# Monte Carlo Ray Tracing

CS 4620 Lecture 22



# Basic ray tracing

- Many advanced methods build on the basic ray tracing paradigm
- Basic ray tracer: one sample for everything
  - one ray per pixel
  - one shadow ray for every point light
  - one reflection ray, possibly one refraction ray, per intersection

# Basic ray traced image

[Glassner 89]



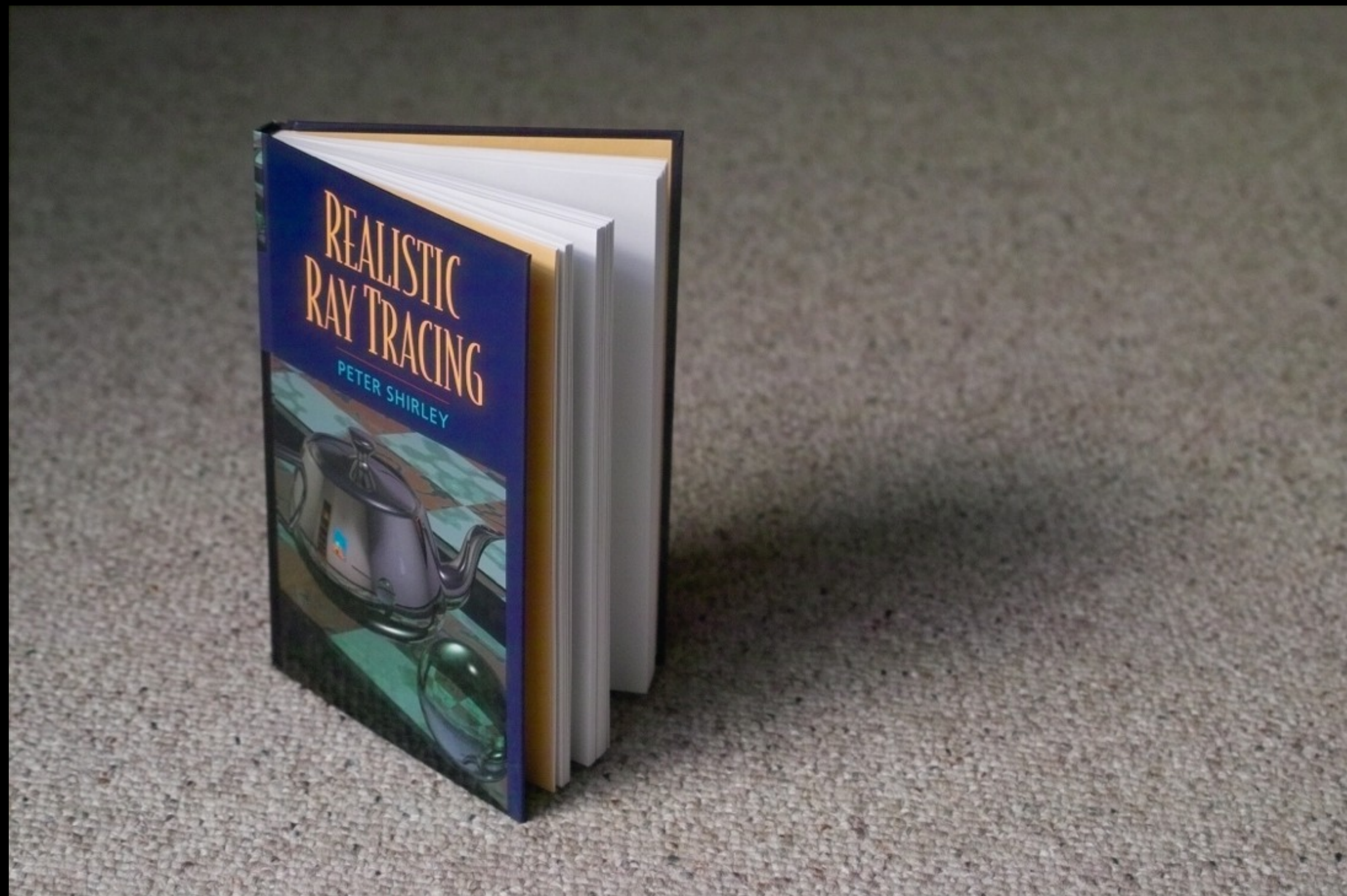
 

