Basic Ray-Tracing Ideas -II

D.A. Forsyth, UIUC

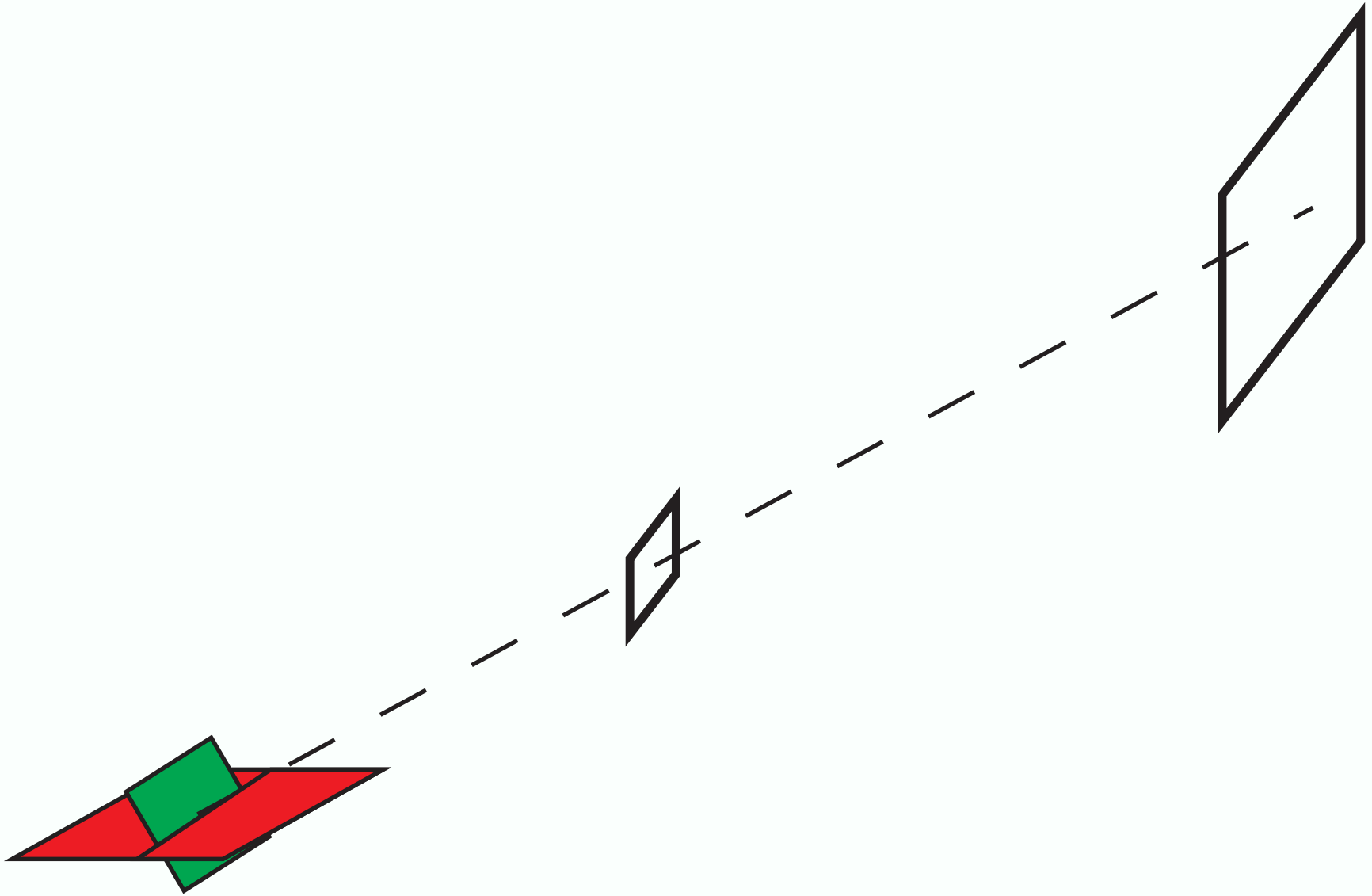
Importance weighting

if $x_i \sim p(x)$

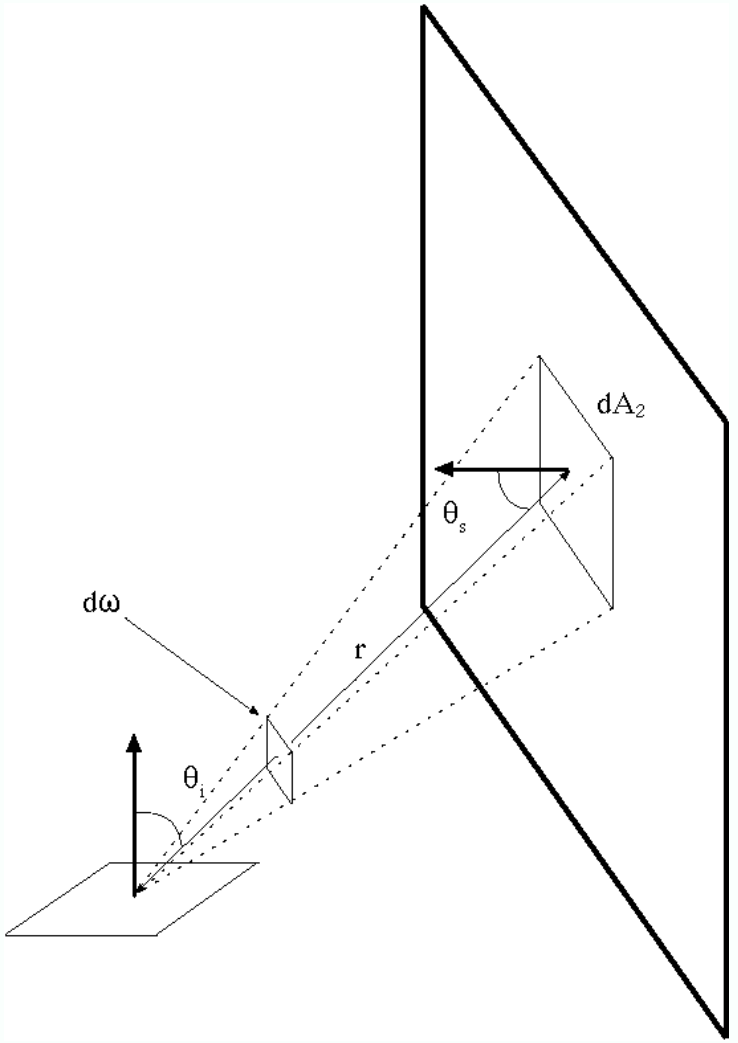
then $\frac{1}{N}\sum_{i=1}^{N} f(x_i) \rightarrow \int f(x)p(x)dx$

If $x_i \sim p(x)$

Then $$\int f(x)dx \approx \frac{1}{N}\sum_i \frac{f(x_i)}{p(x_i)}$$



Area sources

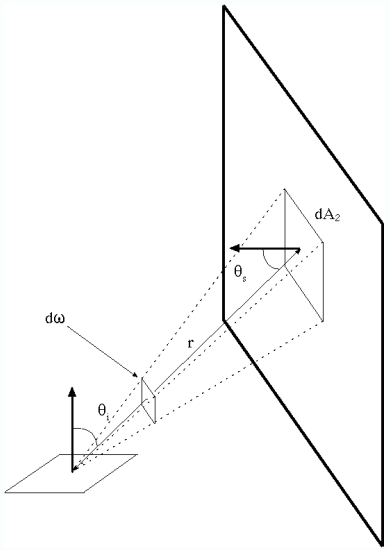


- Examples: diffuser boxes, white walls.
- The radiosity at a point due to an area source is obtained by adding up the contribution over the section of view hemisphere subtended by the source
  - change variables and add up over the source

Radiosity due to an area source

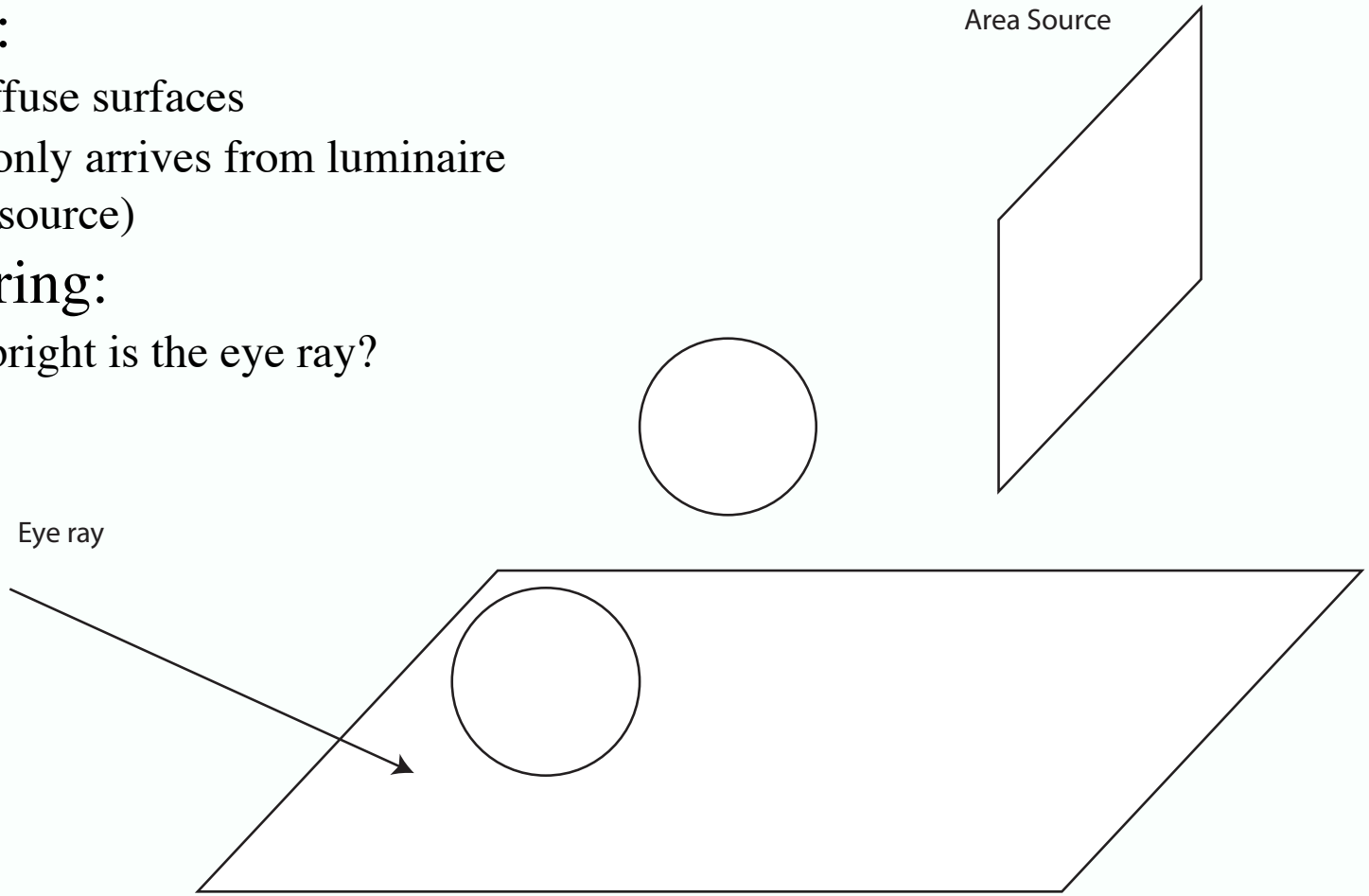
- rho is albedo
- E is exitance
- r(x, u) is distance between points
- u is a coordinate on the source

$$
\begin{aligned}
B(x) &= \rho_d(x)\int_{\Omega} L_i(x, u \rightarrow x)\cos\theta_i d\omega \\
&= \rho_d(x)\int_{\Omega} L_e(x, u \rightarrow x)\cos\theta_i d\omega \\
&= \rho_d(x)\int_{\Omega}\left(\frac{E(u)}{\pi}\right)\cos\theta_i d\omega \\
&= \rho_d(x)\int_{source}\left(\frac{E(u)}{\pi}\right)\cos\theta_i\left(\cos\theta_s\frac{dA_u}{r(x,u)^2}\right) \\
&= \rho_d(x)\int_{source} E(u)\frac{\cos\theta_i\cos\theta_s}{\pi r(x,u)^2}dA_u
\end{aligned}
$$

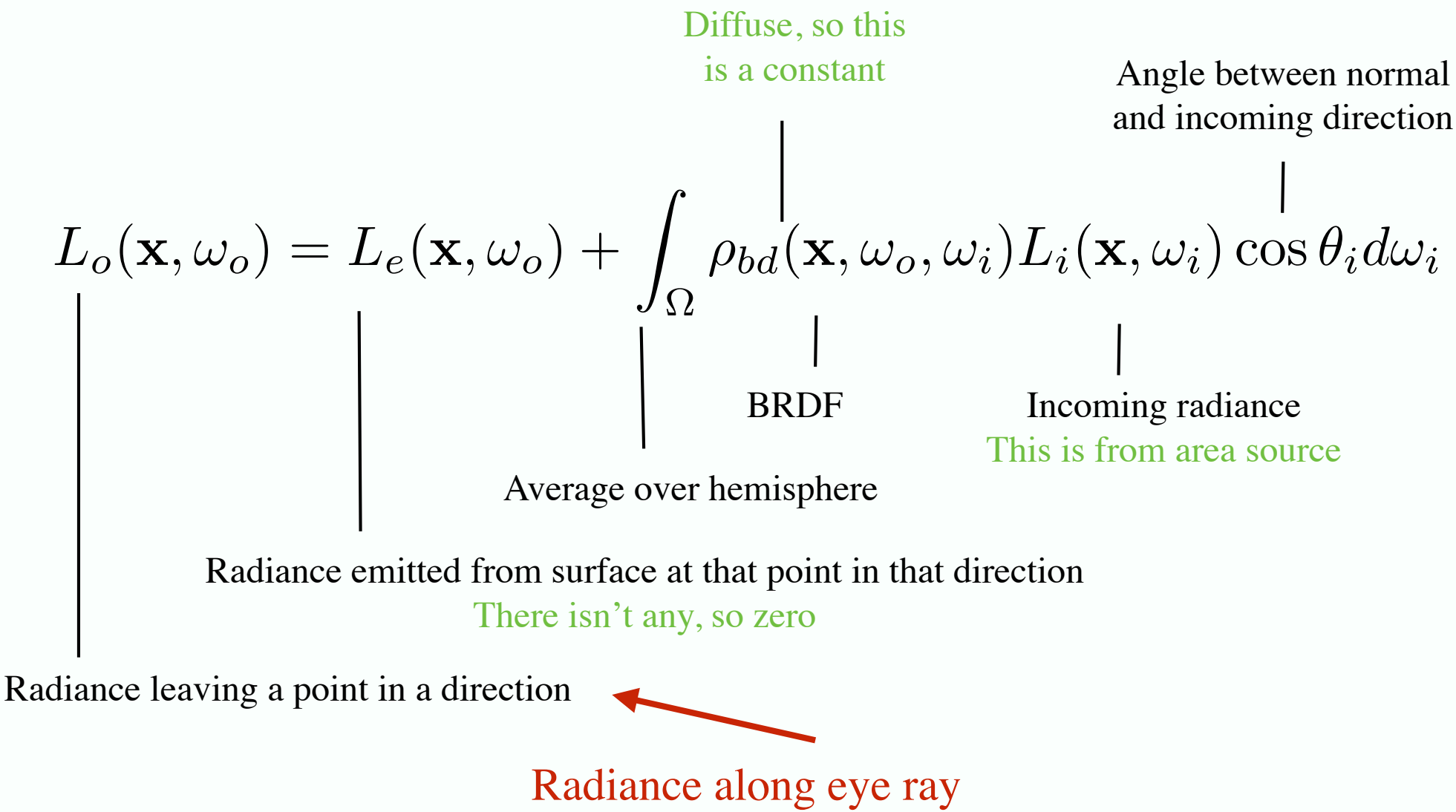


Question: how to ray-trace this?

- Model:
  - all diffuse surfaces
  - light only arrives from luminaire (area source)
- Rendering:
  - how bright is the eye ray?



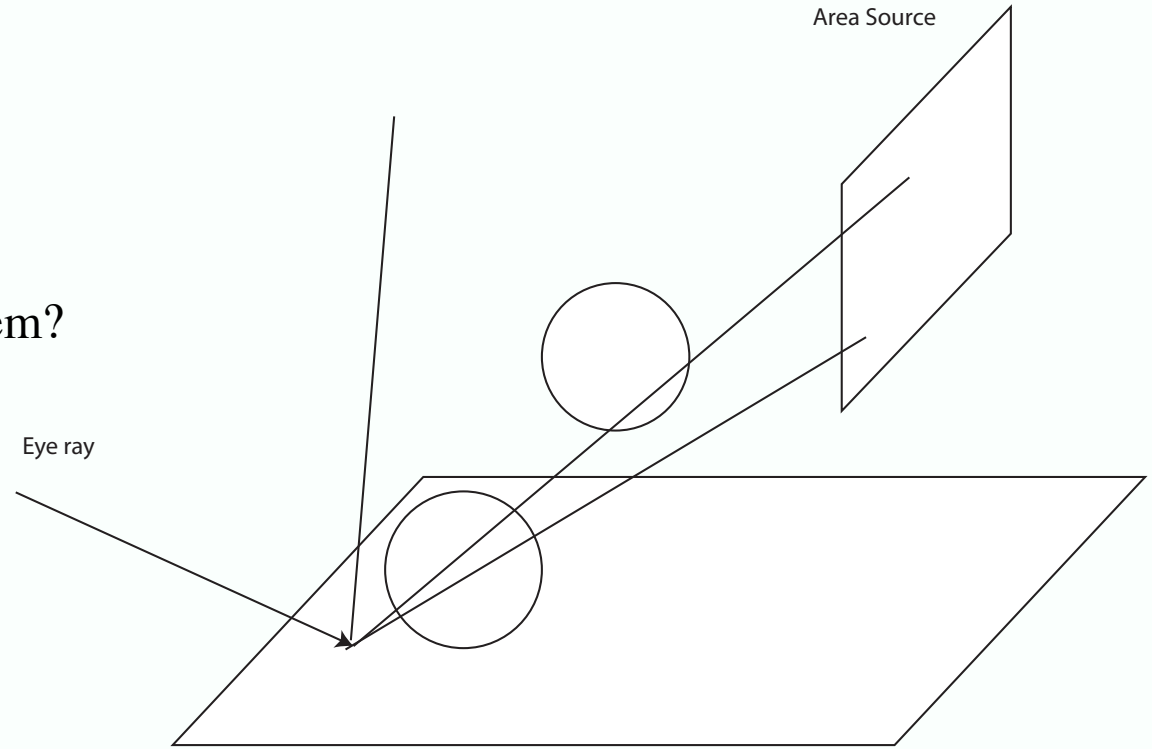
Recall


Diffuse, so this is a constant
Angle between normal and incoming direction
$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_\Omega \rho_{bd}(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i) \cos\theta_i d\omega_i$
BRDF
Incoming radiance
This is from area source
Average over hemisphere
Radiance emitted from surface at that point in that direction
There isn't any, so zero
Radiance leaving a point in a direction
Radiance along eye ray

$$L_o(\mathbf{x}, \omega_o) = \cancelto{0}{L_e(\mathbf{x}, \omega_o)} + \int_\Omega \rho_{bd}(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i) \cos\theta_i d\omega_i$$

$$\frac{1}{N} \sum_{\omega_i \in \text{samples of incoming directions}} \rho L(\mathbf{x}, \omega_i) \cos\theta_i$$

But which directions?
and how should we sample them?

