Computer Graphics

Lecture 10

Shadows

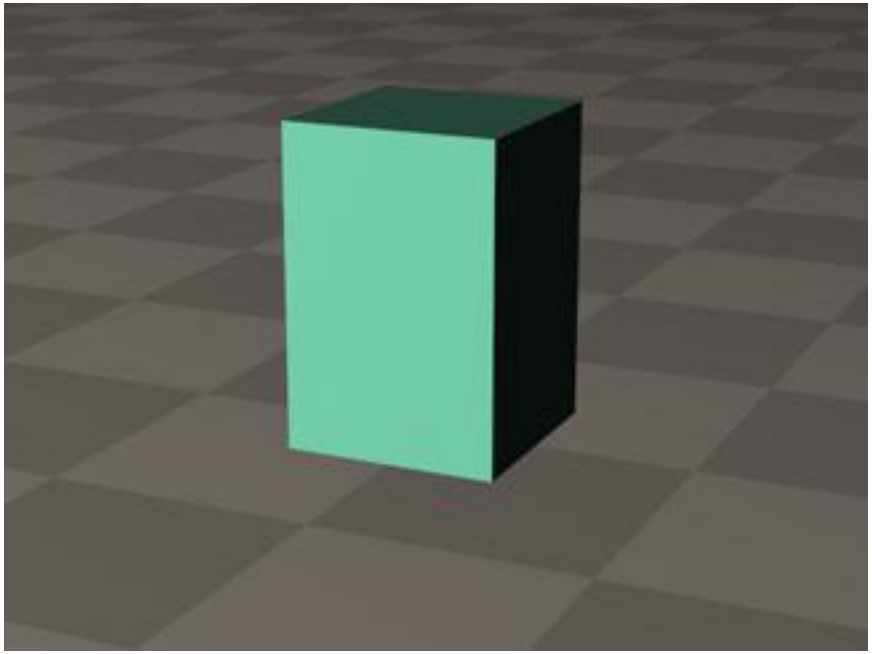
Taku Komura

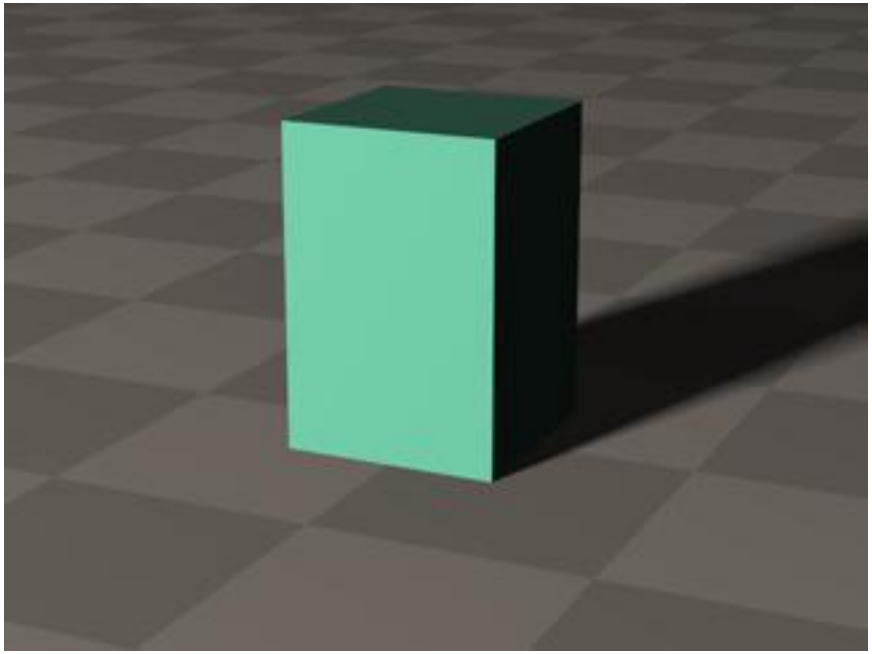
Today

- Shadows
  - Overview
  - Projective shadows
  - Shadow texture
  - Shadow volume
  - Shadow map
  - Soft shadows

Why Shadows?

- Shadows tell us about the relative locations and motions of objects



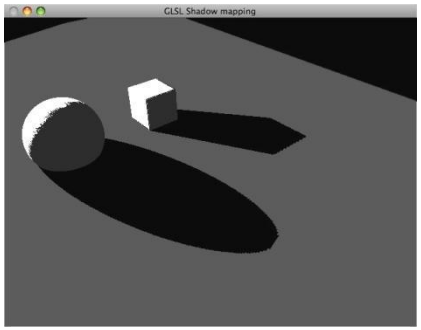


https://www.youtube.com/watch?v=pVTasC-czmw

http://gandalf.psych.umn.edu/users/kersten/kersten-lab/images/ball-in-a-box.mov

What are shadows?

- Shadows can be considered as areas hidden from the light source

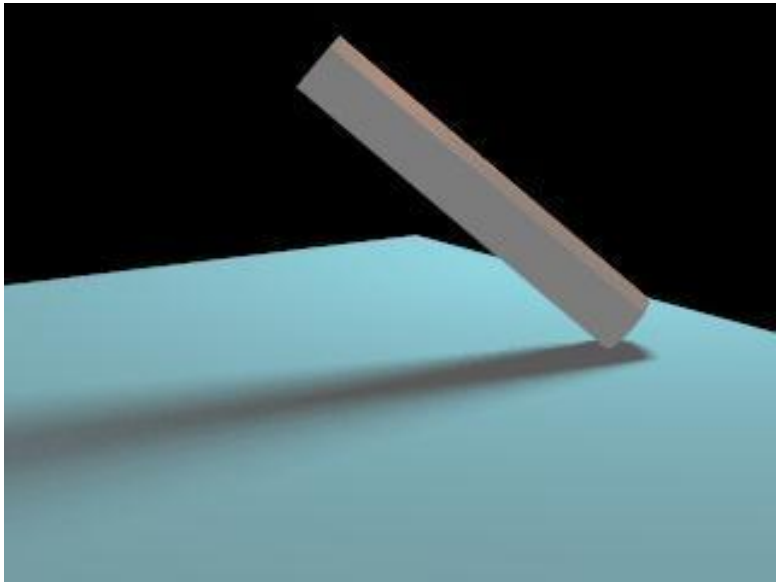


- A shadow on A due to B can be found by projecting B onto A with the light as the center of projection

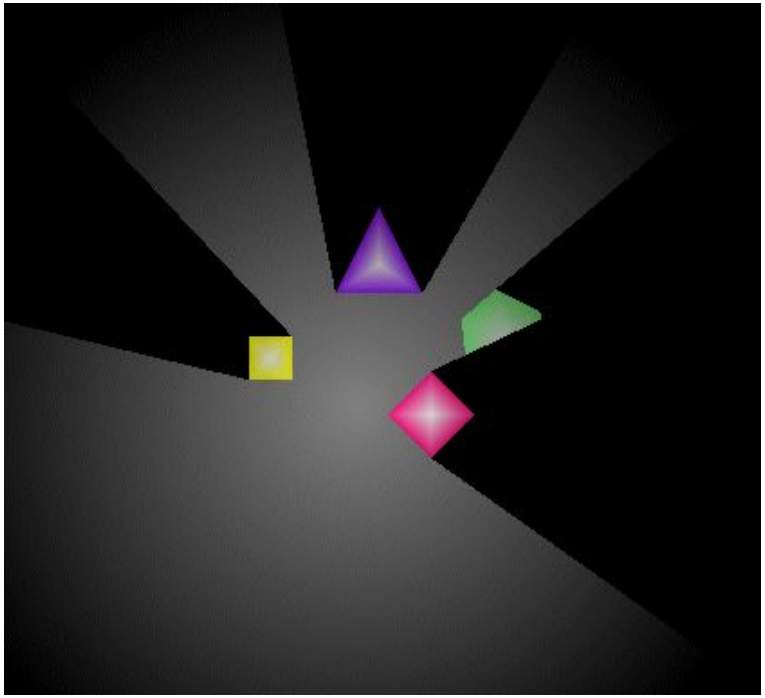
Soft and hard shadows

•Point lights have hard edges, and area lights have soft edges

Soft shadows



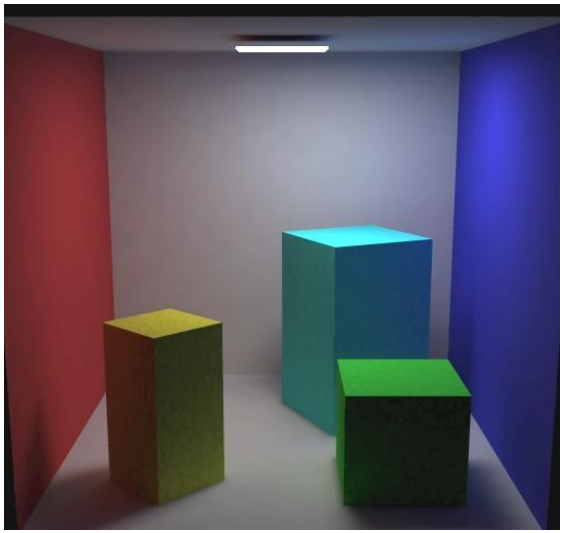
Hard shadows



Rendering Shadows

- Despite its importance, rendering shadows is not very straight forward
- Precise rendering of shadows require **ray tracing or global illumination** techniques, which are very computationally costly
- In order to avoid such intense computation, many techniques have been proposed for the graphics pipeline.