# Discontinuities in basic RT

- 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

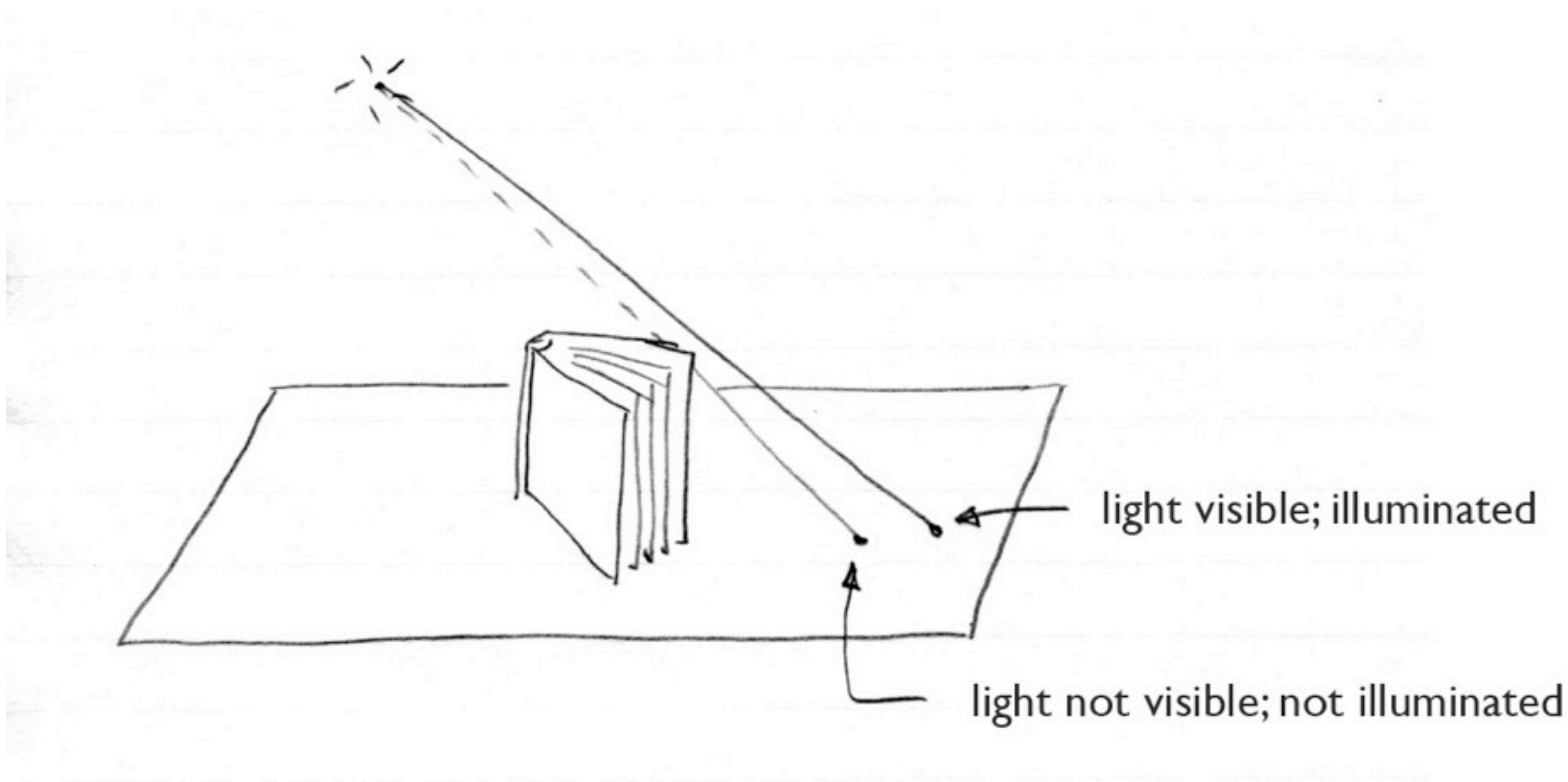
# Soft shadows





# Cause of soft shadows

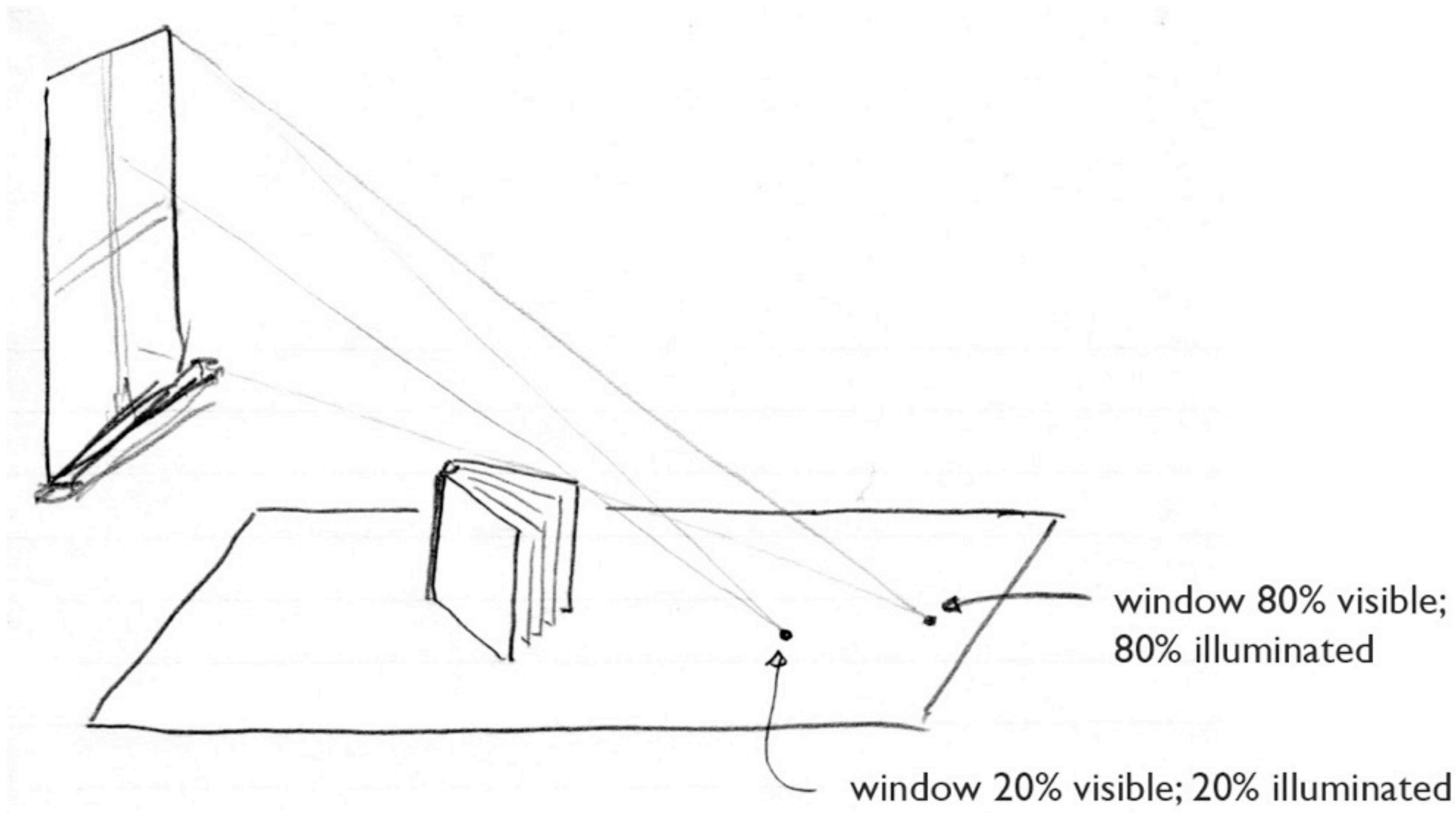



point lights cast hard shadows



# Cause of soft shadows




area lights cast soft shadows

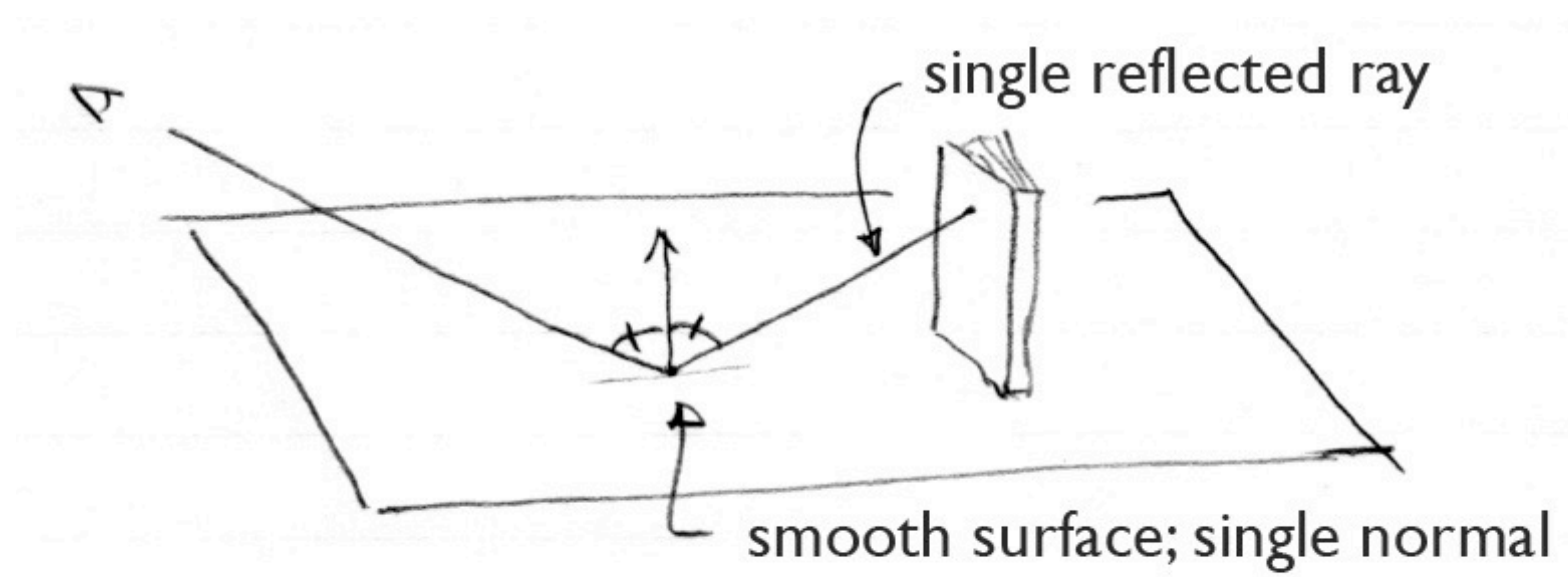
# Glossy reflection


[Lafortune et al. 97]



