# Exploring GPU-Based Monte-Carlo Methods for Global Illumination and Dynamic Irradiance Field Probes: Theory to Implementation

AUSTIN J. MADDISON, Mahidol University International College, Thailand

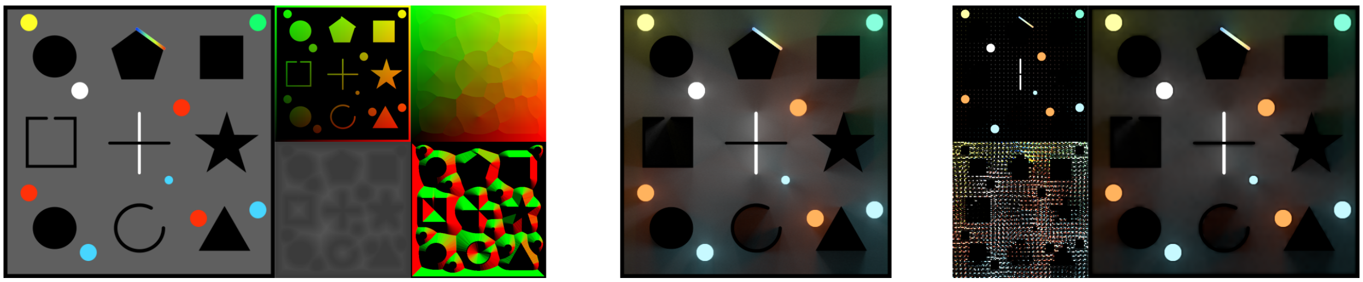


Fig. 1. figure caption

Global illumination (GI) is essential for achieving photorealistic rendering by simulating the indirect interactions of light within a scene. While Monte Carlo ray tracing offers high-fidelity results, its computational cost makes it impractical for real-time applications. This report explores how Monte Carlo-based GI can be adapted for real-time rendering through the use of irradiance field probes. I begin by covering the fundamentals of Monte Carlo ray tracing and the challenges of real-time GI, then introduce irradiance probes as a practical solution to bridge the gap between quality and performance. The implementation is done in a 2D discretized space, using a Signed Distance Field (SDF) representation. While most GI research focuses on 3D scenes, this work explores how similar concepts can be applied in 2D offering potential value for 2D engines aiming for more realistic lighting.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: Global illumination, raytracing, offline rendering, real-time rendering

## 1 INTRODUCTION

Global illumination (GI) is essential for realistic lighting, as it simulates both direct and indirect light interactions within a scene. Unlike direct illumination, GI captures how light reflects, refracts, and diffuses across surfaces, resulting in more cohesive visuals.

This report explores GI algorithms with an emphasis on GPU-based implementations using compute shaders. It begins with Monte Carlo ray tracing, a probabilistic technique foundational to many physically based rendering methods. The goal is to adapt this approach to 2D, using a discretized Signed Distance Field (SDF) instead of traditional 3D geometry.

To support real-time applications, the report then investigates irradiance field probes—an efficient approximation technique that decouples light sampling from scene geometry. This makes it well-suited for dynamic or large environments where full ray tracing is computationally expensive.

I compare the performance and visual quality of irradiance probes with Monte Carlo ray tracing, and conclude by briefly discussing extensions other works that improve the probe system's dynamism and memory efficiency, which is something I can maybe explore in the future.

### 1.1 Report Structure

**Introduction**: Overview of Global Illumination and its significance in realistic rendering (this section).

**Related Work**:

- **Monte Carlo Ray Tracing**: Explanation of the ray tracing method and its role in simulating light transport.
- **Challenges of Real-Time GI**: A discussion on the computational challenges that make GI difficult to perform in real-time.
- **Real-Time GI with Irradiance Probes**: Introduction to irradiance field probes and how they enable real-time indirect lighting.
- **Supercharging Irradiance Fields**: A brief look at extensions to make probes dynamic and space-efficient.