  - Projective shadows
  - Shadow texture
  - Shadow volume
  - Shadow map
  - Soft shadows

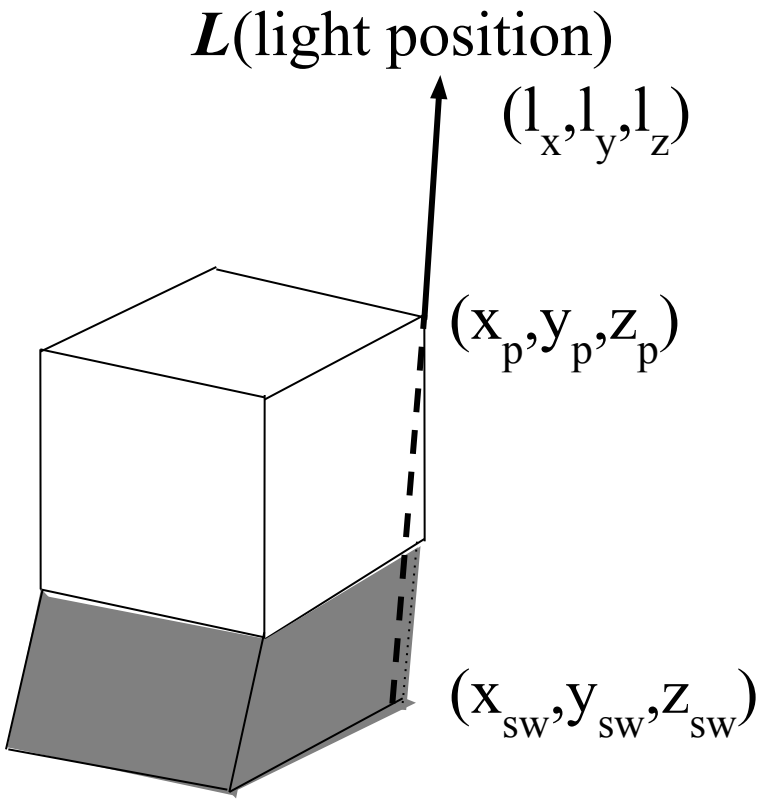


Today

- Shadows
  - Overview
  - **Projective shadows**
  - Shadow texture
  - Shadow volume
  - Shadow map
  - Soft shadows

Projective Shadows

- Here I talk about shadows on **planes** produced by **point light** sources
- These are relatively easy to compute

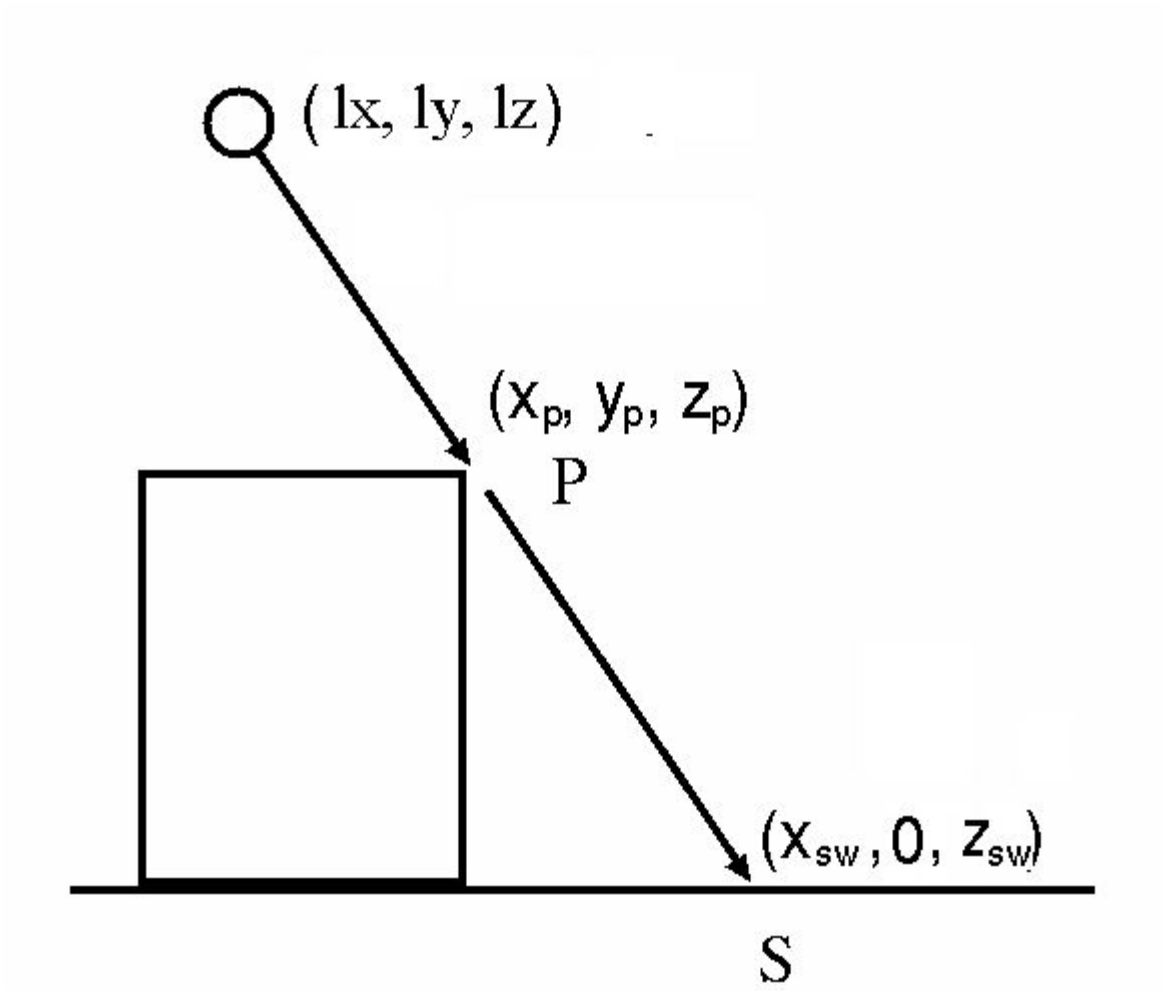


Point Light Shadows

Light source

Object

Floor



Point Light Shadows

- Blinn ’88 gives a matrix that works for local point light sources (based on projection)
  - Takes advantage of perspective transformation (and homogeneous coordinates)

$$\begin{bmatrix} x_{sw} \\ 0 \\ z_{sw} \\ 1 \end{bmatrix} = \begin{bmatrix} l_y & -l_x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -l_z & l_y & 0 \\ 0 & -1 & 0 & l_y \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

Drawing the Shadow

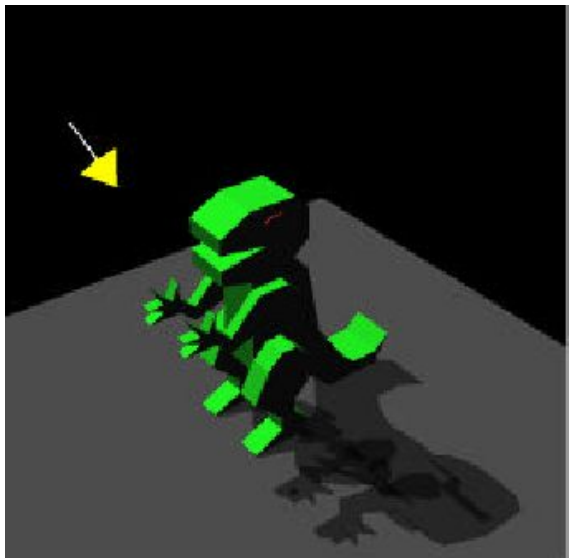
- We now have a matrix that transforms an object into its shadow
- Drawing the shadow:
  - Multiply the shadow matrix into the model transformation
  - Redraw the object in grey with blending on (making them translucent)

