

Thesis Proposal

Bailey Miller

September 2025

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Ioannis Gkioulekas
Keenan Crane
Nicholas Boffi
Ravi Ramamoorthi
Mathieu Desbrun

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Contents

1	Introduction	2
2	Background	4
2.1	Light transport	5
2.2	Discretization-based methods for light transport	5
2.3	Stochastic methods for light transport	6
2.4	Stochastic representations for microparticle geometry	8
3	Stochastic algorithms	10
3.1	Prior work: walk on spheres for the Laplace equation	10
3.2	General boundary conditions via walk on stars	12
3.2.1	Extension to Robin boundary conditions	14
3.3	Differential walk on spheres	16
3.4	Solving PDEs on stochastic geometry	17
4	Stochastic representations	21
4.1	Stochastic opaque solids	21
4.2	Scenes as free-flight distributions	23
5	Proposed work	26
5.1	Iterative hybrid solvers for more general PDEs	26
5.2	Stochastic fast winding numbers	27
5.3	Timeline	28

Abstract

Numerical computing on complex geometry faces two core challenges: representing geometry and performing computation on it. Discretization—voxels, meshes, global solves—remains effective until geometry becomes too detailed or uncertain to resolve. To overcome these limitations, this thesis will develop a complementary paradigm—stochastic geometry primitives (SGPs)—that use randomness to avoid discretization in both representation and computation.

SGPs build on two foundations already established in graphics. Monte Carlo rendering provides an algorithmic primitive that interacts with geometry only through local queries, while participating-media models serve as distributional representations that replace explicit particle interactions with free-flight sampling. Building on these principles, we generalize Monte Carlo partial differential equation (PDE) solvers to handle a wider range of boundary conditions, and we develop stochastic representations of solid geometry that naturally interface with rendering algorithms.

These methods are positioned as general-purpose primitives: black-box operators for physics simulation (elliptic and transport PDEs), geometric computation (harmonic coordinates, distance-driven queries, shape optimization), and machine learning (differentiable PDE layers or stochastic supervision of neural PDE surrogates). In this view, SGPs provide a common interface in place of meshes and global solves, allowing the same primitives to serve simulation, geometry processing, and learning.

To extend this framework further, we propose new work to address current limitations of SGPs. One direction is the treatment of nonlinear PDEs using hybrid neural–Monte Carlo methods, where iterative solves refine neural surrogates under Monte Carlo supervision. Another is a principled, point-based representation of stochastic solid geometry, which can be interpreted as a stochastic form of fast winding numbers. These contributions advance SGPs as general-purpose and efficient building blocks for geometric computing on complex and uncertain domains.

Chapter 1

Introduction

Numerical computing with complex geometry faces two core challenges: accurately representing geometry and efficiently performing computations on it. Traditional geometric computing has long relied on discretized domains, such as voxel grids or volumetric meshes, for tasks like reconstructing surface geometry [42] or solving partial differential equations (PDEs) [19]. While effective, these discretization-based approaches become computationally demanding or even fail when dealing with complex or uncertain geometries. To address these limitations, this thesis will introduce a new paradigm for geometric computing—*stochastic geometry primitives* (SGPs)—which leverages randomness to avoid discretization in both geometric representation and computation.

SGPs serve as building blocks for modeling and simulation that encompass two complementary dimensions.

- **Stochastic algorithms for geometry** recast geometric and physical quantities as (recursive) integral expressions, which are estimated via Monte Carlo sampling. Unlike discretization-based approaches such as radiosity or the finite element method (FEM), stochastic algorithms do not require accounting for all global relationships simultaneously through a mesh or system solve. Instead, they interact with geometry only through local queries (e.g. ray intersections, closest points), forming high-dimensional samples that stochastically encode global relationships.
- **Stochastic representations of geometry** describe geometry in a distributional sense rather than by resolving it explicitly through a mesh or grid. This provides a principled way to handle both extreme geometric complexity—beyond what can be feasibly resolved—and uncertainty arising from limited measurements or incomplete knowledge. Alongside these representations, we develop stochastic analogues of standard geometric queries (e.g. ray intersections, closest points), extending the query-based interface for geometry to the distributional setting.

Besides being important on their own for different graphics or engineering pipelines, these two components can also be combined to achieve algorithms capable of unprecedented scale and efficiency.

Much of the inspiration for these primitives comes from the success of stochastic algorithms and representations in computer graphics. Specifically, Monte Carlo rendering algorithms [40, 84]

demonstrate how stochasticity can enable computation on massive, detailed scenes by limiting geometric interactions to local ray intersection queries with logarithmic complexity. Graphics has also incorporated stochasticity to geometric representations in the form of volumetric models [41, 43, 65] for such smoke, clouds, and tissue which replace explicit particle-level detail with a distributional description. These stochastic models of geometry have made it possible to simulate light transport in settings that would otherwise be computationally intractable, and have even extended their influence beyond graphics to fields such as remote sensing [9, 71] and lidar-based imaging [36, 66]. Despite this success, existing stochastic primitives remain narrowly applicable—algorithmically to light transport and representationally to particle-based models. This thesis aims to extend the power of stochasticity beyond its traditional role in graphics and rendering, developing SGPs as a unified framework for geometric computing.

Our prior work has already begun to lay the foundation for more general purpose SGPs. On the algorithmic side, our work generalizes Monte Carlo PDE solvers to a broader range of boundary conditions [75, 55] and develops differential variants [54] for tasks like shape reconstruction and optimization-driven design. Complementing this, our work also introduces stochastic models of solid geometry [53, 6] that move beyond microparticle formulations to capture uncertainty and complexity in real world scenes. Our work even combines these primitives; applying inverse rendering to stochastic opaque solids for surface reconstruction [53] and extending Monte Carlo PDE solvers to stochastic microparticle geometry [56]. These applications are natural extensions of our work, made possible by the stochastic queries (e.g., ray intersections, closest points) developed alongside our representations, which allow them to serve as drop-in replacements for explicit geometry within existing algorithms.

Building on these foundations, we propose two directions to further generalize SGPs:

- **Algorithmic generalization.** Current stochastic PDE solvers primarily support linear elliptic PDEs or transport problems. We propose extending them to more challenging regimes—nonlinear and hyperbolic PDEs—by developing hybrid neural–Monte Carlo techniques.
- **Representational generalization.** Current point cloud representations of stochastic solid geometry simplify evaluation by assuming pointwise independence. We propose a more expressive class of point-cloud implicit surface models, which account for spatial correlations in queries. These models lead to more expressive stochastic solid geometry, while preserving query efficiency.

We begin by reviewing SGPs as they appear in graphics today (Chapter 2), using light transport as a model problem in geometric computing. This review covers both algorithmic aspects (e.g., Monte Carlo rendering section 2.3) and representational aspects (e.g., participating media section 2.4). We then summarize our own contributions, which lie in advancing Monte Carlo PDE solving algorithms (Chapter 3) and in simultaneously developing stochastic geometric representations that move beyond microparticle models (Chapter 4). Finally, we outline our proposed directions (Chapter 5) for jointly extending algorithms and representations, and conclude with a roadmap (section 5.3) for completing the thesis.

Chapter 2

Background

This chapter surveys the foundations of geometric computing for light transport, which serve as the starting point for SGPs. Over the past two decades, Monte Carlo rendering algorithms have transformed computer graphics by enabling efficient, robust simulation of light transport on massive, extremely detailed scene geometry. In parallel, stochastic representations of geometry have expanded light transport simulation to include simulation with participating media such as clouds and biological tissue. These advances have driven a Monte Carlo renaissance in rendering and visualization, with widespread adoption in applications ranging from visual effects to scientific computing.

Our aim here is not to review light transport for its own sake, but to use it as a case study in how stochastic methods overcome the limits of classical approaches. We organize the discussion around three perspectives:

- **Discretization-based methods.** The radiosity method, introduced by Goral et al. [24], illustrates how the light transport problem can be discretized into a system of equations. This approach highlights both the appeal of discretization and its limitations when facing geometric complexity.
- **Monte Carlo rendering.** Rendering and related algorithms, pioneered by Kajiya [40] and Veach [84], overcome these limitations by replacing discretization with stochastic sampling. These methods scale naturally with complexity and provide the key algorithmic insights that we later generalize in our Monte Carlo PDE framework.
- **Participating Media.** In cases where even explicit geometry becomes intractable, distributional models provide a natural means of abstraction. Such representations make it possible to apply Monte Carlo algorithms to microparticle systems, like clouds or tissue [43, 65] and introduce a distributional view of geometry that we'll later extend beyond microparticles to more general geometric structures such as solids.

These three perspectives—discretization, Monte Carlo rendering, and participating media—anchor our background discussion. They also foreshadow the more general computational framework developed in later chapters, where we extend these ideas beyond rendering (chapter 3) and participating media (chapter 4).

2.1 Light transport

Light transport is governed by a first-order linear PDE that captures how radiance propagates in free space and interacts with surface geometry. The central quantity, radiance $L(x, \vec{\omega})$, measures radiant flux per unit projected area per unit solid angle (refer to Chapter 4 of Pharr et al. [67]). In a spatial domain $\Omega \subset \mathbb{R}^3$, radiance satisfies the following PDE,

$$\begin{aligned} \vec{\omega} \cdot \nabla L(x, \vec{\omega}) &= 0, & x &\in \Omega, \\ L(x, \vec{\omega}) &= L_e(x, \vec{\omega}) + L_s(x, \vec{\omega}), & x &\in \partial\Omega, \vec{n}(x) \cdot \vec{\omega} > 0. \end{aligned} \quad (2.1)$$

where L_e is emitted radiance, L_s is the in-scattered radiance, and $\vec{n}(x)$ is the outward surface normal. The in-scattered term is

$$L_s(x, \vec{\omega}) = \int_{S_+^2} f(x, \vec{\omega}, \vec{\omega}') (\vec{n}(x) \cdot \vec{\omega}') L(x, -\vec{\omega}') d\vec{\omega}' \quad (2.2)$$

where f is the bidirectional scattering distribution function (BSDF). This term couples outgoing radiance to incident radiance at a boundary point. A convenient way to represent incident radiance on the boundary is with an integral equation,

$$L(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_{\partial\Omega} f(x, \vec{\omega}, \vec{x}y) G(x, y) V(x, y) L(y, \vec{y}x) dy, \quad (2.3)$$

Here $\vec{x}y$ is the unit vector from x to y , $V(x, y)$ is a binary visibility function with respect to the geometry $\partial\Omega$, and $G(x, y)$ is a geometry term,

$$G(x, y) = \frac{(\vec{n}(x) \cdot \vec{x}y)(\vec{n}(y) \cdot \vec{y}x)}{\|y - x\|^2}. \quad (2.4)$$

Equation (2.3) is radiosity [24], since it directly couples outgoing radiance to other boundary points. Inside the domain and on the boundary, incident radiance can be expressed recursively as

$$L(x, \vec{\omega}) = L\left(r_{x, \vec{\omega}}^*, -\vec{\omega}\right), \quad (2.5)$$

where $r_{x, \vec{\omega}}^*$ is the first intersection point along direction $\vec{\omega}$. The integral form highlights the core computational difficulty of light transport: every surface point is coupled to every other surface point. Both discretization-based algorithms such as radiosity [24] and stochastic algorithms such as rendering [40] can be viewed as strategies for resolving these global couplings to obtain a solution to the PDE (2.1).

2.2 Discretization-based methods for light transport

The integral formulation of radiance (2.3) reveals that radiance at one surface point depends on contributions from all other surface points, modulated by geometry and visibility terms. Radiosity [24], the canonical discretization-based approach, attempts to explicitly encode these

spatial relationships to compute the radiance at every point on the boundary. It makes two simplifying assumptions.

