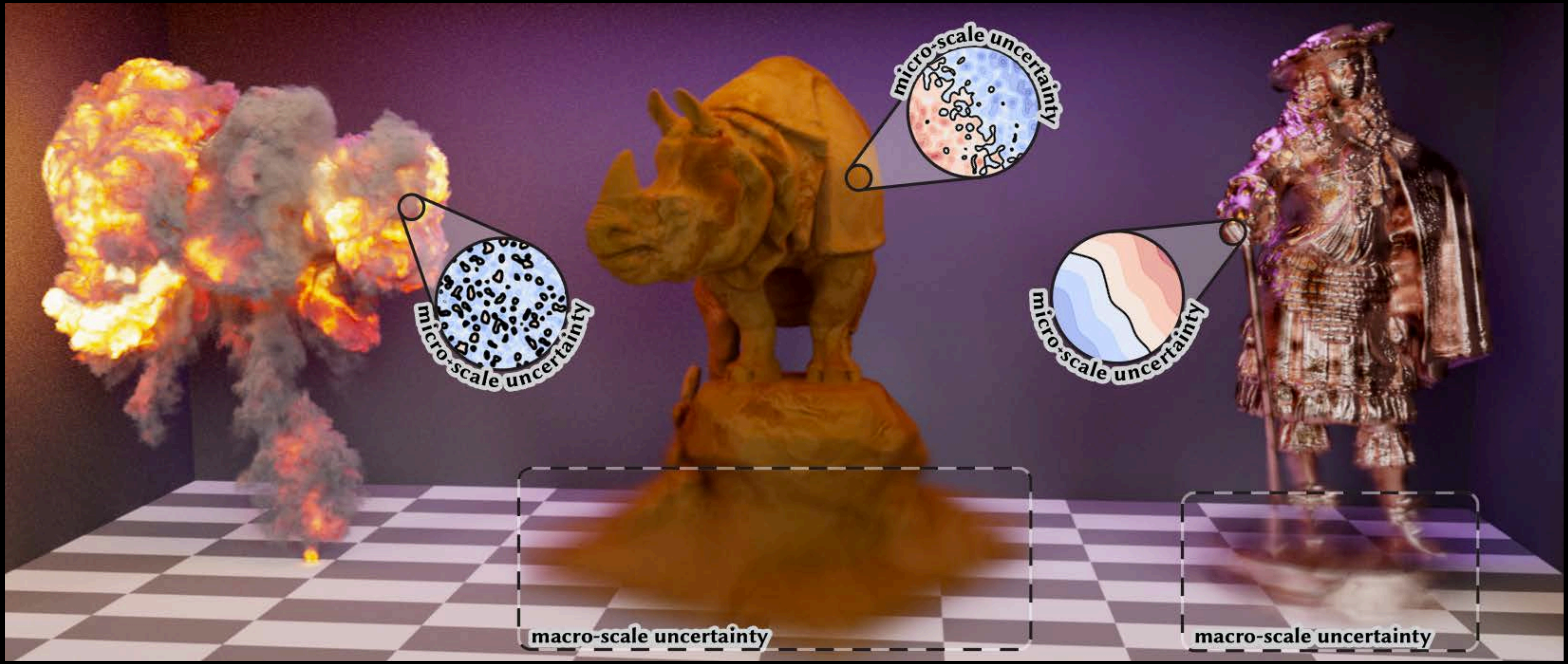
From Microfacets to Participating Media: A Unified Theory of Light Transport with Stochastic Geometry

Dario Seyb, Eugene D’eon, Benedikt Bitterli, Wojciech Jarosz

ACM ToG 2024 (Best Paper Award)



Seungwoo Yoo, KAIST Visual AI Group

Prologue

Prologue: Light Transport Theory in CG

Simulating the behavior of light in a 3D space is crucial in photorealistic rendering, and numerous techniques have been developed over the decades.

Physically Based Rendering: From Theory To Implementation (4th edition)
Matt Pharr, Wenzel Jakob, and Greg Humphreys

Cotton Candy
Chenlin Meng, Hubert Teo, and Jiren Zhu

Prologue: Light Transport Theory in CG

Existing rendering algorithms solve the rendering equation [Kajiya, 1986], under various conditions including geometry, lighting, and material properties.

$$L_o\left(\mathbf{x}, \omega_o, \lambda, t\right) = L_e\left(\mathbf{x}, \omega_o, \lambda, t\right) + L_r\left(\mathbf{x}, \omega_o, \lambda, t\right)$$

"The radiance from a point is the sum of the radiance emitted and reflected at the point."

Prologue: Light Transport Theory in CG

Existing rendering algorithms solve the rendering equation [Kajiya, 1986], under various conditions including geometry, lighting, and material properties.

$$L_o\left(\mathbf{x}, \omega_o, \lambda, t\right) = L_e\left(\mathbf{x}, \omega_o, \lambda, t\right) + \underline{L_r\left(\mathbf{x}, \omega_o, \lambda, t\right)}$$

$$L_r\left(\mathbf{x}, \omega_o, \lambda, t\right) = \int_\Omega f_r\left(\mathbf{x}, \omega_i, \omega_o, \lambda, t\right) L_i\left(\mathbf{x}, \omega_i, \lambda, t\right) \left(\omega_i \cdot \mathbf{n}\right) d\omega_i$$

"The reflected radiance is the sum of all incoming radiance, each weighted by a BRDF."

Prologue: Light Transport Theory in CG

Existing rendering algorithms solve the rendering equation [Kajiya, 1986], under various conditions including geometry, lighting, and material properties.

$$L_o\left(\mathbf{x}, \omega_o, \lambda, t\right) = L_e\left(\mathbf{x}, \omega_o, \lambda, t\right) + L_r\left(\mathbf{x}, \omega_o, \lambda, t\right)$$

$$L_r\left(\mathbf{x}, \omega_o, \lambda, t\right) = \int_\Omega f_r\left(\mathbf{x}, \omega_i, \omega_o, \lambda, t\right) L_i\left(\mathbf{x}, \omega_i, \lambda, t\right)\left(\omega_i \cdot \mathbf{n}\right) d\omega_i$$

The rendering equation is **recursive** by definition.

Prologue: Light Transport Theory in CG

Existing rendering algorithms solve the rendering equation [Kajiya, 1986], under various conditions including geometry, lighting, and material properties.

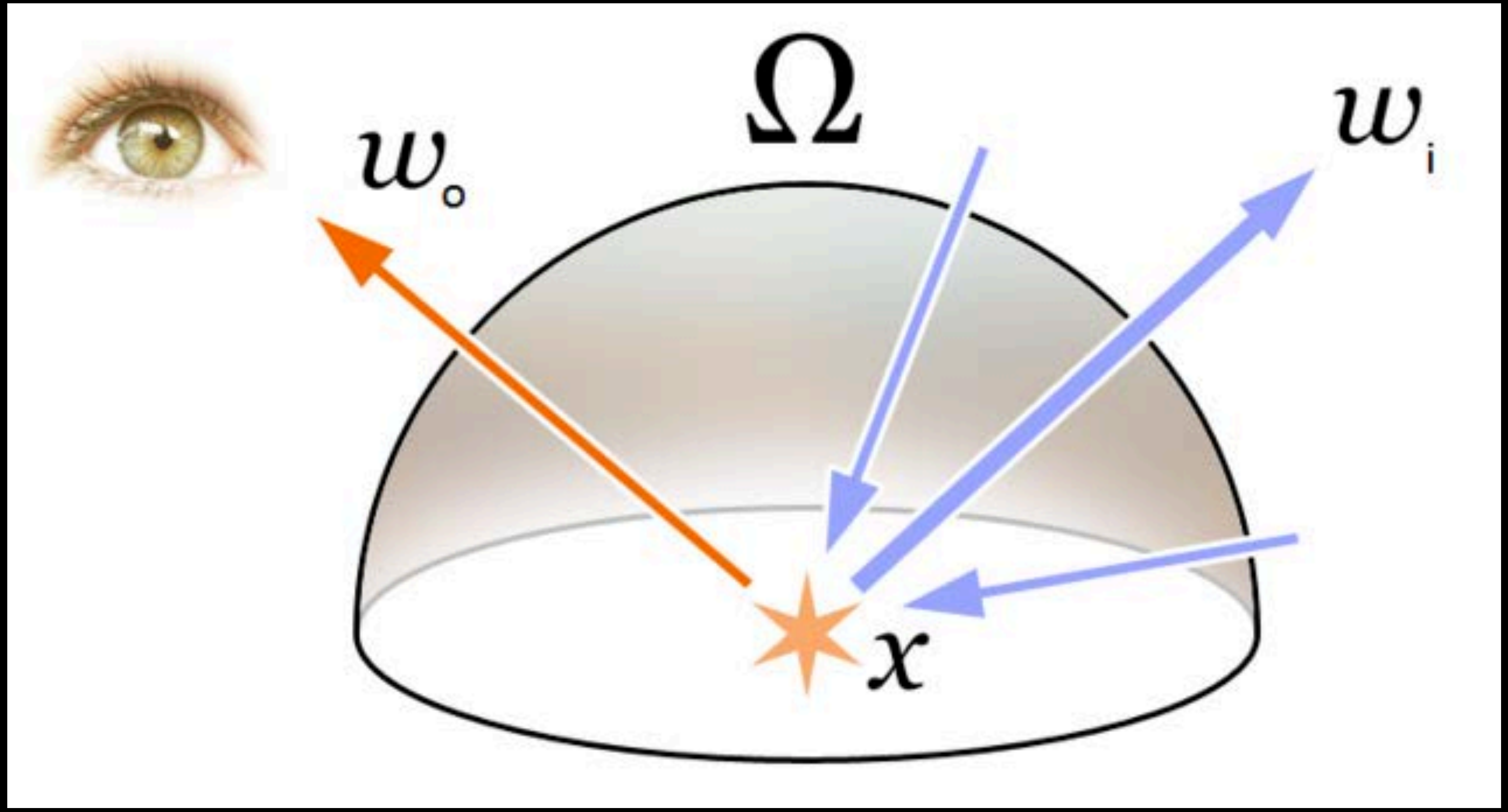
$$L_o\left(\mathbf{x}, \omega_o, \lambda, t\right) = L_e\left(\mathbf{x}, \omega_o, \lambda, t\right) + L_r\left(\mathbf{x}, \omega_o, \lambda, t\right)$$

$$L_r\left(\mathbf{x}, \omega_o, \lambda, t\right) = \boxed{\int_\Omega f_r\left(\mathbf{x}, \omega_i, \omega_o, \lambda, t\right) L_i\left(\mathbf{x}, \omega_i, \lambda, t\right) \left(\omega_i \cdot \mathbf{n}\right) d\omega_i}$$

The rendering equation is an **integral equation**.

Prologue: Light Transport Theory in CG

However, solving the rendering equation demands substantial computational resources due to the need for recursive integral evaluations.



Rendering Equation
Wikipedia

$$\int_{\Omega} f_r\left(\mathbf{x}, \omega_i, \omega_o, \lambda, t\right) L_i\left(\mathbf{x}, \omega_i, \lambda, t\right)\left(\omega_i \cdot \mathbf{n}\right) d\omega_i$$

For all ray direction ω_i's over a unit hemisphere Ω, evaluate the BRDF f_r and incoming radiance L_i.

Prologue: Light Transport Theory in CG

Complex, recursive integrals in the rendering equation is estimated using **the Monte Carlo method** involving random sampling.

$$\int_a^b f(x)dx$$

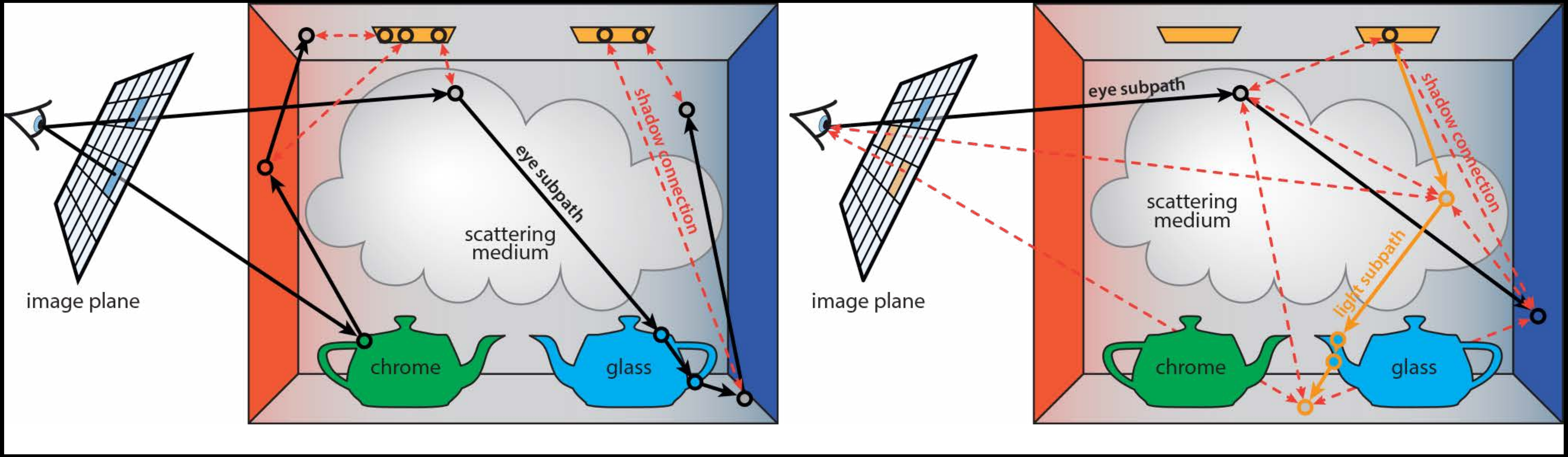
The MC estimator for the integral is:

$$F_n = \frac{b-a}{n} \sum_{i=1}^{n} f(X_i)$$

where $X_i \sim U(a, b)$. The estimator is *unbiased*. That is, $\mathbb{E}[F_n] = \int_a^b f(x)dx$ and its estimate converges to the true value of the integral in average.

Prologue: Light Transport Theory in CG

Ray tracers are programs that merely compute the MC estimate of the solutions of the rendering equation by recursively tracing rays starting from image pixels.



The path to path-traced movies, Foundations and Trends in Computer Graphics and Vision 2016
Per H. Christensen and Wojciech Jarosz

Prologue: Light Transport Theory in CG

Importance Sampling

Acceleration Structures

BRDF Acquisition

Control Variates

Subsurface Scattering

Denoising

Path Space

Inverse Rendering

… and more!

Rendering Distributions of Surfaces

Motivation

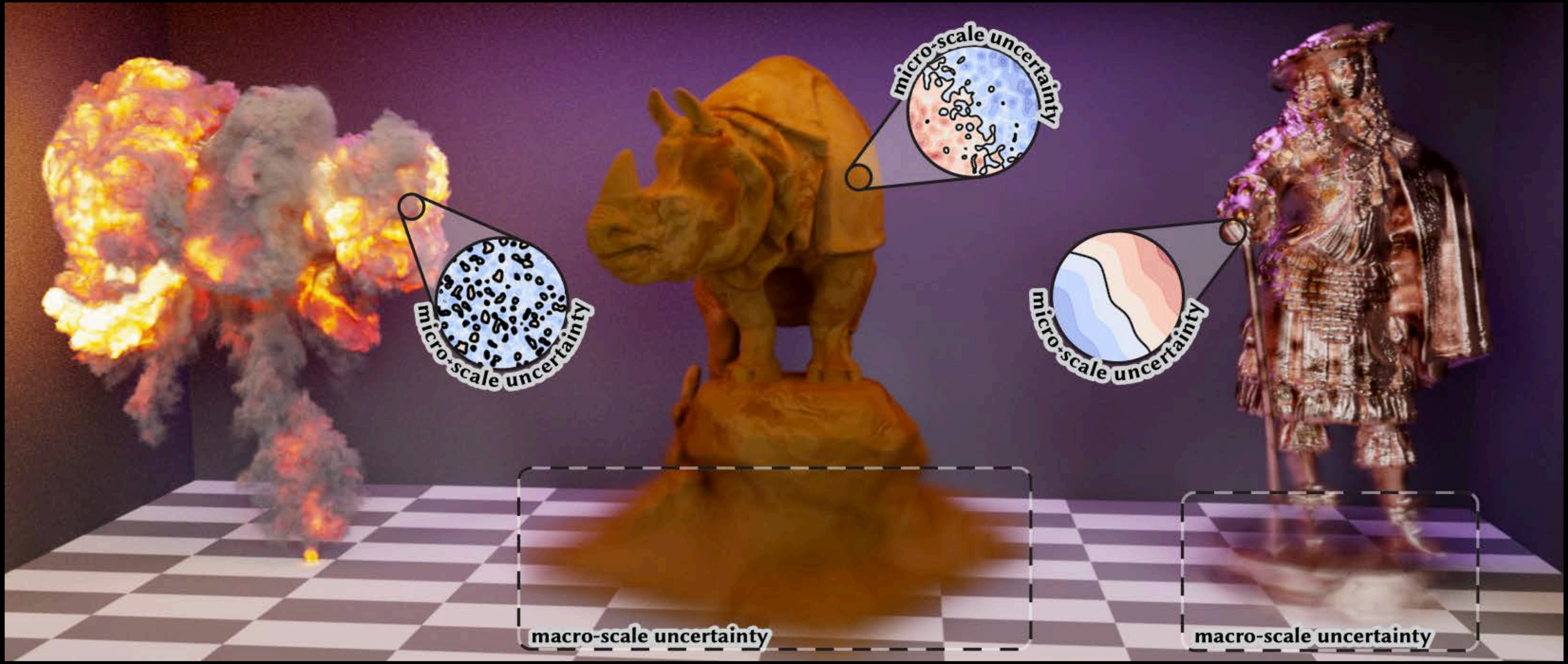
The current light transport theory handles hard surfaces and volumetric participating media differently, struggling to model categories that lie in-between.

