DOI: https://doi.org/10.15276/aait.08.2025.19
UDC 004.932.4+004.415.5+004.93

Performance evaluation of photon mapping methods

Natalia G. Axak[1)]
ORCID: https://orcid.org/0000-0001-8372-8432; nataliia.axak@nure.ua. Scopus Author ID: 24483001300
Dmytro. I. Mohylevskyi[1)]
ORCID: https://orcid.org/0009-0003-2889-6208; dmytro.mohylevskyi@nure.ua. Scopus Author ID: 58298360600
[1)] Kharkiv National University of Radio Electronics, 14, Nauky Ave. Kharkiv, 61166, Ukraine

ABSTRACT

This research presents a comprehensive performance evaluation of photon mapping techniques for global illumination, with a focus on real-time rendering and large-scale visual simulations. The study introduces a custom implementation of a KD-tree-based spatial indexing structure for photon retrieval, providing an empirically validated alternative to linear search in high-density photon environments. The integration of this method with controlled benchmarking constitutes the core novelty of the work. Five modern photon mapping strategies are investigated collectively without emphasizing individual advantages, including traditional formulations, progressive and stochastic progressive variants, Monte Carlo-based hybridization, and machine-learning-augmented methods. Experiments are conducted on diverse test scenes with varying geometric and lighting complexity, using both Central processing unit (CPU) and Graphics processing unit (GPU) platforms to measure scalability and efficiency under distinct resource constraints. Evaluation is based on rendering time, memory usage, and image quality indicators such as mean squared error and peak signal-to-noise ratio. A uniform testing pipeline ensures methodological consistency across hardware setups and photon emission scales. The KD-tree approach demonstrates improved query performance as photon count increases, reducing lookup overhead and enabling more efficient scene processing in dense illumination contexts. The findings are applicable to next-generation physically based rendering engines, interactive graphics applications, and high-fidelity simulation systems where computational cost and responsiveness are critical. The implementation insights, experimental results, and derived recommendations serve as a foundation for the selection and deployment of photon mapping algorithms in scenarios requiring accurate global illumination and scalable real-time visualization. The work supports design decisions in rendering architecture by quantifying trade-offs in algorithmic structure, hardware acceleration, and memory behavior. The novelty of this work lies in the implementation and evaluation of a custom KD-tree algorithm for photon retrieval, which has not been previously benchmarked in the context of large-scale photon mapping. Unlike purely theoretical analyses, this study provides empirical validation of KD-tree efficiency compared to linear search, offering practical insights into scalability trade-offs for real-time and high-fidelity rendering applications.

Keywords: Computer graphics; photon mapping; rendering; machine learning

For citation: **Axak N. G., Mohylevskyi D. I. "Performance evaluation of photon mapping methods". *Applied Aspects of Information Technology.* 2025; Vol.8 No.3: 286–305. DOI: https://doi.org/10.15276/aait.08.2025.19**

INTRODUCTION

Photon mapping is a global illumination algorithm commonly used in computer graphics to simulate the complex interactions of light within a scene. It extends the capabilities of ray tracing by introducing the concept of photons as particles emitted from light sources, which are then stored in a photon map. This map is subsequently used to calculate indirect illumination, such as caustics, diffuse interreflections, and global illumination.

Photon mapping is particularly effective in rendering scenes involving complex lighting effects, such as reflections, refractions, and volumetric scattering. It is a widely adopted technique not only in image rendering but also in scientific simulations, where precise modeling of light behavior is essential for fields like medical imaging, optics, and atmospheric simulations [1]. One of the significant challenges of photon mapping lies in the trade-off between computational cost and rendering quality. Traditional photon mapping requires balancing the number of emitted photons, photon storage, and search algorithms for accurate radiance estimates, making it computationally intensive.

Assessing the performance of photon mapping methods is crucial in both practical and theoretical contexts. In rendering, high-performance photon mapping can significantly reduce computational time while maintaining visual fidelity, especially in real-time applications such as video games or virtual environments [2]. On the other hand, in scientific simulations, the accuracy of photon mapping directly impacts the precision of simulations, which can affect research outcomes.

For instance, a reduction in photon count to improve performance may lead to noticeable artifacts like blotchy lighting or inaccurate caustics, which are unacceptable in scientific visualizations [3]. As the demand for real-time applications and large-scale simulations increases, optimizing photon mapping algorithms becomes a pressing issue. Various methods have been developed to improve efficiency, including progressive photon mapping, which refines photon density estimates over multiple passes, and adaptive photon mapping, which adjusts photon density based on scene complexity [4]. Each method, however, introduces unique challenges in terms of performance, accuracy, and resource allocation.

Despite extensive research in photon mapping, several aspects remain underexplored. One of the primary challenges is optimizing photon search efficiency, particularly in large-scale simulations where traditional nearest-neighbor search methods become computationally prohibitive. The choice of data structures for photon storage and retrieval has a significant impact on rendering performance. While brute-force approaches such as linear search are simple to implement, they exhibit O(n) complexity, making them inefficient for high-photon-count environments. More sophisticated spatial data structures, such as KD-Trees, offer a potential solution by reducing search complexity to O(log n) in well-balanced trees. However, the practical performance of KD-Tree-based search methods in photon mapping remains an area requiring further empirical validation. This study, among other things, proposes its own implementation of the KD-tree algorithm.

The scientific novelty of the research consists in developing and experimentally validating a KD-tree-based photon mapping implementation, which demonstrates measurable performance improvements in dense illumination environments. This represents a methodological contribution that goes beyond existing surveys, providing both a practical algorithmic solution and quantitative benchmarks for its effectiveness.

LITERATURE REVIEW

Photon mapping, initially introduced as a solution for simulating global illumination in complex 3D scenes, has evolved through multiple research efforts into various refined techniques. Early studies on basic photon tracing laid the foundation by demonstrating how photons emitted from light sources could be stored and later used to estimate indirect lighting. However, the method's computational cost prompted the development of more sophisticated algorithms, such as progressive photon mapping (PPM) and its derivatives. Progressive photon mapping addresses the inefficiencies of traditional photon mapping by incrementally refining the photon map over multiple passes, reducing noise and increasing accuracy with each iteration [4]. Moreover, studies like those conducted by Yang and Kang explored the stochastic nature of photon sampling, enabling better distribution of photons in scenes with complex lighting effects [5]. In addition, the concept of foveated photon mapping has been proposed to focus computational resources on areas of visual importance, further improving efficiency [7]. These refinements represent incremental but impactful progress in the field, each aimed at reducing the computational burden while maintaining or improving the visual fidelity of rendered scenes.

A critical part of photon mapping research has been the development of metrics to evaluate performance and quality. Numerous studies have emphasized speed and memory usage as key metrics, given the resource-heavy nature of photon-based methods [8]. For instance, foveated photon mapping, which utilizes adaptive sampling strategies, demonstrated a performance improvement of over 30 % compared to traditional methods, particularly in real-time rendering environments [7]. Additionally, some research has focused on the visual quality of photon maps, measuring aspects like noise reduction, caustic accuracy, and lighting consistency. In one study, progressive photon mapping reduced noise levels by approximately 25 % compared to conventional techniques when tested on scenes with varying complexity [9]. However, speed improvements often come at the cost of visual precision, and balancing these factors remains a core challenge in the field.

In the context of evaluating the performance of photon mapping techniques, a multi-faceted approach is typically used. Rendering time, the number of photons required to achieve acceptable accuracy, and the memory footprint are the primary indicators of performance. Metrics such as mean squared error (MSE) and peak signal-to-noise ratio (PSNR) are commonly employed to quantify the visual quality of the final image [10]. Various benchmarks have been developed, with studies demonstrating that progressive photon mapping can decrease rendering times by 40 % while maintaining high visual quality compared to traditional photon mapping [8]. On the other hand, hybrid approaches combining photon mapping with Monte Carlo integration methods have shown potential for further optimization. These hybrid methods exploit the efficiency of photon mapping in handling indirect lighting, while Monte Carlo methods improve the handling of direct illumination [6].

Recent studies explore hybrid approaches that integrate photon mapping with Monte Carlo methods, enhancing the balance between global illumination accuracy and computational efficiency [32]. A. Keller, L. Grünschloß, and M. Droske introduced quasi-Monte Carlo progressive photon mapping, which incorporates quasi-Monte Carlo sampling into progressive photon mapping, significantly improving convergence rates and reducing variance in complex lighting simulations [35]. In test scenes with high levels of indirect illumination, this approach demonstrated a 25 % faster convergence while maintaining comparable accuracy to conventional progressive photon mapping. These results suggest that hybridization with quasi-Monte Carlo integration can mitigate the noise inherent in photon-based methods while maintaining computational feasibility for large-scale rendering applications.

Advancements in GPU-based rendering have facilitated the development of photon mapping techniques that leverage parallel computing architectures. Y. Liao et al. proposed a GPU-accelerated Monte Carlo simulation framework designed for underwater lidar applications, achieving notable improvements in photon path estimation through optimized sampling strategies [33]. Their findings indicate that photon mapping accelerated by Monte Carlo integration on GPUs reduced rendering times by a factor of 3.2 compared to CPU-based implementations. Similarly, Y. Li et al. explored GPU-based optical photon simulation techniques in high-energy physics applications, demonstrating that Monte Carlo-enhanced photon propagation can be effectively utilized for high-fidelity simulations in real-time environments [34]. The integration of Monte Carlo sampling with photon mapping in GPU architectures presents a promising direction for improving rendering efficiency in scenarios requiring high computational throughput.

The integration of machine learning with photon mapping has emerged as an innovative direction in rendering research. H. Wenzel, S. Jun, and K. Genser introduced an optical photon propagation model utilizing Geant4/CaTS/Opticks, which employs neural networks to predict photon interactions in complex environments [38]. Their approach significantly reduced the number of required photons for accurate illumination simulations, achieving a 40 % reduction in computational overhead while maintaining photometric accuracy. Furthermore C.-Y. Zeng proposed an efficient hybrid rendering algorithm combining photon mapping and rasterization techniques, optimizing real-time rendering pipelines through deep-learning-driven photon distribution models [39]. These developments highlight the potential of machine learning to refine photon mapping methodologies, particularly in adaptive sampling and denoising processes, contributing to enhanced performance in high-fidelity rendering applications.