$$v' = M_s \, M_{l\_to\_w} \, v$$

shadow vertex

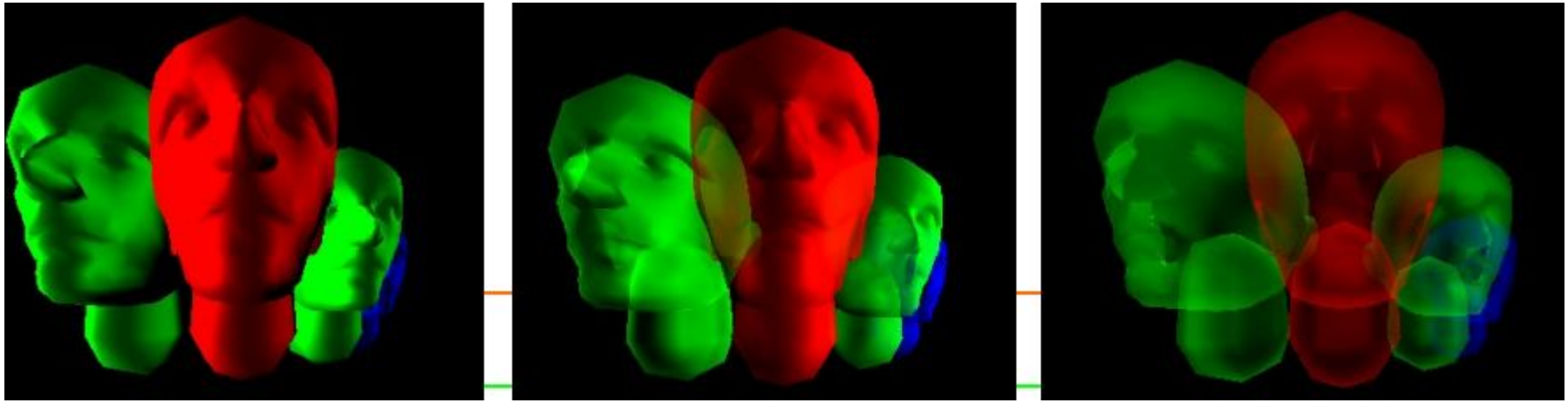
shadow matrix

point in local coordinates

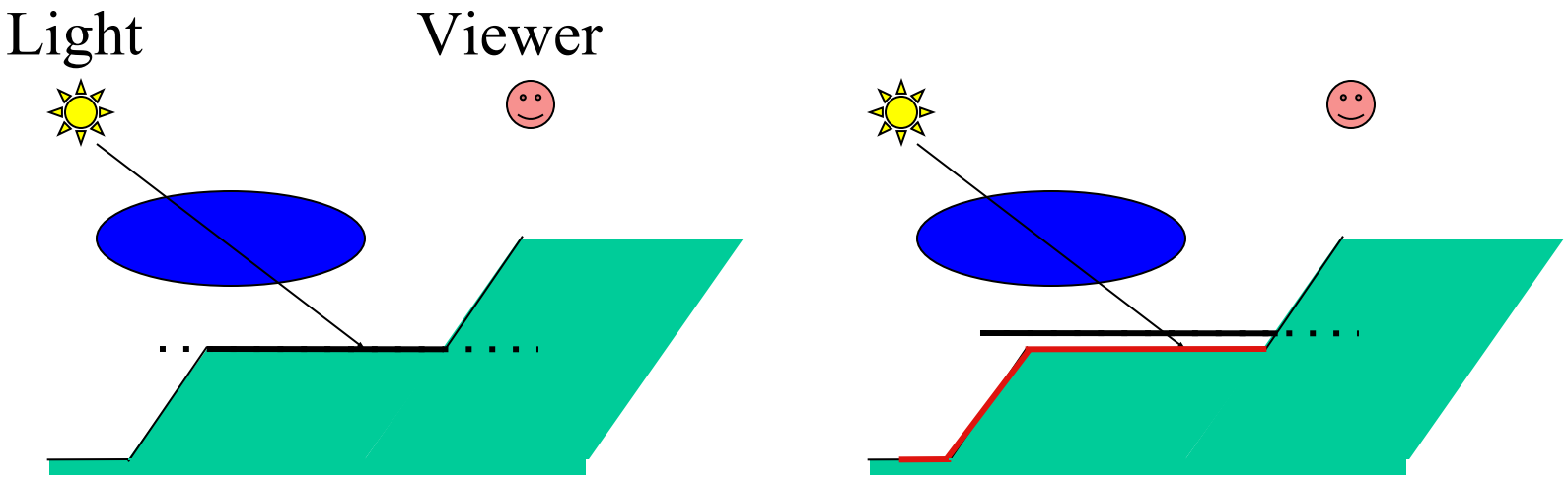


Alpha blending

- A technique to render semi-transparent objects
- Will cover more in the following lecture



Lifting the shadow above the surface



Shadow height equal with hit polygon

Shadow height above hit polygon

- Tricks:
  - Lift the shadow a little off the plane to avoid z-buffer quantization errors

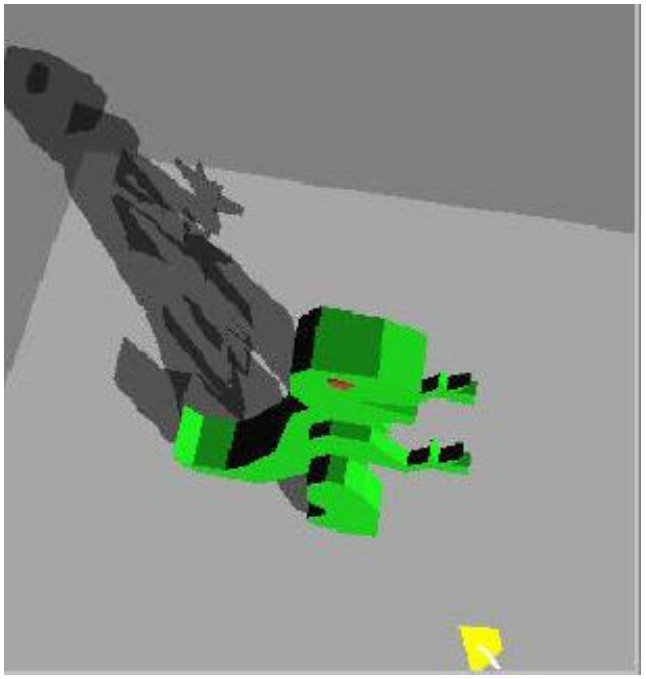
Point Light Shadows : problem

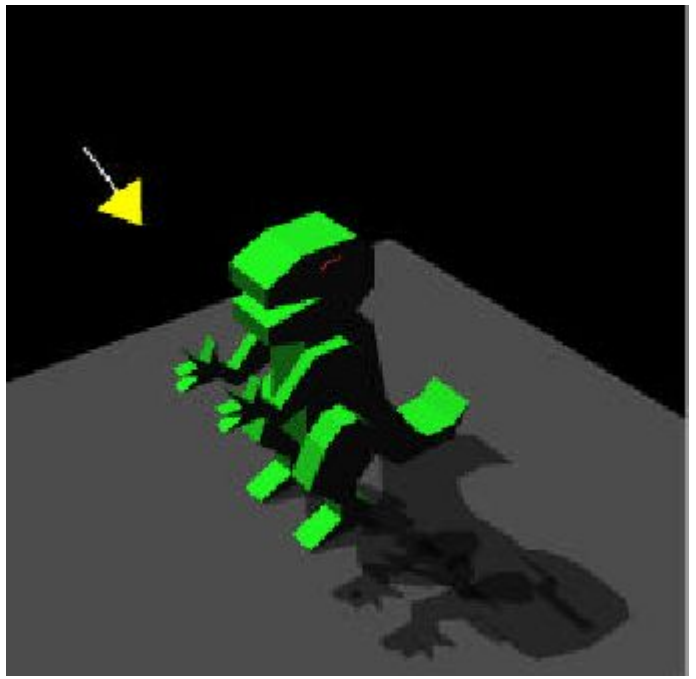
?

?

?

?





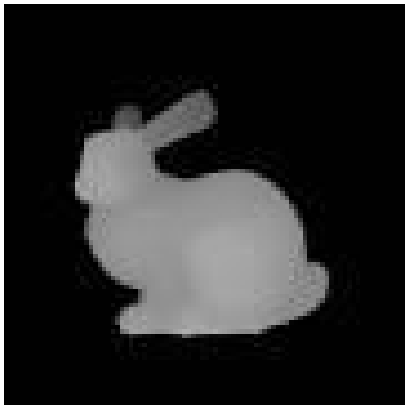
Some review

- Three types of buffers: (images)

Colour buffer

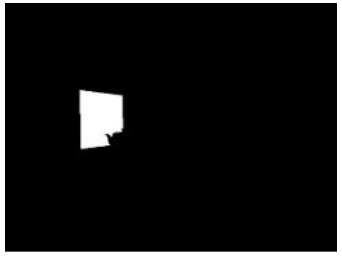
Color

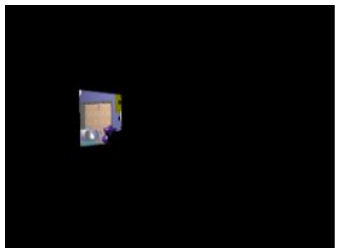
Z-buffer



depth

Stencil buffer



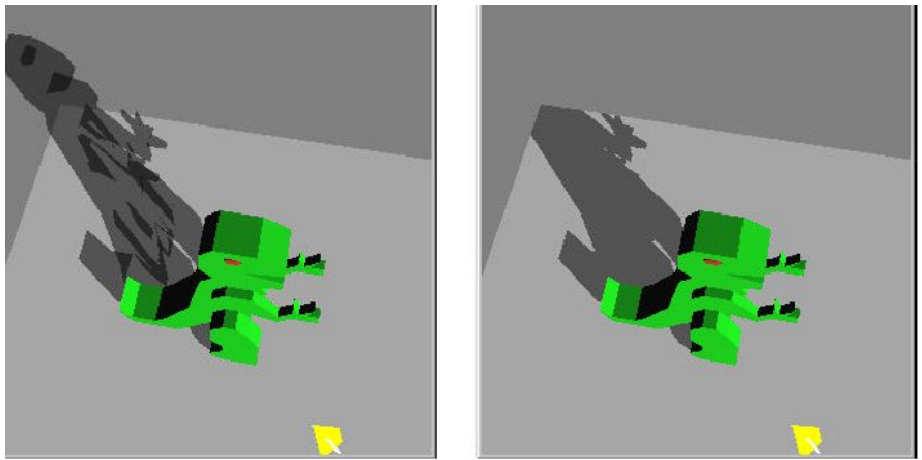


on/off, or integers

Improving Planar Projected Shadows with Stencil

- Set the stencil on for the pixels of the floor
- Only render the shadows on pixels that the stencil is on
- Once a shadow is rendered, the stencil value is set to zero
  - subsequent pixel updates will fail
- No need to lift the shadows above the floor
- Shadows not rendered outside the floor

But still we can only shadow on planes …



Today

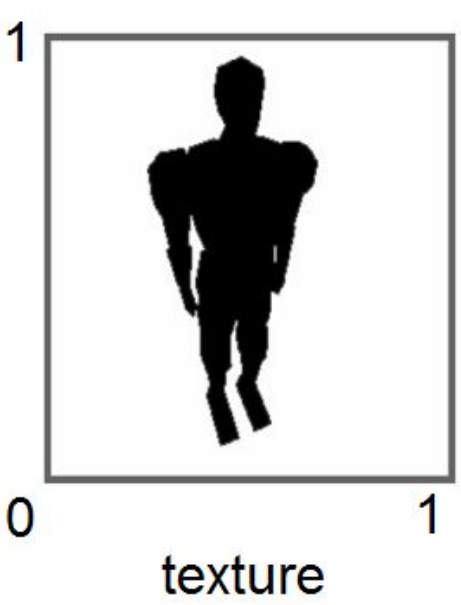
- Shadows
  - Overview
  - Projective shadows
  - **Shadow texture**
  - Shadow volume
  - Shadow map
  - Soft shadows

Shadow Texture

- Use a shadow image as a projective texture
- Generate an image of the occluder from the light's view and color it grey
- Produce a shadow by texture mapping this image onto the background object

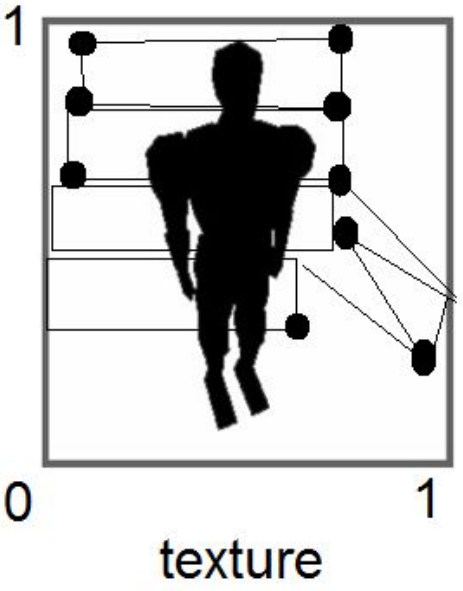
Q: How do we compute the uv coordinates?

