Recursive Global Illumination Using Reflective Sphere Maps and Radiance Cascades—Research Plan

6112ICT Research Methods in IT, Griffith University

Nathan Burg
Griffith University

1 Background

Lighting in computer graphics is an exceptionally broad topic with no universal solution. There are two primary modes of calculating lighting: offline and real-time. Offline calculation, often called baking, is the most general form where a geometry set is defined, and an algorithm is used to calculate lighting for a scene. This mode has no need for interactivity and has no upper bound in computation time, with some algorithms taking several days to render a single image. Real-time lighting operates under the constraint that lighting must be calculated within some fraction of a per-frame time budget. Early real-time graphics had more generous frame time budgets, sometimes upwards of 36 milliseconds (24 frames per second), but modern frame times are shorter, in line with hardware improvements and consumer expectations over time, getting as low as 4 milliseconds (240 frames per second). Despite hardware improvement, achieving these frame times, while also improving resolution, geometric complexity, and quality of effects like lighting, often forces trade-offs. As lighting cost is often coupled to both resolution and geometry, the impact of inefficient lighting algorithms becomes very pronounced at scale.

Direct lighting in real-time graphics has changed very little since its original formulation [1]. Analytical lights—including point lights, directional lights and spot lights—provide a closed form solution to lighting with acceptable quality and can be precisely controlled within a scene. Advancements in analytical lights are typically techniques allowing more lights to be evaluated within a frame budget, such as acceleration or partitioning structures [2]. Analytical lights also require a separate rendering pass to calculate occlusion fields (shadowing). For standard rasterisation pipelines, this process is called shadow mapping [3], which becomes prohibitively expensive if many shadowed lights are required. Analytical lights also cannot solve indirect lighting, which is the light bounced from directly lit surfaces. In scenes with many light interactions, a technique is preferred that can solve the complete radiance field for all emitters and occluders of any shape or size—this is the problem of global illumination.

Global illumination describes the set of techniques in computer graphics aimed to solve the rendering equation [4]. The rendering equation is a five-dimensional recursive integral with no closed form solution, therefore these techniques aim to approximate a solution using various numerical methods. The rendering equation is derived from the radiative transfer equation [5], which is the volumetric integro-differential form of the same transport mechanics. Although the rendering equation has fewer unknowns in theory, the variance of the hemisphere under rotation generally makes the rendering equation more difficult to solve directly.

The most common numerical method to solve the rendering equation is stochastic path tracing [6], which applies the Monte Carlo estimator over the hemisphere to accumulate incoming radiance at a surface point. Modern graphics hardware, like NVIDIA RTX, makes stochastic path tracing viable in real-time graphics [7]. This has led to the near-universal adoption of ray tracing as the basis of contemporary global illumination techniques, such as ReSTIR [8] [9], Dynamic Diffuse Global Illumination (DDGI) [10] [11], and Global Illumination Based on Surfels (GIBS) [12]. While these methods are a significant improvement in real-time global illumination, they are all practically limited by the Monte Carlo estimator, and require temporal accumulation to produce the correct result. These algorithms can take several hundreds of milliseconds across several frames to correctly light a scene, and temporal accumulation itself has a computation and memory cost [13]. These constraints motivate the need for alternative methods.

2 Literature Review

2.1 Lighting in Real-Time Graphics

Direct Lighting Analytical lights can be understood as the standard lighting model in 3D graphics. These lights are represented as a finite set of vectors with no geometry, and when rendering a scene, visible surfaces will calculate approximate visibility and attenuation relative to these vectors as a discrete sum. This calculation does not consider occlusion, meaning these lights are not shadow-casting. To solve this, a technique called shadow mapping is used, where the scene is rendered from the position of the light to record the depth buffer for occlusion testing [3]. This creates a coupled scaling problem in analytical lights—calculating illumination scales with the number of lights per geometry unit, while calculating occlusion scales with the amount of geometry per light unit. The cost of shadow-casting analytical lights grows very quickly, even in modern renderers, to the extent the practical limit of simultaneous shadow-casting lights is between 8 and 16.

There are three primary modes in analytical lighting, with varying cost. Directional lights generally represent light from the sun, and calculate irradiance with local orientation relative to the direction vector described by the light, while ignoring the position of the surface. Calculating occlusion of directional lights is the most common application of shadow mapping, where an orthographic projection is used to render a single pass of the scene. Point lights are more localised lights, such as lamps, that are directionally uniform and calculate attenuation based on distance and normal orientation visibility to a point in space. Calculating occlusion for point lights requires capturing the scene 6 times to construct a cubemap, which is prohibitively expensive, and in practice point light shadowing is only used for set pieces within a scene. Spot lights contain both direction and position vectors. Calculating illumination for spot lights introduces a geometric attenuation term, which generally involves calculating intersection with an oriented cone, but this can be modified to more complex shapes like cosine lobes. Occlusion is captured in one pass from the position of the light, in the direction the cone is oriented, generally as an oriented bounding volume of the light shape using perspective projection.

