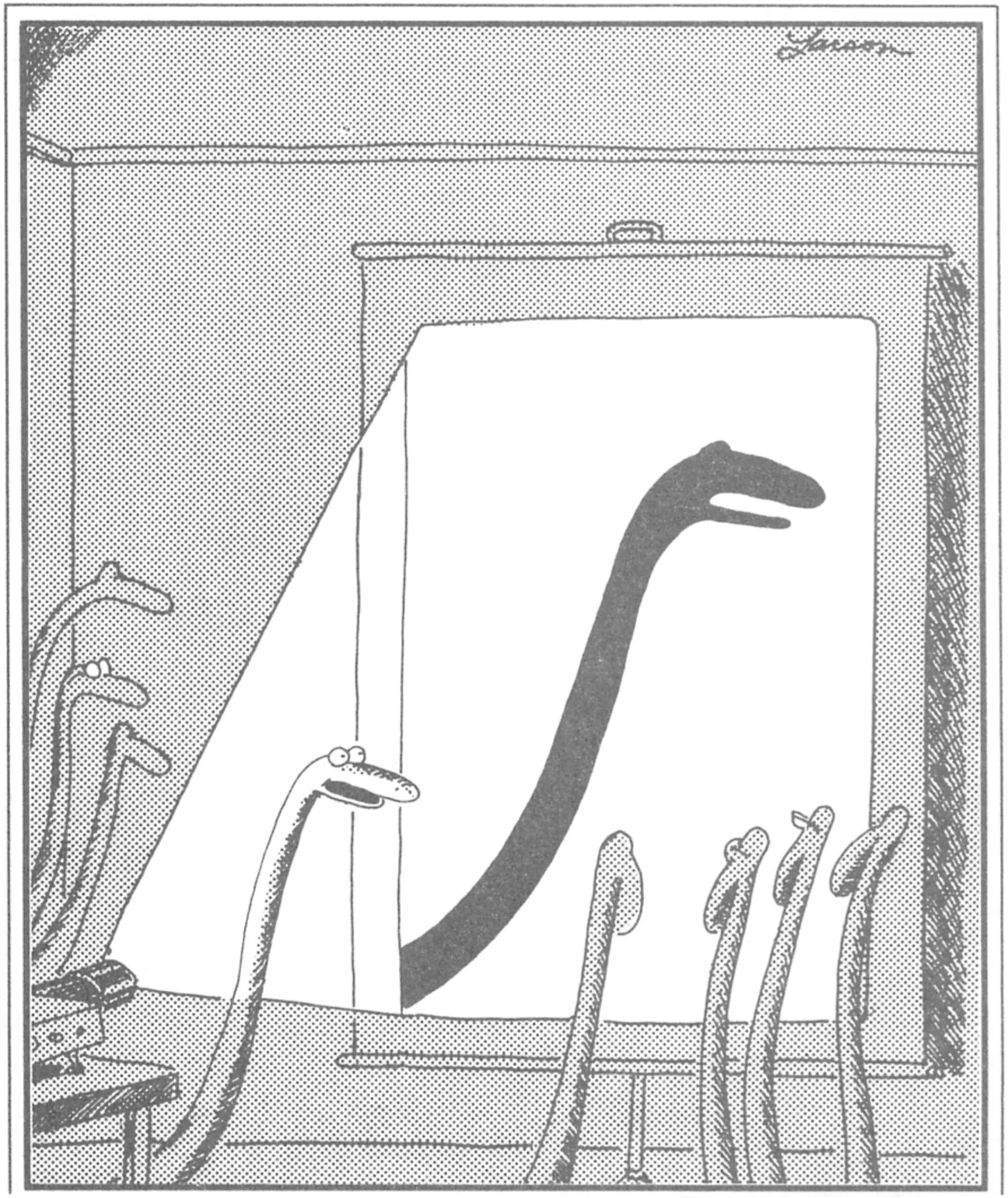
ME-C3100
Computer Graphics, F2015

Jaakko Lehtinen
with lots of material from
Frédo Durand

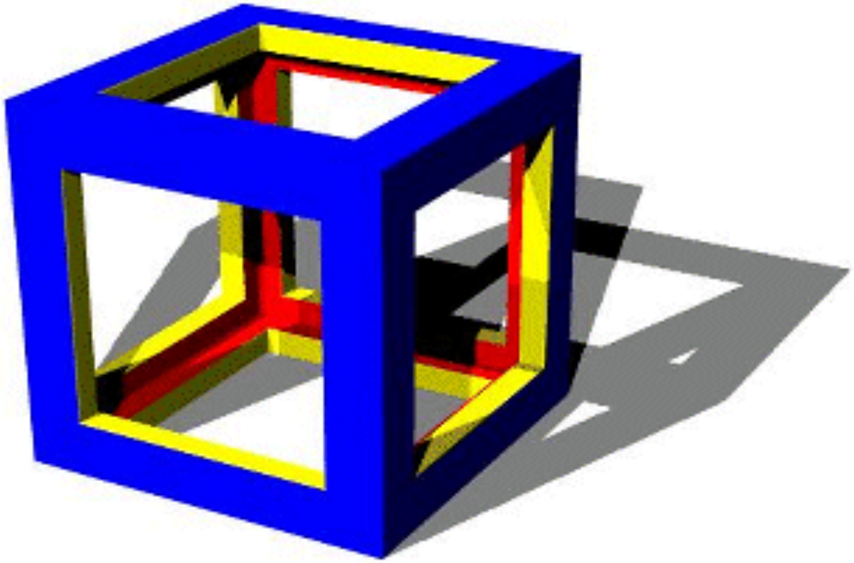
Real-Time Shadows



"Now this is…this is…well, I guess it's another snake."

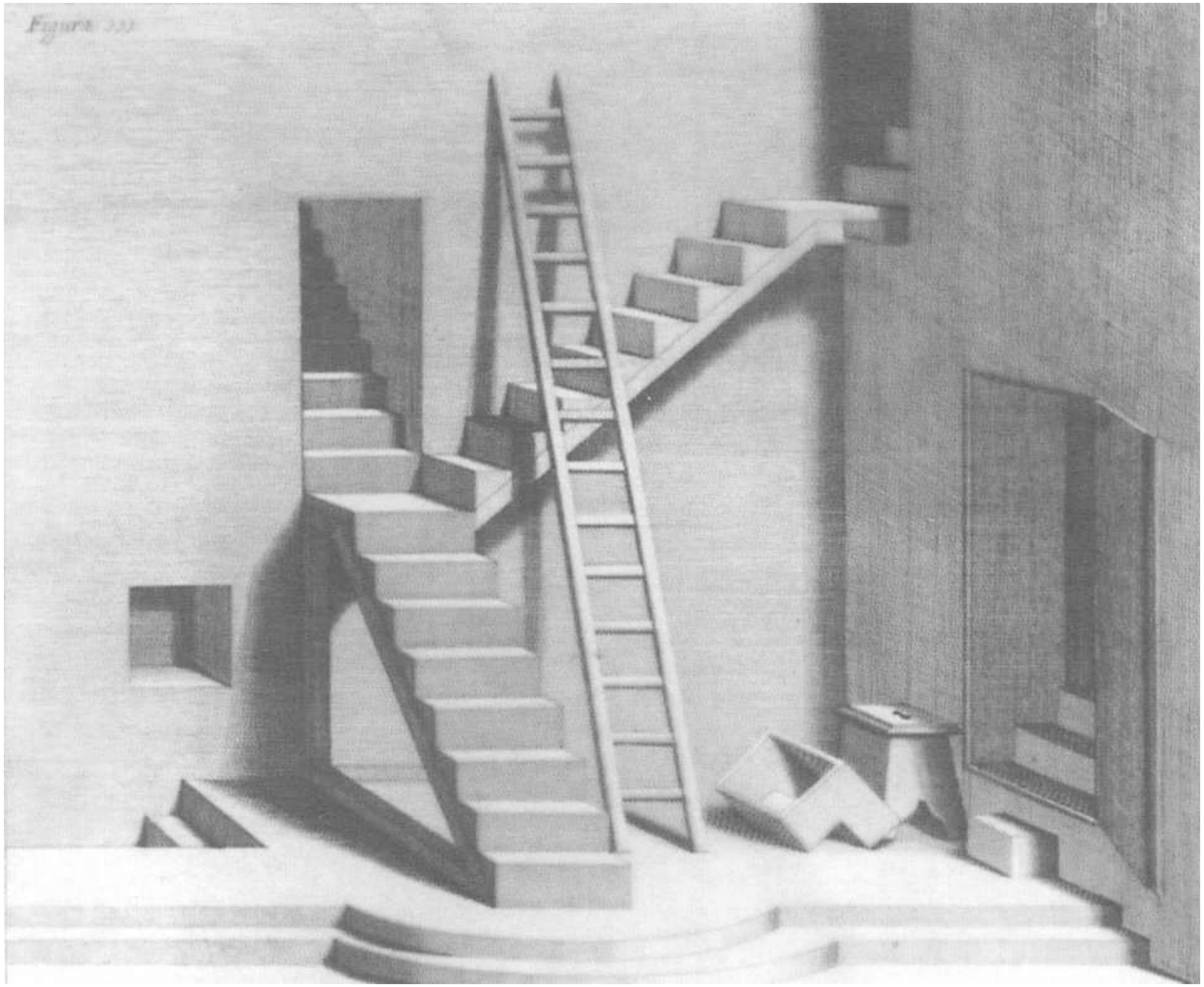
Today

- Shadows
  - Why are Shadows Important?
  - Shadows & Soft Shadows in Ray Tracing
  - Planar Shadows
  - Shadow Maps
  - Shadow Volumes

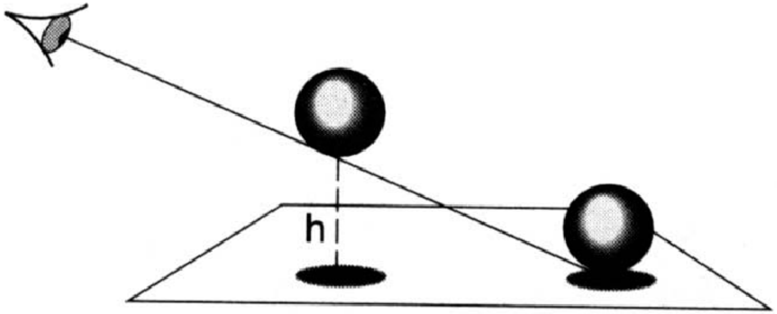


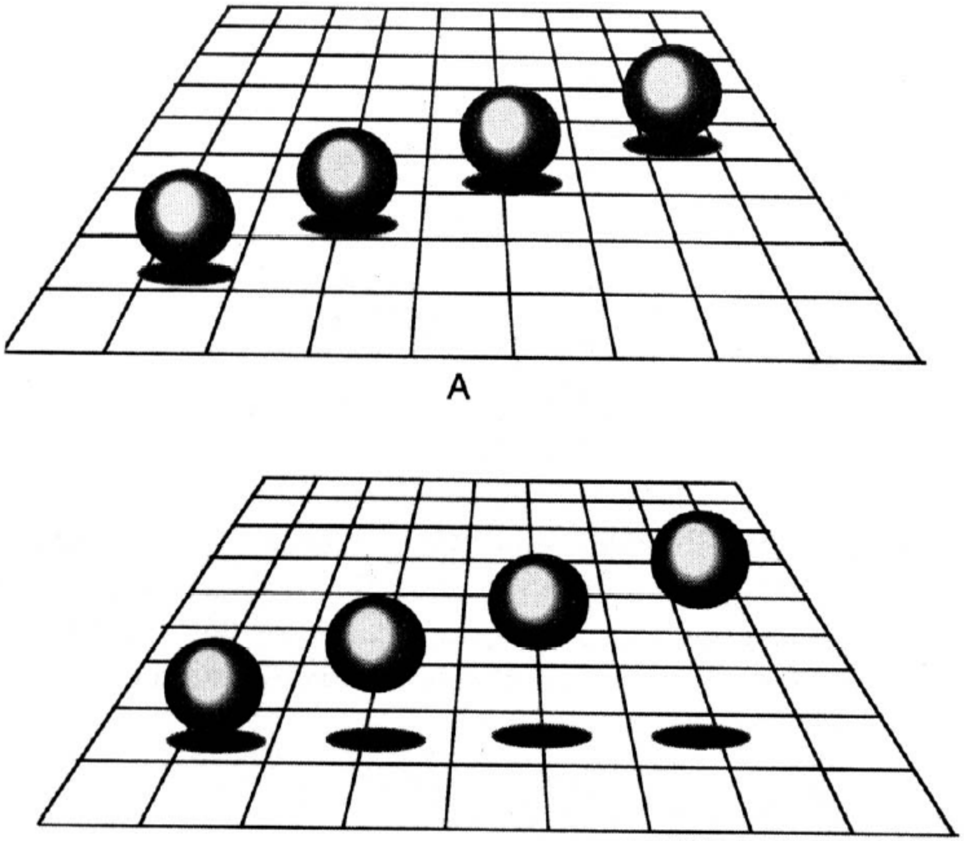
Why are Shadows Important?

- Depth cue
- Scene Lighting
- Realism
- Contact points



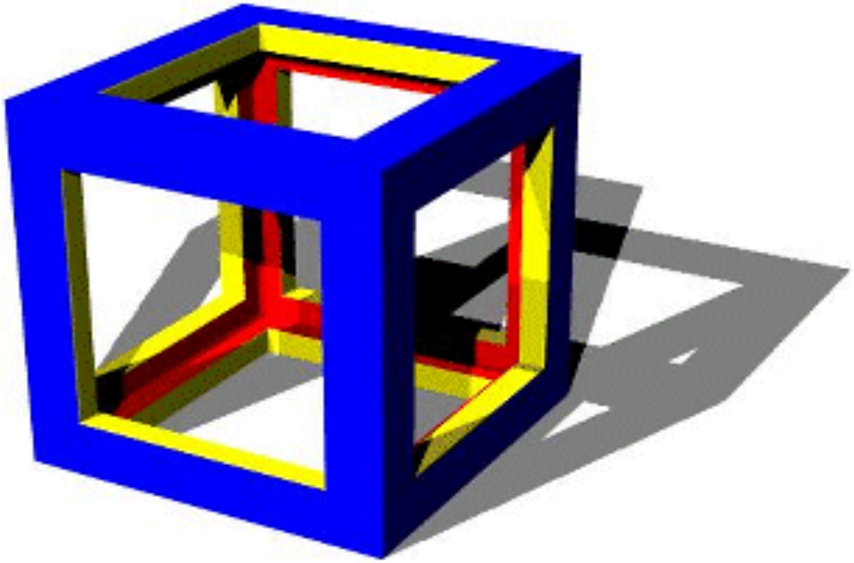
Shadows as a Depth Cue





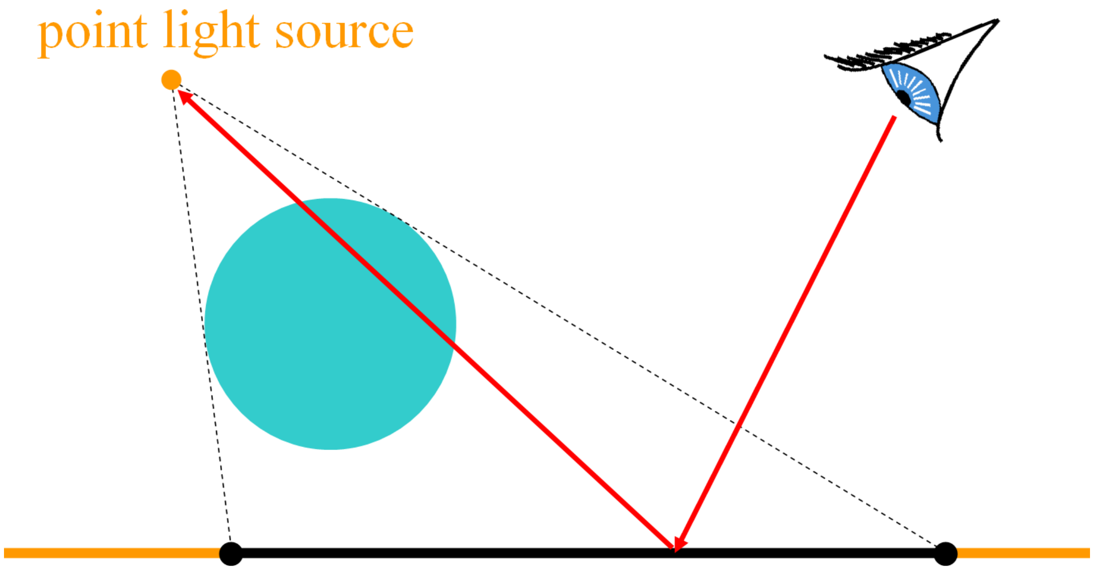
Today

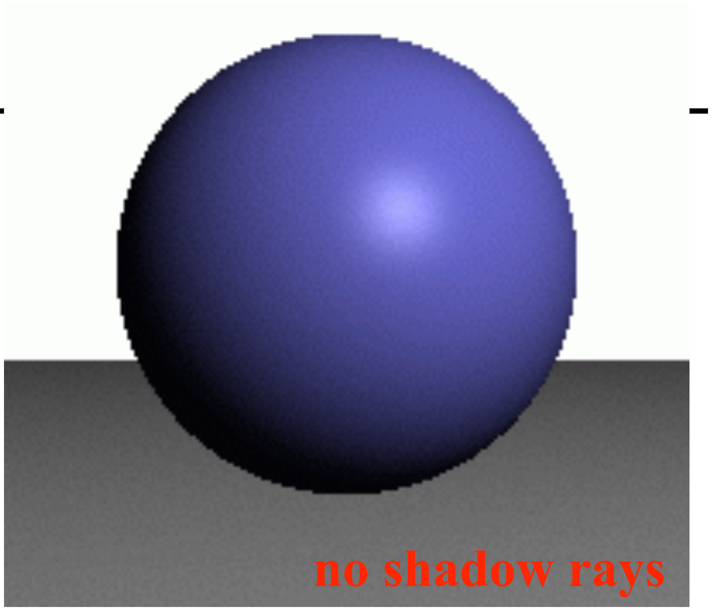
- Why are Shadows Important?
- Shadows & Soft Shadows in Ray Tracing
- Planar Shadows
- Shadow Maps
- Shadow Volumes