"Reflections" – A Star Wars UE4 Real-Time Ray Tracing Cinematic Demo
Epic Games, ILMxLAB, and NVIDIA

A radiative transfer framework for non-exponential media, ACM ToG 2018
Benedikt Bitterli, Srinath Ravichandran, Thomas Müller, Magnus Wrenninge, Jan Novák, Steve Marschner, and Wojciech Jarosz

Motivation

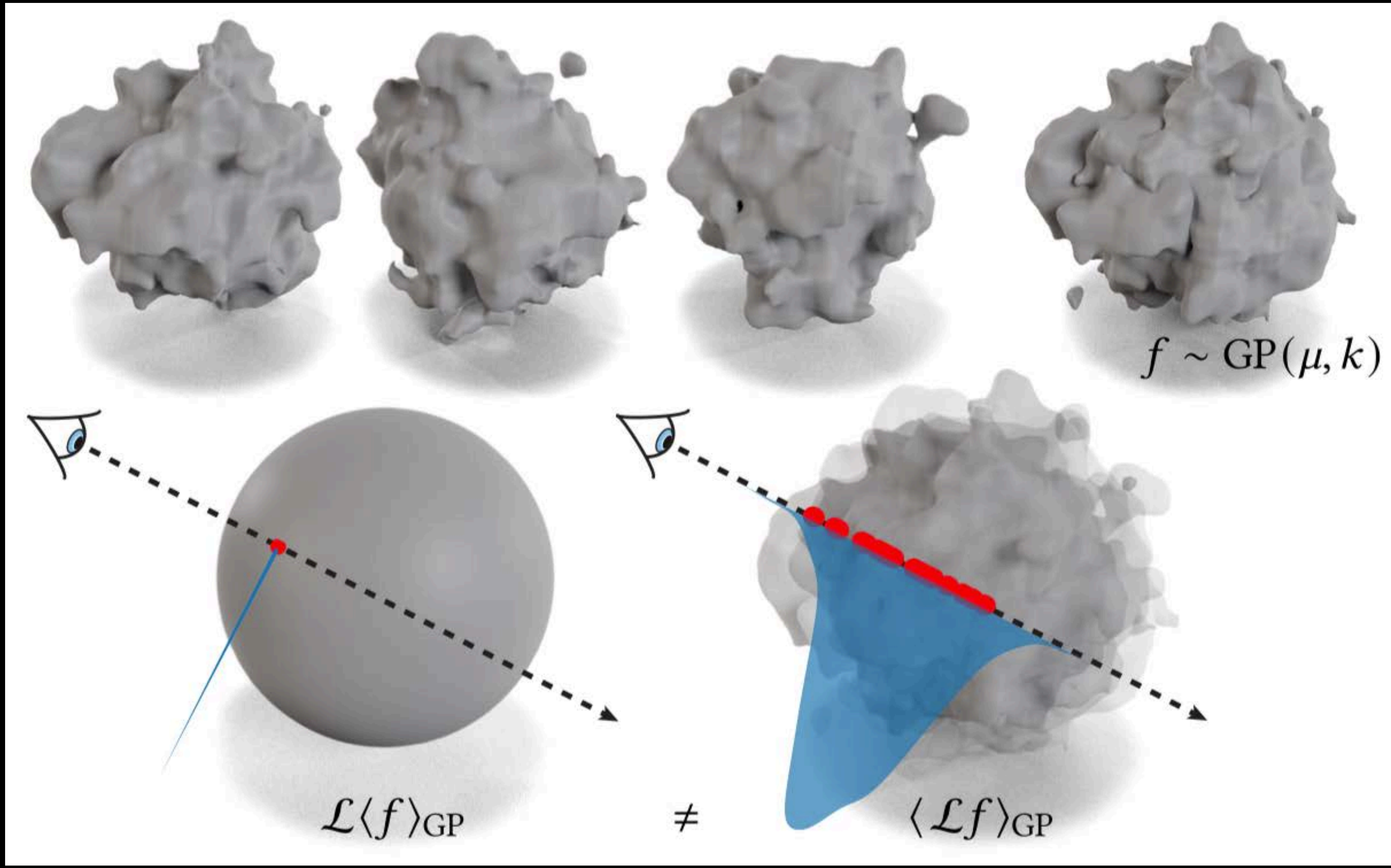
This paper introduces **a unified light transport theory** of surface and participating media, which extends to a wider range of geometry types.



Volumetric **In-Betweens** **Surface**

In a Nutshell…

This paper proposes to represent objects as **stochastic implicit surfaces**, specified by means and covariances of **Gaussian Processes (GPs)**.



Heavy Math Ahead!

There “will” be errors.

Feel free to interrupt me if you have questions.

Background & Notation

Gaussian Processes

A Gaussian process GP$(\mu, k)_\Omega$ is a distribution over functions $f : \Omega \to \mathbb{R}$ such that for any finite set of locations $\mathbf{x}_1, \ldots, \mathbf{x}_n = X \subseteq \Omega$, the evaluations of the function follow an n-dimensional Gaussian distribution

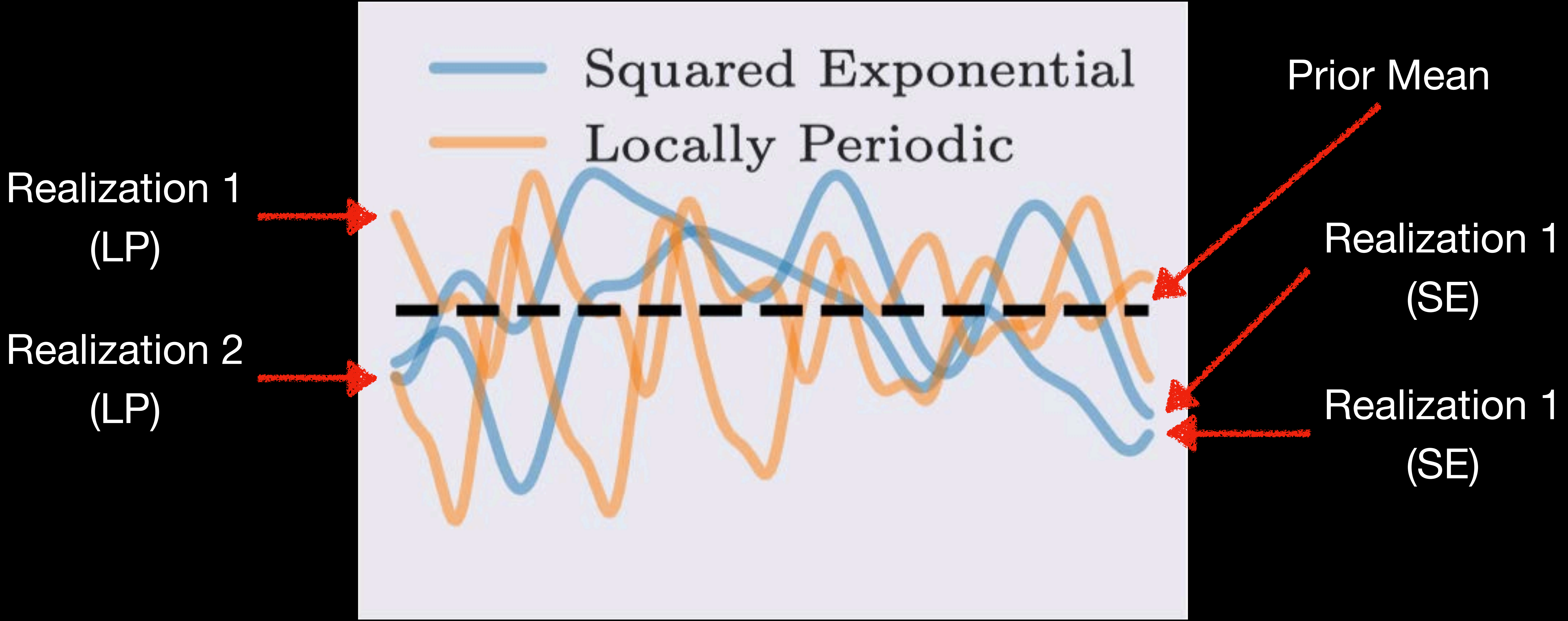
$$f_X \sim \mathcal{N}\left(\mu(X), k(X, X)\right)$$

where $\mu(X) = [\mu\left(\mathbf{x}_1\right), \ldots, \mu\left(\mathbf{x}_n\right)]^T$ is an n-dimensional mean vector and $k(X, X)$ is an $n \times n$ covariance matrix, with entries $k(X, X)^{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$.

The functions μ and k are denoted the *mean function* and *covariance kernel*, respectively.

Background & Notation

Gaussian Processes



Background & Notation

Covariance Kernel Functions

Kernel functions determine the shape of GPs and serve as a design variable for GPs. A kernel describes how similar values are at nearby points in space.

Positive Semi-Definite **Closed under Multiplication and Addition**

In this work, we are interested in particular types of kernels:

Stationary Kernels: $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y} - \mathbf{x})$

Isotropic Kernels: $k(\mathbf{x}, \mathbf{y}) = k(||\mathbf{y} - \mathbf{x}||)$

Non-Stationary Kernels: $k(\mathbf{x}, \mathbf{y}) \neq k(\mathbf{y} - \mathbf{x}) \rightarrow$ Tricky, but useful!

Background & Notation

Restricting Domains of Gaussian Processes

The input domain of a GP can be *restricted* without altering its statistics. We are interested in restricting the domain to points on a line $\mathbf{x}_{\mathbb{R}} = \left\{ \mathbf{x} + t\omega \,|\, t \in \mathbb{R} \right\}$ from the entire 3D space $\mathbb{R}^3$. In general,

$$f\left(\mathbf{x}_{\mathbb{R}}\right) = g(\mathbf{x}_{\mathbb{R}})$$

holds for $f \sim \mathrm{GP}(\mu, k)_{\mathbb{R}^3}$ and $g \sim \mathrm{GP}(\mu, k)_{\mathbf{x}_{\mathbb{R}}}$. This property allows us to evaluate GPs on low-dimensional *slices* without changing the mean or covariance kernel.

Background & Notation

Conditioned Processes

Given the location(s) of point(s) on the graphs of functions sampled from a GP, we can conditionally sample such functions that pass through the point(s).

$$f \sim \text{GP}\left(\mu_{|\zeta_m}(\mathbf{x}), k_{|\zeta_m}\left(\mathbf{x}, \mathbf{y}\right)\right), \text{ where}$$

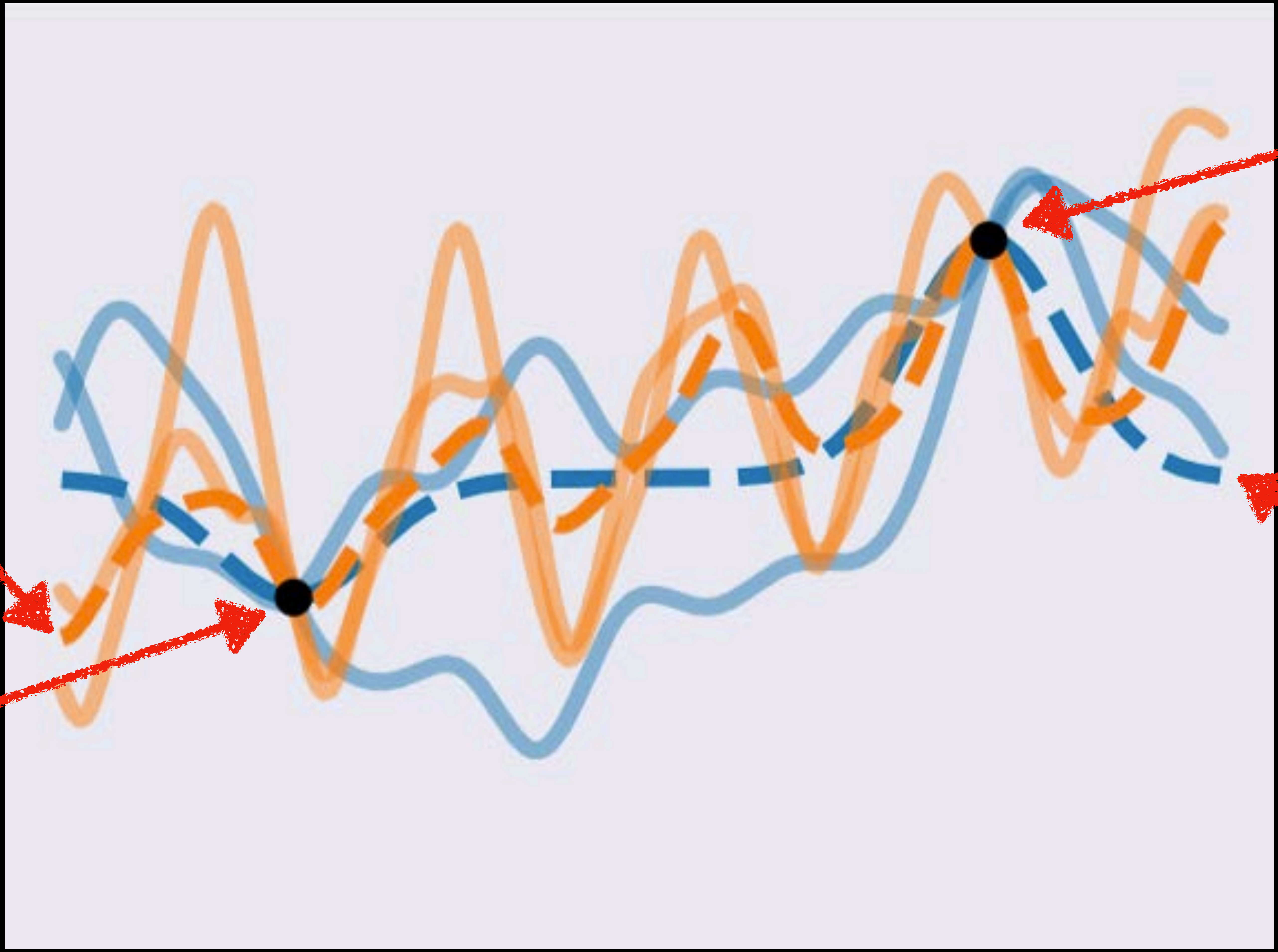
$$\mu_{|\zeta_m}(\mathbf{x}) = \mu(\mathbf{x}) + k(\mathbf{x}, C)k(C, C)^{-1}(m - \mu(C)),$$

$$k_{|\zeta_m}(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, C)k(C, C)^{-1}k(C, \mathbf{x})$$

Conditional sampling involves computing matrix inverse $k(C, C)^{-1}$, which has $\mathcal{O}(n^3)$ time complexity with n observations.

Background & Notation

Conditioned Processes



Posterior Mean (LP)

Observation 2

Observation 1

Posterior Mean (SE)

Background & Notation

Sampling from GPs

Samples at a set of points X can be drawn from a GP(μ, k) via

$$f(X) = \mu(X) + k(X, X)^{\frac{1}{2}}\eta$$

where $\eta \sim \mathcal{N}(0,I)$ and $A^{\frac{1}{2}}$ is the matrix square root. This computation has also time complexity $\mathcal{O}(p^3)$ where $p = |X|$.

Background & Notation

Derivatives of GPs

Thanks to the linearity of the derivative operator, the derivative of a GP is another GP:

$$\text{GP}'\left(\mu\left(\mathbf{x}\right), k\left(\mathbf{x}, \mathbf{y}\right)\right) = \text{GP}\left(\mu'\left(\mathbf{x}\right), k_{\mathbf{x},\mathbf{y}}\left(\mathbf{x}, \mathbf{y}\right)\right)$$

where $k_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y}) = \dfrac{\partial^2 k(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x} \partial \mathbf{y}}$.

Background & Notation

Derivatives of GPs

Furthermore, the joint value-derivative distribution can be computed as:

$$\begin{bmatrix} f(X) \\ f'(Y) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(X) \\ \mu'(Y) \end{bmatrix}, \begin{bmatrix} k(X, X) & k_{\mathbf{y}}(X, Y) \\ k_{\mathbf{x}}(Y, X) & k_{\mathbf{x},\mathbf{y}}(Y, Y) \end{bmatrix} \right)$$

This property will be later exploited when we consider distributions of normal vectors on surfaces derived from GPs.

Background & Notation

Implicit Surfaces

Each function sampled from a GP represents an implicit surface

$$f : \Omega \rightarrow \mathbb{R}, \left\{ \mathbf{x} \in \Omega \,|\, f(\mathbf{x}) = l \right\}$$

where l is the signed distance from $\mathbf{x}$ to the closest surface point.

Computing an intersection between a ray and the surface is equivalent to finding the ray distance s such that $f(\mathbf{x} + s\omega) = 0$. More precisely,

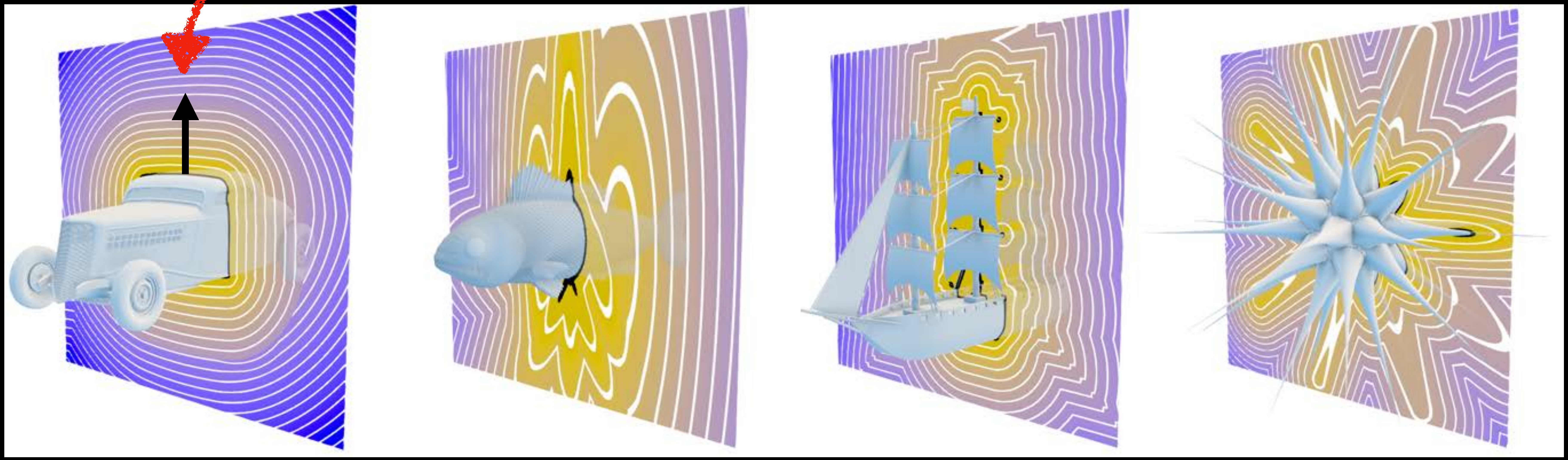
$$s = \text{argmin}_{t \in \mathbb{R}^+} f(\mathbf{x}_t) = 0$$

