Physics-based rendering and its applications in computational photography and imaging

Adithya Pediredla  
Dartmouth College  
adithya.k.pediredla@dartmouth.com

Ioannis Gkioulekas  
Carnegie Mellon University  
igkioule@andrew.cmu.edu
Physics-based rendering and its applications to computational imaging
Physics-based rendering and its applications to computational imaging

- accurate and efficient simulation
- virtually design sensors, optics, and algorithms
Physics-based rendering and its applications to computational imaging

**Forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**Inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems
Physics-based rendering and its applications to computational imaging

forward rendering
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

inverse rendering
- accurate and efficient differentiable simulation
- tractably solve general inverse problems
Physics-based rendering and its applications to computational imaging

forward rendering
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

inverse rendering
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

time-of-flight imaging
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

time-of-flight imaging
non-line-of-sight imaging
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
Physics-based rendering and its applications to computational imaging

forward rendering

- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

inverse rendering

- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
- tactile sensor design
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
- tactile sensor design
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

---

time-of-flight imaging
don-line-of-sight imaging
acousto-optic imaging
ultrafast light imaging

---
speckle imaging
tactile sensor design
differentiable rendering
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

Applications:
- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
- tactile sensor design
- differentiable rendering
- inverse problems
Complex light transport

After [Ritschel et al 2011]
Complex light transport

After [Ritschel et al 2011]
Complex light transport

After [Ritschel et al 2011]
Complex light transport

After [Ritschel et al 2011]
Complex light transport

After [Ritschel et al 2011]
Complex light transport

- Direct light
- Indirect shadow
- Indirect light
- Caustic
- Glossy reflections

After [Ritschel et al 2011]
Complex light transport

- Caustic
- Volumetric scattering

After [Ritschel et al 2011]
Path integral form of light transport

\[ I = \int_{\mathcal{P}} W_e(x_0, x_1) L_c(x_k, x_{k-1}) T(\bar{x}) \, d\bar{x} \]
Path integral form of light transport

$$I = \int_{\mathcal{P}} W_e(x_0, x_1) L_e(x_k, x_{k-1}) T(\bar{x}) \, d\bar{x}$$

light path

space of all light paths
Path integral form of light transport

\[ I = \int_{\mathcal{P}} W_e(x_0, x_1)L_e(x_k, x_{k-1})T(\bar{x}) \, d\bar{x} \]

sensor

weight

image

light path

space of all light paths
Path integral form of light transport

![Diagram showing light transport from a source to a sensor](image)

The image can be represented as:

\[ I = \int_{\mathcal{P}} W_e(x_0, x_1) L_e(x_k, x_{k-1}) T(\bar{x}) \, d\bar{x} \]

where \( I \) is the image, \( W_e \) is the weight from sensor to source, \( L_e \) is the light path, and \( T \) is the transmission through the medium.
Path integral form of light transport

\[ I = \int_{\mathcal{P}} W_e(x_0, x_1) L_e(x_k, x_{k-1}) T(\mathbf{x}) \, d\mathbf{x} \]

\( T(\mathbf{x}) = G(x_0, x_1) \prod_{j=1}^{k-1} f(x_j, x_{j+1}, x_{j-1}) G(x_j, x_{j+1}) \)
Path integral form of light transport

image \[ I = \int_\mathcal{P} W_e(x_0, x_1) L_e(x_k, x_{k-1}) T(\bar{x}) \, d\bar{x} \] light path

path throughput

\[ T(\bar{x}) = G(x_0, x_1) \prod_{j=1}^{k-1} f(x_j, x_{j+1}, x_{j-1}) G(x_j, x_{j+1}) \] BSDF
Path integral form of light transport

\[ I = \int_{\mathcal{P}} W_e(x_0, x_1) L_e(x_k, x_{k-1}) T(\vec{x}) \, d\vec{x} \]

image

space of all light paths

sensor weight

source weight

light path

path throughput

\[ T(\vec{x}) = G(x_0, x_1) \prod_{j=1}^{k-1} f(x_j, x_{j+1}, x_{j-1}) G(x_j, x_{j+1}) \]

BSDF

gometry

\[ f(x_2, x_3, x_1) \]

\[ G(x_2, x_3) \]

\[ G(x_3) \]
Monte Carlo rendering

approximate image as \( I \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(\bar{x}_i)}{p(\bar{x}_i)} \)
Monte Carlo rendering

approximate image as

\[ I \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(\bar{x}_i)}{p(\bar{x}_i)} \]

sum over *randomly* sampled paths
Monte Carlo rendering

approximate image as \[ I \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(\bar{x}_i)}{p(\bar{x}_i)} \]

sum over \textit{randomly} sampled paths
Monte Carlo rendering

approximate image as \( I \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_s(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(\bar{x}_i)}{p(\bar{x}_i)} \)

sum over \textit{randomly} sampled paths

PDF of random path

sensor weight  source weight  path throughput

Monte Carlo rendering

approximate image as $I_j \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(\bar{x}_i)}{p(\bar{x}_i)}$

sum over \textit{randomly} sampled paths

PDF of random path

\textit{Path tracing}: sample path starting from sensor
Monte Carlo rendering

approximate image as \( I_j \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(x_{i,0}, x_{i,1})}{p(x_i)} L_e(x_{i,k}, x_{i,k-1}) T(x_i) \)

sum over \textit{randomly} sampled paths

PDF of random path

Path tracing: sample path starting from sensor
Monte Carlo rendering

approximate image as \( I_j \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(x_{i,0}, x_{i,1})L_e(x_{i,k}, x_{i,k-1})T(\bar{x}_i)}{p(\bar{x}_i)} \)

sum over \textit{randomly} sampled paths

PDF of random path

\textit{Light tracing}: sample path starting from source
Monte Carlo rendering

approximate image as \( I_j \approx \frac{1}{N} \sum_{i=1}^{N} W_e(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(x_i) p(x_i) \)

sum over \textit{randomly} sampled paths

PDF of random path

\textit{Light tracing}: sample path starting from source
Monte Carlo rendering

approximate image as

$$I_j \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(\bar{x}_i)}{p(\bar{x}_i)}$$

sum over \textit{randomly} sampled paths

PDF of random path

\textit{Bidirectional path tracing}: sample path starting from both source and sensor
Monte Carlo rendering

approximate image as $I_j \approx \frac{1}{N} \sum_{i=1}^{N} W_e(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(\bar{x}_i) p(\bar{x}_i)$

sum over *randomly* sampled paths

PDF of random path

*Bidirectional path tracing*: sample path starting from both source and sensor
Monte Carlo rendering

approximate image as \[ I_j \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(x_{i})}{p(x_{i})} \]

sum over \textit{randomly} sampled paths

PDF of random path

\textit{Bidirectional path tracing}: sample path starting from both source and sensor
Monte Carlo rendering

$$I_j \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(x_{i,0}, x_{i,1}) L_e(x_{i,k}, x_{i,k-1}) T(\bar{x}_i)}{p(\bar{x}_i)}$$

sum over randomly sampled paths

PDF of random path

*Bidirectional path tracing*: sample path starting from both source and sensor
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- **time-of-flight imaging**
- **non-line-of-sight imaging**
- **acousto-optic lensing**
- **ultrafast light scanning**

- **speckle imaging**
- **tactile sensor design**
- **differentiable renderer**
- **inverse problems**
Time-of-flight cameras

[Velten et al. ToG, 2013]
Time-of-flight cameras

[Velten et al. ToG, 2013]
Time-of-flight cameras

[Velten et al. ToG, 2013]
Time-of-flight cameras

[Velten et al. ToG, 2013]
Time-of-flight cameras

[Velten et al. ToG, 2013]
Time-of-flight cameras

SwissRanger SR4000
Hamamatsu streak
Brightway VISDOM
Microsoft Kinect
MPD SPAD
Stanford ICCD
Time-of-flight cameras

SwissRanger SR4000
Microsoft Kinect continuous-wave

Hamamatsu streak

MPD SPAD transient

Brightway VISDOM time-gated

Stanford ICCD time-gated
Time-of-flight cameras

intensity  continuous wave  transient  time-gated
Time-of-flight cameras

intensity  continuous wave  transient  time-gated
Rendering time-of-flight cameras: Path space integral

\[ \text{image} = \int f(\text{path}) \]

Path contribution depends on scene properties, light source, and sensor.
Rendering time-of-flight cameras: Path space integral

image = \int f(\text{path})

path contribution, depends on scene properties, light source, and sensor

Monte Carlo rendering:

- randomly sample paths: path_1, path_2, ..., path_N
- approximate image as:

\text{image} \approx \sum_n \frac{f(\text{path}_n)}{\text{prob}(\text{path}_n)}
Rendering time-of-flight cameras: Path space integral

\[ \text{image} = \int f(\text{path}) W(|\text{path}|) \]

Monte Carlo rendering:

- randomly sample paths: \( \text{path}_1, \text{path}_2, \ldots, \text{path}_N \)
- approximate image as:

\[ \text{image} \approx \sum_n \frac{f(\text{path}_n)}{\text{prob}(\text{path}_n)} \]
Rendering time-of-flight cameras: Path space integral

\[
\text{image} = \int f(\text{path}) W(|\text{path}|)
\]

Monte Carlo rendering:

- randomly sample paths: \( \text{path}_1, \text{path}_2, ..., \text{path}_N \)
- approximate image as:

\[
\text{time-of-flight image} \approx \sum_n \frac{f(\text{path}_n)}{\text{prob}(\text{path}_n)} W(|\text{path}_n|)
\]
Rendering continuous wave time-of-flight camera

\[ \text{image} = \int f(\text{path}) W(|\text{path}|) \]
Classification: time-gated camera

\[ \text{image} = \int f(\text{path}) W(|\text{path}|) \]

Light paths
Rendering transient camera

\[ \text{image} = \int f(\text{path}) W(|\text{path}|) \]
Path space integral for time-of-flight cameras

\[
\text{image} = \int f(\text{path}) W(|\text{path}|)
\]

- **Intensity**
  - Path length $|\mathbf{x}|$
  - $W(|\mathbf{x}|)$
  - BDPT, PT, PM, KDE, etc.

- **Continuous-wave**
  - Path length $|\mathbf{x}|$
  - $W(|\mathbf{x}|)$
  - BDPT, PT, PM, KDE, etc.

- **Transient**
  - Path length $|\mathbf{x}|$
  - $W(|\mathbf{x}|)$
  - BDPT, PT, PM, KDE, etc.

- **Time-gated**
  - Path length $|\mathbf{x}|$
  - $W(|\mathbf{x}|)$
  - No efficient renderer

[Jarabo et al., 2014, 2017]
[Marco et al. 2017, 2018]
Path sampling for time-gated rendering is challenging

No control over path length
Path sampling for time-gated rendering is challenging.

No control over path length
Path sampling for time-gated rendering is challenging

No control over path length
Path sampling for time-gated rendering is challenging

No control over path length
Path sampling for time-gated rendering is challenging.

No control over path length.
Path sampling for time-gated rendering is challenging

No control over path length

continuous-wave

transient

time-gated

Path length ($|\vec{x}|$)

#paths
Path sampling for time-gated rendering is challenging. No control over path length.
Path sampling for time-gated rendering is challenging

No control over path length
Path sampling for time-gated rendering is challenging.

