

Adaptive Caustics Rendering in Production with Photon Guiding

Alejandro Conty and Christopher Kulla

Sony Pictures Imageworks



Figure 1: A 2k render of volumetric and surface caustics where the scene is much bigger than the visible part. We render progressively turning the bias into noise so the renderer can adaptively refine based on pixel and sample variance. The most inexpensive result (3 minutes) is already suitable to feed to a denoiser, but we show how the render itself can converge to a clean result if given time. Samples per pixel do not include photon tracing, but are on par since we shot a photon for every two spp. Times are on a 12 core workstation.

Abstract

We present a user controlled technique for achieving fast and stable caustics in a production renderer for both surface and participating media. We combine a progressive photon mapping approach with emission guiding in an on-demand framework that avoids the raytracing overhead of robust bidirectional systems. We also contribute modifications to turn bias into noise while speeding up the render, making the result usable for both adaptive picture refinement and denoisers.

1. Introduction and Motivation

Rendering in the context of film production is now dominated by unidirectional path tracing [Pha18] which has been prized for its simplicity and robustness across a wide range of scenes. One remaining weakness however has been poor performance on sharp, focused caustics. Most renderers to date have chosen to either ignore this effect by clamping light paths, developing coarse approximation such as transparent shadows for glass, or developing custom importance sampling techniques for important subsets of cases [HDF15].

This paper presents our approach at enabling the rendering of hero caustic lighting effects within an existing production path tracer. Rather than solving for all possible caustic paths in a scene,

we specifically target our method to light and object pairs chosen by the artist. This is in contrast to other recent industrial implementations which aim to provide a global caustic solution with a single click [ŠK19]. They solve a much harder problem as they aim to capture all possible light interactions with metallic or transmissive surfaces. In the context of movie visual effects production, we found that artists rarely were interested in all possible paths and instead wanted to more carefully control which paths to add to the image to enhance aspects important to the story. However we do share the goal of wanting to provide an experience for the artist that is free of any technical parameters to tune per scene. Thus our method can be considered a two-click solution: one click to enable caustics on the light source, one click to enable caustics on the casting object.

Our renderer [KCSG18] previously implemented the VCM algorithm [GKDS12] combined with a Metropolis sampler (similar to the work by Šik et. al. [vOHK16]). However, despite this integrator’s high robustness at rendering a variety of challenging scenes, artists found it most useful for rendering caustic passes which were then composited back into the final frame. In other cases, the overhead of combining many techniques outweighed the convergence benefits. This drove us to refocus our efforts specifically on caustics and search for a method that could be seamlessly integrated within our workhorse path tracing integrator.

2. Previous Work

Our work is directly based on the classic photon mapping algorithm [Jen96], and its generalization to volumes [JNSJ11]. The VCM/UPS [GKDS12, HPJ12] and later UPBP [KGH*14] papers further improved the robustness of the original methods from a theoretical perspective as they account for all shortcomings of the individual techniques they combine through multiple importance sampling [VG95]. However in practice, many techniques considered end up being weighted to nearly zero, leading to a large amounts of wasted computation. Our method employs an MIS combination of path tracing and photon mapping only, and further approximates the weights to make the relative weighting between the techniques easy to compute locally with minimal path state requirements.

We are not the first to highlight the importance of caustics in production rendering. We highlight a few industrial strength methods here to motivate our desire to develop our own approach.

Manifold Next-Event-Estimation [HDF15] is a specialized importance sampling technique which focuses on specular-diffuse-specular interactions caused by refractive interfaces. It is specially designed to render caustics within eyes or droplets of water, however, it cannot handle reflected caustic effects which we hoped to include. The original paper also did not address the extension to participating media which seemed challenging.

The Hyperion rendering system [BAC*18] includes a dedicated caustic solver used in the rendering of ocean scenes. As this system was tailored to a specific effect, its feature set was limited to a single distant light and did not include participating media. Nonetheless, our method resembles this system very closely. The remainder of our paper will detail some subtle issues we had to overcome in our implementation.

As mentioned above, the caustic solver integrated in the Corona rendering system [ŠK19] shares many similar goals, particularly in artist usability. Their work also excellently describes shortcomings of previous approaches. On the other hand, they aim to solve the more challenging problem of rendering all potential caustic paths in the scene, while we want to focus computation on only those paths which the artist deems visually significant. As in our system, they found path tracing and photon mapping to be the two techniques most worthwhile to combine. This system has, to our knowledge, not yet been extended to participating media.