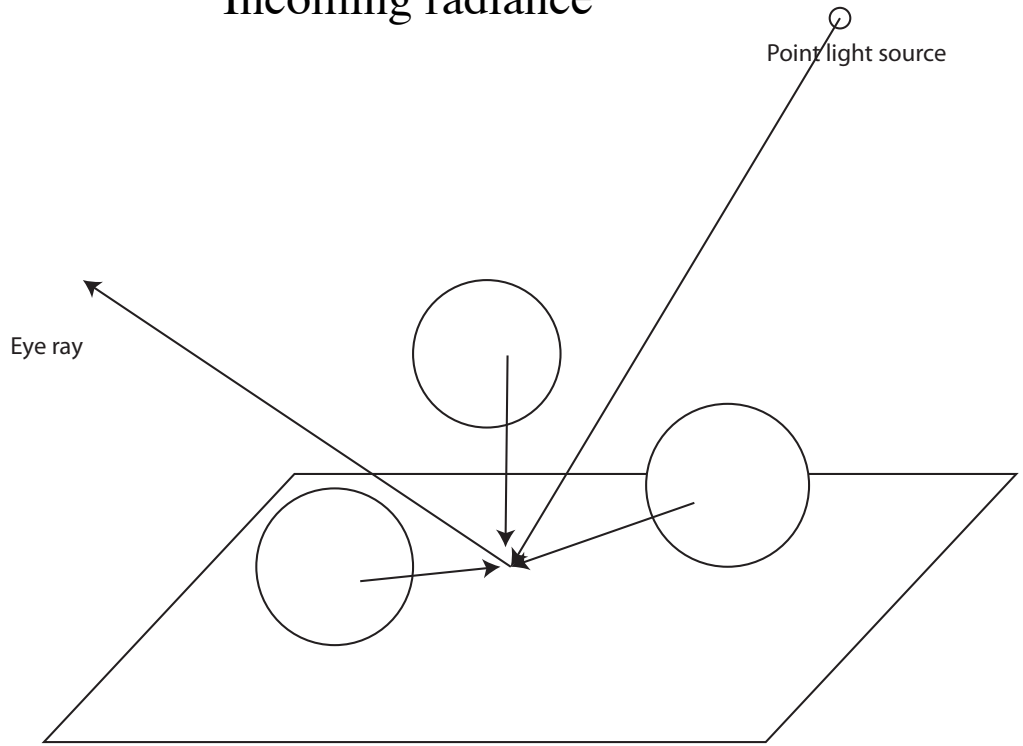


Global illumination

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_\Omega \rho_{bd}(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i) \cos\theta_i d\omega_i$$

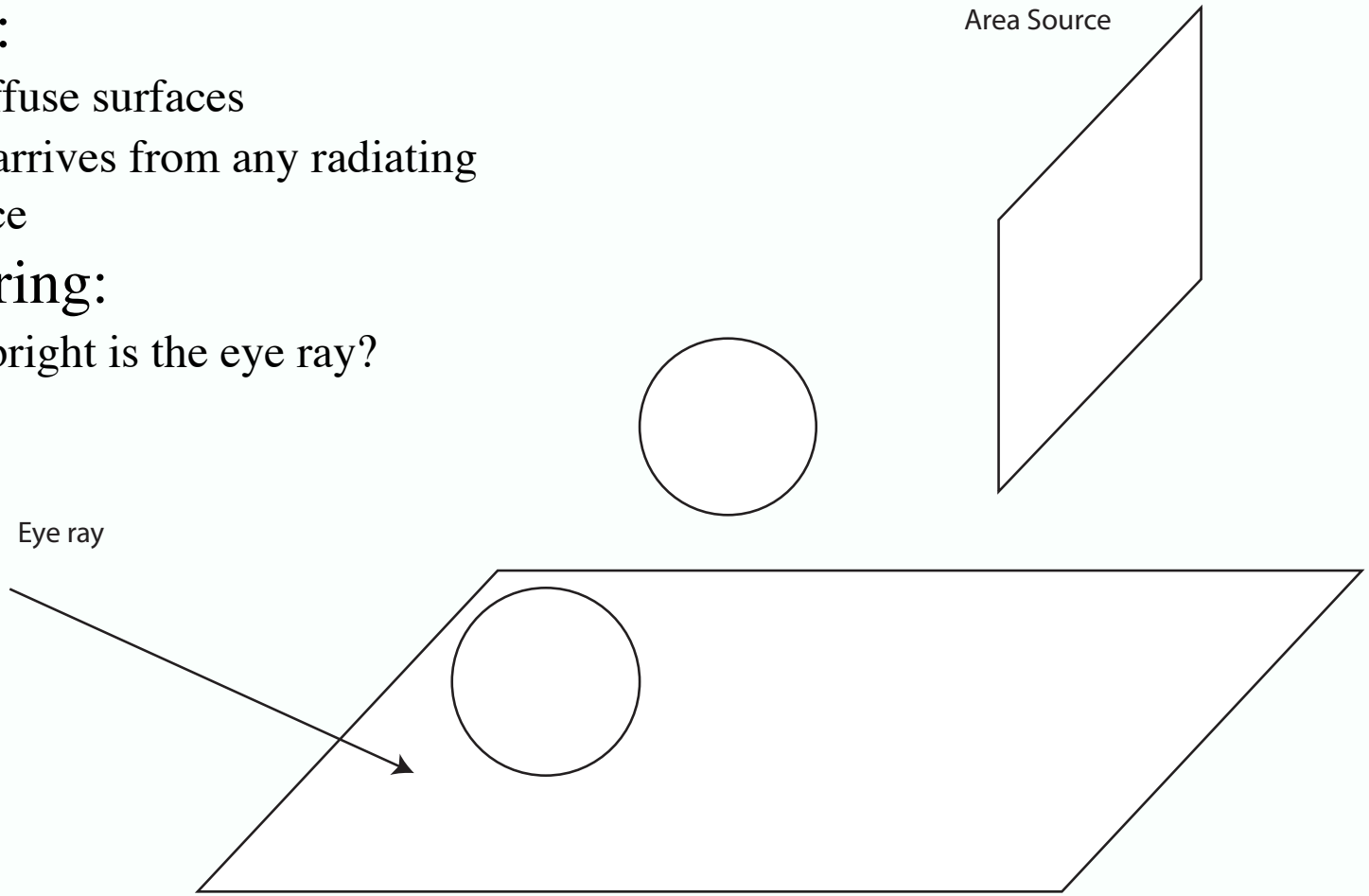
Incoming radiance

- Incoming radiance isn't just from luminaires
  - the reason you can see surfaces is they reflect light
  - other surfaces don't distinguish between reflected light and generated light

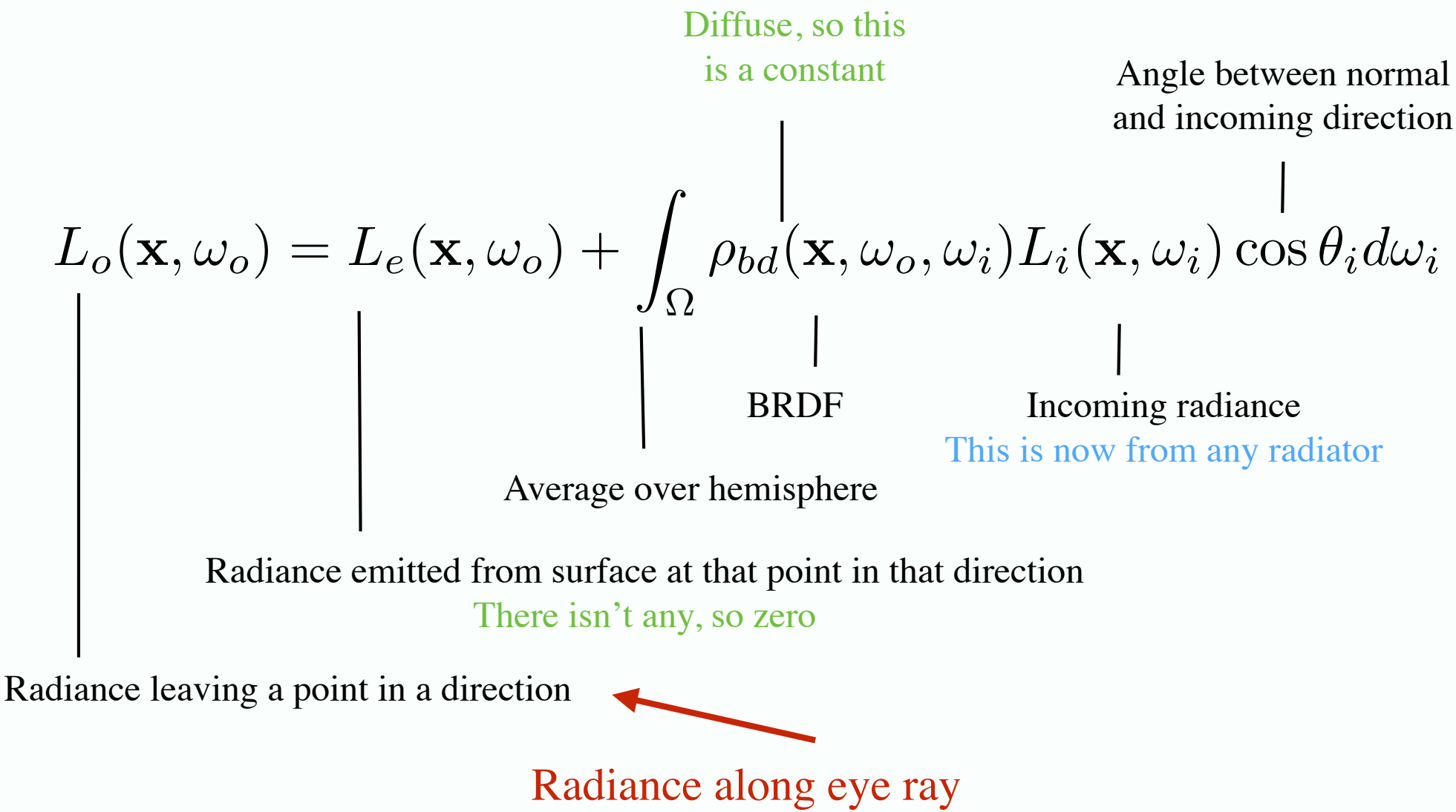


Question: how to ray-trace this?

- Model:
  - all diffuse surfaces
  - light arrives from any radiating surface
- Rendering:
  - how bright is the eye ray?



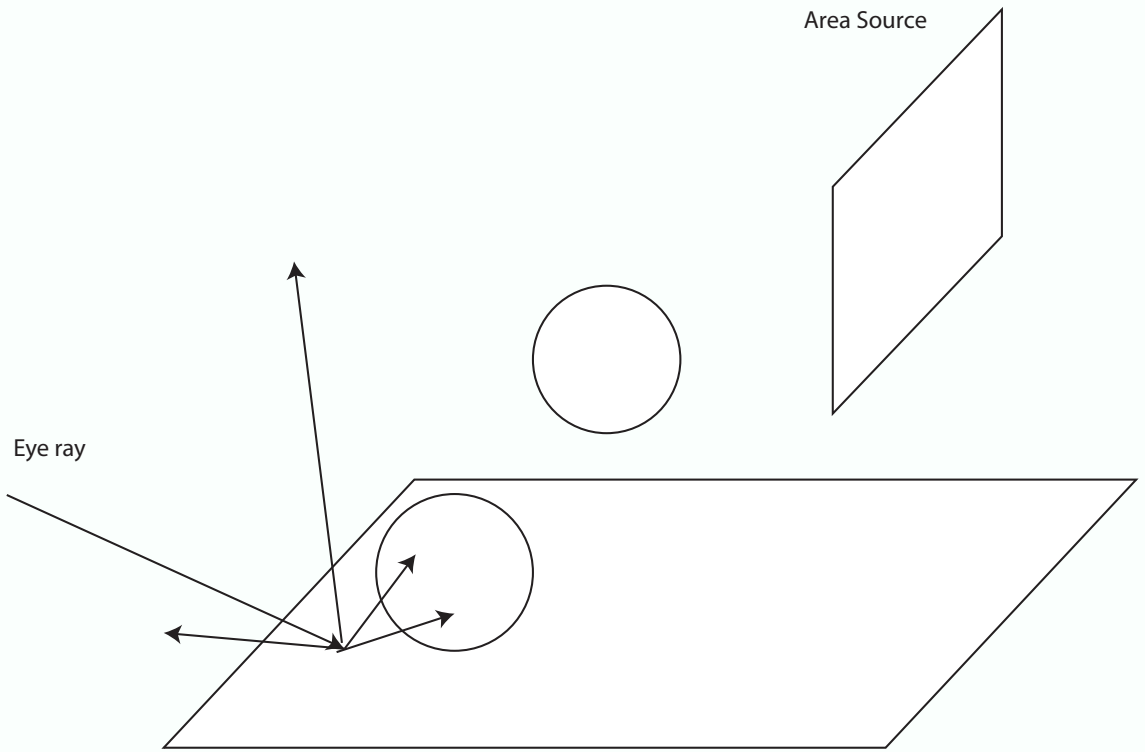
Recall


Diffuse, so this is a constant
Angle between normal and incoming direction
$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_\Omega \rho_{bd}(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i) \cos\theta_i d\omega_i$
BRDF
Incoming radiance
This is now from any radiator
Average over hemisphere
Radiance emitted from surface at that point in that direction
There isn't any, so zero
Radiance leaving a point in a direction
Radiance along eye ray

$$L_o(\mathbf{x}, \omega_o) = \cancelto{0}{L_e(\mathbf{x}, \omega_o)} + \int_\Omega \rho_{bd}(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i) \cos\theta_i d\omega_i$$

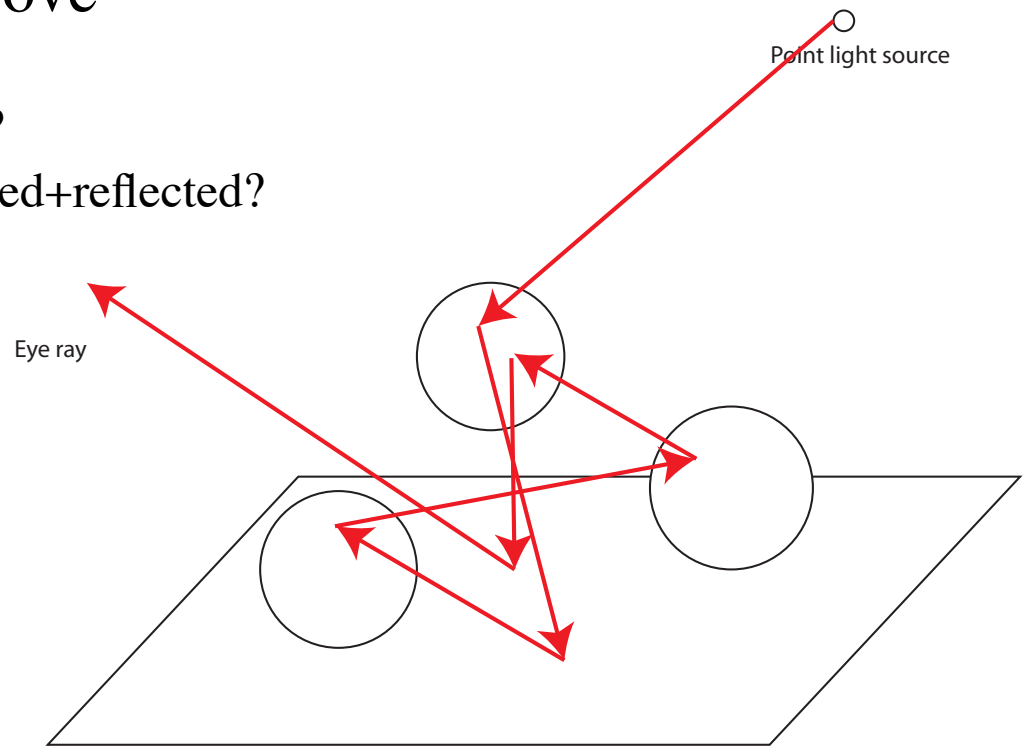
$$\frac{1}{N} \sum_{\omega_i \in \text{samples of incoming directions}} \rho L(\mathbf{x}, \omega_i) \cos\theta_i$$

But which directions?
and how should we sample them?



