *Really* A Problem?

Case study: Doom 3 engine (id Software)

- bump mapping
- per-pixel surface shading
- dynamic and projected lights
- atmospheric effects
- particle effects

} 50%

- shadow volumes

} 50%

“Shadowing accounts for about half of the game’s rendering time.”

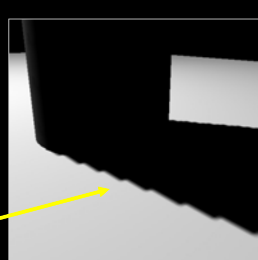
— John Carmack

Two Observations

Two Observations (shadow maps)

Shadow-map aliasing is ugly

But — only noticeable at shadow silhouettes

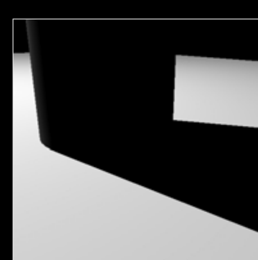


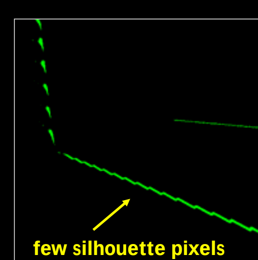
shadow silhouette

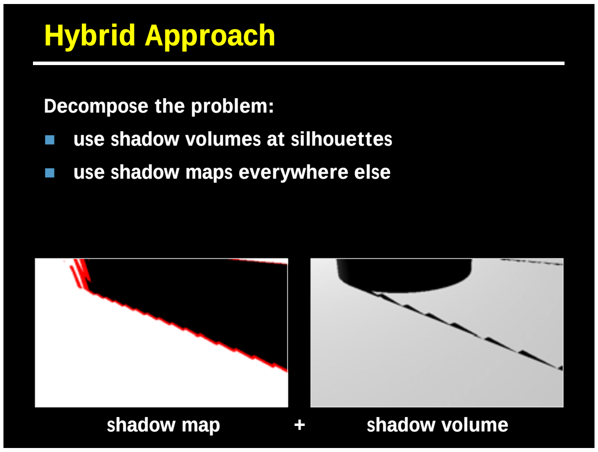
Two Observations (shadow volumes)

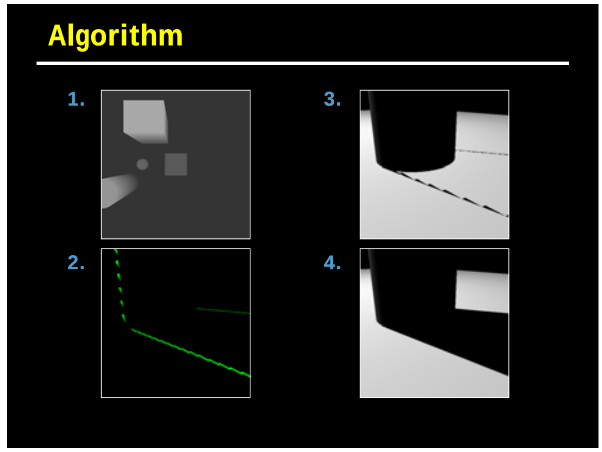
Shadow volumes are accurate everywhere

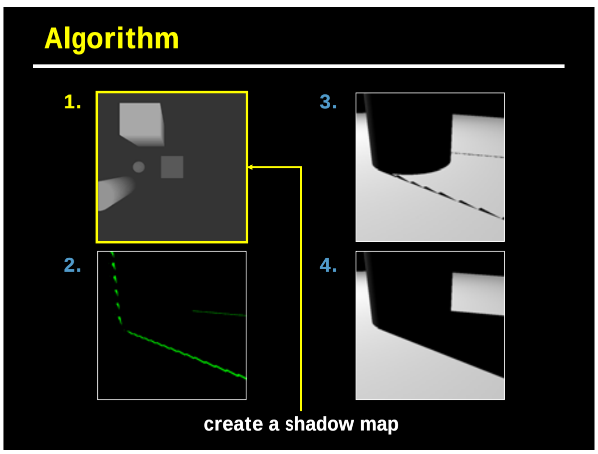
But — accuracy is only needed at silhouettes