Recent work on path guiding [VHH*19] has shown it can be used successfully in a production context. Our work also incorporates a guiding component to try to learn the importance distribution for

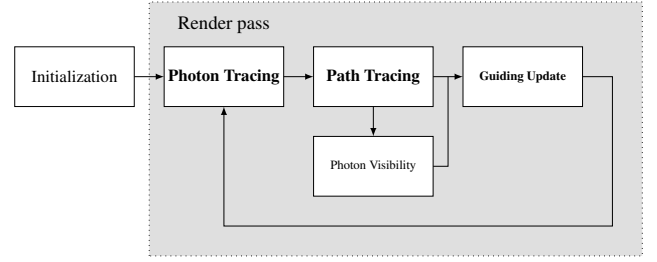


Figure 2: Our rendering process allows feedback from rendering to influence the photon tracing in the next iteration. The photon map is discarded at each pass, but the guiding update allows photon tracing to improve over time.

photons over time. While some work has demonstrated rendering of prominent caustics from purely unidirectional methods [MGN17], our experience was that such global algorithms could still be costly for smaller, localized caustic effects. The extension of such methods to participating media is also challenging, despite recent promising results [HZE*19, SHJD18].

Finally, we draw attention to the fact that many current production rendering systems [KCSG18] [BAC*18] [FHL*18][CFS*18] [GIF*18] use some form of image space adaptive sampling driven by variance to terminate the rendering process. This is desirable because convergence is not always uniform in image space. However when incorporating an algorithm such as photon mapping a unique challenge appears. Unconverged photons (or beams) can show up as splotches (or lines) which could be mistaken for low variance image features. As such, the image space adaptive sampler may decide a pixel has already converged and terminate it without giving an opportunity for more photons to be accumulated in those areas. To our knowledge this interaction has not been explored in previous work, and we aim to address it this paper.

3. Method Overview

Our algorithm is inspired by Progressive Photon Mapping [HOJ08] and interleaves photon tracing with path tracing. Since uniform photon emission is notoriously inefficient [HJ11] we add a guiding update process after we complete each path tracing pass to refine the structure used to control photon emission. Figure 2 summarizes the overall approach. As we expect this overall structure to be familiar, we will focus this paper on the unique choices we made.

Photon tracing only happens between selected light and object pairs (Figure 3). A 2D importance table presents the non-uniform probability of emitting a photon in any given direction. A single 2D table is used for all light-object pairs. This 2D table is refined over time by the guiding update at the end of each rendering pass. We detail this process in Section 4.

During path tracing, we accumulate a visibility signal to indicate which photons and beams were used. This visibility signal is then used to refine the 2D importance table used to guide emission. We detail this process in Section 5.

When gathering photons, we do not use the standard approach of using all photons within the lookup area, and instead randomize the

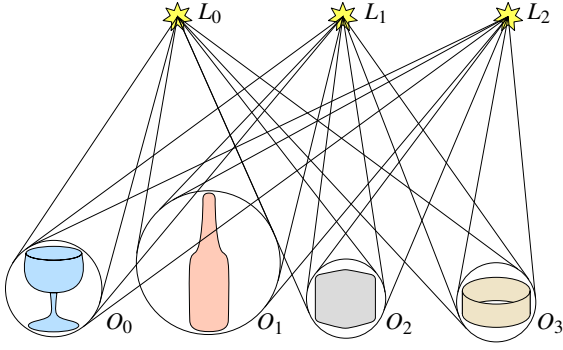


Figure 3: Chosen lights emit photons only towards the bounding spheres of selected objects. The resulting caustics can affect any part of the scene.

photon (or beam) size so that rendering error appears as noise which the image space adaptive sampler can recognize. This prevents adaptive image sampling from terminating refinement early. This is detailed in Section 6.

Finally, we use a simplified form of MIS to combine the probability of constructing the caustic path through photon tracing and path tracing. This ensures we maintain robustness to cases involving large light sources. This is detailed in Section 7.

4. Photon Guiding

In this section we detail how we use a single 2D table to drive the photon tracing process over the lights and objects selected by the artist. This is a form of primary-space guiding, which operates on the random numbers as it has already been approached in the neutron transport field [Boo86] and more specifically to photon tracing [GPGSK18].