# Cause of glossy reflection

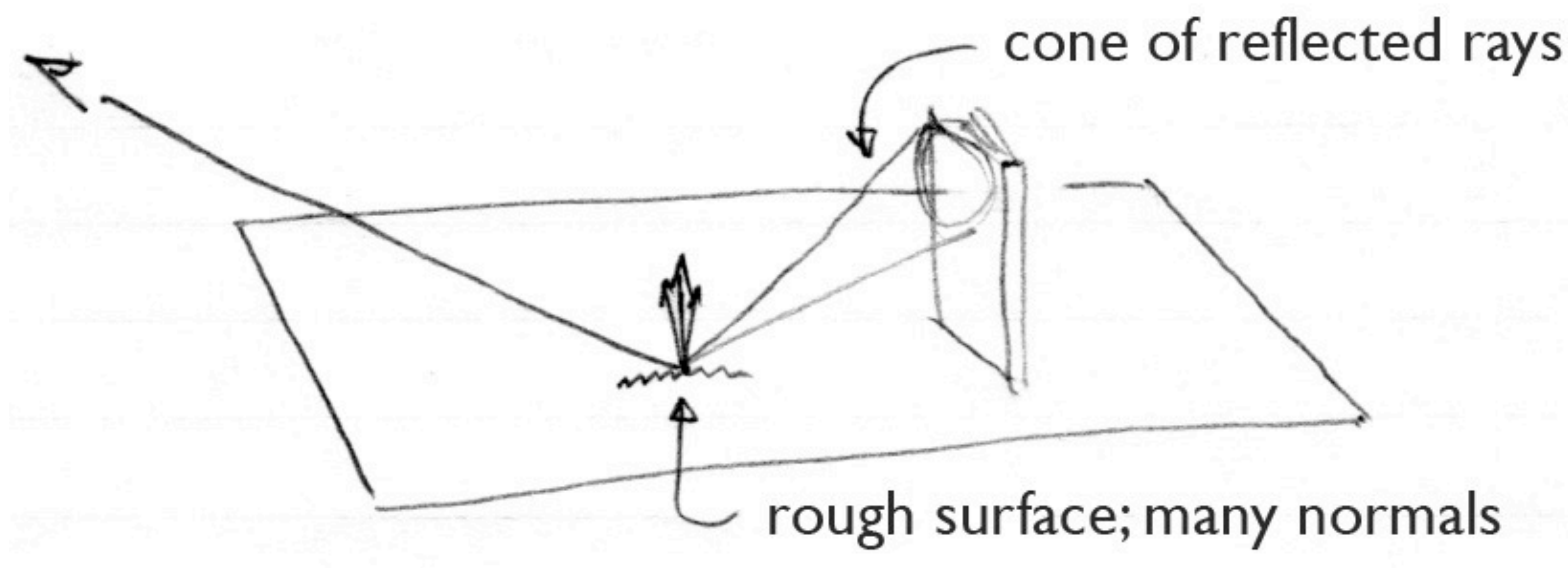



smooth surfaces produce sharp reflections



# Cause of glossy reflection

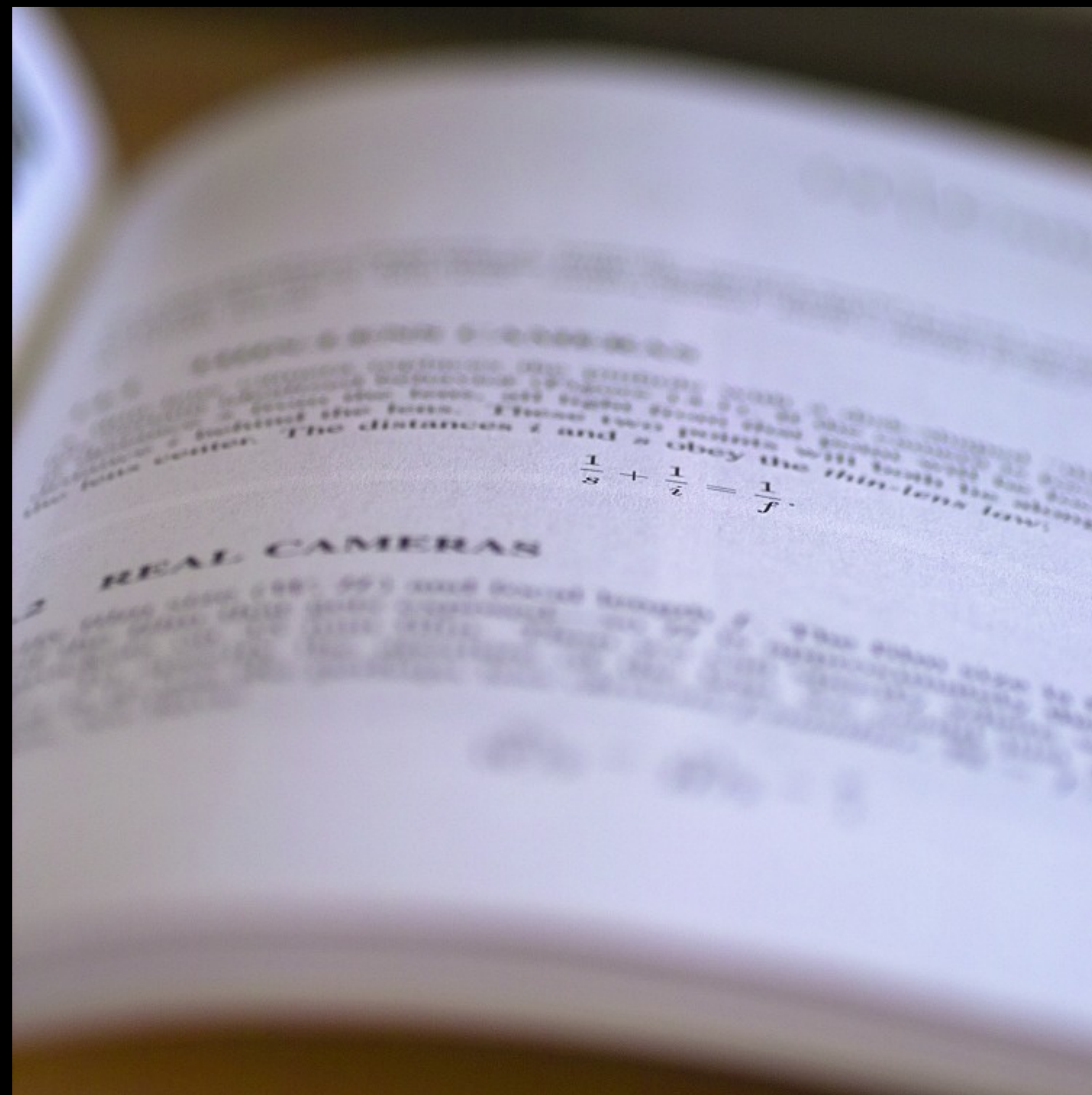



