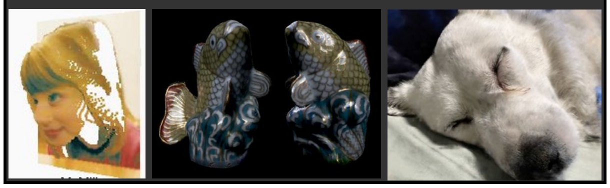
Image-Based Rendering

CSE 274, Lecture 3: Preliminaries

Ravi Ramamoorthi

http://www.cs.ucsd.edu/~ravir



1

Outline

- Will cover many preliminaries today briefly
  - Monte Carlo Integration
  - Path Tracing
  - Volumetric Rendering
  - Deep Learning: CNNs and MLPs
  - (consider vision course for multi-view geometry)
- To Do: Papers to present, etc

2

Motivation: Monte Carlo Integration

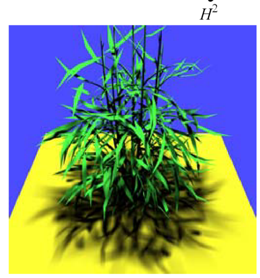
Rendering = integration
- Reflectance equation: Integrate over incident illumination
- Rendering equation: Integral equation

Many sophisticated shading effects involve integrals
- Antialiasing
- Soft shadows
- Indirect illumination
- Caustics

3

Example: Soft Shadows

$$E(x) = \int_{H^2} L_i(x,\omega)\cos\theta\, d\omega$$



Challenges
- **Visibility and blockers**
- **Varying light distribution**
- **Complex source geometry**

Source: Agrawala. Ramamoorthi, Heirich, Moll, 2000

4

Monte Carlo

- Algorithms based on statistical sampling and random numbers
- Coined in the beginning of 1940s. Originally used for neutron transport, nuclear simulations
  - Von Neumann, Ulam, Metropolis, …
- Canonical example: 1D integral done numerically
  - Choose a set of random points to evaluate function, and then average (expectation or statistical average)

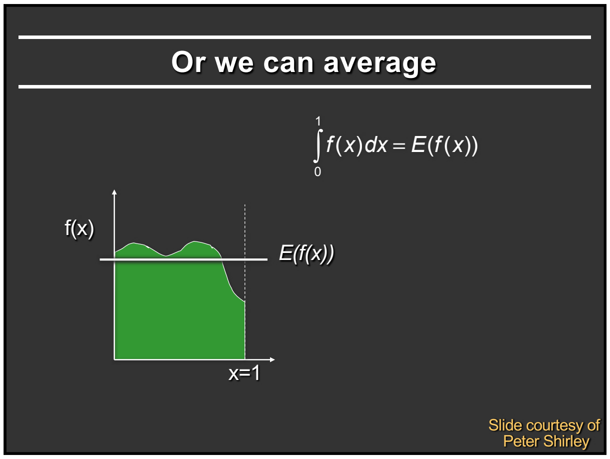
5

Monte Carlo Algorithms

Advantages
- Robust for complex integrals in computer graphics (irregular domains, shadow discontinuities and so on)
- Efficient for high dimensional integrals (common in graphics: time, light source directions, and so on)
- Quite simple to implement
- Work for general scenes, surfaces
- Easy to reason about (but care taken re statistical bias)

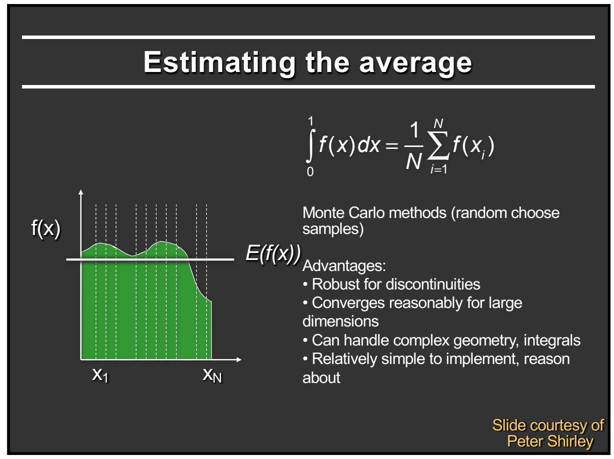
Disadvantages
- Noisy
- Slow (many samples needed for convergence)
- Not used if alternative analytic approaches exist (but those are rare)