Area lights are an extension of point and directional lights in which emission is distributed over a finite surface rather than concentrated at a single point or direction. Illumination from an area light is therefore expressed as an integral over its emitting surface. This form does not have an analytical solution in the general case, and requires numerical sampling methods to approximate [14] [6]. Occlusion testing is also more complicated—for point lights, occlusion testing collapses to a single ray intersection test, while for area lights visibility must be evaluated over a continuous 2D emitter manifold. This defines a swept volume between the point and the light surface, where occluding geometry blocks a subset of the light paths over the area, and contributes to partial shadowing of the emitter. This integral is also solved with numerical methods.

The Rendering Equation While analytical lights are effective tools for structured scenes, they cannot easily generalise to all light interactions. Global illumination expands the explicit emitter model of analytical lights into an implicit emitter and transport coupling model. The rendering equation models light transport as a recursive integral equation:

$$L_o(\mathbf{x}, \omega) = L_e(\mathbf{x}, \omega) + \int_\Omega f_r(\mathbf{x}, \omega_i, \omega) L_i(\mathbf{x}, \omega_i)(\mathbf{n} \cdot \omega_i) d\omega_i \tag{1}$$

This equation has two components that make it difficult to evaluate. The solid-angle integral $\int_\Omega$ is a continuous operator that must be numerically approximated. In graphics, this distinguishes several classes of numerical methods, including stochastic sampling (Monte Carlo and Quasi-Monte Carlo sequences), deterministic quadrature sets, and spectral expansions like spherical harmonics. Furthermore, the incident radiance term L_i introduces a recursive dependence for L_o—if we define $L_o(\mathbf{x}, \omega)$ as the radiance leaving a surface, then $L_i(\mathbf{x}, \omega_i) = L_o(\mathbf{x}', -\omega_i)$, where $\mathbf{x}'$ is the first visible surface along direction ω_i. This couples radiance across all surfaces, and leads to unbounded multi-bounce light transport.

Numerical Methods Solving the rendering equation requires approximating its linear operators. Global illumination algorithms generally discretise the solid angle integral and truncate the recursive coupling to produce practical solutions. To approximate the solid angle integral, we observe that a continuous integral can be approximated by a weighted sum of discrete ordinates within the domain of the integral

$$\int_\Omega f(\omega)\, d\omega \approx \sum_{i=1}^{N} w_i\, f(\omega_i) \tag{2}$$

when substituted into the rendering equation, the linear integral operator is approximated by a finite weighted sum over sampled directions

$$L_o(\mathbf{x}, \omega) = L_e(\mathbf{x}, \omega) + \sum_{i=1}^{N} w_i f_r(\mathbf{x}, \omega_i, \omega) L_o(\mathbf{x}', -\omega_i)(\mathbf{n} \cdot \omega_i) \tag{3}$$

This discretisation effectively encompasses Quasi-Monte Carlo sampling [15], horizon-based and similar quadratures [16] [17] [18], and standard Monte Carlo sampling. As N increases, the approximation converges toward the solution of the integral operator. To approximate the recursive term, we define T as the transport operator

$$(TL)(\mathbf{x}, \omega) = \int_\Omega f_r(\mathbf{x}, \omega_i, \omega) L(\mathbf{x}', -\omega_i)(\mathbf{n} \cdot \omega_i) d\omega_i \tag{4}$$

which generalises the rendering equation to

$$L = L_e + TL \tag{5}$$

then the finite-bounce form of the rendering equation can be understood as a truncated order of the Neumann series for this operator, simplifying the rendering equation for practical solutions to N-bounce lighting.

$$L_N = \sum_{k=0}^{N} T^k L_e \tag{6}$$

Applying both of these approximations reduces the rendering equation to a finite-dimensional coupled linear system. In computational physics, related transport problems are often formulated as discrete ordinates (S_N) [19] or spherical harmonic (P_N) [20] projections, which correspond to large linear systems in angular coefficients that are solved using standard linear system solvers. In contrast, real-time graphics generally operate under strict latency constraints, making globally coupled transport solves impractical at interactive resolutions. Graphics algorithms generally utilise bounded-cost iterative or stochastic estimators [21] [6], that trade physical accuracy for predictable performance.

2.2 Radiance Field Modelling

Ray Tracing Ray tracing assumes global visibility queries are available. Naively, this can be achieved with a brute force intersection test over all geometries, but in practice this structure is generally a tree to enable logarithmic spatial acceleration. Modern formulations in real-time graphics have adopted the Bounding Volume Hierarchy (BVH) as an acceleration structure for ray-triangle intersection [22]. Ray tracing is favourable in the case of perfect specular transport because the solid angle integral is collapsed to a single reflection vector, and each ray cast into a scene has a deterministic transport path [23]. The diffuse case is the worst-case in ray tracing, as the single transport direction expands into the recursive hemisphere integral of the rendering equation. Although ray tracing operates over continuous geometry in three-dimensional space, transport is evaluated along discrete sampled paths, making diffuse transport with high angular resolution prohibitively expensive even in simple scenes. Modern stochastic techniques like ReSTIR attempt to address this by amortising the convergence of diffuse transport over time, but certain categories of transport like reflective caustics have near-zero sampling probability. Ray tracing does not statistically favour this transport due to the combinatorial explosion of sampling paths [6].