No control over path length.
Path sampling for time-gated rendering is challenging

No control over path length
Path sampling for time-gated case

\( \text{Path length } (|\mathbf{x}|) \)

\#paths
Path sampling for time-gated case

step 1: sample path length $|\bar{x}|$
Path sampling for time-gated case

Step 1: Sample path length $|\mathbf{x}|$

Step 2: Generate path with target length $|\mathbf{x}|$
Path sampling for time-gated case

step 1: sample path length $|\mathbf{x}|$

step 2: generate path with target length $|\mathbf{x}|$

bidirectional path tracing (BDPT)
**Path sampling for time-gated case**

step 1: sample path length $|\vec{x}|$

step 2: generate path with target length $|\vec{x}|$

**bidirectional path tracing (BDPT)**

generate light sub-path, camera sub-path
Path sampling for time-gated case

step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path
Path sampling for time-gated case

step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join source sub-path end $x_s$ and camera sub-path end $x_c$
Path sampling for time-gated case

step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join source sub-path end $x_L$ and camera sub-path end $x_C$
Path sampling for time-gated case

step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join $x_l$ and $x_c$ via connecting vertex ($x_e$)
Path sampling for time-gated case

step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

- bidirectional path tracing (BDPT)
- generate light sub-path, camera sub-path
- join $x_l$ and $x_c$ via connecting vertex ($x_e$)

$\text{source sub-path length} + \text{sensor sub-path length} + |x_e \rightarrow x_l| + |x_e \rightarrow x_c| = |\bar{x}|$
Path sampling for time-gated case

step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join $x_l$ and $x_c$ via connecting vertex ($x_e$)

$|x_e \rightarrow x_l| + |x_e \rightarrow x_c| = |\bar{x}| - \text{source + sensor sub-path lengths}$
Path sampling for time-gated case

step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join $x_l$ and $x_c$ via connecting vertex ($x_e$)

Definition of an ellipsoid: $|x_e \rightarrow x_l| + |x_e \rightarrow x_c| = |\bar{x}| - \text{source + sensor sub-path lengths}$
Path sampling for time-gated case

Step 1: Sample path length $|\bar{x}|$

Step 2: Generate path with target length $|\bar{x}|$

Bidirectional path tracing (BDPT)

Generate light sub-path, camera sub-path

Join $x_l$ and $x_c$ via connecting vertex ($x_e$)

Definition of an ellipsoid: $|x_e \to x_l| + |x_e \to x_c| = |\bar{x}| - \text{source + sensor sub-path lengths}$
Path sampling for time-gated case

surface case

volumetric case
Path sampling for time-gated case

- surface case
- volumetric case
Path sampling for time-gated case

surface case

volumetric case
Application: proximity detector for cars

virtual light curtain

proximity detected
Application: proximity detector for cars

- road scene
- standard BDPT
- BDPT w/ ellipsoidal connections

Gate width: 200 ps (1.14% scene)
Rendering time: 10s per frame
Application: proximity detector for cars

road scene

standard BDPT

BDPT w/ ellipsoidal connections

Time: 1.34s

Time: 1.74s
Imaging projects using this renderer

Mitsuba based open source implementation

non-line-of-sight sensor design [White et al. 2022]

depth sensing

tissue imaging

inverse rendering [Wu et al. 2021]

deep learning [Barragan et al. 2021]

SONAR [Reed et al. 2023]

backpropagation neural backprop.
Imaging projects using this renderer

Mitsuba based open source implementation

non-line-of-sight sensor design [White et al. 2022]

depth sensing

tissue imaging

inverse rendering [Wu et al. 2021]

SONAR [Reed et al. 2023]

backpropagation

neural backprop.
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
- tactile sensor design
- differentiable renderer
- inverse problems
Non-line-of-sight imaging
Applications

https://web.media.mit.edu/~raskar/cornar/
A lot of research on non-line-of-sight imaging

**intensity imaging**
- Bouman et al., ICCV 2017
- Saunders et al., Nature 2019
- Saunders et al., COSI 2019
- Maeda et al., ICCP 2019
- Lin et al., COSI 2020
- Sharma, ICCV 2021

**coherence imaging**
- Katz et al., Nat. Photonics 2014
- Lei et al., CVPR, 2019
- Boger-Lombard, Nat. Com. 2019
- Metzler et al., Optica 2020
- Willomitzer et al., Nat. Com. 2021
- Chen et al. SPIE 2022

**time-of-flight imaging**
- Velten et al., Nat. comm. 2012
- Toole et al., Nature 2018
- Liu et al., Nature 2019, Nat. comm. 2020
- Rapp et al. Nat. comm. 2020
- Xin et al., CVPR 2019
- Nam et al., Nat. comm. 2021
How do we image in line-of-sight?
How do we image in line-of-sight?
How do we image in non-line-of-sight?
How do we focus on a voxel?
How do we focus on a voxel?
How do we focus on a voxel?
How do we focus on a voxel?
How do we focus on a voxel?
How do we focus on a voxel?
How do we focus on a voxel?

challenge:
out of focus voxels
How do we focus on a voxel?

challenge: out of focus voxels

solution: time-gate photons
How do we focus on a voxel?

challenge:
non-specular photons
How do we focus on a voxel?

challenge:
non-specular photons

solution:
use a large lens?
expensive !!
How do we focus on a voxel?

challenge:
non-specular photons

solution:
use a large lens?
expensive !!

what does a lens do?
delays rays such that they reach detector at same time instant
Temporal focusing: imitate the lens

challenge:
non-specular photons

solution:
use a large lens? expensive!!

temporal focusing: imitate large lens
Temporal focusing: imitate the lens

challenge:
non-specular photons

description:
use a large lens?
expensive !!

temporal focusing: imitate large lens
Temporal focusing: Illumination should also be an ellipse

Challenge:
non-specular photons

Solution:
use a large lens? expensive!!

temporal focusing:
imitate large lens
Design choices for temporal focusing

- **Ellipse size**
- **Ellipse thickness**

**Time-of-flight cameras**
- ICCD
- SPAD
- PMD
- Kinect
- Streak

Pedireddla et al. ICCP 2019.
Rendering non-line-of-sight imaging by temporal focusing

Gate width: 4 ps (0.4% scene)
Rendering time: 3 hr

* simulation results
Design choices

- **Ellipse size**: Large ellipses result in better resolution.
- **Ellipse thickness**: Thinner ellipses result in better resolution, but loses light.
- **Temporal resolution**: High resolution time-gate results in better resolution, but loses light.
Hardware prototype

Size of ellipses

Thickness of ellipse (laser spot size)

Measured Jitter

Pediredla et al. ICCP 2019.
Hardware prototype

optical component

- picosecond laser
- SPAD
- illumination galvo
- imaging galvo

renderer driven optimization

unoptimized

optimized

Pediredla et al. ICCP 2019.
Hardware prototype

MEMS
Axicon
SLM
OPA

illumination galvo
imaging galvo

optical component

picosecond laser
supercontinuum laser
femtosecond laser
streak
ICCD
kinect
PMD

unoptimized
optimized

renderer driven optimization

Pediredla et al. ICCP 2019.
Results: scanning limited ROI

scanning entire hidden scene

1.5 m
9 cm

previous techniques

7 cm

7 cm

scanning entire hidden scene

1.5 m
9 cm

temporal focusing
Results: scanning limited ROI

SNR of temporal focusing is $>10\times$ higher for small ROI
Results: real-time occupancy detection

Detected signal at focused point as a function of time

Number of photons detected

Time passed (in 100 ms interval)

Measured transient and selected time bin

Time of arrival (in 8 bit interval)

Number of photons

focused voxel (hidden)
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
- tactile sensor design
- differentiable renderer
- inverse problems
Focusing light inside tissue
Focusing light inside tissue
Focusing light inside tissue
Focusing light inside tissue
Focusing light inside tissue
Focusing light inside tissue
Focusing light inside tissue

tumor
Focusing light inside tissue
Focusing light inside tissue

Gradient Refractive Index (GRIN) waveguide

virtual GRIN waveguide

tumor
Focusing light inside tissue

Gradient Refractive Index (GRIN) waveguide

virtual GRIN waveguide
Focusing light inside tissue

Gradient Refractive Index (GRIN) waveguide

virtual GRIN waveguide

tumor
Focusing light inside tissue

Gradient Refractive Index (GRIN) waveguide

virtual GRIN waveguide

tumor
Focusing light inside tissue

Gradient Refractive Index (GRIN) waveguide

virtual GRIN waveguide

tumor
Ultrasonic light guiding inside tissue

[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]
Ultrasonic light guiding inside tissue

- Camera
- Ultrasonic array
- Laser
- 45° mirror

References:
[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]
Ultrasonic light guiding inside tissue

- camera
- ultrasonic array
- laser
- 45° mirror

[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]
Ultrasonic light guiding inside tissue

- Camera
- Ultrasonic array
- Laser
- 45° mirror

[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]
Ultrasonic light guiding inside tissue

- **Camera**
- **Ultrasound array**
- **Laser**
- **45° mirror**

Ultrasound off

Ultrasound on

[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]
Ultrasonic light guiding inside tissue

High-dimensional, highly-non-linear design problem:
- ultrasound frequency
- ultrasound voltage
- placement of transducers
- shape of waveguides
- waveform shape
- and more...
Ultrasonic light guiding inside tissue

High-dimensional, highly-non-linear design problem:
- ultrasound frequency
- ultrasound voltage
- placement of transducers
- shape of waveguides
- waveform shape
- and more...

Efficiently explore using rendering
Ultrasonic light guiding inside tissue

High-dimensional, highly-non-linear design problem:
• ultrasound frequency
• ultrasound voltage
• placement of transducers
• shape of waveguides
• waveform shape
• and more...

Efficiently explore using rendering
Build first rendering algorithm
Rendering continuous refraction and scattering
Rendering continuous refraction and scattering
Rendering continuous refraction and scattering

continuous refraction
Rendering continuous refraction and scattering

Continuous refraction
Rendering continuous refraction and scattering

continuous refraction + scattering
Rendering continuous refraction and scattering

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]

[Chandrasekhar, book, 1960]
[Lenoble, book, 1985]
[Lafortune and Willems, 1996]
[Cammarano and Jensen, 2002]
[Guttirez et al., Com. and Graph. 2006]
[Jarosz et al., Comp. Graph. forum, 2008]
[Jakob et al., ToG, 2010]
[Jarosz et al., ToG, 2011]
[Pediredla et al., JBO, 2016]
[Novak et al., Comp. Graph. forum, 2018]
[Bitterli et al., ToG, 2018]
Rendering continuous refraction and scattering

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

Snell’s law

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

Snell’s law

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

Snell’s law

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languenou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

Snell’s law

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

Snell’s law

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

refractive ray tracing

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

refractive ray tracing

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

refractive ray tracing

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttirez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

refractive ray tracing

[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Guttierrez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]
Rendering continuous refraction and scattering

radiative transfer equation

[Chandrasekhar, book, 1960]
[Lenoble, book, 1985]
[Lafortune and Willems, 1996]
[Cammarano and Jensen, 2002]
[Guttirez et al., Com. and Graph. 2006]
[Jarosz et al., Comp. Graph. forum, 2008]
[Jakob et al., ToG, 2010]
[Jarosz et al., ToG, 2011]
[Pediredla et al., JBO, 2016]
[Novak et al., Comp. Graph. forum, 2018]
[Bitterli et al., ToG, 2018]
Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):
Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath
Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):
1. trace a random emitter subpath
2. trace a random sensor subpath
Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):
1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a straight line
Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):
1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a straight line/curve

\[ \mathbf{x}_0 \rightarrow \mathbf{x}_1 \rightarrow \mathbf{x}_2 \rightarrow \mathbf{x}_3 \rightarrow \mathbf{x}_k \rightarrow \mathbf{x}_{k-1} \rightarrow \mathbf{x}_{k-2} \rightarrow \cdots \]
Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):
1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a straight line curve
Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a straight line curve
Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a straight line curve
Rendering continuous refraction and scattering

\[
\frac{dL}{d\nu_0} = (x_s^* - y)^T \frac{dx_s^*}{d\nu_0},
\]

\[
\frac{dx_s^*}{d\nu_0} = \frac{\partial x_s^*}{\partial \nu_0} + \frac{\partial x_s^*}{\partial s^*} \frac{ds^*}{d\nu_0}.
\]

\[g(x_s^*) = 0 \implies \frac{dg(x_s^*)}{dx_s^*} \frac{dx_s^*}{d\nu_0} = 0 \]

\[
\frac{ds^*}{d\nu_0} = -\frac{dg(x_s^*)}{dx_s^*} \frac{\partial x_s^*}{\partial \nu_0}
\]

\[
\frac{dx_s^*}{d\nu_0} = I_{3 \times 3} - \left( \frac{\partial x_s^*}{\partial s^*} \frac{dg(x_s^*)}{dx_s^*} \right) \frac{\partial x_s^*}{\partial \nu_0}.
\]

---

**Algorithm 2: Symplectic integration for derivative tracing**

**Input:** \( n(x), \nabla n(x), H_n(x), \text{ray}(x,v), \frac{\partial x_s^*}{\partial \nu_0} = O_3, \frac{\partial v_s^*}{\partial \nu_0} = I_3, \)

nSteps, \( s = \text{step size} \)

**Output:** \( \frac{\partial x_s^*}{\partial \nu_0}, \frac{\partial v_s^*}{\partial \nu_0} \)

for \( i = 1 : n\text{Steps} \) do

\[
\text{ray}.v^+ = 0.5s\nabla n(\text{ray}.x);
\]

\[
\frac{\partial v_s^*}{\partial \nu_0} = 0.5sH_n(\text{ray}.x) \frac{\partial x_s^*}{\partial \nu_0};
\]

\[
\text{ray}.x^+ = s - \frac{\text{ray}.v}{n(\text{ray}.x)};
\]

\[
\frac{\partial x_s^*}{\partial \nu_0} = \left( -\frac{\nabla n(\text{ray}.x)}{n(\text{ray}.x)^2} \frac{\partial x_s^*}{\partial \nu_0} + \frac{1}{n(\text{ray}.x)} \frac{\partial v_s^*}{\partial \nu_0} \right);
\]

\[
\text{ray}.v^+ = 0.5s\nabla n(\text{ray}.x);
\]

\[
\frac{\partial v_s^*}{\partial \nu_0} = 0.5sH_n(\text{ray}.x) \frac{\partial x_s^*}{\partial \nu_0};
\]

end

---

Pediredla et al. *Path Tracing Estimators for Refractive Radiative Transfer*, TOG 2020
Application: simulate Luneburg lenses

https://en.wikipedia.org/wiki/Luneburg_lens
Application: simulate Luneburg lenses

https://en.wikipedia.org/wiki/Luneburg_lens
Application: simulate Luneburg lenses

https://en.wikipedia.org/wiki/Luneburg_lens
Application: simulate Luneburg lenses

https://en.wikipedia.org/wiki/Luneburg_lens

\[ n(x) \]
Luneburg lenses
Application: transient rendering

constant refractive index  continuous refractive index
Application: transient rendering

constant refractive index

Time: 0.02 ns

continuous refractive index

Time: 0.02 ns
Application: transient rendering

constant refractive index

continuous refractive index

Time: 1.54 ns

Time: 1.54 ns
Application: transient rendering

constant refractive index

continuous refractive index

Time: 2.50 ns
Application: focusing light inside tissue

High-dimensional, highly-non-linear design problem:
• ultrasound frequency
• ultrasound voltage
• placement of transducers
• waveform shape
• and more...

