

DDISH-GI: Dynamic Distributed Spherical Harmonics Global Illumination

Julius Ikkala, Petrus Kivi, Joel Alanko, Markku Mäkitalo, and
Pekka Jääskeläinen

Tampere University, P.O. Box 553, 33014 Tampere, Finland

Abstract. We propose a real-time hybrid rendering algorithm that off-loads computationally complex rendering of indirect lighting from mobile client devices to dedicated ray tracing hardware on the server with a hybrid real-time computer graphics rendering algorithm. *Spherical harmonics* (SH) light probes are updated with path tracing on the server side, and the final frame is rendered with a fast rasterization-based pipeline that uses the light probes to approximate high quality indirect diffuse lighting and glossy specular reflections. That is, the rendering workload can be split to multiple devices across the network with a small bandwidth usage. It also benefits multi-user and multi-view scenarios by separating indirect lighting computation from camera positioning. Compared to simply streaming fully remotely rendered frames, the approach is more robust to network interruptions and latency. Furthermore, we propose a specular approximation for GGX materials via *zonal harmonics* (ZH). This alleviates the need to implement more computationally complex algorithms, such as screen space reflections, which was suggested in the state-of-the-art *dynamic diffuse global illumination* (DDGI) method. We show that the image quality of the proposed method is similar to that of DDGI, with a 23 times more compact data structure.

Keywords: real-time rendering, path tracing, spherical harmonics, photorealistic rendering, distributed rendering, global illumination

1 Introduction

Dedicated ray tracing acceleration in graphics hardware has become available on NVIDIA RTX GPUs since 2018 [19] and on AMD GPUs as of late 2020, including the recently released video game console generation with AMD graphics hardware [2]. Even though this has made real-time ray and path tracing possible with a high frame rate and screen resolution, high-quality path tracing is still not fast enough for real-time. Denoising methods can be used to make lower-spp path traced images plausible [14,22]. They rely on frame reprojection and suffer from the resulting camera-dependent artefacts, like ghosting artefacts and occlusion issues. Moreover, the lower hardware capabilities of mobile devices, such as standalone *head mounted displays* (HMD), do not offer the performance of

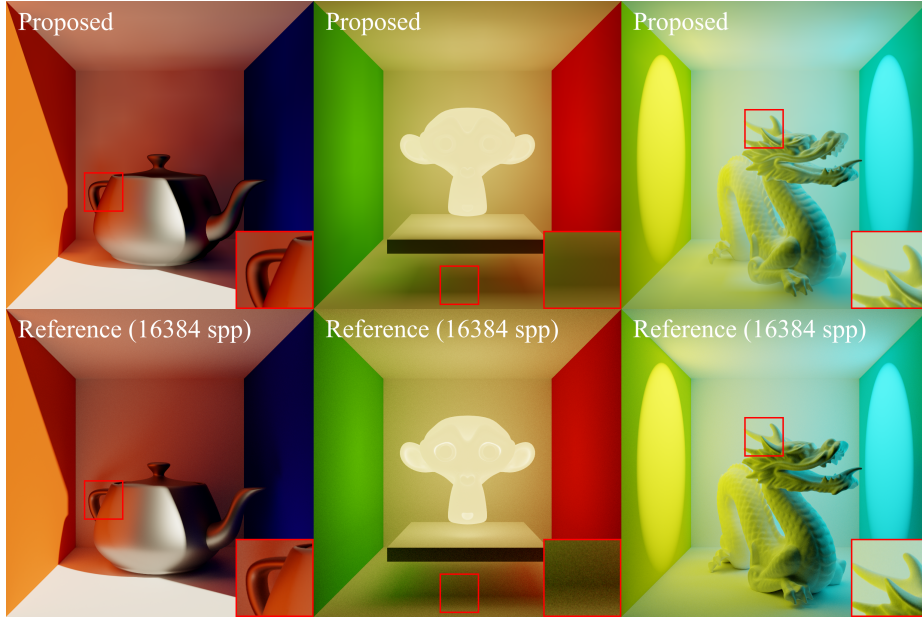


Fig. 1. Three example scenes highlighting different lighting phenomena simulated by DDISH-GI. From left to right: glossy reflections, emissive materials, and diffuse indirect lighting. The proposed method took 1.5 to 2.5 ms to render each image, while our hardware accelerated path tracer took 30 to 140 seconds for each 16384 spp reference image. The images are rendered at 1024×1024 with anti-aliasing implemented by $8\times$ MSAA for DDISH-GI and equivalent box filter sampling for path tracing.

desktop GPUs with highly dedicated hardware. Thus, it is interesting to distribute the rendering effort over the network, specifically with methods that use minimal bandwidth and are tolerant to latency issues.

Various systems for offloading the rendering effort from computationally restricted devices have been developed, including *PlayStation Now* [26] and *Google Stadia* [9]. These streaming services transfer the user input from the client side, update the game state accordingly, perform the actual rendering, and stream the output frames as image sequences back to the client. Even though the streamed frames are compressed with state-of-the-art techniques to lower the bitrate, bandwidth issues still become relevant with higher frame resolutions and refresh rates. Furthermore, the system is entirely reliant on the server and underutilizes the client hardware. In our case, we can fully rely on local rendering hardware if remote rendering resources become unavailable. The game streaming systems along with the rendering pipelines are inherently prone to network problems, including latency issues affecting the interactivity and immersion, as well as total network failure stopping the experience altogether. Moreover, providing photorealistic rendering on HMDs and other multi-view devices and techniques, such as light field displays, requires duplicating the rendering effort for multi-

ple viewports. HMDs have a display for each eye with a typical resolution of 1440×1600 and a refresh rate of 90 Hz [3]. Furthermore, high-end models, such as the Varjo VR-3 [29], exhibit two 1920×1920 and two 2880×2720 displays. For a near-eye light field display, a 5×5 array of 640×800 resolution viewports per eye was demonstrated in [11], yielding a total pixel count of over 25 million, which is 5 to 6 times the amount of a typical HMD. Lighting conditions are shared between the views and, thus, the view-independent aspects of the scene can be rendered irrespective of the viewports.

To tackle the presented challenges, we propose a real-time distributable rendering method that updates *spherical harmonics* (SH) light probes with *path tracing*, and uses them for *global illumination* (GI) approximation and glossy GGX-BRDF-based reflections. Figure 1 highlights some of the different lighting phenomena that the proposed method simulates. The heavy path tracing can be distributed to a server-side desktop system, which streams the dynamically updated SH probe coefficients to be used by a traditional rasterization pipeline on a lightweight mobile client device.