First, all surfaces are assumed to be perfectly diffuse reflectors and emitters. Under this assumption, the outgoing radiance is direction-independent:

$$L(x, \vec{\omega}) = L(x), \quad \vec{\omega} \in \mathcal{S}_+^2. \quad (2.6)$$

Second, the boundary is discretized into a *solution mesh* of N non-overlapping elements $\partial\Omega \approx \bigcup_{i=1}^N e_i$ with $e_i \cap e_j = \emptyset$ for $i \neq j$, and radiance is approximated by a constant L_i on each element. Under these assumptions, the radiance integral (2.3) reduces to the surface balance equations,

$$L_i = L_{ei} + \sum_{j=1}^N F_{ij} L_j. \quad (2.7)$$

Here F_{ij} are known as the *form factors* which encode the spatial relationships—geometric coupling and visibility—between boundary elements (refer to Goral et al. [24]). In matrix form, this is

$$L = L_e + FL \implies L = (I - F)^{-1} L_e \quad (2.8)$$

where $L \in \mathbb{R}^N$ and $F \in \mathbb{R}^{N \times N}$. The radiosity system (2.7) appears deceptively simple: it collapses the light transport PDE into a linear system that can be directly solved. In practice, however, its accuracy depends on the choice of discretization. Each boundary element in the solution mesh must have nearly constant visibility with respect to all others—a requirement rarely met in complex scenes since the binary visibility function $V(x, y)$ introduces sharp discontinuities. Adaptive refinement around these discontinuities can reduce error [46, 26, 14], but detecting and meshing visibility discontinuities is itself a globally defined and difficult problem.

These challenges are not unique to radiosity, but characteristic of discretization-based methods more broadly: one must either uniformly refine the solution mesh, which quickly runs into computational limits, or attempt adaptive refinement, which can be as difficult as solving the original PDE.

2.3 Stochastic methods for light transport

The previous section shows how discretization encodes global coupling of light transport explicitly through form-factor matrices. This approach is naturally tied to the boundary integral formulation of radiance (2.3), which expresses outgoing radiance at one surface point as an integral over all other boundary points and is the foundation of radiosity [24].

Stochastic methods take a different route. Monte Carlo rendering [40, 84] can be formulated in terms of boundary area integrals, but it is often easier to understand through the hemispherical integral (2.2), which sits closer to the PDE form (2.1). From this perspective, constructing an estimator is straightforward: we sample a direction, trace a ray, and evaluate radiance recursively at the first surface hit. In what follows we develop this estimator and discuss some of its advantages.

By starting from the hemispherical integral (2.2), we can express the outgoing radiance at a boundary point $x \in \partial\Omega$ in terms of incident radiance from all directions,

$$L(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_{S_+^2(x)} f(x, \vec{\omega}, \vec{\omega}') (\vec{n}(x) \cdot \vec{\omega}') L(x, -\vec{\omega}') d\vec{\omega}'. \quad (2.9)$$

The key idea of rendering is to directly sample this integral with a single-sample Monte Carlo estimator,

$$\langle L(x, \vec{\omega}) \rangle = L_e(x, \vec{\omega}) + 2\pi f(x, \vec{\omega}, \vec{\omega}') (\vec{n}(x) \cdot \vec{\omega}') L(x, -\vec{\omega}'), \quad \vec{\omega}' \sim \mathcal{U}(S_+^2(x)). \quad (2.10)$$

where $\vec{\omega}'$ is sampled uniformly over the hemisphere and $\langle \cdot \rangle$ denotes an unbiased Monte Carlo estimator. The only unknown in this estimator is the incident radiance $L(x, -\vec{\omega}')$ from direction $\vec{\omega}'$, which we evaluate recursively by finding the first intersection $r_{x, \vec{\omega}'}^*$ along a ray $r_{x, \vec{\omega}'}(t) \equiv x + t\vec{\omega}'$. Substituting this relation makes the estimator recursive,

$$\langle L(x, \vec{\omega}) \rangle = L_e(x, \vec{\omega}) + 2\pi f(x, \vec{\omega}, \vec{\omega}') (\vec{n}(x) \cdot \vec{\omega}') \left\langle L(r_{x, \vec{\omega}'}^*, -\vec{\omega}') \right\rangle, \quad \vec{\omega}' \sim \mathcal{U}(S_+^2(x)). \quad (2.11)$$

To ensure that this estimator converges, we additionally apply Russian roulette which probabilistically terminates recursion while reweighing to avoid bias. At each step, we evaluate the recursive term with probability q

$$\langle L(x, \vec{\omega}) \rangle = L_e(x, \vec{\omega}) + f(x, \vec{\omega}, \vec{\omega}') (\vec{n}(x) \cdot \vec{\omega}') \begin{cases} \frac{1}{q} \left\langle L(r_{x, \vec{\omega}'}^*, -\vec{\omega}') \right\rangle & \text{if } \xi < q \\ 0 & \text{otherwise} \end{cases} \quad \xi \sim \mathcal{U}[0, 1] \quad (2.12)$$

Convergence of this estimator is guaranteed for physically plausible BSDFs, just as in radiosity where this same condition ensures the inverse $(I - F)^{-1}$ exists.

The resulting recursive estimator (2.12) provides an algorithm for computing radiance anywhere in the scene. For boundary points $x \in \partial\Omega$ it is applied directly, while for interior points $x \in \Omega$ we evaluate by first tracing a ray $r_{x, \vec{\omega}}$ to the corresponding boundary point $r_{x, \vec{\omega}}^*$

$$\langle L(x, \vec{\omega}) \rangle = \left\langle L(r_{x, \vec{\omega}}^*, -\vec{\omega}) \right\rangle. \quad (2.13)$$

where we can recursively evaluating the outgoing radiance with equation (2.12). In both cases, the estimator will be noisy and so multiple samples are averaged to reduce variance.

Notice that the rendering algorithm interacts with geometry only through ray-intersection queries used for propagating to the next step in the recursive evaluation. Unlike constructing a form-factor matrix, these queries are local, efficient with $O(\log n)$ complexity with respect to the number of elements n , and scalable when supported by acceleration structures [64, 87]. This minimal interface also enables efficient handling of implicitly defined surfaces, where prior work has already developed fast ray-intersection methods [25, 20, 77].

In addition to avoiding mesh construction, rendering is also output-sensitive: radiance is computed only at points of interest, such as image pixels. This structural characteristic of

rendering also means separate evaluation points can be estimated independently, which enables trivial parallelization that maps well to modern hardware like GPUs [68, 35, 67].

The cost of these benefits is variance: rendering produces noisy estimates. This variance can be reduced by averaging more samples or leveraging specialized strategies, such as importance sampling [85, 86], caching [89, 38], and denoising [15, 62]. In practice, these techniques shift the variance–complexity tradeoff making variance far more manageable than the complexity of adaptive meshing.

These insights point to a broader principle: stochastic algorithms can avoid encoding global spatial relationships explicitly by interacting with geometry only through local queries. In the next section we turn to the other axis of SGPs—stochastic representations—which extend this framework to structures like smoke or sand, where explicit modeling or full simulation is intractable.

2.4 Stochastic representations for microparticle geometry

Stochastic algorithms like rendering can scale to geometry consisting of billions of elements, but some geometric structures still remain computationally intractable. Specifically, media like clouds, smoke, sand, and tissue can consist of quadrillions, or orders of magnitude more, microscopic particles. For these systems, explicit geometry not only exceeds most practical memory limitations, but also makes geometric computation prohibitively expensive. Any attempt at meshing is also futile; the mesh discretization would require more elements than the particles themselves, which already exceed feasible storage.

A useful way to see the difficulty is to imagine the explicit surface

$$O = \bigcup_{i=1}^N B(c_i, R), \quad \partial O = \partial(\mathbb{R}^3 \setminus O), \quad (2.14)$$

where $B(c_i, R)$ is a ball of radius R centered at c_i . Materializing this union is hopeless when N is massive: the only way forward is to model aggregate behavior. In computer graphics, stochastic microparticle models—known as *participating media*—capture the aggregate behavior of particle ensembles such as clouds or smoke. A canonical example is the *Poisson–Boolean model* (PBM), which describes the spatial distribution of particles.

Under a standard PBM, particles are independent, identically shaped, and isotropic, with centers distributed according to a (possibly spatially varying) rate parameter $\lambda : \mathbb{R}^3 \rightarrow \mathbb{R}_+$

$$O \sim \text{PBM}(\lambda) \quad (2.15)$$

Inside any measurable subset $S \subset \mathbb{R}^3$ and conditioned on the number of particles $N(S)$, the centers are i.i.d. proportional to the rate λ ,

$$\text{(number of microparticles)} \quad N(S) \sim \text{Poisson}(\Lambda(S)), \quad (2.16)$$

$$\text{(position of microparticles)} \quad \Pr[c_i = x \mid N(S), c_i \in S] = \frac{\lambda(x)}{\Lambda(S)},$$

$$\Lambda(S) = \int_S \lambda(x) \, dx.$$

Since $\partial\Omega$ is no longer explicitly constructed, geometric queries are resolved stochastically according to the underlying stochastic model. For ray–intersection queries, this means sampling from the *free-flight distribution*, i.e., the distribution of the first intersection distance along a ray. In the case of the PBM, the free-flight distribution is exponential [44]

$$p_{x,\vec{\omega}}^{\text{ff}}(t) = \sigma(r_{x,\vec{\omega}}(t)) \exp\left(-\int_0^t \sigma(r_{x,\vec{\omega}}(s)) \, ds\right). \quad (2.17)$$

Following computer graphics convention, the product of the rate parameter $\lambda(x)$ and particle radius R represents the extinction coefficient $\sigma(x) \equiv \lambda(x) \pi R^2$.

The expected radiance under the PBM is obtained by averaging over the distance to the first intersection, yielding a conservation law along rays that is analogous to the deterministic case,

$$\text{(deterministic)} \quad L(x, -\vec{\omega}) = L(r_{x,\vec{\omega}}^*, -\vec{\omega}), \quad (2.18)$$

$$\text{(stochastic)} \quad \mathbb{E}_O[L(x, -\vec{\omega})] = \int_0^\infty p_{x,\vec{\omega}}^{\text{ff}}(t) \mathbb{E}_O[L(r_{x,\vec{\omega}}(t), -\vec{\omega}) \mid t] \, dt, \quad (2.19)$$

This relationship can be substituted for the recursive radiance term in equation (2.12) and (2.13) to define an estimator for expected radiance over stochastic geometry. For rendering, the conditioning can be ignored without introducing much bias. The resulting recursive sampling procedure is similar to the deterministic case: sample a direction, draw a free-flight distance, and evaluate outgoing radiance at the intersection.

This formulation makes clear that stochastic geometry and stochastic algorithms are natural complements. Because such algorithms depend only on local queries, extending them to stochastic geometry amounts to replacing deterministic outputs with sampled ones. This simple substitution is powerful: it is precisely what enables volume rendering in participating media such as clouds, where explicitly representing microparticle geometry is intractable.

Within volume rendering, researchers have already explored variations in microparticle distributions [4, 37, 17] and structures [34, 27], but two promising opportunities have remained largely overlooked. The first is applying stochastic geometry within alternative algorithms beyond rendering. The second is developing new stochastic geometry models beyond microparticles—such as stochastic solids that can capture uncertainty for surface reconstruction. We develop both of these directions in later chapters.

Chapter 3

Stochastic algorithms

So far we have seen that rendering algorithms are crucial to solving the light transport PDE (2.1) on complex or even stochastic geometry. As rendering algorithms have matured, they have enabled light transport simulation to handle geometric complexity at scales far beyond what is practical in most other areas of geometric computing [10, 18]. This naturally raises the question: why can’t analogous techniques be developed for more general PDEs?

