

Efficient Environment Map Rendering based on Decomposition

Yu-Ting Wu [†] 

¹ yutingwu@mail.ntpu.edu.tw, National Taipei University, Taiwan

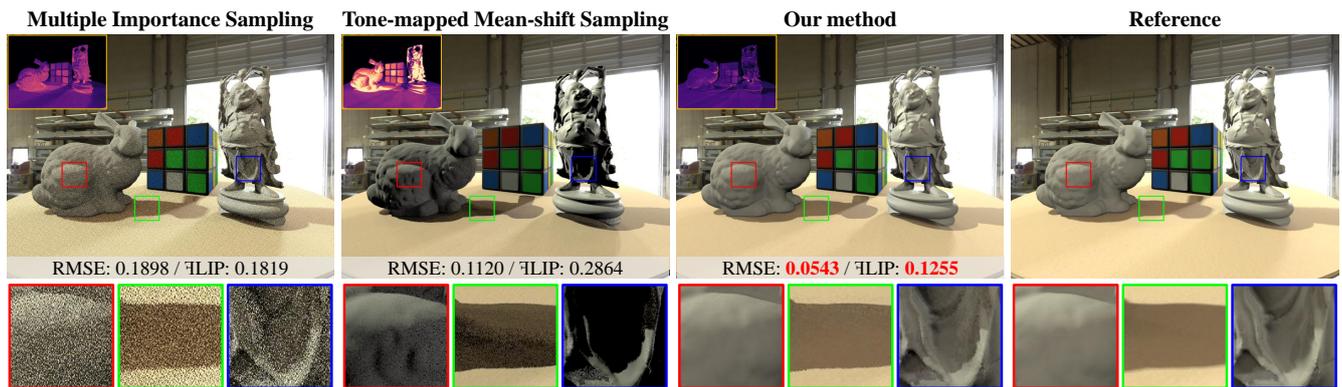


Figure 1: **Equal-time comparisons of various environment map sampling methods.** We compare our methods to multiple importance sampling (MIS) [PJH16] and a global sampling method based on tone-mapped mean-shift (TMMS) [FYWY16]. All methods were rendered using a CPU ray tracer [PJH16] with 4 seconds. TMMS and our method employ interleaved sampling to reduce shadow boundary artifacts caused by evaluating lighting with a small global set of lights for all shading points. Our method generates significantly less noise than MIS and renders more accurate shadows and shading than TMMS. It also achieves the lowest RMSE and FLIP [ANSAM21] errors. The error visualization of FLIP is shown in the top-left corners of the rendered images, with brighter pixels indicating larger errors.

Abstract

This paper presents an efficient environment map sampling algorithm designed to render high-quality, low-noise images with only a few light samples, making it ideal for real-time applications. We observe that bright pixels in the environment map produce high-frequency shading effects, such as sharp shadows and shading, while the rest influence the overall tone of the scene. Building on this insight, our approach differs from existing techniques by categorizing the pixels in an environment map into emissive and non-emissive regions and developing specialized algorithms tailored to the distinct properties of each region. By decomposing the environment lighting, we ensure that light sources are deposited on bright pixels, leading to more accurate shadows and specular highlights. Additionally, this strategy allows us to exploit the smoothness in the low-frequency component by rendering a smaller image with more lights, thereby enhancing shading accuracy. Extensive experiments demonstrate that our method significantly reduces shadow artifacts and image noise compared to previous techniques, while also achieving lower numerical errors across a range of illumination types, particularly under limited sample conditions.

CCS Concepts

• **Computing methodologies** → **Ray tracing; Rasterization; Mixed / augmented reality; Virtual reality;**

1. Introduction

Environment maps are the most prevalent representation for providing real-world illumination to enhance the realism of rendering

virtual objects [Deb98, Deb05a]. These maps capture the direction-varying illumination data at a particular scene location: each pixel’s two-dimensional coordinates correspond to a spherical direction, and the pixel value represents the incoming radiance from that direction. We can simulate realistic lighting on virtual objects by evaluating contributions from all pixels in an environment map.

[†] Corresponding author



Figure 2: **Decomposing lighting in an environment map.** The leftmost image displays the manually labeled classified result of the PINE ATTIC environment map. The second and third images show rendered results using emissive pixels (highlighted in red) and non-emissive pixels (highlighted in blue), respectively. The image rendered with the emissive regions demonstrates high-frequency shading effects, such as pronounced shadows, while the image rendered with the non-emissive regions captures the environment’s ambiance, appearing relatively smooth. The rightmost image is rendered with the complete environment map.

However, performing such a brute-force evaluation is impractical because an environment map usually contains millions of pixels. To accelerate rendering, environment lighting can be estimated by importance sampling the environment map independently at each shading point [VG95, BGH05, TCE05, CJAMJ05, CETC06, CAM08, JCY09, WA09, GKPS12, WC13, KŠV*19] or by using a global set of lights for all shading points [KK03, ARBJ03, ODJ04, Deb05b, WWL05, VD09, FYWY16]. However, these methods usually produce unsatisfactory results with low sample counts, which are common in real-time applications due to the requirement for high frame rates. Fig. 1 illustrates such an example. Methods based on importance sampling, such as multiple importance sampling (MIS) [VG95, PJH16], suffer from severe noise. In contrast, methods using global lights for all shading points, such as tone-mapped mean-shift environment sampling (TMMS) [FYWY16], generate less noise but struggle to render accurate shading and shadows.

This paper introduces a novel approach to generating a global set of representative lights from an environment map. Our approach is designed to produce low-noise images with accurate shading and shadows using only a small number of lights, making it well-suited for real-time applications. The method is founded on the observation that emissive and non-emissive pixels in an environment map create distinct shading effects. Emissive pixels, such as bright light sources or particles, primarily influence high-frequency shading effects, including sharp shadows and specular highlights. Conversely, non-emissive objects contribute to low-frequency shading, affecting the overall tone of the scene. Fig. 2 illustrates an experiment that confirms this observation. We manually categorize pixels in an environment map into emissive and non-emissive groups and use pixels from each group for rendering. The image rendered using the emissive group displays sharp shading and shadows, as depicted in the second image. In contrast, the image rendered using the non-emissive pixels, shown in the third image, captures the ambient color and appears relatively smooth.

Building on the aforementioned observation, our method first automatically categorizes pixels in an environment map into emissive and non-emissive regions. For the emissive regions, we introduce a novel sampling algorithm to generate a set of representative lights, named *EnvDirects*. These lights aim to produce high-frequency shading effects akin to direct illumination. On the other hand, for the non-emissive regions, we present a modified illumina-

tion cut method [CPWAP08] to generate another set of representative lights termed *EnvIndirects*. These lights are utilized to render low-frequency shading effects similar to indirect illumination. Decomposing an environment map into high-frequency and low-frequency components offers two main advantages. First, it facilitates the development of tailored sampling algorithms for each component, ensuring that *EnvDirects* are placed on bright pixels for more precise shadows and specular highlights. Second, it allows our method to explicitly exploit the smoothness of the low-frequency component by rendering it at a lower resolution and then upsampling the results to the original size. This upsampling process significantly cuts rendering costs and enables our method to evaluate more lights within a given time constraint. Fig. 1 demonstrates that our method can significantly improve rendering quality both qualitatively and quantitatively in an equal-time comparison to previous methods.

We evaluated our method using a wide range of environment maps, encompassing indoor and outdoor scenes, day and night settings, and high-frequency and low-frequency lighting conditions. Our method consistently achieves superior results across these diverse test cases compared to previous approaches. Furthermore, we demonstrate that our approach is straightforward to integrate into game engines and provides advantages, including high performance and low noise, compared to existing real-time rendering algorithms like ReSTIR [BWP*20, ZLK*24].

The remainder of the paper is structured as follows. Section 2 examines related work. Section 3 describes the components of our method, detailing the criteria to classify pixels and the algorithms used to generate *EnvDirects* and *EnvIndirects*. Section 4 presents the experiments and discussions. Finally, Section 5 concludes the paper with directions for future work.

2. Related Work

This section concentrates on sampling or rendering algorithms specifically for environment lighting. Methods related to sampling reflectance functions or light paths are considered less relevant and are therefore not included.

