Global Illumination using ReSTIR DI and Photon-Mapped Virtual Point Lights

An improvement on Instant Radiosity

Samuel Bruin[1]

Supervisor(s): Christoph Peters[1]**, Michael Weinmann**[1]**, Elmar Eisemann**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Samuel Bruin
Final project course: CSE3000 Research Project
Thesis committee: Christoph Peters, Michael Weinmann, George Smaragdakis

Abstract

Realistic image synthesis in computer graphics relies heavily on global illumination (GI) to simulate indirect lighting. Computing GI remains one of the most computationally demanding tasks. Traditional approaches like Instant Radiosity (photon mapping) generate virtual point lights (VPLs) to approximate indirect lighting, but become inefficient in complex scenes due to reliance on uniform light sampling. As the number of VPLs grows, this method leads to increased noise and reduced performance. In this paper we explore the integration of ReSTIR DI (Reservoir-based Spatiotemporal Importance Resampling for Direct Illumination) with photon-mapped VPLs to improve the efficiency and quality of global illumination. By leveraging ReSTIR's ability to reuse and resample light paths spatially and temporally, the aim is to address the scalability issues of uniform sampling and enable practical, real-time GI rendering. The approach is evaluated using three test scenes of varying complexity: Cornell Box, Sahur, and a detailed living room scene. Performance metrics include root mean squared error (RMSE) comparison against a reference and average frame times across different sampling methods. Results demonstrate that ReSTIR with photon mapping achieves significantly lower RMSE values and faster convergence compared to uniform sampling, with an average RMSE improvement of 57.4% across all tested scenes. The method shows particularly strong improvements in visually complex scenes, with ReSTIR consistently producing the cleanest results at 1 sample per pixel. However, the integration within the current implementation introduces computational overhead, with frame times increasing by 4-7 times compared to uniform sampling, and occasional blotting artifacts in regions with high VPL density. Despite these limitations, the work establishes that ReSTIR DI can be successfully extended to handle photon-mapped global illumination, providing a foundation for improving VPL-based rendering efficiency while maintaining visual quality.

1 Introduction

In computer graphics, creating realistic images relies heavily on accurately simulating how light behaves in a scene. One of the most important aspects of this is global illumination (GI), which accounts for indirect lighting; light which bounces off surfaces before reaching the camera. While this adds a lot of realism, as illustrated in Figure 1, it is also one of the most computationally expensive parts of rendering, especially in real-time applications like games or interactive simulations. This makes it challenging for renderers to achieve efficient global illumination.

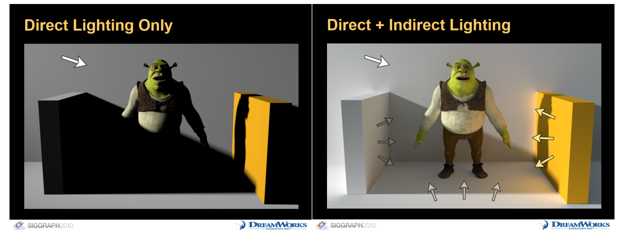

Figure 1: Comparison between direct lighting (left) and full global illumination with both direct and indirect lighting (right) [1].

Photon mapping, detailed in Section 2.2, is one approach that tackles this problem using a two-pass technique. First, it traces photons from light sources through the scene and stores where they hit surfaces. Then, in the second pass, it uses these stored photons to estimate indirect lighting by creating virtual point lights (VPLs) at the intersection points. These VPLs can then approximate the global illumination in the scene.

While photon mapping can perform well in smaller or simpler scenes where a small amount of photons are sufficient, its efficiency drops significantly in more complex environments that require a large number of generated virtual point lights (VPLs) to correctly estimate global illumination. Traditionally, uniform light sampling is used to choose among these VPLs during shading, but as their number increases, this method becomes increasingly inefficient, leading to noisy results.

This project investigates whether Reservoir-based Spatiotemporal Importance Resampling for Direct Illumination (ReSTIR DI), described in Section 2.1, can mitigate these issues. By combining ReSTIR DI with photon-mapped VPLs, the aim is to improve both the quality and speed of global illumination, making the approach more practical for real-time rendering. The study evaluates the effectiveness of this integration in terms of image quality and performance.

The main research question we answer in this paper is:

How can ReSTIR DI be used to improve the performance of global illumination generated through photon mapping?

To break this down, we explore in Section 2.2 how to generate VPLs to simulate indirect lighting, how many we need, and how to distribute their energy. Then in Section 4 we look at how to fit these VPLs into the ReSTIR DI pipeline, which is not designed for indirect lighting. We then compare the performance and quality of this improvement to photon mapping with uniform light sampling, as stated in Section 5.1.

2 Related Work

2.1 ReSTIR DI

Reservoir-based Spatiotemporal Importance Resampling for Direct Illumination (ReSTIR DI) [3] is a technique that increases the quality of rendering scenes by reusing light samples from previous frames and neighboring pixels. It was first developed for direct lighting, but the technique has also been extended to global illumination in ReSTIR GI by Ouyang *et al.* [10]. In this paper, we propose that ReSTIR DI without adaption can be used to also render scenes *with* global illumination. In future work, our technique could be quantitatively and qualitatively compared to ReSTIR GI to better understand the trade-offs and performance differences between the two approaches.

The main problem ReSTIR DI addresses is the inefficiency of traditional Monte Carlo sampling in scenes with many light sources. When a scene contains hundreds or thousands of lights, each pixel must (uniformly) randomly sample from this large set, but most samples contribute little to the final lighting calculation. ReSTIR DI improves this process by reusing high-quality samples both spatially as well as temporally.

The algorithm operates through three phases. First, each pixel generates initial light samples using standard importance sampling. The temporal reuse phase then combines these new samples with samples from the same pixel location in the previous frame. This approach works well because lighting conditions typically remain stable between consecutive frames, especially in static scenes. The spatial reuse phase extends this concept by allowing neighboring pixels to share their samples, taking advantage of the fact that nearby pixels often have similar lighting characteristics. The algorithm in simplified form is also highlighted in Algorithm 1.