A key novelty of the proposed DDISH-GI method is in its use of the low-order SH basis for real-time path traced probes, which allows a very compact representation with rotational invariance, low network bandwidth and no high-frequency artefacts (Section 3). Another novel aspect is how it uses the light probes together with fitted *zonal harmonics* (ZH) representation of GGX (BRDF) lobes, specifically, for indirect glossy reflections (Appendix A), which reduces the need for heavy computation on the client side. We discuss how these aspects benefit distributed computing use cases in Section 4.

2 Related Work

A light probe is a representation of the lighting around a specified point in space. This information can then be used to approximate lighting for nearby surfaces. Environment maps are light probes that act as relatively high resolution globes of the surrounding environment.

A particular representation of a light probe is called a basis. One common basis used in computer graphics is the cubemap basis, which is good for storing precomputed high-resolution environment maps, but can be wasteful for shading rough surfaces. Alternatively, some other bases, like an SH basis or an octahedral basis, provide a more compact representation especially for lower-frequency data.

The SH are a set of special functions that form an orthonormal basis on the surface of a sphere and solve Laplace’s equation. Hence, they are a widely used tool in many fields involving partial differential equations, and they also have a long history in real-time computer graphics, where the SH basis has been used to represent light probes since [21]. In particular, they have proved useful in video games, for precalculating indirect lighting (i.e., GI) for large and static scenes that have only few moving components.

Because games have been relying on light probes for a long time, dynamically updating light probes is an easy-to-integrate method for providing real-time GI. Hence, several rendering methods similar to DDISH-GI have been published.

Dynamic Diffuse Global Illumination (DDGI) is a method that updates light probes in real time using ray tracing, achieving impressive performance. However, in contrast to the proposed method, DDGI probes are limited to diffuse illumination, and a ray tracer is used to fill in glossy parts of the lighting as needed. DDGI uses an octahedral basis for its light probes, with additional visibility information used to improve interpolation [15]. The octahedral representation does not suffer from ringing, unlike SH; on the other hand, it is less compact than the SH representation requiring more bandwidth for distributing computations over a network. DDGI has since been extended to support probe-based glossy reflections as well, although it does not take surface roughness into account [16].

Signed Distance Fields Dynamic Diffuse Global Illumination (SDFDDGI) uses a signed distance field approximation of the scene, which can be used to quickly trace rays even without ray tracing hardware [10]. Like DDGI, this method uses an octahedral basis for the probes, as opposed to the SH in DDISH-GI. It has a fully automatic system for probe placement that relies on the SDF data structure. Probe placement is a challenge of its own, and automated probe placement is out of scope in this paper. Unlike in the proposed method, the probes are not used for reflections in SDFDDGI, and they suggest using other methods such as screen-space reflections or ray tracing for that part.

An SH-based approach for dynamic GI is presented in [23]. Probes are placed in the scene with an algorithm that ensures that each light receiver is visible in some probe. The method uses a real-time lightmap of direct illumination in order to update the probes. Since the probes are placed ahead of time, the contribution of the lightmap pixels to the probe can be precomputed. During runtime, the probes are then updated according to the state of the lightmap. The lighting from relevant probes is then interpolated to calculate lighting at the receiver. The interpolation takes advantage of precalculated visibility information to obtain high-quality lighting. Dynamic geometry is somewhat limited due to the amount of geometry-dependent precalculation needed, and is not directly comparable to the fully dynamic DDISH-GI as such. The method in [23] limits dynamic geometry to certain kinds of occluders and requires more precalculation, but seems to suffer from fewer artefacts and does not use ray tracing during runtime compared to the proposed method. Further, it does allow specular highlights from its probes unlike DDGI, but does this by approximating a directional light source from the probe and calculating the reflection from that.

3 Distributing SH Light Probe Updating and Usage

We propose a hybrid method where the indirect lighting is path traced onto an SH-based light probe approximation. This is used by a rasterization pipeline that handles the direct lighting and applies indirect lighting from the separately calculated light probes. The path tracing and rasterization parts are mostly

independent of each other until their respective results are combined at the end of the rasterization pipeline. Communication between them is minimized by only transferring the SH light probe coefficients, which makes the method suitable for distributed rendering and remote path tracing.

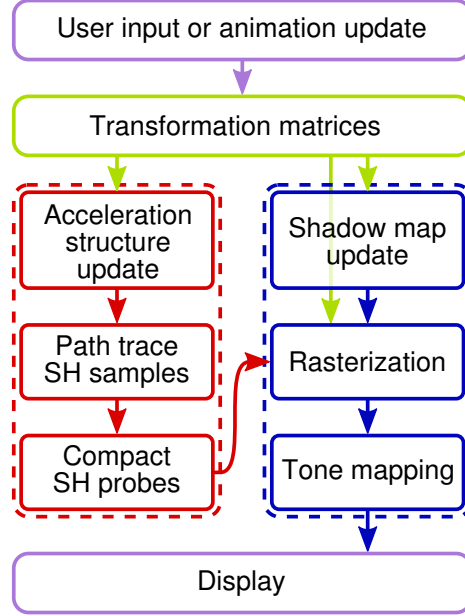


Fig. 2. The pipeline setup of DDISH-GI. The SH update (red) can run asynchronously relative to the SH usage (blue) with few negative effects.

The full rendering pipeline is depicted in Figure 2. User input and animation information, illustrated in green, is used to update all the transformation matrices describing the state of the scene, which is further fed to both the partial path tracing and rasterization pipelines. The path tracing pipeline (SH update), shown in red, updates the SH probes, and the rasterization pipeline (SH usage), shown in blue, renders direct lighting and combines it with the indirect lighting from the updated SH probes, and tone maps the result to be displayed. We explain the SH updating and usage parts in Sections 3.1 and 3.2, respectively.

3.1 Path Tracing and SH Probe Update

In DDISH-GI, path tracing is used solely to update the SH probes that approximate the indirect lighting. This means that the amount of photorealism and accuracy can be flexibly chosen based on hardware capabilities and user preferences. For example, probe count, number of rays and samples used for each probe, and ray bounce depth can be configured. As the updated indirect lighting

component is only needed at the end of the rasterization pipeline before tone mapping, the SH update part of the pipeline can be asynchronous to the SH usage part. Thus, distributing the SH update process to a remote server with dedicated ray tracing hardware over the network is easily achieved, providing computationally restricted mobile devices the capabilities of the server-side devices. Moreover, the indirect lighting does not suffer as much from bandwidth and latency issues due to the compact nature of the SH probe data. Even temporally stale indirect lighting is a decent approximation in the case of a total network failure if the scene, animations, and lighting do not change drastically.