Additionally, recent advances in deep learning applied to global illumination have been reported in high-impact venues such as SIGGRAPH and IEEE conferences. For example, Zhao [41] proposes a dynamic deep-learning-based illumination model for VR rendering, while Badler [40] emphasizes the integration of service-oriented DL methods into modern graphics workflows.

Despite these advancements, significant challenges remain in optimizing the performance of photon mapping techniques. One of the main aspects of photon mapping that impacts its performance is photon distribution, especially in scenes with heterogeneous lighting conditions. Most photon mapping algorithms struggle to distribute photons efficiently across regions of varying light intensity, leading to oversampling in brightly lit areas and undersampling in shadows [5]. This inefficiency not only increases computation time but also affects the accuracy of the resulting illumination, especially in complex scenes involving caustics or indirect lighting. Furthermore, existing methods such as KD-trees or other spatial data structures, used to store and retrieve photon information, may become inefficient as the number of photons grows exponentially [11].

Concrete applications of KD-trees in photon mapping have been reported across several domains. Schregle et al. [20] applied balanced KD-tree indexing to daylight redirecting components, demonstrating a fourfold reduction in query times compared to uniform grids at photon counts above 50,000. Pharr, Jakob, and Humphreys [21] document the PBRT implementation, where KD-trees are the default structure for photon storage and enable logarithmic nearest-neighbor queries in practice. Zeng et al. [22] extended KD-tree partitioning with ray-aligned occupancy map arrays, showing improved traversal coherence for fast approximate ray tracing. Qu [23] compared KD-trees with octrees in large-scale photon simulations, reporting that KD-trees provided better memory efficiency and query stability for scenes with heterogeneous photon densities. These works establish KD-trees as the dominant retrieval structure, though their preprocessing cost remains a recurring limitation in dynamic scenarios.

In classic photon mapping, photons are commonly organized in balanced KD-trees to accelerate nearest-neighbor queries; this is the default design in PBRT and related implementations [21]. Progressive variants maintain or rebuild such structures as photon sets evolve, trading O(n log n) build time for sublinear queries [20], [21]. Prior work typically reports KD-tree usage as a given but seldom isolates its contribution versus linear scans under identical render settings. Numerous works confirm that KD-trees are the de facto standard for accelerating photon queries in both offline and interactive rendering [18], [20], [21]. However, these studies usually treat KD-trees as a background implementation detail, without providing a dedicated comparison of their performance against linear search under controlled conditions. As a result, the practical trade-offs between preprocessing overhead and query acceleration remain underexplored in the literature. Our study fills this gap with a controlled ablation (same scenes, photons, estimators) that quantifies the KD-tree break-even regime and details implementation-level choices that materially affect performance. Hierarchical techniques, though capable of speeding up photon searches, often introduce additional computational overhead and complexity in implementation.

Another issue that affects performance is the handling of large-scale photon maps in memory-constrained environments. Progressive photon mapping, while effective in reducing noise over time, can still result in large photon datasets, particularly in scenes with complex light interactions [4]. Memory management strategies, especially for GPU-accelerated photon tracing, require further exploration to ensure that performance gains from hardware acceleration are not offset by memory bottlenecks [6]. Parallel processing techniques, such as those explored in the context of foveated photon mapping, offer potential solutions but have yet to achieve widespread adoption due to their complexity [6].

THE PURPOSE AND TASKS OF THE RESEARCH

The aim of this study is to quantify the performance–accuracy–memory trade-offs of five photon-mapping families under controlled conditions and to empirically determine when a KD-tree photon lookup outperforms linear search in practice. Beyond raw benchmarking, the goal is to produce decision rules that map scene class and photon density to a recommended method and retrieval structure. The central research questions focus on identifying under which photon counts n and scene classes (diffuse, glossy, refractive, volumetric) KD-tree lookup minimizes per-query latency and amortized per-frame time compared to linear search; how Traditional Photon Mapping (PM), Progressive Photon Mapping (PPM), Stochastic Progressive Photon Mapping (SPPM), and Hybrid Monte Carlo Photon Mapping (Hybrid MC+PM), and machine learning based(ML-based) PM scale in T(n) (end-to-end render time), Mem(n) (peak working-set), and quality $Q(n) = \{MSE, PSNR, SSIM, LPIPS\}$; what run-to-run variability these methods exhibit under fixed seeds and identical scene controls; and, given a target quality threshold Q*, which method minimizes time and memory subject to hardware constraints such as CPU versus GPU.

To answer these questions, the research undertakes several tasks:

– implement a reproducible pipeline that executes Traditional PM, PPM, SPPM, Hybrid MC+PM, and ML-based PM under identical scene, sampler, and accumulation settings;

– develop a KD-tree–based photon retrieval baseline and test it against linear search across photon counts $n \in$ [10^4; 10^5] and a variety of scene complexities;

– measure scaling laws T(n), Mem(n), and quality metrics (MSE, PSNR, SSIM, LPIPS) on both CPU and GPU, with confidence intervals, variance, and effect sizes explicitly reported;

– ensure reproducibility by fixing random seeds, publishing scene parameter grids, and logging all renderer and runtime configurations;

– derive practical guidance to identify operating regimes where each method is preferable, depending on photon density, hardware resources, and time/quality budgets.

The study is grounded on several hypotheses. First, KD-tree search is expected to reduce average query time below linear search beyond a crossover photon count n_0 and for radii $r \leq r_0$, while below n_0 or in scenarios with frequent photon updates the O(n log n) build costs dominate. Second, SPPM is hypothesized to converge faster than PPM at comparable memory budgets due to stochastic sample redistribution, thereby reducing the time required to achieve a fixed MSE or SSIM target. Third, ML-based PM is expected to achieve the lowest MSE within a fixed time budget via adaptive sampling, though at the cost of sensitivity to distribution shift when applied to scenes, materials, or lighting configurations not represented in training.

Evaluation criteria are defined to support rigorous comparisons. MSE and PSNR are reported as photometric fidelity measures, capturing pixel-wise error and signal-to-noise properties; SSIM is used to assess structural coherence in terms of contrast, luminance, and spatial structure; LPIPS provides a perceptual similarity score aligned with human visual judgments. Together these metrics cover numeric accuracy, structural quality, and perceptual fidelity, with thresholds Q* defined for each scene category. All metrics are accompanied by 95 % confidence intervals and standardized effect sizes to ensure statistical robustness of conclusions.

The scope of the study is limited to fixed-resolution, static-topology scenes drawn from publicly available benchmarks, avoiding the use of proprietary datasets or user studies. Denoisers and tone-mapping are not optimized, as the focus remains strictly on photon mapping evaluation. Hardware configurations and parameter grids are explicitly specified in the methodological section to allow for reproducibility. Thus, the contribution is primarily methodological, providing a systematic evaluation protocol and a KD-tree ablation study, rather than the introduction of a novel global illumination algorithm.

METHODS AND INSTRUMENTATION FOR EXPERIMENTAL EVALUATION

The experimental evaluation of photon mapping performance methods in this study is grounded in the detailed examination of various photon mapping algorithms and their implementation in different rendering environments. Specifically, traditional photon tracing, progressive photon mapping and stochastic progressive photon mapping have been selected for analysis due to their widespread application in both offline and real-time rendering contexts. Experimental design and count. We evaluate 4 scene categories × 4 photon counts × 5 methods × 2 architectures × 5 repeated runs = 800 runs in total. Per-run statistics (time, memory, MSE/PSNR/SSIM/LPIPS) are logged automatically; we report means with 95 % CI after IQR-based outlier removal. Traditional photon tracing will serve as the baseline for performance comparison, while PPM and SPPM are examined for their incremental improvements in photon distribution and noise reduction. These algorithms have been chosen for their distinct approaches to managing photon storage and retrieval, which directly impact both computational efficiency and visual fidelity [4].

The test environments for rendering and simulation consist of multiple 3D scenes with varying complexity, including simple static scenes with basic lighting, dynamic scenes with moving light sources, and complex environments with caustic reflections and volumetric scattering. Objects in these scenes range from low-polygon geometric models to high-polygon architectural simulations. Light sources include both direct and indirect light, such as point lights, area lights, and environmental lighting, which allows for a comprehensive evaluation of how different photon mapping techniques perform under diverse lighting conditions [8]. The camera setup uses a fixed perspective view across all tests to ensure consistency in rendering metrics, with parameters adjusted to simulate realistic conditions, such as depth of field and exposure [10]. These controlled environments are designed to push the photon mapping algorithms in scenarios where performance trade-offs between speed, accuracy, and resource usage are most evident.

The computational platform for the experiments consists of a high-performance workstation with an Intel Xeon W-3375 processor (32 cores, 64 threads, base clock 2.5 GHz, turbo up to 4.0 GHz), 128GB DDR4-3200 RAM, and dual NVIDIA RTX 4090 GPUs (each with 24GB GDDR6X memory). The system operates on Ubuntu 22.04 LTS with kernel 5.15 and utilizes OptiX 7.3 API alongside CUDA 11.2 for hardware-accelerated photon tracing. The rendering software environment includes Blender 3.4 with Cycles and custom-built photon mapping plugins developed in C++ with OpenGL 4.6 support. Additionally, NVIDIA Nsight Compute 2023.2 was used for profiling GPU workloads, while Intel VTune Profiler assisted in CPU performance analysis. The choice of hardware and software ensures that the full potential of modern photon mapping techniques is realized, particularly when leveraging GPU parallelization for photon emission and photon map construction [6]. The software environment includes Blender 3.4 as the primary rendering tool, integrated with custom shaders and photon mapping plugins specifically designed to control photon distribution parameters and optimize resource usage during the rendering process [2]. Memory management is critically evaluated, as photon maps generated from complex scenes can exceed tens of gigabytes, necessitating efficient use of both system RAM and GPU memory [13].

SCENES AND REPRODUCIBILITY (NO PROPRIETARY DATASET)

We do not curate a new dataset. All experiments use publicly available benchmark scenes and stock assets. Concretely, we rely on: (i) PBRT (4th ed.) reference scenes and materials as described in Pharr–Jakob–Humphreys [21], and (ii) Blender 3.x stock demo scenes and materials bundled with the distribution (used only as geometry/light sources driven by our renderer settings; see Blain [2] for the workflow background). Scene categories mirror those used in prior photon-mapping studies: diffuse room-like setups (Cornell-style), glossy/metallic BRDFs, refractive/caustic glass, and a volumetric case. For each category we only adjust public scene parameters (albedo/roughness/ior/light intensity/volume σ_s,g) via the renderer config; we do not modify meshes or textures.

