The
University
Of
Sheffield.

COM3503/4503/6503:

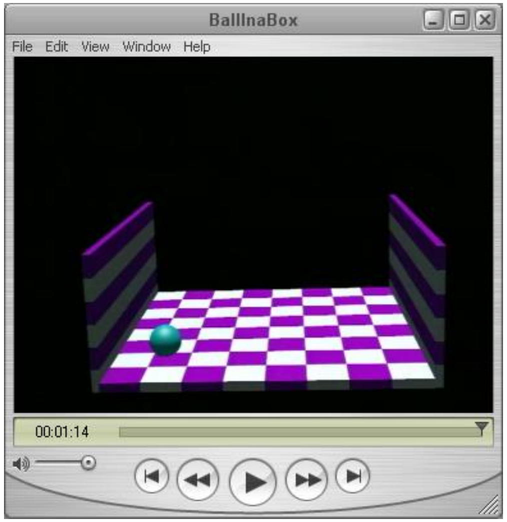
3D Computer Graphics

Lecture 9: Geometric shadows

Dr. Steve Maddock
Room G011, Regent Court
s.maddock@sheffield.ac.uk

1. Introduction

- Shadows increase realism of 3D shaded objects
- Without shadows objects can appear to float
- Shadows give depth cues
- Size depends on light position

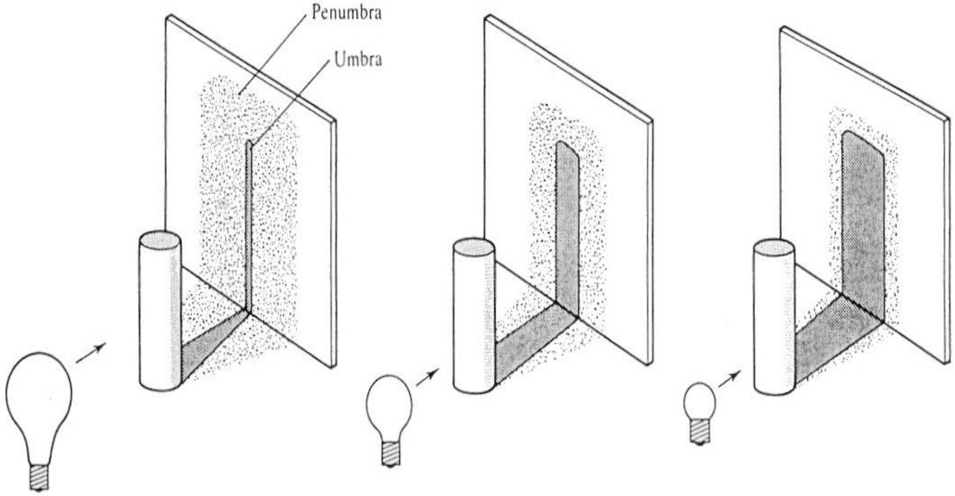


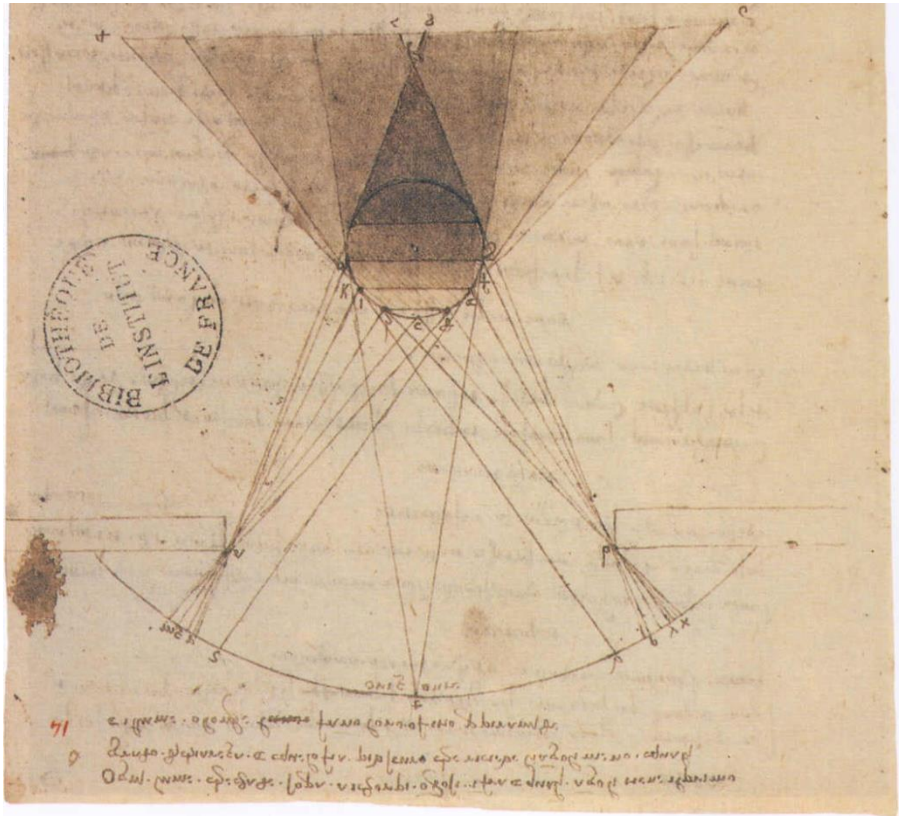
http://gandalf.psych.umn.edu/users/kersten/kersten-lab/demos/shadows.html

Thomas and Johnston, 81

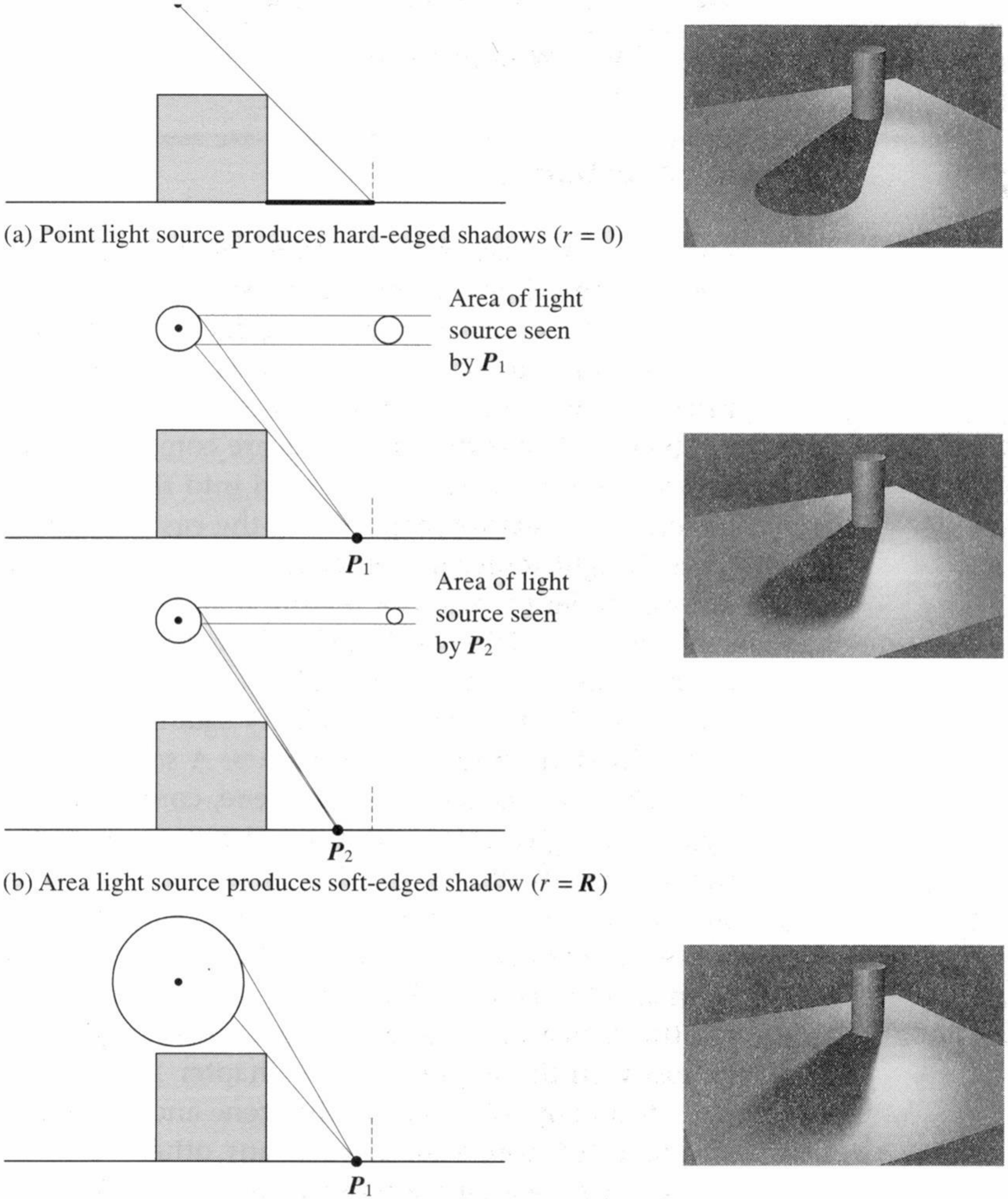
1. Introduction

- Umbra – receiving surface cannot see any part of the light source
- Penumbra – can see some of the light





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°).



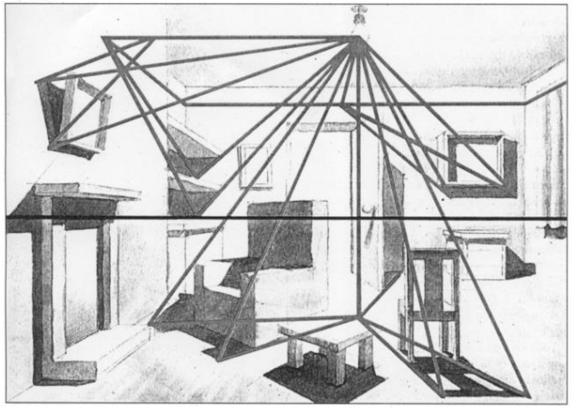
(a) Point light source produces hard-edged shadows ($r = 0$)

(b) Area light source produces soft-edged shadow ($r = \boldsymbol{R}$)

(c) Increasing the light source area softens the shadows more ($r = 2\boldsymbol{R}$)

1. Introduction

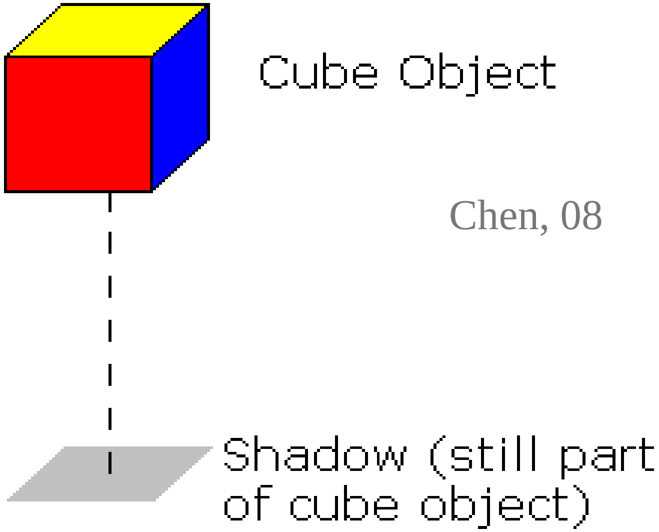
- ‘Geometric shadows’ – Calculate shape of shadow but only guess at its reflected light intensity (Most common approach)
  - Need global illumination algorithm for better light intensity calculation – see later lectures
- Exploit:
  - Shadows are areas hidden from the light source (implies use HSR algorithms)
  - A point light source means no penumbra
  - Shadows do not change in a static scene
- High computational overheads
  - Now ameliorated by GPU
  - Shadows regarded in same way as texture mapping – a quality add-on for standard polygon mesh rendering pipeline



2. Simple shadows

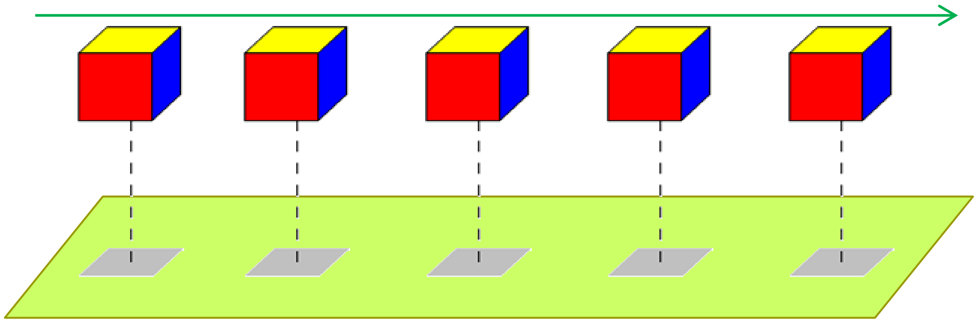
2.1 Fake shadows

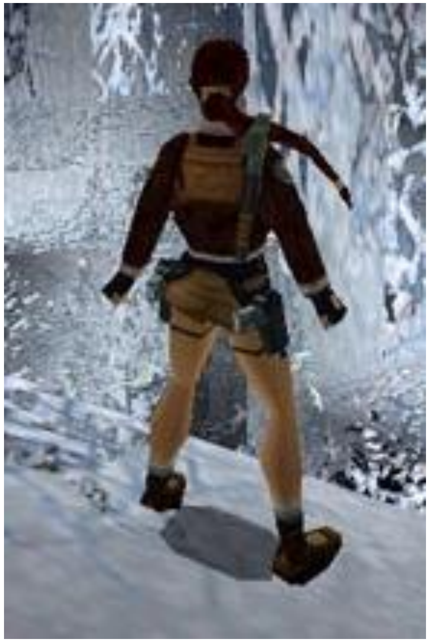
- No exact calculation
- Approximation of shadow position and shape
- Estimated by centre or anchor of object



Chen, 08

- Adv: simple; fast
- Disadv: need flat ground; not exact



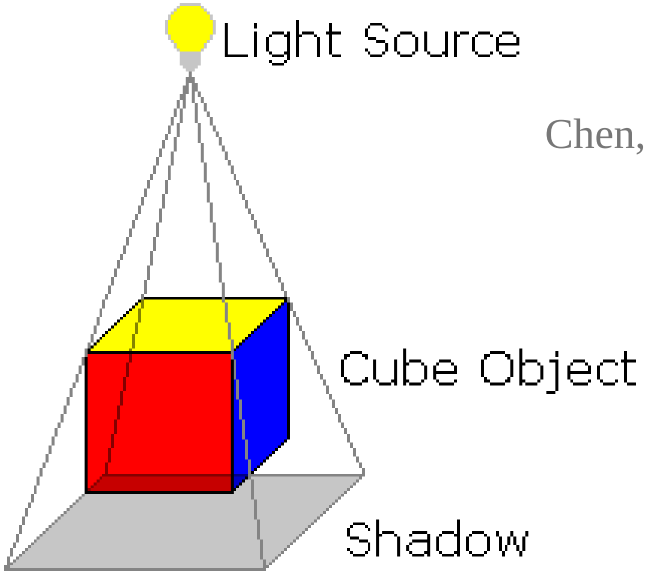


Tomb Raider by Eidos

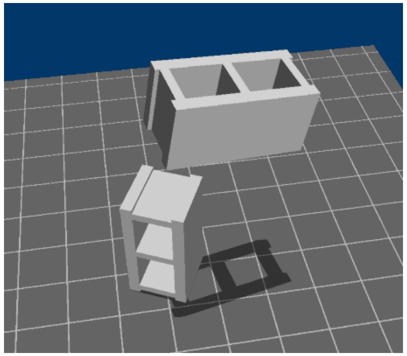
2.1 Vertex projection

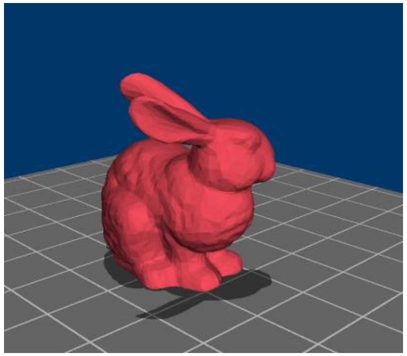
- (see Blinn, 88)
- Object projected to ground
- Exact mathematical calculation

- Adv: simple; exact
- Disadv: flat surface only; no self-shadowing



Chen, 08





Durand, 08

Demo

3. Shadow Algorithms in CG

- Early approaches began with the early rendering approaches
  - E.g. integration with scan-line algorithm (Appel, 68)

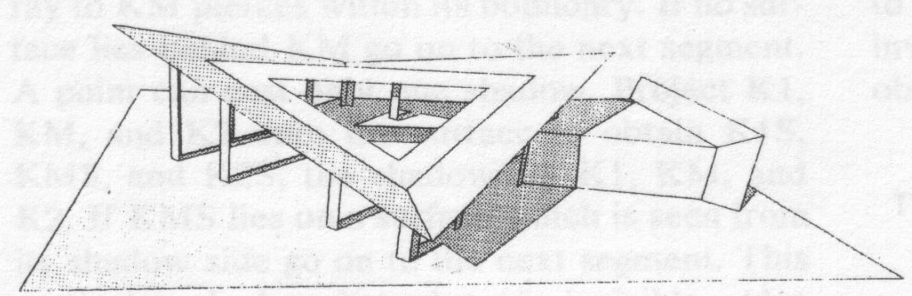


Figure 5 — A higher angle view of the building. 7094 calculation time for this picture was about 30 minutes.

