Computer Graphics

Shadows

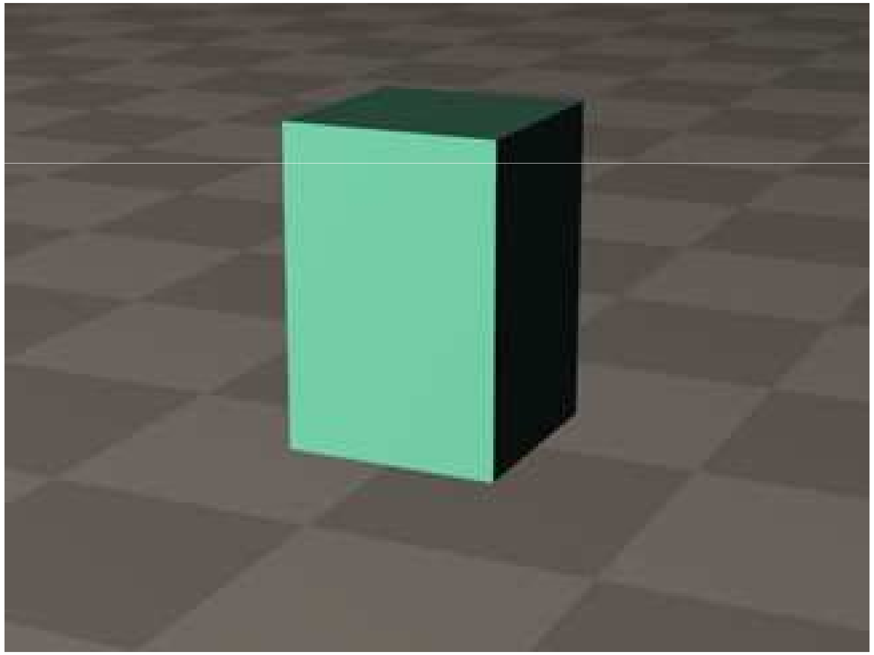
Taku Komura

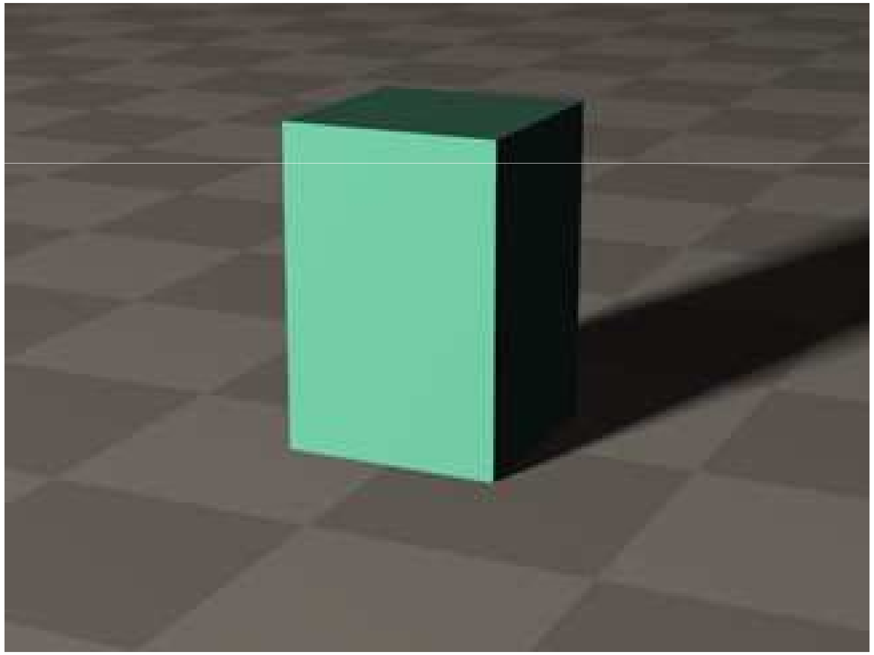
Today

- 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





Shadows and Motion

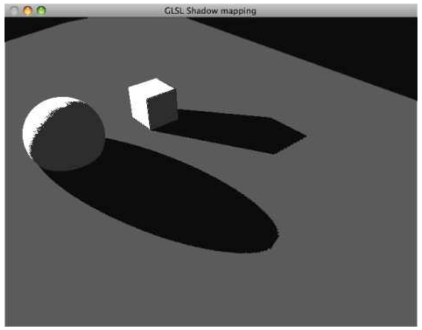
Humans conceive the motion of objects using shadows

Demo movie

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

Facts about Shadows

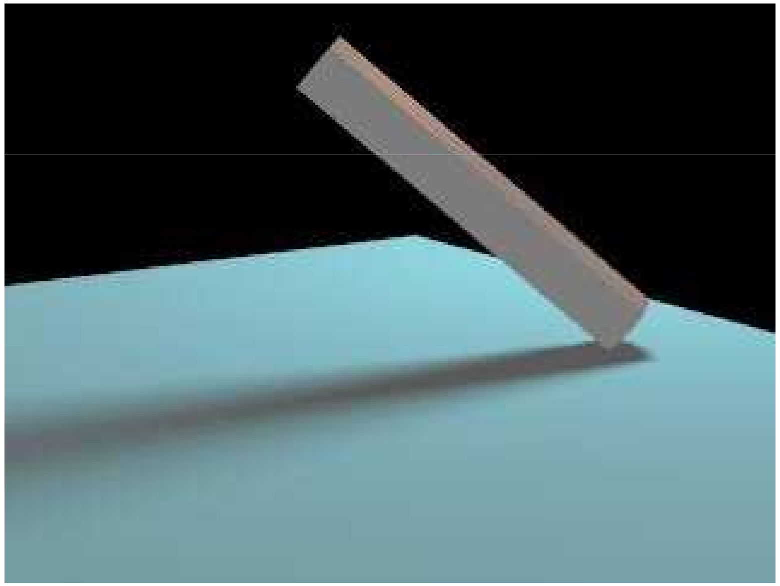
- Shadows can be considered as areas hidden from the light source



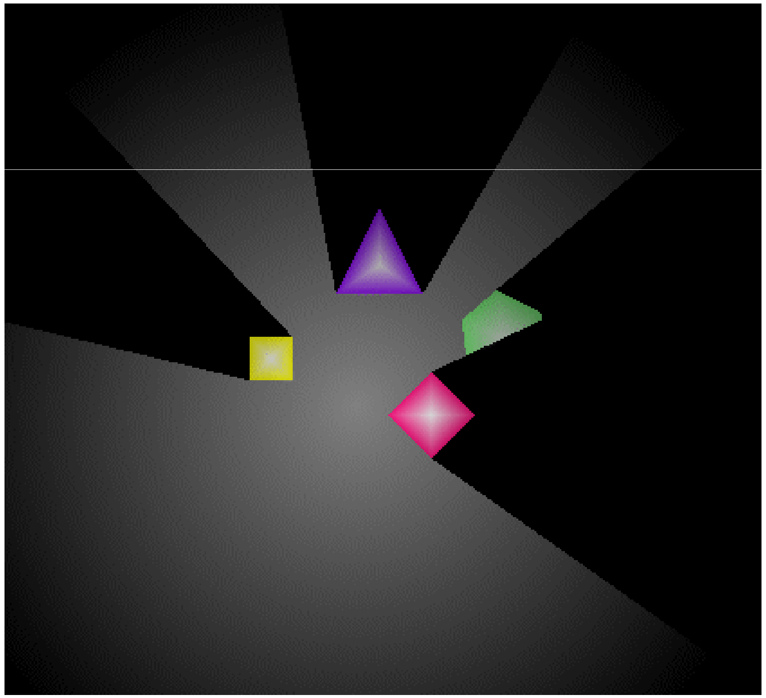
- A shadow on A due to B can be found by projecting B onto A with the light as the center of projection
  - Suggests the use of projection transformations
- Point lights have hard edges, and area lights have soft edges