PDEs form one of the broadest classes of problems in geometric computing. They underlie almost every physical simulation and many geometric tasks without a physical basis. PDEs directly model implicit surfaces [42], enable blending weights for object deformations [31], drive mesh operations such as smoothing [13], and support artistic applications like inflating 2D shapes [39]. In some cases, PDEs act as geometry-aware convolution operators in learning architectures [5, 78]. Given how many applications rest upon existing PDE solvers, developing general Monte Carlo methods for solving PDEs may enable many geometric computing applications to achieve the geometric flexibility and robustness that discretization-based methods like the finite element methods (FEM) often lack. While Monte Carlo solvers will likely never fully supplant these discretization-based methods—and nor should they when geometry is modest enough—the gap between what rendering achieves and what discretization methods currently handle motivates the search for stochastic counterparts.

In this chapter we build on this motivation. We begin by reviewing walk on spheres [57], a Monte Carlo PDE solver from the 1950s that was only recently introduced to computer graphics [73]. This method illustrates how Monte Carlo approaches extend beyond light transport to more general classes of PDEs. Our core contribution is to generalize walk on spheres to Neumann [75] and Robin [55] boundary conditions, enabling a wider range of modeling and simulation problems. We also develop extensions to differentiable solvers [54] and to stochastic geometry [56], such as participating media. These advances position Monte Carlo PDE solvers as SGPs—simple, modular, and scalable building blocks for geometric computing.

3.1 Prior work: walk on spheres for the Laplace equation

The Laplace equation is one of the most fundamental PDEs in both geometry and physics. Throughout this chapter, we focus on the case without screening coefficients or source terms,

[Sawhney and Crane 2020]

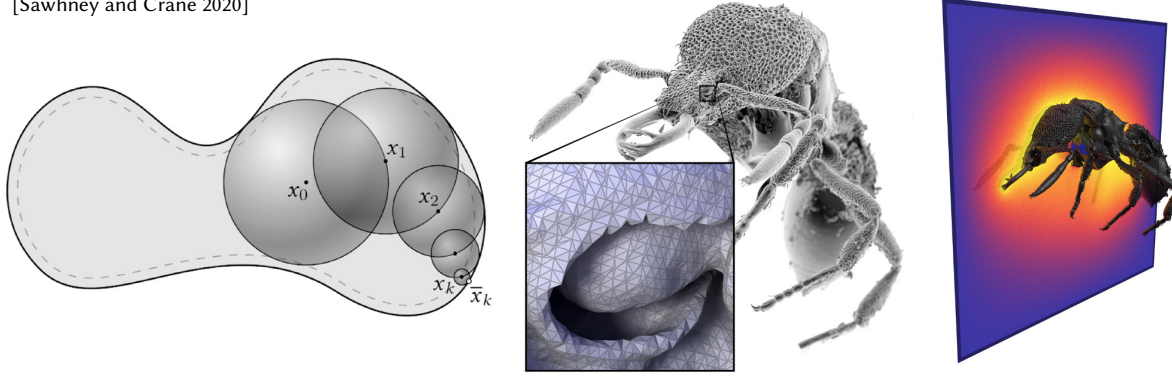


Figure 3.1: Figures from Sawhney and Crane [73]. (Left) Walk on spheres samples the next point on the largest empty sphere of the current point until reaching the boundary. (Right) A Poisson equation is solved on a complex CT scan of an insect using walk on spheres.

though the methods we discuss naturally extend to these variants. With Dirichlet boundary conditions, the Laplace equation can be interpreted as smoothly interpolating g from the boundary $\partial\Omega$ to the interior of the domain Ω :

$$\begin{aligned} \Delta u(x) &= 0 & x &\in \Omega \\ u(x) &= g(x) & x &\in \partial\Omega, \end{aligned} \tag{3.1}$$

where $\Omega \subset \mathbb{R}^d$ is the domain, $\partial\Omega$ its boundary, u the unknown solution, and g the boundary data. The Laplace operator is defined as

$$\Delta u(x) := \sum_{i=1}^d \frac{\partial^2 u(x)}{\partial x_i^2}, \tag{3.2}$$

which intuitively measures the deviation of u at x from the average of its neighbors. This relatively simple PDE underlies a broad range of applications from thermal conduction to harmonic coordinates.

In the 1950s, Muller [57] introduced the *walk on spheres* algorithm as a Monte Carlo solver for the Laplace problem. The key idea is that harmonic functions—solutions u satisfying $\Delta u = 0$ —admit a mean value property:

$$u(x) = \frac{1}{|\partial B(x, r_x)|} \int_{\partial B(x, r_x)} u(y) \, dy, \tag{3.3}$$

where $r_x := \min_{q \in \partial\Omega} \|q - x\|$ is the distance to the nearest boundary point. This representation holds whenever u is harmonic throughout the ball $B(x, r_x)$. Walk on spheres directly estimates this mean value integral using a single-sample Monte Carlo estimator:

$$\langle u(x) \rangle = \langle u(y) \rangle, \quad y \sim \mathcal{U}[\partial B(x, r_x)]. \tag{3.4}$$

The recursion terminates once a walk enters an ϵ -neighborhood of the boundary, where the solution u is approximated with the Dirichlet boundary condition g at the closest boundary point $\pi(x) := \operatorname{argmin}_{q \in \partial\Omega} \|q - x\|$. The final estimator for u accounts for this ϵ -shell termination

$$\langle u(x) \rangle = \begin{cases} g(\pi(x)) & \text{if } r_x < \epsilon \\ \langle u(y) \rangle & \text{otherwise} \end{cases}, \quad y \sim \mathcal{U}[\partial B(x, r_x)]. \quad (3.5)$$

Walk on spheres interacts with geometry only through closest-point queries, which are local and highly scalable [72]. As a result, the method scales gracefully to complex geometry, much like rendering algorithms.

Recently, Sawhney and Crane [73] introduced this method to computer graphics, motivating follow-up work that generalizes walk on spheres to a broader range of governing equations [74, 70]. These works highlight that walk on spheres inherits many of the same benefits as rendering algorithms, including geometric scalability, flexibility to different geometric representations, output sensitivity, and trivial parallelization.

With today's ever-increasing geometric complexity, these strengths make walk on spheres especially appealing. At the same time, several obstacles still limit its practical use:

1. walk on spheres is restricted to Dirichlet boundary conditions,
2. extending to more general PDEs (e.g. nonlinear) remains challenging, and
3. its role as a subroutine is underdeveloped compared to FEM solvers.

The contributions of this thesis directly address these limitations. We generalize walk on spheres to Neumann [75] and Robin [55] boundary conditions, broadening its applicability. We also propose hybrid neural-Monte Carlo methods that use the algorithm as a subroutine, which in turn open a pathway toward tackling more general PDEs. Finally, we explore extensions to differential [54] and volumetric solvers [56], moving walk on spheres closer to feature parity with rendering algorithms and positioning it as a practical, modular algorithmic primitive.

3.2 General boundary conditions via walk on stars

Walk on spheres only supports Dirichlet boundary conditions, but most real systems require more general boundary conditions. While Dirichlet specifies solution values, Neumann conditions constrain normal derivatives—for instance, an insulated wall enforces zero heat flux, or an impermeable boundary enforces normal velocity. Many problems combine both types, with parts of the boundary prescribing values and others prescribing derivatives. Extending walk on spheres to such mixed conditions is therefore essential. Our walk on stars method [75, 55] addresses this need as a direct generalization of walk on spheres to Neumann and Robin boundary conditions, while preserving the performance characteristics of the original method.

We focus here on the homogeneous case, though the walk on stars method naturally extends to non-homogeneous variants including non-zero Neumann boundary conditions. For the

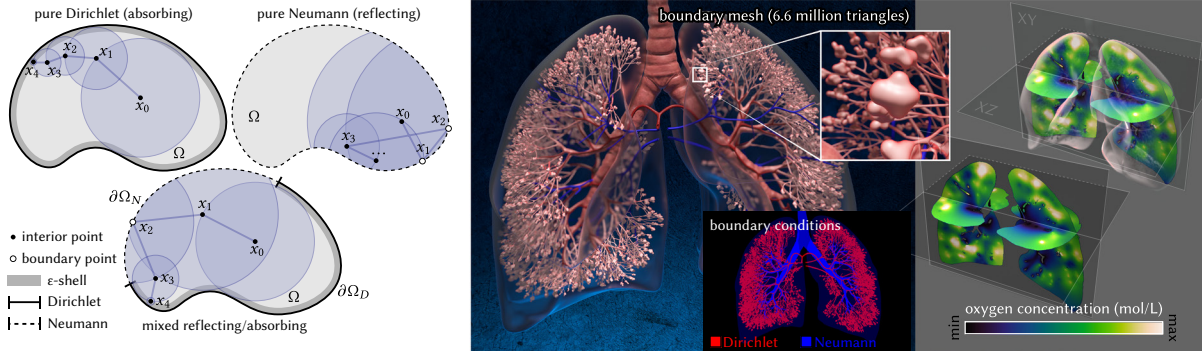


Figure 3.2: (Left) Walk on stars performs a random walk on star-shaped subdomains, reflecting off Neumann boundaries and terminating at the Dirichlet boundary. (Right) A Laplace PDE with mixed Dirichlet-Neumann boundary conditions modeling oxygen diffusion is solved on a model of the lungs using walk on stars.

Laplace equation with mixed Dirichlet-Neumann boundary conditions, a typical setup is

$$\Delta u(x) = 0 \quad x \in \Omega \quad (3.6)$$

$$u(x) = g(x) \quad x \in \partial\Omega_D \quad (3.7)$$

$$\frac{\partial u}{\partial \vec{n}}(x) = 0 \quad x \in \partial\Omega_N, \quad (3.8)$$

where the boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ is partitioned into Dirichlet and Neumann components. Dirichlet conditions prescribe solution values, while Neumann conditions prescribe normal derivatives.

To extend beyond pure Dirichlet boundaries, we use a more general mean value relationship [75]. Instead of the largest empty ball with respect to the entire $\partial\Omega$, we take the intersection of a ball $B(x, s_x)$ with the domain Ω . The star-domain radius s_x is chosen to be maximally large while ensuring that the intersected region $St(x, s_x) = B(x, s_x) \cap \Omega$ excludes all of the Dirichlet boundary $\partial\Omega_D$ and is star-shaped. This means every ray centered at x intersects the boundary $\partial St(x, s_x)$ at most once, hitting either the Neumann boundary $\partial\Omega_N$ or part of the boundary of the ball $\partial B(x, s_x)$. On a star-shaped region $St(x, s_x)$, the solution u satisfies a generalized mean value property,

$$u(x) = \int_{\partial St(x, s_x)} P(x, y) u(y) dy, \quad (3.9)$$

where $P(x, y)$ is the Poisson kernel of the Laplace equation defined for a ball.

$$P_{2D}(x, y) = \frac{(\vec{n}(y) \cdot \vec{x}y)}{2\pi\|x - y\|}, \quad P_{3D}(x, y) = \frac{(\vec{n}(y) \cdot \vec{x}y)}{4\pi\|x - y\|^2}. \quad (3.10)$$

Remarkably, shooting a ray in a random direction from the center x of the star-shaped domain $St(x, s_x)$ and intersecting with the boundary $\partial St(x, s_x)$ produces a point y distributed according

to $P(x, y)$. In other words, the star-shaped domain is perfectly importance sampled by uniform *direction sampling*. This observation yields a direct Monte Carlo estimator:

$$\langle u(x) \rangle = \langle u(y) \rangle, \quad y \sim P(x, \cdot) \text{ on } \partial St(x, s_x). \quad (3.11)$$

Here the notation emphasizes that the Poisson kernel defines a probability distribution supported only on the boundary of the star-shaped domain. We refer to this algorithm as *walk on stars*, because we recursively evaluate the solution on star-shaped subdomains rather than the sphere subdomains used in walk on spheres.