4.1. Light-Object Matrix

The artist identifies L lights and O objects that will participate in caustic rendering. To partition the total budget of photons among these, we assign a probability to each pair that we can visualize as a matrix M of dimension $L \times O$. The entries of the matrix are normalized: $\sum M_{lo} = 1$ to ensure we have a valid discrete PDF to make our selection. For instance:

$$M = \begin{pmatrix} 0.1 & 0.0 & 0.2 & 0 \\ 0.15 & 0.1 & 0 & 0.05 \\ 0.1 & 0 & 0.1 & 0.2 \end{pmatrix},$$

can be directly interpreted as the percentage of photons in our budget to assign to each pair. After selecting a pair, we trace photons by picking a direction from the light toward the bounding sphere of the chosen object as show in Figure 3.

This small matrix can then be combined with a 2D importance map for choosing directions. But because both can be thought of as 2D tables, we collapse both tables into one and maintain a single large 2D table of resolution $m \times n$ that represents both the probability of picking a particular light-object pair *and* a particular direction within the light-object cone. Figure 4 shows an example of this basic

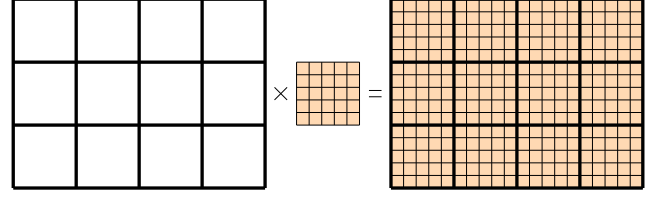


Figure 4: Expanding the light-object table with a fixed grid size allows importance sampling of directions and light-object pairs at the same time, though with sub-optimal resolution allocation.

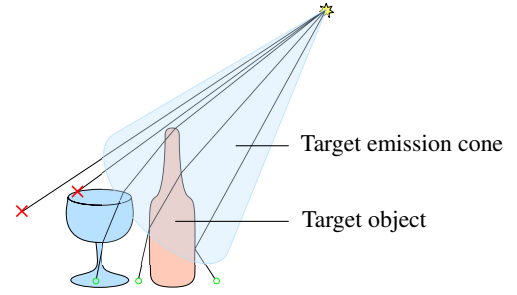


Figure 5: Photons that miss the target object on the first hit are rejected even if the hit object is tagged as caustic. Photons are stored where necessary (green circles) and the walk is terminated when no caustics-tagged object is hit.

layout. In Section 4.2 we will discuss how to relax the layout to better allocate the resolution among the respective pairs.

It is important to note that these sampling cones can overlap. To avoid a complex accounting of this in the emission PDF we simply employ rejection sampling and only continue paths if they hit the expected target object as shown in Figure 5. Beyond this first segment, there is no longer any restriction on the caustic path.

While rejection sampling may seem wasteful, the guiding process will quickly learn to avoid these directions, as well as those that hit the bounding sphere of the object but not the object itself.

4.2. Light-Object Atlas

The simple layout pictured in Figure 4 may not make efficient use of the available resolution if object and light sizes vary within the scene. For this reason, we allow each light-object pair to have a different resolution, while still keeping them all packed into one table. For this we draw inspiration from texture atlas packing algorithms.

We want to assign every light-object (l, o) pair a square area A_{lo} of the guiding table. Unlike the probabilities in the guiding table, this resolution allocation will be fixed for the whole render (guiding will be able to compensate for misallocations to some extent). We use three factors in our heuristic.

- Ω_o : Solid angle on camera of the target o , clamped to the solid angle of the camera plane.
- S_{lo} : A singularity measure in $[0, 1]$ of the light l as seen from the target o .

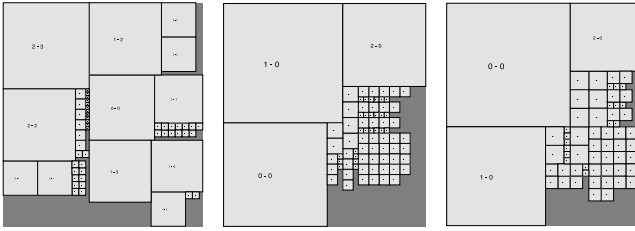


Figure 6: Example layouts from our atlas packing algorithm with three different sets of light-object pairs of random sizes. In practice, utilization stays over 90% for most real world cases.