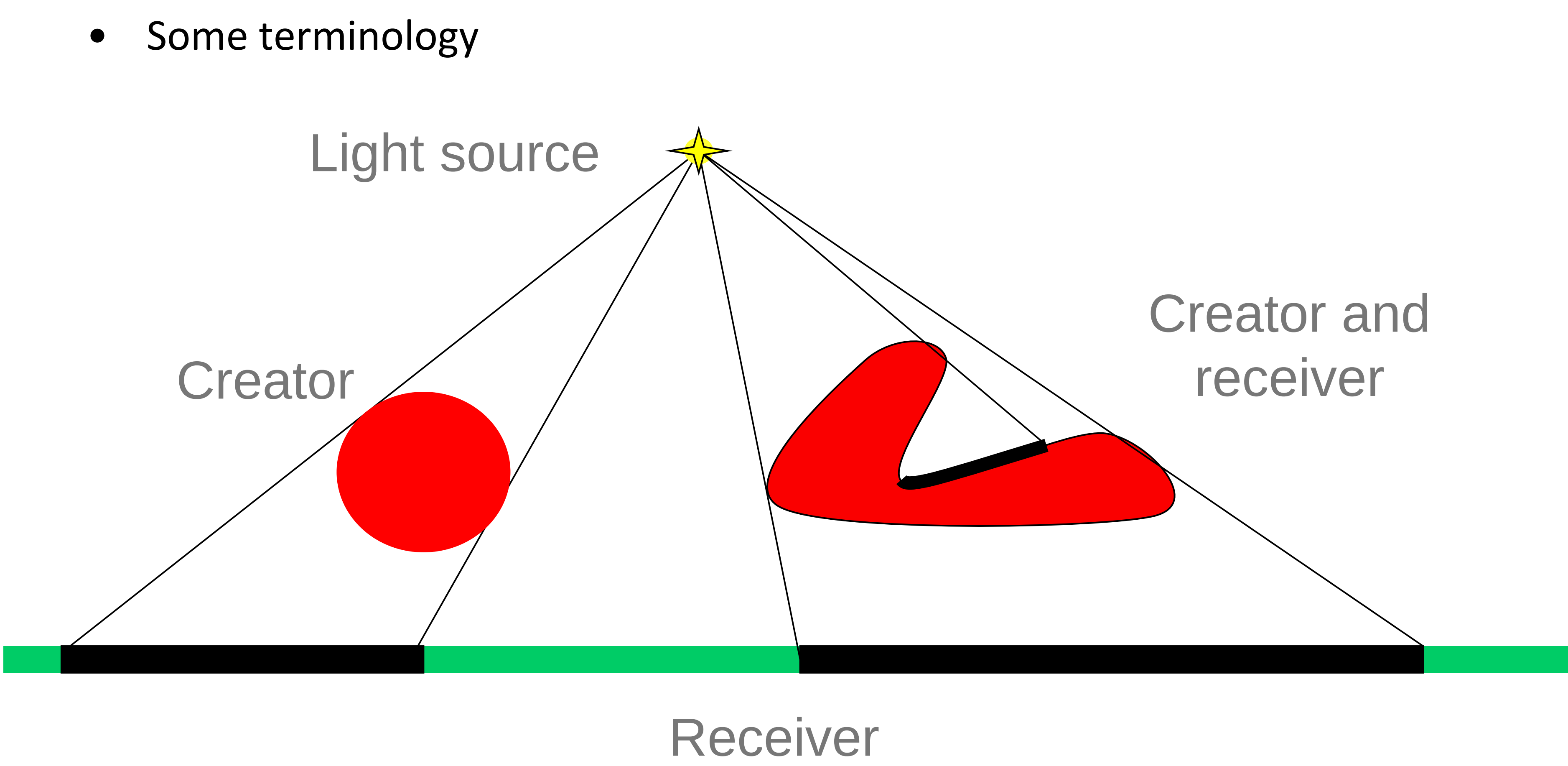
Arthur Appel, IBM Research, 1968

- Now, two most common approaches are:
  - Shadow volumes
  - Shadow z-buffer
- In general, two stages to both of these approaches:
  1. View independent – generate shadow info
  2. View dependent – render

- Global rendering algorithms automatically incorporate shadows with proper lighting – see later lecture

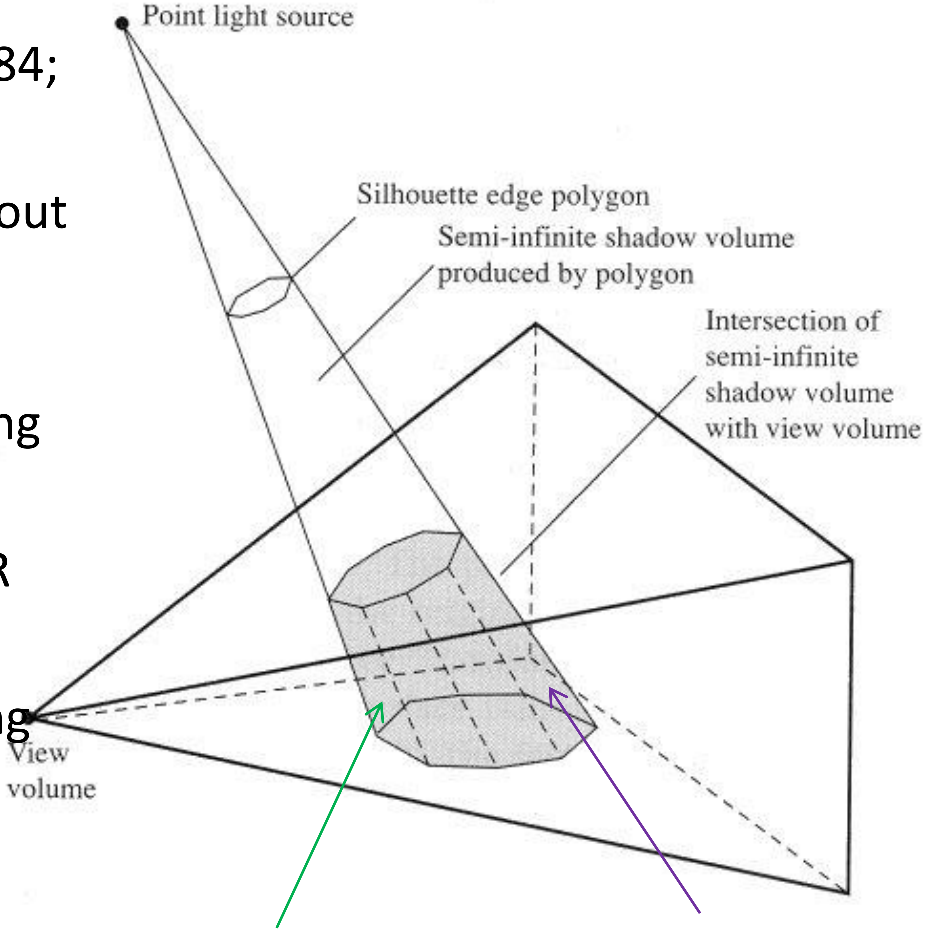
4. Shadow volumes

- Some terminology



4. Shadow volumes

- (Crow, 77; Brotman and Badler, 84; Bergeron, 86)
- Invisible volume of space swept out by shadow of object (occluder/creator)
- Finite volume obtained by clipping against any view volume
- Can be integrated into many HSR algorithms
- Divided into **front** and **back** facing (invisible) shadow polygons
  - Determined using normals

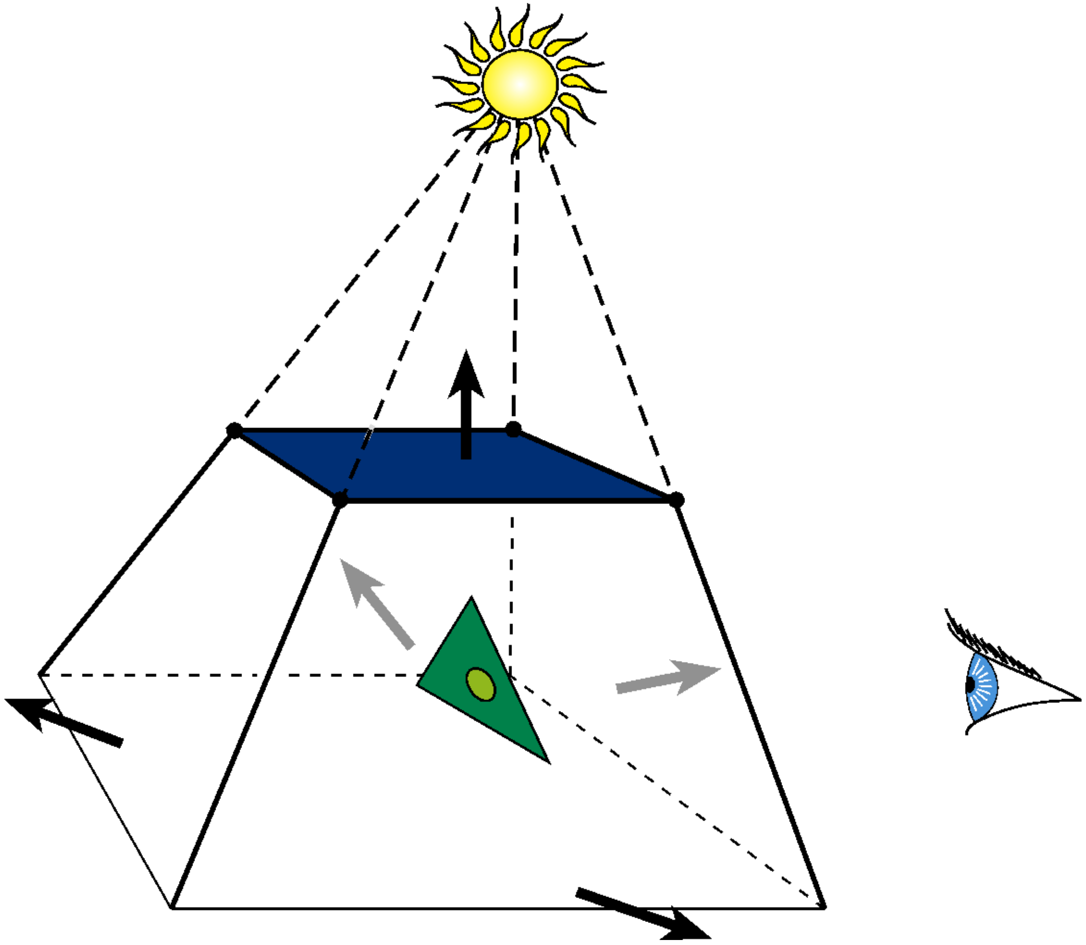

Point light source
Silhouette edge polygon
Semi-infinite shadow volume produced by polygon
Intersection of semi-infinite shadow volume with view volume
View volume

Front facing

Back facing

4. Shadow volumes

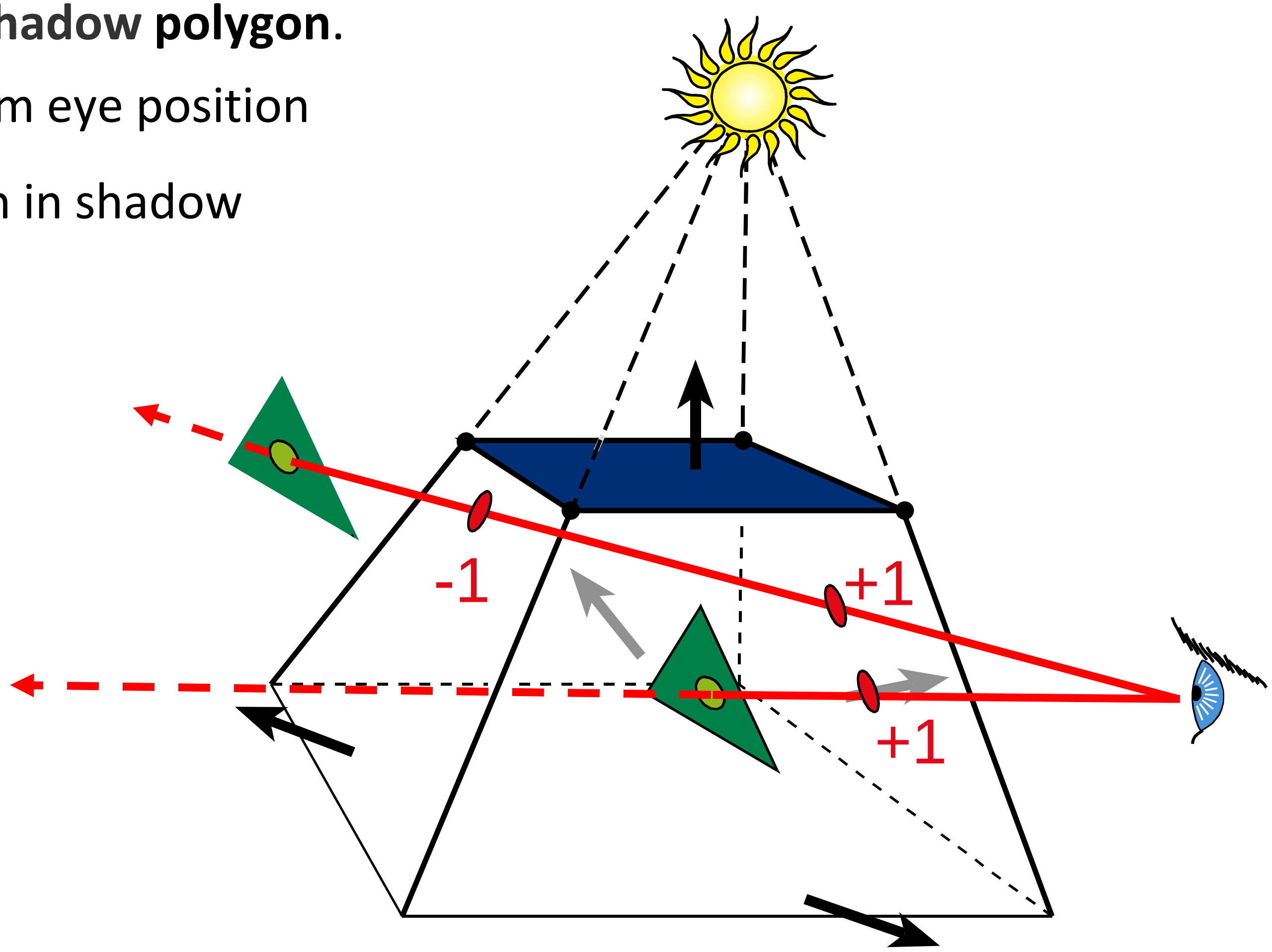
- Polygon is in shadow if behind a **front facing shadow polygon** and in front of a **back facing shadow polygon**



Durand, 08

4. Shadow volumes

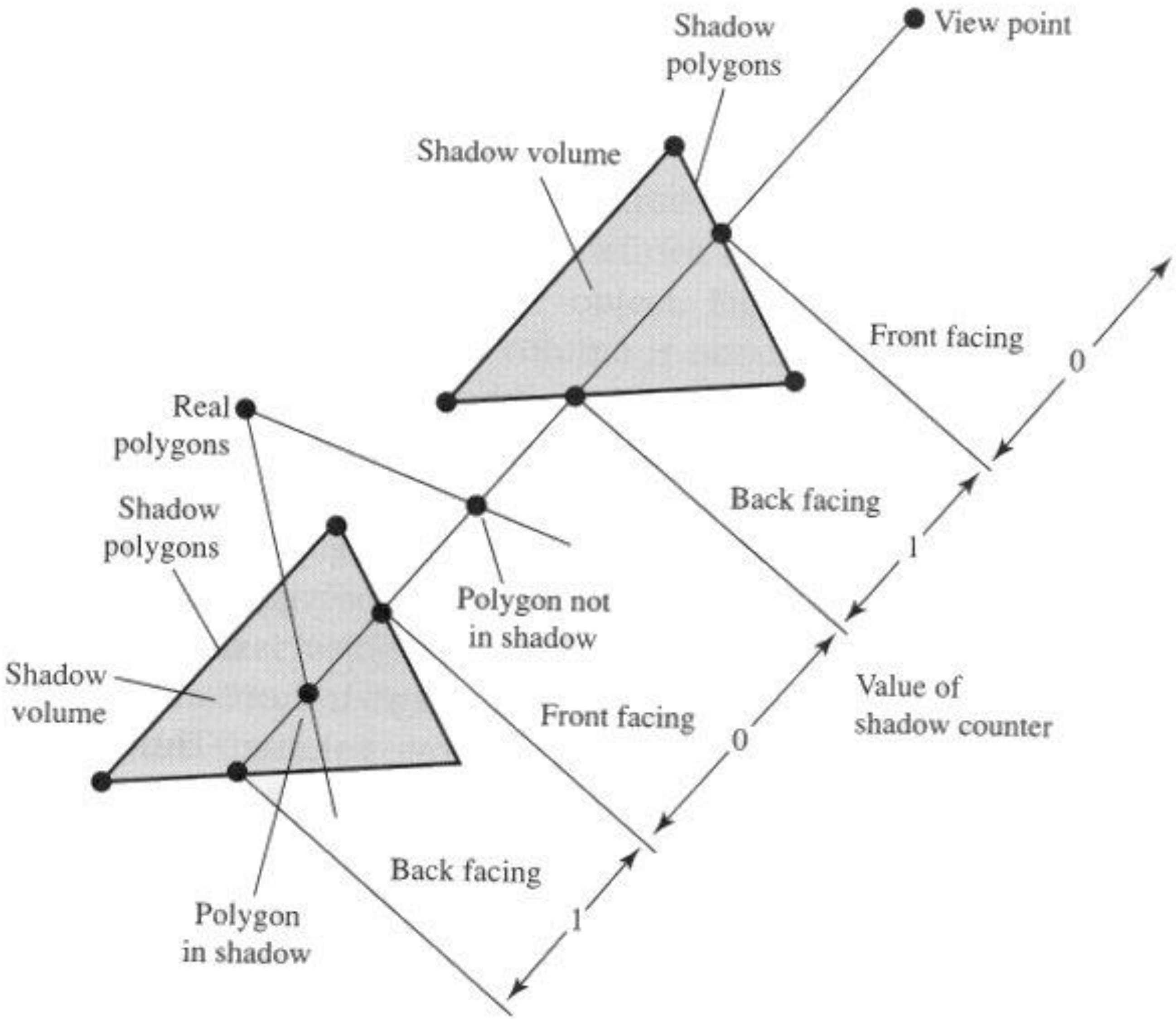
- Polygon is in shadow if behind a **front facing shadow polygon** and in front of a **back facing shadow polygon**.
- Use a counter from eye position
- If counter **≠**0 then in shadow



Durand, 08

4. Shadow volumes

- Polygon is in shadow if behind a **front facing shadow polygon** and in front of a **back facing shadow polygon**
- Use a 'counter' from eye position