Efficiently explore using rendering
Rendering virtual ultrasonic waveguides

real measurement

BDPT
(our technique)

photon mapping
(previous technique)
Validation of simulated data

first focus

50 μm

second focus

experimental data
Validation of simulated data

Experimental data

Rendered data

First focus

50 μm

Irradiance (a. u.)

Second focus

10⁻⁵

10⁻⁴

10⁻³

10⁻²

10⁻¹

10⁰

10⁻⁴

10⁻³

10⁻²

10⁻¹

10⁰
Validation of simulated data

Experimental data vs. rendered data for first and second focus positions.

Irradiance (a.u.) vs. radial distance (μm) for experimental and rendered data, showing close agreement.

Inset 1: Detailed comparison near the focus points.
Inset 2: Expanded view of the peak irradiance region.
Optimized configurations are better than ideal external lens

On human bladder (10 scattering lengths, 2.67 mm thick)
- 50% higher focusing performance than external lens.
- 300% higher focusing performance than previous designs.

On brain tissue (50 scattering lengths, 7.5 mm thick)
- 15% higher focusing performance than external lens.
- Experimentally validated on tissue phantoms.

[Pediredla et al., to appear in Nature Communications]
Optimized configurations are better than ideal external lens

On human bladder (10 scattering lengths, 2.67 mm thick)
- 50% higher focusing performance than external lens.
- 300% higher focusing performance than previous designs.

On brain tissue (50 scattering lengths, 7.5 mm thick)
- 15% higher focusing performance than external lens.
- Experimentally validated on tissue phantoms.

[Pediredla et al., to appear in Nature Communications]
Physics-based rendering and its applications to computational imaging

**Forward Rendering**
- Accurate and efficient simulation
- Virtually design sensors, optics, and algorithms

**Inverse Rendering**
- Accurate and efficient differentiable simulation
- Tractably solve general inverse problems

- Time-of-flight imaging
- Non-line-of-sight imaging
- Acousto-optic lensing
- Ultrafast light scanning
- Speckle imaging
- Tactile sensor design
- Differentiable renderer
- Inverse problems
scanning with galvos (1 kHz) is slow due to moving parts
our scanning technique (1MHZ) is fast without moving parts

1000 × faster
*not up to scale
our scanning technique (1MHz) is fast without moving parts

projector  

microscopy  

lidar

1000 × faster  
*not up to scale
our scanning technique (1MHZ) is fast without moving parts

- ultrafast optics
- physics

projector

microscopy

lidar

1000 × faster
*not up to scale

physics
ultrafast optics

1000 × faster
*not up to scale
our scanning technique (1MHz) is fast without moving parts

- ultrafast optics
- synchronization

projector
microscopy
lidar

1000 × faster
*not up to scale
our scanning technique (1MHz) is fast without moving parts

- ultrafast optics
- synchronization
- signal processing

1000 × faster

*not up to scale

projector
microscopy
lidar
Physics sound, light, and matter interaction

2D visualization

water
Physics sound, light, and matter interaction

2D visualization

water
Physics sound, light, and matter interaction

2D visualization

transducer

water
Physics sound, light, and matter interaction

nm scale
(exaggerated)
Physics sound, light, and matter interaction
Physics sound, light, and matter interaction
Physics sound, light, and matter interaction

Focus travels at the speed of 1.5 km/s
Ultrafast optics and synchronization

ultrafast synchronized optics that illuminates same spot
Ultrafast optics and synchronization

2D visualization

transducer

illuminating two spots per cycle
Dot projector

transducer-1 phase

transducer-1

transducer-2

transducer-2 phase
Hardware experiments

galvo

mirror

ultrasound transducers

SPAD

beam splitter

camera

lens

laser
Hardware results: dot projector

- galvo steering (1 kHz)
- proposed steering (1 MHz)
- mirror
- camera
- laser
- galvo (moving mirrors)
- ultrasound transducers (without moving parts)
- camera

exposure = 1 ms
Hardware results: dot projector

Galvo steering (1 kHz)

Proposed steering (1 MHz)

Laser lens

Ultrasound transducers (without moving parts)

Exposure = 50 ms

Exposure = 50 ms
Hardware results: Lidar

- Lidar beam splitter
- SPAD
- Laser

Depth map (cm) transient peak value

- 20 (photons)
- 16
- 12
- 8
- 4
- 130
- 140
- 150
- 160
- 170
- 180 (cm)

100 × faster than standard lidar
Hardware results: adaptive depth measurement

standard galvo (depth error = 51.3 cm)

our technique (depth error = 3.28 cm)

15 × depth accuracy
Limitation

Pediredla et al. CVPR 2023.
Physics-based rendering and its applications to computational imaging

**forward rendering**

- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**

- accurate and efficient differentiable simulation
- tractably solve general inverse problems

---

time-of-flight imaging  
non-line-of-sight imaging  
acousto-optic lensing  
ultrafast light scanning  
speckle imaging  
tactile sensor design  
differentiable renderer  
inverse problems
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning

- speckle imaging
- tactile sensor design
- differentiable renderer
- inverse problems
Coffee break
See you at 3.30 PM
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
- tactile sensor design
- differentiable renderer
- inverse problems
Physics-based rendering and its applications to computational imaging

forward rendering
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

inverse rendering
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

• time-of-flight imaging
• non-line-of-sight imaging
• acousto-optic lensing
• ultrafast light scanning

• speckle imaging
• tactile sensor design
• differentiable renderer
• inverse problems
Rendering wave-optics effects

what real laser images look like
what real laser images look like
Rendering wave-optics effects

- speckle: noise-like pattern
- what real laser images look like
- what standard Monte Carlo renderings look like
Rendering wave-optics effects

- Speckle: noise-like pattern
- What real laser images look like
- What standard Monte Carlo renderings look like
- Projected speckle image
- Scattering volume

Laser beam
Rendering wave-optics effects

speckle: noise-like pattern
what real laser images look like

what standard Monte Carlo renderings look like

projected speckle image

scattering volume

laser beam
Applications of memory effect

**Non-invasive single-shot imaging through scattering layers and around corners via speckle correlations**

Ori Katz, Pierre Heidmann, Mathias Fink, and Sylvain Gigan

Optical imaging through and inside complex samples is a difficult challenge with important applications in many fields. The fundamental problem is that inhomogeneous samples such as biological tissue randomly scatter and diffuse light. A central problem is to prevent the formation of diffraction-limited images. Despite many recent advances, no current method can perform non-invasive imaging in real-time using diffused light. Here, we show that, owing to the ‘memory effect’ for speckle correlations, we can achieve a single high-resolution imaging of the scattered light, captured with a standard camera, encodes sufficient information to image through visually opaque layers and around corners using spatially incoherent light and various samples, from white paint to dynamic biological samples. Our single-shot lensless technique, which does not require wavefront-shaping nor time-gated or interferometric detection, is demonstrated here using a camera phone. It has the potential to enable imaging in currently inaccessible scenarios.

Diffractive-limited optical imaging is an indispensable tool in many fields of research, such as optometry, cell biology and material sciences. However, the inherent scattering of complex materials, such as biological tissue, induces light scattering, which diffuses any optical beam into a complex speckle pattern, limiting the resolution and penetration depth of this optical imaging technique. Many approaches to overcome this fundamental, yet practical, problem have been put forward over the years, with promising requirements in holography dating back to just a few years after the invention of the laser. However, in practice, no approach allows real-time imaging using diffused light. Modern environments that are based on using only uncoated, ‘bullish’, light, such as optical coherence tomography and two-photon microscopy, have proven very useful, but are inherently limited to shallow depths where a measurable amount of unscattered photons is present. Adapting optical techniques can near-perfectly correct for distortions using deformable mirrors, but require the presence of a bright point-source ‘guide star’ or a high initial image contrast. Recent exciting advances in controlled wavefront shaping have allowed focusing and imaging through highly scattering samples. However, these techniques either require initial access to unscattered light or a highly scattering medium, or, the presence of a high-quality ‘guide star’ or a known object is required. Here, we propose an alternative approach to the ‘small angle’ imaging, which does not require a ‘guide star’ or a high initial image contrast. We introduce a technique that exploits the ‘memory effect’ for speckle correlations, captured with a standard camera, encodes sufficient information to image through visually opaque layers and around corners using spatially incoherent light. This technology is demonstrated here using a camera phone. It has the potential to enable imaging in currently inaccessible scenarios.

**Principle**

A schematic of the experiment for imaging through a scattering medium, as well as a numerical example, of the scattering medium shown in Fig. 1a–d. An object is hidden behind a wall at depth D behind a scattering medium of thickness L. The object is illuminated by a collimated laser beam propagating through a scattering medium. The scattered light is detected by the camera and the image is reconstructed. The camera is placed at a distance of the scattering medium behind the object. The scattering medium is illuminated by an aberration-free diffraction-limited laser beam. The scattered light is captured by the camera and the image is reconstructed. The camera is placed at a distance D behind the object. The scattering medium is illuminated by an aberration-free diffraction-limited laser beam. The scattered light is captured by the camera and the image is reconstructed.
Speckle-based fluorescence microscopy

- Fluorescent particles
- Scattering
- Microscope objective
- Sample
- Captured image
- Prototype

Diagram showing the setup of the speckle-based fluorescence microscopy system.
Speckle-based fluorescence microscopy

fluorescent particles → scattering microscope objective and sample → captured image → scattering-free image

Prototype setup with moving laser source, tube lenses, beam splitter, objectives, scattering sample, and camera.
Use the memory effect to image through scattering

\[ O \rightarrow I_{\text{ideal}} \]

\[ O \rightarrow I = O \ast S \]

[Katz et al., 2014]
Use the memory effect to image through scattering

$$I_{\text{ideal}} = O \ast S$$

[Katz et al., 2014]
Use the memory effect to image through scattering

\[ O \rightarrow \text{imaging system} \rightarrow \text{Ideal image} \]

\[ I = O \ast S \]

[Phase Retrieval]

[Katz et al., 2014]
Use the memory effect to image through scattering

\[ O \ast S = I_{\text{ideal}} \]

Phase Retrieval

[O]  \[ C(\theta) \]

memory-effect range

[θ_max]  \[ \theta \]

Autocorrelation

camera image's autocorrelation

reconstruction

[O]  \[ I = O \ast S \]

Phase Retrieval

[Katz et al., 2014]
Speckle-based fluorescence microscopy

[PIs: Gkioulekas, Levin]
Speckle-based fluorescence microscopy

fluorescent particles

scattering microscope objective captured sample

captured image

autocorrelation and phase retrieval

scattering-free image

Speckle-based fluorescence microscopy

[PIs: Gkioulekas, Levin]
Speckle-based fluorescence microscopy

Performance strongly depends on:
- speckle statistics
- image priors
- tissue parameters
Recap: Monte Carlo (volumetric) rendering

Image = \int_{\text{paths}} f(\text{path})
Recap: Monte Carlo (volumetric) rendering

\[
\text{Image} = \int f(\text{path}) \quad \text{paths}
\]

[Bar et al., 2019]
Recap: Monte Carlo (volumetric) rendering

Path contribution, depends on the scattering material

\[
\text{Image} = \int_{\text{paths}} f(\text{path})
\]

[Bar et al., 2019]
Recap: Monte Carlo (volumetric) rendering

\[
\text{Image} = \int f(\text{path})
\]

Path contribution, depends on the scattering material

[Bar et al., 2019]
Recap: Monte Carlo (volumetric) rendering

\[ \text{Image} = \int f(\text{path}) \]

Path contribution, depends on the scattering material

volumetric density (extinction coefficient) \( \sigma \)

[Bar et al., 2019]
Recap: Monte Carlo (volumetric) rendering

\[ \text{Image} = \int f(\text{path}) \]

Path contribution, depends on the scattering material

- Volumetric density (extinction coefficient) \( \sigma \)
- Scattering albedo \( a \)

[Bar et al., 2019]
Recap: Monte Carlo (volumetric) rendering

Image = \int f(path)