Algorithm 1 ReSTIR Sampling Algorithm

// Standard importance sampling
for each pixel $p \in$ Image **do**
    `reservoirs`$[p] \leftarrow$ RIS(p)
end for
// Reuse samples from previous frame at same pixel
for each pixel $p \in$ Image **do**
    $p' \leftarrow$ pickTemporalNeighbor(p)
    `reservoirs`$[p] \leftarrow$ TemporalReuse(S_p, p')
end for
// Reuse samples from neighboring pixels
for each pixel $p \in$ Image **do**
    $p' \leftarrow$ pickSpatialNeighbor(p)
    `reservoirs`$[p] \leftarrow$ SpatialReuse(S_p, p')
end for

The core mechanism behind ReSTIR is weighted reservoir sampling. Each pixel maintains a reservoir that stores one sample at a time, but can replace this sample based on the relative importance weights of candidate samples. Over multiple frames, this process naturally converges toward keeping the most significant light contributions.

2.2 Instant Radiosity

Using Instant Radiosity [8], otherwise known as photon mapping, we can 'convert' indirect lighting to direct lighting; we can generate virtual point lights (VPLs) at each bounce point, as can be seen in Figure 2, which allows ReSTIR DI to directly estimate the indirect illumination component without adaption.

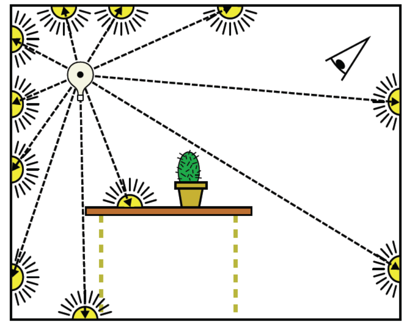

Figure 2: A visual description of Instant Radiosity estimating first-bounce light contribution. Rays are sent out from the light source into the scene. At their hit points, a virtual point light is placed [12].

The Instant Radiosity algorithm works by tracing paths from the light sources into the scene and placing virtual point lights at surface intersection points. Each VPL represents the radiance that would be emitted from that location, effectively capturing the indirect lighting contribution at that point. The intensity and color of each VPL are determined by the incoming radiance at the intersection point and the material properties of the surface, more specifically the bidirectional reflectance distribution function (BRDF) of that surface. As rays bounce through the scene, Russian roulette termination is performed to probabilistically terminate paths, preventing infinite bouncing while maintaining unbiased results by appropriately weighting the surviving paths.

Once all VPLs have been generated, the indirect lighting problem becomes equivalent to a direct lighting problem with many small light sources. This transformation is particularly advantageous for our approach because ReSTIR DI is specifically designed to handle scenes with numerous light sources efficiently. Instead of computing complex path integrals for global illumination, we can now apply ReSTIR's temporal and spatial resampling techniques directly to the VPL set.

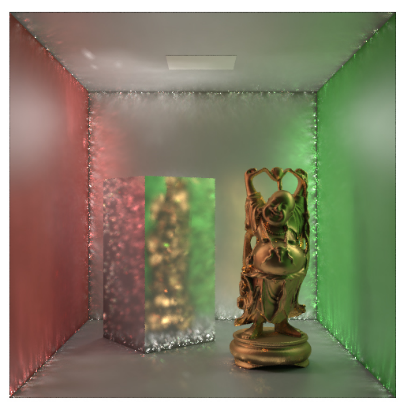

Figure 3: A render of the Cornell Box using 10k VPLs showcases the appearance of spiky artifacts when using VPLs for indirect illumination [14].

However, Instant Radiosity does have some limitations. As noted by Tokuyoshi [16], VPL-based methods can suffer from spiky artifacts, as can be seen in Figure 3, caused by singularities that occur when the point light approximation becomes inaccurate at very close distances to surfaces. Additionally, the quality of the indirect lighting approximation depends heavily on the number and distribution of VPLs, requiring careful tuning of the path tracing parameters to balance quality and performance.

3 Methodology

To investigate whether ReSTIR DI can improve VPL sampling for indirect lighting, we built a framework including three components. We start by implementing a baseline renderer that handles direct illumination using ReSTIR DI, RIS, and uniform sampling, along with a path-traced reference for ground truth comparisons. Next, we develop a photon mapping system that generates VPLs to represent indirect lighting throughout the scene. We then combine these systems to test how ReSTIR's spatiotemporal resampling performs with photon-mapped VPLs compared to standard sampling methods.

This approach allows us to directly compare different sampling strategies for VPL-based GI. The implementation details and design decisions for each component are described in the following subsections.

3.1 Renderer and Baseline Setup

We first construct a baseline renderer, written in C++ using SDL2 [13] and TinyBVH [2], capable of rendering scenes including only direct illumination using ReSTIR DI. This setup also includes implementations of Resampled Importance Sampling (RIS) [15] and uniform light sampling. The system is capable of rendering textured scenes lit by both colored area lights and point lights. Surface interaction is modeled using Lambertian reflectance, which simplifies the shading model while remaining suitable for evaluating indirect lighting techniques. The target function our renderer aims to solve is described in Equation 1 [11], where $L_e(x, \omega_i)$ represents the emitted radiance at point x from direction ω_i, $V(x, \omega_i)$ is the visibility term, $f_r(\omega_i, x, \omega_o)$ denotes the BRDF at point x for incoming direction ω_i and outgoing direction ω_o, and $G(x)$ is the geometric term that accounts for the cosine angle between the surface normal and incident light direction as well as distance-based attenuation.

$$f(x) = L_e(x, \omega_i)\, V(x, \omega_i)\, f_r(\omega_i, x, \omega_o)\, G(x) \quad (1)$$

Our ReSTIR DI implementation follows the standard three-phase approach: initial sample generation, temporal reuse, and spatial reuse. For temporal reuse, we maintain sample history across 32 frames and cap the temporal confidence weight M_r in ReSTIR reservoirs ($M_{cap} = 20$) to limit temporal accumulation and maintain stability as suggested by Lin *et al.* [9]. This prevents the algorithm from becoming unstable due to unbounded sample accumulation while preserving the benefits of temporal coherence. For spatial reuse, we select $k = 8$ neighboring pixels within a 10-pixel radius using uniform random sampling.

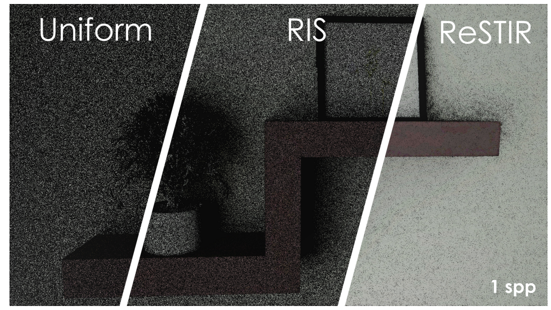

Figure 4: Visual comparison between **uniform light sampling**, **RIS**, and **ReSTIR** in our renderer (1 spp). RIS considers $m = 32$ candidates. Spatial reuse is done with $k = 8$ within a 10-pixel radius, uniformly random. Temporal reuse spans across 32 frames with $M_{cap} = 20$.