4. Shadow volumes

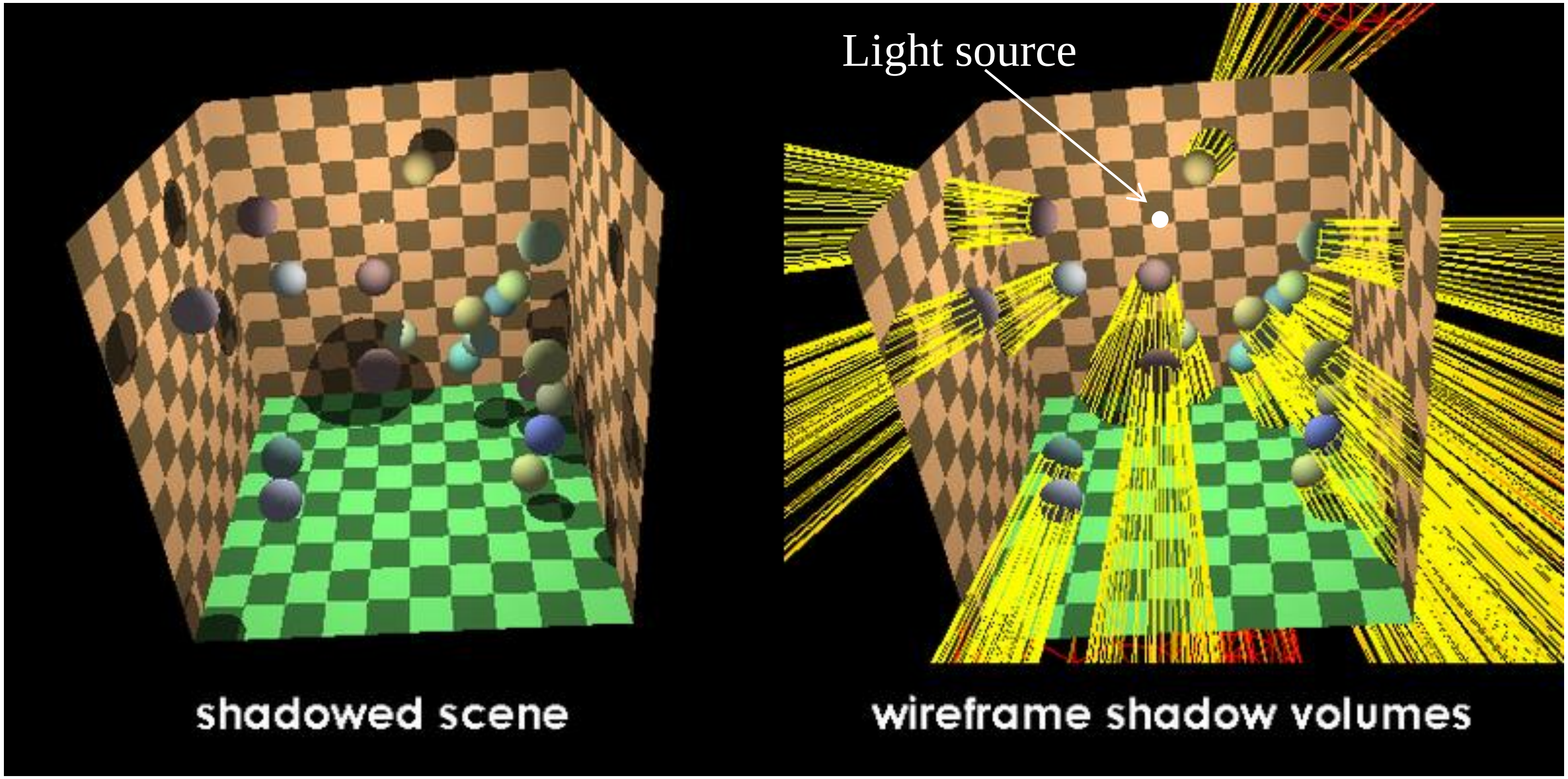
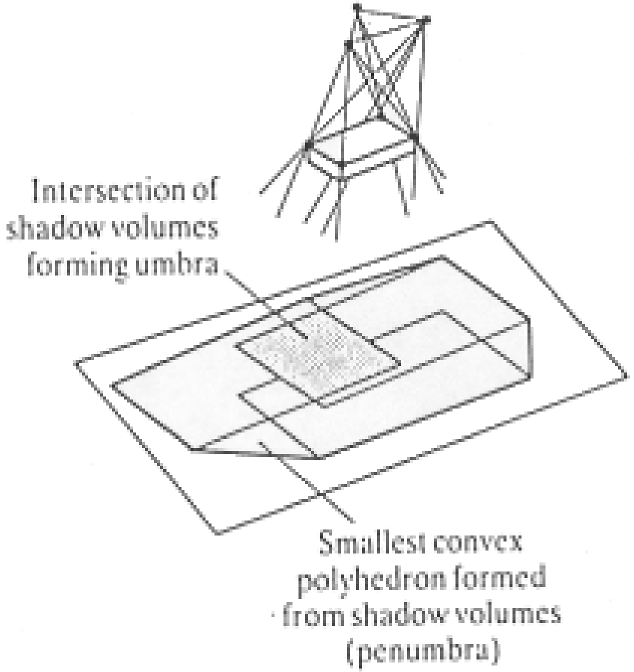


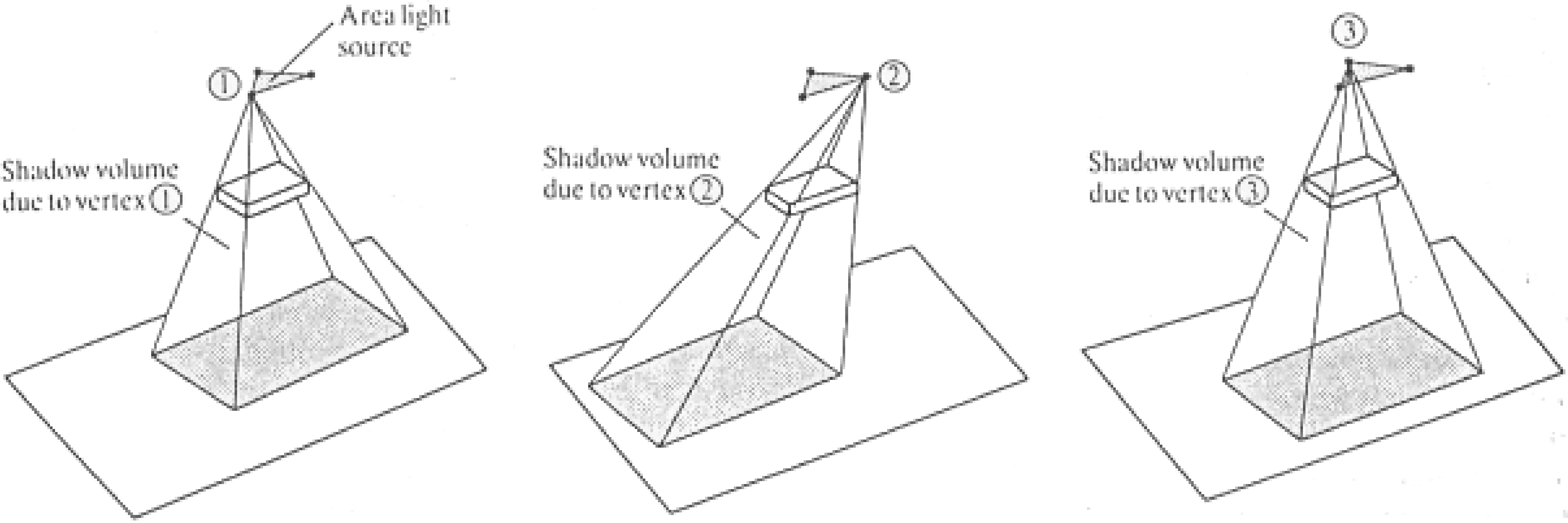
Illustration of shadow volumes. The image above at left shows a scene shadowed using shadow volumes. At right, the shadow volumes are shown in wireframe. Note how the shadows form a large conical area pointing away from the light source (the bright white point).

http://en.wikipedia.org/wiki/Shadow_volume

4.1 Area light source

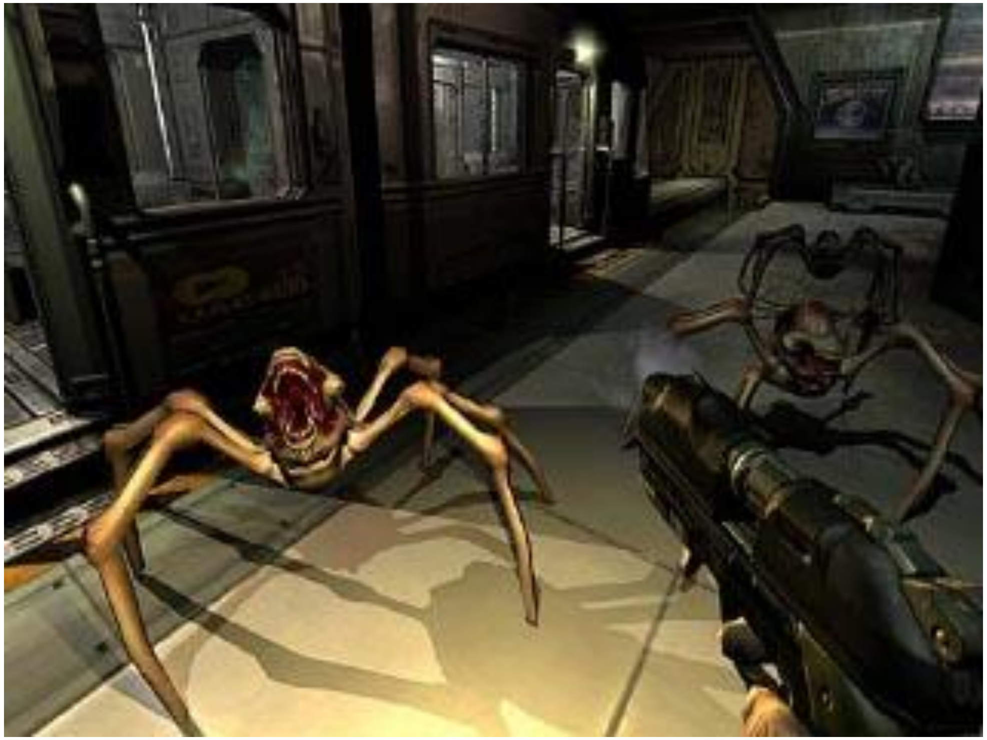
Penumbra/umbra derivations due to an area light source (from an illustration in [NISH85])

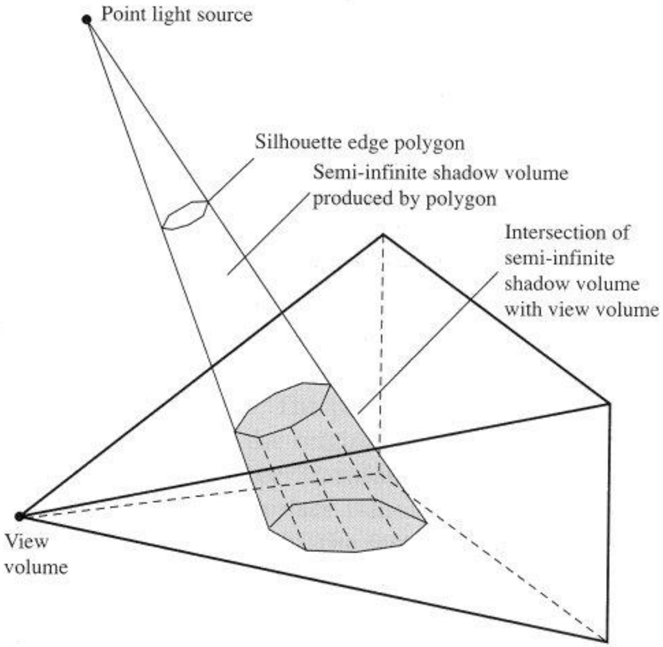

Intersection of shadow volumes forming umbra
Smallest convex polyhedron formed from shadow volumes (penumbra)


Area light source
①
Shadow volume due to vertex ①
②
Shadow volume due to vertex ②
③
Shadow volume due to vertex ③

4.2 Real-time shadow volumes

- Use stencil buffer for fast hardware implementation
  - E.g. Doom 3



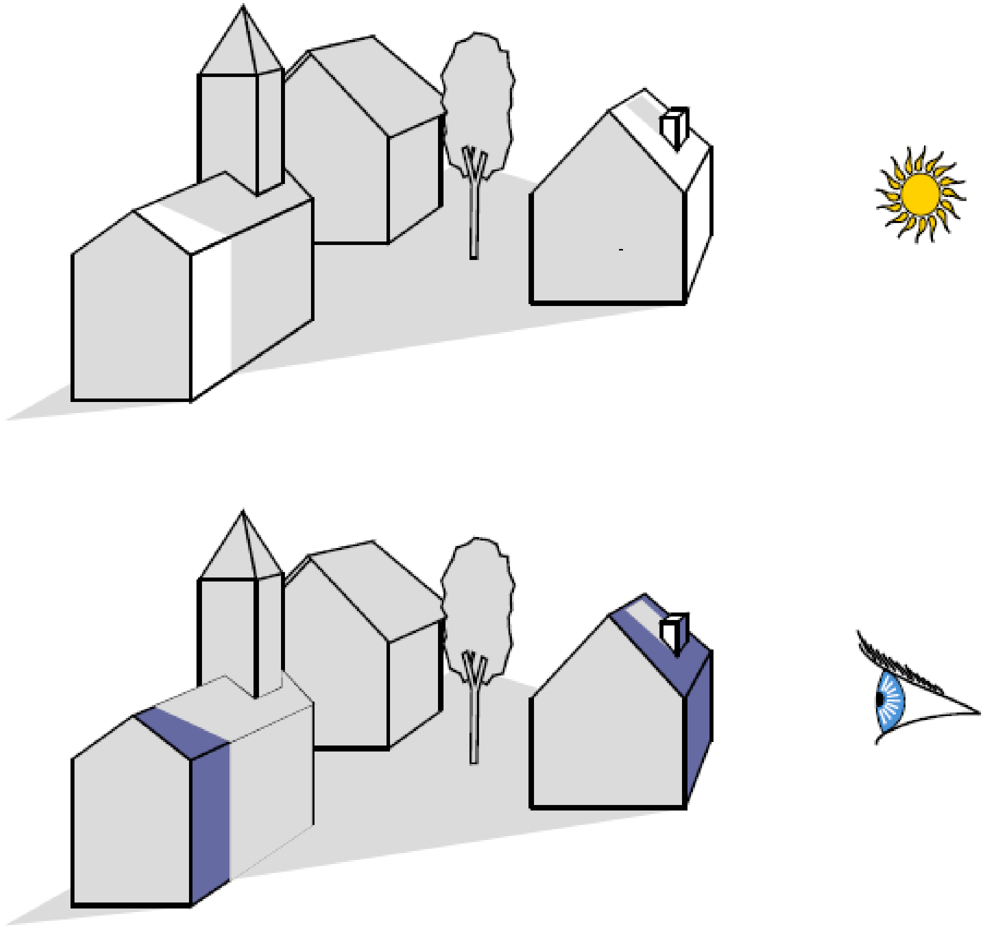

Point light source
Silhouette edge polygon
Semi-infinite shadow volume produced by polygon
Intersection of semi-infinite shadow volume with view volume
View volume

Note: objects can self-shadow

http://en.wikipedia.org/wiki/Shadow_volume; http://www.idsoftware.com

5. Shadow Z-buffer / Shadow mapping (Williams, 78; Reeves et al, 87)

- Reminder: shadow/view duality
  - A point is lit if it is visible from the light source
  - A point is in shadow if it is not visible from the light source
  - Points rendered in blue are visible from the viewpoint
- Shadow computation is similar to view computation



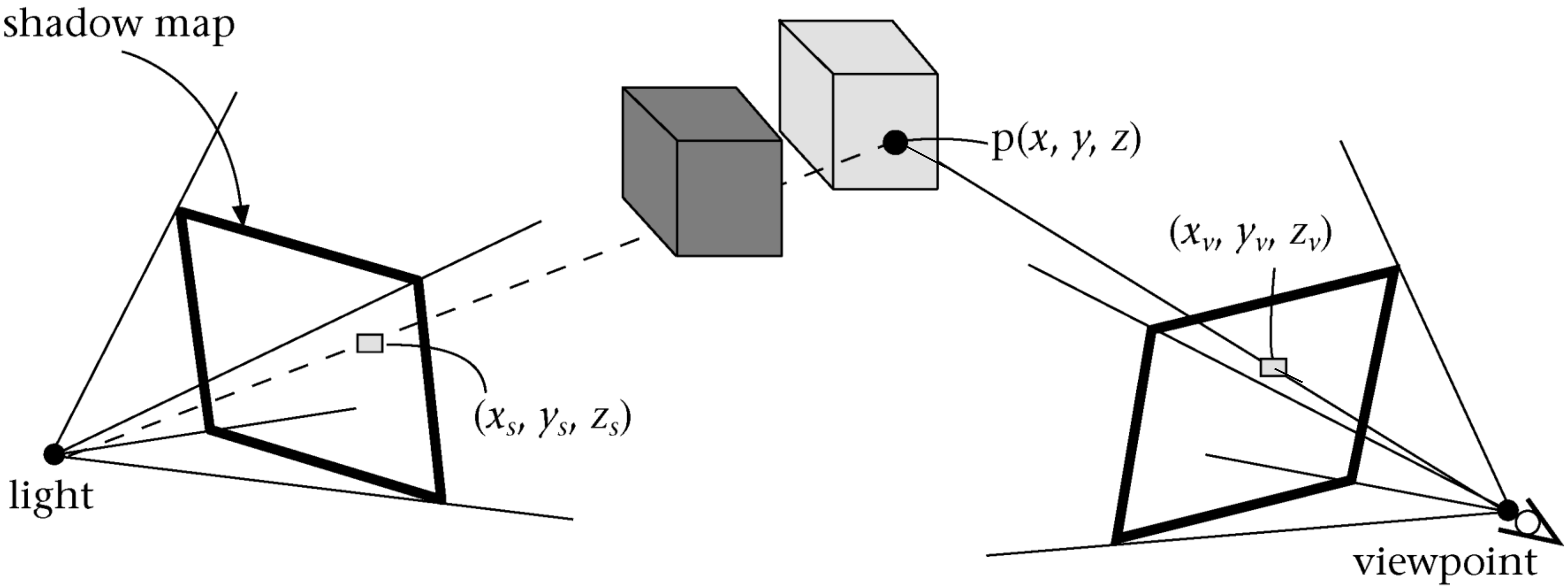
Durand, 08

5. Shadow Z-buffer / Shadow mapping (Williams, 78; Reeves et al, 87)

Two-stage algorithm , with a rendering pass in each stage

Stage 1. 'Render' scene from light's point of view (no lighting calculations)