Path contribution, depends on the scattering material

volumetric density (extinction coefficient) \sigma
scattering albedo a
phase function \rho_\theta

[Bar et al., 2019]
Recap: Monte Carlo (volumetric) rendering

Image = \int f(\text{path})

Path contribution, depends on the scattering material

material = \begin{pmatrix} \sigma \\ a \\ p_\theta \end{pmatrix}

[Bar et al., 2019]
Recap: Monte Carlo (volumetric) rendering

\[ \text{Image} = \int f(\text{path}) \]

Path contribution, depends on the scattering material

To compute efficiently: importance sampling of paths

\[
\text{material} = \begin{pmatrix}
\sigma \\
a \\
p_{\theta}
\end{pmatrix}
\]

[Bar et al., 2019]
Memory effect: simulate covariance of speckle images

Covariance = $\int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$

[Bar et al., 2019]
Memory effect: simulate covariance of speckle images

$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

Need to consider products of pairs of paths

[Bar et al., 2019]
Memory effect: simulate covariance of speckle images

\[
\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)
\]

Need to consider products of *pairs* of paths

[Bar et al., 2019]
Memory effect: simulate *covariance* of speckle images

\[
\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)
\]

Need to consider products of *pairs* of paths

[Bar et al., 2019]
Memory effect: simulate covariance of speckle images

\[
\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)
\]

Need to consider products of pairs of paths

[Bar et al., 2019]
Memory effect: simulate \textit{covariance} of speckle images

Each path contributes a complex number \( u \):

\[ \text{light}_1 \rightarrow \text{view}_1 \]

\[ \text{light}_2 \rightarrow \text{view}_2 \]

Path 1: \( \text{light}_1 \rightarrow \text{view}_1 \)

Path 2: \( \text{light}_2 \rightarrow \text{view}_2 \)

Covariance:

\[ \int u(\text{path}_1) \cdot u^*(\text{path}_2) \]

\( u = |u| e^{i \cdot \text{phase}} \)

Need to consider products of \textit{pairs} of paths

Each path contributes a complex number \( u \)

[Bar et al., 2019]
Memory effect: simulate covariance of speckle images

Covariance = \int u(path_1) \cdot u^*(path_2)

u = |u| \ e^{i \cdot \text{phase}}

Need to consider products of pairs of paths

Each path contributes a complex number u

phase \propto \text{Length (path)}

[Bar et al., 2019]
Memory effect: simulate covariance of speckle images

Each path contributes a complex number $u$

$$u = |u| e^{i \cdot \text{phase}}$$

Need to consider products of pairs of paths

Each path contributes a complex number $u$

$$\text{phase} \propto \text{Length ( path )}$$

$$\Delta \text{phase} \propto \text{Length ( path}_1\text{ )} - \text{Length ( path}_2\text{ )}$$

$$\text{Covariance} = \int u(\text{path}_1) \cdot u^*(\text{path}_2)$$

[Bar et al., 2019]
Memory effect: simulate covariance of speckle images

Observation: need to consider only path pairs that share the same nodes (except start and end)

All other path pairs are averaged out in the integration

Covariance = \[ \int u(\text{path}_1) \cdot u^*(\text{path}_2) \]
Memory effect: simulate covariance of speckle images

Covariance = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)

Observation: need to consider only path pairs that share the same nodes (except start and end)

All other path pairs are averaged out in the integration

[Bar et al., 2019]
Memory effect: simulate covariance of speckle images

Covariance = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)

Observation: need to consider only path pairs that share the same nodes (except start and end)

Efficiency of our algorithm comes from the ability to neglect all other paths pairs

All other path pairs are averaged out in the integration

[Bar et al., 2019]
Memory effect: simulate **covariance** of speckle images

Efficiency of our algorithm comes from the ability to neglect all other paths pairs

**Covariance** = \( \int u(path_1) \cdot u^*(path_2) \)

Observation: need to consider only path pairs that share the same nodes (except start and end)

[Bar et al., 2019]
Comparison with wave-equation solver and real measurements

match wave equation solvers, $10^5$x faster

match real measurements of memory effect

[Bar et al., 2019]
Speckle-based fluorescence microscopy

[Alterman et al., 2021]
Speckle-based fluorescence microscopy

- Fluorescent particles
- Scattering microscope objective sample
- Captured image
- Autocorrelation and phase retrieval
- Scattering-free image

[Alterman et al., 2021]
Speckle-based fluorescence microscopy

Performance strongly depends on:

• speckle statistics
• image priors
• tissue parameters

[Alterman et al., 2021]
Evaluate the memory effect

Analytical solution based on diffusion approximation

[Alterman et al., 2021]
Evaluate the memory effect

This approximation is correct for some materials and configurations

Analytical solution based on diffusion approximation

[Alterman et al., 2021]
Evaluate the memory effect

This approximation is correct for some materials and configurations.

In practice ME extent is often wider.

Analytical solution based on diffusion approximation

[Alterman et al., 2021]
Evaluate the memory effect

This approximation is correct for some materials and configurations.

In practice ME extent is often wider.

Currently: measured *empirically* in the lab.

Analytical solution based on diffusion approximation.

[Alterman et al., 2021]
Evaluate the memory effect

This approximation is correct for some materials and configurations. In practice, ME extent is often wider than predicted.

Currently: measured empirically in the lab.

Analytical solution based on diffusion approximation

Now we can efficiently compute ME curves for all scattering parameters.

[Alterman et al., 2021]
Evaluate the memory effect

Analytical solution based on diffusion approximation

This approximation is correct for some materials and configurations. In practice, ME extent is often wider than initially computed empirically in the lab. Currently, measured empirically.

Now we can efficiently compute ME curves for all scattering parameters.

[Alterman et al., 2021]
Better algorithms for fluorescence microscopy

[Alterman et al., 2021]
Acquisition of scattering materials

Use differentiable speckle rendering to recover material parameters from speckle images

- Acquisition camera: records speckle correlations
- Rotating illuminator: two laser beams at 4° separation
- Motorized sample mount: 8 degrees of freedom
- Material samples
- High-power coherent monochromatic laser
- Optical fiber
Rendering wave optics is a very active area.

A Generic Framework for Physical Light Transport

SHLOMI STEINBERG, University of California, Santa Barbara, USA
LING-QI YAN, University of California, Santa Barbara, USA
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic imaging
- ultrafast light scanning
- speckle imaging
- tactile sensor design
- differentiable renderer
- inverse problems
Why tactile sensing?

Object perception
[Huang et.al. 2022]

Advanced manufacturing
[Li et.al. 2014]

Robotic manipulation
[Yuan et.al. 2018]
[Wilson et.al. 2023]

Neuroprosthetics
[Gu et.al. 2023]
Tactile sensors

BioTac
ionically-conductive fluid based

ReSkin
Magnetic field based

GelSight
Vision-based
Tactile sensors

BioTac
ionically-conductive fluid based

ReSkin
Magnetic field based

GelSight
Vision-based
Vision-based tactile sensors: **working principle**

[Johnson and Adelson, 2021]
Vision-based tactile sensors: photometric stereo

With photometric stereo, GelSight can encode surface normals as an RGB image.

[Johnson and Adelson, 2021]
Vision-based tactile sensors: design variants

**FlatGel GelSight**
Dong et. al. 2017
Agarwal et. al. 2021

**GelSlim Family**
Donlon et. al. 2018
Ma et. al. 2019
Hogan et. al. 2020
Taylor et. al. 2021

**RoundTip GelSight**
Romero et. al. 2020
Designing VBTS is hard
Designing VBTS is hard

- Diversity of sensor shape and required form-factor

[Agarwal et al., 2023]
Designing VBTS is hard

- Diversity of sensor shape and required form-factor
- Complex light interaction
Curved tactile sensor

[Agarwal et al., 2023]
Curved tactile sensor

[Agarwal et al., 2023]
Curved tactile sensor: *working principle*

[Agarwal et al., 2023]
Curved tactile sensor: **working principle**

[Agarwal et al., 2023]
Curved tactile sensor: *working principle*

Refraction at rough interface

[Agarwal et al., 2023]
Curved tactile sensor: *working principle*

Refraction at rough interface

Reflection at glossy surfaces

[Agarwal et al., 2023]
Curved tactile sensor: **working principle**

- Refraction at rough interface
- Reflection at glossy surfaces

[Agarwal et al., 2023]
Curved tactile sensor: working principle

- Refraction at rough interface
- Reflection at glossy surfaces

[Agarwal et al., 2023]
Curved tactile sensor: *working principle*

- Refraction at rough interface
- Reflection at glossy surfaces

[Agarwal et al., 2023]
Curved tactile sensor: *working principle*

- Refraction at rough interface
- Reflection at glossy surfaces

[Agarwal et al., 2023]
Sensor design framework: \textit{optical simulation}

- Sensor geometry
- Material specifications
- Placement of camera and lights
Sensor design framework: *optical simulation*

- Sensor geometry
- Material specifications
- Placement of camera and lights

Section view

Tactile image  [Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

Diffuse surface

(Relatively) easy cases

Hard case

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

![Diagram showing optical simulation](image)

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

![Diagram of sensor design framework](image)

(Relatively) easy cases  
Hard case

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

![Diagram](Image)

[Diffuse surface](Image) (Relatively) easy cases

[Mirror](Image) Hard case

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

![Diffuse surface and Mirror](image)

*(Relatively) easy cases*  
*Hard case*

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

[Diffuse surface](#)

- (Relatively) easy cases

[ Mirror](#)

- Hard case

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

Diffuse surface

(Mirror) Smooth glass

(Relatively) easy cases

Hard case

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

  - Diffuse surface
  - Mirror
  - Smooth glass

  (Relatively) easy cases

  Hard case

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

[Diffuse surface] (Relatively) easy cases

[Mirror] Smooth glass

[Diffuse/Glossy surface] Rough glass

[Agarwal et al., 2023]
Sensor design framework: \textit{optical simulation}

- What makes simulation challenging

Diffuse surface

(Relatively) easy cases

Mirror

Smooth glass

Hard case

Diffuse/Glossy surface

Rough glass

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging

  - Diffuse surface
  - Mirror
  - Diffuse/Glossy surface

  (Relatively) easy cases

  Hard case

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination
- Require **Markov Chain Monte Carlo** rendering techniques
  - Key idea: Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination

- Require [Markov Chain Monte Carlo](https://en.wikipedia.org/wiki/Monte_Carlo_method) rendering techniques
  - Key idea: Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination

- Require **Markov Chain Monte Carlo** rendering techniques
  - Key idea: Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Sensor design framework: \textit{optical simulation}

• What makes simulation challenging
  • SDS light paths
  • Indirect illumination

• Require \textbf{Markov Chain Monte Carlo} rendering techniques
  • Key idea \hspace{1cm} \textcolor{red}{\textbf{Slowly mutate paths to generate useful paths}}

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination
- Require **Markov Chain Monte Carlo** rendering techniques
  - Key idea: Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Sensor design framework: \textit{optical simulation}

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination

- Require \textbf{Markov Chain Monte Carlo} rendering techniques
  - Key idea \rightarrow Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination
- Require [Markov Chain Monte Carlo](https://en.wikipedia.org/wiki/Monte_Carlo_method) rendering techniques
  - Key idea: Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination

- Require **Markov Chain Monte Carlo** rendering techniques
  - Key idea: Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination

- Require **Markov Chain Monte Carlo** rendering techniques
  - Key idea: Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Sensor design framework: \textit{optical simulation}

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination

- Require \textit{Markov Chain Monte Carlo} rendering techniques
  - Key idea: Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Sensor design framework: *optical simulation*

- What makes simulation challenging
  - SDS light paths
  - Indirect illumination

- Require **Markov Chain Monte Carlo** rendering techniques
  - Key idea: Slowly mutate paths to generate useful paths

[Agarwal et al., 2023]
Optimization:
- Stochastic Gradient Descent (SGD)

MCMC sampling:
- Langevin Monte Carlo (LMC)

[Luan et al., 2020]
Optimization problem:
\[
\max_x f(x)
\]

Gradient descent/ascent:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1})
\]

[Luan et al., 2020]
GD OVERVIEW

Optimization problem:

$$\max_x f(x)$$

Gradient descent/ascent:

$$x_t = x_{t-1} + s_{t-1}\nabla f(x_{t-1})$$

[Luan et al., 2020]
Optimization problem:
\[
\max_x f(x)
\]

Gradient descent/ascent:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1})
\]

[Luan et al., 2020]
Optimization problem:
\[
\max_x f(x)
\]

Gradient descent/ascent:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1})
\]

[Luan et al., 2020]
Optimization problem:
$$\max_x f(x)$$

Gradient descent/ascent:
$$x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1})$$

[Luan et al., 2020]
GD OVERVIEW

Optimization problem:
\[
\max_x f(x)
\]

Gradient descent/ascent:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1})
\]