Light paths

- Recursively expand, as above
  - sample the incoming directions
    - what radiance is coming in?
    - go to far end - what is emitted+reflected?
    - recur
-



Light paths - II

$$v = \int_\Lambda \int_D \int_\Omega \int_T w(\mathbf{x}, \lambda, \omega, t) L(\mathbf{x}, \omega, t) dt d\omega dx d\lambda \approx \sum_{i \in \text{rays}} g(\text{ray}) L(\text{ray}$$

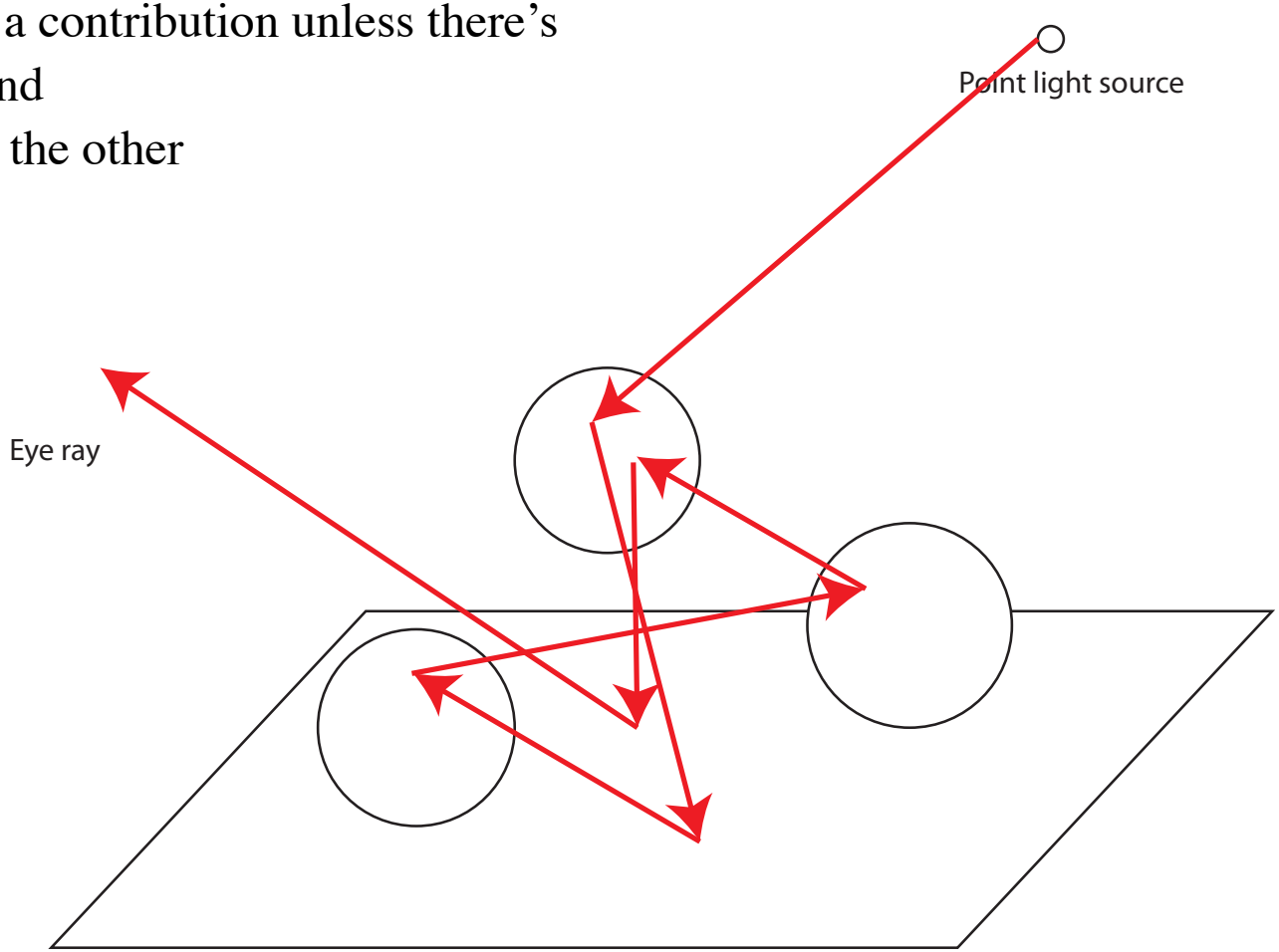
- But this is really (suppressing wavelength and time)

$$\int_D \int_\Omega L(\mathbf{x}, \omega) w(\mathbf{x}, \omega) dx d\omega \approx \frac{1}{N} \sum_{paths} (\text{contribution of path})$$

Light paths - III

- Now consider contribution of path
  - it doesn't make a contribution unless there's
    - eye at one end
    - luminaire at the other
- We can write

L (Something) E

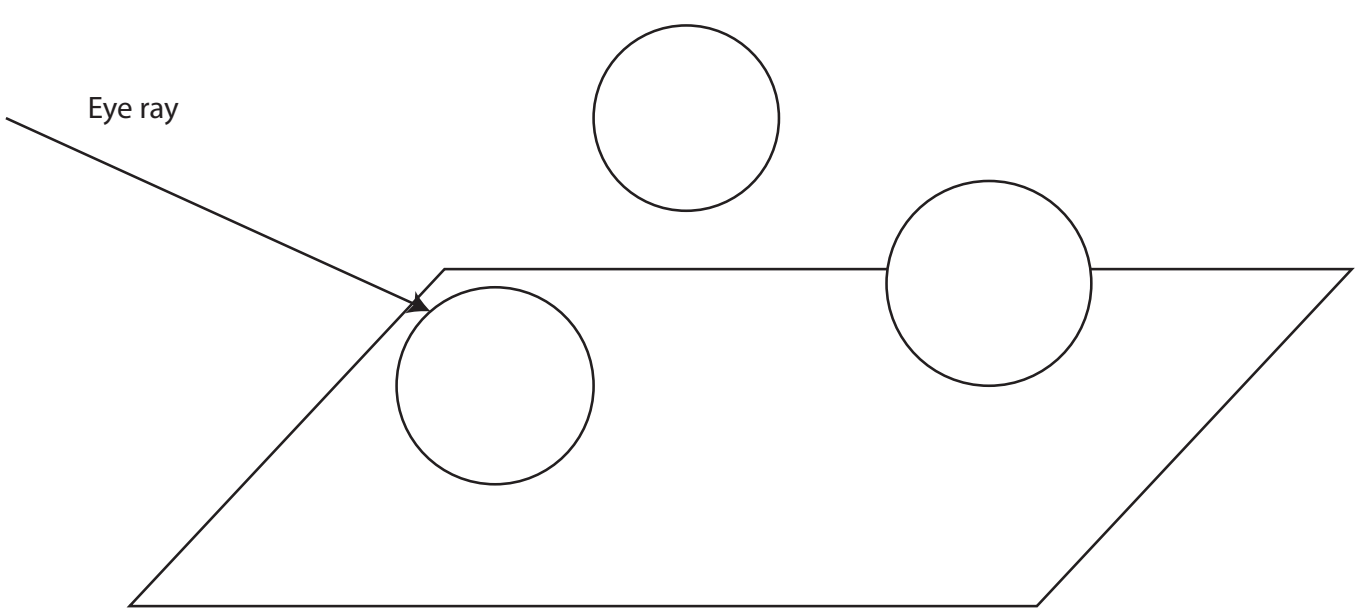


Some light paths are harder than others

- We have already seen how to render
  - LDE - (light diffuse eye)
    - eye ray to diffuse surface, can it see light?
  - LSE - (light specular eye)
    - eye ray to specular surface, reflect and hit diffuse, can it see light?
  - Actually, can do:
    - LDS*E - (light diffuse 0 or more specular bounces eye)
- How about
  - LDDE - (light diffuse diffuse eye)
    - easy geometry likely high variance
  - LS+DE - (light diffuse at least one specular eye)
    - rather harder

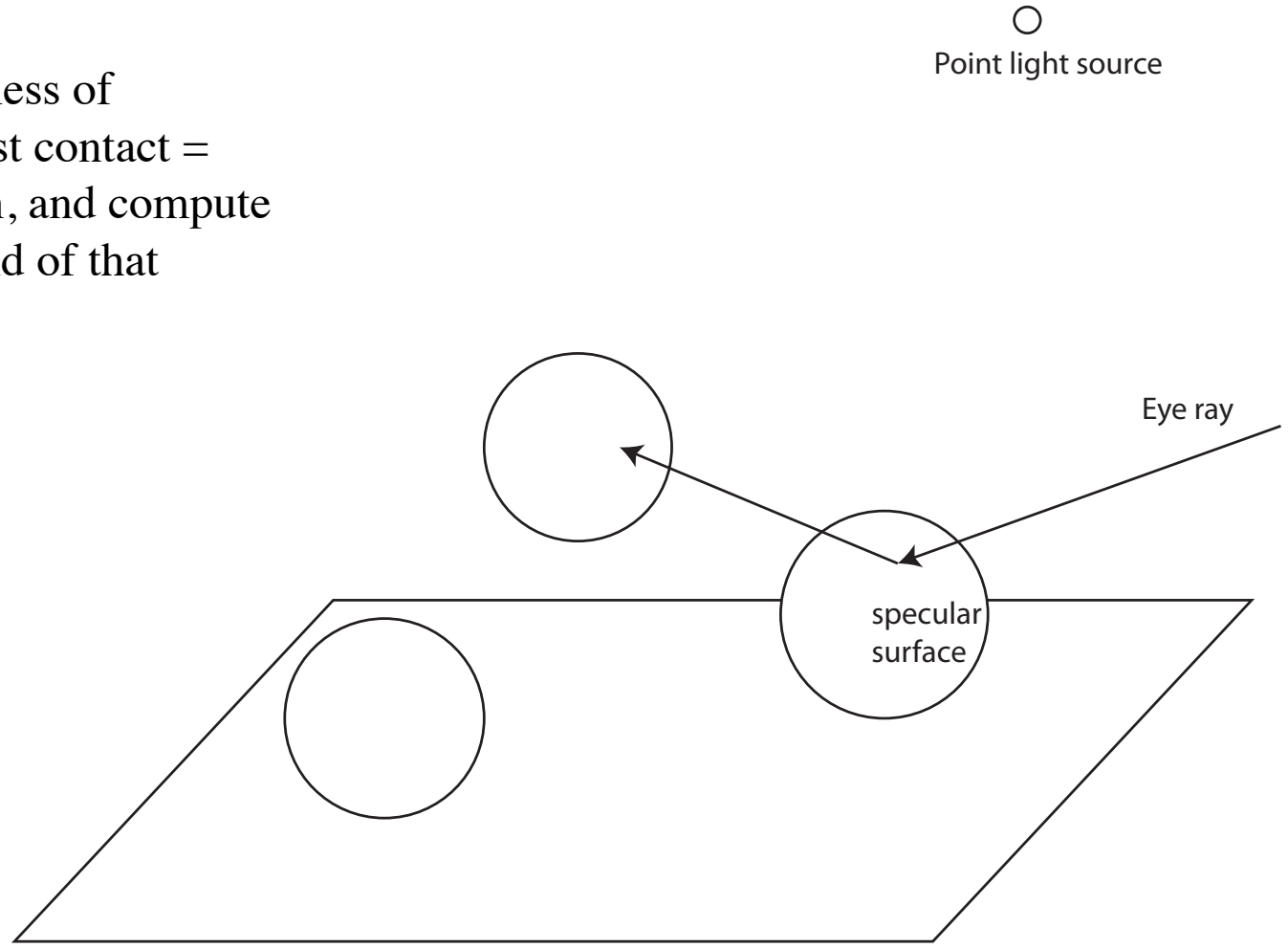
Eye ray strikes diffuse surface - LDE

Compute brightness of diffuse surface at first contact = Can it see the light sources ?= Is there an object in line segment connecting point to source?



Eye ray strikes specular surface - LDSE

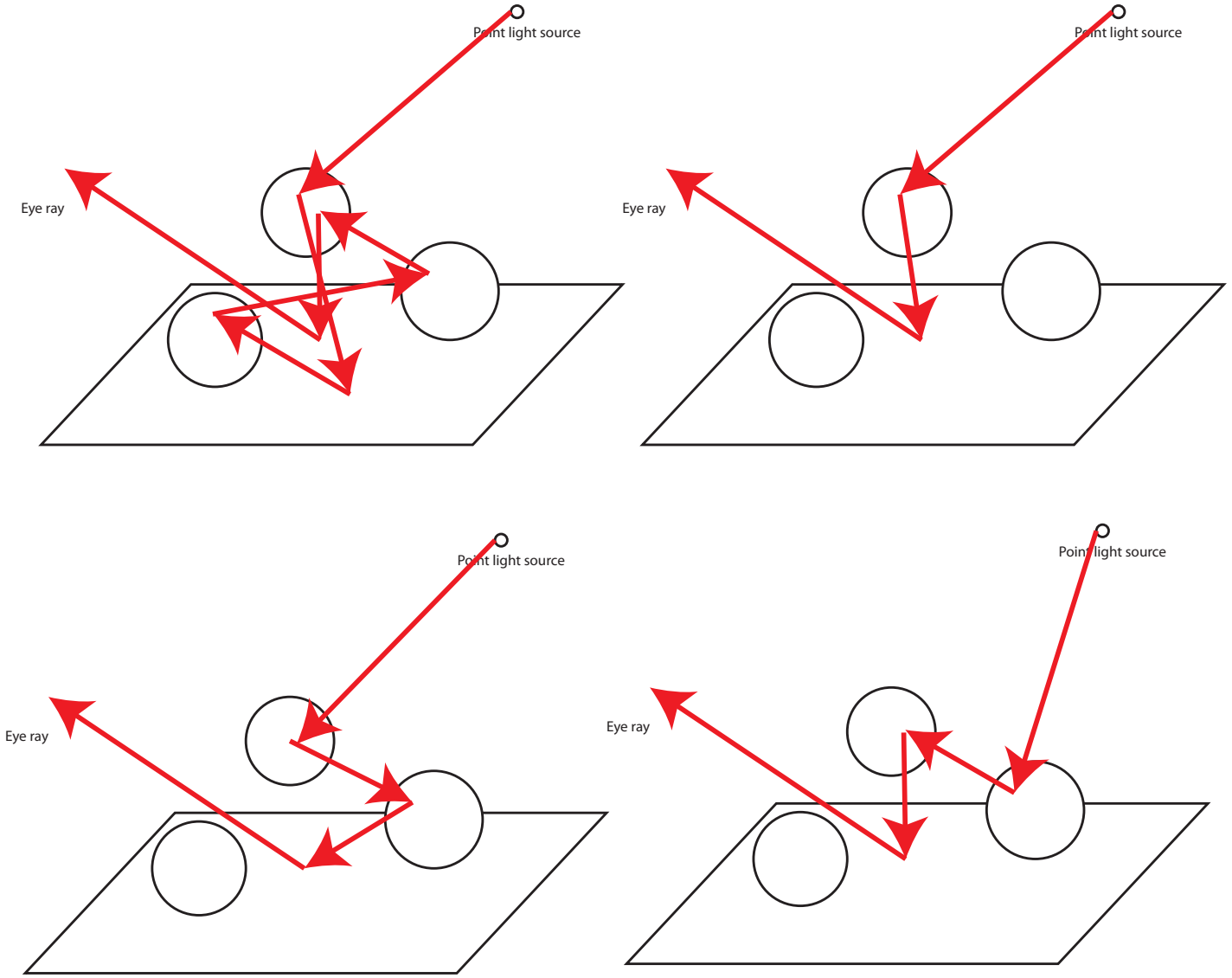
Compute brightness of specular surface at first contact = eye ray changes direction, and compute brightness at the end of that



Diffuse VS Specular (also translucent)

- Diffuse surfaces:
  - any incoming direction can cause light to travel down the eye ray
    - so you do not know from which directions contributions will arrive
  - when an eye ray arrives, it must create multiple query rays
- Specular surfaces:
  - only one incoming direction can cause light to travel down the eye ray
    - so you do know from which directions contributions will arrive
  - when an eye ray arrives, it creates only one query ray
- Translucent surfaces are like specular surfaces:
  - different geometry for the query ray

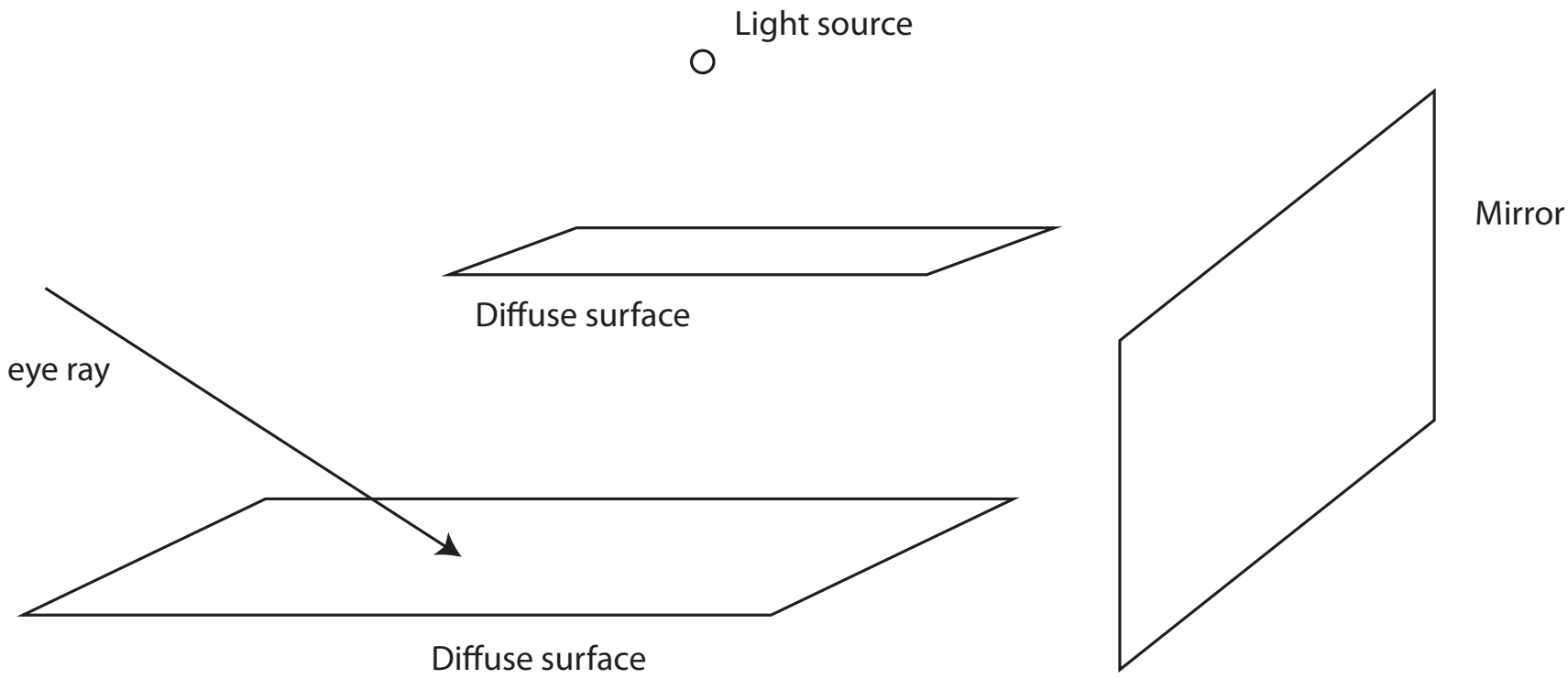
LDD+E - easy geometry, but variance



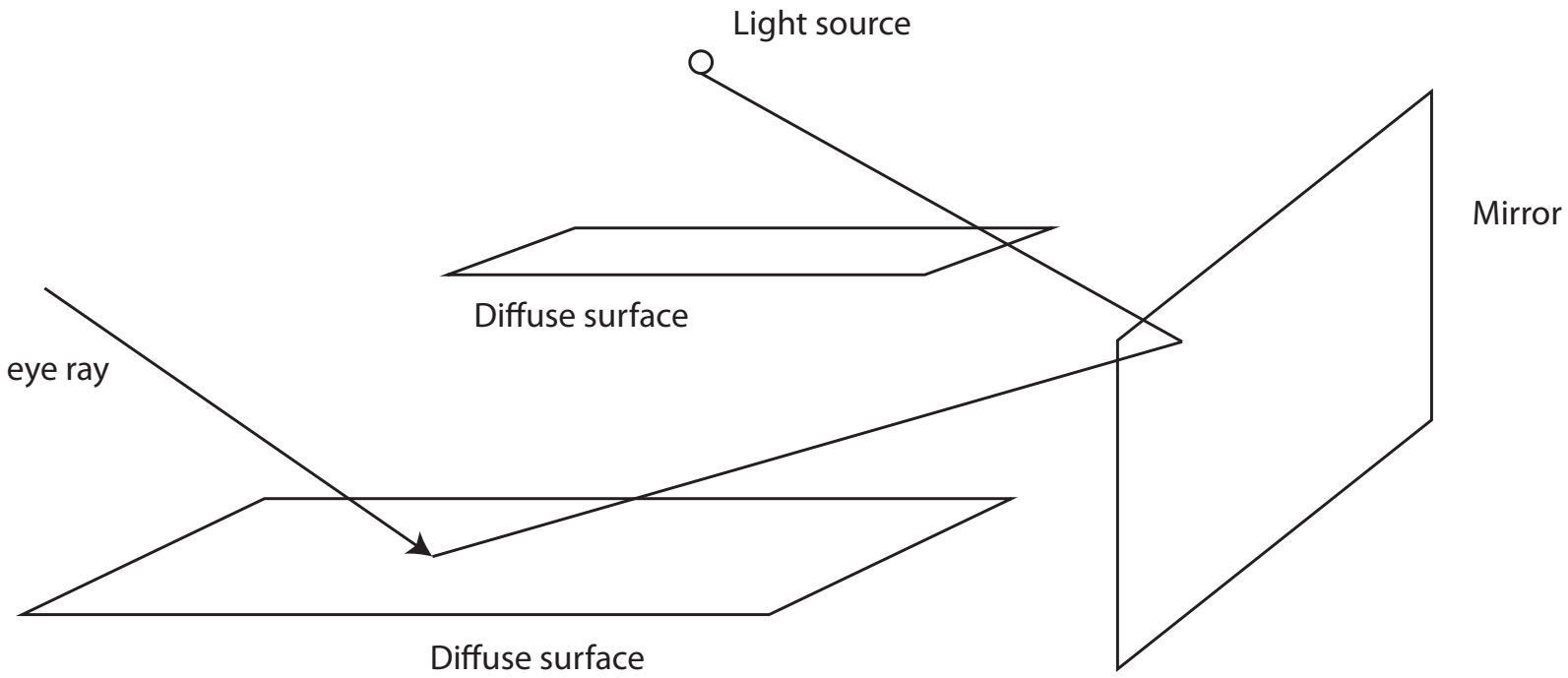
LDD+E - variance control (sketch!)