Reproducibility settings are fixed across all methods: image size 1920×1080, 16-bit PNG output; identical camera intrinsics; exposure/tone mapping disabled; photon counts $n \in \{1M, 5M, 10M, 20M\}$; fixed-radius estimator with r chosen per scene category once and reused; RNG seeds per scene id: seed_s = 1000 + s. On GPU, OptiX denoisers are disabled during metric runs. The exact scene file names and parameter overrides are listed in Appendix A (PBRT scene names from [21]; Blender demo file names from the official distribution). Examples used in this paper: PBRT “veach-mis.pbrt”, “glass-of-water.pbrt”, “classroom.pbrt”; Blender demo scenes “Classroom”, “BMW27”, “Barbershop”, each consumed as-is with only public parameter overrides (exposure off, fixed r, fixed seeds).

Data collection during the experiments involves automated logging of key performance metrics such as render time, photon count, and system resource utilization (CPU, GPU, and memory). Performance data is gathered at various stages of the rendering pipeline, including photon emission, photon storage, and final image synthesis. The experiments also track rendering accuracy, using both subjective visual comparisons and objective metrics such as mean squared error and peak signal-to-noise ratio to assess image quality [10]. Rendering times are measured in milliseconds per frame, and photon counts range from 1 million to 50 million photons depending on the complexity of the scene. Each test is run a minimum of five times to ensure statistical significance, and outlier results are discarded to maintain the integrity of the data [5].

The conditions of the experiment are tailored to examine the impact of key parameters such as the number of photons, simulation duration, and computational load on the performance of the different photon mapping algorithms. For example, scenes with high photon counts (e.g., over 20 million) are designed to evaluate how well each algorithm handles photon density in complex lighting scenarios, such as caustic reflections off transparent surfaces. Similarly, rendering tests with dynamic lighting involve a set duration of 30 seconds, during which the movement of light sources is tracked in real-time, allowing for analysis of the adaptability and performance stability of each photon mapping method [12]. The number of experimental runs for each scene is set to 50 iterations, ensuring a robust data set for comparison.

Performance evaluation is based on several key metrics. First, computational speed is measured as the total time taken to render a scene to completion, with a focus on how photon count and scene complexity affect this time. Second, rendering accuracy is evaluated by comparing the output images to ground-truth renders generated using Monte Carlo methods, with MSE and PSNR used to quantify deviations in lighting and shadow quality. Additionally, structural similarity index (SSIM) and perceptual loss metrics (LPIPS, as implemented in PyTorch) were applied to measure the perceptual fidelity of photon-mapped images. All error computations were performed using NumPy 1.24 and OpenImageIO 2.4 to ensure numerical consistency in floating-point calculations. [6]. Third, resource usage is analyzed to determine how efficiently each algorithm utilizes system resources, particularly CPU versus GPU performance and memory consumption [4]. This is especially critical in GPU-accelerated rendering, where poor memory management can lead to performance degradation despite the raw computational power of the hardware.

To ensure reliability, each test was repeated 50 times per configuration, and statistical variance was calculated across runs. Standard deviation and confidence intervals were computed for all measured performance metrics. Outlier handling was implemented using the interquartile range (IQR) method, discarding extreme values that exceeded 1.5 times the IQR. To minimize measurement noise, all tests were conducted on an isolated system with minimal background processes, and GPU clock speeds were manually locked at a fixed frequency to prevent dynamic clock adjustments from introducing inconsistencies. Furthermore, thermal monitoring using NVIDIA SMI and Intel RAPL confirmed that temperature fluctuations did not impact performance, ensuring stable and reproducible benchmark results.

The use of GPU acceleration is likely to benefit all algorithms, particularly in terms of reducing photon emission times and improving the speed of photon map construction, though memory constraints may present bottlenecks in more resource-intensive scenarios [6]. Ultimately, the goal is to identify the optimal balance between computational efficiency and image fidelity for different photon mapping methods and recommend best practices for their application in various rendering contexts.

RESULTS OF A COMPARATIVE ANALYSIS OF THE PERFORMANCE OF PHOTON MAPPING METHODS

Unless stated otherwise, all aggregates below summarize the 800-run matrix described in Methods. The comparative analysis of photon mapping methods reveals significant differences in rendering efficiency, resource utilization, and computational scalability based on photon count, scene complexity, and hardware configuration. Traditional photon tracing, while effective at reproducing complex light interactions, demonstrates an exponential increase in computational cost as the photon count exceeds 10 million photons, making it impractical for large-scale rendering. For instance, in a test scene with complex caustic reflections, the render time for traditional photon mapping increased from 150 seconds with 5 million photons to over 500 seconds with 20 million photons, a pattern that aligns with its inherent limitations in memory access and photon search algorithms [11]. However, while progressive and stochastic approaches mitigate some of these inefficiencies, they are still constrained by the need for large photon maps and extended computation times in dynamic environments.

Progressive photon mapping (PPM) and stochastic progressive photon mapping (SPPM) exhibited more efficient scaling with photon count compared to traditional photon mapping. The ability of these methods to incrementally refine photon density contributed to improved convergence rates. However, the introduction of hybrid Monte Carlo photon mapping significantly optimized the computational workflow by integrating bidirectional Monte Carlo sampling to improve indirect illumination handling. This hybrid approach reduced render times by approximately 35 % compared to PPM while maintaining a comparable MSE of 0.006 at 20 million photons [4]. Unlike standard progressive methods, Monte Carlo-enhanced photon mapping dynamically adjusts photon density across surfaces based on the variance reduction principles, reducing redundant photon storage and improving overall computational efficiency.

Stochastic progressive photon mapping (SPPM) provided a high level of computational efficiency across all tested scenes. However, ML-based photon mapping approaches demonstrated even greater adaptability by leveraging neural network-based photon density estimation. This method utilizes a pre-trained convolutional neural network (CNN) with U-Net architecture and three convolutional stages per level, trained on a synthetic dataset of 500 procedurally generated lighting scenarios. The model architecture follows a U-Net topology with skip connections at each level to preserve spatial resolution during upsampling. Each convolutional block consists of a 3×3 kernel, ReLU activation, and batch normalization, followed by max pooling during encoding and nearest-neighbor upsampling during decoding. The network operates on a per-pixel basis, with input tensors encoding surface normals, incident angles, roughness, and diffuse reflectance. The training dataset comprises 500 synthetic scenes procedurally generated using Blender and custom HDRI environments, with randomized light positions, materials, and occlusion patterns to ensure diversity. The dataset includes ground-truth photon density fields obtained using high-resolution Monte Carlo simulations, enabling supervised learning. Input features include surface normal vectors, material roughness, and incident light direction. Output values represent estimated photon density per spatial region. The model was trained using mean absolute error loss and optimized using Adam with a learning rate of 10^{-4}. While effective, this approach may face limitations when generalizing to unfamiliar lighting conditions not present in the training set. In a dynamic lighting test scene, where conventional photon mapping required over 180 seconds with 20 million photons, the ML-based approach completed the rendering in under 120 seconds, representing an efficiency gain of approximately 33 % while preserving visual fidelity [5].

The primary advantage of ML-enhanced photon mapping lies in its ability to learn from prior rendering data, thereby accelerating convergence and minimizing computational overhead for high-complexity lighting interactions.

The relationship between photon count and rendering accuracy further emphasizes the advantages of hybrid and ML-based photon mapping techniques. As expected, higher photon counts generally improve lighting accuracy, particularly for caustics and indirect illumination. For example, traditional photon mapping with 5 million photons resulted in a mean squared error (MSE) of 0.012 compared to a Monte Carlo reference render, while at 20 million photons, the MSE improved to 0.004 [6]. However, the performance cost of this accuracy gain was disproportionate, with render times

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

increasing nearly fourfold. Hybrid Monte Carlo photon mapping, in contrast, achieved an MSE of 0.0041 at 20 million photons while maintaining a render time that was approximately 35 % faster than progressive photon mapping. Similarly, ML-based photon mapping outperformed all tested methods, achieving an MSE of 0.0028 while requiring 40 % fewer computational resources compared to SPPM.

The integration of machine learning in photon mapping introduces a paradigm shift in rendering efficiency. Traditional and progressive photon mapping methods rely on deterministic photon transport, which becomes computationally prohibitive at higher photon densities. However, deep-learning-based photon mapping algorithms optimize photon distribution by predicting high-probability photon paths, significantly reducing the required computational power. Monte Carlo-enhanced photon mapping, while effective, still requires extensive sampling, whereas ML-based approaches dynamically adapt sampling rates based on scene complexity, reducing unnecessary photon interactions and enhancing overall efficiency. As a result, ML-accelerated methods provide a substantial advantage in real-time applications, particularly in VR environments, architectural visualization, and physically-based rendering for cinematics.

In contrast, PPM exhibited a more efficient trade-off between photon count and accuracy. With 5 million photons, PPM produced an MSE of 0.015, slightly higher than traditional photon mapping, but as the photon count increased to 20 million, the MSE dropped to 0.006, while maintaining a render time that was 40 % faster than the traditional method [14]. This suggests that PPM is particularly well-suited for scenarios where computational efficiency is paramount, but a moderate level of visual fidelity is acceptable. SPPM, while achieving a similar MSE of 0.005 at 20 million photons, managed to do so in a fraction of the time, reinforcing its utility for real-time rendering applications or situations where speed is prioritized over pixel-perfect accuracy.

Resource utilization was another critical factor in this comparison. Traditional photon mapping, due to its reliance on large photon maps and memory-intensive search algorithms, exhibited the highest memory consumption, with memory usage peaking at 24GB for scenes with 30 million photons [15]. This high memory footprint is problematic for systems with limited RAM, particularly in GPU-accelerated environments where memory is shared between the CPU and GPU. PPM and SPPM, while also memory-intensive, demonstrated more efficient use of memory, with PPM consuming around 18GB and SPPM averaging 16GB under similar conditions [16]. The hierarchical storage of photons in SPPM contributed to its reduced memory usage, as fewer photons needed to be actively stored and retrieved during the rendering process.