[Luan et al., 2020]
GD OVERVIEW

Optimization problem:
\[
\max_x f(x)
\]

Gradient descent/ascent:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1})
\]

scalar step size

[Luan et al., 2020]
Sampling problem:

\[ x_t \sim f \]

Langevin MC:

\[ x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I) \]

Apply Metropolis Hastings to accept/reject

Paul Langevin

[Luan et al., 2020]
Sampling problem: 
\[ x_t \sim f \]

Langevin MC:
\[ x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I) \]

Apply Metropolis Hastings to accept/reject

[Luan et al., 2020]
Sampling problem: \( x_t \sim f \)

Langevin MC:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I)
\]

Apply Metropolis Hastings to accept/reject

[Luan et al., 2020]
Sampling problem:
\[ x_t \sim f \]

Langevin MC:
\[ x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I) \]

Apply Metropolis Hastings to accept/reject

[LMC OVERVIEW]

[Luan et al., 2020]
Sampling problem:
\[ x_t \sim f \]

Langevin MC:
\[ x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I) \]

Apply Metropolis Hastings to accept/reject

[Luan et al., 2020]
Sampling problem:
\[ x_t \sim f \]

Langevin MC:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I)
\]

Apply Metropolis Hastings to accept/reject

[Luan et al., 2020]
Sampling problem:

\[ x_t \sim f \]

Langevin MC:

\[ x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I) \]

Apply Metropolis Hastings to accept/reject

[\text{Luan et al., 2020}]
LMC OVERVIEW

Sampling problem:
\[ x_t \sim f \]

Langevin MC:
\[ x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I) \]

Apply Metropolis Hastings to accept/reject

[Luan et al., 2020]
Sampling problem:

\[ x_t \sim f \]

Langevin MC:

\[ x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I) \]

Apply Metropolis Hastings to accept/reject

[LUAN et al., 2020]
Sampling problem:

\[ x_t \sim f \]

Langevin MC:

\[ x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I) \]

Apply Metropolis Hastings to accept/reject

[Luan et al., 2020]
Sampling problem:

\[ x_t \sim f \]

Langevin MC:

\[ x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I) \]

Apply Metropolis Hastings to accept/reject

[Luan et al., 2020]
GD VS. LMC

Optimization:
\[
\max_x f(x)
\]
GD:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1})
\]

Sampling:
\[
x_t \sim f
\]
LMC:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I)
\]

[Luan et al., 2020]
GD VS. LMC

Optimization:
\[
\max_x f(x)
\]
GD:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1})
\]

Sampling:
\[
x_t \sim f
\]
LMC:
\[
x_t = x_{t-1} + s_{t-1} \nabla f(x_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 I)
\]

[\text{Luan et al., 2020}]

convert optimization algorithms into rendering algorithms
LANGEVIN MONTE CARLO RENDERING WITH GRADIENT-BASED ADAPTATION

Prior Ours 10x acceleration
Evaluating optical simulation framework

Real-world prototype

[Agarwal et al., 2023]
Evaluating optical simulation framework

Real-world prototype

Rasterization

[Agarwal et al., 2023]
Evaluating optical simulation framework

Real-world prototype

Rasterization

Path tracing

[Agarwal et al., 2023]
Evaluating optical simulation framework

Rasterization

Real-world prototype

Path tracing

Our simulation result

[Agarwal et al., 2023]
Evaluating optical simulation framework

Rasterization

Real-world prototype

Our simulation result

Path tracing

[Rasterization, Path tracing]

[Agarwal et al., 2023]
Curved sensor material design

[Sensing surface]

[Agarwal et al., 2023]
Curved sensor illumination design

Light source

Spot Light  IES Light  Area Light

Background Image

Indenter Image

Color Signal

0.930  0.800  0.973
Curved sensor shape design

• Used gradient-free optimization, CMA-ES, for optimizing the sensor shape
• Optimized sensor design is 35% better than initial design
• Optimized sensor design outperforms human-expert design in 3D shape reconstruction

<table>
<thead>
<tr>
<th></th>
<th>2D Curve</th>
<th>CAD Model</th>
<th>Simulated Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Design</td>
<td><img src="image1.png" alt="Initial Design" /></td>
<td><img src="image2.png" alt="Initial Design" /></td>
<td>Score = 0.832</td>
</tr>
<tr>
<td>Human Expert Design</td>
<td><img src="image3.png" alt="Human Expert Design" /></td>
<td><img src="image4.png" alt="Human Expert Design" /></td>
<td>Score = 0.973</td>
</tr>
<tr>
<td>Optimized Design</td>
<td><img src="image5.png" alt="Optimized Design" /></td>
<td><img src="image6.png" alt="Optimized Design" /></td>
<td>Score = 1.125</td>
</tr>
</tbody>
</table>

- Sensing Surface
- Interface between shell and elastomer
- Shell Surface
- Hard shell
- Soft elastomer
Results: Robotic grasping

Robotic arm with optimized tactile sensor

Common Objects

Tactile Images

[Agarwal et al., 2023]
Results: surface inspection

Experiment setup

Text description:
Tactile Sensing

Experiment 1: Detection accuracy vs text size

Human expert design
Optimized design

Text size = 1.5 mm

Text size = 1.0 mm

“Tactile Sensing”
“Tacolle Sensing”

[Agarwal et al., 2023]
Results: surface inspection

Experiment setup

Text description: Feeling of Touch

Experiment 2: Detection accuracy vs contact location

Human expert design

Optimized design

[Agarwal et al., 2023]
Results: surface inspection

[Agarwal et al., 2023]
Physics-based rendering and its applications to computational imaging

**Forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**Inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning

- speckle imaging
- tactile sensor design
- differentiable renderer
- inverse problems
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
- tactile sensor design
- differentiable renderer
- inverse problems
Forward rendering

digital scene specification (geometry, materials, optics, light sources)

physically-accurate rendering

photorealistic simulated image
Inverse rendering

digital scene specification (geometry, materials, camera, light sources)

physically-accurate inverse rendering

photorealistic synthetic image
Inverse rendering

digital scene specification (geometry, materials, camera, light sources)

physically-accurate inverse rendering

image measurements
Analysis by synthesis (a.k.a. inverse rendering)

$$\min_{\text{unknowns } m} \left\| \text{image}(m) - \text{image}(m) \right\|^2$$

(tissue properties)
Analysis by synthesis (a.k.a. inverse rendering)

\[
\min \| \text{image}(m) \|^2 - \text{captured measurements}
\]

unknowns \( m \) (tissue properties)
Analysis by synthesis (a.k.a. inverse rendering)

\[ \text{min} \: || \text{unknowns} \: m \: (\text{tissue properties}) \: - \: \text{image}(m) \: ||^2 \]

- Solve with exhaustive search
- For all parameters \( m \)
  - Evaluate \( \text{loss}(m) \)
    - Computed with rendering

\( \text{known parameters} \: m \: (\text{tissue properties}) \: \rightarrow \: \text{physically-accurate rendering} \: \rightarrow \: \text{image}(m) \: \rightarrow \: \text{unknown parameters} \: m \: (\text{camera, shape}) \)
Analysis by synthesis (a.k.a. inverse rendering)

\[
\min \| \text{image}(m) \|^2 - \text{image}(m)
\]

unknowns \( m \) (tissue properties)

solve with gradient descent

while (not converged)

update \( m \) with \( \frac{\partial \text{loss}(m)}{\partial m} \)

computed with differentiable rendering

\( \frac{\partial \text{image}(m)}{\partial m} \)

differentiable rendering

unknown parameters \( m \) (tissue properties)

known parameters (camera, shape)
Analysis by synthesis (a.k.a. inverse rendering)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene unknowns } \pi \right) \right]$$
Analysis by synthesis (a.k.a. inverse rendering)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene unknowns } \pi \right) \right]$$
Analysis by synthesis (a.k.a. inverse rendering)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene unknowns } \pi \right) \right]$$

Stochastic gradient descent (e.g., Adam):

initialize $$\pi \leftarrow \pi_0$$

while (not converged)

update $$\pi \leftarrow \pi + \eta \cdot \frac{\text{dloss}(\pi)}{\text{d}\pi}$$
Analysis by synthesis (a.k.a. inverse rendering)

Analysis-by-synthesis optimization:

\[
\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene} \left( \text{unknowns } \pi \right) \right) \right]
\]

Stochastic gradient descent (e.g., Adam):

initialize \( \pi \leftarrow \pi_0 \)

while (not converged)

update \( \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi} \)
How do we differentiate light transport?

\[ I = \int_{\mathcal{P}} W_e(x_0, x_1) L_e(x_k, x_{k-1}) T(\bar{x}) \, dx \]

image \quad \text{space of all light paths}

\[ T(\bar{x}) = G(x_0, x_1) \prod_{j=1}^{k-1} f(x_j, x_{j+1}, x_{j-1}) G(x_j, x_{j+1}) \]

path throughput

BSDF

geometry
REMINDER (?) FROM CALCULUS
Reminder from calculus

\[
\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx \quad ? =
\]
Reminder from calculus

\[ \frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx \quad ? = \]
Reminder from calculus

Differentiation under the integral sign
Also known as the Leibniz integral rule

\[ \frac{d}{d\pi} \int_{a(\pi)} f(x, \pi) \, dx \]
Reminder from calculus

**Differentiation under the integral sign**

Also known as the Leibniz integral rule

\[
\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) \, dx \\
+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi} \\
+ \sum_{i} \left( f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi) \right) \frac{dc_i(\pi)}{d\pi}
\]
Reminder from calculus

Differentiation under the integral sign
Also known as the Leibniz integral rule

\[ \frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) \, dx \]

Move derivative inside integral

\[ + f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(\alpha(\pi); \pi) \frac{da(\pi)}{d\pi} \]

\[ + \sum_{i} \left( f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi) \right) \frac{dc_i(\pi)}{d\pi} \]
Reminder from calculus

Differentiation under the integral sign
Also known as the Leibniz integral rule

\[
\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) \, dx \quad \text{Move derivative inside integral}
\]

Account for changes in integration limits

\[
+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(\alpha(\pi); \pi) \frac{da(\pi)}{d\pi} + \sum_{i} (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi}
\]
Reminder from calculus

Differentiation under the integral sign
Also known as the Leibniz integral rule

\[ \frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) \, dx \]

Move derivative inside integral

Account for changes in integration limits

\[ + f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(\alpha(\pi); \pi) \frac{da(\pi)}{d\pi} \]

Account for discontinuities of integrand that depend on \( \pi \)

\[ + \sum_{i} \left( f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi) \right) \frac{dc_i(\pi)}{d\pi} \]
A simple example

\[ f(x, \pi) = \begin{cases} 
0 & \text{if } x < 2\pi \\
1 & \text{if } x \geq 2\pi 
\end{cases} \]

\[ \frac{d}{d\pi} \int_{0}^{4\pi} f(x, \pi) \, dx = \]
A simple example

\[ f(x, \pi) = \begin{cases} 
0 & \text{if } x < 2\pi \\
1 & \text{if } x \geq 2\pi 
\end{cases} \]

\[ \frac{d}{d\pi} \int_{0}^{4\pi} f(x, \pi) dx = \]
A simple example

\[ f(x, \pi) = \begin{cases} 
0 & \text{if } x < 2\pi \\
1 & \text{if } x \geq 2\pi 
\end{cases} \]

\[
\frac{d}{d\pi} \int_0^{4\pi} f(x, \pi) \, dx = \int_0^{2\pi} \frac{d}{d\pi} 0 \, dx + \int_{2\pi}^{4\pi} \frac{d}{d\pi} 1 \, dx \quad \text{Move derivative inside integral}
\]
A simple example

\[ f(x, \pi) = \begin{cases} 
0 & \text{if } x < 2\pi \\
1 & \text{if } x \geq 2\pi 
\end{cases} \]

\[
\frac{d}{d\pi} \int_{0}^{4\pi} f(x, \pi) \, dx = \int_{0}^{2\pi} \frac{d}{d\pi} 0 \, dx + \int_{2\pi}^{4\pi} \frac{d}{d\pi} 1 \, dx \\
\text{Move derivative inside integral}
\]

Account for changes in integration limits

\[
+ \frac{1}{\pi} \frac{d(4\pi)}{d\pi} - 0 \frac{d0}{d\pi}
\]
A simple example

\[ f(x, \pi) = \begin{cases} 
0 & \text{if } x < 2\pi \\
1 & \text{if } x \geq 2\pi 
\end{cases} \]

\[
\frac{d}{d\pi} \int_0^{4\pi} f(x, \pi) \, dx = \int_0^{2\pi} \frac{d}{d\pi} 0 \, dx + \int_{2\pi}^{4\pi} \frac{d}{d\pi} 1 \, dx
\]

Account for changes in integration limits

\[ + \frac{d(4\pi)}{d\pi} - 0 \frac{d0}{d\pi} \]

