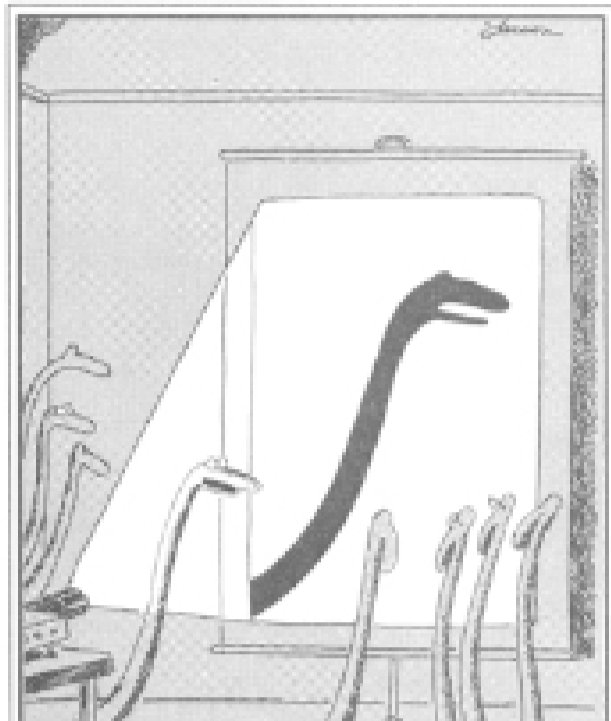


Shadows

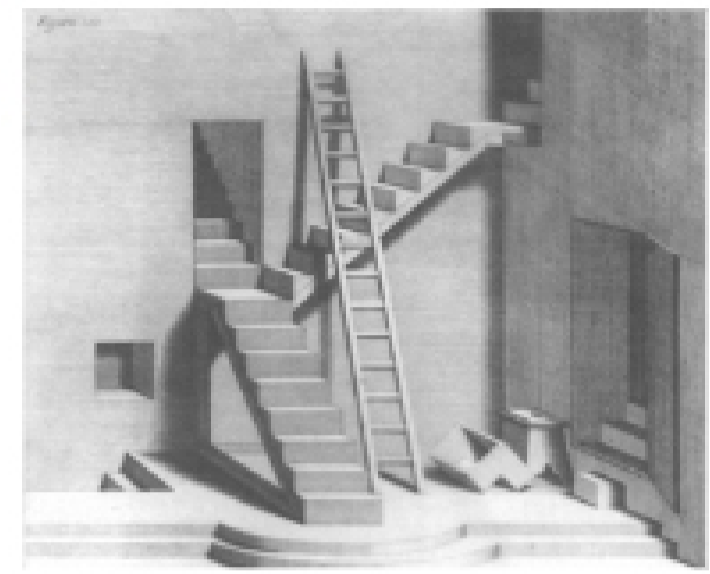
Thanks to:
Frédo Durand
and Seth Teller
MIT