Walk on stars is recursive like walk on spheres, yet continues even if y is sampled on the Neumann boundary $\partial\Omega_N$. Eventually y is sampled within the ϵ -shell of the Dirichlet boundary $\partial\Omega_D$ where the walk terminates

$$\langle u(x) \rangle = \begin{cases} g(\pi(x)) & \text{if } r_x < \epsilon \\ \langle u(y) \rangle & \text{otherwise} \end{cases}, \quad y \sim P(x, \cdot) \text{ on } \partial St(x, s_x). \quad (3.12)$$

In the special case where no Neumann boundary is present, the star-shaped region reduces to a ball, and walk on stars reduces to walk on spheres.

Crucially walk on stars interacts with geometry only through local queries to construct star-shaped subdomains and therefore scales well with complex geometry. In practice, the star-domain radius s_x is computed as the minimum of two queries: a closest-point query to the Dirichlet boundary and a closest-visibility silhouette query on the Neumann boundary. The latter query requires minimal changes to a bounding volume hierarchy (BVH) and retains a computational complexity $O(\log n)$ with respect to the number of boundary elements n . We describe the full details of this query and its acceleration structures in our work introducing walk on stars [75].

3.2.1 Extension to Robin boundary conditions

Having extended walk on spheres to Neumann boundaries, we next consider Robin conditions. Robin conditions are important in practice, since real materials are neither perfectly insulating (Neumann) nor perfectly absorbing (Dirichlet), but instead a mixture of both behaviors. Robin conditions naturally capture such realistic boundaries, making them essential in applications ranging from thermal analysis to fluid flow. Our “walkin’ Robin” modification [55] generalizes walk on stars to handle Robin boundary conditions with only minor changes to the algorithm. As before, we focus on the homogeneous case for clarity,

$$\Delta u(x) = 0 \quad x \in \Omega \quad (3.13)$$

$$u(x) = g(x) \quad x \in \partial\Omega_D \quad (3.14)$$

$$\frac{\partial u}{\partial \vec{n}}(x) - \mu(x) u(x) = 0 \quad x \in \partial\Omega_R, \quad (3.15)$$

where $\mu \geq 0$ is the Robin coefficient and the boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_R$ is partitioned into Dirichlet and Robin components. With Robin boundary conditions, we obtain a modified mean

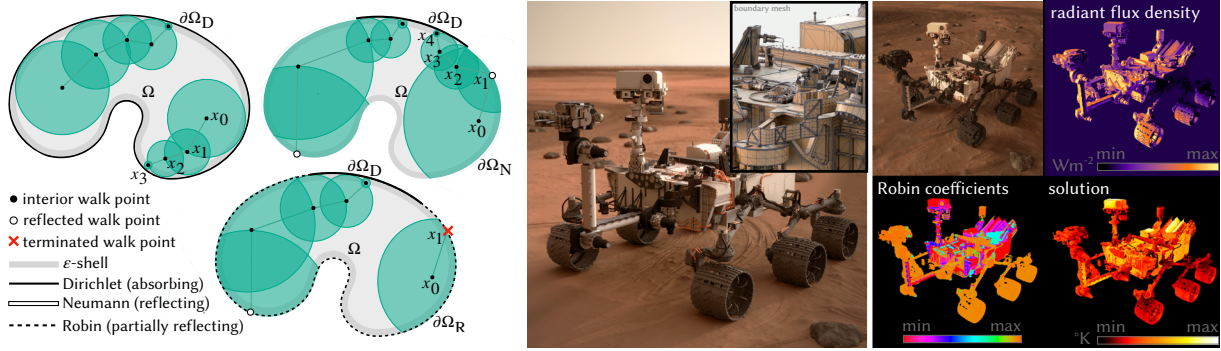


Figure 3.3: (Left) Walk on stars generalized to Robin boundary conditions: the size of star-shaped subdomains is reduced to bound reflectance, and walks can terminate early via Russian roulette. (Right) Thermal analysis of the NASA Curiosity Rover on an artist-generated model. Here Robin boundary conditions correspond to convective heat transfer, modeling thermal conduction on the rover subject to radiant flux at the boundary.

value relation

$$u(x) = \int_{\partial St(x, s_x)} \rho_\mu(y) P(x, y) u(y) dy \quad (3.16)$$

Here the mean value relation includes a reflectance term

$$\rho_\mu(x, y) = \begin{cases} 1 - \mu(y) \frac{G(x, y)}{P(x, y)} & \text{if } x \in B(x, s_x) \\ 1 & \text{otherwise} \end{cases} \quad (3.17)$$

where $G(x, y)$ is the Green's function for the Laplace equation defined for the ball,

$$G_{2D}(x, y) = \frac{\log(r/\|x - y\|)}{2\pi}, \quad G_{3D}(x, y) = \frac{1}{4\pi} \left(\frac{1}{r} - \frac{1}{\|x - y\|} \right). \quad (3.18)$$

Just as before, we form a recursive estimator with direction sampling and introduce an ϵ -shell on the Dirichlet boundary,

$$\langle u(x) \rangle = \rho_\mu(x, y) \begin{cases} g(\pi(x)) & \text{if } r_x < \epsilon \\ \langle u(y) \rangle & \text{otherwise} \end{cases}, \quad y \sim P(x, \cdot) \text{ on } \partial St(x, s_x). \quad (3.19)$$

The only difference from the original walk on stars is that we accumulate reflectance terms—similar to an albedo in rendering—with each recursive step. To ensure convergence, we require the reflectance to be strictly less than one, $|\rho_\mu| < 1$. We enforce this by reducing the star-domain radius to $s_x^\mu < s_x$, chosen so that $|\rho_\mu(x, y)| < 1$ for all points on the Robin boundary contained in the star-domain. Because reflectance is a geometric quantity, this safe radius s_x^μ can be computed efficiently in logarithmic time using a modified BVH. We describe this query and the data structure fully in our paper introducing walk on stars with Robin boundary conditions [55]. With reflectance bounded below 1, we can apply Russian roulette with survival probability equal to reflectance ρ_μ which reduces walk length and improves performance.

The Robin case requires only modest changes to walk on stars: shrinking the star-radius to control reflectance, and accumulating reflectance weights along the walk. These adjustments preserve the locality and scalability of the method while extending it to more realistic Robin boundary conditions. In this way, the same algorithm now accommodates Dirichlet, Neumann, and Robin conditions within a unified framework.

3.3 Differential walk on spheres

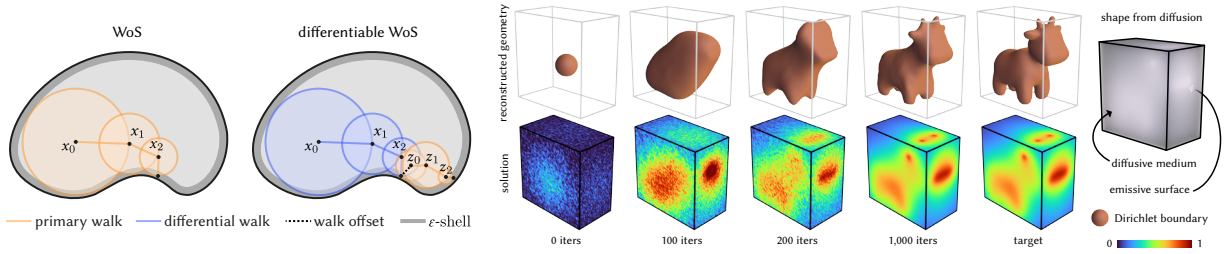


Figure 3.4: (Left) Differential walk on spheres solves a nested PDE to compute the derivative of the solution with respect to scene parameters. (Right) An emissive surface is reconstructed using diffusion profiles observed on a bounding box, via stochastic gradient descent. Unlike conventional mesh-based approaches we evaluate derivatives on regions of interest without computing a global solution.

Many modern applications of PDE solvers require *differentiable* variants. Optimization-driven design [81, 29], inverse reconstruction [30], and modern learning-based architectures [5, 78] all depend on computing derivatives of PDE solutions with respect to scene parameters. Formally, given a solution $u(x, \pi)$, we define its derivative

$$\dot{u}(x, \pi) := \frac{\partial u(x, \pi)}{\partial \pi}. \quad (3.20)$$

with u the solution of a Laplace equation with parameterized boundary condition $g(x, \pi)$ and geometry $\Omega(\pi)$

$$\begin{aligned} \Delta u(x, \pi) &= 0 & x &\in \Omega(\pi) \\ u(x, \pi) &= g(x, \pi) & x &\in \partial\Omega(\pi). \end{aligned} \quad (3.21)$$

While Monte Carlo solvers for analogous differential quantities in light transport have already enabled breakthroughs in design [60] and reconstruction [33], differential variants of walk on spheres remain underexplored. Our recent work [54] introduces such differential Monte Carlo solvers, enabling derivatives with respect to both boundary conditions and geometry. Along with other concurrent efforts [93, 92], this nascent area opens applications ranging from electrical impedance tomography [7] to thermal design [12].

To build such a differential solver we first recall an important observation from shape optimization: the derivative of the *primal solution* with respect to the parameters satisfies a corresponding nested PDE [29]

$$\begin{aligned} \Delta \dot{u}(x, \pi) &= 0 & x \in \Omega(\pi) \\ \dot{u}(x, \pi) &= V_{\vec{n}}(x) \left(\frac{\partial g(x, \pi)}{\partial \vec{n}} - \frac{\partial u(x, \pi)}{\partial \vec{n}} \right) + \frac{\partial g(x, \pi)}{\partial \pi} & x \in \partial\Omega(\pi), \end{aligned} \quad (3.22)$$

where the boundary normal velocity describes how boundary positions change along normal directions with respect to parameters:

$$V_{\vec{n}}(x) = \vec{n} \cdot \frac{\partial x(\pi)}{\partial \pi} \quad (3.23)$$

The insight of our work [54] is that the derivative $\dot{u}$ can be estimated using the same walk on spheres algorithm as the primal solution,

$$\langle \dot{u}(x, \pi) \rangle = \begin{cases} V_{\vec{n}}(x) \left(\frac{\partial g(x, \pi)}{\partial \vec{n}} - \frac{\partial u(x, \pi)}{\partial \vec{n}} \right) + \frac{\partial g(x, \pi)}{\partial \pi} & \text{if } r_x < \epsilon \\ \langle \dot{u}(y, \pi) \rangle & \text{otherwise} \end{cases}, \quad y \sim \mathcal{U}[\partial B(x, r_x)]. \quad (3.24)$$

The only difference from the primal solver is at the boundary, where we must evaluate the normal derivative of u . This can be approximated with a one-sided finite difference,

$$\left\langle \frac{\partial u(x, \pi)}{\partial \vec{n}} \right\rangle \approx \frac{g(x, \pi) - \langle u(x - \epsilon \vec{n}, \pi) \rangle}{\epsilon} \quad (3.25)$$

which requires an additional primal walk on spheres evaluation just inside the boundary. This adjoint-based strategy, long used in shape optimization [29], adapts naturally to the Monte Carlo regime. While we have so far focused on Dirichlet boundary geometry, the framework extends in principle to more general PDEs and boundary conditions. Such generalizations often require higher-order boundary derivatives, whose efficient evaluation remain an active area of research.

Building the differential solver on walk on spheres allows it to inherit the forward solver's benefits: scalability to complex geometry, compatibility with diverse representations, and output-sensitive evaluation. These advantages become even more important in optimization, where alternative discretization-based approaches would require repeated remeshing (explicit surfaces) or redistancing (implicit surfaces). Moreover, the stochastic nature of Monte Carlo methods enables noisy but efficient gradient estimates, making stochastic gradient descent feasible for shape optimization. This stochasticity can help avoid local minima and regularize optimization—an effect we demonstrate empirically in our work [54].