rough surfaces produce soft (glossy) reflections



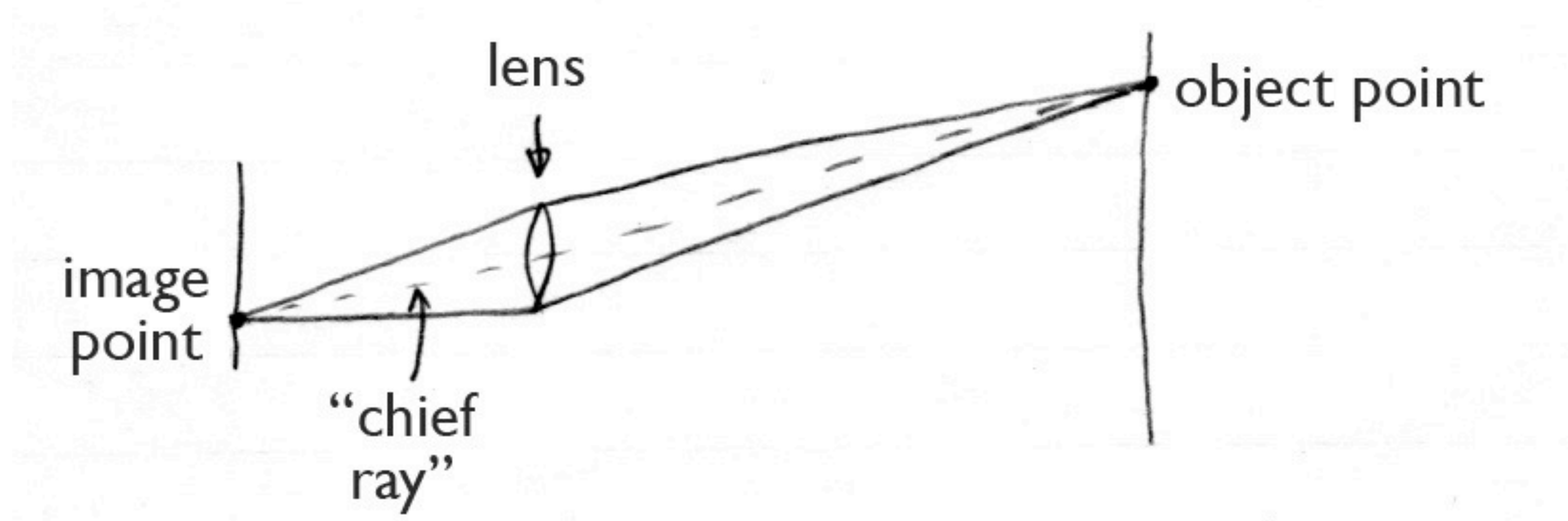
# Depth of field





# Cause of focusing effects

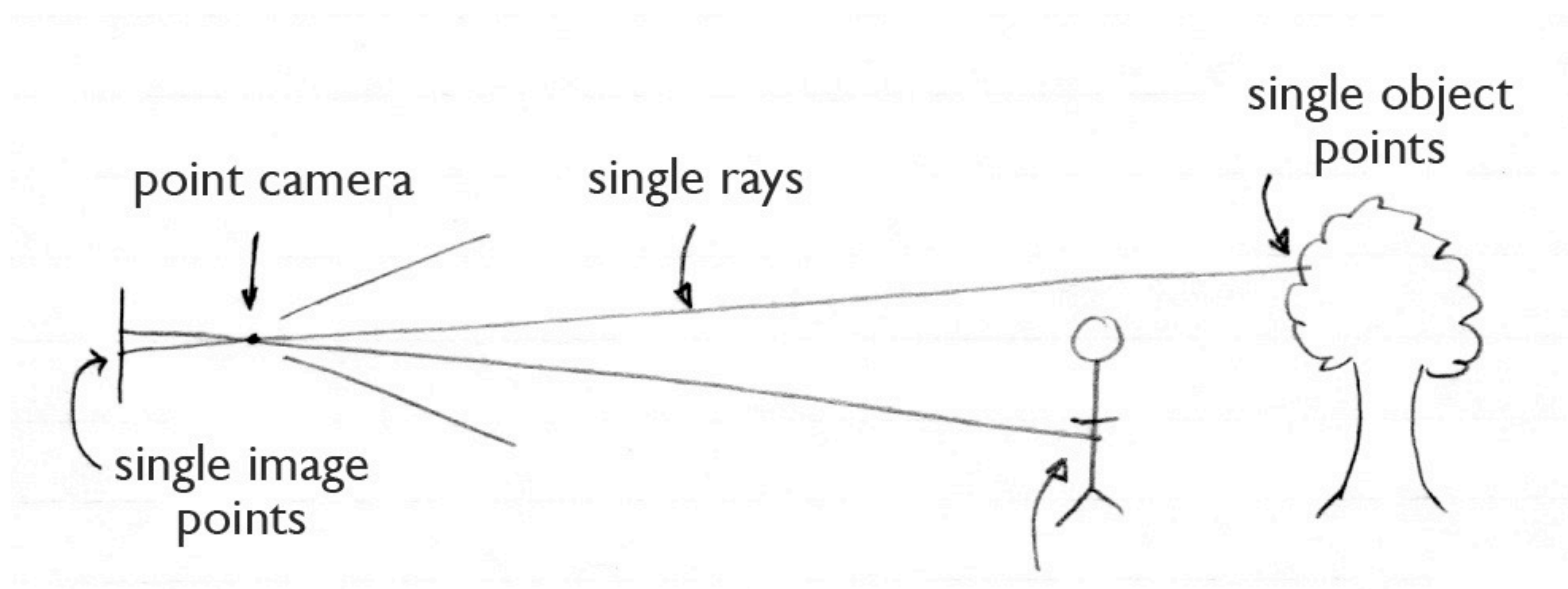



what lenses do (roughly)