To ensure a reliable point of comparison, a path-tracing-based ground truth renderer is developed in parallel as well. We extend the original renderer with a path-tracing mode. This path-traced reference shares most of the core code base, such as scene handling, materials, and geometry, with the main renderer. The path tracer implements next event estimation for direct lighting computation, uniformly sampling one light per intersection. For path termination, we use Russian roulette sampling at each bounce. The continuation probability is calculated from the maximum RGB component of the current throughput, clamped between 0.05 and 0.95, with throughput compensation applied when paths continue.

The complete source code for our renderer implementation, including both the ReSTIR-based system and the path-traced ground truth renderer, is made publicly available[1] to facilitate reproducibility and future research. The repository is developed and tested on multiple platforms. It includes complete test scene data used in our evaluation. Additionally, we provide utility scripts for result analysis, enabling researchers to easily reproduce our experimental pipeline and extend our work with minimal setup overhead.

3.2 Photon Mapping for VPL Generation

To generate the VPLs, we implement a photon mapping system inspired by the Instant Radiosity technique [8]. Photons are emitted from the light sources and traced through the scene to create a set of VPLs. While this method works well for small scenes, it becomes less efficient in complex environments; many photons miss camera-visible areas, resulting in high computational burden [7]. Optimizing photon placement to better target regions of interest, such as areas visible to the camera or regions with high shading complexity, remains an open problem. Future work could explore adaptive or guided photon emission strategies to increase VPL coverage in perceptually important areas, improving both efficiency and visual quality.

For our implementation, we establish a target number of VPLs to generate before terminating the photon tracing process. Photons are emitted from area light sources with uniform distribution across the light surface, where each photon

[1] The code repository can be accessed at: https://github.com/ancientkingg/restir-vpl

carries an initial flux value equal to the light's total radiant power divided by the number of emitted photons. When a photon intersects a surface, we sample a new direction based on the material's bidirectional scattering distribution function (BSDF). The photon's flux is updated according to the surface BSDF and the probability density of the sampled direction, as can be seen in Equation 2

$$\Phi_{i+1} = \Phi_i \cdot \frac{f_r(\omega_i, \omega_o, \mathbf{n}) \cdot (\mathbf{n} \cdot \omega_i)}{p(\omega_o)} \tag{2}$$

We implement Russian roulette, as described in Algorithm 2, to prevent tracing low-energy photons indefinitely. We enforce minimum and maximum bounce limits to ensure reasonable path lengths. At each bounce after the minimum threshold, we evaluate the photon's current flux and use the maximum RGB component as our survival heuristic. To prevent extreme cases where photons always survive or always terminate, we clamp the survival probability to a reasonable range (typically between 0.05 and 0.95). When a photon survives the Russian roulette test, we scale its flux appropriately to compensate for the terminated photons, maintaining energy conservation throughout the process.

Algorithm 2 Russian Roulette Path Termination

if bounces $\geq$ MIN_BOUNCES **then**
  max_flux $\leftarrow \max$ (flux.r, flux.g, flux.b)
  $x \leftarrow$ clamp(max_flux, 0.05, 0.95)
  if rand() $> x$ **then**
    Terminate
  end if
  flux $\leftarrow \frac{\text{flux}}{x}$
end if

At each surface intersection, we create a VPL that represents a hemispherical light source oriented along the surface normal at the intersection point. Each VPL emits light uniformly across the hemisphere above the surface, mimicking how light would naturally scatter from that location. The VPL stores the intersection position, surface normal (which defines the hemisphere orientation), and the photon's flux at that location. The process continues until we reach our target VPL count.

To aid with the debugging of our photon mapping implementation, we developed a visualization mode that renders all geometry using Phong shading and renders generated VPLs as colored spheres. Blue for VPLs intended to replace direct lighting from scene lights and red for VPLs intended to mimic GI.

Since complex scenes generate thousands of VPLs, we use a k-d tree from the nanoflann library [4] for spatial acceleration. An example scenario of the VPL generation and debug mode can be found in Figure 5.

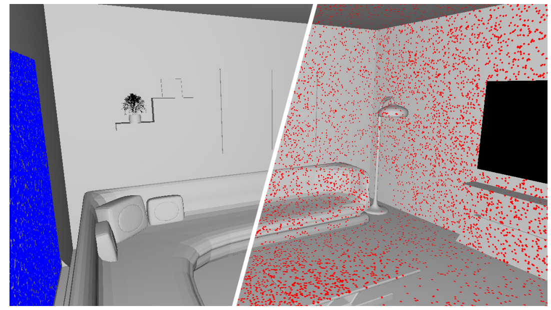

Figure 5: A render in debug mode with Phong shading. Comparison of VPL generation in the living room scene between **direct lighting only** (left) and **direct + indirect lighting** (right). 100k VPLs are generated to replace all scene lights (blue) and 100k VPLs are generated for GI (red).

4 ReSTIR DI with Photon Mapping

4.1 Pipeline Overview and Sampling Strategy

The rendering pipeline proceeds in two main stages:

1. **Photon Mapping Pass**: Photons are emitted from light sources and traced through the scene to produce a set of VPLs. Each VPL stores position, normal, light contribution, and incoming direction at the first hit.
2. **ReSTIR-Based Shading Pass**: For each pixel, a set of candidate VPLs is selected (either uniformly or using RIS), and a local reservoir is formed based on their contributions. ReSTIR's spatiotemporal resampling is then used to refine these candidates by borrowing samples from neighboring pixels and the previous frame.

By structuring the renderer this way, we bypass the need for full path tracing of indirect bounces during rendering, relying instead on precomputed VPLs and real-time resampling.

4.2 Unifying GI with Direct Sampling via VPLs

The core contribution of this project is the integration of photon-mapped VPLs into a ReSTIR DI pipeline. All light sources in the renderer, including physical area lights and photon-derived VPLs, are unified under a single abstraction: a large set of point lights. This simplifies the integration with ReSTIR, which operates over light samples without needing to differentiate between their physical origin. The ReSTIR candidate sampling is then applied over this combined pool of VPLs, enabling spatial and temporal reuse for both direct and indirect components without needing to differentiate between different types of lighting.

To achieve this unification, we convert area lights into discrete point lights through importance-based sampling, as described in Algorithm 3.