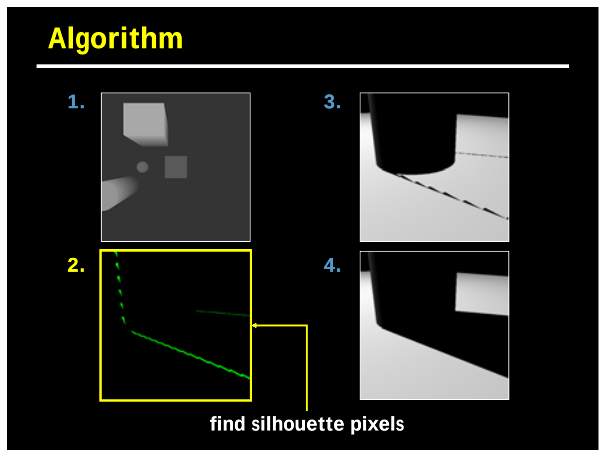


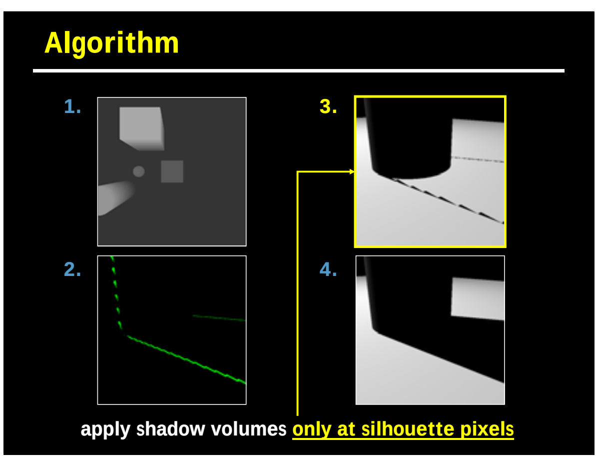


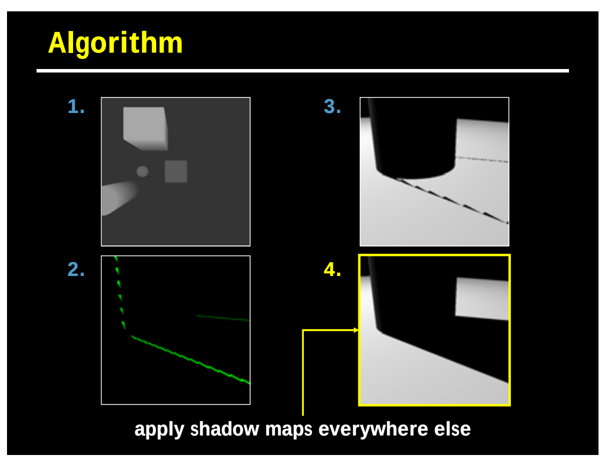

Hybrid Approach
Decompose the problem:
use shadow volumes at silhouettes
use shadow maps everywhere else
shadow map
+
shadow volume


Algorithm
1.
2.
3.
4.


Algorithm
1.
2.
3.
4.
create a shadow map

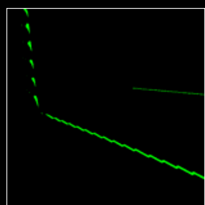

Algorithm
1.
2.
3.
4.
find silhouette pixels

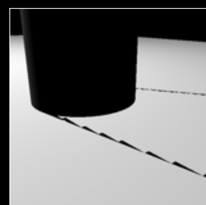

Algorithm
1.
2.
3.
4.
apply shadow volumes only at silhouette pixels


Algorithm
1.
2.
3.
4.
apply shadow maps everywhere else

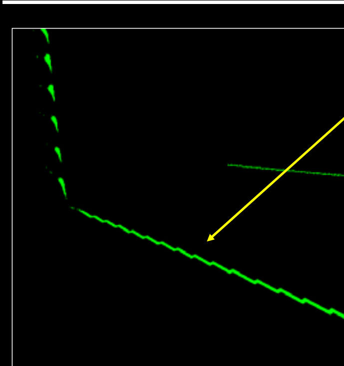
Algorithm Details

Questions:
- how to find silhouette pixels?
- how to rasterize <u>only</u> silhouette pixels?