Soft and hard shadows

Soft shadows



Hard shadows

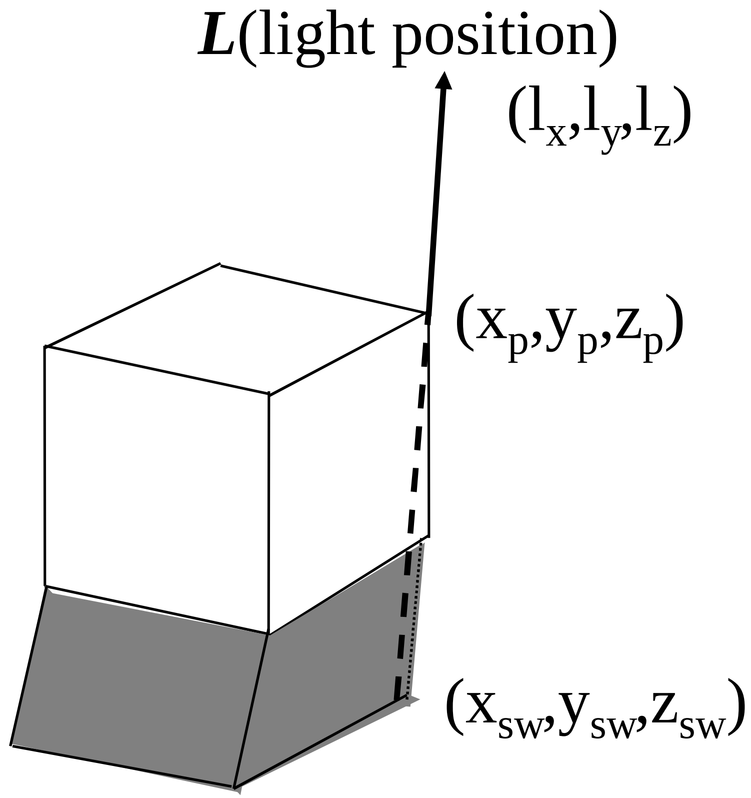


Today

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

Ground Plane Shadows

- Shadows cast by **point light** sources **onto planes** are an important case that is relatively easy to compute
  - Shadows cast by objects (cars, players) onto the ground

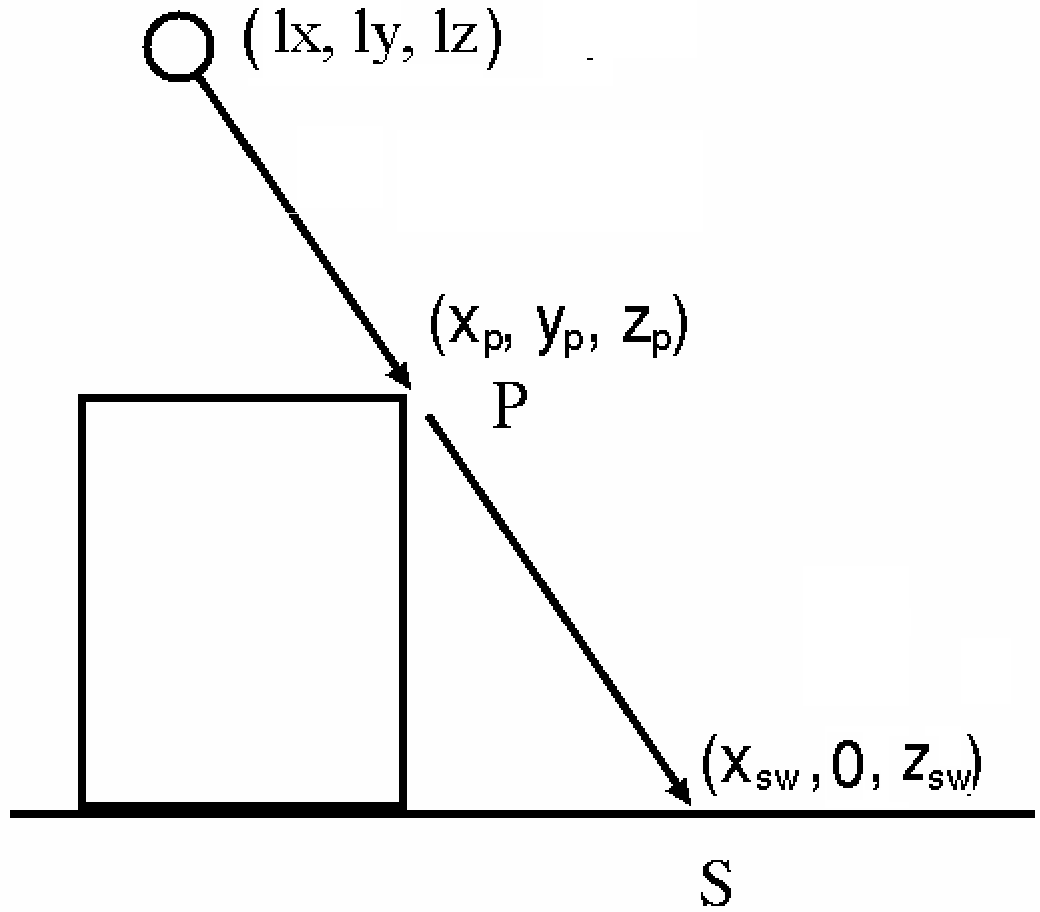


Point Light Shadows

Light source

Object

Floor



Point Light Shadows

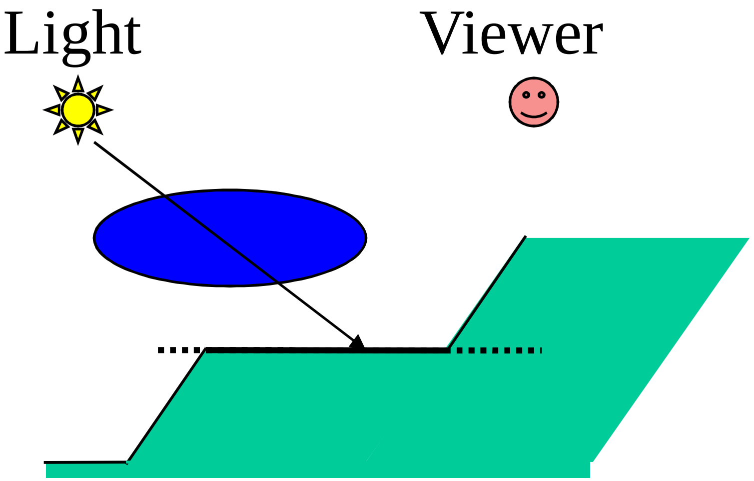
- Blinn ’88 gives a matrix that works for local point light sources
  - Takes advantage of perspective transformation (and homogeneous coordinates)