- For mapping the shadow texture, we need to know the uv coordinates of the texture at every vertex of the background mesh



Q: How do we compute the uv coordinates?

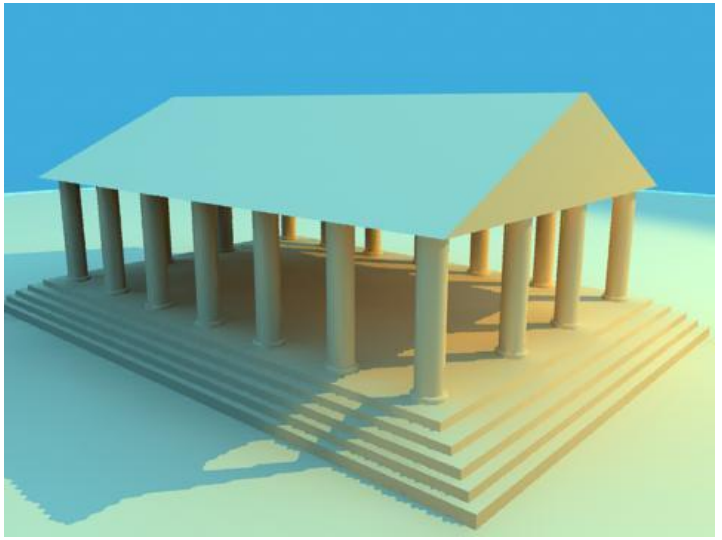
- View the object from the light source
- Project the background object onto the projection plane used to produce the shadow texture
- Obtain the (x,y) coordinates and normalize
  - This becomes the uv coordinates



Shadow Texture : Cons and Pros

Pros

- How is it good compared to projective shadows?
- The shadow does not need to be recomputed if the occluder does not move.

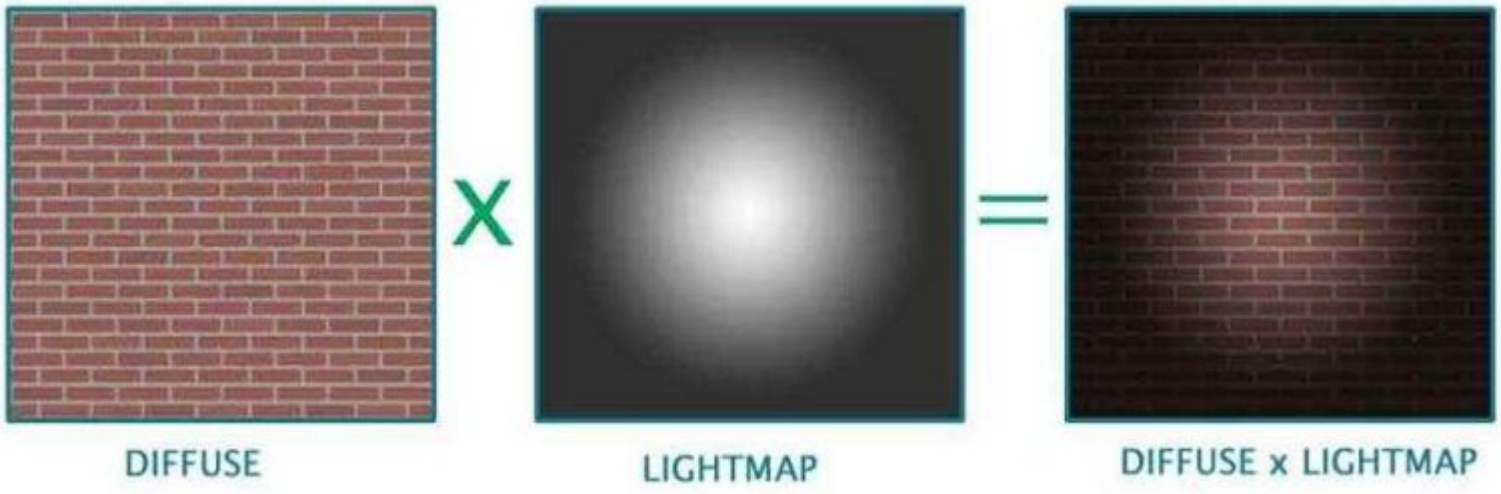


Cons

1. ?
2. ?
3. ?

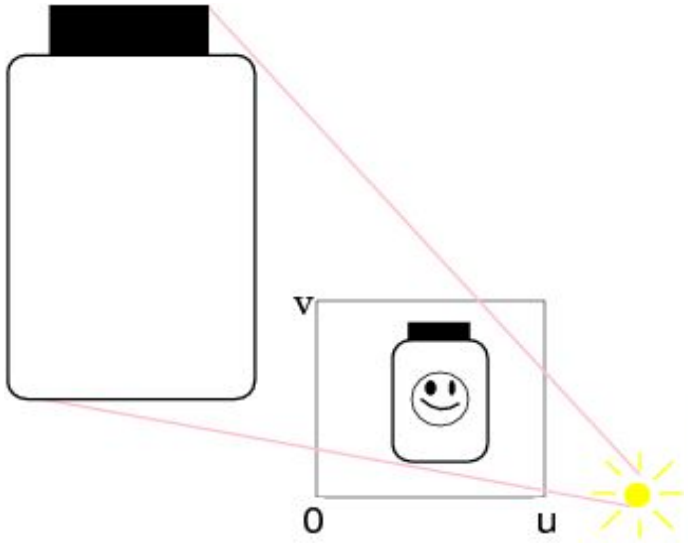
extra: Light Mapping

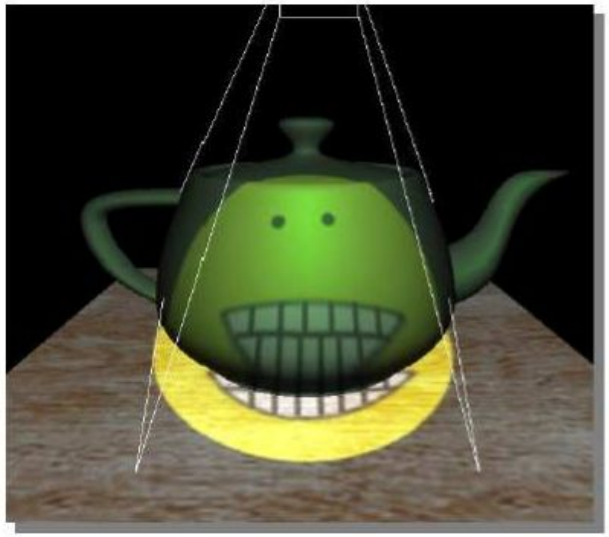
- Producing spotlight effects
  - Not so easy in ordinary graphics pipeline
- Multiply the light map to the original diffuse texture and map to the surface

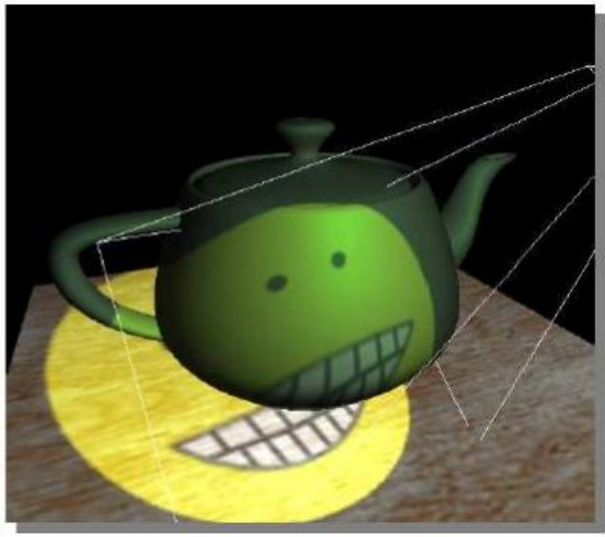


extra: Light maps by projective textures

1. View the objects from the light source
2. Compute the uv coordinates by projecting the object(s) onto the screen space





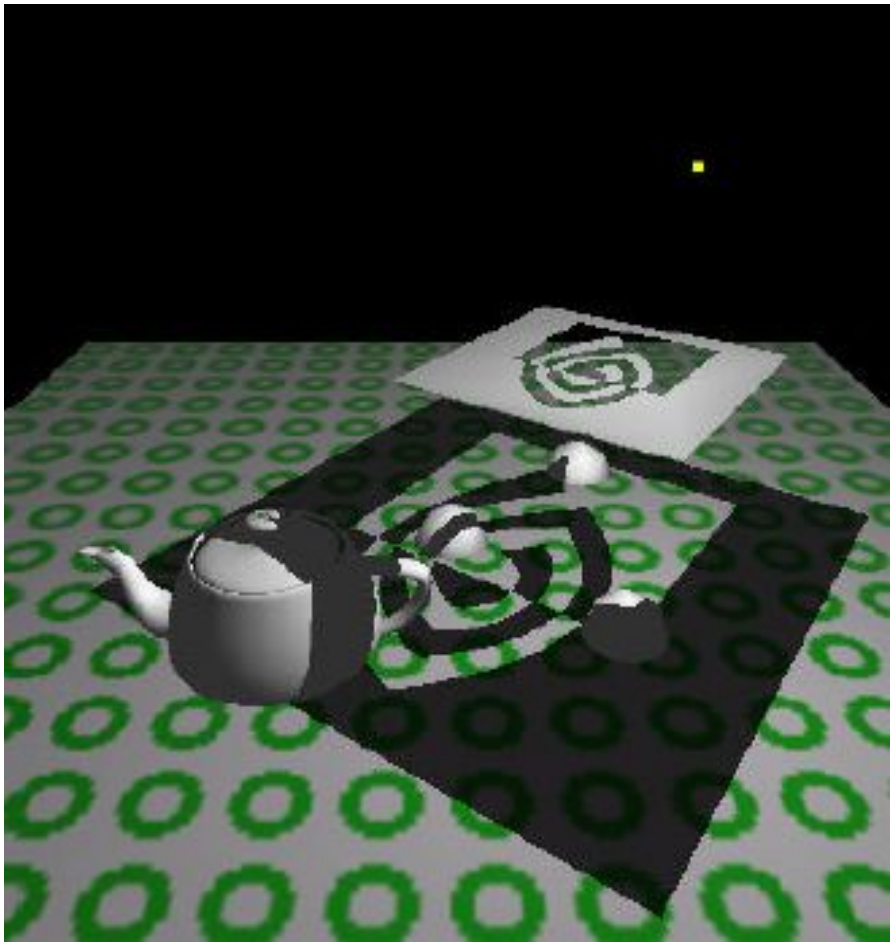


Today

- Shadows
  - Overview
  - Projective shadows
  - Shadow texture
  - **Shadow volume**
  - Shadow map
  - Soft shadows

