Advanced Rendering

CS 4620 Lecture 37

Announcements

- Welcome back!

- A5 grading today
- A6 grading will not have demos
- A7 due later this week

- Prelim next Thu: Dec 10th, 7pm

The Blue Umbrella

- Pixar short
- Made partly to showcase new more photorealistic rendering
  - –much of it based on the ideas in this lecture

worth a look:
https://vimeo.com/131090328
http://rainycitytales332.tumblr.com

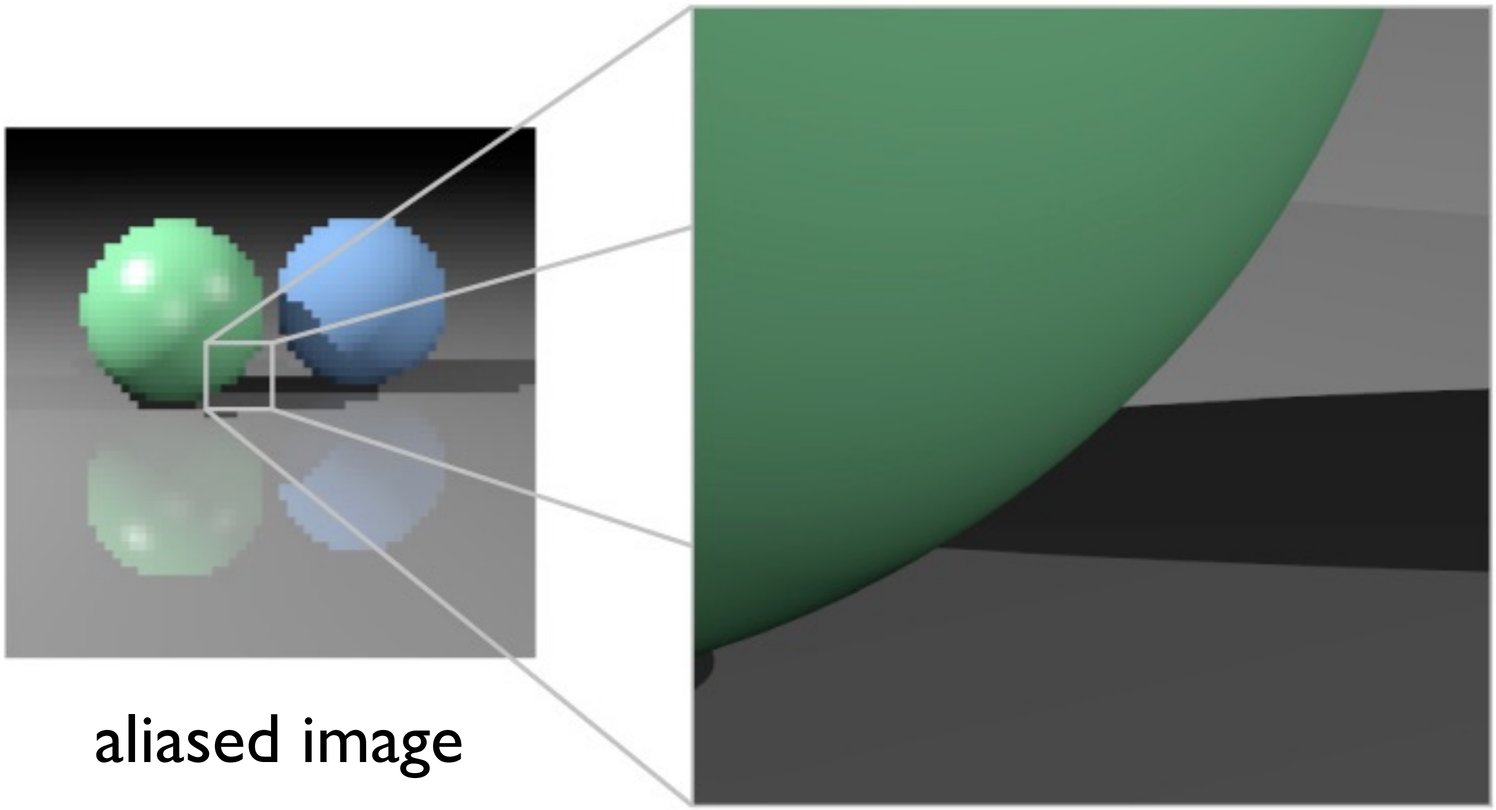
Basic ray tracing

- Basic ray tracer: one sample for everything
  - –one ray per pixel
  - –one shadow ray for every point light
  - –one reflection ray per intersection
    - one refraction ray (if necessary) per intersection
- Many advanced methods build on the basic ray tracing paradigm

Discontinuities in basic RT

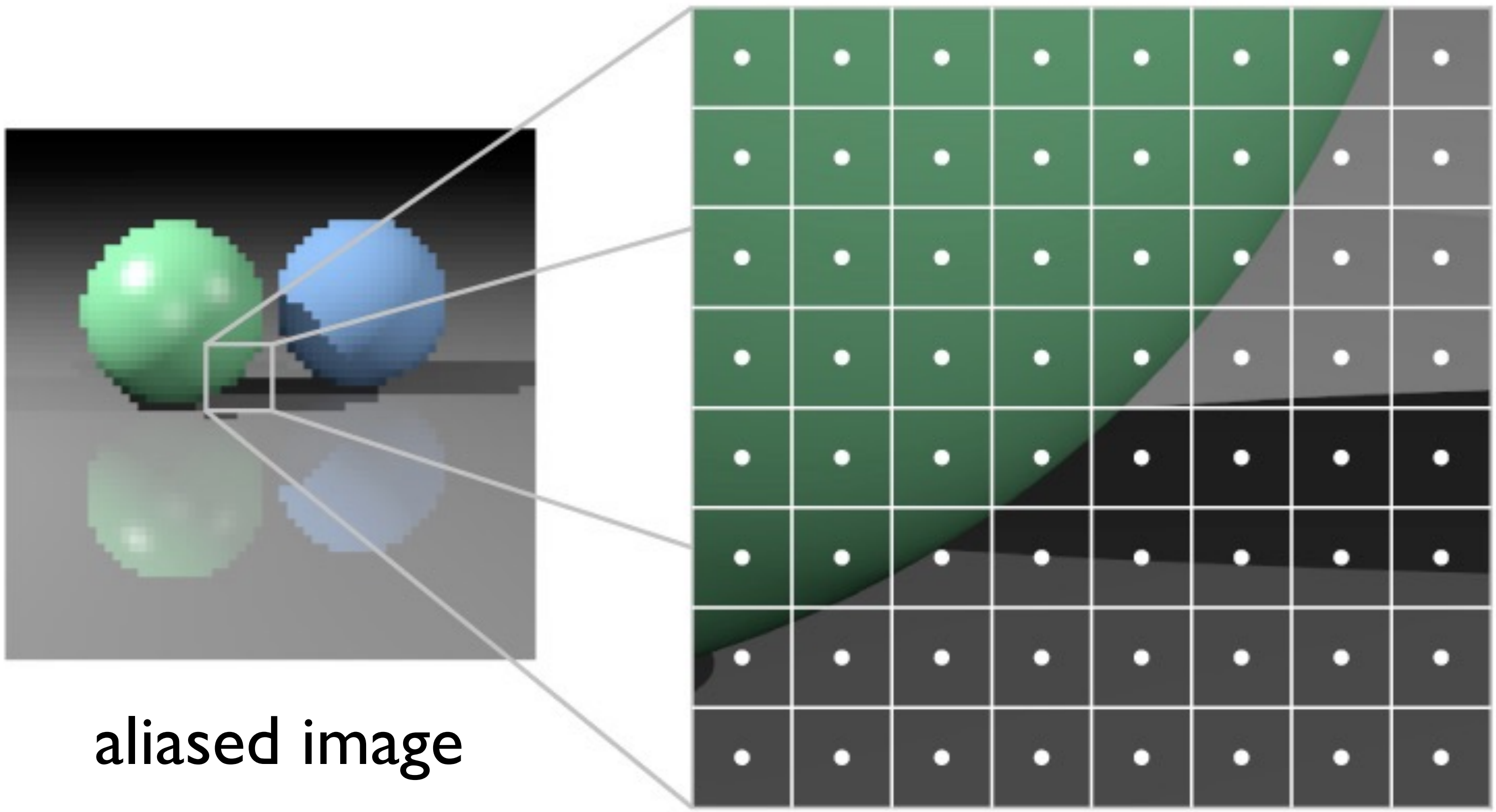
- Perfectly sharp object silhouettes in image
  - –leads to aliasing problems (stair steps)
- Perfectly sharp shadow edges
  - –everything looks like it's in direct sun
- Perfectly clear mirror reflections
  - –reflective surfaces are all highly polished
- Perfect focus at all distances
  - –camera always has an infinitely tiny aperture
- Perfectly frozen instant in time (in animation)
  - –motion is frozen as if by strobe light