Find Silhouette Pixels

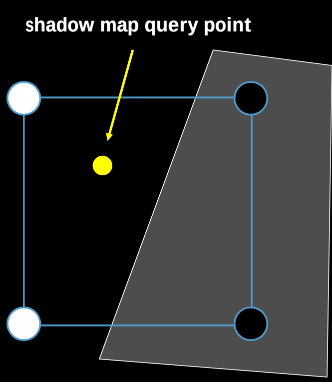


Silhouette pixels

Look for depth discontinuities

Use nearest 2x2 depth samples of the shadow map

Find Silhouette Pixels (example)



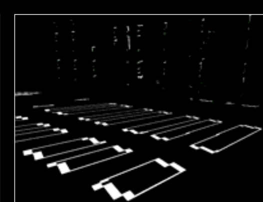
Check results:
- 2 in shadow
- 2 visible

Disagreement!
- silhouette pixel

Restricted Rasterization

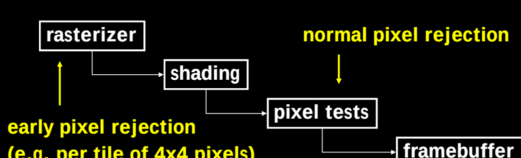
Use a mask to limit rasterization:
- tag silhouette pixels in framebuffer
- mask off all other pixels


example scene


mask

Computation Mask

We need a <u>computation mask</u>
- user-specified mask
- hardware early pixel rejection
- reduces rasterization, shading, memory bandwidth



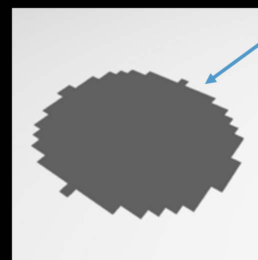
Hardware Support

Current hardware doesn't have computation mask
- but — hardware already has early z culling!
- minimal changes needed for native mask support
- our implementation uses a <u>simulated</u> mask

Results

- 2.6 GHz Pentium 4
- NVIDIA GeForce 6 (NV40) + crazy blue power supply

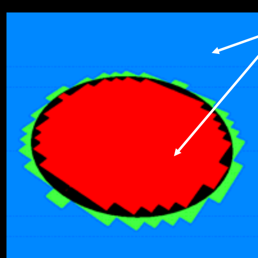
Hybrid Algorithm Example



Aliased shadow of a ball

standard shadow map

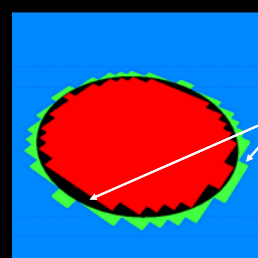
Hybrid Algorithm Example



Blue and red regions handled by shadow maps

visualization

Hybrid Algorithm Example

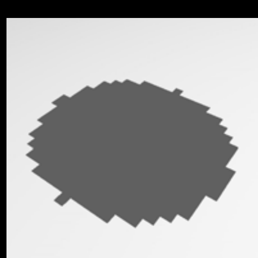


Blue and red regions handled by shadow maps

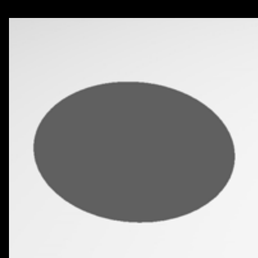
Black and green regions handled by shadow volumes