Background & Notation

Implicit Surfaces

The normal vector at the intersection $\mathbf{x}_s$ can be easily computed as

$$\mathbf{n}_s = \frac{\nabla f(\mathbf{x}_s)}{\|\nabla f(\mathbf{x}_s)\|}$$



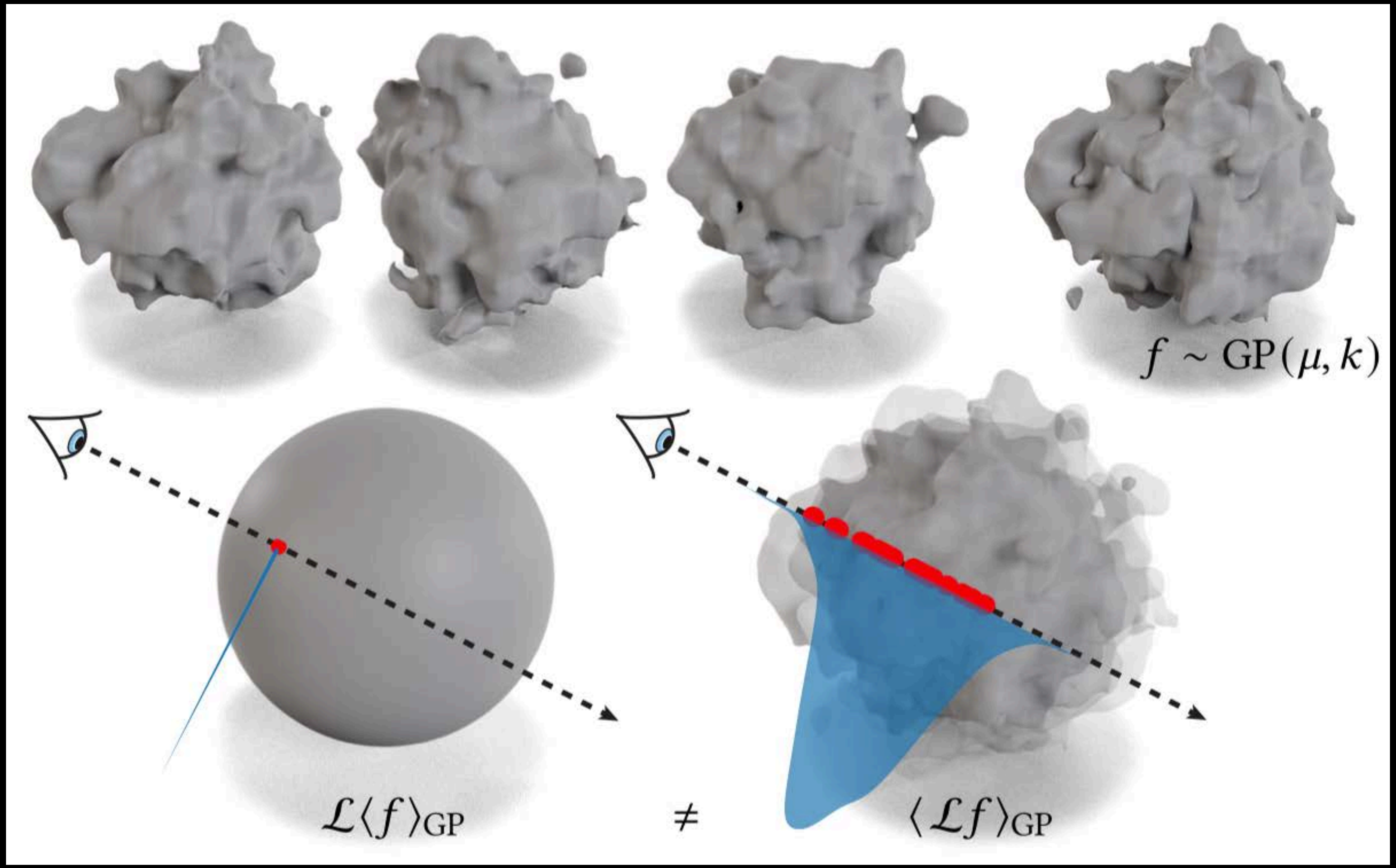
A Dataset and Explorer for 3D Signed Distance Functions, i3D 2022
Towaki Takikawa, Andrew Glassner, and Morgan McGuire

Background & Notation

Stochastic Implicit Surfaces

A **stochastic implicit surface (SIS)** is the distribution of level sets defined by a stochastic process. Our interest is **Gaussian Process Implicit Surfaces (GPIS)**.

Light Transport
(*Mean* Implicit Surface)



***Mean* Light Transport**
(Over All Realizations)

Key Idea

Ensemble-Averaged Light Transport in GPISes

First, consider the surface rendering equation [Kajiya 1986] in a scene defined by an implicit surface f:

$$L^f(\mathbf{x}, \boldsymbol{\omega}) = \int_{S^2} \rho(\mathbf{x}_s^f) L^f(\mathbf{x}_s^f, \boldsymbol{\omega}_s^f) d\boldsymbol{\omega}_s,$$

where $\rho(\mathbf{x}_s) = \rho(\mathbf{x}_s, -\boldsymbol{\omega}, \boldsymbol{\omega}_s, \mathbf{n}_s) \,|\, \mathbf{n}_s \cdot \boldsymbol{\omega}_s \,|$ is the cosine-weighted BRDF.

This equation explains light transport on for a *fixed* surface f.

Key Idea

Ensemble-Averaged Light Transport in GPISes

Now, assume that f is a realization of a GP. The **ensemble averaged light transport** over all realization of f of the GP $\mathrm{GP}(\mu, k \mid \zeta)$ is defined as:

$$\langle L^f(\mathbf{x}, \boldsymbol{\omega}) \rangle_\zeta = \int_{\mathrm{GP}(\mu,k|\zeta)} L^f(\mathbf{x}, \boldsymbol{\omega}) d\gamma_{\mu,k}(f \mid \zeta),$$

LT over a single, fixed surface f

$$L^f(\mathbf{x}, \boldsymbol{\omega}) = \int_{S^2} \rho(\mathbf{x}_s^f) L^f(\mathbf{x}_s^f, \boldsymbol{\omega}_s^f) d\boldsymbol{\omega}_s$$

Probability density of $f \sim \mathrm{GP}(\mu, k \mid \zeta)$

Key Idea

Ensemble-Averaged Light Transport in GPISes

A Monte Carlo estimator for the equation naturally follows as

$$\langle L^{f}(\mathbf{x}, \boldsymbol{\omega}) \rangle_{\zeta} \approx \frac{1}{N} \sum_{j=1}^{N} L^{f_j}(\mathbf{x}, \boldsymbol{\omega}),$$

where $f_j \sim \mathrm{GP}(\mu, k \,|\, \zeta)$ is one realization of implicit functions implied by the GP.

"Iteratively sample f from GP and simulate light transport using MC!"

🤩

Experiments (…🤔)

Key Idea

Ensemble-Averaged Light Transport in GPISes

For each sample L^{f_j}, **an entire 3D realization of f_j** must be constructed!

Assume that our GPIS is discretized into a volume of sidelength $\mathcal{O}(n)$.

What is the time complexity of constructing each f_j?

HINT

$$f \sim \text{GP}\left(\mu_{|\zeta_m}(\mathbf{x}), k_{|\zeta_m}(\mathbf{x}, \mathbf{y})\right), \text{ where}$$

$$\mu_{|\zeta_m}(\mathbf{x}) = \mu(\mathbf{x}) + k(\mathbf{x}, C)k(C, C)^{-1}(m - \mu(C)),$$

$$k_{|\zeta_m}(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, C)k(C, C)^{-1}k(C, \mathbf{x})$$

$$f(X) = \mu(X) + k(X, X)^{\frac{1}{2}}\eta$$

Key Idea

Ensemble-Averaged Light Transport in GPISes

For each sample L^{f_j}, **an entire 3D realization of f_j** must be constructed!

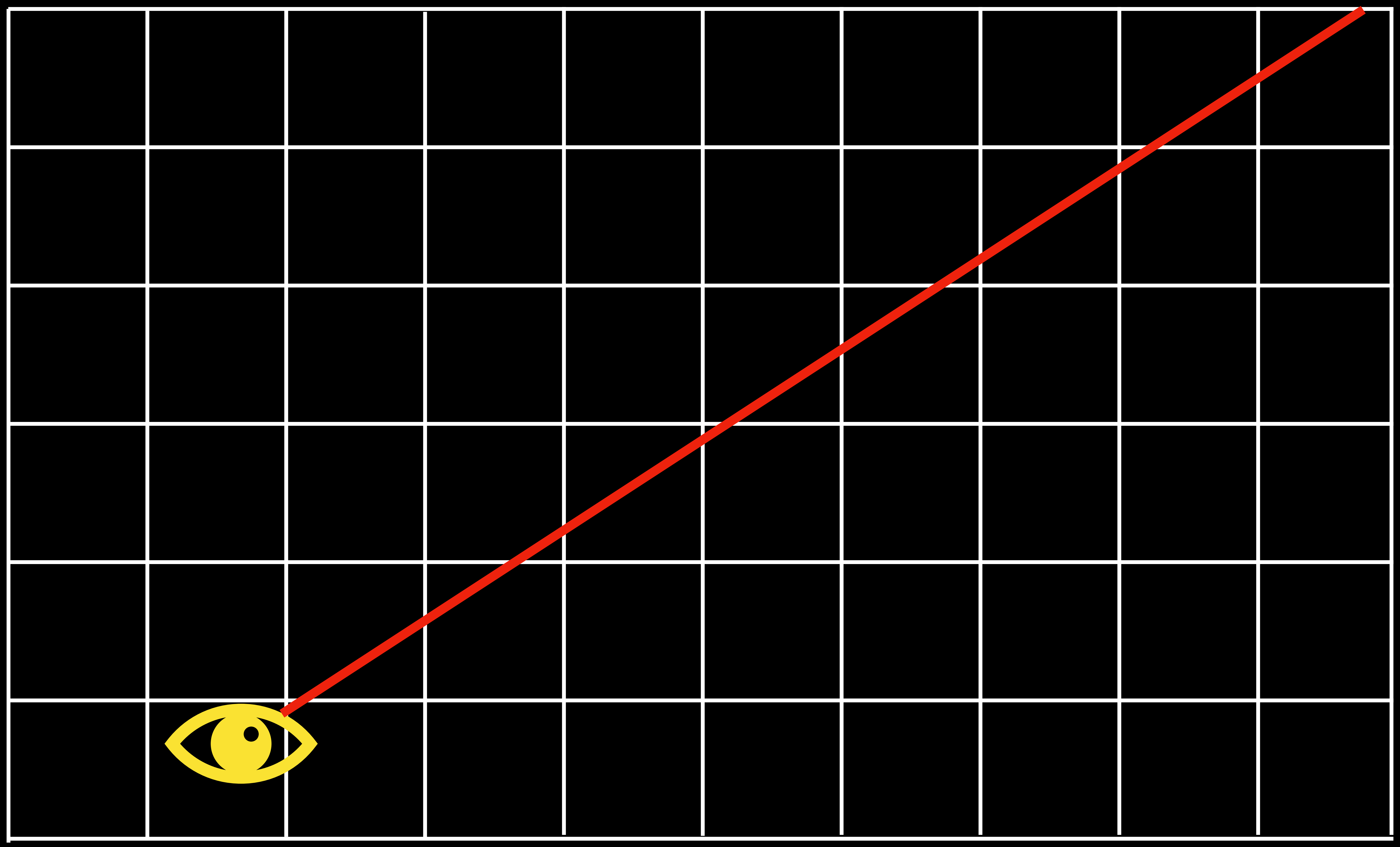
Assume that our GPIS is discretized into a volume of sidelength $\mathcal{O}(n)$.

What is the time complexity of constructing each f_j?

