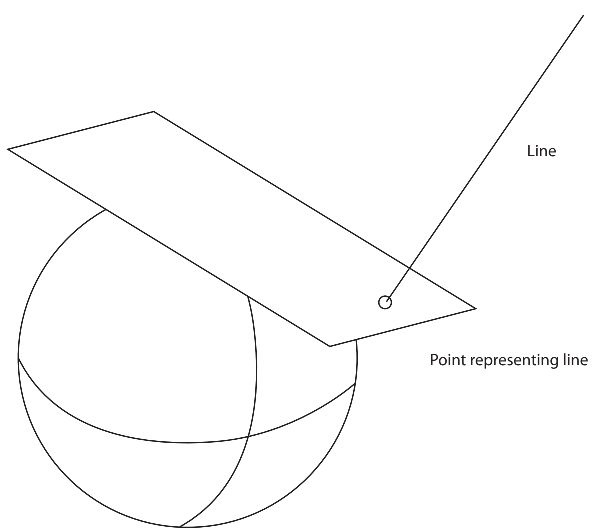
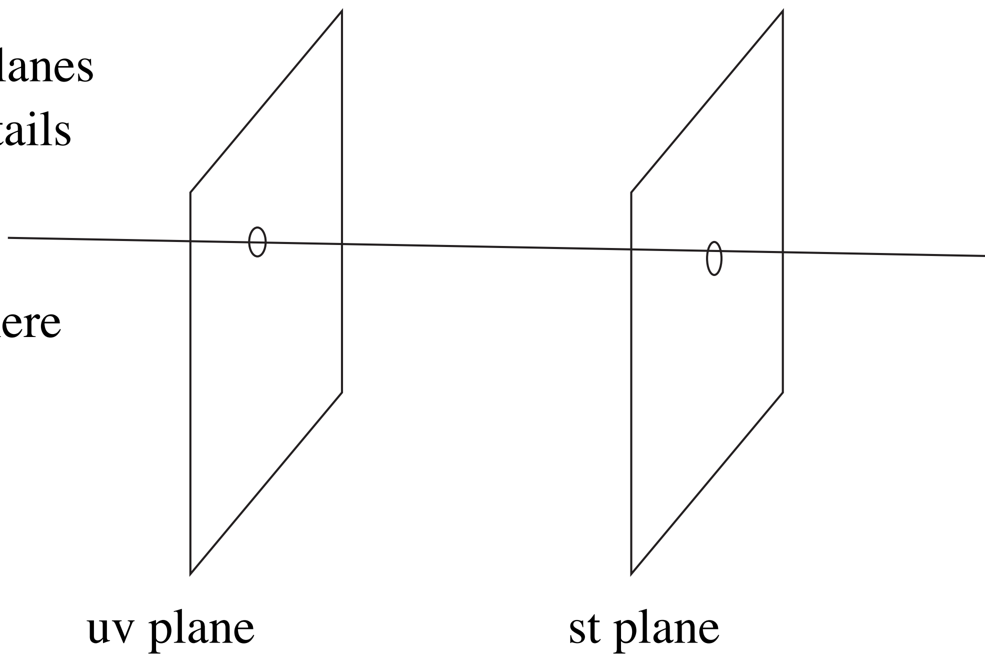
Heading into NERF

D.A. Forsyth, UIUC

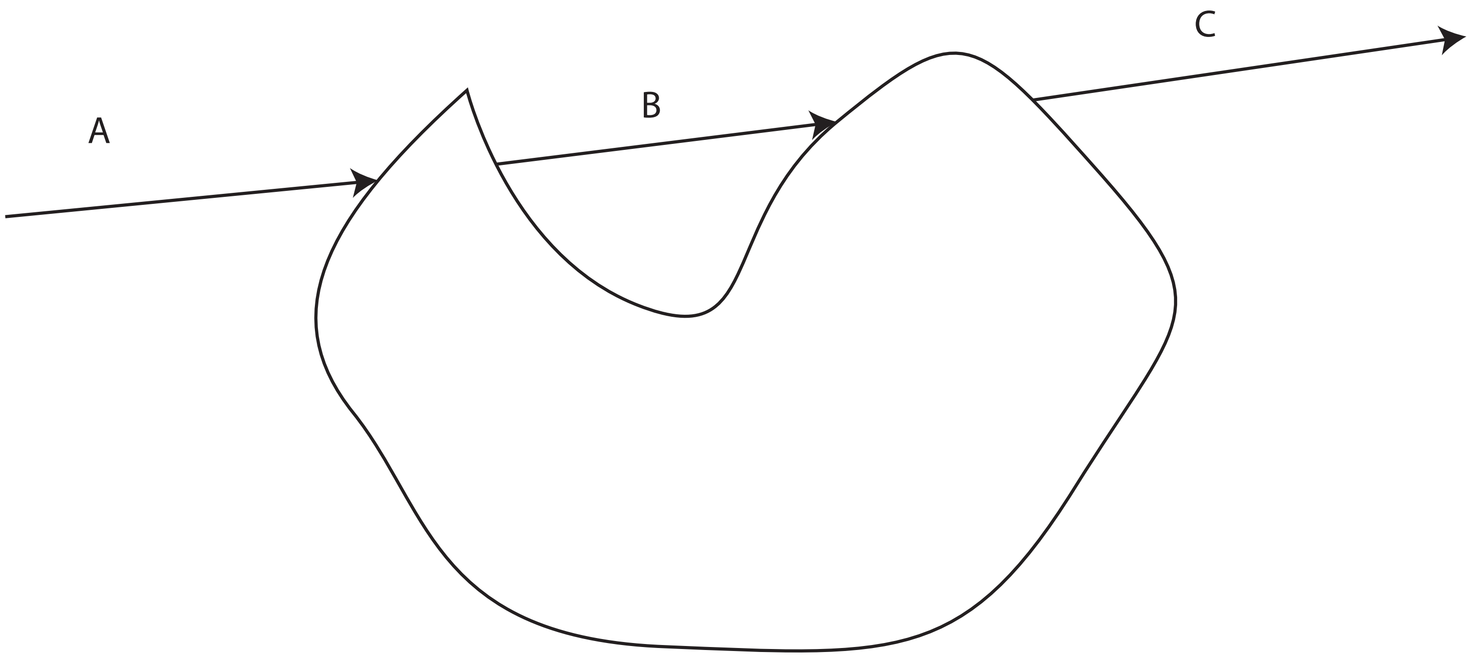
Lines in 3D (if it's empty!)

- 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



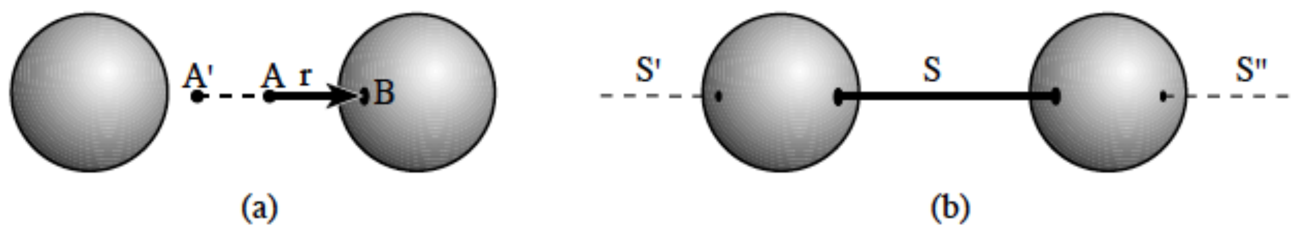
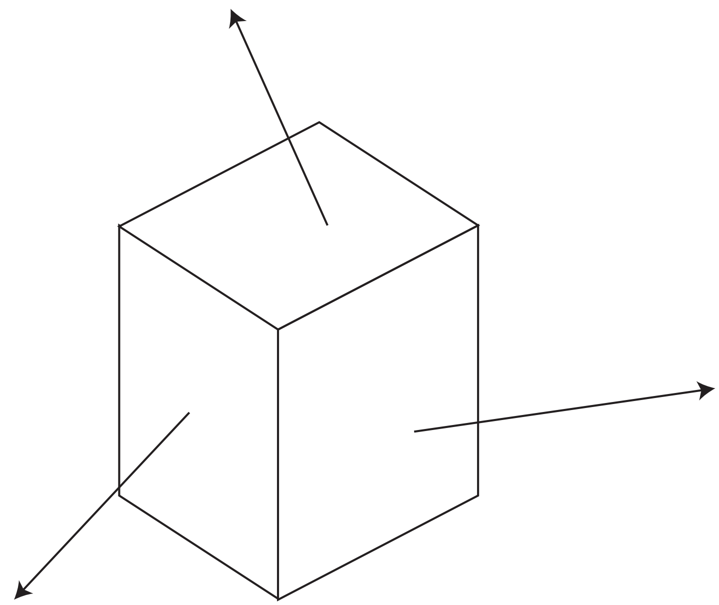


Fig. 1. Maximal free segment. (a) All the rays collinear to r whose origin is between the two spheres "see" point B. (b) These rays are grouped into a *maximal free segment* S. Two other maximal free segments S' and S'' are collinear to S.

Durand et al 02

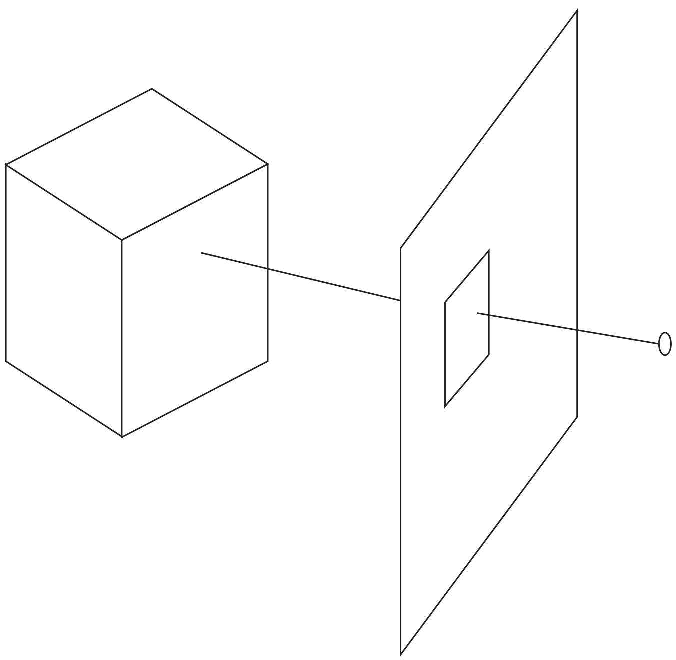
Simplify

- 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



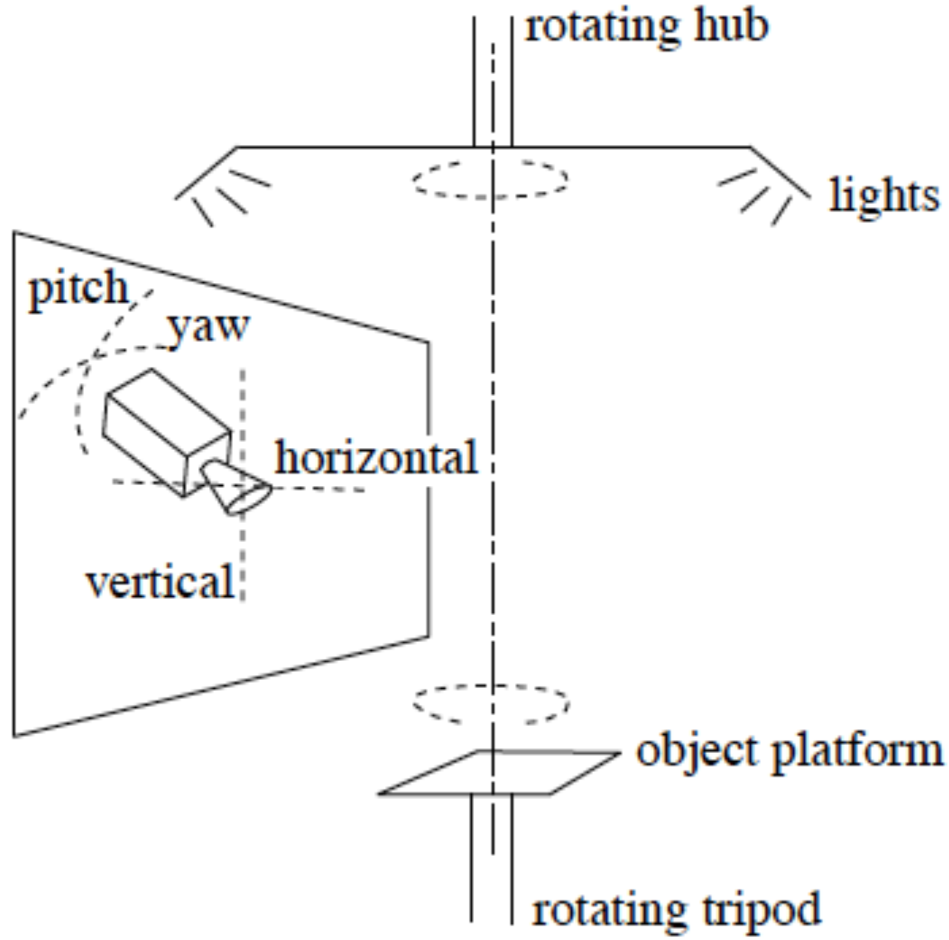


Figure 10: Object and lighting support. Objects are mounted on a Bogen fluid-head tripod, which we manually rotate to four orientations spaced 90 degrees apart. Illumination is provided by two 600W Lowell Omni spotlights attached to a ceiling-mounted rotating hub that is aligned with the rotation axis 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