- In principle, easily sampled recursively
- Preferentially sample paths that make large contributions
  - these are paths that connect light, eye, via high albedo surfaces
    - “Russian roulette”
      - continue path with probability = albedo
      - weight by (1/albedo)
- OR Cache results
  - propagating a path:
    - check: is there something in the cache?
      - yes: use it
      - no: propagate path and cache results

LS+DE - harder - where is the light?



LS+DE - harder - where is the light?



Problem: which direction leaving diffuse surface will hit the light?

Strategies

- Bidirectional ray tracing
  - Trace a lot of rays from light through specular surfaces to first diffuse
  - Trace a lot of eye rays to first diffuse
  - Join paths
  - Variance control
    - weight paths as if they'd been found in different ways (Veach +Guibas)
- Markov chain monte carlo
  - take a path and mutate it; reweight contribution of mutated path
- Caching
  - Photon cache - trace many rays from light through specular to diffuse
    - cache at diffuse
    - query with eye ray

Biased vs unbiased rendering

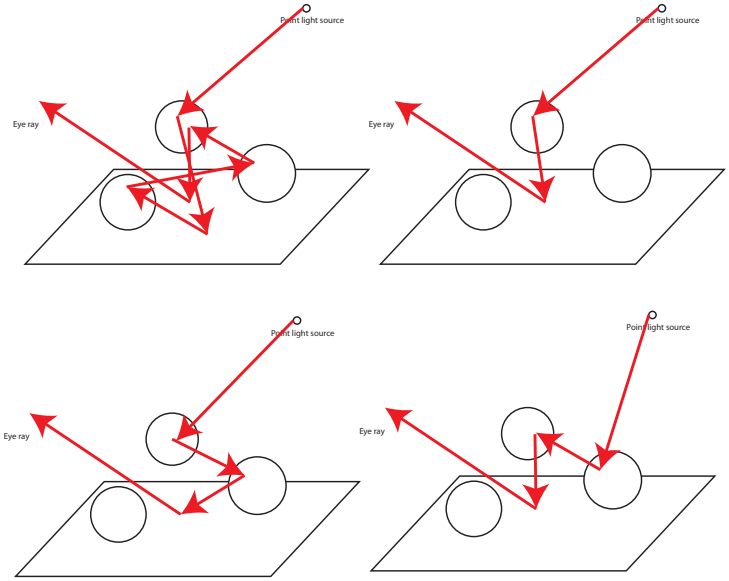
- Unbiased renderer
  - pixel value is value of random variable (different paths=different values)
  - E(estimate)=True value
  - sometimes essential
    - eg estimate the amount of light in a museaum hall
- Biased renderer
  - E(estimate)=True value + Bias
  - eg Photon map (as above)
    - Bias because we do not know how many photons to stick in cache
  - Often more realistic
- Crucial point
  - Very few people can tell if a render is nearly physically correct
    - and no-one can reliably spot exactness

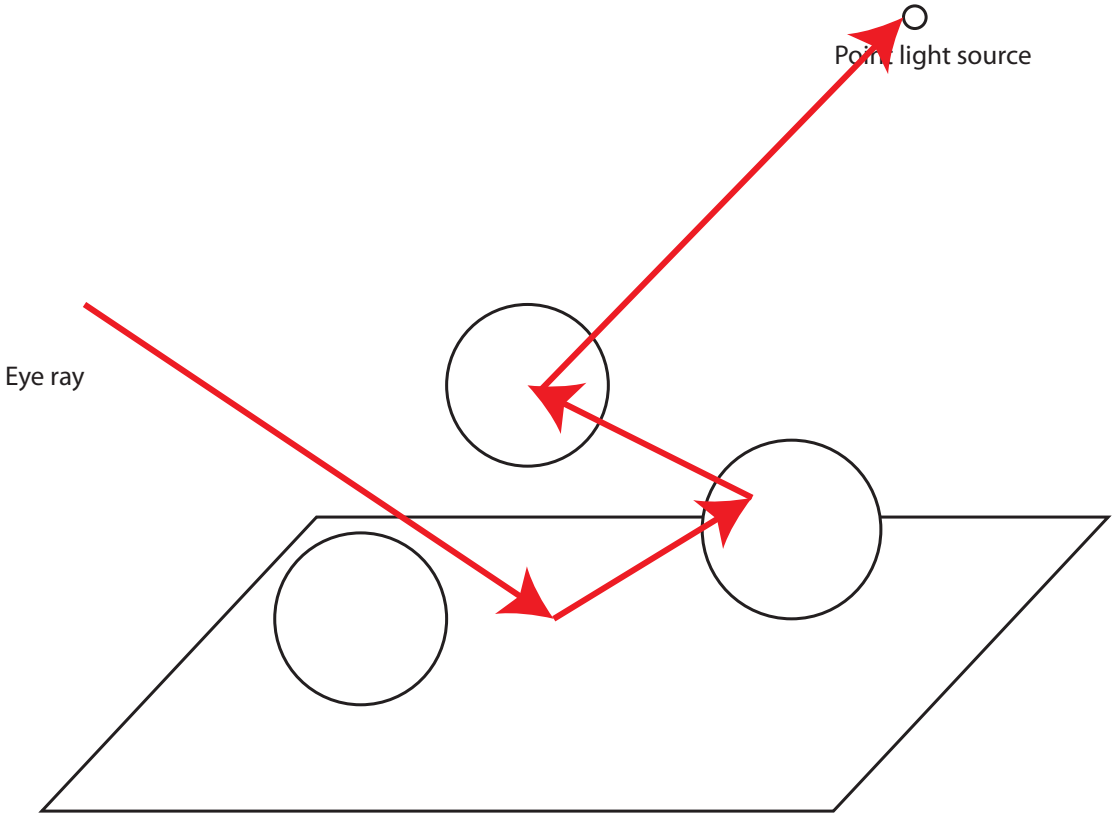
The plenoptic function

- We are repeatedly sampling radiance
  - as function of point, direction
- What if we had a function that could report that?

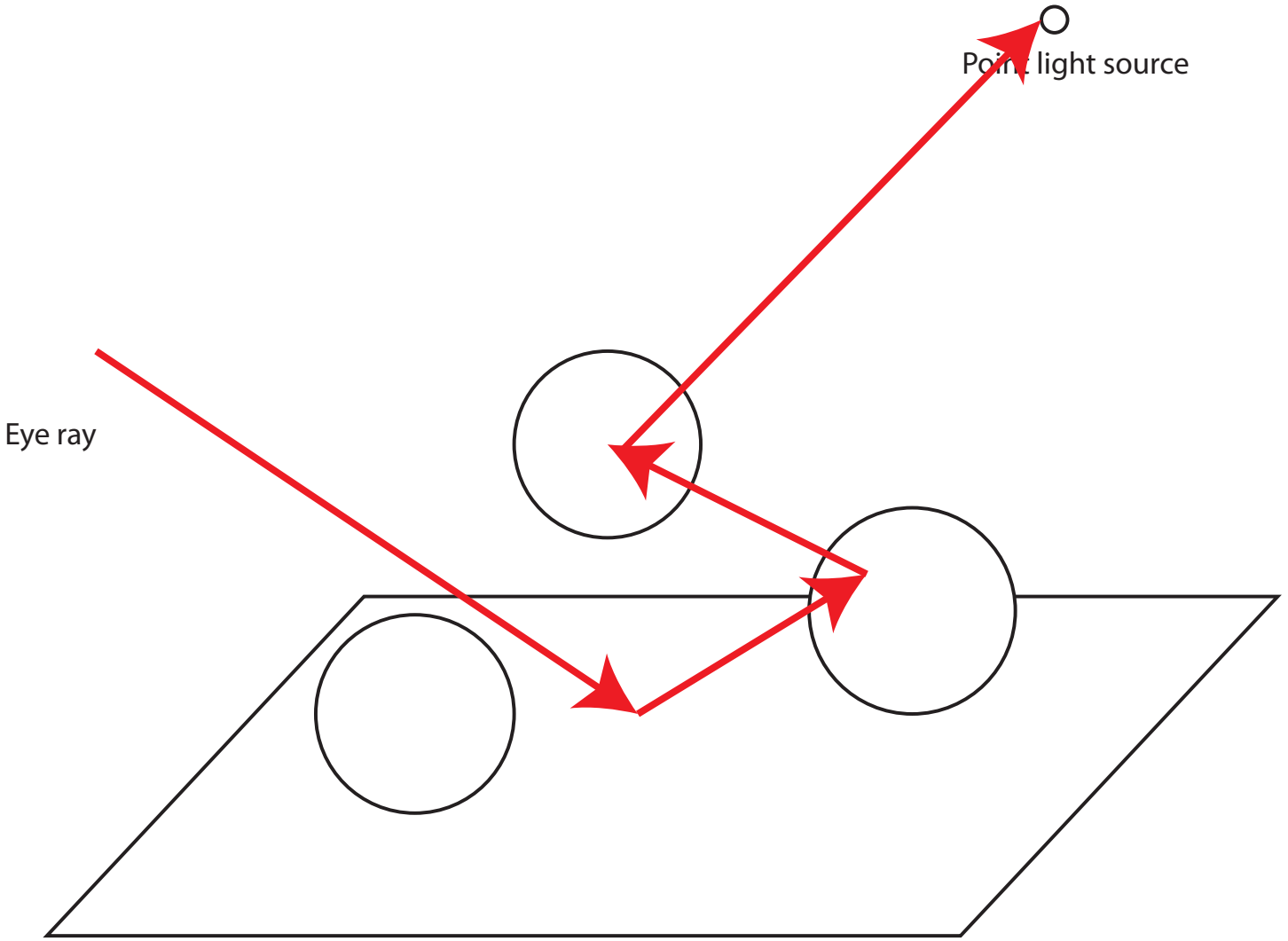
- Plenoptic function
  - radiance along a directed line
  - space of lines is rather nastier than you might think

The plenoptic function as a cache





The plenoptic function as a cache



The plenoptic function - careful!

- This is a function whose domain is nasty
  - all maximal directed line segments (lines) in free space
  - the domain can get very complicated
    - easy when there aren't any objects
    - otherwise, much harder
  - domain is sometimes called a visibility complex

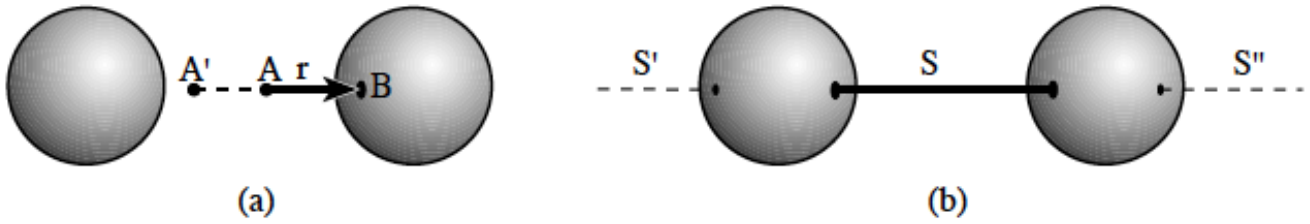
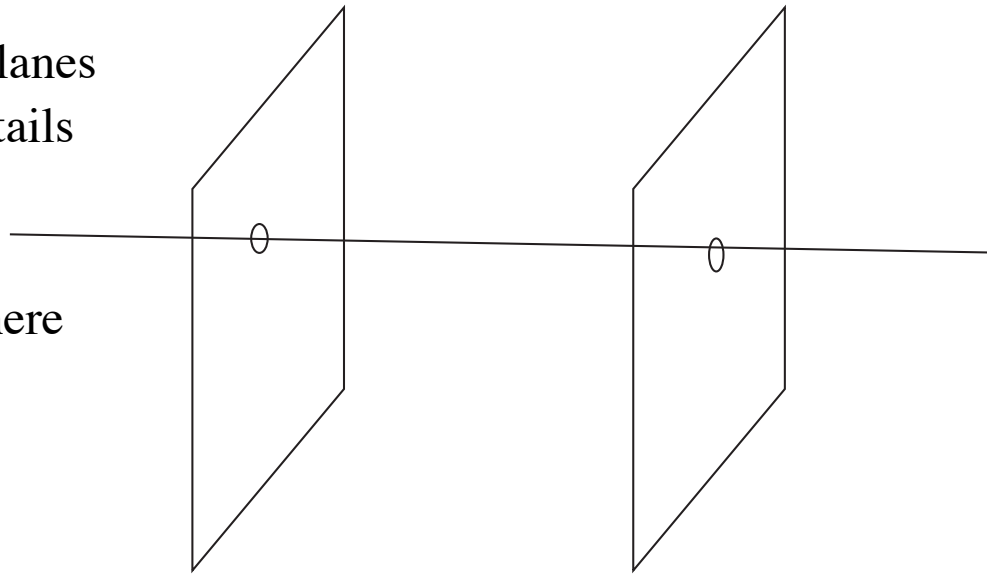


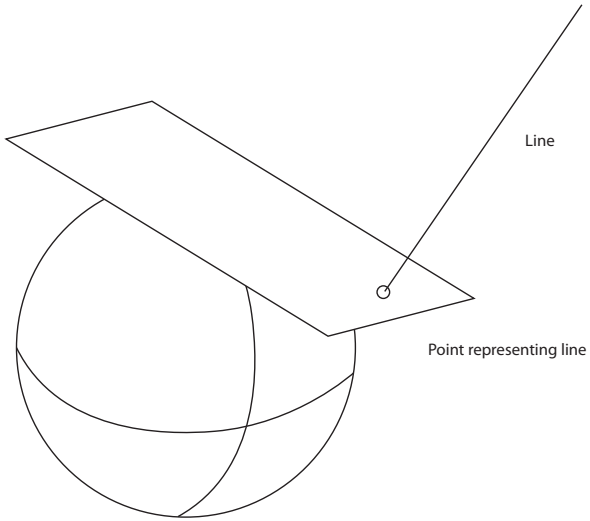
Fig. 1. Maximal free segment. (a) All the rays collinear to r whose origin is between the two spheres "see" point B. (b) These rays are grouped into a *maximal free segment* S. Two other maximal free segments S' and S'' are collinear to S.

Durand et al 02

Lines in 3D (if it's empty!)