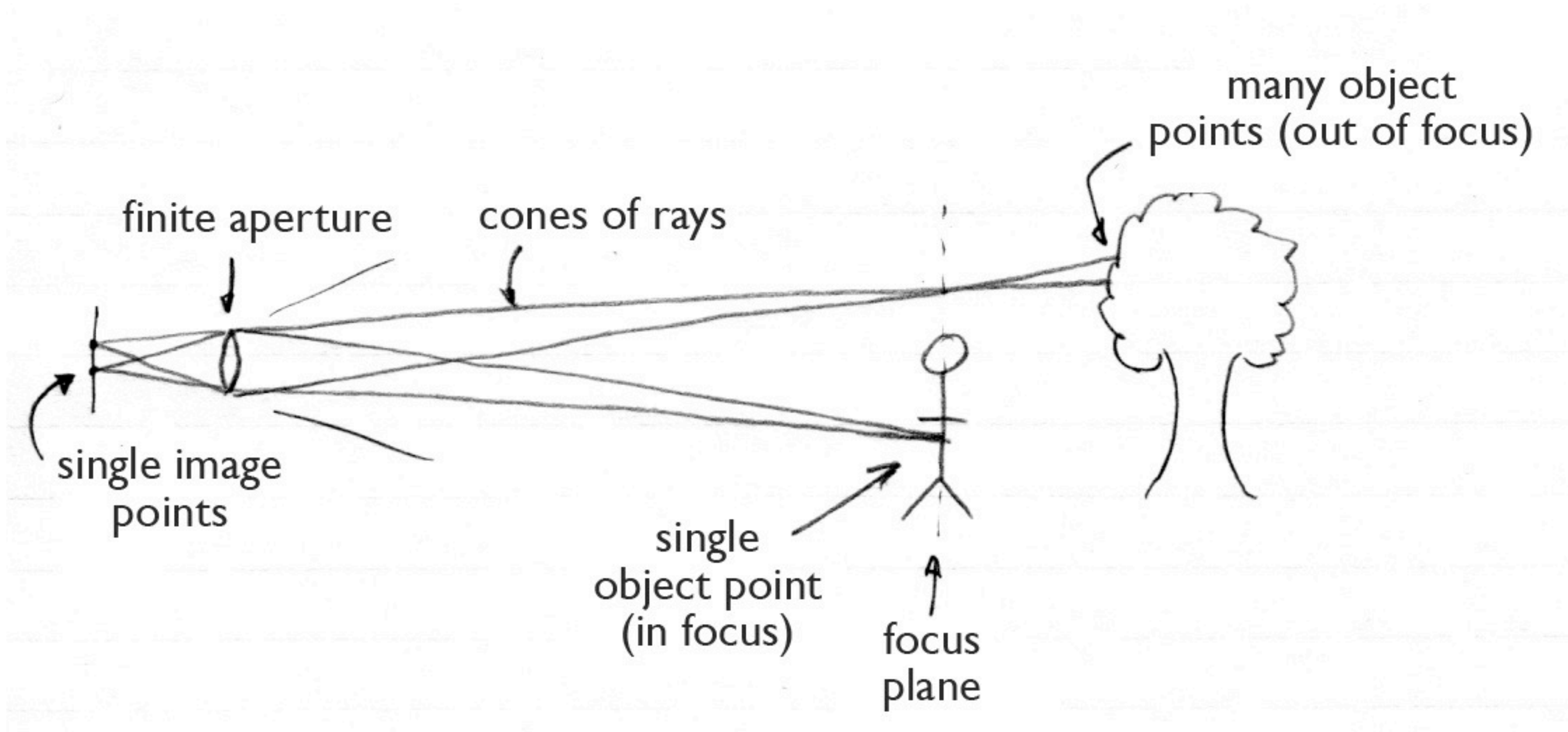
 

# Cause of focusing effects




point aperture produces always-sharp focus

# Cause of focusing effects




finite aperture produces limited depth of field

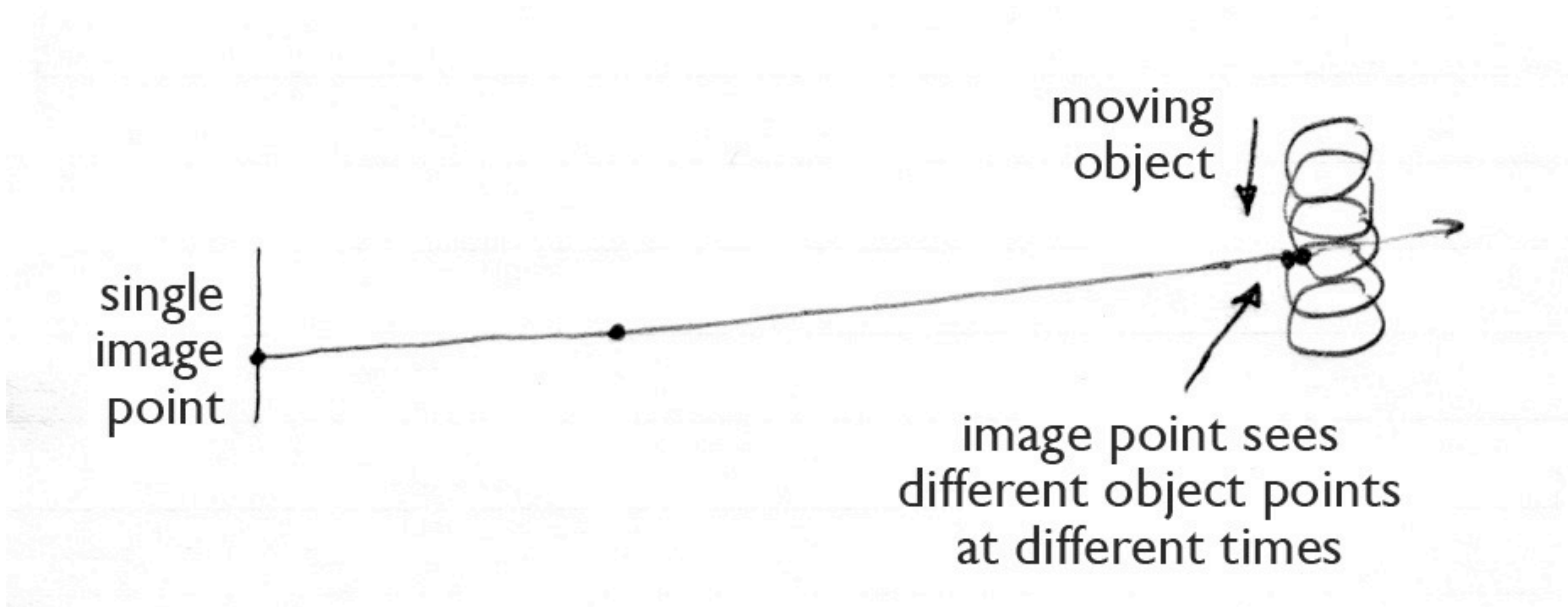


# Motion blur

[Cook, Porter, Carpenter 1984]

# Cause of motion blur



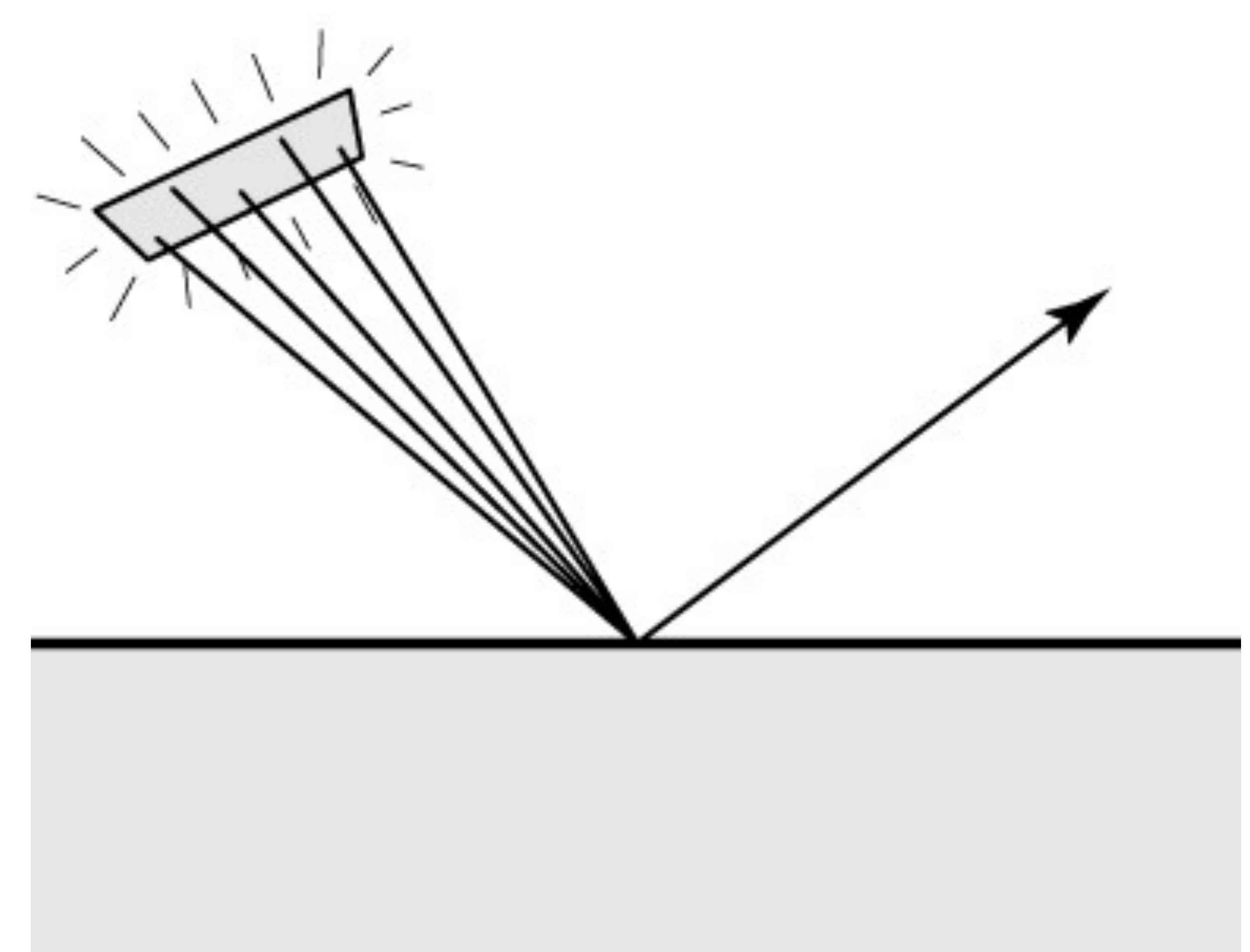


Pixar—*Monsters University* (2013)