Virtual Point Lights Virtual Point Lights (VPLs) refer to the image-space projection of lights onto surfaces. Evaluating standard point lights at a surface involves only a discrete sum across a set of known vectors. In the case of VPLs, the discrete light vectors are unknown. In the original formulation used in the instant radiosity algorithm [15], a linear search is used to approximate a subset of the total VPLs. Particularly, a quasi-Monte Carlo walk based on the Halton sequence is used to discretise the hemisphere into N rays and approximate attenuation from discrete samples on the VPL texture. Solving VPLs is not strictly limited to quasi-Monte Carlo sampling—standard Monte Carlo, and other discretisations, can also resolve VPLs with varying quality and performance. Most image-based lighting techniques, like screen space and environment mapping, can be loosely defined as an application or extension of virtual point lights, where pixels in a projected texture compose the radiance field.

Screen Space Screen space techniques can be understood as an application of instant radiosity, where the shaded surface is also the VPL surface. The earliest forms of screen space global illumination are in the form of horizon-based ambient occlusion [18], which uses the surface normal and depth buffer as a geometry proxy to approximate local visibility. This idea is extended to screen space reflection [24], which treats the depth buffer as a heightfield to cast a ray through visible geometry, resolving specular reflection at the hit point. Horizon-based indirect lighting [25] [26] and screen space global illumination [27] extend this again, using a finite quadrature to cast rays into the scene and calculate indirect lighting. Screen space ray marching is effectively analogous to VPL sampling, and screen space lighting techniques extend instant radiosity by approximating occlusion using the depth buffer. The primary limitation of screen space techniques is information—visible surfaces are evaluated as the complete radiance field, so off-screen or occluded emitters are lost.

Environment Mapping Environment mapping is a method of calculating approximate global illumination by omitting the spatial term in the rendering equation. In this formulation, the radiance field is evaluated as a spherical function at infinite distance. Environment maps most simply represent specular reflection [28], but an equivalent diffuse map can be constructed by downsampling the environment map, blurring the high frequency radiance field. Different environment mapping techniques introduce new terms or other optimisations that make it more useful in modern graphics.

The standard representation of environment maps is the cubemap, but other representations often introduce favourable properties. For example, equirectangular projection and octahedron unwrapping [29] stores the sphere within a single 2D texture. Octahedrons are typically preferred as the sample distribution is more even, and mapping a direction to octahedron is less computationally expensive. Environment maps can also be represented by spherical harmonic functions [30]. For low-frequency diffuse lighting, an entire environment map may be collapsed into just 9 spherical harmonic coefficients.

Parallax corrected cubemaps [31] are a modern technique that defines proxy geometry to fit the sphere to, which can be an axis-aligned bounding box (AABB), an oriented bounding box (OBB), or some other definition across multiple bounding boxes, including signed distance functions (SDFs) [32]. This allows analytical fitting of geometry for very efficient local illumination in specific scenes, which differs from the approach of distance maps.

Distance impostors are a cubemap probe that encodes world-space distance as a buffer, allowing distance to first surface to be found in any direction using the hardware cubemap sampler. The original form uses this distance encoding as proxy geometry for parallax-correct environment mapping [33]. Later formulations were extended to include refraction and caustic transport [34], and then again to utilise depth peeling [35]. Depth peeling partially solves the depth buffer occlusion problem by introducing more layers of geometry, though this makes the technique much more expensive in practice. In production renderers, this technique has little adoption, where screen space techniques and analytical parallax-corrected cubemaps are generally preferred.

Reflective Shadow Maps Reflective shadow maps (RSMs) extend standard shadow mapping techniques by using the standard shadow pass of analytical lights to generate VPLs within the shadow map [36]. Standard sampling strategies can be used to sample the RSM, and in terms of information, VPLs outside the view frustum or occluded by geometry will properly contribute to the radiance field. RSMs are limited to geometry directly visible to the light, meaning in the simplest case where only the global shadow map (sun) is reflective, this technique will not work indoors and further analytical lights are required.

Light Propagation Volumes Light propagation volumes (LPVs) can be understood as a sampling strategy for reflective shadow maps [37] [38]. At a high level, LPVs are a uniform grid of irradiance probes in world space that store 2nd-order spherical harmonic coefficients. To populate these probes, a reflective shadow map will aggregate its virtual point lights into whatever grid cell it falls into, calculating a local spherical harmonic basis at each grid cell. In the propagation pass, these coefficients are propagated through the grid using a method similar to short characteristics sweeping. LPVs are mechanically similar to SHDOM in computational physics [39], and is one of the few global illumination algorithms that does not inherently rely on temporal accumulation or stochastic sampling. LPVs typically fail around high-frequency geometry, as the spherical harmonic basis and probe grid cannot resolve high-frequency occlusion.