The most common approach to estimate environment lighting involves sampling a set of lights and combining their shading results using an estimator. Importance sampling is an effective strat-

egy for reducing sampling variance. This technique generates samples based on the (partial) product of the rendering function components, including the illumination from the environment map, the reflectance function at each shading point, and the visibility function between the light source and the surface point.

Some methods convert an environment map into a global set of representative lights, typically selected based on pixel intensity using either importance sampling or clustering. Light stratification is also important to consider, as it accounts for contributions from different directions. Kollig and Keller [KK03] enhance the sample distribution generated with Lloyd’s relaxation method by iteratively adding new lights into the light cluster with the highest energy. Agarwal *et al.* [ARBJ03] and Ostromoukhov *et al.* [ODJ04] introduced hierarchical stratification algorithms for efficient generation of representative lights. Wan *et al.* [WWL05] subdivide the environment map into equal quadrilaterals proportional to solid angles. The median cut algorithm [Deb05b] recursively divides an environment map into disjoint blocks with equal energy and generates a representative light at the energy centroid of each region. Viriyothai and Debevec [VD09] further refined this method by splitting regions to minimize overall variance. Feng *et al.* [FYWY16] proposed a method to generate light clusters with irregular shapes. They first use the mean-shift algorithm to obtain initial clusters. Then, the clusters are adaptively splitting and merging clusters based on their importance. Our method falls into this category. Unlike previous methods that apply a single sampling algorithm for all pixels, we developed specialized sampling approaches for the emissive and non-emissive regions in an environment map, resulting in more accurate shadow rendering and better exploitation of the smoothness in the low-frequency component.

Using a global set of lights for every shading point has two primary advantages. First, it enhances run-time performance because light samples can be precomputed offline. Second, it can produce noise-free images. However, a significant drawback is that the results usually show visible shadow boundary artifacts when the number of samples is limited. Kollig and Keller [KK03] addressed this issue by employing interleaved sampling, which transforms these shadow artifacts into minor noises.

Representative lights can also be selected uniquely for each pixel by taking into account the product of the illumination, surface reflectance, and visibility function. The product of illumination and reflectance function can be approximated using parametric models, such as spherical harmonics [JCJ09], wavelets [CJAMJ05, CAM08], or spherical Gaussian [TCJW08], or through hierarchical partition based on a sum-area table [CETC06]. Multiple importance sampling (MIS) methods provide another solution by combining samples drawn from various techniques [VG95, BGH05, TCE05, KŠV*19]. Additionally, the illumination of an environment map can be represented by virtual point lights to aid sampling [WA09, GKPS12, WC13]. Incorporating surface reflectance or visibility functions along with illumination improves sampling quality, especially for glossy materials. However, due to the spatial variability of the surface reflectance and visibility functions, a unique sampling function must be created for each shading point. This leads to higher sampling costs compared to using a global set of representative lights, which only requires constructing the

sampling function a single time. Moreover, random sampling introduces noise, necessitating more samples or extensive filtering to achieve a noise-free image.

Based on normalizing flows [MMR*19], Rodriguez-Pardo *et al.* [RPFGLM23] recently proposed a data-driven method that can learn to generate light samples and their sampling probabilities directly from an environment map. The method can shorten the time to draw samples by avoiding tabulating cumulative distribution functions.

In other approaches, Mei *et al.* [MJH06] combine median cut with stochastic sampling to account for spatially-varying surface reflectance functions. Barsi and Szirmay-Kalos [BSK05] divide the environment map into a finite number of directional domains using Lloyd’s relaxation and approximate lighting, reflectance, and visibility functions for each region. Madsen *et al.* [MSV03] approximate an environment map with an ambient term and a set of directional lights generated from an optimization process that considers the diffuse shading of the Phong lighting model.

Rhee *et al.* [RPAC17] proposed a mixed-reality rendering framework for computing lighting from a panoramic video. Their method decomposes environment lighting into diffuse and specular components. They also proposed a light detection algorithm to generate directional lights from the environment map and use them for rendering shadows. Compared to our method, their diffuse and specular components are computed without considering visibility. Thus, the shading and shadows are not physically correct.

One popular approach for real-time rendering is to approximate environment lighting using spherical harmonics (SH) [RH01, RH02]. However, these methods struggle with rendering high-frequency shading like specular highlights. It is also expensive to incorporate visibility into SH lighting [RWS*06]. Precomputed radiance transfer methods address these issues by baking visibility per vertex in an offline process [SKS02] or by using other basis functions such as wavelets [NRH03, NRH04] or spherical radial basis functions (SRBFs) [TS06, CDAS20]. However, these solutions come at the cost of time-consuming precomputation and a restriction to static geometry.

Recently ReSTIR [BWP*20, OLK*21, LKB*22, ZLK*24] has provided a real-time solution for rendering fully dynamic scenes with complex lighting using ray tracing. While this method has many advantages, it still depends heavily on denoising algorithms to minimize noise and necessitates high-end graphics hardware for real-time performance. In comparison, our method, integrated into the game engine (Section 4.4), achieves higher frame rates, demands less powerful graphics hardware, and delivers noise-free results.

3. Method

3.1. Algorithm overview

Fig. 3 illustrates the flowchart of our method. Starting with an environment map as input, we first decompose it into emissive and non-emissive regions. We then generate two sets of representative lights, *EnvDirects* and *EnvIndirects*, to approximate the contributions of these regions. *EnvDirects* are used to render the high-frequency

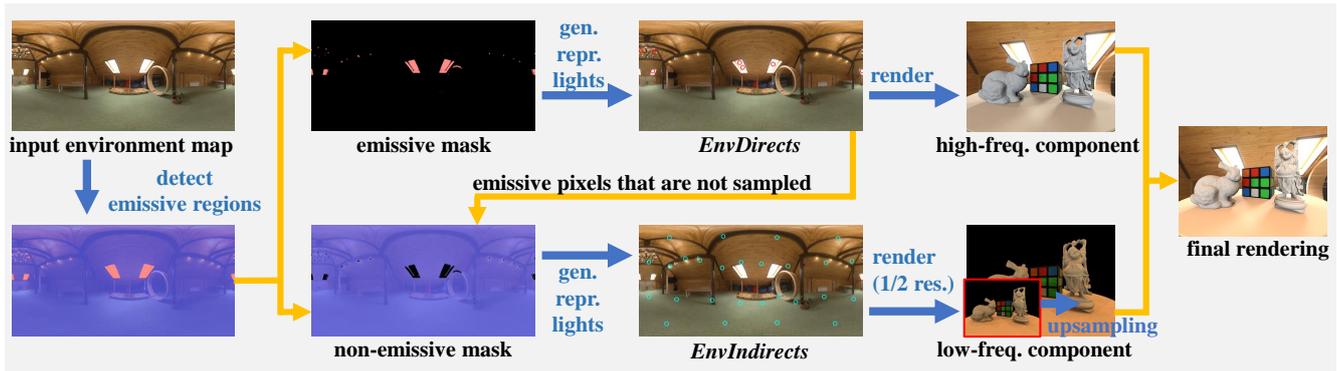


Figure 3: **Flowchart of our method.** Given an environment map as input, we first classify the pixels into emissive and non-emissive regions. Subsequently, we develop distinct sampling algorithms to generate two sets of representative lights, *EnvDirects* and *EnvIndirects*, from these regions. These representative lights are then utilized to render the high-frequency and low-frequency components, respectively. To take advantage of the spatial coherence in the low-frequency component, we render it at a lower resolution and then upsample the result to the original size. This process enables us to evaluate more lights within the given time constraints. Finally, the high-frequency and low-frequency components are combined to produce the final rendered image.

component, while *EnvIndirects* handle the low-frequency component. To exploit the smooth characteristics of the low-frequency component, we render it at a lower resolution and upsample the result to the target size. Finally, we combine the high-frequency and low-frequency components to produce the final result.

3.2. Environment map decomposition

The emissive regions correspond to the bright pixels in an environment map. We determine whether a pixel is in the emissive regions based on the following three criteria:

- The pixel should have a higher brightness than most pixels in the environment map.
- The pixel’s brightness should significantly exceed the average brightness of the environment map.
- The pixel’s luminance should not be significantly smaller than the maximum luminance of the environment map.