# 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

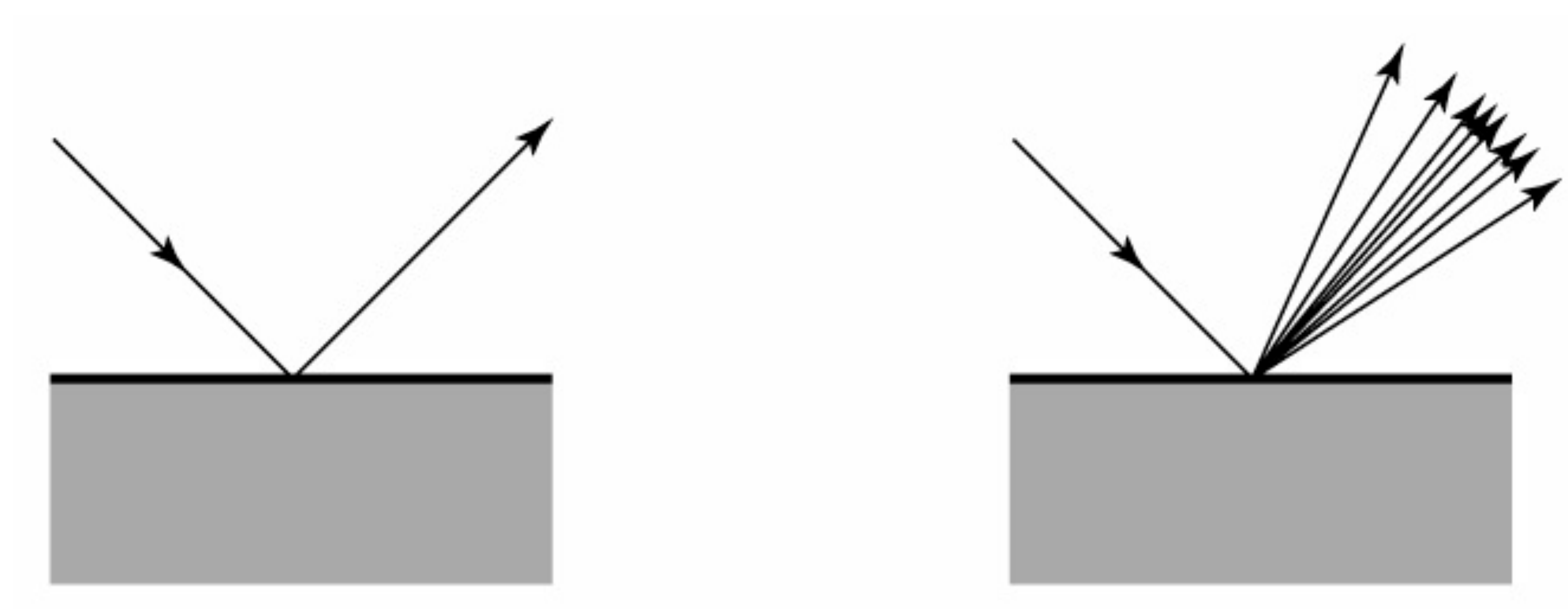




# Creating glossy reflections

- 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

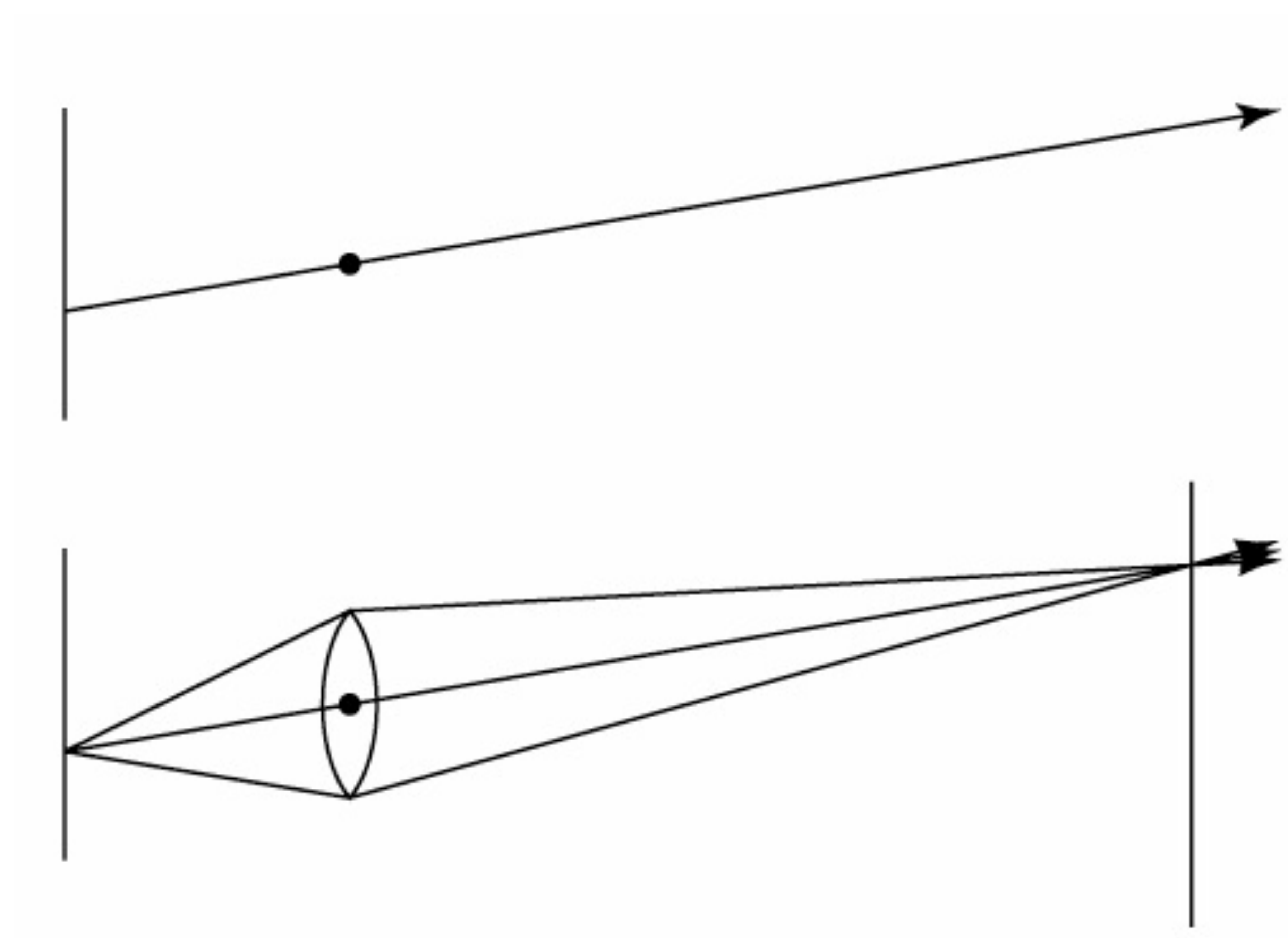




# Depth of field

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





# 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

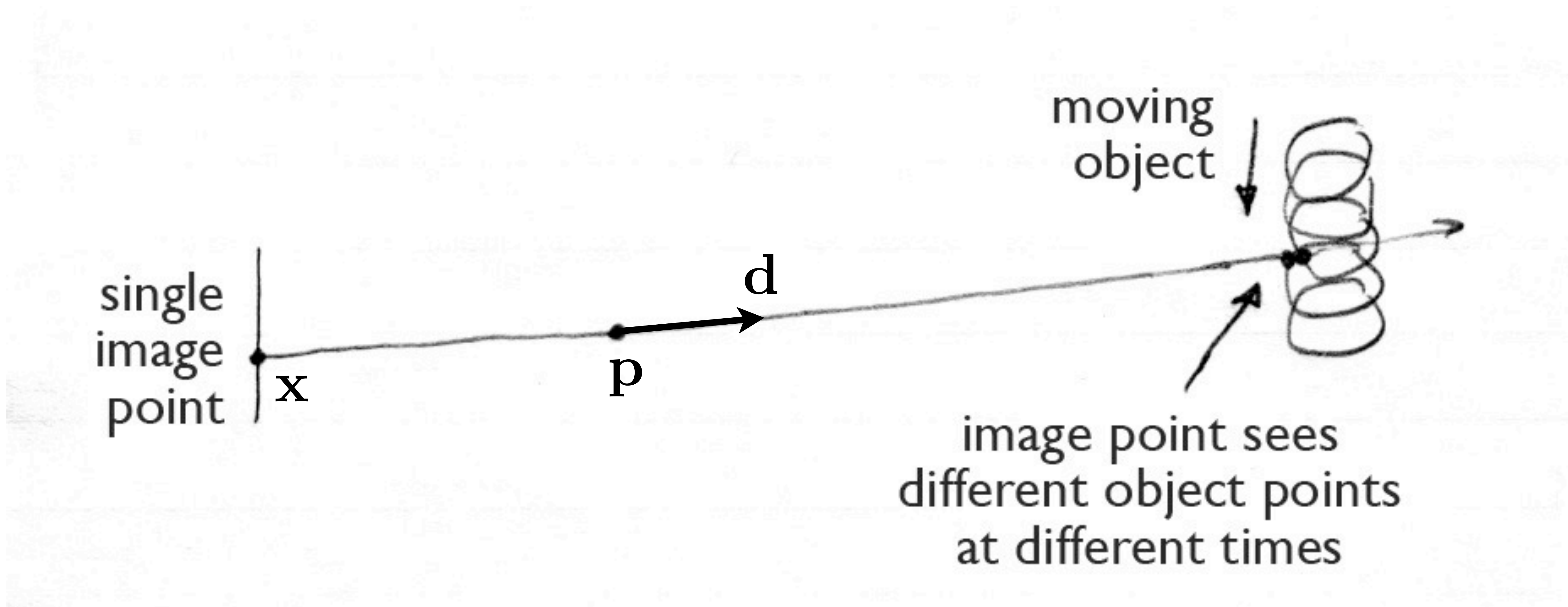


# But how, exactly?

- A key tool for getting all these effects accurately in a ray tracer is Monte Carlo integration
- Step 1: all these effects are actually integration problems
- Step 2: they can be solved using Monte Carlo integration