6



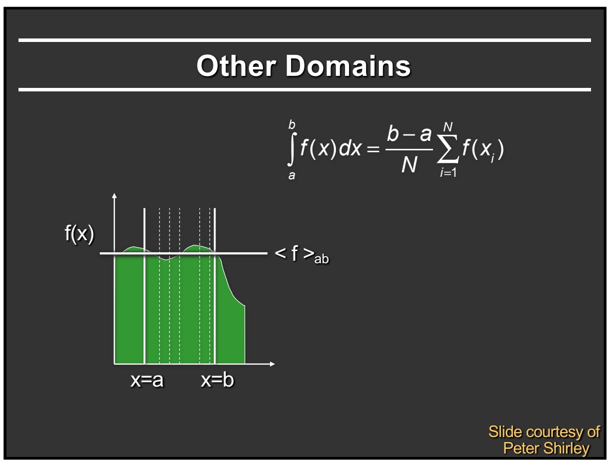
Or we can average
$\int_0^1 f(x)dx = E(f(x))$
f(x)
E(f(x))
x=1
Slide courtesy of Peter Shirley

7



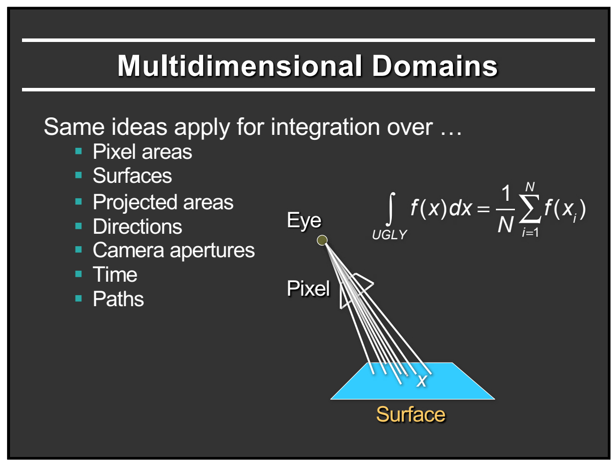
Estimating the average
$\int_0^1 f(x)dx = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$
f(x)
E(f(x))
x_1 x_N
Monte Carlo methods (random choose samples)
Advantages:
• Robust for discontinuities
• Converges reasonably for large dimensions
• Can handle complex geometry, integrals
• Relatively simple to implement, reason about
Slide courtesy of Peter Shirley

8



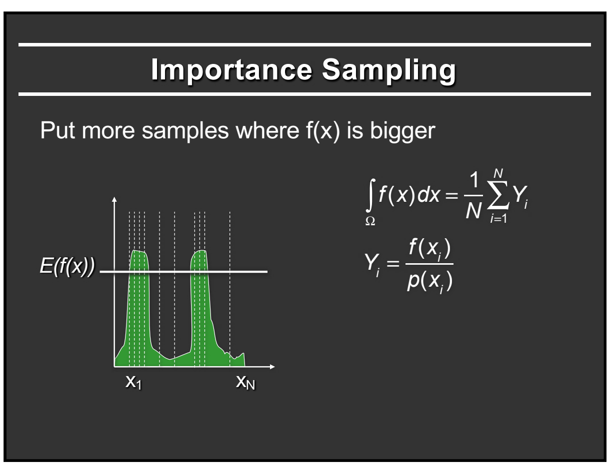
Other Domains
$\int_a^b f(x)dx = \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$
f(x)
$< f >_{ab}$
x=a x=b
Slide courtesy of Peter Shirley

9



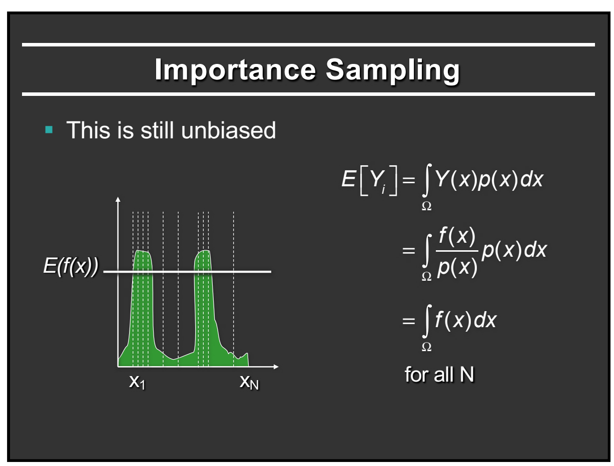
Multidimensional Domains
Same ideas apply for integration over …
Pixel areas
Surfaces
Projected areas
Directions
Camera apertures
Time
Paths
$\int_{UGLY} f(x)dx = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$
Eye
Pixel
x
Surface