Account for discontinuities of integrand that depend on \( \pi \)

\[ + (0 - 1) \frac{d(2\pi)}{d\pi} \]
Leibniz integral rule

Differentiation under the integral sign
Also known as the Leibniz integral rule

\[
\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) \, dx
\]

Move derivative inside integral

Account for changes in integration limits

\[
+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(\alpha(\pi); \pi) \frac{da(\pi)}{d\pi}
\]

Account for discontinuities of integrand that depend on \( \pi \)

\[
+ \sum_i \left( f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi) \right) \frac{dc_i(\pi)}{d\pi}
\]
Leibniz integral rule

Differentiation under the integral sign
Also known as the Leibniz integral rule

\[
\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) \, dx + f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi} + \sum_{i} \left( f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi) \right) \frac{dc_i(\pi)}{d\pi}
\]
Leibniz integral rule

\[
\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) \, dx
\]

**Differentiation under the integral sign**
Also known as the Leibniz integral rule

Account for changes in integration limits

Account for discontinuities of integrand that depend on \( \pi \)

**Interior integral**

Move derivative inside integral

**Boundary terms**

\[+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(\alpha(\pi); \pi) \frac{da(\pi)}{d\pi} \]

\[+ \sum_i \left( f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi) \right) \frac{dc_i(\pi)}{d\pi} \]
Reynolds transport theorem

\[ \frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) \]
Reynolds transport theorem

\[ \frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) \]
Reynolds transport theorem

\[
\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) \quad ?
\]

Reynolds transport theorem [1903]
Generalization of the Leibniz rule
Reynolds transport theorem

\[
\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x, \pi)}{d\pi} dA(x) + \int_{\partial\Omega(\pi)} g(x, \pi) dl(x)
\]

Reynolds transport theorem [1903]
Generalization of the Leibniz rule
Reynolds transport theorem

\[
\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x,\pi)}{d\pi} dA(x) + \int_{\partial \Omega(\pi)} g(x, \pi) dl(x)
\]

Reynolds transport theorem [1903]
Generalization of the Leibniz rule
Reynolds transport theorem

\[
\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x,\pi)}{d\pi} dA(x) + \int_{\partial\Omega(\pi)} g(x, \pi) dl(x)
\]

Boundary domain

\[\text{discontinuity points } \cup \text{ boundary of domain } \Omega \]

(if they depend on \(\pi\))
Reynolds transport theorem

\[
\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x,\pi)}{d\pi} dA(x) + \int_{\partial \Omega(\pi)} g(x, \pi) dl(x)
\]

Discontinuity points \( \cup \) boundary of domain \( \Omega \) (if they depend on \( \pi \))

Reynolds transport theorem [1903]
Generalization of the Leibniz rule
Reynolds transport theorem

\[
\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x,\pi)}{d\pi} dA(x) + \int_{\partial\Omega(\pi)} g(x, \pi) dl(x)
\]

Reynolds transport theorem [1903]
Generalization of the Leibniz rule
DIFFERENTIATING DIRECT ILLUMINATION
Direct illumination integral

Radiance from $x$:

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$
Direct illumination integral

Radiance from $x$:

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$

Unit hemisphere
Direct illumination integral

Radiance from $x$:

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$

Unit hemisphere
Direct illumination integral

Radiance from $x$:

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$

Reflectance (BRDF)  Incident radiance

Unit hemisphere
Direct illumination integral

Radiance from $x$:

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$

Reflectance (BRDF)  Incident radiance  Shading wrt normal $n$

Unit hemisphere
Direct illumination integral

Radiance from $x$:

$$ I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i) $$

- Reflectance (BRDF)
- Incident radiance
- Shading wrt normal $n$

Monte Carlo rendering:
Direct illumination integral

Radiance from $x$:

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$

Monte Carlo rendering:
- Sample random directions $\omega_i^s$ from PDF $p(\omega_i)$
Direct illumination integral

Radiance from $x$:

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$

Monte Carlo rendering:

- Sample random directions $\omega_i^s$ from PDF $p(\omega_i)$
- Form estimator

$$I \approx \sum_s \frac{f_r(\omega_i^s, \omega_o) L_i(\omega_i^s) (n \cdot \omega_i^s)}{p(\omega_i^s)}$$
Differential direct illumination

Differential radiance from $x$:

$$\frac{dl}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$
Differential direct illumination: local parameters

Differential radiance from $x$:

$$
\frac{dL}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}_2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)
$$
Differential direct illumination: local parameters

Differential radiance from $x$:

$$\frac{dI}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$

π: local parameters
- BRDF parameters
- shading normal
- illumination brightness
Differential direct illumination: local parameters

Differential radiance from $x$:

$$\frac{dI}{d\pi} = \int_{\mathbb{S}^2} \frac{d}{d\pi} \left\{ f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \right\} d\sigma(\omega_i)$$

$\pi$: local parameters
- BRDF parameters
- shading normal
- illumination brightness

Just move derivative inside integral
Differential direct illumination: local parameters

\[ \frac{dI}{d\pi} = \int_{\mathbb{H}^2} \frac{d}{d\pi} \left\{ f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \right\} d\sigma(\omega_i) \]

Just move derivative inside integral

Monte Carlo differentiable rendering:

- Sample random directions \( \omega_i^s \) from PDF \( p(\omega_i) \)
- Form estimator

\[ \frac{dI}{d\pi} \approx \sum_s \left( \frac{d}{d\pi} \left\{ f_r(\omega_i^s, \omega_o) L_i(\omega_i^s) (n \cdot \omega_i^s) \right\} \right) p(\omega_i^s) \]

Just differentiate numerator

\[ \pi: \text{local parameters} \]
- BRDF parameters
- shading normal
- illumination brightness

[Khungurn et al. 2015, Gkioulekas et al. 2015]
Alternative estimator

Differential radiance from $x$:

\[
\frac{dl}{d\pi} = \int_{\mathbb{S}^2} \frac{d}{d\pi} \{f_r(\omega_i, \omega_o, \pi)L_i(\omega_i)(n \cdot \omega_i)\}d\sigma(\omega_i)
\]

Just move derivative inside integral

Monte Carlo estimation:

- Sample random directions $\omega_i^s$ from PDF $p(\omega_i, \pi)$
- Form estimator

\[
\frac{dl}{d\pi} \approx \sum_s \frac{d}{d\pi} \{f_r(\omega_i^s, \omega_o, \pi)L_i(\omega_i^s)(n \cdot \omega_i^s)\} \frac{1}{p(\omega_i^s, \pi)}
\]

$\pi$: local parameters
- BRDF parameters
Alternative estimator

Differential radiance from $x$:

\[
\frac{dI}{d\pi} = \int_{\mathbb{H}_2^2} \frac{d}{d\pi} \{ f_r(\omega_i, \omega_o, \pi)L_i(\omega_i)(n \cdot \omega_i) \} d\sigma(\omega_i)
\]

Just move derivative inside integral

Monte Carlo estimation:

- Sample random directions $\omega_i^s$ from PDF $p(\omega_i, \pi)$
- Form estimator

\[
\frac{dI}{d\pi} \approx \sum_s \frac{d}{d\pi} \left\{ \frac{f_r(\omega_i^s, \omega_o, \pi)L_i(\omega_i^s)(n \cdot \omega_i^s)}{p(\omega_i^s, \pi)} \right\}
\]

$\pi$: local parameters
- BRDF parameters

Differentiate entire contribution

[Zeltner et al. 2021]
Differential direct illumination: global parameters

Differential radiance from $x$:

$$\frac{dI}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)$$

**$\pi$: global parameters**
- shape and pose of different scene elements (camera, sources, objects)
Differential direct illumination: global parameters

Differential radiance from $x$:

$$
\frac{dI}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{S}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)
$$

$$
= \int_{\mathbb{S}^2} \frac{d}{d\pi} \{f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i)\} \, d\sigma(\omega_i)
$$

Need to use full Reynolds transport theorem

$\pi$: global parameters
- shape and pose of different scene elements (camera, sources, objects)
Discontinuities in the integrand

\[
I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i)(n \cdot \omega_i) d\sigma(\omega_i)
\]

\[f(\omega_i)\]

\[f(\omega_i)\]

\[\pi: \text{size of the emitter}\]

Integrand

Discontinuous points (\(\pi\)-dependent)
Applying the Reynolds transport theorem

\[ I = \int_{\mathbb{H}^2} f(\omega_i, \omega_o) d\sigma(\omega_i) \]
Applying the Reynolds transport theorem

\[ I = \int_{\mathbb{H}^2} f(\omega_i, \omega_o) d\sigma(\omega_i) \]

\[ \frac{dI}{d\pi} = \int_{\mathbb{H}^2} \frac{df}{d\pi} d\sigma + \int_{\partial \mathbb{H}^2} g dl \]

Interior integral (same as for local parameters)

Boundary integral

[Ramamoorthi et al. 2007, Li et al. 2019]
Reparameterizing the direct illumination integral

**Hemispherical integral**

\[ I = \int_{\mathbb{H}^2} f(\omega_i) \, d\sigma(\omega_i) \]
Reparameterizing the direct illumination integral

**Hemispherical** integral

\[ I = \int_{\mathbb{H}^2} f(\omega_i) \, d\sigma(\omega_i) \]

**Surface** integral

\[ I = \int_{\mathcal{L}(\pi)} f(y \rightarrow x) \, G(x, y) \, dA(y) \]
Reparameterizing the direct illumination integral

**Hemispherical** integral

\[ I = \int_{\mathbb{H}^2} f(\omega_i) \, d\sigma(\omega_i) \]

**Surface** integral

\[ I = \int_{\mathcal{L}(\pi)} f(y \rightarrow x) \, G(x, y) \, dA(y) \]

Includes visibility, fall-off, and foreshortening terms.
Reparameterizing the direct illumination integral

**Hemispherical integral**

\[
I = \int_{\mathbb{H}^2} f(\omega_i) \, d\sigma(\omega_i)
\]

**Surface integral**

\[
I = \int_{L(\pi)} f(y \rightarrow x) \, G(x, y) \, dA(y)
\]
Reparameterizing the direct illumination integral

**Hemispherical integral**

\[ I = \int_{\mathbb{H}^2} f(\omega_i) \, d\sigma(\omega_i) \]

**Surface integral**

\[ I = \int_{\mathcal{L}(\pi)} f(y \rightarrow x) \, G(x, y) \, dA(y) \]
Reparameterizing the direct illumination integral

**Hemispherical** integral

\[ I = \int_{\mathbb{H}_2} f(\omega_i) \, d\sigma(\omega_i) \]

classical domain

**Surface** integral

\[ I = \int_{\mathcal{L}(\pi)} f(y \rightarrow x) \, G(x, y) \, dA(y) \]

evolving domain

\[ I = \int_{\mathcal{L}(\pi)} f(y \rightarrow x) \, G(x, y) \, dA(y) \]
Differentiating the hemispherical integral

\[ I = \int_{\mathbb{H}^2} f(\omega_l) d\sigma(\omega_l) \]

\[ \frac{dI}{d\pi} = \int_{\mathbb{H}^2} \frac{df}{d\pi} d\sigma + \int_{\partial \mathbb{H}^2} g \, dl \]

\( \pi \): size of the emitter

Discontinuities of \( f \)

Reynolds transport theorem
Differentiating the area integral

\[ I = \int_{L(\pi)} f(y \to x) G(x, y) dA(y) \]

\[ \frac{dI}{d\pi} = \int_{L(\pi)} \frac{d(fG)}{d\pi} dA + \int_{\partial L(\pi)} g \, dl \]

\( \pi \): size of the emitter

Boundary of \( L(\pi) \)

Reynolds transport theorem
Sources of discontinuities

- Boundary edge
- Sharp edge
- Silhouette edge
Sources of discontinuities

Boundary edge

Sharp edge

Silhouette edge
Sources of discontinuities

- Boundary edge
- Sharp edge
- Silhouette edge
Sources of discontinuities

- Boundary edge
- Sharp edge
- Silhouette edge
Sources of discontinuities

- Boundary edge
- Sharp edge
- Silhouette edge

Topology-driven

Visibility-driven
Sources of discontinuities

- Boundary edge
- Sharp edge
- Silhouette edge

Topology-driven

Visibility-driven
Sources of discontinuities

- We still need to account for discontinuities when using smooth closed surfaces (e.g., neural SDFs)
Sources of discontinuities

- We still need to account for discontinuities when using smooth closed surfaces (e.g., neural SDFs).
Sources of discontinuities

- We still need to account for discontinuities when using smooth closed surfaces (e.g., neural SDFs)