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

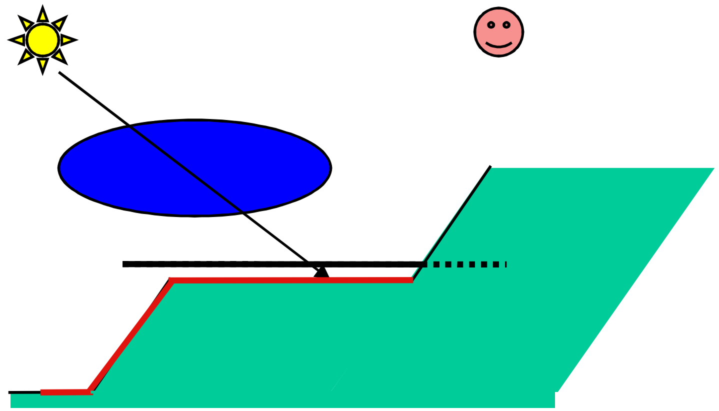
Drawing the Shadow

- We now have a matrix that transforms an object into its shadow
- Drawing the shadow:
  - Draw the polygon
  - Multiply the shadow matrix into the model transformation
  - Redraw the object in grey with blending on (making them translucent)
- Tricks:
  - Lift the shadow a little off the plane to avoid z-buffer quantization errors (can be done with extra term in matrix)

Lifting the shadow above the surface



Shadow height equal with hit polygon
Z-buffer quantization errors



Shadow height above hit polygon

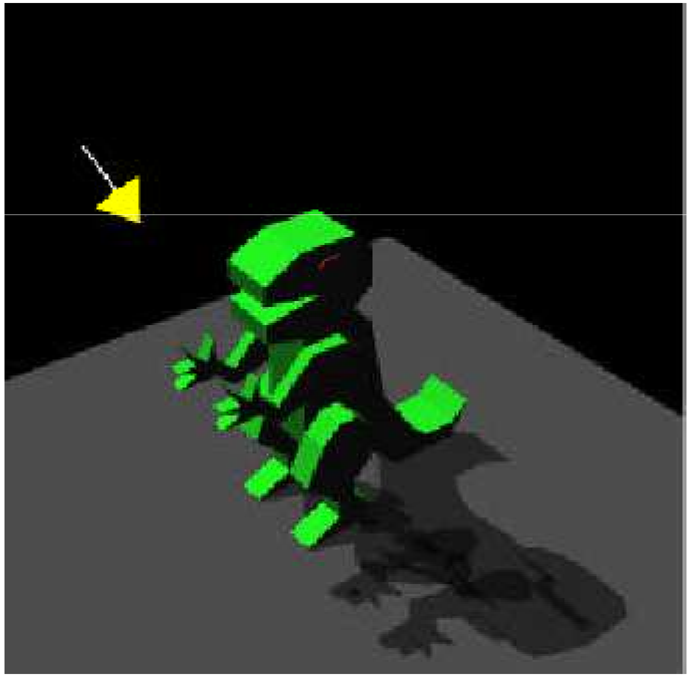
Point Light Shadows : problem

?

?

?

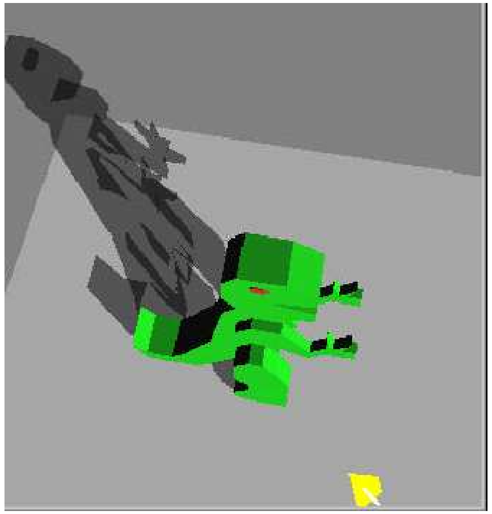
?

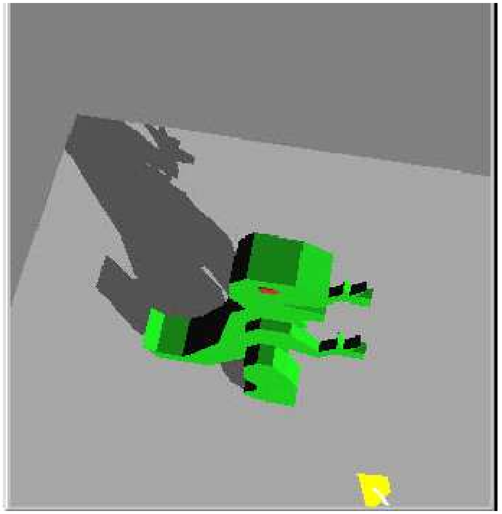


Improving Planar Projected Shadows with Stencil

- Only pixels tagged with the ground plane's unique stencil value will be updated when the shadow is rendered.
- When a shadow is rendered, the stencil value is set to zero
  - subsequent pixel updates will fail
- Lifting the shadows above the floor not necessary,
  - stencil buffer used to avoid rendering the floor over the shadow region

But still we can only shadow on planes …

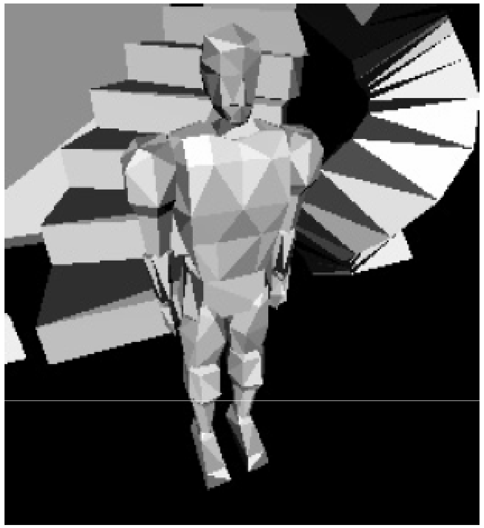
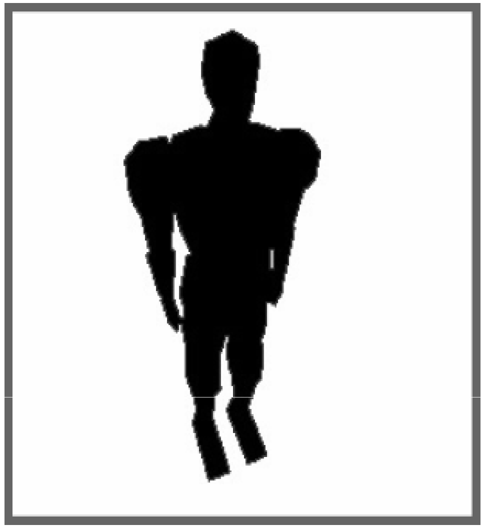
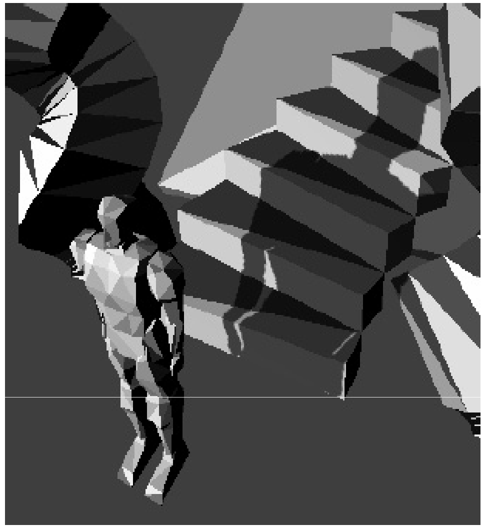