Shadow Volume

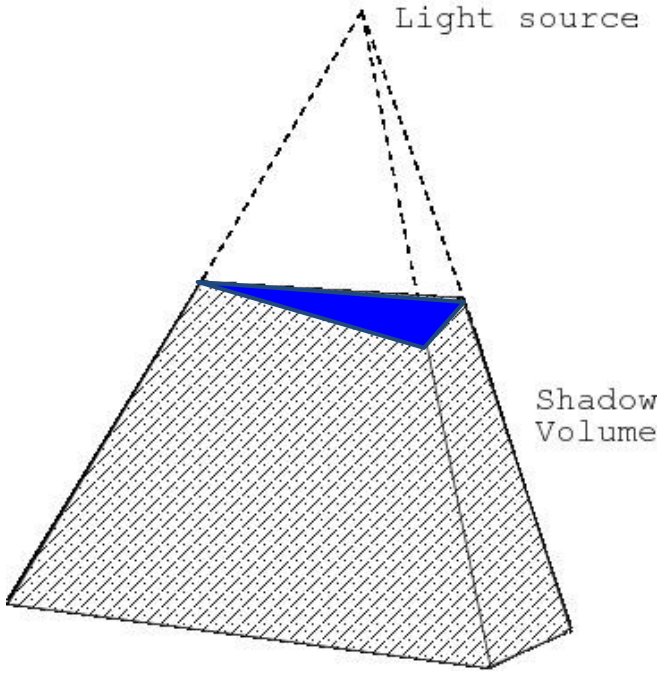
- In the real world, the shadow cast by an object blocking a light is a *volume*, not merely some two-dimensional portion of a plane.
- An algorithm that models shadow regions as volumes.



Using Shadow Volumes to Render Shadows

Two stages

- Compute the *shadow volume* formed by a light source and a set of shadowing objects.
- Every triangle produces a shadow volume



Using Shadow Volumes to Render Shadows

Two stages

- Compute the *shadow volume* formed by a light source and a set of shadowing objects.
- Every triangle produces a shadow volume
- Check whether the point is inside / outside the shadow volume
  - inside → shadowed
  - Outside → illuminated by light source

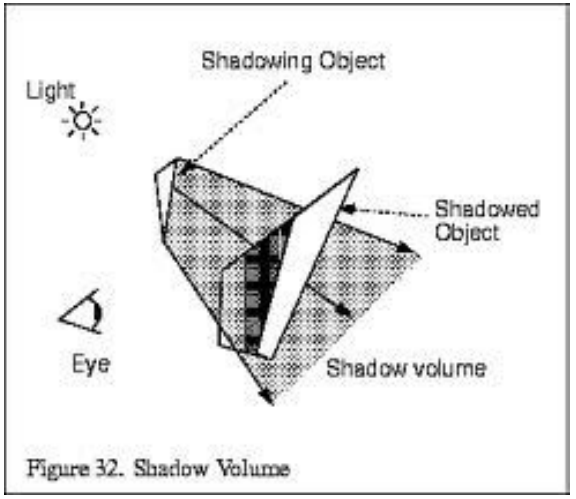
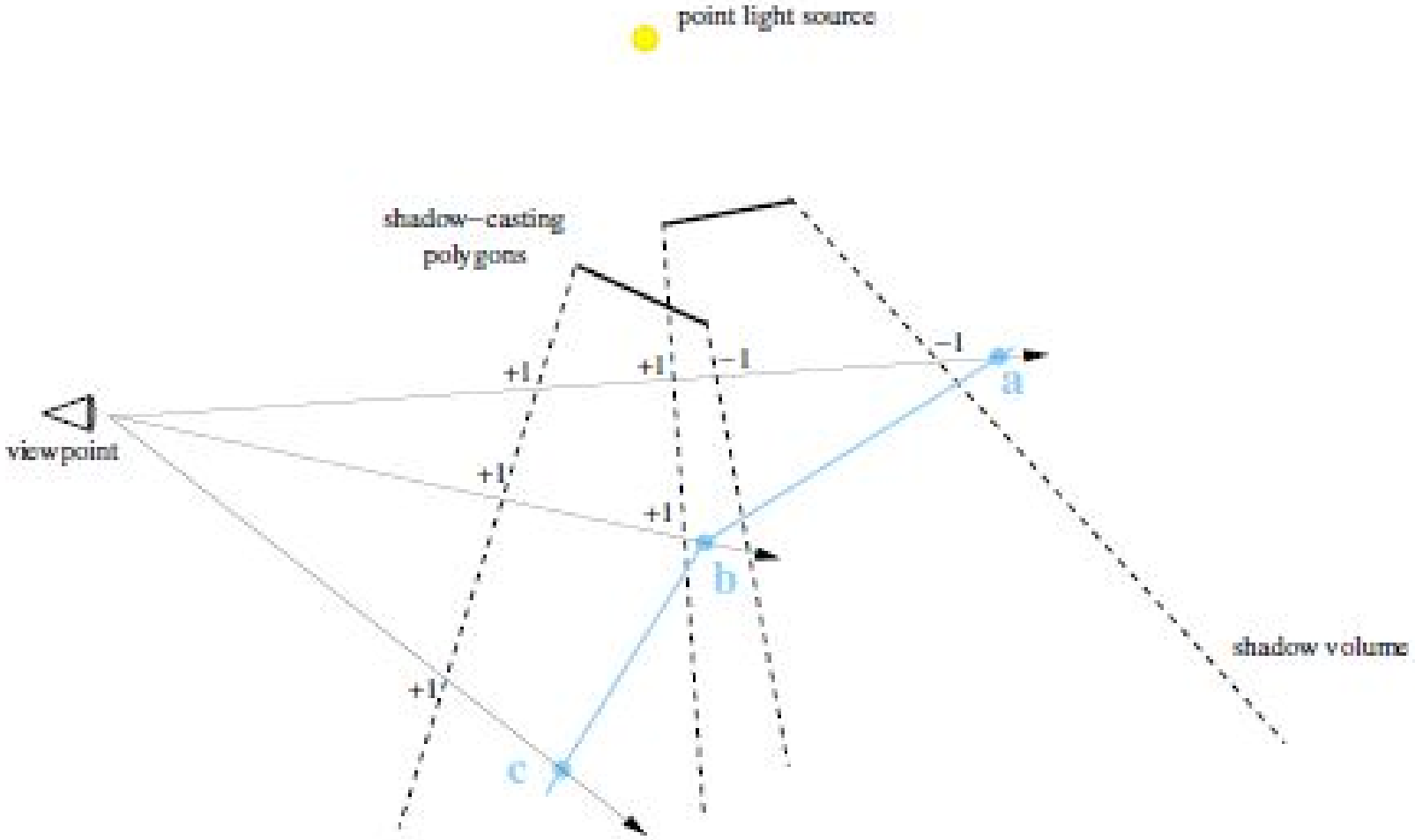
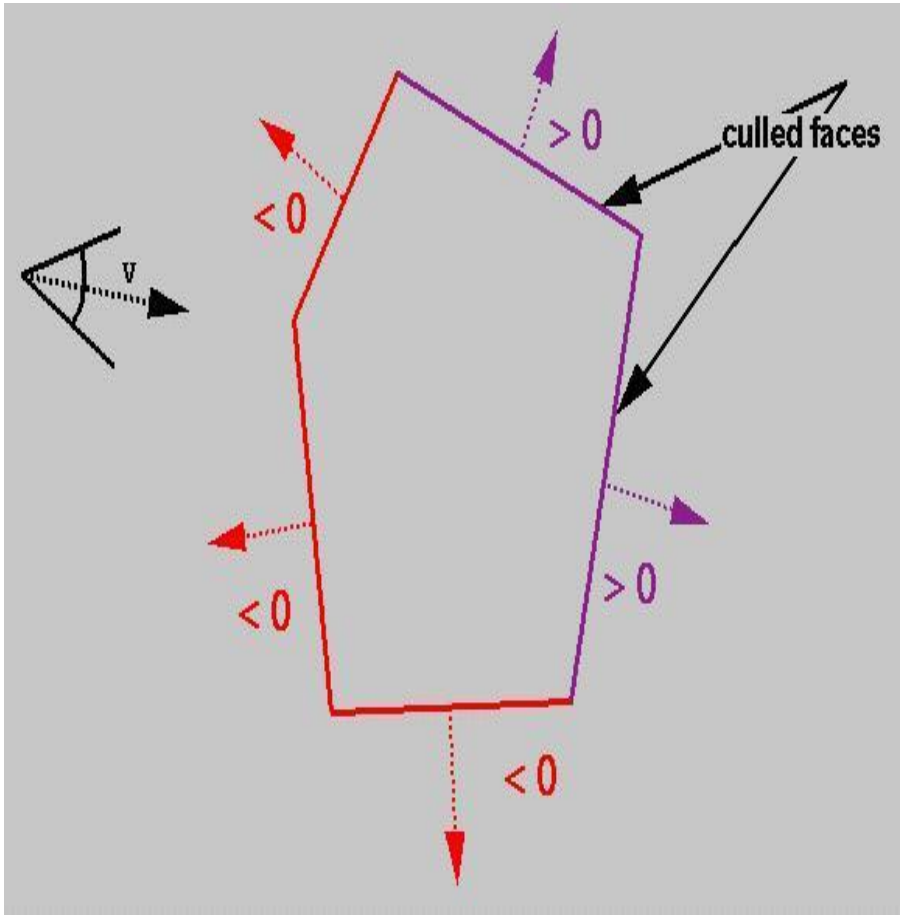


Figure 32. Shadow Volume


point light source
shadow−casting
polygons
viewpoint
+1
+1
−1
−1
+1
+1
+1
a
b
c
shadow volume

Back Face Culling

- We do not draw polygons facing the other direction
- Test z component of surface normals. If negative – cull, since normal points away from viewer.
- Or if $N \cdot V > 0$ we are viewing the back face so polygon is obscured.