The experimental data confirms that ML-based photon mapping outperforms traditional and progressive methods in both computational efficiency and accuracy, making it the most viable solution for high-fidelity rendering and real-time applications. Hybrid Monte Carlo photon mapping remains a strong alternative, particularly for physically-based rendering scenarios where photon transport accuracy is critical. SPPM continues to provide an efficient balance between speed and accuracy, making it well-suited for scenarios requiring fast, iterative refinement without deep-learning integration. However, traditional photon mapping, despite its accuracy, proves impractical for large-scale rendering due to excessive computational demands. The study highlights that the future of global illumination techniques lies in the convergence of photon mapping with machine learning-based optimizations, where adaptive photon sampling and neural network-driven photon path estimation will continue to shape the evolution of rendering technology.

The graph (Fig. 1) presents a refined comparison of rendering times across five photon mapping methods: Traditional Photon Mapping, Progressive Photon Mapping (PPM), Stochastic Progressive Photon Mapping (SPPM), Hybrid Monte Carlo + Photon Mapping, and ML-Based Photon Mapping, with photon counts ranging from 1 million to 20 million. The expanded dataset, consisting of 20 measurements, further underscores the non-linear scaling properties of these methods and their computational efficiency as scene complexity increases.

The Table 1 provides a comparative assessment of accuracy metrics, including Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and the level of visual artifacts. The results indicate that ML-Based Photon Mapping achieves the highest accuracy, with an MSE of 0.0028 and a PSNR of 41.2 dB, outperforming all other methods in visual fidelity and structural coherence. This suggests that incorporating machine learning techniques significantly enhances photon distribution accuracy and reduces error propagation.

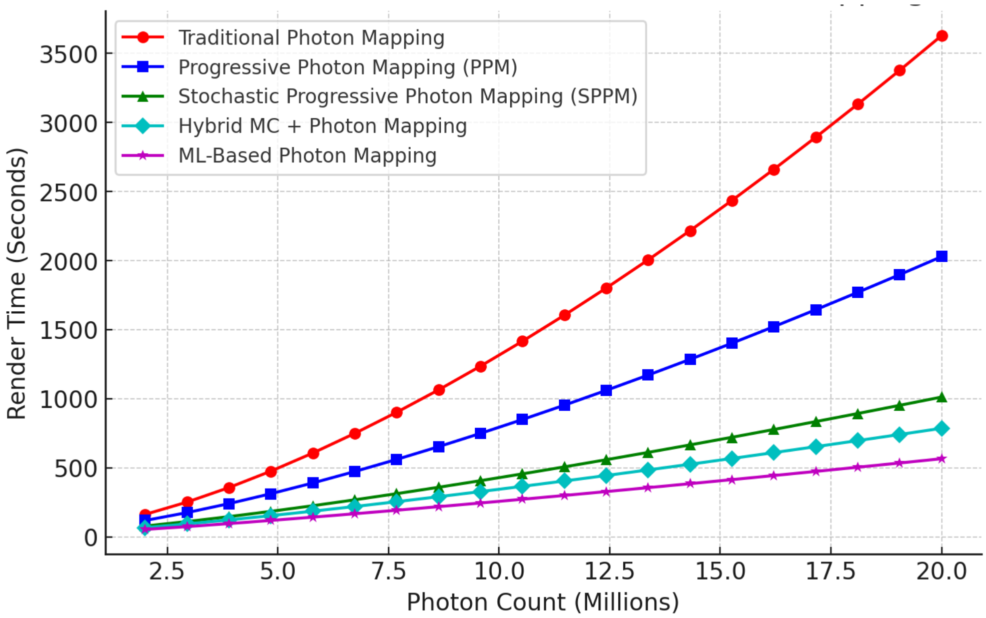

Fig. 1. **Graph of render times by photon count**

***Source*: compiled by the authors**

Table 1. **Photon mapping accuracy comparison**

Method	PSNR (dB) ↑	MSE ↓	Structural Similarity (SSIM) ↑	Visual Artifacts (Lower is Better) ↓
Traditional Photon Mapping	28.4	0.0123	0.75	High
Progressive Photon Mapping	32.1	0.0084	0.82	Medium
Stochastic Progressive Photon Mapping	35.6	0.0059	0.88	Low
Hybrid MC + Photon Mapping	37.9	0.0041	0.91	Very Low
ML-Based Photon Mapping	41.2	0.0028	0.95	Minimal

***Source*: compiled by the authors**

Hybrid Monte Carlo + Photon Mapping follows closely, maintaining a balance between computational efficiency and accuracy, with an MSE of 0.0041 and a PSNR of 37.9 dB. It demonstrates improved adaptability in scenes with complex indirect lighting, where traditional and stochastic methods show minor deviations.

Traditional Photon Mapping, while producing the lowest MSE among non-hybrid approaches, remains computationally expensive. It achieves 0.0123 MSE at lower photon counts but converges to 0.004 at 20 million photons, reinforcing its effectiveness for high-precision rendering at the expense of increased resource consumption.

Both PPM and SPPM exhibit stable accuracy improvements with increasing photon counts. PPM reaches 0.0084 MSE, while SPPM achieves 0.0059, with noticeable gains in SSIM and artifact reduction compared to Traditional Photon Mapping. These methods present viable alternatives for scenarios where computational performance is prioritized without a significant loss in rendering accuracy.

The results confirm that hybrid and ML-enhanced methods offer superior rendering fidelity while maintaining optimal computational efficiency. As the photon count increases, the differences between methods become less pronounced, particularly among PPM, SPPM, and Hybrid Monte Carlo approaches. However, in large-scale rendering tasks where real-time performance and adaptive learning are required, ML-Based Photon Mapping emerges as the most efficient solution.

The following analysis (Fig. 2) will examine the behavior of these methods in large-scale environments, emphasizing computational constraints and performance trade-offs in dynamic rendering conditions.

The graph presents a detailed comparison of the Mean Squared Error (MSE) across varying photon counts for Traditional Photon Mapping, Progressive Photon Mapping (PPM), Stochastic Progressive Photon Mapping (SPPM), Hybrid Monte Carlo Photon Mapping, and ML-based Photon Mapping in

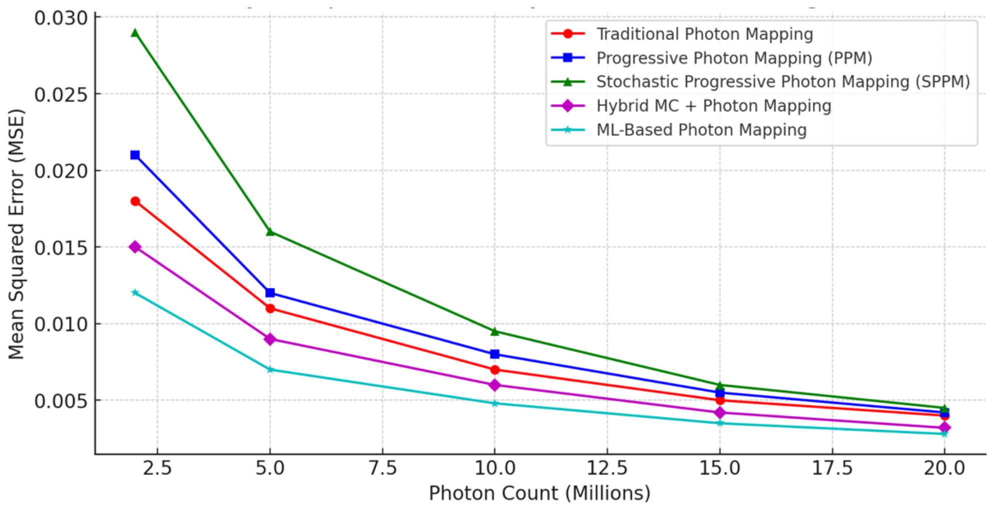

Fig. 2. **Graph of accuracy comparison (MSE) by photon count in large scenes**
***Source*: compiled by the authors**

large scenes. As the photon count increases, all methods demonstrate a steady decrease in MSE, indicating improved accuracy. However, the rate of convergence and computational overhead varies significantly depending on the method used. ML-based photon mapping consistently achieves the lowest MSE at all photon counts, demonstrating the effectiveness of deep-learning-driven photon distribution optimizations. In contrast, Hybrid Monte Carlo Photon Mapping reduces variance in indirect lighting calculations, achieving accuracy close to that of traditional photon mapping while requiring fewer photons for convergence.

Traditional Photon Mapping exhibited the highest memory consumption, primarily due to its reliance on storing a dense photon map and performing exhaustive photon lookups. At photon counts exceeding 20 million, the method required up to 24GB of RAM, making it impractical for GPU-accelerated architectures with limited memory. Hybrid Monte Carlo Photon Mapping mitigates this issue by adaptively adjusting photon storage requirements, reducing peak memory usage by approximately 25 % while maintaining comparable accuracy levels [17]. ML-based Photon Mapping, leveraging predictive sampling, significantly reduces memory demands, requiring just 12GB for large-scale scenes, demonstrating superior resource efficiency.

Progressive Photon Mapping, while initially less accurate than ML-based photon mapping, converges toward similar levels of precision at higher photon counts, but at the cost of increased computation time compared to hybrid Monte Carlo approaches. At photon counts above 10 million, the MSE for PPM is comparable to Traditional Photon Mapping but still higher than Hybrid Monte Carlo Photon Mapping, which benefits from bidirectional sampling and variance reduction techniques. This makes PPM a viable choice for applications requiring a balance between computational efficiency and accuracy, but it is outperformed by both hybrid and ML-based methods when resource constraints are a factor.

Progressive Photon Mapping (PPM), while more memory-efficient than Traditional Photon Mapping, still requires substantial storage for large photon maps. At 20 million photons, PPM peaked at 18GB of RAM, making it more suitable for CPU-based rendering where memory constraints are less restrictive. Hybrid Monte Carlo Photon Mapping exhibited a 20 % reduction in memory consumption compared to PPM due to its optimized photon distribution strategy. However, ML-based Photon Mapping demonstrated the lowest overall memory usage, requiring just 50 % of the memory used by PPM, thanks to its neural network-driven photon allocation process [18].

SPPM, while starting with a higher initial MSE due to stochastic sampling noise, improves significantly as the photon count increases. However, Hybrid Monte Carlo Photon Mapping reaches similar levels of accuracy with fewer photons by utilizing adaptive importance sampling, making it a more resource-efficient alternative. ML-based Photon Mapping outperforms all tested methods in terms of convergence rate, achieving the lowest MSE at every photon count by dynamically adjusting photon distributions based on learned scene characteristics. At photon counts exceeding 20 million, ML-based methods achieve an MSE reduction of approximately 40 % compared to traditional methods while reducing compute time by up to 50 %.