4.3 Adapting ReSTIR to Handle Large VPL Sets

To manage the large amount of (dynamically) generated VPLs produced via photon mapping, we implemented Resampled Importance Sampling (RIS) in conjunction with ReSTIR. RIS helps guide the sampling process toward higher-contribution VPLs, improving convergence speed. Combined

Algorithm 3 Generate point lights from triangle lights

$\mathcal{L} \leftarrow$ list of triangular lights in the scene
$W \leftarrow \sum_{\ell \in \mathcal{L}} \text{intensity}(\ell) \cdot \text{area}(\ell)$
for each light $\ell \in \mathcal{L}$ **do**
    $w \leftarrow \frac{\text{intensity}(\ell) \cdot \text{area}(\ell)}{W}$
    $n \leftarrow \lfloor w \cdot N \rceil$ ▷ N: total number of photons
    for $i = 1$ to n **do**
        Sample position p on ℓ
        $n_p \leftarrow$ surface normal of light at p
        $F \leftarrow \frac{\text{intensity}(\ell) \cdot \text{area}(\ell)}{N}$
        Add new point light at p with color c_ℓ, normal n_p and flux F
    end for
end for

with temporal and spatial reuse in ReSTIR, this reduces per-frame noise, as can be seen in Figure 6.

Figure 6: Comparison between RIS (left) and uniform light sampling (right). 5 frame accumulation. RIS considers $m = 32$ candidates. The variance for RIS is 1121.28, significantly lower than the 4728.34 observed with uniform sampling, highlighting RIS's effectiveness in noise reduction.

5 Experimental Setup and Results

5.1 Evaluation and Metrics

All methods are evaluated in controlled test scenes with varying complexity and lighting configurations. We perform two distinct types of evaluation:

- **Visual Quality Assessment**: We accumulate frames over time to generate high-quality reference images and measure visual fidelity using RMSE against path-traced ground truth images. We also compare visual fidelity of ReSTIR against images rendered with high sample counts using uniform light sampling.
- **Noise Comparison**: We compare the variance between our three methods (RIS, ReSTIR, and uniform light sampling) over various sample counts (SPP) to compare noise characteristics.

5.2 Hardware Setup

All rendering experiments were conducted on a Windows PC equipped with an Intel i5-8600K processor to ensure consistent computational conditions across all test runs. However, due to the nature of our evaluation focusing on visual quality and noise characteristics rather than absolute performance metrics, the specific hardware configuration is not particularly relevant to our findings.

5.3 Data used for Evaluation

For our evaluation, we use three test scenes that provide adequate opportunities to showcase the benefits of improved indirect lighting sampling. These scenes range from simple geometric setups to complex real-world environments, allowing us to test our VPL sampling approach under different conditions. Images of the scenes can be found in Appendix A.

The first scene is the Cornell Box [5] (Figure 14), a standard benchmark in global illumination research.

We also test on an open box scene containing a detailed figure (Figure 15). The scene consists of a detailed figure with colored walls and basic geometric objects, making it easy to observe color bleeding and indirect lighting effects. We will refer to this scene as *Sahur* throughout the paper.

Our most complex test case is a detailed living room scene [6] (Figure 16) that features realistic furniture arrangements and numerous occluded areas. The scene was chosen specifically because it offers extensive opportunities for indirect lighting contributions, particularly in shadowed regions behind tables, couches, and other furniture where direct lighting cannot reach. These areas depend heavily on bounced light from multiple surface interactions, making them ideal for testing the effectiveness of different VPL sampling approaches.

To maintain consistent lighting conditions across experiments, we configured each scene with appropriate light sources. The Cornell Box uses a standard area light positioned at the ceiling. The Sahur scene contains a single large area light placed off to the side off-camera to allow for shadow casting within the box. For the living room, we placed a large area light source directly outside the room to serve as the primary light source.

5.4 Lighting and Sampling Parameters

For all experiments, we generate 200,000 VPLs in total: 100,000 to replace the original scene lights and another 100,000 to simulate global illumination bounces. This VPL count was chosen based on preliminary testing that showed that insufficient VPL density leads to noticeable lighting inaccuracies and artifacts, as demonstrated in Figure 7. Previous research has established that adequate VPL sampling density is crucial for maintaining lighting quality, with lower counts resulting in artifacts [8].

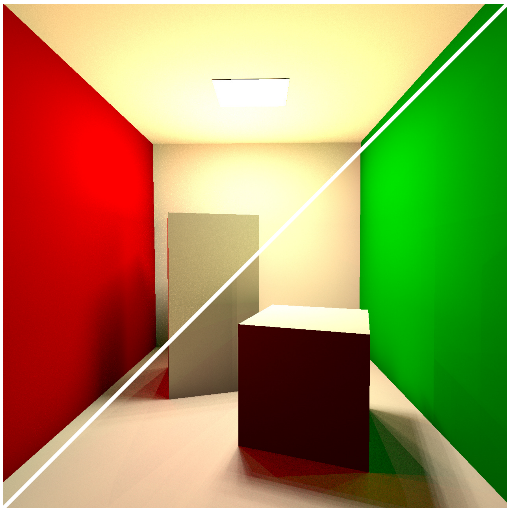

Figure 7: Comparison between photon mapping using 100 VPLs (right) and 100,000 VPLs (left). It showcases how low VPL count can cause artifacts.

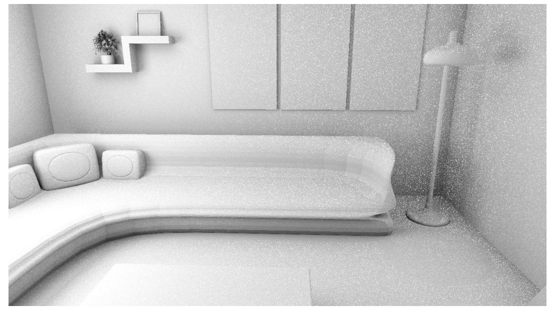

Figure 8: Living room scene rendered using uniform light sampling with 10,000 frames. Despite the high sample count, significant noise artifacts are still visible throughout the image, particularly in the shadowed areas and on surfaces with complex lighting interactions. This demonstrates why uniform sampling becomes computationally prohibitive for scenes with many light sources, necessitating the use of importance sampling techniques like ReSTIR.

A key challenge with VPL-based rendering is the occurrence of firefly artifacts caused by singularities in the distance function, as originally identified by Keller [8] in the Instant Radiosity framework. When a shading point lies very close to a VPL, the $1/d^2$ distance attenuation factor can produce extremely bright contributions that manifest as bright pixel outliers. To address this issue, our implementation uses Yuksel's point light attenuation method [17], which modifies the traditional distance falloff to prevent these singularities while maintaining visually plausible lighting behavior. The effectiveness of this approach in reducing firefly artifacts can be observed in Figure 10.