2.3 Radiance Cascades

Definition Radiance cascades are an alternative numerical structure formulated to exploit the observation that angular error of a finite quadrature increases with distance from a sample point, while the effect of spatial error decreases [40]. This observation is called the penumbra condition. Transport literature had previously identified this property [41], however radiance cascades is designed to exploit it. If we use an adaptive quadrature that increases the angular resolution with distance, we can effectively mitigate the increase in angular error, but practically, tracing ray segments is equally expensive to tracing full length rays, making this discrete segmentation redundant. Transport solvers typically assume that the sampling position is invariant [42], while radiance cascades deliberately introduces a small amount of spatial error to construct a probe hierarchy, which distributes one probe at the current interval level across 4 or more probes at the lower level. This spatial error introduces numerical diffusion, which can be favourable in optically thick media [43]. With this reconstruction, each cascade level doubles the number of logical rays, while the angular scaling halves the number of physical ray segments traced at each level. Radiance cascades were originally designed as a global illumination solution in graphics, but further application demonstrates general application to transport problems.

NLTE Radiative Transfer As radiance cascades formulate volumetric transport in the general case, applying the structure to the non-LTE radiative transfer equation is an emergent use case in computational astrophysics [44]. When compared to short and long characteristics solvers of the discrete ordinates system, radiance cascades completely mitigate the ray effects artefact.

Global Illumination The original radiance cascades formulation was designed for global illumination in computer graphics. Radiance cascades enable the approximation of radiance fields with no temporal accumulation or stochastic sampling, much like LPVs, but the structure of radiance cascades permit greater angular and spatial resolution almost universally. There is some flexibility in the modes available depending on scene requirements and hardware. The standard formulation solves 2D global illumination in the volumetric form, which does not generalise well to 3D due to memory cost. Practical 3D implementations only solve volumetric transport near surfaces, as solving the full radiance field is prohibitively expensive.

The screen space [45] and screen-probes, world-intervals (SPWI) [46] formulations construct probe hierarchies on a manifold of the depth buffer as a heightfield. Radiance is gathered using either screen-space information, or with ray tracing, and merged using a double bilinear interpolation based on position relative to parent probes on lateral and depth axes. There is also a two-layer min-max formulation used to mitigate depth discontinuity error that arises from interval merging near boundaries. These distributions are more like a 2D solution in practice, and are both prone to information loss and instability in motion. The screen space formulation with the min-max double manifold formulation is used in Path of Exile 2, which avoids this instability by using a fixed-angle camera. Convolution prefiltering [47] is an optimisation used by the screen space method to approximate cone tracing in screen space rays [48], which improves coverage for each ray, similar to voxel cone tracing techniques [49].

The sparse formulation uses a hashmap-backed sparse octree in 3D to localise high spatial resolution probes around surfaces [50], using upper cascades to distribute computation across nearby surfaces. This method is mechanically similar to surfels-based methods, but querying this structure is much simpler, and more performant in the general case. Sparse radiance cascades are invariant to camera motion, with similar memory cost to screen space probes, but are more expensive computationally.

3 Research Questions

Existing formulations of global illumination based on radiance cascades exclusively use either screen space ray marching or ray tracing to measure radiance, while the literature reveals several methods that do not have the same limitations as these techniques. The broad question is ‘how can an image-space radiance field be integrated with radiance cascades to achieve high-fidelity, multi-bounce global illumination within real-time hardware constraints?’

This question is motivated by the existing techniques—screen space is severely information limited, and ray tracing is too expensive to update in real-time. The proposed method also motivates two sub-questions to be addressed:

1. What effect does combined projection and interpolation (octahedron encoding and convolution prefiltering) have on the coverage, accuracy and performance of ray accumulation in global illumination problems?

2. How does the proposed formulation scale, compared to other common techniques, in computation cost, memory cost, and image quality, and what are the failure modes of each formulation?

4 Aims and Significance

The primary **aim** of this research is to develop and evaluate a novel global illumination framework that decouples the cost of radiance field integration from scene complexity. This study aims to:

1. Implement and refine the technique to a local optimum through empirical observation, within a typical real-time rendering environment, to evaluate practical utility when paired with other rendering systems like particles and animation.

2. Develop a unified testing environment within NVIDIA Falcor to quantify the performance and fidelity trade-offs of this approach against state-of-the-art techniques.

The **significance** of this work arises when considering the techniques the proposed method relies on. The techniques used between 1995 and 2005 were designed for significantly weaker hardware than is available today, so if we can deliver a mathematically elegant solution that has the potential to outperform modern algorithms, we can generalise global illumination to most consumer hardware without temporal accumulation, frame generation or upscaling. The technique is also based purely on raster graphics, which means ray acceleration hardware like NVIDIA RTX is not required, and that it is theoretically suitable for mobile devices.