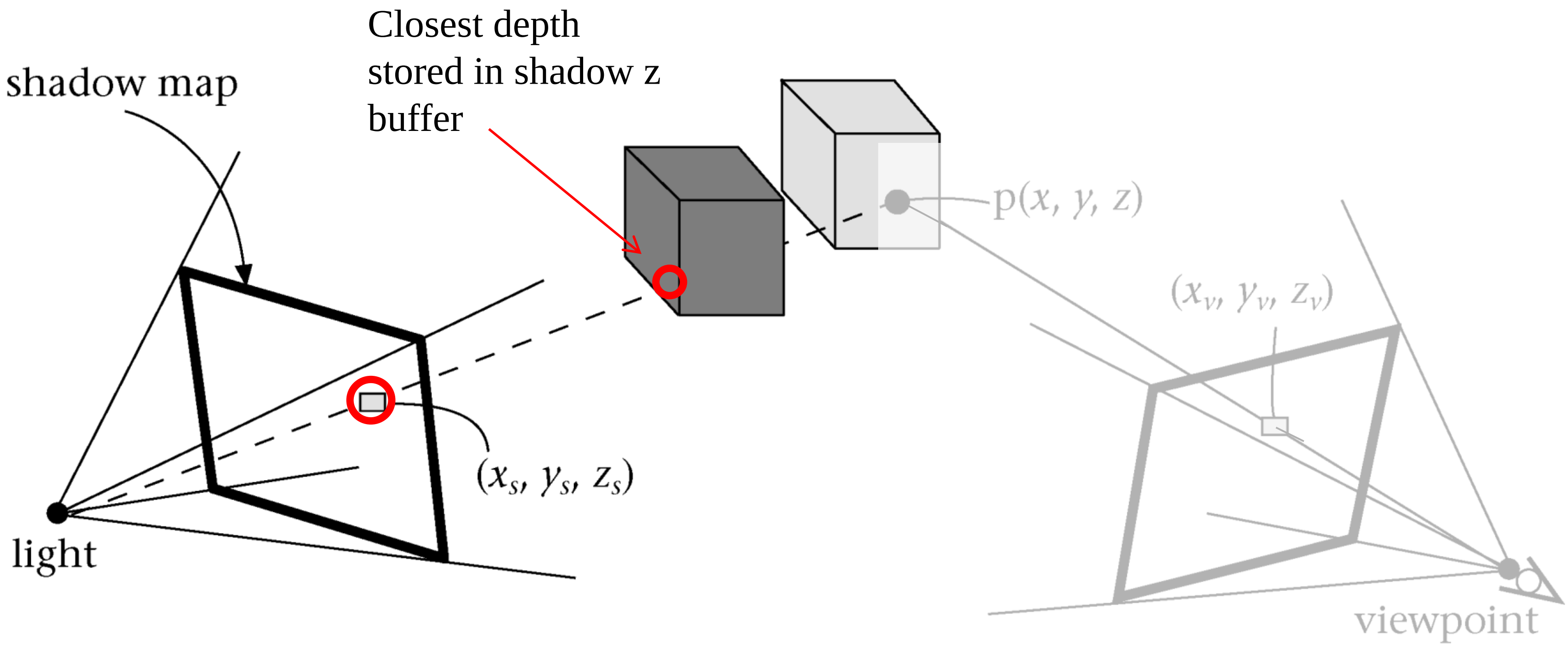
Stage 2. Render scene from eye viewpoint (including lighting calculations)



Slater et al , 02

5.1 Stage 1: ‘Render’ scene from light’s point of view

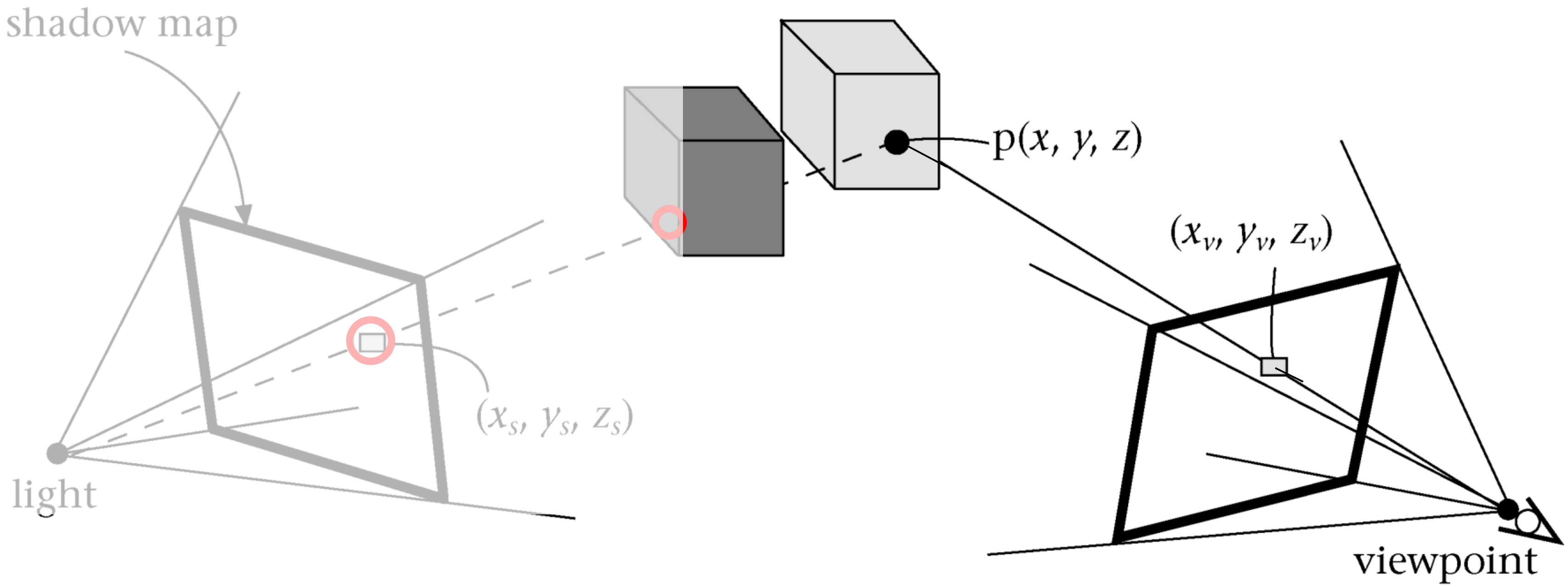
- No lighting calculations
- Store depth of each pixel in shadow map (also called depth map because is is a z buffer)



Slater et al , 02

5.2 Stage 2: Render scene from eye viewpoint

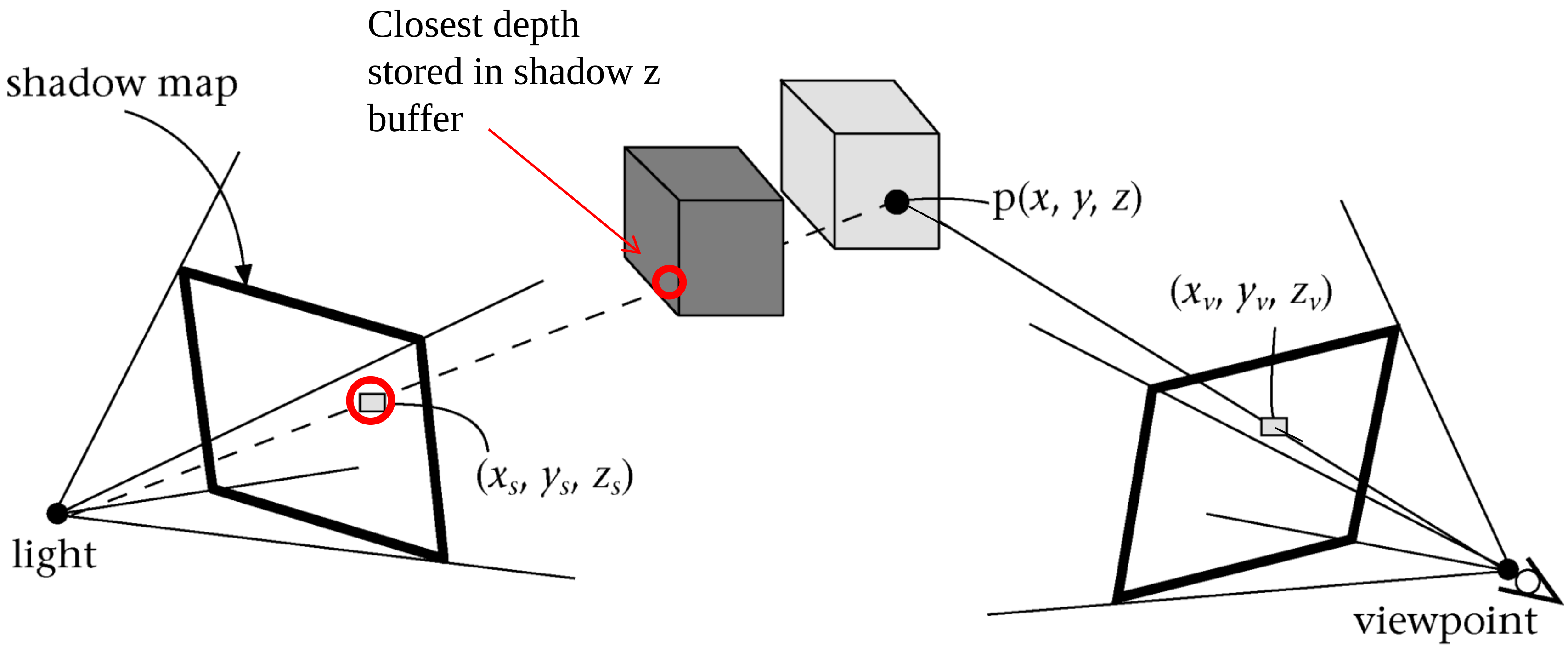
- Include lighting calculations, i.e. standard zbuffer-based HSR
- Check shadow map to determine shadows



Slater et al , 02

5.2 Stage 2: Render scene from eye viewpoint

- Include lighting calculations, i.e. standard zbuffer-based HSR
- **Check shadow map to determine shadows - How?**

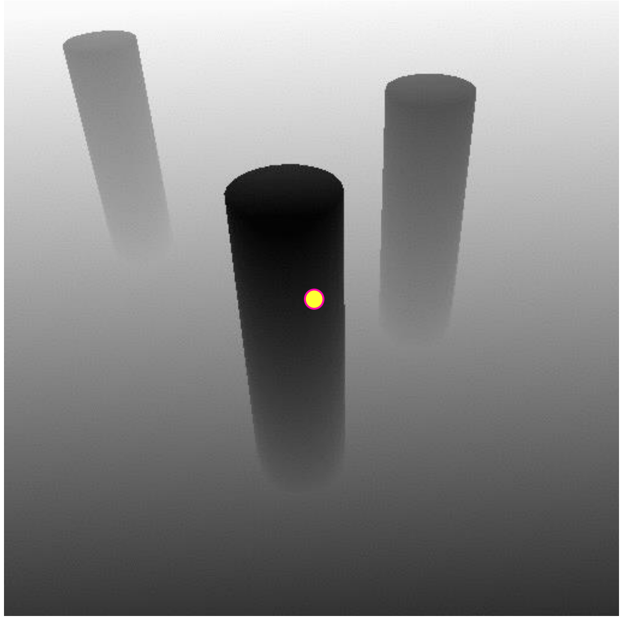


Slater et al , 02

5.2 Stage 2: Render scene from eye viewpoint

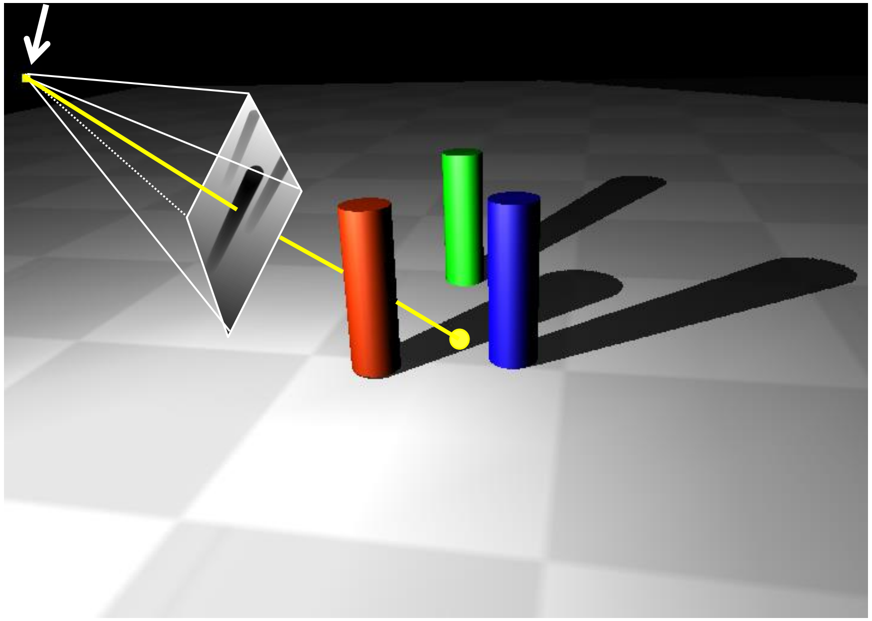
- Transform a surface point ● from eye (x_e,y_e,z_e) into light coordinate system (x_l,y_l,z_l)
- Compare resulting depth z’ to stored depth in shadow map
- If z’ > stored_depth(x_l,y_l), then pixel is in shadow

Shadow map



Light source

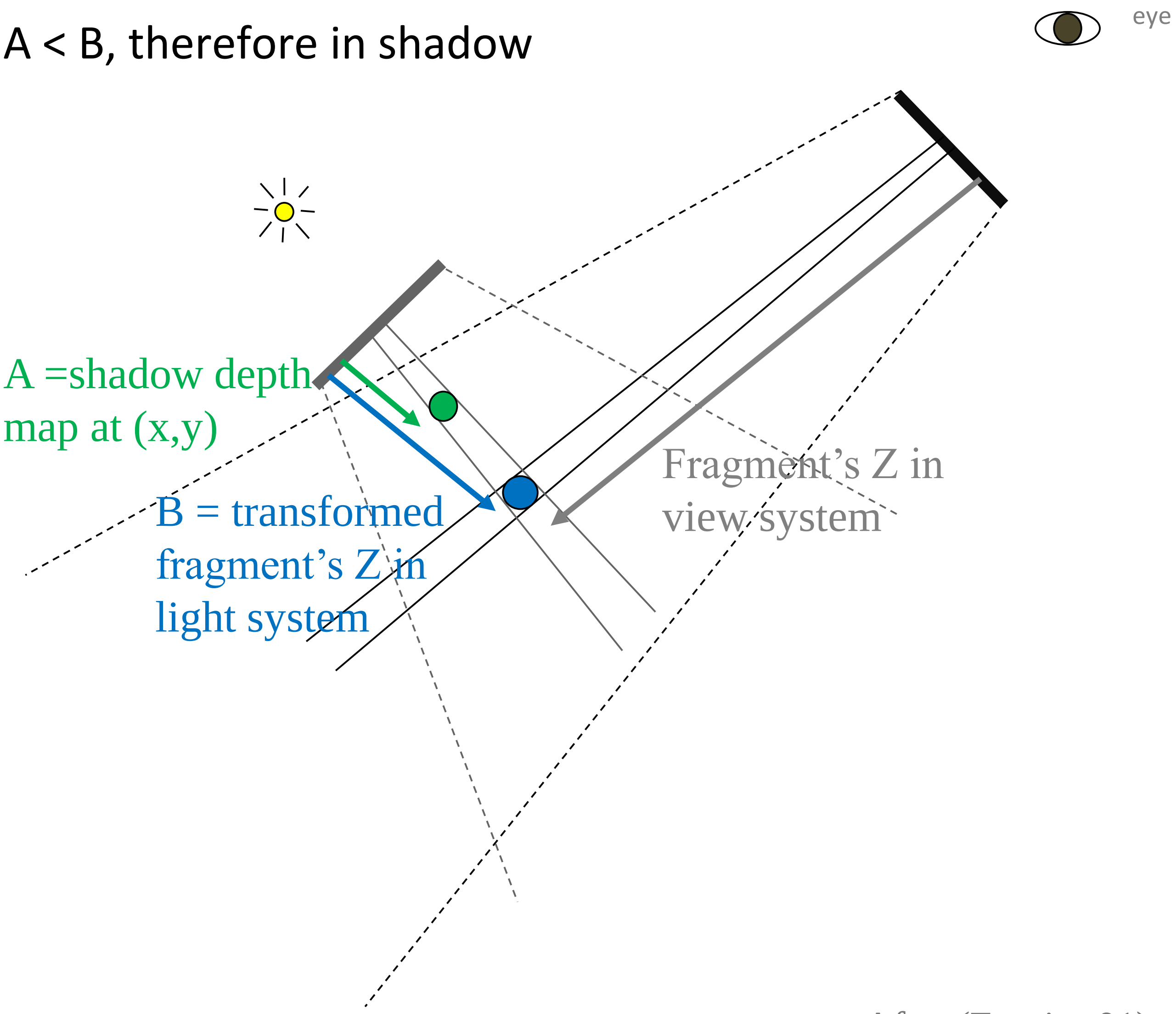
Eye viewpoint



Brabec,04

5.3 Comparing the depths

- The case A < B, therefore in shadow



eye

A =shadow depth map at (x,y)

