



# Shadows & occlusion

- Shadow - occlusion duality
- Floor shadows
- Shadow buffer
- Soft shadows

# Motivation

---

- **Visibility**
  - Most objects in a scene are hidden from any given view point
  - Rendering them wastes GPU cycles
  - Need efficient techniques for rejecting what is obviously hidden
    - Rejection tests can be exact, conservative (reject more), approximate
- **Shadows**
  - Objects in the shadow do not reflect direct illumination
  - Want to know which light sources illuminate an object
  - Need a quick test for being in the shadow
- **Visibility and shadows are defined in terms of occluders**
  - Surfaces or objects that may block the passage of light
  - When placed between the object and the viewer/light

# Shadows provide depth clues

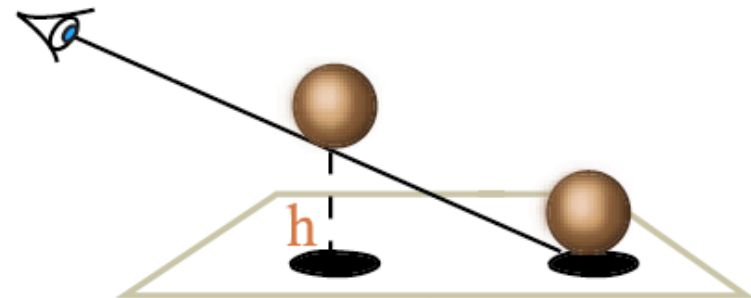
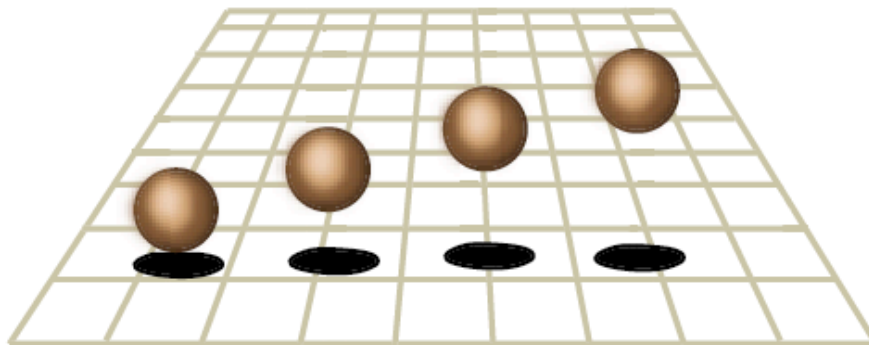
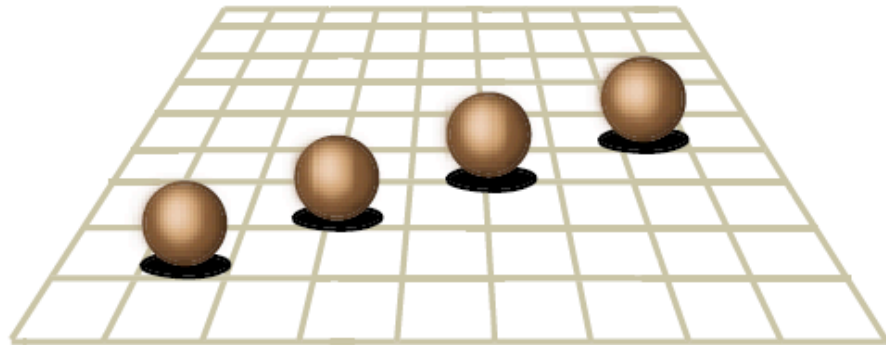


Image adapted from:

Palmer, Stephen E. *Vision Science: Photons to Phenomenology*. MIT Press. May 1999. ISBN: 0-262-16183-4.