$$O(n^9) = O\left(\left(n^3\right)^3\right)$$

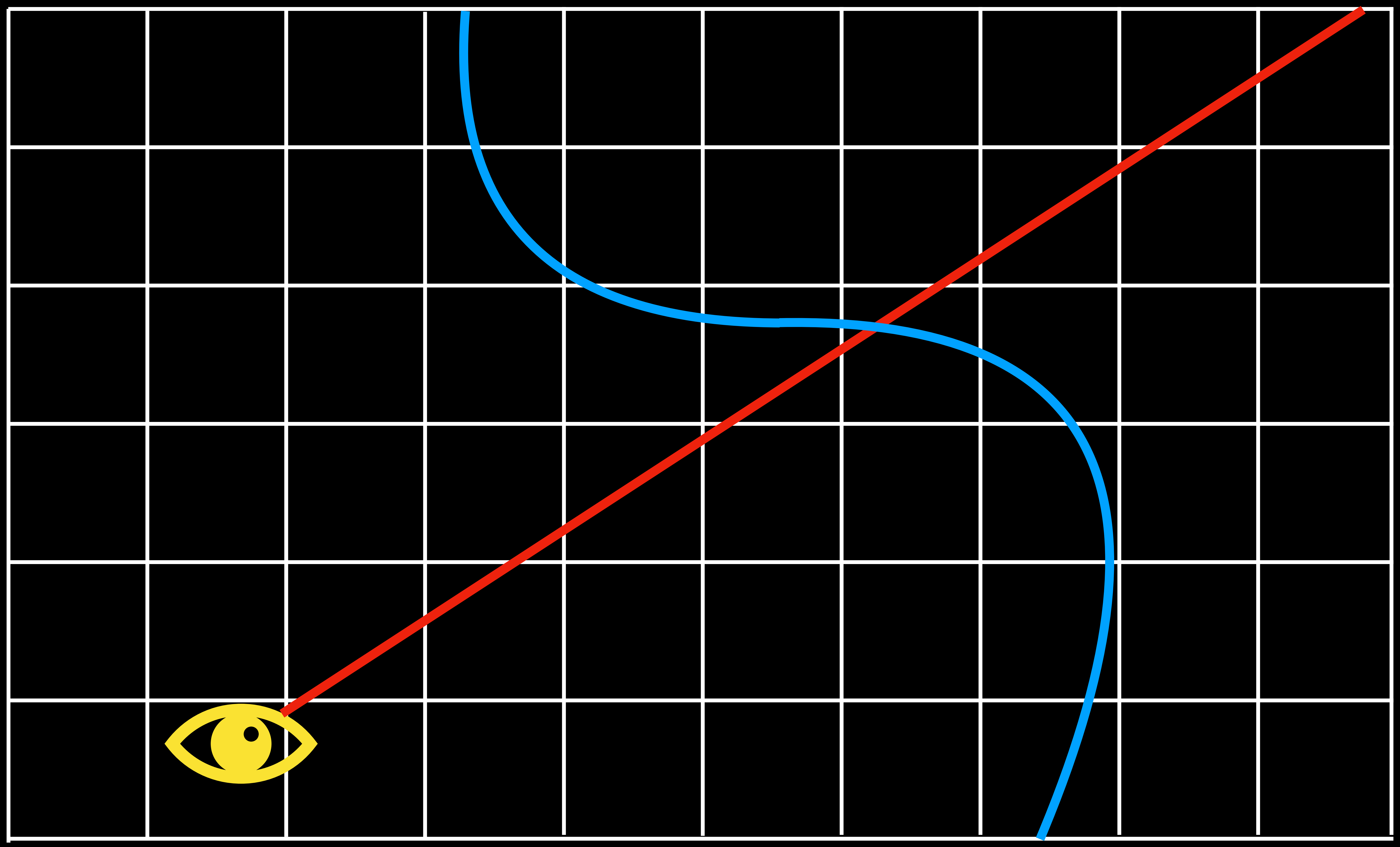
Key Idea

Ensemble-Averaged Light Transport in GPISes



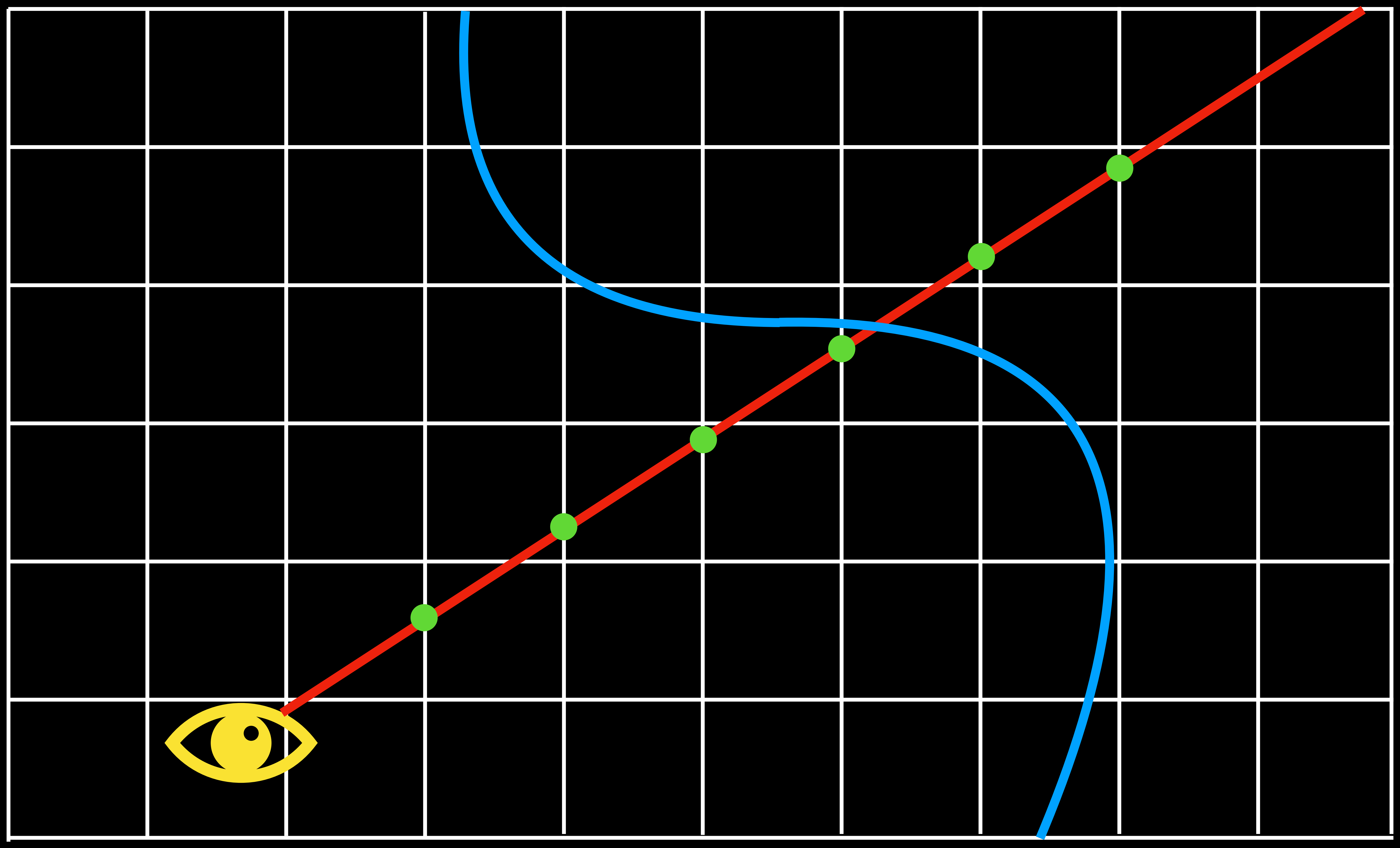
Key Idea

Ensemble-Averaged Light Transport in GPISes



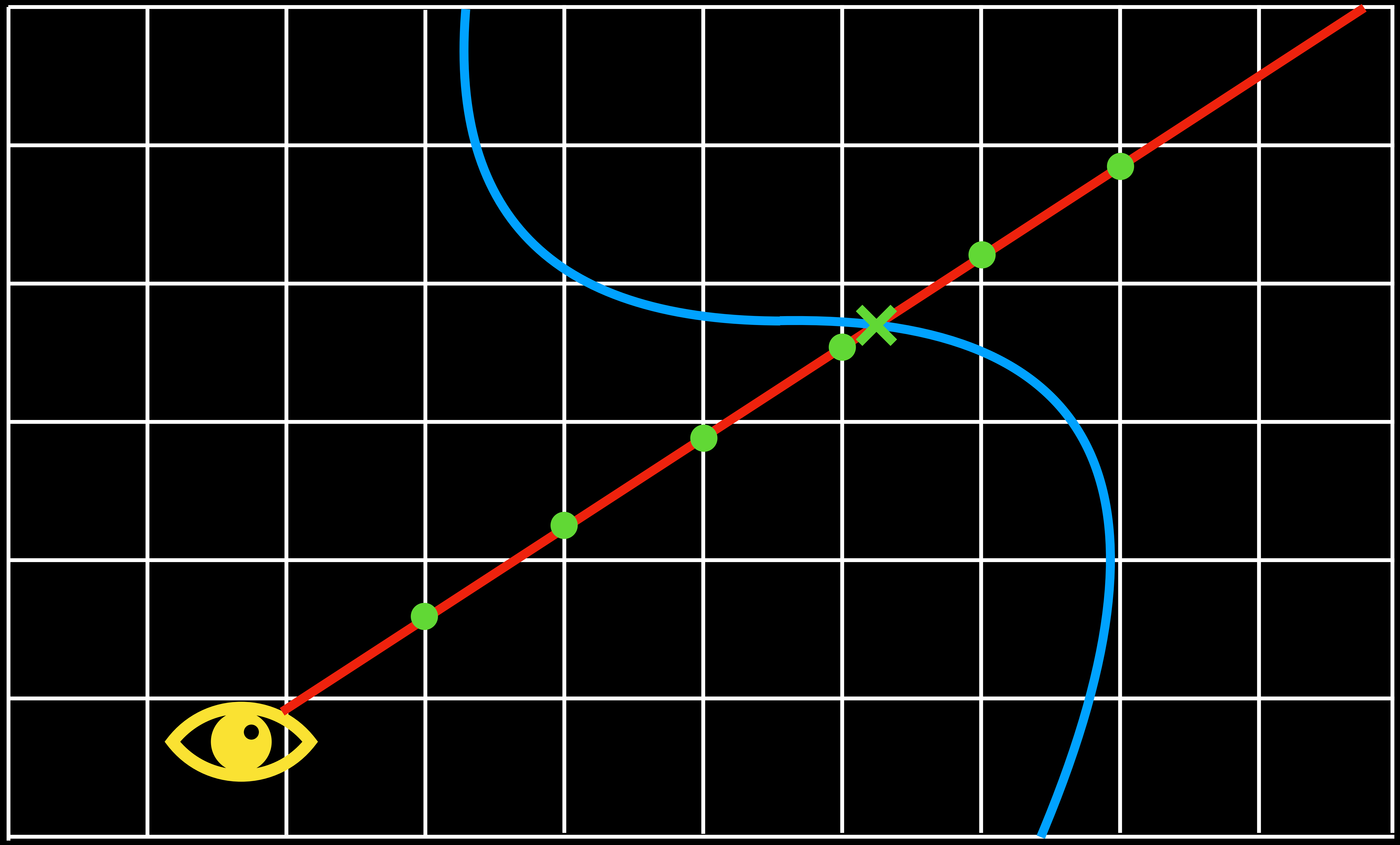
Key Idea

Ensemble-Averaged Light Transport in GPISes



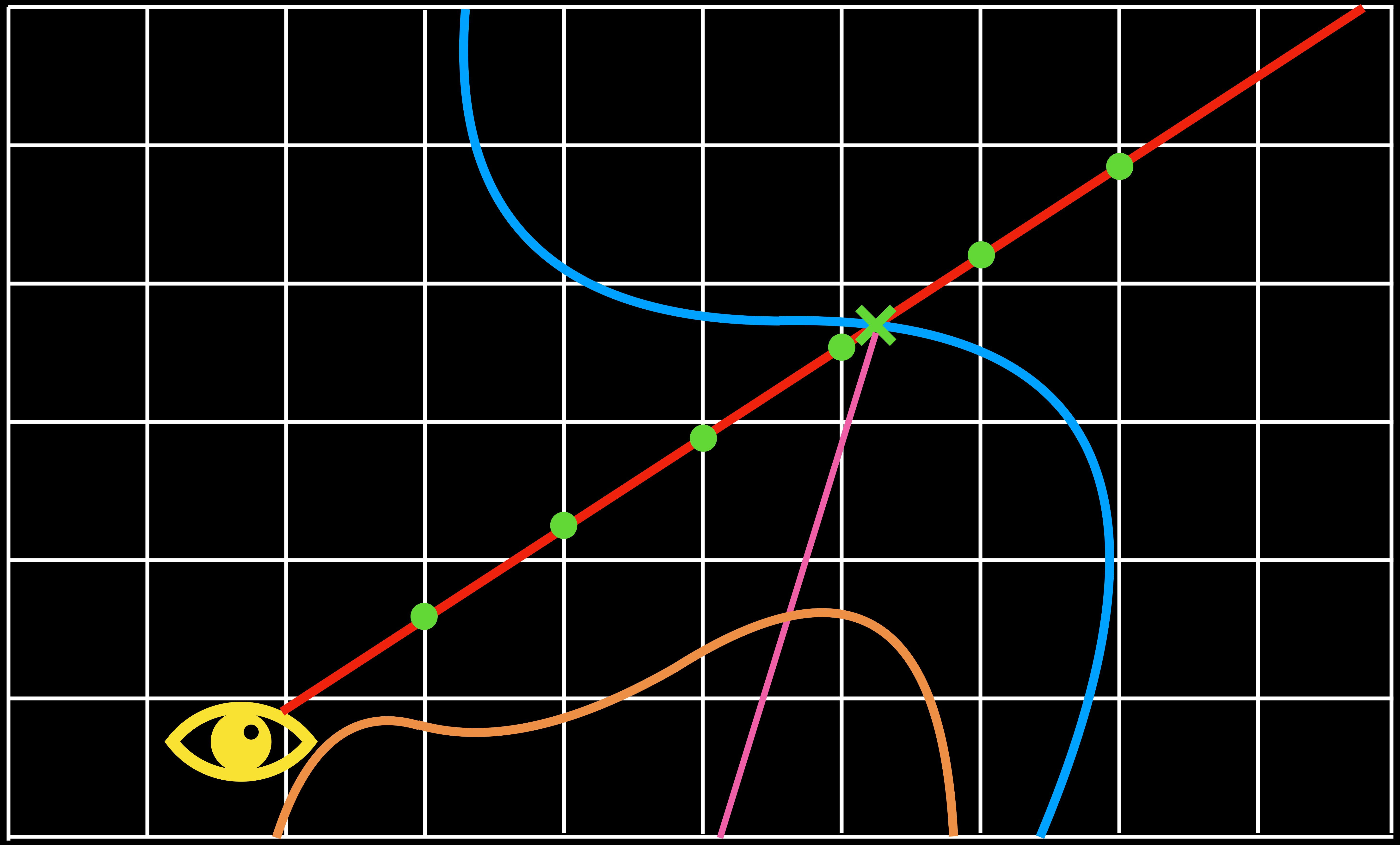
Key Idea

Ensemble-Averaged Light Transport in GPISes



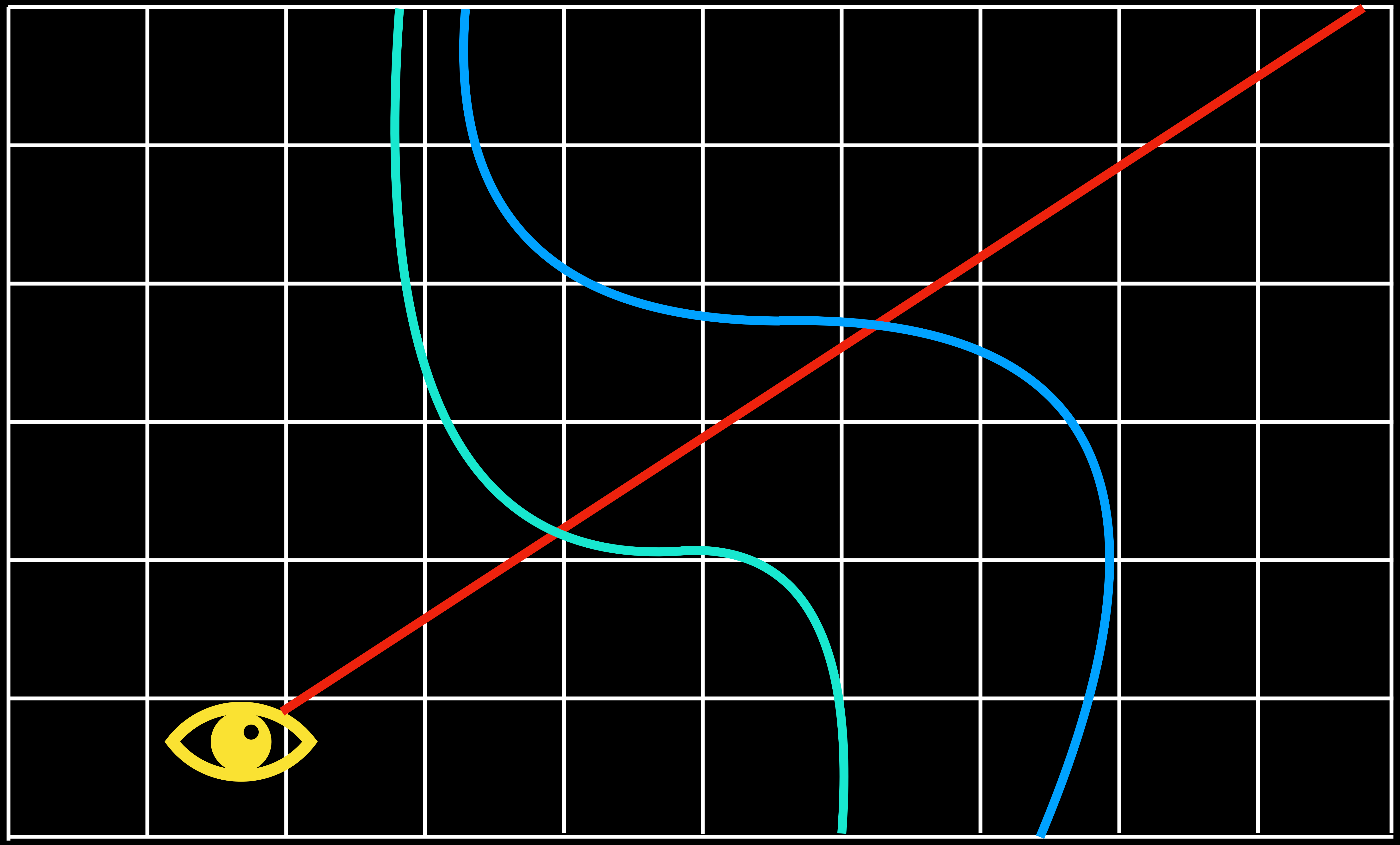
Key Idea

Ensemble-Averaged Light Transport in GPISes



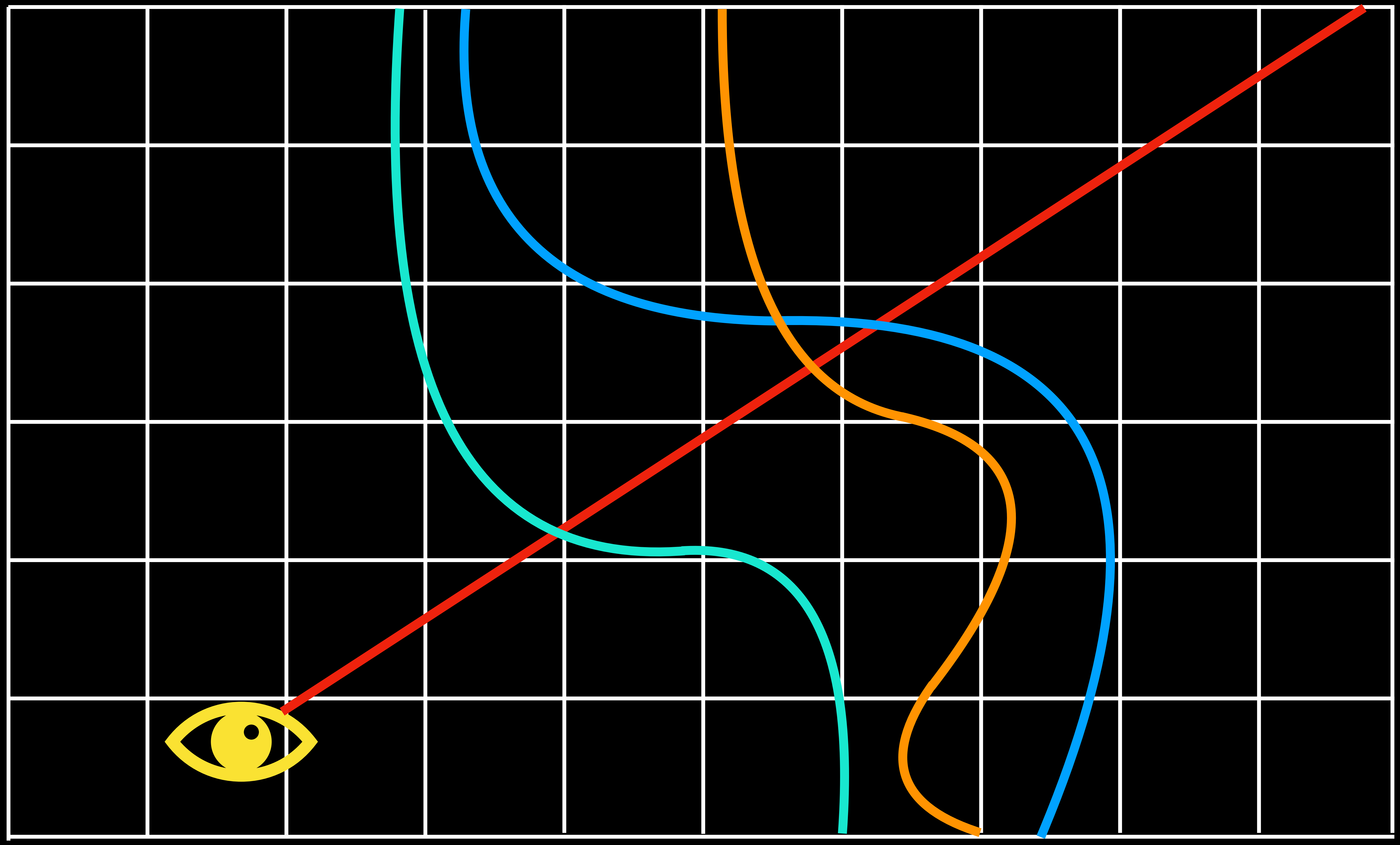
Key Idea

Ensemble-Averaged Light Transport in GPISes



Key Idea

Ensemble-Averaged Light Transport in GPISes



Key Idea

Ensemble-Averaged Light Transport in GPISes

Free-flight distributions of f are delta functions for fixed implicit surfaces:

$$L^f(\mathbf{x}, \boldsymbol{\omega}) = \int_0^\infty \int\int_{S^2} \rho(\mathbf{x}_t)\delta^f(\mathbf{x}_t, \mathbf{n}) I^f(0,t) L^f(\mathbf{x}_t, \boldsymbol{\omega}_t) d\boldsymbol{\omega}_t d\mathbf{n} dt,$$

where $\delta^f(\mathbf{x}_t, \mathbf{n}) = \delta\left(f\left(\mathbf{x}_t\right) - 0\right) \cdot \delta\left(\frac{\nabla f\left(\mathbf{x}_t\right)}{\|\nabla f\left(\mathbf{x}_t\right)\|} - \mathbf{n}\right)$ and

$$I^f(0,t) = \begin{cases} 1 & \forall s \in (0,t) : f(\mathbf{x}_s) > 0 \\ 0 & \text{otherwise} \end{cases}.$$

Key Idea

Ensemble-Averaged Light Transport in GPISes

Since the BRDF ρ is independent of realizations f, the ensemble-averaged LT can be computed as