Antialiasing in ray tracing



aliased image

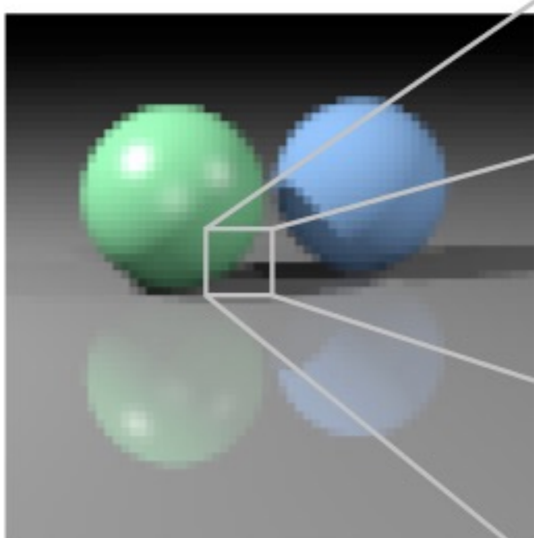
Antialiasing in ray tracing



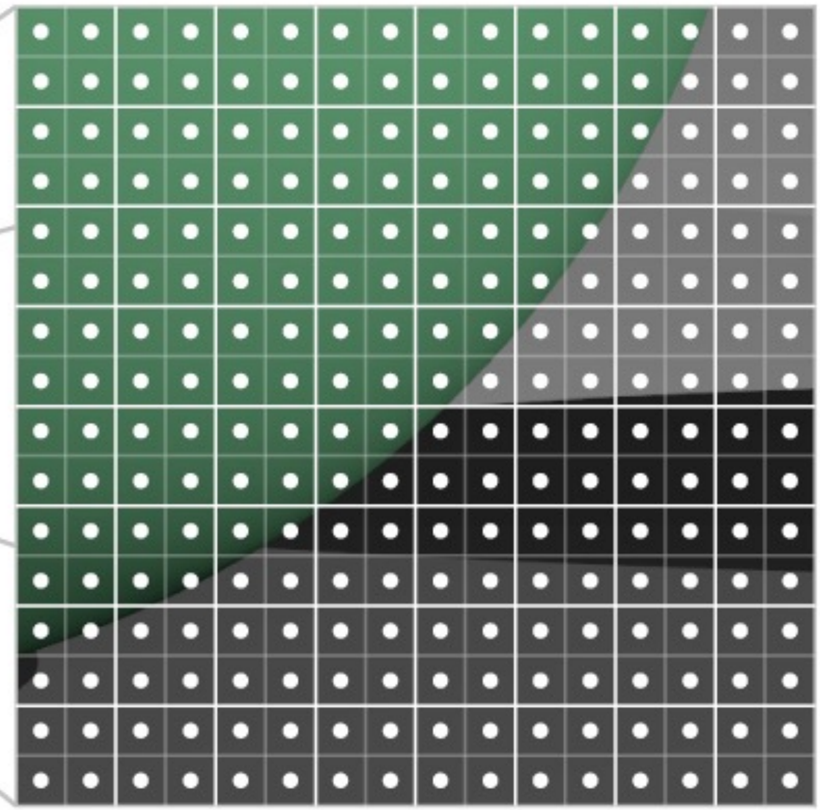
aliased image

one sample per pixel

Antialiasing in ray tracing

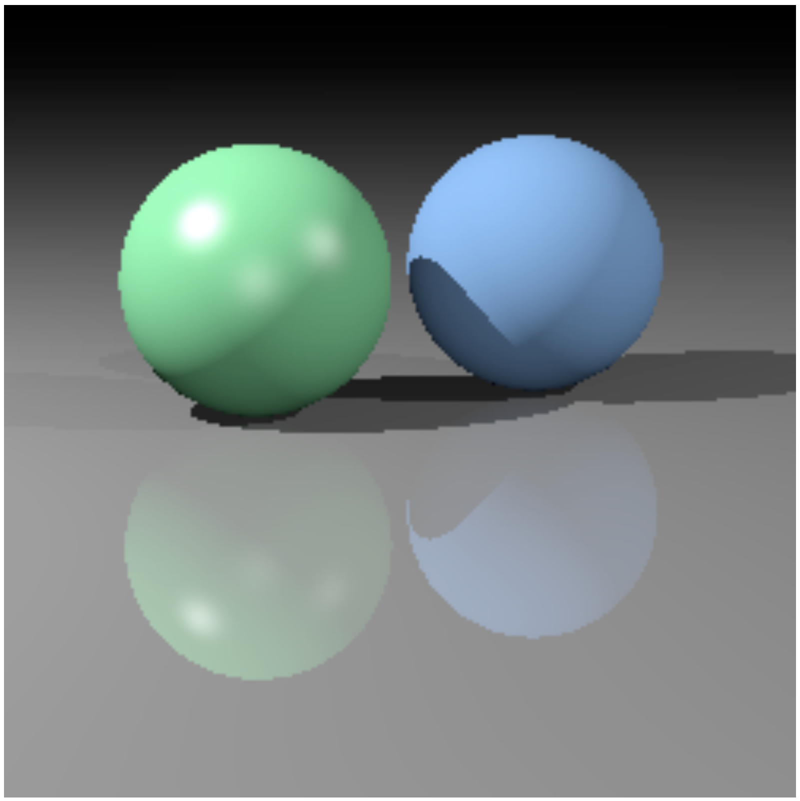


antialiased image

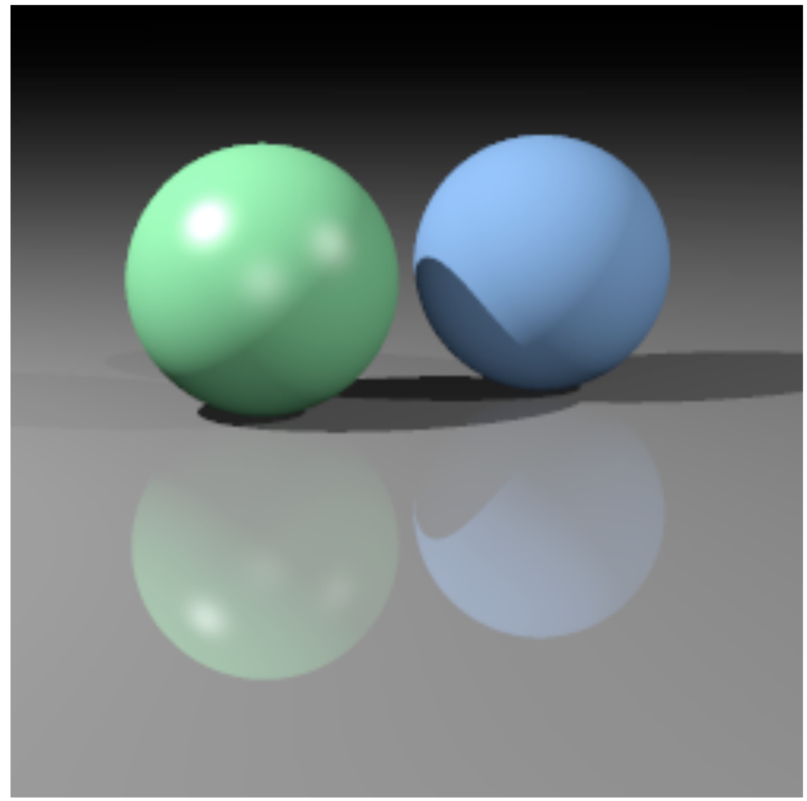


four samples per pixel

Antialiasing in ray tracing



one sample/pixel

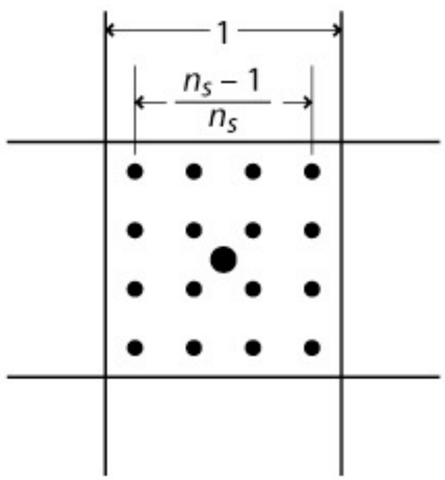


9 samples/pixel

Details of supersampling

- For image coordinates with integer pixel centers:

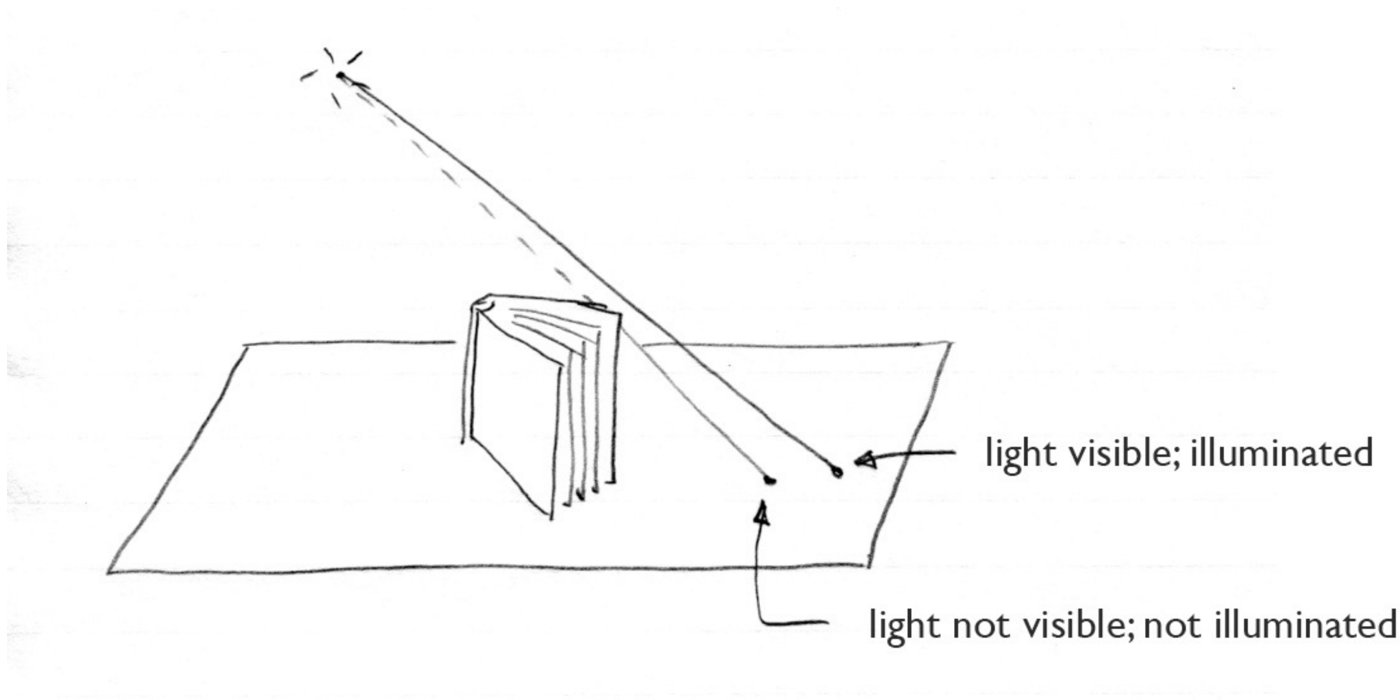
```
// one sample per pixel
for iy = 0 to (ny-1) by 1
  for ix = 0 to (nx-1) by 1 {
    ray = camera.getRay(ix, iy);
    image.set(ix, iy, trace(ray));
  }
```




```
// ns^2 samples per pixel
for iy = 0 to (ny-1) by 1
  for ix = 0 to (nx-1) by 1 {
    Color sum = 0;
    for dx = -(ns-1)/2 to (ns-1)/2 by 1
      for dy = -(ns-1)/2 to (ns-1)/2 by 1 {
        x = ix + dx / ns;
        y = iy + dy / ns;
        ray = camera.getRay(x, y);
        sum += trace(ray);
      }
    image.set(ix, iy, sum / (ns*ns));
  }
```

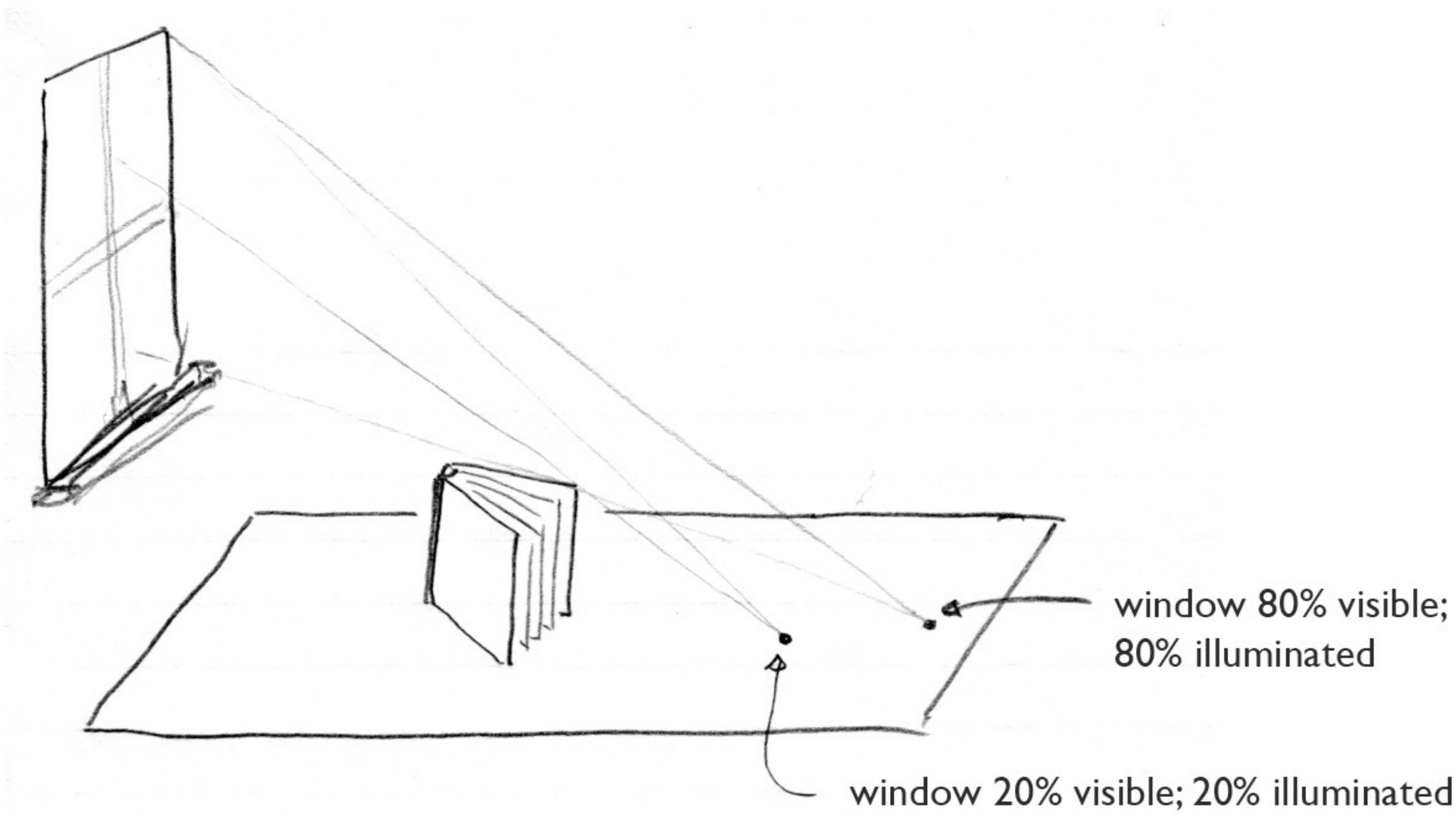
Soft shadows

Cause of soft shadows



point lights cast hard shadows

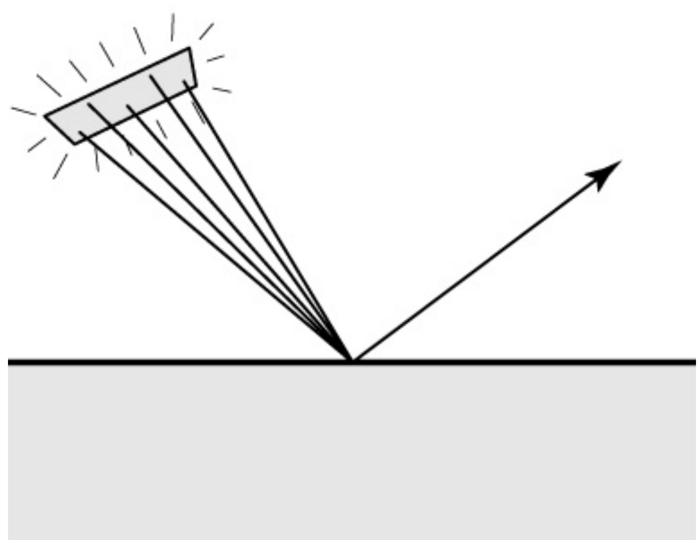
Cause of soft shadows



area lights cast soft shadows

Creating soft shadows

- For area lights: use many shadow rays
  - –and each shadow ray gets a different point on the light
- Choosing samples
  - –general principle: start with uniform in square