3.4 Solving PDEs on stochastic geometry

Explicit geometry is not always available or tractable for PDE solvers. In some cases, such as clouds, sand, or tissue, the boundary consists of quadrillions of microscopic particles whose exact arrangement cannot practically be represented in explicit form. In others, such as surface

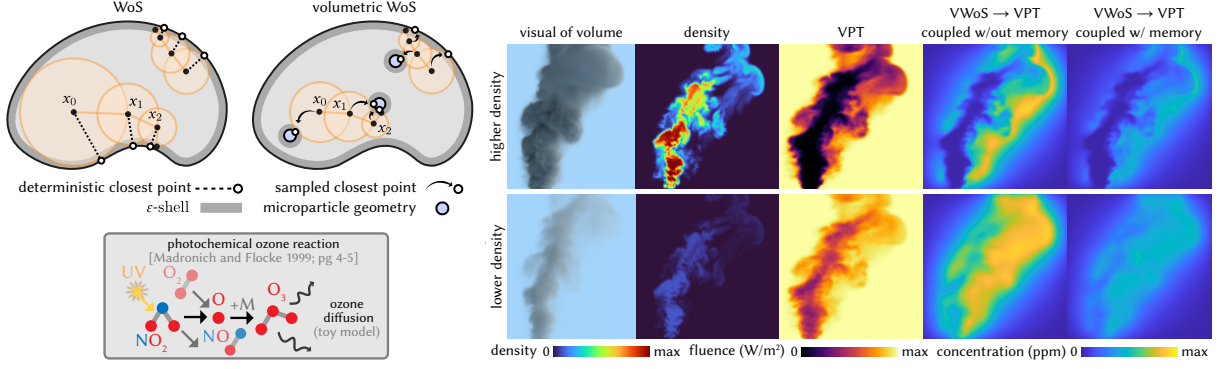


Figure 3.5: (Upper Left) Volumetric walk on spheres (VWoS) solves a Laplace PDE on stochastic microparticle geometry by sampling closest points. (Right, Lower Left) A toy photochemical system is simulated [49] in a participating medium: incident radiance, simulated with volumetric path tracing (VPT), drives the production of ozone, which then diffuses outward, simulated with VWoS. Because the solvers are coupled and interact with the same stochastic microparticle geometry, they must share conditioning or “memory” across the coupling.

reconstruction from measurements, the geometry itself is uncertain. In both settings it is natural to replace explicit geometry with a *random geometry* $\mathcal{O}$ that captures either unresolvable complexity or uncertainty. This idea has roots in stochastic geometry [8], the homogenization of PDEs in random media [63], and in computer graphics under the study of participating media [4, 37, 17]. In these settings, the primary goal is to compute or approximate the *expected* value of a solution,

$$\bar{u}(x) := \mathbb{E}_{\mathcal{O}}[u(x)] \quad (3.26)$$

under each geometry realization $\mathcal{O}$. In our case, the solution u satisfies a PDE such as the Laplace equation on $\Omega = \mathbb{R} \setminus \mathcal{O}$

$$\begin{aligned} \Delta u(x) &= 0 & x &\in \Omega, \\ u(x) &= g(x) & x &\in \partial\Omega. \end{aligned} \quad (3.27)$$

In the previous chapter we saw how rendering generalizes to these settings through volumetric approaches: deterministic ray–intersection queries are replaced with free-flight distribution samples. Our recent work [56] shows that the same strategy applies to walk on spheres, since it interacts with geometry only through closest-point queries to compute the largest empty sphere.

In particular, our *volumetric* walk on spheres method computes the mean solution $\bar{u}(x)$ for stochastic microparticle geometry, such as clouds or smoke. We model this geometry with a PBM $\mathcal{O} \sim \text{PBM}$, which is the very same abstraction used in computer graphics to represent participating media [41]. The key idea, as in volume rendering, is that instead of evaluating a deterministic closest point, we sample one according to the PBM. This modification requires only minor changes to the classical algorithm, while largely preserving its structure and performance.

Formally, under any stochastic geometry $\mathcal{O}$, we can express the mean solution to a Laplace equation via the expected mean value relation,

$$\bar{u}(x) = \mathbb{E}_{\mathcal{O}} \left[\int_{\partial B(x, r_x)} P(x, y) u(y) dy \right], \quad (3.28)$$

where $r_x = \|q^* - x\|$ is the distance to the closest boundary point and $q^* = \operatorname{argmin}_{q \in \partial\Omega} \|q - x\|$ is the closest boundary point. This expectation can be rewritten by conditioning on the location of that closest point q^* ,

$$\bar{u}(x) = \int_{\mathbb{R}^3} p_x(q) \mathbb{E}_{\mathcal{O}} \left[\int_{\partial B(x, r_x)} P(x, y) u(y) dy \mid q^* = q \right] dq. \quad (3.29)$$

Conditioned on the closest-point, everything inside the inner integral is deterministic except for the unknown solution $u(y)$. Pushing the expectation inside yields

$$\bar{u}(x) = \int_{\mathbb{R}^3} p_x(q) \int_{\partial B(x, r_x)} P(x, y) \bar{u}(y \mid q^* = q) dy dq. \quad (3.30)$$

where $\bar{u}(x \mid \cdot) := \mathbb{E}[u(x) \mid \cdot]$. Unlike rendering, the conditioning cannot be ignored without introducing significant bias. Therefore, we accumulate the conditional events in a set $M = \{q_1^* = q_1, q_2^* = q_2, \dots\}$ and condition the closest-point pdf on this set

$$\bar{u}(x \mid M) = \int_{\mathbb{R}^3} p_x(q \mid M) \int_{\partial B(x, r_x)} P(x, y) \bar{u}(x \mid M \cup \{q^* = q\}) dy dq. \quad (3.31)$$

We discuss the full details of the conditional closest point sampling for the PBM in our paper [56]. The resulting volumetric walk on spheres algorithm though is identical in structure to standard walk on spheres, except for the presence of an additional sampling decision and the “memory” M accumulated with each step

$$\langle \bar{u}(x \mid M) \rangle = \begin{cases} g(x) & \text{if } \|x - q\| < \epsilon \\ \langle \bar{u}(y \mid M \cup \{q^* = q\}) \rangle & \text{otherwise} \end{cases}, \quad \begin{matrix} q \sim p_x(\cdot \mid M) \\ y \sim \mathcal{U}[\partial B(x, \|x - q\|)] \end{matrix}. \quad (3.32)$$

The common theme across all generalizations of walk on spheres is that the algorithmic structure remains unchanged. As a result, we inherit the benefits and characteristics of the original algorithm. Additionally, unlike homogenization methods, which only approximate solutions for stochastic geometry [21], our approach yields an unbiased mean solution without assumptions on particle size or PBM rate parameters.

As in volume rendering [22], we anticipate that differentiable variants of the algorithm could be developed for inverse tasks such as estimating medium parameters. In cases where stochastic geometry reflects uncertainty, these variants could further support scene reconstruction, as recent volumetric approaches in computer vision demonstrate [52].

In volumetric methods for walk on spheres and rendering, the focus has largely been on microparticle collections. These models are mathematically convenient and broadly applicable. In geometric computing, however, we typically reason about solid objects—especially in applications

like surface reconstruction. In such cases, uncertainty or unresolved complexity is more naturally expressed through geometric models of solids rather than microparticles. Addressing this shift requires new stochastic representations—comprising both geometric models and query mechanisms—that fit naturally within stochastic algorithms. The next chapter develops this theme, introducing stochastic solids and, more broadly, learnable query distributions as new forms of stochastic representations.

Chapter 4

Stochastic representations

Representations of geometry—meshes, point clouds, implicit surfaces—are central to geometric computing. They provide the substrate on which algorithms operate, typically through pointwise queries such as ray-intersection or closest-point evaluation. Two fundamental challenges arise in practice: *complexity*, where the number of geometric elements becomes overwhelming, and *uncertainty*, where the geometry is known up to some noise or intrinsically random.

Geometric complexity has long been the province of computer graphics, where stochastic representations are well established. From extensive models of stochastic microparticle geometry [4, 37, 17, 27, 34] to stochastic surface detail models [95, 28], diverse representations of stochastic microparticle geometry make it possible to evaluate the aggregate effect of microparticle ensembles. By contrast, representations of uncertainty in solid geometry have not been developed to the same extent, at least not in a form usable by stochastic algorithms. Yet these representations are a natural fit for a broad range of applications including surface reconstruction in computer vision [11, 83], geometric reasoning in robotics [16], and uncertainty quantification in engineering design [90, 47].

Our main contribution in this chapter is to develop principled representations for stochastic opaque solids [53], enabling stochastic algorithms to operate on uncertain solid geometry. While our focus is on volume rendering and more specifically free-flight distributions—the distribution of first intersection events along a ray—the same principles in theory extend to other stochastic algorithms and geometric queries such as closest-point evaluation [44, 56]. Beyond stochastic opaque solids, we discuss our work on prior-free models for free-flight distributions, which enable learning representations of scene geometry without specifying a stochastic geometric model a priori. By casting geometric representations in stochastic terms and interfacing with them through distributional queries, SGPs enable new forms of geometric computing on both complex and uncertain geometry.

4.1 Stochastic opaque solids

Volumetric scene representations have long been central to resolving measurement uncertainty in computer vision [11] and have gained renewed attention with the advent of neural radiance fields (NeRF) [52]. NeRF demonstrates the power of volumetric models for novel view synthesis

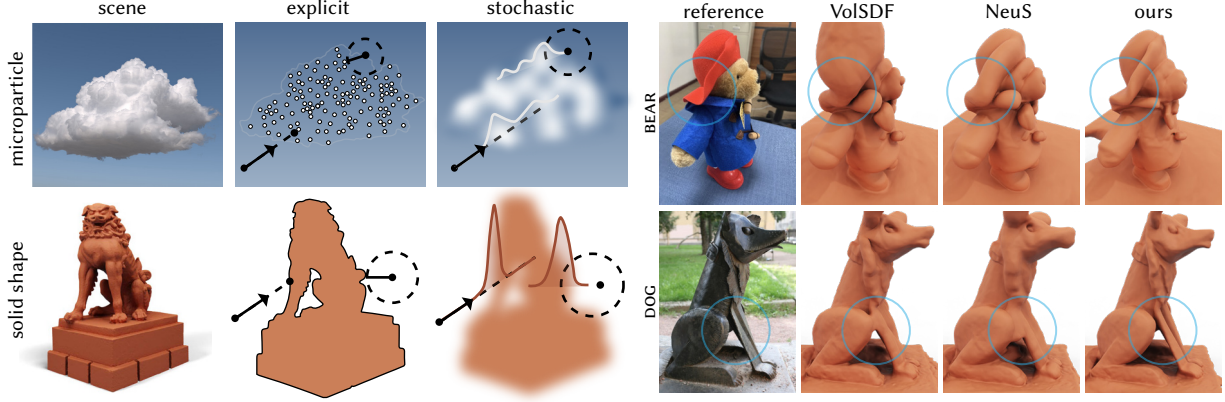


Figure 4.1: (Left) Stochastic opaque solids extend solid geometry with uncertainty, analogous to how stochastic microparticle models capture geometric complexity. Deterministic queries in the explicit setting are replaced with distributional equivalents. (Right) We leverage these representations for surface reconstruction [53], showing that a probabilistic formulation generalizes and improves upon more heuristic models [91, 88].

by representing scenes as participating media, e.g. stochastic microparticle geometry. Yet most real-world environments are composed of solid geometry rather than microparticles. If our goal is not only view synthesis, but also reconstructing surfaces and reasoning about physical structure, we need representations that enable rendering algorithms to account for uncertainty in solid geometry.