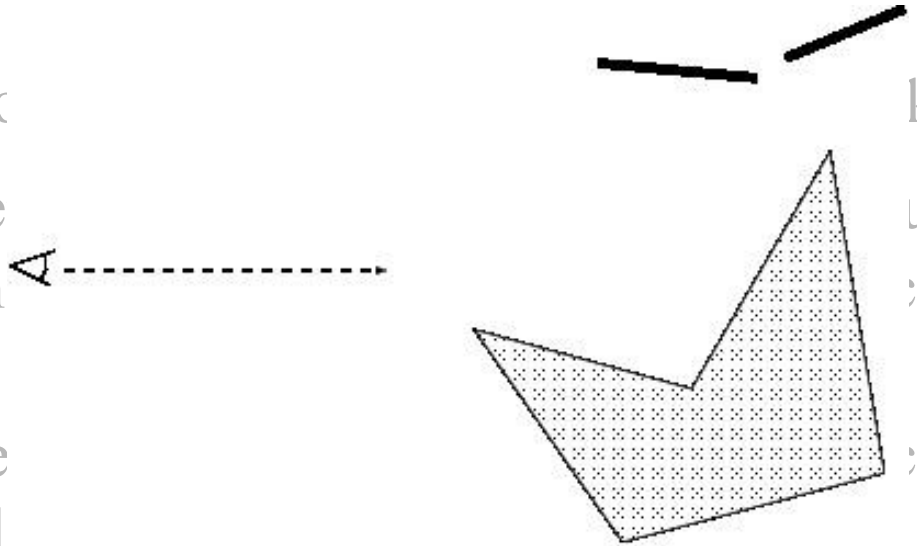


Shadow volume check by stencil buffer

- Render the scene with ambient light
- Clear the stencil buffer, and render the shadow volume with the colour buffer off and back face culling on
- Whenever a rendered fragment of the shadow volume is closer than the depth of the other objects, increment the stencil value for that pixel
- Turn on the front face culling and turn off the back face culling
- Whenever a rendered fragment of the shadow volume is closer than the depth of the other objects, decrement the stencil value for that pixel
- Render the scene using diffuse and specular reflection for the area that the stencil value is 0

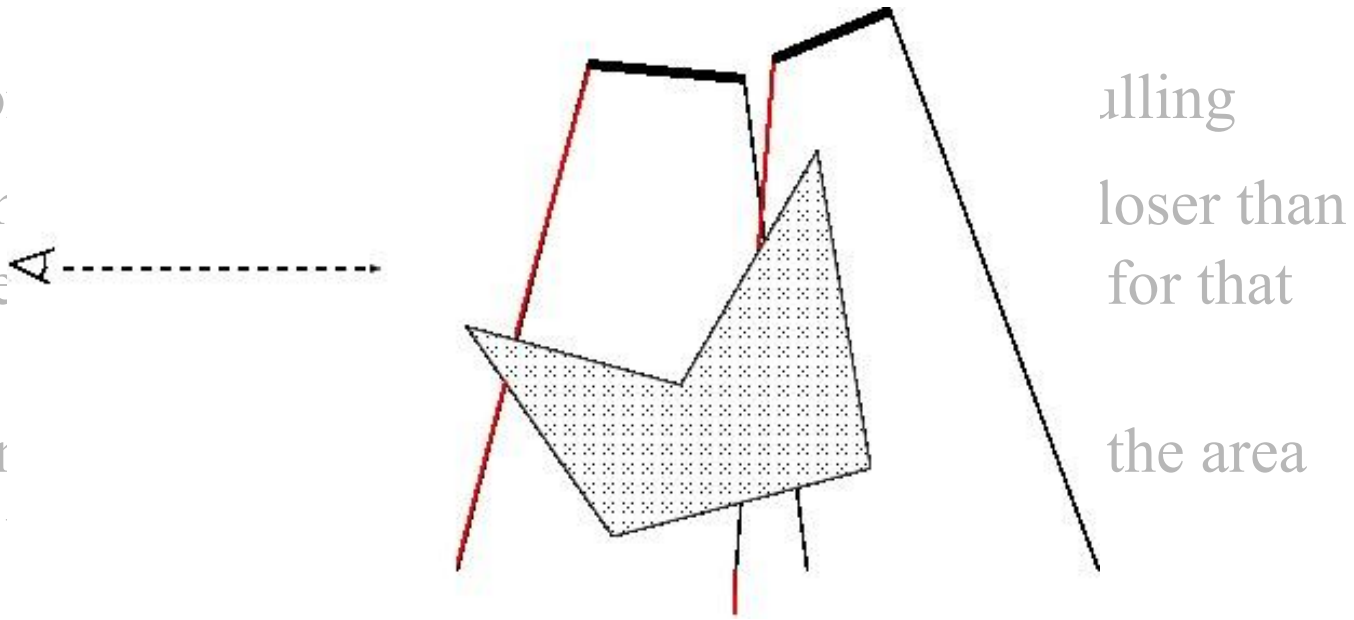
Doing this using a stencil buffer

- Render the scene with ambient light
- Clear the stencil buffer, and render the shadow volume with the colour buffer off and back face culling on
- Whenever a rendered fragment of the shadow volume is closer than the depth of the other objects, increment the stencil value for that pixel
- Turn on the fr k face culling
- Whenever a re ume is closer than the depth of th il value for that pixel
- Render the sce tion for the area that the stencil



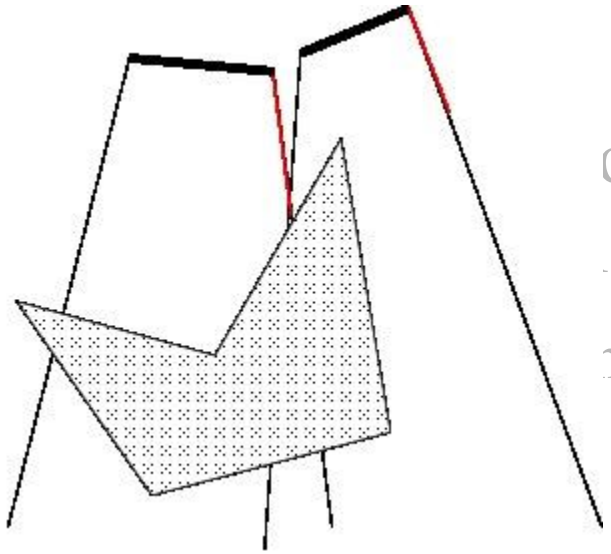
Doing this using a stencil buffer

- Render the scene with ambient light
- Clear the stencil buffer, and render the shadow volume with the colour buffer off and back face culling on
- Whenever a rendered fragment of the shadow volume is closer than the depth of the other objects, increment the stencil value for that pixel
- Turn on the fro ılling
- Whenever a rer loser than the depth of the for that pixel
- Render the scer the area that the stencil



Doing this using a stencil buffer

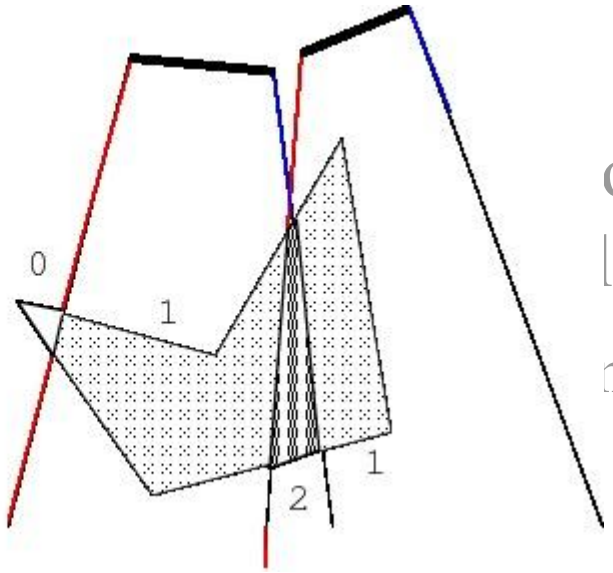
- Render th
- Clear the s ow volume with the c ing on
- Whenever a ne is closer than the depth of value for that pixel



- Turn on the front face culling and turn off the back face culling
- Whenever a rendered fragment of the shadow volume is closer than the depth of the other objects, decrement the stencil value for that pixel
- Render the scene using diffuse and specular reflection for the area that the stencil value is 0

Doing this using a stencil buffer

- Render th
- Clear the ow volume with the c ling on
- Whenever a me is closer than the depth of value for that pixel
- Turn on the front face culling and turn off the back face culling
- Whenever a rendered fragment of the shadow volume is closer than the depth of the other objects, decrement the stencil value for that pixel
- Render the scene using diffuse and specular reflection for the area that the stencil value is 0



Advantage / Disadvantages of Shadow Volume

- Advantage
  - Do not need to manually specify the shadowed objects
  - The occluder can shadow itself
  - High precision
- Disadvantage
  - Bottleneck at the rasterizer
  - Many shadow volumes covering many pixels

Today

- Shadows
  - Overview
  - Projective shadows
  - Shadow texture
  - Shadow volume
  - **Shadow map**
  - Soft shadows

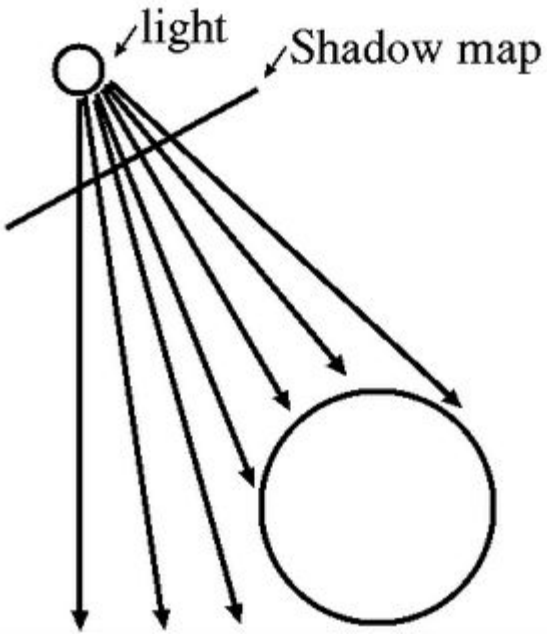
Shadow Mapping

- Another method that can handle the shadow of multiple objects casted onto objects of arbitrary shapes
- Make use of the Z-buffer