5 Methodology

This research works within the positivist paradigm [51], pursuing objective and reproducible outcomes through controlled experimentation. The methodology follows the Design Science Research (DSR) framework [52], in which the primary contribution is the construction and evaluation of a novel artifact. The artifact is iteratively refined through empirical observation before quantitative evaluation against state-of-the-art techniques.

5.1 Proposed Formulation

The specific formulation we wish to test is distance maps, used as both a reflective shadow map and local occlusion proxy, integrated into screen space hierarchy of radiance cascades. The objective is to find a dynamic global illumination algorithm that does not rely on ray tracing hardware, stochastic sampling, or temporal accumulation. The basis of this formulation is the quality and speed of the ray approximation in distance map ray marching, and retention of off-screen information, which together effectively extends screen space techniques with similar performance and improved quality.

We also wish to extend the use case of octahedron environment maps, as the performance and memory cost is significantly improved over cubemaps in the general case, and with this form we can test reprojection of screen-space irradiance into the octahedron proxy to approximate multi-bounce global illumination.

Proposed Graphics Pipeline The complete form of the solution, at a high level, can be understood as several working components.

1. The initial implementation is a reformulation of the SPWI method discussed in Section 2.3, so we must generate the min-max probe manifold by downsampling the screen space depth buffer.
2. Capture a camera-local distance map of the scene, which becomes our global radiance source.
3. Convert the camera-local distance map from a cubemap to an unwrapped octahedron, then perform depth-aware downsampling on this texture to generate a mip chain.
4. Generate radiance cascades for the min-max probe manifold by ray marching into the distance map. The sampled mip level should correspond to the current cascade, increasing coverage of coarse samples to improve performance.
5. Using the final merged probes, for each pixel, calculate the hemisphere contribution of the 8 nearest probes on the min-max manifold.
6. Optionally, reproject this final irradiance into the distance map, and repeat the process N times.

5.2 Artifact Construction

Vulkan Renderer A custom graphics engine has been developed for the purpose of practical integration. In a practical rendering environment, we can refine the technique by evaluating common edge case scenarios that would not appear in static testing environments. The renderer is implemented in C, using the Vulkan graphics API for cross-platform support, GLFW as a windowing abstraction, and Open Dynamics Engine for dynamic scene experimentation. The renderer uses a modern instancing-first approach with bindless resources, deferred shading and composable frame graph. The renderer also includes live reloading of shader programs to reduce friction in experimentation. To satisfy this experiment, we must pass some practical, observable test cases:

1. The implementation should exhibit the basic emission and occlusion properties of a global radiance field, including correct isotropic energy attenuation and penumbra shadow casting, in line with formal radiance cascades.
2. The implementation should produce a stable simulation with minimal artefacts when the camera or scene moves or rotates.
3. The implementation should be stable across various scene geometries at varying distances. The simulation should not deform at grazing angles, near distances, far distances, and both high and low depth complexity scenes.

If we can cover these qualitative cases and refine the method to a local optimum, we can understand the quality and performance characteristics of the method in real rendering environments. This practical application should include test scenes that cover these cases appropriately.

Test Scenes While forming a complete interactive application like a game is not feasible within the scope of this project, we can use open-source test scenes to evaluate the quality of our solution in various scenarios. In graphics literature, there are many sample models and scenes used to test various lighting models.

Choosing the correct models depends on the strengths and weaknesses of our solution. The Cornell box is a common test scene in computer graphics [21], which can demonstrate the diffuse interreflection of surfaces. For our method, the result would be identical to a screen space implementation, as the Cornell box fits entirely within the view frustum. This can be interpreted as both a limitation and a benchmark, as visual parity with established methods will be a positive indicator of correct transport.

The Sponza atrium model [53] is another scene commonly used in global illumination research, as it tests several scenarios simultaneously. The center of the atrium can be directly lit by sunlight, with brightly coloured curtains contributing to indirect lighting. The first floor corridors are unlit, but have small openings around the curtains. The second floor has many pillars, which introduce depth discontinuity and high-frequency transitions between lit and unlit space. Sponza can effectively stress test light leaking on the first floor, and surface stability on the second, making it an important scene in computer graphics research.

The Amazon Lumberyard Bistro [54] is a large outdoor scene with a varying geometric detail, useful for evaluating global illumination in a realistic scenario. It combines large, open spaces, large buildings, archways, foliage and very thin geometry like fences and lamp posts. As this presents many tests at once, we can quickly identify the most significant failure modes in practical scenarios. The Bistro Interior is an extension of this scene, which is a small room with very high frequency geometry that image-space techniques would struggle to resolve.

The purpose of the initial implementation is to gather qualitative data for the technique, and catch edge cases that may disrupt our assumptions about the problem.

Falcor Renderer To evaluate how our method compares to both the ground truth solution and the current state of the art techniques, quantitative analysis on constant, replicable scenes is required. NVIDIA Falcor [55] has an open-source ecosystem of different global illumination solutions, which will allow quick replication of these techniques in a uniform environment. Adapting our technique for this renderer is far simpler than adapting all other techniques for our renderer. We can use the same test scenes in this renderer, however either a fixed camera position or a carefully authored and replicable camera path are required to demonstrate different techniques fairly.