Our work develops these representations with the introduction of stochastic opaque solids [53]. Specifically, we model a stochastic solid $\mathcal{O}$ in terms of a random indicator function

$$I : \mathbb{R}^3 \rightarrow \{0, 1\}, \quad \mathcal{O} \equiv \{x \in \mathbb{R}^3 \mid I(x) = 1\}. \quad (4.1)$$

The indicator specifies whether a point lies inside the solid ($I(x) = 1$) or outside ($I(x) = 0$). From this, we define the occupancy and vacancy fields as the probabilities

$$o(x) = \Pr[I(x) = 1], \quad (4.2)$$

$$v(x) = 1 - o(x). \quad (4.3)$$

To simulate light transport on $\mathcal{O}$, we start from the radiance conservation law used for deterministic geometry,

$$L(x, \vec{\omega}) = L(r_{x, \vec{\omega}}^*, -\vec{\omega}). \quad (4.4)$$

For surface reconstruction applications, this relationship simplifies since we model the outgoing radiance as purely emissive just as in NeRF,

$$L(x, \vec{\omega}) = L_e(r_{x, \vec{\omega}}^*, -\vec{\omega}). \quad (4.5)$$

We then take the expectation over the stochastic solid O , which introduces an integral over the distribution of the first intersection distance,

$$\mathbb{E}_O[L(x, \vec{\omega})] = \mathbb{E}_O\left[L_e\left(r_{x,\vec{\omega}}^*, -\vec{\omega}\right)\right] = \int_0^\infty p_{x,\vec{\omega}}^{\text{ff}}(t) L_e(r_{x,\vec{\omega}}(t), -\vec{\omega}) dt. \quad (4.6)$$

Here $r_{x,\vec{\omega}}(t) \equiv x + t\vec{\omega}$ denotes a point along the ray, and $r_{x,\vec{\omega}}^* \equiv r_{x,\vec{\omega}}(t_{x,\vec{\omega}}^*)$ is the first intersection with $t_{x,\vec{\omega}}^* \equiv \min\{t \geq 0 \mid r_{x,\vec{\omega}}(t) \in O\}$ the first intersection distance. We can then evaluate this integral using either Monte Carlo integration or quadrature [50, 52]. If the indicator I is a Markov process along the ray $r_{x,\vec{\omega}}(t)$ with respect to distance t , then we prove in our paper [53] that the first intersection distance follows an exponential distribution,

$$p_{x,\vec{\omega}}^{\text{ff}}(t) = \sigma(r_{x,\vec{\omega}}) \exp\left(-\int_0^t \sigma(r_{x,\vec{\omega}}(s)) ds\right), \quad \sigma(x, \vec{\omega}) = \frac{|\vec{\omega} \cdot \nabla v(x)|}{v(x)}. \quad (4.7)$$

with rate parameter $\sigma(x, \vec{\omega})$ determined by the vacancy function and its gradient. This form makes clear that the intersection probability depends on the local vacancy and the alignment of the ray with the gradient of the vacancy field, i.e. the surface normal. The exponential structure provides a principled bridge between classic volumetric models and stochastic opaque solids, and explains why exponential media already mimic uncertain solid surfaces to some degree.

In our paper [53], we generalize this framework in several directions. One extension factors out the directional term to model a distribution of surface normals. Another extension parameterizes the vacancy field with a stochastic implicit surface. These extensions reveal a design space for stochastic opaque solid geometry, which unifies prior methods such as VolSDF [91] and NeuS [88] as different choices of underlying implicit surface and normal distributions. This perspective enables principled modifications—such as Gaussian process implicit surfaces or spatially varying anisotropy—that improve surface reconstruction.

In follow-up work, we have shown that the formulation is agnostic to the choice of stochastic implicit surface representation, allowing it to be combined with point-based formulations such as fast dipole sums [6]. Other subsequent work [76, 94] has introduced new stochastic geometry models for solids, incorporating correlations between evaluation points or mutually exclusive interactions with geometry. Building on these directions, in the next section we introduce a query-centric, prior-free model for the free-flight distribution, enabling scene reconstruction without assuming a specific stochastic geometry model a priori.

4.2 Scenes as free-flight distributions

A recurring theme of this proposal is that stochastic algorithms interface with geometry—whether deterministic or stochastic—only through queries. From this perspective, the role of a stochastic geometric model is mainly to parameterize a query such as the free-flight distribution. However, this reliance can be limiting: it forces us to select a stochastic model for a scene a priori, and more critically, it often complicates evaluation. Computing the free-flight distribution typically requires integrating a rate parameter along a ray. This process can be slow, and it introduces sensitivity to sampling that leads to inaccuracies.

For surfaces, this manifests as missed thin structures or failure to capture the sharp peaks in the free-flight distribution where surface interactions occur. But if all we require is the query itself, why not learn it directly? Instead of prescribing a stochastic model, we can predict a parameterized free-flight distribution that serves as a universal approximator (up to the number of parameters) and is integration-free by design. This is the central idea of free-flight distribution networks (FreD).

First, we describe a parameterized model of *transmittance* which by construction ensures valid transmittance functions and whose negated derivatives produce analytic free-flight distributions. Let $a, b \in \mathbb{R}$ be scalar, signed-distances on any oriented line l with respect to the line origin $o_l \equiv \operatorname{argmin}_{q \in l} \|q\|$. We define transmittance as the product of elementary transmittance functions,

$$T_l(a, b) = \prod_{i=1}^N T_l^{(i)}(a, b) \quad (4.8)$$

So long as each elementary transmittance function is reciprocal $T_l^{(i)}(a, b) = T_l^{(i)}(b, a)$, monotonically decreasing, and restricted to the range $[0, 1]$ then so too will T_l satisfy these properties. To enable analytic evaluation and adherence to the transmittance properties, we choose the elementary transmittance functions to be soft, step functions,

$$T_l^{(i)}(a, b) = 1 - \alpha_i (\Phi(\max(a, b) \mid \mu_i, \sigma_i) - \Phi(\min(a, b) \mid \mu_i, \sigma_i)) \quad (4.9)$$

where $\Phi(\cdot \mid \mu_i, \sigma_i)$ is the CDF of a location-scale distribution, such as a Gaussian, with mean $\mu_i \in \mathbb{R}$ and variance $\sigma_i \in \mathbb{R}_+$. An additional scale α_i weighs the contribution of each primitives. Since each elementary transmittance acts as a step function, the product transmittance is a universal approximation of transmittance for sufficiently many elementary terms. For such a transmittance representation, the corresponding free-flight distribution is also a closed-form expression,

$$p_{l,a}^{\text{ff}}(b) = -\frac{\partial T_l^{(i)}(a, b)}{\partial b} = \sum_{i=1}^N \left[\prod_{j \neq i} T_l^{(j)}(a, b) \right] \alpha_i \phi(b \mid \mu_i, \sigma_i) \quad (4.10)$$

where the free-flight defines the distribution of distance $|b - a|$ to the first intersection along l in direction $\operatorname{sign}(b - a)$. Given this parameterized free-flight distribution with $3N$ parameters, $\pi_i = \{\alpha_i, \mu_i, \sigma_i\}$ and $\pi = \{\pi_i\}_{i=1}^N$, all that remains is to predict these parameters for any given line l in the scene. To do so, we bound the line l using a scene radius $[-R, R]$ and then,

1. looks up M uniformly spaced, D dimensional features along the ray from a background feature volume such as dense voxel grid or hash-grid [58],
2. divide the ray up into equally spaced intervals with $\frac{M}{N}$ features per interval,
3. and pass ordered features into a network in both the forward and backward order to predict parameters for two elementary transmittance functions per interval.

Since the free-flight distribution is insensitive to the order in which the primitives are indexed, the final step ensures order invariance for features.

FreD differs from prior ray query models that bypass integration [80, 1, 48]. Instead of predicting a single ray query, FreD predicts scene structure along an entire chord l using

uniformly spaced features. This yields stable training and reciprocity in transmittance by construction rather than through regularization. Although FreD still samples modestly (e.g., 128 points per ray), evaluation remains integration-free: no up-sampling or quadrature is needed. Because primitives are learned continuously along the chord, these fixed samples offer much higher effective precision than a coarse NeRF, especially for surface geometry where stochastic solid models [88, 91, 61, 53] require heavy up-sampling to capture concentrated regions in the free-flight distribution.

Since this project is in its final stages, we present it as existing work and note its anticipated completion date in Section 5.3.

Chapter 5

Proposed work

We propose advances on two complementary fronts: stochastic algorithms, extending Monte Carlo PDE solvers into general-purpose iterative methods, and stochastic representations, developing stochastic implicit surfaces based on the Gaussian Free Field.

Monte Carlo PDE solvers are general, but so far confined to direct estimation. We propose extending them into iterative hybrid solvers—combining walk on spheres with neural components—to tackle nonlinear and coupled PDEs in complex geometry.

Additionally, we introduce a stochastic implicit surface representation based on the Gaussian Free Field, whose mean recovers the winding number and whose covariance can be evaluated efficiently via Barnes–Hut acceleration, yielding a principled and scalable model for surface geometry in simulation and inverse problems.

5.1 Iterative hybrid solvers for more general PDEs

We introduce *WoS-Net*, a general framework for solving nonlinear PDEs by embedding walk on spheres inside an iterative fixed-point solver. Our starting point is the nonlinear Dirichlet problem

$$\begin{aligned}\Delta u(x) + F(u(x)) &= f(x) & x \in \Omega \\ u(x) &= g(x) & x \in \partial\Omega,\end{aligned}\tag{5.1}$$

We observe that the inverse of the Laplacian can be realized stochastically via walk on spheres,

$$\Delta^{-1}f(x) \sim g(x_N) + \sum_{i=1}^N G(x_i, y_i) f(y_i).\tag{5.2}$$

and so we can naturally apply this inverse to form a Picard iteration of equation (5.1),

$$u_k = \Delta^{-1}(f - F(u_{k-1})),\tag{5.3}$$

where for brevity we drop spatial arguments of the functions. We can also generalize to more stable operator-splitting schemes such as Douglas–Rachford [23], where we alternate between

pointwise nonlinear inversions and spatial linear inversions,

$$v_k = [I + \gamma F]^{-1}(u_k - f) \quad (5.4)$$

$$u_{k+1} = [\Delta - \gamma]^{-1}v_k, \quad (5.5)$$

enabling accelerated convergence.

Rather than store solutions on a grid, we realize these updates by supervising a neural surrogate $u(x, \theta)$ against the stochastic inverse. We plan to use an update procedure similar to Mehta et al. [51]’s neural level-set evolution, where a stochastic target is computed and network parameters are guided toward it via gradient descent rather than an exact update.

Compared to prior neural walk on spheres methods [45, 59], we are not using walk on spheres as a direct solver but rather to facilitate iterative solves. Some prior works have also explored the use of walk on spheres in the context of operator splitting [70, 32, 82], they usually do so in a time-stepping regime where relatively denoised, global solves are required to ensure a stable step. In the steady state regime, our expectation is that iterative solvers may be more localized and even noisier, so long as step sizes are adjusted accordingly.

Finally, while this may superficially resemble PINNs [69], there are a few key differences. First, we avoid higher order gradients by leveraging the stochastic inverse of the Laplacian. Unlike conventional PINNs, we also stop the gradient through $F(u)$, so updates occur only across iterations—yielding a true iterative solver driven by Monte Carlo supervision. We also don’t have to explicitly enforce a boundary loss since it’s captured naturally by the stochastic inversion which enables better handling of complex geometry.