The first two rules focus on identifying relatively bright pixels. The first criterion selects emissive pixels based on their relative intensity ranking, while the second criterion assesses them based on their absolute intensity values. The third rule is designed to detect very bright light sources within a small solid angle in the environment map, such as the sun. Using the above criteria, we determine the luminance threshold L_T for classifying a pixel as emissive using the following metric:

$$L_T = \max(L_b, \lambda \cdot L_{avg}, \gamma \cdot L_{max}), \quad (1)$$

where L_b represents the luminance value that exceeds 98% of the pixels in the environment map. L_{avg} and L_{max} denote the average and maximum luminance values of the environment map, respectively. We empirically set λ to 3 and γ to 0.001. A pixel is categorized as belonging to the emissive regions if its luminance exceeds L_T ; otherwise, it is considered part of the non-emissive regions. Fig. 4 demonstrates some examples of our classification results.

The subsequent subsections detail our sampling algorithms for



Figure 4: **Environment map classification results.** Emissive and non-emissive pixels are highlighted in red and blue, respectively.

generating the representative lights from the emissive and non-emissive regions. With a total budget of K lights, we determine the respective number of representative lights in *EnvDirects* and *EnvIndirects* by evaluating the importance of the emissive and non-emissive regions using the metric [ARBJ03]:

$$\Gamma(L, \Delta\omega) = L^a \Delta\omega^b. \quad (2)$$

Here, L and Δ represent a region’s integrated luminance and solid angle. We set the parameters a to 1 and b to 1/4 according to the visibility-based variance analysis in the original paper. Let Γ_{em} and Γ_{nem} be the importance measurements of the emissive and non-emissive regions, respectively. The number of representative lights N in *EnvDirects* is determined by $K \cdot (\Gamma_{em} / (\Gamma_{em} + \Gamma_{nem}))$. The remaining budget $M = K - N$ is allocated to *EnvIndirects*.

3.3. EnvDirects generation

Although several methods have been proposed to approximate an environment map with a global set of lights based on pixel intensity and sample stratification [ARB03, KK03, ODJ04, WWL05, Deb05b, VD09, FYWY16], these methods are designed to sample the complete environment map. In our approach of generating *EnvDirects*, we focus exclusively on emissive pixels, which usually count a small fraction and are distributed across the environment map. Consequently, we developed a new method to generate lights based on intensity while ensuring stratification specifically within the emissive regions.

Our method generates N representative lights iteratively. In each iteration, we begin by drawing a light sample using conventional importance sampling based on the pixels' intensities [PJH16]. Specifically, we construct an importance map of the emissive regions using pixels' luminance values. To sample a light based on the importance map, we first sample a row of the map according to its marginal density, then sample a pixel within the row using conditional probability. The selected pixel becomes the initial position of a representative light. After generating a light sample, we update the importance values of its neighboring region Ω to zero to enforce stratification. In our implementation, Ω includes pixels within an angular distance of $s \cdot 4\pi/K$ from the sample, where K is the total number of representative lights, and we empirically set s to 0.15.

After generating N samples, we further enhance the stratification of lights by applying k-means clustering to refine their directions. Using the N light samples as the initial cluster centers, each k-means iteration assigns the emissive region pixels to the cluster with the nearest center. The cluster centers are then updated to the energy centroids of all cluster members. This process continues until no further changes in the clusters occur. Upon convergence, we position a representative light at the energy centroid of each cluster and set the light color to the sum of the pixel values within the region. Pixels that are too far away from the nearest center ($2 \cdot \Omega$) are excluded, and their labels are reset to non-emissive.

Fig. 5 compares the light distributions of a test environment map (PINE ATTIC) generated by various methods. Previous methods [Deb05b, VD09, FYWY16], which use a single sampling approach for the entire environment map, might generate strong lights that deviate from the bright light sources, resulting in incorrect shadows. In contrast, our method's decomposition of emissive pixels ensures that the *EnvDirects* are positioned on the bright pixels, leading to more accurate shadows.

3.4. EnvIndirects generation

The representative lights in *EnvIndirects* approximate the contributions from non-emissive pixels. To generate this set of lights, we begin by masking the emissive pixels in the environment map and only consider the remaining part. Compared to the emissive regions, the non-emissive regions typically cover a larger area of the environment map but have lower energy. This suggests that representative lights should be evenly distributed across the entire environment map to accommodate contributions from varying directions, and should have similar energies to prevent high-frequency

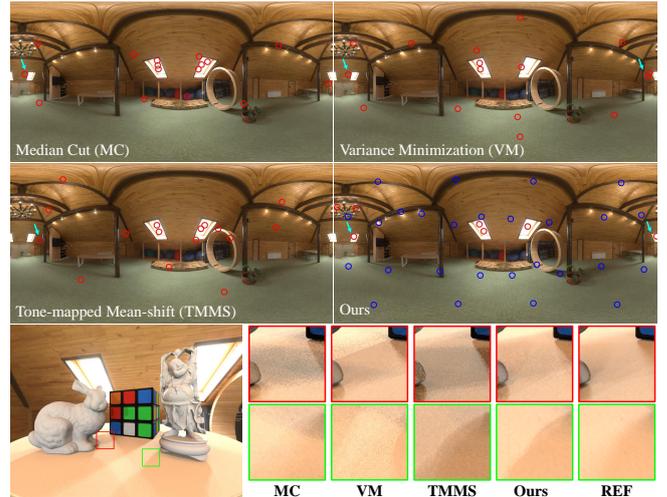


Figure 5: Light distribution comparisons with previous methods, including MC [Deb05b], VM [VD09], and TMMS [FYWY16]. For these methods, red circles denote representative lights. In our method, red circles denote *EnvDirects* while blue ones denote *EnvIndirects*. As indicated by the cyan arrows, our method better ensures that *EnvDirects* (red circles) are on the bright pixels than prior methods. When using these lights for rendering, our method produces the most accurate shadows.

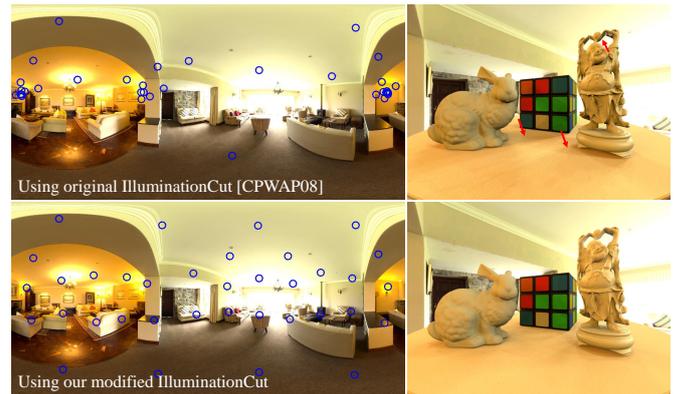


Figure 6: Our modified illumination cut algorithm generates better light distributions of *EnvIndirects*, producing higher-quality images with fewer shadow artifacts.

shading. Furthermore, since the number of *EnvIndirects* is determined by the relative importance of non-emissive region, the algorithm must be able to generate an arbitrary number of lights.

We developed our algorithm for generating *EnvIndirects* based on the illumination cut algorithm [CPWAP08], as it enables the generation of an arbitrary number of lights with minimal variance. The original illumination cut algorithm converts an environment map to many uniformly sampled lights, organized into a binary light tree based on their directions. Each leaf node of the light tree corresponds to an individual light, while internal nodes represent light clusters. To determine a clustering configuration, the method



Figure 7: 32 environment maps used to evaluate the various environment map sampling methods. These test cases encompass a wide range of illumination conditions, including indoor and outdoor scenes, day and night settings, and high-frequency and low-frequency lighting.

evaluates a cut in the tree by iteratively selecting the cut node with the largest internal luminance variance and replacing the node with its two children. Although the illumination cut algorithm performs well in scenarios of many lights, we found that it produces sub-optimal clustering for a small number of lights (8-32). The reason is that the method only considers variance in its splitting criteria, which can result in clusters with large energy or directional extent in homogeneous regions. To address this issue, we incorporate two additional terms to constrain the total energy and the directional extent of a cluster, leading to the following objective function $\text{Obj}(n)$:

$$\text{Obj}(n) = w_V \cdot V(n) + w_D \cdot D(n) + w_E \cdot E(n), \quad (3)$$

where $V(n)$, $D(n)$, and $E(n)$ represent the variance, directional extent, and total energy of the node n , respectively. We normalize these terms to the range $[0, 1]$ by dividing them by the corresponding maximum values of all candidate nodes. The parameters w_V , w_D , and w_E control the weights of the three factors. We set w_V and w_E to 1.0 and w_D to 5.0 empirically. In each iteration, the cut node with the largest $\text{Obj}(n)$ is selected for replacement by its two children. Fig. 6 demonstrates that our modified illumination cut algorithm generates better light distributions than the original version and reduces shadow artifacts. More detailed comparisons are provided in the supplementary materials.