Today

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

Shadow Texture

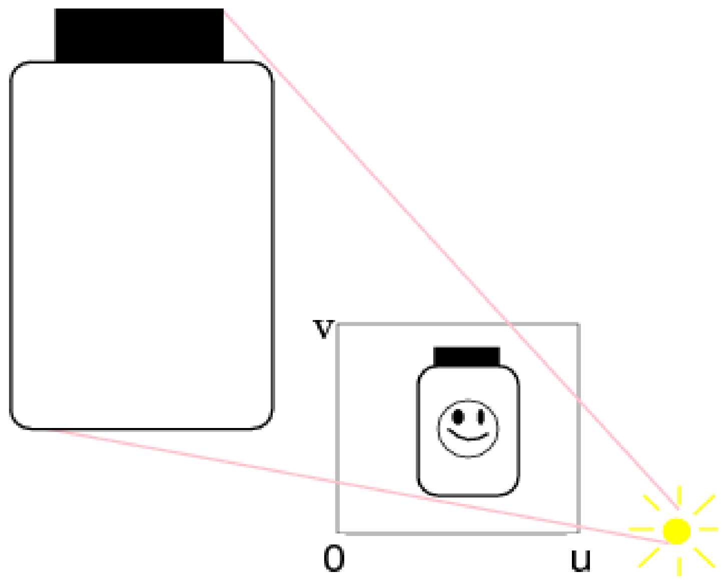




- Use a shadow image as a projective texture
- Generate an image of the occluder from the light’s view and color it grey
- Produce a shadow by texture mapping this image onto the background object
- Can render shadows over curved surfaces

Q: How do we compute the uv coordinates?
A: Use projective textures

- View the object from the light source
- Project the background object onto the projection plane (texture coordinate) and compute the uv coordinates per vertex
- Interpolate the uv coordinates inside the triangles using hyperbolic interpolation

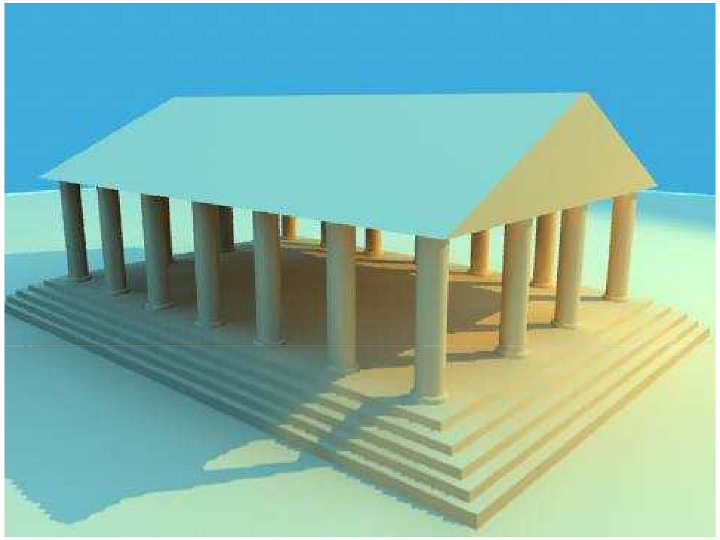
Projective textures are also useful for light mapping



Shadow Texture : Cons and Pros

Pros

- ?
- The shadow does not need to be recomputed if the occluder does not move.

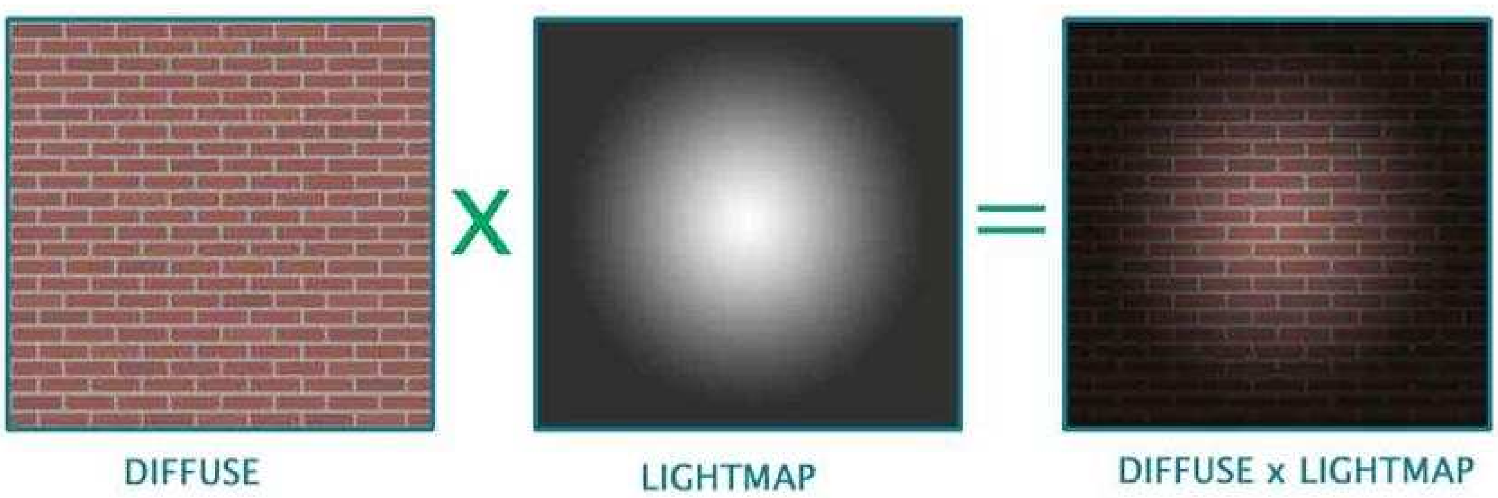


Cons

1. ?
2. ?
3. ?

Light mapping

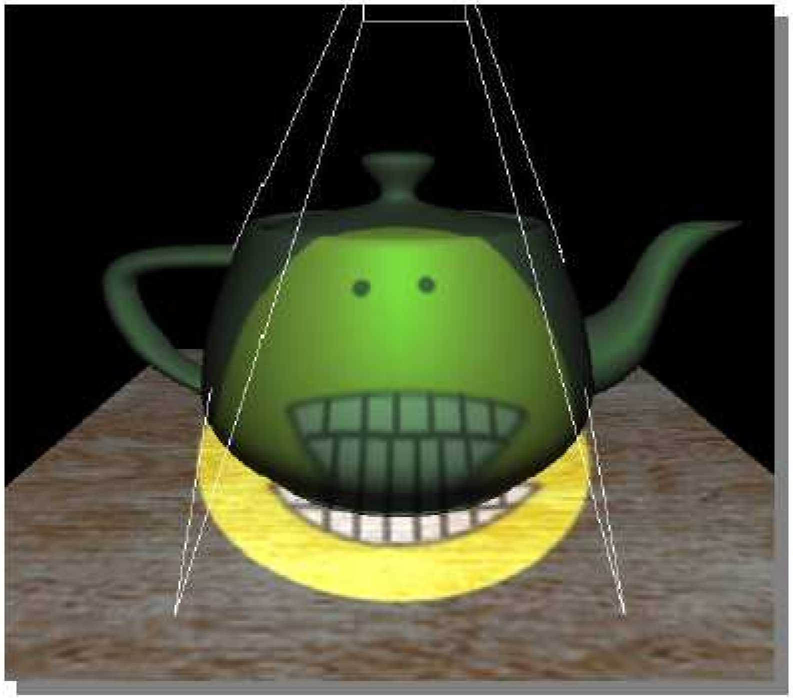
- Phong shading is expensive as need per pixel lighting
- Light mapping can produce Phong shading-like effects (like spot lights) using texture mapping
- Can produce spotlights on low count polygons
- Multiply the light map to the original diffuse texture and map to the surface