- Space of lines is 4 dimensional
  - can specify a line by:
    - where it intersects each of two planes
      - some missing lines, some details
  - alternative
    - directed line
    - point on the tangent plane of sphere





Lines in 3D with object can be nasty

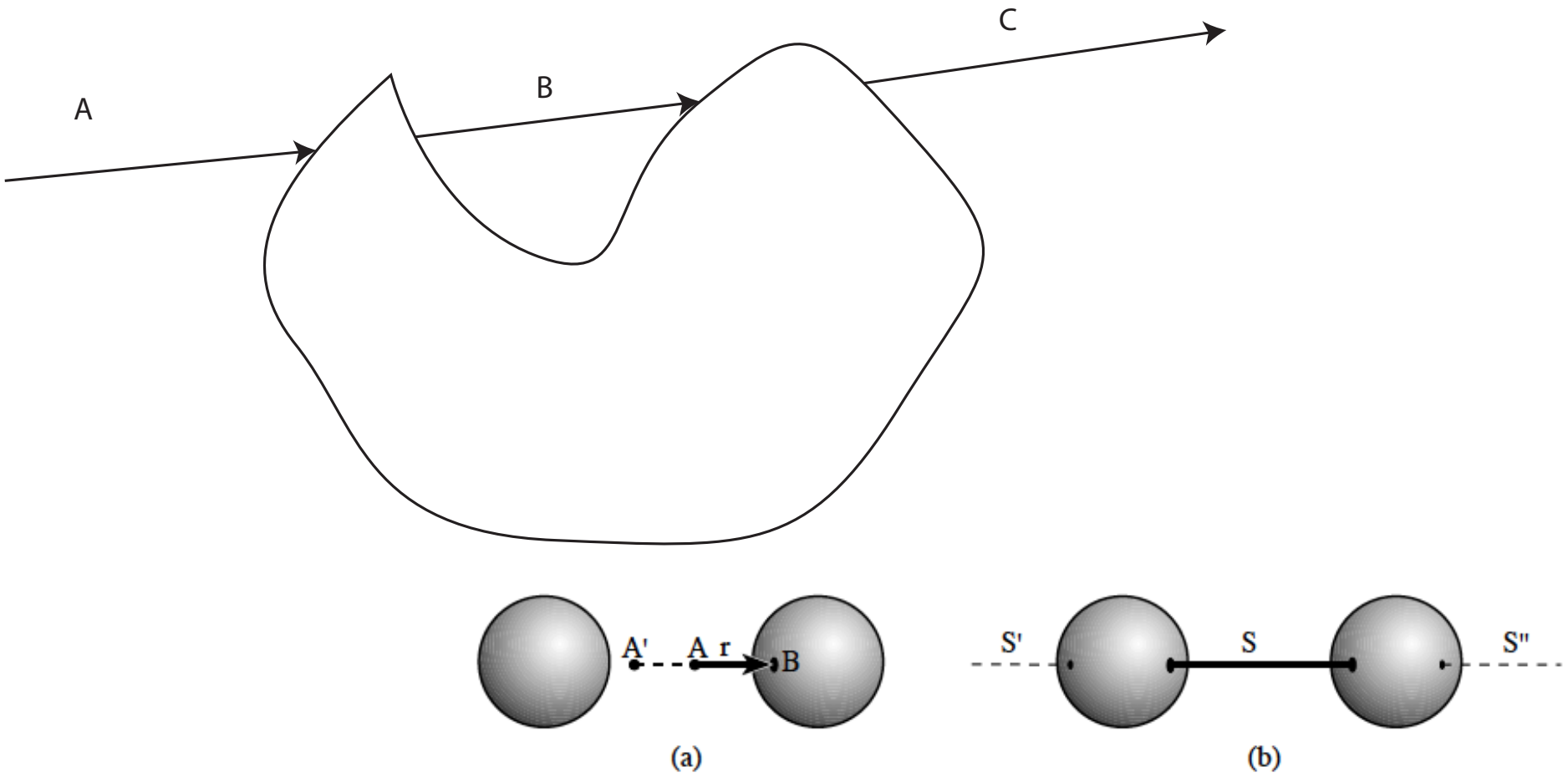
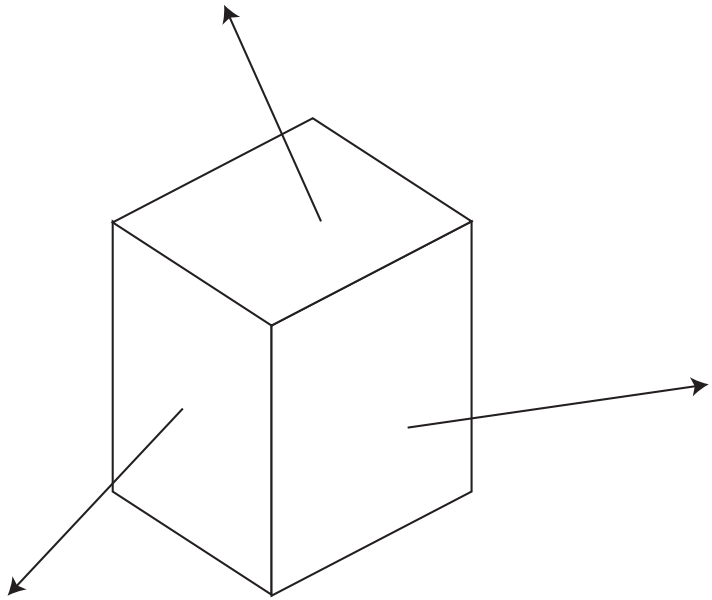


Fig. 1. Maximal free segment. (a) All the rays collinear to r whose origin is between the two spheres "see" point B. (b) These rays are grouped into a *maximal free segment* S. Two other maximal free segments S' and S'' are collinear to S.

Durand et al 02

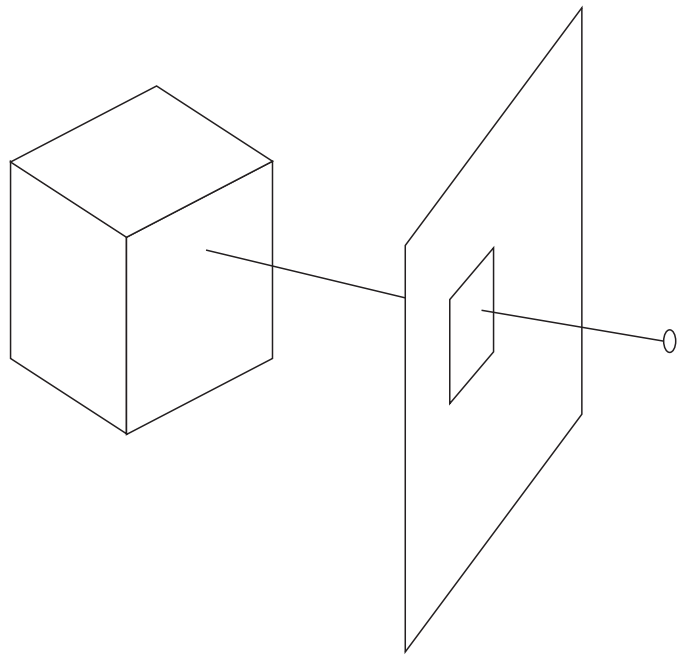
Simplify

- Place an object in a box
  - record radiance for each ray leaving box
- Easy to ray trace
  - look up eye ray in rays leaving box
    - report that value
- Capture is relatively easy



Capturing this representation

- Obtain an awful lot of images from calibrated cameras
  - each image is a set of rays leaving the box
  - calibration



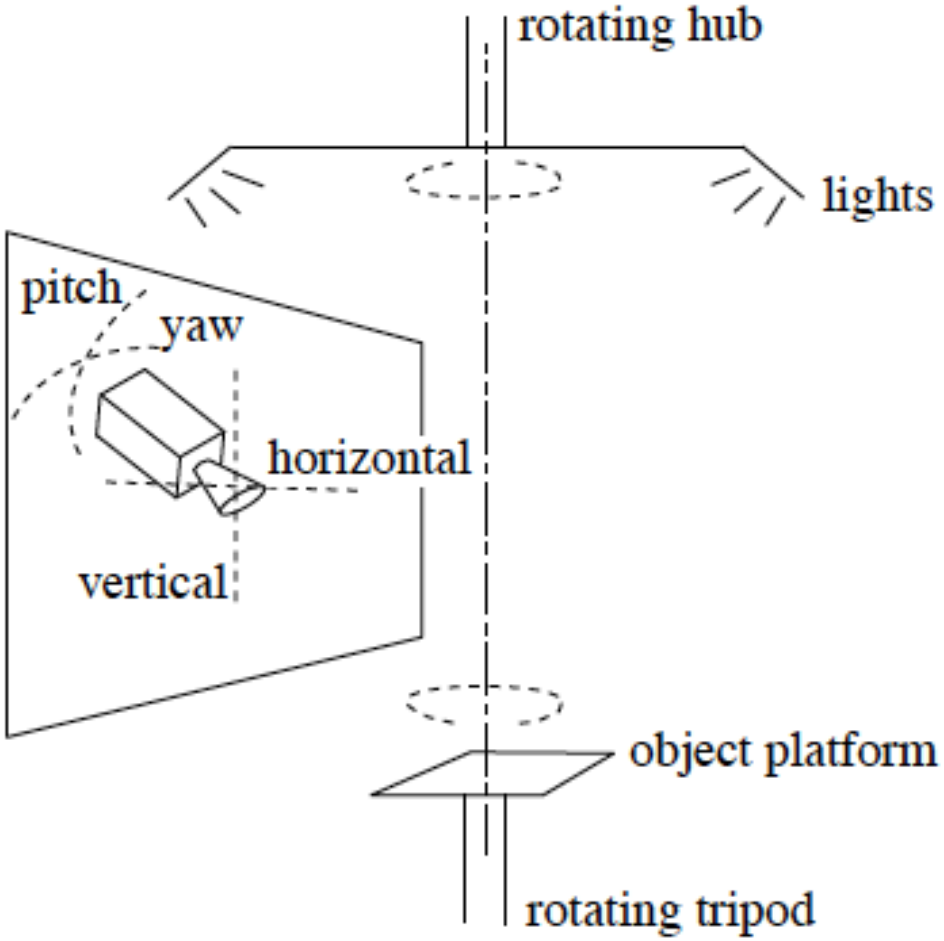


Figure 10: Object and lighting support. Objects are mounted on a Bogen fluid-head tripod, which we manually rotate to four orientations spaced 90 degrees apart. Illumination is provided by two 600W Lowell Omni spotlights attached to a ceiling-mounted rotating hub that is aligned with the rotation axis of the tripod. A stationary 6' x 6' diffuser panel is hung between the spotlights and the gantry, and the entire apparatus is enclosed in black velvet to eliminate stray light.

Levoy and Hanrahan, 96

Rendering

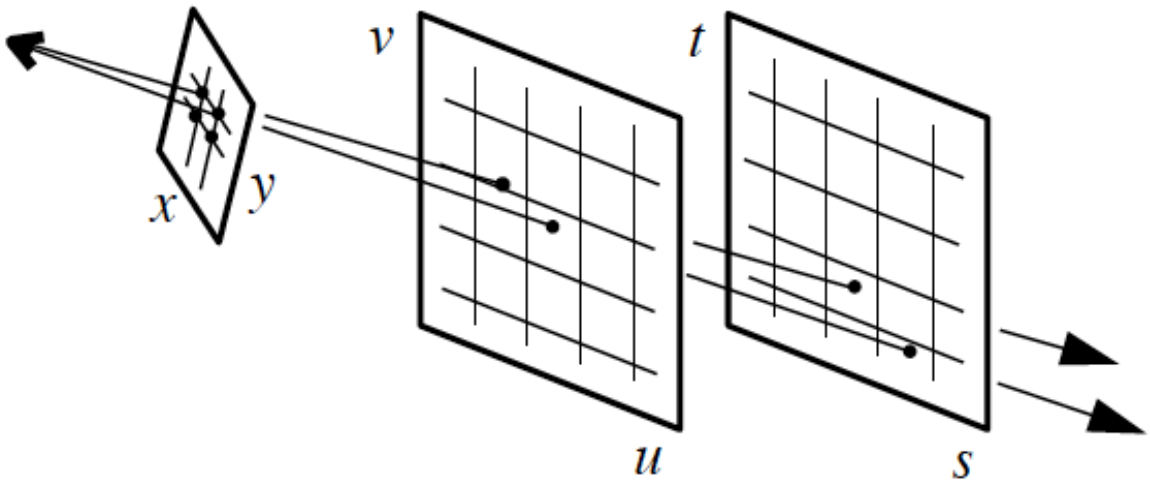


Figure 12: The process of resampling a light slab during display.

Levoy+Hanrahan, 96

Issue: Sampling and Interpolation

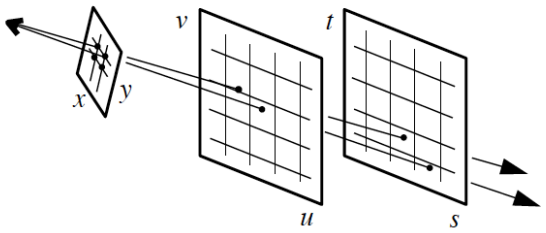


Figure 12: The process of resampling a light slab during display.

- Almost every eye ray ends up “between” uv, st samples
  - we must interpolate (smooth; something)
  - Traditional: multilinear interpolation

Interpolation helps, but..

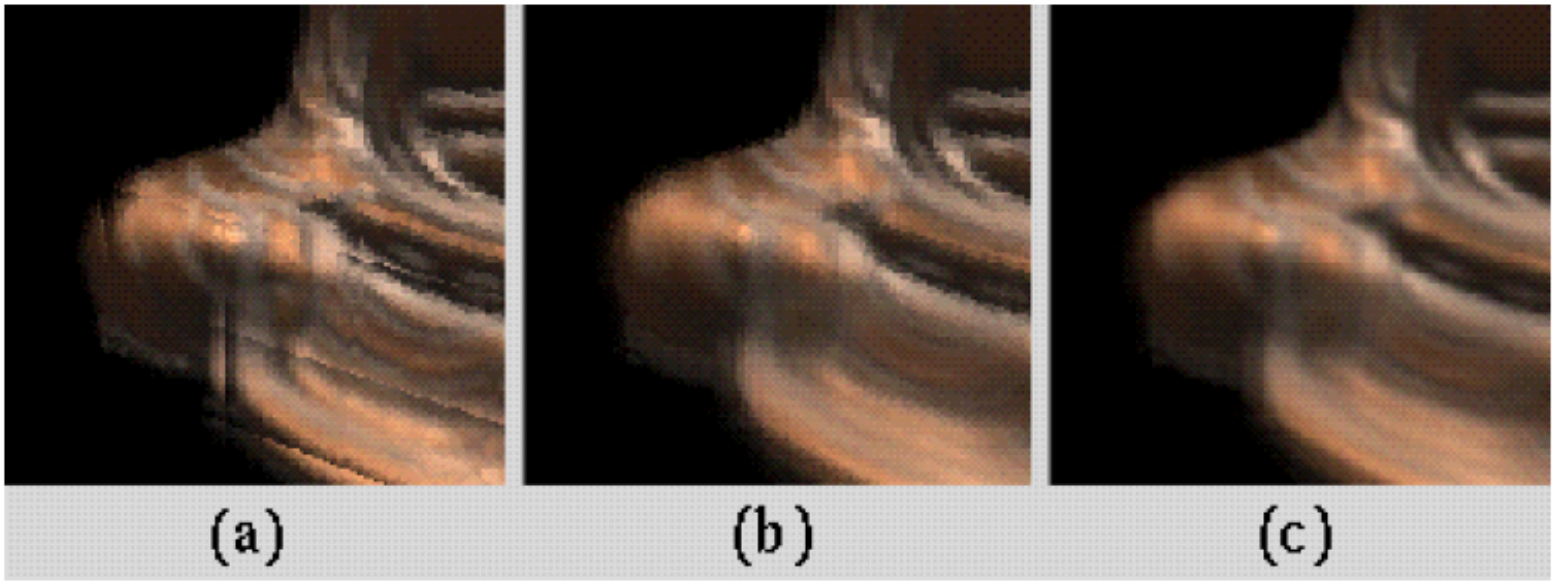
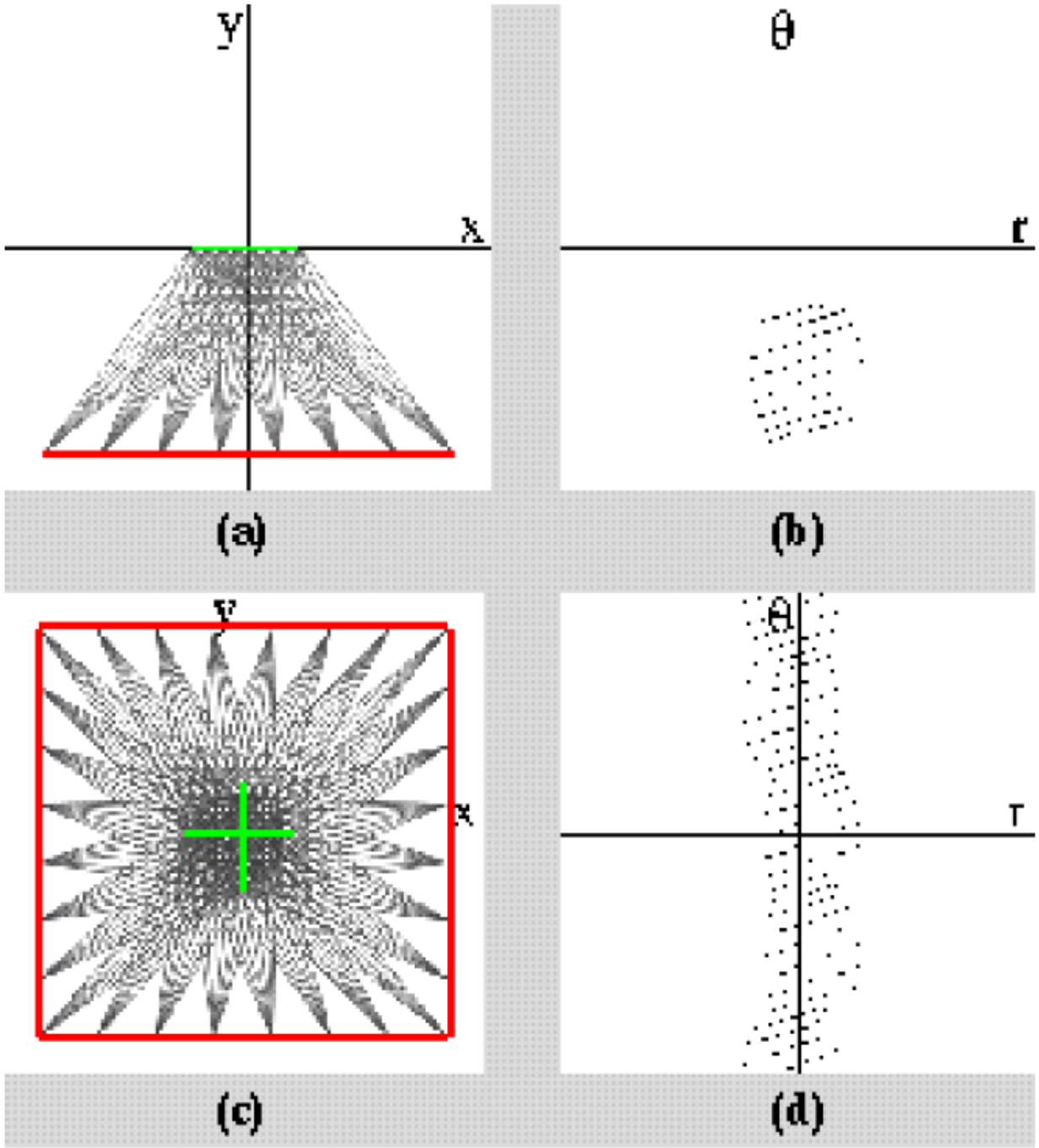


Figure 13: The effects of interpolation during slice extraction. (a) No interpolation. (b) Linear interpolation in uv only. (c) Quadra-linear interpolation in uvst.