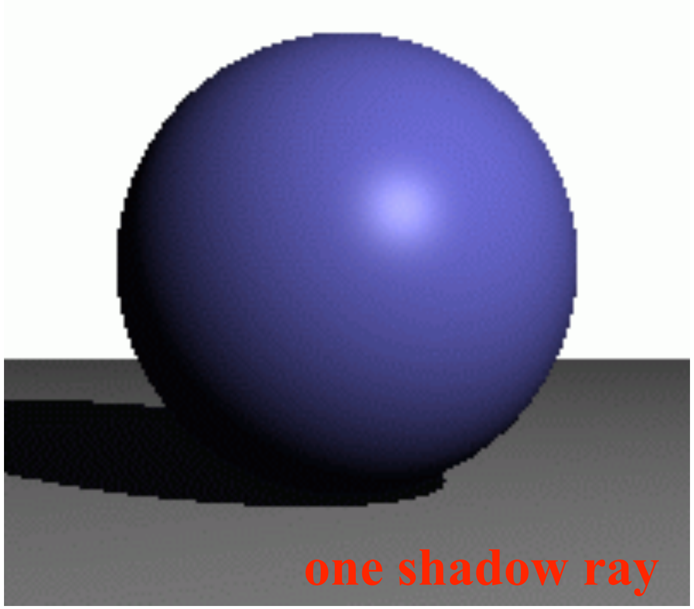


Shadows

- One shadow ray per intersection per point light source

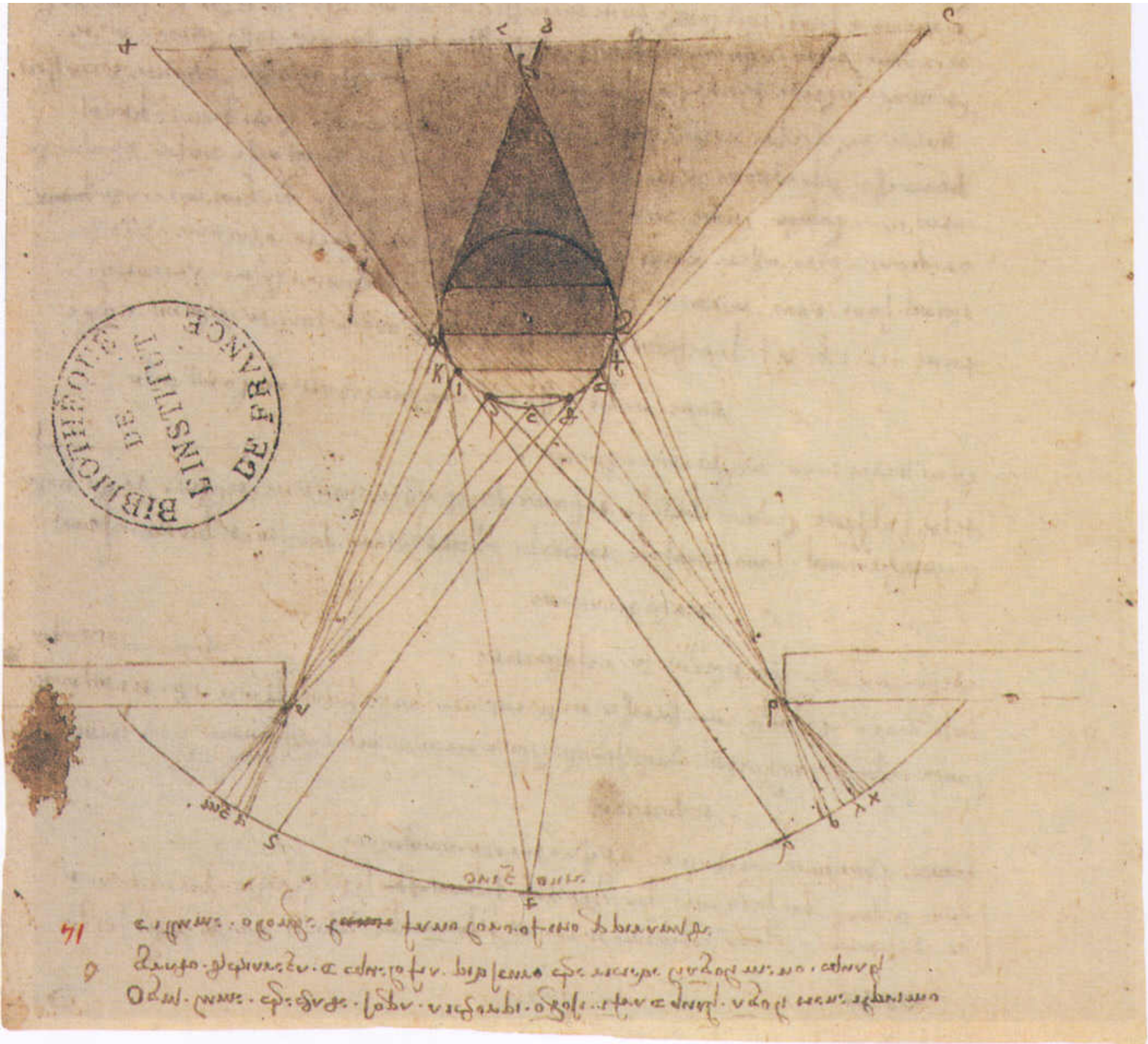






Soft Shadows

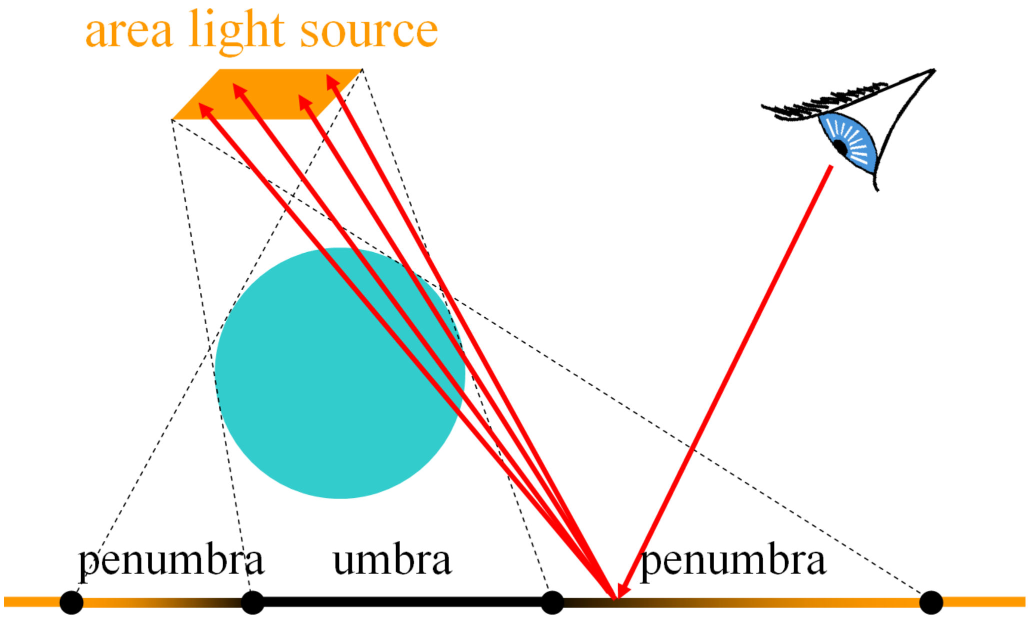
- Caused by extended light sources
- Umbra
  - source completely occluded
- Penumbra
  - Source partially occluded
- Fully lit

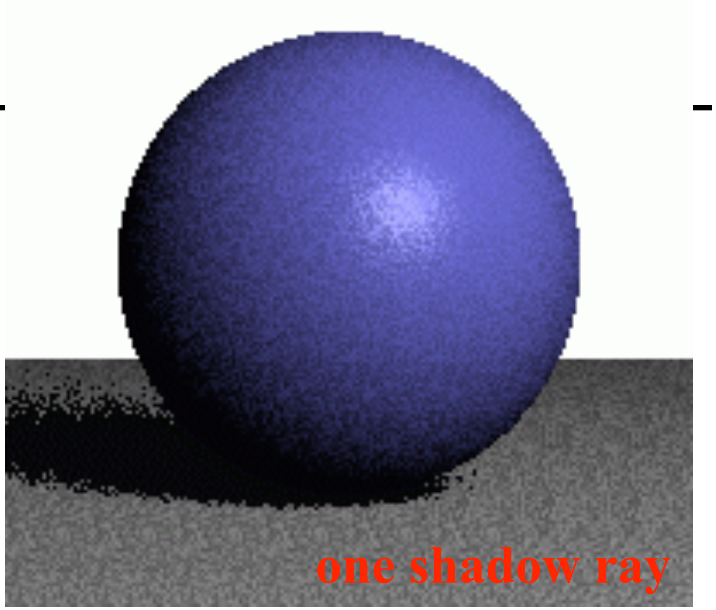


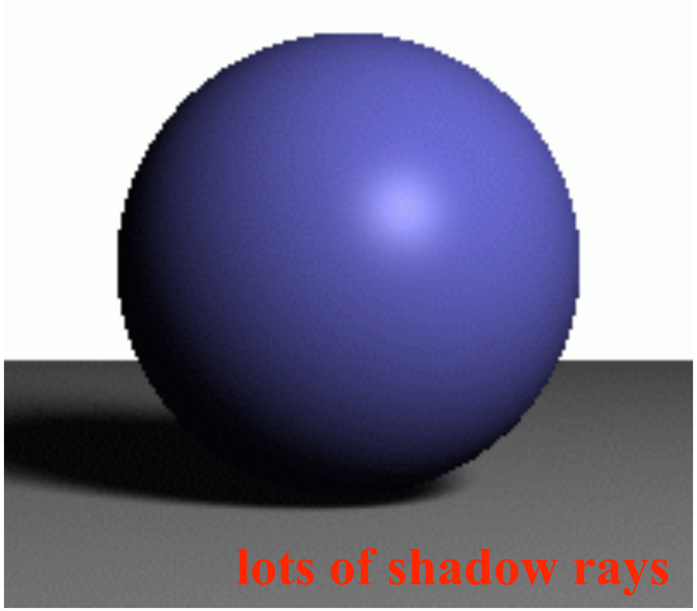
XVI. Léonard de Vinci (1452-1519). Lumière d'une fenêtre sur une sphère ombreuse avec (en partant du haut) ombre intermédiaire, primitive, dérivée et (sur la surface, en bas) portée. Plume et lavis sur pointe de métal sur papier, 24 x 38 cm. Paris, Bibliothèque de l'Institut de France (ms. 2185 ; B.N. 2038, f° 14 r°).

Soft Shadows

- Multiple shadow rays to sample area light source

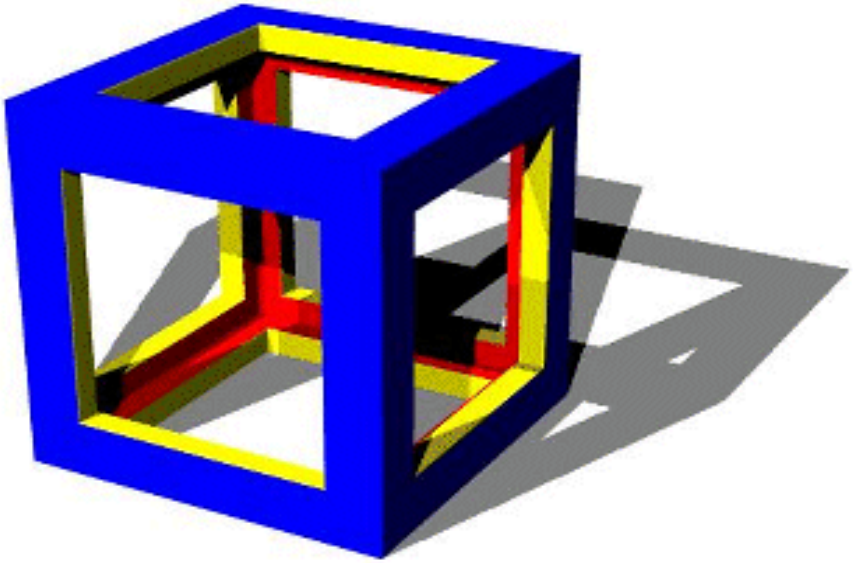






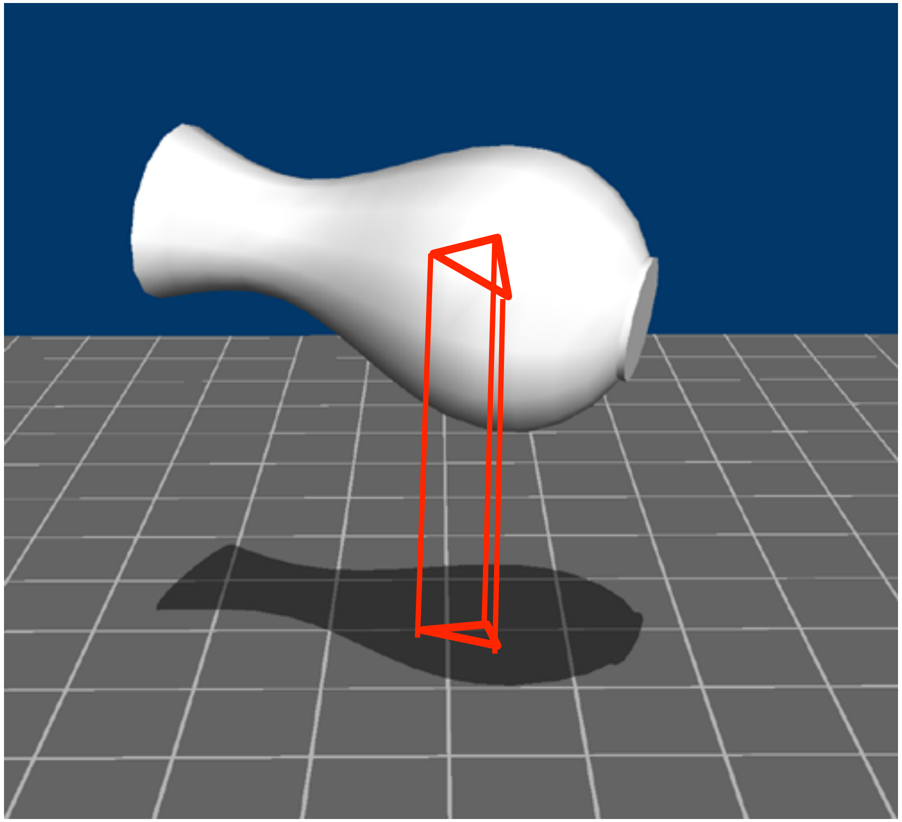
Today

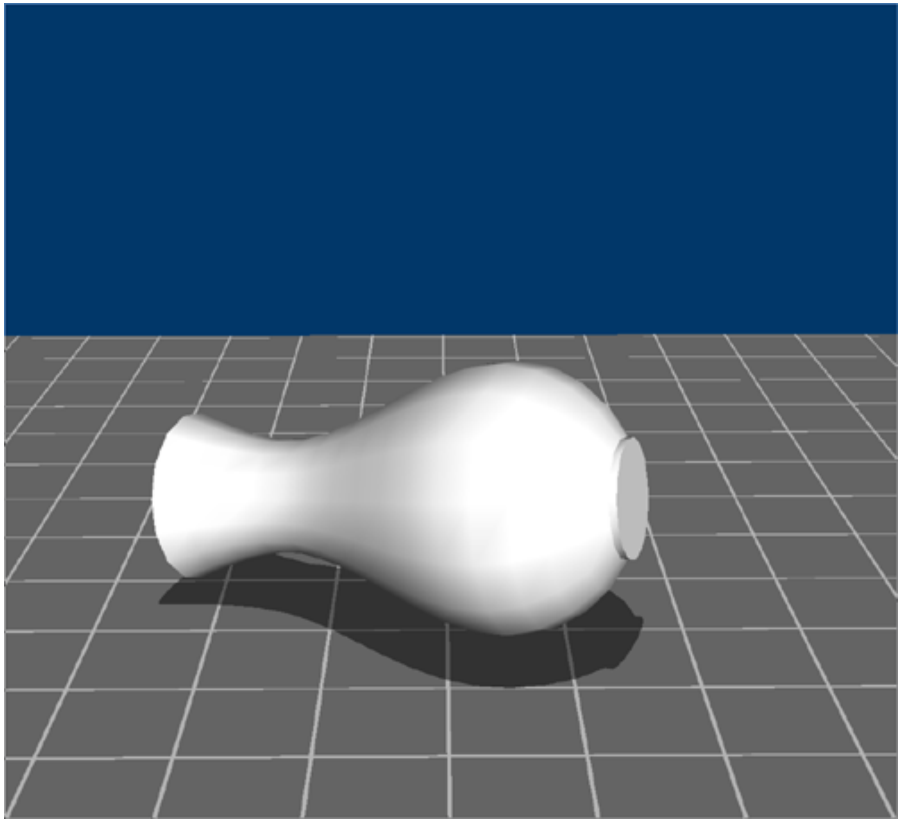
- Why are Shadows Important?
- Shadows & Soft Shadows in Ray Tracing
- Planar Shadows
- Shadow Maps
- Shadow Volumes