Our ReSTIR implementation uses both temporal and spatial reuse. We configure the spatial reuse with $k = 8$ uniformly random neighbor selection within a 20-pixel radius, set $M_{cap} = 20$ for reservoir capacity, and use $m = 32$ candidate samples per pixel. For single frame comparisons, reservoirs accumulate samples with temporal reuse over 32 frames to build up sufficient sample history. The RIS baseline uses the same candidate count of $m = 32$ for fair comparison.

We run each test for 20 frames and compare the results against a 4000-frame ground truth rendered using the same VPL setup with RIS. While RIS is theoretically unbiased only in the limit of infinite samples [15], the finite-sample bias introduced with 4000 frames is negligible compared to the substantial variance reduction achieved, making it a practical choice for ground truth generation within our computational constraints. Additionally, we generate a path-traced reference image using 2000 samples per pixel to evaluate visual quality differences between our VPL-based approach and traditional path tracing methods. We were unable to generate ground truth references using uniform light sampling, as preliminary experiments showed significant visual noise, as can be seen in Figure 8, even with sample counts upwards of 10,000 frames, which exceeded our four-hour computational budget per rendering session, as detailed in Section 6.1.

To better showcase the shading behavior and indirect lighting effects, we disable image textures in all scenes, allowing the geometric and lighting interactions to be more clearly visible. Both our photon mapping implementation and the path tracing reference use a maximum ray depth of 8 bounces with Russian roulette termination to handle longer light paths efficiently.

All images are rendered at either 1280×1280 resolution for square scenes like the Cornell Box, or 1280×720 for wider scenes like the living room, providing sufficient detail to analyze the lighting quality and sampling effectiveness.

5.5 Results

The results are presented across several sections, each focusing on a different parts of the evaluation. These include comparisons between direct and global illumination, the impact of distance attenuation models, performance analysis, visual quality with path tracing, single frame comparisons, and RMSE behavior across accumulated frames. All experiments were conducted under consistent conditions as detailed in Section 5.4.

While our implementation shows clear improvements over uniform sampling in most areas, we encountered several unexpected issues during development that affected both performance and visual quality. These problems, along with the successful aspects of our approach, are discussed in detail below.

We begin by examining the difference between direct illumination (DI) and global illumination (GI) using photon mapping. As shown in Figure 9, the image rendered with GI (right) captures significantly more indirect light bounces, resulting in a more realistic and evenly lit scene compared to the DI-only variant (left). This highlights the contribution of global illumination in producing visually accurate results, especially in enclosed scenes.

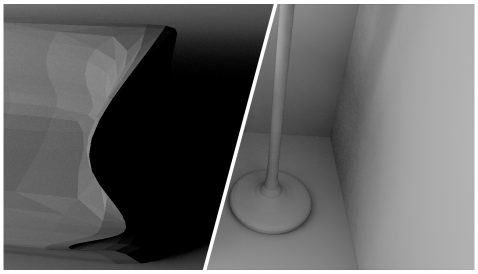

Figure 9: Comparison between global illumination using photon mapping (right) and only direct lighting (left), both using ReSTIR DI. 500 frame accumulation.

Next, we evaluate the effect of distance attenuation on rendering artifacts. When using the standard $1/d^2$ distance attenuation, firefly artifacts become apparent, particularly in ReSTIR, as seen in Figure 10. These artifacts tend to appear in regions where surfaces meet at sharp angles. By replacing the inverse-square distance attenuation with the modified attenuation function proposed by Yuksel [17], the fireflies are effectively removed in both RIS and ReSTIR, improving visual quality.

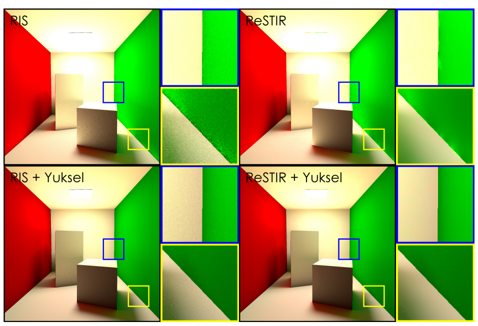

Figure 10: Comparison of Yuksel's attenuation function with inverse square distance attenuation for both ReSTIR and RIS. 500 frame accumulation. Fireflies are noticable at the edges where two planes meet, especially in ReSTIR, but dissapear with Yuksel's attenuation function.

The performance results in Table 5.5 reveal a significant drawback to our current implementation. Both RIS and ReSTIR introduce substantial overhead compared to uniform sampling, with ReSTIR consistently showing the worst performance across all tested scenes. The frametimes are roughly 4-7 times longer than uniform sampling, which represents a considerable computational cost.

This poor performance stems from our decision to implement the algorithms exactly as described in the original pseudocode by Bitterli *et al.* Although this approach ensured algorithmic correctness, it came at the cost of performance, as our code was not optimized for efficiency.

A qualitative comparison with a path-traced (PT) reference image is shown in Figure 11. While ReSTIR with photon mapping achieves comparable global illumination effects using significantly fewer frames, the PT reference image contains some noticeable artifacts. The reference shows ceiling inconsistencies and shadow cutoffs. We hypothesize that these inconsistencies are caused from inaccuracies in ray-scene interactions, likely caused by improper surface offsets resulting in BVH traversal errors.

Scene	Uniform	RIS	ReSTIR
Cornell Box	796 ms	3314 ms	4304 ms
Sahur	647 ms	2995 ms	3996 ms
Living Room	633 ms	3232 ms	4360 ms

Table 1: Average frametime (ms) for uniform light sampling, RIS, and ReSTIR, gathered from 500-frame accumulations across 3 scenes.

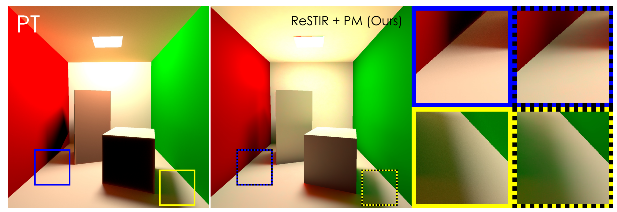

Figure 11: Visual comparison between path tracing (PT, left) and our method (ReSTIR + PM, right). Both images show similar global illumination, with key similarities highlighted in blue and yellow boxes. Differences, such as ceiling artifacts, are also visible. PT was rendered over 2000 frames, while our method used only 500.