Creating soft shadows

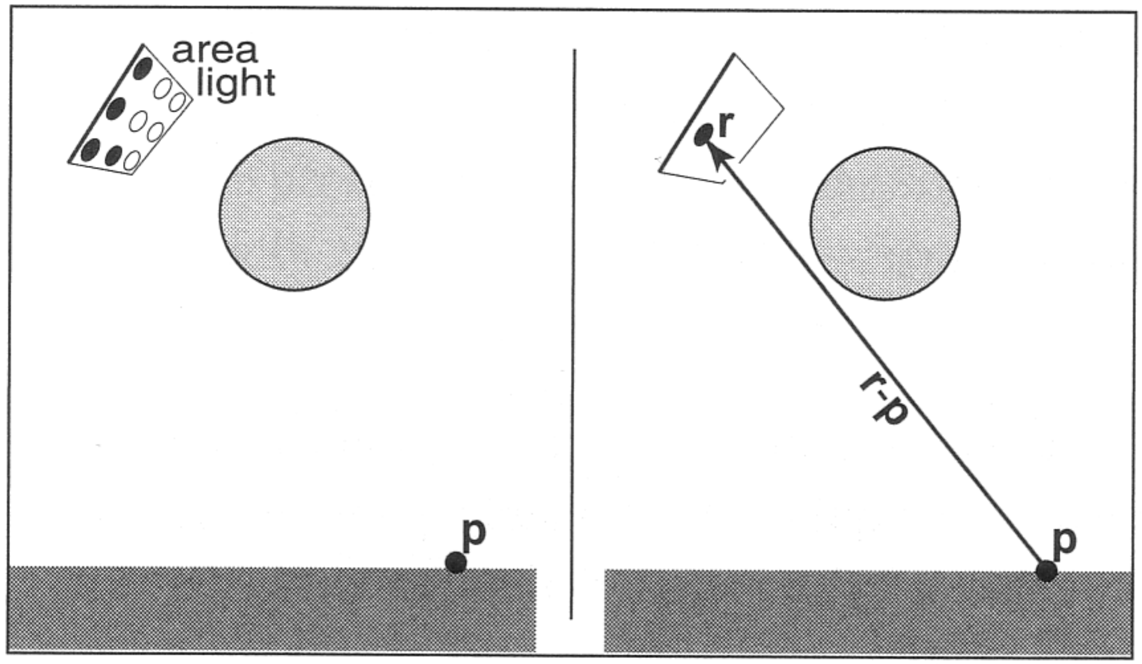
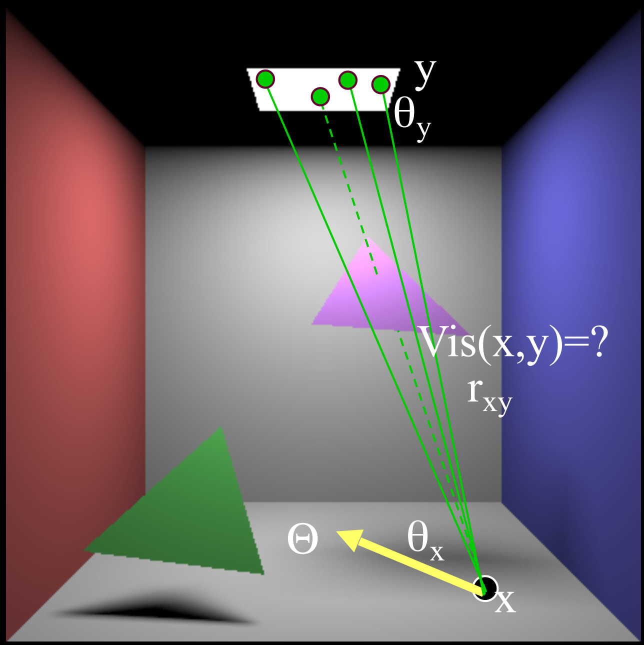


Figure 13.13. Left: an area light can be approximated by some number of point lights; four of the nine points are visible to **p** so it is in the penumbra. Right: a random point on the light is chosen for the shadow ray, and it has some chance of hitting the light or not.

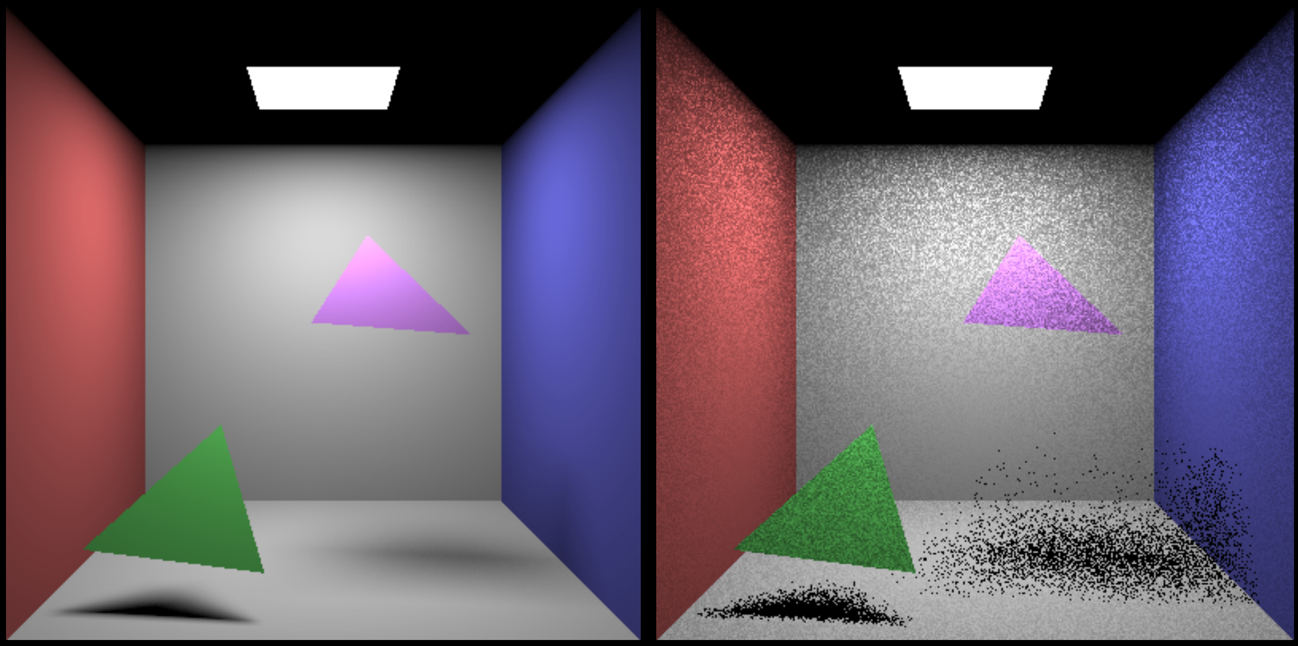
Generating direct paths

- Pick surface points y_i on light source
- Evaluate direct illumination integral



$$\langle L(x \rightarrow \Theta) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f_r(\ldots)L(\ldots)G(x, y_i)}{p(y_i)}$$