WebGL Renderer The final artifact is a live WebGL implementation of the technique as an appendix, that supports its generality through an observable and practical demonstration. This implementation will help us understand the performance characteristics on different hardware, including mobile devices, but it is not the aim of this study to extend the qualitative or quantitative analysis of the method to this implementation. The Vulkan and Falcor renderers cannot be trivially replicated and deployed across devices, while the WebGL version makes this process effortless, therefore it is an ideal appendix of the research.

5.3 Evaluation

As global illumination techniques are generally an approximation of ground truth, we must rely on both qualitative and quantitative metrics to derive an optimal solution. Combined, these metrics will provide evidence for or against the proposed solution.

Qualitative Certain metrics are difficult to quantify practically. Specific failure modes will be unique to implementation, meaning their evaluation is largely perceptual and qualitative. For the early experimentation phase, metrics like perceived stability in motion must be evaluated by direct observation. In the comparative analysis, this can extend to disjoint metrics like ghosting in ReSTIR and stability in our method, where the impact of these artefacts require the subjective evaluation.

Quantitative Quantitative analysis can be used when comparing techniques directly and on the same basis. In this case, the primary quantitative measure will be PSNR to the ground truth—which is standard in graphics literature—where then we can compare it against other methods, and extend to other metrics like FLIP [56]. We can also profile the techniques to measure performance—as real-time graphics are time sensitive, frame time distributions for each technique are a useful metric to evaluate their performance. Finally, the WebGL implementation serves as a demonstrability evaluation, providing informal evidence of generalisability across hardware configurations without ray tracing acceleration.

5.4 Rigour, Validity, and Ethics

Rigour All research programs will be available as an appendix of the research, to ensure scientific reproducibility. The Vulkan and Falcor demos will be available to demonstrate the high-performance case, and the WebGL demo for the compatibility case. The three artifacts allow for diverse testing to validate the technique independent of specific hardware.

Validity Unified frameworks are utilised to ensure the experimentation is controlled. Standard and peer-reviewed quantitative analysis techniques (PSNR, FLIP [56]) for computer graphics evaluation will be used for all comparative metrics. Data will be generated on controlled and consistent hardware, with both a warmup period to ensure thermal throttling does not affect results, and average results will be used over thousands of individual samples to measure variance in this regard.

Ethics This research does not use human data and therefore is not meaningfully subject to ethical clearance in this regard. The three primary research artifacts are dependent on free and open source libraries, tools and assets. Licensing requirements of these utilities will be strictly observed. All performance data is generated through controlled algorithmic execution, where no results are selectively omitted or fabricated to favour the proposed method.

6 Research Schedule

This research is divided into three sequential stages.

- **Stage 1 (3 months) – Literature Review** A comprehensive literature review is developed to unify global illumination and transport theory, and identify techniques that will apply to the problem of real-time global illumination. This has been completed over the preceding three months.
- **Stage 2 (1 month) – Experimentation** In this stage we test different configurations to find a local optimum of performance and perceived fidelity. As we already have a working renderer for this case, this is expected to take one month.
- **Stage 3 (3 months) – Analysis** The final phase of the research will be analysis. The Falcor environment will be developed to evaluate the optimised form of the technique against ground truth and state-of-the-art methods. This is expected to use the remaining three months of the academic year. This phase will also include development of the WebGL demonstration appendix.

References

[1] B. T. Phong, "Illumination for computer generated pictures," *Communications of the ACM*, vol. 18, pp. 311–317, June 1975.

[2] O. Olsson, M. Billeter, and U. Assarsson, "Clustered deferred and forward shading," in *Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics*, EGGH-HPG'12, (Goslar, DEU), pp. 87–96, Eurographics Association, 2012.

[3] L. Williams, "Casting curved shadows on curved surfaces," *ACM SIGGRAPH Computer Graphics*, vol. 12, pp. 270–274, Aug. 1978.

[4] J. T. Kajiya, "The rendering equation," *ACM SIGGRAPH Computer Graphics*, vol. 20, pp. 143–150, Aug. 1986.

[5] S. Chandrasekhar, *Radiative Transfer*. Dover Books on Intermediate and Advanced Mathematics, Dover Publications, 1960.

[6] E. Veach, *ROBUST MONTE CARLO METHODS FOR LIGHT TRANSPORT SIMULATION*. PhD, Stanford University, 1997.

[7] NVIDIA, "NVIDIA Turing GPU Architecture," tech. rep., NVIDIA Corporation, 2018.

[8] B. Bitterli, C. Wyman, M. Pharr, P. Shirley, A. Lefohn, and W. Jarosz, "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting," *ACM Transactions on Graphics*, vol. 39, pp. 148:1–148:17, Aug. 2020.

[9] Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, and J. Pantaleoni, "ReSTIR GI: Path Resampling for Real-Time Path Tracing," *Computer Graphics Forum*, vol. 40, pp. 17–29, Dec. 2021.