3.2 Rasterization and SH Probe Usage

As in many rendering methods utilizing ray tracing [14,22], DDISH-GI also rasterizes direct lighting. Our rasterization pipeline is also used to combine all components into the final rendered image, because most GPU hardware is still very optimized to the pipeline steps needed in rasterization yielding a better performance than only using path tracing. The path tracer that updates the SH probes is configured separately from the rasterization, consequently making path tracing independent of the screen resolution. Moreover, the rasterization performance is not affected by the path traced sample count or bounce depth.

Producing direct lighting and realistic shadows with rasterization is not unambiguous and can be handled in many ways. In this publication, we do not propose a new method for rasterizing primary visibility and direct lighting. Rather, we refer to [32] which shows performant and power efficient rasterization on mobile devices. We used the method in [21] to implement diffuse indirect lighting.

It is possible to store and sample the coefficients of SH probes as volumes in 3D textures. GPUs traditionally have hardware accelerated operations to sample interpolated values from the textures, meaning trilinear interpolation in the case of a 3D texture. This acceleration can be naturally utilized in the case of the SH coefficients stored in the probes in 3D space, because values for SH coefficients in-between probes are valid when linearly interpolated. We do not use windowing for the SH approximation because while windowing helps to smooth out ringing in certain scenarios, it also produces less precise lighting data.

4 Distributed Computing

One of our main goals when designing DDISH-GI was to lower the amount of data transferred between the SH update and SH usage parts of the rendering pipeline. Our chosen L2 SH light probes use 56 bytes for the raw representation of one probe. Compared to the state-of-the-art DDGI system, which uses a minimum of 1280 bytes per probe, in our largest scene using a grid of 1024 probes, our L2 SH basis only needs $57 \cdot 10^3$ bytes instead of $1.3 \cdot 10^6$ bytes, yielding a $23\times$ reduction in bandwidth. Considering the refresh rate of 90 Hz on typical HMDs, updating indirect illumination for each frame would require a bandwidth of 5 MB/s in the proposed method, which is reasonable on current networking

standards compared to 120 MB/s for DDGI if it were to be used in the same way. The benefit is exemplified on light field displays with dozens of individual viewports, such as [11], which can share the indirect lighting between viewports.

A secondary goal for utilizing distributability and remote hardware capabilities was to split the rendering effort such that the degradation or loss of a network connection to the remote server would not hinder the interactivity of the system. Simultaneously, the computationally most laborious rendering effort, global illumination, could be distributed separately to dedicated hardware. This was achieved with the pipeline structure presented in Section 3 and Figure 2. The pipeline structure together with the view-independence of the SH probes also lends to multi-client use cases: the indirect lighting can be calculated only once to all users which independently rasterize their viewports with relatively lightweight local direct lighting calculations.

The two parts of the pipeline need not be synchronized. The only requirement for an interactive experience is client-side hardware capable of rendering the rasterization part of the pipeline, and the path tracing part of the pipeline can be utilized based on the network capabilities. Examples include situations where we have a large bandwidth and low latency connection, a high latency network, or no network at all. In the first case, the SH probes can easily be updated and transferred to the client side for each rasterized frame for indirect lighting updated in real time. In the second case, the SH probe coefficients can be asynchronously transferred when sufficient lighting or scene changes warrant an update. Precomputed SH probe indirect lighting suits the last case. Interactivity and frame rates are sustained irrespective of connection quality and only the plausibility of indirect lighting is affected by it.

Compared to interactive game streaming services that send controller inputs from the client side and receive fully rendered image sequences from the server side, DDISH-GI is not fully dependent on the server side producing the rendered frames. The streaming services are prone to latency issues manifesting as input-to-rendered-image-latency to the client or full termination of the service on network loss. Some of their latency issues could be mitigated through advances in low-latency compression techniques and their hardware support, for example by using the recent JPEG XS mezzanine compression standard [7] that allows for sub-frame latencies. However, such solutions have not received widespread adoption yet. Furthermore, they do not address the problem of possible network loss to the service. In the proposed method, total network connection loss results only in indirect lighting becoming static instead of dynamically updated.

A possible use case for our rendering method is as follows. A virtual reality headset or HMD with low-end rendering hardware, such as the Oculus Quest 2 [8], capable of running basic rasterization, can be utilized as the SH usage part of our pipeline. Then, a high-end desktop on the server side equipped with a GPU with dedicated ray tracing hardware can be used for the path tracing, or SH update, part of the pipeline. Because XR experiences are sensitive to latency in terms of immersion and user comfort [1], our scenario illustrates how occasional

network latency issues will not affect the overall virtual experience – rather, it only influences the quality and plausibility of indirect lighting.

5 Results

DDISH-GI is compared to the closest equivalent method, DDGI, which is based on a different, less compact but more accurate octahedral probe basis. The original DDGI [15] did not support using the probes for glossy reflections, but the extension was added in [16]. They simply reuse the irradiance-filtered probe data instead of filtering the probes for each roughness value individually. They propose that this approximation be used only for second-order reflections, whereas the first-order reflection should be ray traced. We compare against this extended version of DDGI implemented in the G3D research framework [18].

Three commonly used test scenes are used for the measurements: the Sibenik Cathedral, Breakfast Room and Sponza scenes [17]. These scenes and the camera angles shown were selected such that indirect lighting has a major impact to the image output, which emphasizes the effects the compared methods have on the overall lighting. The irradiance volumes / probe grids are placed manually. The Sponza scene is rendered using $16 \cdot 8 \cdot 8 = 1024$ probes, Breakfast Room with $8 \cdot 4 \cdot 8 = 256$ probes and Sibenik Cathedral with $16 \cdot 8 \cdot 4 = 512$ probes. These probe densities were chosen such that they are not unrealistically dense, yet can represent most lower-frequency local lighting details to a visually acceptable degree. Some of the probes in the volumes are located inside geometry. The probes are placed in the exact same positions in both of the compared methods.

The quality of both methods is evaluated by comparing them against a 16384 spp path traced reference image of the test scene, using the PSNR and SSIM [31] metrics. Each ray was allowed to bounce a maximum of 32 times in the scene before termination. While DDGI technically models infinite bounces, we found that this number of bounces matches its output very closely. We use the implementation in the G3D framework for DDGI measurements, but the material code is modified to match the BSDF we use in these scenes. Furthermore, we use the DDGI glossy reflection approximation in first-order reflections, since both methods only use the probes for indirect lighting in this comparison.