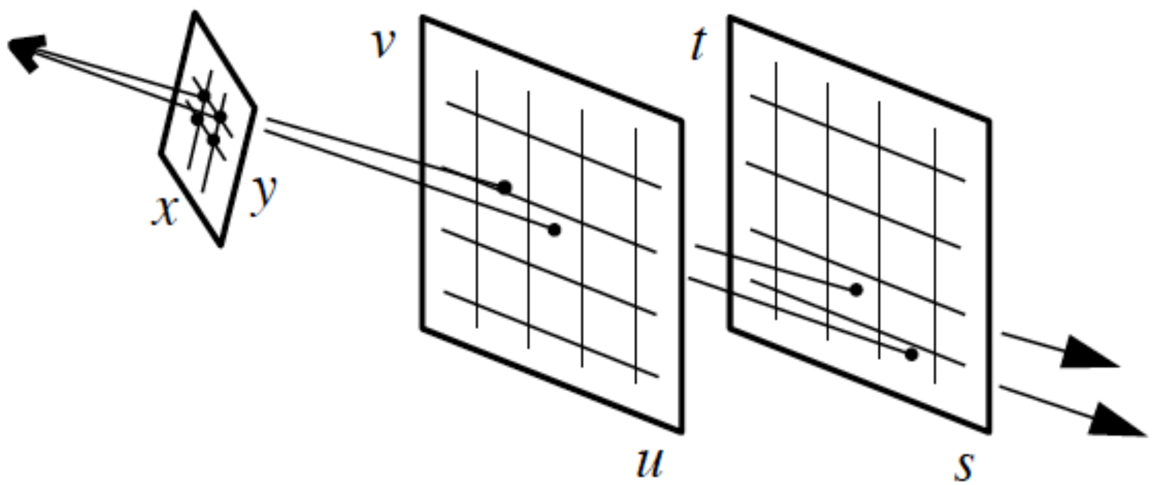


Figure 12: The process of resampling a light slab during display.

Levoy+Hanrahan, 96

Issue: Sampling and Interpolation

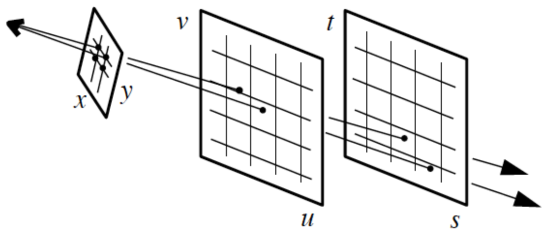


Figure 12: The process of resampling a light slab during display.

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

Interpolation helps, but..

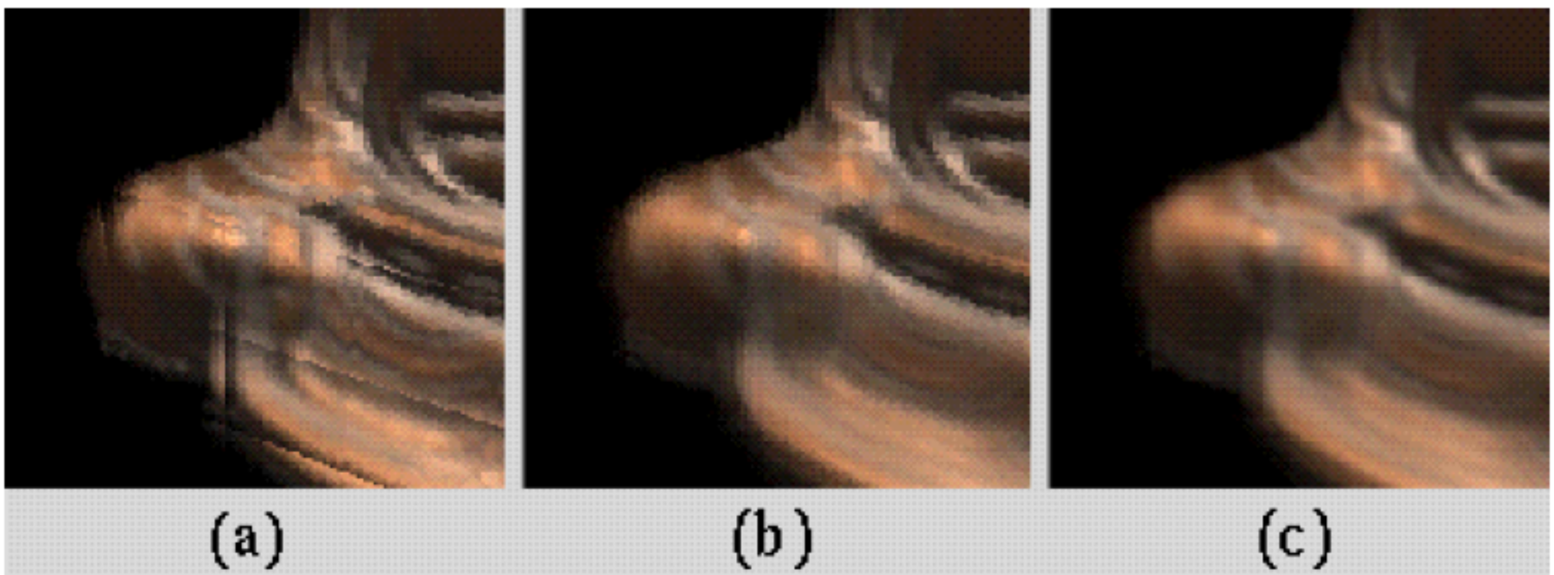
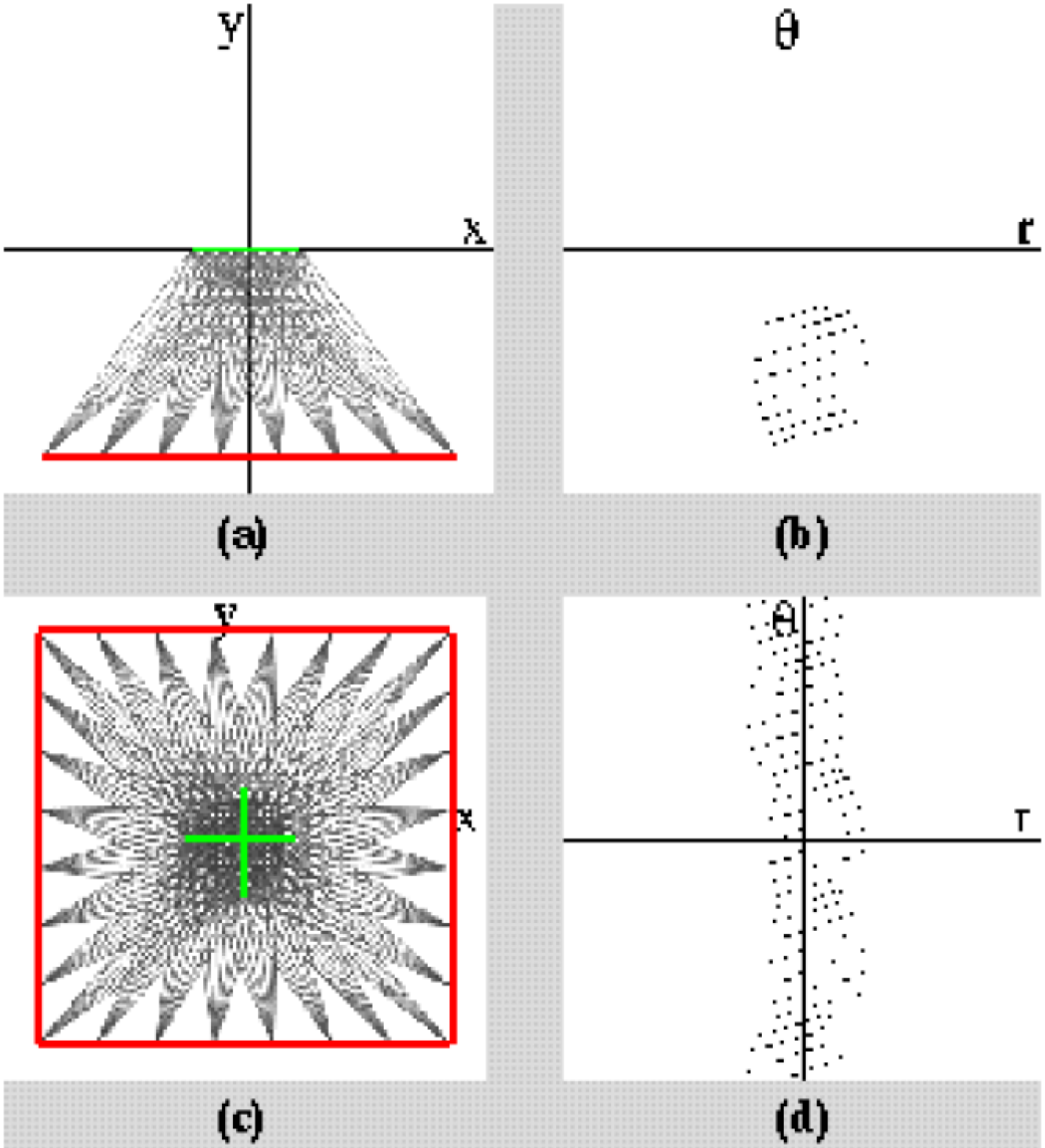


Figure 13: The effects of interpolation during slice extraction. (a) No interpolation. (b) Linear interpolation in uv only. (c) Quadralinear interpolation in uvst.