Shadow Map

Preparation

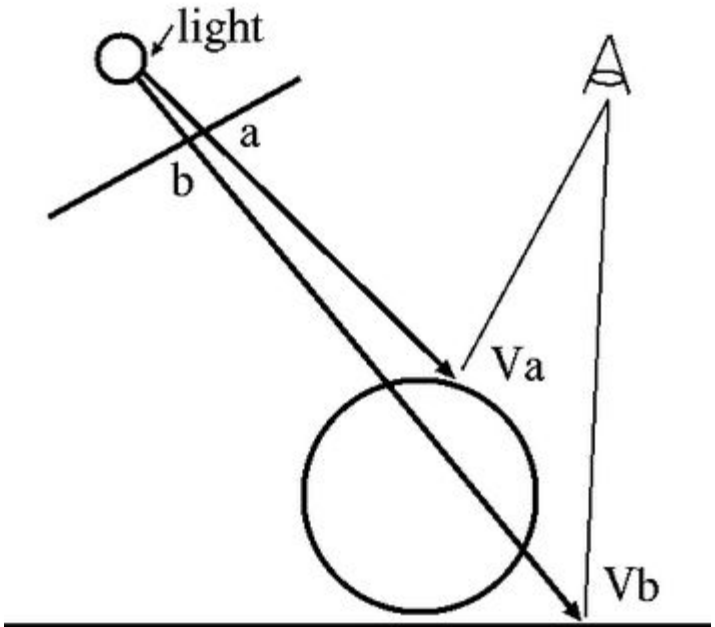
- Prepare a depth buffer for each light
- Render the scene from the light position
- Save the depth information in the depth buffer



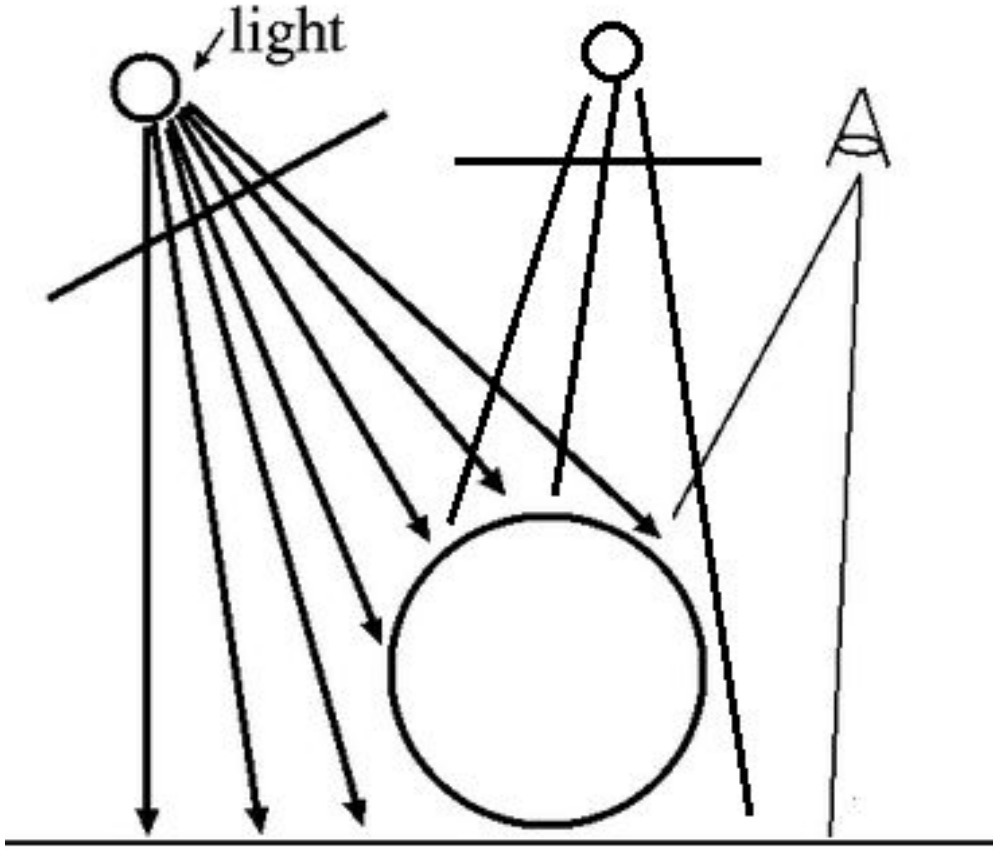
Shadow Map

Rendering the scene

1. Render the objects; whenever rendering an object, check if it is shadowed or not by transforming its coordinate into the light space
2. After the transformation, if the depth value is larger than that in the light's depth buffer it should be shadowed



Can Handle Multiple Light Source



Shadow Map - comparison

- Self shadows?
- Need to specify the occluder / occludee?

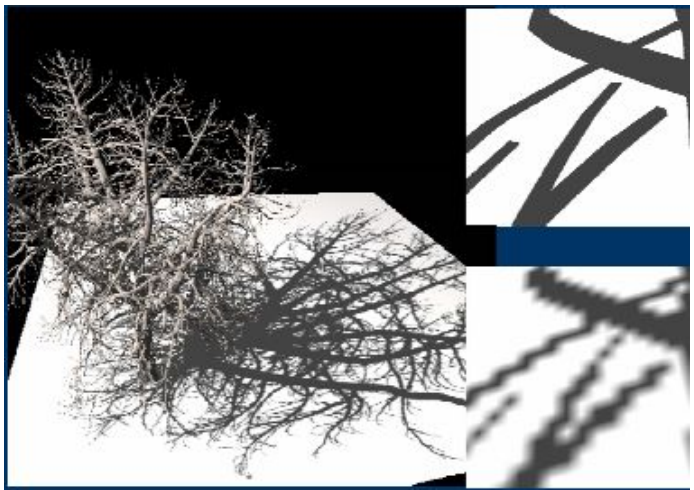
Shadow Map - comparison (2)

Advantage

- When there are many objects/light sources, it is faster than shadow volume – why?

Disadvantage

- How is the precision compared with shadow volume?



Adaptive Shadow Mapping

- The resolution of the shadow maps is lower than that of the rendered image
  - Many box shape artifacts
- Need to use shadow map of higher resolutions
- Changing the resolution of the shadow map according to the viewpoint

- Adaptive Shadow Maps,
  - Fernando et al. SIGGRAPH 2001

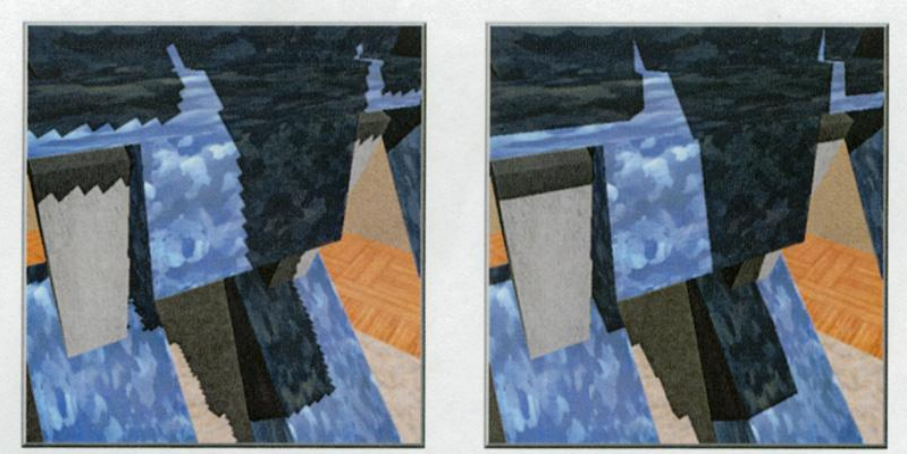


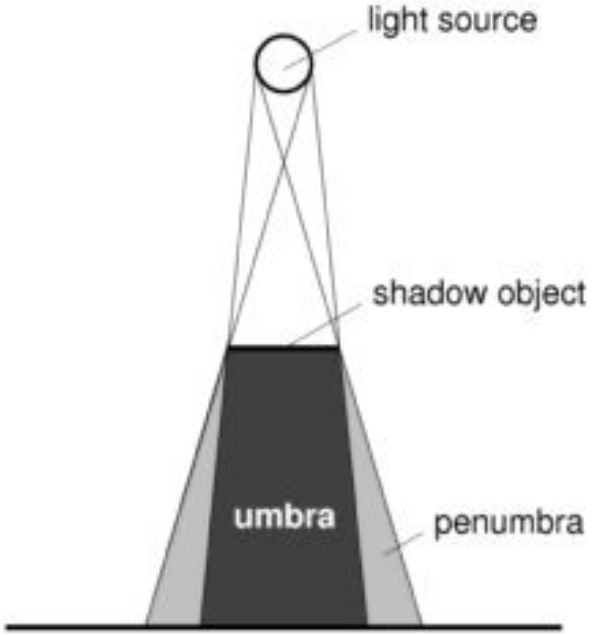
Figure 2: A conventional 2,048×2,048 pixel shadow map (left) compared to a 16 MB ASM (right). EFFECTIVE SHADOW MAP SIZE: 65,536×65,536 PIXELS.