All images are rendered at a resolution of 1920×1080 without anti-aliasing. A single NVIDIA RTX 3090 is used for all timing measurements. The L2 basis is used for DDISH-GI, as we found L3 and L4 significantly slower with negligible quality impact. DDGI is set to use 8×8 resolution for the irradiance data and 16×16 for the visibility data, as suggested in the paper. The proposed method is using 256 path traced samples per probe per frame, which are then temporally blended with data from previous frames. DDGI is given 256 rays per probe per frame, as it functions more similar to ray casting. A hysteresis parameter of $\alpha = 0.99$ is used in both methods. This parameter determines the ratio of reused data from the previous frame to the new data, which means that only 1% of new probe data is blended in with the old in each frame. We experimented with different values ranging from 0.80 to 0.99 (visualized in the supplemental video)

and observed a trade-off between temporal flickering caused by the relatively low spp in probe path tracing and temporal staleness and “ghosting“ due to data re-usage of indirect lighting. Because abrupt changes were much noticeable than slow lagging in indirect lighting, we decided to favor a much higher hysteresis parameter value. Furthermore, direct lighting is still unaffected by the parameter.

Because both compared rendering methods contain temporal components, the quality is measured after the lighting has visually stabilized. Performance measurements were averaged over 50 frames. While we have implemented a visibility-based interpolation scheme into our method, we choose not to use it in this comparison since it provides little quality benefit in these scenes but has a significant cost in the rasterization step. Instead, we use backface culling in probe updates, which prevents dark artefacts from occurring when probes are placed inside geometry. We enabled this same kind of backface culling for DDGI in Sponza and Sibenik Cathedral, because doing so results in better quality in those scenes, and some code for this option existed also in the G3D framework. Comparison images are shown in Appendix B.

Table 1. Quality and performance measurements for the Sponza scene.

Method	PSNR (dB)	SSIM	Probe update (ms)	Rasterization (ms)	Total (ms)
DDISH-GI	23.01	0.911	1.68	0.64	2.32
DDGI	19.05	0.843	5.29	1.14	6.43
16384spp PT	N/A	N/A	N/A	N/A	1866440

Table 2. Quality and performance measurements for the Breakfast Room scene.

Method	PSNR (dB)	SSIM	Probe update (ms)	Rasterization (ms)	Total (ms)
DDISH-GI	24.06	0.917	1.02	0.42	1.43
DDGI	20.41	0.907	3.76	0.90	4.66
16384spp PT	N/A	N/A	N/A	N/A	1114910

Table 3. Quality and performance measurements for the Sibenik Cathedral scene.

Method	PSNR (dB)	SSIM	Probe update (ms)	Rasterization (ms)	Total (ms)
DDISH-GI	23.44	0.873	0.97	0.43	1.40
DDGI	21.82	0.905	3.52	0.96	4.48
16384spp PT	N/A	N/A	N/A	N/A	521877

5.1 Quality

The obtained PSNR and SSIM results are presented in Tables 1–3. Overall, the image quality produced by DDISH-GI is slightly better or on par with DDGI, with quite minimal differences in the Breakfast Room scene and close in terms of SSIM in the Sibenik Cathedral. The poor PSNR results for DDGI in the Sponza scene were assumed to be caused by us forcing the glossy approximation to occur in the first-order reflection, but this was not the case since disabling that resulted in a very similar but slightly lower PSNR score of 18.64 dB.

Because we had to modify the material model of G3D to match our rendering framework, we verified that they match by ensuring that we get the same results

when rendering only direct lighting. Aside from very slight differences in shadow map pixel alignment and biasing (G3D uses a different type of shadow map biasing than our framework), we were able to produce the same image in both renderers and thus concluded that the material models match. Additional modifications had to be made to the glossy approximation, as it originally assumed an extended form of an isotropic variation of the Ashikhmin–Shirley BRDF model [5] instead of the GGX BRDF that we use. In terms of temporal flicker, DDGI was notably stabler with these settings in the Sibenik Cathedral scene and performed admirably in all scenes. Our method had some noticeable flicker in the Sibenik cathedral, but in the other two scenes temporal instability was barely visible through display color banding. This can be worked around by either taking more samples than 256 per probe or using a very aggressive hysteresis value such as $\alpha = 0.998$. Since DDGI reuses previous smooth probe lighting to render additional bounces, they do not have a similar noise source as our path tracing based approach. The downside of that approach is lower precision in lighting, as probe positioning and resolution directly affects secondary bounces as well. This spread of probe inaccuracy may be why the Sibenik Cathedral scene looks significantly more evenly lit in DDGI than the reference.

The main weakness we identify in our method is that the SH L2 basis used naturally blurs out sharp details when fit to the path traced indirect lighting. However, as indirect lighting is usually low-frequency in nature, the loss of high-frequency detail is not as noticeable as in direct lighting. In addition, our current implementation wastes some effort on keeping the SH probes active even if they are inside objects, and thus, not affecting the lighting. We have also experimented with adding a simple, visibility-based probe interpolation to tackle light leaking issues and saw negligible improvements in most scenes.

5.2 Performance

As shown in Tables 1–3, DDISH-GI is significantly faster in both probe update (average 1.22 ms vs. 4.19 ms) and rasterization (average 0.50 ms vs. 1.00 ms) when compared to the DDGI implementation in G3D. The G3D framework uses a different graphics API (OpenGL) than our framework (Vulkan). OpenGL doesn’t support ray tracing directly, so G3D uses the NVIDIA OptiX API [20] for this purpose. This interoperation may include buffer transfers that can slow down the probe update process. Further, we use forward rendering for the rasterization stage, which is less bandwidth-intensive than the deferred rendering used in G3D. The probe basis used in DDGI is also more complicated to filter than SH probes.

We found that the cost of probe interpolation in DDGI is significant and experienced a performance decrease when using interpolation based on visibility. We believe this is caused by the additional bandwidth and computation workload imposed by reading and interpolating probes manually. Some of this cost is avoided by storing probe data in a set of 3D textures and limiting to trilinear interpolation, where we can better utilize texture sampling hardware.

To match the infinite number of light bounces that DDGI simulates, we set the number of light bounces at 32. While realistic, this number is unreasonably

high for plausible indirect lighting in general, and it should be sufficient to use just a few bounces in most scenes. Doing so would have a significant performance benefit in the probe update step of our method. Since our method is based on path tracing, most optimization techniques related to it are usable. We used *Russian roulette* sampling [4] to increase the performance of using so many bounces at the cost of some additional noise in individual samples.

