EUROGRAPHICS 2025 / A. Bousseau and A. Dai
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 2

Many-Light Rendering Using ReSTIR-Sampled Shadow Maps

Song Zhang[1] Daqi Lin[2] Chris Wyman[2] and Cem Yuksel[1]

[1]University of Utah, USA
[2]NVIDIA, USA

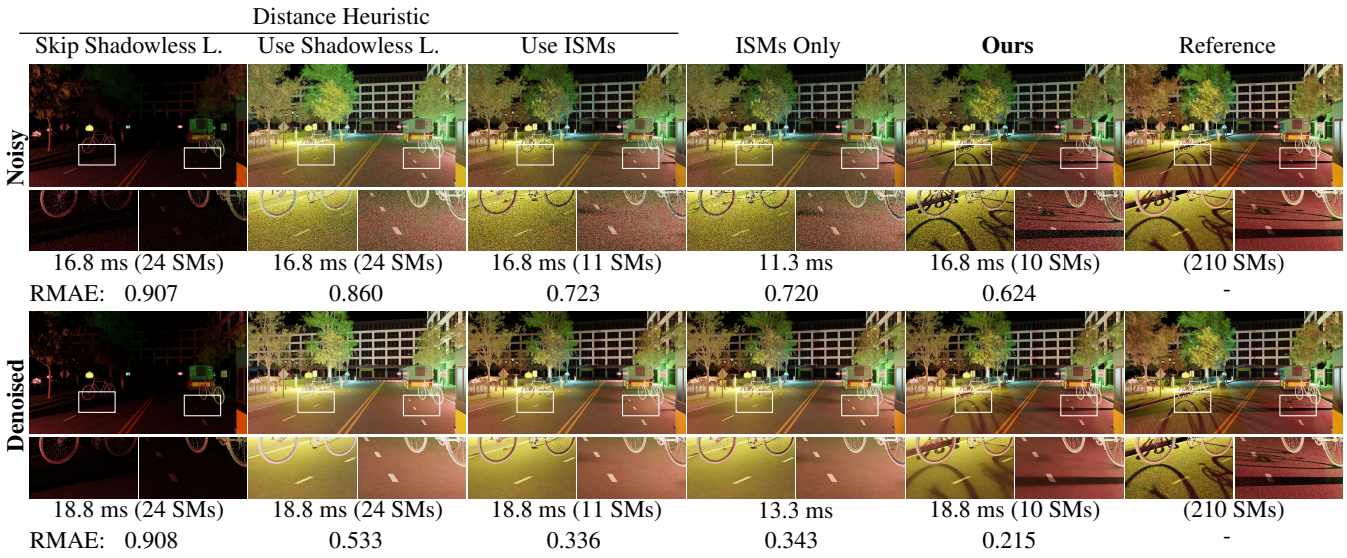


Figure 1: *Our new ReSTIR-sampled stochastic technique automatically selects and generates shadow maps for a few lights in each frame; we handle other lights using imperfect shadow maps (ISMs). This allows more high-quality shadow-casting lights in raster engines without resorting to low-quality ISMs. Notice that our method substantially improves the shadow quality both visually and numerically, as compared to only ISMs and distance heuristics (where we pick closest lights to camera to generate shadow maps and other lights use a fixed visibility value or ISMs). We also show the additional noise introduced by ReSTIR can be filtered by the denoiser [SKW*17]. For reference, we generate shadow maps for all the lights and accumulate frames to remove the noise. This Emerald Square scene uses 35 omnidirectional lights (6 shadow maps for each) to approximate the emissive mesh lights.*

Abstract

We present a practical method targeting dynamic shadow maps for many light sources in real-time rendering. We compute full-resolution shadow maps for a subset of lights, which we select with spatiotemporal reservoir resampling (ReSTIR). Our selection strategy automatically regenerates shadow maps for lights with the strongest contributions to pixels in the current camera view. The remaining lights are handled using imperfect shadow maps, which provide low-resolution shadow approximation. We significantly reduce the computation and storage compared to using all full-resolution shadow maps and substantially improve shadow quality compared to handling all lights with imperfect shadow maps.

CCS Concepts

• ***Computing methodologies*** → ***Rendering;***

1. Introduction

Recent research extends Monte Carlo algorithms to quickly sample from non-analytic probability densities using spatiotemporal reservoir resampling (ReSTIR). This improves importance sampling, including for many-light direct illumination [BWP*20], volumetric lighting [LWY21], and complex specular light paths [LKB*22].

© 2025 The Authors. Computer Graphics Forum published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

However, these algorithms require ray tracing, which is not supported or sufficiently fast across all target platforms of today.

In this paper, we extend the light sampling benefits of ReSTIR to platforms that do not have ray tracing support. Our goal is to allow complex lighting beyond the capabilities of traditional rendering methods without relying on ray tracing.

In a dynamic scene with a large number of light sources, computing shadows becomes the main challenge. Without ray tracing, the obvious alternative is using shadow maps [Wil78]. However, updating each shadow map requires an entire render pass, in addition to the costs of storing and sampling them during rendering. The *imperfect shadow maps* (ISMs) method [RGK*08] offers a fast alternative by computing low-resolution shadow map approximations for a large number of lights using a single render pass, thereby solving both the computation and the storage problems. However, it cannot produce high-quality shadows that are suitable for direct lighting.

Our solution is using ReSTIR to combine shadow maps with imperfect shadow maps. One interesting observation about resampling is that it tends to extract coherence. For direct lighting, ReSTIR-selected samples in neighbor pixels tend to lie on similar lights [WP21]. This indicates that ReSTIR can identify important lights where accurate shadows are essential, enabling us to allocate our limited shadow map budget more effectively to these lights. Thus, our method selects the important lights in the scene for a given camera view using ReSTIR and computes shadow maps only for those lights. The shadows of the remaining lights are approximated using ISMs. This enables real-time rendering of scenes with hundreds of lights while using an order of magnitude fewer shadow maps.

Specific contributions in this paper include:

- The first raster algorithm driven by ReSTIR;
- Defining how to identify important shadow maps using ReSTIR; and
- Avoiding flicker as lights selected for shadow maps evolve.

Unlike ray tracing, where ReSTIR can independently query visibility to any location on any light at every pixel, relying on shadow mapping imposes some hard constraints. Light visibility must be shadow map friendly, e.g., from points, spotlights, or approximated area sources using percentage closer filtering [RSC87] or PCSS [Fer05]. Also, given the limited shadow map budget, we must define the behavior of queries for less important lights and establish a strategy for handling transitions when movement or animation causes a new light to become more significant than the current selections.

2. Related Work

Many approaches can render scenes with complex lighting: basic shadow algorithms (e.g., shadow maps [Wil78]) can be used per-light, instant radiosity [Kel97] can divide complex sources into points, imperfect shadow maps [RGK*08] approximate visibility to point lights, light cuts [WFA*05] aggregate contributions from nearby sources, importance sampling [War94] focuses samples on important lights, tiled shading [OA11] limits the lights touched per tile, structures like BVHs [MPC22] and grids [LY19] reduce the cost of finding important lights, and stochastic data structures [LY20] can reduce costs further.

Mathematically, we have two choices: explicitly sum light from all sources or randomly sample the lighting with a distribution that gives an expected value of this desired integral. Prior work either defines or optimizes algorithms accomplishing one of these goals.

Most real-time applications today either precompute lighting [SSS*20] or sum incident light from a few sources using shadow mapping or ray tracing; these shadow casting lights are usually predefined by artists or other users. There are other approaches like clustered deferred shading and hardware-supported virtual cube map [OSK*14; OBS*15]. But they only use point lights with a finite sphere of influence, and requires a culling pass over shadow-casting geometry.

2.1. Spatiotemporal Reservoir Resampling (ReSTIR)

Recently, spatiotemporal reservoir resampling [BWP*20] is introduced, allowing for quick estimation of per-pixel lighting using very few ray queries. As a stochastic approach that does not require evaluating every light, it significantly increases the number of lights that can be utilized in real time, though it depends on ray tracing to test visibility.

ReSTIR builds on resampled importance sampling [TCE05] to pick samples from a candidate pool of lights in each pixel. Each candidate may be low quality, but the act of reweighting candidates and picking a subset improves the output sample distribution. ReSTIR supercharges this by iterating resampling over samples borrowed from spatial and temporal neighbors.

