Deep Illumination-Driven Light Probe Placement

Andreas Tarasidis

Diploma Thesis

Supervisor: Ioannis Fudos

Ioannina, July 2025

ΤΜΉΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΉΣ

ΠΑΝΕΠΙΣΤΉΜΙΟ ΙΩΑΝΝΊΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UNIVERSITY OF IOANNINA

Dedication

To my mother, father, and brother, who always helped me achieve my goals.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof, Ioannis Fudos, for his invaluable guidance, feedback, and support. As an advisor, he always made time, was extraordinarily patient, and his suggestions were instrumental in the completion of this thesis.

I am also grateful to the friends I've made over the years. Their camaraderie has made my academic journey truly worthwhile.

This journey wouldn't have been possible without them, so finally, but certainly not least, I extend my deepest gratitude to my family. Without their encouragement, boundless support, and unwavering belief in my potential, my works would never have reached fruition.

I dedicate this thesis to them.

Abstract

Realistic lighting is a cornerstone of visually compelling 3D graphics. Unity's Light-Probe system offers an efficient way to capture and interpolate baked Global Illumination (GI) data across dynamic objects in scenes. However, manual placement of light probes in complex scenes is both time-consuming and error-prone, greatly delaying the iteration process when making 3D applications. This thesis presents an automated, deep learning-based approach that predicts per-point importance scores for light probe placement using a PointNet++ -inspired neural network.

We first generate a regular 3D point grid that conforms to the user-defined arbitrarily-shaped bounds of the scene. We sample per-point lighting information, including spherical harmonics, light-, normal-, and RGB- variance, and occlusion factor as well. These features capture important information that drive GI accuracy. The data is then converted into a concise feature vector at each location, used to then train the PointNet++ -style AI model that consumes an arbitrary-length list of such feature vectors and outputs a probability in the range 0-1, depicting how vital it is to place a light probe at each point on the grid.

To deploy in Unity, the trained model is then exported to an .ONNX file and imported via Sentis, the official Unity package for handling AI models inside a Unity Runtime; at edit-time, it ingests per-point scene data and returns per-point importance values. Predicted high-importance locations are then used to populate a Unity LightProbeGroup object, giving developers immediate, visually appropriate probe distributions, with easy to control thresh-holding if higher- or lower-importance locations are desired.

We demonstrate that our AI model generalizes across grid sizes and shapes without retraining, as well as giving immediate results for any scene. Although our evaluation remains mostly qualitative, based on visual inspection of GI results and light-probe placement across a variety of indoor and outdoor scenes, we consistently observe that the generated probe layouts capture important scene light-data with minimal or no manual tweaking. By replacing manual probe placement with a simpler AI-based workflow, artists and developers save time and achieve a faster iteration process throughout the development of a 3D application.

Περίληψη

TODO

List of Figures

Table of Contents

Chapter 1

Introduction

Modern interactive 3D applications, like video games, VR/AR apps, simulators etc., depend on believable lighting interactions with the objects of a 3D scene to achieve the desired visual goals, while trying to maintain real-time frame-rate budgets, typically above 30 Frames per Second (FPS). Achieving visual fidelity and performance can be a difficult task and sometimes impossible with the given hardware specifications of the device. For that reason, modern real-time rendering engines, e.g. Unity, Unreal Engine, Godot and others, depend on a number of methods to balance those metrics.

The illumination of any scene can be split into two very simple categories. Direct Illumination, the light that travels unoccluded from a light source to a surface of an object, is typically handled with techniques like shadow-mapping or screen-space shadows, yielding crisp, high-framerate-capable shadows, but lack in inter-surface light transport situations. In contrast, Indirect Illumination, or Global Illumination (GI), captures light that has bounced or refracted off one or more surfaces, producing soft shadows, color bleeding, and contextually rich shading.

The field-standard for accurate lighting and shadows in a scene is Path-Tracing, a method that tracks every light ray and any interactions it has with the objects of a 3D scene and calculates the resulting color for each pixel of the screen. Such approach remains prohibitively expensive for most interactive applications, so real-time systems employ pre-computation and approximation of the illumination of the scene; static geometry is baked into lightmaps that store per-texel irradiance, while dynamic elements sample from irradiance volumes or light probes, sparse 3D points whose spherical-harmonic coefficients are interpolated at runtime.

Screen-space GI methods typically approximate a limited number of light-ray bounces directly from the camera's depth buffer, but suffer from missing contextual information outside the camera's view frustum and temporal instability. Voxel-based approaches (e.g., cone-tracing through a low resolution 3D grid) enable more dynamic multi-bounce effects at the cost of memory, processing cost and potential blurring of fine detail.

Across all these techniques, the central challenge is allocating a strict millisecond-scale budget to indirect illumination while maintaining consistency across static and dynamic scene content, avoiding visible seams when blending baked and runtime solutions and fitting within GPU memory constraints.

Light probes, in particular, represent a compelling middle-ground, flexible enough to illuminate moving objects without rebaking yet compact enough for real-time evaluation, making their optimal placement a critical factor in any high-quality GI pipeline.

1.1 Related Works

There is an abundance of work in the literature addressing the problem of Global Illumination. These studies aim to achieve realistic lighting in 3D scenes by employing various approaches and techniques, each offering unique advantages and disadvantages, but they share a common goal: to minimize computational cost while maximizing visual fidelity.

1.1.1 Offline Methods

Offline Illumination methods refer to techniques that are not viable for real-time applications and are therefore used only in situations where the importance of high visual fidelity far outweighs the need for computational speed, typically in non-interactive 3D renders, most commonly in movies or pre-rendered scenes. Classic Path-Tracing, first introduced in 1986 (Kajiya 1986), tracks the movement of a photon ray emitted from a source, typically the camera, and simulates physics interactions to calculate the color of each screen pixel accurately. The immense computational cost of path-tracing led to the development of performance improvements, such as the Metropolis Light Transport (MLT) method introduced in 1997 (Veach and Guibas 1997), and variants like bi-directional Path-Trace (Lafortune and Willems 1993), which build on Monte-Carlo algorithms (Lafortune 1996).

1.1.2 Online Methods

non-AI based methods

AI based methods

1.2 Thesis Structure

TODO

Chapter 2

Review of the Literature

theoretical background, what light probes are, technologies used

2.1 Light Probes

TODO

Chapter 3

Our Approach

what we did and how why etc

Chapter 4

Experiments

examples and stuff

Chapter 5

Conclusions and Future Work

conclusions and future works

Bibliography

Kajiya, James T. (Aug. 1986). "The rendering equation". In: *SIGGRAPH Comput. Graph.* 20.4, 143–150. ISSN: 0097-8930. DOI: 10.1145/15886.15902. URL: https://doi.org/10.1145/15886.15902.

Lafortune, E (1996). "Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering". URL: https://web.archive.org/web/20160617231944/http://www.graphics.cornell.edu/~eric/thesis/index.html.

Lafortune, Eric P. and Yves D. Willems (1993). "Bi-directional path tracing". In: *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*. Alvor, Portugal, pp. 145–153. URL: https://graphics.cs.kuleuven.be/publications/BDPT/.

Veach, Eric and Leonidas J. Guibas (1997). "Metropolis light transport". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. USA: ACM Press/Addison-Wesley Publishing Co., 65–76. ISBN: 0897918967. DOI: 10.1145/258734.258775. URL: https://doi.org/10.1145/258734.258775.