6 Conclusion

We proposed a novel method of utilizing SH probes in a highly distributable rendering pipeline called DDISH-GI. Compared to the G3D implementation [18] of the state-of-the-art method DDGI, the computational performance of our method was $3\times$ as fast on the tested scenes and hardware, and the objective image quality was similar or slightly better. Moreover, our compact SH probe structure could be sent over a network connection to the client side with a $23\times$ reduction in bandwidth compared to DDGI. Unlike approaches based on streaming image frames, it is also robust to network connection issues due to being view-independent and inherently falls back into static indirect lighting with fully local rendering if connection is fully lost.

We showed that our method excels in producing a highly compact representation of realistic indirect light with sufficient quality compared to the state-of-the-art, which can be utilized even in high-latency low-bandwidth server-client connection scenarios. For future work, discarding probes inside geometry and compressing the probe data before transfer would be interesting. It could also be beneficial to try a more robust probe interpolation method in conjunction with non-aligned grid-like probe placement.

Acknowledgements

This project has received funding from the ECSEL Joint Undertaking (JU) under Grant Agreement No 783162 (FitOptiVis). The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Netherlands, Czech Republic, Finland, Spain, Italy. The project is also supported in part by the Academy of Finland under Grant 325530.

Appendix A GGX approximation with ZH lobes

ZH is a subset of SH that consists of the SH functions that are rotationally symmetric with respect to the z -axis. In particular, if we assume that our physically-based material model has rotationally symmetric BRDF lobes, we can represent those lobes more compactly and efficiently with the ZH functions instead of requiring a full SH representation. We refer the reader to [24] and [25] for general details on SH and ZH, respectively.

For the physically-based material model in our renderer, we chose to use the common GGX material model [30], also known as Trowbridge-Reitz distribution [28], to support the glTF 2.0 specification [27] as closely as possible. The GGX BSDF has parameters that allows it to be used for a large variety of different real-world materials with a good degree of realism [30]. It is quite fast to evaluate, which is why it has seen a lot of use in the real-time rendering industry.

SH has been previously used as an approximation for arbitrary BRDFs [13]. Hence, in addition to using an SH approximation for path traced indirect lighting, we also investigate using ZH for GGX-based rotationally invariant BRDF material approximations. The indirect lighting from the SH probes is convolved with the GGX material approximation for the final lighting contribution. As shown in [24], the convolution between a rotationally symmetric function g and some function f – in our case the ZH GGX basis and the SH light probe basis, respectively – has projection coefficients satisfying

$$(g * f)_l^m = \sqrt{\frac{4\pi}{2l+1}} g_l^0 f_l^m, \quad (1)$$

where the left-hand side of the equation denotes the projection coefficients of the basis of g convolved with f , m is the order of the basis function coefficient, and l is the degree of the basis function coefficient. Thus, a convolution of SH over the ZH is simple because it only amounts to scaling each degree of the SH basis with the respective ZH. On the other hand, the SH-over-SH convolution would require several multiplications added up for each coefficient.

Utilizing SH probes and SH approximation for surface materials was proposed in [6] for specular lighting. Their method uses several texture lookups and has to rotate the SH approximation, making it computationally challenging. As the SH probes are already an approximation for the indirect lighting components with smoothed expressiveness at lower degree coefficients, using a sharper convolution lobe for the material, like in [6], is wasteful and only highlights the low-frequency nature of the SH probes. Furthermore, our novel contribution is applying the ZH GGX approximation only for indirect glossy highlights from the SH probes. For the final indirect lighting contribution, we decided to adopt a less complex approach utilizing the *split sum approximation* published in [12]:

$$\int_{\Omega} \frac{L_i(\omega_i) f(\omega_i, \omega_o) \mathbf{n} \cdot \omega_i}{p(\omega_i, \omega_o)} d\omega_i \approx \left(\int_{\Omega} L_i(\omega_i) d\omega_i \right) \left(\int_{\Omega} \frac{f(\omega_i, \omega_o) \mathbf{n} \cdot \omega_i}{p(\omega_i, \omega_o)} d\omega_i \right), \quad (2)$$

where $L_i(\omega_i)$ is the incoming radiance from the direction ω_i , $f(\omega_i, \omega_o)$ is the BRDF from direction ω_i to the direction ω_o , $\mathbf{n} \cdot \omega_i$ is the angle between ω_i and the surface normal \mathbf{n} , and $p(\omega_i, \omega_o)$ is the probability of sampling from ω_i to ω_o based on the BRDF. In our case, the left side integral of the product is calculated as the convolution between the nearest interpolated SH probes and our rotationally symmetric ZH approximation of the GGX specular lobe (similarly as in [12]). This convolution produces a new rotationally symmetric ZH function with coefficients calculated by Equation 1. Furthermore, the right side integral of the split sum is evaluated as an *environment BRDF* which is

encoded in a 2D 2-channel texture varying by surface roughness on one axis and $\mathbf{n} \cdot \boldsymbol{\omega}_i$ on the other.

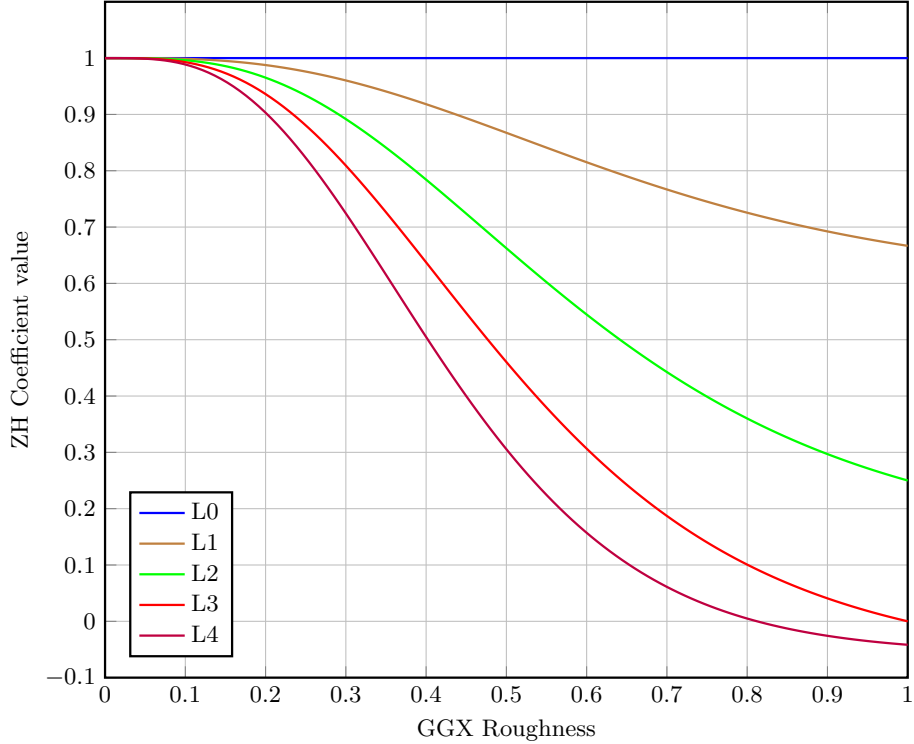


Fig. 3. Values of the precomputed ZH coefficients for approximating GGX specular lobes.