Resampled importance sampling replaces an importance sampled Monte Carlo estimator with a RIS estimator:

$$\langle F\rangle_{mc} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \qquad \langle F\rangle_{ris} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{\hat{p}(x_i)}\frac{1}{M_i}\sum_{j=1}^{M_i}\frac{\hat{p}(x_{ij})}{p(x_{ij})}$$

For $F = \int f(x)dx$, p being the probability of sampling x_i or x_{ij}, and $\hat{p}$ being the (unnormalized) *target distribution* we really want to sample (which may be non-analytic or otherwise intractable). ReSTIR simply observes that $\langle F\rangle_{ris}$ includes a Monte Carlo estimate of $\int \hat{p}(x)dx$, which can be replaced with another RIS estimator. This iterative replacement allows sample sharing between neighbor pixels.

A key benefit of ReSTIR, for direct lighting, is it allows evaluating the rendering equation's visibility term at lower frequency. For instance, $\hat{p}$ may not always include a visibility query. This allows focusing shadow queries on important lights, rather than testing visibility of all randomly selected candidates x_{ij}. As an example, fast ReSTIR implementations for direct lighting [BWP*20; WP21] apply RIS to select one of M candidate lights using unshadowed $\hat{p}$ contributions at each pixel. A visibility test is performed for this selected (important) light, setting its weight to zero if occluded. After spatiotemporal resampling (reusing samples from other pixels), another visibility test is performed to shade the pixel (computing f). Although only performed for one out of M samples, the initial visibility test significantly reduces variance as it prevents occluded samples from propagating to neighboring pixels [BWP*20].

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

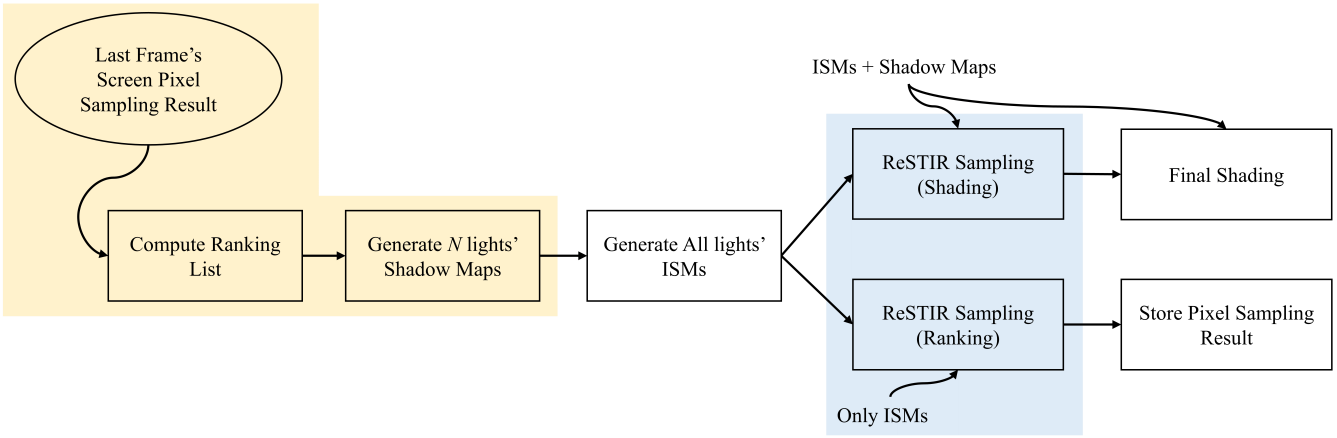


Figure 2: *An overview of our pipeline, highlighting key aspects. The yellow region builds a light ranking using ReSTIR (Section 3.1). The blue region reduces flickering from frame-to-frame differences in light selection (Section 3.3). We combine visibility from both shadow maps and ISMs for higher quality during shading, but use only imperfect shadow maps for light ranking to encourage ranking stability over frames.*

2.2. Imperfect Shadow Maps

While widely used in modern rasterized games, shadow mapping [Wil78] requires computing separate shadow maps for each important light. Rendering costs thus increase linearly when adding new lights. Imperfect shadow maps (ISMs) [RGK*08] instead generate hundreds of lower quality shadow maps in one pass, improving scalability in exchange for reduced shadow quality.

Typically, ISMs are rendered by discretizing a scene into points and randomly distributing these points into low resolution z-buffers during rendering. This leaves holes, which are approximately filled with a push-pull algorithm [MKC07]. The quality of this noisy visibility estimate varies with point sampling density, ISM resolution, and the distribution of points (e.g., quality improves when using an importance-based distribution [REH*11]).

Usually, ISMs are used for visibility of indirect illumination and inaccurate visibility is mitigated by averaging lookups from many virtual point lights. In this work, we demonstrate that when combined with ReSTIR-based light query selection and regular shadow maps for a view-dependent set of important lights, imperfect shadow maps can also deliver reasonable visibility for direct lighting.

3. ReSTIR-Sampled Shadow Maps

As shadow map computation and storage costs grow linearly with light count, using shadow maps for all lights quickly becomes impractical. This strictly limits how many shadow maps can be used when the rendering budget is fixed.

However, each light's contribution to the rendered image varies; depending on the viewpoint, some lights provide significantly more illumination than others. This observation motivates our approach: generate shadow maps for the most important lights and approximate less important ones with imperfect shadow maps (ISMs) [RGK*08]. We identify the important subset of lights by ranking light relevance with ReSTIR [BWP*20], and update this set every frame, automatically adapting to animation of lights, geometry, and camera. We also use ReSTIR for final shading to prevent independent accesses to hundreds of light paths per pixel, which is impractical for real-time applications.

An overview of our rendering pipeline is shown in Figure 2. Below we describe how to use ReSTIR for our ranking (Section 3.1), how to derive a practical direct lighting algorithm based on that (Section 3.2), and then how to achieve stability across frames (Section 3.3).

3.1. Selecting Lights with Shadow Maps Using ReSTIR

Let us first assume perfect accumulation of direct light (no random sampling) and focus on selecting which lights need to generate shadow maps.

Given a budget of N shadow maps, we aim to select a good subset of lights that minimize the visual difference between using shadow maps for all lights and using shadow maps for an N-light subset (and ISMs for the rest). Be aware that simply ranking lights according to the sensor importance of their direct illumination might not truly capture their visual importance. For example, a very bright light might only visibly affect a single pixel because of distance-based attenuation, where the world space distance between shading points in two pixels can vary significantly.

Observing that shadow boundaries become more obvious in regions where one light dominates, our heuristic prioritizes lights that impact a larger number of pixels and exhibit relative strength in their influential regions. To rank using this heuristic, we define the score of a light as the sum of relative contributions at all pixels, i.e. for light ℓ_j, the score $S_j = \sum_{i=1}^{\#\text{pixels}} \left(f_i(\ell_j) / \sum_{k=1}^{\#\text{lights}} f_i(\ell_k) \right)$ where $f_i(\ell_k)$ is the direct lighting contribution of light ℓ_k at pixel i.