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

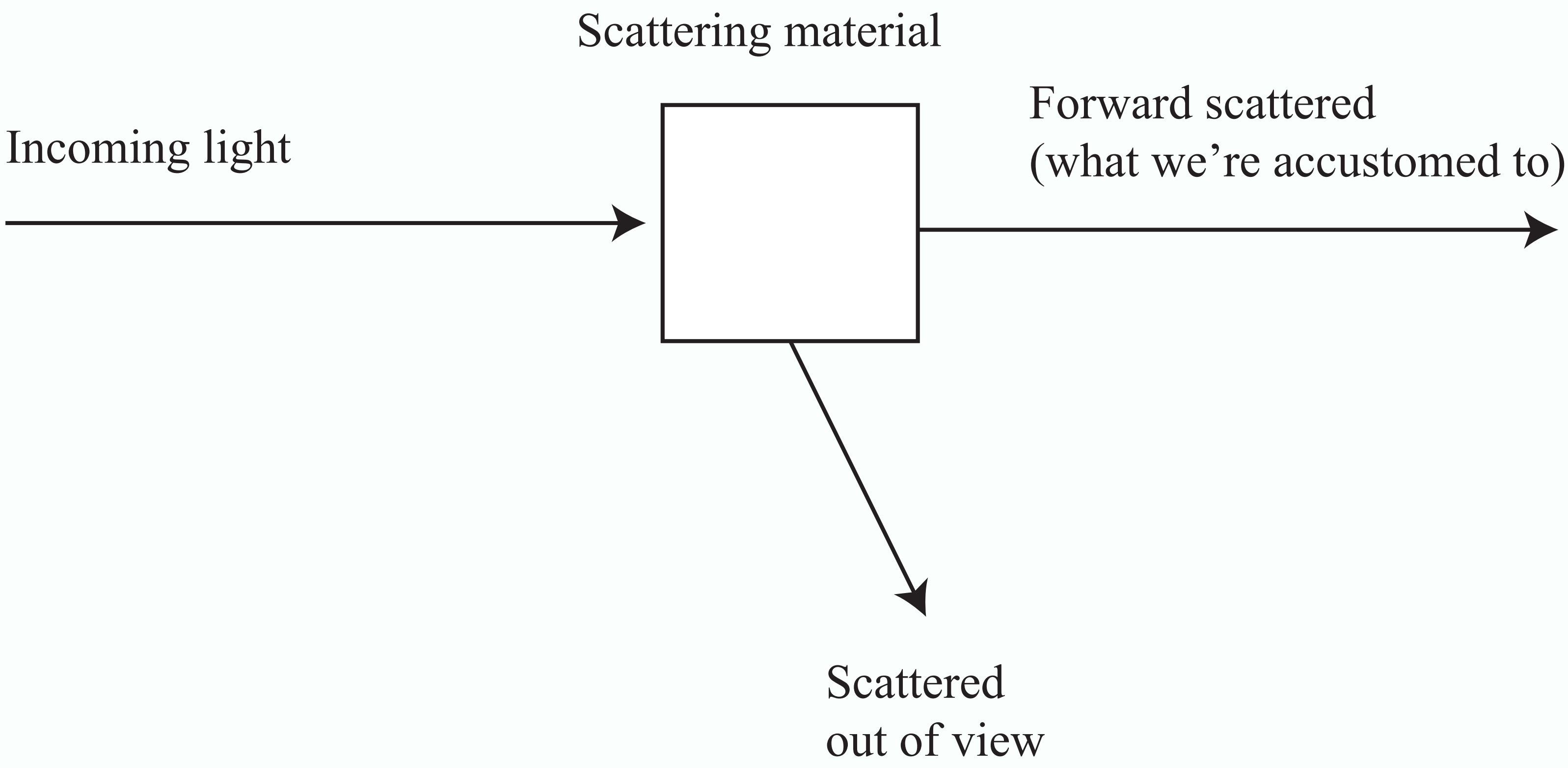
Scattering

- 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

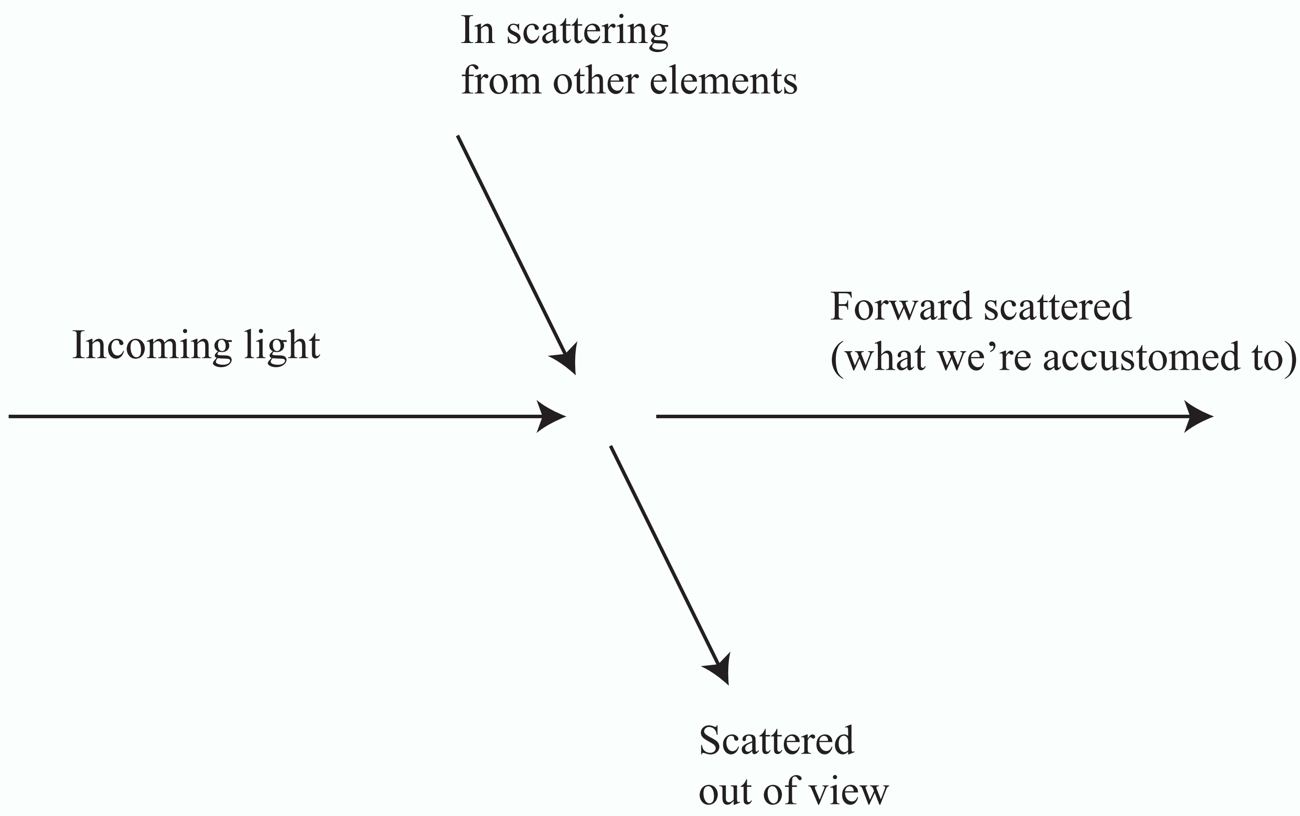
Participating media

- 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

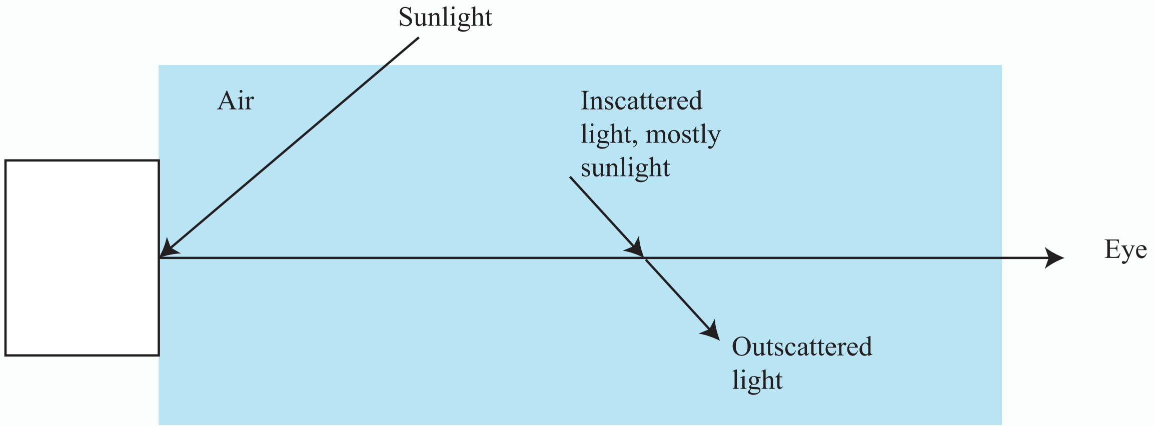


A ray passing through scattering material

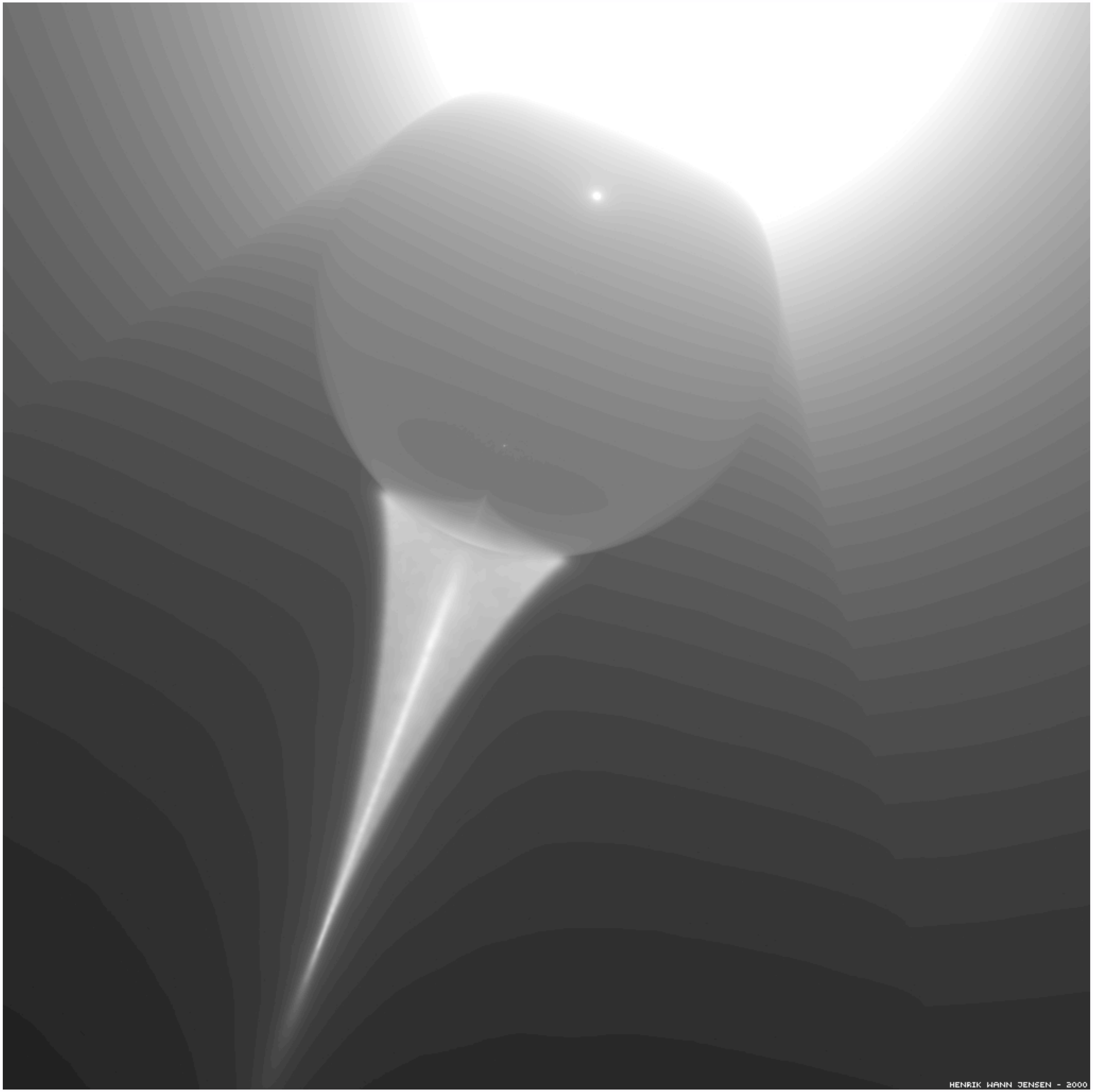


From Lynch and Livingstone, Color and Light in Nature

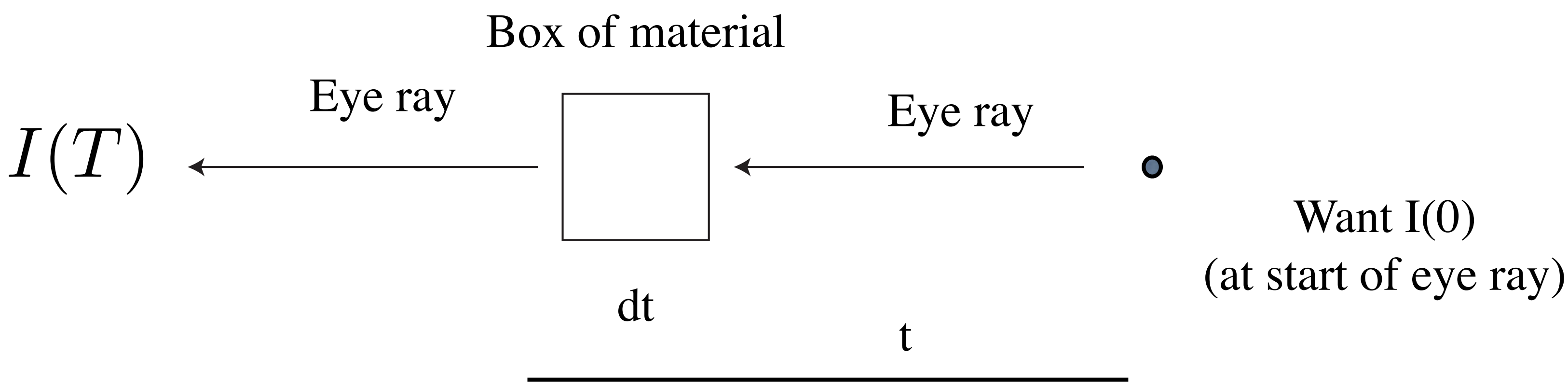
Airlight as a scattering effect



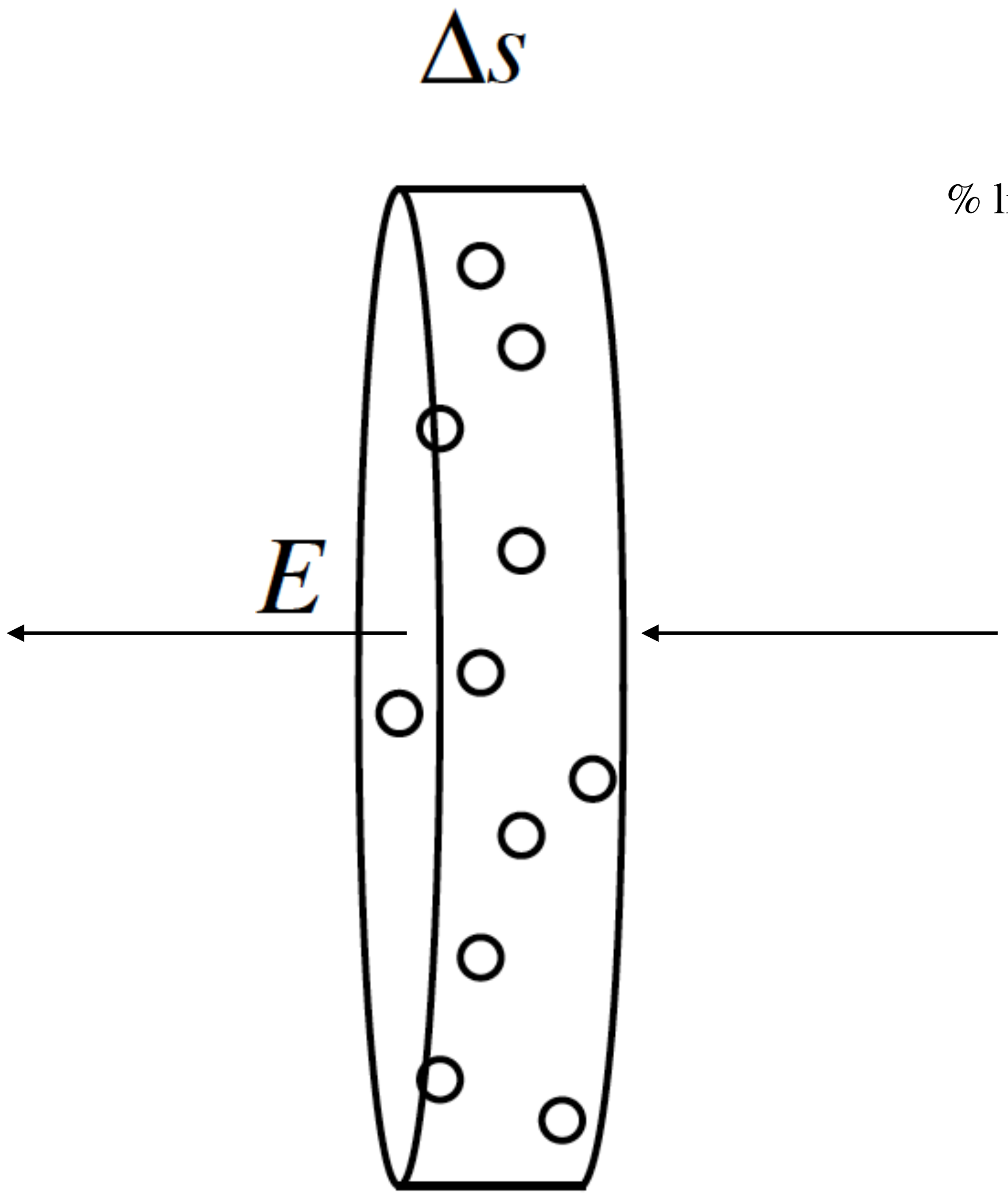
From Lynch and Livingstone, Color and Light in Nature


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?



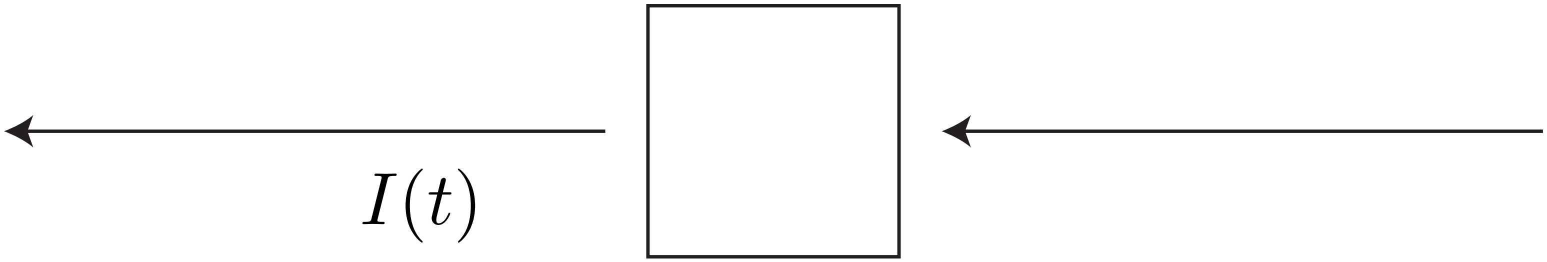
Cross sectional area of "slab" is E
Contains particles, radius r, density rho

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

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

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

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

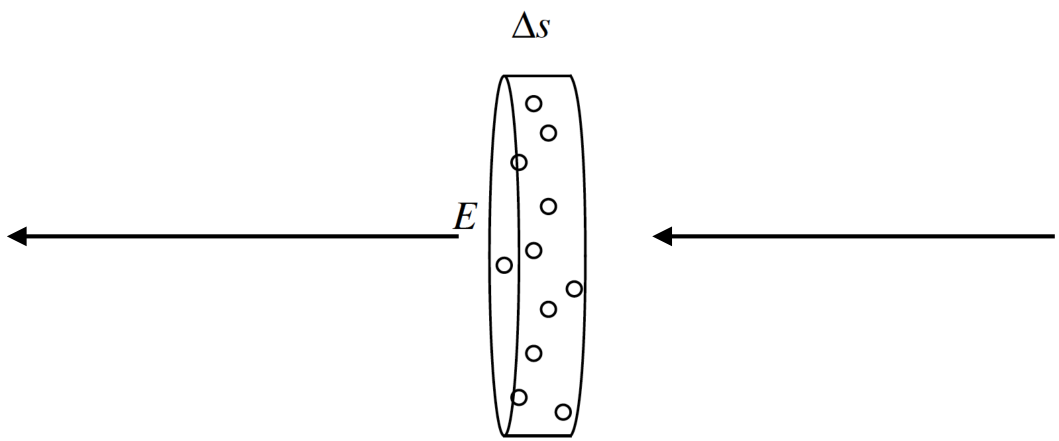
$$I(0) = I(T) e^{-\int_0^T \sigma(t) dt}$$

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

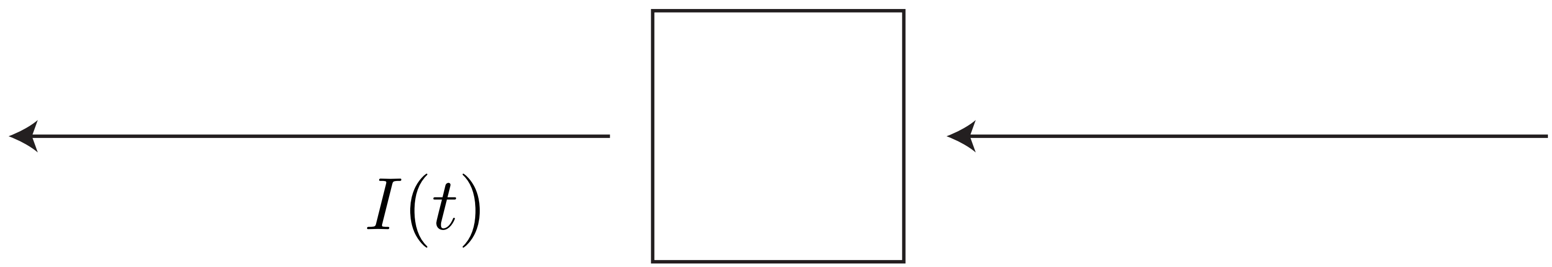
Too few to overlap when projected

Light out = Light in -
Light absorbed+
Light generated

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

which is

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