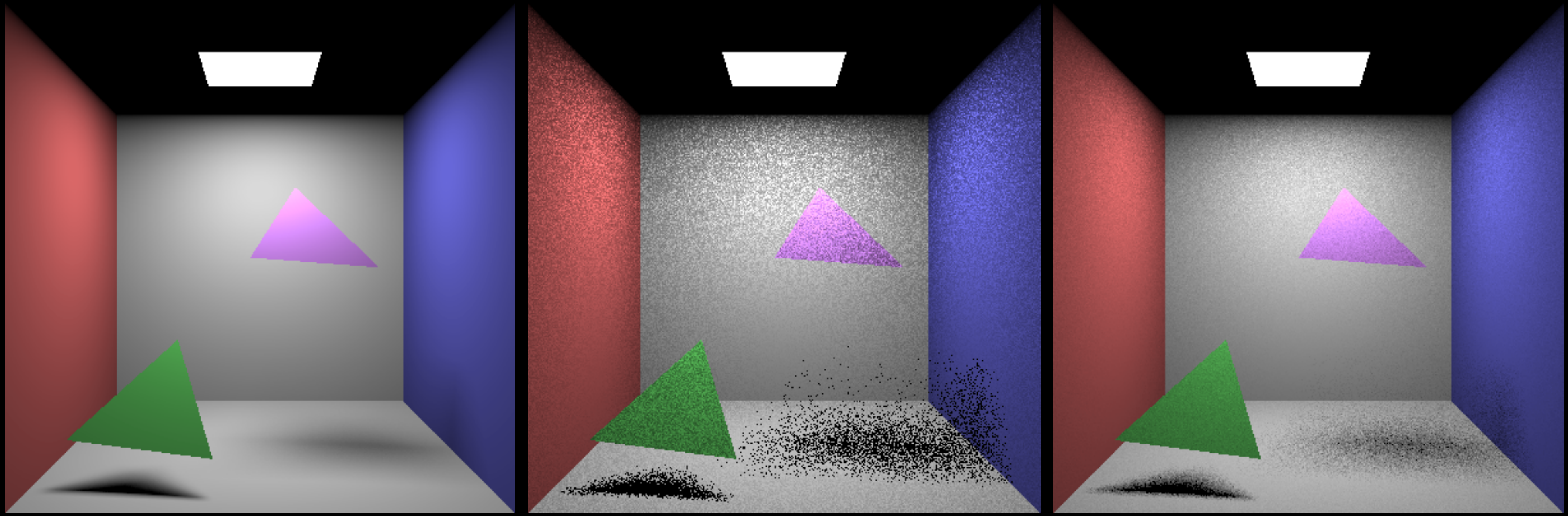
Applied to direct illumination



$$p(y) = \frac{1}{Area_{source}}$$

$$E(x) = Area_{source} L_{source} f_r \frac{\cos\theta_x \cos\theta_{\bar{y}}}{r_{x\bar{y}}^2} Vis(x, \bar{y})$$

More points ...

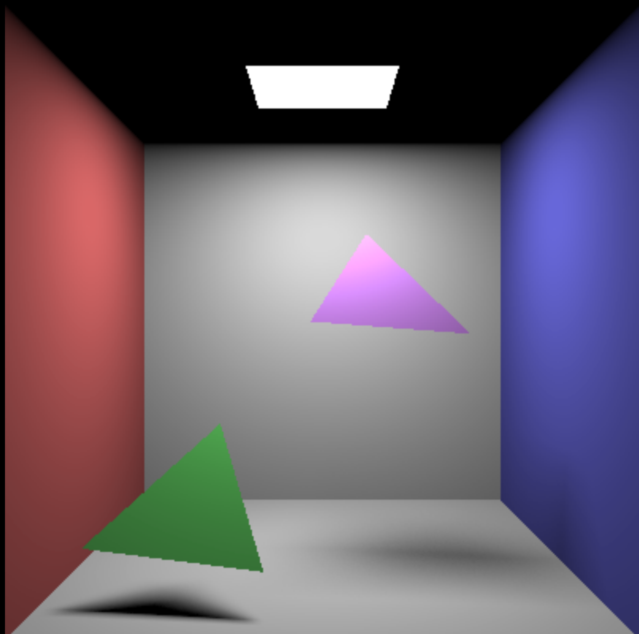


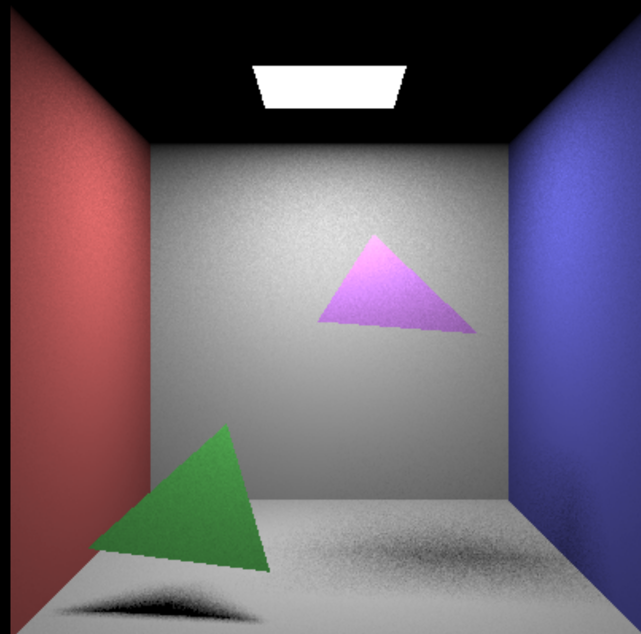
1 shadow ray

9 shadow rays

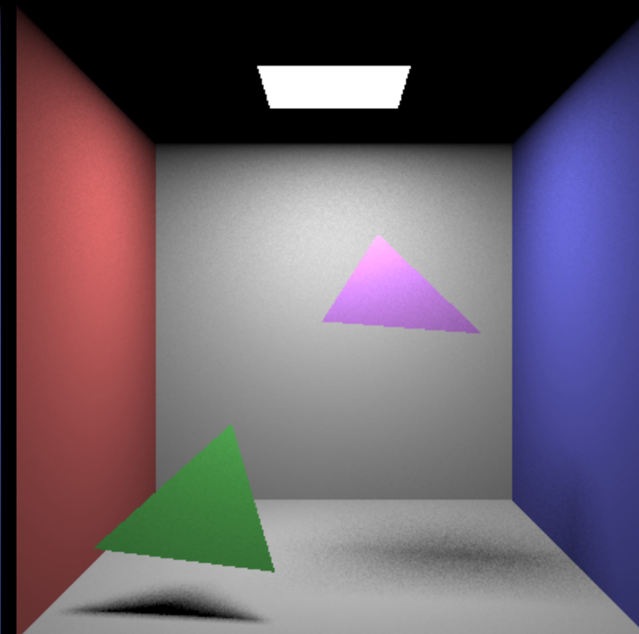
$$E(x) = \frac{Area_{source} f_r L_{source}}{N} \sum_{i=1}^{N} \frac{\cos\theta_x \cos\theta_{\bar{y}_i}}{r_{x\bar{y}_i}^2} Vis(x, \bar{y}_i)$$

Even more points ...





36 shadow rays



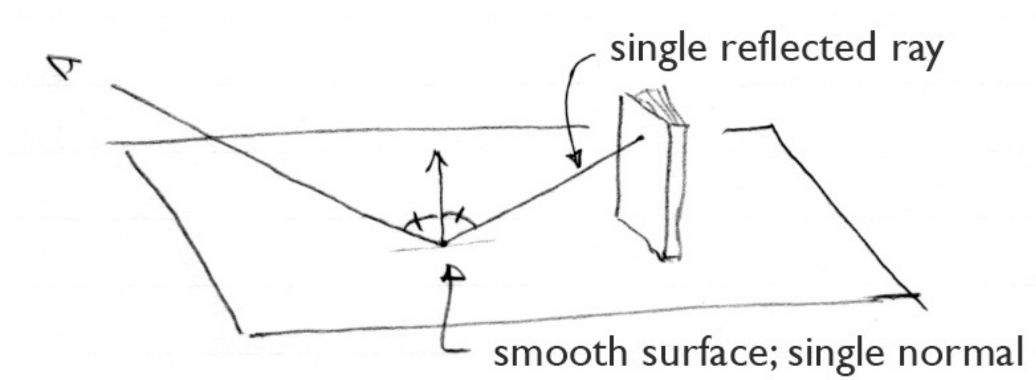
100 shadow rays

$$E(x) = \frac{Area_{source} f_r L_{source}}{N} \sum_{i=1}^{N} \frac{\cos\theta_x \cos\theta_{\bar{y}_i}}{r^2_{x\bar{y}_i}} Vis(x, \bar{y}_i)$$

Glossy reflection

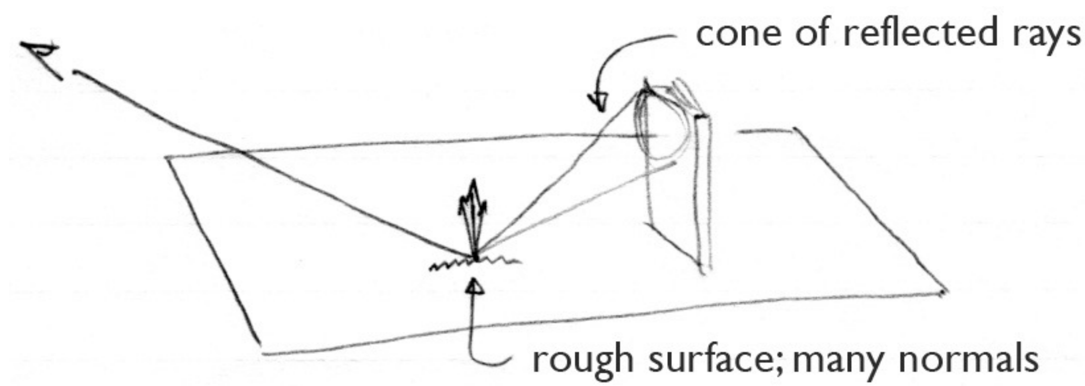
[Lafortune et al. 97]