Cast Shadows on Planar Surfaces

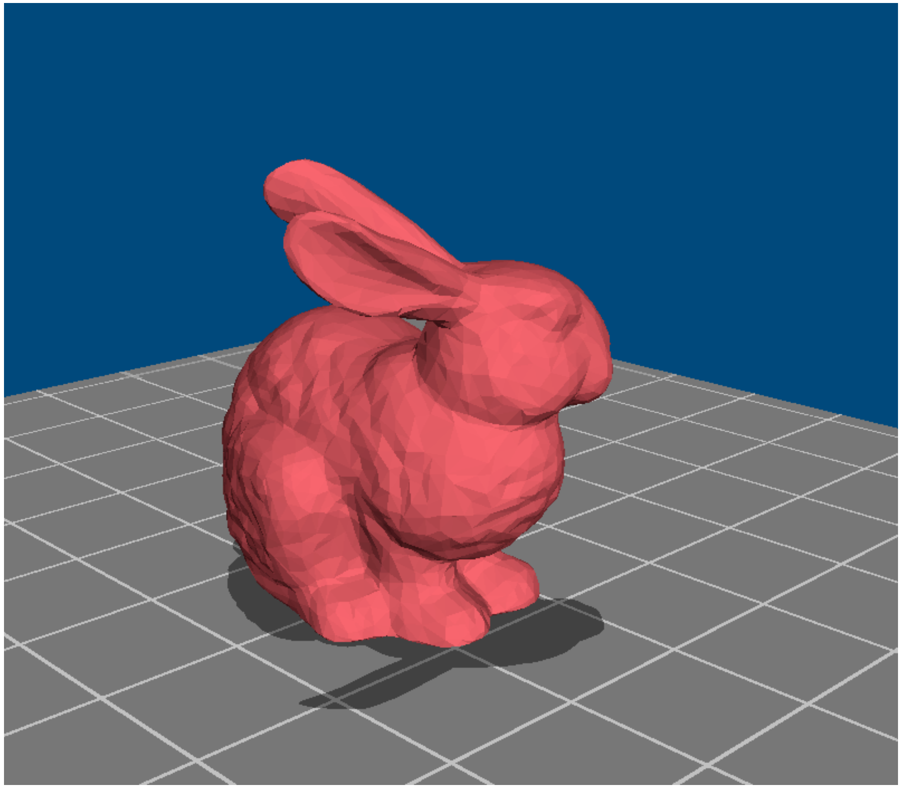
- Draw the object primitives a second time, projected to the ground plane, draw as dark translucent color

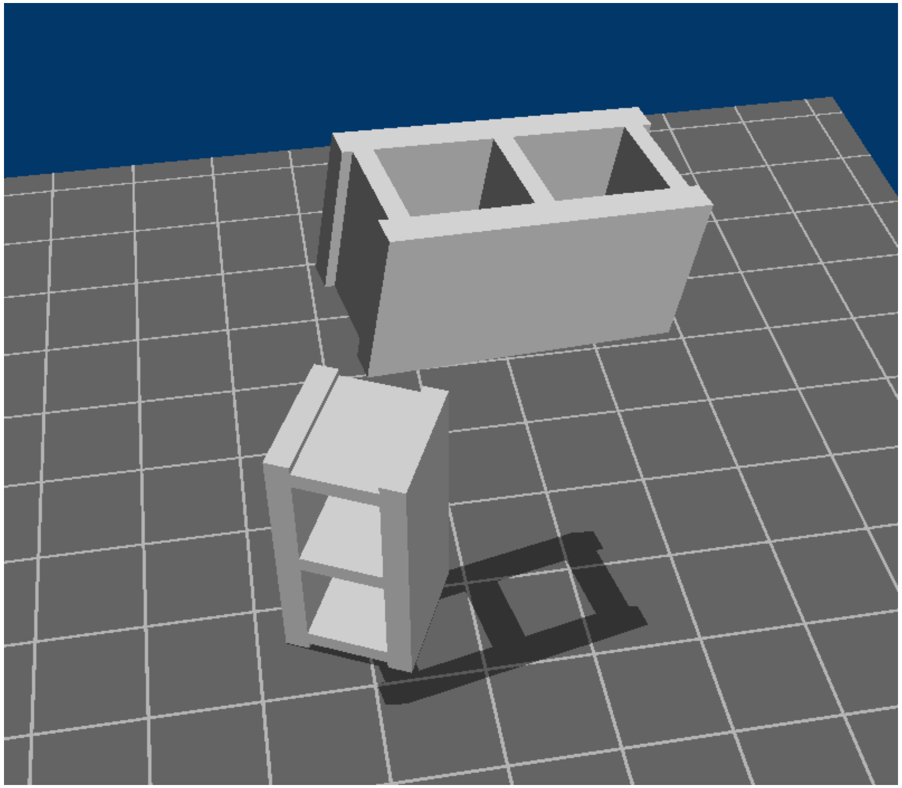




Limitations of Planar Shadows

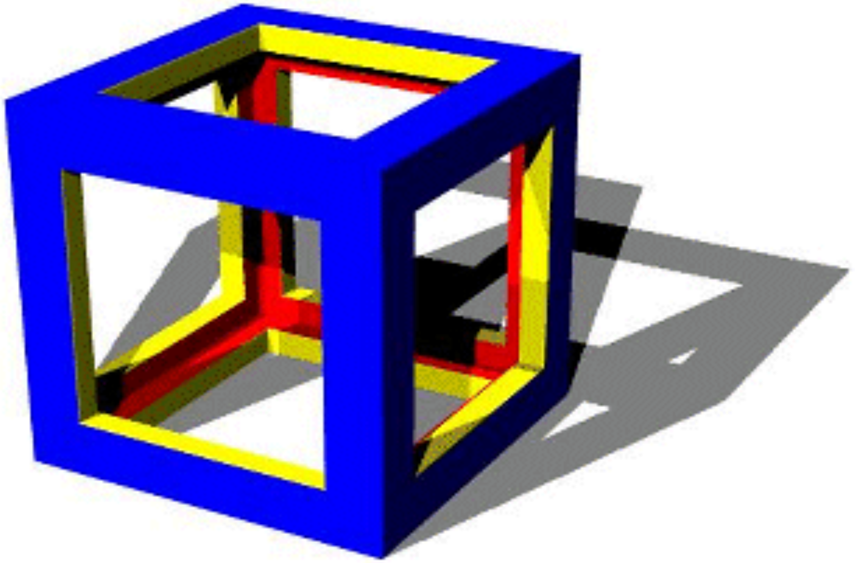
- Does not produce self-shadows, shadows cast on other objects, shadows on curved surfaces, etc.





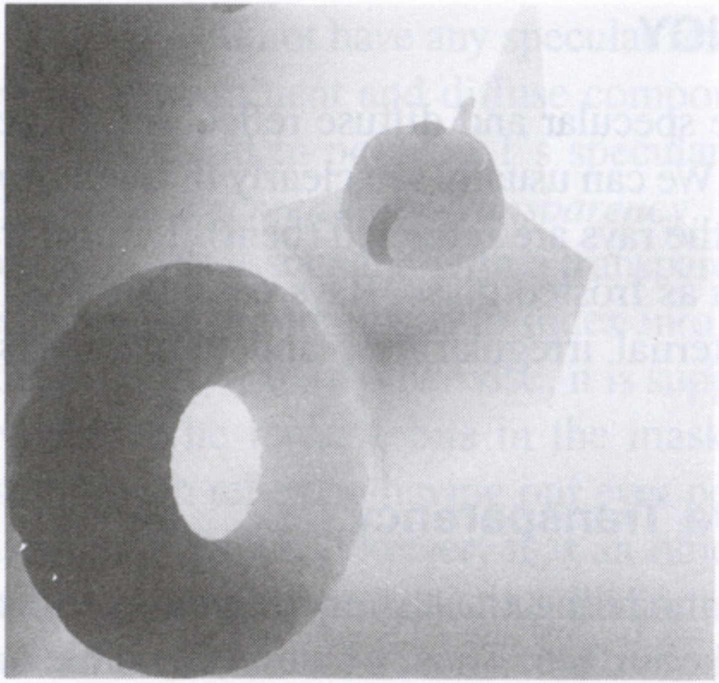
Today

- Why are Shadows Important?
- Shadows & Soft Shadows in Ray Tracing
- Planar Shadows
- Shadow Maps
  - Shadow/View Duality
  - Texture Mapping
- Shadow Volumes



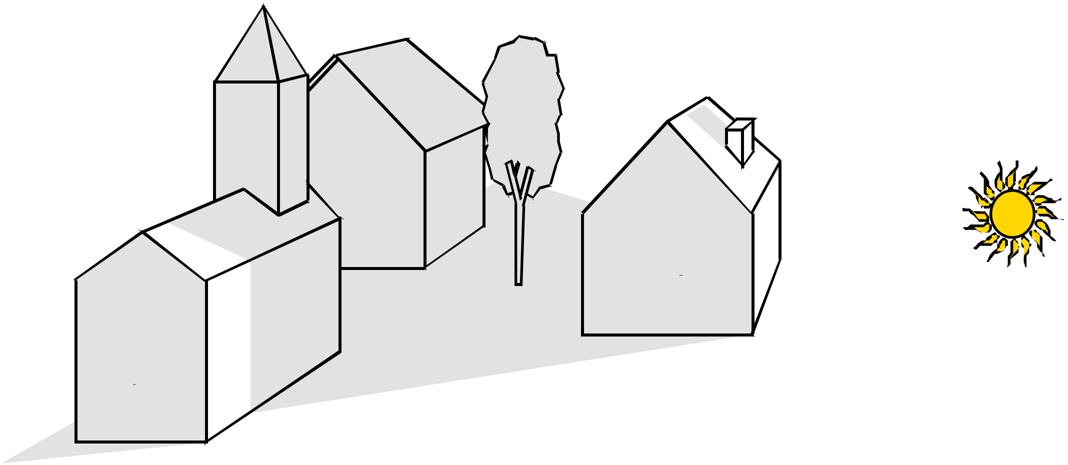
Shadow Maps Are(Were) Important

- Both in high-end production software...
  - Pixar’s RenderMan
- ... and in real-time rendering (e.g. games)

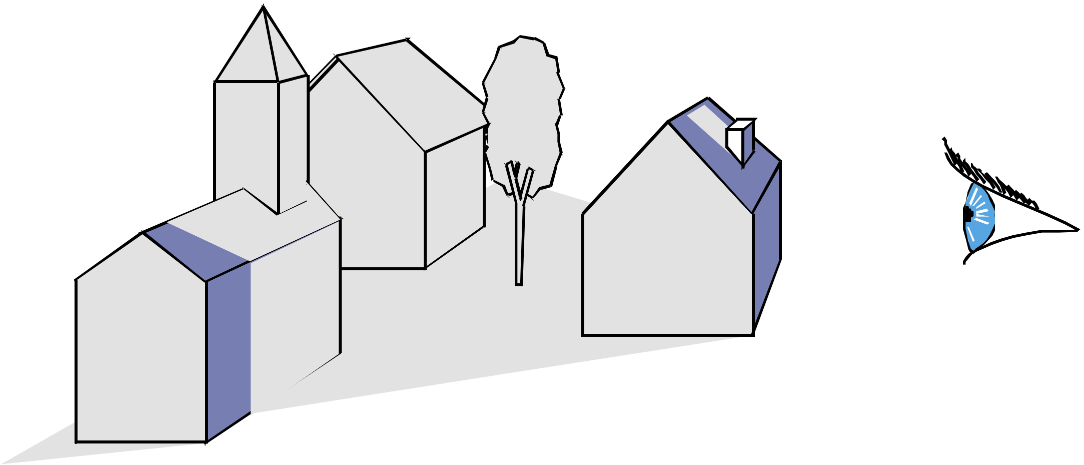


Shadow/View Duality

- A point is lit if it is visible from the light source

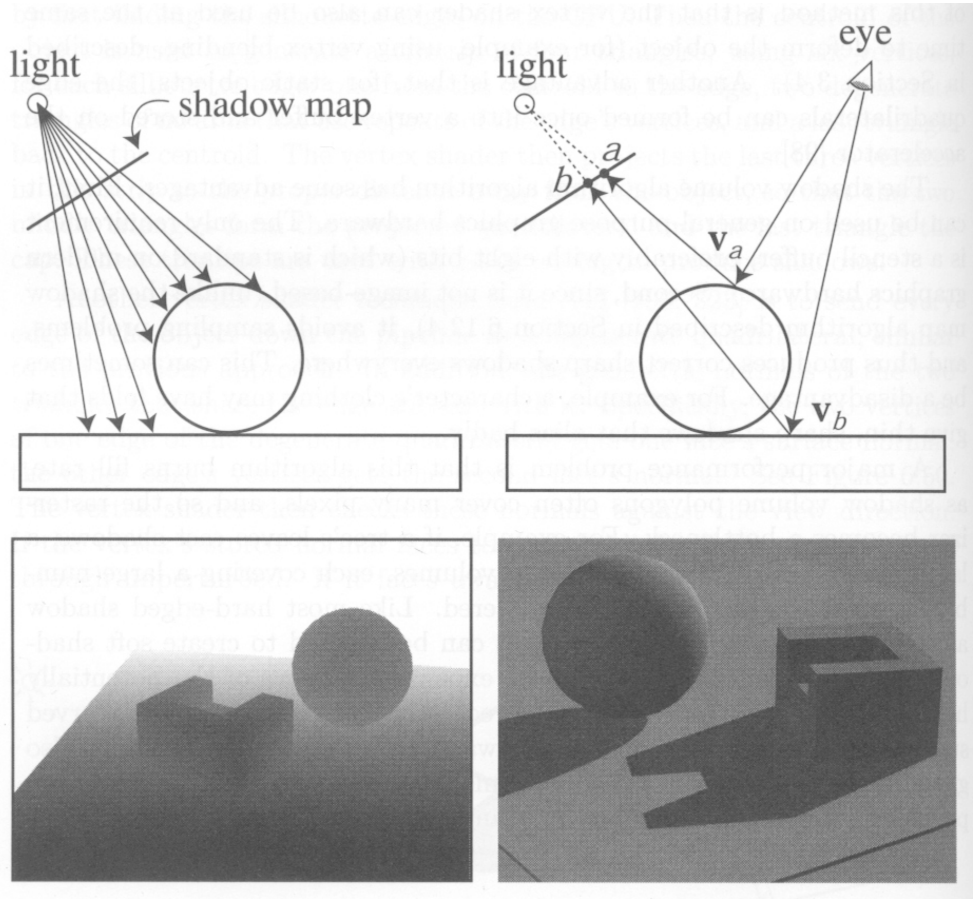


- Shadow computation similar to view computation



Shadow Mapping (=”shadow Z-buffer”)

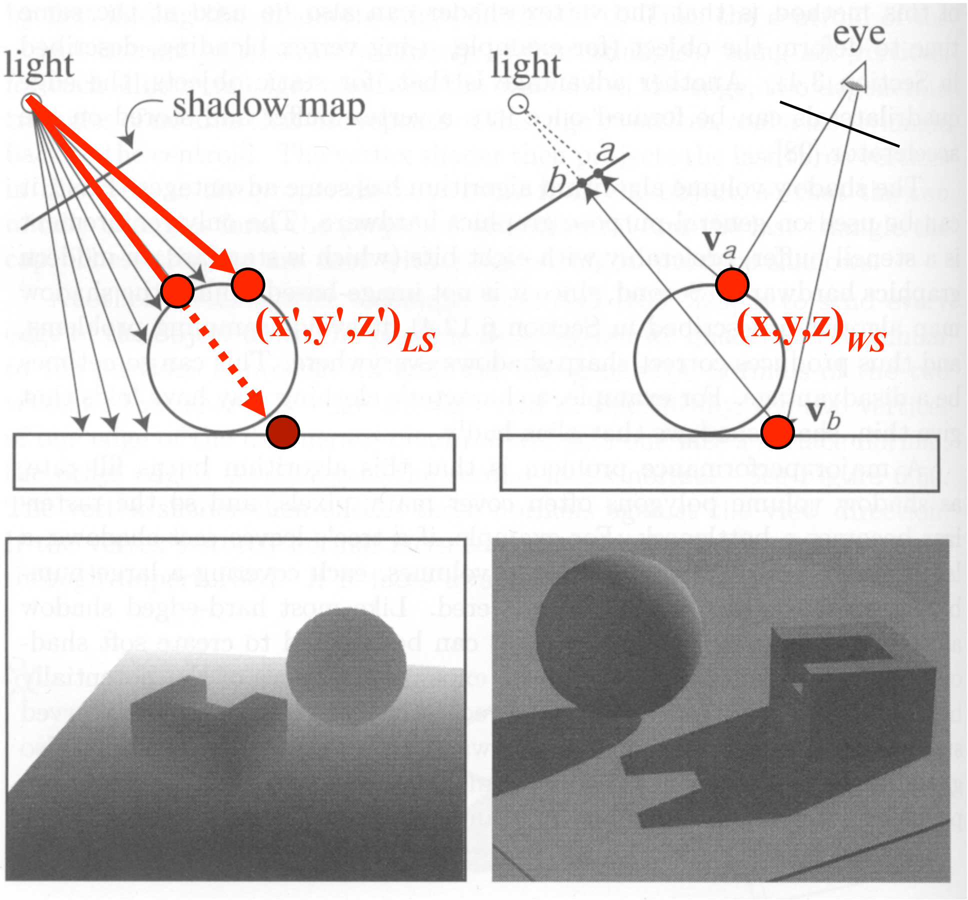
- Texture mapping with depth information
- 2 passes
  - Compute shadow map == depth from light source
    - You can think of it as a z-buffer as seen from the light
  - Render final image, check shadow map to see if points are in shadow



Foley et al. “Computer Graphics Principles and Practice”

Shadow Map Look Up

- We have a 3D point $(x,y,z)_{WS}$
- How do we look up the depth from the shadow map?
- Use the 4x4 perspective projection matrix from the light source to get $(x',y',z')_{LS}$
- ShadowMap(x',y') < z'?