Two plane representation and sampling

Levoy+Hanrahan, 96

Figure 3: Using line space to visualize ray coverage. (a) shows a single

Revise model

- We need:
  - better interpolation
  - easier capture
  - some way to deal with the awkwardness of line representations
- Ideas:
  - move to scattering/volume rendering based representation
    - this will make the line representation easier to deal with
  - use a multilayer perceptron to represent relevant functions

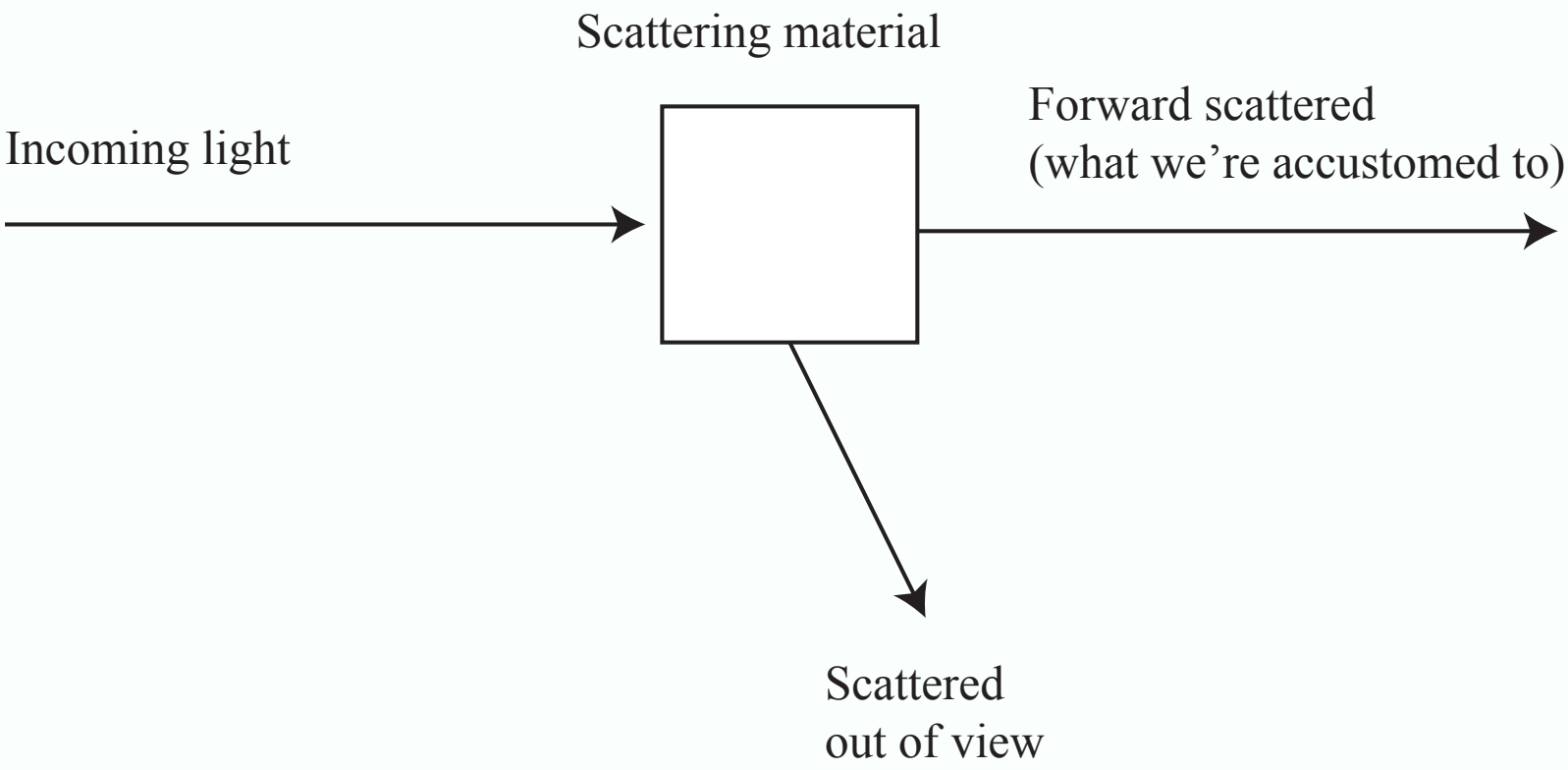
Scattering

- Fundamental mechanism of light/matter interactions
- Visually important for
  - slightly translucent materials (skin, milk, marble, etc.)
  - participating media
- In fact, it's the mechanism underlying reflection

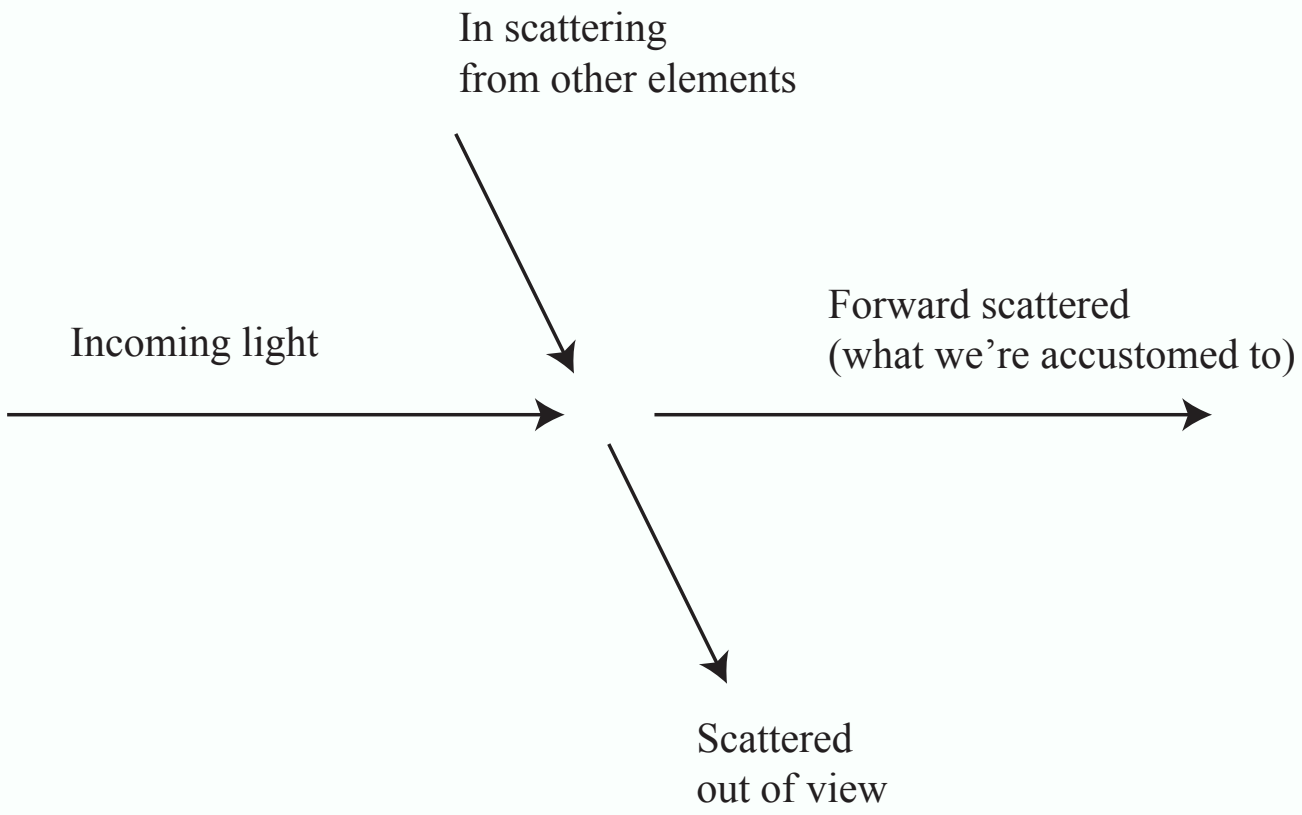
Participating media

- for example,
  - smoke,
  - wet air (mist, fog)
  - dusty air
  - air at long scales
- Light leaves/enters a ray travelling through space
  - leaves because it is scattered out
  - enters because it is scattered in
- New visual effects

Light hits a small box of material

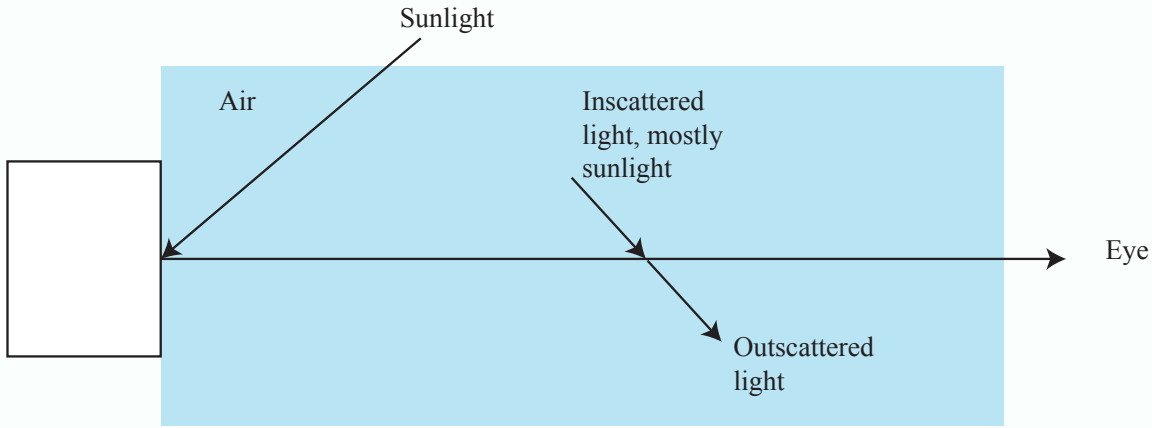


A ray passing through scattering material

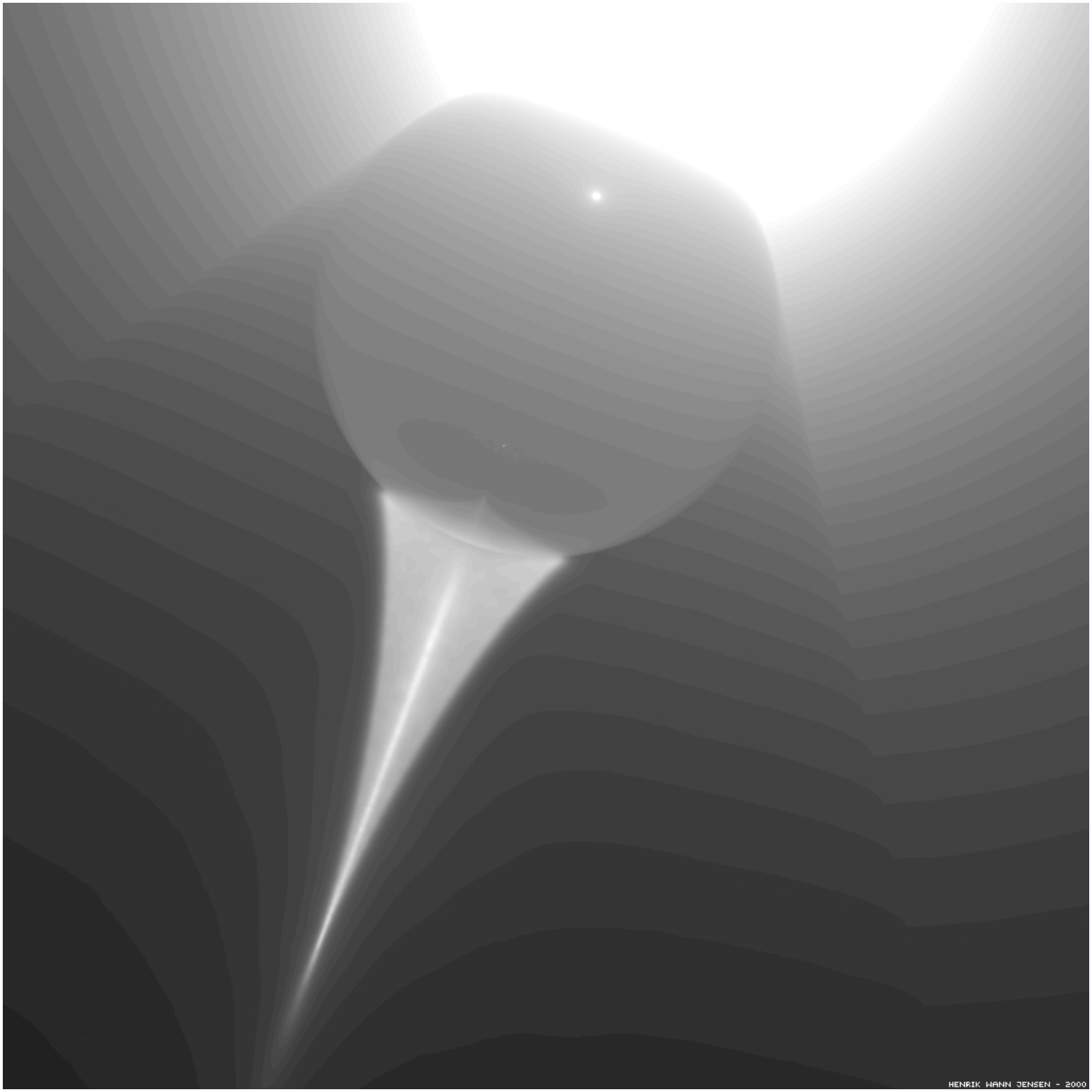


From Lynch and Livingstone, Color and Light in Nature

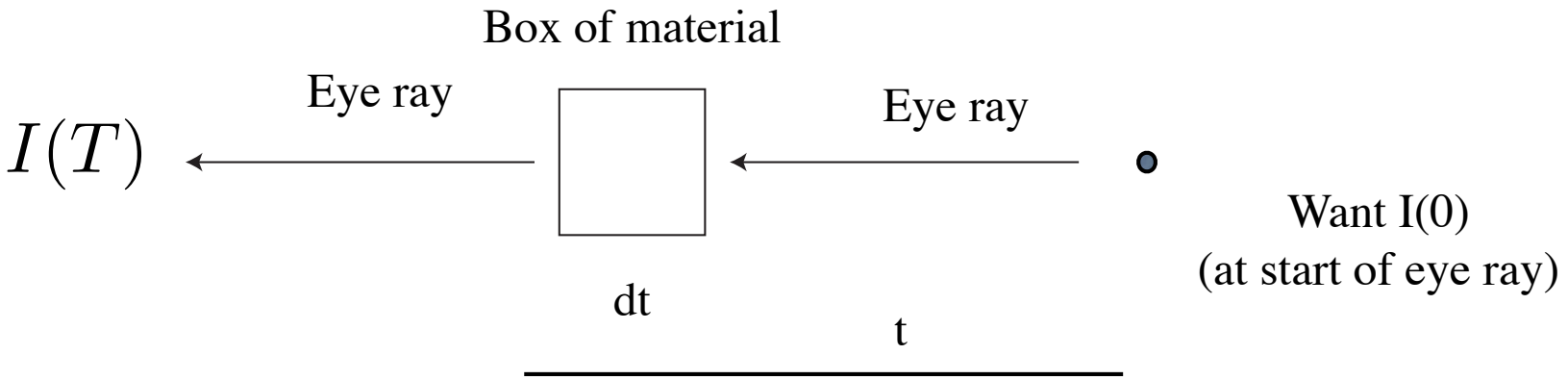
Airlight as a scattering effect



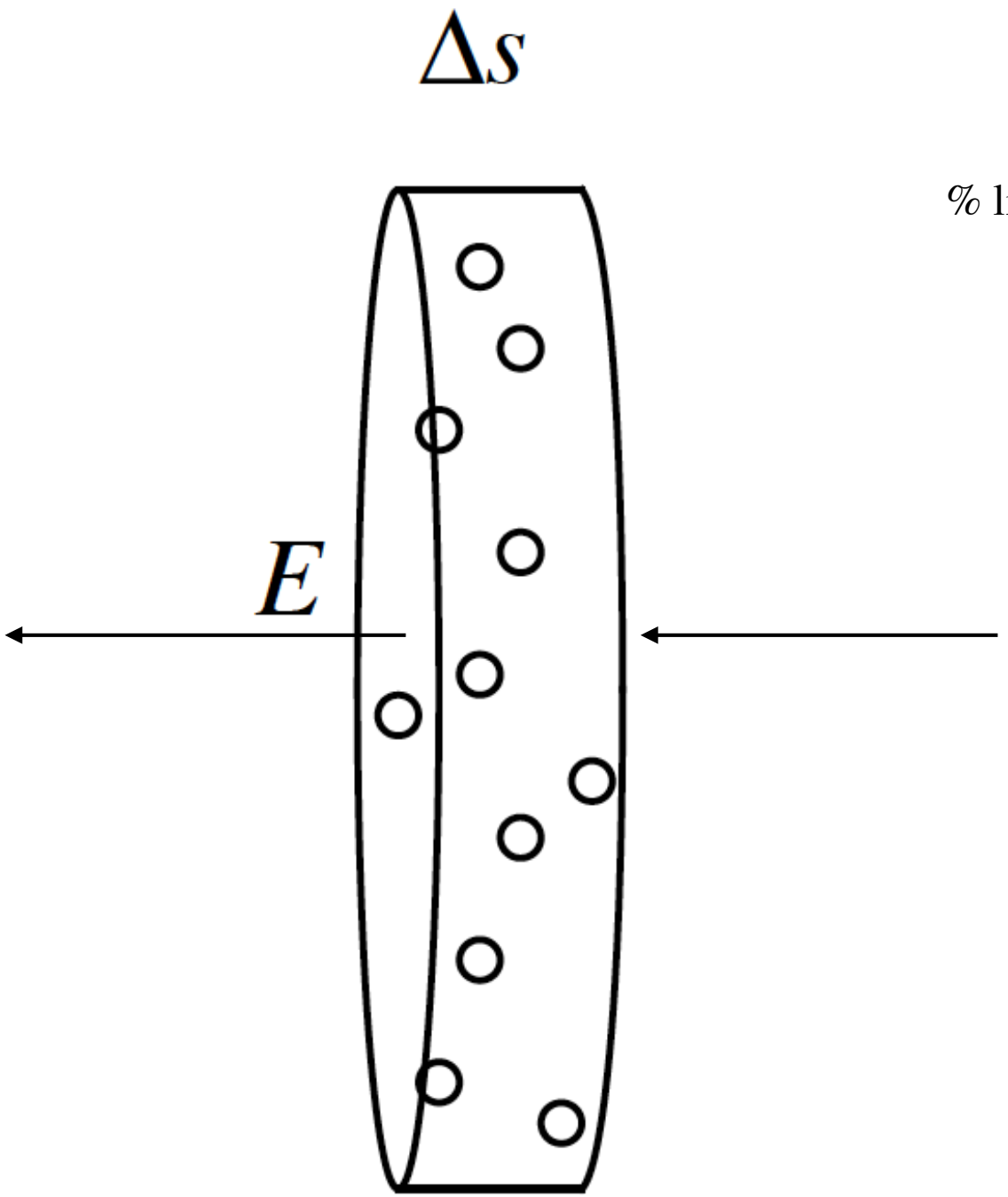
From Lynch and Livingstone, Color and Light in Nature


HENRIK WANN JENSEN - 2000

Absorption



- Ignore in-scattering
  - only account for forward scattering
- Assume there is a source at t=T
  - of intensity I(T)
  - what do we see at t=0?