In pursuing this project, our goal is to demonstrate that we can retain the capabilities of walk on spheres that make it appealing—scalability to extremely complex geometry—while finally demonstrating PDEs that go beyond the standard linear elliptic regime. Towards this end, we’ll also explore practical improvements throughout this project as needed including anything from PDE-aware feature representations to effective sample reuse.

5.2 Stochastic fast winding numbers

We propose a stochastic extension of the winding number, yielding a principled distribution over implicit surfaces that retains its efficiency while exposing both a mean and covariance query. We define a random field $u : \mathbb{R}^3 \rightarrow \mathbb{R}$ as a Gaussian process [79]

$$u \sim \mathcal{GP}\left(0, G_\epsilon^\Omega(x, y)\right), \quad (5.6)$$

Here, G_ϵ^Ω is the free-space regularized Green’s function of the Laplacian [3] on domain Ω . The use of the regularized Green’s kernel has a similar effect as in Chen et al. [6] where it helps to avoid strong singularities that create artifacts. For now, consider $\Omega = \mathbb{R}^3$ for simplicity where the Green’s function is known in closed form,

$$G_\epsilon^{\mathbb{R}^3}(x, y) = \frac{\text{erf}(\|x-y\|/\epsilon)}{4\pi\|x-y\|} \quad (5.7)$$

Conditioning this field on a point cloud $\mathcal{P} = \{p_i \vec{n}_i, \alpha_i\}_{i=1}^N$ yields a distribution over harmonic implicit surfaces whose mean surface is the generalized winding number,

$$\mathbb{E}[u(x) \mid \mathcal{P}] = \sum_{i=1}^N \alpha_i P_\epsilon^\Omega(x, p_i, \vec{n}_i). \quad (5.8)$$

Here $P_\epsilon^\Omega(x, p, \vec{n}) = \vec{n} \cdot V_\epsilon^\Omega(x, p)$ is the regularized free-space Poisson kernel of the Laplacian,

$$V_\epsilon^\Omega(x, y) = \frac{(x - p)}{\|x - p\|^3} S(\|x - p\|/\epsilon) \quad (5.9)$$

with $S(t) \equiv \text{erf}(t) - \frac{2t}{\sqrt{\pi}} \exp(-t^2)$. Since u is a Gaussian process, we can also evaluate the covariance with a similar summation

$$\text{Cov}(u(x), u(y) \mid \mathcal{P}) = G_\epsilon^\Omega(x, y) - \sum_{i=1}^N \alpha_i V_\epsilon^\Omega(x, p_i) \cdot V_\epsilon^\Omega(y, p_i) \quad (5.10)$$

This perspective generalizes fast winding numbers [2] into a stochastic fast winding number (SFWN), where covariance evaluation remains efficient: the sums in (5.10) admit the same Barnes–Hut–style fast summation used for winding number evaluation. Moreover, SFWN can condition on domains other than free space by substituting the appropriate regularized Green’s function G_ϵ^Ω in the summation. The Green’s function for a domain satisfies a Poisson equation and so it can be evaluated efficiently with walk on spheres.

Finally, Seyb et al. [76] introduce volume rendering algorithms for Gaussian process implicit surfaces (GPIS). Because SFWN is itself a GPIS, these rendering algorithms apply directly. Beyond volume rendering, Seyb et al. [76] also suggest that such techniques may extend to walk on spheres, analogous to our approach with stochastic microparticle geometry in Section 3.4, provided a stochastic closest-point query can be developed for the GPIS. We will explore these extensions as part of this project, demonstrating that SFWN enables scalable, robust rendering and simulation on stochastic surface representations.

5.3 Timeline

We are in the process of completing two projects, “Scenes as Free-Flight Distributions” and “Stochastic Fast Winding Number,” which we expect to finish by November 2025 and January 2026, respectively. In parallel, we will begin work on the WoS-Net project, with the goal of completing it around May 2026. Dissertation writing will begin in early summer, with the defense planned for late summer. An overview is provided below.

- September 2025 – Propose thesis.
- November 2025 – Submit “Scenes as Free-Flight Distributions” (e.g. FreD) to CVPR.
- January 2026 – Submit “Stochastic Fast Winding Number” to SIGGRAPH.
- May 2026 – Complete “WoS-Net” and submit to TBD venue.
- June 2026 – Begin dissertation writing.
- August 2026 – Complete dissertation and defend thesis.

Bibliography

- [1] Tristan Aumentado-Armstrong, Stavros Tsogkas, Sven Dickinson, and Allan D Jepson. Representing 3d shapes with probabilistic directed distance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19343–19354, 2022.
- [2] Gavin Barill, Neil G Dickson, Ryan Schmidt, David IW Levin, and Alec Jacobson. Fast winding numbers for soups and clouds. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.
- [3] J Thomas Beale, Wenjun Ying, and Jason R Wilson. A simple method for computing singular or nearly singular integrals on closed surfaces. *Communications in Computational Physics*, 20(3):733–753, 2016.
- [4] Benedikt Bitterli, Srinath Ravichandran, Thomas Müller, Magnus Wrenninge, Jan Novák, Steve Marschner, and Wojciech Jarosz. A radiative transfer framework for non-exponential media. *ACM Trans. Graph.*, 37(6), December 2018. ISSN 0730-0301. doi: 10.1145/3272127.3275103. URL <https://doi.org/10.1145/3272127.3275103>.
- [5] Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International conference on machine learning*, pages 1407–1418. PMLR, 2021.
- [6] Hanyu Chen, Bailey Miller, and Ioannis Gkioulekas. 3d reconstruction with fast dipole sums. *ACM Trans. Graph.*, 43(6), November 2024. ISSN 0730-0301. doi: 10.1145/3687914. URL <https://doi.org/10.1145/3687914>.
- [7] Margaret Cheney, David Isaacson, and Jonathan C Newell. Electrical impedance tomography. *SIAM review*, 41(1):85–101, 1999.
- [8] Sung Nok Chiu, Dietrich Stoyan, Wilfrid S Kendall, and Joseph Mecke. *Stochastic geometry and its applications*. John Wiley & Sons, 2013.
- [9] Jacek Chowdhary, Peng-Wang Zhai, Emmanuel Boss, Heidi Dierssen, Robert Frouin, Amir Ibrahim, Zhongping Lee, Lorraine A Remer, Michael Twardowski, Feng Xu, et al. Modeling atmosphere-ocean radiative transfer: A pace mission perspective. *Frontiers in Earth Science*, 7:100, 2019.
- [10] Per H. Christensen and Wojciech Jarosz. The path to path-traced movies. *Found. Trends. Comput. Graph. Vis.*, 10(2):103–175, October 2016. ISSN 1572-2740. doi: 10.1561/06000000073. URL <https://doi.org/10.1561/06000000073>.

- [11] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.
- [12] Talib Dbouk. A review about the engineering design of optimal heat transfer systems using topology optimization. *Applied Thermal Engineering*, 112:841–854, 2017.
- [13] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’99, page 317–324, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0201485605. doi: 10.1145/311535.311576. URL <https://doi.org/10.1145/311535.311576>.
- [14] Frédo Durand, George Drettakis, and Claude Puech. Fast and accurate hierarchical radiosity using global visibility. *ACM Trans. Graph.*, 18(2):128–170, April 1999. ISSN 0730-0301. doi: 10.1145/318009.318012. URL <https://doi.org/10.1145/318009.318012>.
- [15] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. A frequency analysis of light transport. *ACM Trans. Graph.*, 24(3): 1115–1126, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073320. URL <https://doi.org/10.1145/1073204.1073320>.
- [16] Hugh F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal on Robotics and Automation*, 4(1):23–31, 1988. doi: 10.1109/56.768.
- [17] Eugene d’Eon. A reciprocal formulation of nonexponential radiative transfer. 1: Sketch and motivation. *Journal of Computational and Theoretical Transport*, 47(1-3):84–115, 2018. doi: 10.1080/23324309.2018.1481433. URL <https://doi.org/10.1080/23324309.2018.1481433>.
- [18] Luca Fascione, Johannes Hanika, Rob Pieké, Ryusuke Villemin, Christophe Hery, Manuel Gamito, Luke Emrose, and André Mazzone. Path tracing in production. In *ACM SIGGRAPH 2018 Courses*, SIGGRAPH ’18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450358095. doi: 10.1145/3214834.3214864. URL <https://doi.org/10.1145/3214834.3214864>.
- [19] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’01, page 15–22, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 158113374X. doi: 10.1145/383259.383260. URL <https://doi.org/10.1145/383259.383260>.
- [20] Mark Gillespie, Denise Yang, Mario Botsch, and Keenan Crane. Ray tracing harmonic functions. *ACM Trans. Graph.*, 43(4), July 2024. ISSN 0730-0301. doi: 10.1145/3658201. URL <https://doi.org/10.1145/3658201>.
- [21] Arianna Giunti, Richard Höfer, and Juan J. L. Velázquez. Homogenization for the poisson equation in randomly perforated domains under minimal assumptions on the size of the holes. *Communications in Partial Differential Equations*, 43(9):1377–1412, 2018. doi: 10.1080/03605302.2018.1531425. URL <https://doi.org/10.1080/03605302.2018.1531425>.

- [22] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Trans. Graph.*, 32(6), November 2013. ISSN 0730-0301. doi: 10.1145/2508363.2508377. URL <https://doi.org/10.1145/2508363.2508377>.
- [23] Roland Glowinski, Stanley J. Osher, and Wotao Yin. *Splitting Methods in Communication, Imaging, Science, and Engineering*. Scientific Computation. Springer International Publishing, Cham, 1st ed. 2016. edition, 2016. ISBN 3-319-41587-5.
- [24] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, page 213–222, New York, NY, USA, 1984. Association for Computing Machinery. ISBN 0897911385. doi: 10.1145/800031.808601. URL <https://doi.org/10.1145/800031.808601>.
- [25] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996.
- [26] Paul S. Heckbert. Discontinuity meshing for radiosity. In *Third Eurographics Workshop on Rendering*, pages 203–226, 1992.
- [27] Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. The sggx microflake distribution. *ACM Trans. Graph.*, 34(4), July 2015. ISSN 0730-0301. doi: 10.1145/2766988. URL <https://doi.org/10.1145/2766988>.
- [28] Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. Multiple-scattering microfacet bsdfs with the smith model. *ACM Transactions on Graphics (TOG)*, 35(4):1–14, 2016.
- [29] Antoine Henrot and Michel Pierre. *Shape variation and optimization*, volume 28. European Mathematical Society Publishing House, Zuerich, Switzerland, 2018.
- [30] Victor Isakov. *Inverse problems for partial differential equations*. Springer, 2006.
- [31] Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4), July 2011. ISSN 0730-0301. doi: 10.1145/2010324.1964973. URL <https://doi.org/10.1145/2010324.1964973>.
- [32] Pranav Jain, Ziyin Qu, Peter Yichen Chen, and Oded Stein. Neural monte carlo fluid simulation. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705250. doi: 10.1145/3641519.3657438. URL <https://doi.org/10.1145/3641519.3657438>.
- [33] W. Jakob, Y. Li, Z. Wang, B. Nicolet, W. Tan, and A. Goldan. Inverse rendering for pet image reconstruction. In *2024 IEEE Nuclear Science Symposium (NSS), Medical Imaging Conference (MIC) and Room Temperature Semiconductor Detector Conference (RTSD)*, pages 1–1, 2024. doi: 10.1109/NSS/MIC/RTSD57108.2024.10654967.
- [34] Wenzel Jakob, Adam Arbree, Jonathan T. Moon, Kavita Bala, and Steve Marschner. A radiative transfer framework for rendering materials with anisotropic structure. *ACM Trans. Graph.*, 29(4), July 2010. ISSN 0730-0301. doi: 10.1145/1778765.1778790. URL <https://doi.org/10.1145/1778765.1778790>.