10



Importance Sampling
Put more samples where f(x) is bigger
$\int_\Omega f(x)dx = \frac{1}{N}\sum_{i=1}^{N} Y_i$
$Y_i = \frac{f(x_i)}{p(x_i)}$
E(f(x))
x_1 x_N

11

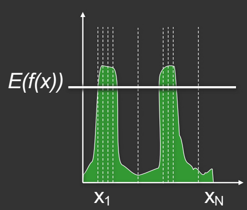


Importance Sampling
This is still unbiased
$E[Y_i] = \int_\Omega Y(x)p(x)dx$
$= \int_\Omega \frac{f(x)}{p(x)} p(x)dx$
$= \int_\Omega f(x)dx$
for all N
E(f(x))
x_1 x_N

12

Importance Sampling

- Zero variance if p(x) ~ f(x)



$$p(x) = cf(x)$$

$$Y_i = \frac{f(x_i)}{p(x_i)} = \frac{1}{c}$$

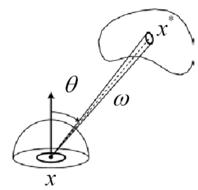
$$Var(Y) = 0$$

Less variance with better importance sampling

13

Direct Lighting – Directional Sampling

$$E(x) = \int_{\Omega} L(x,\omega)\cos\theta \, d\omega$$



Ray intersection $x^*(x,\omega)$

Sample ω **uniformly by** Ω

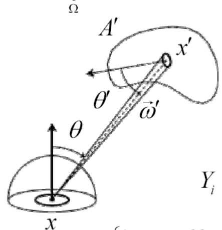
$$Y_i = L(x^*(x,\omega_i), -\omega_i)\cos\theta \, 2\pi$$

CS348B Lecture 6 Pat Hanrahan, Spring 2004

14

Direct Lighting – Area Sampling

$$E(x) = \int_{\Omega} L_i(x,\omega)\cos\theta \, d\omega = \int_{A'} L_o(x',\omega')V(x,x')\frac{\cos\theta\cos\theta'}{|x-x'|^2}dA'$$



Ray direction $\omega' = x - x'$

Sample x' **uniformly by** A'

$$Y_i = L_o(x_i', \omega_i')\, V(x, x_i')\frac{\cos\theta\cos\theta_i'}{|x - x_i'|^2} A$$

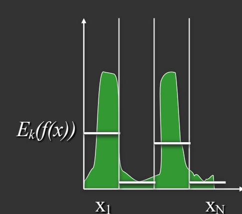
$$V(x,x') = \begin{cases} 0 & \neg visible \\ 1 & visible \end{cases}$$

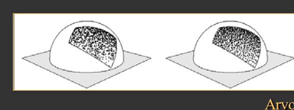
CS348B Lecture 6 Pat Hanrahan, Spring 2004

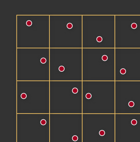
15

Stratified Sampling

- Estimate subdomains separately



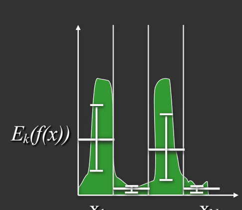




16

Stratified Sampling

- Less overall variance if less variance in subdomains



$$Var\left[F_N\right] = \frac{1}{N^2}\sum_{k=1}^{M} N_i Var\left[F_i\right]$$

17

More Information

- Veach PhD thesis chapter (linked to from website)
- Course Notes (links from website)
  - *Mathematical Models for Computer Graphics*, Stanford, Fall 1997
  - *State of the Art in Monte Carlo Methods for Realistic Image Synthesis*, Course 29, SIGGRAPH 2001

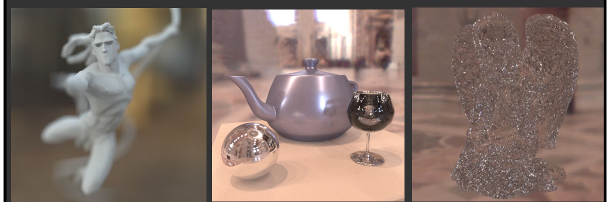
18

Motivation: Monte Carlo Path Tracing

- Core method to solve rendering equation
- Widely used production+realtime (with denoising)
- General solution to rendering, global illumination
- Suitable for a variety of general scenes
- Based on Monte Carlo methods
- Enumerate all paths of light transport

