ECE 595 / CS 491 / CS 591

Real-Time Rendering & Graphics Hardware

Pradeep Sen
Advanced Graphics Lab

Class 13
March 5, 2007

Announcement

- In class mid-term on Wednesday
- I will post study guide today

Last time

- Talked about texturing pipeline
- Discussed techniques to combat aliasing artifacts

Today

- Finally, we start talking about illumination
- First up: shadowing!

Shadows are an important part of any image

- Help define the spatial relationship between objects

source: artis.imag.fr

Doom 3, id (2004)

Final Fantasy, Square (2001)

Shadows are an important part of any image

- Help define the spatial relationship between objects
- Create “mood” for scene

Toy Story 2, Pixar (1999)

Shadows are an important part of any image

- Help define the spatial relationship between objects
- Create “mood” for scene
- Add to game play

Doom 3, id (2004)

Shadows are an important part of any image

- Help define the spatial relationship between objects
- Create “mood” for scene
- Add to game play
- Make an image look photorealistic

Final Fantasy, Square (2001)

Why are shadows so hard to compute?

- This is the first example of non-local light transport we are looking at
- Consider our rendering model: as we rasterize each triangle and shade it we don’t have information about the rest of the scene!
- To render shadows, we have to know what else is going on in the scene
- An example of “global illumination”

Kinds of shadows

- Hard shadows
  - Shadows cast from a point light source
  - Everything is either lit or not
- Soft shadows
  - Shadows cast from an area light source
  - Some points see the entire light source, some see none of it, and some see part of it

Examples

Hard shadows

Sen et al. SIGGRAPH 2003

Soft shadows

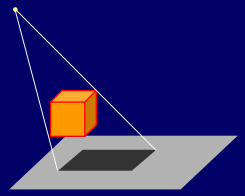
Ng et al. SIGGRAPH 2004

Hard shadows

- There are several algorithms to cast hard shadows in real-time:
  - Precomputed
  - Approximations
  - Correct self-shadowing

Planar shadows

- Used to cast shadows onto a flat plane
- Very common in games



Planar shadows

- Used to cast shadows onto a flat plane
- Very common in games

Madden 2007, Electronic Arts (2006)

In this class

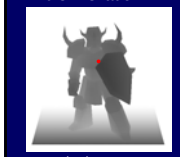
- We are interested in real-time algorithms that compute correct hard shadows for dynamic objects
- Currently there are two approaches for real-time hard shadows:
  - Shadow maps
  - Shadow volumes

Shadow maps

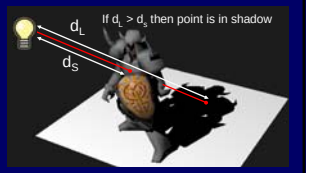
- Developed by Lance Williams (78)
- Fast, image-based technique
- Based on the observation that if the point light source cannot “see” a surface, then it is in shadow
- How do you implement visibility? Z-buffer!
- Implementation requires two passes: one from the light source, one from the final view position

Shadow mapping algorithm

1. Render scene depth from light source
2. To shade a point, measure its distance to the light and compare it to the value stored in the depth map
3. If point is farther to the light than the value in the depth map, it is in shadow



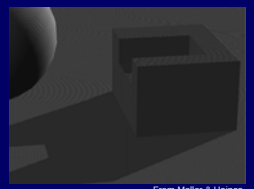
depth map



Shadow mapping demo

A few technical details

- You need to add a small bias to avoid self-shadowing



From Moller & Haines

A few technical details

- You need to add a small bias to avoid self-shadowing
- To create a point light source you need to render six views

Problems with shadow maps

- Aliasing artifacts because of quantization of depth map!

Proposed solutions for artifacts

- For the past 20 years, there have been many proposed solutions to address these quantization artifacts
  - Percentage Closer Filtering (Reeves et al.)
  - Perspective Shadow Maps (Stamminger et al.)
  - Adaptive Shadow Maps (Fernando et al.)
  - Silhouette Shadow Maps (Sen et al.)
  - etc.