**Approach**:

- **SDF-Based Render Pipeline**: Description of the custom 2D rendering pipeline and use of Signed Distance Fields to represent geometry.
- **Monte Carlo GI in 2D**: Adaptation and implementation of Monte Carlo ray tracing in a 2D SDF environment.
- **Irradiance Probes**: Details on implementing irradiance probes on the GPU for fast and scalable GI.

**Performance Evaluation**:

- **Benchmarks**: Performance measurements of the irradiance probe system vs. traditional Monte Carlo ray tracing.
- **Image Comparisons**: Visual comparisons to evaluate the trade-offs between quality and speed.

**Conclusion**:

- Summary of findings, limitations, and suggestions for future work.

Through this project, I aim to investigate how core GI algorithms work and written for the GPU and how they can be translated into real-time rendering a simplified 2D context.

Author's Contact Information: Austin J. Maddison, austin.mad@student.mahidol.edu, Mahidol University International College, Nakhon Pathom, Nakhon Pathom Province, Thailand.

## 2 RELATED WORK:

Global illumination is the sum of all incoming irradiance at a given point, including both direct and indirect light contributions.

### 2.1 Monte Carlo Ray Tracing

One of the most widely used techniques for simulating global illumination is Monte Carlo ray tracing, a probabilistic method that traces rays from light sources and simulates their interactions with surfaces in a scene. The diffuse global illumination (GI) irradiance $L_0$ can be recursively [Kajiya 1986]. defined as:

$$L_0(p) = L_e(p) + \rho(p) \int_{\Omega} L_0(p', \omega') f_r(p, \omega, \omega')(\omega' \cdot n)\, d\omega' \quad (1)$$

where:

- $L_0(p)$: The outgoing radiance at point $p$.
- $L_e(p)$: The emitted radiance at point $p$.
- $\rho(p)$: The surface reflectance at $p$.
- $f_r(p, \omega, \omega')$: The bidirectional reflectance distribution function (BRDF).
- $\omega'$: The incoming light direction.
- $n$: The surface normal at $p$.
- $\Omega$: The hemisphere above the surface.

However, Monte Carlo ray tracing can be computationally expensive, requiring many samples to achieve noise-free results.

### 2.2 Challenges of Real-Time GI

Global illumination techniques like Monte Carlo ray tracing are too slow for real-time use due to their high computational demands. To speed up GI, two main approaches are used: scene representation proxies (like simplified geometry or voxel grids) and precomputing or caching (such as baked lighting in textures or vertices). While these methods improve performance, they struggle with dynamic scenes, as changes require recalculation of precomputed data.

Irradiance fields fall into the category of caching techniques but offer a sparser and more flexible solution. By decoupling lighting information from scene geometry and storing indirect lighting in a grid or probe-based structure, irradiance fields enable real-time GI approximations. This approach allows for cheaper recalculations, striking a balance between performance and visual fidelity.

### 2.3 Real-Time GI with Irradiance Probes

To address the challenges of real-time global illumination, irradiance probes provide an efficient solution for approximating indirect lighting. Irradiance probes decouple light sampling from scene geometry, enabling the calculation of indirect lighting in a more computationally feasible manner. By storing irradiance values in a grid or field throughout the scene, they allow for quick retrieval of lighting information, significantly reducing the cost of real-time global illumination. This method works particularly well in dynamic or large-scale environments where full ray tracing would be too costly. The trade-off is a reduction in accuracy, but in many cases, irradiance probes offer a good balance between performance and visual fidelity.

### 2.4 Supercharging Irradiance Fields

While irradiance probes offer a practical solution for real-time GI, several advanced techniques can enhance their efficiency and adaptability. Spherical Harmonics (SH) are often used to compactly represent irradiance data, enabling efficient storage and retrieval [Schneider et al. 2017; Sloan 2008]. Visibility information can be incorporated into probes to better account for occlusions, improving the accuracy of indirect lighting [Silvennoinen and Lehtinen 2017]. Dynamic updates with state tracking allow probes to adapt to changes in the scene, ensuring responsiveness in dynamic environments [Datta et al. 2023; Sedláček 2019]. Irregular and variably sparse placement of probes, guided by scene complexity or lighting conditions, can further optimize memory usage and computational overhead [Vardis 2015]. These enhancements collectively improve the quality and performance of irradiance fields in real-time applications.