Directly computing scores is impractical, as it requires evaluating every light at each pixel. However, by noting that perfect importance sampling at each pixel selects a light with a probability proportional to its contribution f, we can easily see that S_j equals the expected number of pixels that select ℓ_j with perfect importance

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

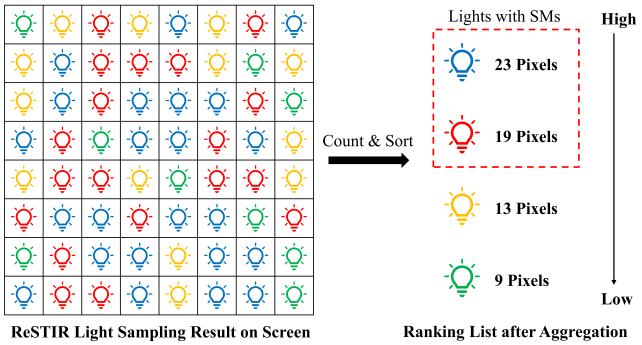


Figure 3: *A standard ReSTIR pass [BWP*20] identifies light samples for each pixel (left). We count each light's samples to rank them (right) and decide which light generates shadow maps in this frame.*

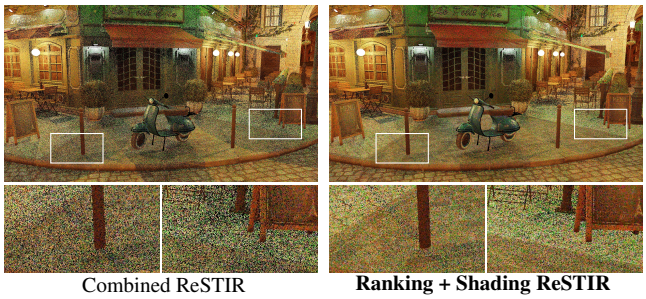


Figure 4: *Compare noise using (left) one versus (right) two ReSTIRs. To avoid flickering, the Combined ReSTIR here uses solely ISMs for sample selection and reserves shadow maps for final shading. With two ReSTIRs, the shading ReSTIR uses both shadow map and ISM queries during sample selection while the ranking ReSTIR only uses ISMs. Noise is measured with relative mean absolute error (RMAE).*

sampling (assuming the samplers are independent)[†]. To approximate perfect importance sampling, we can run a standard ReSTIR process [BWP*20] and obtain a score by counting how many reservoirs select the light (and are visible to the light) (see Figure 3). Under the rasterization context, we use ISMs for the visibility queries in the target function $\hat{p}$ that approximates f, avoiding shadow map generation. This also makes the scores approximate. However, we found that the top-N light set generated this way generally agrees with the top-N lights with accurate scores (see Figure 7). Since this ReSTIR process is only used to produce the ranking, it allows us to skip evaluating f in the final shading. In the ranking process, shadow maps are generated for the top-N lights, while the rest utilize ISMs for the shading procedure described in Section 3.2.

3.2. A Practical Rendering Algorithm with ReSTIR-Selected Shadow Maps

Knowing which lights will use shadow maps, we now want to derive a practical real-time rendering algorithm. Simply accumulating lighting from all light sources can be expensive, so our solution also incorporates ReSTIR for shading. Note that this *shading ReSTIR* is separate from the *ranking ReSTIR* used to select a light subset and requires an additional reservoir per pixel to store. During shading ReSTIR, visibility queries either use a shadow map or an ISM, depending on whether the light was selected to generate a shadow map. Although it is called "shading ReSTIR", visibility queries can happen either during the resampling process or final shading stage (recall Section 2.1).

To minimize the overhead of two ReSTIR processes, we parallelize the computations whenever possible. To prevent delays caused by ReSTIR in identifying the top-N lights, we utilize statistics from the previous frame to choose the top lights, enabling shadow map generation to start at the beginning of each frame, prior to the ReSTIR calculations. See Figure 2 for an illustration.

Specifically, the following methods allow us to maximize shared computations. Initial light candidates are generated once and are used in both shading and ranking ReSTIRs. When checking candidates' visibility, we perform two visibility tests (one for each shading and ranking ReSTIR) only when the selected candidate has a shadow map. During sample reuse, both ReSTIR reservoirs select their own spatiotemporal neighbors. However, the random number for selecting spatial neighbors can be identical, as can the random number for weighted reservoir sampling.

3.3. Maintaining Temporal Stability

One might further reduce overhead with a single ReSTIR to do both ranking and shading. However, it is hard to get comparable quality as the two-ReSTIR version. With a combined approach, shading impacts the next frame's light ranking. This leads to severe flickering as shading and ranking mutually affect each other (see the supplemental video for an example). To avoid flickering, we only use shadow maps for final shading, while all other visibility queries only use ISMs. This prevents final shading from affecting the light ranking. Unfortunately, this Combined ReSTIR somewhat degrades the rendering quality (see left in Figure 4), as visibility during sample selection (only using ISMs) and final shading (SMs + ISMs) can disagree. However, separating these into two passes (see right in Figure 4) eliminates this additional noise, as the shading ReSTIR ensures consistent visibility.

Our two-ReSTIR approach can also flicker due to ReSTIR's sampling noise. Subtle random score variations can change the top-N lights. To address this, we explicitly reason about this flickering: When a light increases its ranking, we upgrade it to use a shadow map. Similarly, if a light's ranking drops, it gets downgraded and loses its shadow map. With a fixed shadow map budget, if one light is upgraded another must be downgraded. Lights around this boundary may alternatively be upgraded and downgraded between frames, leading to objectionable flickering.

To solve this problem, we propose a variance-based light selection strategy. We skip updating the top N list if upgraded lights are

[†] Defining an binary-valued indicator random variable X_i which is 1 if pixel i selects the light j and 0 otherwise, then $\mathbb{E}[\sum_{i=1}^{\#\text{pixels}} X_i] = \sum_{i=1}^{\#\text{pixels}} \mathbb{E}[X_i] = S_j$.

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

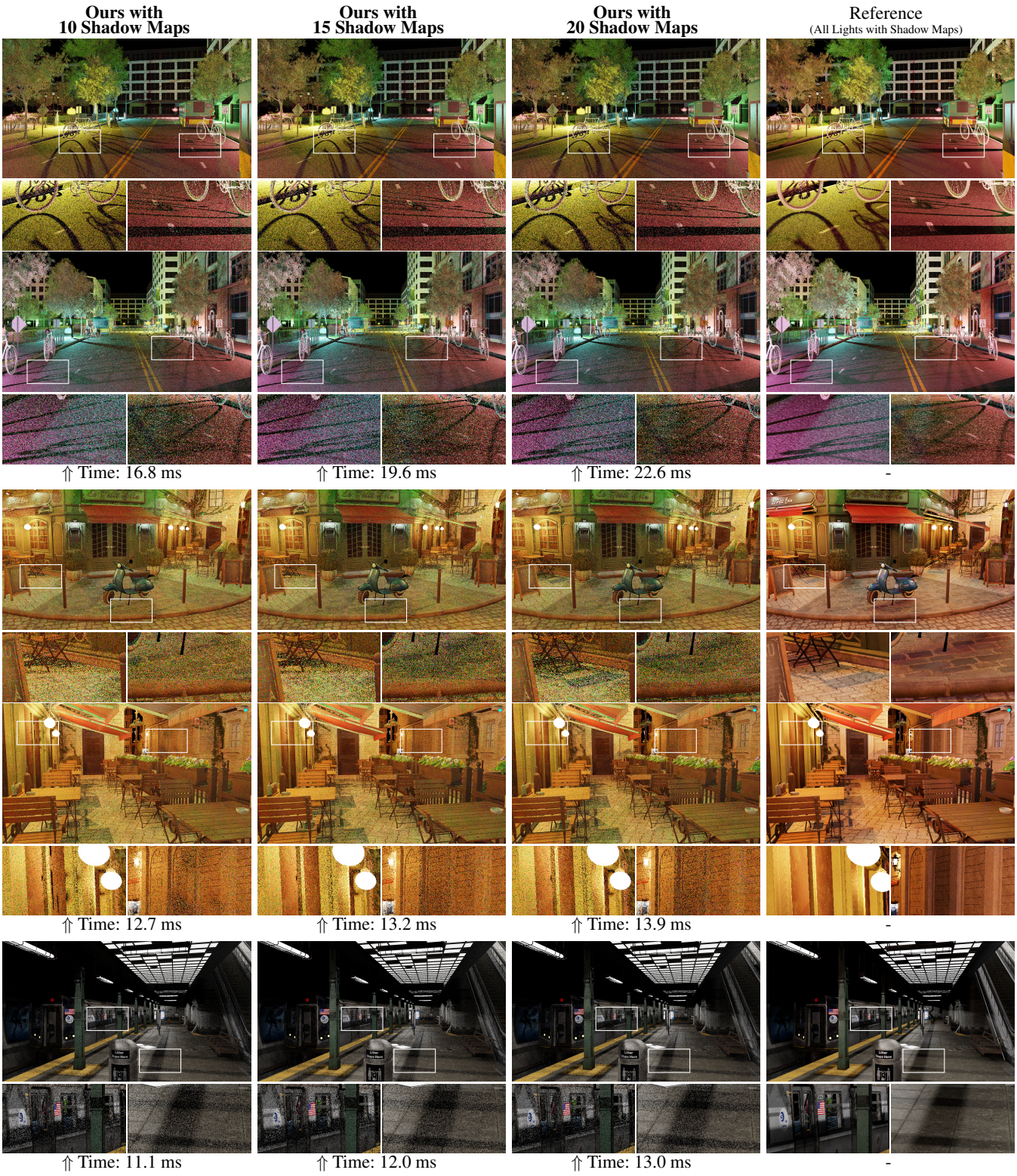


Figure 5: *Comparing different number of shadow maps in our method to reference. The reference is using all shadow maps in the scene (i.e. 210 for Emerald Square, 611 for Bistro, and 535 for Subway, and accumulated enough frames to remove the noise. We pick three representative values from 10 to 20 shadow maps, and select the best one for each view visually. Here, the selected values in Bistro are 20 and 15 (for third and forth row). In Subway and Emerald Square with flying teapots, this value is 10. Note that, what appears as "noise" in the first two rows' reference images is actually aliasing caused by the high-frequency normal map on the road.*