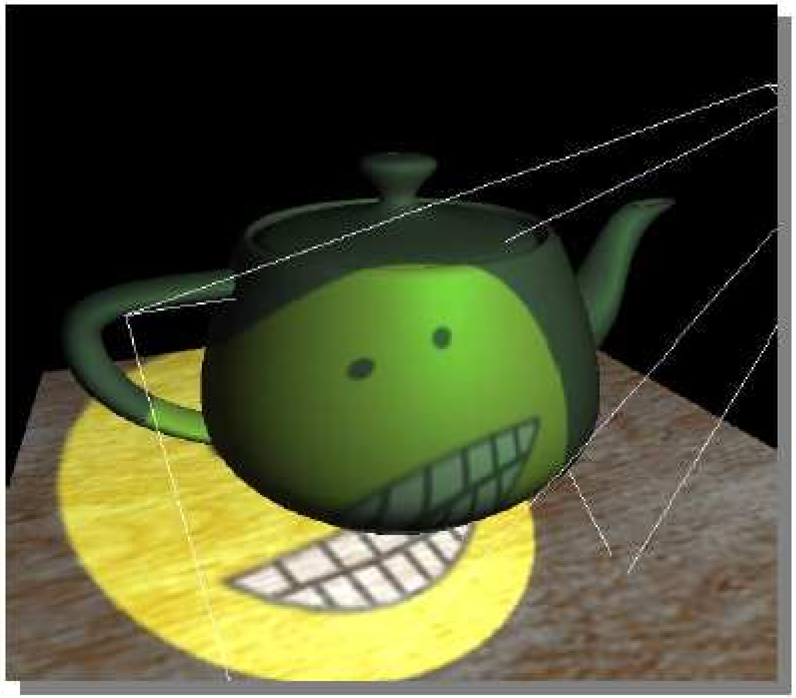


Light maps by projective textures

Compute the uv mapping by projecting the object onto the texture coordinates

Multiply the light map to the texture / diffuse colour





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

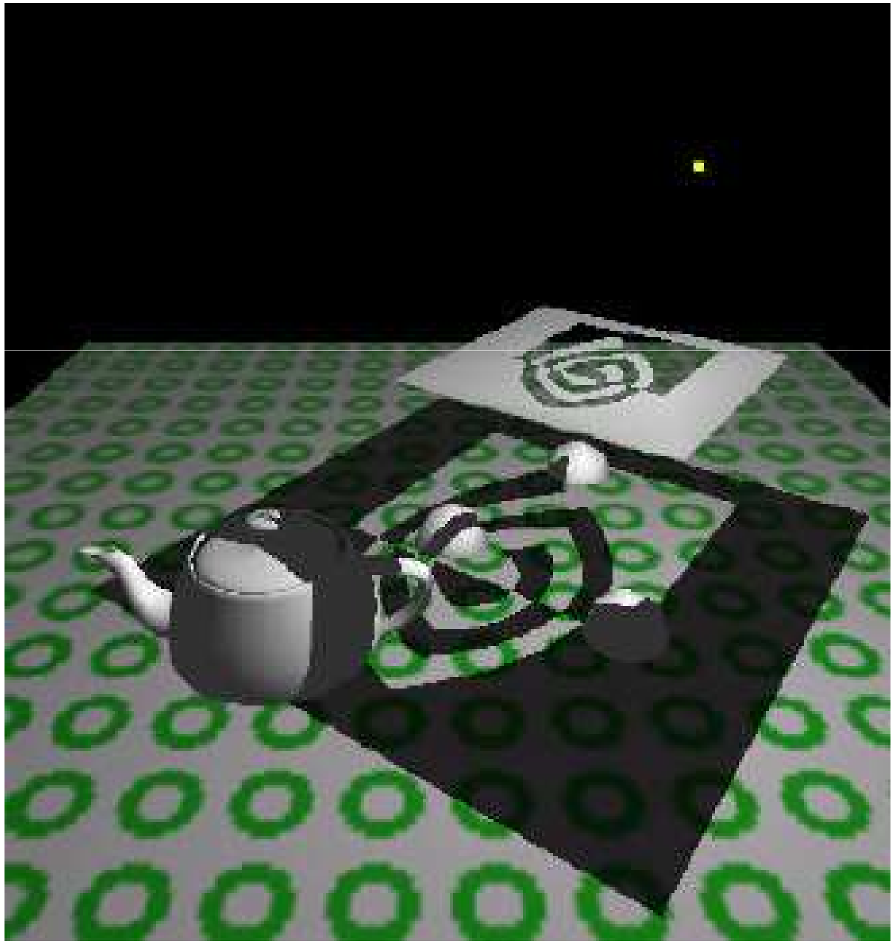
Figure 2: A conventional 2,048×2,048 pixel shadow map (left) compared to a 16 MB ASM (right). EFFECTIVE SHADOW MAP SIZE: 65,536×65,536 PIXELS.

Today

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

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
- Check whether the point is inside / outside the shadow volume
  - inside → shadowed
  - Outside → illuminated by light source