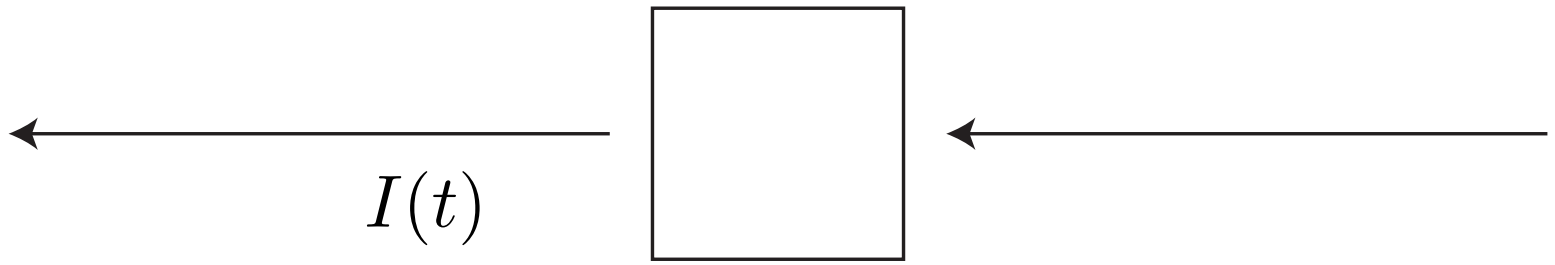
Cross sectional area of “slab” is E
Contains particles, radius r, density rho

Too few to overlap when projected

% light absorbed = (area of projected particles)/ (area of slab)

This is:

$$\frac{(\rho E \Delta s)\pi r^2}{E} = \sigma(s)\Delta s$$



$$I(t - \delta t) = I(t) - \sigma(t) I(t) \delta(t)$$

Extinction coefficient

$$\frac{dI}{dt} = \sigma(t) I(t)$$

$$\frac{d \log I}{dt} = \sigma(t)$$

$$I(T) = I(0) e^{\int_0^T \sigma(t) dt}$$

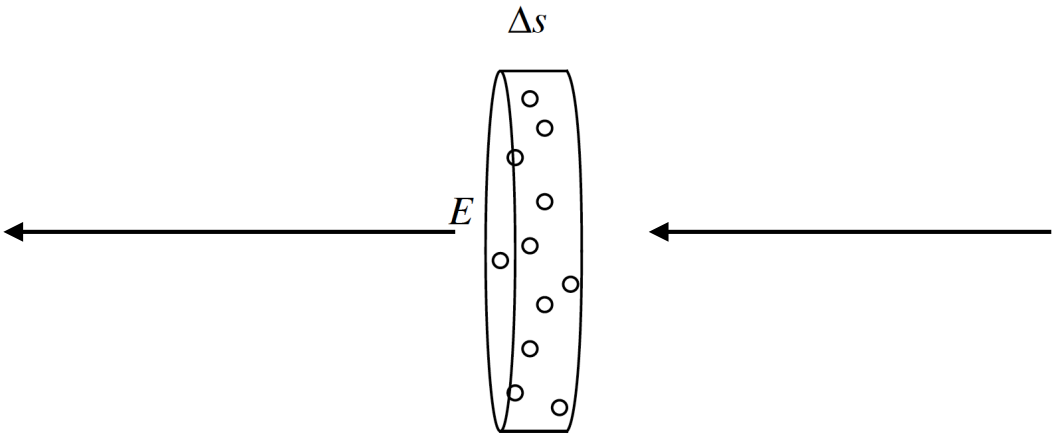
$$I(0) = I(T) e^{-\int_0^T \sigma(t) dt}$$

Eye is at 0

Intensity at T

More interesting…

- Intensity is "created along the ray"
  - by (say) airlight
  - Model - the particles glow with intensity C(x)



Cross sectional area of "slab" is E
Contains particles, radius r, density rho

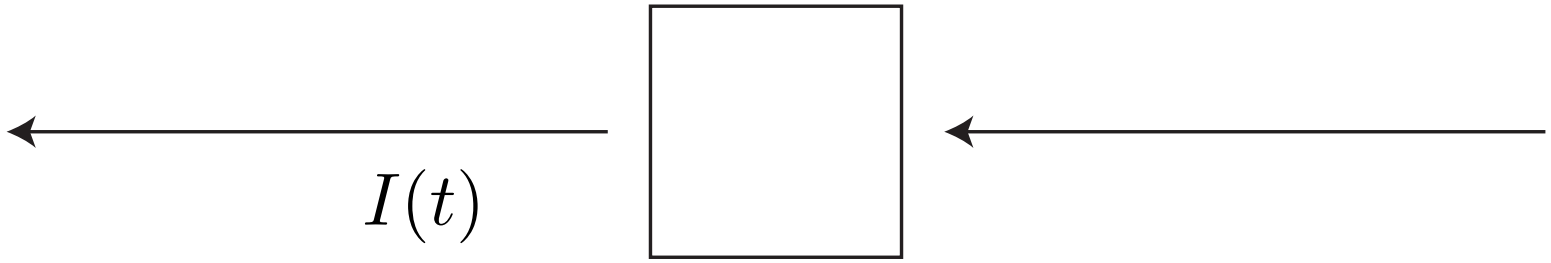
Too few to overlap when projected

Light out = Light in -
Light absorbed+
Light generated

Light generated: C x (area fraction of proj. particles)

which is

$$C(\mathbf{x}(s))\frac{(\rho E \Delta s)\pi r^2}{E} = C(\mathbf{x}(s))\sigma(s)\Delta s$$



$$I(t-\delta t) = I(t) - \sigma(t)I(t)\delta t + \mathbf{c}(\mathbf{x}(t))\sigma(t)\delta t$$

Absorption

Generation

$$I(0) = \int_0^T \mathbf{c}(\mathbf{x}(s))\sigma(s)e^{-\int_0^s \sigma(u)du} ds$$

$$I(0) = \int_0^T \underbrace{\mathbf{c}(\mathbf{x}(s))\sigma(s)}_{\text{Made at s}} \underbrace{e^{-\int_0^s \sigma(u)du}}_{\substack{\text{Absorbed in transit} \\ \text{from s to 0}}} ds$$

Accumulate along ray

Yields

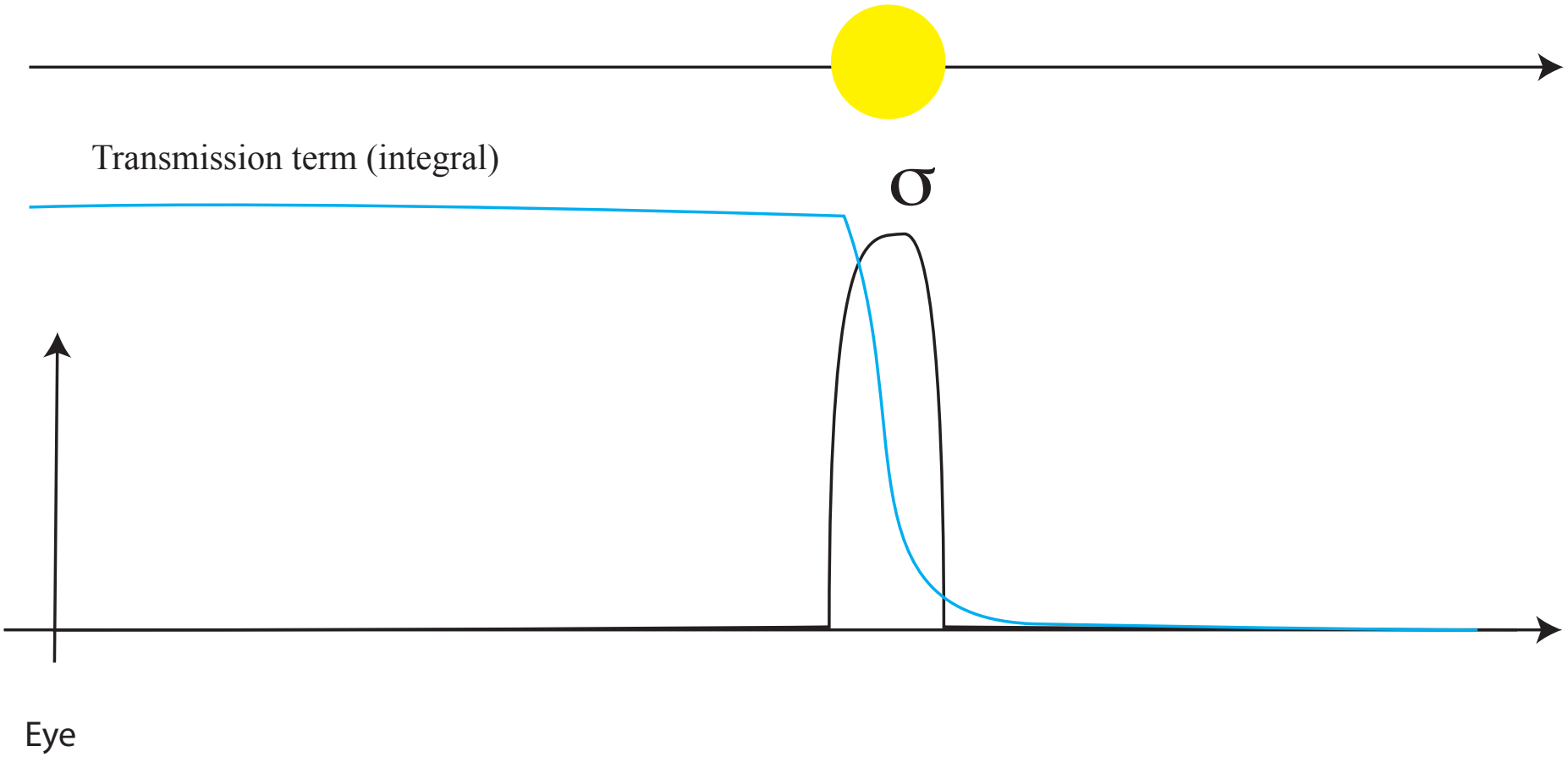
The volume density $\sigma(\mathbf{x})$ can be interpreted as the differential probability of a ray terminating at an infinitesimal particle at location $\mathbf{x}$. The expected color $C(\mathbf{r})$ of camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with near and far bounds t_n and t_f is:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right). \quad (1)$$

Mildenhall et al, 20

Actual rendering…

- Integration problem
  - walk back along ray from viewpoint, sampling
    - collect color at sample point
    - accumulate transmission
      - if weight is too small, stop walking
- This could be nasty…
  - variety of strategies, depending on what we know about c, sigma, eg
    - known in voxels
      - cut ray into segments (per voxel), compute integral per segment
    - parametric function
      - cut ray into uniform segments, one sample per segment
- but the integral is a differentiable function of c, sigma


Transmission term (integral)
σ
Eye

NERF representations

- Build neural network to predict
  - c, sigma as functions of position, direction
- Render this object with a volume renderer
- Learn using images geometrically calibrated to one another

NERF representations

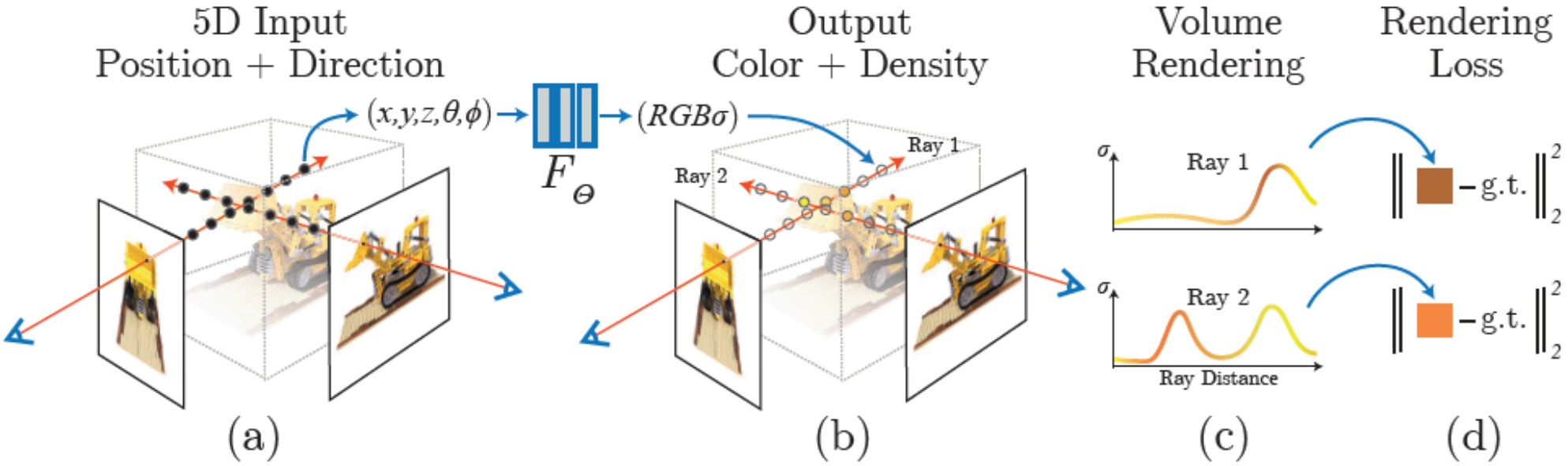


Fig. 2: An overview of our neural radiance field scene representation and differentiable rendering procedure. We synthesize images by sampling 5D coordinates (location and viewing direction) along camera rays (a), feeding those locations into an MLP to produce a color and volume density (b), and using volume rendering techniques to composite these values into an image (c). This rendering function is differentiable, so we can optimize our scene representation by minimizing the residual between synthesized and ground truth observed images (d).

Mildenhall et al, 20

Integrator

Length of interval

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i \,, \text{ where } T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right)$$

Generated in interval

Transmission to eye

Recall:

$$1 - e^{-\sigma\delta} \approx 1 - (1 - \sigma\delta + ...) = \sigma\delta$$

Mildenhall et al, 20

What works: Radiance, pos'n encoding

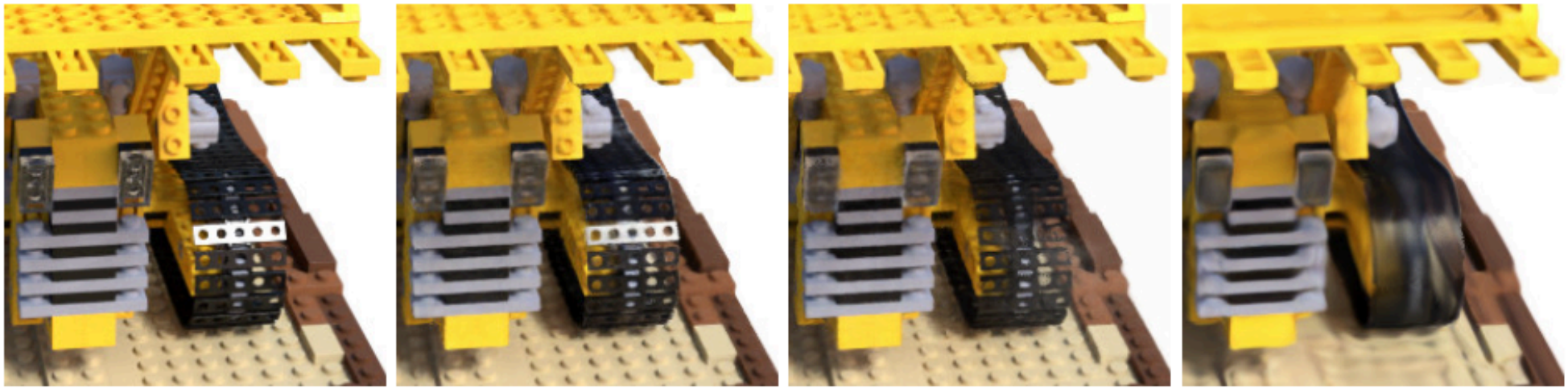


Fig. 4: Here we visualize how our full model benefits from representing view-dependent emitted radiance and from passing our input coordinates through a high-frequency positional encoding. Removing view dependence prevents the model from recreating the specular reflection on the bulldozer tread. Removing the positional encoding drastically decreases the model's ability to represent high frequency geometry and texture, resulting in an oversmoothed appearance.

Mildenhall et al, 20

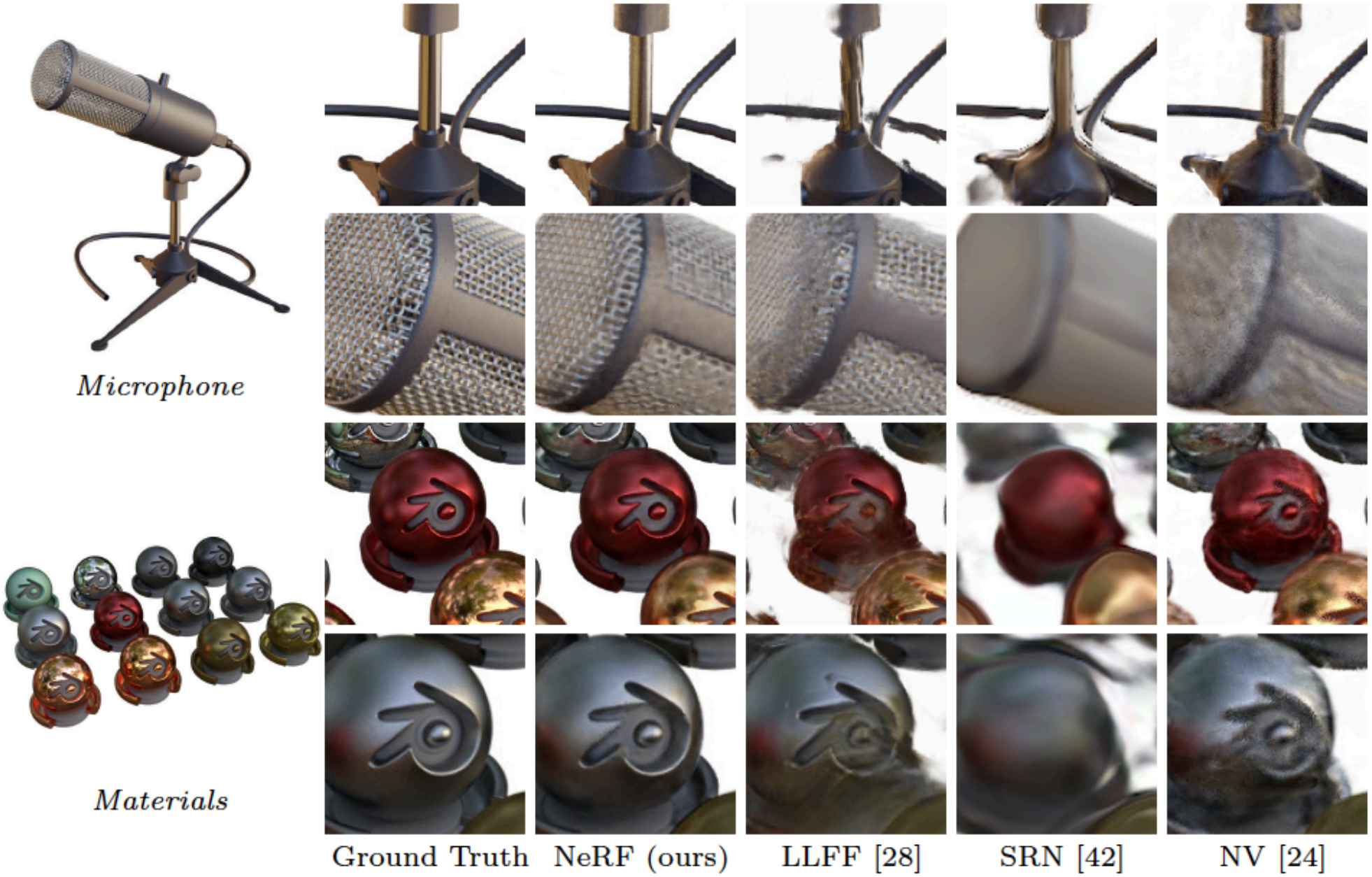
Controlling the integrator

Our rendering strategy of densely evaluating the neural radiance field network at N query points along each camera ray is inefficient: free space and occluded regions that do not contribute to the rendered image are still sampled repeatedly. We draw inspiration from early work in volume rendering [20] and propose a hierarchical representation that increases rendering efficiency by allocating samples proportionally to their expected effect on the final rendering.