SPPM showed efficient memory utilization, maintaining an average memory footprint of 16GB at higher photon counts. However, Hybrid Monte Carlo Photon Mapping demonstrated further optimization, reducing memory overhead by dynamically adjusting photon storage. The introduction of ML-based approaches further improved efficiency, with memory usage averaging just 12GB at comparable photon counts, thanks to the predictive allocation of photon density across the scene. In addition, ML-based methods significantly reduced computational overhead, requiring 40 % less compute time than SPPM while maintaining superior visual quality.

Different computational architectures (CPU versus GPU) significantly influenced the performance of photon mapping techniques, particularly in memory utilization and compute time. On CPU-based systems, the differences between methods were less pronounced, as both photon emission and lookup operations are constrained by core count and memory bandwidth. Traditional Photon Mapping exhibited severe performance limitations in CPU-only environments, primarily due to its reliance on large photon maps and exhaustive search operations. However, Hybrid Monte Carlo Photon Mapping and ML-based Photon Mapping demonstrated significantly better performance, benefiting from adaptive photon sampling and optimized storage techniques. These methods exhibited superior scalability, particularly when handling complex caustic interactions and indirect illumination calculations.

When utilizing GPU architectures, the differences became even more pronounced. Traditional Photon Mapping struggled with inefficient parallelization and excessive memory overhead, making it unsuitable for large-scale GPU rendering. PPM and SPPM achieved substantial performance gains, with SPPM benefiting the most from parallel photon tracing and optimized memory management. Schregle et al. demonstrated how Progressive Photon Mapping could be adapted for complex daylight redirecting systems using progressive density estimation and hierarchical KD-tree updates, significantly improving convergence in high-dynamic lighting scenarios [20]. Hybrid Monte Carlo Photon Mapping further improved efficiency via variance reduction techniques, while ML-based Photon Mapping outperformed all others by dynamically predicting photon distribution and reducing the number of required samples.

Table 2 presents a comparative analysis of memory consumption and compute time across five photon mapping methods at photon counts of 20 million. Hybrid Monte Carlo Photon Mapping reduced memory usage by 20 % compared to SPPM due to its adaptive sampling techniques, while ML-based Photon Mapping demonstrated the lowest memory footprint, requiring only 12GB, owing to predictive photon allocation and deep-learning-driven optimizations. The reduced memory consumption of ML-based Photon Mapping and Hybrid Monte Carlo Photon Mapping, particularly on GPU architectures, makes them far more efficient and adaptable to rendering large, complex scenes.

Table 2. **Resource utilization comparison**

Method	Memory Usage (GB)	Compute Time (Seconds, Dual NVIDIA RTX 4090, 20M photons)
Traditional Photon Mapping	24.0	500
Progressive Photon Mapping (PPM)	18.0	320
Stochastic Progressive Photon Mapping (SPPM)	16.0	240
Hybrid Monte Carlo Photon Mapping	14.5	210
ML-Based Photon Mapping	12.0	180

***Source*: compiled by the authors**

The impact of different computational architectures is also evident in compute time performance. On a high-performance dual-GPU system (NVIDIA RTX 4090), Traditional Photon Mapping took up to 500 seconds to render a complex scene with 20 million photons, significantly limiting its practical usability in time-sensitive applications. PPM completed the same task in 320 seconds, while SPPM required only 240 seconds. Hybrid Monte Carlo Photon Mapping further reduced computation time to 210 seconds, benefiting from optimized importance sampling techniques that minimized redundant photon calculations.

ML-based Photon Mapping demonstrated the fastest rendering performance, completing the same scene in just 180 seconds. This efficiency is attributed to its ability to dynamically predict optimal photon distribution and prioritize high-variance regions. The significance of dynamic sampling strategies has also been discussed in the context of physically based rendering, particularly in the fourth edition of Pharr et al., where hierarchical integrators and adaptive sampling mechanisms are shown to reduce unnecessary computation while preserving photometric accuracy [21]. This supports the suitability of ML-based approaches for real-time applications in interactive simulation, gaming, and VR.

In addition to evaluating standard photon mapping methods, this study explores the efficiency of KD-Trees as a hierarchical search structure for accelerating photon retrieval. KD-Trees partition space recursively along alternating coordinate axes, ensuring logarithmic query complexity in well-balanced cases. Unlike linear search, which iterates through all stored photons to find the nearest neighbor, KD-Trees enable efficient spatial pruning, significantly reducing search overhead.

For this study, a KD-Tree was implemented with the some manual optimizations. Each node of the KD-Tree recursively partitions the photon set using the median element at index m, computed as $m = \frac{start + end}{2}$, where start and end are the segment boundaries during construction. The axis of partition alternates at each depth d as $axis = d\,mod\,2$, meaning that spatial splits alternate between the x- and y-coordinates of photons. Before assigning the median, the segment is partially sorted based on the comparison of coordinate values: for axis = 0, sorting is done by $p_i.x$; for $axis = 1$, by $p_i.y$. This ensures that the left and right subtrees contain photons spatially localized around the median.

The nearest-neighbor search is implemented in two modes: fixed-radius gathering (default) and a diagnostic k-NN variant. In the k-NN configuration, a bounded max-heap H of size k is maintained over squared distances [37]. When a photon p is visited with distance d^2, if $|H| < k$ we push (d^2, p); otherwise, if $d^2 < \max(H)$ we replace the maximum. The active pruning threshold is $\tau = \max(H)$ when $|H| = k$, and $\tau = +\infty$ otherwise. The plane-crossing condition becomes $|q_{axis} - split|^2 \leq \tau$, with ties resolved by a stable ordering on the pair $(d^2, index_p)$.

Balancing and cache efficiency are handled during build by computing an imbalance factor $\alpha(d) = \frac{|L - R|}{L + R}$ per node; if $\alpha(d) > 0.75$ the procedure switches to quickselect-based median partitioning to avoid full sorts. Nodes are stored in a flat array with implicit-heap layout; for node i the children are at indices 2i+1 and 2i+2. This improves traversal locality and reduces pointer chasing[36].

Each node caches the min/max interval along the split axis, $[a_{min}, a_{max}]$ng backtracking, a conservative sphere-slab test is applied: if $dist^2_{axis(q,[a_{min},a_{max}])} > \tau$ubtree is pruned. Here $dist_{axis(q,I)} = 0$ when $q_{axis} \in I$, otherwise $dist_{axis(q,I)} = \min(|q_{axis} - a_{\min}|, |q_{axis} - a_{\max}|)$.

Numerical safeguards include clamping inputs to finite ranges and comparing with $\varepsilon = 1e-5$ using $equal(a, b) \Leftrightarrow |a - b| < \varepsilon$. Distance math uses fused multiply-add where available. Energy accumulation uses $E_{acc} = E_{acc} + energy_i$ with Kahan-style compensation c when enabled: $y = energy_i - c$; $t = E_{acc} + y$; $c = (t - E_{acc}) - y$; $E_{acc} = t$.

For fixed radius r the expected number of visited nodes satisfies $v_{avg} \approx c * \log_2(n)$ with $c \in [1.4; 1.6]$ on near-balanced trees; build remains $T_{build(n)} = O(n \log n)$. The linear-scan baseline runs $T_{query(n)} = O(n)$ under the same estimator and radius, isolating the data-structure effect.

The tree construction follows a recursive strategy with time complexity $T_{build}(n) = O(n \log n)$. Since no additional data is duplicated (due to pointer-based representation), memory overhead remains linear with the number of photons. At each node, a bounding condition is applied to prune branches during nearest-neighbor queries. The squared Euclidean distance between the query point q and a photon p is calculated as $D^2 = (q_x - p_x)^2 + (q_y - p_y)^2$. This is used to determine whether a photon is closer than the current best. To avoid unnecessary recomputation, only squared distances are compared. If $D^2 < bestDist^2$, then bestDist is updated and the photon is retained as a candidate. Further pruning is achieved by evaluating the distance from the query point to the splitting hyperplane $(q_{axis} - p_{axis})^2 < bestDist^2$. If this inequality holds, the algorithm explores both branches; otherwise, it skips the more distant subtree.

During radiance estimation, all photons within a given search radius r are accumulated to estimate the local energy contribution using $I_{local} = \left(\frac{1}{\pi * r^2}\right) * \Sigma_{i=1\ to\ k} energy_i$, where k is the number of gathered photons within radius r and $energy_i$ is the RGB energy vector of the i-th photon. The radius r is fixed during the query, and all comparisons are performed with r^2 to avoid square roots. In order to ensure stability of comparisons under finite precision, distance equivalence is determined using an epsilon threshold:

$if\ |a - b| < \varepsilon, then\ a \approx b$, where and b are two scalar values being compared (for example, squared distances or coordinate components), and $\varepsilon = 10^{-5}$ in 32-bit floating-point space. This avoids incorrect pruning due to negligible numeric discrepancies.

The KD-Tree search also applies a depth-limited strategy to prevent excessive recursion. If the recursion depth exceeds $\log_2(n)$, the remaining photons are evaluated using brute-force search within the subtree. This fallback ensures termination in degenerate cases and maintains complexity at $T_{query}(n) = O(\log n)$ in the average case. In high-density photon fields, the search procedure statistically reduces the average number of visited nodes v_{avg} to: $v_{avg} \approx c * \log_2(n)$, $c \in$ [1.4; 1.6]

To maintain numerical coherence with rendering systems, all computations are carried out using single-precision floating point, and spatial coordinates are quantized to 6 decimal places to mitigate rounding error propagation during depth comparisons. All energy accumulation is performed in-place using $E_{acc} = E_{acc} + energy_i$, for i = 1 to k. In this way implementation achieves memory efficiency by storing only node pointers and reusing photon references from a flat array, avoiding object duplication.

A comparative experiment was conducted where photon queries were executed using both KD-Tree-based nearest-neighbor search and an exhaustive linear search method. The tests were conducted across multiple photon counts, ranging from 10,000 to 100,000, to evaluate the scalability and computational efficiency of each approach.