# 4x4 matrix formulation (review)

u	0	0	0
0	v	0	0
0	0	w	0
0	0	0	1

$S(u,v,w)$

1	0	0	u
0	1	0	v
0	0	1	w
0	0	0	1

$T(u,v,w)$

$U_x$	$V_x$	$W_x$	0
$U_y$	$V_y$	$W_y$	0
$U_z$	$V_z$	$W_z$	0
0	0	0	1

$R(U,V,W)$

C	-S	0	0
S	C	0	0
0	0	1	0
0	0	0	1

$R_Z(a)$

$U_x$	$V_x$	$W_x$	u
$U_y$	$V_y$	$W_y$	v
$U_z$	$V_z$	$W_z$	w
0	0	0	1

$M \cdot P$

x
y
z
1

=

$U_x x + V_x y + W_x z + u$
$U_y x + V_y y + W_y z + v$
$U_z x + V_z y + W_z z + w$
1

# Rotation matrix

- $\|U\|=\|V\|=1$
- $U \cdot V = 0$
- $W=U \times V$
- $R = \{ U \ V \ W \}$ , 3x3 matrix
- $R' = \triangleleft R$ , inverse = transpose
- Rotation to map Z to W
  - $U := Y \times W$  or  $X \times W$
  - $U := U / \|U\|$
  - $V := W \times U$
  - $R_{ZtoW} = \{ U \ V \ W \}$
- Rotation to map W to Z axis
  - $\triangleleft R_{ZtoW}$

$U_x$	$V_x$	$W_x$	<b>0</b>
$U_y$	$V_y$	$W_y$	<b>0</b>
$U_z$	$V_z$	$W_z$	<b>0</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>

$U_x$	$V_x$	$W_x$
$U_y$	$V_y$	$W_y$
$U_z$	$V_z$	$W_z$

R

inverse

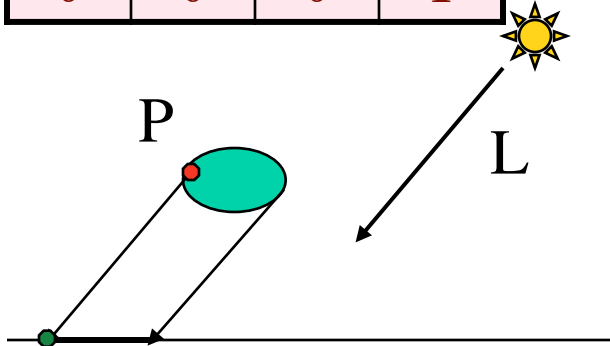
$U_x$	$U_y$	$U_z$
$V_x$	$V_y$	$V_z$
$W_x$	$W_y$	$W_z$

R'

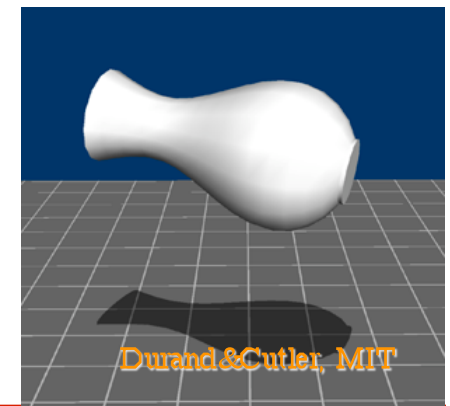
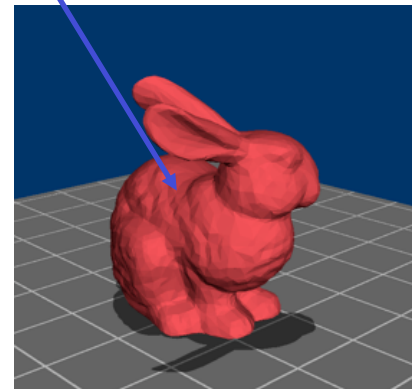
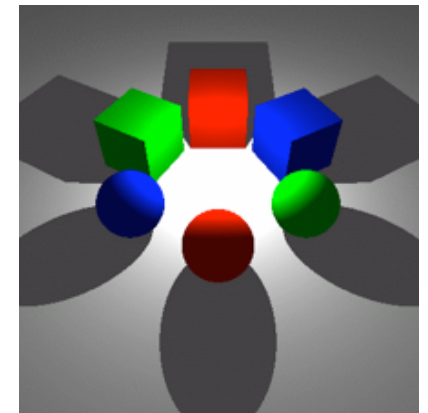
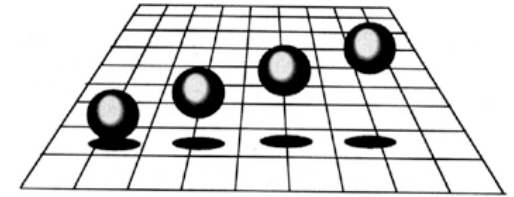
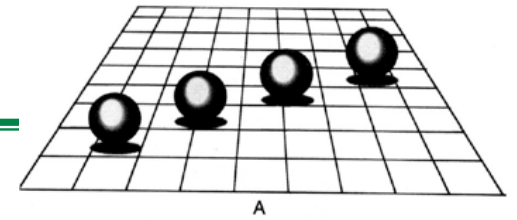
# Floor shadows