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

likely to get immediately downgraded. We measure this via a running mean μ and variance σ [Wes79] of each light's score s, using

$$\mu = (1-\alpha)\mu_* + \alpha s$$
$$\sigma^2 = (1-\alpha)\sigma_*^2 + \alpha(s-\mu_*)(s-\mu)$$

where μ_* and σ_* are the prior frame mean and variance, score s is recomputed every frame, and α is a blending factor (we use $\alpha = 0.1$) similar to TAA [YLS20]. α controls the relative impact of old and new samples when computing variance σ. Larger α increases recent frames' influence.

When changed rankings propose downgrading one light and upgrading another, we compare their scores. Specifically, the lights entering (i.e. current frame) and leaving (i.e. previous frame) the same position in the top-N ranking form a pair. If their difference is below a multiple β of either lights' variance, we reject the change. Flicker reduction is greater with larger β, but larger values can delay upgrading important lights during animation. We found $\beta = 3.0$ the best trade-off in our tests. Essentially, for a light to be upgraded, its score improvement must exceed what could be attributed to random chance. This significantly reduces flickering from sampling noise.

4. Results

We built a prototype using NVIDIA's Falcor framework [KCK*22]. All our test scenes include hundreds of hand-placed (point or spot) light sources. All methods use the same scene geometry, lighting, framework, and pipeline. Our results are measured on an NVIDIA RTX 4090 24GB, AMD Ryzen 9 7950X (16 cores with 4.5GHz), and 64GB RAM.

Please note that we do not take advantage of static geometry and lights. All shadow maps and ISMs are rendered anew each frame to show performance and quality assuming full dynamism. Thus, our implementation for Bistro and Emerald Square scenes (without animation) is identical to that for the Subway scene with moving trains. To approximate the emissive mesh lights in these scenes, we place either analytic point or spot lights inside them, based on whether they are spherical or rectangular. For Emerald Square scene, we manually place 35 emissive flying teapots (i.e. 35 flying point lights) with animation to simulate complex dynamic lighting in games.

All methods use ReSTIR and only shade one light from each pixel's reservoir. We follow the default settings of biased RTXDI [NVI21], using $M = 32$ candidate samples, selecting one random spatial neighboring pixel within a 30-pixel radius, and limiting the effective temporal sample weight to be at most 20 times that of each frame's new candidate. The spatial rejection heuristic in [BWP*20] is used for both shading and ranking ReSTIR.

We rasterize a V-buffer [BH13] for primary hits in all methods. To remove shadow aliasing, we use hardware percentage closer filtering (PCF) [RSC87] for both ISMs and shadow maps. We use 1024×1024 shadow maps and 32×32 ISMs with one pull-push step.

Picking reasonable number of shadow maps. As the required number of shadow maps is scene dependent, in Figure 5, we explore how to pick a reasonable value. We also demonstrate varying numbers of shadow maps in different scenes. To pick a right number, we focus on *large shadow difference* in these comparisons. Overall, Bistro needs more shadow maps (20 for third row and 15 for forth row) as many lights illuminate the scene simultaneously. This is noticable in the shadows of the chair and Vespa (third row insets); the artifacts caused by ISMs (forth row insets). For other scenes (Subway and Emerald Square), 10 shadow maps works reasonably.

Shadow approximation with ISMs is important. Although ISMs' approximate shadows are crude for direct lighting, Figure 6 shows we cannot simply turn on/off the lights outside of top-N rank. While not as important as the top-N lights, at least some approximation is needed for their shadows. If we approximate other lights to always have visibility of either 0 or 1, results appear excessively dark or overly bright. Utilizing ISMs gives a more satisfactory approximation that closely resembles the reference.

Using only ISMs in ranking ReSTIR is enough. In Figure 7, we show that using ISMs for visibility approximation when ranking lights is also crucial as it properly estimates the score defined in Section 3.1. Although ISMs poorly approximate shadows, only using ISMs for ranking (in Section 3.2) does not dramatically impact final shading. The ranking list need not rely on exact visibility, e.g., expensive full resolution shadow maps. For lights with large contributions, stochastic sampling regularly samples them. Even if ISMs are inaccurate, enough pixels will sample them to keep these lights in the top N after visibility testing.

Adaptive ranking updates. As explained in Section 3.1, our method selects which lights will render shadow maps using ReSTIR. The example in Figure 8 shows how light intensity impacts this process. As light intensity rises, our scores for the light are updated automatically, possibly resulting in a shadow map assignment. ReSTIR factors light intensity into sample weights, leading to higher prominence in the rankings. Greater intensity boosts the weight of the light, prompting more pixels to choose it, thus elevating it into the top-N ranking. This example also shows the importance of using shadow maps for strong lights, as ISM approximations are error-prone and lack detail.

Figure 9 shows our method adaptively upgrading lights as the view changes. When moving the camera toward a light's shadow, ReSTIR elevates the likelihood of selecting that light across more pixels, thereby enhancing its ranking score. This process eventually results in the light being allocated a shadow map.

Improved temporal coherence. The supplementary video shows our method under animation. With our approach, ISMs automatically switch to shadow maps as the camera moves. We fade in shadow maps during these upgrades, giving the transition from low- to high-quality shadows a natural look (e.g., see the shadows of the tables and chairs in Bistro). This fade-in effect is achieved by interpolating the visibility between shadow map and ISM (or fully lit), depending on whether it's our method or distance heuristic, across a 60 frames time window. Even for moving geometry or lights (e.g. the Subway train or Emerald Square with flying teapots), we can