$$I(t - \delta t) = I(t) - \underbrace{\sigma(t) I(t) \delta t}_{\text{Absorption}} + \underbrace{\mathbf{c}(\mathbf{x}(t)) \sigma(t) \delta t}_{\text{Generation}}$$

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

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

Made at s

Absorbed in transit
from s to 0

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

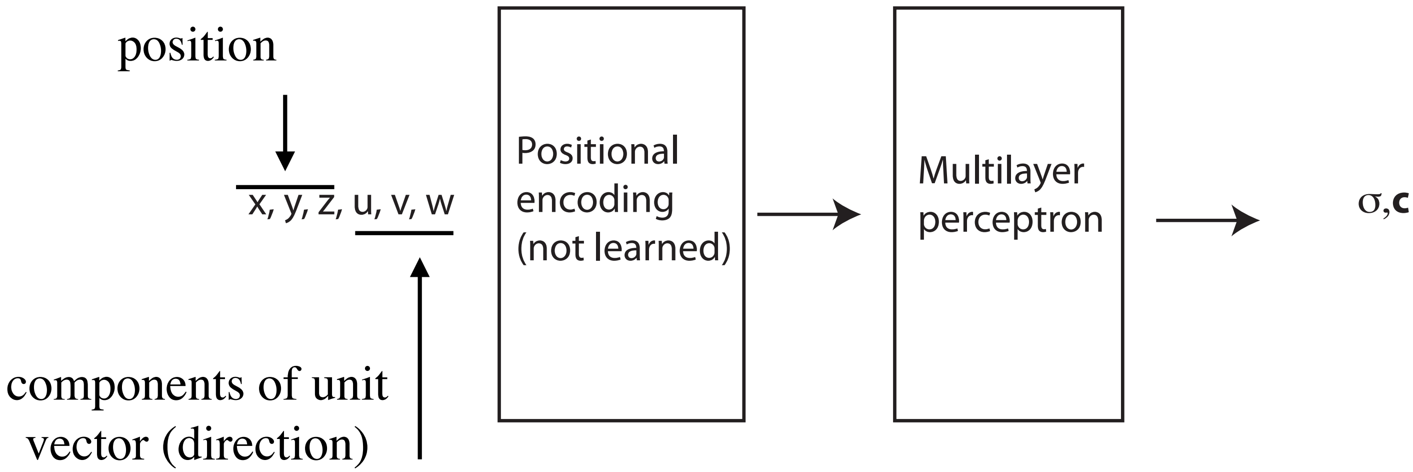
Actual rendering…

- 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

NERF representations

- Build neural network to predict
  - c, sigma as functions of position, direction, parameters
- Render this object with a volume renderer
  - to make images
- Learn this object by
  - adjusting parameters (gradient descent etc)
    - so that images it produces, with renderer are the same as
      - known images
      - geometrically calibrated to one another

Positional encoding



We leverage these findings in the context of neural scene representations, and show that reformulating F_Θ as a composition of two functions $F_\Theta = F'_\Theta \circ \gamma$, one learned and one not, significantly improves performance (see Fig. 4 and Table 2). Here γ is a mapping from $\mathbb{R}$ into a higher dimensional space $\mathbb{R}^{2L}$, and F'_Θ is still simply a regular MLP. Formally, the encoding function we use is:

$$\gamma(p) = \left(\sin\left(2^0 \pi p\right), \cos\left(2^0 \pi p\right), \cdots, \sin\left(2^{L-1} \pi p\right), \cos\left(2^{L-1} \pi p\right) \right). \qquad (4)$$