[10] Z. Majercik, J.-P. Guertin, D. Nowrouzezahrai, and M. McGuire, "Dynamic Diffuse Global Illumination with Ray-Traced Irradiance Fields," *Journal of Computer Graphics Techniques (JCGT)*, vol. 8, pp. 1–30, June 2019.

[11] Z. Majercik, A. Marrs, J. Spjut, and M. McGuire, "Scaling Probe-Based Real-Time Dynamic Global Illumination for Production," *Journal of Computer Graphics Techniques (JCGT)*, vol. 10, pp. 1–29, May 2021.

[12] H. Halen, A. Brinck, K. Hayward, and X. Guan, "Global Illumination Based on Surfels," Aug. 2021. Published: Presentation slides, ACM SIGGRAPH Courses: Advances in Real-Time Rendering in Games.

[13] D. Nehab, P. V. Sander, J. Lawrence, N. Tatarchuk, and J. R. Isidoro, "Accelerating real-time shading with reverse reprojection caching," in *Proceedings of the 22nd ACM SIGGRAPH/EUROGRAPHICS Symposium on Graphics Hardware*, GH '07, (Goslar, DEU), pp. 25–35, Eurographics Association, 2007.

[14] E. Heitz, J. Dupuy, S. Hill, and D. Neubelt, "Real-time polygonal-light shading with linearly transformed cosines," *ACM Transactions on Graphics*, vol. 35, pp. 1–8, July 2016.

[15] A. Keller, "Instant radiosity," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, (Not Known), pp. 49–56, ACM Press, 1997.

[16] K. Lathrop and B. Carlson, "DISCRETE ORDINATES ANGULAR QUADRATURE OF THE NEUTRON TRANSPORT EQUATION," Tech. Rep. LA-3186, Sept. 1964.

[17] B. Carlson, "TRANSPORT THEORY: DISCRETE ORDINATES QUADRATURE OVER THE UNIT SPHERE.," Tech. Rep. LA-4554, Jan. 1970.

[18] M. Mittring, "Finding next gen: CryEngine 2," in *ACM SIGGRAPH 2007 courses*, (San Diego California), pp. 97–121, ACM, Aug. 2007.

[19] B. Carlson, "NUMERICAL SOLUTION OF TRANSIENT AND STEADY-STATE NEUTRON TRANSPORT PROBLEMS," Tech. Rep. LA-2260, May 1959.

[20] J. H. Jeans, "The Equations of Radiative Transfer of Energy," *Monthly Notices of the Royal Astronomical Society*, vol. 78, pp. 28–36, Nov. 1917.

[21] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, "Modeling the interaction of light between diffuse surfaces," in *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pp. 213–222, ACM, Jan. 1984.

[22] T. L. Kay and J. T. Kajiya, "Ray tracing complex scenes," *ACM SIGGRAPH Computer Graphics*, vol. 20, pp. 269–278, Aug. 1986.

[23] T. Whitted, "An improved illumination model for shaded display," *Communications of the ACM*, vol. 23, pp. 343–349, June 1980.

[24] M. McGuire and M. Mara, "Efficient GPU Screen-Space Ray Tracing," *Journal of Computer Graphics Techniques (JCGT)*, vol. 3, pp. 73–85, Dec. 2014.

[25] B. Mayaux, "Horizon-Based Indirect Lighting," tech. rep., Mar. 2018.

[26] O. Therrien, Y. Levesque, and G. Gilet, "Screen Space Indirect Lighting with Visibility Bitmask," *The Visual Computer*, vol. 39, pp. 5925–5936, Nov. 2023. arXiv:2301.11376 [cs].

[27] T. Ritschel, T. Grosch, and H.-P. Seidel, "Approximating dynamic global illumination in image space," in *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, (Boston Massachusetts), pp. 75–82, ACM, Feb. 2009.

[28] J. F. Blinn and M. E. Newell, "Texture and reflection in computer generated images," *Communications of the ACM*, vol. 19, pp. 542–547, Oct. 1976.

[29] T. Engelhardt and C. Dachsbacher, "Octahedron Environment Maps," in *Vision, Modelling and Visualization (VMV 2008)*, 2008.

[30] R. Ramamoorthi and P. Hanrahan, "Frequency space environment map rendering," *ACM Transactions on Graphics*, vol. 21, pp. 517–526, July 2002.

[31] S. Lagarde and A. Zanuttini, "Local image-based lighting with parallax-corrected cubemaps," in *ACM SIGGRAPH 2012 Posters on - SIGGRAPH '12*, (Los Angeles, California), p. 1, ACM Press, 2012.

[32] J. C. Hart, "Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces," *The Visual Computer*, vol. 12, pp. 527–545, Dec. 1996.

[33] G. A. Patow, "Accurate Reflections Through a Z-Buffered Environment Map," tech. rep., LIFIA, Universidad Nacional de La Plata, Buenos Aires, Argentina, 1995.