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

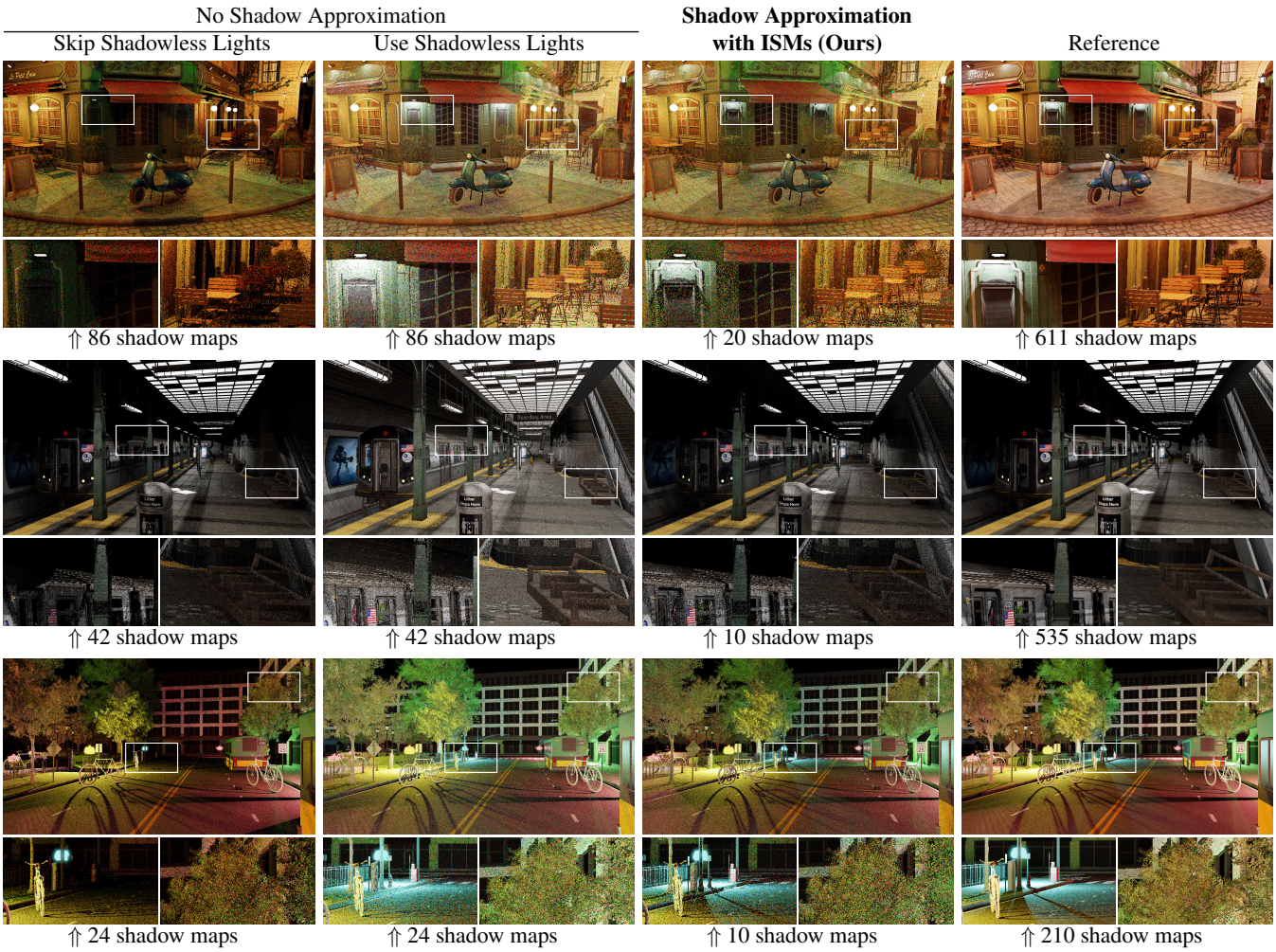


Figure 6: *Equal-time comparison of different visibility evaluations (both shading and ranking) for non-shadow-map lights. With ISMs to approximate shadows, our method achieves much closer results to the reference, especially in terms of overall shadowing quality. From top to bottom, rendering time in Bistro is 14 ms, Subway is 11 ms, and Emerald Square is 17 ms. Note that for both no shadow approximation methods, their ranking visibility is set to 1, which suggests their ranking is the same. For pixels selecting the top-N lights, it uses shadow maps for testing visibility. For pixels selecting other lights, the shading visibility is set to 0 (i.e. first column, always occluded) or 1 (i.e. second column, always illuminated), or evaluated by ISMs (Ours). In reference image, all lights use shadow maps.*

still identify important shadow maps, though some light leaks remain where our shadow map budget is insufficient to upgrade all of the important lights.

Our supplementary video also contains an ablation study showing that variance-based light selection is necessary for our temporal coherence. Further, we show that the flickering in the naive combined ReSTIR cannot simply be removed by variance-based light selection. Incorporating two ReSTIRs along with variance-based light selection is essential.

General quality and performance evaluation. Figure 1 and Figure 10 compare our new method to a renderer with only ISMs or distance heuristic which picks the closest N lights to the camera, or only shadow maps. All comparisons use ReSTIR for direct lighting, just with different visibility terms. For both ISMs and our method, 100 million pre-generated point samples are rendered onto all lights' ISMs. As can be seen in these figures, our method significantly improves visual shadow quality at a relatively low cost, compared to generating all lights' shadow maps.

Using ISMs alone introduces many problems that our approach ameliorates. In particular, small occluders close to the light sources are frequently under-sampled with ISMs, leaving large holes in occluders that cause significant light leaks, even after a pull-push pass (e.g. the tables and chairs in the second row of Figure 10). In addition to the absence of shadow detail, the ISM pull-push pass could also lead to incorrect shadow outcomes in certain areas (e.g. the station floor in the forth row of Figure 10).

In an equal-time comparison with three variations of the distance heuristic, which attempts to mimic how most games choose

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

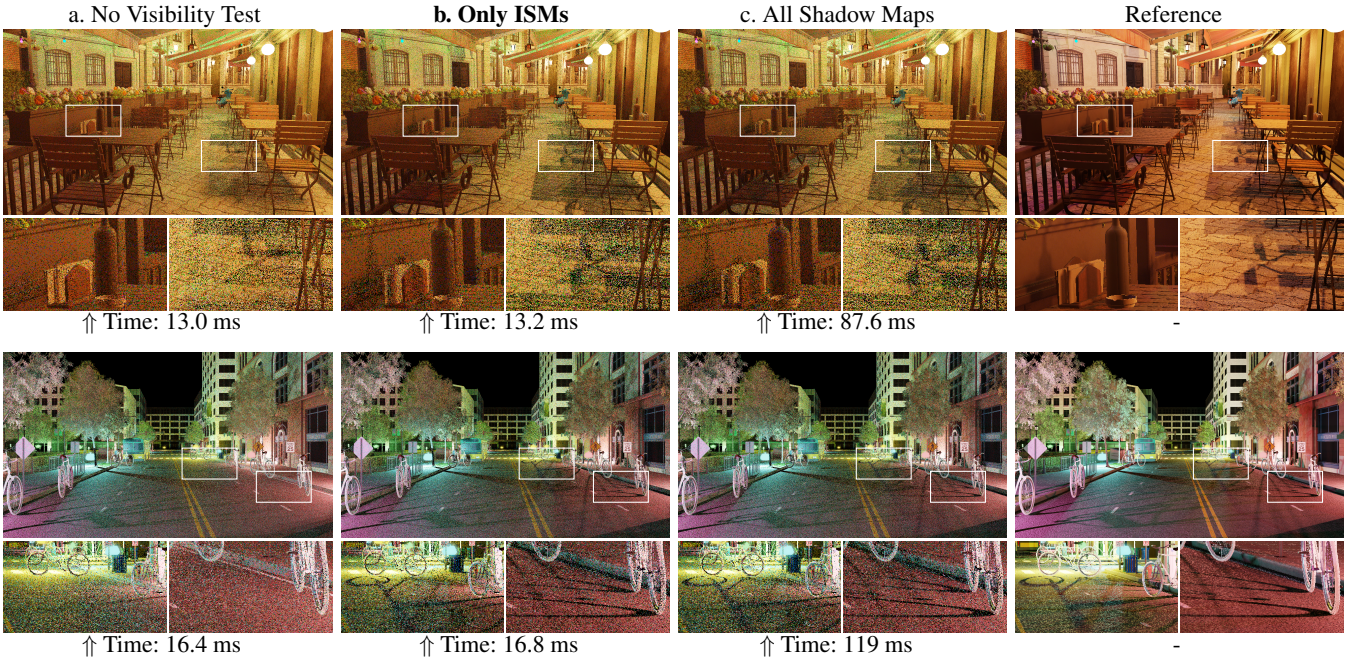


Figure 7: *We compare different options for light ranking. Ignoring visibility in Ranking ReSTIR produces excessive light leaks in the final shading result (a). We use ISMs for visibility tests in Ranking ReSTIR and count how many reservoirs (pixels) select the light for computing a light's score, which produces much better results than no visibility (b). Compare to use all shadow maps, ground-truth ranking (c), the ranking produced by only ISMs is very similar (e.g. in Bistro, only **2** lights are not in both lists), resulting in nearly identical final rendered outcomes. All methods use the same number of shadow maps (i.e. 15 in Bistro and 10 in Emerald Square with flying teapots), and the shading ReSTIR is also the same (i.e. ISMs and shadow maps). The reference uses 611 and 210 shadow maps respectively in Bistro and Emerald Square.*