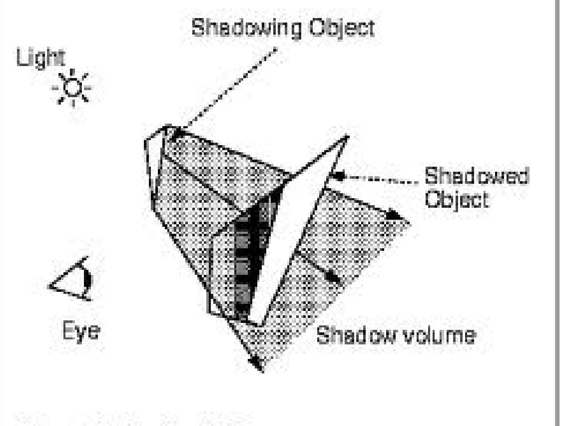
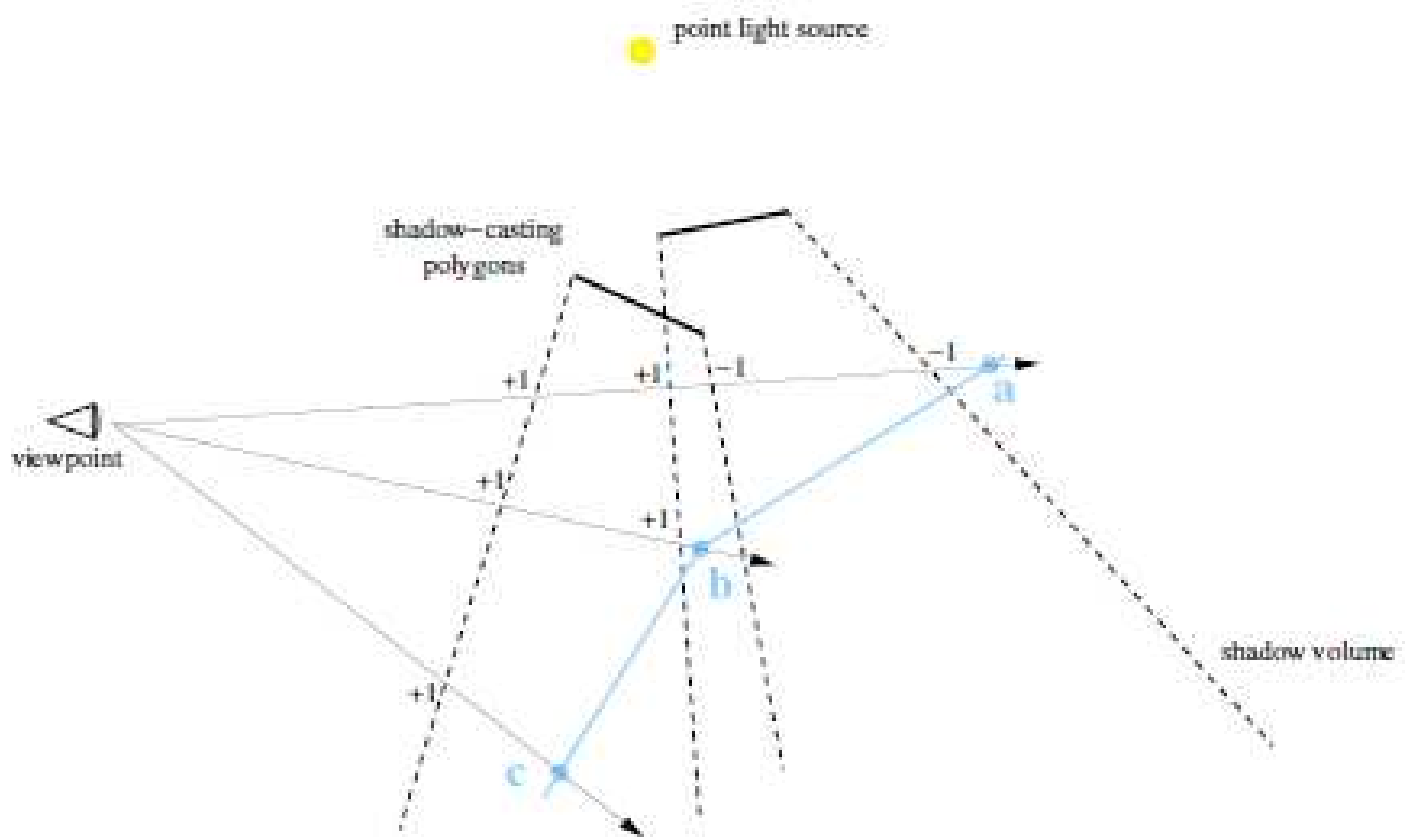


Figure 32. Shadow Volume

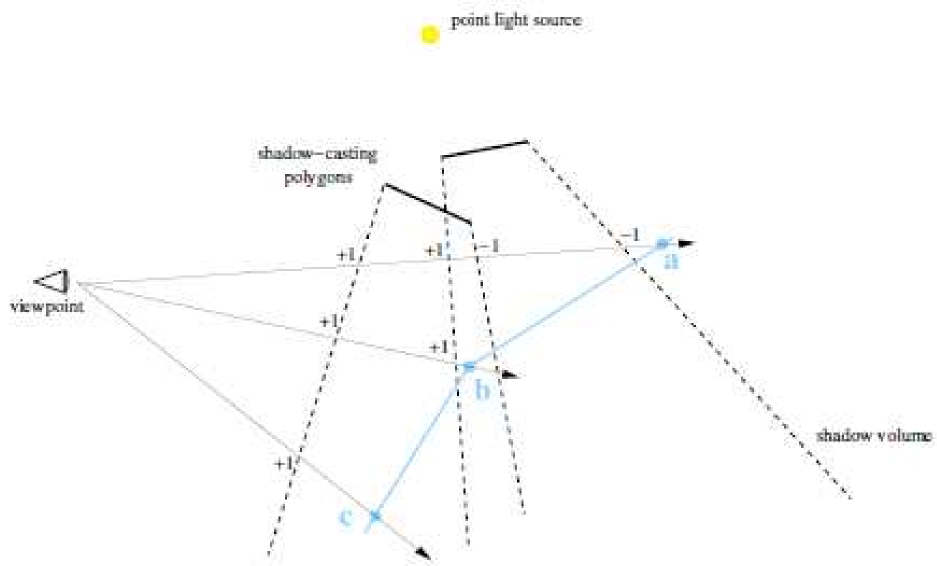

point light source
shadow−casting
polygons
viewpoint
+1
+1
−1
−1
+1
+1
+1
a
b
c
shadow volume

Procedure

- Get the polygonal boundary representation for the shadow volume
- 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
  - Whenever a rendered fragment of the shadow volume is closer than the depth of the other objects, decrement the stencil value for that pixel
  - After this process, if the stencil count is 0, that point is outside the shadow

Procedure

- After rendering, if the stencil value is zero, it is outside the shadow, so must be lit
- Otherwise inside the shadow
- Render the scene with the diffuse & specular components active, but only for pixels the stencil is zero



Advantage / Disadvantages of Shadow Volume

- Advantage
  - Do not need to manually specify the shadowed objects
  - The occluder can shadow itself

- Disadvantage
  - Bottle neck at the rasterizer
  - Many shadow volumes covering many pixels