19

From UCB class many years ago



20

Simplest Monte Carlo Path Tracer

For each pixel, cast n samples and average
- Choose a ray with *p*=camera, *d*=(θ,ϕ) within pixel
- Pixel color += (1/n) * TracePath(*p*, *d*)

TracePath(*p*, *d*) returns (r,g,b) [and calls itself recursively]:
- Trace ray (*p*, *d*) to find nearest intersection *p*′
- Select with probability (say) 50%:
  - Emitted:
    return 2 * (Le_{red}, Le_{green}, Le_{blue}) // 2 = 1/(50%)
  - Reflected:
    generate ray in random direction *d*′
    return 2 * f_r(*d* → *d*′) * (*n*•*d*′) * TracePath(*p*′, *d*′)

Weight = 1/probability
Remember: unbiased
requires having f(x) / p(x)

21

Simplest Monte Carlo Path Tracer

For each pixel, cast n samples and average
- Choose a ray with *p*=camera, *d*=(θ,ϕ) within pixel
- Pixel color += (1/n) * TracePath(*p*, *d*)

TracePath(*p*, *d*) returns (r,g,b) [and calls itself recursively]:
- Trace ray (*p*, *d*) to find nearest intersection *p*′
- Select with probability (say) 50%:
  - Emitted:
    return 2 * (Le_{red}, Le_{green}, Le_{blue}) // 2 = 1/(50%)
  - Reflected:
    generate ray in random direction *d*′
    return 2 * f_r(*d* → *d*′) * (*n*•*d*′) * TracePath(*p*′, *d*′)

Path terminated when
Emission evaluated

22

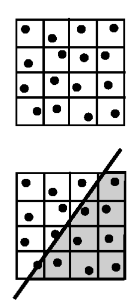
Path Tracing: Include Direct Lighting

```
Step 1. Choose a camera ray r given the
   (x,y,u,v,t) sample
   weight = 1;
   L = 0
Step 2. Find ray-surface intersection
Step 3.
   L += weight * Lr(light sources)
   weight *= reflectance(r)
   Choose new ray r' ~ BRDF pdf(r)
       Go to Step 2.
```

CS348B Lecture 14 Pat Hanrahan, Spring 2009

23

Stratified Sampling



Stratified sampling like jittered sampling

Allocate samples per region

$$N = \sum_{i=1}^{m} N_i \quad F_N = \frac{1}{N}\sum_{i=1}^{m} N_i F_i$$

New variance

$$V[F_N] = \frac{1}{N^2}\sum_{i=1}^{m} N_i V[F_i]$$

Thus, if the variance in regions is less than the overall variance, there will be a reduction in resulting variance

For example: An edge through a pixel

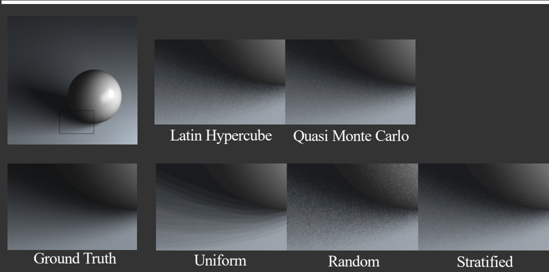
$$V[F_N] = \frac{1}{N^2}\sum_{i=1}^{\sqrt{N}} V[F_i] = \frac{V[F_E]}{N^{1.5}}$$

CS348B Lecture 9 Pat Hanrahan, Spring 2002

D. Mitchell 95, Consequences of stratified sampling in graphics

24

Comparison of simple patterns



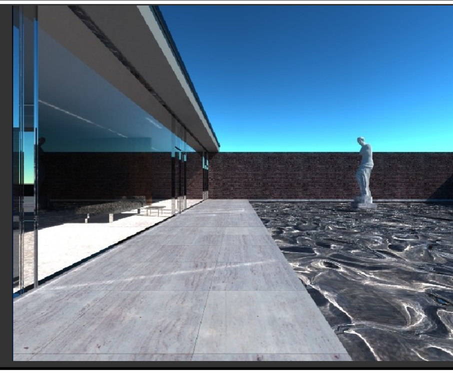
16 samples for area light, 4 samples per pixel, total 64 samples

If interested, see my recent paper "A Theory of Monte Carlo Visibility Sampling"

Figures courtesy Tianyu Liu

25

Mies House: Swimming Pool



26

Optional Path Tracing Assignment

- If you have not taken CSE 168 or done path tracer
- Follow CSE 168 on edX or edX edge (ask me for access if needed), build path tracer
- Includes guide for raytracing if not already done
- For your benefit only, optional do not turn in (since many people wanted it for knowledge)
- You can use it in final project, but don't need to, and may be better off using off-the-shelf renderer
- If you do use it in final project, document it
- Again, it is optional and not directly graded