To further exploit the smooth characteristic of the low-frequency component, we render the image shaded by *EnvIndirects* at a reduced resolution and then upsample the result to the target resolution using joint bilateral upsampling [KCLU07]. The process utilizes surface normal, depth map, and albedo map as auxiliary feature buffers to preserve edges. The standard deviations of the Gaussian for the surface normal, depth, and albedo are set to 0.005. The standard deviation for the spatial Gaussian is set to 1.0, resulting in a 5×5 kernel. The final result is obtained by combining the high-frequency component rendered by *EnvDirects* and the upsampled low-frequency component rendered by *EnvIndirects*.

3.5. Interleaved sampling

Using a global set of lights for rendering produces noise-free images; however, the results usually contain noticeable artifacts near shadow boundaries. Kollig and Keller [KK03] demonstrate that using interleaved sampling can reduce the undesired artifacts at the

expense of introducing slight noise. Following this idea, after obtaining a set of *EnvDirects* and *EnvIndirects*, we generate an additional 7 sets of representative lights by randomly perturbing the directions of the individual lights in *EnvDirects* and *EnvIndirects*. To balance between artifacts and noise, we restrict the perturbing range to 0.1 multiplied by the cluster directional extent.

4. Experiments

Compared methods. We compare our new method with several environment map sampling approaches, including multiple importance sampling [VG95] (MIS, using PBRT3's implementation [PJH16]), fast hierarchical importance sampling (FHIS) [ODJ04], median cut (MC) [Deb05b], a variant of median cut based on variance minimization (VM) [VD09], and a method based on tone-mapped mean-shift (TMMS) [FYWY16]. For our method, we also include a version that renders the low-frequency component at the original resolution, denoted as Ours (w/o U). To reduce shadow boundary artifacts for global light methods, we generate 8 sets of representative lights for interleaved sampling for FHIS, MC, VM, TMMS, and our method. Reference images are rendered using MIS with 4096 light samples. We evaluate all methods using root-mean-square error (RMSE) and a perceptual error metric, $\mathcal{H}LIP$ [ANSAM21].

Test environment. We implement all compared methods on the PBRT3 system [PJH16]. All images in this paper were rendered at 1600×1200 resolutions with 4 sub-pixel samples for antialiasing, using a desktop with an Intel i7-10700 CPU at 2.90 GHz, 72GB of RAM, and a NVIDIA RTX A4000 graphics card.

Test environment maps. We assess all compared methods with 32 high-resolution HDR environment maps obtained from the PolyHaven website [PH:23]. Fig. 7 displays the thumbnails of these maps. The test cases encompass a variety of environmental content, such as indoor and outdoor scenes, day and night settings, and high-frequency and low-frequency illumination.

Test scenes. We design four test scenes to evaluate the compared methods. Each scene consists of three to four objects with materials that vary from perfectly diffuse to semi-glossy.

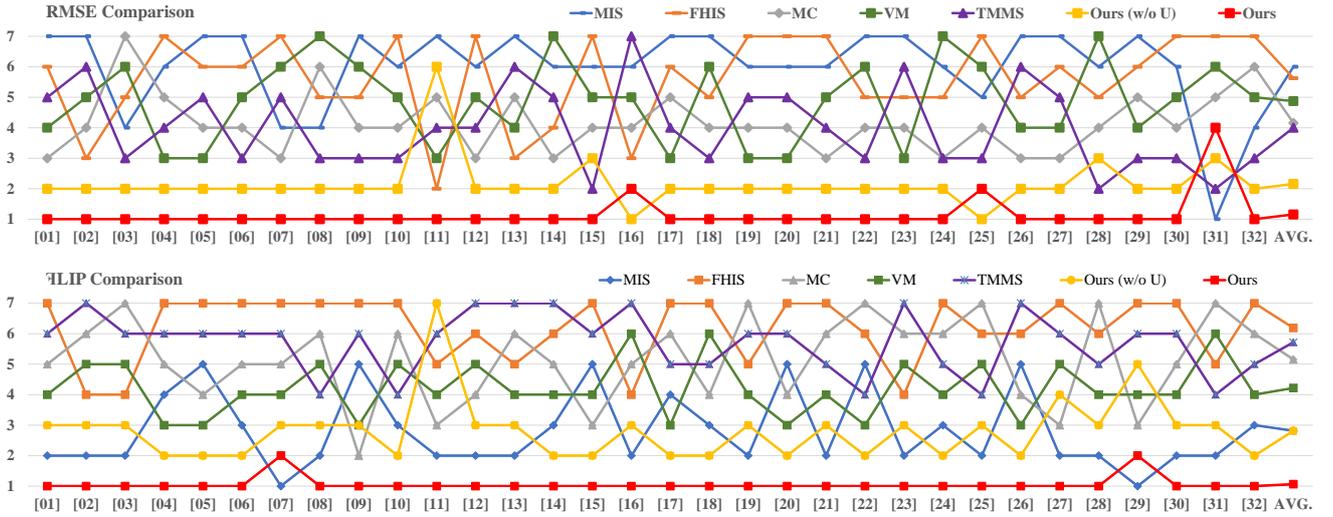


Figure 8: **RMSE and FLIP ranking comparison.** Our full methods (red lines) demonstrate the most robust results among all compared methods. Please refer to the supplementary material for indexing environment maps.

Methods	RMSE ↓ (ranking ↓)	FLIP ↓ (ranking ↓)
MIS	0.1579 (6.00)	0.1212 (2.81)
FHIS	0.1572 (5.63)	0.2217 (6.19)
MC	0.0881 (4.16)	0.1819 (5.16)
VM	0.1035 (4.88)	0.1546 (4.22)
TMMS	0.0886 (4.00)	0.1895 (5.72)
Ours (w/o U)	0.0579 (2.16)	0.1191 (2.81)
Ours	0.0438 (1.16)	0.0865 (1.06)

Table 1: **Quantitative comparisons of various environment map sampling methods.** The RMSE and FLIP are computed by averaging the errors over 32 diverse environment maps and 4 test scenes. The numbers in brackets indicate the average ranking among all compared methods. Our full method achieves the best average ranking and the lowest average errors in RMSE and FLIP.

4.1. Quantitative comparisons

To conduct the quantitative comparisons, we first convert the 32 test environment maps into discrete representative lights using all compared methods. We then use these lights to render the four test scenes and compute the differences from the reference images. Table 1 presents the evaluation results. The RMSE and FLIP values represent the average values across the four test scenes. For each test environment map, we also rank the compared methods and display the average ranking of each method in brackets. Please note that TMMS has a higher average RMSE but a lower ranking than MC because it produces significantly larger errors in some cases. Fig. 8 visualizes the individual RMSE and FLIP ranking of the 32 cases. The evaluation results indicate that our environment lighting decomposition without upsampling the low-frequency component already produces better results than previous methods, and our complete method can further effectively mitigate errors. Moreover, our methods achieve more robust results throughout the 32 cases

than prior methods. MIS produces the second-place FLIP values but the largest RMSE. FHIS, MC, VM, and TMMS show fluctuating rankings between test inputs. Our full method (the red line) consistently outperforms other competitors in almost all cases.