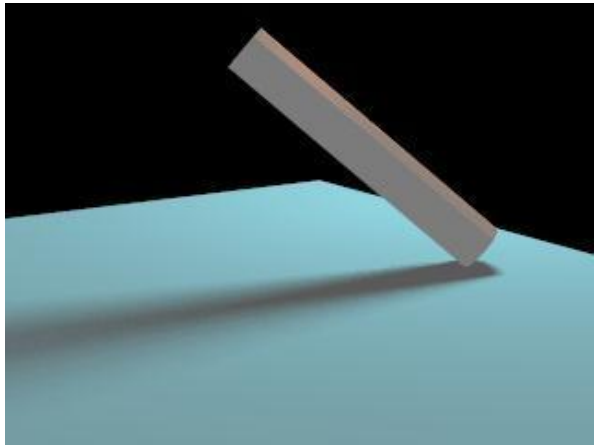
Today

- Shadows
  - Overview
  - Projective shadows
  - Shadow texture
  - Shadow volume
  - Shadow map
  - **Soft shadows**

Soft Shadows

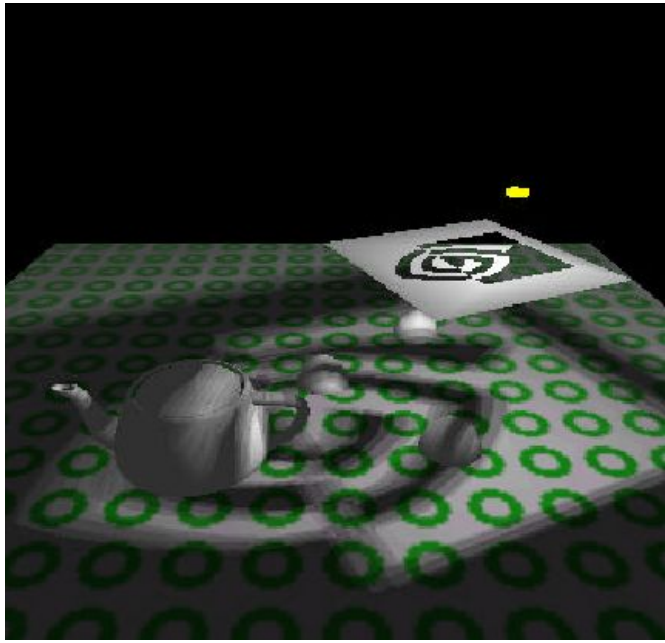
- Made by area light
  - umbra – totally blocked from the light source
  - Penumbra – partially blocked from the light source
- Can be modelled by a collection of point light sources





Soft shadowing by multiple point light sources

- Additive blending is used to accumulate the contribution of each light
- Can apply both planar projected shadows approach or shadow volume approach
- Then apply convolution
- The softness of the shadow depends on an adequate number of samples.
- The time to render the scene increases linearly with the number of samples used to approximate an area light source.
- Artifacts are introduced if not enough samples are used
- Drawback: slow

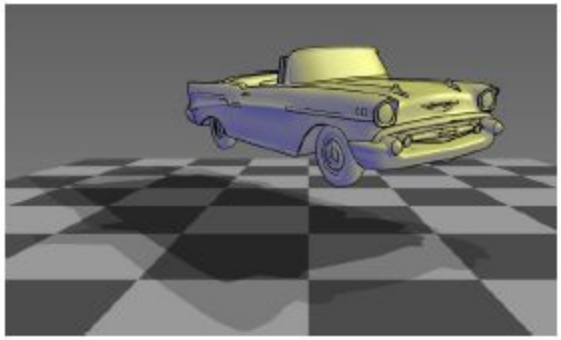


Other techniques to generate soft shadow : Gooch et al.

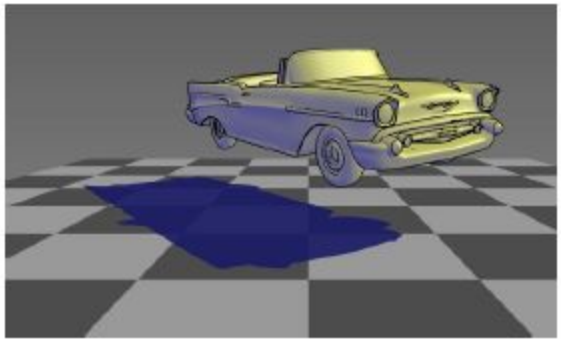
- Moving the projected plane up and down instead of moving the light source
- The projections cast upon it are averaged
- Can use projective shadows :

  Applying the same texture multiple times to planes of different heights and overlapping them
- “Interactive technical illustration”

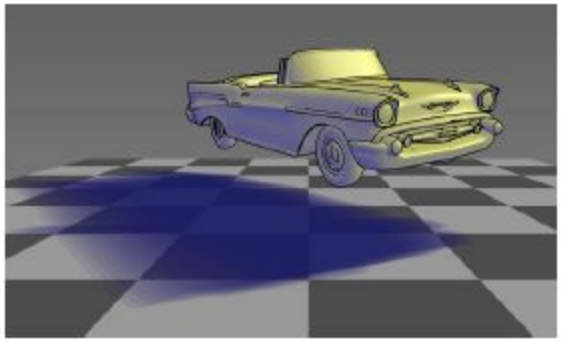
  Gooch et al. I3D 1998



(a) Hard penumbra and hard umbra.



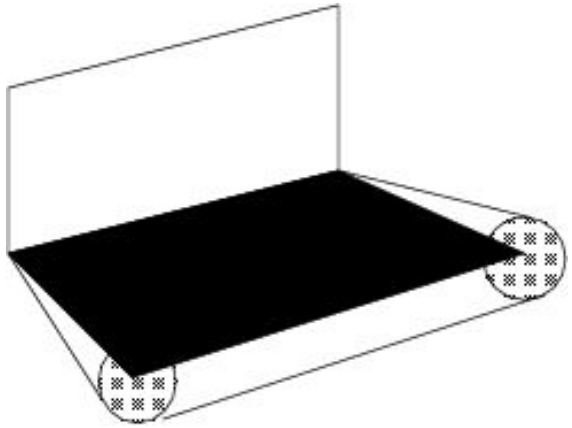
(b) Single hard, colored shadow.



(c) Colored soft shadow.

Other techniques to generate soft shadow : Haines, 2001

- First create a hard shadow and then paint the silhouette edges with gradients that go from dark in the center to white on the edges
  - The gradient areas have a width proportional to the height of the silhouette edge casting the shadow



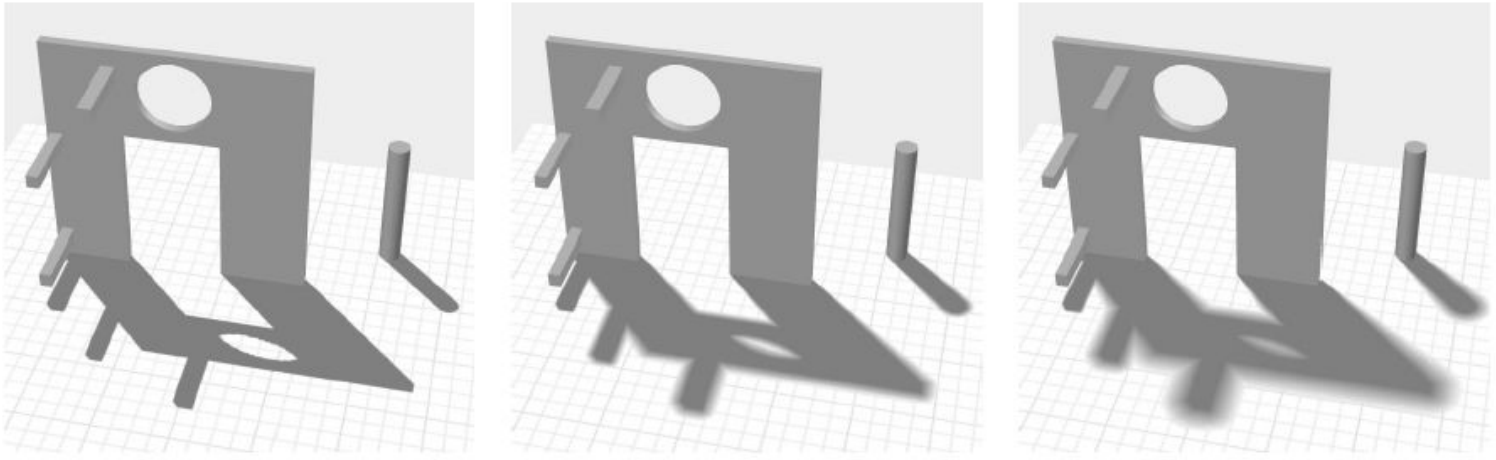
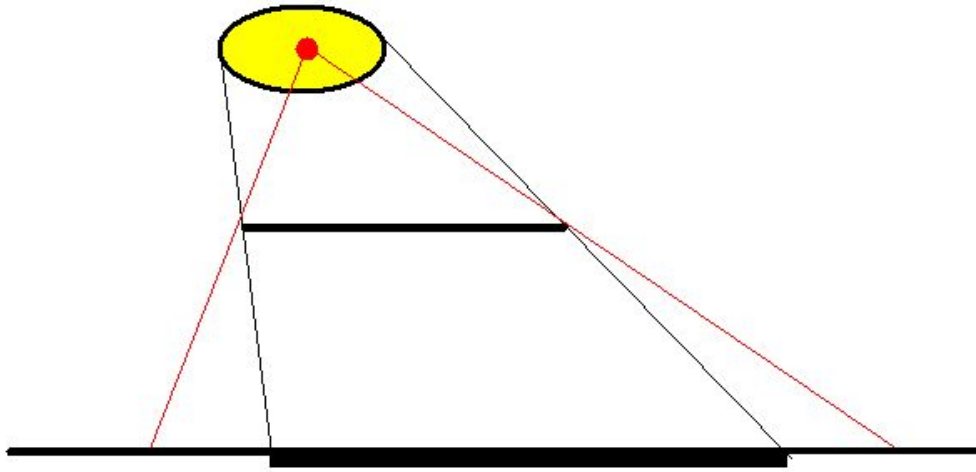


Figure 3: On the left is a hard shadow, the middle shows the effect of a small area light source, the right a larger light source.

Problems with Gooch et al. and Haine's method

- The umbra (the dark shadowed area) becomes too large as it is produced by a point light

- An area light usually decreases the size of the umbra



Summary

- Shadows need to be added with extra processing in graphics pipelines.
- Introduced the following methods
  - Projective shadows
  - Shadow texture
  - Shadow volume
  - Shadow map
  - Soft shadows

Readings

- Real-time Rendering 2nd Edition Chapter 6.12
- Real-Time Shadows , Eric Haines, Tomas M¨ oller, GDC 2001, CMP, 335-352
- Foley, Chapter 16.4
- Many OpenGL demos
- http://www.opengl.org/resources/code/samples/advanced/advanced96/programs.html