27

Volumetric Scattering

28

Volumetric Rendering

- Participating Media (light participates via scattering)
  - Volumetric phenomena like clouds, smoke, fire
  - Subsurface scattering, translucency (wax, human skin)
  - Medium is often known as a participating medium
  - ***For IBR/NeRF, only absorption, emission no scattering***
- Surface Rendering: Radiance Constant along Ray
  - Only true in absence of participating media
  - *No longer true for volumetric scattering*
  - *Often replace ray tracing with ray marching in medium*
- Volumetric Properties
  - BRDF replaced by phase function
  - Must consider absorption and scattering in medium

29

Full Volumetric Rendering Formulation

Slide credit: Novak et al 2018, Monte Carlo methods for physically based volume rendering

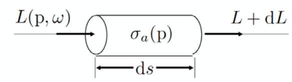
30

Volumetric Interactions

- 4 different processes affect radiance of a beam
  - Absorption
  - Out-Scattering (not used in IBR/NeRF)
  - Emission
  - In-Scattering (not used in IBR/NeRF)

31

Absorption



$$\mathrm{d}L(\mathrm{p},\omega) = -\sigma_a(\mathrm{p})\, L(\mathrm{p},\omega)\, \mathrm{d}s$$

Absorption cross section: $\sigma_a(\mathrm{p})$

- **Probability of being absorbed per unit length**
- **Units: 1/distance**

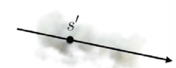
32

Absorption

33

Transmittance

$$\mathrm{d}L(\mathrm{p},\omega) = -\sigma_a(\mathrm{p})\, L(\mathrm{p},\omega)\, \mathrm{d}s$$



$$\frac{\mathrm{d}L(\mathrm{p},\omega)}{L(\mathrm{p},\omega)} = -\sigma_a(\mathrm{p})\, \mathrm{d}s$$

$$\log L(\mathrm{p}+s\omega,\omega) = -\int_0^s \sigma_a(\mathrm{p}+s'\omega,\omega)\, \mathrm{d}s' = -\tau(s)$$

Optical distance (depth): $\tau(s) = \int_0^s \sigma_a(\mathrm{p}')\, \mathrm{d}s'$

$$\mathrm{p}' = \mathrm{p} + s'\omega$$

Homogeneous medium–constant σ_a: $\tau(s) = \sigma_a s$

34

Transmittance and Opacity

$$\mathrm{d}L(\mathrm{p},\omega) = -\sigma_a(\mathrm{p})\, L(\mathrm{p},\omega)\, \mathrm{d}s$$

$$\frac{\mathrm{d}L(\mathrm{p},\omega)}{L(\mathrm{p},\omega)} = -\sigma_a(\mathrm{p})\, \mathrm{d}s$$

$$\log L(\mathrm{p}+s\omega,\omega) = -\int_0^s \sigma_a(\mathrm{p}+s'\omega,\omega)\, \mathrm{d}s' = -\tau(s)$$

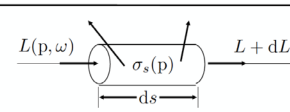
$$L(\mathrm{p}+s\omega,\omega) = \mathrm{e}^{-\tau(s)} L(\mathrm{p},\omega) = T(s)\, L(\mathrm{p},\omega)$$

Transmittance: $T(s) = \mathrm{e}^{-\tau(s)}$

Opacity: $\alpha(s) = 1 - T(s)$

35

Out-Scattering



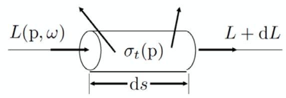
$$\mathrm{d}L(\mathrm{p},\omega) = -\sigma_s(\mathrm{p})\, L(\mathrm{p},\omega)\, \mathrm{d}s$$

Scattering cross-section: σ_s

- **Probability of being scattered per unit length**

36

Extinction



$$\mathrm{d}L(\mathrm{p},\omega) = -\sigma_t(\mathrm{p})\, L(\mathrm{p},\omega)\,\mathrm{d}s$$

Total cross section: $\sigma_t = \sigma_a + \sigma_s$

Albedo: $W = \dfrac{\sigma_s}{\sigma_t} = \dfrac{\sigma_s}{\sigma_a + \sigma_s}$

Optical distance from absorption and scattering:

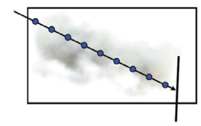
$$\tau(s) = \int_0^s \sigma_t(\mathrm{p}')\,\mathrm{d}s'$$

37

Ray Marching for Transmittance

$$\tau(s) = \int_0^s \sigma_t(x + s'\omega)\,\mathrm{d}s'$$

$$T(s) = \mathrm{e}^{-\tau(s)}$$



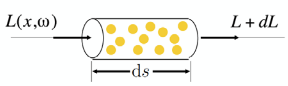
Monte Carlo not necessary for 1D—can use a Riemann sum:

$$\tau(s) \approx \frac{s}{N}\sum_i^N \sigma_t(x_i)$$

$$x_i = x + \frac{i + 0.5}{N}\omega$$

38

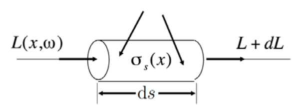
Emission



$$\mathrm{d}L(\mathrm{p},\omega) = \sigma_a(\mathrm{p}) L_\mathrm{e}(\mathrm{p},\omega)\,\mathrm{d}s$$

39

In-Scattering



$$S(\mathrm{p},\omega) = \sigma_s(\mathrm{p}) \int_{S^2} p(\omega' \to \omega)\, L(\mathrm{p},\omega')\,\mathrm{d}\omega'$$

Phase function: $p(\omega' \to \omega)$

Reciprocity: $p(\omega' \to \omega) = p(\omega \to \omega')$

Energy conservation: $\int_{S^2} p(\omega' \to \omega)\,\mathrm{d}\omega' = 1$

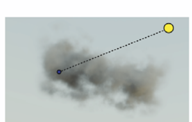
40

Direct Illumination in a Volume

$$S_\mathrm{d}(\mathrm{p}',\omega) = \sigma_s(\mathrm{p}') \int_{S^2} p(\omega' \to \omega)\, L_\mathrm{d}(\mathrm{p}',\omega')\,\mathrm{d}\omega'$$

Can treat like direct illumination at a surface

- **Sample from phase function's distribution**
- **Sample from light source distributions**
- **Weight using multiple importance sampling**



41

Direct Illumination in a Volume

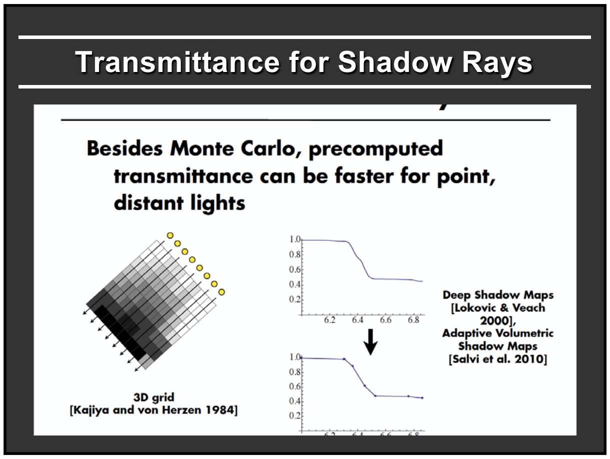
$$S_\mathrm{d}(\mathrm{p}',\omega) = \sigma_s(\mathrm{p}') \int_{S^2} p(\omega' \to \omega)\, L_\mathrm{d}(\mathrm{p}',\omega')\,\mathrm{d}\omega'$$

Estimator: $\sigma_s(\mathrm{p}')\dfrac{1}{N}\sum_i^N \dfrac{p(\omega_i \to \omega)\, L_d(\mathrm{p}',\omega_i)}{p(\omega_i)}$

Computing direct lighting, L_d can be expensive

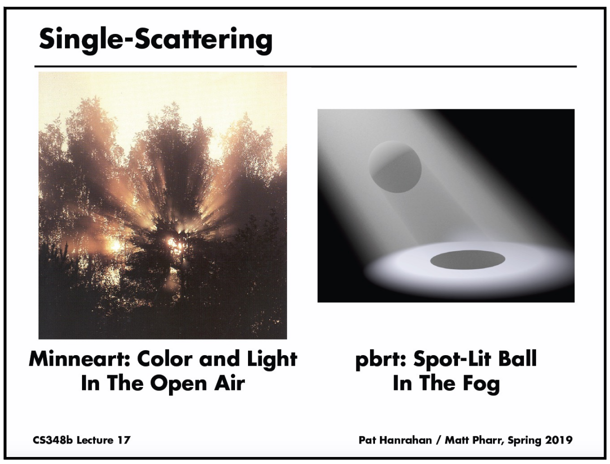
- **Not just a shadow ray—need to compute transmittance**