Today

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

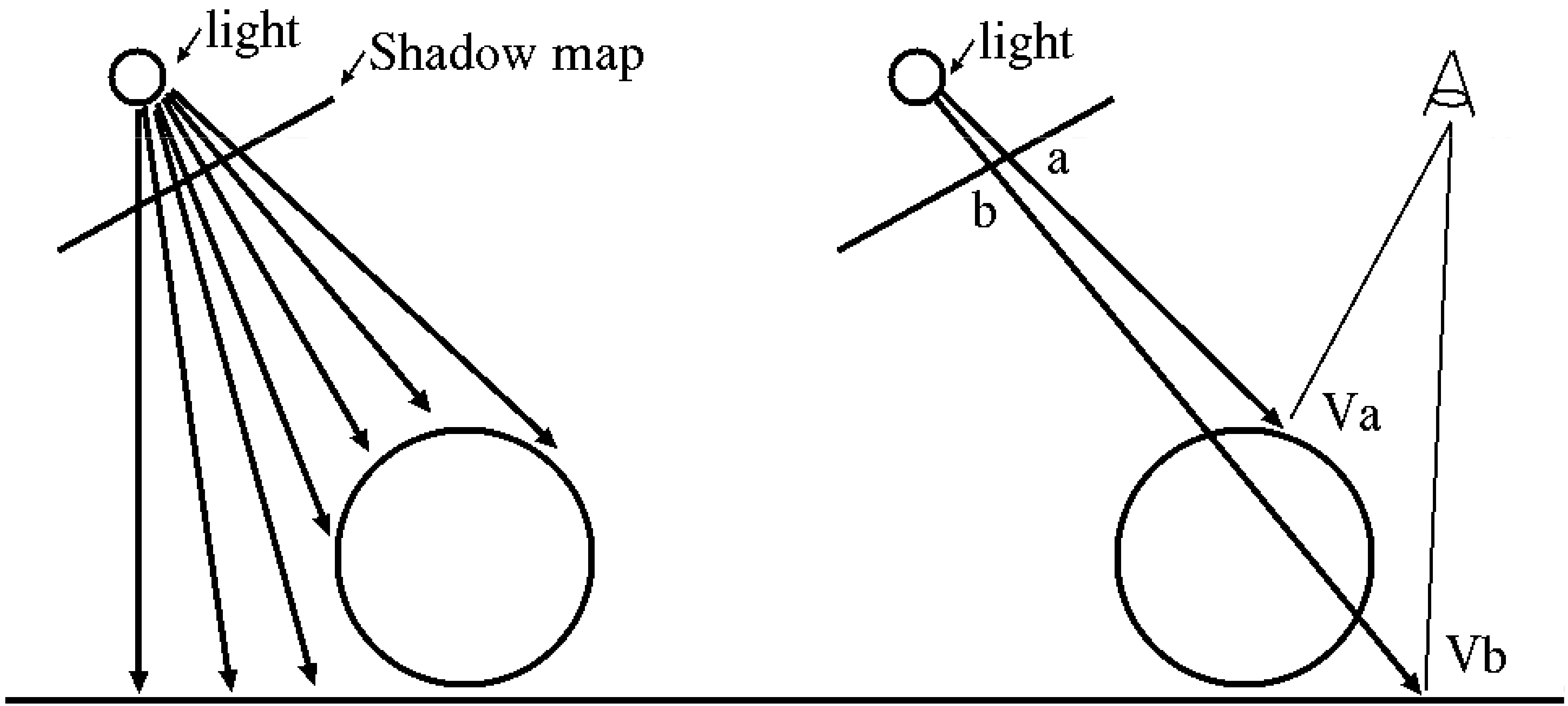
Shadow Mapping

- A method using the Z-buffer

- Render the scene from the light source using the Z-buffer algorithm
  - The Z-buffer contains the distance to the object
  - *shadow depth map / shadow buffer*
- Render the scene from the view point
  - Compute the coordinates of the sampled points in each light space
  - If the rendered vertex is farther away from the value in the Z-buffer, it is in the shadow

Shadow Map

- Checking whether Va,Vb is closer to the light



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

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

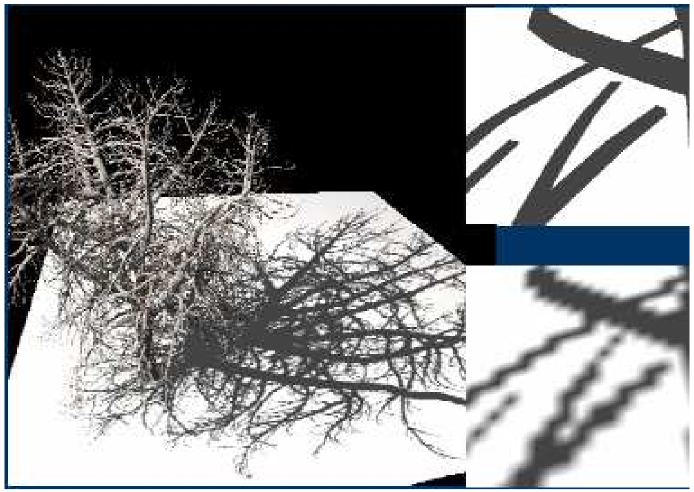
Shadow Map

Advantage

- Don't need a stencil buffer
- When there are many shadows, it is faster then shadow volume – why?

Disadvantage

- How is the precision compared with shadow volume?

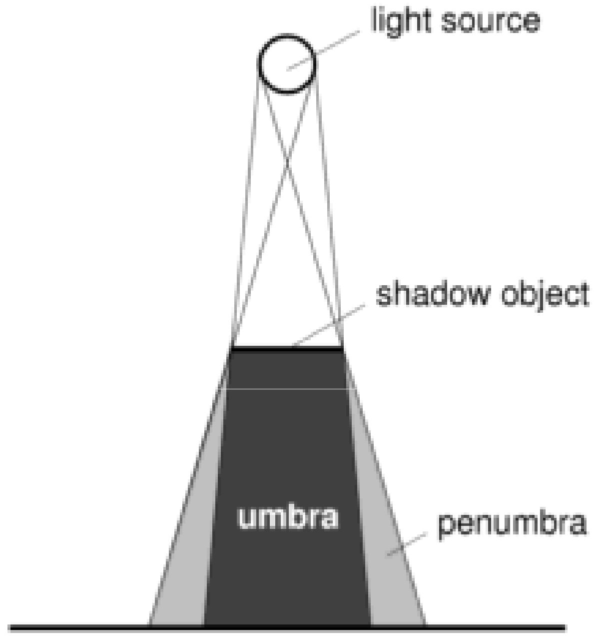


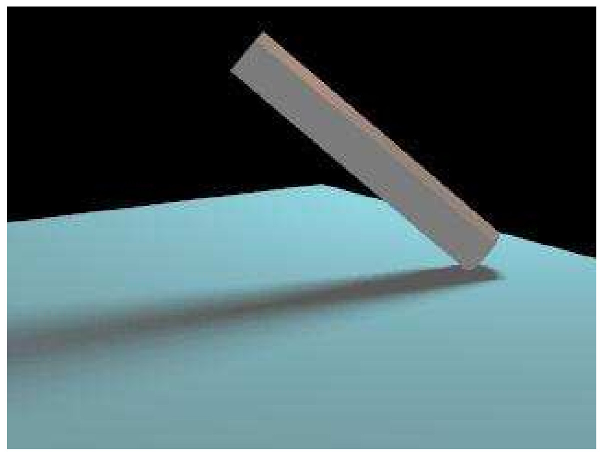
Today

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

Soft Shadows

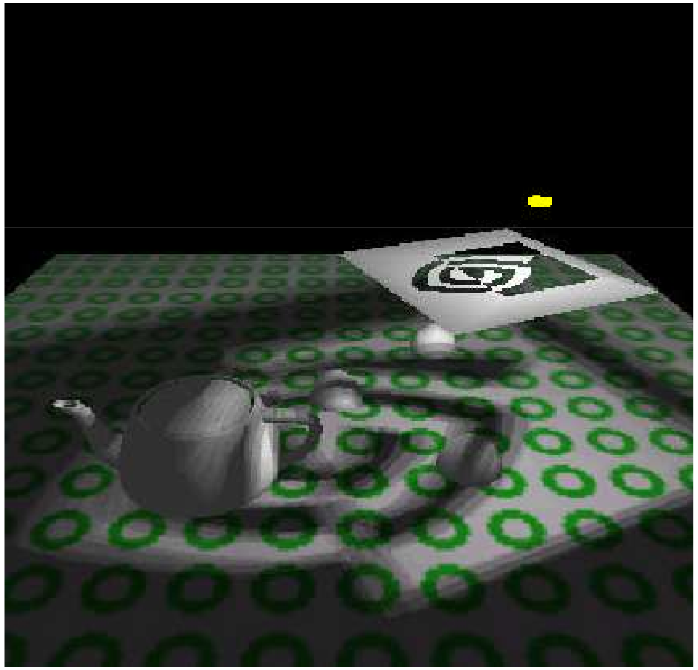
- Made by area light
  - umbra – totally blocked from the light source
  - Penumbra – partially blocked from the light source
- Can be modelled by a collection of point light sources





Soft shadowing by multiple point light sources

- Additive blending is used to accumulate the contribution of each light.
- 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
- Apply convolution
- Can apply both planar projected shadows approach or shadow volume approach
- Drawback: slow

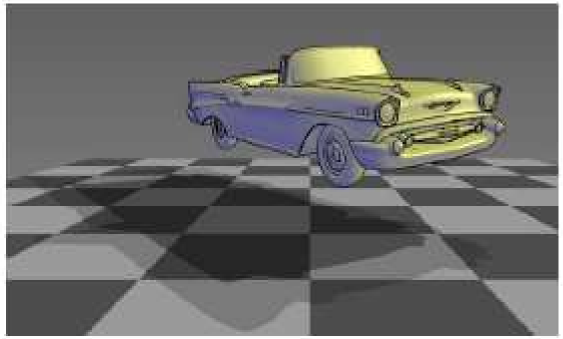


Other techniques to generate soft shadow : Gooch et al.