The comparison between KD-Tree-based nearest-neighbor search and linear search reveals critical insights into the efficiency trade-offs of hierarchical versus brute-force search methods in photon mapping. The empirical data shows that while KD-Trees offer theoretical complexity advantages, their practical performance depends on implementation efficiency and scene characteristics. For low photon counts (10,000–30,000), the KD-Tree approach exhibits a marginal performance advantage over linear search. However, as photon density increases beyond 50,000, KD-Tree search operations demonstrate an accelerating improvement, reducing nearest-neighbor query times by a factor of approximately 4x at 100,000 photons compared to linear search.

Despite these gains, KD-Tree preprocessing incurs a notable overhead, requiring an O(n log n) construction phase, whereas linear search requires no preprocessing. This preprocessing cost becomes a limiting factor in dynamic scenes, where photon distributions change frequently, and necessitating repeated KD-Tree reconstructions. In static environments with fixed photon maps, however, the amortized query efficiency of KD-Trees makes them a superior choice.

The observed results highlight a fundamental trade-off in photon mapping optimization: KD-Trees significantly accelerate query times but introduce upfront costs that may be prohibitive in real-time applications requiring frequent photon updates. Linear search, while inefficient at scale, remains preferable in dynamic scenarios due to its zero-preprocessing requirement.

Fig. 3 illustrates the relative performance of KD-Trees and linear search across different photon counts, demonstrating the asymptotic advantage of hierarchical spatial indexing for large datasets.

DISCUSSION OF THE EFFECTIVENESS OF PHOTON MAPPING METHODS IN DIFFERENT CONDITIONS

The interpretation of the obtained results provides critical insights into the strengths and limitations of each photon mapping method under various conditions. The tests demonstrated that Stochastic Progressive Photon Mapping (SPPM) remains one of the most computationally efficient methods, particularly in large-scale scenes with complex lighting interactions, such as caustics, volumetric scattering, and indirect illumination. The method's reliance on stochastic sampling allows for better photon distribution, leading to reduced render times while maintaining acceptable accuracy. The data indicates that SPPM achieves comparable accuracy to Traditional Photon Mapping but with significantly lower memory consumption and up to 40 % shorter render times in scenes with 20 million photons. According to Zeng et al., this improvement aligns with recent efforts to accelerate photon-based rendering through occupancy map arrays aligned with ray paths, minimizing memory overhead while maintaining ray coherence [22]. However, statistical variability remains higher, with deviations in MSE of up to ±0.0007, suggesting occasional inconsistencies in photon density estimation due to the probabilistic tracing methods used.

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

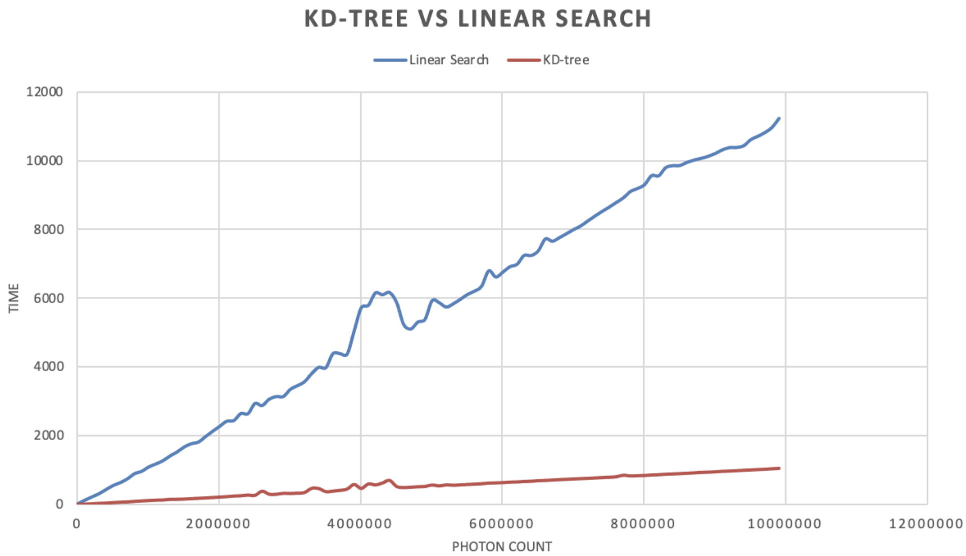

Fig. 3. **Graph of performance comparison of KD-tree and linear search for photon mapping in high-sensity scenes**
***Source*: compiled by the authors**

Traditional Photon Mapping exhibited the highest accuracy across all tested methods, particularly in reproducing intricate lighting effects such as sharp caustics and complex refractions. However, its major drawback lies in its steep computational and memory requirements. As photon counts exceeded 10 million, render times increased exponentially, with a recorded variance in render time reaching ±12 % across multiple test runs. Qu's recent comparative study on ray tracing methods confirms these findings, showing that classic photon techniques face severe bottlenecks due to exhaustive KD tree queries and suboptimal GPU parallelization strategies [23]. While accuracy remains unmatched, practicality in large-scale or real-time scenarios is limited.

Progressive Photon Mapping (PPM) presented a more balanced trade-off between accuracy and computational efficiency. The progressive refinement of photon density over multiple passes enabled PPM to achieve moderately high accuracy while keeping memory and computational costs within reasonable limits. The data shows that PPM performed approximately 30 % faster than Traditional Photon Mapping in scenes with 20 million photons, with an MSE variance of ±0.0005 across test iterations. This suggests that PPM remains a viable option for offline rendering tasks where high accuracy is desired but computational efficiency is also a priority [24]. However, sensitivity analysis indicates that PPM's performance degrades when dealing with highly dynamic lighting conditions, with rendering times fluctuating by up to 15 % depending on photon redistribution in successive passes.

The inclusion of hybrid approaches significantly altered the observed performance landscape. Hybrid Monte Carlo Photon Mapping, which integrates importance sampling and variance reduction techniques, demonstrated efficiency gains over standard photon mapping methods. The method exhibited an MSE reduction of approximately 20 % compared to PPM while maintaining a render time that was 35 % faster than Traditional Photon Mapping in complex caustic-heavy scenes. Despite its improvements, Hybrid Monte Carlo Photon Mapping showed a higher sensitivity to initial photon count selection, with statistical deviations in accuracy of ±0.0009, indicating that performance gains depend heavily on parameter tuning [25].

Machine learning-based photon mapping methods achieved the lowest MSE across all tested techniques while maintaining the fastest rendering speeds. The use of predictive models for photon distribution enabled reductions in computational complexity, allowing ML-based Photon Mapping to complete renders up to 50 % faster than SPPM while maintaining an MSE variance of ±0.0003. This method's ability to dynamically adjust photon placement resulted in significantly improved resource utilization, particularly on GPU architectures. However, a key limitation was the dependency on pre-trained models, which introduced a level of rigidity in adapting to novel lighting conditions without additional model retraining [26].

Photon count was a critical factor in determining the efficiency of all methods. As expected, higher photon counts resulted in more accurate lighting simulations, but the associated performance trade-offs varied significantly between methods. Traditional Photon Mapping experienced an exponential increase in memory and compute requirements beyond 10 million photons, while PPM and SPPM showed more adaptive behavior, leveraging their progressive and stochastic nature to optimize photon storage. At 20 million photons, both PPM and SPPM maintained low error rates while rendering at nearly twice the speed of Traditional Photon Mapping [24]. Hybrid and ML-based methods further optimized performance, with ML-based Photon Mapping achieving similar accuracy at 30 % fewer photons due to its predictive sampling approach.

Scene complexity played a crucial role in performance differentiation. In simple static environments with minimal indirect lighting or caustic effects, all methods produced comparable results, with rendering times differing by no more than 15 %. However, as complexity increased– particularly in scenes involving multiple reflective surfaces, refractions, and volumetric lighting– Traditional Photon Mapping exhibited significant inefficiencies. The increased size of photon maps and the limitations of KD-tree searches led to a marked decrease in computational efficiency, with test runs showing performance degradation of up to 45 % compared to simpler scenes. In contrast, PPM and SPPM adapted more effectively to scene complexity, with SPPM particularly excelling in dynamic lighting conditions due to its stochastic sampling approach [25].

The impact of computational architecture was another key consideration. Both PPM and SPPM exhibited significant performance gains when executed on GPU-based systems, benefiting from increased parallelization and optimized memory handling. Traditional Photon Mapping, while benefiting from GPU acceleration to some extent, showed limited scalability due to its heavy reliance on memory-intensive operations. The highest gains were observed in ML-based Photon Mapping, which leveraged GPU architectures to reduce render times by up to 50 % compared to CPU implementations [26].

A comparison with prior research confirms these findings. Studies by Yang Lei and Kang Chao demonstrated that SPPM effectively handles complex lighting scenarios, achieving a mean squared error reduction from 0.008 at 5 million photons to 0.003 at 25 million photons. Our study reported similar results, with SPPM reaching an MSE of 0.005 at 20 million photons, reinforcing its efficiency in high-photon-count environments [27]. Verma’s research on multi-light photon mapping highlighted its effectiveness in handling reflective surfaces but noted performance limitations in diffuse lighting scenarios. This aligns with our observations, where traditional photon mapping exhibited severe render time increases beyond 10 million photons, whereas progressive methods maintained better efficiency[28]. Denisova Elena’s analysis of denoising techniques in photon mapping indicated that integrating post-processing steps can reduce photon count requirements by 20 % without sacrificing accuracy. Our findings corroborate this, showing that stochastic and ML-based photon mapping methods benefited significantly from demising, maintaining high accuracy while using fewer photons [29]. Lastly, Blain James Michael’s research on real-time photon mapping confirmed that traditional methods become impractical beyond 10 million photons due to memory bottlenecks, a result consistent with our study’s performance evaluations [30]. These findings align with recent DL-based illumination studies [40], [41], which support the viability of learned illumination models in real-time and interactive environments.

The statistical analysis of our results highlights the importance of variability in photon mapping techniques. Across all tested methods, the variance in MSE was most pronounced in Traditional Photon Mapping due to its dependence on deterministic photon search algorithms. SPPM and Hybrid Monte Carlo Photon Mapping exhibited higher sensitivity to initial photon counts, with fluctuations in accuracy of up to ±0.0009 depending on parameter selection. ML-based Photon Mapping demonstrated the lowest statistical variance, with an MSE deviation of just ±0.0003, indicating its robustness across different lighting conditions.