# Motion blur by integration




$$I(\mathbf{x}) = L(\mathbf{p}, \mathbf{d}(\mathbf{x}), t_0)$$

instantaneous light measurement

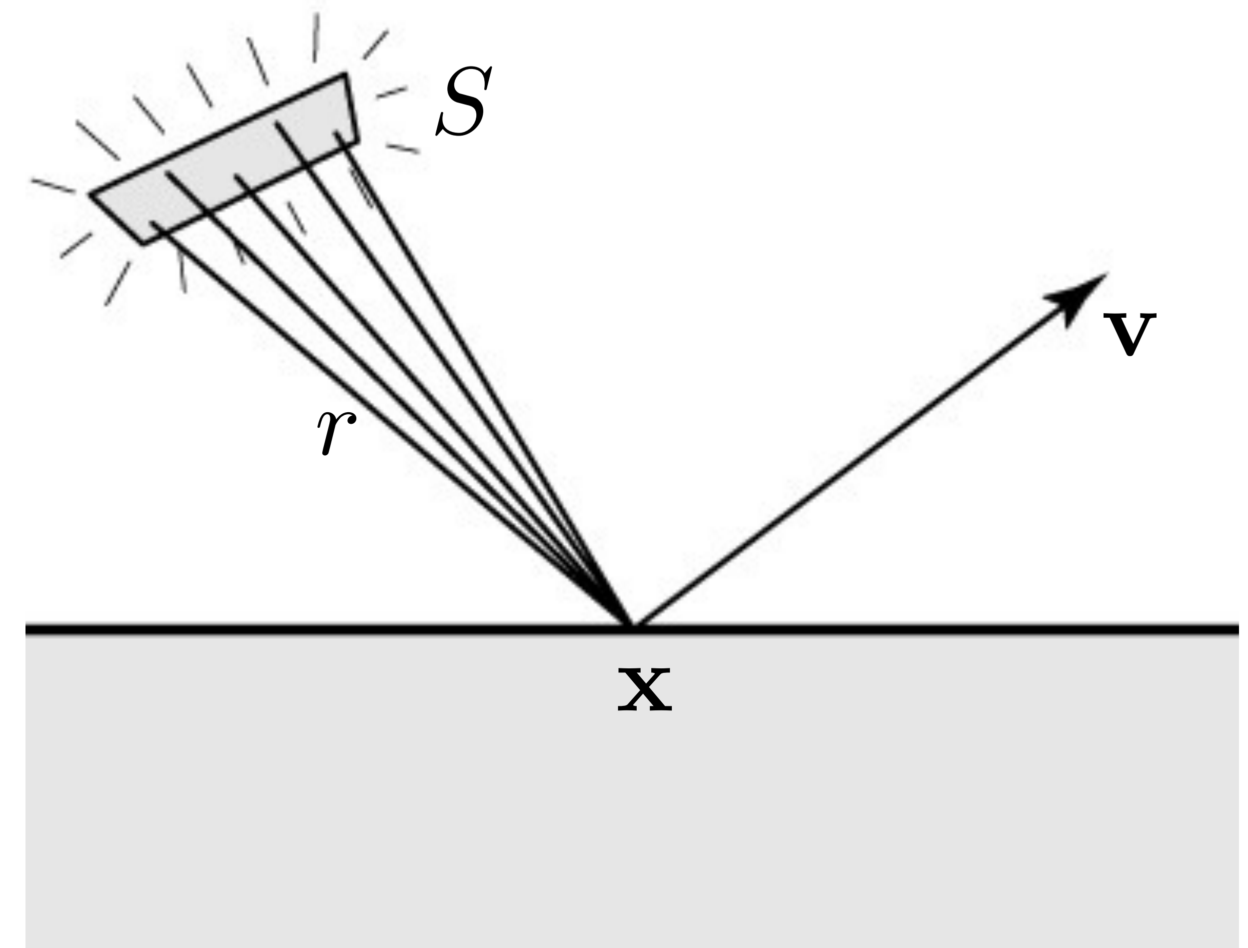
$$I(\mathbf{x}) = \frac{1}{|t_1 - t_0|} \int_{t_0}^{t_1} L(\mathbf{p}, \mathbf{d}(\mathbf{x}), t) dt$$

light averaged over shutter interval



# Soft shadows by integration




$$L(\mathbf{x}, \mathbf{v}) = \frac{I \ \cos\theta}{r^2}$$

illumination from single point

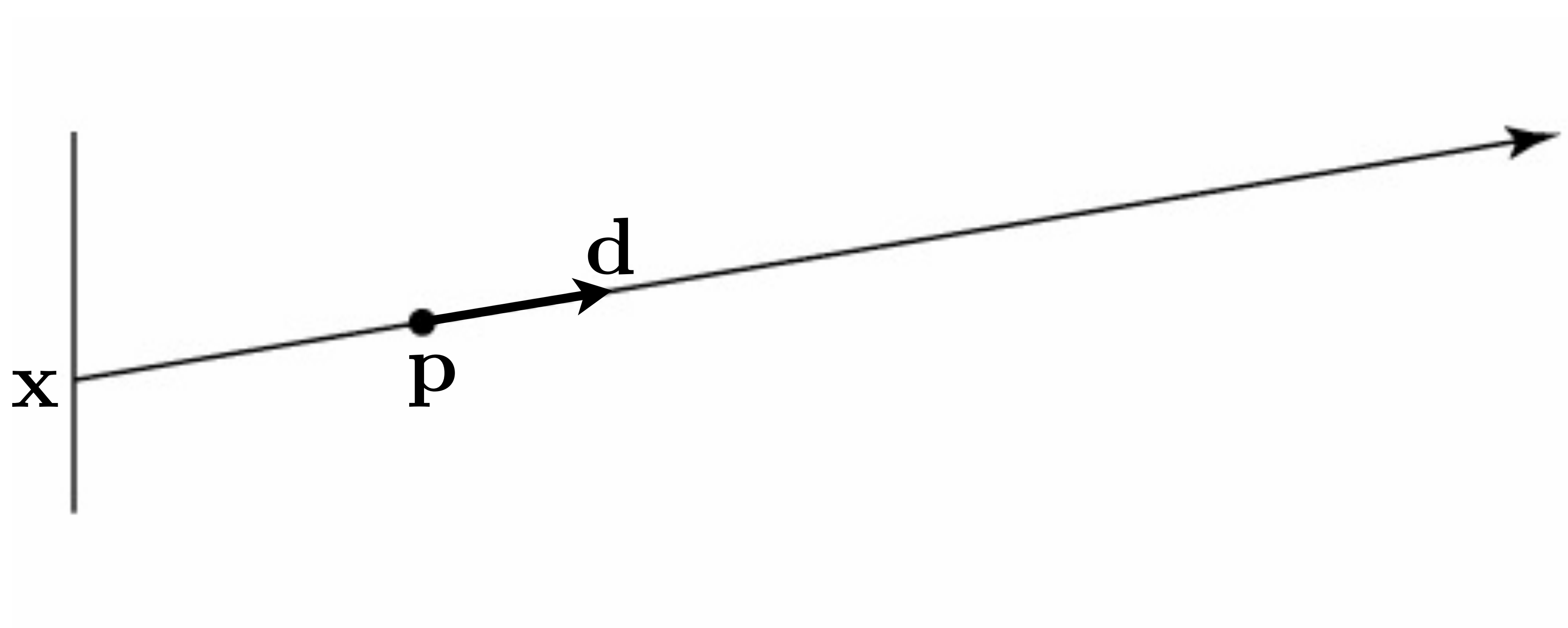
$$L(\mathbf{x}, \mathbf{v}) = \frac{1}{|S|} \int_S \frac{I \ \cos\theta}{r^2} dA$$

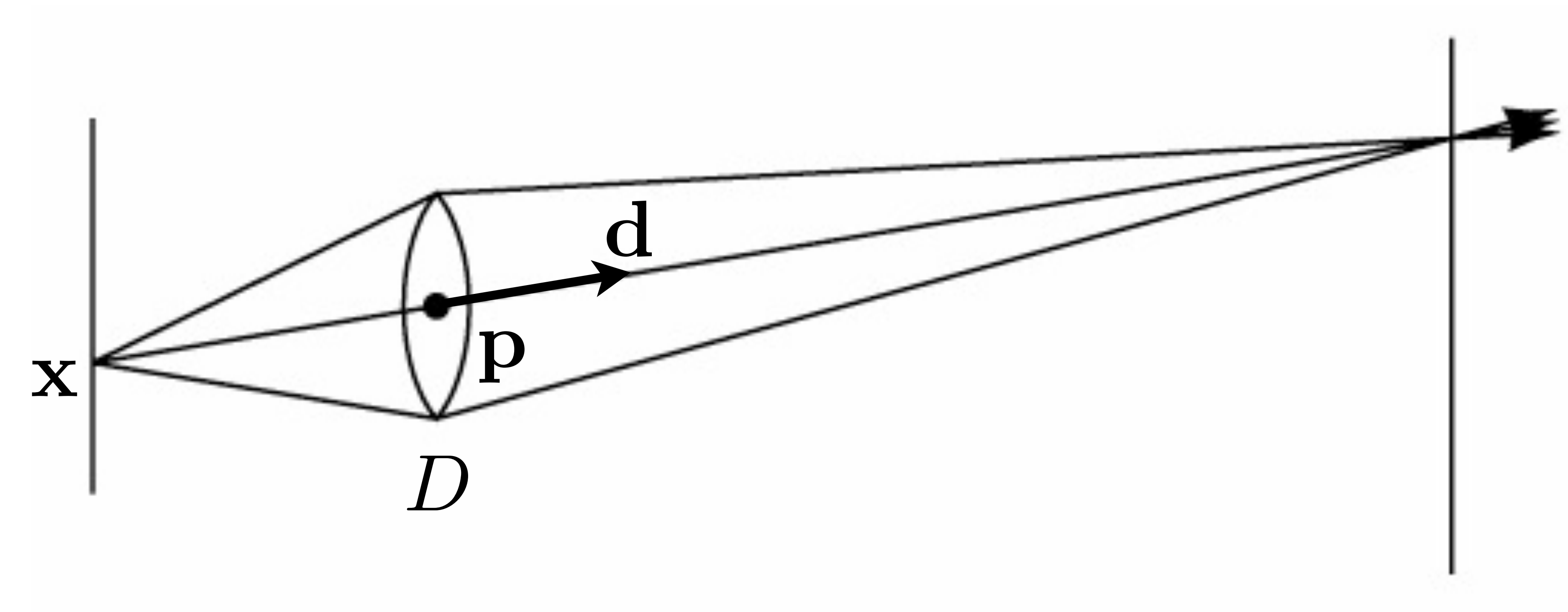
illumination averaged over light source area