Cause of glossy reflection



smooth surfaces produce sharp reflections

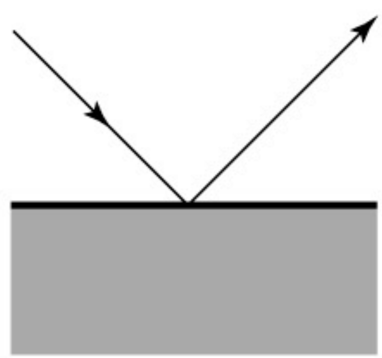
Cause of glossy reflection

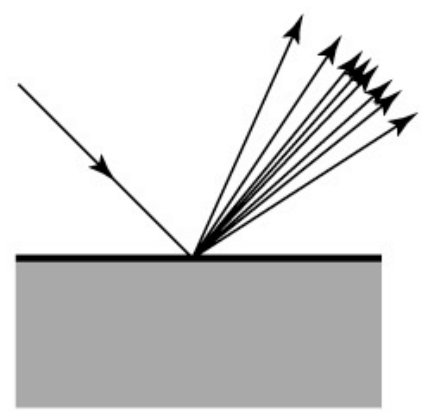


rough surfaces produce soft (glossy) reflections

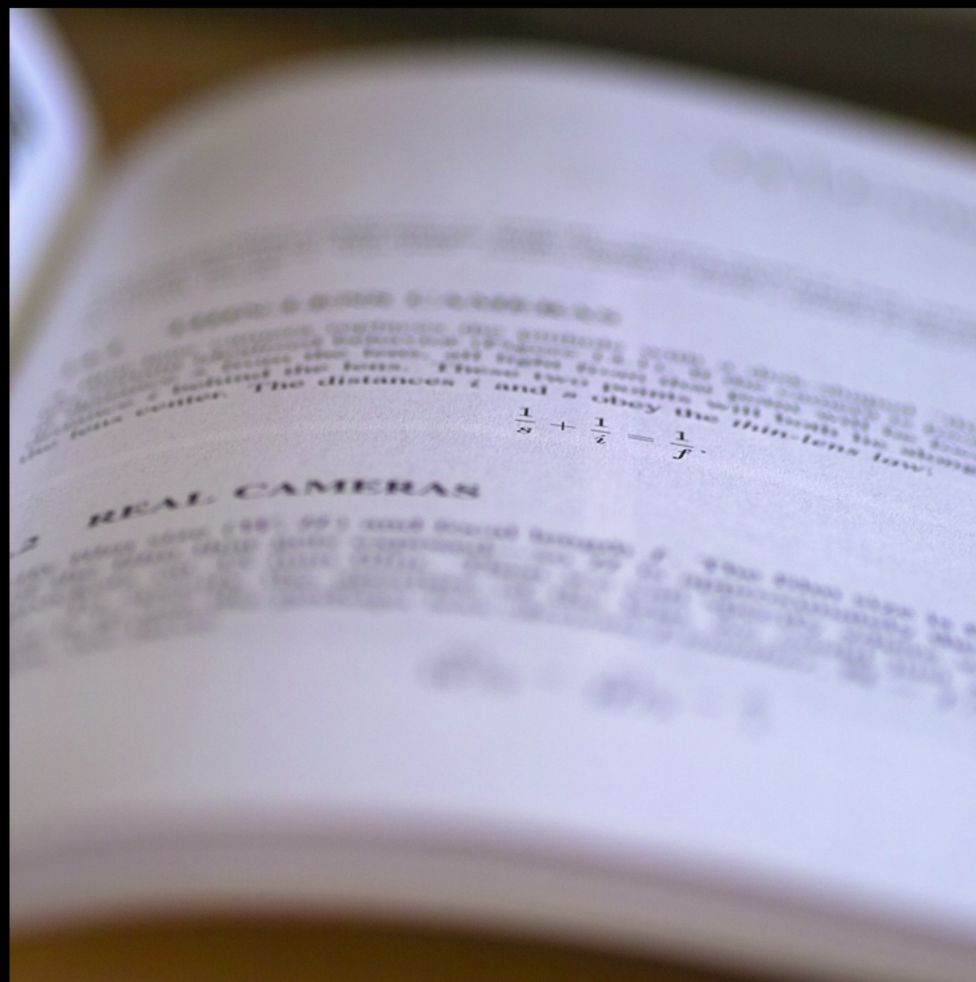
Creating glossy reflections

- Jitter the reflected rays
  - –Not exactly in mirror direction; add a random offset
  - –Can work out math to match Phong exactly
  - –Can do this by jittering the normal if you want