42

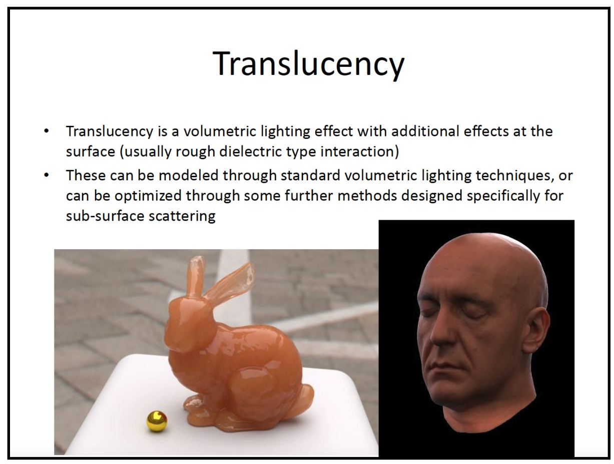


43

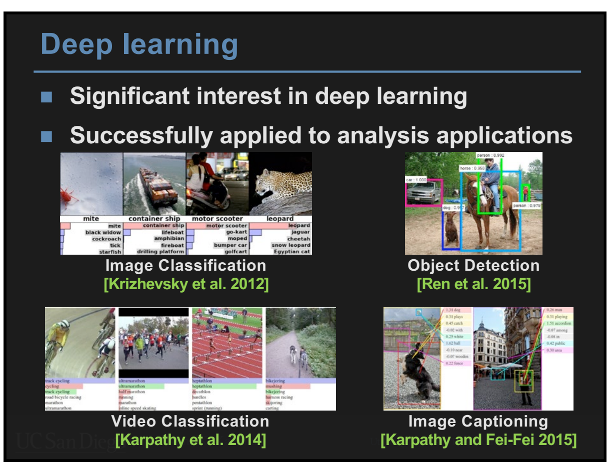
44



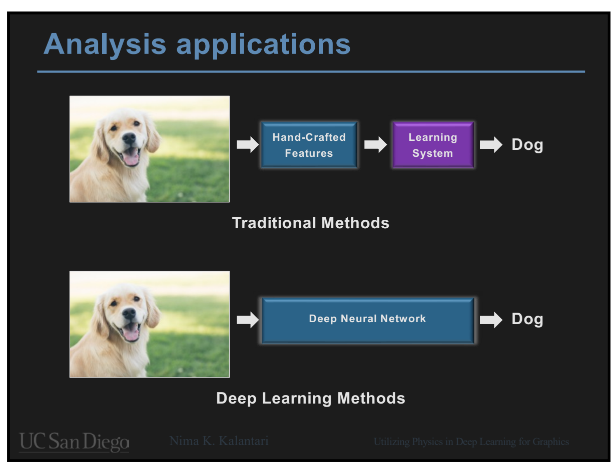
45



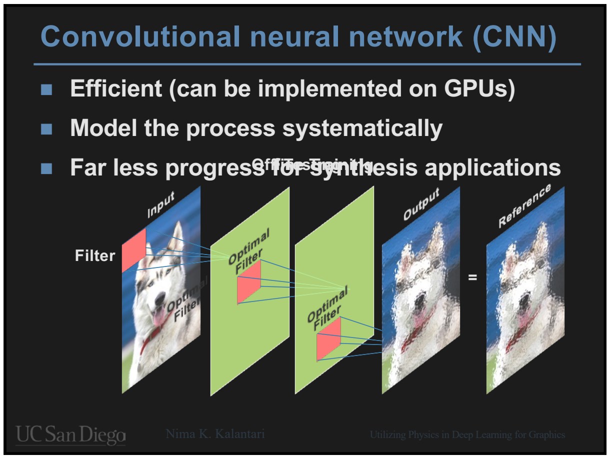
46



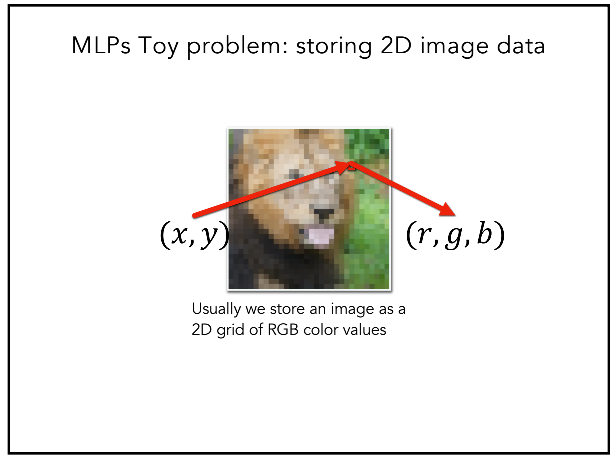
47



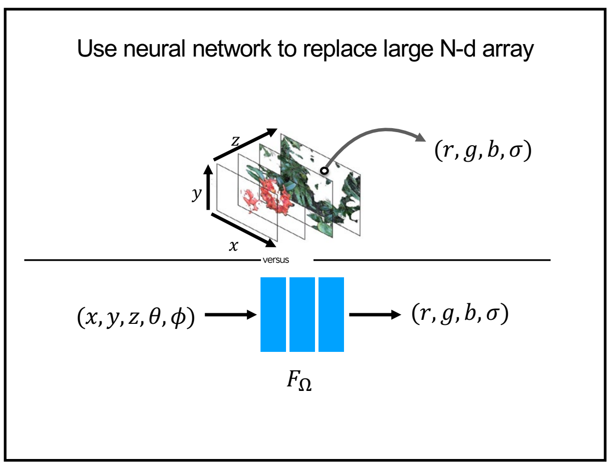
48


Convolutional neural network (CNN)
Efficient (can be implemented on GPUs)
Model the process systematically
Far less progress for synthesis applications
Input
Filter
Optimal Filter
Optimal Filter
Output
Reference
=
UC San Diego
Nima K. Kalantari
Utilizing Physics in Deep Learning for Graphics

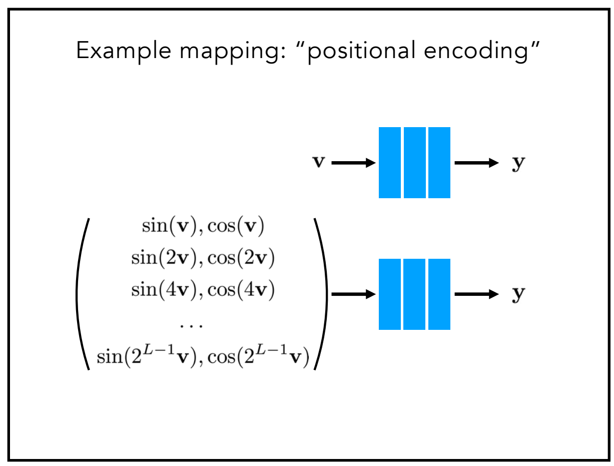
49


MLPs Toy problem: storing 2D image data
(x, y)
(r, g, b)