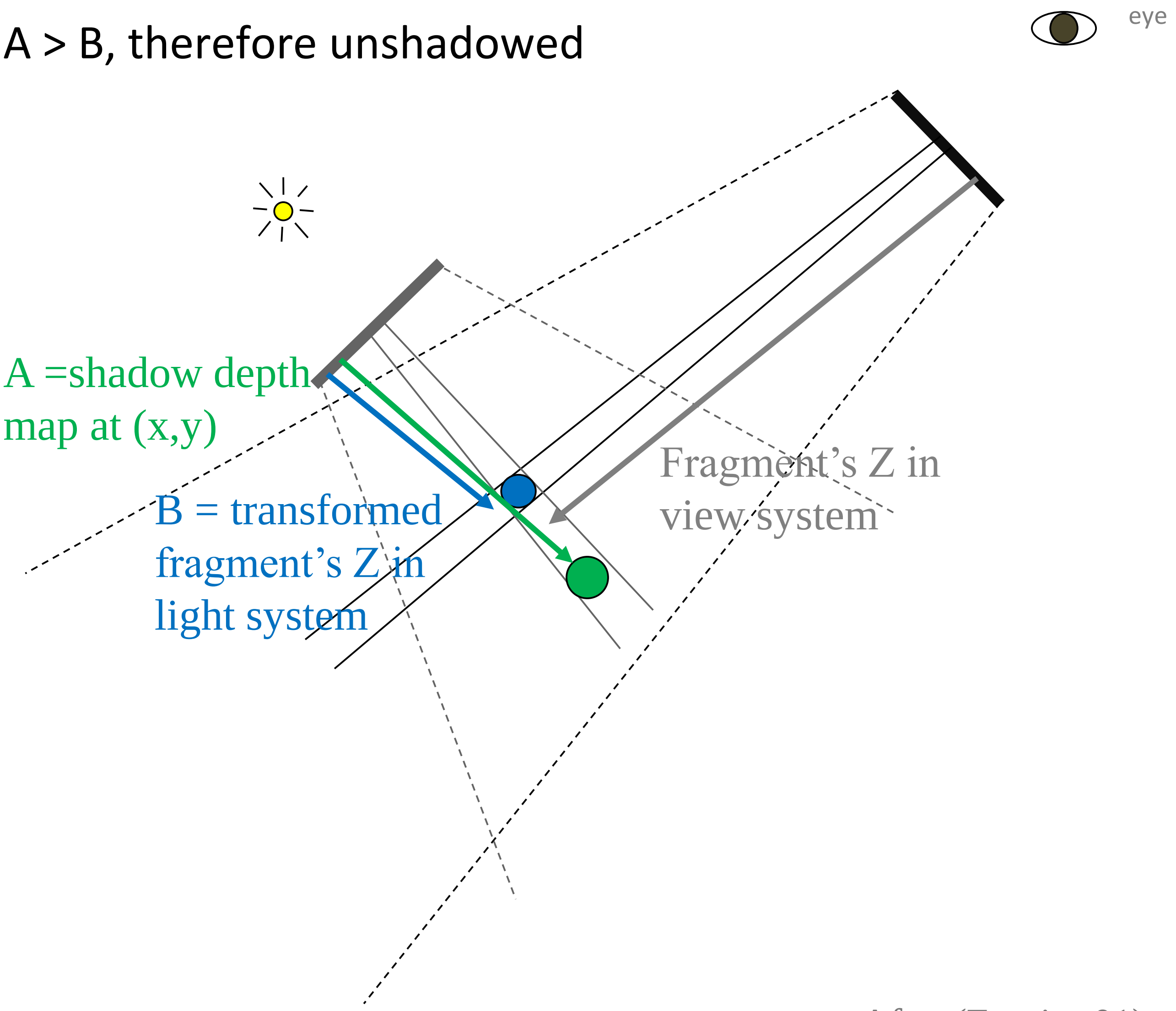
B = transformed fragment's Z in light system

Fragment's Z in view system

After (Everitt, 01)

5.3 Comparing the depths

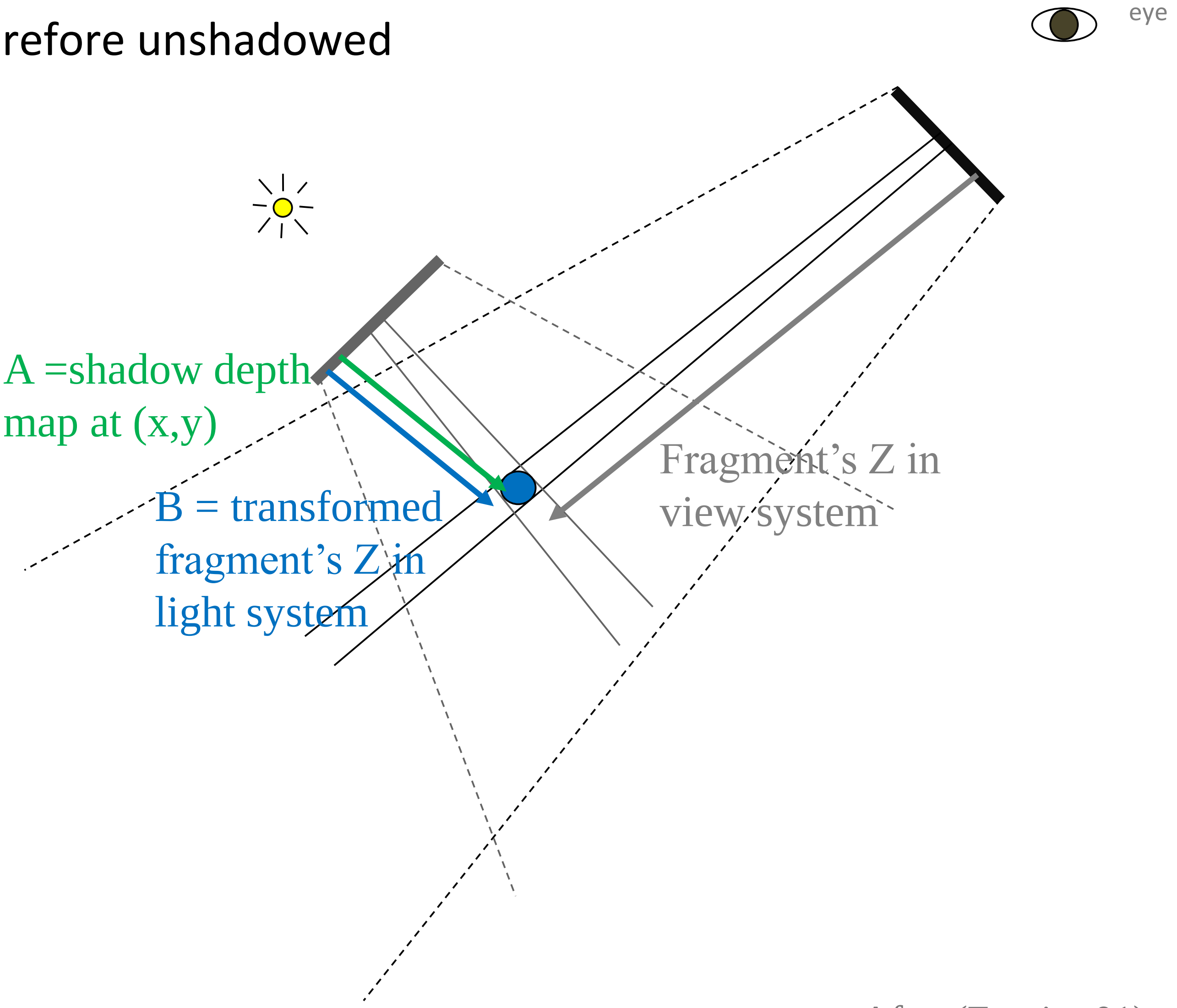
- The case A > B, therefore unshadowed


eye
A =shadow depth map at (x,y)
B = transformed fragment's Z in light system
Fragment's Z in view system

After (Everitt, 01)

5.3 Comparing the depths

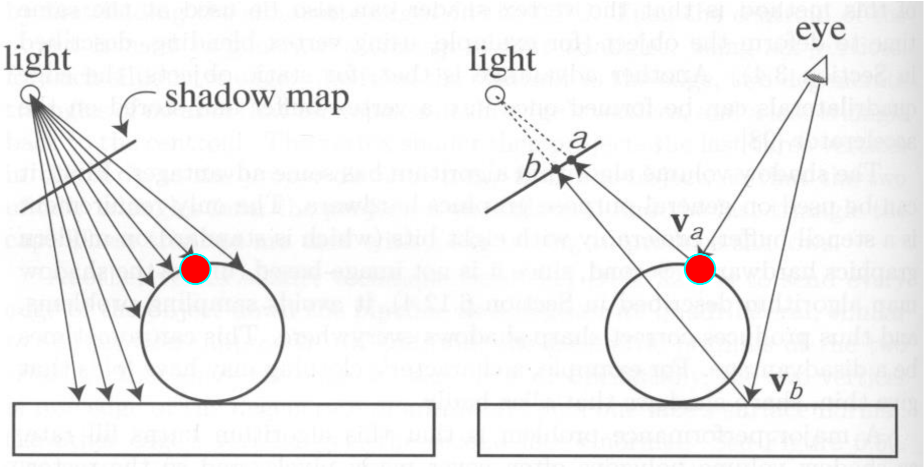
- A≅ B, therefore unshadowed

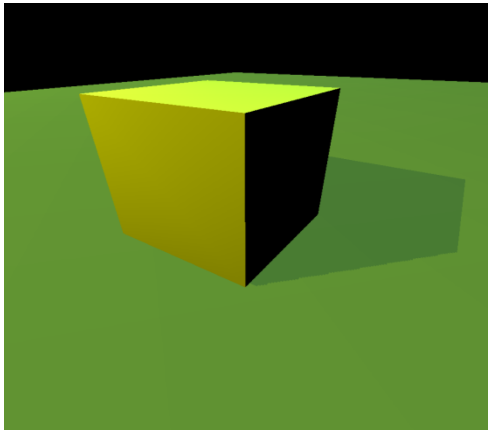


After (Everitt, 01)

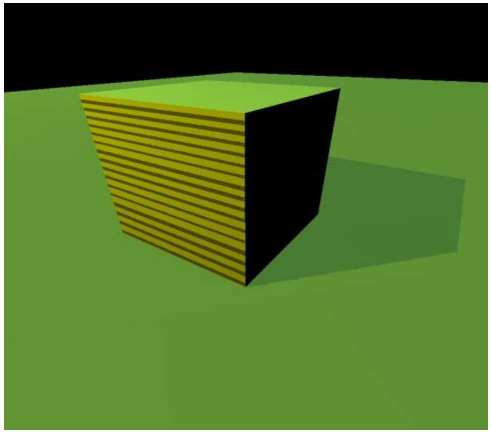
5.4 Bias

- Shadowmap (x,y) ≅ z’
- Erroneous self-shadowing
- Add a little:
  - Shadowmap(x,y)+bias < z’
- Choosing a good bias value can be very tricky

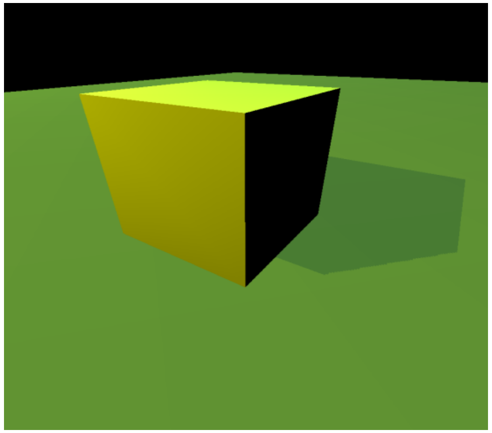




Correct image



Not enough bias

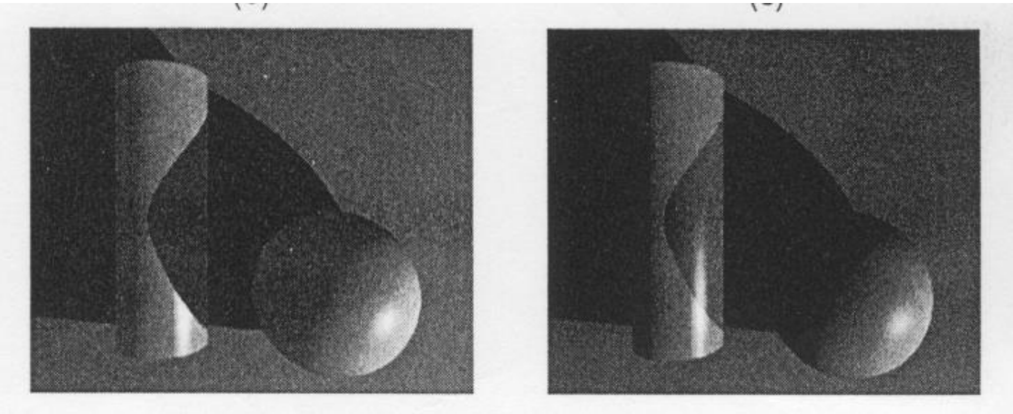


Way too much bias

Durand, 08

5.5 Multiple lights

- Separate shadow Z (depth)-buffer for each light source.
- Incorrectly shades highlights – they are just darkened



Foley et al, 90

- Objects can self-shadow

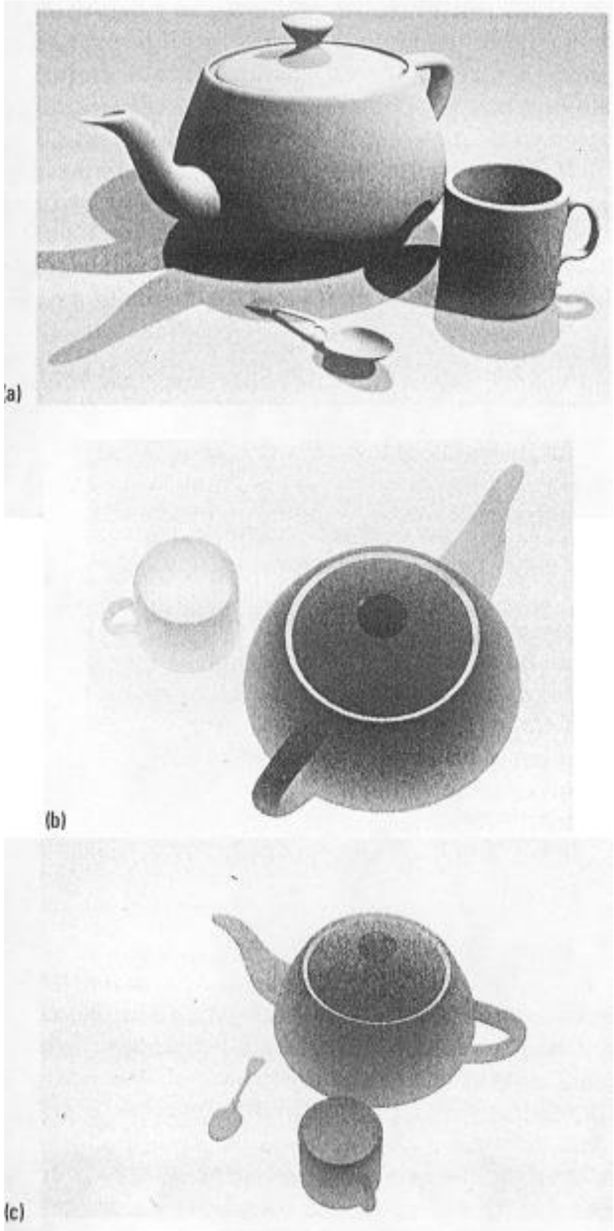
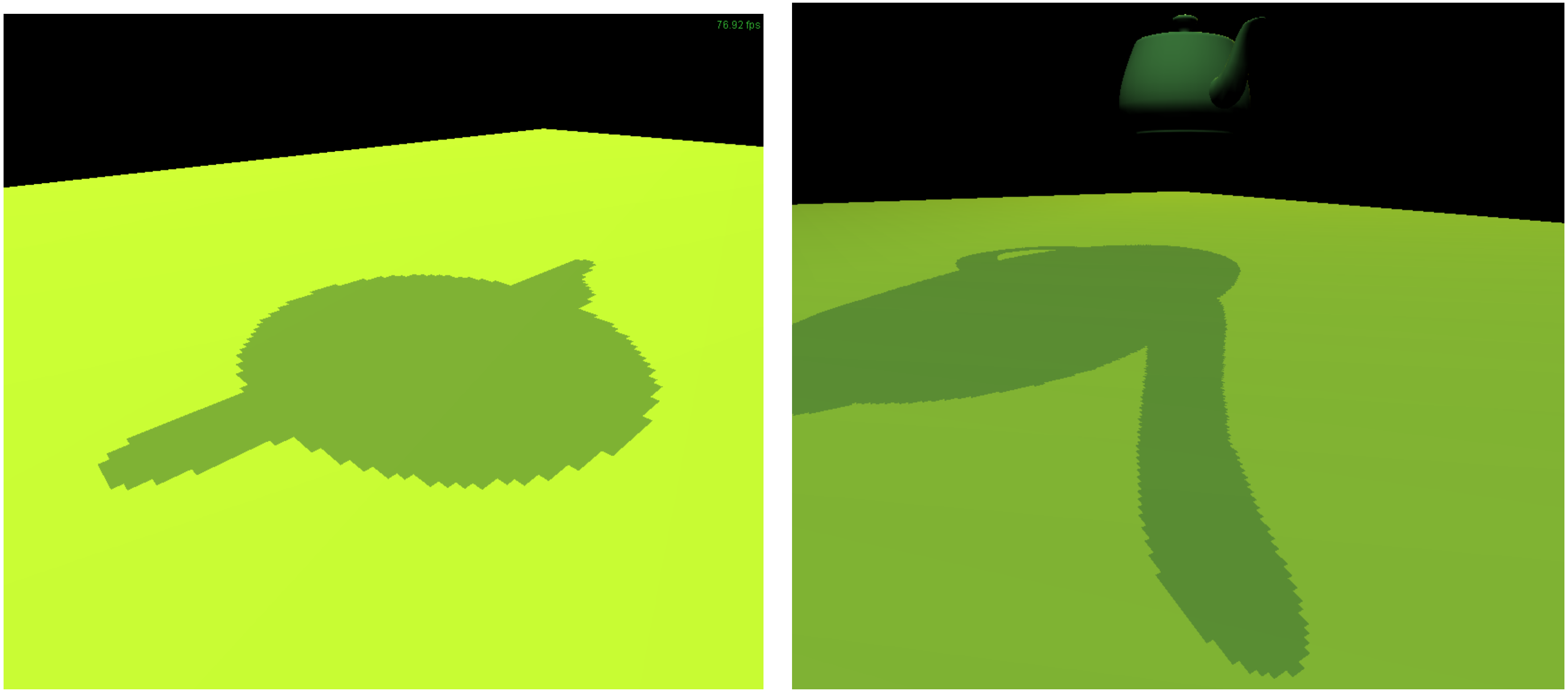


Figure 5.12 The shadow Z-buffer technique. (a) A scene ri
with two light sources. (b) Depth map for the first light sour
(c) Depth map for the second light source.