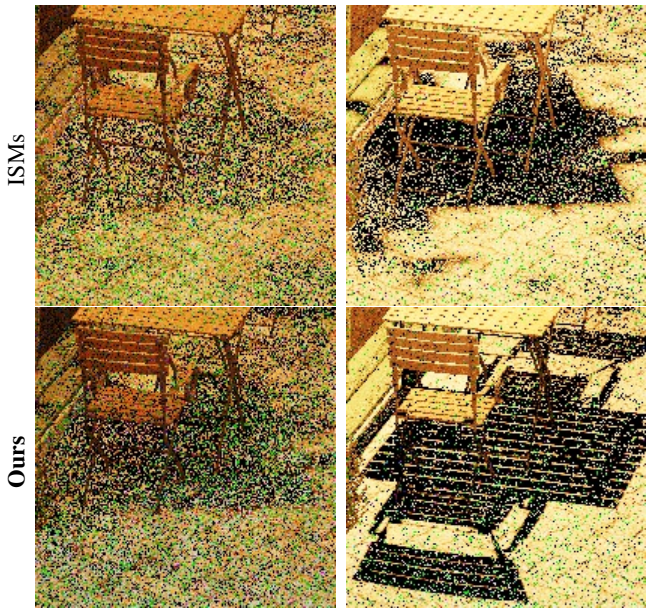


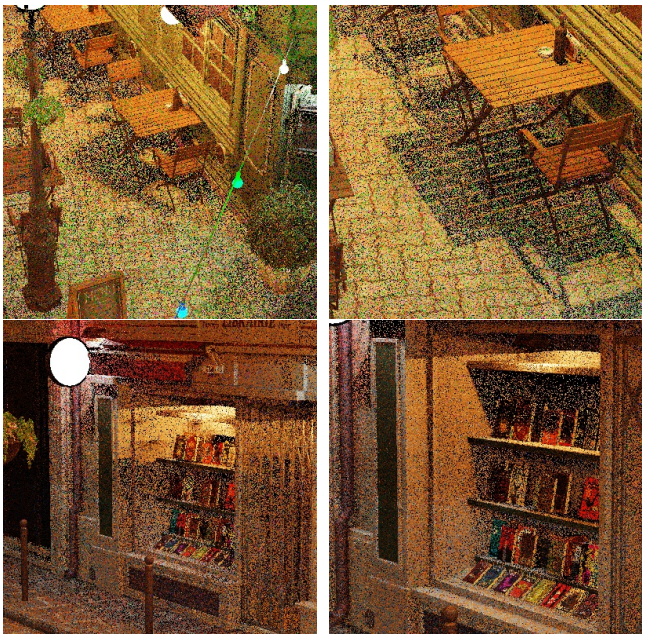
Figure 8: *Increasing light intensity encourages our method (bottom) to transition to more accurate shadow maps, while using ISMs (top) the low-resolution of the map simply becomes more obvious.*

lights for casting shadows, our technique still achieves superior results. Like seen in Figure 6, without the approximation of ISMs for non-top-N lights' shadows, distance heuristics also have excessively dark (e.g. Bistro and Emerald Square) or overly bright (e.g. Subway) problem. Moreover, for lights that are distant from the camera yet cast notable shadows (such as the bicycle's shadow in the first row and Figure 1), the distance heuristic turns out to be ineffective.

In terms of numerical errors, skipping shadowless lights has much lower RMAE than using those lights, and it also has similar RMAE with Ours in some scenes. This is because when we set shadowless lights' visibility to 0, it also terminates all the potential noise in ReSTIR from sampling the lights (especially the color noise in Bistro), where those noise still exists in other methods. However, in Figure 11, we show modern denoiser [SKW*17] can adeptly filter out these ReSTIR sampling noise. With denoising applied, our method produces larger reduction on numerical errors.

Table 1 gives a performance breakdown. We assume a fully-dynamic scene, so frame time includes rendering *all* shadow maps and ISMs from scratch. It is important to point out that our approach does not incorporate space partitioning or precomputed visibility culling, meaning every surface is considered a potential shadow caster. Consequently, the time needed for shadow map creation might be longer, but occlusion culling [BWPP04; BMSW11] could potentially reduce these times.

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.



Original camera Moving camera closer

Figure 9: *Showing how our results evolve under motion. We show two views in the Bistro (left) and move towards interesting shadows (right). As we approach, localized lights have larger impact and typically transition from ISMs to higher resolution shadow maps.*

Rendering shadow maps for all lights is simply infeasible as light counts increases. ISMs are significantly faster, though ISM generation still is the bottleneck. The performance overhead of our light ranking and the secondary ReSTIR reservoir are below 1.0 ms. The main cost of our method over ISMs is the additional shadow map generation passes.

We also provide performance of ReSTIR with ray-traced shadows. Unlike our prototype, this implementation is highly optimized, and provides superior render time. Since we assume fully-dynamic scenes for the shadow mapping methods, Table 1 also includes acceleration structure build times for ray tracing. Depending on the scene animation, this cost can be reduced by refitting rather than rebuilding the acceleration structure.

5. Limitations and Future Work

While ReSTIR identifies key lights, building shadow maps for even this subset may strain our budget. This occurs if many lights cover different parts of the image but no single light dominates (e.g. the third row of Figure 10). When our shadow map budget is insufficient, some important lights must use ISMs. One direction for future work could explore reusing shadow maps across frames, as some may contain largely static data; this could amortize shadow map cost.

While ISMs approximate visibility, they provide relatively poor shadow estimation, especially for direct lighting. This largely stems from their low resolution and uniform point distributions. Another future direction is to develop a more adaptive point distribution to improve ISM accuracy. Additionally, creating intermediate ISMs with different resolutions might allow ReSTIR to bin lights into multiple buckets, using higher resolutions as light importance increases during motion.

Our method can also be used with other shadow approximation techniques than ISMs, e.g., screen-space ray tracing. This would be another interesting direction for future research.

6. Conclusion

We introduced a real-time method handling many lights with shadow maps in dynamic scenes. Our method uses ReSTIR to automatically select lights with the greatest contribution and a secondary ReSTIR pass with a running variance test to provide stability. By getting rid of the shadow rays (for instance from [BWP*20]), we show ReSTIR can also be applied in rasterization. Even with only a small number of shadow maps, we provide much better shadows than classic distance heuristic and ISMs alone, with only a minor overhead. Our approach achieves a quality level comparable to exhaustive shadow mapping for all lights. Beyond this, our prototype has neither constraints on the light properties (e.g. influence range), nor any specific culling on shadow-casting geometries. Thus, it can be applied in existing raster-based rendering systems with little modification.

Our novel approach to shadow map generation aims to showcase the effectiveness of ReSTIR-sampled importance, providing artists with greater flexibility in managing lights and shadow maps within a game engine.

References

[BH13] BURNS, CHRISTOPHER A. and HUNT, WARREN A. "The Visibility Buffer: A Cache-Friendly Approach to Deferred Shading". 2013 6.

[BMSW11] BITTNER, JIŘÍ, MATTAUSCH, OLIVER, SILVENNOINEN, ARI, and WIMMER, MICHAEL. "Shadow caster culling for efficient shadow mapping". *Symposium on Interactive 3D Graphics and Games* (2011) 8.

[BWP*20] BITTERLI, BENEDIKT, WYMAN, CHRIS, PHARR, MATT, et al. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39.4 (July 2020). DOI: 10/gg8xc7 1–4, 6, 9.

[BWPP04] BITTNER, JIŘÍ, WIMMER, MICHAEL, PIRINGER, HARALD, and PURGATHOFER, WERNER. "Coherent hierarchical culling: Hardware occlusion queries made useful". *Computer Graphics Forum*. Vol. 23. 3. Wiley Online Library. 2004, 615–624 8.

[Fer05] FERNANDO, RANDIMA. "Percentage-closer soft shadows". *International Conference on Computer Graphics and Interactive Techniques*. 2005 2.

[KCK*22] KALLWEIT, SIMON, CLARBERG, PETRIK, KOLB, CRAIG, et al. *The Falcor Rendering Framework*. https://github.com/NVIDIAGameWorks/Falcor. Mar. 2022. URL: https://github.com/NVIDIAGameWorks/Falcor 6.

[Kel97] KELLER, ALEXANDER. "Instant Radiosity". *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. 1997, 49–56. DOI: 10.1145/258734.258769 2.