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

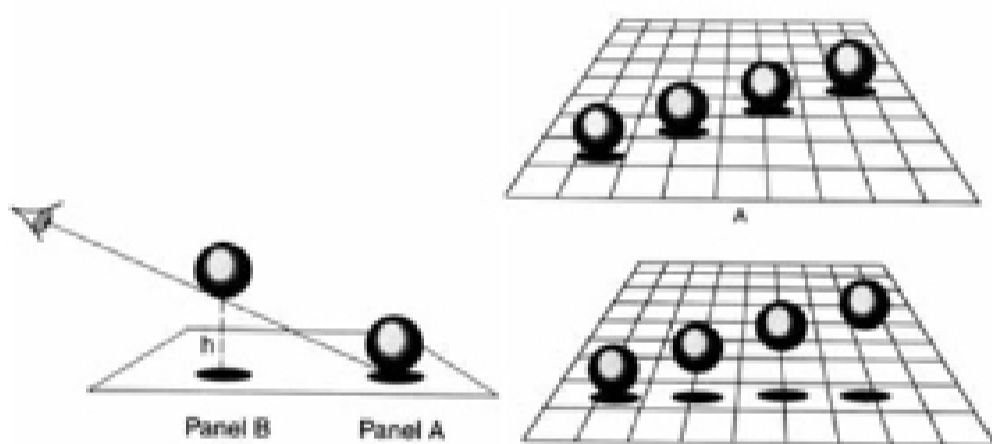
Shadows

- Realism
- Depth cue



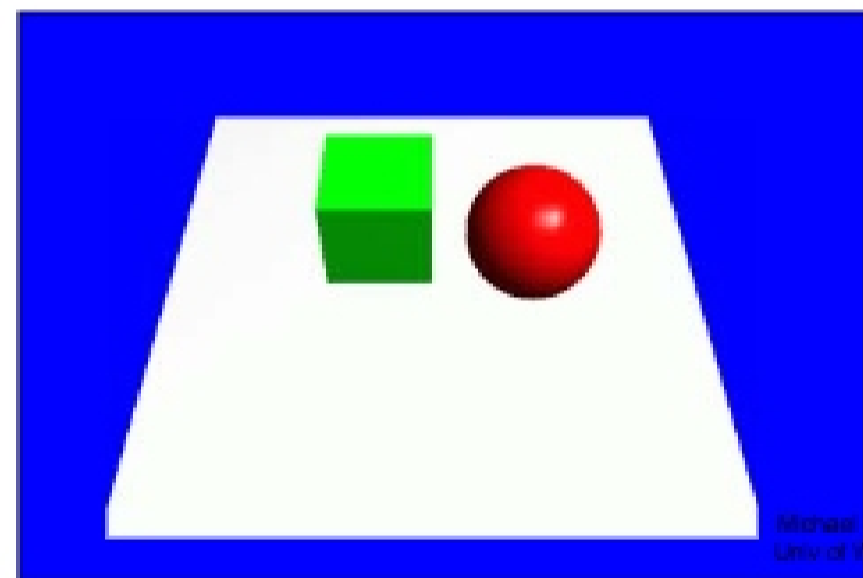
2

Shadows as depth cue



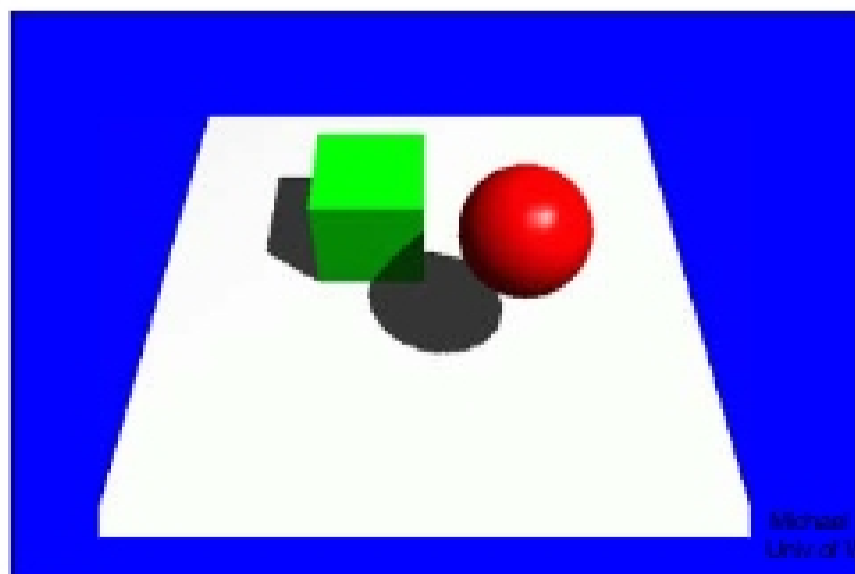
3

Spatial relationship between objects



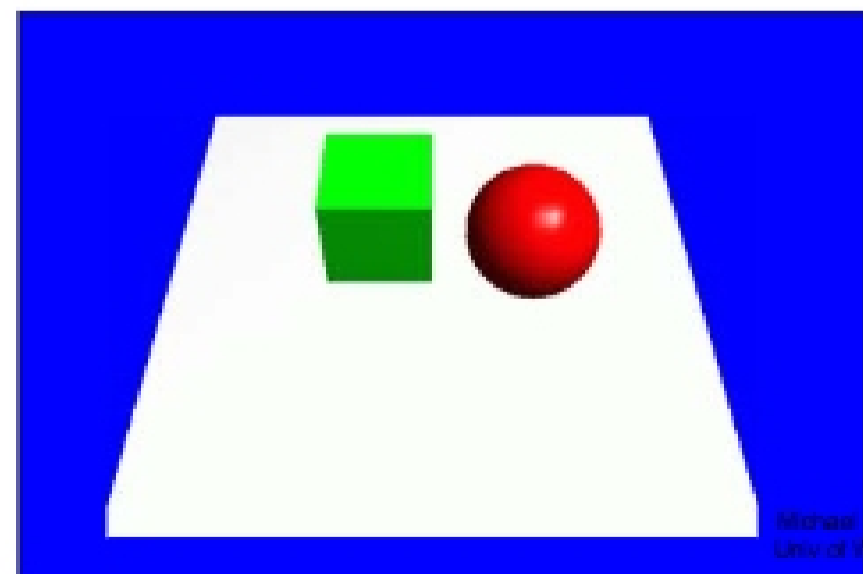
Michael McCool
Univ. of Waterloo

Spatial relationship between objects



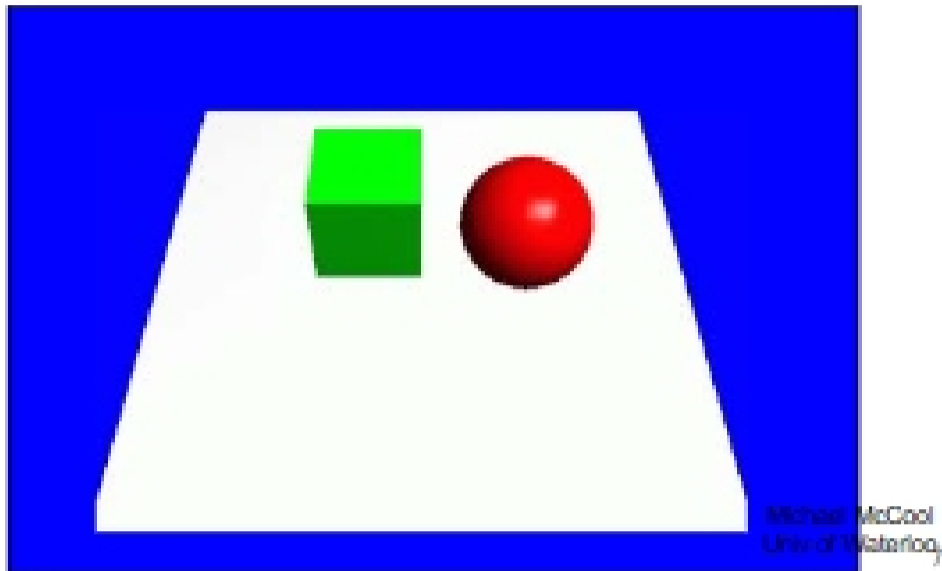
Michael McCool
Univ. of Waterloo