- [35] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. Dr.jit: a just-in-time compiler for differentiable rendering. *ACM Trans. Graph.*, 41(4), July 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530099. URL <https://doi.org/10.1145/3528223.3530099>.
- [36] Adrian Jarabo, Julio Marco, Adolfo Munoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. A framework for transient rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 33(6), November 2014. doi: 10/gfznb8.
- [37] Adrian Jarabo, Carlos Aliaga, and Diego Gutierrez. A radiative transfer framework for spatially-correlated materials. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201282. URL <https://doi.org/10.1145/3197517.3201282>.
- [38] Henrik Wann Jensen. Global illumination using photon maps. In *Eurographics workshop on Rendering techniques*, pages 21–30. Springer, 1996.
- [39] Pushkar Joshi and Nathan A. Carr. Repoussé: Automatic Inflation of 2D Artwork. In Christine Alvarado and Marie-Paule Cani, editors, *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. The Eurographics Association, 2008. ISBN 978-3-905674-07-1. doi: /10.2312/SBM/SBM08/049-055.
- [40] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, page 143–150, New York, NY, USA, 1986. Association for Computing Machinery. ISBN 0897911962. doi: 10.1145/15922.15902. URL <https://doi.org/10.1145/15922.15902>.
- [41] James T. Kajiya and Brian P Von Herzen. Ray tracing volume densities. *SIGGRAPH Comput. Graph.*, 18(3):165–174, January 1984. ISSN 0097-8930. doi: 10.1145/964965.808594. URL <https://doi.org/10.1145/964965.808594>.
- [42] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, page 61–70, Goslar, DEU, 2006. Eurographics Association. ISBN 3905673363.
- [43] Eric P Lafortune and Yves D Willems. Rendering participating media with bidirectional path tracing. In *Eurographics Workshop on Rendering Techniques*, pages 91–100. Springer, 1996.
- [44] Günter Last and Mathew Penrose. *Lectures on the Poisson process*, volume 7. Cambridge University Press, 2018.
- [45] Zilu Li, Guandao Yang, Xi Deng, Christopher De Sa, Bharath Hariharan, and Steve Marschner. Neural caches for monte carlo partial differential equation solvers. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023.
- [46] D. Lischinski, F. Tampieri, and D.P. Greenberg. Discontinuity meshing for accurate radiosity. *IEEE Computer Graphics and Applications*, 12(6):25–39, 1992. doi: 10.1109/38.163622.
- [47] Dishu Liu, Alexander Litvinenko, Claudia Schillings, and Volker Schulz. Quantification of air-foil geometry-induced aerodynamic uncertainties—comparison of approaches. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):334–352, 2017. doi: 10.1137/15M1050239. URL <https://doi.org/10.1137/15M1050239>.

- [48] Zhuoman Liu, Bo Yang, Yan Luximon, Ajay Kumar, and Jinxi Li. Raydf: neural ray-surface distance fields with multi-view consistency. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.
- [49] Sasha Madronich and Siri Flocke. The role of solar radiation in atmospheric chemistry. *Environmental photochemistry*, pages 1–26, 1999.
- [50] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 2002.
- [51] Ishit Mehta, Manmohan Chandraker, and Ravi Ramamoorthi. A level set theory for neural implicit evolution under explicit flows. In *European conference on computer vision*, pages 711–729. Springer, 2022.
- [52] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.
- [53] Bailey Miller, Hanyu Chen, Alice Lai, and Ioannis Gkioulekas. Objects as volumes: A stochastic geometry view of opaque solids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 87–97, June 2024.
- [54] Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. Differential walk on spheres. *ACM Transactions on Graphics (TOG)*, 43(6):1–18, 2024.
- [55] Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. Walkin’ robin: Walk on stars with robin boundary conditions. *ACM Trans. Graph.*, 43(4), July 2024. ISSN 0730-0301. doi: 10.1145/3658153. URL <https://doi.org/10.1145/3658153>.
- [56] Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. Solving partial differential equations in participating media. *ACM Trans. Graph.*, 44(4), jul 2025. ISSN 0730-0301. doi: 10.1145/3731152. URL <https://doi.org/10.1145/3731152>.
- [57] Mervin E Muller. Some continuous monte carlo methods for the dirichlet problem. *The Annals of Mathematical Statistics*, pages 569–589, 1956.
- [58] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL <https://doi.org/10.1145/3528223.3530127>.
- [59] Hong Chul Nam, Julius Berner, and Anima Anandkumar. Solving poisson equations using neural walk-on-spheres. *arXiv preprint arXiv:2406.03494*, 2024.
- [60] Baptiste Nicolet, Felix Wechsler, Jorge Madrid-Wolff, Christophe Moser, and Wenzel Jakob. Inverse rendering for tomographic volumetric additive manufacturing. *ACM Trans. Graph.*, 43(6), November 2024. ISSN 0730-0301. doi: 10.1145/3687924. URL <https://doi.org/10.1145/3687924>.
- [61] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5589–5599, 2021.

- [62] Ryan S. Overbeck, Craig Donner, and Ravi Ramamoorthi. Adaptive wavelet rendering. *ACM Trans. Graph.*, 28(5):1–12, December 2009. ISSN 0730-0301. doi: 10.1145/1618452.1618486. URL <https://doi.org/10.1145/1618452.1618486>.
- [63] George C Papanicolaou. Diffusion in random media. In *Surveys in applied mathematics*, pages 205–253. Springer, 1995.
- [64] Steven G. Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, and Martin Stich. Optix: a general purpose ray tracing engine. *ACM Trans. Graph.*, 29(4), July 2010. ISSN 0730-0301. doi: 10.1145/1778765.1778803. URL <https://doi.org/10.1145/1778765.1778803>.
- [65] Mark Pauly, Thomas Kollig, and Alexander Keller. Metropolis light transport for participating media. In *Eurographics Workshop on Rendering Techniques*, pages 11–22. Springer, 2000.
- [66] Adithya Pediredla, Ashok Veeraraghavan, and Ioannis Gkioulekas. Ellipsoidal path connections for time-gated rendering. *ACM Trans. Graph.*, 38(4), July 2019. ISSN 0730-0301. doi: 10.1145/3306346.3323016. URL <https://doi.org/10.1145/3306346.3323016>.
- [67] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. MIT Press, 2023.
- [68] Timothy J. Purcell, Ian Buck, William R. Mark, and Pat Hanrahan. Ray tracing on programmable graphics hardware. *ACM Trans. Graph.*, 21(3):703–712, July 2002. ISSN 0730-0301. doi: 10.1145/566654.566640. URL <https://doi.org/10.1145/566654.566640>.
- [69] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [70] Damien Rioux-Lavoie, Ryusuke Sugimoto, Tümay Özdemir, Naoharu H Shimada, Christopher Batty, Derek Nowrouzezahrai, and Toshiya Hachisuka. A monte carlo method for fluid simulation. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022.
- [71] Katherine Salesin, Kirk D. Knobelspiesse, Jacek Chowdhary, Peng-Wang Zhai, and Wojciech Jarosz. Unifying radiative transfer models in computer graphics and remote sensing, part i: A survey. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 314: 108847, 2024. ISSN 0022-4073. doi: <https://doi.org/10.1016/j.jqsrt.2023.108847>. URL <https://www.sciencedirect.com/science/article/pii/S0022407323003655>.
- [72] Rohan Sawhney. Fcpw: Fastest closest points in the west, 2021.
- [73] Rohan Sawhney and Keenan Crane. Monte carlo geometry processing: A grid-free approach to pde-based methods on volumetric domains. *ACM Transactions on Graphics*, 39(4), 2020.
- [74] Rohan Sawhney, Dario Seyb, Wojciech Jarosz, and Keenan Crane. Grid-free monte carlo for pdes with spatially varying coefficients. *ACM Trans. Graph.*, 41(4), July 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530134. URL <https://doi.org/10.1145/3528223.3530134>.

- [75] Rohan Sawhney, Bailey Miller, Ioannis Gkioulekas, and Keenan Crane. Walk on stars: A grid-free monte carlo method for pdes with neumann boundary conditions. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301. doi: 10.1145/3592398. URL <https://doi.org/10.1145/3592398>.
- [76] Dario Seyb, Eugene d'Eon, Benedikt Bitterli, and Wojciech Jarosz. From microfacets to participating media: A unified theory of light transport with stochastic geometry. *ACM transactions on graphics*, 43(4), 2024.
- [77] Nicholas Sharp and Alec Jacobson. Spelunking the deep: guaranteed queries on general neural implicit surfaces via range analysis. *ACM Trans. Graph.*, 41(4), July 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530155. URL <https://doi.org/10.1145/3528223.3530155>.
- [78] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.*, 41(3), March 2022. ISSN 0730-0301. doi: 10.1145/3507905. URL <https://doi.org/10.1145/3507905>.
- [79] Scott Sheffield. Gaussian free fields for mathematicians. *Probability theory and related fields*, 139(3):521–541, 2007.
- [80] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *Proc. NeurIPS*, 2021.
- [81] Jan Sokolowski and Jean-Paul Zolésio. Introduction to shape optimization. In *Introduction to Shape Optimization: Shape Sensitivity Analysis*, pages 5–12. Springer, 1992.
- [82] Ryusuke Sugimoto, Christopher Batty, and Toshiya Hachisuka. Velocity-based monte carlo fluids. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705250. doi: 10.1145/3641519.3657405. URL <https://doi.org/10.1145/3641519.3657405>.
- [83] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [84] Eric Veach. *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford, CA, USA, 1998. AAI9837162.
- [85] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, page 419–428, New York, NY, USA, 1995. Association for Computing Machinery. ISBN 0897917014. doi: 10.1145/218380.218498. URL <https://doi.org/10.1145/218380.218498>.
- [86] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, page 65–76, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0897918967. doi: 10.1145/258734.258775. URL <https://doi.org/10.1145/258734.258775>.
- [87] Ingo Wald, Sven Woop, Carsten Benthin, Gregory S. Johnson, and Manfred Ernst. Embree: a kernel framework for efficient cpu ray tracing. *ACM Trans.*

- Graph.*, 33(4), July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601199. URL <https://doi.org/10.1145/2601097.2601199>.
- [88] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.
 - [89] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, page 85–92, New York, NY, USA, 1988. Association for Computing Machinery. ISBN 0897912756. doi: 10.1145/54852.378490. URL <https://doi.org/10.1145/54852.378490>.
 - [90] Dongbin Xiu and Daniel M. Tartakovsky. Numerical methods for differential equations in random domains. *SIAM Journal on Scientific Computing*, 28(3):1167–1185, 2006. doi: 10.1137/040613160. URL <https://doi.org/10.1137/040613160>.
 - [91] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in neural information processing systems*, 34:4805–4815, 2021.
 - [92] Ekrem Fatih Yilmazer, Delio Vicini, and Wenzel Jakob. Solving inverse pde problems using grid-free monte carlo estimators. *ACM Trans. Graph.*, 43(6), November 2024. ISSN 0730-0301. doi: 10.1145/3687990. URL <https://doi.org/10.1145/3687990>.
 - [93] Zihan Yu, Lifan Wu, Zhiqian Zhou, and Shuang Zhao. A differential monte carlo solver for the poisson equation. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705250. doi: 10.1145/3641519.3657460. URL <https://doi.org/10.1145/3641519.3657460>.
 - [94] Ziyi Zhang, Nicolas Roussel, and Wenzel Jakob. Many-worlds inverse rendering. *arXiv preprint arXiv:2408.16005*, 2024.
 - [95] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Building volumetric appearance models of fabric using micro ct imaging. *ACM Transactions on Graphics (TOG)*, 30(4):1–10, 2011.