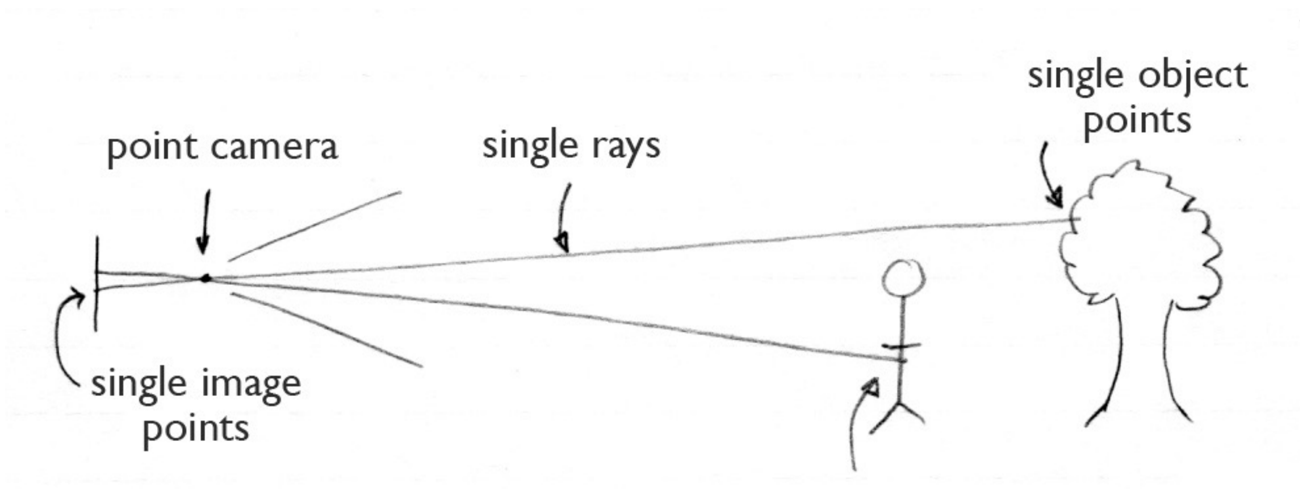




Depth of field

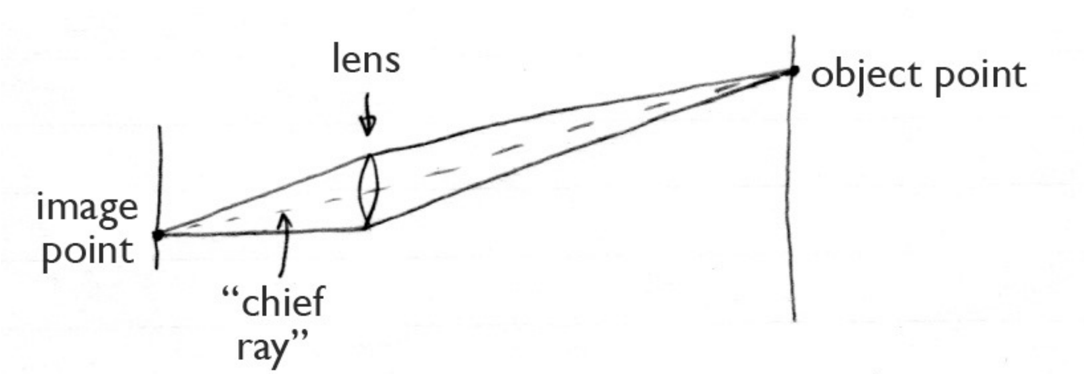


Cause of focusing effects



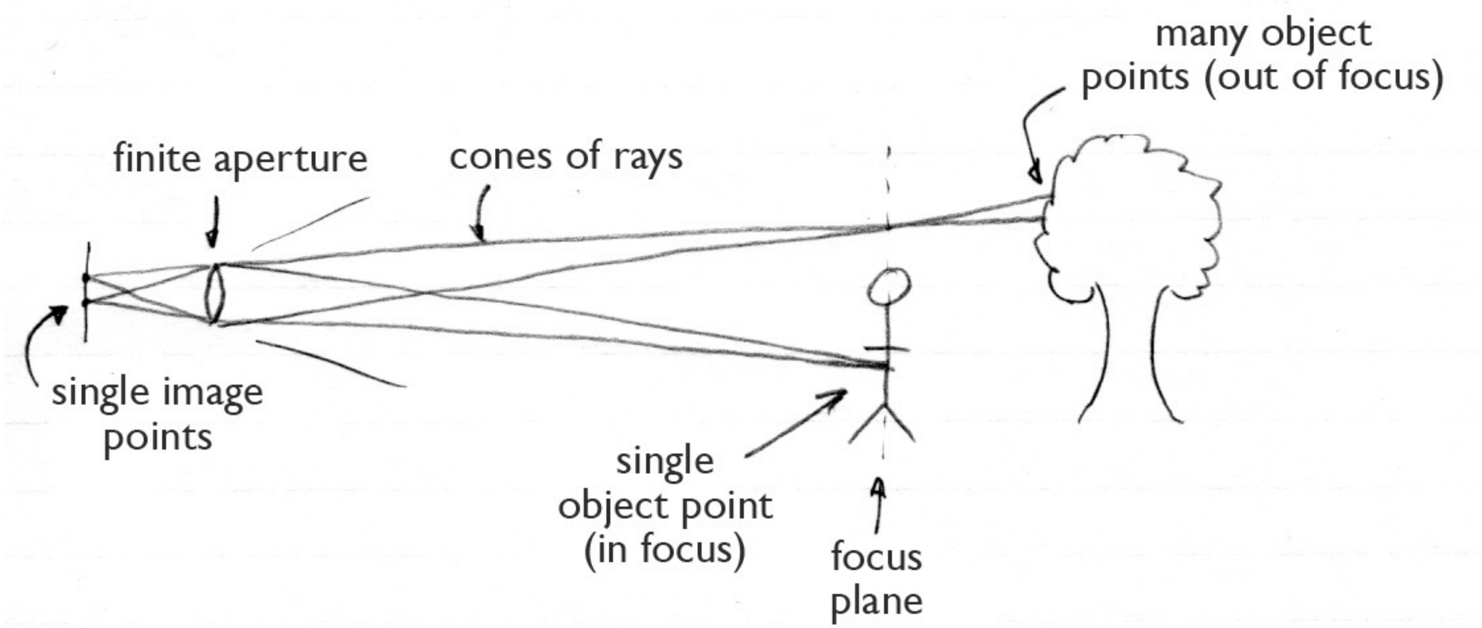
point aperture produces always-sharp focus

Cause of focusing effects



what lenses do (roughly)

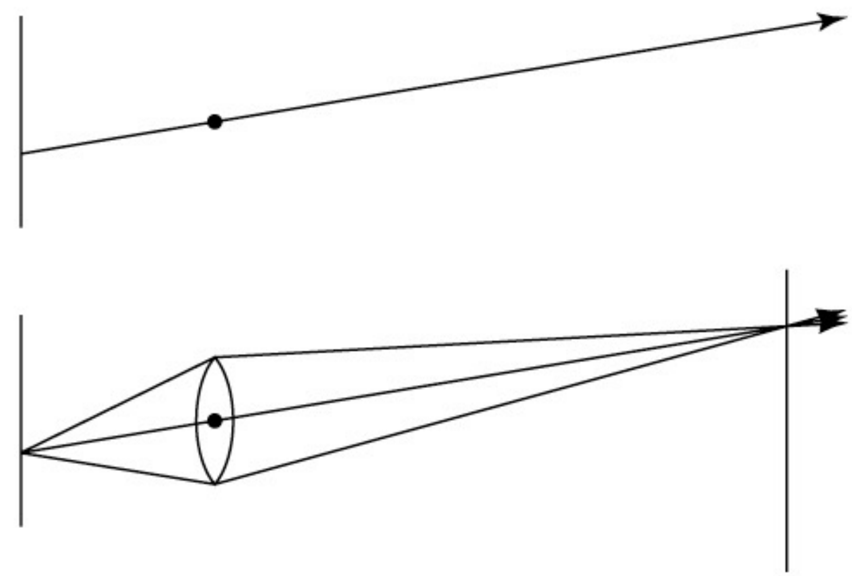
Cause of focusing effects



finite aperture produces limited depth of field

Depth of field

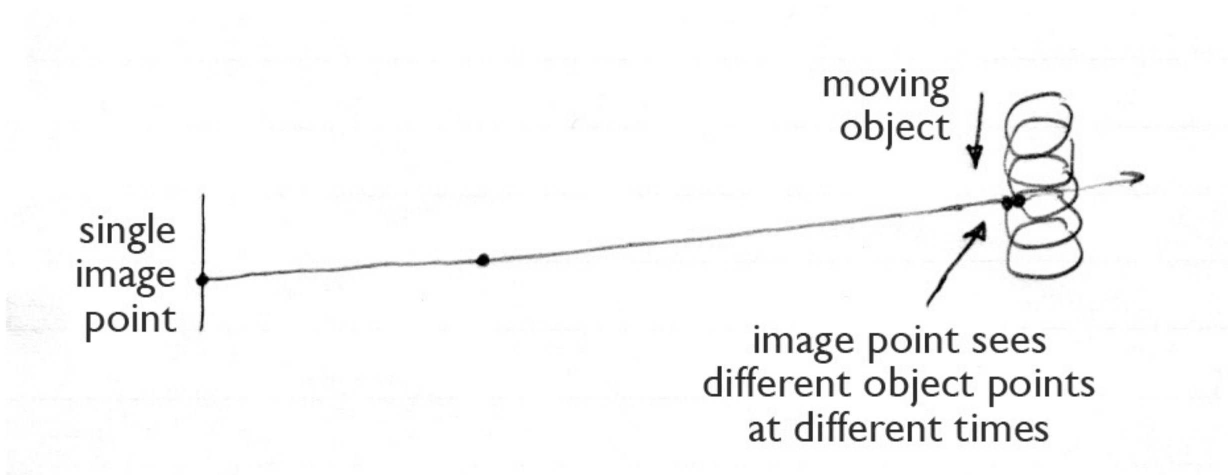
- Make eye rays start at random points on aperture
  - –always going toward a point on the focus plane



Motion blur

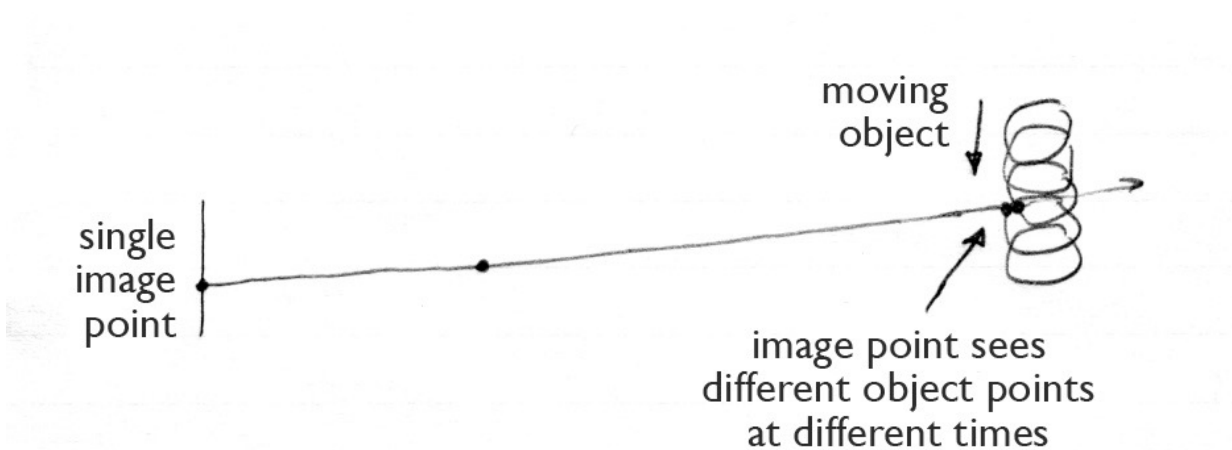
[Cook, Porter, Carpenter 1984]

Cause of motion blur



Motion blur

- Caused by finite shutter times
  –strobing without blur
- Introduce time as a variable throughout the system
  –object are hit by rays according to their position at a given time
- Then generate rays with times distributed over shutter interval