This function $\gamma(\cdot)$ is applied separately to each of the three coordinate values in $\mathbf{x}$ (which are normalized to lie in $[-1, 1]$) and to the three components of the

NERF representations

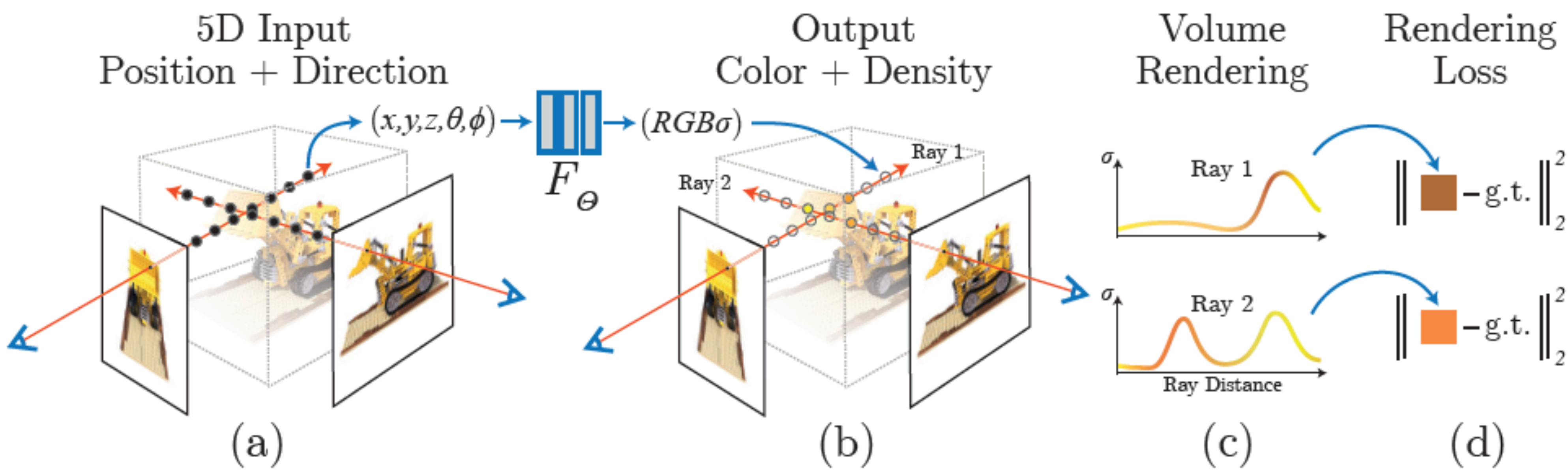


Fig. 2: An overview of our neural radiance field scene representation and differentiable rendering procedure. We synthesize images by sampling 5D coordinates (location and viewing direction) along camera rays (a), feeding those locations into an MLP to produce a color and volume density (b), and using 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

We numerically estimate this continuous integral using quadrature. Deterministic quadrature, which is typically used for rendering discretized voxel grids, would effectively limit our representation's resolution because the MLP would only be queried at a fixed discrete set of locations. Instead, we use a stratified sampling approach where we partition $[t_n, t_f]$ into N evenly-spaced bins and then draw one sample uniformly at random from within each bin:

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\ t_n + \frac{i}{N}(t_f - t_n)\right]. \tag{2}$$

Integrator

Length of interval

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

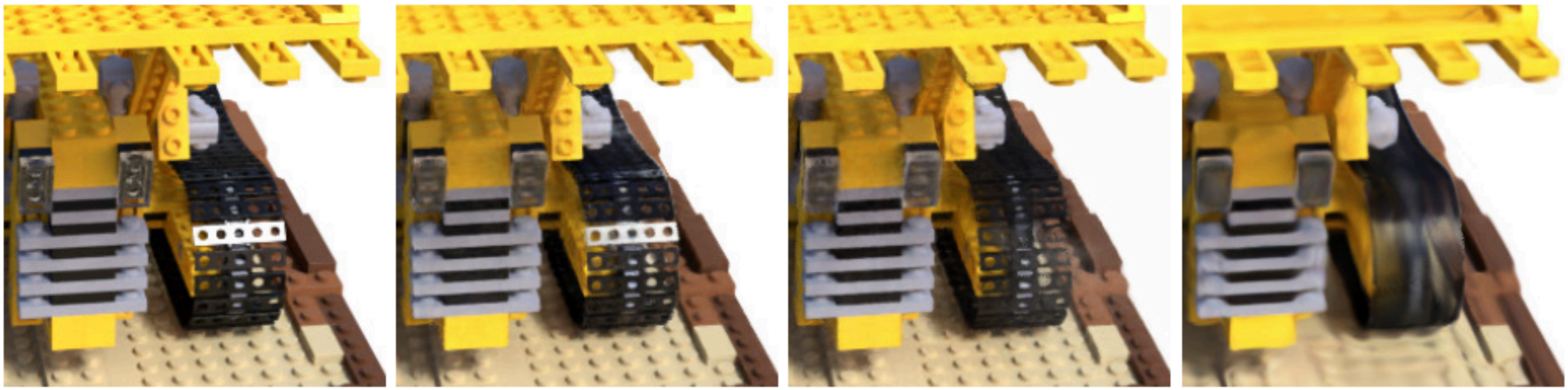
Transmission to eye

Recall:

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

Mildenhall et al, 20

What works: Radiance, pos'n encoding

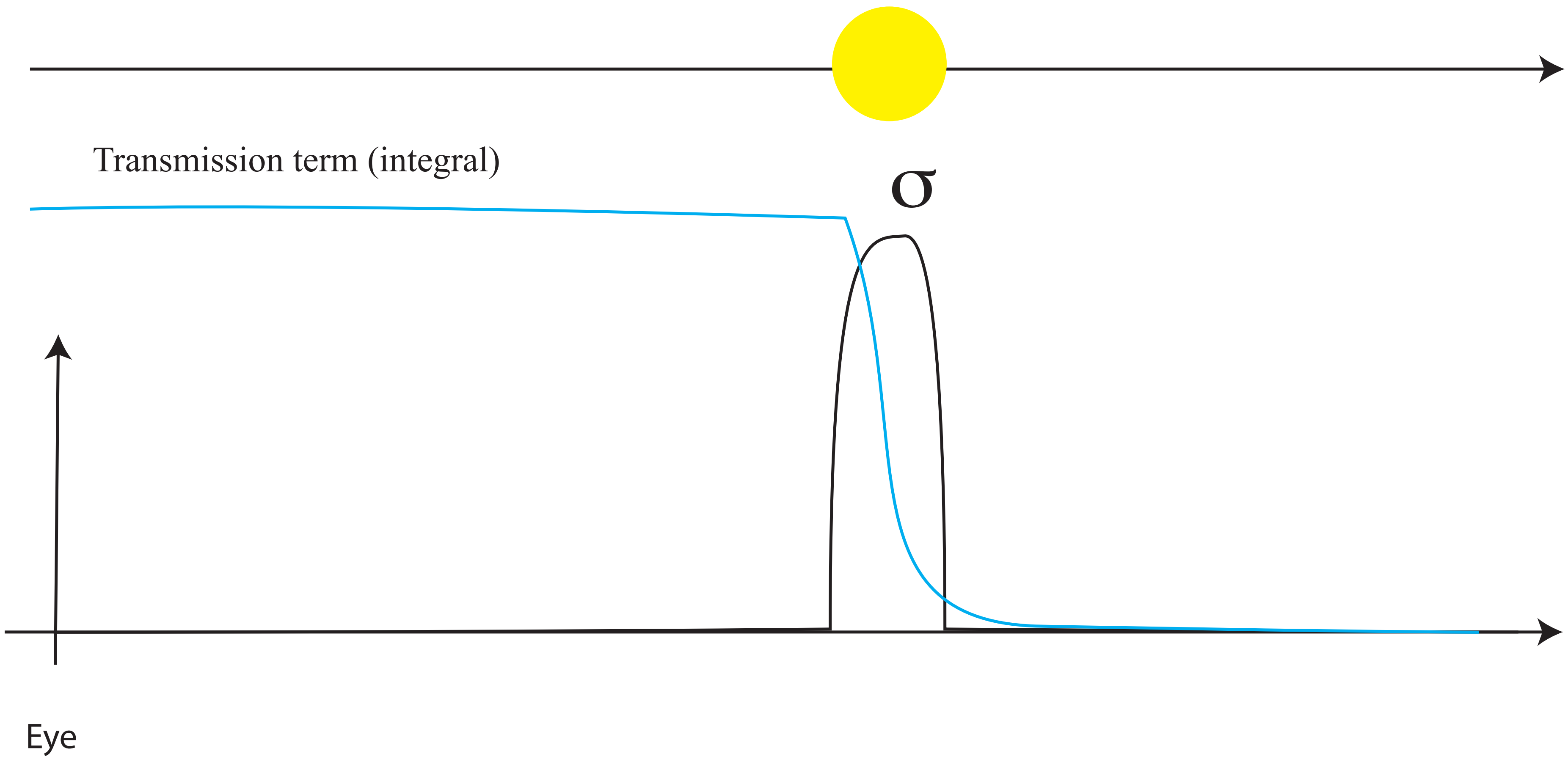


Ground Truth | Complete Model | No View Dependence | No Positional Encoding

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

Integration is hard

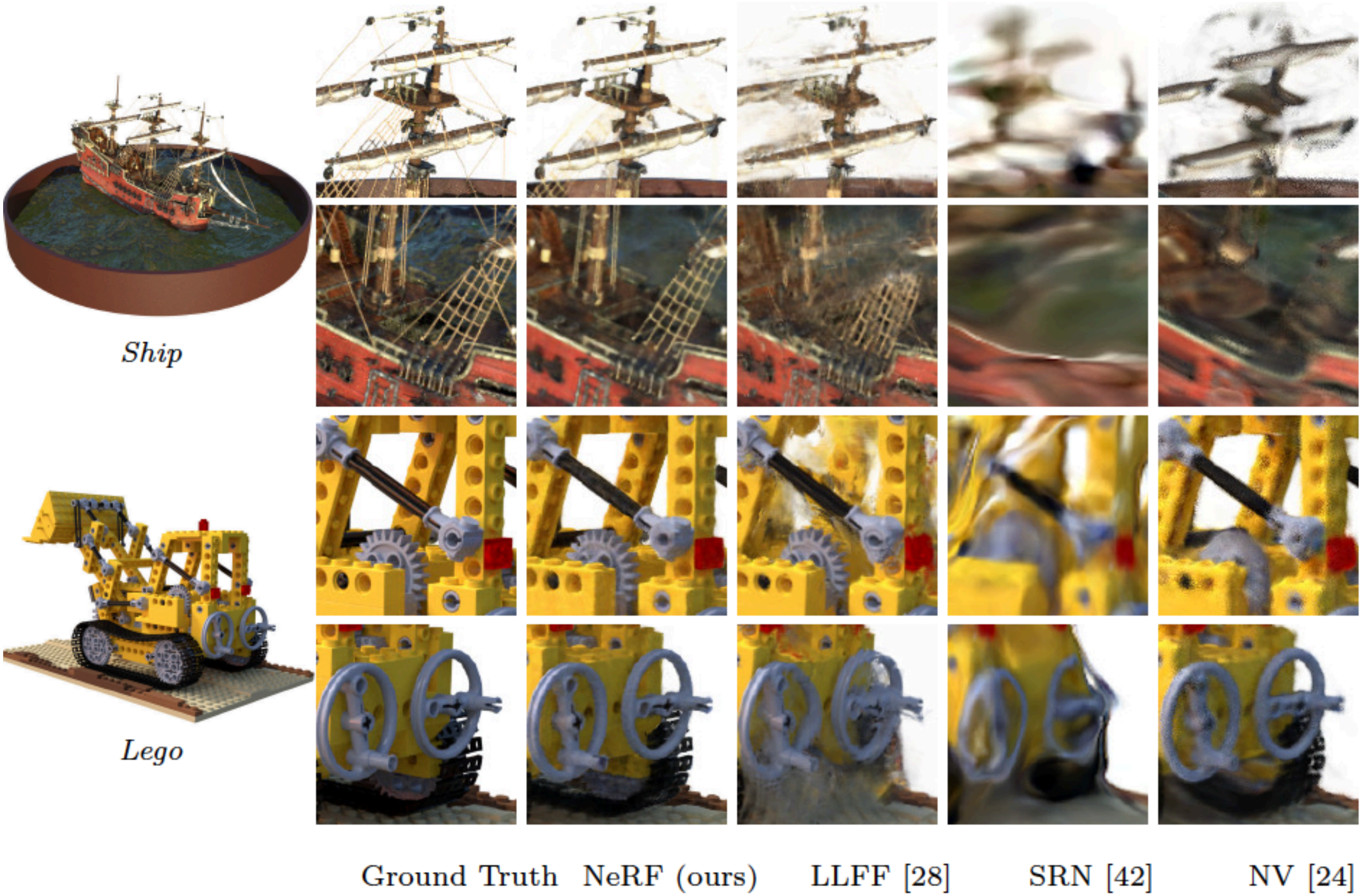


Controlling the integrator

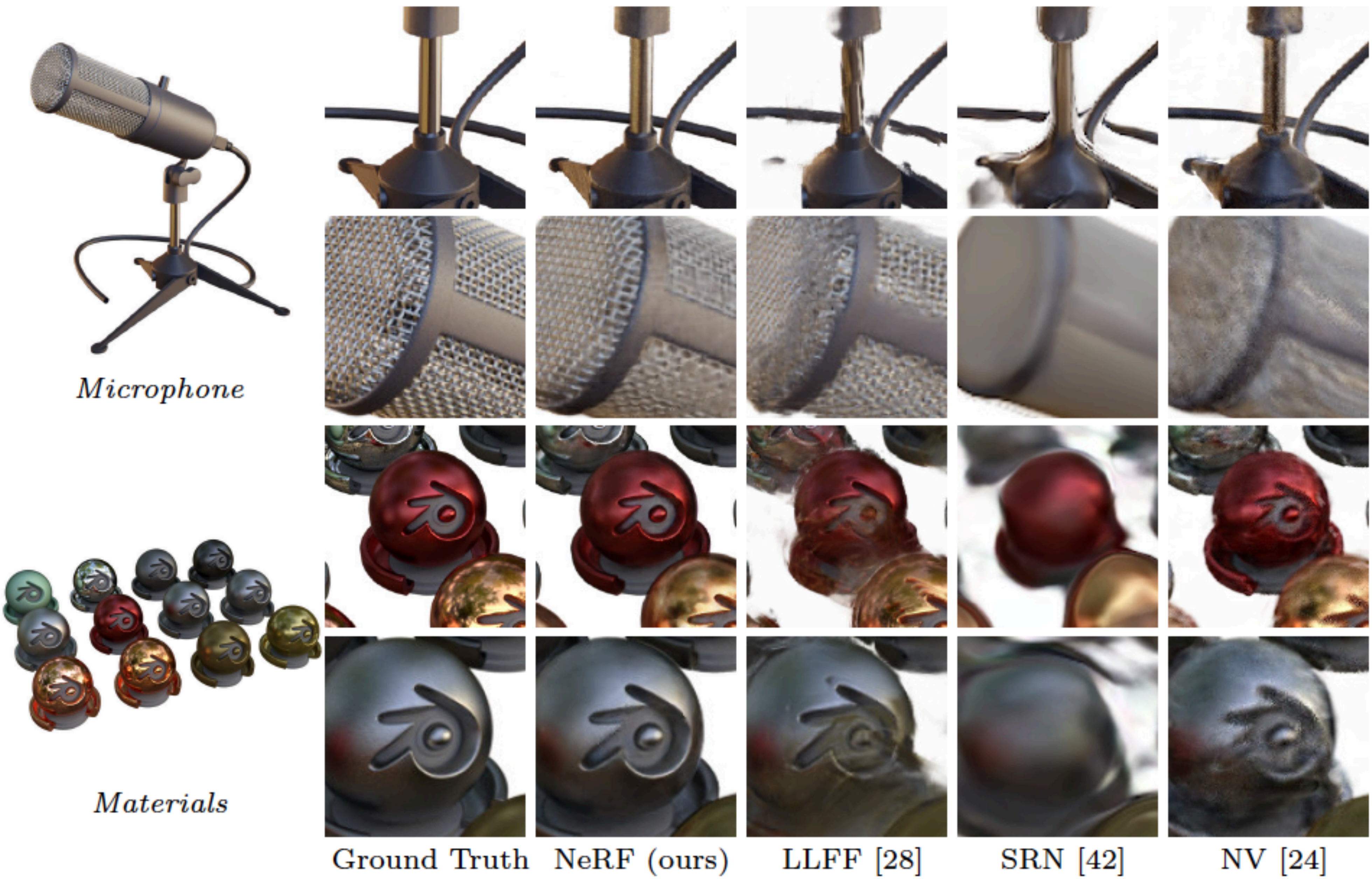
Our rendering strategy of densely evaluating the neural radiance field network at N query points along each camera ray is inefficient: free space and occluded regions that do not contribute to the rendered image are still sampled repeatedly. We draw inspiration from early work in volume rendering [20] 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:

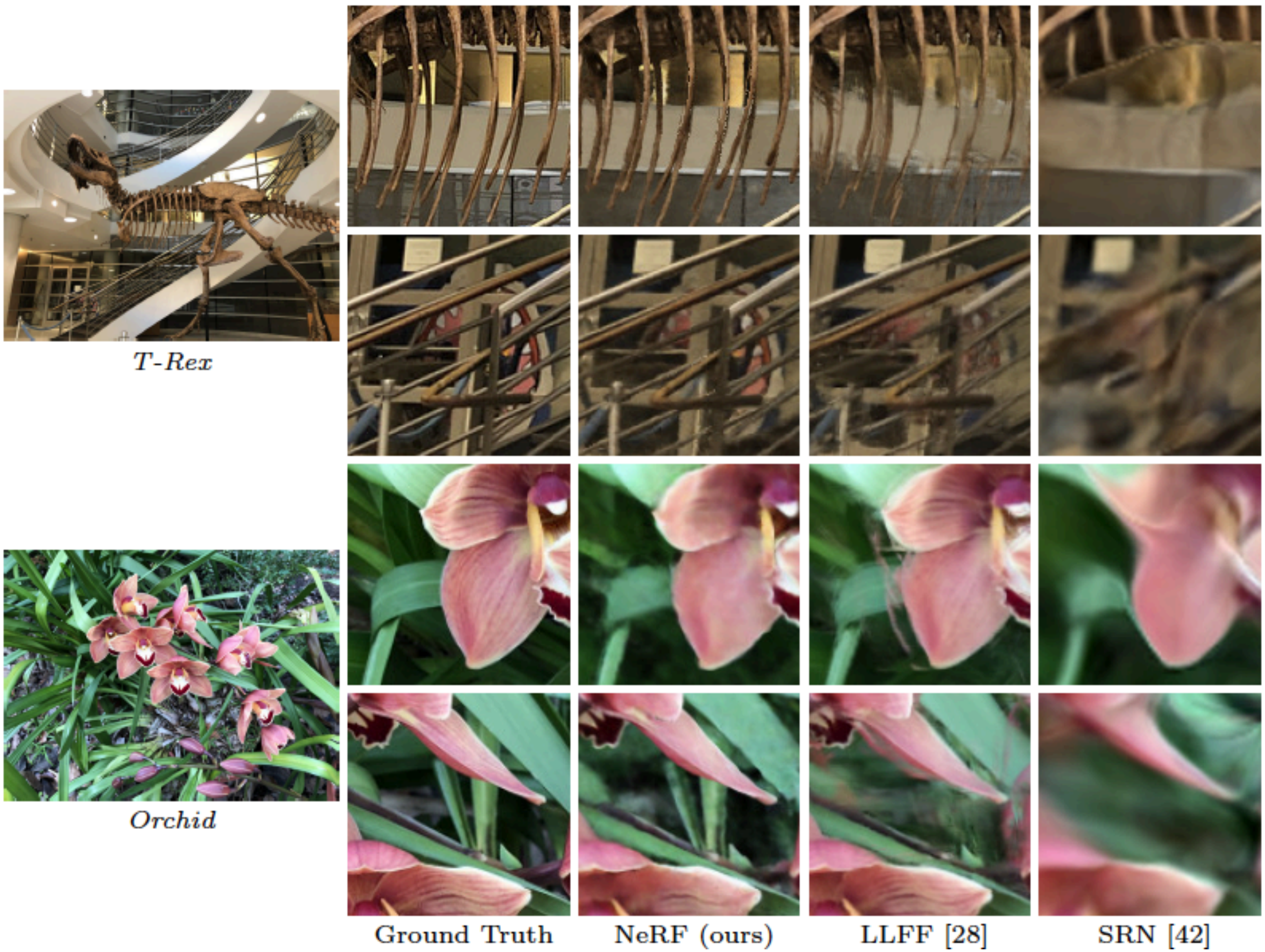
Mildenhall et al, 20



Mildenhall et al, 20



Mildenhall et al, 20



Mildenhall et al, 20

Quiz: what could go wrong?

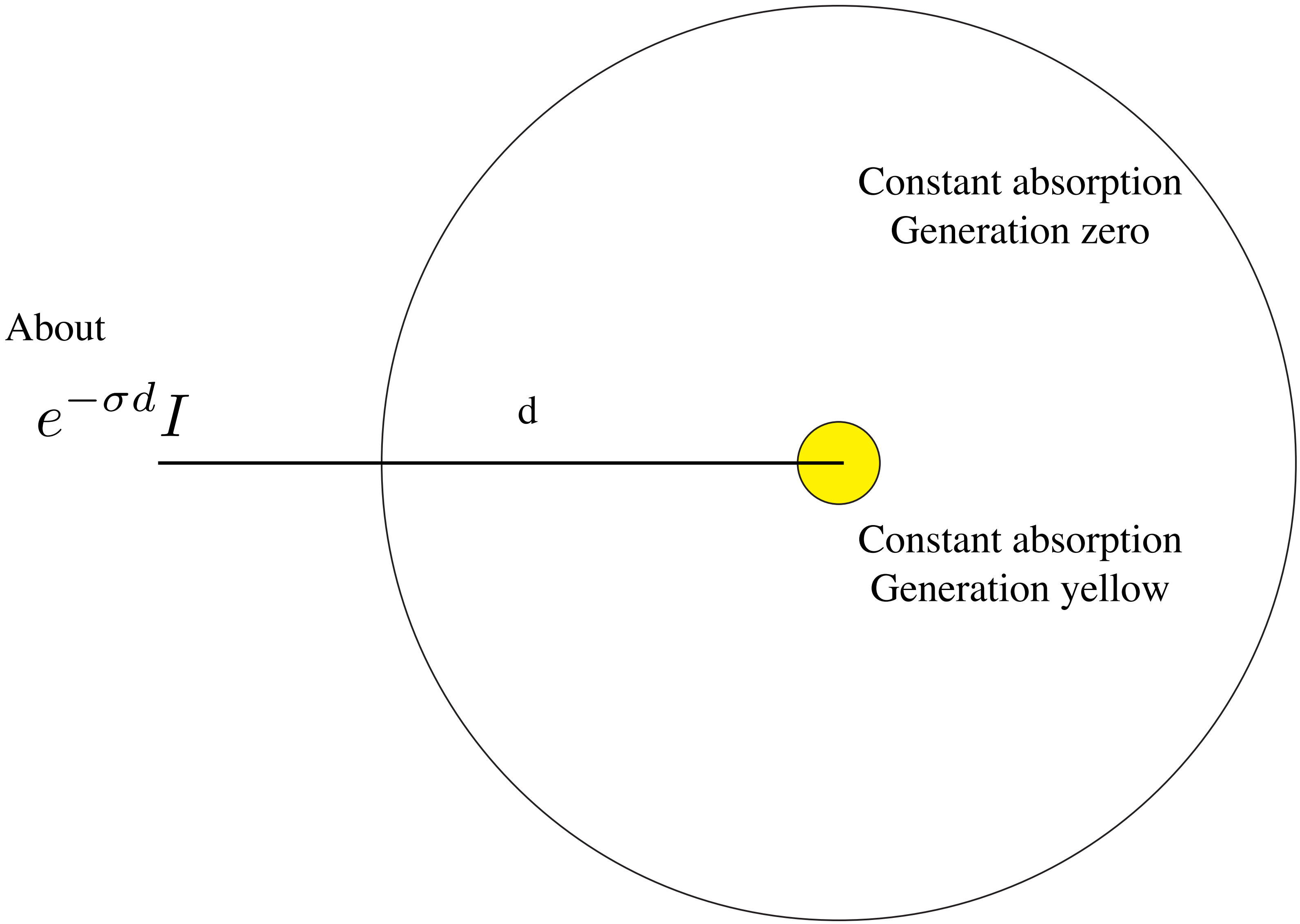
Quiz: what could go wrong?