[LKB*22] LIN, DAQI, KETTUNEN, MARKUS, BITTERLI, BENEDIKT, et al. "Generalized Resampled Importance Sampling: Foundations of ReSTIR". *ACM Trans. Graph.* 41.4 (2022), 75:1–75:23. DOI: 10/gqjn7b 1.

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

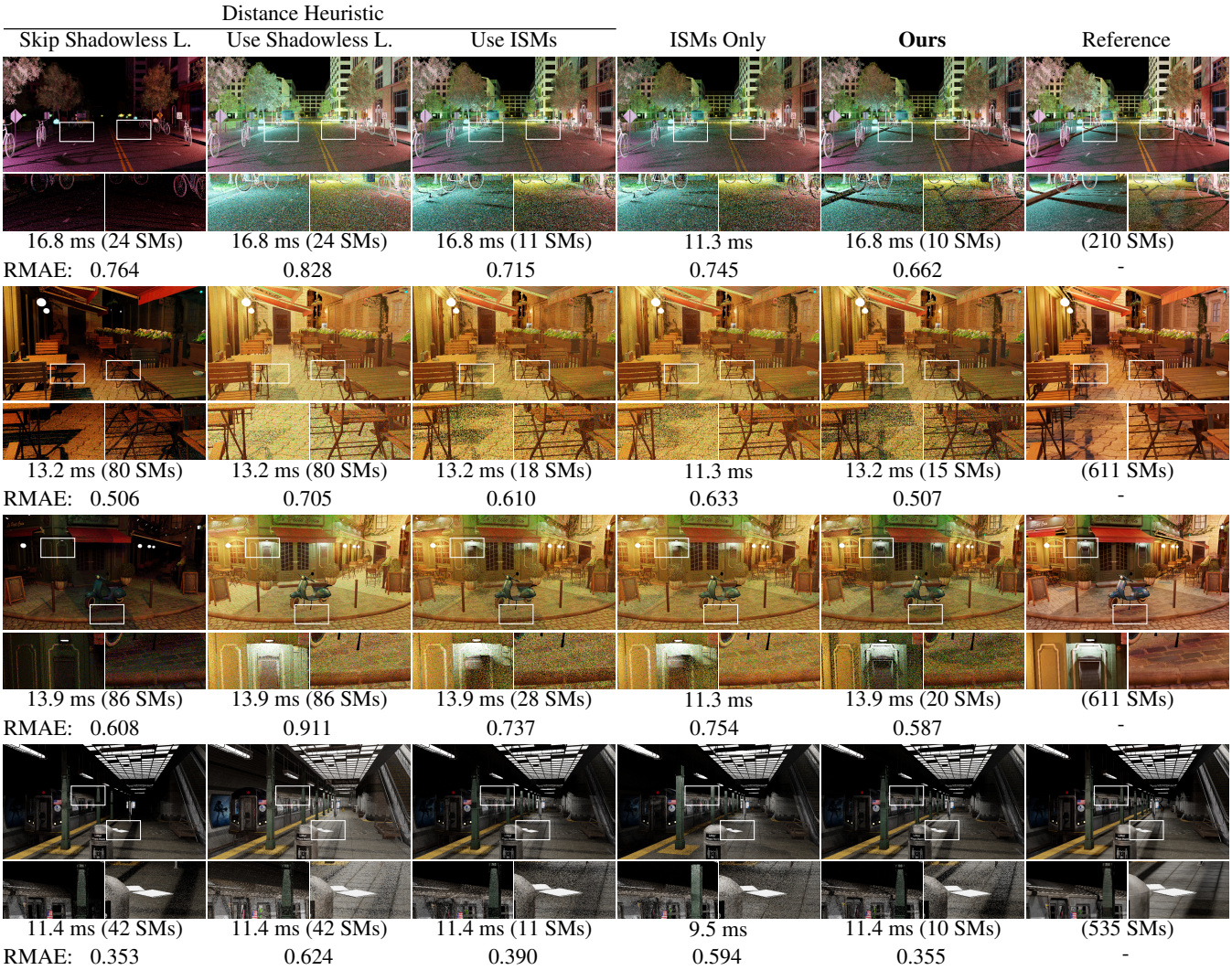


Figure 10: *Compared to using ISMs for all lights and distance heuristic to pick the top-N lights (closest to camera), our method is able to produce better shadow approximation results on both strong hard shadows (e.g. shadows from the bike in Emerald Square), and overall soft shadows (e.g. Bistro), thanks to the powerful ReSTIR ranking. Across all distance heuristic techniques, the ranking ReSTIR remains unchanged. However, for shading ReSTIR, visibility is determined as either 0 or 1, or it is evaluated using ISMs.*

[LWY21] LIN, DAQI, WYMAN, CHRIS, and YUKSEL, CEM. "Fast Volume Rendering with Spatiotemporal Reservoir Resampling". *ACM Trans. Graph.* 40.6 (Dec. 2021), 279:1–279:18. DOI: 10/grrjd6 1.

[LY19] LIN, DAQI and YUKSEL, CEM. "Real-Time Rendering with Lighting Grid Hierarchy". *Proc. ACM Comput. Graph. Interact. Tech.* 2.1 (June 2019). DOI: 10.1145/3321361 2.

[LY20] LIN, DAQI and YUKSEL, CEM. "Real-Time Stochastic Lightcuts". *Proc. ACM Comput. Graph. Interact. Tech.* 3.1 (May 2020). DOI: 10.1145/3384543 2.

[MKC07] MARROQUIM, RICARDO, KRAUS, MARTIN, and CAVALCANTI, PAULO ROMA. "Efficient Point-Based Rendering Using Image Reconstruction". *PBG@Eurographics*. 2007 3.

[MPC22] MOREAU, P., PHARR, M., and CLARBERG, P. "Dynamic Many-Light Sampling for Real-Time Ray Tracing". *Proceedings of the Conference on High-Performance Graphics*. HPG '19. 2022, 21–26. DOI: 10.2312/hpg.20191191 2.

[NVI21] NVIDIA. *RTX Direct Illumination (RTXDI)*. https://developer.nvidia.com/rtx/ray-tracing/rtxdi. Apr. 2021. URL: https://developer.nvidia.com/rtx/ray-tracing/rtxdi 6.

[OA11] OLSSON, OLA and ASSARSSON, ULF. "Tiled Shading". *Journal of Graphics, GPU, and Game Tools* 15.4 (2011), 235–251. DOI: 10.1080/2151237X.2011.621761 2.

[OBS*15] OLSSON, OLA, BILLETER, MARKUS, SINTORN, ERIK, et al. "More Efficient Virtual Shadow Maps for Many Lights". *IEEE Transactions on Visualization and Computer Graphics* 21 (2015), 701–713. URL: https://api.semanticscholar.org/CorpusID:7665065 2.

[OSK*14] OLSSON, OLA, SINTORN, ERIK, KÄMPE, VIKTOR, et al. "Efficient virtual shadow maps for many lights". *Proceedings of the 18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2014). URL: https://api.semanticscholar.org/CorpusID:15883444 2.