Figure 12 illustrates the visual quality at 1 sample per pixel (spp) across three sampling techniques in the living room scene using the parameters referenced in Section 5.4. Refer to Appendix B for more comparisons. RIS and ReSTIR both achieve major reductions in noise compared to uniform sampling. ReSTIR, while providing the cleanest result overall, introduces slight blotting artifacts in some regions. These artifacts are likely a result of the large number of virtual point lights (VPLs) considered during sampling. We believe that this is due to the weighting in candidate generation. At random one of the pixels will select a very good candidate with a large weight. Due to the spatial reuse this candidate will then be spread out to the neighboring pixels, causing this blotting.

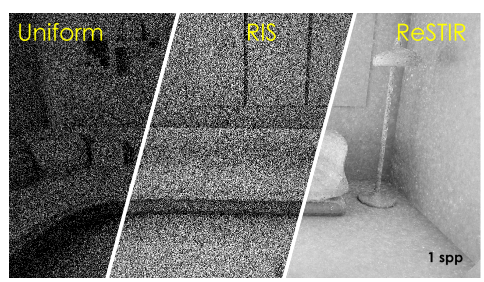

Figure 12: Comparison of sampling techniques using 1 sample per pixel (spp) in the living room scene. From left to right: uniform sampling, reservoir importance sampling (RIS), and ReSTIR.

Finally, we analyze the root mean squared error (RMSE) over accumulated frames (Figure 13). Both RIS and ReSTIR

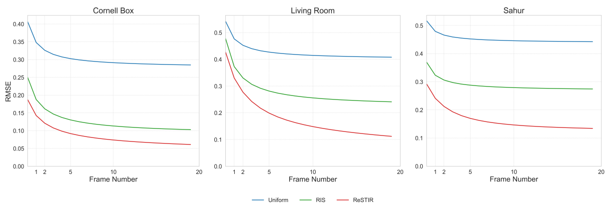

Figure 13: RMSE over 20 accumulated frames comparing different sampling methods. ReSTIR achieves lower RMSE across all scenes, indicating higher rendering quality. See Section C for a larger version.

show significantly faster convergence compared to uniform sampling with an average improvement of 57.4% for ReSTIR and 38.9% for RIS across all scenes. ReSTIR consistently produces the lowest RMSE across all tested scenes. The advantage is especially clear in more complex scenes like in the living room and in Sahur, where ReSTIR converges noticeably faster than RIS. However, for the simpler Cornell Box scene, the convergence rates between RIS and ReSTIR are more comparable, with a smaller margin of improvement. A more detailed overview of the statistics can be found in Appendix C.

Notably, uniform sampling appears to converge much more slowly, and may even look like it is plateauing at a non-zero RMSE. This could mean that it requires thousands of frames to converge, which is also consistent with our earlier observations of visual noise even at high sample counts. Alternatively, the convergence patterns could indicate systematic bias in our evaluation setup, where our RIS-generated ground truth inherently favors importance sampling methods, causing uniform sampling to converge toward a different solution. Future work could address this potential bias by generating ground truth references using uniform light sampling, despite the computational expense, to provide a more definitive evaluation of the relative performance of these sampling methods.

6 Responsible Research

6.1 Environmental Impact Disclosure

We performed all rendering experiments on our personal computing hardware using our own resources and electricity. Given that rendering research can be computationally intensive, we imposed a practical four-hour limit per rendering session. This limit helped maintain reasonable development cycles while avoiding excessive power consumption and hardware stress from extended runs.

Rather than running exhaustive parameter sweeps that could require days of continuous computation, we focused on targeted test cases designed to demonstrate ReSTIR's key capabilities. We were able to gather meaningful performance data, as can be seen in this paper, without requiring extensive computational resources.

6.2 Reproducibility

We have made the complete implementation, including all source code, build configurations, and test scenes used in this work, available in the linked repository. The code base includes setup instructions and dependency specifications that enable compilation across different systems. We documented all rendering parameters and scene configurations through configuration files and inline comments.

The repository structure allows researchers to reproduce the exact results presented in this paper and provides a starting point for further development. By making our implementation freely available, we aim to lower the barrier for others to explore ReSTIR techniques and build upon the foundations established here.

6.3 LLM Usage

Large language models were used to assist with grammar checking and improving sentence clarity during the writing process. All research design, implementation, data analysis, and conclusions were developed independently and were not assisted with AI, to maintain the integrity of this research.

7 Conclusions

In this paper, we proposed an extension to photon mapping by integrating ReSTIR for direct illumination sampling. The goal was to evaluate whether spatiotemporal reuse techniques, originally designed for direct illumination, could improve sample efficiency and visual quality in a photon mapping context.

The results show that ReSTIR, when combined with photon mapping, produces lower RMSE values and faster convergence, in terms of sample count, compared to both uniform sampling and RIS. These improvements were most noticeable in visually complex scenes, where reuse of samples across pixels and frames helped reduce noise and stabilize lighting. However, this came with some trade-offs. ReSTIR introduced blotting artifacts in certain areas, likely due to sometimes randomly sampling a light with a large weight from the large set of virtual point lights. Frame times also increased by 4-7 times compared to uniform sampling, with ReSTIR consistently showing the worst performance across all tested scenes. This is largely due to an unoptimized implementation that prioritized clarity and correctness over efficiency.

In summary, the integration of ReSTIR into a photon mapping renderer improved sample reuse and convergence behavior but introduced new challenges in terms of artifact handling and performance. These results suggest that while the approach is promising, further refinement is needed, particularly in optimizing the implementation and better managing the bias introduced by the large number of VPLs.

Overall, our work establishes that ReSTIR DI can be successfully extended to handle photon-mapped global illumination, providing a way for improving the efficiency and quality of VPL-based rendering. While performance optimization remains a concern for practical use, the implementation demonstrates clear advantages in terms of convergence speed and noise reduction, particularly in scenarios with complex geometry.

8 Future Work

While our renderer demonstrates that photon mapping can be effectively combined with ReSTIR DI to handle thousands of VPLs, there are several areas where this work could be extended to achieve better performance and practical applicability.

The most significant limitation of our work lies in performance optimization. Our implementation follows the original ReSTIR DI pseudocode from Bitterli *et al.* almost exactly, prioritizing algorithmic correctness over performance. This design choice, while valuable for validating the approach, results in the relatively poor frame times we observed earlier. In addition, a GPU implementation would be essential for practical real-time applications. It could provide substantial performance gains and enable interactive frame rates with higher VPL counts and more complex scenes.