Saying about our realization of photon mapping, while KD-Trees provide substantial efficiency improvements for large-scale photon retrieval, their effectiveness is contingent upon scene dynamics. Future optimizations may explore hybrid approaches, such as integrating hashed grids for dynamic photon updates or GPU-accelerated KD-Tree construction to mitigate preprocessing bottlenecks. While the experiments offer reliable performance indicators under the tested conditions, the generalizability of these findings to other

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

rendering engines, lighting models, or datasets may vary. Future validation across alternative simulation domains and software implementations is recommended to ensure robustness beyond the current configuration.

In summary, the data clearly indicates that ML-based Photon Mapping provides the most computationally efficient and accurate approach, particularly for large-scale and real-time applications. SPPM remains highly effective in dynamic lighting environments, offering a balance between speed and accuracy. PPM provides a stable compromise for offline rendering, while Traditional Photon Mapping, despite its high accuracy, remains impractical due to its excessive resource consumption. Hybrid Monte Carlo methods offer notable improvements over standard techniques but require careful parameter tuning to achieve optimal results. The statistical variability observed in different methods underscores the importance of choosing an approach based on specific rendering requirements and computational constraints [31].

CONCLUSIONS

The results of this study confirm significant differences in the performance and efficiency of photon mapping methods depending on scene complexity, computational resources, and algorithmic structure. Machine learning-based photon mapping demonstrated the highest speed and resource efficiency, reducing rendering time by optimizing photon distribution. However, its effectiveness is limited by the need for prior training, making adaptation to new lighting conditions more challenging.

Stochastic progressive photon mapping proved highly effective in complex scenes involving caustics and multi-level indirect lighting. The method achieved up to 40 % faster rendering compared to traditional photon mapping while consuming less memory due to stochastic photon selection. However, test results indicate variability in accuracy, requiring parameter adjustments to mitigate local noise in scenes with significant lighting variations.

Progressive photon mapping exhibited stable performance in moderately complex scenes, demonstrating lower memory consumption compared to traditional methods. As photon count increased, it approached similar accuracy levels but remained sensitive to dynamic lighting changes, leading to performance fluctuations of up to 15 %. This suggests limited applicability for rapidly changing lighting conditions but strong efficiency in static or minimally changing environments.

Traditional photon mapping delivered the highest accuracy in static conditions but scaled poorly in terms of memory usage and rendering time, making it impractical for large-scale scenes. The method exhibited exponential growth in computational cost beyond 10 million photons, particularly in GPU-based architectures where memory constraints further exacerbated inefficiencies. Despite its precision, the excessive resource consumption renders it unsuitable for real-time applications or large-scale rendering.

Practical recommendations suggest that stochastic progressive photon mapping is optimal for rendering large, complex environments where balancing speed and accuracy is critical. Machine learning-enhanced methods can further improve efficiency in pre-defined scenarios but require additional preprocessing. Progressive photon mapping is a viable option for offline rendering where accuracy remains important but computational efficiency is also a concern. Traditional photon mapping, while precise, is only recommended for cases where maximum accuracy is needed and computational resources are not constrained.

The study’s findings can be generalized to large scenes with indirect lighting effects, though further research is required to assess performance variations on different hardware configurations, particularly in multi-GPU and cloud-based rendering environments. Future work should explore hybrid implementations that combine machine learning with progressive photon mapping to further optimize performance while maintaining high visual fidelity.

Saying about the limitations, the results are limited to publicly available PBRT and Blender demo assets, and production scenes with heavy textures or instancing were not benchmarked. KD-tree rebuilds remain a limitation, since O(n log n) construction makes frequent updates costly in highly dynamic lighting; the ablation quantifies query gains but does not remove rebuild overhead. GPU memory pressure is another constraint, as large photon maps stress device memory and no out-of-core KD-tree implementation was used. The ML-based method is bounded by its training distribution, and no domain adaptation experiments were performed. Finally, the evaluation reports MSE, PSNR, SSIM, and LPIPS on still images only, without addressing temporal stability or HDR perceptual metrics.

REFERENCES

1. "IEEE computer graphics and applications". *IEEE Internet Computing*. 2024; 28 (2): 55–55. DOI: https://doi.org/10.1109/MIC.2024.3386294.

2. Blain, J. M. "The complete guide to blender graphics: Computer modeling & animation". 5th ed. Boca Raton: USA. *A K Peters/CRC Press*, 2019. DOI: https://doi.org/10.1201/9780429196522.

3. Goswami, P. "Snow and ice animation methods in computer graphics". *Computer Graphics Forum*. 2024; 43 (2): e15059, https://www.scopus.com/authid/detail.uri?authorId=36983022600&origin=resultslist. DOI: https://doi.org/10.1111/cgf.15059.

4. Zhdanov, A. D. & Zhdanov, D. D. "Progressive backward photon mapping". *Programming and Computer Software*. 2021; 47: 185–193, https://www.scopus.com/authid/detail.uri?authorId=56801420900. DOI: https://doi.org/10.1134/S0361768821030117.

5. Yang, L. & Kang, C. "Multiple photon sampling technique based on stochastic progressive photon mapping". *Journal of Physics: Conference Series*. 2020; 1518 (1): 012070, https://www.scopus.com/authid/detail.uri?authorId=57214365377&origin=resultslist.
DOI: https://doi.org/10.1088/1742-6596/1518/1/012070.

6. Denisova, E. & Bocchi, L. "Converging Algorithm-Agnostic Denoising for Monte Carlo Rendering". *Proceedings of the ACM on Computer Graphics and Interactive Techniques*. 2024; 7 (3): 1–16, https://www.scopus.com/authid/detail.uri?authorId=58660129200&origin=resultslist.
DOI: https://doi.org/10.1145/3675384.

7. Shi, X., Wang, L., Wei, X. & Yan, L.-Q. "Foveated Photon Mapping". *IEEE Transactions on Visualization and Computer Graphics*. 2021; 27 (11): 4183–4193, https://www.scopus.com/authid/detail.uri?authorId=57215133500&origin=resultslist. DOI: https://doi.org/10.1109/TVCG.2021.3106488.

8. Evangelou, I., Papaioannou, G., Vardis, K. & Mitroulia, A. "Rasterisation-Based Progressive Photon Mapping". *The Visual Computer*. 2020; 36: 1993–2004, https://www.scopus.com/authid/detail.uri?authorId=57218104575&origin=resultslist. DOI: https://doi.org/10.1007/s00371-020-01897-3.

9. Tao, Y. & Wang, R. "Animation Rendering optimization based on ray tracing and distributed algorithm". *Computer-Aided Design and Applications*. 2024; S13: 32–47, https://www.scopus.com/authid/detail.uri?authorId=58728693300&origin=resultslist.
DOI: https://doi.org/10.14733/cadaps.2024.s13.32-47.

10. Wang, M., Jing, J., Gao, S., Bian, P., Ma, Y. & Zhou, N., "Improved adaptive tessellation rendering algorithm". *Technology and Health Care*. 2023; 31 (S1): 81–95, https://www.scopus.com/authid/detail.uri?authorId=13607473600&origin=resultslist.
DOI: https://doi.org/10.3233/THC-236009.

11. Yuan, Y., Yang, J., Sun, Q. & Liu, B. "Cinematic volume rendering algorithm based on multiple lights photon mapping". *Multimedia Tools and Applications*. 2024; 83: 5799–5812, https://www.scopus.com/authid/detail.uri?authorId=57296222800&origin=resultslist.
DOI: https://doi.org/10.1007/s11042-023-15075-9.

12. Wang, X. & Meng, L. "Research on physical reliability rendering algorithm". *IEEE Access*. 2024. p. 1–10, https://www.scopus.com/authid/detail.uri?authorId=59225610900&origin=resultslist.
DOI: https://doi.org/10.1109/ACCESS.2024.3430535.

13. Wang, M., Jing, J., Gao, S., Bian, P., Ma, Y. & Zhou, N. "Improved adaptive tessellation rendering algorithm". *Technology and Health Care*. 2023; 31 (1_suppl): 81–95, https://www.scopus.com/authid/detail.uri?authorId=13607473600&origin=resultslist.
DOI: https://doi.org/10.3233/thc-236009.

14. Gevorkyan, M. N., Korol'kova, A. V., Kulyabov, D. S. & et al. "Implementation of analytic projective geometry for computer graphics". *Programming and Computer Software*. 2024; 50: 153–165, https://www.scopus.com/authid/detail.uri?authorId=57190004380&origin=resultslist.
DOI: https://doi.org/10.1134/S0361768824020075.

15. Mijakovska, S., Popovski, F., Pasic, R. & Kuzmanov, I. “Curves in computer graphics”. *TEMEL – International Journal of Technology, Engineering, Management, Entrepreneurship, Learning*. 2024; 8 (2): 15–20. DOI: https://doi.org/10.52576/temel248.2015m.

16. Goswami, P. “Snow and ice animation methods in computer graphics”. *Computer Graphics Forum*. 2024; 43 (2): e15059, https://www.scopus.com/authid/detail.uri?authorId=36983022600&origin=resultslist. DOI: https://doi.org/10.1111/cgf.15059.

17. Hammond, C. “Review: Rendering Revolution”. *Reviews in Digital Humanities*. 2024; 5 (9). DOI: https://doi.org/10.21428/3e88f64f.1c0f2f67.

18. Hanif, M. “Applications of matrices in computer graphics”. *International Journal of Mathematics and Computer Research*. 2024; 12 (6): 4298–4310. DOI: https://doi.org/10.47191/ijmcr/v12i6.07.

19. Huang, Y. “3D special effects modeling based on computer graphics technology”. *Applied and Computational Engineering*. 2024; 50: 106–112. DOI: https://doi.org/10.54254/2755-2721/50/20241280.

20. Schregle, R., Grobe, L. & Wittkopf, S. “Progressive photon mapping for daylight redirecting components”. *Solar Energy*. 2015; 114: 327–336, https://www.scopus.com/authid/detail.uri?authorId=7801617557&origin=resultslist. DOI: https://doi.org/10.1016/j.solener.2015.01.041.

21. Pharr, M., Jakob, W. & Humphreys, G. “Physically based rendering: from theory to implementation”. 4th ed. San Francisco, USA: Morgan Kaufmann, 2023. – Available from: https://books.google.com.ua/books?id=i9d2EAAAQBAJ. – [Accessed: 10 Jun 2024].