In order to use the ZH basis to approximate GGX materials with different roughnesses, we numerically integrated 1024 samples on the 0 to 1 roughness interval and fit a ZH basis function onto each GGX lobe value. The coefficients of the ZH basis are plotted as a function of the material roughness in Figure 3. The figure shows how the specularity intensifies on smoother materials with large coefficients on the left, and dampens down to diffuse rough materials with small coefficients on the higher degrees. We experimented with different basis degrees from L0 to L4 and found that the L2 basis was a good trade-off between coefficient compactness and approximation quality. The fit is exactly correct to the Trowbridge-Reitz GGX lobe at the impulse direction of the surface normal.

Even though the split sum approximation doesn't take into account the skewness of the GGX lobe at grazing angles, it is an accepted trade-off in the industry and works well enough together with the SH-probes. In [12], the authors used

Equation 2 to compute the separate integrals in advance for environment maps on cubemap bases, whereas our novel contribution is applying this to an SH light probe basis for glossy specular highlights with the GGX lobe further approximated by a ZH basis. The split sum method constrains the materials’ BRDF lobes to be axially symmetric, which means that in our implementation of the SH probe basis, we only consider the rotationally symmetric basis which is exactly the ZH basis. This provides us with the aforementioned faster SH-over-ZH convolution. In each degree of the SH basis, the ZH is unique and so, we can refer to the different degrees from L0 to L4 uniquely.

As discussed earlier, our SH probe approximations are only for the indirect lighting component produced by path tracing and do not consider effects from direct lights. In order to support specular highlights from direct lighting, we approximate direct lights as almost singular points and directly sample the BRDF, which is less accurate for larger lights. However, it serves as a decent approximation as long as the surface is not perfectly smooth and has some roughness present.

Appendix B Comparison images

We present comparison images between DDGI, the proposed method DDISH-GI, and a 16384 spp path traced reference from Sponza (Figure 4), Breakfast Room (Figure 5), and Sibenik Cathedral (Figure 6). We observe that DDGI exhibits overly spread out indirect lighting in occluded areas whereas the proposed method is closer to the reference in those situations, as seen behind the benches of the Sibenik Cathedral. Both probe-based methods have challenges with high frequency detail in indirect lighting effects, such as in the corners of the Breakfast Room, due to the limited spatial resolution of the probes.

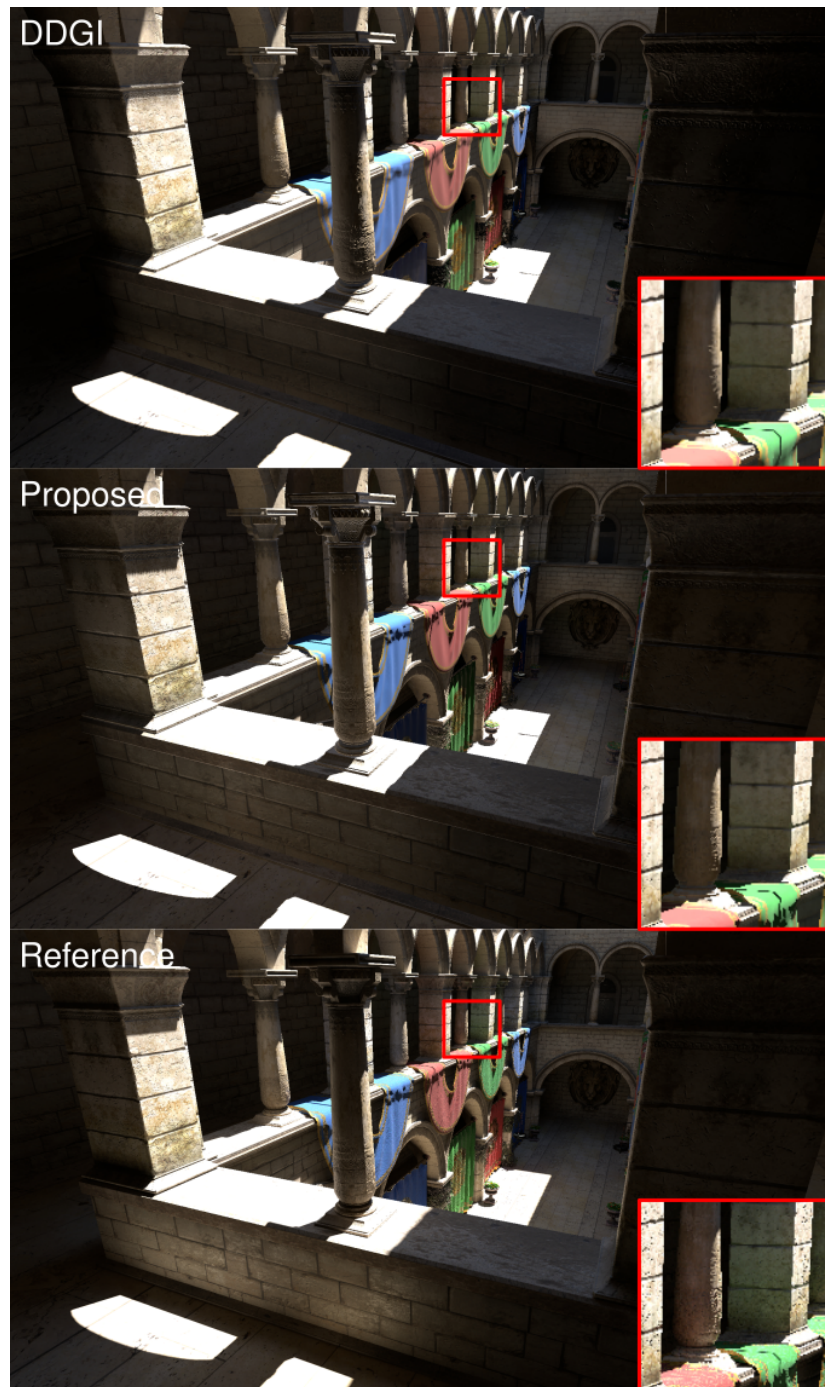


Fig. 4. Comparison images from Sponza.