Percentage closer filtering

- Developed by Reeves et al. SIGGRAPH 1987 to address these artifacts
- Observation: multiple samples can be used to reduce quantization artifacts
- The trick is how to filter these samples properly

Adaptive shadow maps

- Developed by Fernando et al. SIGGRAPH 2001
- Observation: combat aliasing by dedicating more samples to the region that the user is currently viewing
- They develop a hierarchical structure that stores depth maps of varying resolutions

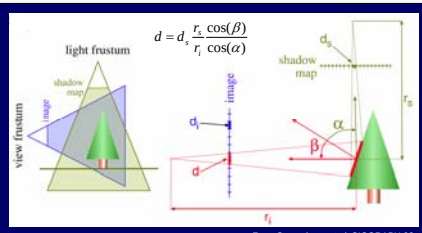
standard shadow maps

adaptive shadow maps

Perspective shadow maps

- Developed by Stamminger and Drettakis (SIGGRAPH 2002)
- Observation: one source of artifacts is the differences in perspective projection between the light source and viewer
- Combat these artifacts by transforming the objects by the eye's perspective projection before computing the shadow map

Sources of artifacts

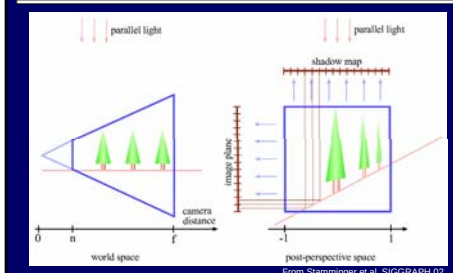


From Stamminger et al. SIGGRAPH 02

Perspective aliasing

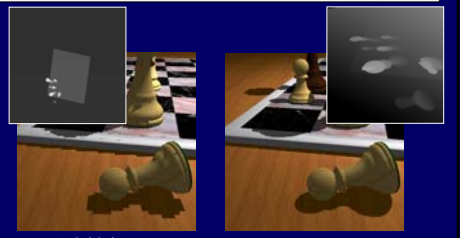
Projective aliasing

Perspective shadow maps



From Stamminger et al. SIGGRAPH 02

Perspective shadow maps results



standard shadow maps

perspective shadow maps

Problems with perspective shadow maps

- Lots of special cases that are difficult to implement

Shadow silhouette maps

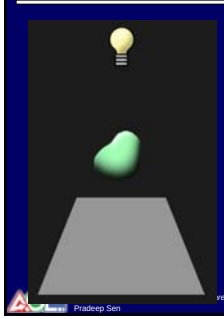
- Observation: shadow maps work well in regions that are fully lit and regions that are fully shadowed
- The quantization artifacts are due to the approximation of the sample as a box
- Can we use the silhouette map algorithm to deform the box to match shadow discontinuities?

Shadow silhouette map algorithm

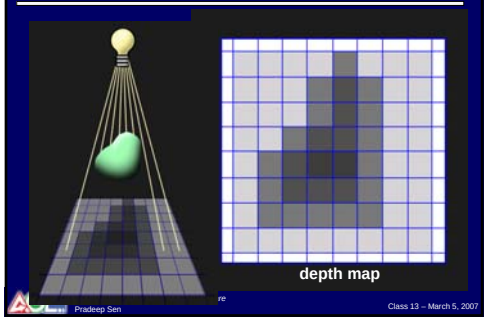
Enhance the depth map with a new texture: the silhouette map