5.6 Aliasing

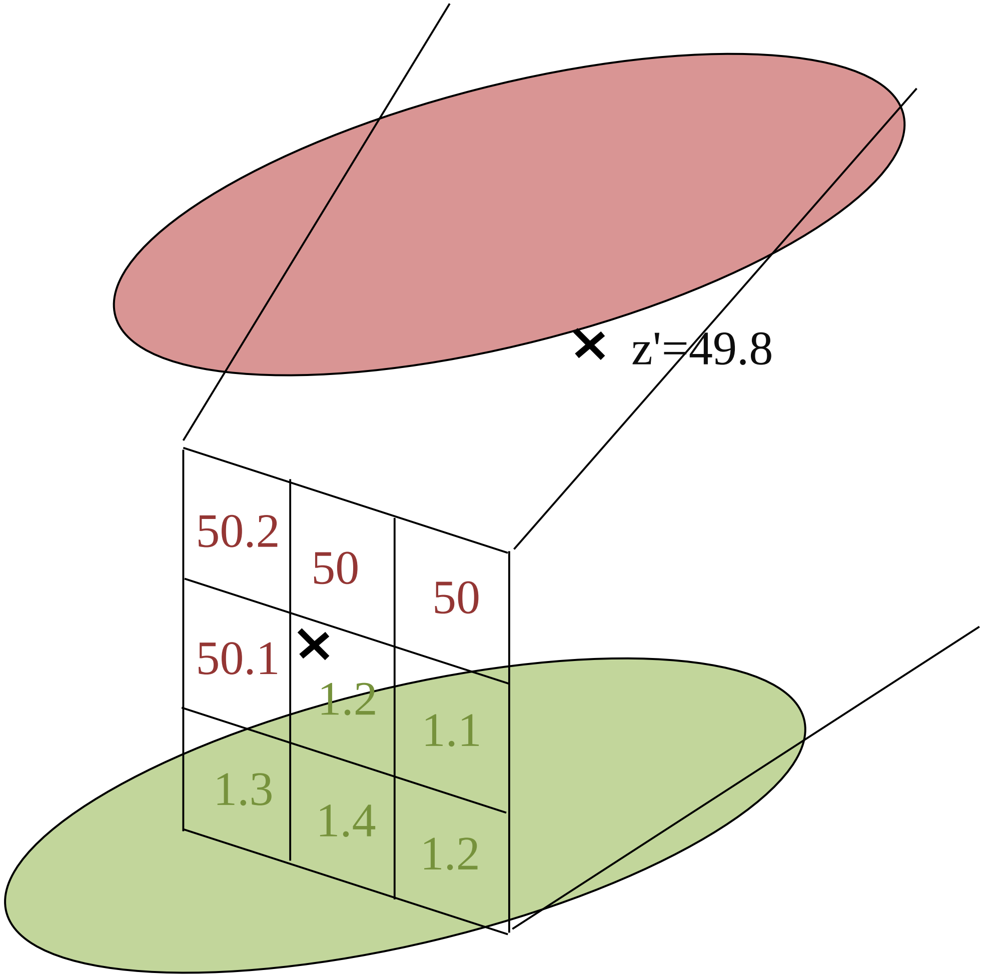
- Sampling issues
- Reprojection aliasing – especially bad when the camera & light are opposite each other



Durand, 08

5.7 Filtering

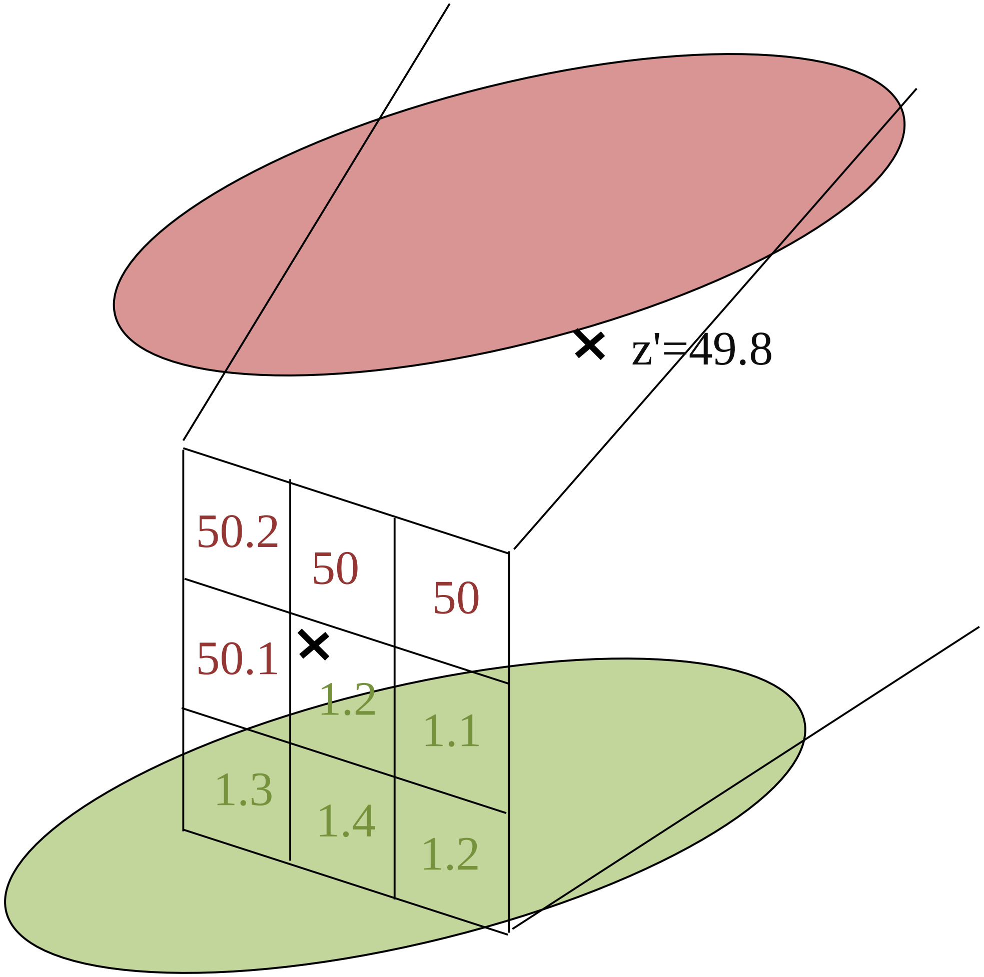
- Look at neighbours
- Filter the depth?
  - weighted average of neighbouring depth values
  - = 22.9
  - Compare 22.9 < 49.8, therefore in shadow
- No... filtering depth is not meaningful



After (Durand, 08)

5.8 Percentage closer filtering

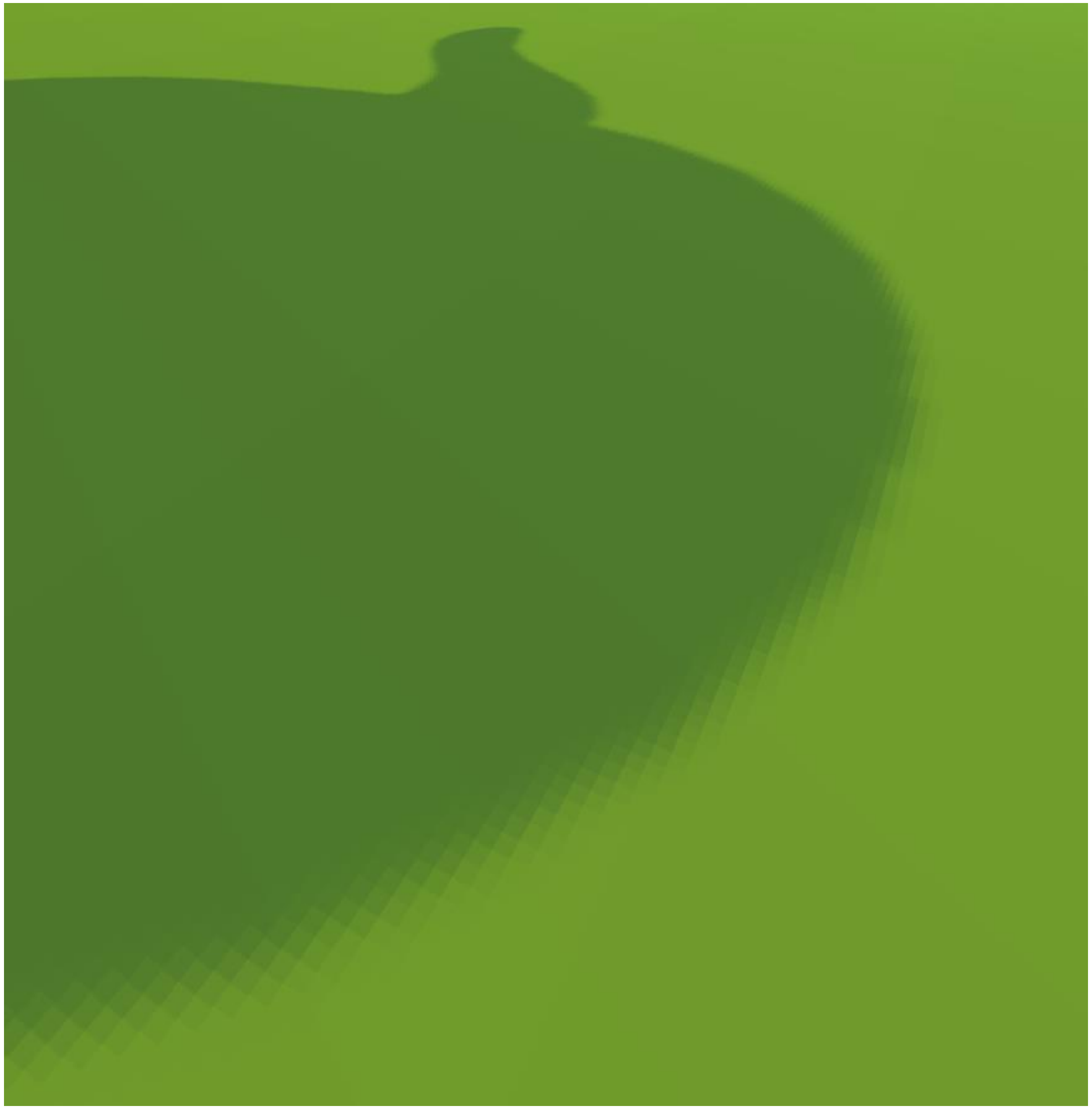
- Instead filter the result of the depth test for each neighbour
  - weighted average of comparison results
  - Compare $a_{ij} < 49.8$
  - 0 0 0
    0 1 1
    1 1 1
  - Average = 0.56
  - So, shadow value is 56%
- But makes the bias issue trickier to get right



After (Durand, 08)