Spatial relationship between objects

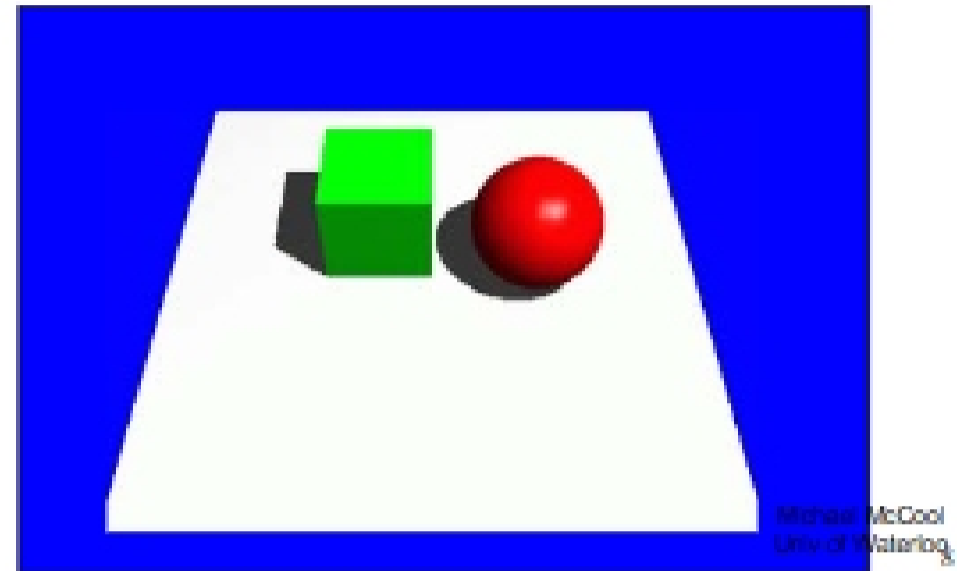


Michael McCool
Univ. of Waterloo

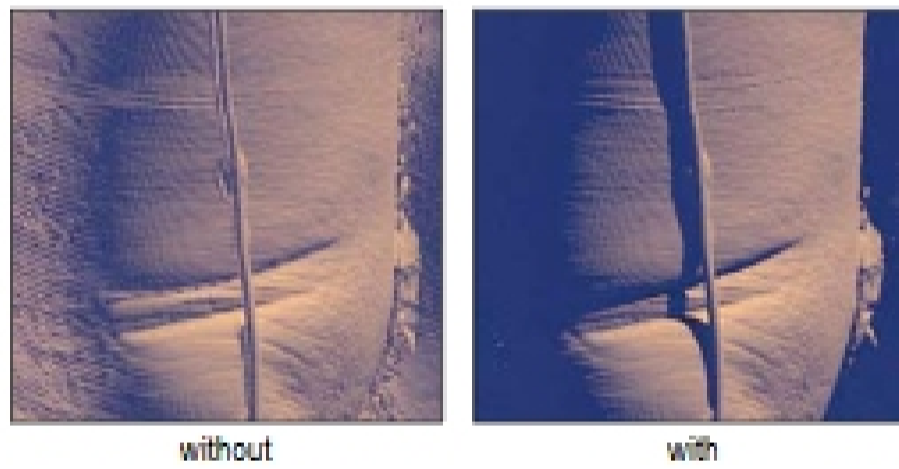
Spatial relationship between objects



Spatial relationship between objects



Shadows add visual acuity



9

Shadows and art

- Only in Western pictures (here Caravaggio)



10

Shadows and art

- Shadows as the origin of painting

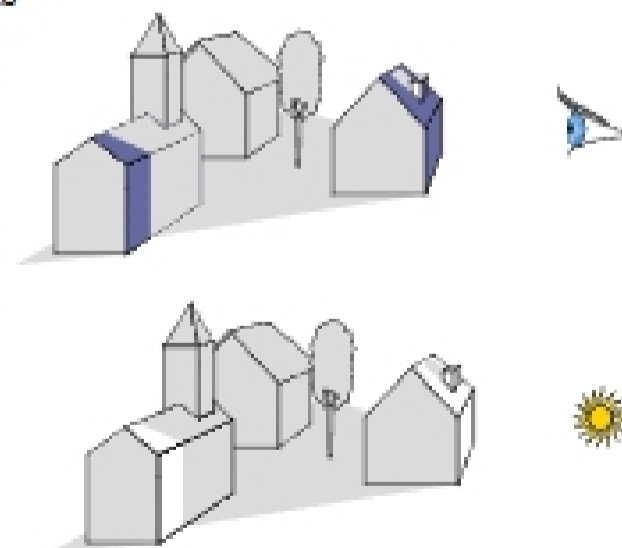


Plate 10
David Allan,
The Origin of Painting
(*The Maid of Curich*),