depth map **silhouette map**

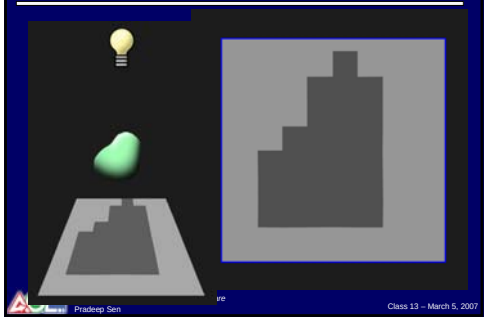
Standard depth map algorithm



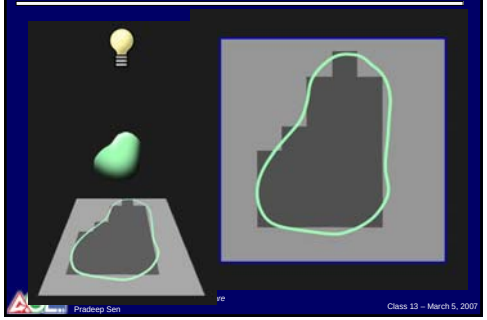
Standard depth map algorithm



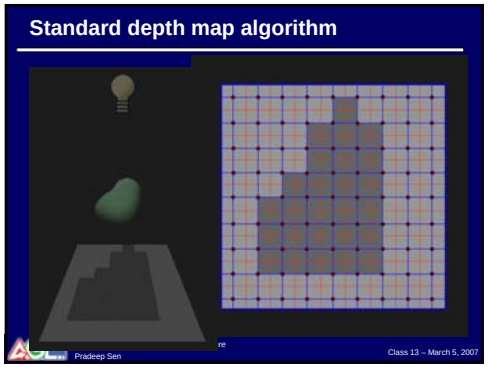
Standard depth map algorithm



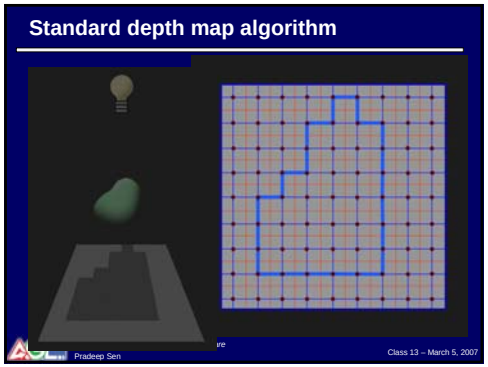
Standard depth map algorithm



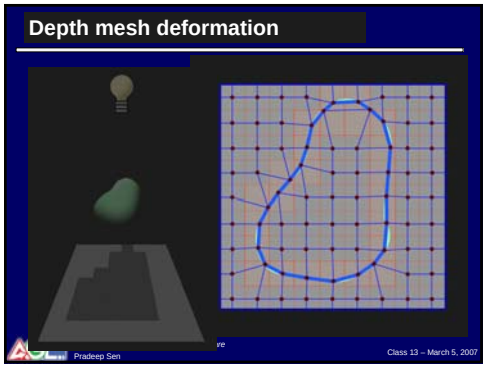
Standard depth map algorithm



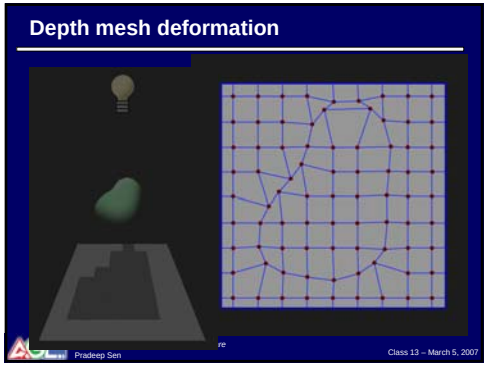
Standard depth map algorithm



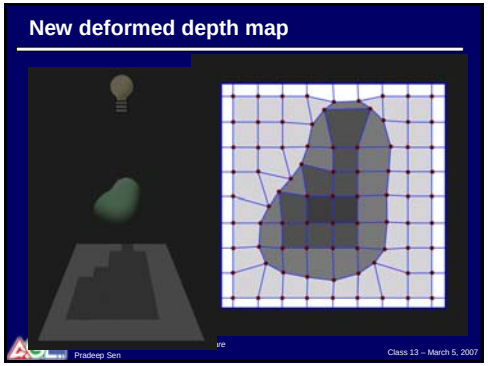
Depth mesh deformation



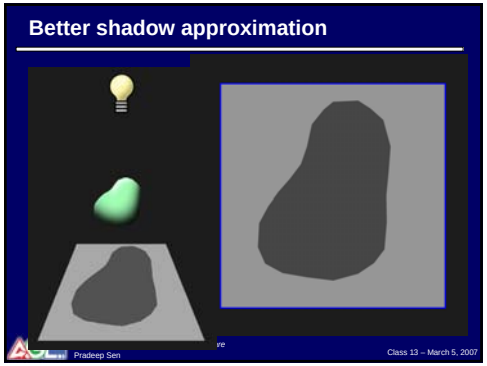
Depth mesh deformation



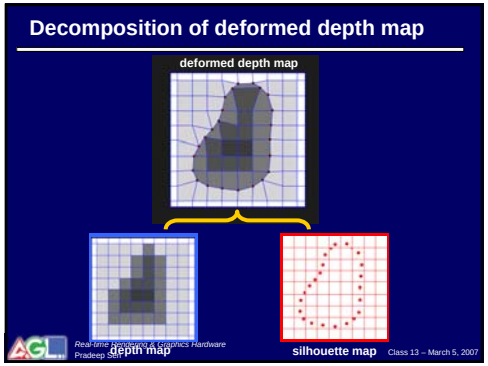
New deformed depth map



Better shadow approximation

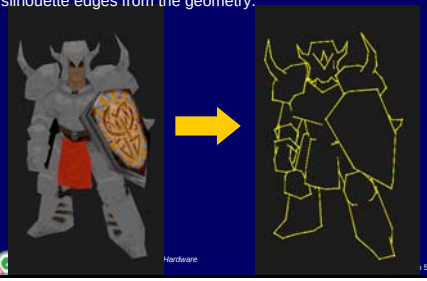


Decomposition of deformed depth map



Extract silhouette edges

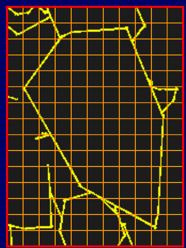
To render the silhouette map, we must first extract the silhouette edges from the geometry.



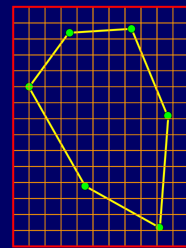
Rasterize edges into silhouette map

We rasterize the silhouette edges using a technique that guarantees 4-connected lines

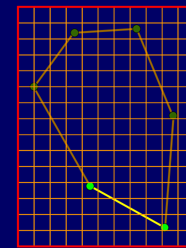
Generating silhouette map