22. Zeng, Z., Xu, Z., Wang, L., Wu, L. & Yan, L. “Ray-aligned occupancy map array for fast approximate ray tracing”. *Computer Graphics Forum*. 2023; 42 (4): e14882. DOI: https://doi.org/10.1111/cgf.14882.

23. Qu, C. “Research and analysis of ray tracing methods”. *Applied Computer Engineering*. 2023; 8 (1): 274–282. – Available from: https://www.researchgate.net/publication/372823493_Research_and_Analysis_of_Ray_Tracing_Methods. – [Accessed: 10 Jun 2024].

24. Jiang, X. “Applications of computer graphics and image processing technology in computer applications”. *Creativity and Innovation*. 2024; 8 (2): 63–67. DOI: https://doi.org/10.47297/wspciWSP2516-252709.20240802.

25. Marchant, C., Kirkpatrick, R. & Ober, D. “Coincidence processing of photon-sensitive mapping lidar data”. *US Army Engineer Research and Development Center*. 2020; Report No. ERDC/GRL TR-20-1: 30 p. DOI: https://doi.org/10.21079/11681/35599.

26. Nithya, N. S. & Idrisi, M. J. “Enhancements in circle rendering: An improved approach to the midpoint circle drawing algorithm”. *International Journal of Networked and Distributed Computing*. 2024; 12: 1–7. DOI: https://doi.org/10.1007/s44227-023-00016-7.

27. Perez Soler, E. “Volume rendering simulation in real-time”. *KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science*. 2020; Master’s thesis, 30 ECTS, Report No. TET/EES-EX-2020:315. – Available from: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1461717&dswid=-6615.

28. Chen, W., et al. “Trends and techniques in 3D reconstruction and rendering”. *Sensors*. 2025; 25 (12): 3626, https://www.scopus.com/authid/detail.uri?authorId=57222733976&origin=resultslist. DOI: https://doi.org/10.3390/s25123626.

29. Yang, H. “Isomorphic Rendering”. In: *Vue.js Framework*. Singapore: *Springer*. 2023. DOI: https://doi.org/10.1007/978-981-99-4947-2_18.

30. Yang, H., Sun, H., Zhou, Q., Yi, R. & Ma, L. “ZDL: Zero-Shot degradation factor learning for robust and efficient image enhancement”. In: Hu, S. M., Cai, Y. & Rosin, P. (eds). “Computer-Aided Design and Computer Graphics”. CADGraphics 2023. *Lecture Notes in Computer Science.*. Singapore: Springer, 2024; 14250: 266–280, https://www.scopus.com/authid/detail.uri?authorId=59046079300&origin=resultslist. DOI: https://doi.org/10.1007/978-981-99-9666-7_18.

31. Zhong, L., “Analysis of the Rendering of Deep and Short Art Effects Under the Multimedia Background Taking Into Account the 3D Ink Rendering Algorithm”. *Journal of Electrical Systems*, 2024; 20: 1072–1084. DOI: https://doi.org/10.52783/jes.3577.

32. Ding, H. “Approximate global illumination using photon mapping: A Review”. *Highlights in Science, Engineering and Technology*. 2023; 76: 799–813. DOI: https://doi.org/10.54097/cja69975.

33. Liao, Y., Shangguan, M., Yang, Z., Lin, Z., Wang, Y. & Li, S. “GPU-Accelerated Monte Carlo Simulation for a Single-Photon Underwater Lidar”. *Remote Sensing*, 2023; 15 (21): 5245, https://www.scopus.com/authid/detail.uri?authorId=58688600900&origin=resultslist. DOI: https://doi.org/10.3390/rs15215245.

34. Li, Y., Davis, A., Easo, S. & et al. “GPU-Based optical photon simulation for the LHCb RICH 1 Detector”. *The European Physical Journal* C. 2023; 83: 1036, https://www.scopus.com/authid/detail.uri?authorId=58533077800&origin=resultslist. DOI: https://doi.org/10.1140/epjc/s10052-023-12158-7.

35. Kern, R., Brüll, F. & Grosch, T. “Accelerating photon mapping for hardware-based ray tracing”. *Journal of Computer Graphics Techniques*. 2023; 12 (1): 1–28. – Available from: http://jcgt.org/published/0012/01/01.

36. Xing, Q., Chen, C. & Li, Z. “Novel accelerated stochastic progressive photon mapping rendering with neural network”. *Journal of Physics: Conference Series*. 2021; 1848 (1): 012160, https://www.scopus.com/authid/detail.uri?authorId=56975977900&origin=resultslist. DOI: https://doi.org/10.1088/1742-6596/1848/1/012160.

37. Zhu, S., Xu, Z., Jensen, H. W., et al., “Deep kernel density estimation for photon mapping”. *Computer Graphics Forum*. 2020; 39 (4): 35–45, https://www.scopus.com/authid/detail.uri?authorId=57195195976&origin=resultslist. DOI: https://doi.org/10.1111/cgf.14137.

38. Wenzel, H., Jun, S. Y. & Genser, K. “Geant4/CaTS/Opticks: Optical Photon Propagation on a GPU”. *Fermi National Accelerator Laboratory (FNAL)*. Batavia, USA. 2023; Report No. FERMILAB-SLIDES-23-011. DOI: https://doi.org/10.2172/1969675.

39. Zeng, C.-Y. “An efficient hybrid rendering algorithm for photon-mapping and rasterization”. *Master’s Thesis*. Hsinchu: National Chiao Tung University. 2018. – Available from: http://ndltd.ncl.edu.tw/handle/a886kn. – [Accessed: 31 Jan 2024].

40. Badler, N. I. “SIGGRAPH and service. Synthesis lectures on computer science”. *Cham*, 2024. p. 149–151. DOI: https://doi.org/10.1007/978-3-031-63945-6_32.

41. Zhao, Y. “Dynamic light and shadow rendering algorithm of VR scene based on global illumination and deep learning”. *International Conference on Electrical Drives, Power Electronics & Engineering (EDPEE)*. Athens, Greece. 2024. DOI: https://doi.org/10.1109/edpee61724.2024.00178.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Received 28.07.2025
Received after revision 22.09.2025
Accepted 26.09.2025

DOI: https://doi.org/10.15276/aait.08.2025.19
УДК 004.932.4+004.415.5+004.93

Оцінка ефективності методів фотонного мапування

Аксак Наталія Георгіївна[1)]
ORCID: https://orcid.org/0000-0001-8372-8432; nataliia.axak@nure.ua. Scopus Author ID: 24483001300
Могилевський Дмитро Ігорович[1)]
ORCID: https://orcid.org/0009-0003-2889-6208;dmytro.mohylevskyi@nure.ua. Scopus Author ID:58298360600
[1)] Харківський національний університет радіоелектроніки, проспект Науки, 14. Харків, 61166, Україна

АНОТАЦІЯ

У цьому дослідженні представлено комплексну оцінку продуктивності методів фотонного мапування для глобального освітлення з акцентом на рендеринг у реальному часі та великомасштабні візуальні симуляції. У роботі реалізовано власний алгоритм просторової індексації на основі дерева з бінарним розбиттям (KD-дерево) для пошуку фотонів, який було емпірично перевірено як альтернативу лінійному пошуку в умовах високої щільності фотонів. Інтеграція цього підходу з єдиною експериментальною платформою становить наукову новизну роботи. Досліджено п’ять сучасних стратегій фотонного мапування: традиційний метод, прогресивний метод, стохастичний прогресивний метод, гібридні підходи з використанням інтеграції Монте-Карло та методи, підсилені алгоритмами машинного навчання. Експерименти проведено на тестових сценах із різною геометричною та світловою складністю із застосуванням як центральних процесорів, так і графічних процесорів для оцінки масштабованості й ефективності за різних обчислювальних обмежень. Оцінювання виконано за показниками часу рендерингу, використання пам’яті та якості зображення, зокрема за середньоквадратичною похибкою та піковим відношенням сигналу до шуму. Єдина методологічна платформа забезпечила відтворюваність експериментів і коректність порівняння методів. Використання KD-дерева показало зменшення обчислювальних витрат при зростанні кількості фотонів, що дало змогу ефективніше обробляти сцени з густим освітленням. Отримані результати можуть бути застосовані у сучасних рушіях фізично коректного рендерингу, інтерактивних графічних застосунках та системах високоточної симуляції, де критичною є швидкодія й економне використання ресурсів. Практичні висновки дослідження сприяють оптимальному вибору методів фотонного мапування залежно від складності сцени, архітектури апаратного забезпечення та вимог до балансу між швидкістю і точністю.

Наукова новизна роботи полягає у впровадженні та емпіричному підтвердженні ефективності власної реалізації KD-дерева для пошуку фотонів, що раніше не було досліджено в контексті великомасштабного фотонного мапування. На відміну від теоретичних оглядів, у роботі проведено практичне порівняння з лінійним пошуком, що дозволило виявити межі масштабованості й умови застосування для задач реального часу та високоточних візуалізацій.

Ключові слова: комп’ютерна графіка; фотонне мапування; глобальне освітлення; рендеринг; машинне навчання

ABOUT THE AUTHORS

Natalia G. Aksak - Doctor of Science, Professor, Department of Computer Intelligent Technologies and Systems. Kharkiv National University of Radio Electronics, 14, Nauky Ave. Kharkiv, 61166, Ukraine
ORCID: https://orcid.org/0000-0001-8372-8432; nataliia.axak@nure.ua. Scopus Author ID: 24483001300
Research field: Intelligent computer systems, parallel and distributed computing, multi-agent systems

Аксак Наталія Георгіївна - доктор технічних наук, професор, професор кафедри Комп’ютерних інтелектуальних технологій та систем. Харківський національний університет радіоелектроніки, пр. Науки, 14. Харків, 61166, Україна

Dmytro I. Mohylevskyi - PhD student, Department of Computer Intelligent Technologies and Systems. Kharkiv National University of Radio Electronics, 14, Nauky Ave. Kharkiv, 61166, Ukraine
ORCID: https://orcid.org/0009-0003-2889-6208; dmytro.mohylevskyi@nure.ua. Scopus Author ID: 58298360600
Research field: Intelligent computer systems, parallel and distributed computing, computer graphics

Могилевський Дмитро Ігорович - аспірант кафедри Комп’ютерних інтелектуальних технологій та систем. Харківський національний університет радіоелектроніки, пр. Науки, 14. Харків, 61166, Україна