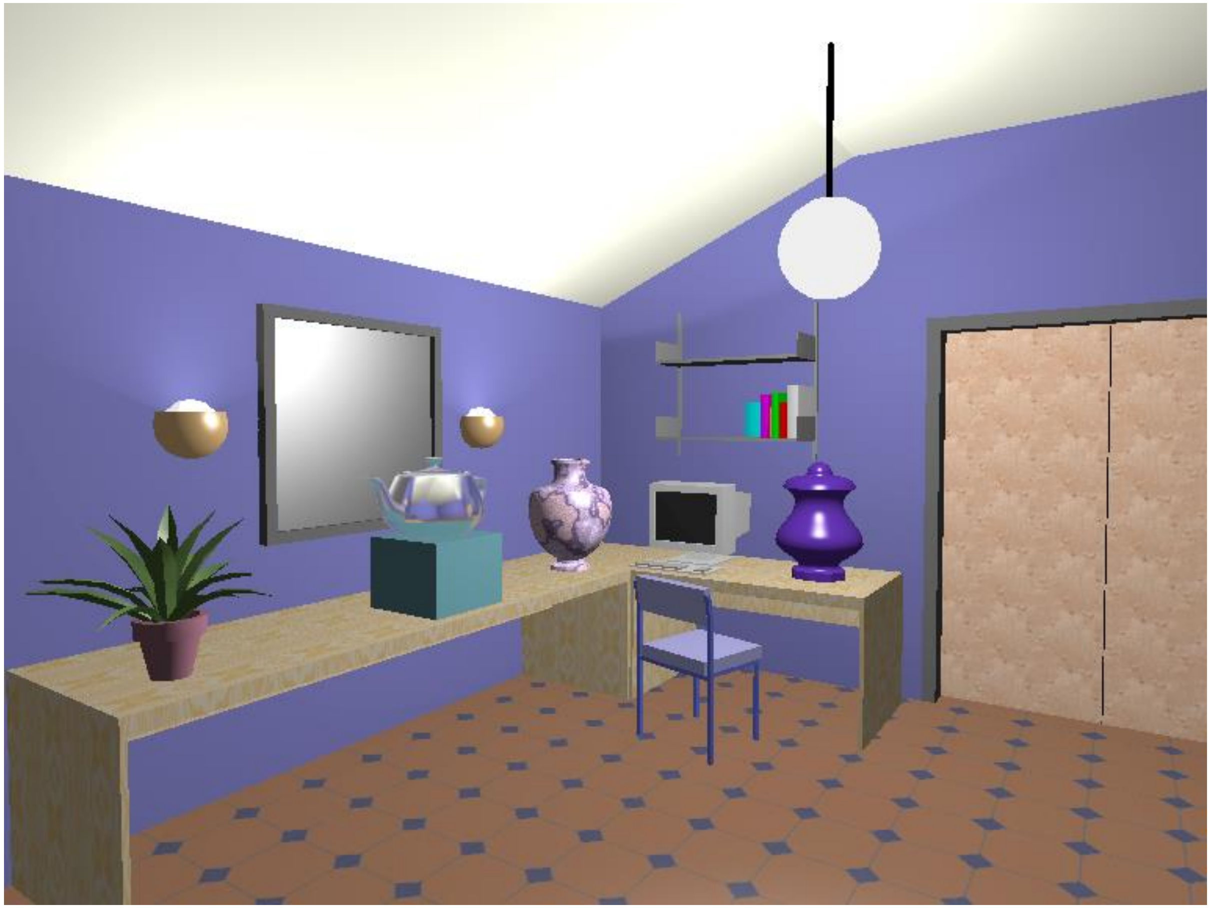
5.8 Percentage closer filtering

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



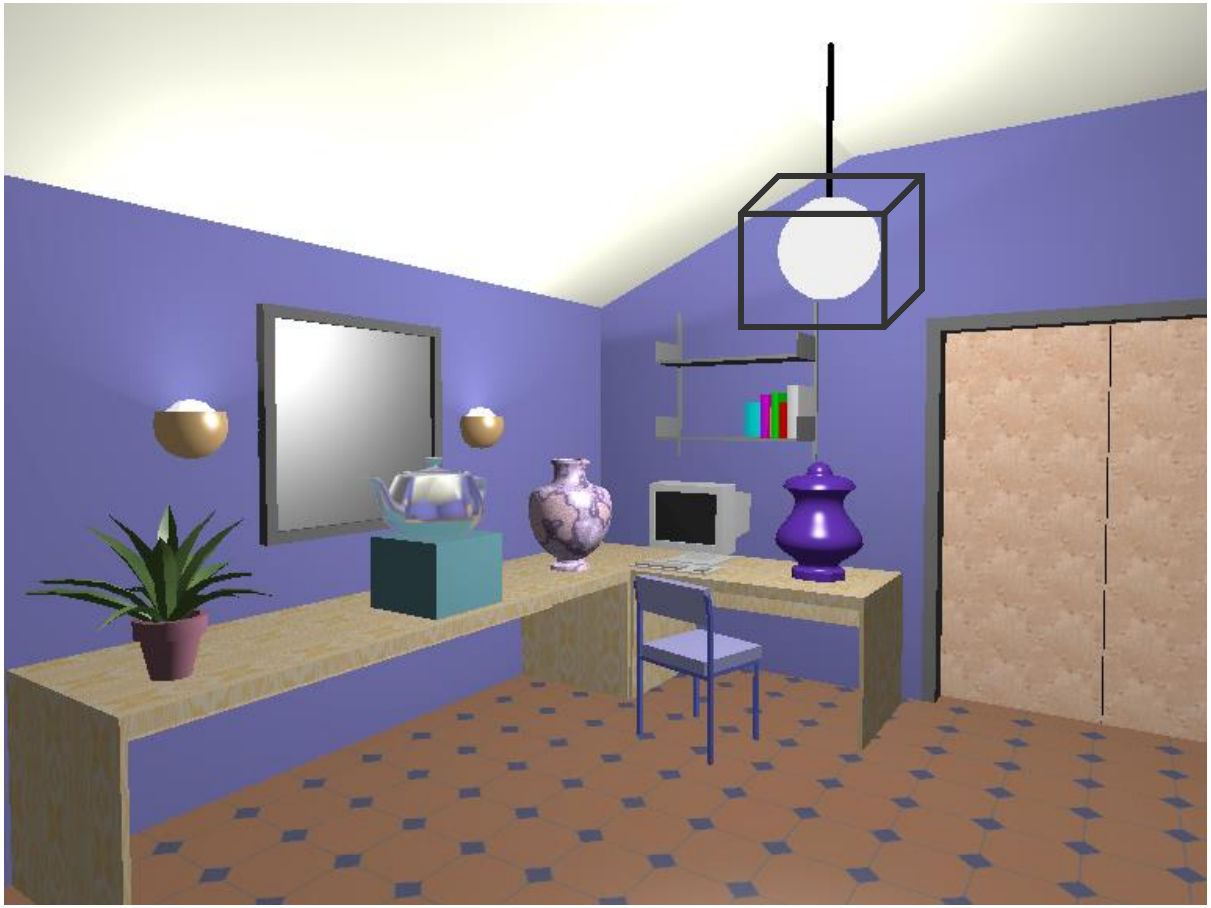
Durand, 08

6. An example

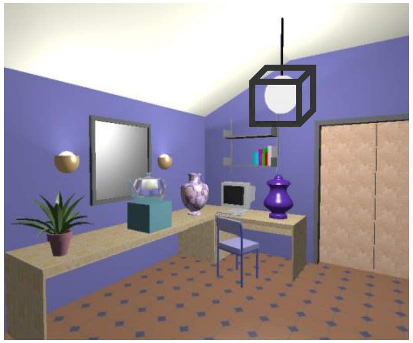


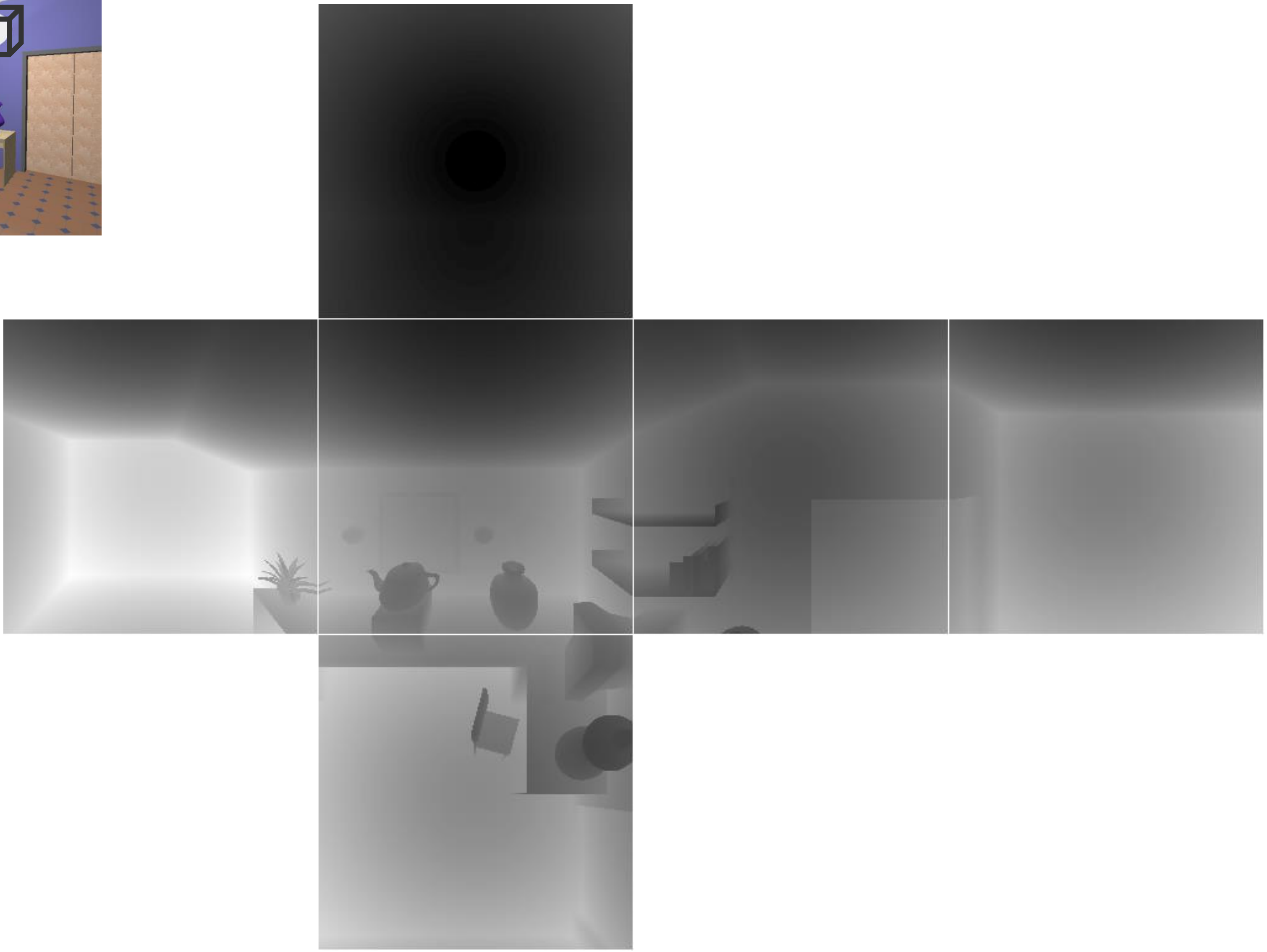
No shadows

6. An example

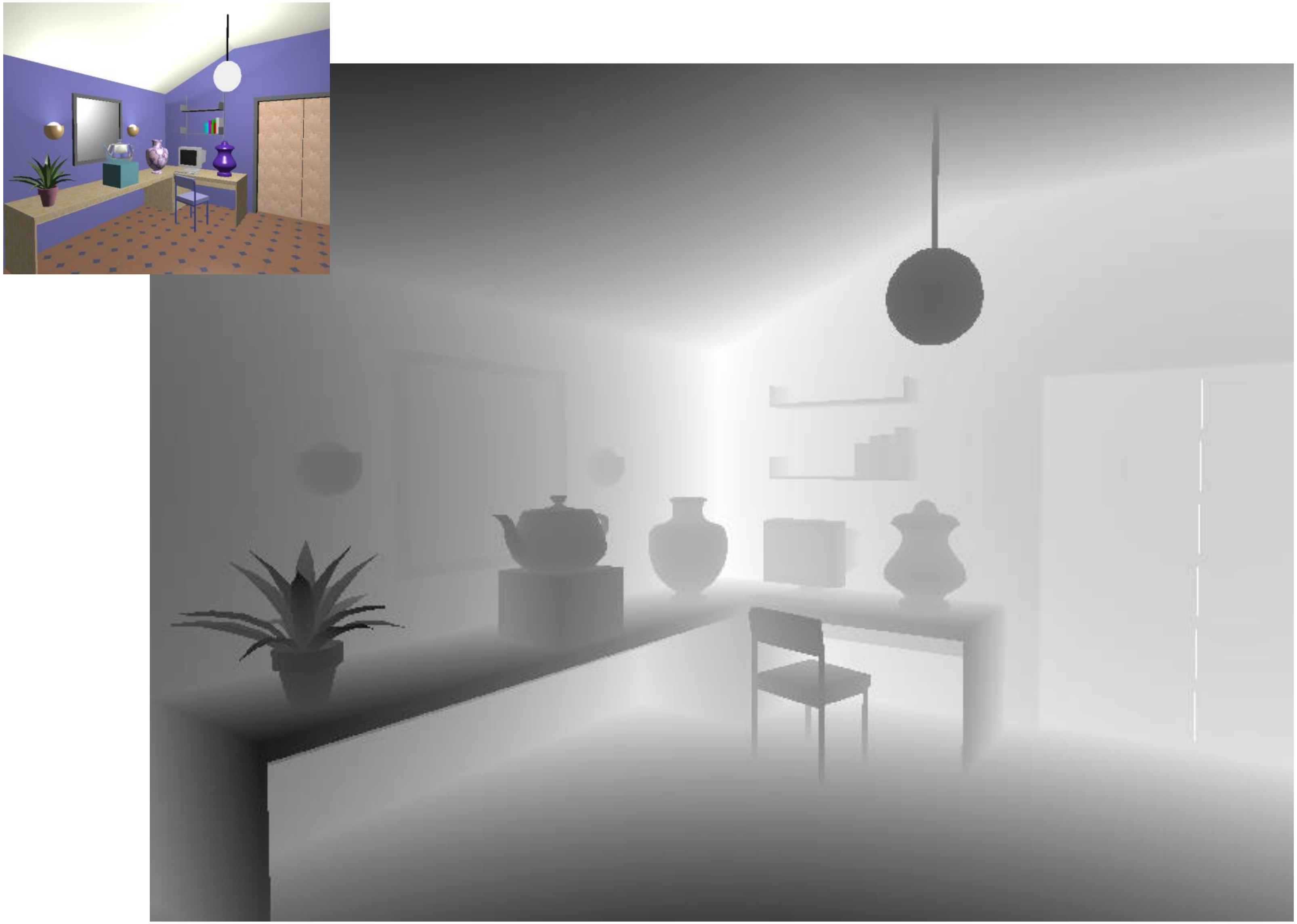


‘render’ scene from light source – fit cube and render through each face

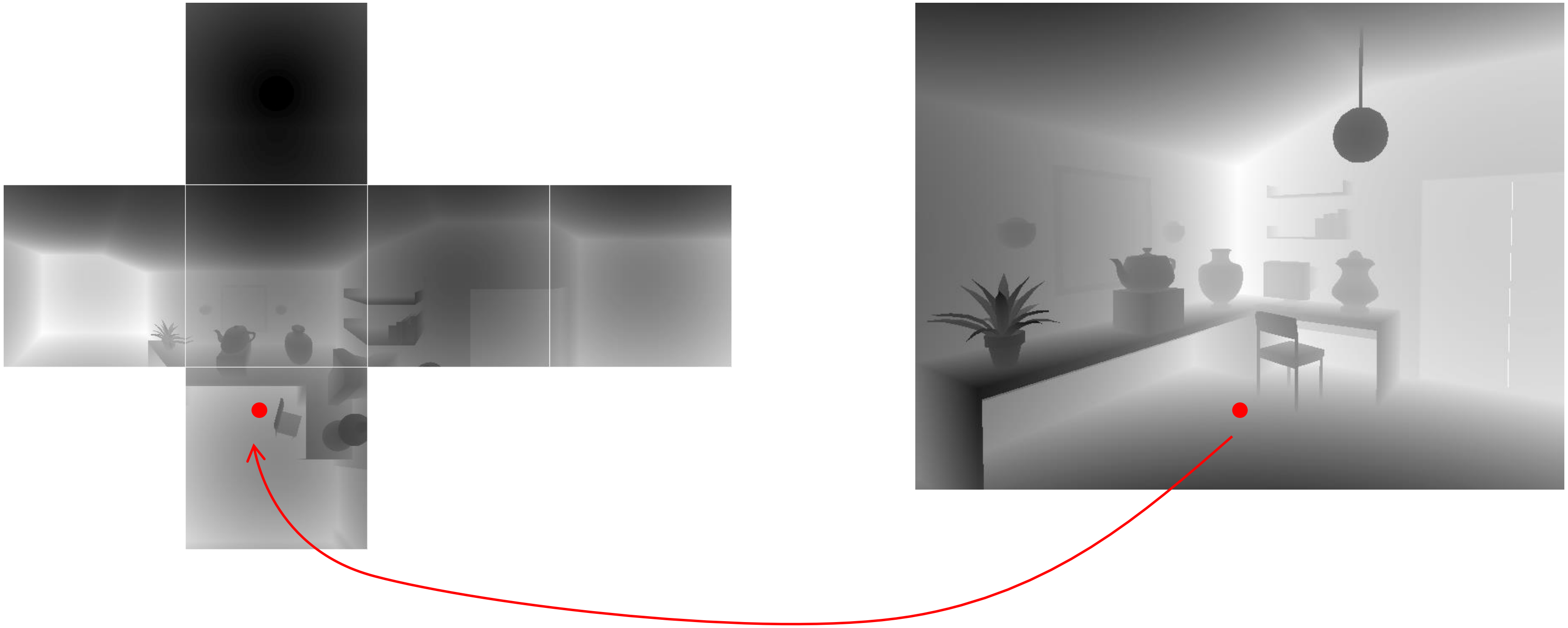




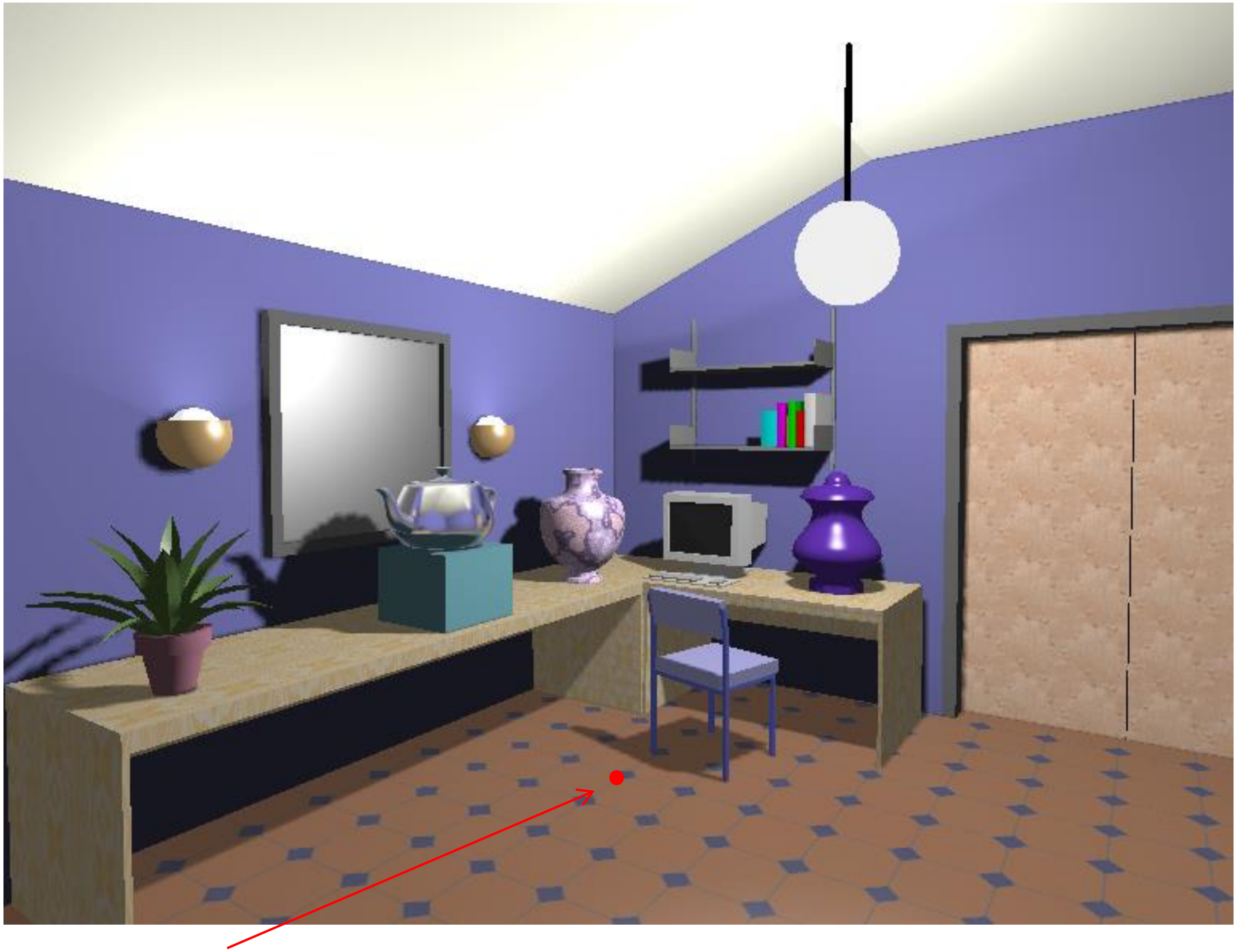
Depth map for the point light source



Depth map for scene from proper viewpoint

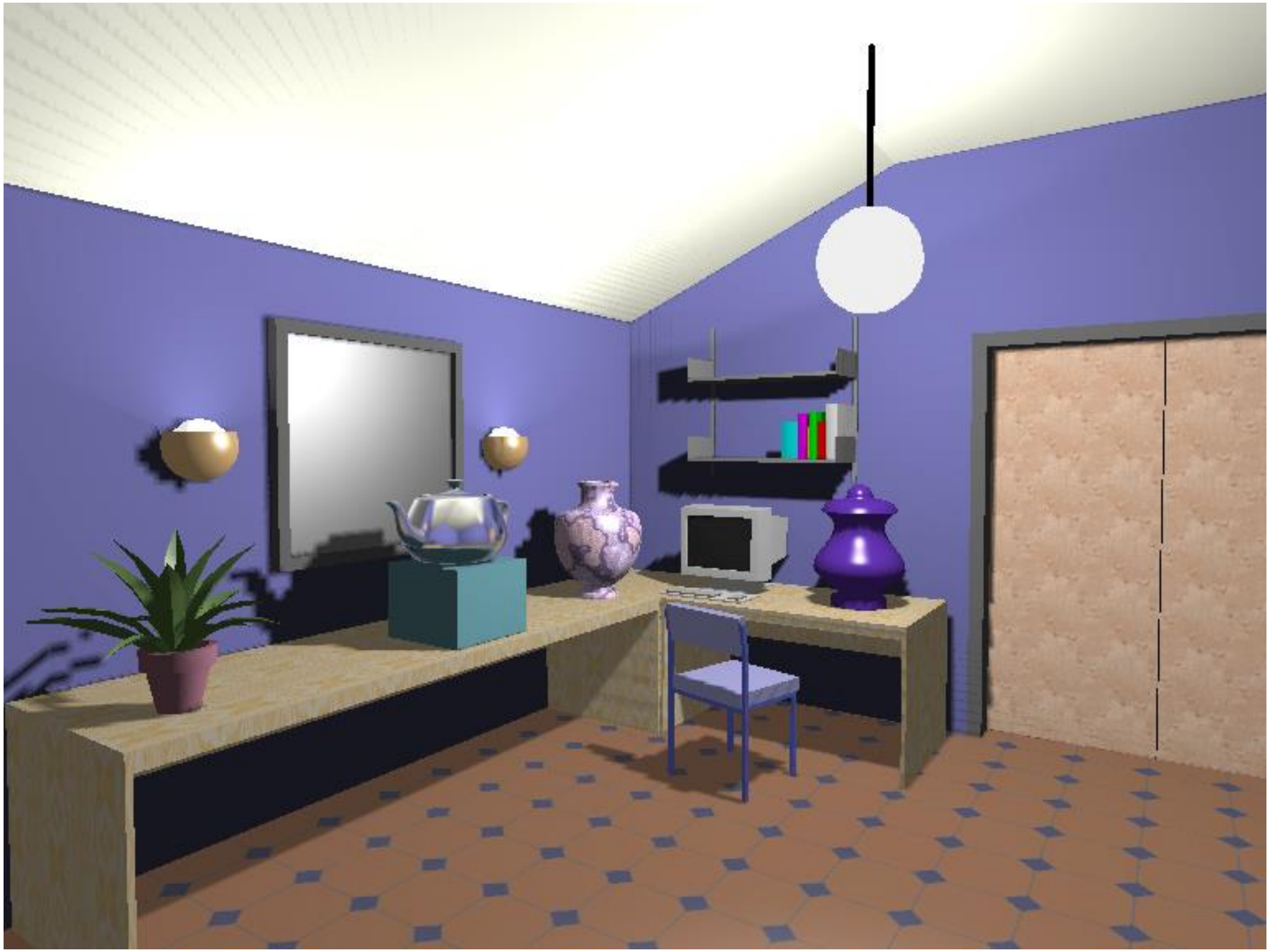


Transform fragment z to light system z’
and compare with z value in shadow map

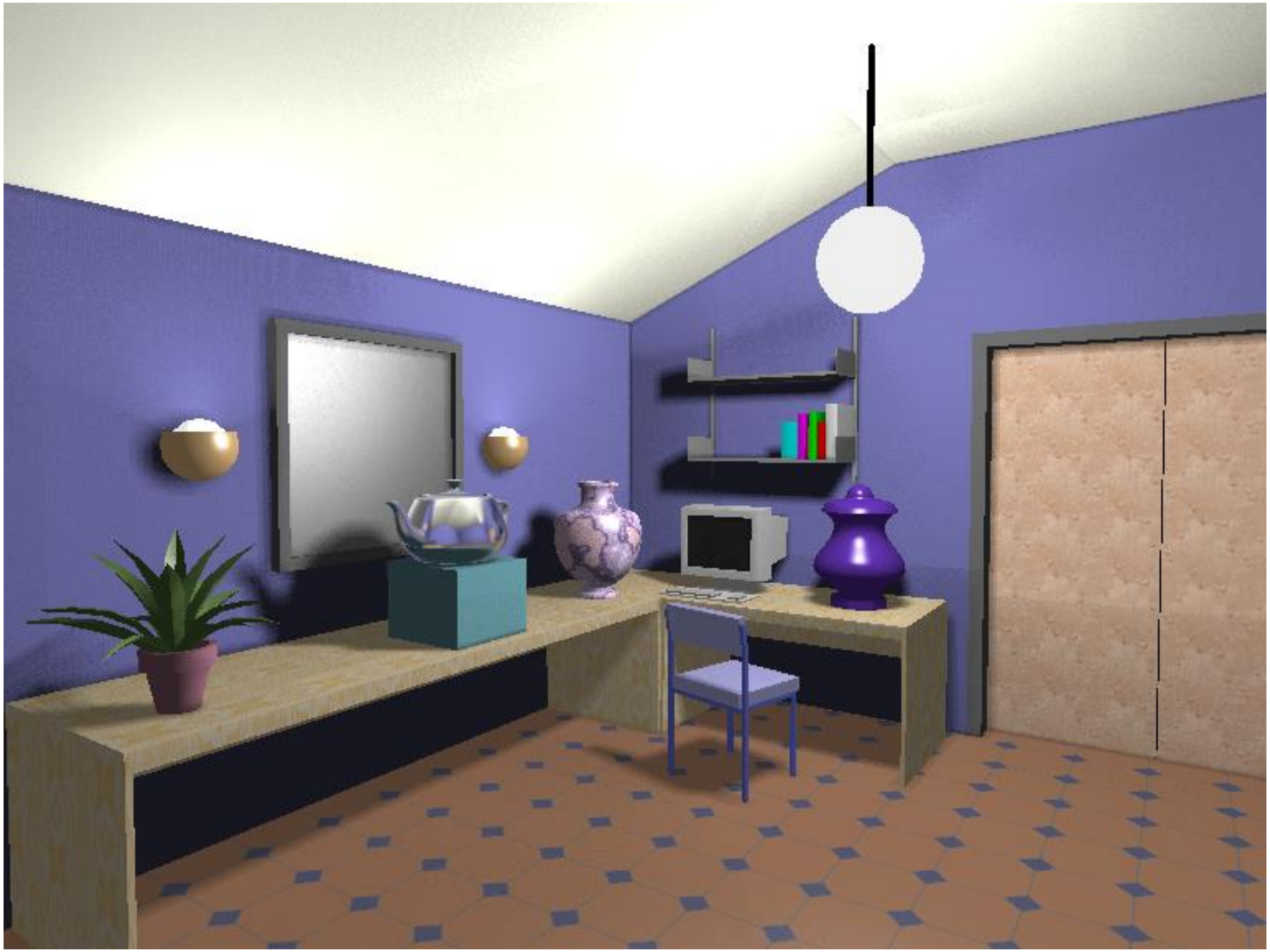


Not in shadow

With shadows



Lowering the shadow map resolution increases the blocky look at the shadow edges and also causes shadows to break up, e.g. shadow of the leaves