(for those counting: I have hidden a cosine factor in $I$, don't worry about it today)

# Depth of field by integration




$$I(\mathbf{x}) = L(\mathbf{p}, \mathbf{d}(\mathbf{x}))$$

light along
a single ray

$$I(\mathbf{x}) = \frac{1}{|D|} \int_D L(\mathbf{p}, \mathbf{d}(\mathbf{x}, \mathbf{p})) dA(\mathbf{p})$$

light averaged over
rays through aperture



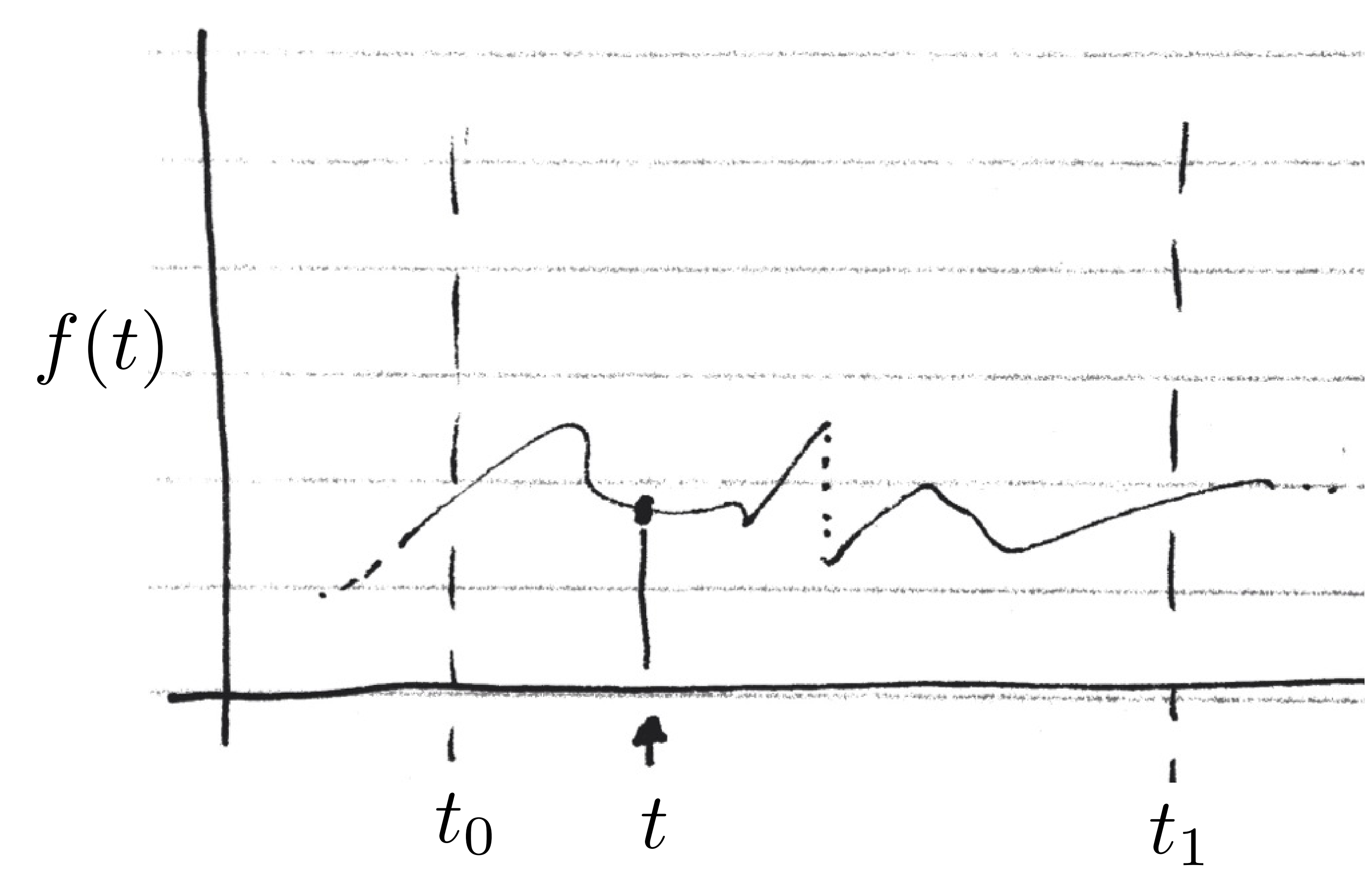
 

# Monte Carlo integration

- How to integrate a function we don't know much about?
  - we can only evaluate it by tracing rays
  - reasoning about how light changes from one ray to the next is tricky
- Idea: evaluate at a random place and call use that *sample* to make an *estimate* of the average value (and thereby integral) of the function

Knowing only the value at t
and the size of the interval
my best estimate of the integral
is:

$$g(t) = f(t)|t_1 - t_0|$$





# Monte Carlo integration

- If I do this many times, what is the expected value?
  - when there are finitely many possibilities, outcome *k* will happen in a fraction *p*(*k*) of trials, hence

$$E\{g[k]\} = \sum_k g[k]p[k]$$

here *p*(*k*) is the *probability* of outcome *k*

  - in our continuous case this becomes an integral

$$E\{g(t)\} = \int_{t_0}^{t_1} g(t)p(t)dt$$

here *p*(*t*) is a *probability density* for outcomes around *t*

  - for the estimator on the previous slide

$$E\{g(t)\} = E\{f(t)|t_1 - t_0|\} = \int_{t_0}^{t_1} f(t)|t_1 - t_0|\, p(t)\, dt$$

but $p(t) = \dfrac{1}{|t_1 - t_0|}$

$$= \int_{t_0}^{t_1} f(t)\, dt$$



# Monte Carlo integration

- In general Monte Carlo integration works like this
  - choose **x** randomly in some domain *D* with some probability density *p*(**x**)
  - evaluate *f*(**x**) and form the estimator

$$g(\mathbf{x}) = \frac{f(\mathbf{x})}{p(\mathbf{x})}$$

  - the expected value of g(x) will then be

$$E\{g(\mathbf{x})\} = \int_D f(\mathbf{x})\, d\mathbf{x}$$

- Get better and better approximations to that expected value by averaging together a lot of independent samples



# Monte Carlo in rendering

- Motion blur: select random *t* in the shutter interval

$$g(t) = L(\mathbf{p}, \mathbf{d}(\mathbf{x}), t)\,|t_1 - t_0|$$

- Depth of field: select random **p** uniformly over the aperture *D*

$$g(\mathbf{p}) = L(\mathbf{p}, \mathbf{d}(\mathbf{x}, \mathbf{p}))\,|D|$$

- Area light: select source point **y** uniformly over the light source *S*

$$g(\mathbf{y}) = \frac{I \cos\theta(\mathbf{y})}{r(\mathbf{y})^2}\,|S|$$



# Monte Carlo for surface reflection

- Key integral to be evaluated is

$$L(\mathbf{x}, \mathbf{v}) = \int_{H^2} f_r(\mathbf{v}, \mathbf{w}) L_i(\mathbf{x}, \mathbf{w}) (\mathbf{w} \cdot \mathbf{n}) \, d\mathbf{w}$$

  – and a common approach is to sample with

$$p(\mathbf{w}) \propto f_r(\mathbf{v}, \mathbf{w})$$

# The Blue Umbrella

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

worth a look:
**http://rainycitytales332.tumblr.com**