## 3 APPROACH

### 3.1 SDF-Based Render Pipeline

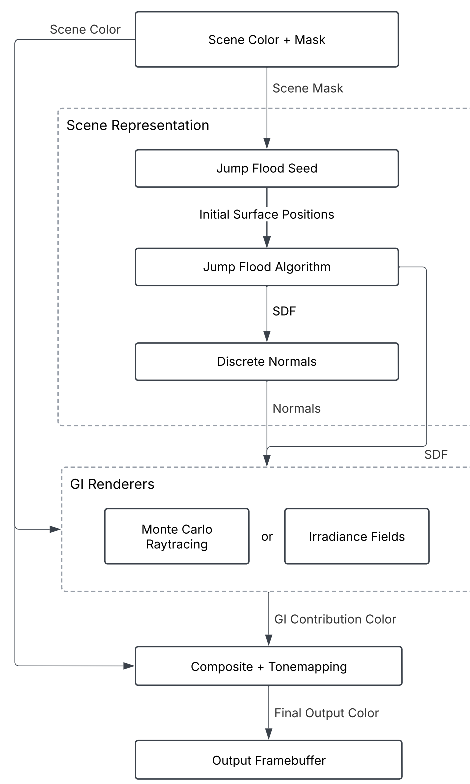



Fig. 2. Overview of the 2D rendering pipeline. The input consists of a color image and a mask, which are processed to generate a Signed Distance Field (SDF) using the Jump Flooding Algorithm. Normals are derived from the SDF, and along with the scene color, are passed to the Global Illumination (GI) renderer. The GI renderer, using either Monte Carlo ray tracing or irradiance fields, computes indirect lighting. Finally, the GI result is composited with the scene color, tonemapped, and rendered to the output framebuffer.

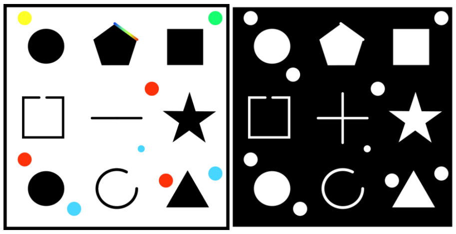


Fig. 3. Left: Scene color image. Right: Scene mask image.

Fig. 4. Left: Surface positions image. Right: Jump flooded scene positions.

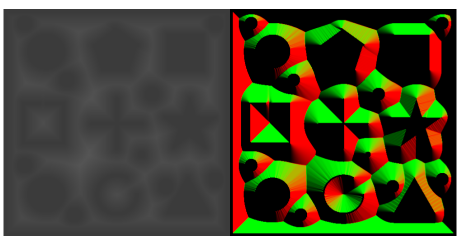


Fig. 5. Left: Signed Distance Field. Right: Normals.

## 3.2 Monte Carlo GI in 2D

We use the 2D buffers shown above to then trace scene using 2D raymarching. This kind of intersection testing is efficient because the SDF provides the minimum distance to all surfaces and at any given position. This allows the ray to skip large sections of the scene while marching. The high level algorithm is the following... (detailed algorithm look into the source code in Section 6).

### *3.2.1 Algorithm*

(1) Initialize ray position at every **scene position** random unit circle direction.
(2) Initialize ray direction with random unit circle direction.
(3) Set total distance and color contribution to zero.
(4) Repeat for a fixed number of bounces:
    (a) Perform ray marching to detect object intersection.
    (b) If no intersection, break the loop.
    (c) Add color contribution based on hit properties.
    (d) Compute new half-unit circle ray direction for the next bounce.
(5) Blend results if using multiple samples. (Accumilation)
(6) Write final color to global illumination output texture.

## 3.3 Irradiance Probes

To implement irradiance probes using a sparse grid, we first define the grid layout and perform ray tracing for each probe position. The results are stored in a map layout for efficient querying during rendering.