Additionally, a direct comparison between our photon mapping + ReSTIR DI approach and ReSTIR GI would provide valuable insights into the trade-offs between different indirect lighting strategies. While ReSTIR GI samples indirect lighting paths directly, our VPL-based method precomputes indirect lighting in the form of VPLs that can be reused across frames. Understanding the trade-offs between these strategies could prove valuable.

An important area for future investigation concerns potential bias in our evaluation methodology. Our RMSE results show uniform sampling converging much more slowly and potentially plateauing at non-zero error levels, which could indicate either extremely slow convergence requiring thousands of frames or systematic bias in our RIS-generated ground truth. Future work should generate ground truth references using uniform light sampling, despite the computational expense, to provide more definitive evaluation of relative sampling method performance.

Finally, while we currently address firefly artifacts using Yuksel's point light attenuation method [17], this approach introduces bias to maintain acceptable visual quality. Although this biased solution works reasonably well for our current implementation, future work could explore more principled variance reduction techniques that minimize fireflies without compromising the unbiased nature of the Monte Carlo estimation.

A Scenes used in Experiments

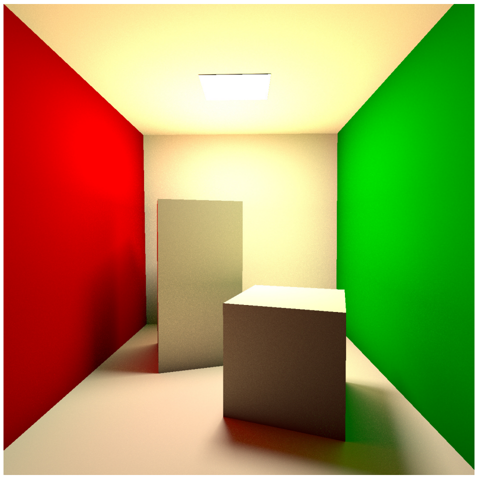

Figure 14: Cornell Box scene with an area light positioned at the ceiling. The scene contains two colored cubes inside of a box.

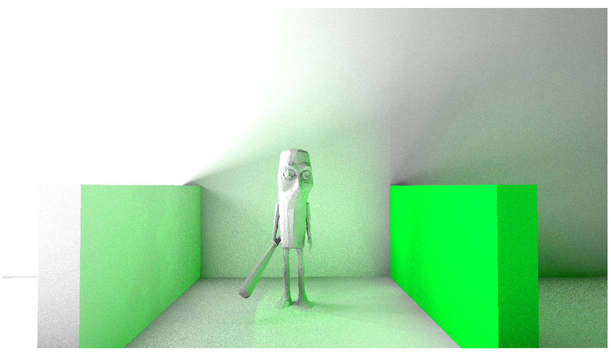

Figure 15: Sahur scene with an area light positioned off-camera (left). The scene contains an open box, of which the right wall is colored, with a detailed figure inside.

Figure 16: Living room scene with a single large area light positioned outside the room (left). The scene contains furniture such as a table and a couch.

B Single Frame Comparisons between Sampling Techniques

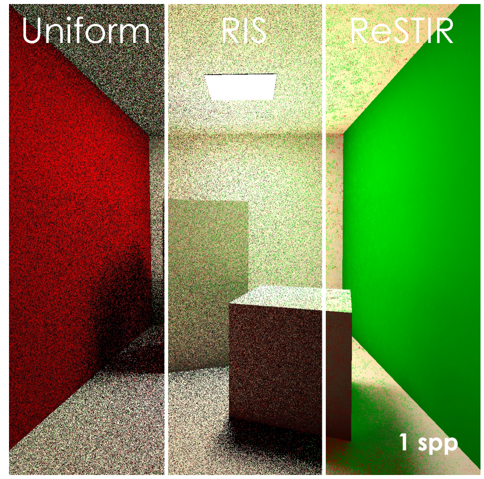

Figure 17: Comparison of sampling techniques using 1 sample per pixel (spp) in the Cornell Box scene. From left to right: uniform sampling, reservoir importance sampling (RIS), and ReSTIR.

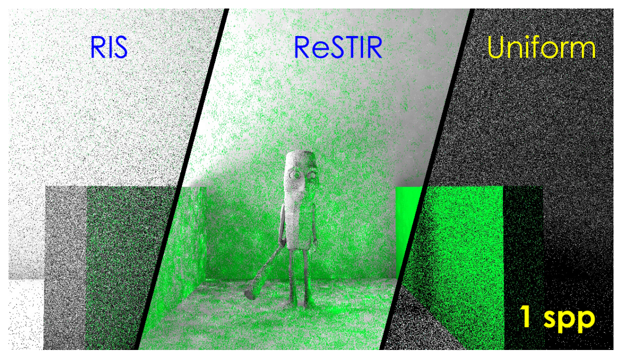

Figure 18: Comparison of sampling techniques using 1 sample per pixel (spp) in the Sahur scene. From left to right: uniform sampling, reservoir importance sampling (RIS), and ReSTIR.

C Detailed RMSE Results

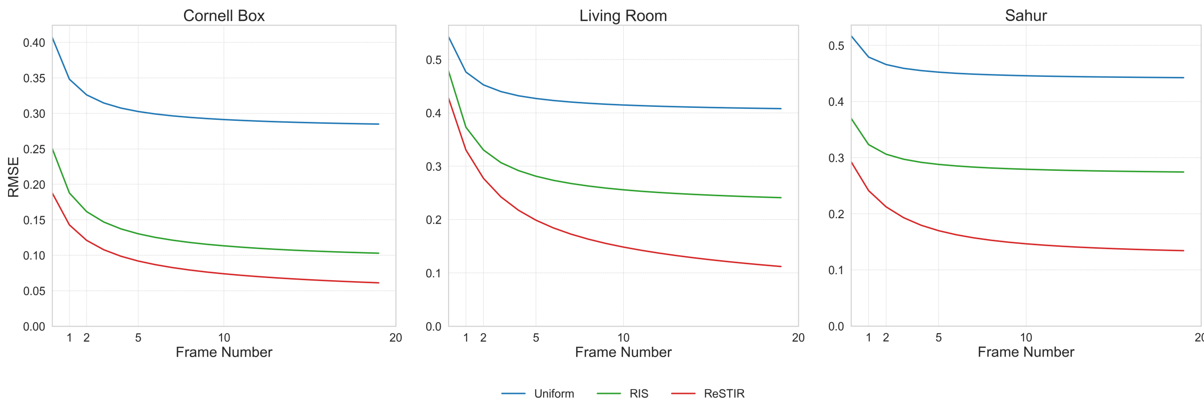

Figure 19: RMSE over 20 accumulated frames for three different scenes, Cornell Box, Living Room, and Sahur, comparing Uniform (blue), RIS (green), and ReSTIR (red) sampling methods. ReSTIR achieves lower RMSE across all scenes, indicating higher rendering quality. In more complex scenes, such as the living room and Sahur, ReSTIR converges noticeably faster than RIS, suggesting improved efficiency in handling difficult lighting scenarios. For the simpler Cornell Box scene, both ReSTIR and RIS show similar convergence rates, with a smaller performance gap.