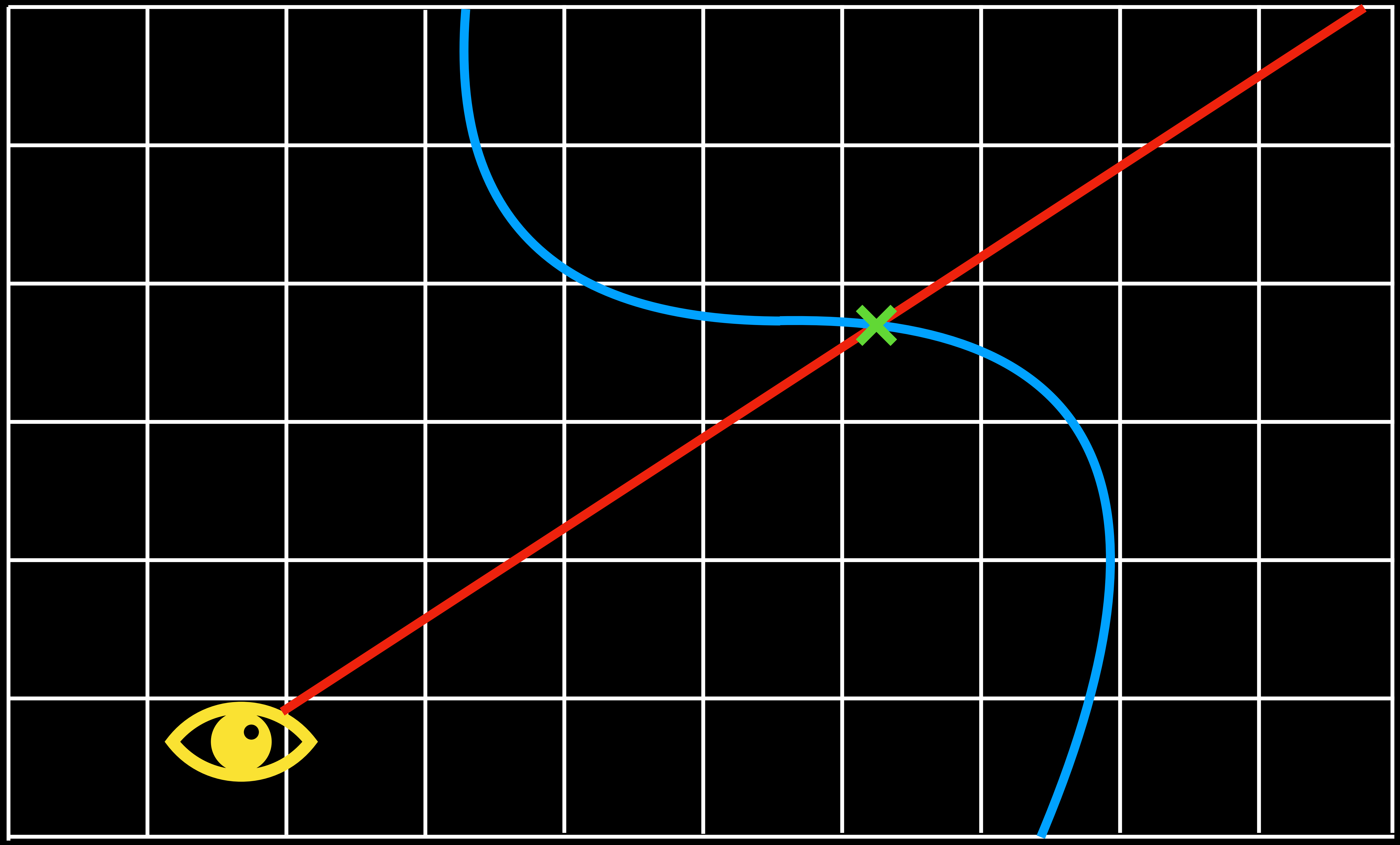
$$\langle L^f(\mathbf{x}, \boldsymbol{\omega}) \rangle_\zeta = \int_0^\infty \iint_{S^2} \rho(\mathbf{x}_t) \Big\langle \delta^f(\mathbf{x}_t, \mathbf{n}) I^f(0,t) L^f(\mathbf{x}_t, \boldsymbol{\omega}_t) \Big\rangle_\zeta d\boldsymbol{\omega}_t d\mathbf{n} dt,$$

where $\delta^f(\mathbf{x}_t, \mathbf{n}) = \delta\left(f\left(\mathbf{x}_t\right) - 0\right) \cdot \delta\left(\dfrac{\nabla f\left(\mathbf{x}_t\right)}{\|\nabla f\left(\mathbf{x}_t\right)\|} - \mathbf{n}\right)$ and

$$I^f(0,t) = \begin{cases} 1 & \forall s \in (0,t) : f(\mathbf{x}_s) > 0 \\ 0 & \text{otherwise} \end{cases}.$$

Key Idea

Ensemble-Averaged Light Transport in GPISes



Key Idea

Ensemble-Averaged Light Transport in GPISes

$$\langle L^f(\mathbf{x},\boldsymbol{\omega})\rangle_\zeta = \int_0^\infty \iint_{S^2} \rho(\mathbf{x}_t) \Big\langle \delta^f(\mathbf{x}_t,\mathbf{n}) I^f(0,t) L^f(\mathbf{x}_t,\boldsymbol{\omega}_t) \Big\rangle_\zeta d\boldsymbol{\omega}_t d\mathbf{n} dt$$

Repeat:

1. **Sample $f \sim$ GP$(\mu, k \,|\, \zeta)$; $\rightarrow \mathcal{O}(n^9)$ time complexity**
2. Compute $L^f(\mathbf{x}_t, \boldsymbol{\omega}_t)$ via path tracing;
3. Use the path tracing result when evaluating the expectation if,
   A. $\mathbf{x}_t$ is the point where the ray initially intersects with f (from δ^f and I^f);
   B. The normal at $\mathbf{x}_t$ is aligned with $\mathbf{n}$ (from δ^f).

Key Idea

Ensemble-Averaged Light Transport in GPISes

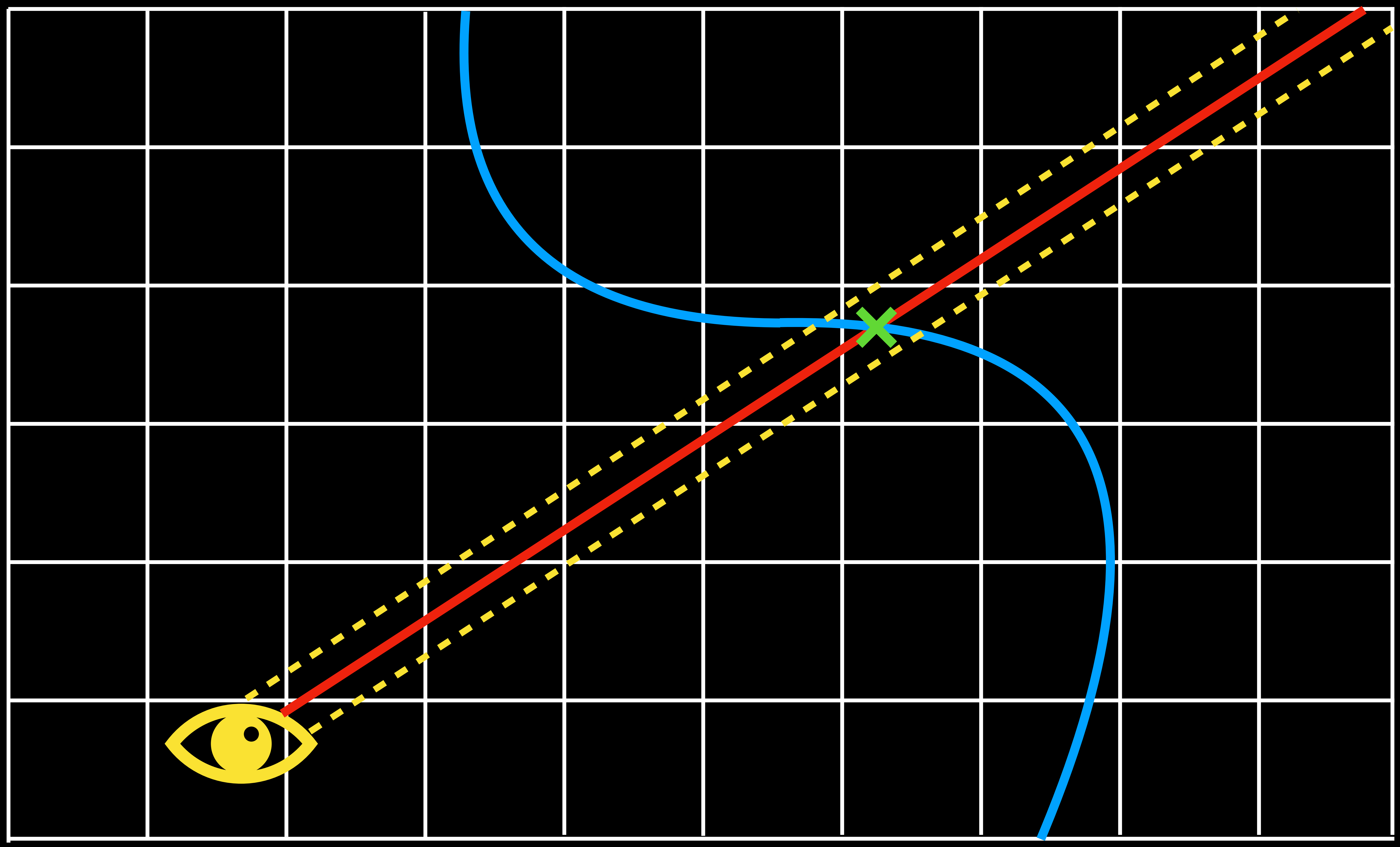
Instead of *filtering* realizations after sampling, we can average over realizations having the required intersection point $\mathbf{x}_t$ and normal $\mathbf{n}$:

$$\langle L^f(\mathbf{x}, \boldsymbol{\omega}) \rangle_\zeta = \int_0^\infty \iint_{S^2} \rho(\mathbf{x}_t) \gamma_{\mathbf{x}_t}(0, \mathbf{n} \,|\, \zeta) \left\langle I^f(0,t) L^f(\mathbf{x}_t, \boldsymbol{\omega}_t) \right\rangle_{\zeta \wedge \zeta_\delta} d\boldsymbol{\omega}_t d\mathbf{n} dt,$$

where $\zeta_\delta = \left(f\left(\mathbf{x}_t\right) = 0 \wedge \nabla f(\mathbf{x}_t)/||\nabla f(\mathbf{x}_t)|| = \mathbf{n} \right)$ and $\gamma_{\mathbf{x}_t}(0, \mathbf{n} \,|\, \zeta)$ is the density of sampling realizations that satifsy the condition ζ_δ.

Key Idea

Ensemble-Averaged Light Transport in GPISes



Key Idea

Ensemble-Averaged Light Transport in GPISes

Since $I^f(0,t)$ only depends on f over 1D ray segment $(\mathbf{x}, \mathbf{x}_t)$, we decompose sampling f into two steps:

1. Sample the values $f_{\mathbf{x},\mathbf{x}_t}$ along the ray segment;
2. Continue sample f over the remainder of the domain.

Theoretically, this decomposition does not alter the statistics of a GP.

Key Idea

Ensemble-Averaged Light Transport in GPISes

The final equation for the ensemble-averaged light transport is

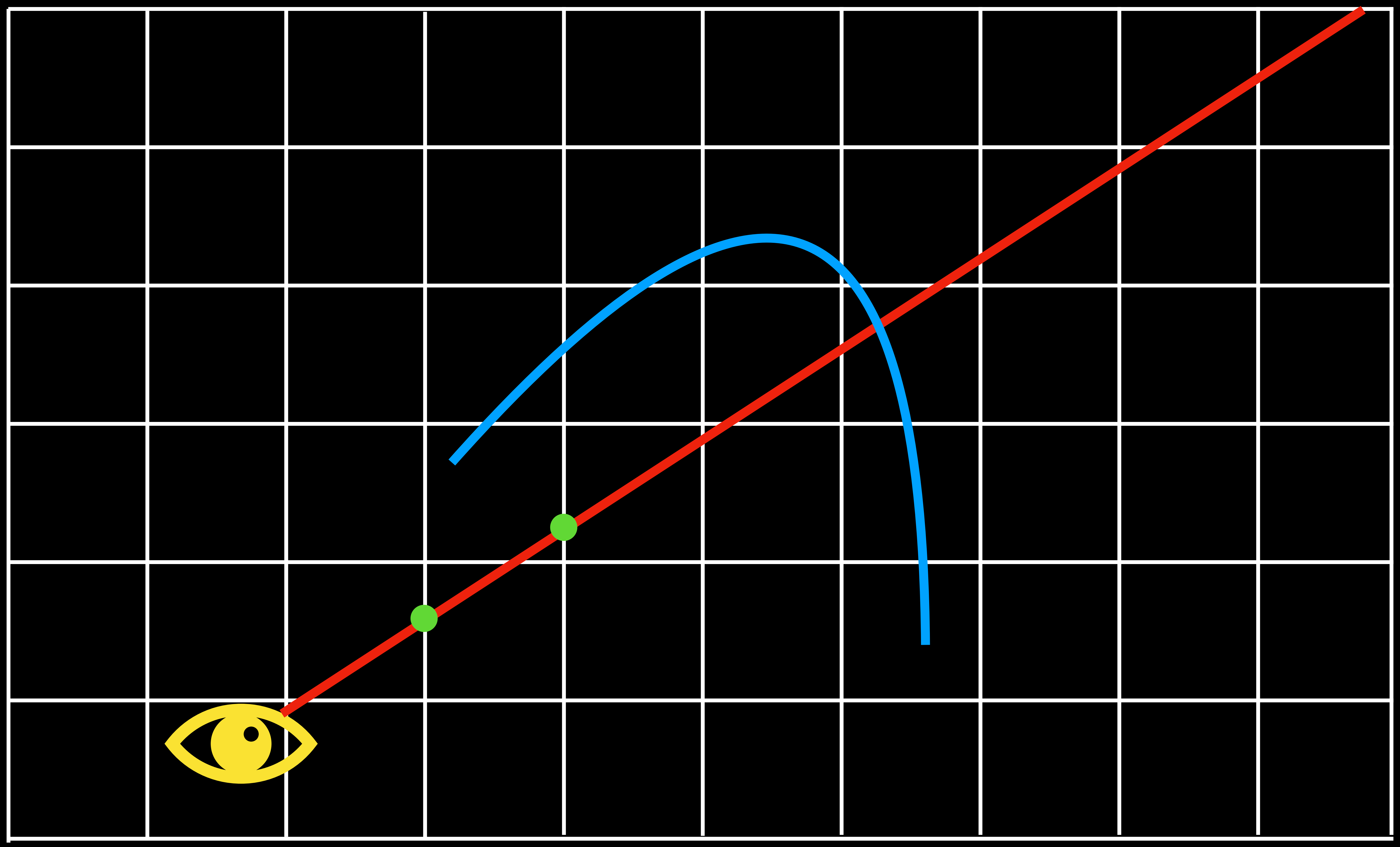
$$\langle L^f(\mathbf{x}, \boldsymbol{\omega}) \rangle_\zeta = \int_0^\infty \iint_{S^2} \rho(\mathbf{x}_t) \gamma_{\mathbf{x}_t}(0, \mathbf{n} \mid \zeta) \left\langle I^f(0,t) \left\langle L^f(\mathbf{x}_t, \boldsymbol{\omega}_t) \right\rangle_{\zeta \wedge \zeta_\delta \wedge \zeta_{(\mathbf{x},\mathbf{x}_t)}} \right\rangle_{\zeta \wedge \zeta_\delta}^{(\mathbf{x},\mathbf{x}_t)} d\boldsymbol{\omega}_t d\mathbf{n} dt$$

where $\langle \cdot \rangle_\zeta^{(\mathbf{x},\mathbf{x}_t)}$ is the conditioned average over realization restricted to a path segment.