*3.3.1 Probe Position Initialization* We define a regular grid of probe positions across the scene, with a parameter $n$ where $n \times n$ is the total number of probes. Each probe stores irradiance data computed using the ray tracing algorithm above into a texture. After the radiance has been computed, it can be queried by any position in the scene to get the interpolated GI irradiance.

### *3.3.2 Algorithm Compute*

(1) Initialize ray position at every **probe position** random unit circle direction.
(2) Initialize ray direction with random unit circle direction.
(3) Set total distance and color contribution to zero.
(4) Repeat for a fixed number of bounces:
    (a) Perform ray marching to detect object intersection.
    (b) If no intersection, break the loop.
    (c) Add color contribution based on hit properties.
    (d) Compute new half-unit circle ray direction for the next bounce.
(5) Blend results if using multiple samples. (Accumilation)
(6) Write final color to probe's texture.

### *3.3.3 Algorithm Query*

(1) For a given scene position, find the nearest probe positions in the sparse grid.
(2) Interpolate the irradiance values from the neighboring probes using a weighted average based on distance.
(3) Apply the interpolated irradiance to compute the indirect lighting contribution for the scene position.

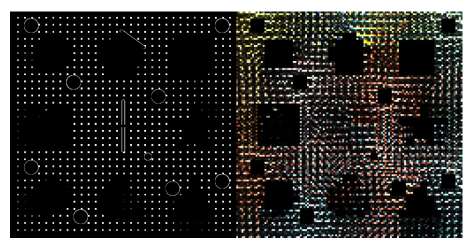


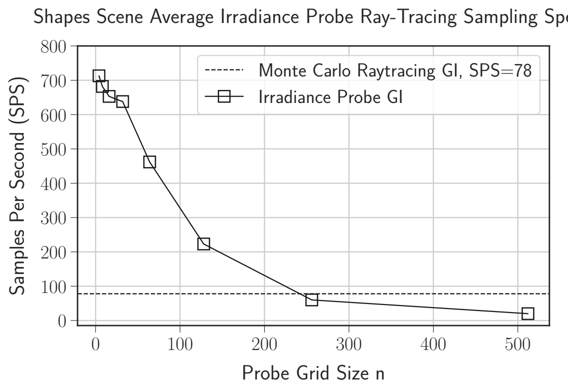
Fig. 6. Left: Irradiance probe positions (marked as the square dots). Right: Irraidance probe irradiance data texture

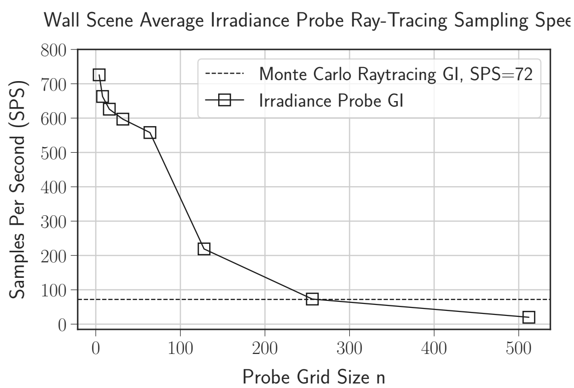
# 4 PERFORMANCE EVALUATION

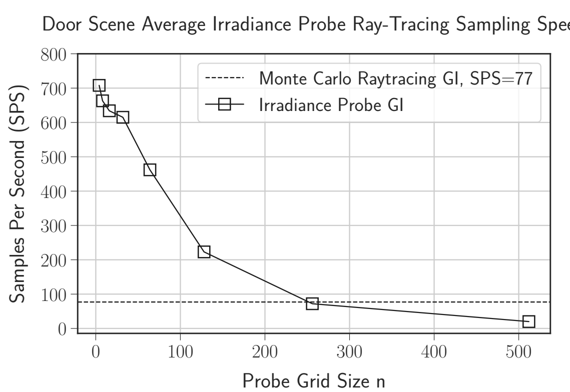
## 4.1 Benchmarks

The performance of both the Monte Carlo ray tracer and the irradiance probes for global illumination was evaluated against different scenes. The metric used was samples per second (sps), where a higher sps indicates better performance. The output resolution was fixed at 512x512, and the scenes were enclosed to ensure that rays reached the maximum bounce limit, which was set to 8.