4.2. Qualitative comparisons

Fig. 9 and Fig. 10 compare the rendered results produced by various sampling methods on indoor and outdoor environment maps, respectively. In all cases, MIS exhibits significant noise across the entire image and results in the highest RMSE. Conversely, methods that use a global set of representative lights, including FHIS, MC, VM, TMMS, and our methods, show noticeable artifacts on shadow boundaries. We employ interleaved sampling to balance between bias and image noise. As the figures demonstrate, FHIS, MC, VM, and TMMS generate substantial shadows and shading errors in some image regions. For instance, MC, VM, and TMMS produce incorrect shadows on the Rubik’s cube in LYTHWOOD LOUNGE, and FHIS shows incorrect shading on the vase in LAPA. FHIS and TMMS over-darken the dragon’s face in DRACHENFELS CELLAR, while FHIS, MC, and VM fail to reconstruct shadows accurately in NEUER ZOLLHOF. Our method yields images closest to the reference image in all cases. The FLIP values and error visualizations also indicate that our method produces the most perceptually accurate images. In NEUER ZOLLHOF, our method without upsampling the low-frequency component performs slightly better in terms of RMSE than our full method, as the upsampling process may introduce additional errors. Please refer to the supplementary material for more image comparisons.

One interesting observation is that our decomposition of the environment map reduces the noise generated by interleaved sampling compared to other methods. We found that when strong lights are created from clusters with large spatial extents, the perturbation from interleaved sampling may generate more noise due to visibil-

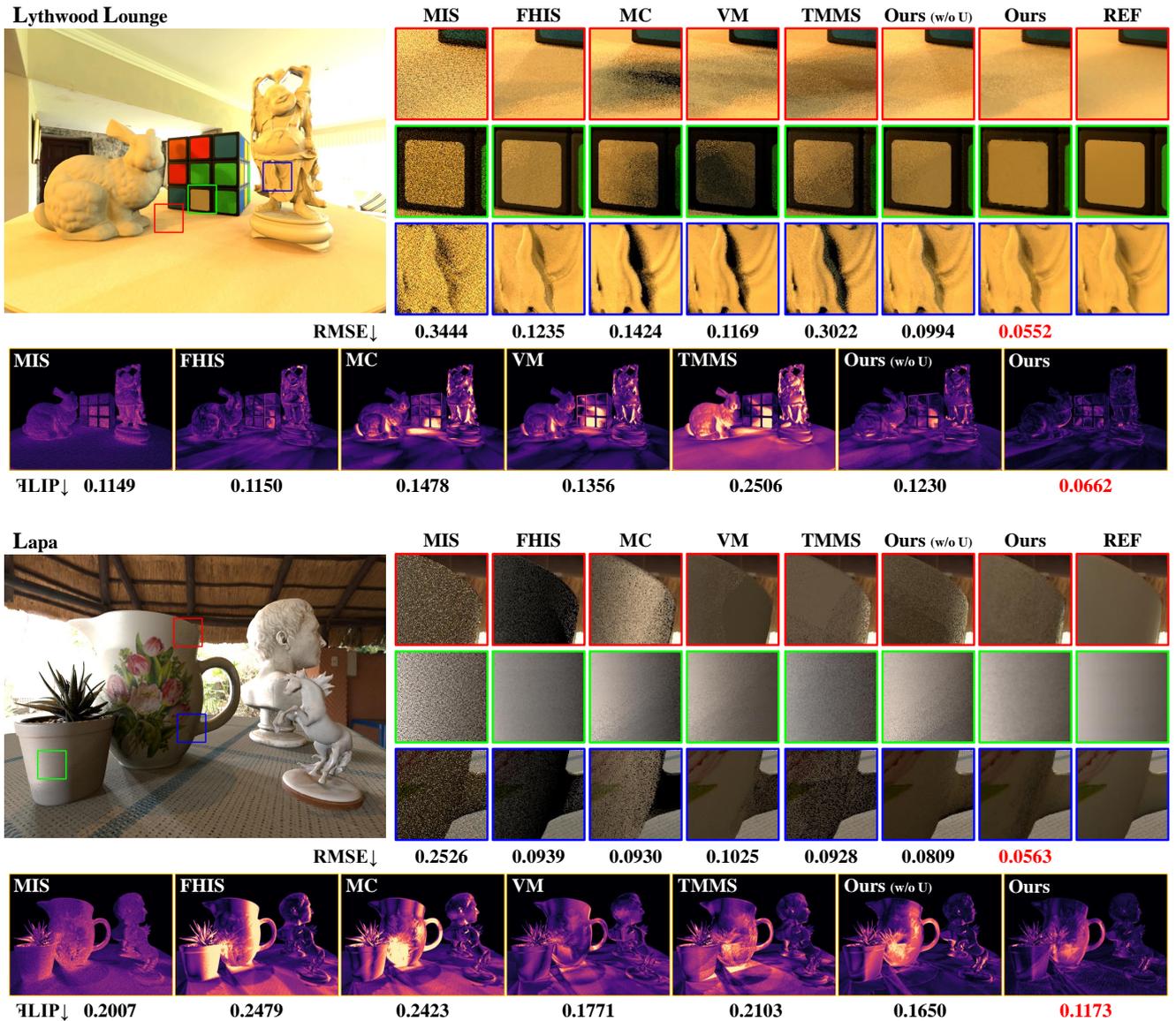


Figure 9: Equal-time comparisons (4 seconds) of various environment map sampling methods on two indoor environment maps, LYTHWOOD LOUNGE and LAPA. The bottom row visualizes the Υ LIP error. Our method showcases superior rendering quality with more accurate shadows and shading in both cases, achieving the lowest RMSE and Υ LIP values.

ity variance. In our method, the strong lights in *EnvDirects* have smaller spatial extents because they are confined to emissive pixels. While the lights in *EnvIndirects* may have larger extents, they exhibit low intensity variance and produce low-frequency shading. Consequently, our method produces significantly less noise and fewer shadow artifacts than FHIS, MC, VM, and TMMS.

4.3. Glossy materials

Highly glossy scenes have long been challenging for methods that render using a global set of lights, as the limited number of representative lights may fail to capture the reflected directions. To assess the performance of various methods in such scenarios, we de-

signed a scene featuring several highly glossy materials. As shown in Fig. 11, our method consistently outperforms other global sampling methods (MC and TMMS) in both visual quality and numerical accuracy when dealing with highly glossy materials. The specular highlights are more precise due to improved light distribution. Compared to MIS, which also samples based on surface reflectance, our method produces significantly less noise. However, the global lighting approach and the absence of surface reflectance considerations might result in less accurate specular highlights.