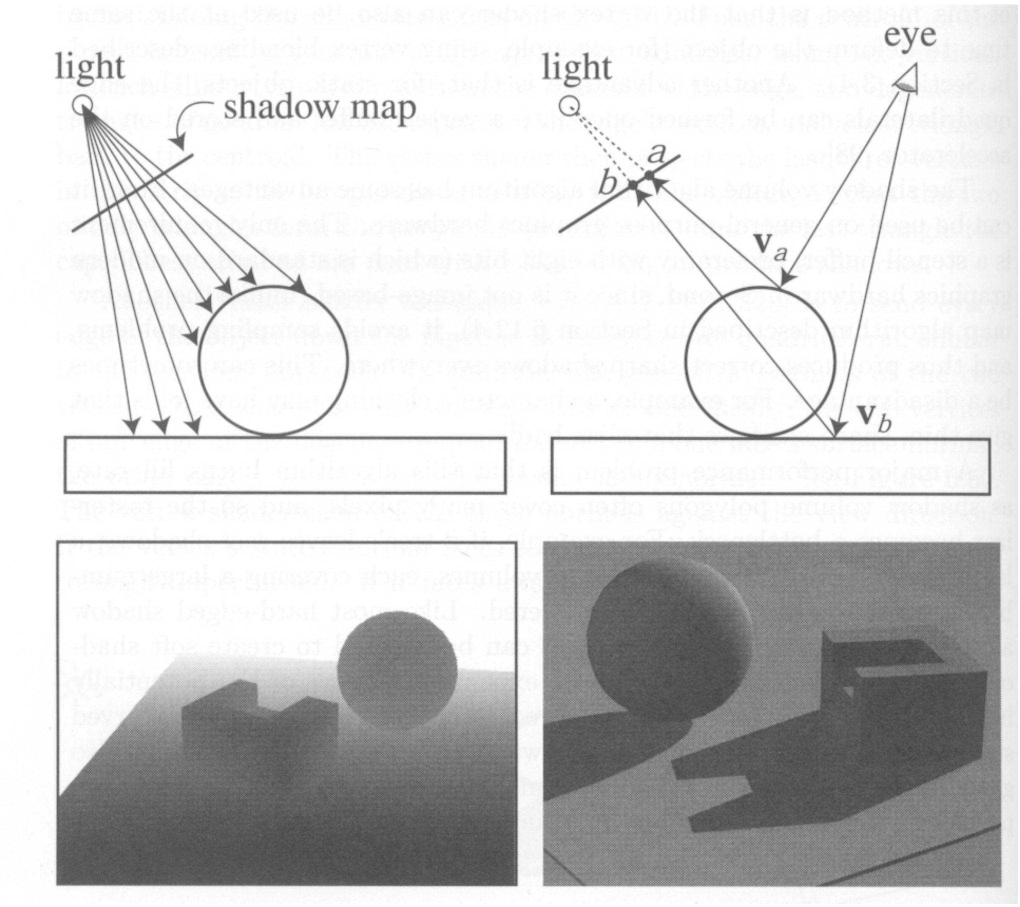
Foley et al. “Computer Graphics Principles and Practice”

Questions?

Plate 52 Grandville, *The Shadows (The French Cabinet)* from *La Caricature*, 1830.

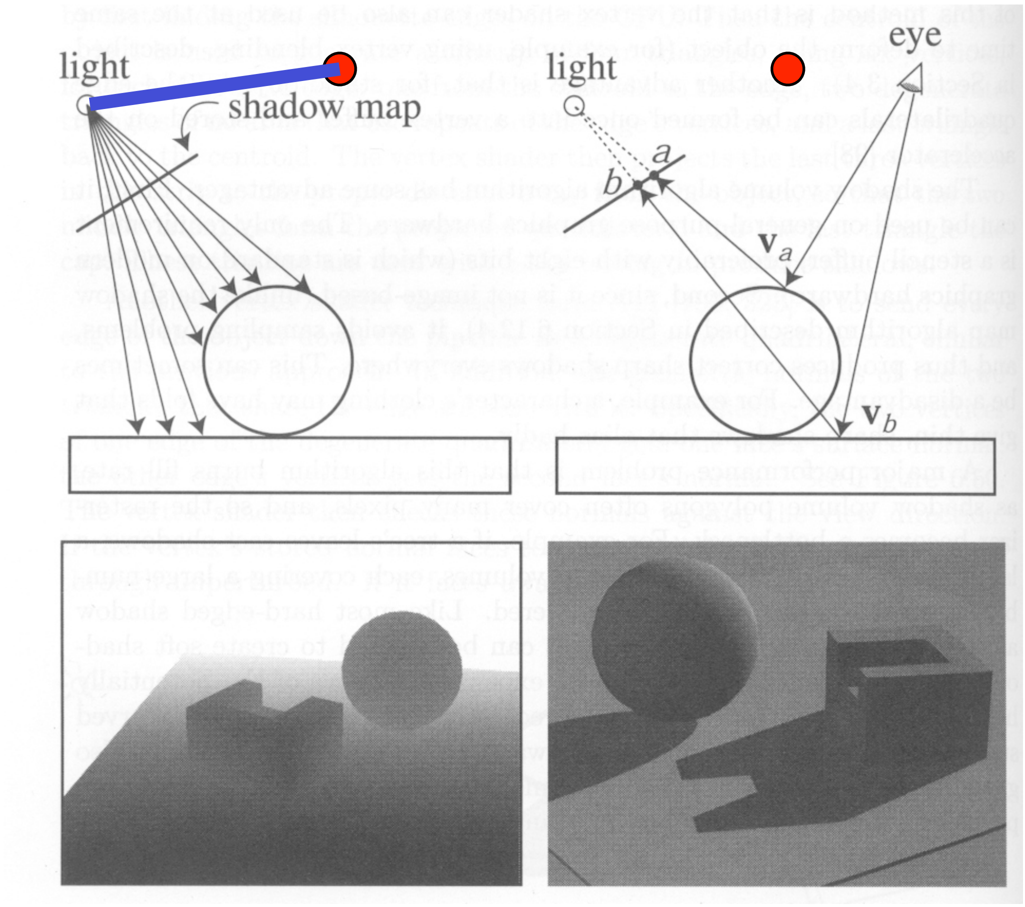
Limitations of Shadow Maps

1. Field of View
2. Bias (Epsilon)
3. Aliasing



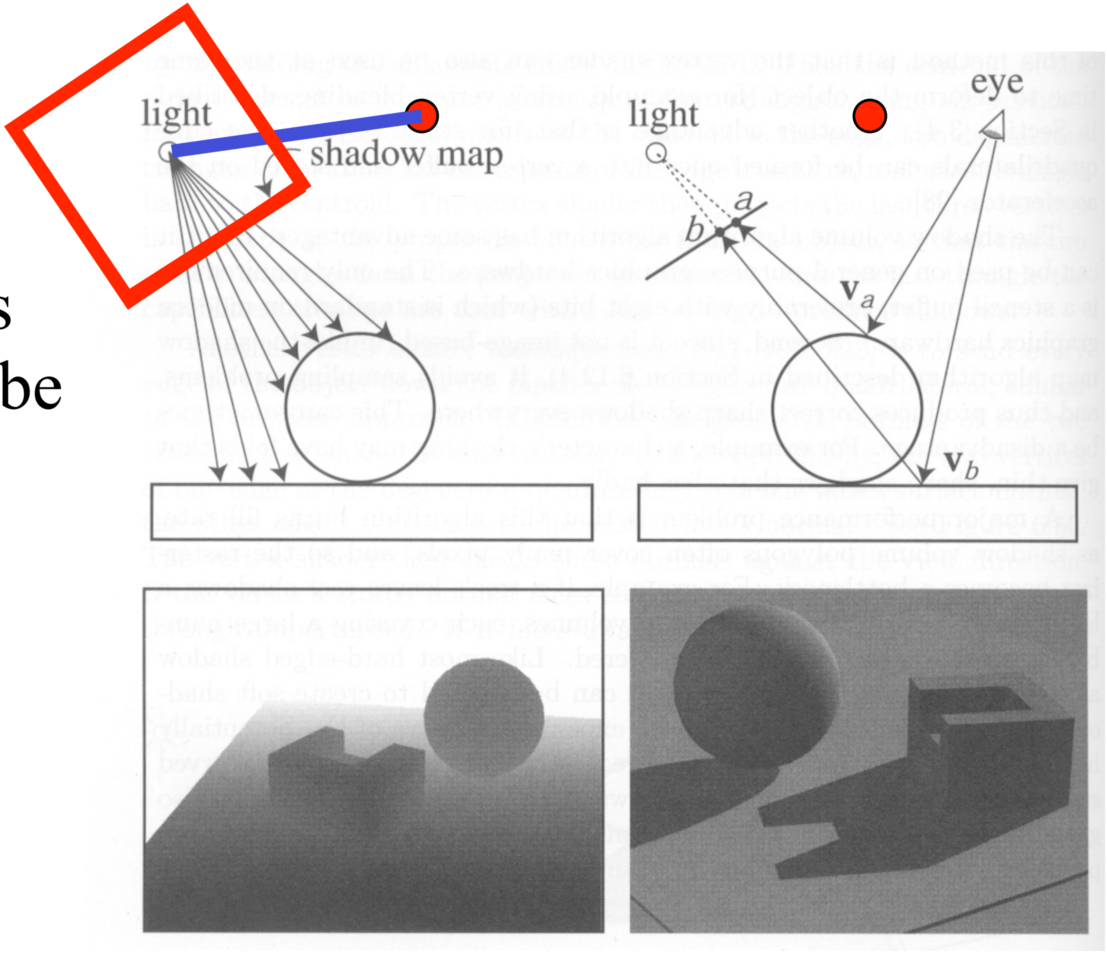
1. Field of View Problem

- What if point to shadow is outside field of view of shadow map?



1. Field of View Problem

- What if point to shadow is outside field of view of shadow map?
  - Use 6 shadow maps on the faces of a cube
  - ... or use only spot lights O:-)



2. The Bias (Epsilon) Nightmare

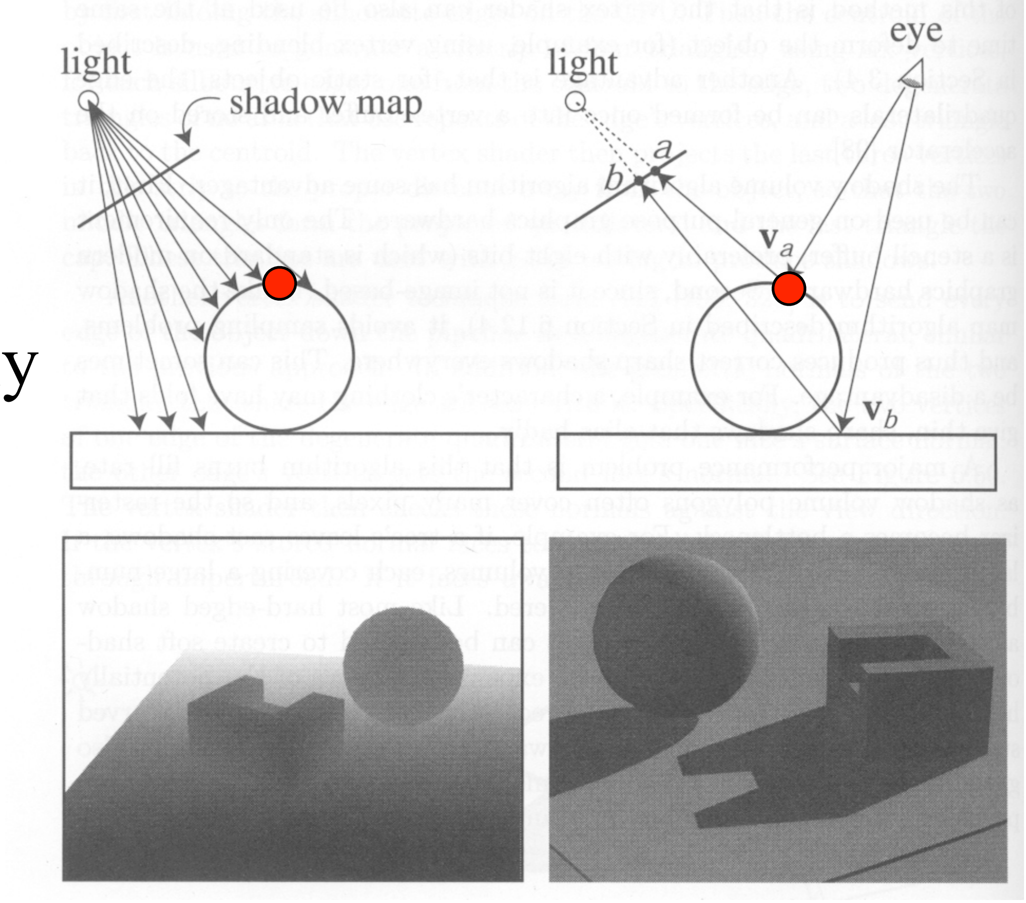
- For a point visible from the light source

  ShadowMap(x’,y’) ≈ z’

  – But due to rounding errors the depths never agree exactly

- How can we avoid erroneous self-shadowing?

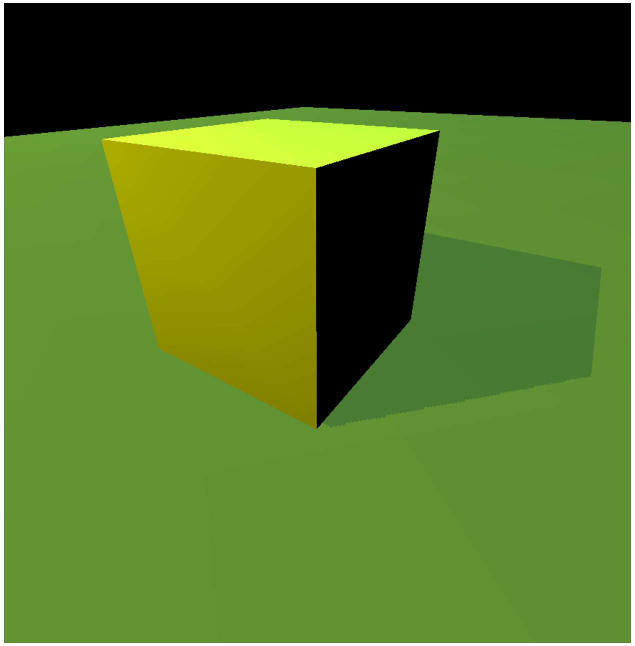
  – Add bias (epsilon)



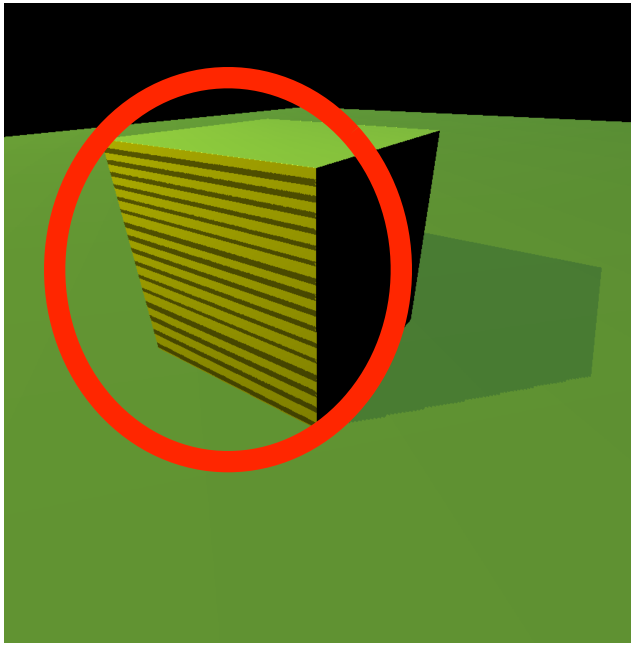
2. Bias (Epsilon) for Shadow Maps

ShadowMap(x’,y’) + bias < z’

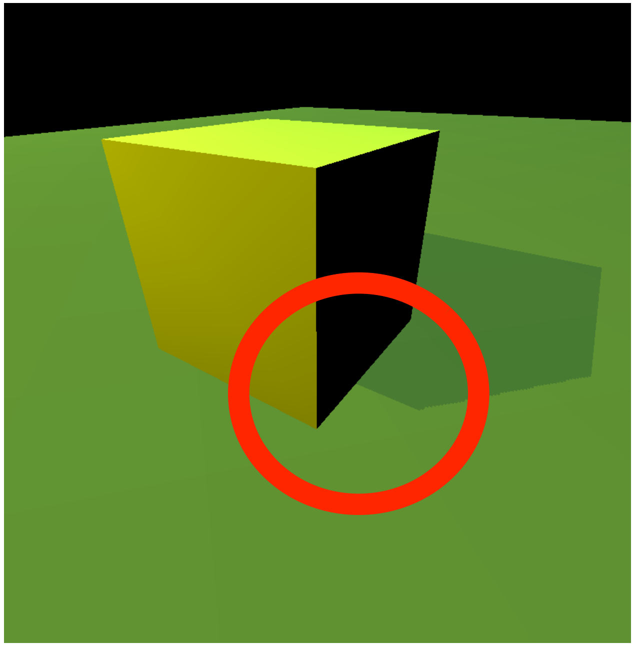
Choosing a good bias value can be very tricky



Correct image



Not enough bias (“surface acne”)