- 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

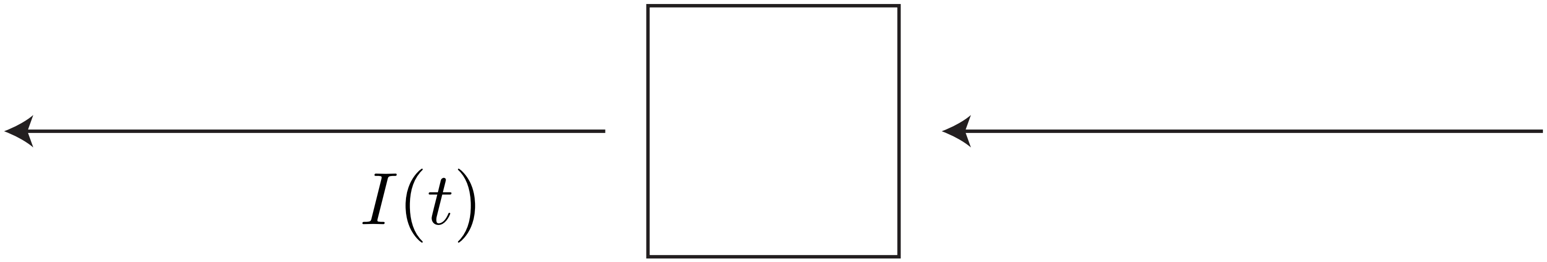
  - but this might not be a bad thing…



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

Tomography (rapid summary!)

- Pass x-rays through the body in many different directions
  - record result
- Reconstruct 3D density from x-rays
  - absorption only - no local generation of light
-



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

Extinction
coefficient

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

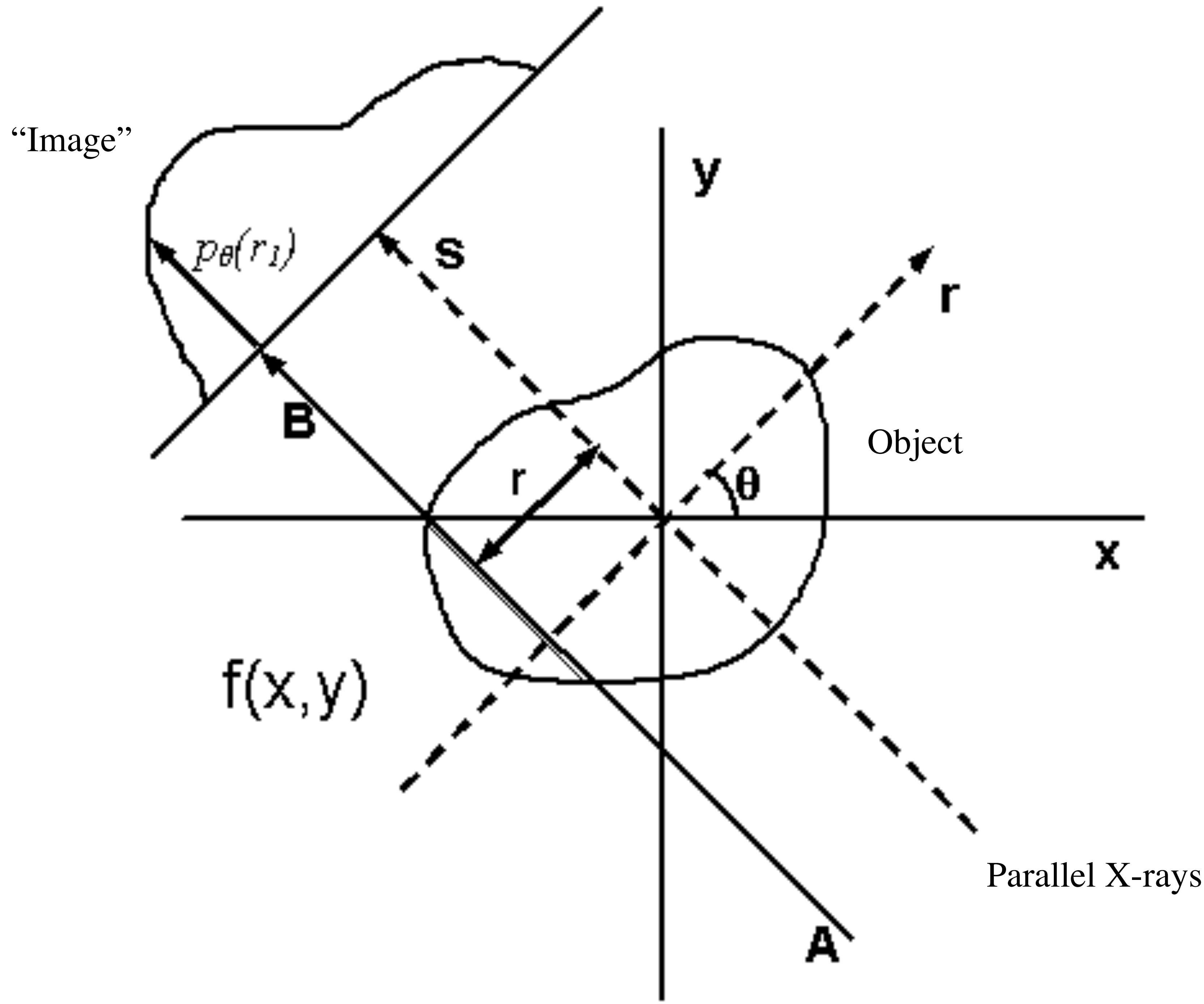
$$\frac{d \log I}{dt} = \sigma(t)$$

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

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

Eye is at 0

Intensity at T


"Image"
y
$p_\theta(r_1)$
s
r
B
Object
r
θ
x
f(x,y)
Parallel X-rays
A

The x-ray image

- We have, for one pixel

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

Eye is at 0

Intensity of x-ray source

$$-\log \frac{I(0)}{I(T)} = \int_0^T \sigma(t(\mathbf{x}))dt$$

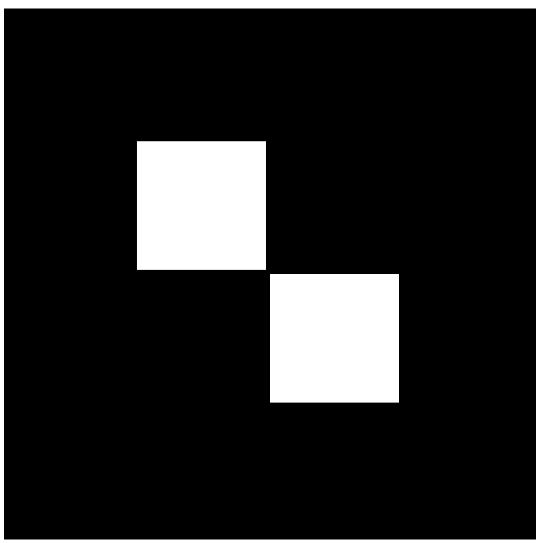
Want

Observe

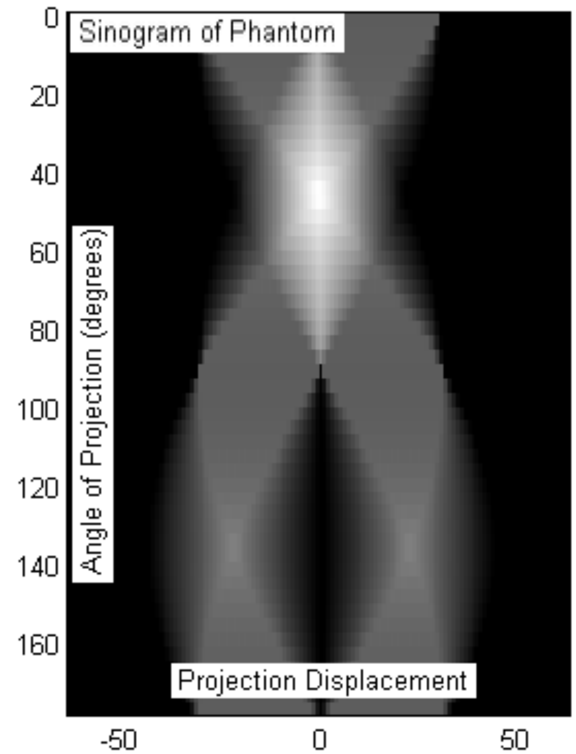
2D example

- Lines (rays) encoded by angle, distance to origin
- Sinogram
  - plot of density observed as a function of angle, distance to origin

Object



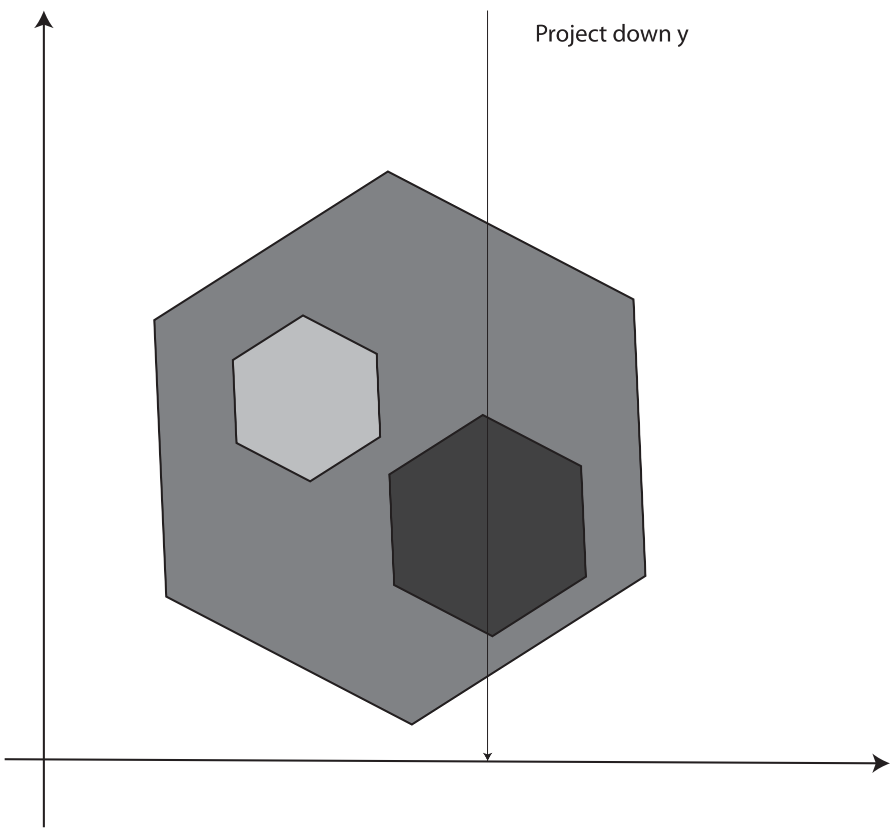
Sinogram



Tomography

- Q: Can you get object from sinogram?
  - yes
    - (sketch follows)
    - important limits from sampling issues
- Q: Is it unique?
  - i.e. is mapping from object to sinogram injective?
  - yes
    - important limits from sampling issues
    - some noise
- Q: what role do deep networks have here?
  - A: largely worked out (see papers)
    - helpful, but should be constrained by sampling theory, etc
      - or else they make fake structures

Roughly why you can reconstruct - I



$$d(x) = \int_{-\infty}^{\infty} \sigma(x, y) dy$$

Observe

Want

Roughly why you can reconstruct - II

$$d(x) = \int_{-\infty}^{\infty} \sigma(x, y) dy$$

Fourier transform of d (which we can compute)

$$\mathcal{F}\left\{d\right\}(\omega_x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} \sigma(x, y) dy\right] e^{-j\omega_x x} dx$$

$$\mathcal{F}\left\{\sigma\right\}(\omega_x, \omega_y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sigma(x, y) e^{-j(\omega_x x + \omega_y y)} dx dy$$

$$\mathcal{F}\left\{d\right\}(\omega_x) = \mathcal{F}\left\{\sigma\right\}(\omega_x, 0)$$

Roughly why you can reconstruct - III

$$\mathcal{F}\left\{d\right\}\left(\omega_x\right) = \mathcal{F}\left\{\sigma\right\}\left(\omega_x, 0\right)$$

Fourier transform of d is a “slice” of Fourier transform of sigma

Each different projection direction yields a different “slice”

- Strategy:
  - collect many different directions (= slices)
  - this yields an approximation to FT of sigma
  - invert FT - you now have sigma
- Issues:
  - sampling
    - we don’t see FT of d exactly, just samples
    - we don’t see FT of sigma exactly, only samples of slices

Various features + complications

- Features
  - uniqueness of FT yields uniqueness of reconstruction
- Complications
  - it's a nuisance to take x-rays orthographically
    - rather use a point source (= perspective; = fan beam)
    - mild mathematical complications follow
    - practical complications follow - more samples, etc.
- This *ISN'T* the NeRF reconstruction problem
  - no internal source
    - SPECT - single photon emission computed tomography
      - swallow some radiating material, then get imaged!