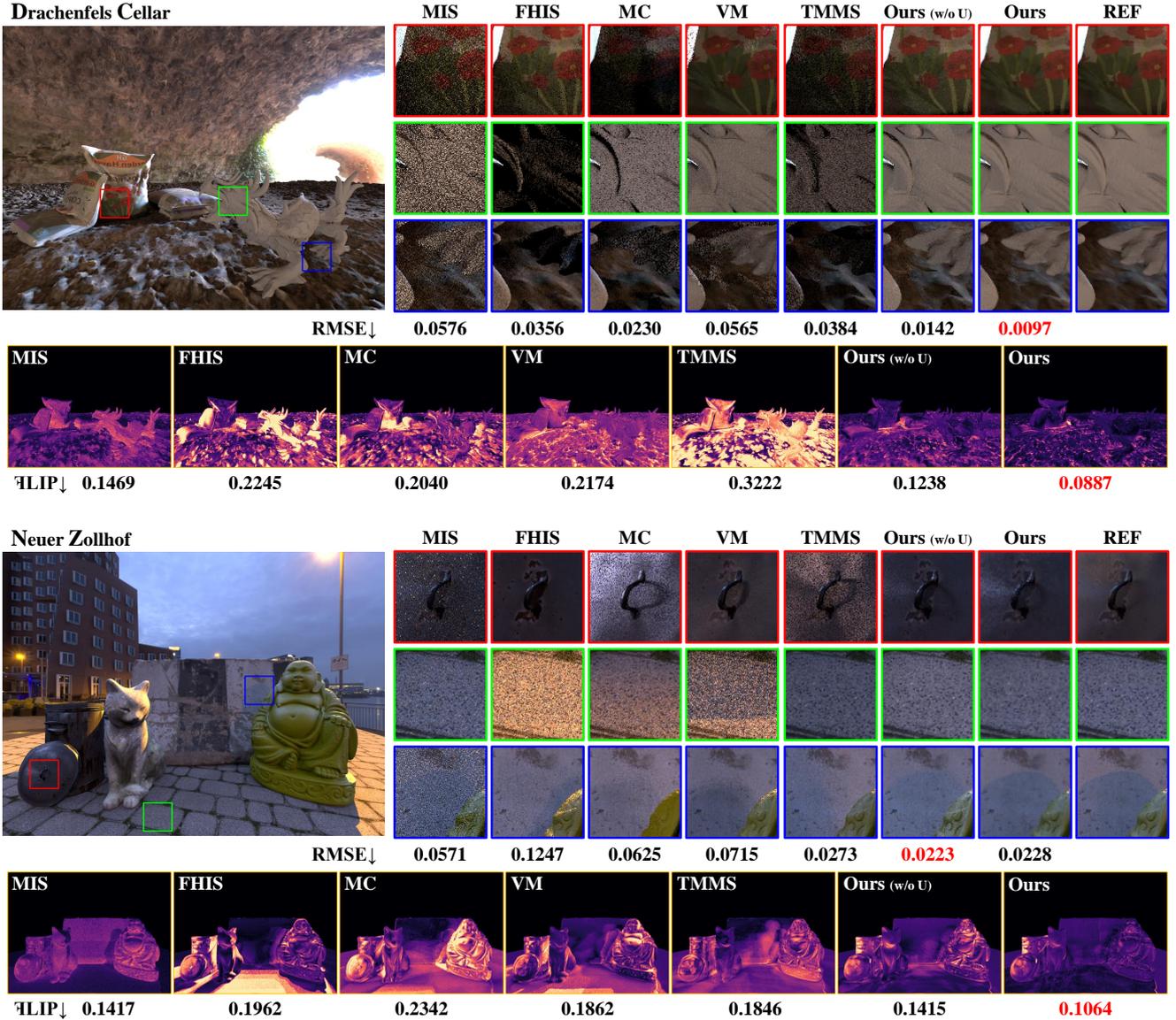


Figure 10: Equal-time comparisons (4 seconds) of various environment map sampling methods on two outdoor environment maps, DRACHENFELS CELLAR and NEUER ZOLLHOF. The bottom row visualizes the FLIP error. Our method produces images with the fewest noises and shadow artifacts, and the most accurate shading compared to prior methods.

4.4. Integration to game engines

Fig. 12 showcases the rendered results of our method implemented in the Unity engine [JBT*20]. The test scene consists of 161.8K triangles. Our method produces visually pleasing images at a rate of 160 frames per second on an NVIDIA RTX A4000 GPU in this scene. Compared to the CPU implementation on PBRT3, we generate the auxiliary feature buffers for joint-bilateral filtering using multiple render targets and determine visibility using shadow maps instead of casting rays, as Unity is a rasterization-based system. We chose not to use interleaved sampling because Unity's built-in soft shadow maps help reduce shadow boundary artifacts by blurring

the edges. This approach also simplifies the implementation while producing noise-free images.

Fig. 13 presents comparisons between the rendered results of our Unity-based implementation with the latest ReSTIR approach [ZLK*24], using the authors' code built on the Falcor framework [KCK*22]. The scene features both diffuse and glossy objects, with a total of 2.1 million triangles. While there are minor shading differences due to variations in shader implementations between the two engines, the strengths of both ReSTIR and our method are evident. ReSTIR excels in producing high-quality specular reflections and soft shadows. However, when rendering a 1600×1200 image with one sample per pixel, it achieves only

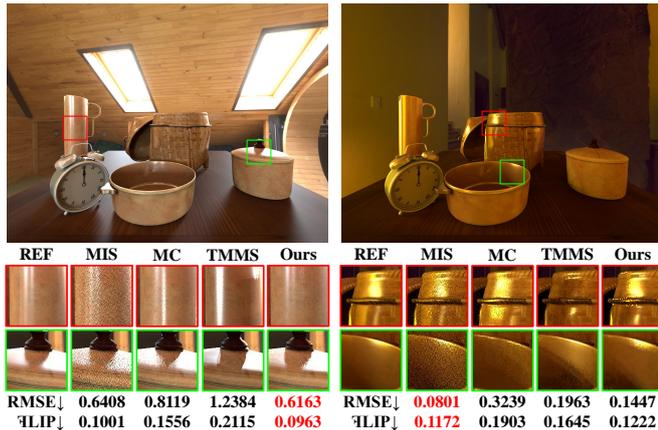


Figure 11: Equal-time comparisons (4 seconds) of various environment map sampling methods on glossy materials, illuminated by the PINE ATTIC (left) and FIREPLACE (right) environment maps, demonstrate that our method produces images closest to the references compared to other global sampling techniques. While our method generates less noise than MIS, it may occasionally misestimate specular highlights due to the reliance on a global set of lights and the absence of surface reflectance considerations.

Method for <i>EnvDirects</i>	RMSE ↓	FLIP ↓
IS-only	0.0599	0.0988
IS+masking	0.0560	0.0958
IS+kmeans	0.0466	0.0877
IS+masking+kmeans	0.0438	0.0865

Table 2: **Ablation studies of various configurations to generate *EnvDirects*.** The RMSE and FLIP values, averaged across 32 environment maps and 4 test scenes, show that the algorithm incorporating all three components: conventional importance sampling (IS), local neighborhood masking (masking), and k-means clustering (kmeans), achieves the lowest numerical errors.

about 18 frame per second in our test environment (NVIDIA RTX A4000), with noticeable noise in the output. Although denoising techniques can mitigate this noise, they introduce additional processing time and memory usage. In contrast, our method delivers noise-free, visually appealing results at 70 – 80 frames per second on the same hardware. The shadows and specular highlights generated by our method match those of ReSTIR, despite the challenges of using a global set of lights and soft shadow maps for visibility approximation.

4.5. Discussions

Partial algorithms for generating *EnvDirects*. Our sampling algorithm for generating *EnvDirects* comprises three major components: importance sampling a light based on luminance (**IS**), masking the local neighborhood of a sampled light to enforce stratification (**masking**), and applying k-means clustering to refine the directions of representative lights (**kmeans**). We conducted an ablation study to investigate various combinations of the three com-

ponents. The experimental results, shown in Table 2, underscore the importance of sample location and stratification in achieving high-quality rendered images when using a global set of representative lights. The **IS-only** and **IS+kmeans** configurations lack sample stratification, while the **IS-only** and **IS+masking** setup do not adjust the light positions to the energy centroid of an emissive region. These configurations produce less optimal results compared to the full algorithm.

Light generation time. On average, our method takes 185.7 milliseconds to generate the representative lights for rendering the results presented in this paper across the 32 testing environment maps. The time taken for each environment map can vary due to differences in the numbers of *EnvDirects* and *EnvIndirects*. The most time-consuming step of our method is using tabulated importance sampling [PJH16] to determine the initial positions of *EnvDirects*, which involves binary search steps to find the sample probability. Employing NEnv [RPFGLM23] could potentially accelerate this step. Although our light generation time is longer than that of MC and VM, which takes about ten milliseconds by speeding up using summed-area tables, it is much shorter than TMMS, which might take seconds due to its expensive mean-shift, adaptive splitting, and adaptive merging processes.

Limitations. Fig. 14 illustrates two challenging scenarios for our methods. In BASEMENT BOXING RING, the narrow, elongated shadows from bright, slender light sources present a challenge due to their soft edges. In BLUE PHOTO STUDIO, numerous small, bright light sources from various angles create smooth shadows. Both scenarios are difficult to accurately reconstruct with a limited number of discrete lights. While all compared methods struggle to render visually appealing shadows, our method delivers the best image quality and lowest errors compared to the others.

Another limitation of the proposed method is that shadow boundary artifacts persist, even with interleaved sampling. One possible short-term solution to mitigate the shadow boundary artifacts is to compute partial visibility between a shading point and the area covered by a representative light. In rasterization-based systems, this can be approximated using soft shadow map techniques. In ray-tracing-based systems, this can be accomplished by tracing multiple shadow rays toward the solid angles covered by the light cluster when generating the representative light, although this would increase computation time.

5. Conclusion

This paper introduces a novel method for approximating an environment map using directional lights. By recognizing that bright pixels in the environment map primarily determine shadow shapes while dim pixels contribute to shading tones, we classify the map into emissive and non-emissive regions and develop specialized sampling algorithms to generate representative lights from these two regions, leveraging their characteristics. Additionally, we enhance shading accuracy by exploiting the smoothness of the low-frequency component through upsampling. The final result is obtained by combining the high-frequency and low-frequency components. Extensive experiments demonstrate that our method consistently outperforms previous approaches in scenarios with limited light sources.