To analyze the impact of varying the grid size $n$, the irradiance probe system was tested with grid sizes ranging from $n = 4$ to $n = 512$.




As anticipated, irradiance probes exhibited significantly better performance by reducing the number of positions requiring irradiance calculations, with runtime scaling according to the grid size $n^2$. Performance remained consistent across different scenes, which is expected since the enclosed space ensures all rays reach the maximum bounce depth, as they cannot escape. In an open scene, performance differences might emerge as some rays would terminate earlier, reducing the number of required bounces.

## 4.2 Image Comparisons

The images illustrate how GI quality decreases with lower grid resolutions $n$. Image D ($n = 32$) is the minimum acceptable resolution, as lower values show noticeable artifacts. Beyond image F $n = 64$, quality improvements are minimal. Irradiance probes at $n = 64$ were on average 7.4 times faster than Monte Carlo ray tracing.

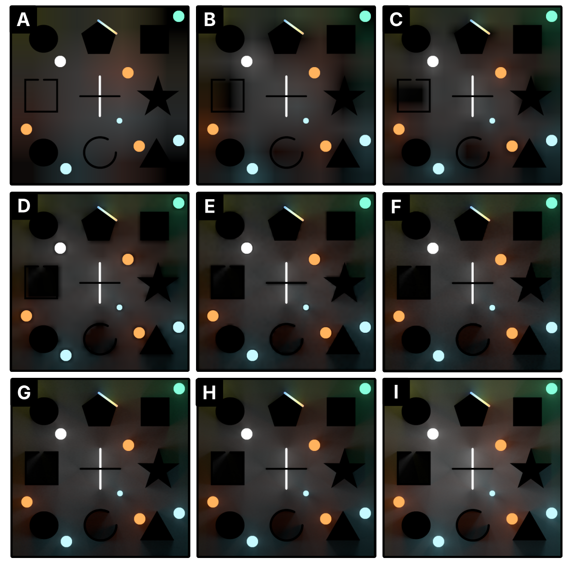



Fig. 7. Shapes Scene: A to I represent irradiance probes with increasing grid resolution (A: n=4, B: n=8, C: n=16, D: n=32, E: n=64, F: n=128, G: n=256, H: n=512), and I represents Monte Carlo Ray Tracing.

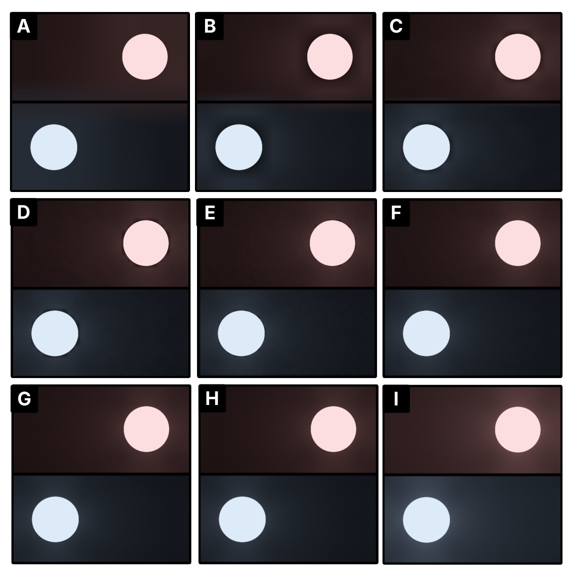



Fig. 8. Wall Scene: A to I represent irradiance probes with increasing grid resolution (A: n=4, B: n=8, C: n=16, D: n=32, E: n=64, F: n=128, G: n=256, H: n=512), and I represents Monte Carlo Ray Tracing.