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

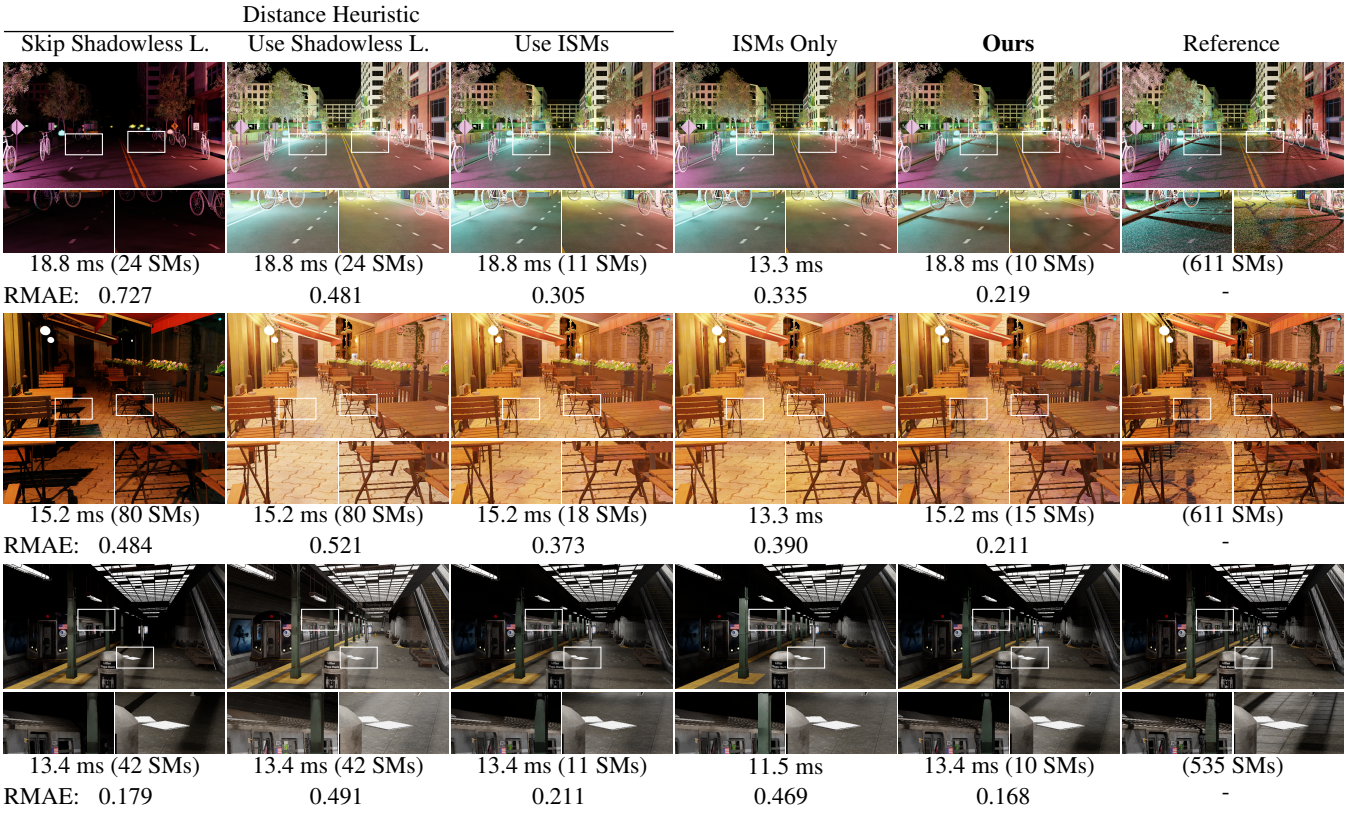


Figure 11: ***(Denoised Results)*** *Modern denoiser can adaptively filter out ReSTIR sampling noise. All settings are the same as Figure 10, and denoising pass takes about 2 ms in all scenes.*

[REH*11] RITSCHEL, TOBIAS, EISEMANN, ELMAR, HA, INWOO, et al. "Making Imperfect Shadow Maps View-Adaptive: High-Quality Global Illumination in Large Dynamic Scenes". *Computer Graphics Forum* 30 (2011) 3.

[RGK*08] RITSCHEL, TOBIAS, GROSCH, THORSTEN, KIM, MIN H., et al. "Imperfect Shadow Maps for Efficient Computation of Indirect Illumination". *ACM Trans. Graph. (Proc. of SIGGRAPH ASIA 2008)* 27.5 (2008) 2, 3.

[RSC87] REEVES, WILLIAM T., SALESIN, D., and COOK, ROBERT L. "Rendering antialiased shadows with depth maps". *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987) 2, 6.

[SKW*17] SCHIED, CHRISTOPH, KAPLANYAN, ANTON, WYMAN, CHRIS, et al. "Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination". *Proceedings of High Performance Graphics*. 2017, 1–12 1, 8.

[SSS*20] SEYB, DARIO, SLOAN, PETER-PIKE, SILVENNOINEN, ARI, et al. "The Design and Evolution of the UberBake Light Baking System". *ACM Trans. Graph.* 39.4 (Aug. 2020). DOI: 10.1145/3386569.3392394 2.

[TCE05] TALBOT, JUSTIN, CLINE, DAVID, and EGBERT, PARRIS. "Importance Resampling for Global Illumination". *Eurographics Symposium on Rendering*. 2005. ISBN: 3-905673-23-1. DOI: 10.2312/EGWR/EGSR05/139-146 2.

[War94] WARD, GREGORY J. "Adaptive Shadow Testing for Ray Tracing". *Photorealistic Rendering in Computer Graphics*. Ed. by BRUNET, P. and JANSEN, F. W. 1994, 11–20 2.

[Wes79] WEST, DAVID H. "Updating mean and variance estimates: an improved method". *Commun. ACM* 22 (1979), 532–535. URL: https://api.semanticscholar.org/CorpusID:30671293 6.

[WFA*05] WALTER, BRUCE, FERNANDEZ, SEBASTIAN, ARBREE, ADAM, et al. "Lightcuts: A Scalable Approach to Illumination". *ACM Trans. Graph.* 24.3 (July 2005), 1098–1107. DOI: 10.1145/1073204.1073318 2.

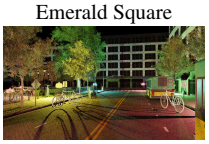
[Wil78] WILLIAMS, LANCE. "Casting Curved Shadows on Curved Surfaces". *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '78. 1978, 270–274. DOI: 10.1145/800248.807402 2, 3.

[WP21] WYMAN, CHRIS and PANTELEEV, ALEXEY. "Rearchitecting Spatiotemporal Resampling for Production". *High-Performance Graphics*. 2021, 23–41. ISBN: 978-3-03868-156-4. DOI: 10/grrjkk 2.

[YLS20] YANG, LEI, LIU, SHIQIU, and SALVI, MARCO. "A Survey of Temporal Antialiasing Techniques". *Computer Graphics Forum* 39 (2020). URL: https://api.semanticscholar.org/CorpusID:220514131 6.

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

Table 1: *Performance breakdown of shadow computation methods using ReSTIR light sampling.*

		Bistro 	Subway 	Emerald Square 
Shadow Maps (for all lights)	Generate Shadow Maps	82.1 ms	95.9 ms	114.2 ms
	ReSTIR Sampling	3.9 ms	3.8 ms	3.6 ms
	Shading	1.6 ms	1.0 ms	1.2 ms
	Render Time	**87.6 ms**	**100.7 ms**	**119.0 ms**
ISMs (for all lights)	Generate ISMs	8.0 ms	6.0 ms	7.6 ms
	ReSTIR Sampling	2.0 ms	2.6 ms	2.5 ms
	Shading	1.3 ms	0.9 ms	1.2 ms
	Render Time	**11.3 ms**	**9.5 ms**	**11.3 ms**
Ours with 10 shadow maps	Compute Ranking	0.1 ms	0.1 ms	0.1 ms
	Update Light Data	0.2 ms	0.2 ms	0.2 ms
	Generate Shadow Maps	1.4 ms	1.8 ms	5.6 ms
	Generate ISMs	8.0 ms	6.0 ms	7.6 ms
	(Two) ReSTIR Sampling	2.1 ms	2.1 ms	2.1 ms
	Shading	0.9 ms	0.9 ms	1.2 ms
	Render Time	**12.7 ms**	**11.1 ms**	**16.8 ms**
Ours with 20 shadow maps	Compute Ranking	0.1 ms	0.1 ms	0.1 ms
	Update Light Data	0.2 ms	0.2 ms	0.2 ms
	Generate Shadow Maps	2.6 ms	3.7 ms	11.4 ms
	Generate ISMs	8.0 ms	6.0 ms	7.6 ms
	(Two) ReSTIR Sampling	2.1 ms	2.1 ms	2.1 ms
	Shading	0.9 ms	0.9 ms	1.2 ms
	Render Time	**13.9 ms**	**13.0 ms**	**22.6 ms**
Shadow Rays (for all lights)	ReSTIR Sampling	3.2 ms	2.9 ms	4.4 ms
	Shading	0.9 ms	0.7 ms	1.5 ms
	Render Time	**4.1 ms**	**3.6 ms**	**5.9 ms**
	BVH Build Time	**9.4 ms**	**20.3 ms**	**9.0 ms**

© 2025 The Authors.
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.