Pixar—*Monsters University* (2013)

Lightcuts

Direct only (relative cost 1x)

Direct+Indirect (1.3x)

Direct+Indirect+Fog (1.8x)

Direct+Indirect+Fog+Motion (2.2x)

Images and Displays

Representative display technologies

Direct-view displays

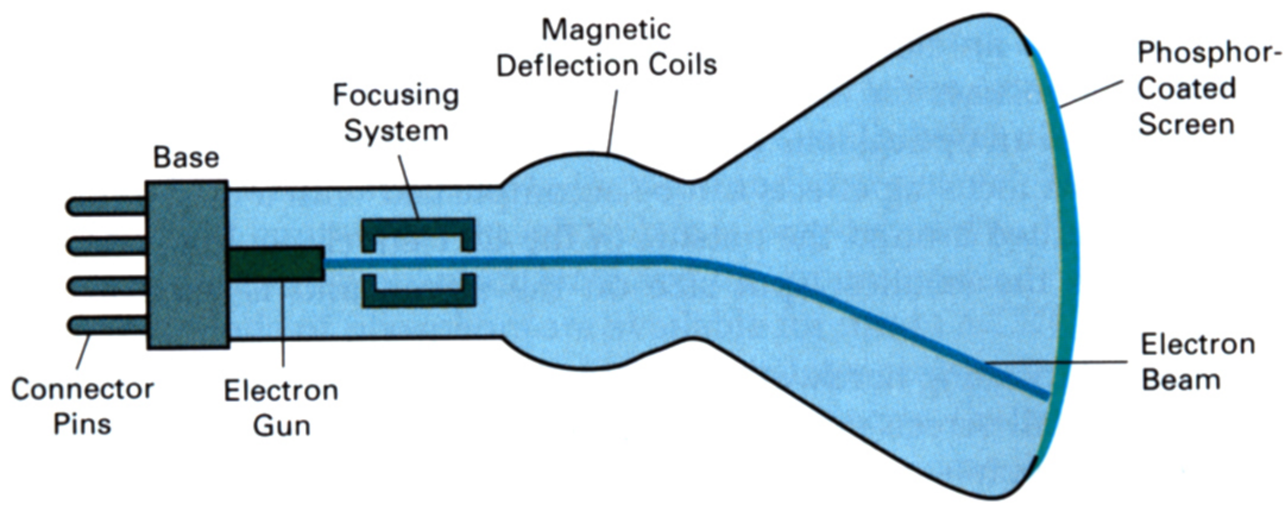
- Raster CRT display
- LCD display
- LED display

Printers

- Laser printer
- Inkjet printer

Cathode ray tube

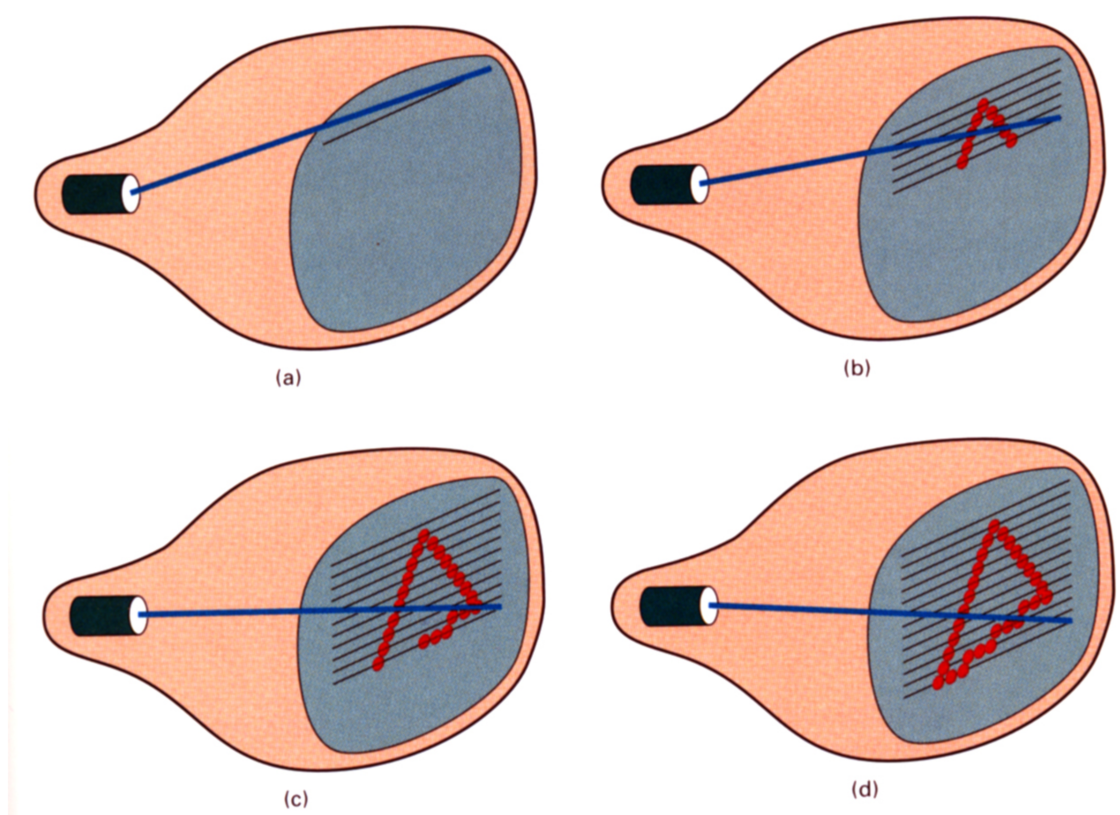
- First widely used electronic display
  - developed for TV in the 1920s–1930s



[H&B fig. 2-2]

Raster CRT display

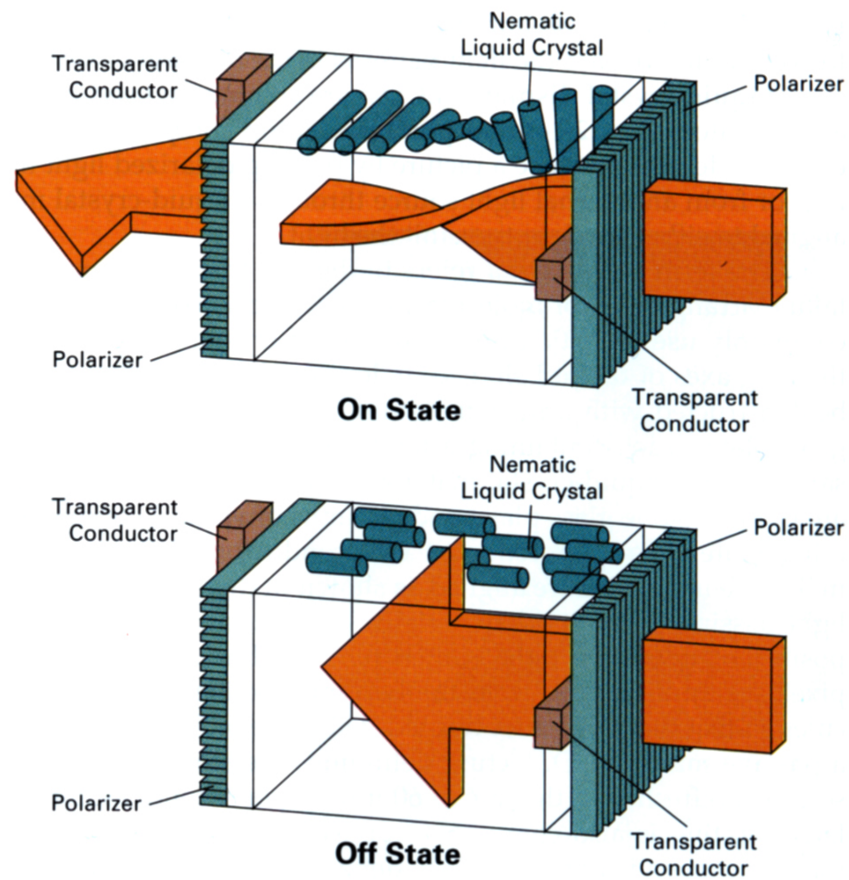
- Scan pattern fixed in display hardware
- Intensity modulated to produce image
- Originally for TV
  - (continuous analog signal)
- For computer, intensity determined by contents of *framebuffer*



[H&B fig. 2-7]

LCD flat panel display

- Principle: block or transmit light by twisting its polarization
- Illumination from backlight (either fluorescent or LED)
- Intermediate intensity levels possible by partial twist
- Fundamentally raster technology
- Fixed format



[H&B fig. 2-16]

LED Displays

[Wikimedia Commons]

[Google—Nexus 4]