[34] L. Szirmay-Kalos, B. Aszódi, I. Lazányi, and M. Premecz, "Approximate Ray-Tracing on the GPU with Distance Impostors," *Computer Graphics Forum*, vol. 24, pp. 695–704, Sept. 2005.

[35] C. Zhang, H.-H. Hsieh, and H.-W. Shen, "REAL-TIME REFLECTIONS ON CURVED OBJECTS USING LAYERED DEPTH TEXTURES," pp. 276–286, IADIS Press, 2008.

[36] C. Dachsbacher and M. Stamminger, "Reflective shadow maps," in *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, (Washington District of Columbia), pp. 203–231, ACM, Apr. 2005.

[37] A. Kaplanyan, "Light Propagation Volumes in CryEngine 3," in *ACM SIGGRAPH 2009 Courses: Advances in Real-Time Rendering in Games*, 2009.

[38] A. Kaplanyan and C. Dachsbacher, "Cascaded light propagation volumes for real-time indirect illumination," in *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, (Washington D.C.), pp. 99–107, ACM, Feb. 2010.

[39] K. F. Evans, "The Spherical Harmonics Discrete Ordinate Method for Three-Dimensional Atmospheric Radiative Transfer," *Journal of the Atmospheric Sciences*, vol. 55, pp. 429–446, Feb. 1998.

[40] A. Sannikov, "Radiance Cascades: A Novel Approach to Calculating Global Illumination[WIP]," tech. rep., 2023.

[41] F. Durand, N. Holzschuch, C. Soler, E. Chan, and F. X. Sillion, "A frequency analysis of light transport," *ACM Transactions on Graphics*, vol. 24, pp. 1115–1126, July 2005.

[42] H. P. Jones and A. Skumanich, "The Formation of Resonance Lines in Multidimensional Media. II. Radiation Operators and Their Numerical Representation," *The Astrophysical Journal*, vol. 185, p. 167, Oct. 1973.

[43] P. Kunasz and L. H. Auer, "Short characteristic integration of radiative transfer problems: Formal solution in two-dimensional slabs," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 39, pp. 67–79, Jan. 1988.

[44] C. M. J. Osborne and A. Sannikov, "Radiance cascades: a novel high-resolution formal solution for multidimensional non-LTE radiative transfer," *RAS Techniques and Instruments*, vol. 4, p. rzae062, Jan. 2025.

[45] G. A. Dan and M. Jensen, "Screen-Space Global Illumination Using Radiance Cascades in 3D Video Games," Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, 2025.

[46] K. Gavras, "Exploration and Optimization of Radiance Cascades for Real-Time Applications," Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, 2025.

[47] A. E. Fabris and A. R. Forrest, "Antialiasing of curves by discrete pre-filtering," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, (Not Known), pp. 317–326, ACM Press, 1997.

[48] Y. Uludag and W. Engel, "Hi-Z Screen-Space Cone-Traced Reflections," in *GPU Pro 5*, pp. 149–192, A K Peters/CRC Press, 0 ed., Dec. 2018.

[49] C. Crassin, F. Neyret, M. Sainz, S. Green, and E. Eisemann, "Interactive Indirect Illumination Using Voxel Cone Tracing," *Computer Graphics Forum*, vol. 30, pp. 1921–1930, Sept. 2011.

[50] D. A. Alcantara, A. Sharf, F. Abbasinejad, S. Sengupta, M. Mitzenmacher, J. D. Owens, and N. Amenta, "Real-time parallel hashing on the GPU," *ACM Transactions on Graphics*, vol. 28, pp. 1–9, Dec. 2009.

[51] Y. S. Park, L. Konge, and A. R. Artino, "The Positivism Paradigm of Research," *Academic Medicine*, vol. 95, pp. 690–694, May 2020.

[52] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, pp. 45–77, Dec. 2007.

[53] Khronos Group, "gltf sample assets: Sponza." `https://github.com/KhronosGroup/glTF-Sample-Assets/tree/main/Models/Sponza`, 2023. Originally created by Marko Dabrovic (2002), updated by Frank Meinl (Crytek, 2010), and adapted to glTF 2.0 PBR by the Khronos Group. Accessed: 2026.

[54] A. Lumberyard, "Amazon lumberyard bistro, open research content archive (orca)," July 2017. `http://developer.nvidia.com/orca/amazon-lumberyard-bistro`.

[55] S. Kallweit, P. Clarberg, C. Kolb, T. Davidovič, K.-H. Yao, T. Foley, Y. He, L. Wu, L. Chen, T. Akenine-Möller, C. Wyman, C. Crassin, and N. Benty, "The Falcor rendering framework," 8 2022. `https://github.com/NVIDIAGameWorks/Falcor`.

[56] P. Andersson, J. Nilsson, T. Akenine-Möller, M. Oskarsson, K. Åström, and M. D. Fairchild, "FLIP: A Difference Evaluator for Alternating Images," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 3, pp. 1–23, Aug. 2020.