$$\mathcal{O}(m^3)$$

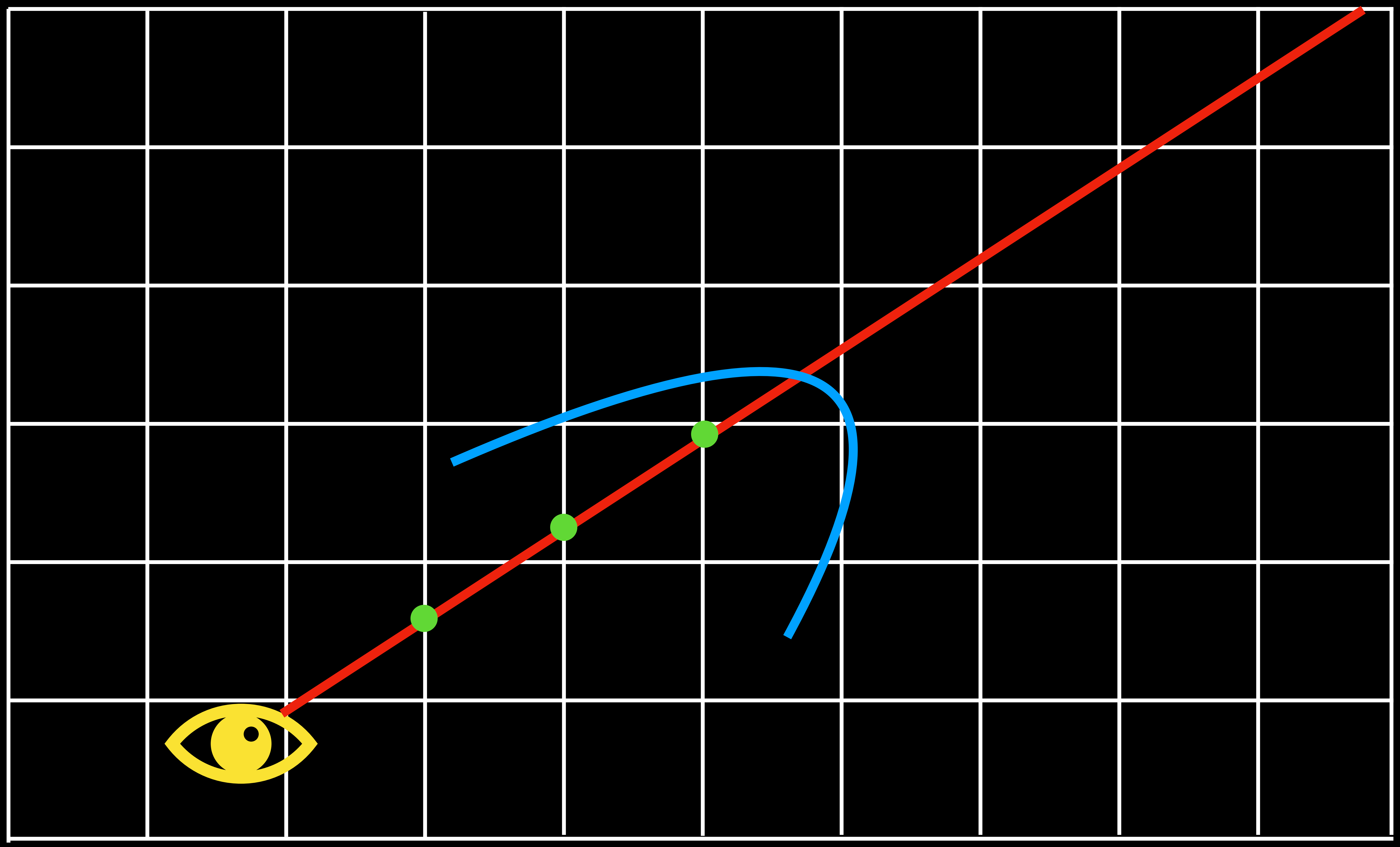
Key Idea

Ensemble-Averaged Light Transport in GPISes



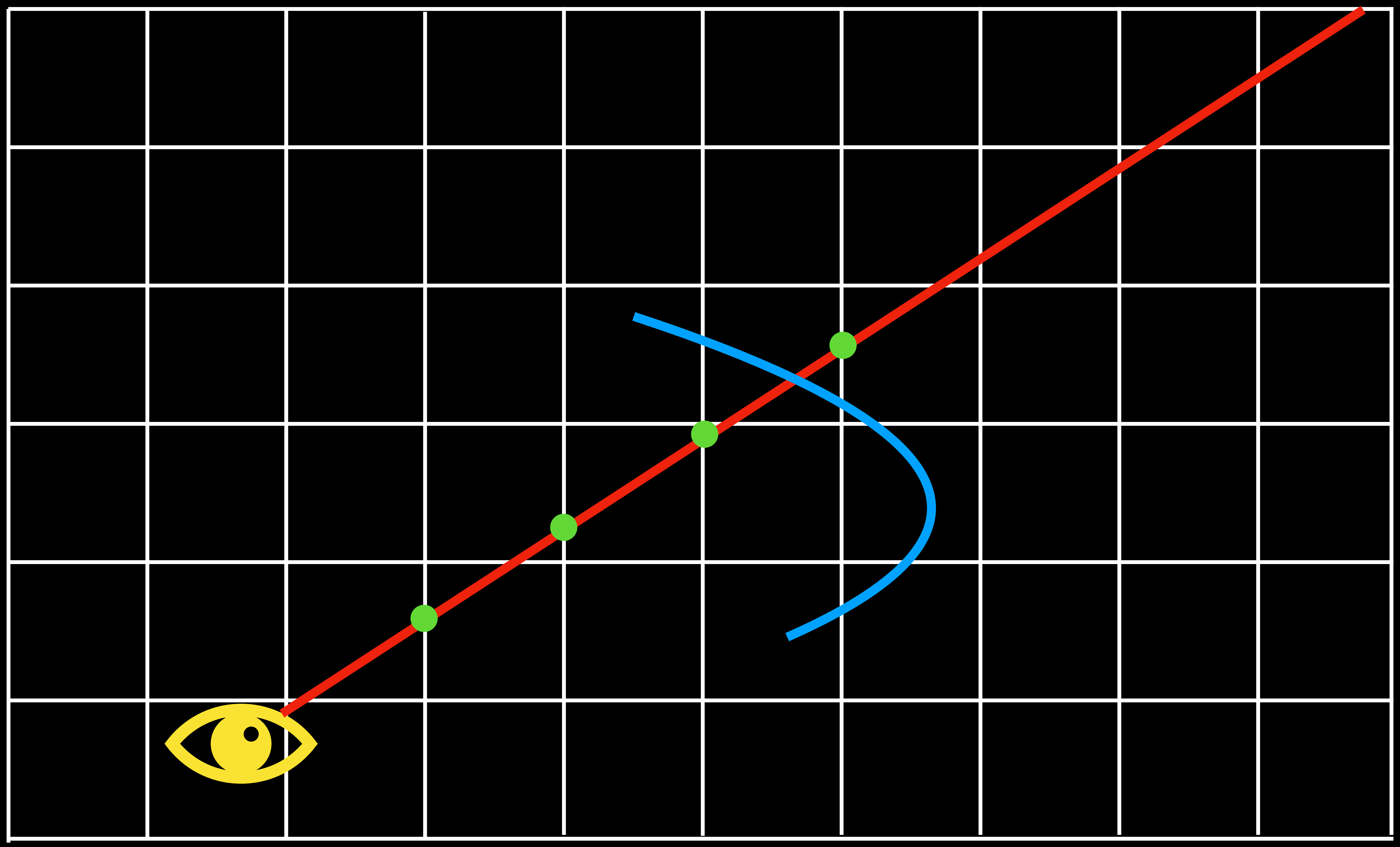
Key Idea

Ensemble-Averaged Light Transport in GPISes



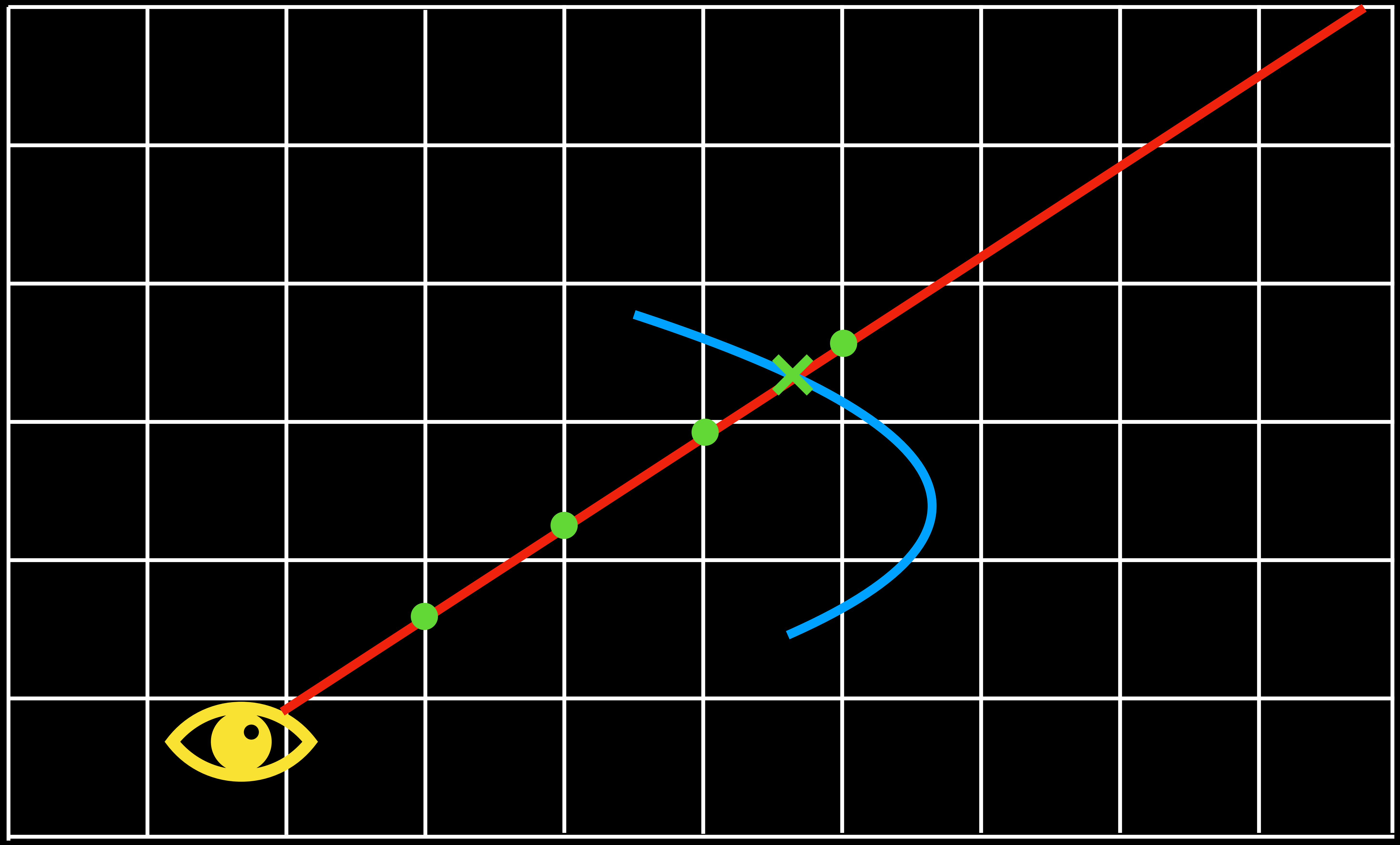
Key Idea

Ensemble-Averaged Light Transport in GPISes



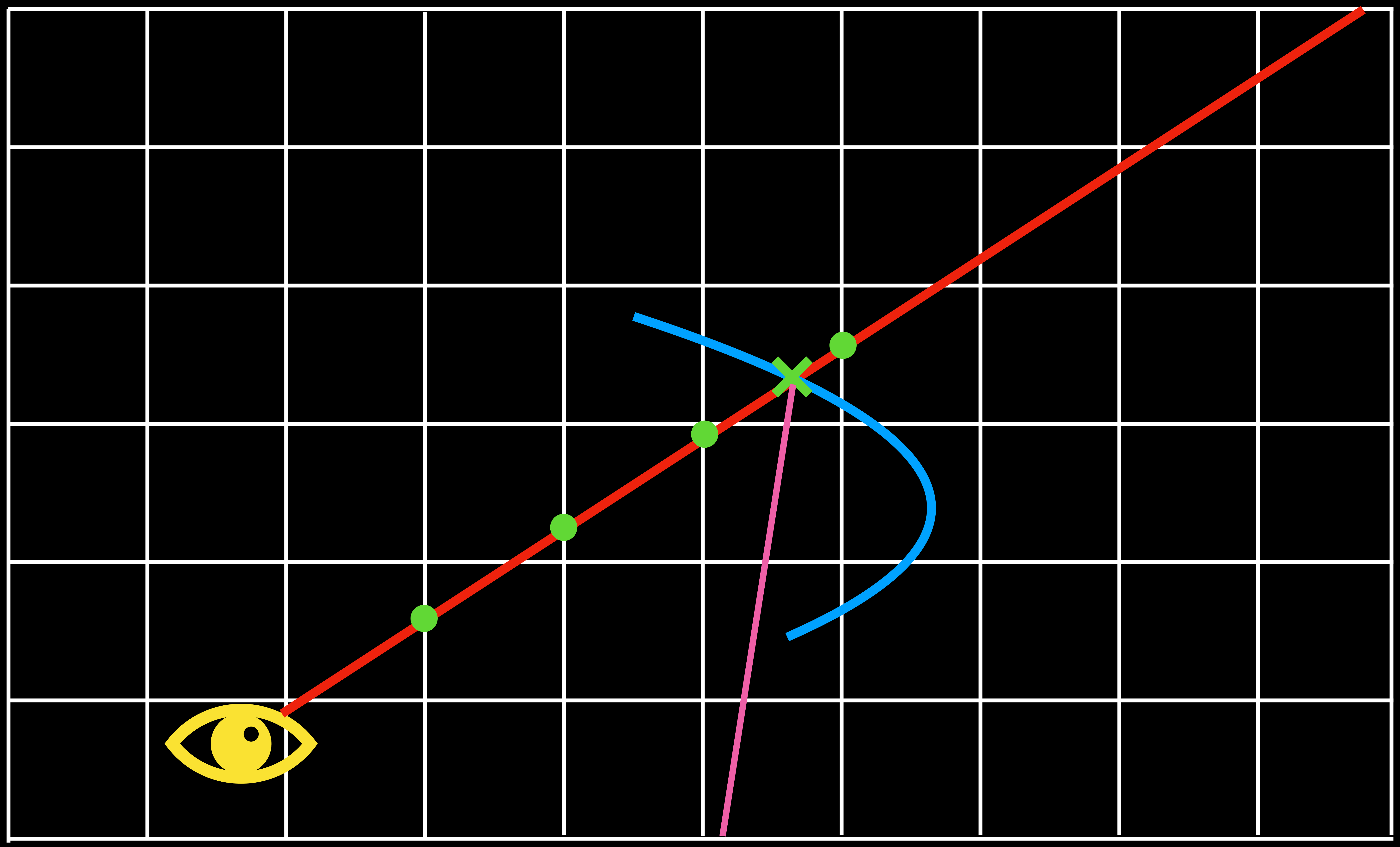
Key Idea

Ensemble-Averaged Light Transport in GPISes



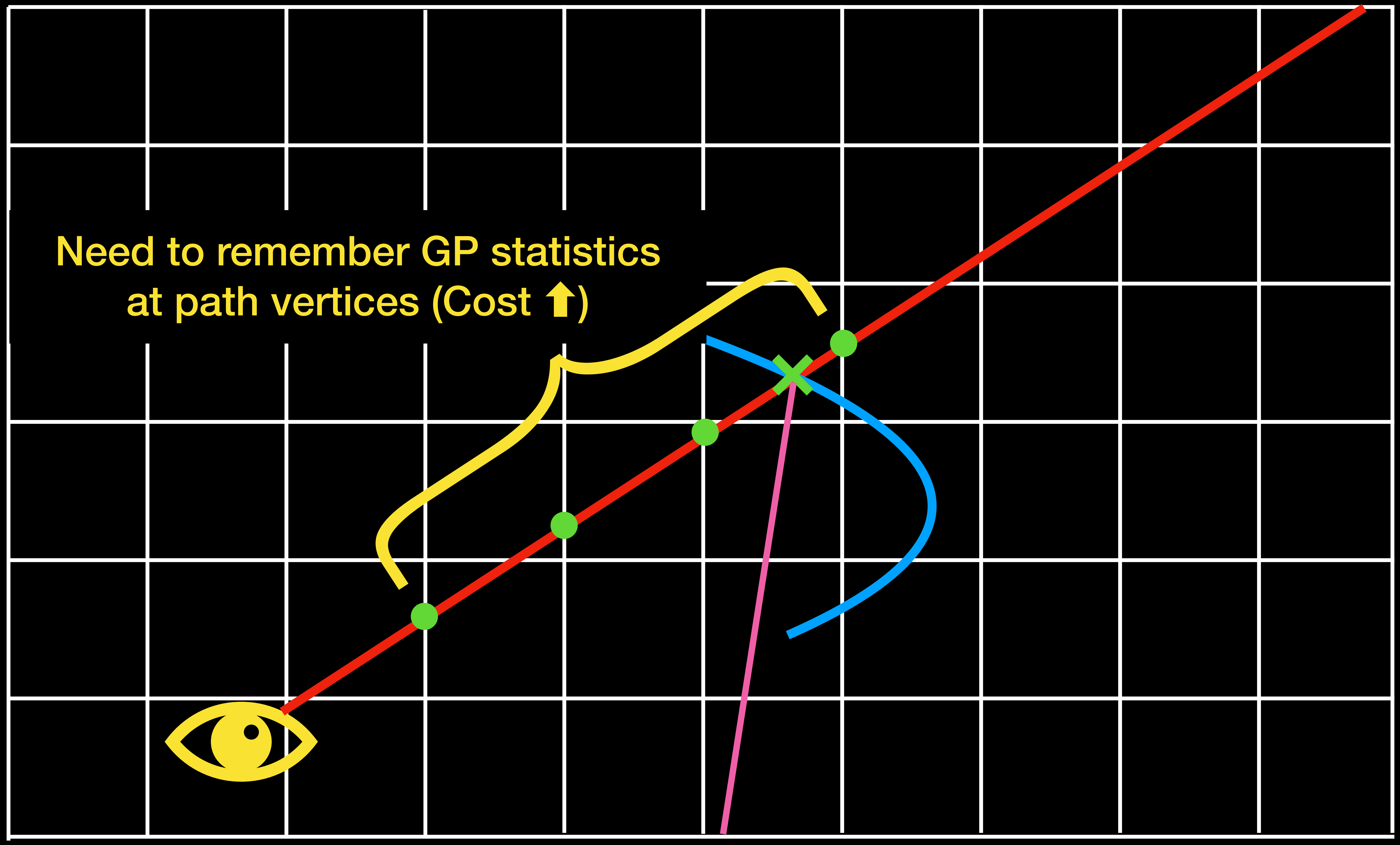
Key Idea

Ensemble-Averaged Light Transport in GPISes



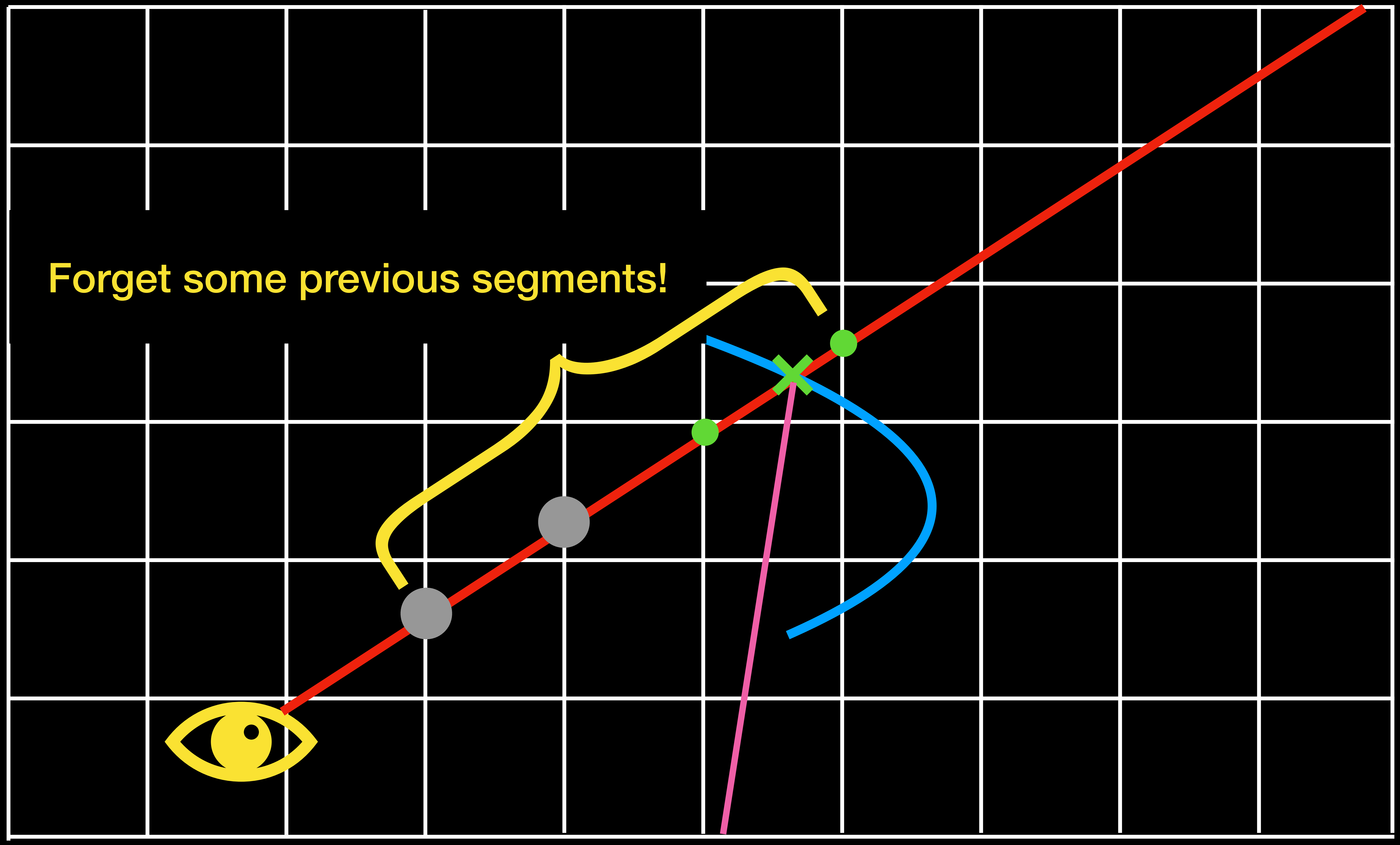
Key Idea

Ensemble-Averaged Light Transport in GPISes



Key Idea

Ensemble-Averaged Light Transport in GPISes



Key Idea

Memory Models for Rendering Equation Evaluation

✅ ❌ ✅

Key Idea

Progressive Sampling via Function-Space GPs

$$\langle L^f(\mathbf{x},\omega)\rangle_\zeta = \int_0^\infty \int\int_{S^2} \rho(\mathbf{x}_t)\gamma_{\mathbf{x}_t}(0,\mathbf{n}\,|\,\zeta)\left\langle I^f(0,t)\left\langle L^f\left(\mathbf{x}_t,\boldsymbol{\omega}_t\right)\right\rangle_{\zeta\wedge\zeta_\delta\wedge\zeta_{(\mathbf{x},\mathbf{x}_t)}}\right\rangle_{\zeta\wedge\zeta_\delta}^{(\mathbf{x},\mathbf{x}_t)} d\boldsymbol{\omega}_t d\mathbf{n} dt$$

$$\langle L_i\widehat{(\mathbf{x}^u,\boldsymbol{\omega})}\rangle_\zeta = \frac{\rho(\mathbf{x}_t)\Gamma(t,\mathbf{n}\,|\,\zeta)}{p(t,\mathbf{n},\boldsymbol{\omega}_t,f_{(\mathbf{x},\mathbf{x}_t)})}\langle L_i\widehat{(\mathbf{x}_t,\boldsymbol{\omega}_t)}\rangle'_\zeta,$$

where $t,\mathbf{n},\boldsymbol{\omega}_t,f_{(\mathbf{x},\mathbf{x}_t)} \sim p(t,\mathbf{n},\boldsymbol{w}_t,f_{(\mathbf{x},\mathbf{x}_t)})$