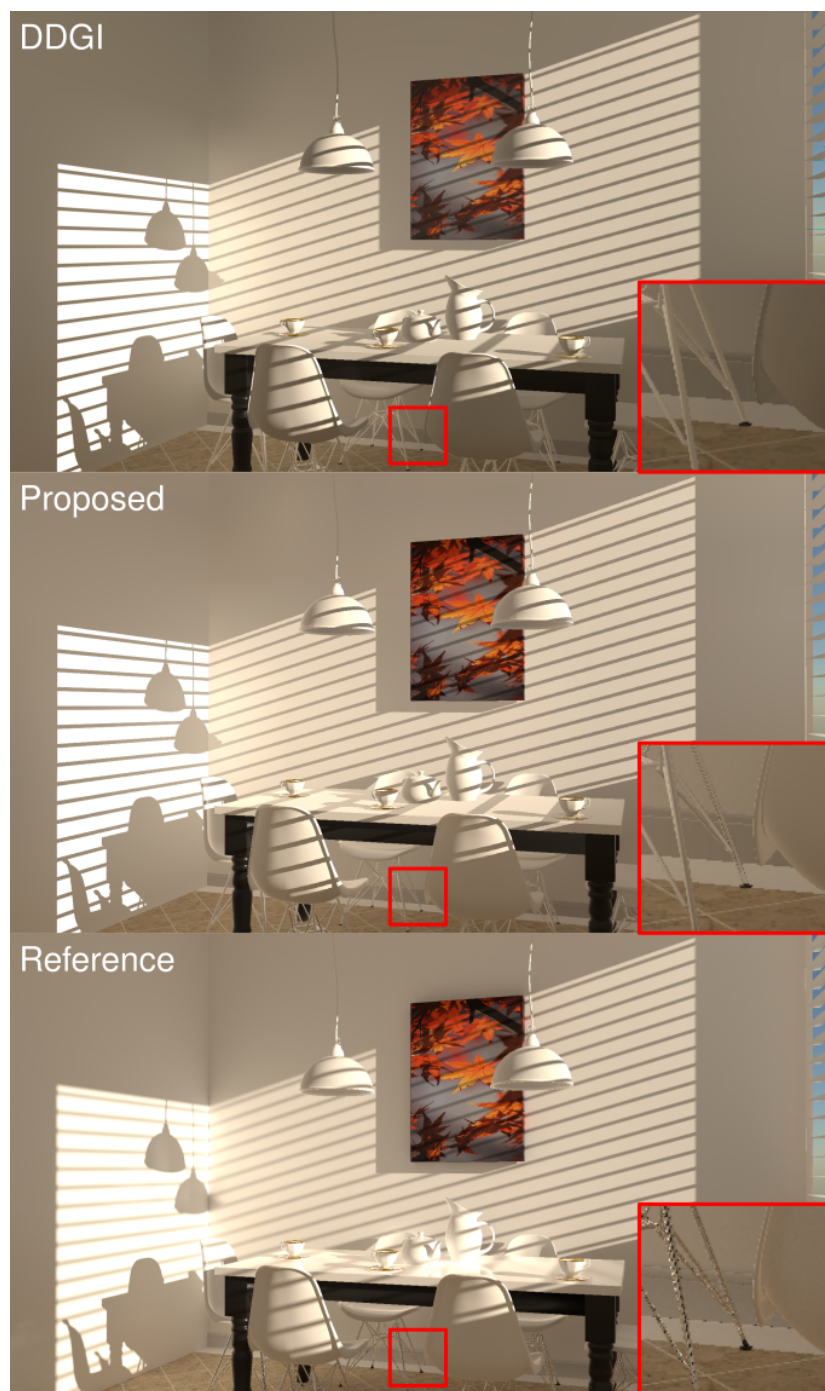


Fig. 5. Comparison images from Breakfast Room.