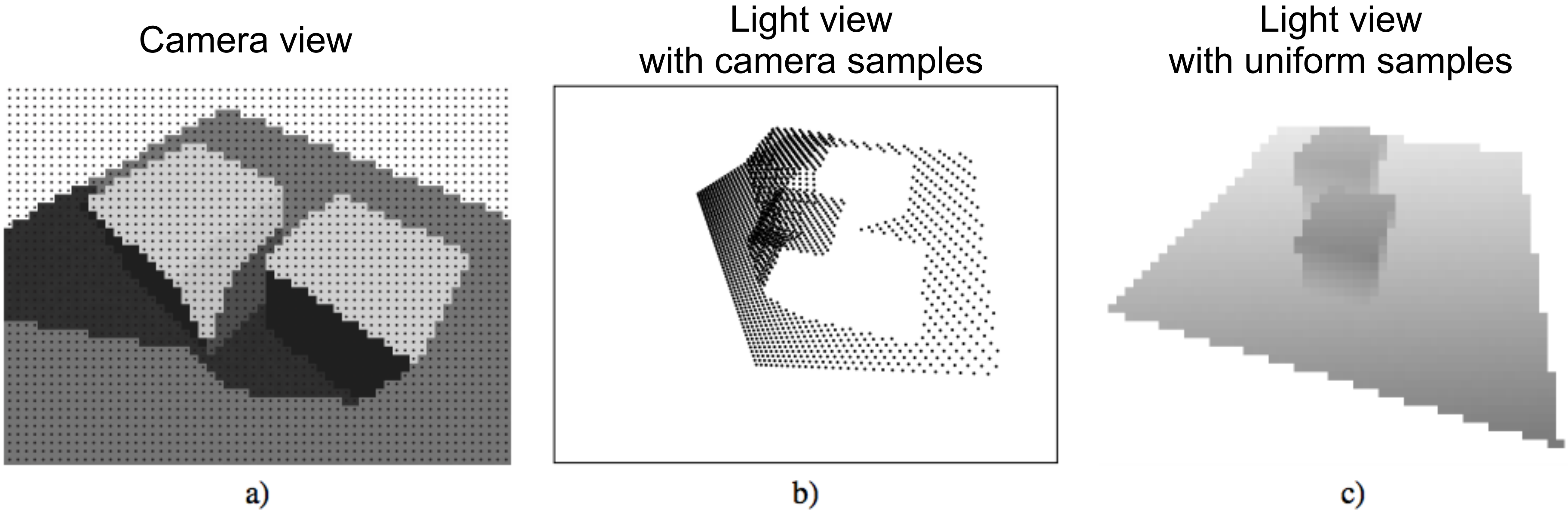


Way too much bias

Fixes

- "Depth slope bias"
  - Works in real time, doesn't fix all issues
- "Alias-free Shadow Maps"
  - Aila & Laine 2004
  - Transform all view samples into light space, perform non-uniform rasterization at these points to find depth precisely in the correct locations
  - Also called "irregular Z-buffer"
  - Incidentally: gives precisely the same result as a ray tracer. It's *really* just a different ordering of the loops :)

Alias-Free Shadow Maps



Camera view

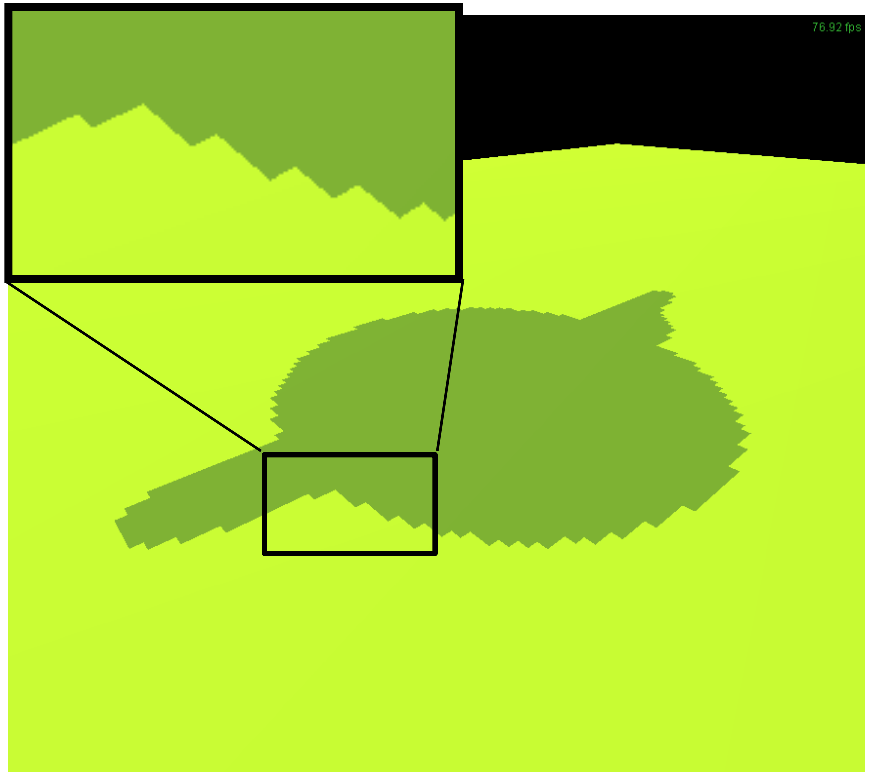
Light view
with camera samples

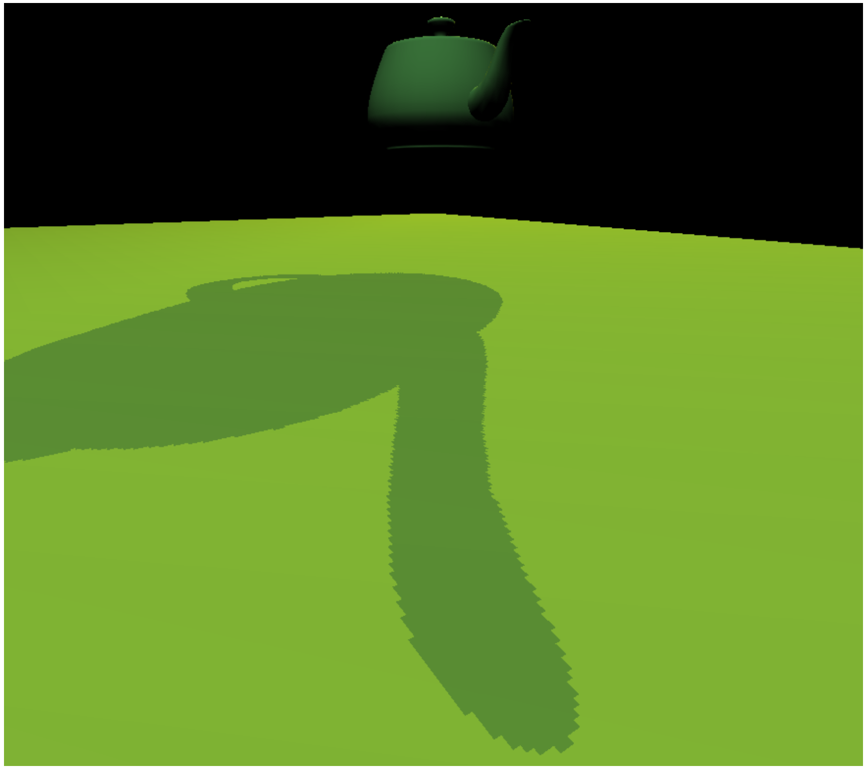
Light view
with uniform samples

Figure 2: *a) A simple test scene with shadows, as seen from the camera. The black dots are the pixel centers. b) The visible pixels of (a) transformed into the image plane of the light source. The dots are used as sampling points when the scene is rasterized to the shadow map. The large empty areas correspond to regions that are not visible from the camera, and thus need no shadow information. c) The corresponding traditional shadow map is shown for comparison purposes only. In a traditional shadow map algorithm, the regularly sampled map (c) would be tested exactly at the sampling points shown in (b). Clearly, the regular structure of (c) is not suitable for accurately answering the queries.*

3. Shadow Map Aliasing

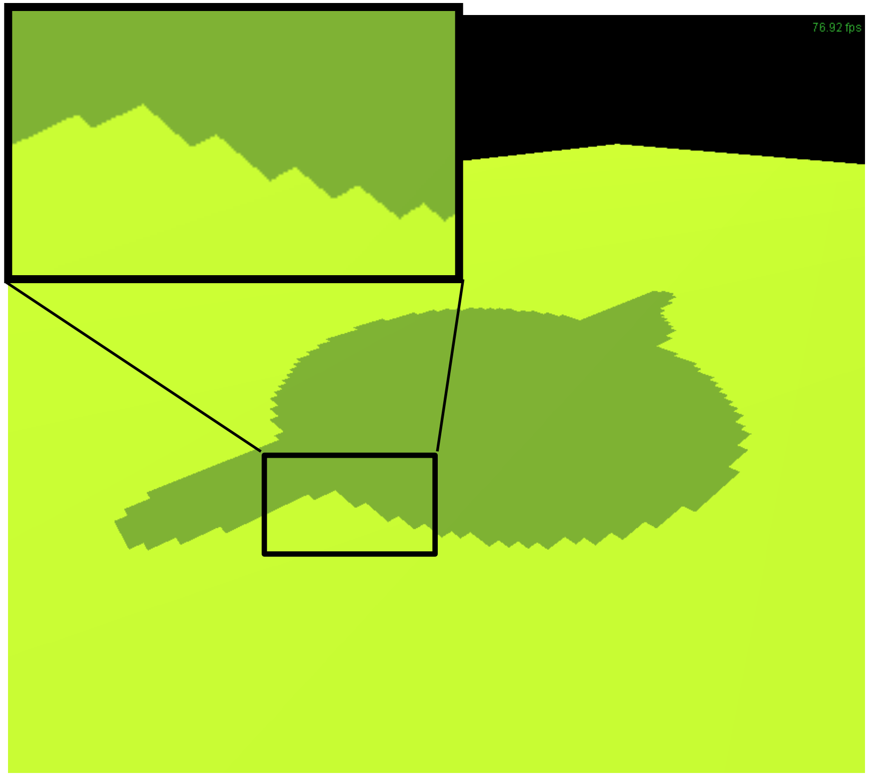
- Under-sampling of the shadow map
  - Jagged shadow edges

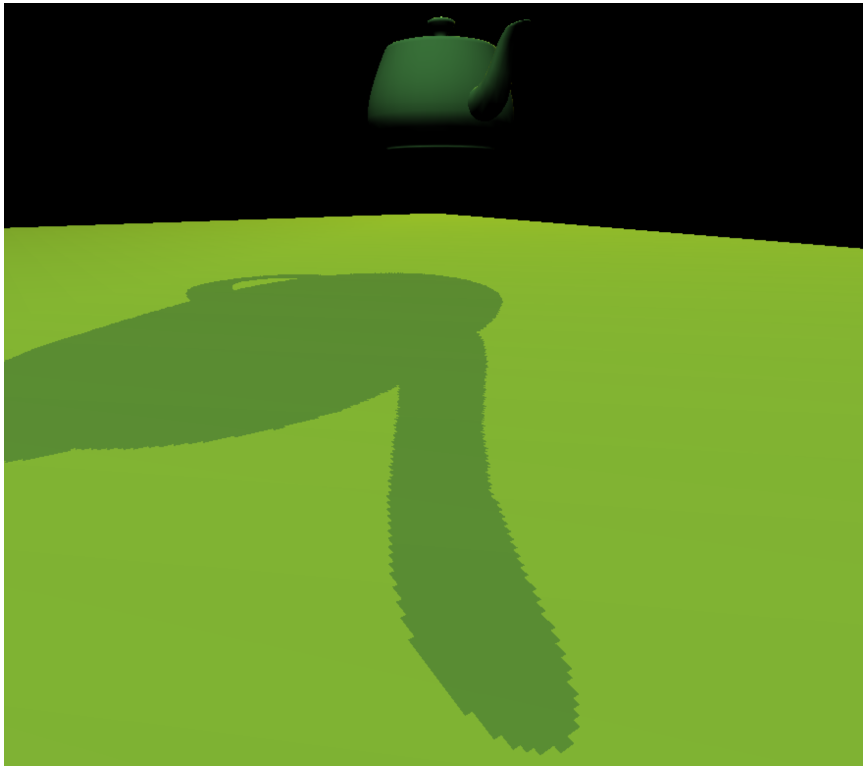




3. Shadow Map Aliasing

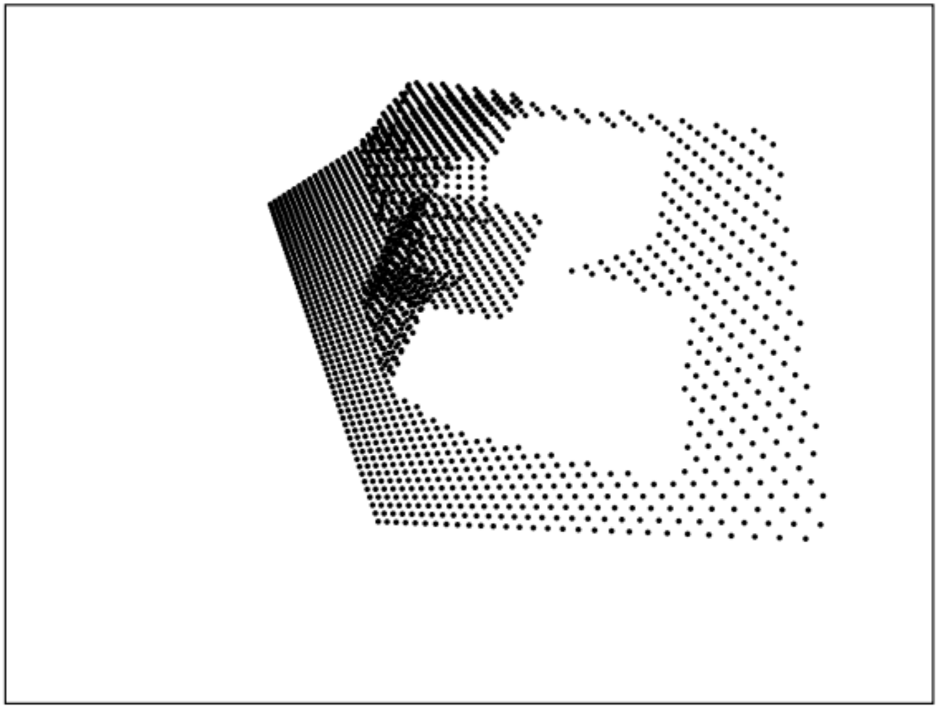
- Why is this?
  - The shadow map stores exact results for the rays that correspond to shadow map pixels... but not screen pixels!