Usually we store an image as a
2D grid of RGB color values

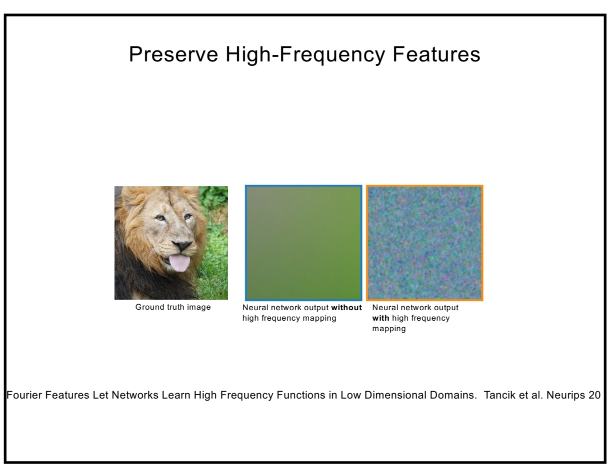
50


Use neural network to replace large N-d array
(r, g, b, σ)
z
y
x
versus
(x, y, z, θ, ϕ)
(r, g, b, σ)
F_Ω

51


Example mapping: “positional encoding”
$\mathbf{v}$
$\mathbf{y}$
$\sin(\mathbf{v}), \cos(\mathbf{v})$
$\sin(2\mathbf{v}), \cos(2\mathbf{v})$
$\sin(4\mathbf{v}), \cos(4\mathbf{v})$
$\dots$
$\sin(2^{L-1}\mathbf{v}), \cos(2^{L-1}\mathbf{v})$
$\mathbf{y}$

52


Preserve High-Frequency Features
Ground truth image
Neural network output without high frequency mapping
Neural network output with high frequency mapping
Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. Tancik et al. Neurips 20

53