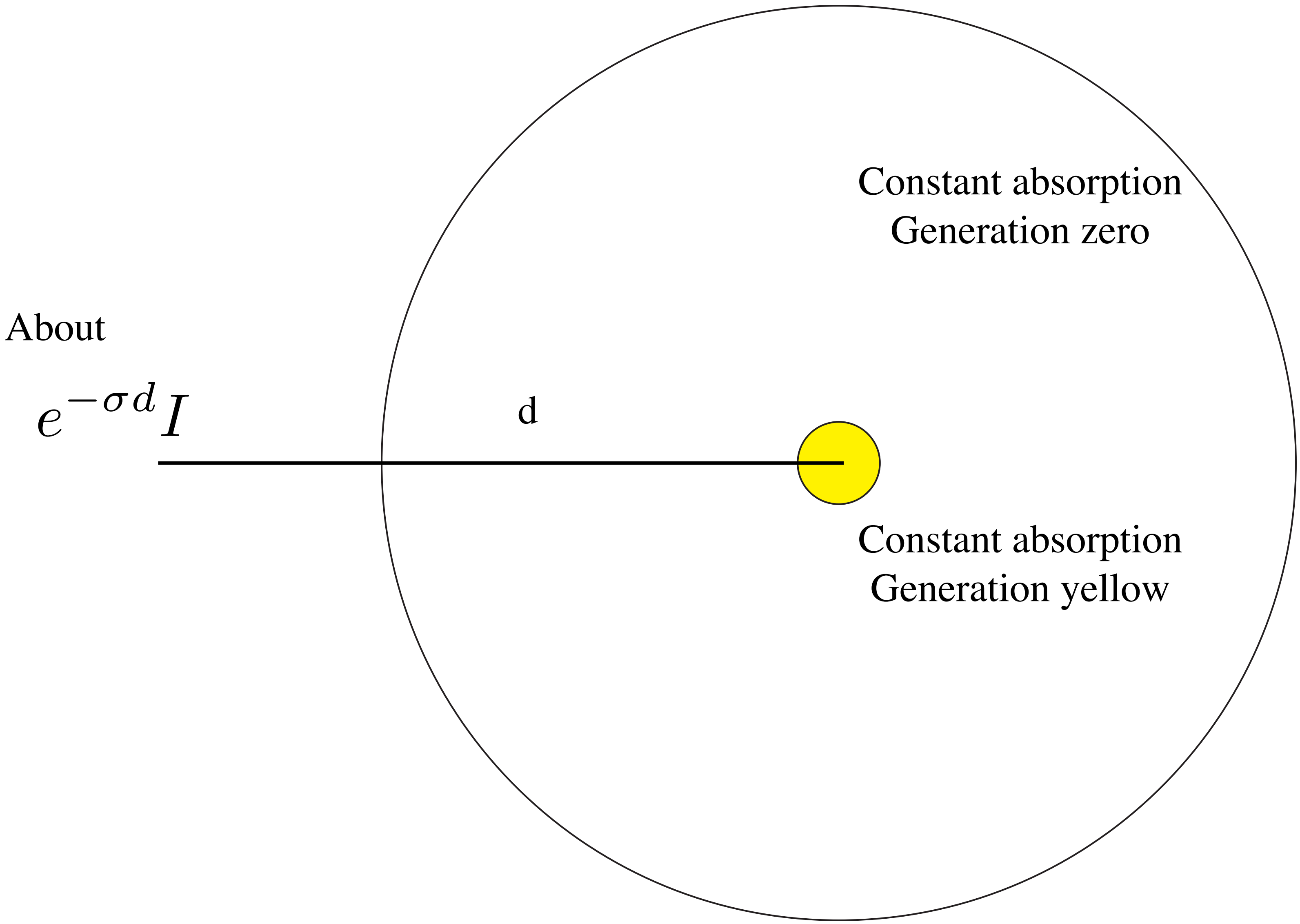
Current rough state of math

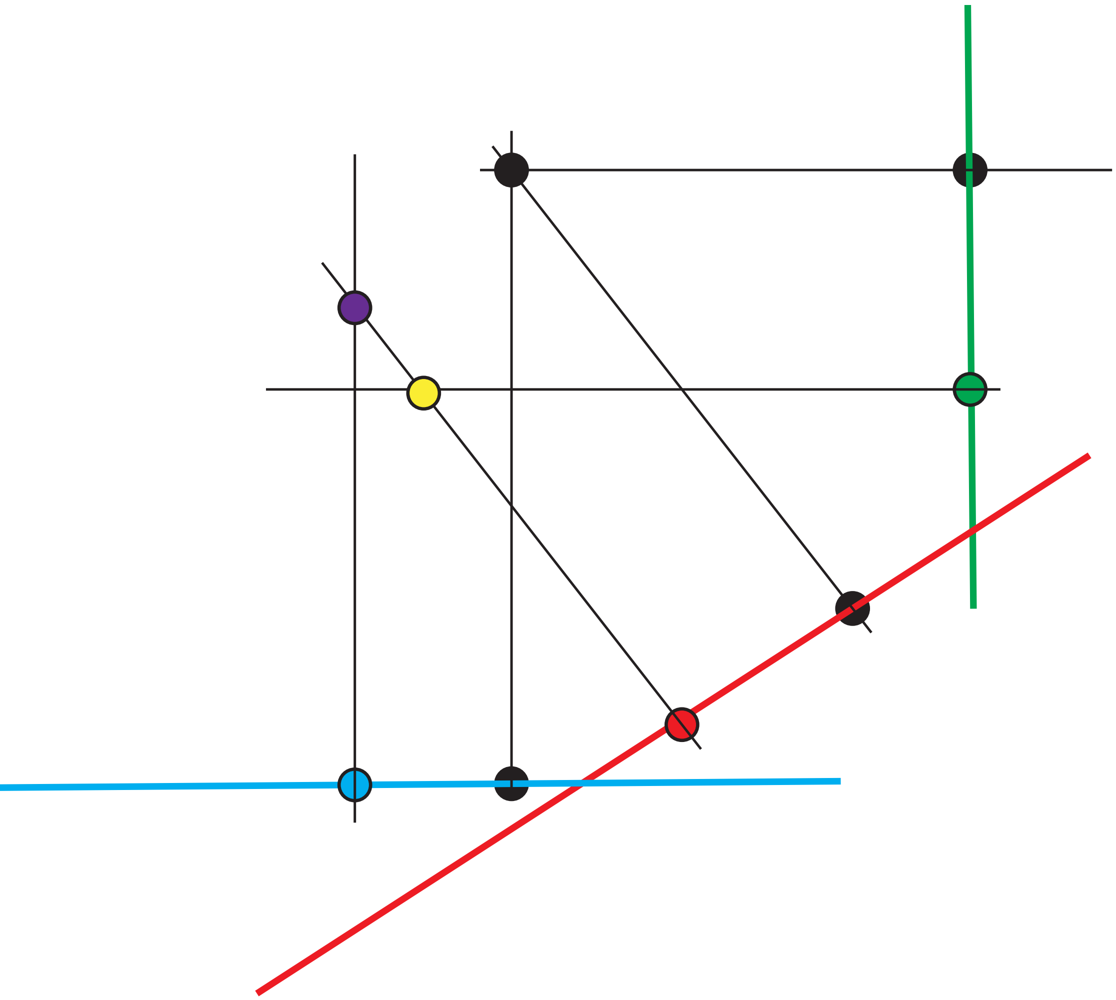
- Versions of the NERF repn. are studied in tomography
  - Note if sigma is known, mapping from C to I is linear
  - If sigma is known constant, then it's injective
    - i.e. an infinite set of images has a unique C
  - If sigma isn't known, and depends on angle, not much is known
    - pretty clearly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

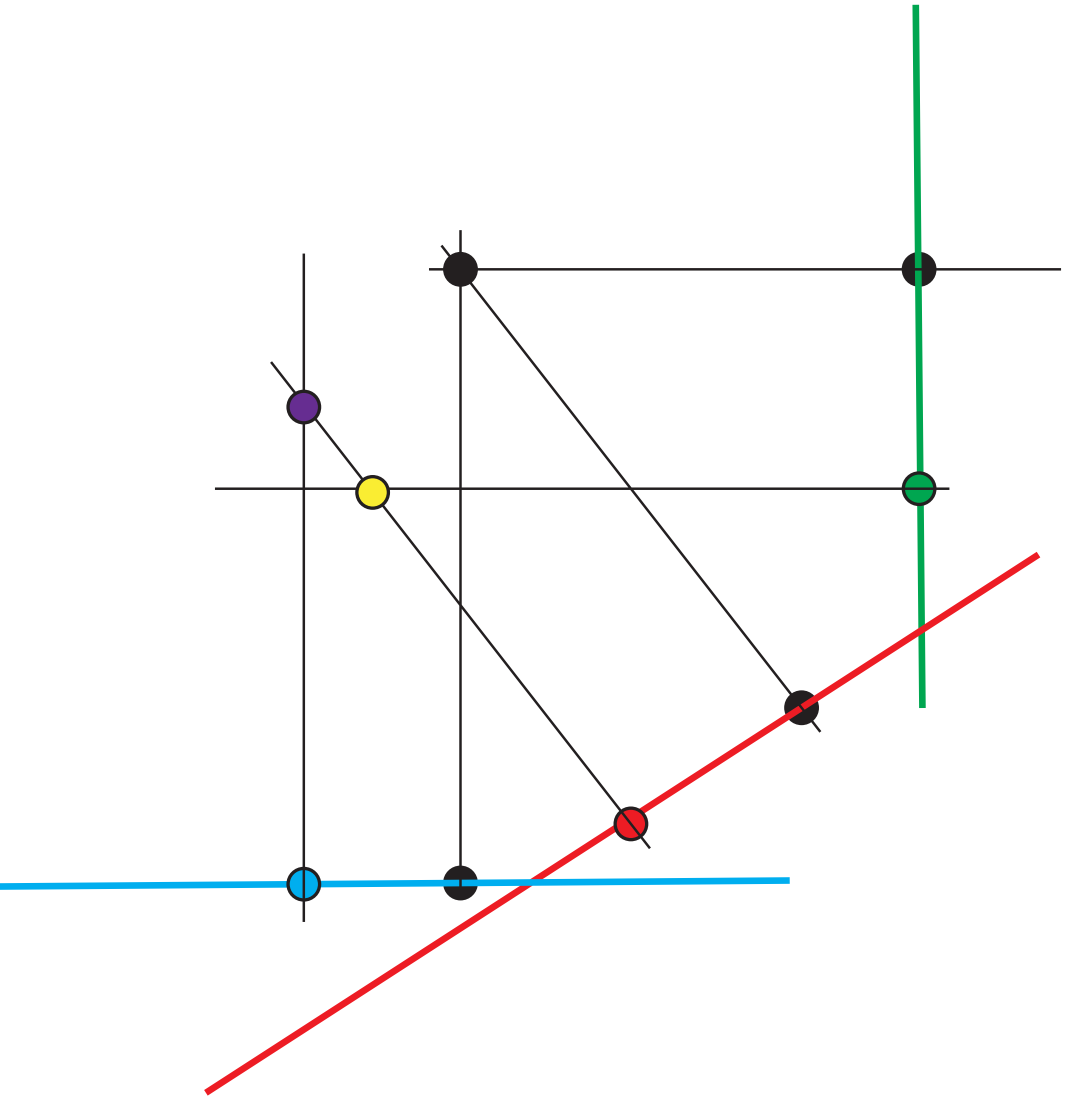
- Don’t need a reconstruction
  - need to predict renderings
- Failures of uniqueness as in picture are irrelevant
  - basic point is you can’t see them
- It may be possible to fudge localization difficulties
  - in a useful way


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



- 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