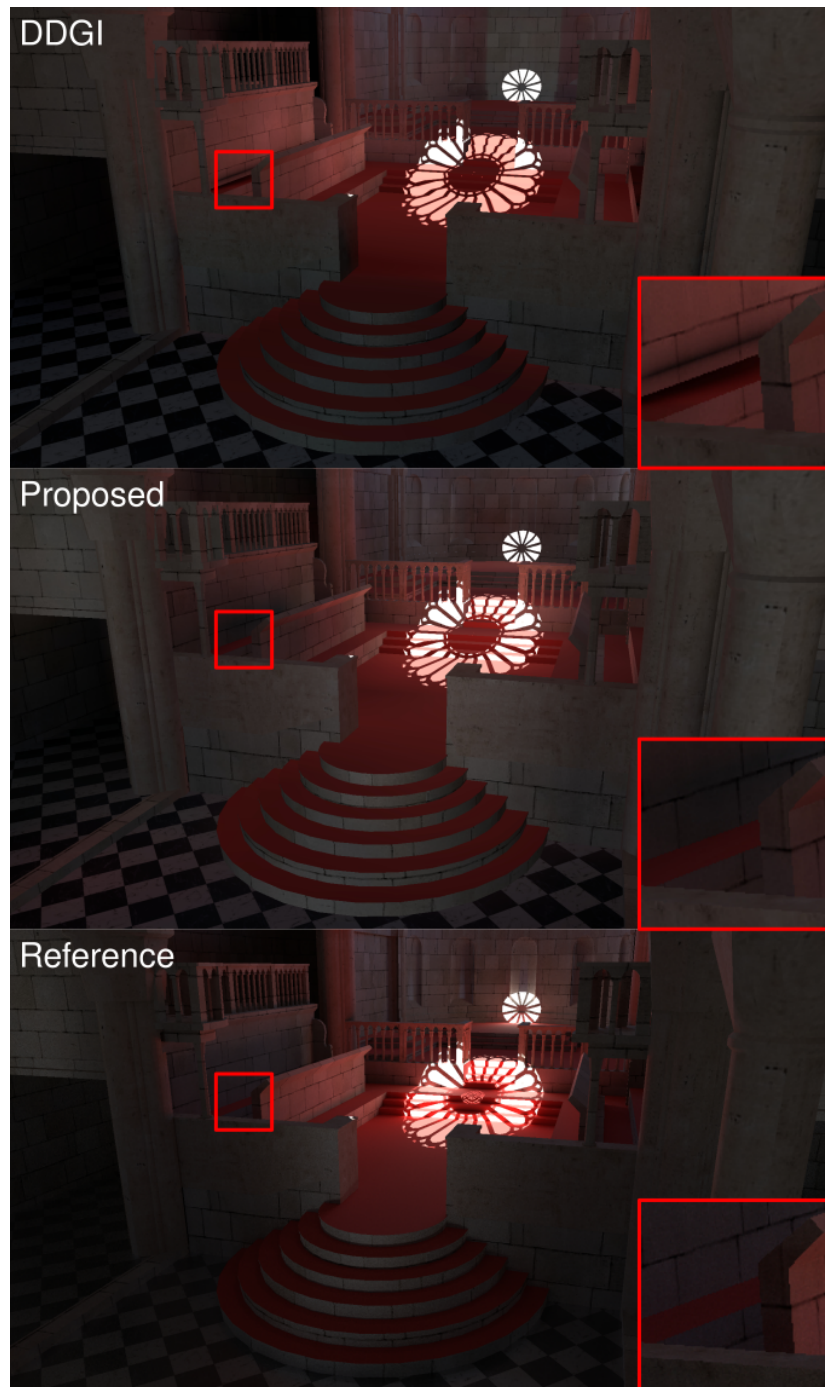


Fig. 6. Comparison images from Sibenik Cathedral.

References

1. Abrash, M.: What VR could, should, and almost certainly will be within two years. Steam Dev Days, Seattle 4 (2014)
2. Advanced Micro Devices, Inc: RDNA 2 architecture. <https://www.amd.com/en/technologies/rdna-2>, accessed: 2020-10-18
3. Angelov, V., Petkov, E., Shipkovenski, G., Kalushkov, T.: Modern virtual reality headsets. In: Proceedings of the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE (2020)
4. Arvo, J., Kirk, D.: Particle transport and image synthesis. In: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '90, ACM (1990)
5. Ashikhmin, M., Shirley, P.: An anisotropic Phong BRDF model. *Journal of Graphics Tools* 5(2) (2000)
6. Chen, H., Liu, X.: Lighting and material of Halo 3. In: ACM SIGGRAPH 2008 Games. New York, NY, USA (2008)
7. Descampe, A., Keinert, J., Richter, T., Föföel, S., Rouvroy, G.: JPEG XS, a new standard for visually lossless low-latency lightweight image compression. In: Proceedings of the Applications of Digital Image Processing XL (2017)
8. Facebook Technologies LLC: Oculus quest 2. <https://www.oculus.com/quest-2/features/>, accessed: 2021-02-22
9. Google LLC: Stadia. <https://stadia.google.com/>, accessed: 2020-10-18
10. Hu, J., Yip, M., Alonso, G.E., Gu, S., Tang, X., Jin, X.: Signed distance fields dynamic diffuse global illumination. *arXiv preprint* (2020)
11. Huang, F.C., Luebke, D., Wetzstein, G.: The Light Field Stereoscope. In: ACM SIGGRAPH 2015 Emerging Technologies. Association for Computing Machinery, New York, NY, USA (2015)
12. Karis, B., Games, E.: Real Shading in Unreal Engine 4. In: Proceedings of Physically Based Shading in Theory and Practice. vol. 4, p. 3 (2013)
13. Kautz, J., Sloan, P.P., Snyder, J.: Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In: Proceedings of the 13th Eurographics Workshop on Rendering. EGRW '02, Eurographics Association (2002)
14. Koskela, M., Immonen, K., Mäkitalo, M., Foi, A., Viitanen, T., Jääskeläinen, P., Kultala, H., Takala, J.: Blockwise Multi-Order Feature Regression for Real-Time Path-Tracing Reconstruction. *ACM Trans. Graph.* 38(5) (2019)
15. Majercik, Z., Guertin, J.P., Nowrouzezahrai, D., McGuire, M.: Dynamic diffuse global illumination with ray-traced irradiance fields. *Journal of Computer Graphics Techniques (JCGT)* 8(2) (June 2019), <http://jcgt.org/published/0008/02/01/>
16. Majercik, Z., Marrs, A., Spjut, J., McGuire, M.: Scaling probe-based real-time dynamic global illumination for production (2020)
17. McGuire, M.: Computer Graphics Archive (July 2017), <https://casual-effects.com/data>
18. McGuire, M., Mara, M., Majercik, Z.: The G3D innovation engine (01 2017), <https://casual-effects.com/g3d>, <https://casual-effects.com/g3d>
19. NVIDIA Corporation: Nvidia RTX ray tracing. <https://developer.nvidia.com/rtx/raytracing>, accessed: 2020-10-18
20. Parker, S.G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., Stich, M.: OptiX: A general purpose ray tracing engine. *ACM Trans. Graph.* 29(4) (2010)

21. Ramamoorthi, R., Hanrahan, P.: An efficient representation for irradiance environment maps. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '01, ACM (2001)
22. Schied, C., Kaplanyan, A., Wyman, C., Patney, A., Chaitanya, C.R.A., Burgess, J., Liu, S., Dachsbacher, C., Lefohn, A., Salvi, M.: Spatiotemporal variance-guided filtering: Real-time reconstruction for path-traced global illumination. In: Proceedings of High Performance Graphics. Association for Computing Machinery (2017)
23. Silvennoinen, A., Lehtinen, J.: Real-time global illumination by precomputed local reconstruction from sparse radiance probes. *ACM Trans. Graph.* 36(6) (2017)
24. Sloan, P.P., Kautz, J., Snyder, J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.* 21(3) (2002)
25. Sloan, P.P., Luna, B., Snyder, J.: Local, deformable precomputed radiance transfer. *ACM Trans. Graph.* 24(3) (2005)
26. Sony Interactive Entertainment LLC: Playstation Now. <https://www.playstation.com/en-us/ps-now/>, accessed: 2021-2-23
27. The Khronos Group Inc.: glTF version 2.0 - specification, <https://github.com/KhronosGroup/glTF/blob/master/specification/2.0/README.md>
28. Trowbridge, T.S., Reitz, K.P.: Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am.* 65(5) (May 1975)
29. Varjo Technologies Oy: Varjo VR-3. <https://varjo.com/products/vr-3/>, accessed: 2021-1-16
30. Walter, B., Marschner, S.R., Li, H., Torrance, K.E.: Microfacet models for refraction through rough surfaces. In: Proceedings of the 18th Eurographics Conference on Rendering Techniques. EGSR'07, Goslar, DEU (2007)
31. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4) (2004)
32. Zhang, Y., Ortin, M., Arellano, V., Wang, R., Gutierrez, D., Bao, H.: On-the-fly power-aware rendering. *Computer Graphics Forum* 37(4) (2018)