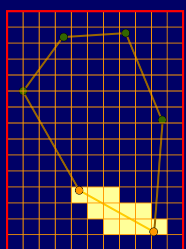
Generating silhouette map



Generating silhouette map

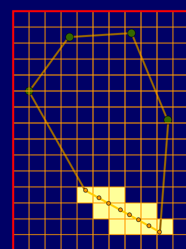


Generating silhouette map



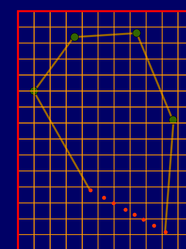
- Conservatively rasterize pixels which intersect line segment.
- Pass end points of line segment as interpolants to the fragment prog.

Generating silhouette map



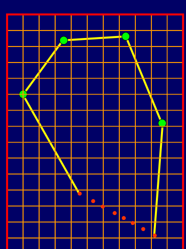
- Conservatively rasterize pixels which intersect line segment.
- Pass end points of line segment as interpolants to the fragment prog.
- Fragment prog. picks a point on the line segment inside the texel.

Generating silhouette map



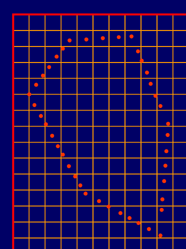
- Conservatively rasterize pixels which intersect line segment.
- Pass end points of line segment as interpolants to the fragment prog.
- Fragment prog. picks a point on the line segment inside the texel.

Generating silhouette map



- Conservatively rasterize pixels which intersect line segment.
- Pass end points of line segment as interpolants to the fragment prog.
- Fragment prog. picks a point on the line segment inside the texel.
- Repeat for all silhouette edges in scene. New line segments overwrite old values.