Cornell Box					
Sampling Technique	t=1	t=2	t=5	t=10	t=20
Uniform	0.41	0.35	0.31	0.29	0.28
RIS	0.25 (-39.0%)	0.19 (-45.7%)	0.14 (-54.8%)	0.12 (-58.6%)	0.10 (-64.3%)
ReSTIR	0.19 (-53.7%)	0.14 (-60.0%)	0.10 (-67.7%)	0.08 (-72.4%)	0.06 (-78.6%)

Living Room					
Sampling Technique	t=1	t=2	t=5	t=10	t=20
Uniform	0.54	0.48	0.43	0.42	0.41
RIS	0.48 (-11.1%)	0.37 (-22.9%)	0.29 (-32.6%)	0.26 (-38.1%)	0.24 (-41.5%)
ReSTIR	0.43 (-20.4%)	0.33 (-31.3%)	0.22 (-48.8%)	0.16 (-61.9%)	0.11 (-73.2%)

Sahur Scene					
Sampling Technique	t=1	t=2	t=5	t=10	t=20
Uniform	0.52	0.48	0.46	0.45	0.44
RIS	0.37 (-28.8%)	0.32 (-33.3%)	0.29 (-37.0%)	0.28 (-37.8%)	0.27 (-38.6%)
ReSTIR	0.29 (-44.2%)	0.24 (-50.0%)	0.18 (-60.9%)	0.15 (-66.7%)	0.13 (-70.5%)

Table 2: RMSE over 20 accumulated frames for three different scenes: Cornell Box, Living Room, and Sahur. Percentage improvements shown for RIS and ReSTIR relative to the Uniform baseline.

References

[1] DreamWorks Animation. *Direct vs. Indirect Lighting Comparison*. Presented at SIGGRAPH 2010. Image courtesy of DreamWorks Animation, shown in SIGGRAPH 2010 materials. 2010. URL: https://www.siggraph.org/.

[2] Jacco Bikker. *TinyBVH*. Version 1.5.5. URL: https://github.com/jbikker/tinybvh.

[3] Benedikt Bitterli et al. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting". In: *ACM Trans. Graph.* 39.4 (Aug. 2020). ISSN: 0730-0301. DOI: 10.1145/3386569.3392481. URL: https://doi.org/10.1145/3386569.3392481.

[4] Jose Luis Blanco and Pranjal Kumar Rai. *nanoflann: a C++ header-only fork of FLANN, a library for Nearest Neighbor (NN) with KD-trees*. https://github.com/jlblancoc/nanoflann. 2014.

[5] Michael F. Cohen et al. "A radiosity solution for complex environments". In: *Proceedings of the 11th annual conference on Computer graphics and interactive techniques (SIGGRAPH '84)*. 1984, pp. 323–331. DOI: 10.1145/800031.808601.

[6] dylanheyes. *White Modern Living Room*. https://sketchfab.com/3d-models/white-modern-living-room-afb8cb0cbee1488caf61471ef14041e9. Sketchfab. 2023.

[7] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. Natick, MA, USA: A K Peters, Ltd., 2001. ISBN: 1-56881-147-0.

[8] Alexander Keller. "Instant radiosity". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 49–56. ISBN: 0897918967. DOI: 10.1145/258734.258769. URL: https://doi.org/10.1145/258734.258769.

[9] Daqi Lin et al. "Generalized resampled importance sampling: foundations of ReSTIR". In: *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301. DOI: 10.1145/3528223.3530158. URL: https://doi.org/10.1145/3528223.3530158.

[10] Y. Ouyang et al. "ReSTIR GI: Path Resampling for Real-Time Path Tracing". In: *Computer Graphics Forum* 40 (Nov. 2021), pp. 17–29. DOI: 10.1111/cgf.14378.

[11] Christoph Peters. *My toy renderer, part 4: Ray tracing*. Blog post discussing ray traced shadows with polygonal and linear lights for diffuse and specular surfaces. 2021. URL: https://momentsingraphics.de/ToyRenderer4RayTracing.html.

[12] Daniel Scherzer et al. "A Survey on Temporal Coherence Methods in Real-Time Rendering". In: *A Survey on Temporal Coherence Methods in Real-Time Rendering*. Jan. 2011.

[13] SDL Development Team. *Simple DirectMedia Layer 2*. Latest stable release; zlib license. SDL Development Team. 2025. URL: https://www.libsdl.org.

[14] Florian Simon, Johannes Hanika, and Carsten Dachsbacher. "Rich-VPLs for Improving the Versatility of Many-Light Methods". In: *Computer Graphics Forum* 34.2 (2015), pp. 575–584. DOI: https://doi.org/10.1111/cgf.12585. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12585. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12585.

[15] Justin Talbot, David Cline, and Parris Egbert. "Importance Resampling for Global Illumination". In: *Eurographics Symposium on Rendering (2005)*. Ed. by Kavita Bala and Philip Dutre. The Eurographics Association, 2005. ISBN: 3-905673-23-1. DOI: /10.2312/EGWR/EGSR05/139-146.

[16] Yusuke Tokuyoshi. "Virtual spherical gaussian lights for real-time glossy indirect illumination". In: *Virtual spherical gaussian lights for real-time glossy indirect illumination*. New York, NY, USA: Association for Computing Machinery, 2014. ISBN: 9781450328951. DOI: 10.1145/2669024.2669025. URL: https://doi.org/10.1145/2669024.2669025.

[17] Cem Yuksel. "Point Light Attenuation Without Singularity". In: *ACM SIGGRAPH 2020 Talks*. SIGGRAPH 2020. New York, NY, USA: ACM, Aug. 2020. DOI: 10.1145/3388767.3407364. URL: https://doi.org/10.1145/3388767.3407364.