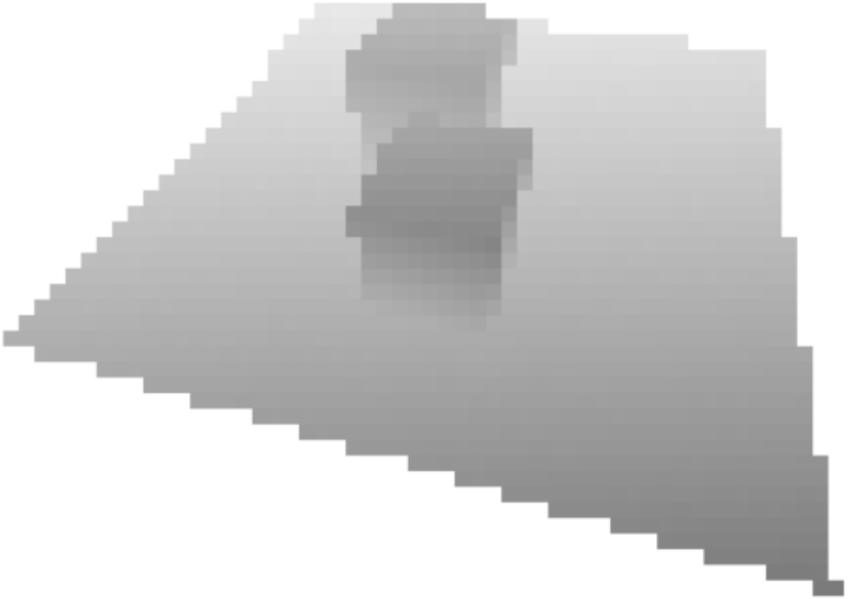
Resolution / Sample Mismatch

Light view
with camera samples



This is where we **want** to know
distance from light source

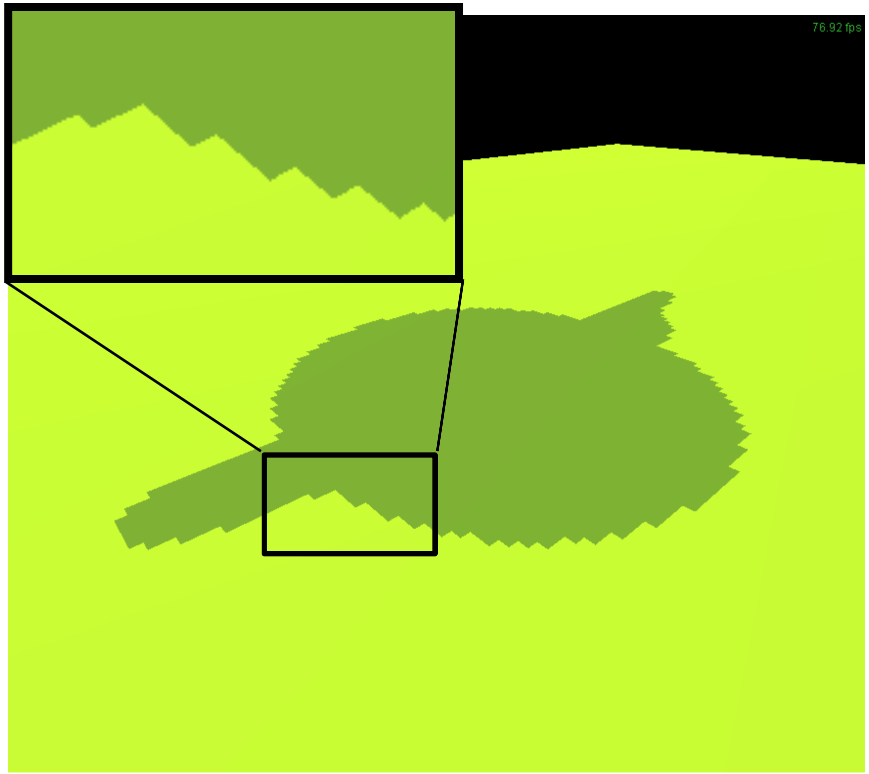
Light view
with uniform samples

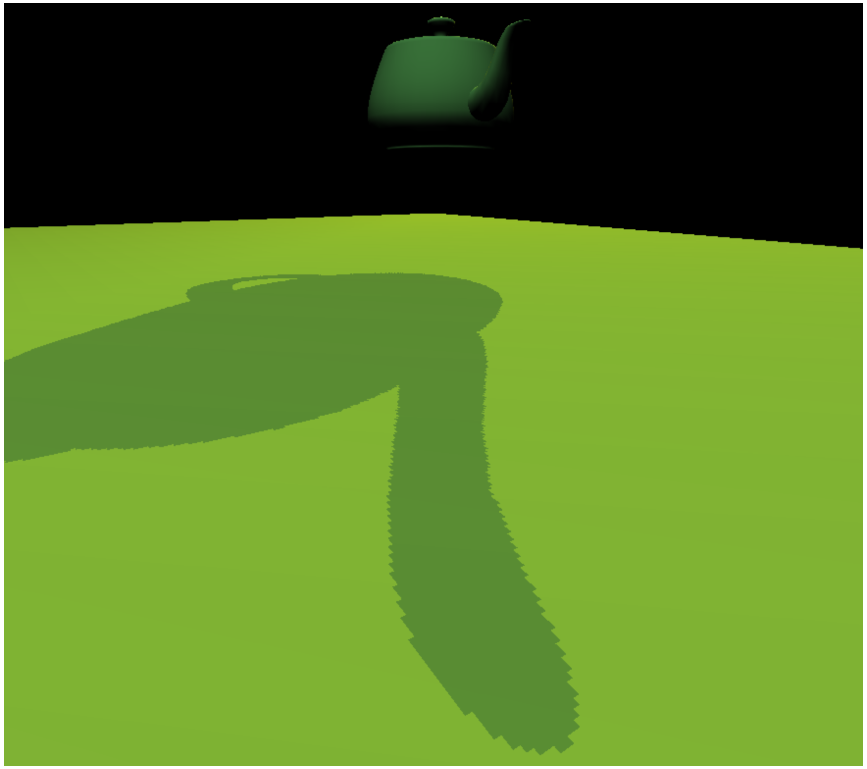


This is where the shadow map tells
us the distance from the source

3. Shadow Map Aliasing

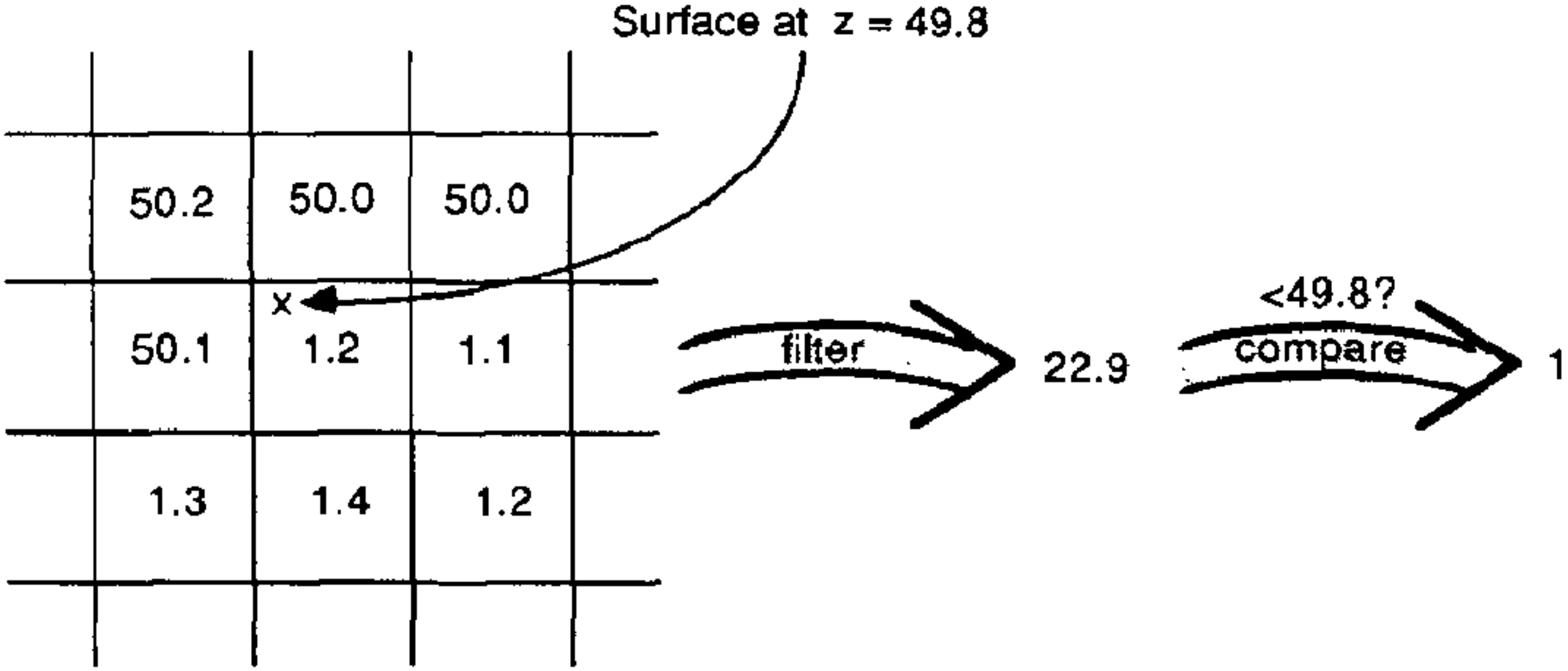
- Can you think of ways to get around this?





3. Shadow Map Filtering

- Should we filter the depth like a texture? (weighted average of neighboring depth values)
- No... filtering depth is not meaningful

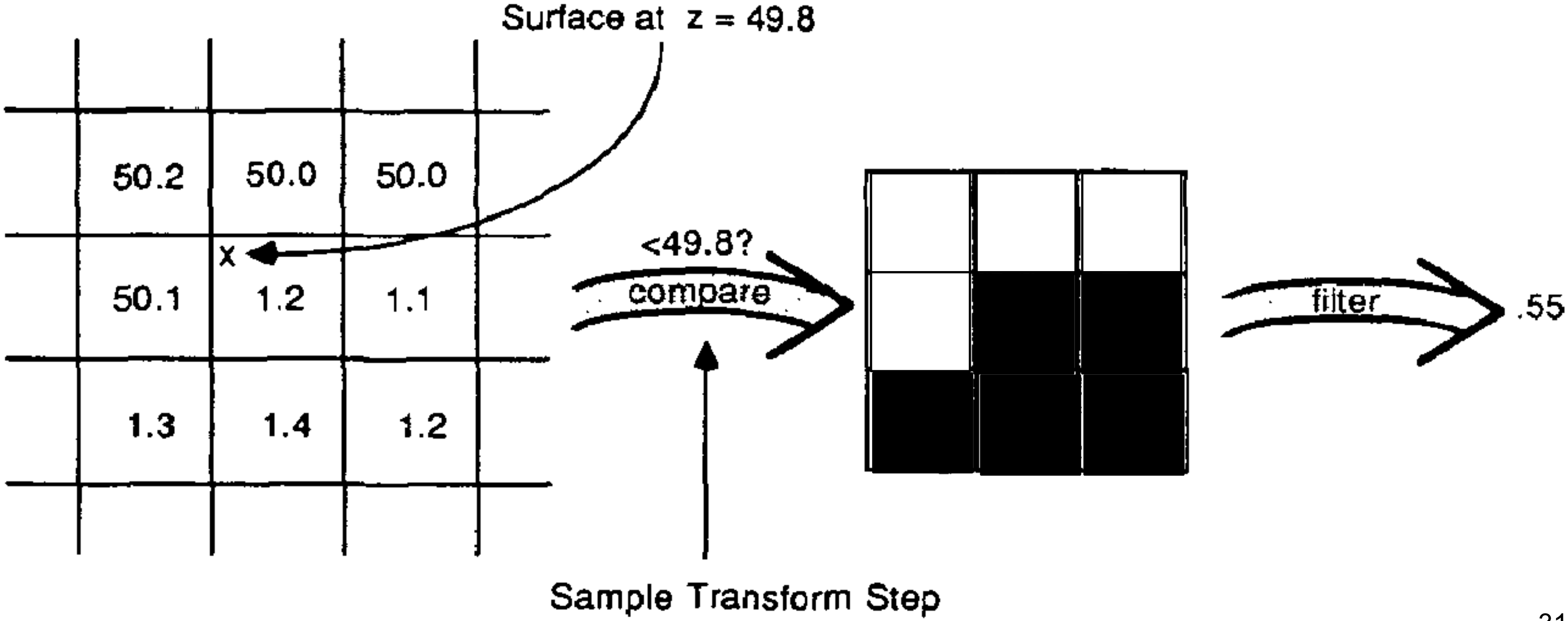


50.2	50.0	50.0
50.1	1.2	1.1
1.3	1.4	1.2

a) Ordinary texture map filtering. Does not work for depth maps.

3. Percentage Closer Filtering

- Instead we need to filter the *result* of the shadow test (weighted average of comparison results)

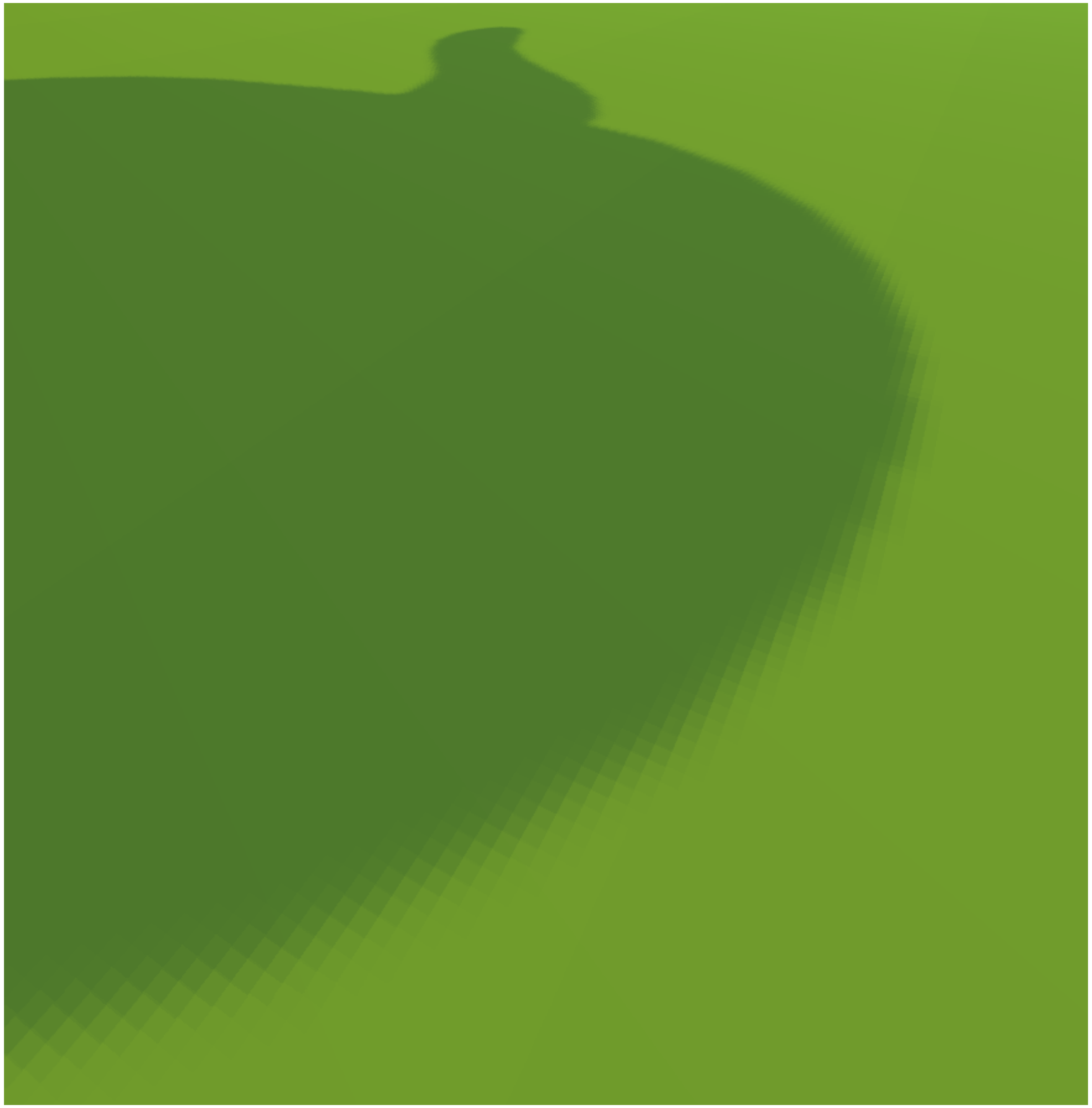


50.2	50.0	50.0
50.1	1.2	1.1
1.3	1.4	1.2

□	□	□
□	■	■
■	■	■

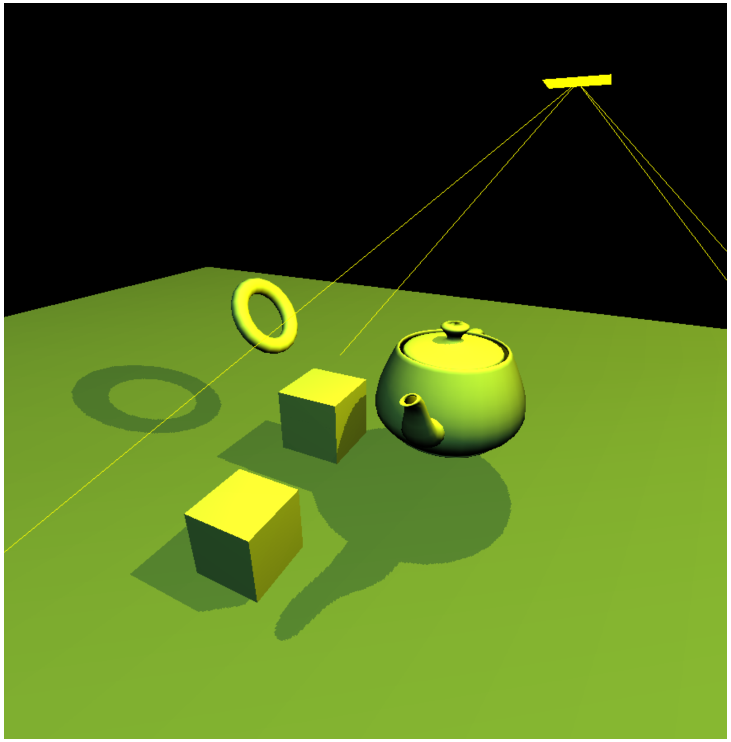
3. “Percentage Closer Filtering”

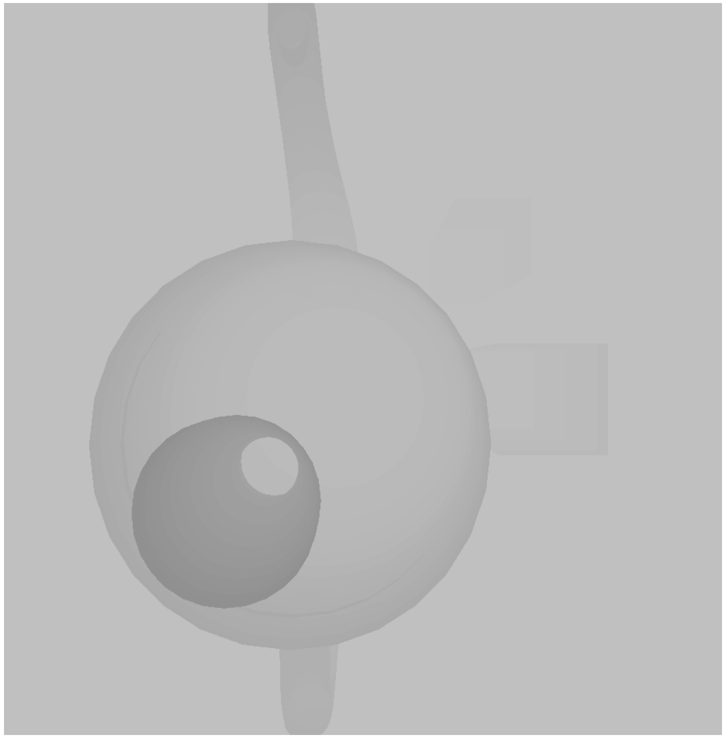
- 5x5 samples
- Nice antialiased shadow
- Using a bigger filter produces (fake) soft shadows
  - Why fake?
- Setting bias is tricky



Hardware Shadow Maps

- Can be done with hardware texture mapping
  - Texture coordinates u,v,w generated using 4x4 matrix
  - Modern hardware even does percentage closer filtering

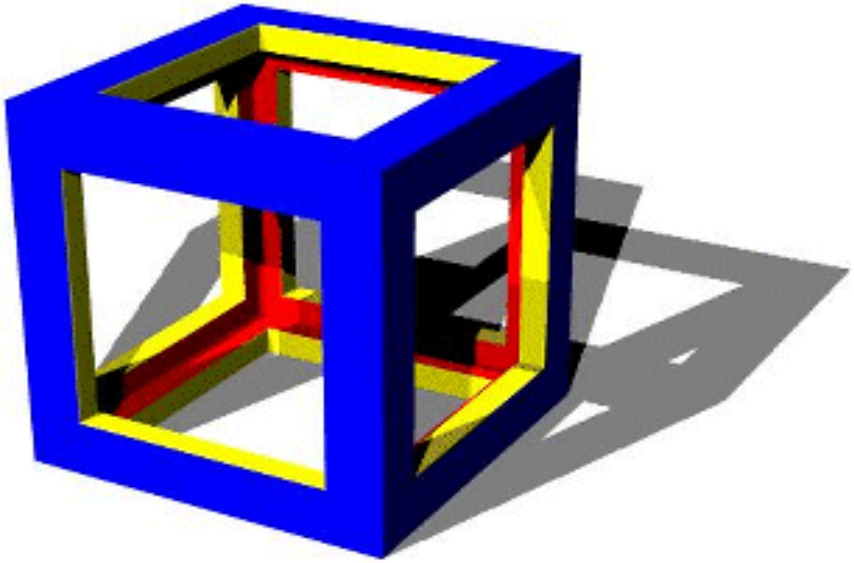




Questions?