[Gargallo et al., ICCV 2007]
Significance of the boundary integral

Original image

Derivative image w.r.t. vertical offset of the area light and the cube

Derivative image w/o boundary integral
Significance of the boundary integral

Original image

Derivative image w.r.t. vertical offset of the area light and the cube

Derivative image w/o boundary integral
Gradient Accuracy Matters

Inverse-rendering results with *identical* optimization settings

**INIT. MESH**: 
0.0115

**SOFTRAS**: 
0.0039

**PYTORCH3D**: 
0.0091

**MITSUBA 2**: 
0.0065

**NVDIFFRAST**: 
0.0022

**Luan et al. 2021**: 
0.0016

**GROUND TRUTH**: 
head

**Relative err**: 0%

**0.0010**: 
maneki
Differential Global Illumination
Very active area of research

Path-Space Differentiable Rendering
CHENG ZHANG, University of California, Irvine
BAILEY MILLER, Carnegie Mellon University
KALYAN, University of California, Irvine

Mitsuba 2: A Retargetable Forward and Inverse Renderer
MERLIN NIMIER-DAVID, Ecole Polytechnique Federale de Lausanne
DELIO VICINI, Ecole Polytechnique Federale de Lausanne
TIZIAN ZELTNER, Ecole Polytechnique Federale de Lausanne
WENZEL JAKOB, Ecole Polytechnique Federale de Lausanne

Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time
DELIO VICINI, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland
SEBASTIEN SPEIERER, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland
WENZEL JAKOB, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland

Reparameterizing Discontinuous Integrands for Differentiable Rendering
GUILLAUME LOUBET, Ecole Polytechnique Federale de Lausanne (EPFL)
NICOLAS HOLZSCHUCH, Inria, Univ. Grenoble-Alpes, CNRS, UK
WENZEL JAKOB, Ecole Polytechnique Federale de Lausanne (EPFL)

Differentiable Monte Carlo Ray Tracing through Edge Sampling
TZU-MAO LI, MIT CSAIL
MIKKI AITTALA, MIT CSAIL
FREDO DURAND, MIT CSAIL
JAAKKO LEHTINEN, Aalto University & NVIDIA

(a) Initial state (b) Target state (c) Ours (d) RB (bias)
Remember: Path Integral for Global Illumination

\[ I = \int_{\Omega} f(\bar{x}) \, d\mu(\bar{x}) \]

Pixel value
Measurement contribution
Path space
Area-product measure

Light path \( \bar{x} = (x_0, x_1, x_2, x_3) \)

[Zhang et al., 2020]
Differential Path Integral

Path-space differentiable rendering

\[ \frac{d}{d\theta} \left( \int_{\Omega} f(\boldsymbol{x}) d\mu(\boldsymbol{x}) \right) = \int_{\Omega} f(\boldsymbol{x}) d\mu(\boldsymbol{x}) + \int_{\partial\Omega} g(\boldsymbol{x}) d\mu'(\boldsymbol{x}) \]

Interior integral  Boundary integral

[Zhang et al., 2020]
Differential Path Integral

Path-space differentiable rendering

\[
\frac{d}{d\theta} \left( \int_{\Omega} f(\mathbf{x}) d\mu(\mathbf{x}) \right) = \int_{\Omega} f(\mathbf{x}) d\mu(\mathbf{x}) + \int_{\partial\Omega} g(\mathbf{x}) d\mu'(\mathbf{x})
\]

Interior integral  Boundary integral

We now derive \(\delta\nu/\delta v\) in Eq. (25) using the recursive relations provided by Eqs. (21) and (24). Let

\[
\begin{align*}
\hat{h}_n^{(0)}(x_0; x_N) &= \int_{M^{N-1}} \left( h_n^{(0)} \right)_{x_0} dA(x_0) + \sum_{n' = 1}^{N} \int \partial \hat{h}_n^{(0)}(x_0; x_{n'-1}) d\mu'(x_{n'-1}), \\
\hat{h}_n^{(0)}(x_0; x_{n'-1}) &= \sum_{n = 1}^{N} h(x_{n'-1}) V(x_{n'-1}) dA(x_{n'-1}), \\
\Delta \hat{h}_n^{(0)}(x_0; x_{n'-1}) &= \hat{h}_n^{(0)}(x_0; x_{n'-1}) - \hat{h}_n^{(0)}(x_0; x_{n'-1})/g(x_{n'-2}; x_{n'-1}),
\end{align*}
\]

for \(0 \leq n < n' \leq N\). We omit the dependencies of \(h_n^{(0)}, h_n^{(1)}\), and \(\Delta h_n^{(0)}\) on \(x_0, \ldots, x_N\) for notational convenience. We now show that, for all \(0 \leq n < N\), it holds that

\[
\hat{h}_n(x_0; x_{n-1}) = \int_{M^{N-n}} \left( h_n^{(0)} \right)_{x_0} dA(x_0) + \sum_{n' = 1}^{N} \int \partial h_n^{(0)}(x_0; x_{n'-1}) d\mu'(x_{n'-1}),
\]

and

\[
\begin{align*}
\hat{h}_{n-1}(x_0; x_{n-2}) &= \int_{M^{N-n+1}} \left( h_{n-1}^{(0)} \right)_{x_0} dA(x_0) + \sum_{n' = 1}^{N} \int \partial \hat{h}_{n-1}^{(0)}(x_0; x_{n'-2}) d\mu'(x_{n'-2}), \\
\Delta \hat{h}_{n-1}^{(0)}(x_0; x_{n'-2}) &= \hat{h}_{n-1}^{(0)}(x_0; x_{n'-2}) - \hat{h}_{n-1}^{(0)}(x_0; x_{n'-2})/g(x_{n'-3}; x_{n'-2}).
\end{align*}
\]

Notice that \(\hat{h}_0^{(0)} = f\) and \(\Delta \hat{h}_0^{(0)} = \Delta f\), where \(\Delta f\) follows the definition in Eq. (28). Letting \(n = 0\) in Eq. (56) yields

\[
\hat{h}_0(x_0) = \int_{M^N} f(x_0) + \sum_{n' = 1}^{N} \int \partial f(x_0; x_{n'-1}) d\mu'(x_{n'-1}),
\]

Lastly, based on the assumption that \(h_0\) is continuous in \(x_0\), Eq. (25) can be obtained by differentiating Eq. (23):

\[
\frac{\partial}{\partial v} = \frac{\partial}{\partial v} \int_{M^N} h_0(x_0) dA(x_0).
\]

(The full derivation is quite involved...)
Differential Path Integral

Path-space differentiable rendering

\[
\frac{d}{d\theta} \left( \int_{\Omega} f(\mathbf{x}) d\mu(\mathbf{x}) \right) = \int_{\Omega} f(\mathbf{x}) d\mu(\mathbf{x}) + \int_{\partial\Omega} g(\mathbf{x}) d\mu'(\mathbf{x})
\]

**Interior integral**

- Defined on the ordinary path space \( \Omega \)
- The integrand \( f \) can be obtained by differentiating the ordinary measurement contribution function \( f \)

[Zhang et al., 2020]
Differential Path Integral

Path-space differentiable rendering

\[
\frac{d}{d\theta} \left( \int_{\Omega} f(\mathbf{x}) d\mu(\mathbf{x}) \right) = \int_{\Omega} f(\mathbf{x}) d\mu(\mathbf{x}) + \int_{\partial\Omega} g(\mathbf{x}) d\mu'(\mathbf{x})
\]

**Boundary integral**

- Defined on the boundary path space \( \partial\Omega \)
- A **boundary light path** is the same as an original one except having exactly one **boundary segment**
Path-Space Differentiable Path Tracing

*Unidirectional* estimator

- **Interior**: *unidirectional* path tracing
- **Boundary**: *unidirectional* sampling of subpaths

[Zhang et al., 2020]
Path-Space Differentiable Path Tracing

**Unidirectional** estimator

- **Interior**: unidirectional path tracing
- **Boundary**: unidirectional sampling of subpaths

**Bidirectional** estimator

- **Interior**: bidirectional path tracing
- **Boundary**: bidirectional sampling of subpaths

[Zhang et al., 2020]
Application: neural rendering

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis [MildenHall et al. ECCV 2020]
Acquisition of scattering materials

mustard
whole milk
hand cream
shampoo
olive oil
robittusin
curacau
mixed soap

wine
milk soap
liquid clay
reduced milk

mustard
coffee

[Gkioulekas et al., 2013]
Acquisition setup

[Gkioulekas et al., 2013]
Acquisition setup

[Gkioulekas et al., 2013]
Acquisition setup

Invert using differentiable rendering

[Gkioulekas et al., 2013]
Synthetic renderings

mixed soap

glycerine soap

olive oil

curacao

whole milk

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions

unknown nanodispersion

\[ \text{dispersing medium} \]
\[ \text{particle material} \]

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions

unknown nanodispersion

dispersing medium

particle material

measurements

size

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions
unknown nanodispersion

 dispersing medium
 particle material

measurements

size

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions

very precise dispersions (NIST Traceable Standards)

polystyrene  aluminum oxide

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions

very precise dispersions (NIST Traceable Standards)

polystyrene 1
polystyrene 2
polystyrene 3
aluminum oxide

[Ground-truth]

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions

very precise dispersions (NIST Traceable Standards)

polystyrene 1
polystyrene 2
polystyrene 3
aluminum oxide

$\theta$

$p(\theta)$

$\theta$

$\theta$

ground-truth

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions

very precise dispersions (NIST Traceable Standards)

polystyrene 1  polystyrene 2  polystyrene 3  aluminum oxide

[Ground-truth]

polystyrene 1  polystyrene 2  polystyrene 3  aluminum oxide

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions

very precise dispersions (NIST Traceable Standards)

polystyrene 1
polystyrene 2
polystyrene 3
aluminum oxide

[Gkioulekas et al., 2013]
Particle sizing of industrial nanodispersions

polystyrene 1  polystyrene 2  polystyrene 3  aluminum oxide

very precise dispersions (NIST Traceable Standards)

[Gkioulekas et al., 2013]
Optical tomography

- Camera
- Thick smoke cloud
- Simulated camera measurements
- Reconstructed cloud volume
- Slice through the cloud

[Gkioulekas et al., 2015]
Computational Fabrication

Determining the material configuration for individual voxels in full-color inkjet 3D printing

[Nindel et al. 2021]
Active area of research

- Woven fabrics
  - Khungurn et al. 2015, Zhao et al. 2016

- Efficient algorithms
  - Nimier-David et al. 2019, 2020

- Industrial dispersions
  - Gkioulekas et al. 2013

- Computed tomography
  - Geva et al. 2018

- 3D printing
  - Elek et al. 2019, Nindel et al. 2021

- Cloud tomography
  - Levis et al. 2015, 2017, 2020
Non-line-of-sight (NLOS) imaging
Non-line-of-sight (NLOS) imaging
Non-line-of-sight (NLOS) imaging
Non-line-of-sight (NLOS) imaging

Time-of-flight measurements
Non-line-of-sight (NLOS) imaging

Time-of-flight measurements

visible wall

scan point

source & sensor

occluder

NLOS object

intensity (# photons)

time

x

y

Time-of-flight measurements

LOS signal

NLOS signal
SPAD-based lidar

- Single-photon avalanche photodiode (SPAD)
- Picosecond laser
- Galvo mirror
NLOS shape optimization

Simulated time-of-flight data

100,000 vertices

[Tsai et al., 2019]
Underwater 3D using imaging SONAR

Underwater robot with sonar

Robotics Institute High Bay

[Qadri et al., 2023]
Underwater 3D using imaging SONAR

Test structure

Ground truth mesh obtained using a laser scan

Sonar image collection points

[Qadri et al., 2023]
Underwater 3D using imaging SONAR

1 degree

back-projection

virtual aperture

differentiable rendering

14 degrees

28 degrees

[Qadri et al., 2023]
Millimeter-accuracy underwater 3D reconstructions using data captured with an acoustic sonar mounted on robot.