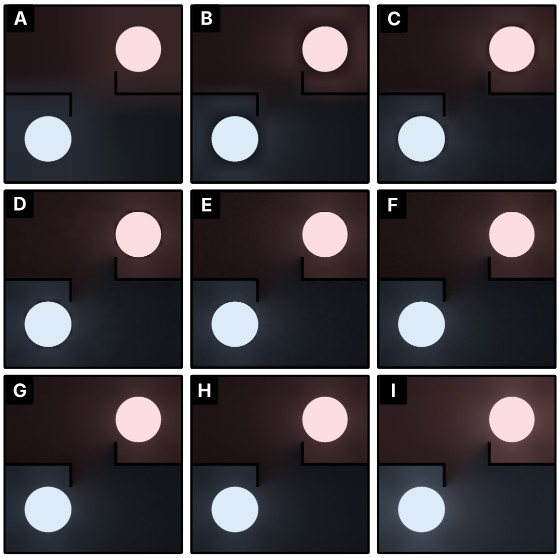



Fig. 9. Door Scene: A to I represent irradiance probes with increasing grid resolution (A: n=4, B: n=8, C: n=16, D: n=32, E: n=64, F: n=128, G: n=256, H: n=512), and I represents Monte Carlo Ray Tracing.

## 5 CONCLUSION

Introducing irradiance fields to query global illumination proves to be faster than Monte Carlo ray-tracing every pixel, but at the cost of accuracy. This trade-off is evident in the benchmarks and image comparisons, where irradiance fields significantly reduce computation time while maintaining acceptable visual quality at higher grid resolutions. However, the reduced accuracy can lead to artifacts such as light leaking or loss of detail in complex lighting scenarios.

Future work for this project could be exploring the techniques that I mention in Section 2.4, where people have extended irradiance probes to be more efficient, flexible and dynamic.

In conclusion, this project provided an opportunity to explore both the theoretical and practical aspects of light transport and its optimization for real-time rendering. Additionally, it facilitated learning how to create advanced GPU programs using compute shaders, random number generation, and other techniques essential for crafting bespoke rendering solutions.

## 6 SOURCE CODE

The full source code and implementation details (although messy) can be found here: https://github.com/AustinMaddison/2D-GI.

## ACKNOWLEDGMENTS

All my friends and peers <3.

## REFERENCES

S. Datta, N. Goli, and J. Zhang. 2023. Adaptive Dynamic Global Illumination. *arXiv preprint arXiv:2301.05125* (2023). https://arxiv.org/abs/2301.05125

James T. Kajiya. 1986. The Rendering Equation. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 143–150. doi:10.1145/15922.15902

A. Schneider, S. Schönborn, B. Egger, L. Frobeen, and T. Vetter. 2017. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV).* https://en.wikipedia.org/wiki/Spherical_harmonic_lighting

Š. Sedláček. 2019. *Real-time Global Illumination using Irradiance Probes.* Master's thesis. Czech Technical University in Prague. https://dcgi.fel.cvut.cz/wp-content/wpallimport-dist/theses/pdf/theses-2019-sedlasi1-thesis.pdf

A. Silvennoinen and J. Lehtinen. 2017. Real-time Global Illumination by Precomputed Local Reconstruction from Sparse Radiance Probes. *ACM Transactions on Graphics* 36, 6 (2017), 230. https://www.cs.mcgill.ca/~ywang411/ProjectPage/Thesis_RadianceProbePlacement.pdf

P. Sloan. 2008. Stupid Spherical Harmonics (SH) Tricks. In *ResearchGate.* https://www.researchgate.net/publication/215506209_Stupid_Spherical_Harmonics_SH_Tricks

K. Vardis. 2015. *Efficient Illumination Algorithms for Global Illumination in Interactive Rendering.* Ph. D. Dissertation. Athens University of Economics and Business. https://graphics.cs.aueb.gr/graphics/docs/PhD_Thesis_Kostas_Vardis_webversion_lowres.pdf

Last modified Sunday 6th April, 2025