visualization

Hybrid Algorithm Example

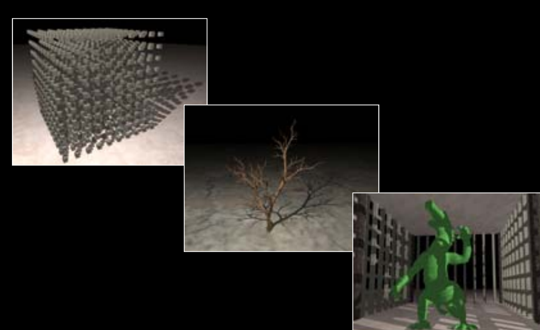


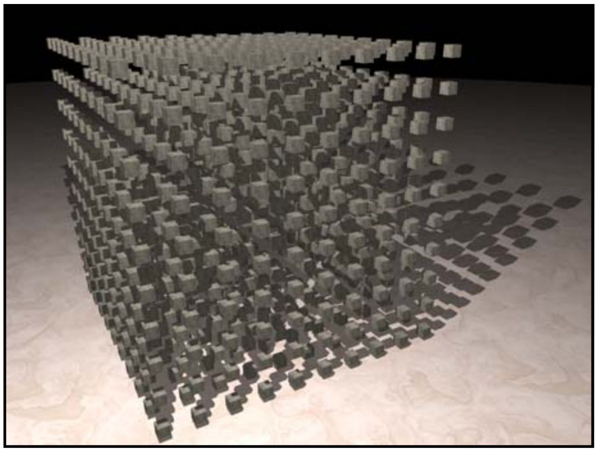
standard shadow map



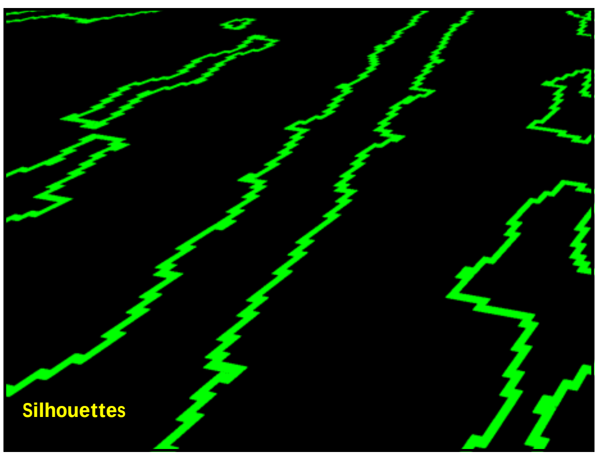
hybrid algorithm

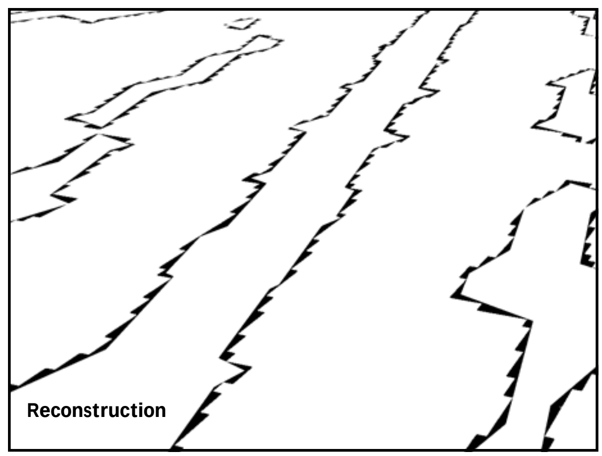
Test Scenes

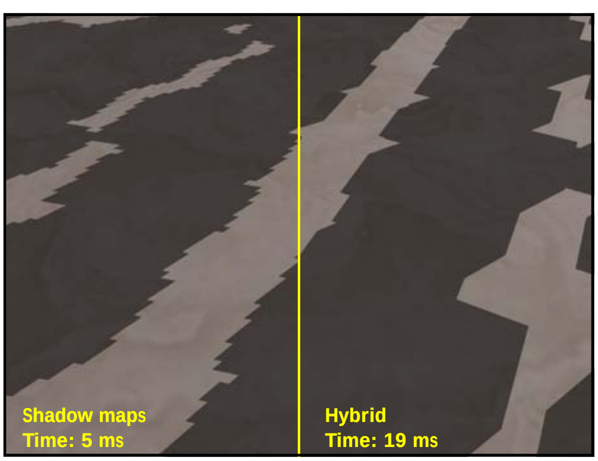


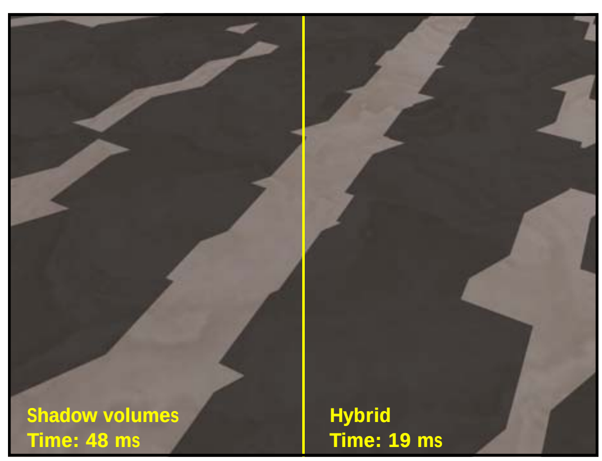



Shadow maps


Silhouettes

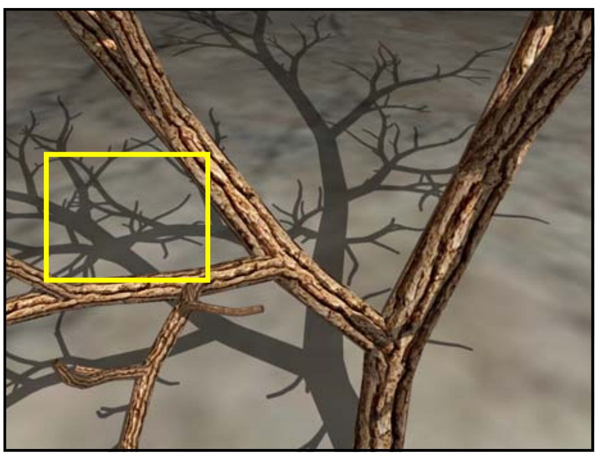

Reconstruction


Shadow maps
Time: 5 ms
Hybrid
Time: 19 ms


Shadow volumes
Time: 48 ms
Hybrid
Time: 19 ms

Artifacts

Low-resolution shadow map ⟶ discretization errors
Misclassified silhouette pixels ⟶ missing features
Difficult cases: fine geometry

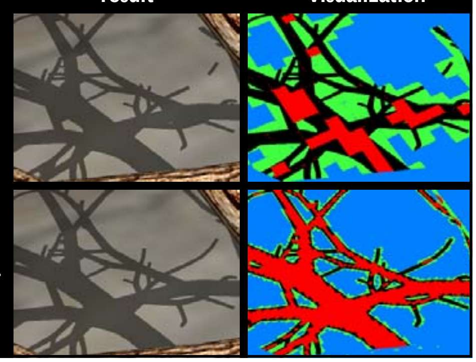


Example of Missing Features

result visualization

256x256

1024x1024



Discussion

Algorithm designed to help fillrate-bound applications:

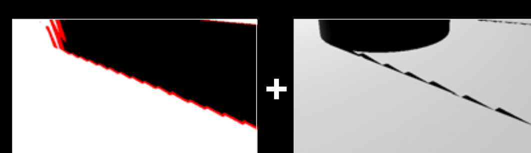
- requires an extra rendering pass
- 30% to 100% speedup in our test scenes
- performance depends a lot on culling hardware

More details in the paper and web page ...

- tradeoff analysis
- comparison to related work
- implementation details
- more performance and image comparisons

Summary

Hybrid shadow algorithm



Screen-space decomposition:

- most pixels use fast (but inexact) algorithm
- a few pixels use accurate (but expensive) algorithm

Computation Masks

Why?

- pixels are not created equal
- programmer marks “interesting” pixels
- fast reject all other pixels
- not just for shadows!
- useful in general for multipass algorithms
- hardware is (mostly) already there

Acknowledgments

Nick Triantos and Mark Kilgard (NVIDIA)

Jan Kautz and Addy Ngan (MIT)

Timo Aila

ASEE NDSEG Fellowship