[Qadri et al., 2023]
Kaleidoscopic 3D scanning

[Ahn et al., CVPR 2023, Tuesday PM]
Kaleidoscopic 3D scanning

[Ahn et al., CVPR 2023, Tuesday PM]
Kaleidoscopic 3D scanning

[Ahn et al., CVPR 2023, Tuesday PM]
Kaleidoscopic 3D scanning

kaleidoscopic system

camera view

virtual cameras

[Ahn et al., CVPR 2023, Tuesday PM]
Kaleidoscopic 3D scanning

Kaleidoscope

camera view

virtual cameras

[Ahn et al., CVPR 2023, Tuesday PM]
Example 3D scans

photograph

3D reconstruction

[Ahn et al., CVPR 2023, Tuesday PM]
Optimizing Gradient-Index (GRIN) Optics

Luneburg Lens  

GRIN Fiber

[Teh et al., 2022]
Nonlinear Ray Tracing

\( \eta(\mathbf{x}) \): refractive index of the volume at location, \( \mathbf{x} \)

[Teh et al., 2022]
Nonlinear Ray Tracing

[Teh et al., 2022]
Nonlinear Ray Tracing

$x_0$

[Teh et al., 2022]
Nonlinear Ray Tracing

[Teh et al., 2022]
Nonlinear Ray Tracing

\[ \frac{dx}{dt} = \mathbf{v} \quad \frac{d\mathbf{v}}{dt} = \eta \nabla \eta \]

[Teh et al., 2022]
Nonlinear Ray Tracing

\[ \frac{dx}{dt} = v \quad \frac{dv}{dt} = \eta \nabla \eta \]

[Teh et al., 2022]
Nonlinear Ray Tracing

\[ x_i = x_{i-1} + v_i \Delta t \quad v_i = v_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t \]

[Teh et al., 2022]
Nonlinear Ray Tracing

\[ \mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t \quad \mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t \]

[Teh et al., 2022]
Nonlinear Ray Tracing

\[ \mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t \]

\[ \mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t \]

[Teh et al., 2022]
Nonlinear Ray Tracing

\[ x_i = x_{i-1} + v_i \Delta t \quad v_i = v_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t \]

[Teh et al., 2022]
Nonlinear Ray Tracing

\[ \mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t \]
\[ \mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t \]

[Teh et al., 2022]
Nonlinear Ray Tracing

$x_i = x_{i-1} + v_i \Delta t$

$v_i = v_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t$

[Teh et al., 2022]
Nonlinear Ray Tracing

\[ \mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t \quad \mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t \]

[Teh et al., 2022]
Nonlinear Ray Tracing

[Teh et al., 2022]
Nonlinear Ray Tracing

[Teh et al., 2022]
Nonlinear Ray Tracing

\[ \min_{\eta} \| \hat{\mathbf{x}} - \mathbf{x}_f \|^2 \]

[Teh et al., 2022]
Nonlinear Ray Tracing

\[
\frac{d}{d\eta} \| \hat{x} - x_f \|^2
\]

[Teh et al., 2022]
Nonlinear Ray Tracing in reverse

[Teh et al., 2022]
Nonlinear Ray Tracing in reverse

[Teh et al., 2022]
Nonlinear Ray Tracing in reverse

[Teh et al., 2022]
Nonlinear Ray Tracing in reverse

\[
\min_\eta \sum_{i=1}^N \mathcal{F}_i \left[ \int_{(x_0, v_0) \in \Omega} C_i \left( x \left( \sigma_f; \eta, x_0, v_0 \right), v \left( \sigma_f; \eta, x_0, v_0 \right) \right) \, dx_0 \, dv_0 \right]
\]

s.t. \( x \left( \sigma; \eta, x_0, v_0 \right) = v \), \( \forall \sigma \in [0, \sigma_f] \),

\( v \left( \sigma; \eta, x_0, v_0 \right) = \eta \nabla \eta \), \( \forall \sigma \in [0, \sigma_f] \),

\( x \left( 0; \eta, x_0, v_0 \right) = x_0 \),

\( v \left( 0; \eta, x_0, v_0 \right) = v_0 \),

\( \hat{x} \)

(15)

\[
\hat{\lambda} = - \left( \nabla \eta \left( \nabla \eta \right)^T + \eta \text{Hess} \left( \eta \right) \right) \mu, \quad \forall \sigma \in [0, \sigma_f] \quad (19)
\]

\[
\hat{\mu} = - \lambda, \quad \forall \sigma \in [0, \sigma_f] \quad (20)
\]

\[
\lambda \left( \sigma_f \right) = \frac{\partial C}{\partial x}, \quad (21)
\]

\[
\mu \left( \sigma_f \right) = \frac{\partial C}{\partial v}, \quad (22)
\]

\[
x_{i-1} = x_i - v_i \Delta \sigma, \quad (28)
\]

\[
v_{i-1} = v_i - \eta \left( x_{i-1} \right) \nabla \eta \left( x_{i-1} \right) \Delta \sigma, \quad (29)
\]

\[
\lambda_{i-1} = \lambda_i + \left( \nabla \eta \left( x_{i-1} \right) \left( \nabla \eta \left( x_{i-1} \right) \right)^T + \eta \left( x_{i-1} \right) \text{Hess} \left( \eta \left( x_{i-1} \right) \right) \right) \mu_i \Delta \sigma, \quad (30)
\]

\[
\mu_{i-1} = \mu_i + \lambda_{i-1} \Delta \sigma. \quad (31)
\]
Optimizing Gradient-Index (GRIN) Optics

Luneburg Lens

GRIN Fiber

[Teh et al., 2022]
Optimizing Gradient-Index (GRIN) Optics

Luneburg Lens

GRIN Fiber

[Teh et al., 2022]
Luneburg Lens
Luneburg Lens
Luneburg Lens
Luneburg Lens
Luneburg Lens
Luneburg Lens

\[ \eta(x) = \sqrt{2 - \|x\|^2} \]

[Luneburg, R. K. 1944]
Luneburg Lens

[Teh et al., 2022]
Luneburg Lens

[Teh et al., 2022]
Luneburg Lens

\[
\min_{\eta} \sum_{i \in \text{rays}} \| x_{\text{target}} - x_i \|^2
\]

[Teh et al., 2022]
Luneburg Lens

\[
\min_{\eta} \sum_{i \in \text{rays}} \| x_{\text{target}} - x_i \|^2
\]

[Teh et al., 2022]
Luneburg Lens

[Teh et al., 2022]
Luneburg Lens

[Teh et al., 2022]
Luneburg Lens

[Teh et al., 2022]
Luneburg Lens

Refractive Index

Position

[Optimized]

[Teh et al., 2022]
Luneburg Lens

[Teh et al., 2022]
GRIN Fiber

https://en.wikipedia.org/wiki/Optical_fiber
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

Parabolic

Target

Modal dispersion

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
GRIN Fiber

[Teh et al., 2022]
Multiview Display

[Teh et al., 2022]
Multiview Display

[Teh et al., 2022]
Differentiable rendering for wildfire monitoring

**Ours**: Low-flying, granular resolution; team scouts fire plume and environment up-close

**Other’s product**: 2D fire map, low-res

**Our Product**: granular 3D map of environment and fire plume

Sensor wavelength trades smoke penetration and resolution

[USDA NIFA project jointly with Sebastian Scherer and Katia Sycara]
What differentiable rendering does not give us
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \frac{\text{scene}}{\text{unknowns } \pi} \right) \right]$$
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

\[
\min_{\text{scene \ unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene \ unknowns } \pi \right) \right]
\]
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

\[
\min_{\text{scene, unknowns } \pi} \text{loss} \left[ \text{render} \left( \begin{array}{c}
\text{scene} \\
\text{unknowns } \pi
\end{array} \right) \right]
\]

Stochastic gradient descent (e.g., Adam):

initialize \( \pi \leftarrow \pi_0 \)

while (not converged)

update \( \pi \leftarrow \pi + \eta \cdot \frac{\text{dloss}(\pi)}{\text{d}\pi} \)
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene unknowns } \pi \right) \right]$$

Stochastic gradient descent (e.g., Adam):

initialize $$\pi \leftarrow \pi_0$$

while (not converged)

update $$\pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$$

Differentiable rendering
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

\[
\min_{\text{scene unknowns } \pi} \text{loss} \left[ \begin{array}{c}
\text{scene, render} \\
\text{unknowns } \pi
\end{array} \right]
\]

Stochastic gradient descent (e.g., Adam):

initialize \( \pi \leftarrow \pi_0 \)

while (not converged)

update \( \pi \leftarrow \pi + \eta \cdot \frac{\text{dloss}(\pi)}{d\pi} \)
Why we need good initializations

• Analysis-by-synthesis objectives are highly non-convex, non-linear
  • Multiple *local* minima
Why we need good initializations

• Analysis-by-synthesis objectives are highly non-convex, non-linear
  • Multiple *local* minima

• Ambiguities exist between different parameters
  • Multiple *global* minima
Why we need good initializations

• Analysis-by-synthesis objectives are highly non-convex, non-linear
  • Multiple local minima

• Ambiguities exist between different parameters
  • Multiple global minima

Ambiguities between BRDF and lighting
[Romeiro and Zickler 2010]

Ambiguities between shape and lighting
[Xiong et al. 2015]

Ambiguities between scattering parameters [Zhao et al. 2014]
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene unknowns } \pi \right) \right]$$

Stochastic gradient descent (e.g., Adam):

**Initialize** $\pi \leftarrow \pi_0$

while (not converged)

**Update** $\pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$

Differentiable rendering
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

\[
\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render (scene unknowns } \pi) \right]
\]

Stochastic gradient descent (e.g., Adam):

- initialize \( \pi \leftarrow \pi_0 \)
- while (not converged)
  - update \( \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi} \)

Differentiable rendering
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

\[
\min_{\text{scene, unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene, unknowns } \pi \right) \right]
\]

Learned initializations help:
- avoid local minima
- accelerate convergence

Stochastic gradient descent (e.g., Adam):

- initialize \( \pi \leftarrow \pi_0 \)
- while (not converged)
  - update \( \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi} \)
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

\[
\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene unknowns } \pi \right) \right]
\]

Learned initializations help:
- avoid local minima
- accelerate convergence

Stochastic gradient descent (e.g., Adam):

- initialize \( \pi \leftarrow \pi_0 \)
- while (not converged)
  - update \( \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi} \)

Differentiable rendering
Why we need discriminative loss functions

- Well-designed loss functions can help reduce ambiguities
- Perceptual losses can help emphasize design aspects that matter
- Differentiable rendering can be combined with any loss function that can be backpropagated through

VGG-based perceptual loss [Johnson et al. 2016]
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

$$\min_{\text{scene}} \text{loss} \left[ \text{render} \left( \text{scene}, \text{unknowns } \pi \right) \right]$$

Stochastic gradient descent (e.g., Adam):

- **initialize** \( \pi \leftarrow \pi_0 \)
- **while** (not converged)
  - **update** \( \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi} \)

\( \pi \): BRDF
\( \pi \): illumination
\( \pi \): scattering
\( \pi \): camera pose
\( \pi \): 3D shape and pose
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

\[
\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene unknowns } \pi \right) \right]
\]

Stochastic gradient descent (e.g., Adam):

- initialize \( \pi \leftarrow \pi_0 \)
- while (not converged)
  - update \( \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi} \)
The extent to which we can improve upon an initialization strongly depends on the signal-to-noise ratio of our measurements.
High signal-to-noise ratio is critical

- The extent to which we can improve upon an initialization strongly depends on the signal-to-noise ratio of our measurements.

Non-line-of-sight imaging [Tsai et al. 2019]
The extent to which we can improve upon an initialization strongly depends on the signal-to-noise ratio of our measurements.

We need reliable camera models (noise, aberrations, other non-idealities).

Non-line-of-sight imaging [Tsai et al. 2019]
High signal-to-noise ratio is critical

- The extent to which we can improve upon an initialization strongly depends on the signal-to-noise ratio of our measurements
- We need reliable camera models (noise, aberrations, other non-idealities)

Non-line-of-sight imaging [Tsai et al. 2019]

Optical gradient descent [Chen et al. 2020]
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
- tactile sensor design
- differentiable rendering
- inverse problems
Take-Home Messages

• Great progress has been made in physics-based rendering
  • Capable of handling multiple types of imaging systems beyond RGB cameras (e.g., time-of-flight, sonar, tactile sensors).

transient rendering of static scene
time-gated rendering of dynamic scene
Take-Home Messages

• Great progress has been made in physics-based rendering:
  • Capable of handling multiple types of imaging systems beyond RGB cameras (e.g., time-of-flight, sonar, tactile sensors).
  • Capable of handling more general scene models and light-matter interactions (e.g., speckle, continuous refraction and scattering).
Take-Home Messages

• Great progress has been made in physics-based rendering
  • Capable of handling multiple types of imaging systems beyond RGB cameras (e.g., time-of-flight, sonar, tactile sensors).
  • Capable of handling more general scene models and light-matter interactions (e.g., speckle, continuous refraction and scattering).
  • Capable of acting as digital twins for scientific imaging applications.

non-line-of-sight imaging  light throughput enhancement  ultrafast light scanners  imaging through scattering media  tactile sensor design
Take-Home Messages

• Great progress has been made in physics-based rendering
  • Capable of handling multiple types of imaging systems beyond RGB cameras (e.g., time-of-flight, sonar, tactile sensors).
  • Capable of handling more general scene models and light-matter interactions (e.g., speckle, continuous refraction and scattering).
• Capable of acting as digital twins for scientific imaging applications.
• Capable of differentiation for general inverse rendering problems.
Monte Carlo rendering for more general physics and sensing

Simulation of general diffusion processes like heat transfer and oxygen flow

Joint work with Rohan Sawhney, Bailey Miller, Keenan Crane
SIGGRAPH 2023
Many thanks to our collaborators
Many thanks to our sponsors

See Below the Skin
Physics-based rendering and its applications to computational imaging

**forward rendering**
- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

**inverse rendering**
- accurate and efficient differentiable simulation
- tractably solve general inverse problems

- time-of-flight imaging
- non-line-of-sight imaging
- acousto-optic lensing
- ultrafast light scanning
- speckle imaging
- tactile sensor design
- differentiable rendering