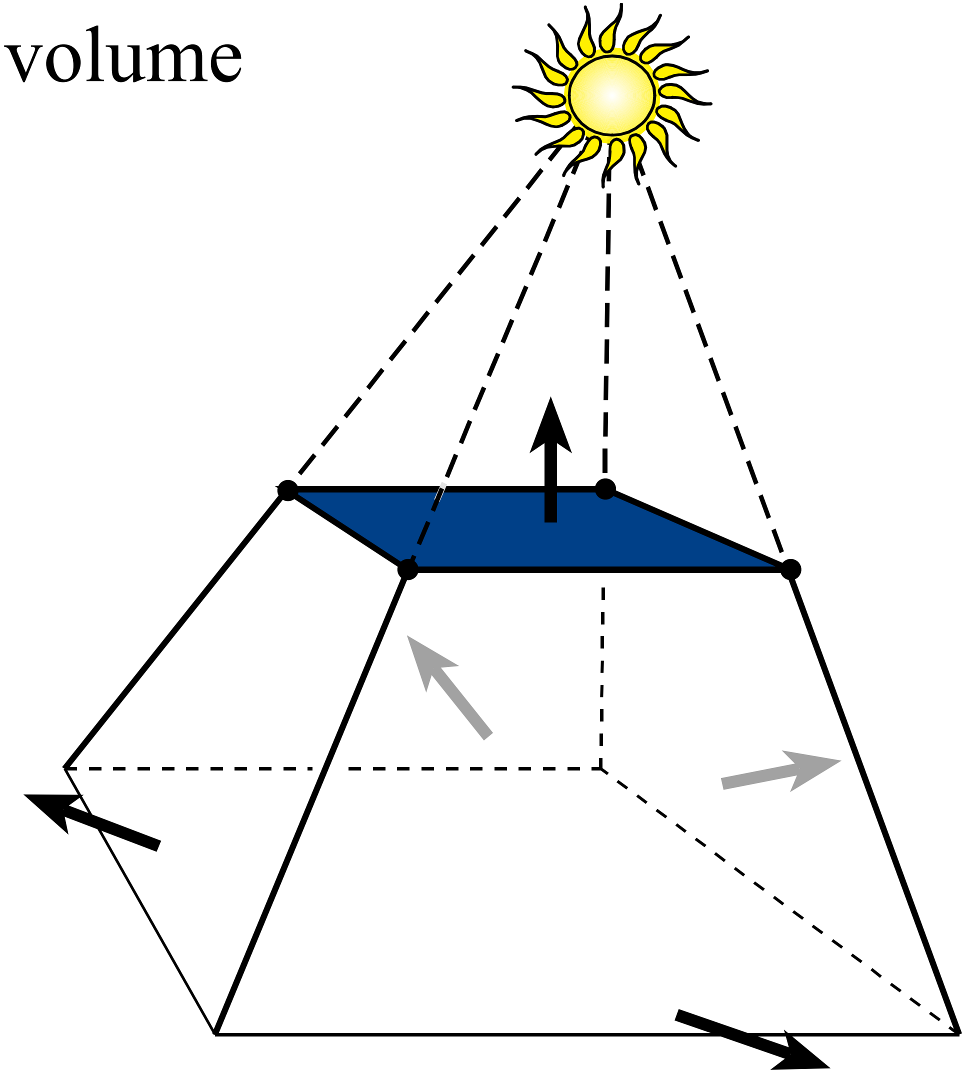
Today

- Why are Shadows Important?
- Shadows & Soft Shadows in Ray Tracing
- Planar Shadows
- Shadow Maps
- Shadow Volumes
  - (The Stencil Buffer)



Shadow Volumes

- Explicitly represent the volume of space in shadow
- For each polygon
  - Pyramid with point light as apex
  - Include polygon to cap
- Shadow test similar to clipping

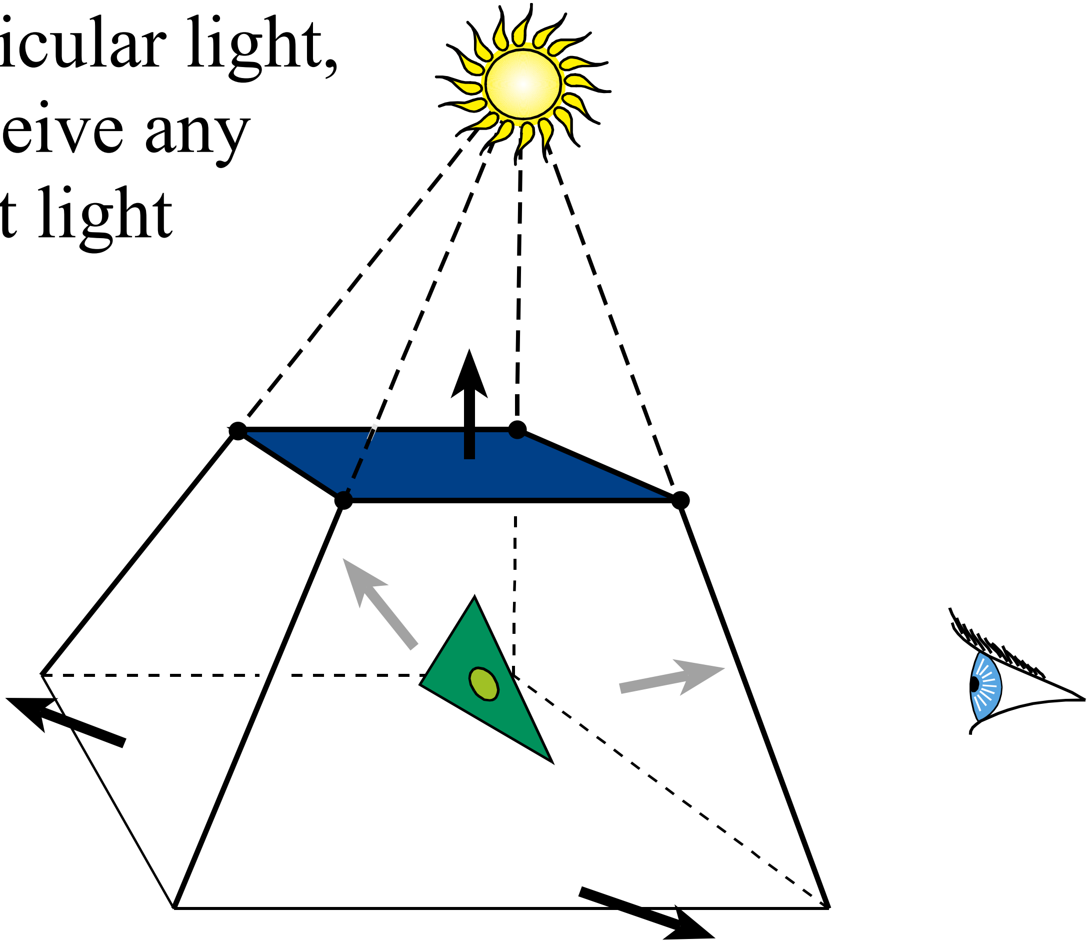


Shadow Volumes

- If a point is inside a shadow volume cast by a particular light, the point does not receive any illumination from that light

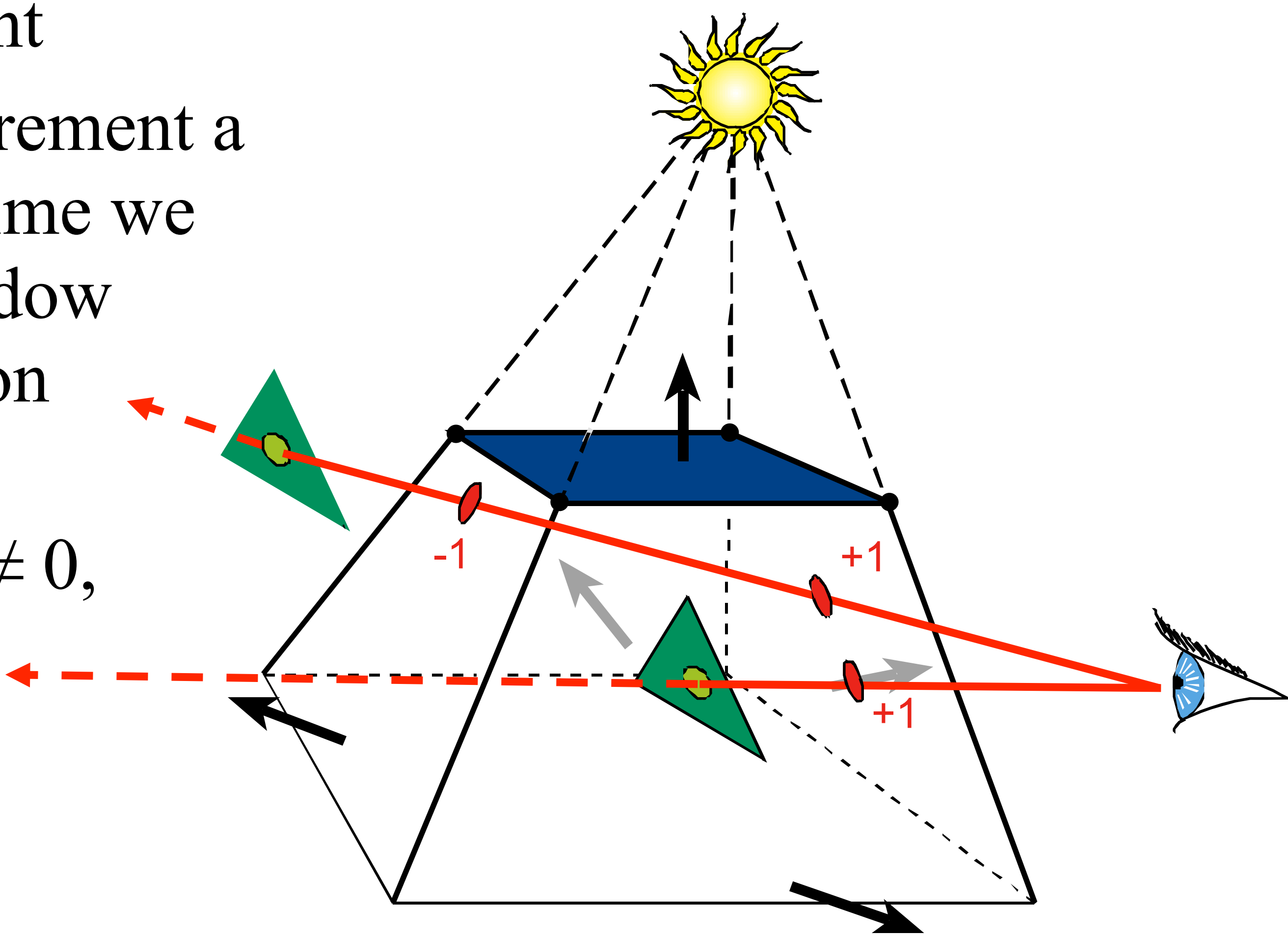
- Cost of naive implementation:

  #polygons * #lights



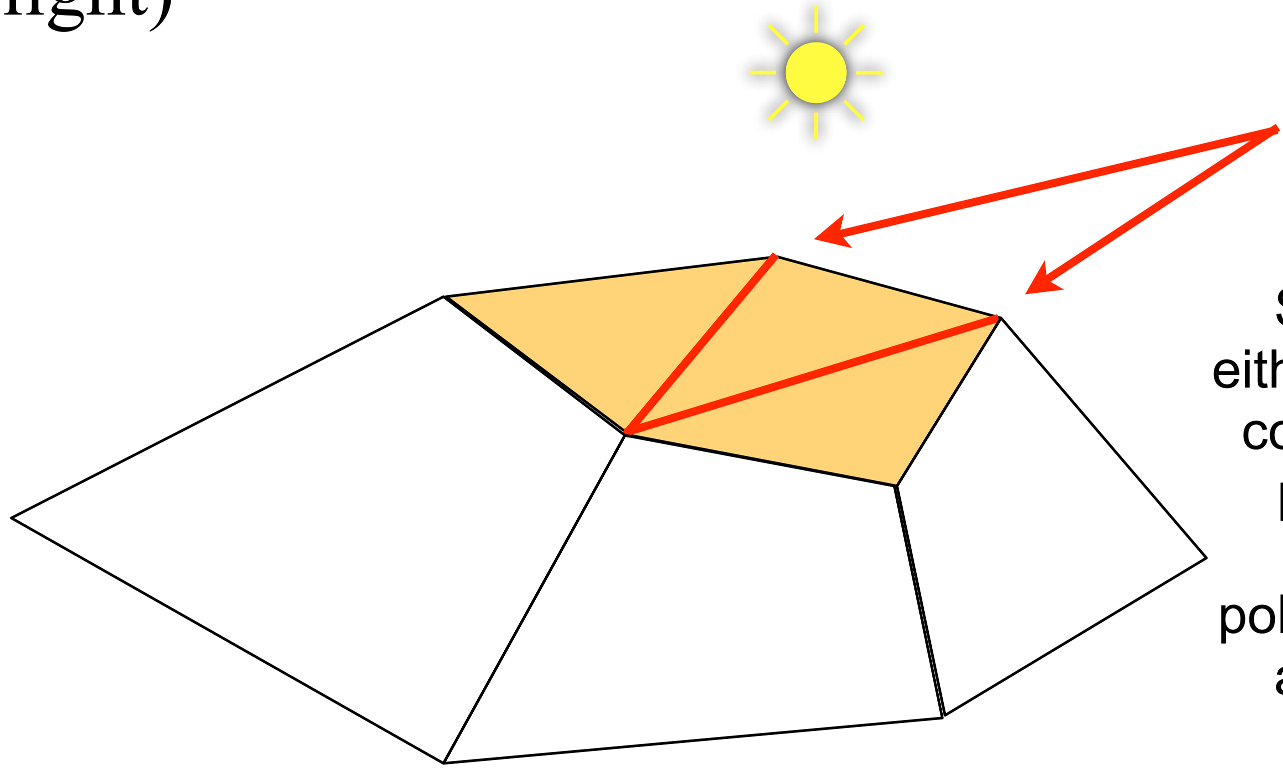
Better Shadow Volumes

- Shoot a ray from the eye to the visible point
- Increment/decrement a counter each time we intersect a shadow volume polygon
- If the counter ≠ 0, the point is in shadow



Smarter Shadow Volumes

- We don’t really need the shadow volume for each individual primitive, because edges of actual shadow regions only happen on *silhouettes* (as seen from light)



These edges do not need shadow volume polygons

Silhouette edges are either 1) edges that are connected to only one polygon, or 2) edges connected to one polygon facing the light and one facing away from the light.

Shadow Volumes Are Sort of Passé

- It is possible to implement shadow volumes rather efficiently using the stencil buffer
  - John Carmack’s Doom 3 engine did this very well
  - Shadow volumes had their 15 minutes of popularity around that time
  - NVIDIA even came up with hardware acceleration (“UltraShadow”)
- Further info
  - Wikipedia
  - Aila & Akenine-Möller: Hierarchical Shadow Volumes, Proc. Graphics Hardware 2004

Id Software

Shadow Volumes Are Sort of Passé

- But the need to extract silhouette edges and rasterize the resulting shadow polygons is just too much work when scene complexities and image resolutions grow

???

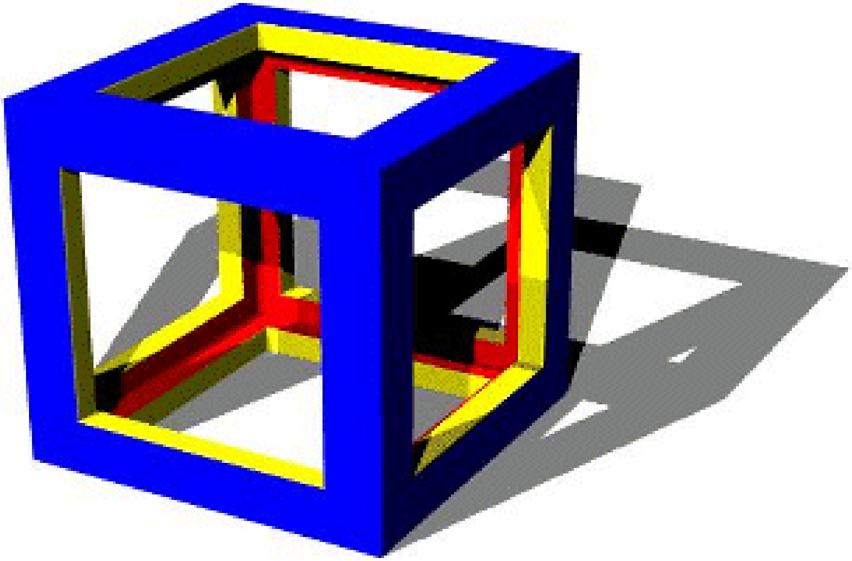
Shadow Volumes Are Sort of Passé

- But the need to extract silhouette edges and rasterize the resulting shadow polygons is just too much work when scene complexities and image resolutions grow
- Shadow maps are bad, but they're the best we've got!

???