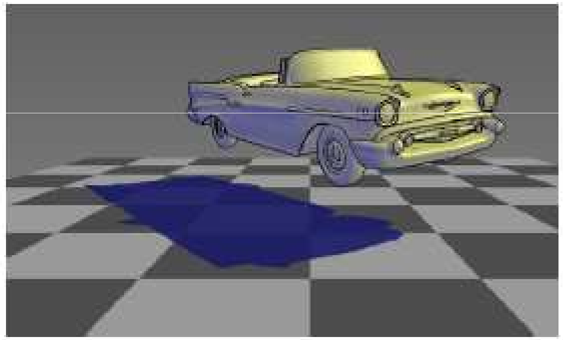
- Moving the projected plane up and down instead of moving the light source
- The projections cast upon it are averaged
- Can use projective shadows :

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

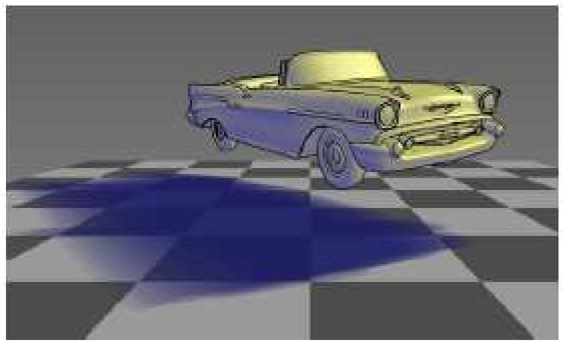
  Gooch et al. I3D 1998



(a) Hard penumbra and hard umbra.



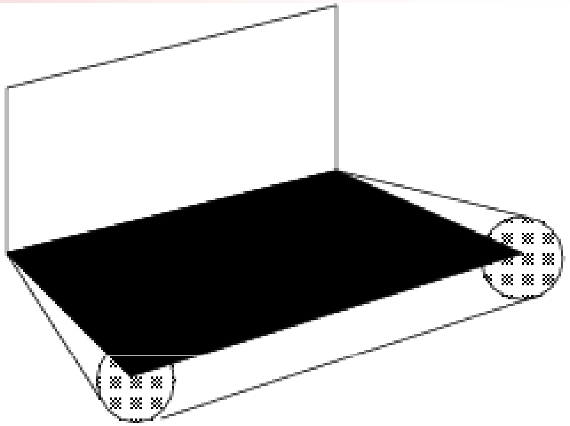
(b) Single hard, colored shadow.



(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



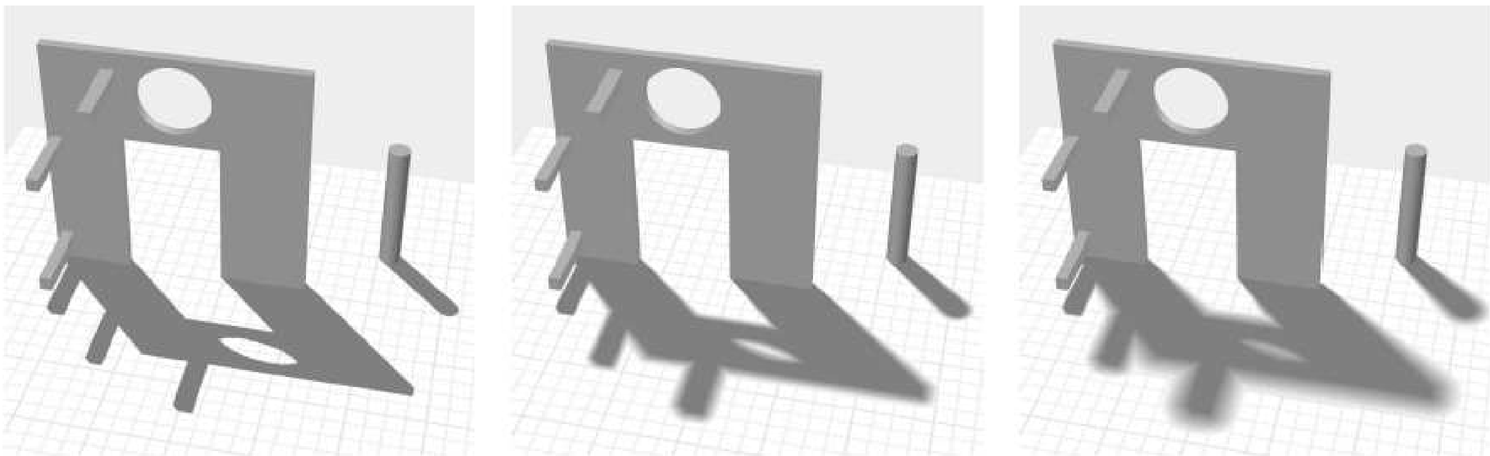
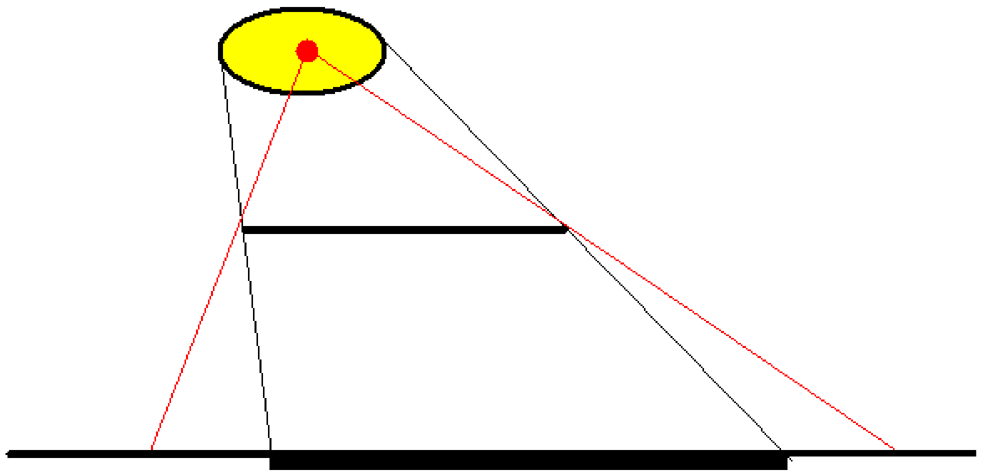


Figure 3: On the left is a hard shadow, the middle shows the effect of a small area light source, the right a larger light source.

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



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