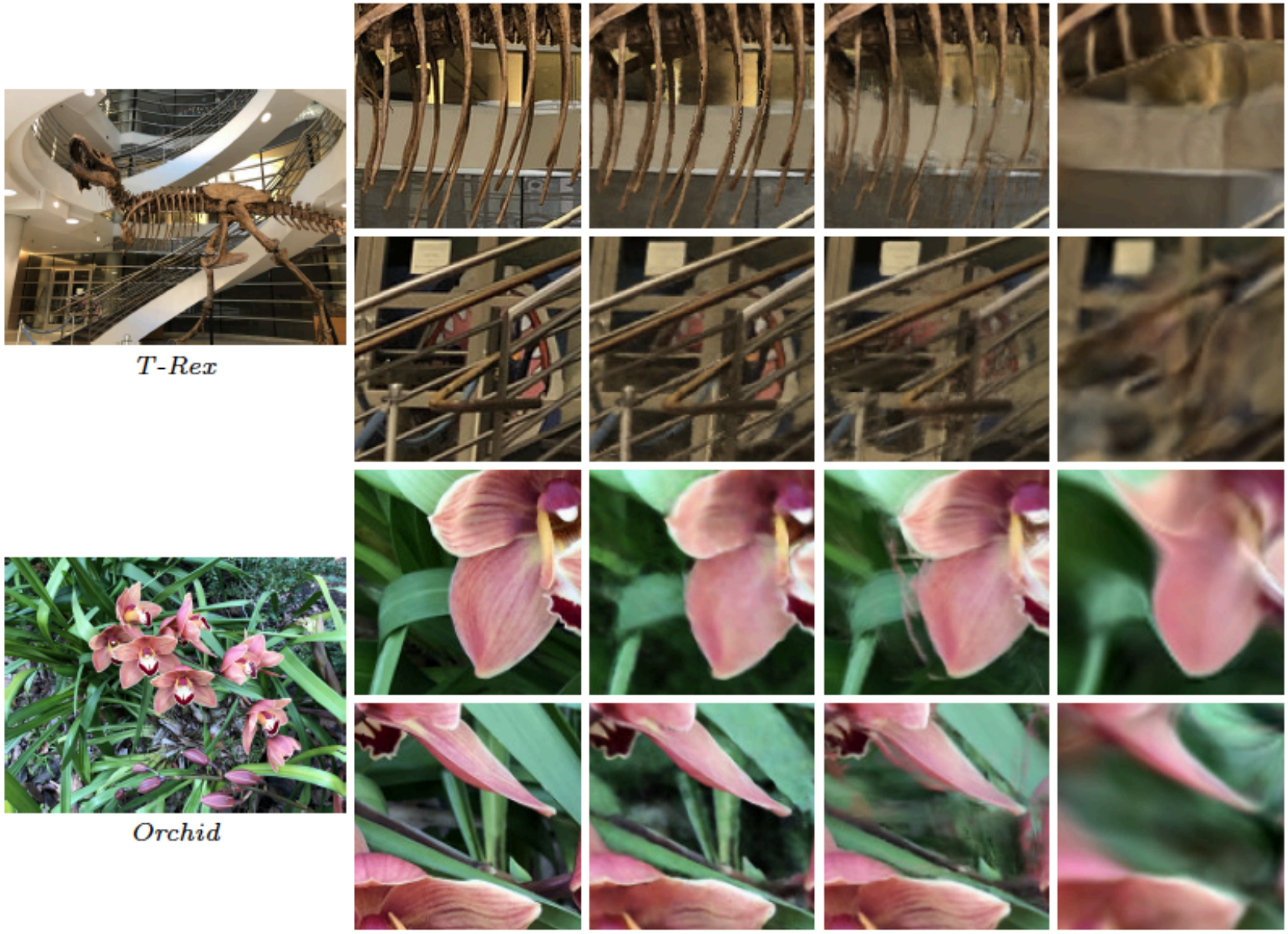
Instead of just using a single network to represent the scene, we simultaneously optimize two networks: one "coarse" and one "fine". We first sample a set of N_c locations using stratified sampling, and evaluate the "coarse" network at these locations as described in Eqns. 2 and 3. Given the output of this "coarse" network, we then produce a more informed sampling of points along each ray where samples are biased towards the relevant parts of the volume. To do this, we first rewrite the alpha composited color from the coarse network $\hat{C}_c(\mathbf{r})$ in Eqn. 3 as a weighted sum of all sampled colors c_i along the ray:

Mildenhall et al, 20

Mildenhall et al, 20



Mildenhall et al, 20



Mildenhall et al, 20

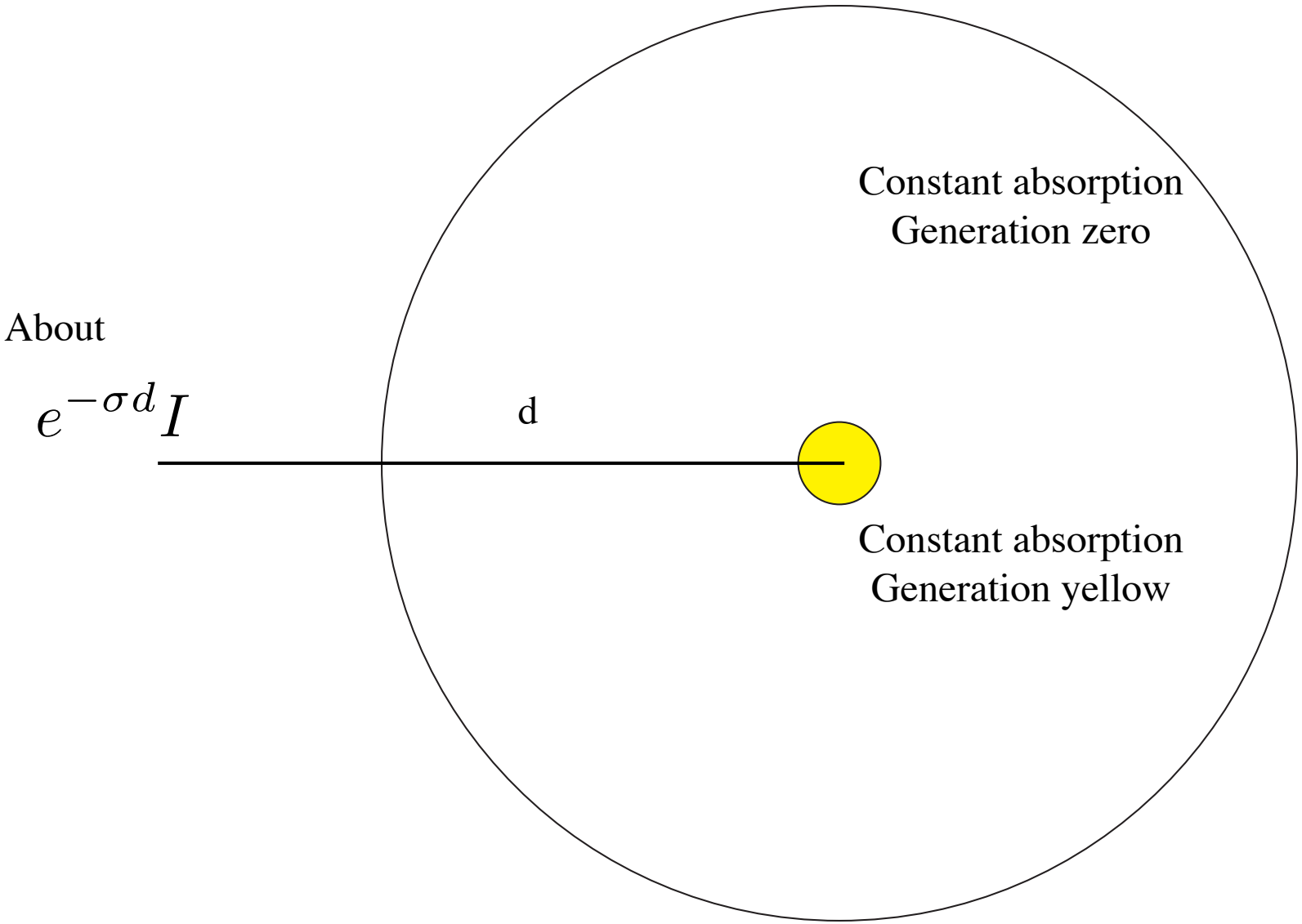
Quiz: what could go wrong?

Quiz: what could go wrong?

- A1:
  - variance in integral estimate makes learning slow
    - symptom is present, diagnosis ?
- A2:
  - different c, sigma give the same images
    - pretty much guaranteed to be true
- A3:
  - good representations may require many views
    - see above
- A4:
  - for surface objects, c, sigma are very odd functions
    - may also contribute to learning problems
    - what is prior to be?

NERF - worrying issues

- Is the reconstruction unique?
  - Why to worry:
    - If not, test image may not be what you want
- Almost certainly not
  - example on next slide suggests I can construct a NERF
    - large norm
    - near zero image
  - but this might not be a bad thing…


Constant absorption
Generation zero
About
$e^{-\sigma d} I$
d
Constant absorption
Generation yellow

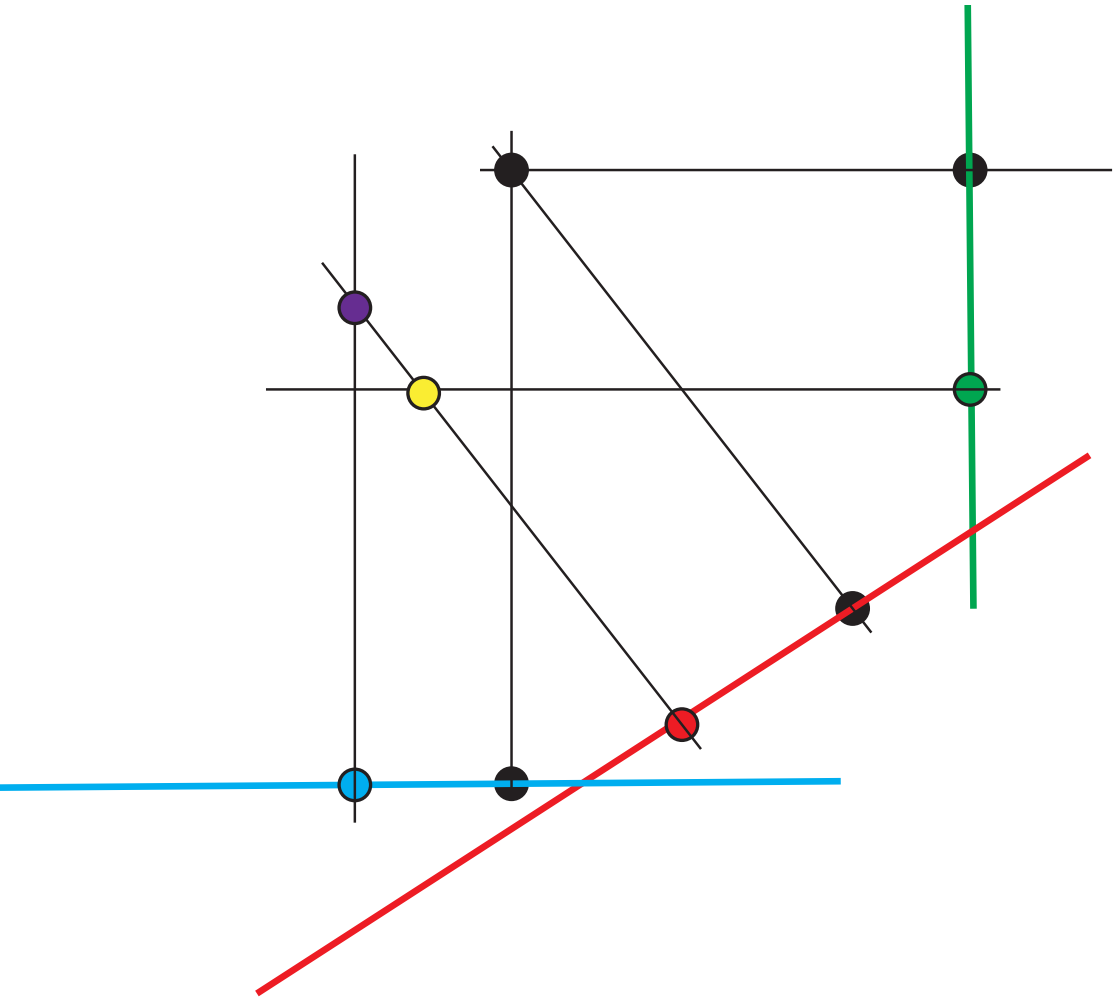
More on uniqueness

- Versions of the NERF repn. are studied in tomography
  - Note if sigma is known, mapping from C to I is linear
  - If sigma is known constant, then it's injective
    - i.e. an infinite set of images has a unique C
  - If sigma isn't known, and depends on angle, not much is known
    - pretty clearly mapping (sigma, c) -> images isn't injective

$$I(0) = \int_0^T \mathbf{c}(\mathbf{x}(s))\sigma(s)e^{-\int_0^s \sigma(u)du} ds$$

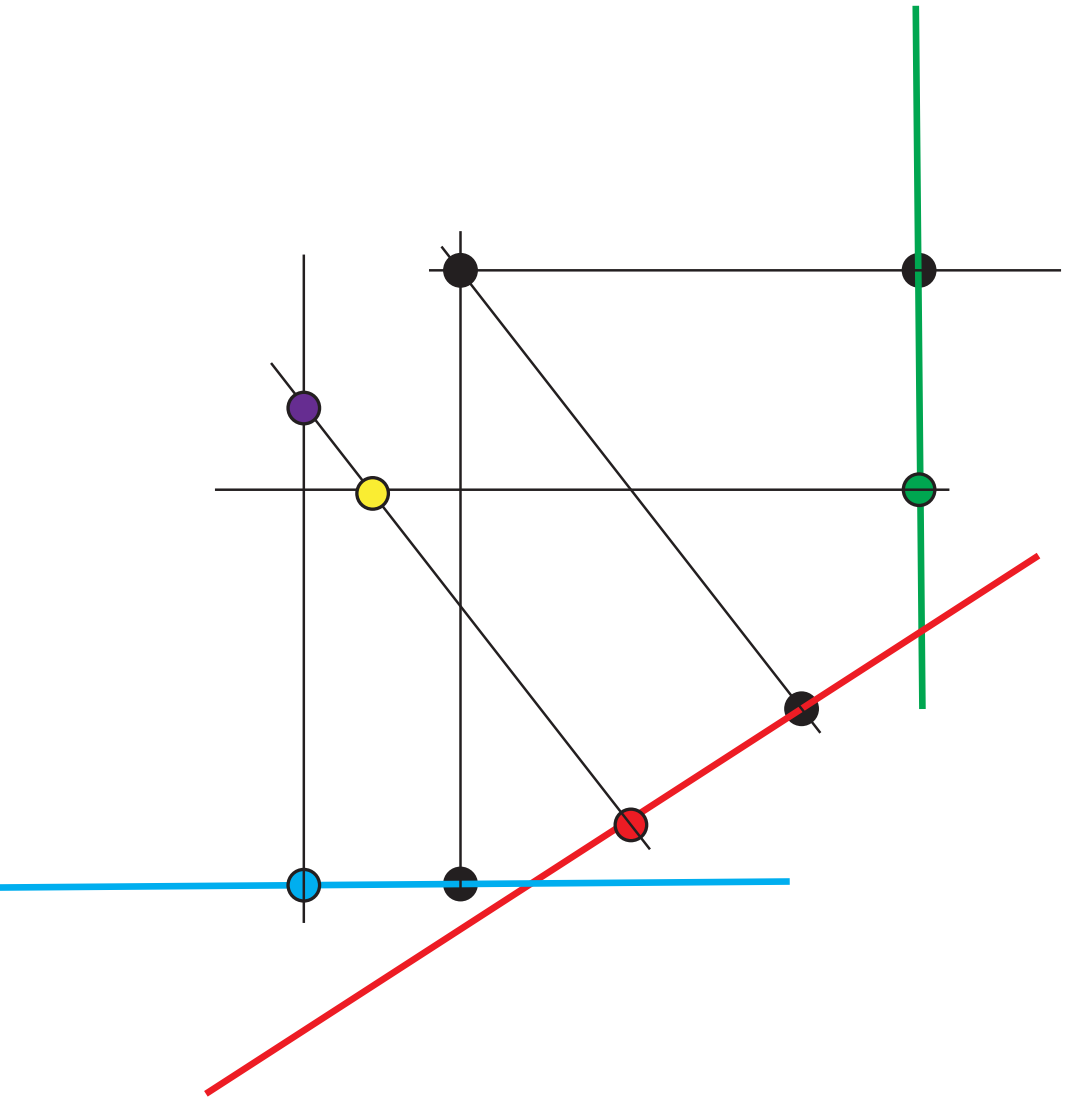
Why this might not be a bad thing

- Don’t need a reconstruction
  - need to predict renderings
- Failures of uniqueness as in picture are irrelevant
  - basic point is you can’t see them
- It may be possible to fudge localization difficulties
  - in a useful way



- Three orthographic cameras see two points
  - Black point correctly localized
  - Other is not
    - B, R reconstruct purple point
    - R, G reconstruct yellow point
  - What to do?
    - traditional:
      - find a point that minimizes least square reprojection error
    - NERF (?):
      - put together a density in triangle that “behaves well”

What is a well behaved density…



- Looks like a point
  - ie localized opacity, occlusion
  - from each intermediate view
- Could it exist?
  - yes
    - C and sigma depend on
      - position and direction