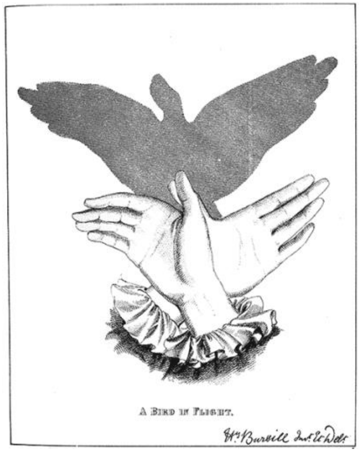


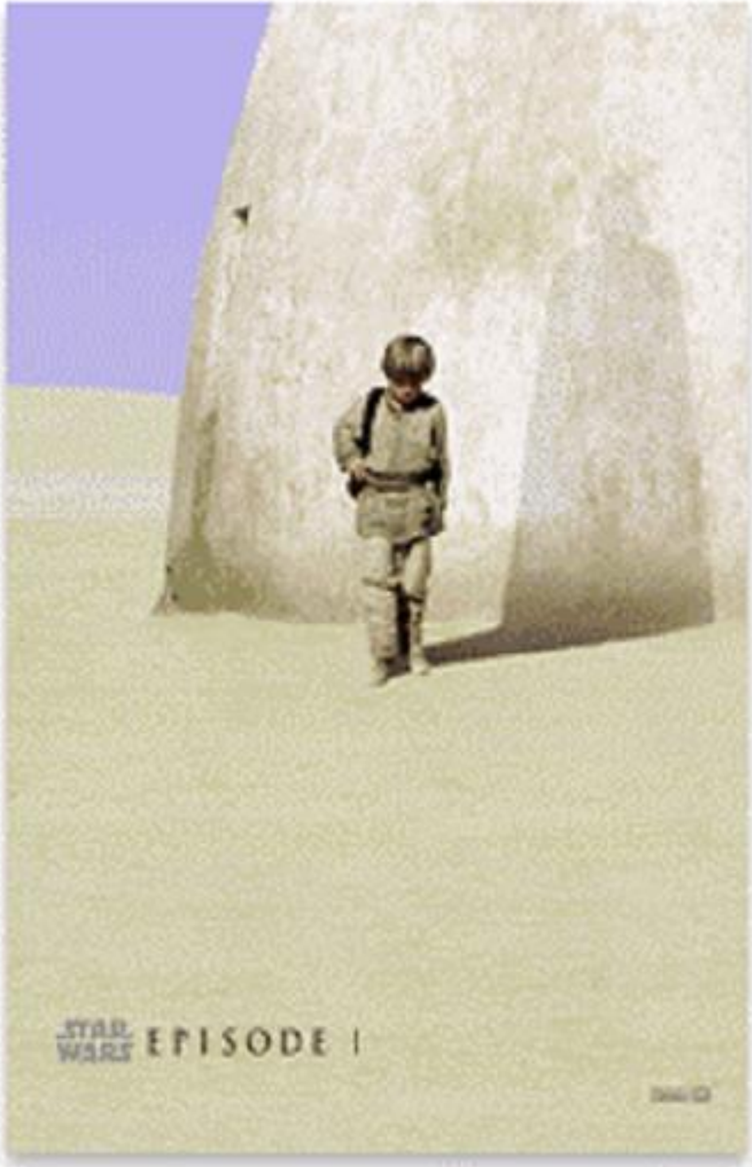
Increasing the shadow map resolution diminishes the blockiness

7. Summary

- ‘Geometric shadows’ since we calculate the shape of the shadow but only guess at its reflected light intensity
  - We need a global illumination algorithm to determine a better light intensity – see later lectures
- Exploit:
  - Shadows are hidden from the light source
  - Shadows do not change in a static scene
- Shadow z-buffer can be done in hardware
  - Less sensitive to geometric complexity than shadow volumes
  - Issues with aliasing
- Because of the high computational overheads, shadows have been regarded in much the same way as texture mapping – as a quality add-on

But, shadows might not be what they seem…





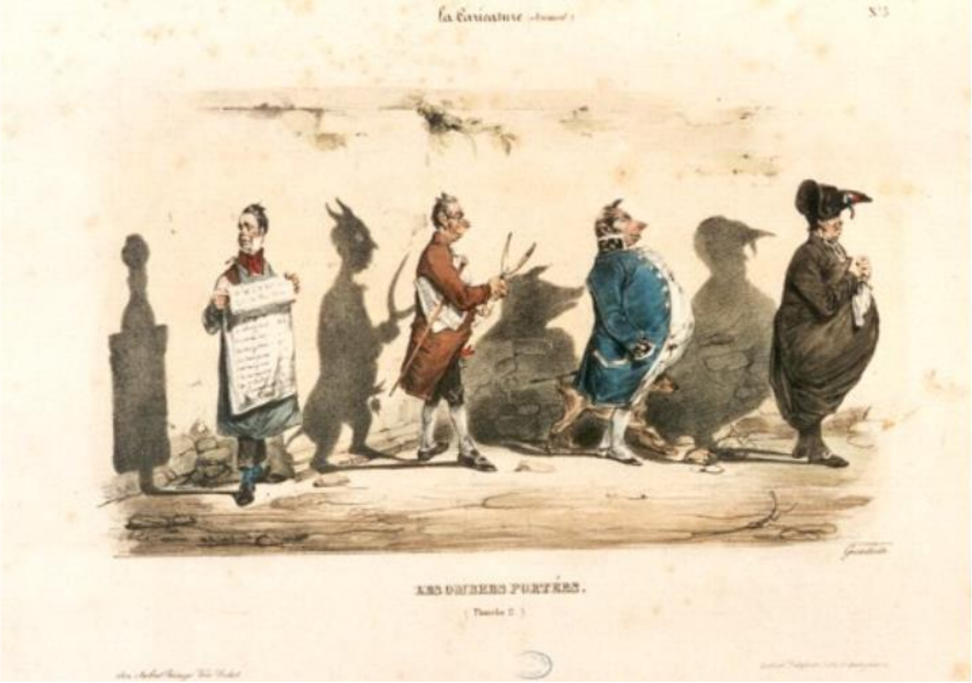
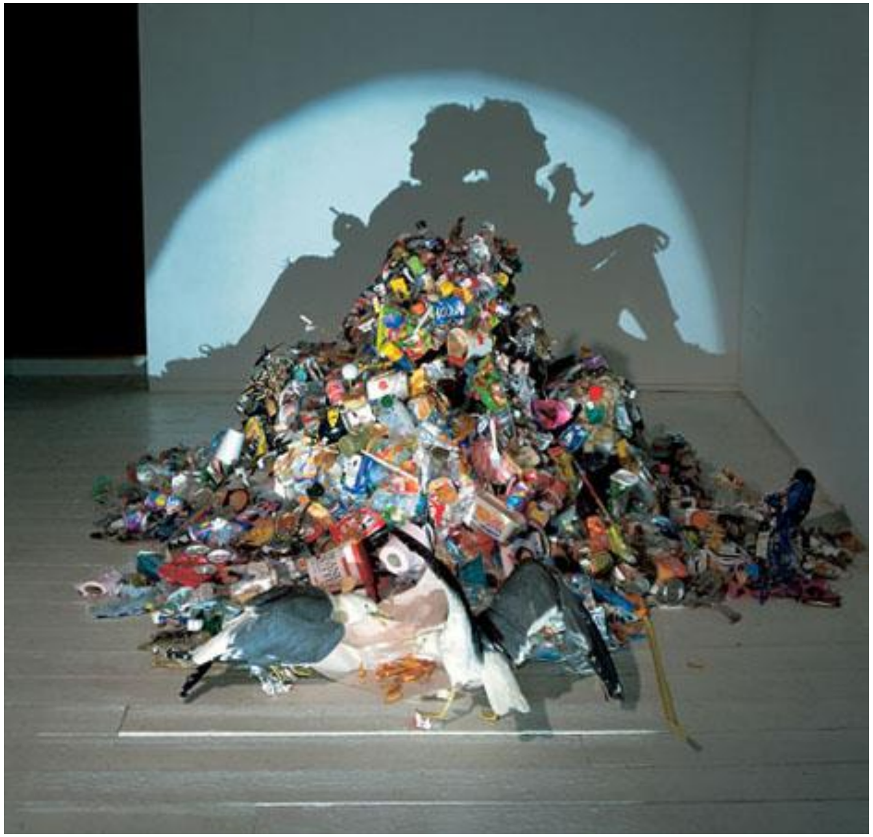


Plate 52 Grandville, *The Shadows (The French Cabinet)* from *La Caricature*, 1830.



"Dirty White Trash (with Gulls)", 1998

http://en.wikipedia.org/wiki/Tim_Noble_and_Sue_Webster

More examples of similar work:

http://visualfunhouse.com/snapshot_illusions/shadow-illusions.html

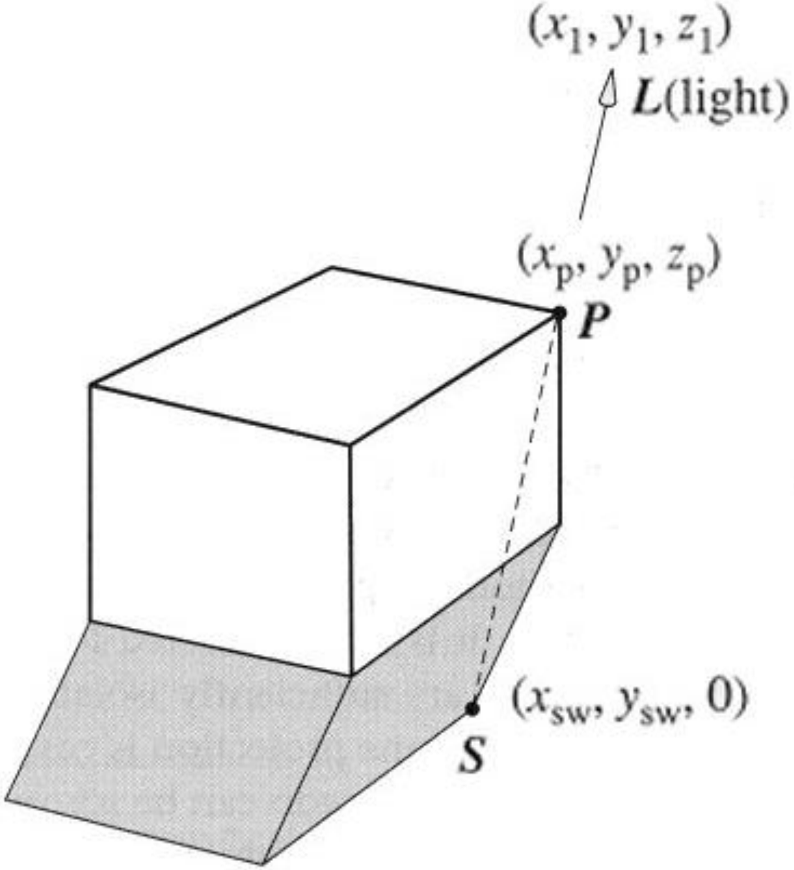
John Lewis Christmas 2007 TV ad "Shadows

References

- A. Appel, “Some Techniques for Shading Machine Renderings of Solids", Proc. AFIPS JSCC, vol. 32,1968, pp. 37-45.
- P. Atherton, K. Weiler, D. Greenberg, “Polygon Shadow Generation", Computer Graphics, 12(3), August 1978, pp. 275-281.
- P. Bergeron. A general version of Crow´s shadow volumes. IEEE Computer Graphics & Applications, 6(9):17–28, 1986.
- J. Blinn, "Me and my (fake) shadow", IEEE Computer Graphics and Applications, January 1988.
- W. Bouknight, K. Kelley, “An Algorithm for Producing Half-Tone Computer Graphics Presentations Shadows and Movable Light Sources", AFIPS Conf. Proc., vol. 36, 1970, pp. 1-10.
- S. Brabec, Shadow Techniques for Interactive and Real-Time Applications, PhD Thesis, MPI Informatik, Saarbrücken, Germany, 2004.
- L. Brotman, N. Badler, “Generating Soft Shadows with a Depth Buer Algorithm", IEEE CG&A, 4(10), October 1984, pp. 71-81
- Chen, J.X., Lecture notes for CS 752 Interactive Graphics Software, George Mason University, 2008. *www.cs.gmu.edu/~jchen/cs662/cjx1shadow.ppt*
- F. Crow. Shadows Algorithms for Computer Graphics. Computer Graphics (Proceedings of SIGGRAPH 77), July 1977, pages 242-248
- Durand, F., Course notes for MIT course 6.837: Computer Graphics, 2008
- W. Reeves, D. Salesin, R. Cook, “Rendering Anti-Aliased Shadows with Depth Maps", Computer Graphics, 21(4), July 1987, pp. 283-291.
- M. Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran, Paul Haeberli ., “Fast Shadows and Lighting Effects Using Texture Mapping,” SIGGRAPH 92
- Slater,M., A.Steed, Y. Chrysanthou, “Computer Graphics and Virtual Environments”,: From Realism to Real-Time”, Addison Wesley, 2002.
- Thomas, F. and O.Johnston, “Disney Animation: The Illusion of Life”, Abbeyville Publishers, NY, 1981.
- L. Williams,”Casting Curved Shadows on Curved Surfaces", Computer Graphics, 12(3), August 1978, pp. 270-274.
- A.Woo P. Poulin A. Fournier. A survey of Shadow Algorithms. IEEE Computer Graphics and Applications, Volume 10, Issue 6 (November 1990) pp. 13 - 32

A1. Simple shadows on a ground plane (Blinn, 88)

- Single objects throwing shadows on flat ground plane.
- Ground plane projection.
- Z-buffer: scan ground plane, then shadow, then object.
- Approximation: single point source at an infinite distance.



A1. Simple shadows on a ground plane (Blinn, 88)

- Parallel light rays in a direction **L** = (x_l,y_l,z_l).
- A point on object **P** = (x_p,y_p,z_p) casts a shadow at **S** = $(x_{sw},y_{sw},0)$.

  S = **P** - α**L**

- and given that $z_{sw} = 0$, we have:

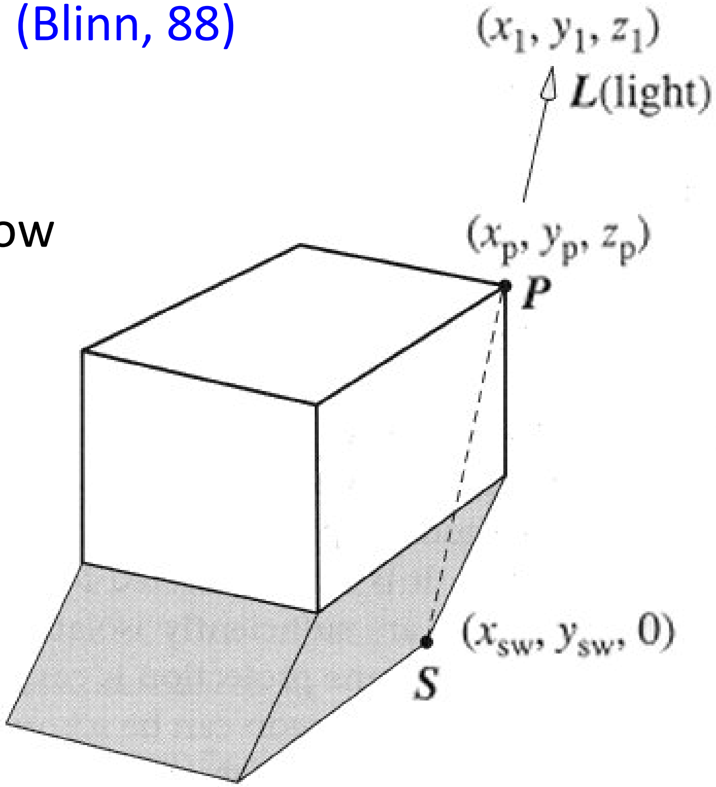
  $0 = z_p - \alpha z_l$

  $\alpha = z_p/z_l$

- and $x_{sw} = x_p - (z_p/z_l)\, x_l$

  $y_{sw} = y_p - (z_p/z_l)\, y_l$

- As a homogeneous transformation →
- If light source at finite distance:
  - **S** = **P** + α(**P**-**L**)



$$\begin{bmatrix} x_{sw} \\ y_{sw} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_1/z_1 & 0 \\ 0 & 1 & -y_1/z_1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$