- Depth cue, realism, light position
- Draw object twice
  - Second time: projected on the ground
- **Does not support self-shadows**

1	0	$L_x/L_z$	0
0	1	$L_y/L_z$	0
0	0	0	0
0	0	0	1



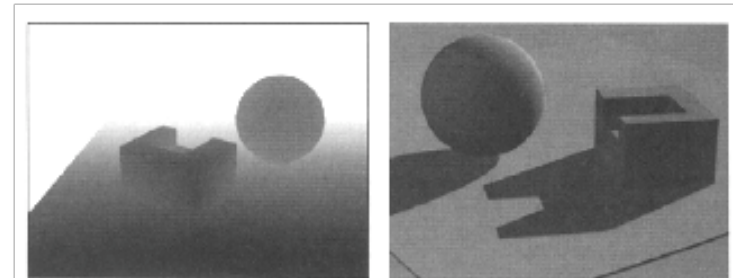
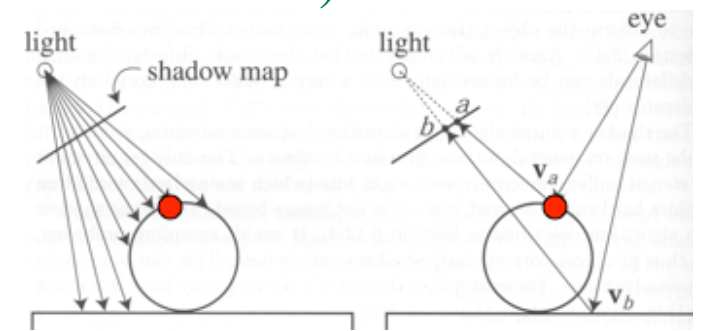
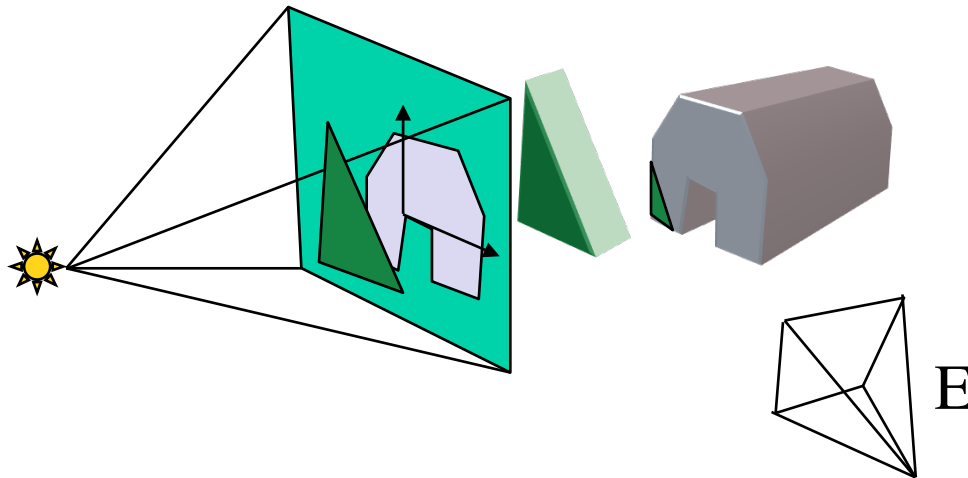
$$S = P + (P_z/L_z) L$$