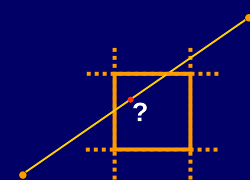
Generating silhouette map



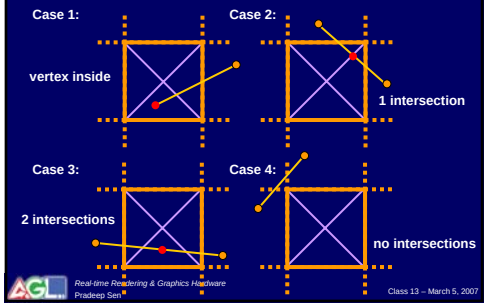
- Conservatively rasterize pixels which intersect line segment.
- Pass end points of line segment as interpolants to the fragment prog.
- Fragment prog. picks a point on the line segment inside the texel.
- Repeat for all silhouette edges in scene. New line segments overwrite old values.

Selecting a point on the line segment

Our fragment program must pick a point on the line segment which is inside the texel



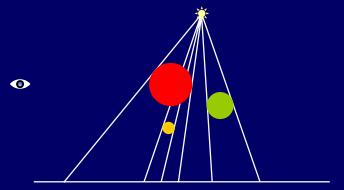
Our algorithm to pick silhouette points



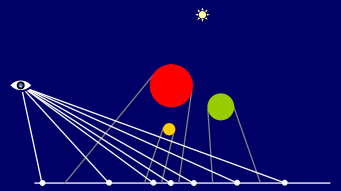
Shadow volumes

- The other leading approach for real-time hard shadows
- Developed by Crow (1978)
- Based on the observation that you can determine whether a point is in shadow or not by counting how many times it has entered/exited the shadow volumes

Shadow volume algorithm



Shadow volume algorithm



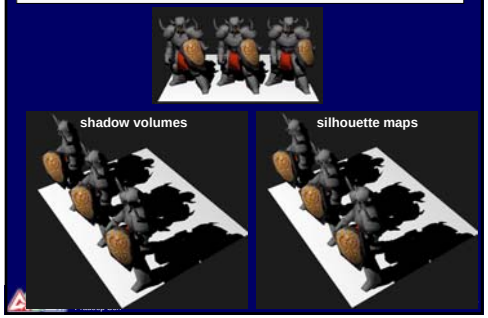
Shadow volume optimization

- Developed by Heidmann (1991)
- Observation: we can use the stencil buffer to do the counting for us
- Also known as “stencil shadows”

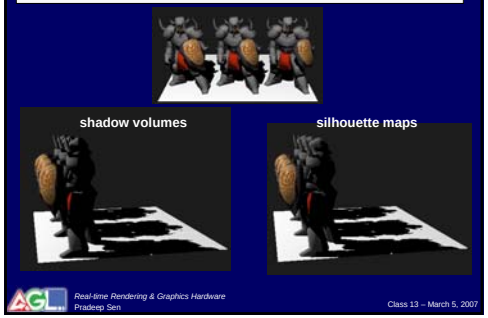
Problems with shadow volumes

- Large rasterization cost

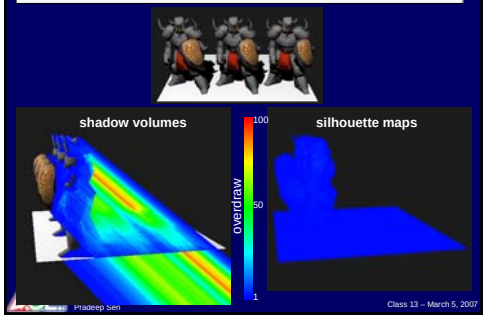
Overdraw problem



Overdraw problem



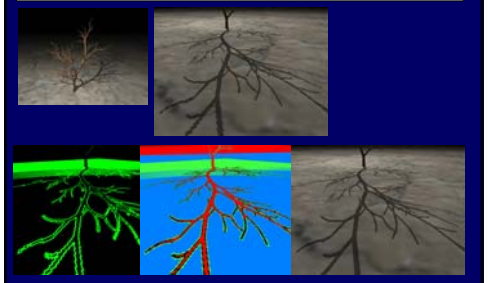
Overdraw problem



Hybrid algorithms

- Combine the best of both worlds: shadow maps and shadow volumes
- For example: Chan et al. 2004

Hybrid results



Reading

- Moller & Haines, Section 6.12