Figure 12: Implemented in Unity, our method achieves an average frame rate of 160 frames per second in this scene (161.8K triangles) on a desktop equipped with an NVIDIA RTX A4000 graphics card. The environment maps used for rendering, from left to right, are PINE ATTIC, LAPA, KIARA 1 DAWN, and RED WALL.



Figure 13: Comparisons between our method implemented in Unity and ReSTIR [ZLK*24]. While ReSTIR generates more accurate specular highlights and shadows, it is prone to sampling noise (visible upon zooming into the images) and operates at just about 18 frames per second in our test environment. In contrast, our method delivers visually appealing, noise-free images at a much faster rate of 70 – 80 frames per second. It’s important to note that our method is integrated into Unity, whereas the ReSTIR approach is implemented using the Falcor framework [KCK*22]. The shading differences between the two are due to variations in shader implementations across the engines. The environment maps used for rendering, from left to right, are PINE ATTIC, INDUSTRIAL PIPE AND VALVE 02, and FACTORY YARD. The scene consists of 2.1M triangles.

Our method assumes a static environment map as input. Moving forward, we would like to explore algorithms for dynamic environment maps to accommodate real-time illumination changes.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and the creators or providers of the models and textures used in this paper, downloaded from McGuire Computer Graphics Archive [McG17] and Poly Haven Website [PH:23]. This work was supported in part by the National Science and Technology Council (NSTC) under grants 111-2222-E-305-001-MY2.

Conflict of Interest Statement

We have no conflicts of interest to disclose.

submitted to COMPUTER GRAPHICS Forum (9/2024).

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

- [ANSAM21] ANDERSSON P., NILSSON J., SHIRLEY P., AKENINE-MÖLLER T.: Visualizing errors in rendered high dynamic range images. In *Eurographics Short Papers* (Vienna, Austria, 2021), The Eurographics Association. doi:10.2312/egs.20211015. 1, 6
- [ARB03] AGARWAL S., RAMAMOORTHY R., BELONGIE S., JENSEN H. W.: Structured importance sampling of environment maps. *ACM Trans. Graph. (Proc. SIGGRAPH)* 22, 3 (jul 2003), 605–612. doi: 10.1145/882262.882314. 2, 3, 4, 5
- [BGH05] BURKE D., GHOSH A., HEIDRICH W.: Bidirectional importance sampling for direct illumination. In *Proc. Eurographics Conference on Rendering Techniques* (Konstanz, Germany, 2005), The Eurographics Association, p. 147–156. doi:/10.2312/EGWR/EGSR05/147-156. 2, 3

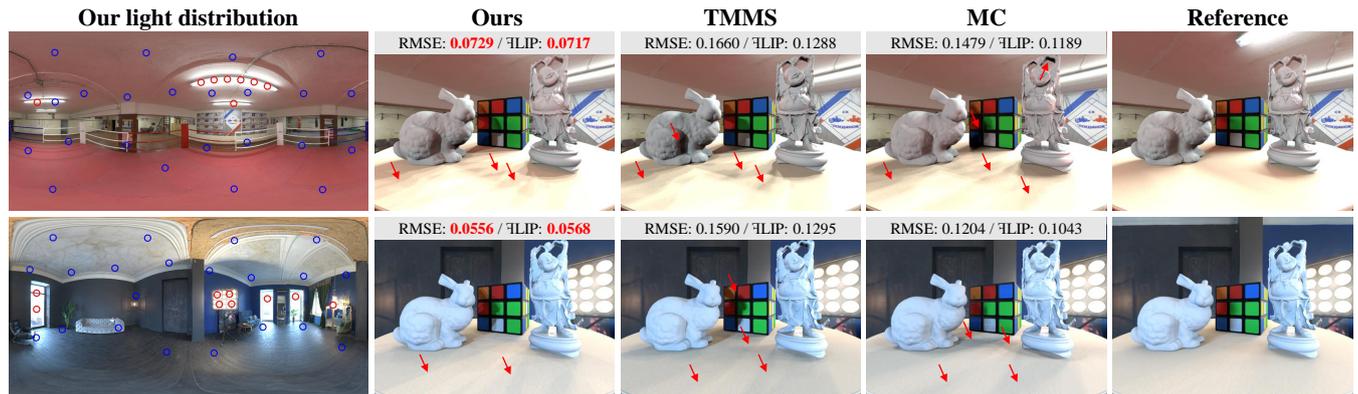


Figure 14: **Challenging cases for our methods.** The slender, bright light sources in BASEMENT BOXING RING (top) and the numerous small light sources in BLUE PHOTO STUDIO (bottom) pose challenges for rendering with a limited number of lights, resulting in inaccurate shadows on the table. Despite these challenges, our method outperforms previous methods in terms of both visual quality and numerical errors.

- [BSK05] BARSÍ A., SZIRMAI-KALOS L.: Image-based illumination on the GPU. *Machine Graphics & Vision International Journal* 14 (01 2005), 3
- [BWP*20] BITTERLI B., WYMAN C., PHARR M., SHIRLEY P., LEFOHN A., JAROSZ W.: Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph. (Proc. SIGGRAPH)* 39, 4 (aug 2020). doi:10.1145/3386569.3392481. 2, 3
- [CAM08] CLARBERG P., AKENINE-MÖLLER T.: Practical product importance sampling for direct illumination. *Computer Graphics Forum (Proc. Eurographics)* 27, 2 (2008), 681–690. doi:https://doi.org/10.1111/j.1467-8659.2008.01166.x. 2, 3
- [CDAS20] CURRIUS R. R., DOLONIUS D., ASSARSSON U., SINTORN E.: Spherical gaussian light-field textures for fast precomputed global illumination. *Computer Graphics Forum (Proc. Eurographics)* 39, 2 (2020), 133–146. doi:https://doi.org/10.1111/cgf.13918. 3
- [CETC06] CLINE D., EGBERT P. K., TALBOT J. F., CARDON D. L.: Two stage importance sampling for direct lighting. In *Proc. Eurographics Conference on Rendering Techniques* (Nicosia, Cyprus, 2006), The Eurographics Association, p. 103–113. doi:/10.2312/EGWR/EGSR06/103-113. 2, 3
- [CJAMJ05] CLARBERG P., JAROSZ W., AKENINE-MÖLLER T., JENSEN H. W.: Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Trans. Graph. (Proc. SIGGRAPH)* 24, 3 (jul 2005), 1166–1175. doi:10.1145/1073204.1073328. 2, 3
- [CPWAP08] CHESLACK-POSTAVA E., WANG R., AKERLUND O., PEL-LACINI F.: Fast, realistic lighting and material design using nonlinear cut approximation. In *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* (2008), Association for Computing Machinery. doi:10.1145/1457515.1409081. 2, 5
- [Deb98] DEBEVEC P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proc. SIGGRAPH* (1998), Association for Computing Machinery, p. 189–198. doi:10.1145/280814.280864. 1
- [Deb05a] DEBEVEC P.: Image-based lighting. In *ACM SIGGRAPH Courses* (2005), Association for Computing Machinery. doi:10.1145/1198555.1198709. 1
- [Deb05b] DEBEVEC P.: A median cut algorithm for light probe sampling. In *SIGGRAPH Poster* (2005), Association for Computing Machinery. doi:10.1145/1186954.1187029. 2, 3, 5, 6
- [FYWY16] FENG W., YANG Y., WAN L., YU C.: Tone-mapped mean-shift based environment map sampling. *IEEE Trans. on Visualization and Computer Graphics* 22, 9 (2016), 2187–2199. doi:10.1109/TVCG.2015.2500236. 1, 2, 3, 5, 6
- [GKPS12] GEORGIEV I., KRIVÁNEK J., POPOV S., SLUSALLEK P.: Importance caching for complex illumination. *Computer Graphics Forum (Proc. Eurographics)* 31, 2pt3 (may 2012), 701–710. doi:10.1111/j.1467-8659.2012.03049.x. 2, 3
- [JBT*20] JULIANI A., BERGES V.-P., TENG E., COHEN A., HARPER J., ELION C., GOY C., GAO Y., HENRY H., MATTAR M., LANGE D.: Unity: A general platform for intelligent agents, 2020. arXiv:1809.02627. 9
- [JCJ09] JAROSZ W., CARR N. A., JENSEN H. W.: Importance sampling spherical harmonics. *Computer Graphics Forum (Proc. Eurographics)* 28, 2 (2009), 577–586. doi:https://doi.org/10.1111/j.1467-8659.2009.01398.x. 2, 3
- [KCK*22] KALLWEIT S., CLARBERG P., KOLB C., DAVIDOVIĆ T., YAO K.-H., FOLEY T., HE Y., WU L., CHEN L., AKENINE-MÖLLER T., WYMAN C., CRASSIN C., BENTY N.: The Falcor rendering framework, 8 2022. URL: https://github.com/NVIDIAGameWorks/Falcor. 9, 11
- [KCLU07] KOPF J., COHEN M. F., LISCHINSKI D., UYTENDAELE M.: Joint bilateral upsampling. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26, 3 (jul 2007), 96–es. doi:10.1145/1276377.1276497. 6
- [KK03] KOLLIG T., KELLER A.: Efficient illumination by high dynamic range images. In *Proc. Eurographics Workshop on Rendering* (Leuven, Belgium, 2003), The Eurographics Association, p. 45–50. doi:10.2312/EGWR/EGWR03/045-051. 2, 3, 5, 6
- [KŠV*19] KARLÍK O., ŠIK M., VÉVODA P., SKŘIVAN T., KRIVÁNEK J.: MIS compensation: Optimizing sampling techniques in multiple importance sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 38, 6 (2019). doi:10.1145/3355089.3356565. 2, 3
- [LKB*22] LIN D., KETTUNEN M., BITTERLI B., PANTALEONI J., YUKSEL C., WYMAN C.: Generalized resampled importance sampling: foundations of ReSTIR. *ACM Trans. Graph. (Proc. SIGGRAPH)* 41, 4 (jul 2022). doi:10.1145/3528223.3530158. 3
- [McG17] MCGUIRE M.: Computer graphics archive, July 2017. URL: https://casual-effects.com/data. 11
- [MJH06] MEI X., JAEGER M., HU B.: An effective stratified sampling