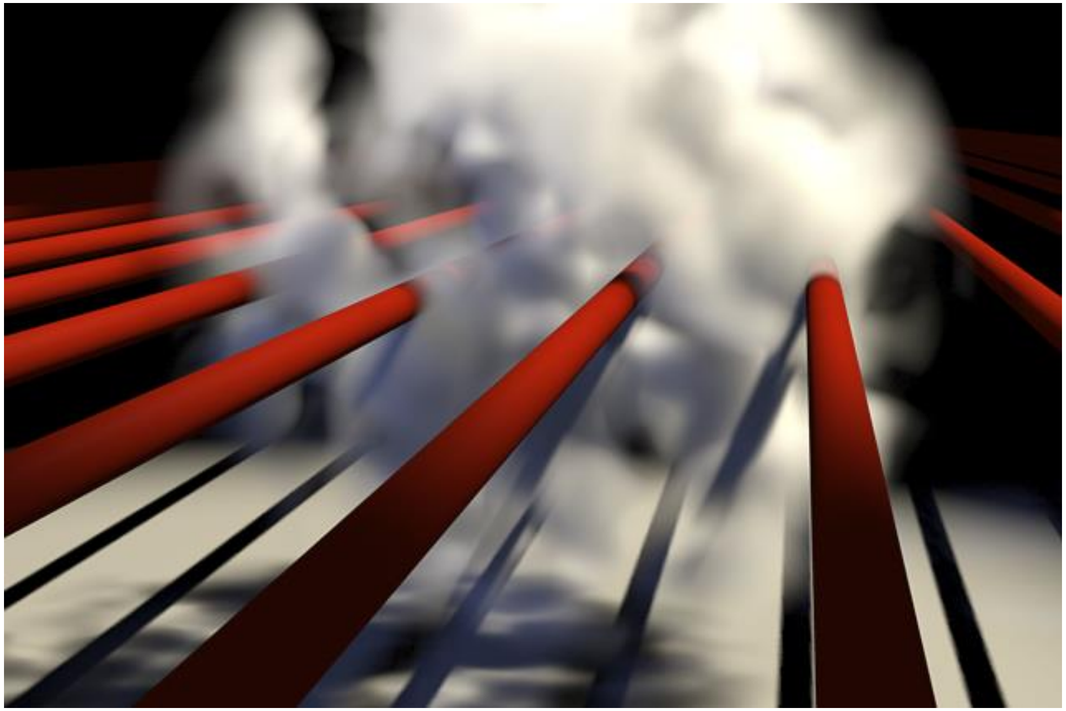
Today

- Shadow Maps
- Shadow Volumes
  - (The Stencil Buffer)
- Deep Shadow Maps



Deep shadow maps

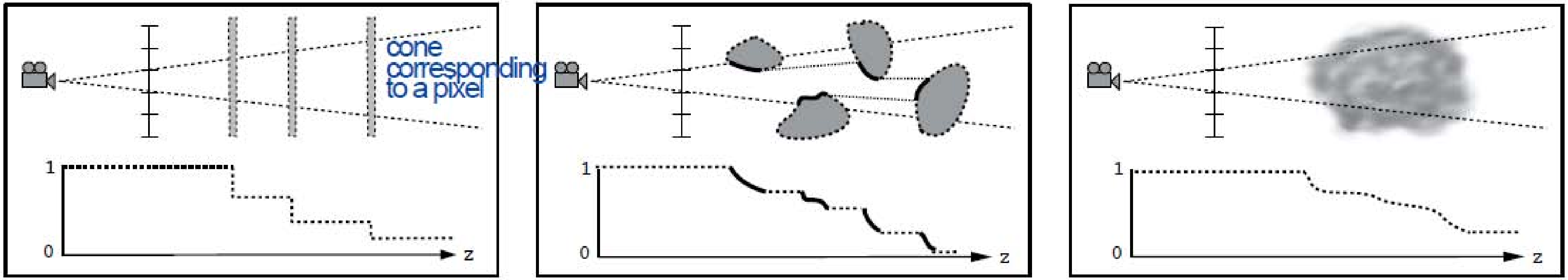
- Lokovic & Veach, Pixar
- Shadows in participating media like smoke, inside hair, etc.
  - They represent not just depth of the first occluding surface, but the attenuation along the light rays
- Note: shadowing only, no scattering



© ACM. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

Visibility function along depth

- Fraction of a pixel occluded, as a function of depth
- Due to
  - small occluders, semi-transparent objects, smoke & volumetric effects



(a) A stack of semitransparent objects

(b) Partial coverage by opaque blockers

(c) Volume attenuation due to smoke

Figure 3: *Visibility functions in flatland. Each diagram shows a beam of light that starts at the shadow camera origin (i.e. the light source) and passes through a single pixel of the deep shadow map, accompanied by that pixel's visibility function.* ***(a)*** *The beam's power is reduced as it passes through consecutive semitransparent surfaces.* ***(b)*** *The blockers are opaque, but each covers only part of the pixel's area; the emphasized segments of the function correspond to visible portions of the blockers.* ***(c)*** *Passage through smoke reduces the beam's power in a more continuous manner.*

© ACM. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

Deep shadow maps

- Preprocess:
  Compute dense visibility function for each pixel of shadow map
  - send lots of rays or rasterize at a high resolution
  - compress visibility function at each pixel
    - to reduce memory cost
- At render time, shadow query
  - Transform visible point into light coordinates
    - just like shadow maps
  - Read visibility value from compressed function

Compression algorithm

- Approximation: Piecewise linear
- Set an error bound
- Decide which vertices to keep
- Greedy from zero to far

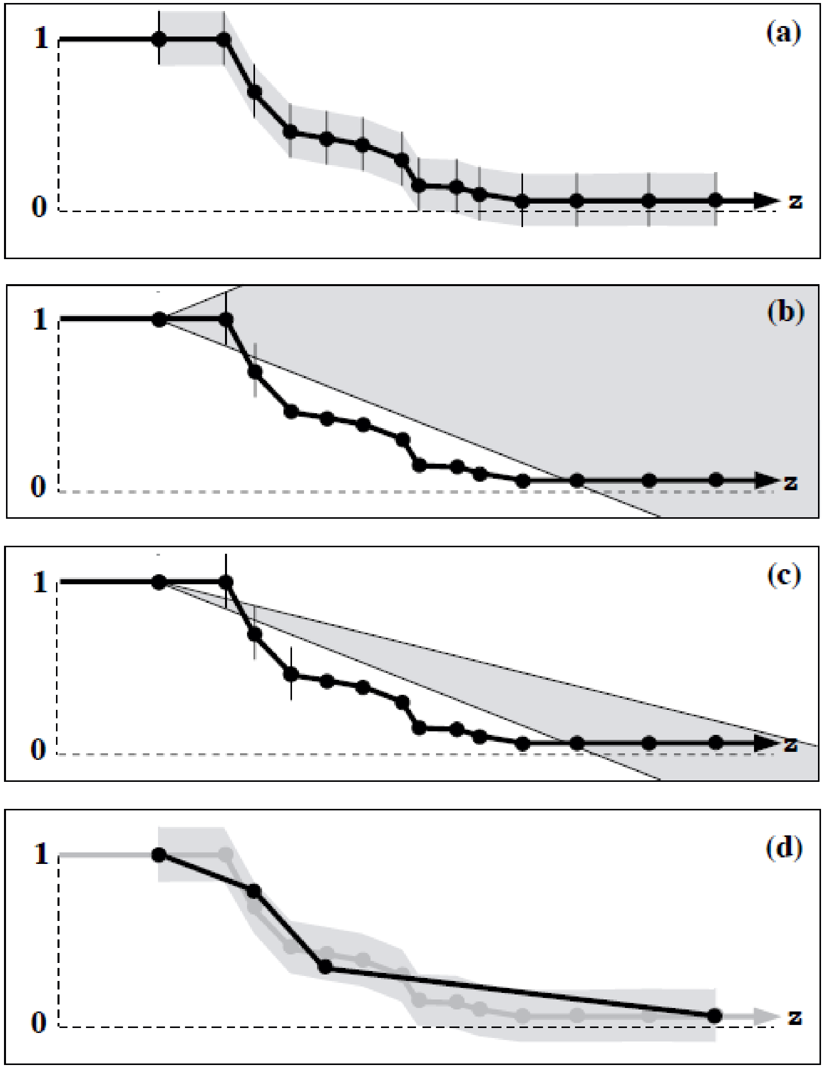


***Figure 5:** Our compression algorithm. **(a)** A piecewise linear curve and an illustration of its error bound. **(b)** Each input vertex defines a* target window *that constrains the slope of the next output segment. **(c)** The current slope range is intersected with each target window until it would become empty. **(d)** The output segment is extended to the current z value with a slope equal to the midpoint of the current slope range, and this process is repeated.*

© ACM. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

Deep shadow map results

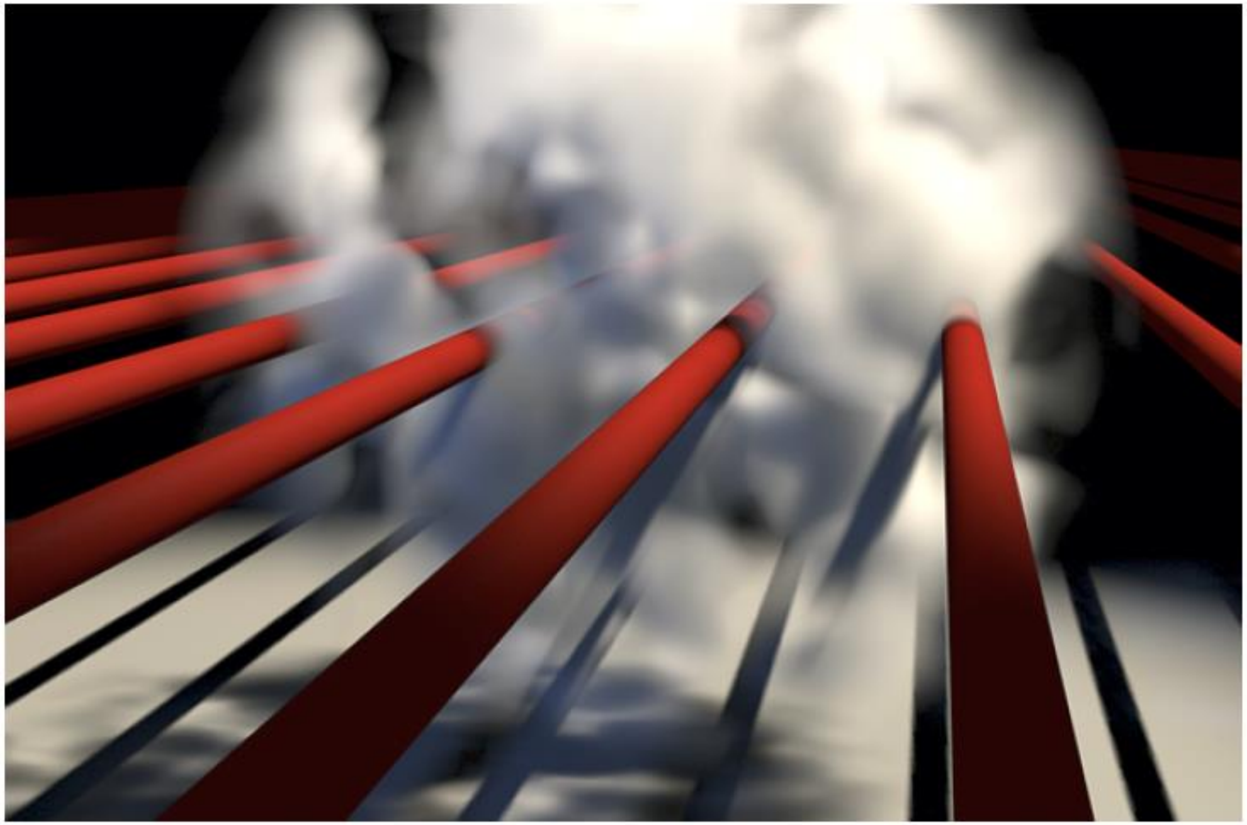


Figure 11: *A cloud with pipes. Notice the shadows cast from surfaces onto volumetric objects and vice versa. A single deep shadow map contains the shadow information for the cloud as well as the pipes.*

© ACM. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

Deep shadow map results

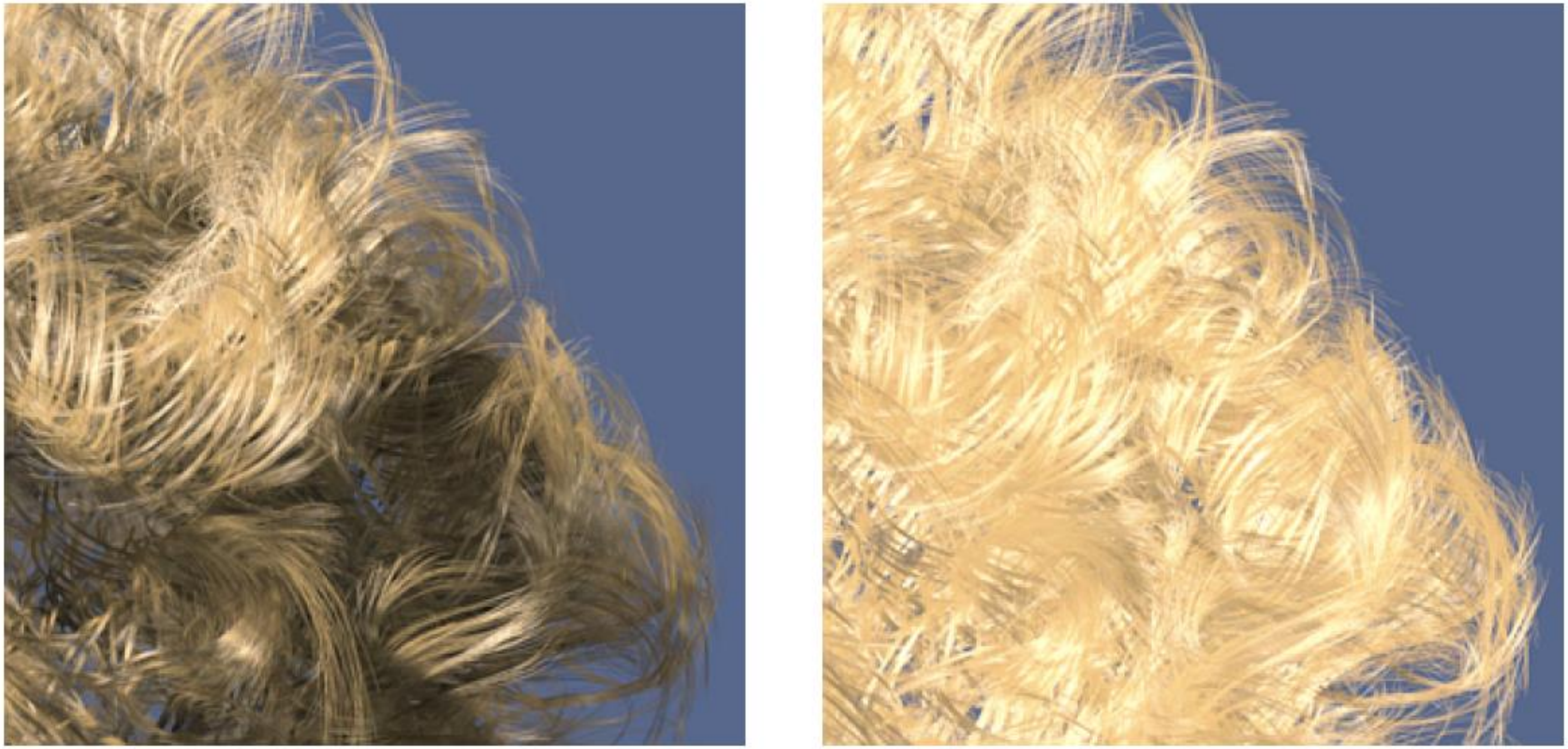
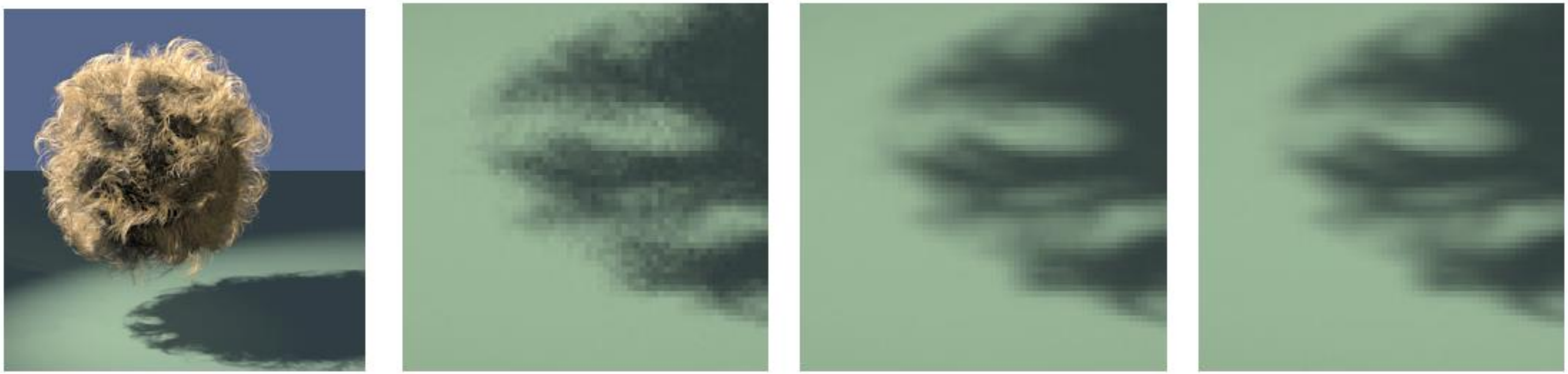


Figure 1: *Hair rendered with and without self-shadowing.*

© ACM. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

Deep shadow map results

- Advantage of deep shadow map over higher-resolution normal shadow map:
  Pre-filtering for shadow antialiasing



(a) Ball with 50,000 hairs *(b) 512×512 Normal shadow map* *(c) $4k\times4k$ Normal shadow map* *(d) 512×512 Deep shadow map*

© ACM. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

Further Reading on Shadows

- Some recent techniques allow filtering the shadow map instead of just the tests (percentage closer)
  - Works pretty well in some situations, although not general
  - Variance Shadow Maps (Donnelly, Lauritzen I3D 2006)
  - Convolution Shadow Maps
    - Annen et al., SIGGRAPH 2008
    - Annen et al., EGSR 2007
- An interesting hybrid between shadow maps and ray tracing
  - Aila and Laine, Alias-Free Shadow Maps, EGSR 2004

Annen et al. 2008