# Shadow map (supports self-shadows)

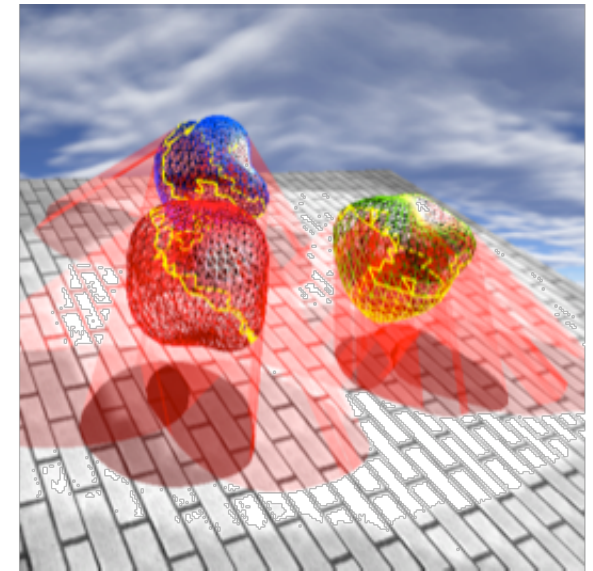
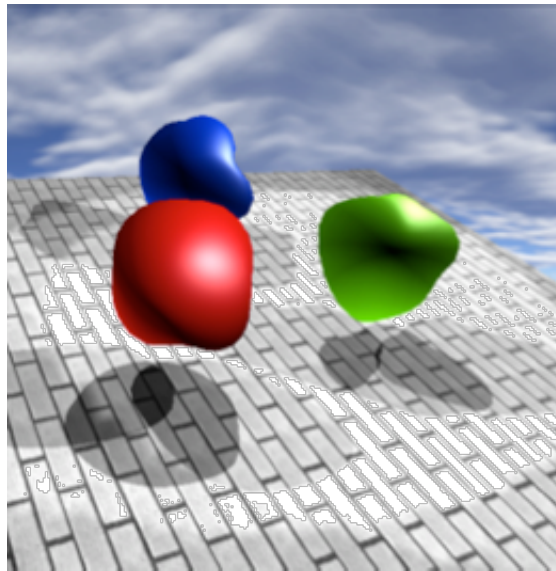
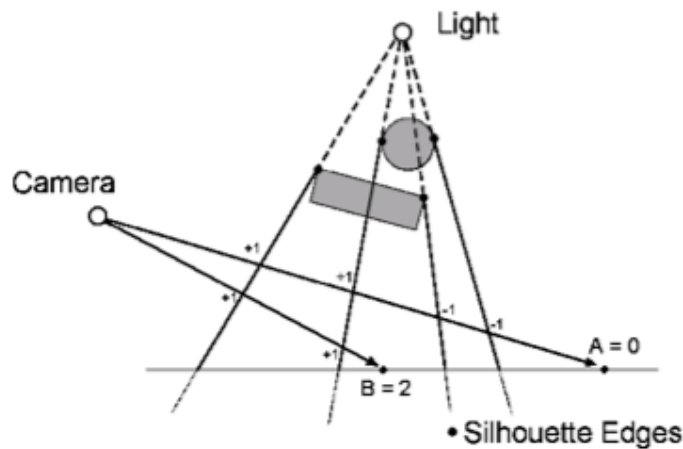
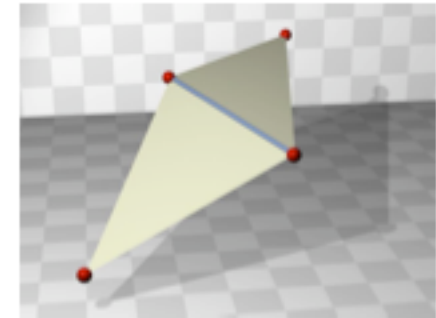
- Pre-render from light : store z-buffer as shadow map (texture)
- While rendering, check whether fragments are in shadow
  - Use GPU to perform the check (compare z to texture)