1725. Oil on wood,
38.7 x 31 cm, Edinburgh,
National Gallery of
Scotland.

11

Duality shadow-view

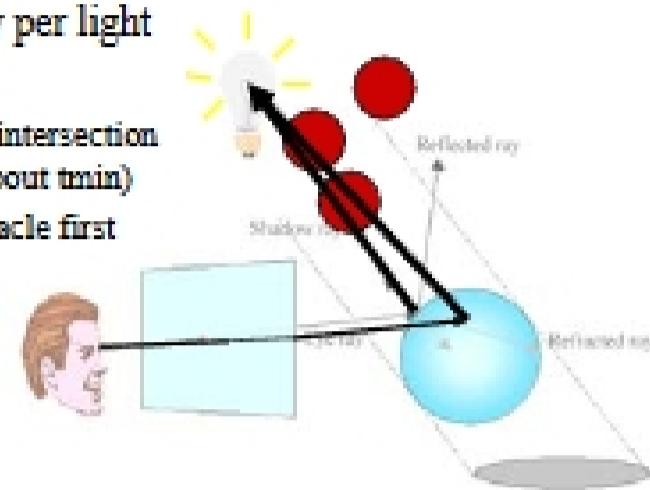
- A point is lit if it is visible from the light source
- Shadow computation very similar to view computation



12

Shadow ray

- Ray from visible point to light source
- If blocked, discard light contribution
- One shadow ray per light
- Optimization?
 - Stop after first intersection (don't worry about tmin)
 - Test latest obstacle first