- scheme for environment maps with median cut method. In *Proc. International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06)* (2006), IEEE Computer Society, pp. 384–389. doi:10.1109/CGIV.2006.19. 3
- [MMR*19] MÜLLER T., MCWILLIAMS B., ROUSSELLE F., GROSS M., NOVÁK J.: Neural importance sampling. *ACM Trans. Graph.* 38, 5 (oct 2019). doi:10.1145/3341156. 3
- [MSV03] MADSEN C. B., SORENSEN M. K. D., VITTRUP M.: Estimating positions and radiances of a small number of light sources for real-time image-based lighting. In *Eurographics Short Presentations* (Goslar, Germany, 2003), Eurographics Association. doi:/10.2312/egs.20031065. 3
- [NRH03] NG R., RAMAMOORTHY R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 22, 3 (jul 2003), 376–381. doi:10.1145/882262.882280. 3
- [NRH04] NG R., RAMAMOORTHY R., HANRAHAN P.: Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph. (Proc. SIGGRAPH)* 23, 3 (aug 2004), 477–487. doi:10.1145/1015706.1015749. 3
- [ODJ04] OSTROMOUKHOV V., DONOHUE C., JODOIN P.-M.: Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph. (Proc. SIGGRAPH)* 23, 3 (aug 2004), 488–495. doi:10.1145/1015706.1015750. 2, 3, 5, 6
- [OLK*21] OUYANG Y., LIU S., KETTUNEN M., PHARR M., PANTALEONI J.: ReSTIR GI: Path resampling for real-time path tracing. *Computer Graphics Forum (Proc. High-Performance Graphics)* 40, 8 (2021), 17–29. doi:https://doi.org/10.1111/cgf.14378. 3
- [PH:23] Poly Haven. <https://polyhaven.com/hdris>, 2023. Accessed: 2023-08-15. 6, 11
- [PJH16] PHARR M., JAKOB W., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Nov. 2016. 1, 2, 5, 6, 10
- [RH01] RAMAMOORTHY R., HANRAHAN P.: An efficient representation for irradiance environment maps. In *Proc. SIGGRAPH* (Los Angeles, California, USA, 2001), Association for Computing Machinery, p. 497–500. doi:10.1145/383259.383317. 3
- [RH02] RAMAMOORTHY R., HANRAHAN P.: Frequency space environment map rendering. In *Proc. SIGGRAPH* (San Antonio, Texas, USA, 2002), Association for Computing Machinery, p. 517–526. doi:10.1145/566654.566611. 3
- [RPAC17] RHEE T., PETIKAM L., ALLEN B., CHALMERS A.: MR360: Mixed reality rendering for 360° panoramic videos. *IEEE Trans. on Visualization and Computer Graphics* 23, 4 (2017), 1379–1388. doi:10.1109/TVCG.2017.2657178. 3
- [RPFGLM23] RODRIGUEZ-PARDO C., FABRE J., GARCES E., LOPEZ-MORENO J.: NEnv: Neural environment maps for global illumination. *Computer Graphics Forum (Proc. EGSR)* 42, 4 (2023), e14883. doi:https://doi.org/10.1111/cgf.14883. 3, 10
- [RWS*06] REN Z., WANG R., SNYDER J., ZHOU K., LIU X., SUN B., SLOAN P.-P., BAO H., PENG Q., GUO B.: Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. In *ACM Trans. Graph. (Proc. SIGGRAPH)* (2006), Association for Computing Machinery, p. 977–986. doi:10.1145/1179352.1141982. 3
- [SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph. (Proc. SIGGRAPH)* 21, 3 (jul 2002), 527–536. doi:10.1145/566654.566612. 3
- [TCE05] TALBOT J. F., CLINE D., EGBERT P.: Importance resampling for global illumination. In *Proc. Eurographics Conference on Rendering Techniques* (Konstanz, Germany, 2005), The Eurographics Association, p. 139–146. doi:/10.2312/EGWR/EGSR05/139-146. 2, 3
- [TCJW08] TSAI Y.-T., CHANG C.-C., JIANG Q.-Z., WENG S.-C.: Importance sampling of products from illumination and brdf using spherical radial basis functions. *Visual Computer* 24, 7 (jul 2008), 817–826. doi:10.1007/s00371-008-0263-7. 3
- [TS06] TSAI Y.-T., SHIH Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 25, 3 (jul 2006), 967–976. doi:10.1145/1141911.1141981. 3
- [VD09] VIRIYOTHAI K., DEBEVEC P.: Variance minimization light probe sampling. In *SIGGRAPH Poster* (2009), Association for Computing Machinery. doi:10.1145/1599301.1599393. 2, 3, 5, 6
- [VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proc. SIGGRAPH* (1995), Association for Computing Machinery, p. 419–428. doi:10.1145/218380.218498. 2, 3, 6
- [WA09] WANG R., AKERLUND O.: Bidirectional importance sampling for unstructured direct illumination. *Computer Graphics Forum (Proc. Eurographics)* (2009). doi:10.1111/j.1467-8659.2009.01366.x. 2, 3
- [WC13] WU Y.-T., CHUANG Y.-Y.: VisibilityCluster: Average directional visibility for many-light rendering. *IEEE Trans. on Visualization and Computer Graphics* 19, 9 (2013), 1566–1578. doi:10.1109/TVCG.2013.21. 2, 3
- [WWL05] WAN L., WONG T.-T., LEUNG C.-S.: Spherical Q2-tree for sampling dynamic environment sequences. In *Proc. Eurographics Conference on Rendering Techniques* (2005), The Eurographics Association, p. 21–30. doi:/10.2312/EGWR/EGSR05/021-030. 2, 3, 5
- [ZLK*24] ZHANG S., LIN D., KETTUNEN M., YUKSEL C., WYMAN C.: Area ReSTIR: Resampling for real-time defocus and antialiasing. *ACM Trans. Graph. (Proc. SIGGRAPH)* 43, 4 (jul 2024). doi:10.1145/3658210. 2, 3, 9, 11