Appearance Models

Appearance Spaces of GPISes

Joint Distribution of Free-Flight Distances and Normals

Using the Renewal+, or Renewal models, our rendering equation can be simplified to

GPIS Density

$$\langle L(\mathbf{x},\boldsymbol{\omega})\rangle_{\zeta} \approx \int_0^{\infty}\iint \rho\left(\mathbf{x}_t\right)\Gamma\left(t,\mathbf{n}\,|\,\zeta\right)\left\langle L\left(\mathbf{x}_t,\boldsymbol{\omega}_t\right)\right\rangle_{\zeta\wedge\zeta'} d\boldsymbol{\omega}_t d\mathbf{n} dt,$$

where $\Gamma\left(t,\mathbf{n}\,|\,\zeta\right) = \gamma_{\mathbf{x}_t}\left(0,\mathbf{n}\,|\,\zeta\right) T\left(\mathbf{x}_t\,|\,\zeta\right)$ with the transmittance

$$T(\mathbf{x}_t\,|\,\zeta) = \int_{\mathrm{GP}_{(\mathbf{x},\mathbf{x}_t)}|\zeta\wedge\zeta_\delta} I^f\left(0,t\right) d\gamma\left(f_{\left(\mathbf{x},\mathbf{x}_f\right)}\,|\,\zeta\wedge\zeta_\delta\right)$$

Appearance Spaces of GPISes

Surface-Type GPISes

Similarly to GPIses, microfacet surfaces are regarded as realizations of a stochastic process (*e.g.*, Beckmann model).

One important attribute is the **distribution of visible normals (vNDF)** $D_v(\mathbf{n} \mid \boldsymbol{\omega})$. In this framework, it is elegantly derived from $\Gamma(t, \mathbf{n} \mid \zeta)$ as

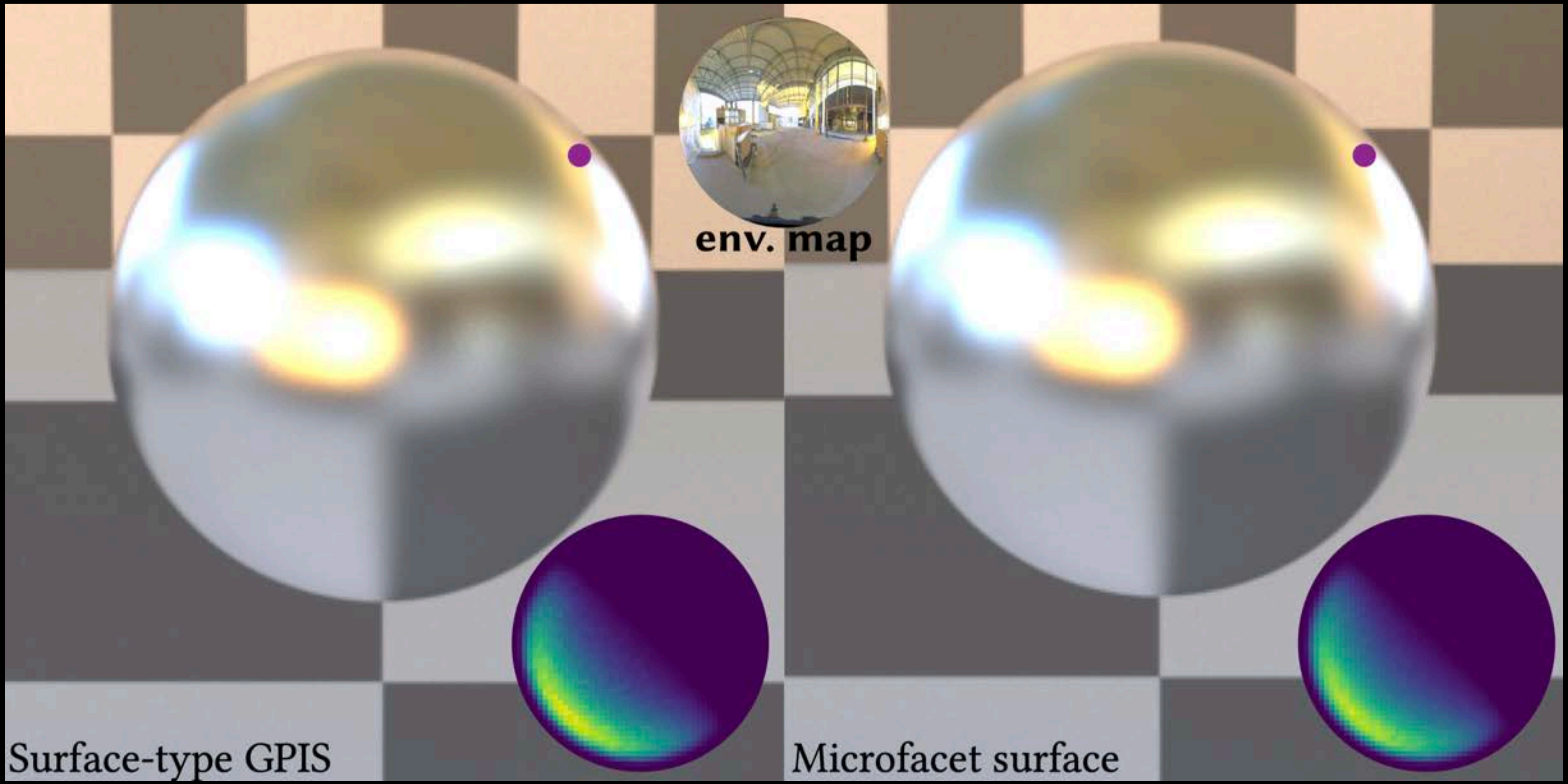
$$D_v(\mathbf{n} \mid \boldsymbol{\omega}) = \int_0^\infty \Gamma\left(t, \mathbf{n} \mid \boldsymbol{\omega}, \zeta\right) dt,$$

which describes the distribution of normals $\mathbf{n}$ visible from direction $\boldsymbol{\omega}$.

Appearance Spaces of GPISes

Surface-Type GPISes

Surface-Type GPISes reproduce existing microfacet model while reducing errors caused by approximations used in classical methods.



Appearance Spaces of GPISes

Volume-Type GPISes

The free-flight distribution is the central quantity in volumetric light transport. This can be also derived from the GPIS density $\Gamma(t, \mathbf{n} \,|\, \zeta)$

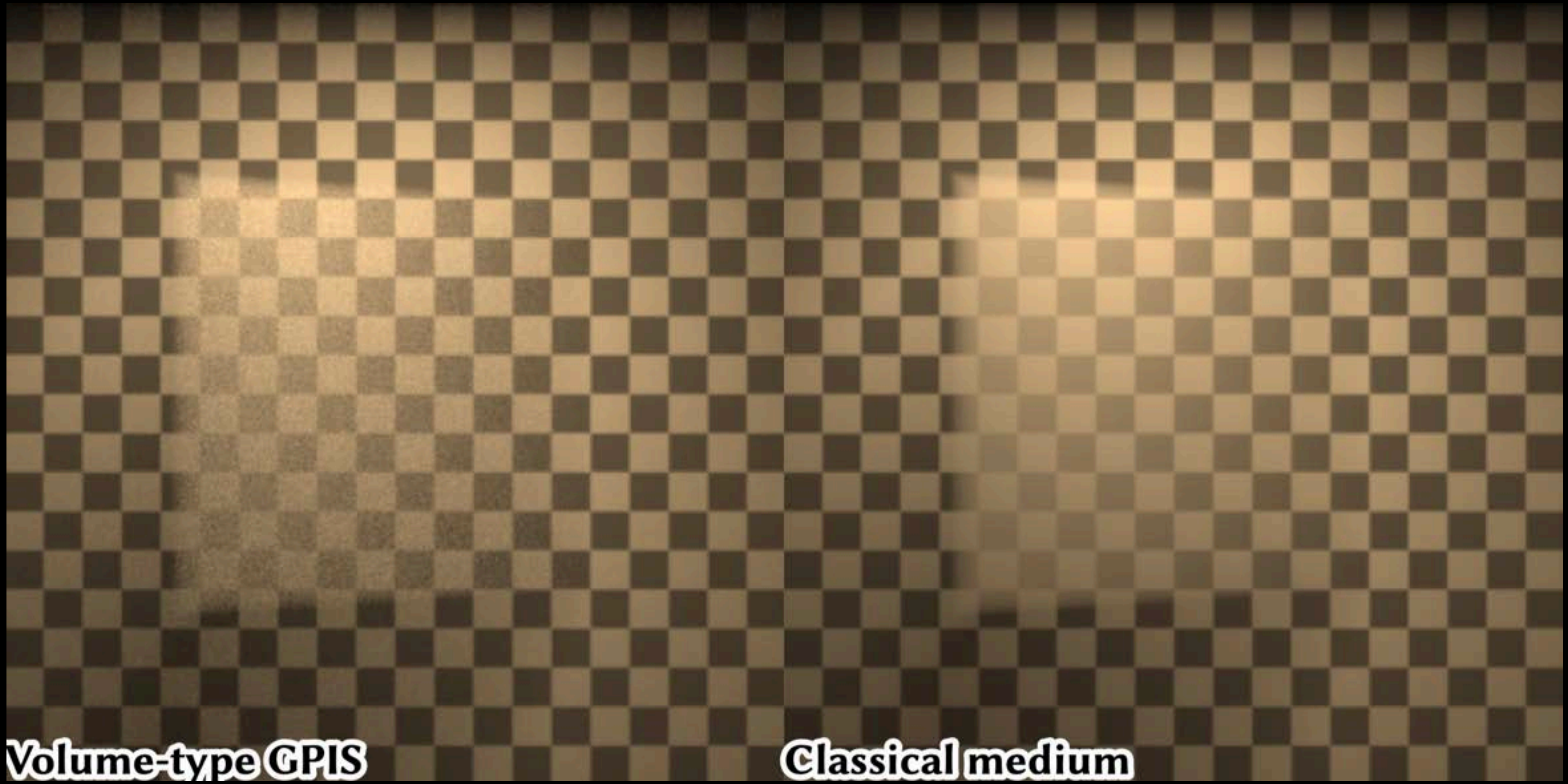
$$\Gamma(t \,|\, \zeta) = \int_{S^2} \Gamma(t, \mathbf{n} \,|\, \zeta) d\mathbf{n},$$

which is the probability density of finding the first zero crossing of the GPIS at distance t.

Appearance Spaces of GPISes

Volume-Type GPISes

Similarly to microfacets, GPISes can reproduce light transport through classical volumetric media by carefully adjusting their mean and covariance.



Non-Stationary GPIS Models

A Non-Stationary Kernel

Now, we consider **non-stationary GPISes** defined by

- A prior mean function and covariance kernel parameterized with Φ
- A set of conditioning points C with each point $\mathbf{c} \in C$ has
  - A location $\mathbf{c}_{\mathbf{x}}$
  - A value $\mathbf{c}_{v}$
  - A normal derivative direction $\mathbf{c}_{\nabla}$

Non-Stationary GPIS Models

A Non-Stationary Kernel

Following the definition of GPs introduced earlier, the mean and covariance of the GP are

$$\mu_{\Phi|C}(\mathbf{x}) = \mu_\Phi(\mathbf{x}) + k_\Phi(\mathbf{x}, C_\mathbf{x})k_\Phi(C_\mathbf{x}, C_\mathbf{x})^{-1}(C_v - \mu_\Phi(C_\mathbf{x})),$$

$$k_{\Phi|C}(\mathbf{x}, \mathbf{y}) = k_\Phi(\mathbf{x}, \mathbf{x}) - k_\Phi(\mathbf{x}, C_\mathbf{x})k_\Phi(C_\mathbf{x}, C_\mathbf{x})^{-1}k_\Phi(C_\mathbf{x}, \mathbf{y}).$$

Assuming a zero-mean function, a prior covariance kernel k_Φ is the only remaining attribute that determines the apperance of a GPIS.

Non-Stationary GPIS Models

A Non-Stationary Kernel

Local Variance

$$\sigma_\Phi : \mathbb{R}^3 \to \mathbb{R}$$

Local Anisotropy

$$\Sigma_\Phi : \mathbb{R}^3 \to \mathbf{S}^3_+$$

The authors employ the non-stationary covariance kernel from [Paciorek and Schervish, 2006]

$$k_\Phi^{NS}(\mathbf{x}, \mathbf{y}) = \sigma_\Phi(\mathbf{x})\sigma_\Phi(\mathbf{y}) \frac{|\Sigma_\Phi(\mathbf{x})|^{\frac{1}{4}} \, |\Sigma_\Phi(\mathbf{y})|^{\frac{1}{4}}}{\left|\frac{\Sigma_\Phi(\mathbf{x}) + \Sigma_\Phi(\mathbf{y})}{2}\right|^{-\frac{1}{2}}} k_\Phi^S\left(\sqrt{Q_\Phi(\mathbf{x}, \mathbf{y})}\right),$$

where

$$Q_\Phi(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \left(\frac{\Sigma_\Phi(\mathbf{x}) + \Sigma_\Phi(\mathbf{y})}{2} \right)^{-1} (\mathbf{x} - \mathbf{y})$$

Non-Stationary GPIS Models

A Non-Stationary Kernel

Need to remain PSD after interpolation!

The mean, variance, and anisotropy fields, $\mu_\Phi(\mathbf{x})$, $\sigma_\Phi(\mathbf{x})$, and $\Sigma_\Phi(\mathbf{x})$ are stored on a voxel grid and values are retrieved via interpolation.

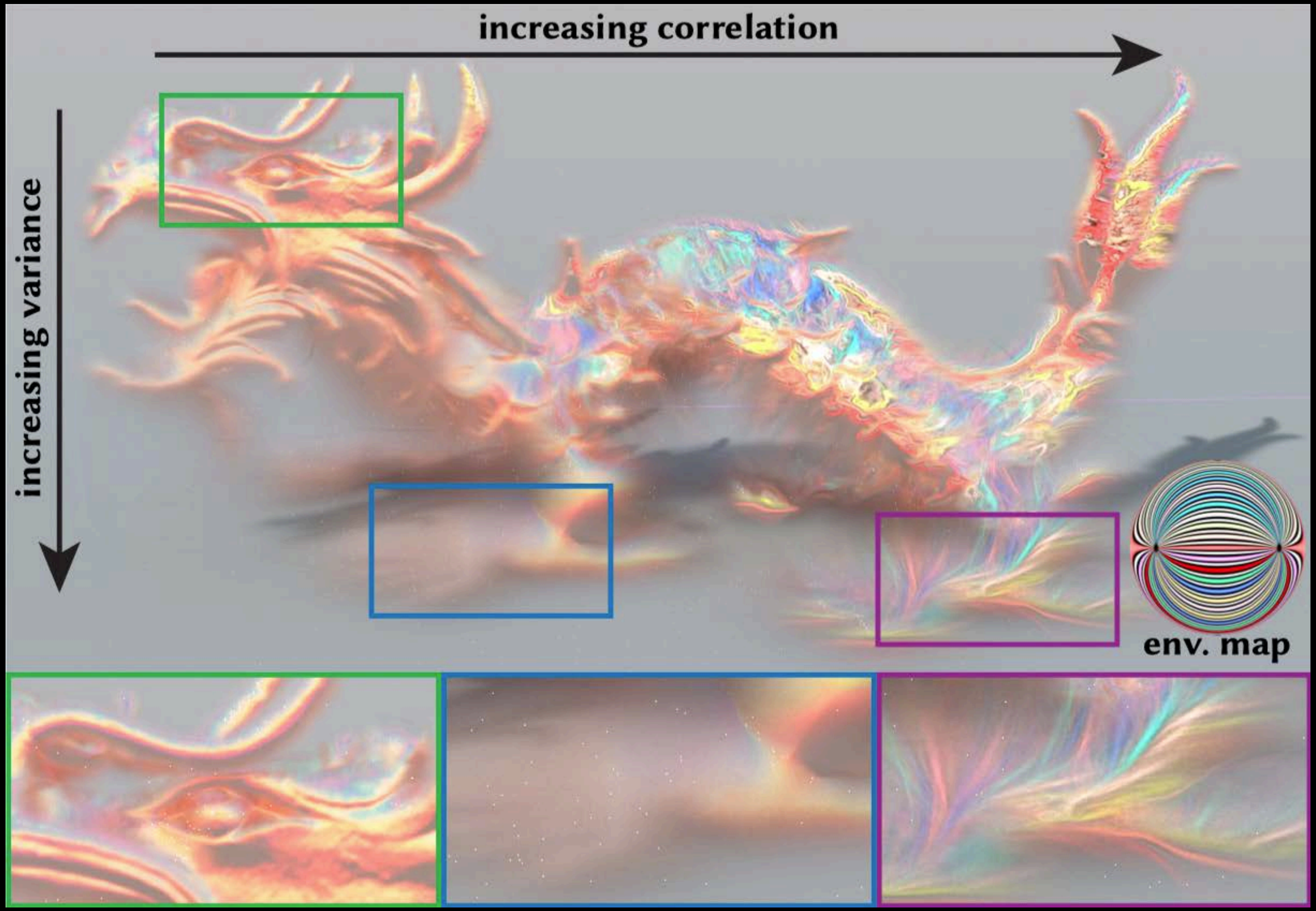
The SGGX Microflake Distribution, ACM ToG 2015

Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher

Non-Stationary GPIS Models

A Non-Stationary Kernel

Spatially varying kernels allow appearance change within a single object.