Ray-casting shadows

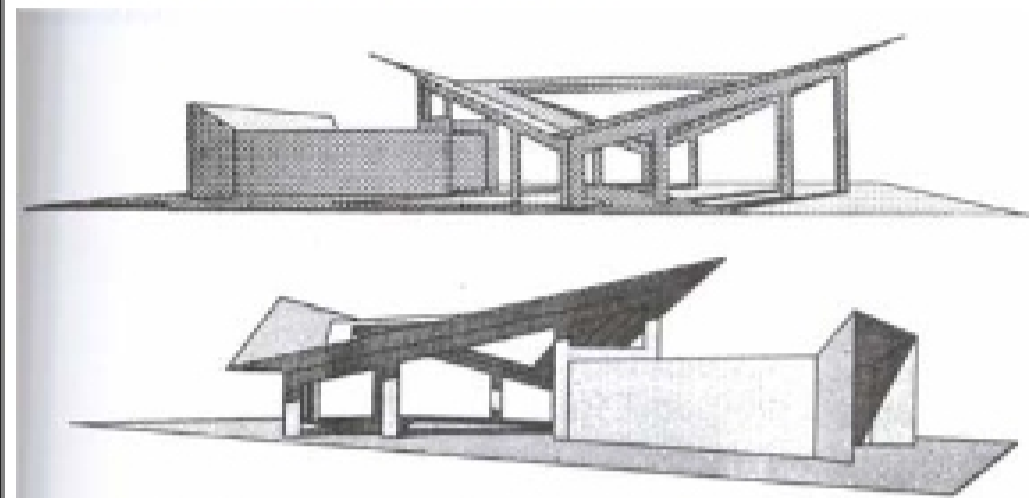
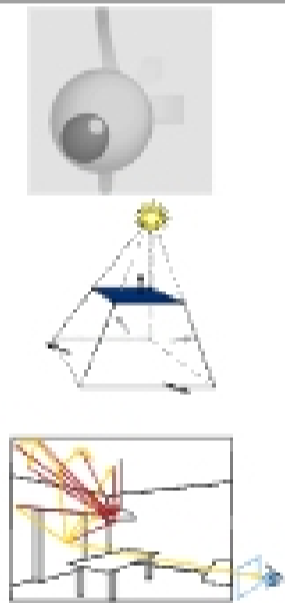


Fig. 16.52 Early pictures rendered with ray tracing. (Courtesy of Arthur Appel, IBM T.J. Watson Research Center.)

14

Overview

- Project Shadows
- Shadow map
 - Image-precision, texture mapping
- Shadow volume
 - Object space
- Soft shadows



15

Planar Projection

- Render a ground-plane
- Render an object
- Then render the object again, but this time
 - Projected onto the plane
 - Without light, so that the shadow is black
 - Half transparent (using blending), to avoid completely dark shadows
 - Avoid multiple “darkening” on one spot by using ordinary z-buffer checks

16

Planar Shadows (& demo)



Shadow is projected into the plane of the floor.

17

Constructing a Shadow Matrix

```
void shadowMatrix(GLfloat shadowMat[4][4], GLfloat groundPlane[4], GLfloat lightVec[3])
{
    GLfloat det;
    // Find det product between light position vector and ground plane normal.
    det = groundPlane[0] * lightVec[2] -
        groundPlane[1] * lightVec[1] +
        groundPlane[2] * lightVec[0] -
        groundPlane[3] * lightVec[3];
    shadowMat[0][0] = det * lightVec[0] * groundPlane[0];
    shadowMat[1][0] = det * lightVec[0] * groundPlane[1];
    shadowMat[2][0] = det * lightVec[0] * groundPlane[2];
    shadowMat[3][0] = det * lightVec[0] * groundPlane[3];
    shadowMat[0][1] = det * lightVec[1] * groundPlane[0];
    shadowMat[1][1] = det * lightVec[1] * groundPlane[1];
    shadowMat[2][1] = det * lightVec[1] * groundPlane[2];
    shadowMat[3][1] = det * lightVec[1] * groundPlane[3];
    shadowMat[0][2] = det * lightVec[2] * groundPlane[0];
    shadowMat[1][2] = det * lightVec[2] * groundPlane[1];
    shadowMat[2][2] = det * lightVec[2] * groundPlane[2];
    shadowMat[3][2] = det * lightVec[2] * groundPlane[3];
    shadowMat[0][3] = det * lightVec[3] * groundPlane[0];
    shadowMat[1][3] = det * lightVec[3] * groundPlane[1];
    shadowMat[2][3] = det * lightVec[3] * groundPlane[2];
    shadowMat[3][3] = det * lightVec[3] * groundPlane[3];
}
```

18