Shadow-casting  
fragments must be in  
light frustum



# Shadow volumes on GPU

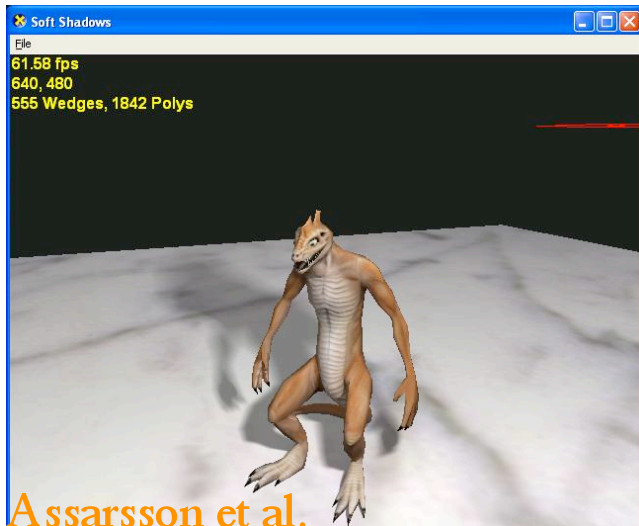
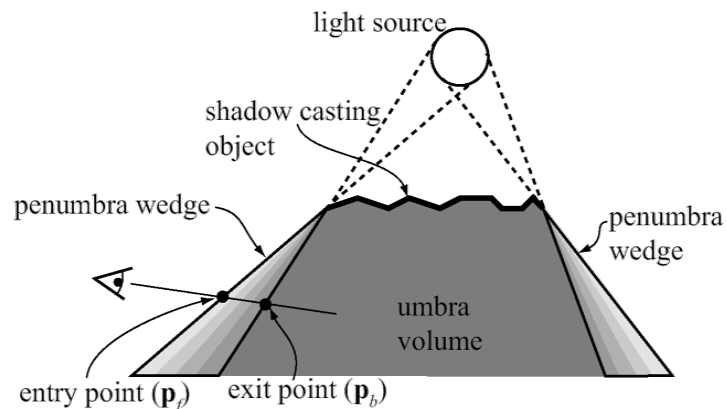
- Heidmann (IRIS Universe 1991), Everitt (nVidia 2002)
  - Shadow volume = triangles facing light + silhouette extrusions
- Brabec-Seidel (EUROGRAPHICS 2003)
  - Add silhouette identification in hardware



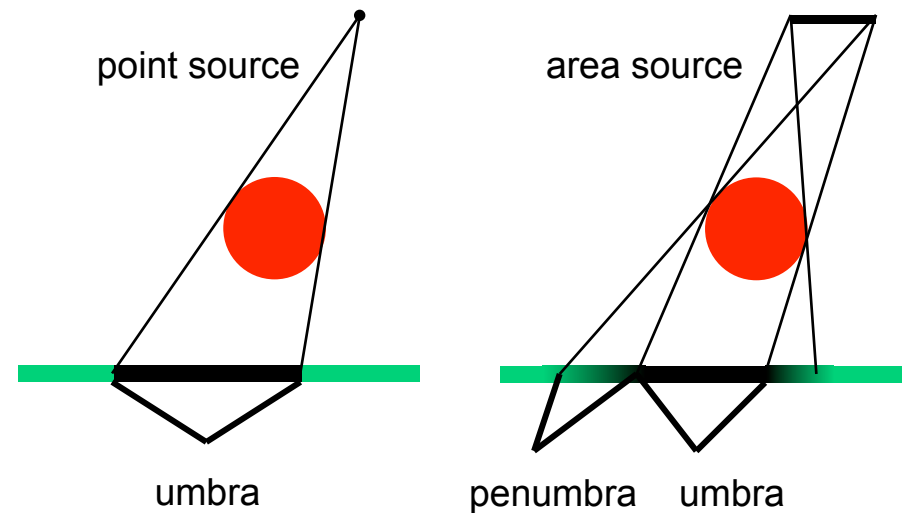


# Soft shadows (area light sources)

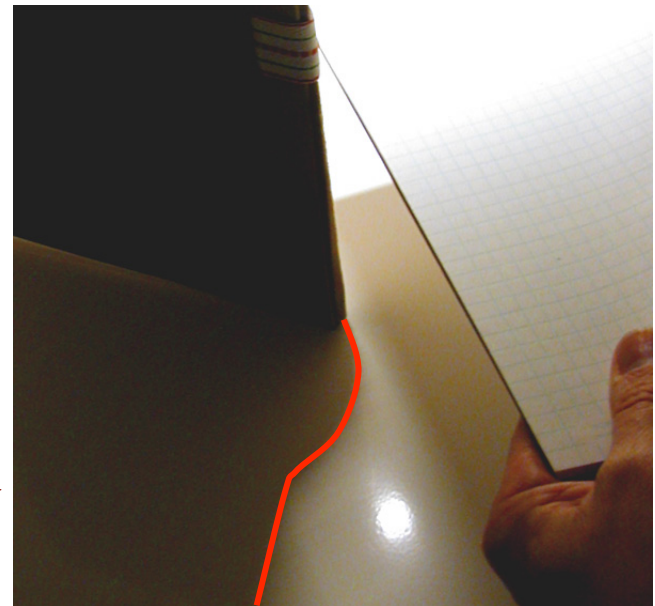
- Polygonal area light source



Assarsson et al.



Polygons  
cast curved  
shadows



# Polygons cast curved shadows!

---