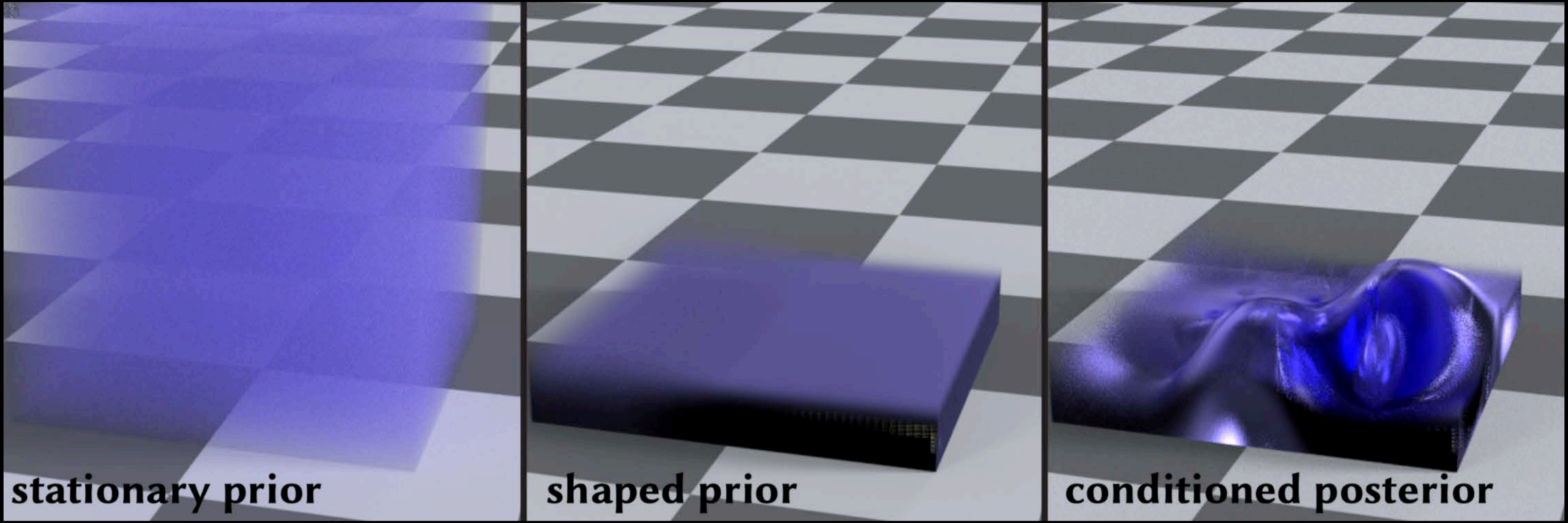


Applications of GPISes

Creating and Acquiring GPISes

Manual Annotation

Similarly to classical scene components (e.g., meshes), GPISes can be manually annotated by manipulating the mean and covariance over a volume.

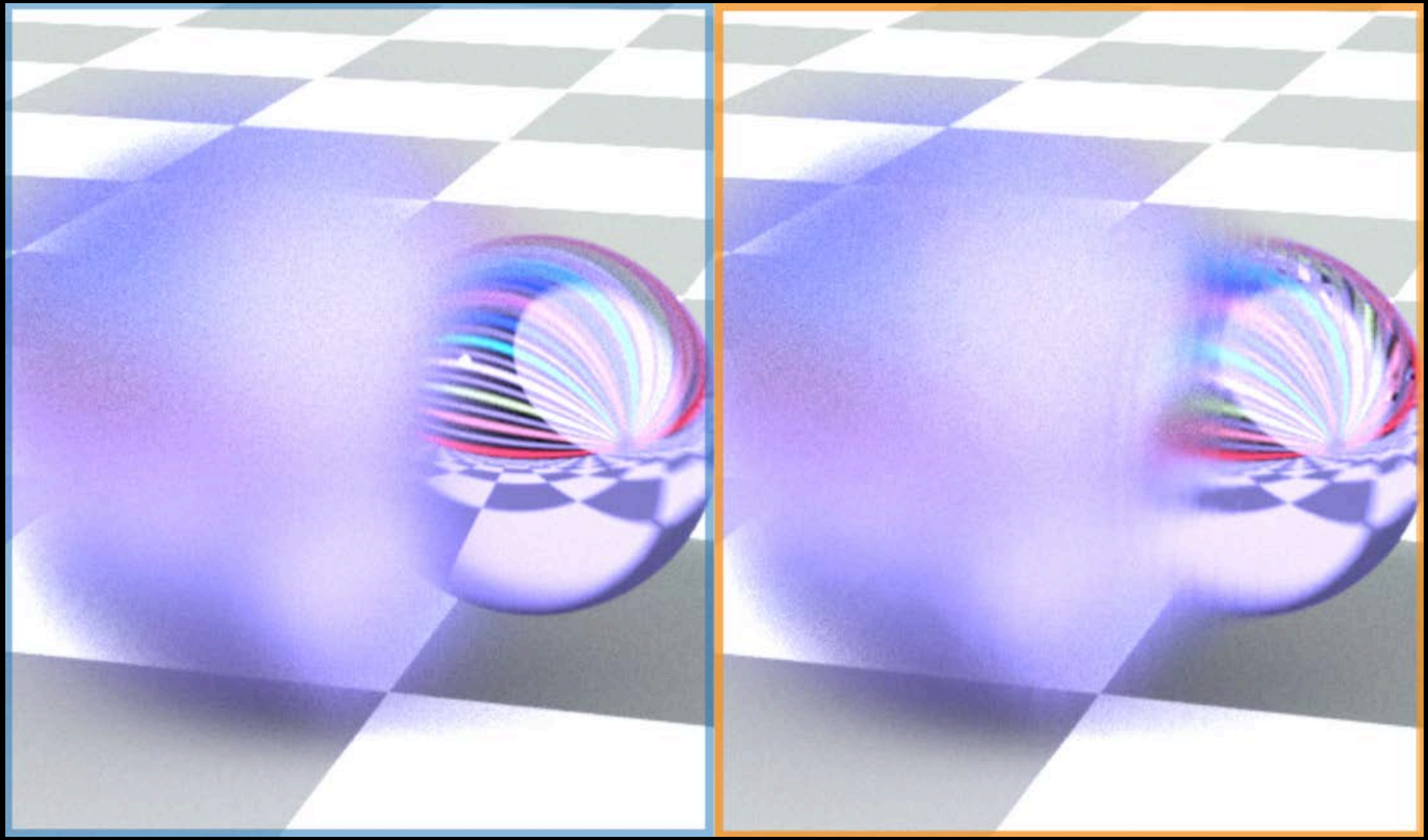


Creating and Acquiring GPISes

Constructive Solid Geometry (CSG)

GPISes also support several CSG operations for intuitive editing.

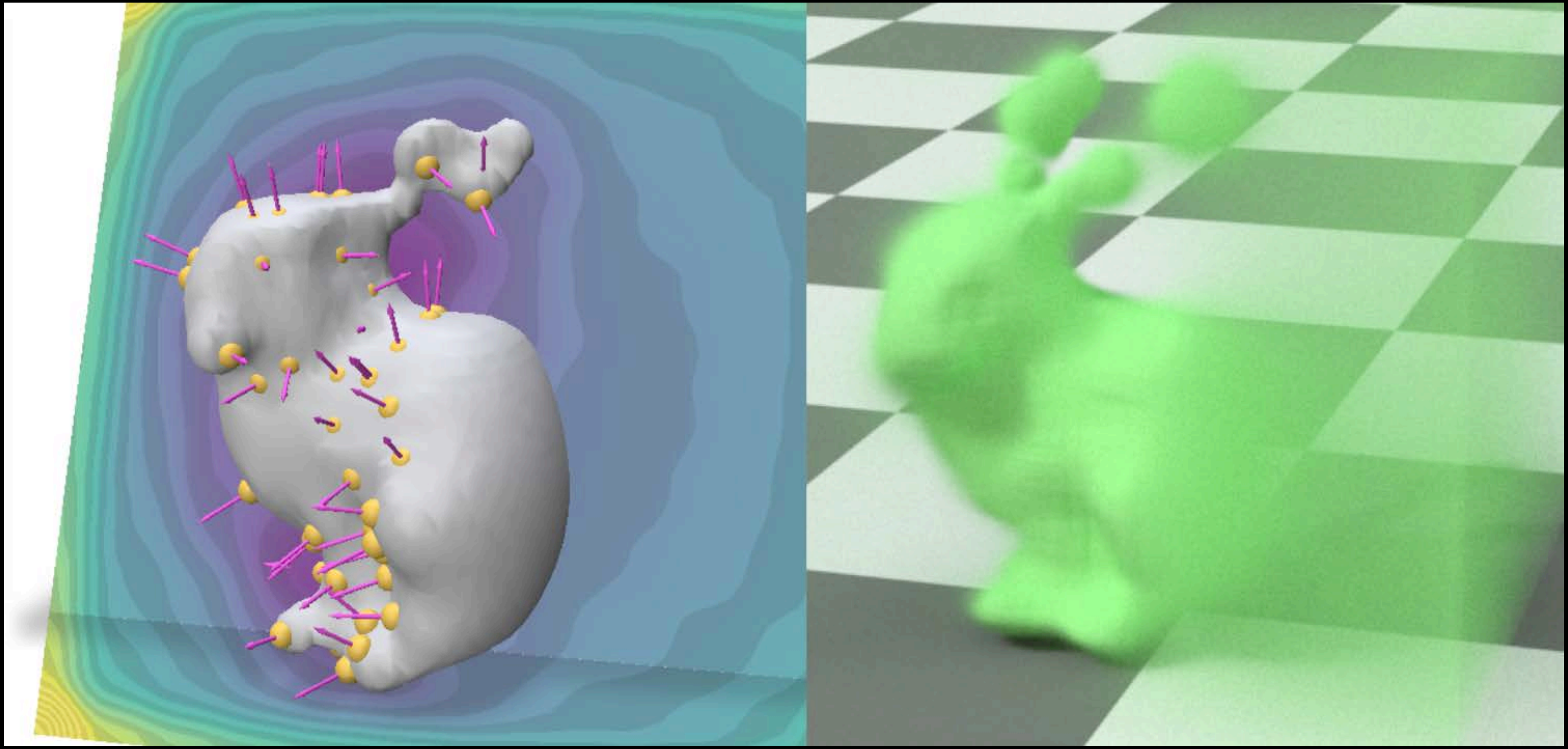
Sample & Compose



Fit to a GPIS

Stochastic Poisson Surface Reconstruction

The proposed rendering algorithm can be used to visualize stochastic implicit surfaces reconstructed via SPSR [Sellán and Jacobson, 2022].



Filtering Implicit Surfaces

An implicit function $f(\mathbf{x})$ can be fitted to a Gaussian process that is most likely to sample it to improve efficiacy.

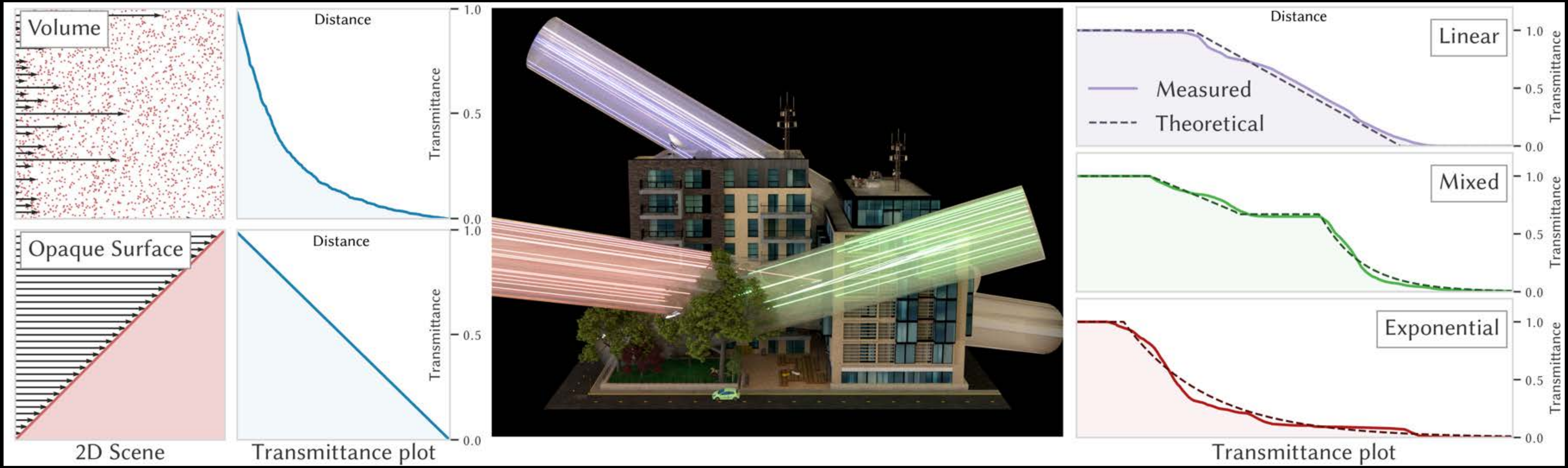
Input Function Residual Residual (Freq.) Reconstruction (Freq.)

Filtering Implicit Surfaces

Compaired to maximum likelihood estimation that only captures the mean function, GPISes capture high-frequency details by fitting covariances.

Filtering Implicit Surfaces

Alternative optimization objectives, such as appearance-based losses, can be used to recover GPIS parameters from images via differentiable rendering.



A Non-Exponential Transmittance Model for Volumetric Scene Representations, ACM ToG 2021

Delio Vicini, Wenzel Jakob, and Anton Kaplanyan

Discussion

Limitations

- Rendering implicit surfaces, including GPISes, are **typically slower** than rendering triangular meshes;
- Does not extend to **differentiable rendering** due to the lack of analytic expressions for transmittance and normal distributions;
- Texture mapping is non-trivial due to complex **parameterization**;
- **Choice of step sizes** along ray marching may impact rendering quality;

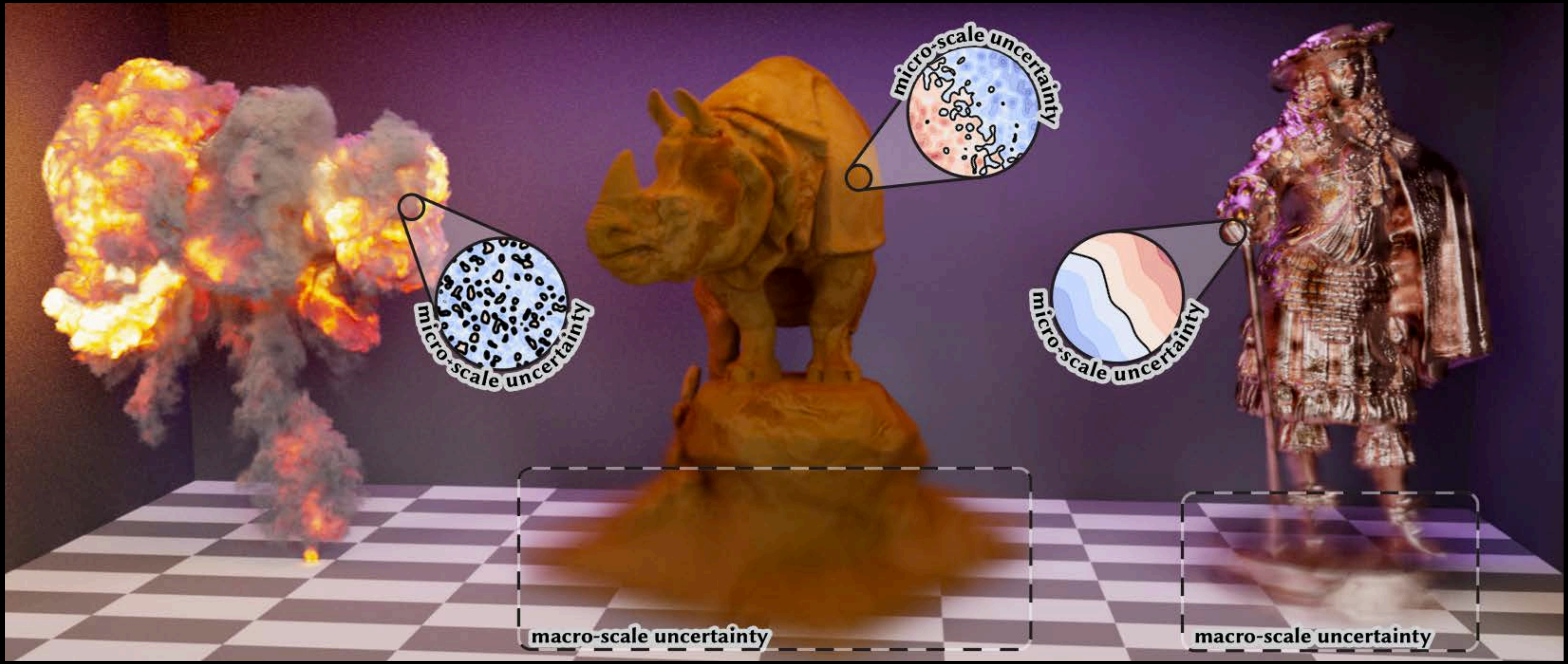
Conclusion

- Proposes **a novel Monte Carlo rendering algorithm** for stochastic implicit surfaces which is significantly faster than a naive implementation;
- Proposes an approximate, yet reasonable **memory model** to trade-off accuracy and performance;
- Demonstrates **the capability of GPISes** in representing widely used geometry types, as well as those existing models struggle to handle;
- Showcases **various applications** where the proposed algorithm can be potentially useful.

From Microfacets to Participating Media: A Unified Theory of Light Transport with Stochastic Geometry

Dario Seyb, Eugene D'eon, Benedikt Bitterli, Wojciech Jarosz

ACM ToG 2024 (Best Paper Award)



Seungwoo Yoo, KAIST Visual AI Group