- w_{lo} : Average weight (radiance over PDF) of the samples on light l as sampled from the center of target o , in solid angle space.

The first factor Ω is trivial to compute based on the bounding sphere size and distance to camera. For the second and third we perform a small integration of the light from the target’s center to get the average radiance and at the same time compute the average PDF P_{avg} . We use:

$$S_{lo} = \frac{1}{1 + 1/P_{\text{avg}}^2}, \quad (1)$$

as a heuristic for the relevance of a light in the photon mapped caustics context. Lights that have a large average pdf must be close to singular and therefore should get more resolution. As the average pdf goes down the light source must be getting larger and therefore benefits less from photon mapping, thus requiring less resolution allocated in the guiding table.

From these three factors we can define a light-object pair importance as:

$$I_{lo} = \Omega_j S_{lo} w_{lo}. \quad (2)$$

We assign a square A_{I_o} to each pair with area proportional to this importance I_{I_o} . Each area is filled with a uniform probability to seed the learning process. Since our importance table is discrete, we work in integer coordinates to layout these boxes.

We use a simple greedy algorithm to perform this layout. It keeps track of a the list rectangular "holes" over the domain that are not yet occupied. The algorithm starts with a hole covering the whole table. Every time we place an element into the table, the holes get fragmented, but we keep them sorted by distance to the origin and size to pack larger boxes first. We show some example packings for a large collection of light-object pairs of random sizes in Figure 6.

After the layout is done, each pixel in our guiding table corresponds to a single light-object pair or is simply wasted space in which case we fix its probability at 0. This way space left over from the packing procedure does not impact the sampling at all. We found that wasting a bit of space in the atlas packing was preferable to stretching the domains to try and achieve better occupancy.

4.3. Guided Photon Emission

Once the light-object atlas is computed, a discrete 2D table M has to be built over it. Its aspect ratio will depend on the atlas layout, which will tend to be square. We choose its size m, n coupled with

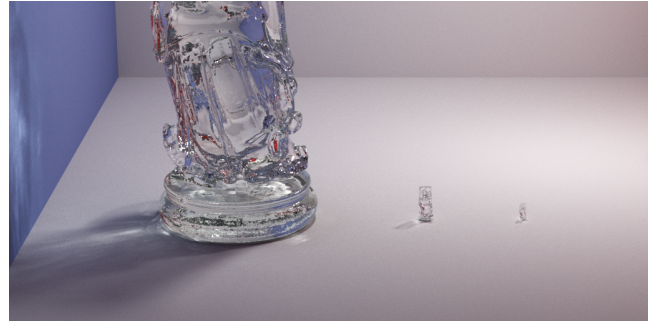


Figure 7: Three differently sized Buddhas casting caustics from a pair of lights. The visibility signal for this image is shown in Figure 8.

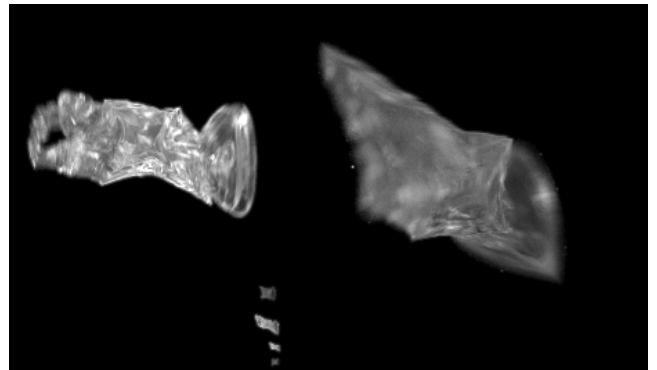


Figure 8: Visibility signal integrated for photon guiding while rendering Figure 7. There are two lights and three objects, so we see six shapes where the two big ones correspond to the bigger Buddha. Table utilization is 90%.

the output image resolution so $m \cdot n = 0.25 \cdot \rho^2 \cdot \text{width} \cdot \text{height}$, where ρ is a user supplied parameter and the number of photons per pass is $\rho^2 \cdot \text{spp} \cdot \text{width} \cdot \text{height}$. This way when $\text{spp} = 1$, in the first pass, we will emit 4 photons per guiding cell in M .

Emitting photons from a light usually involves sampling a point on the surface of the light, then a direction. This requires four random numbers $\xi = (\xi_0, \xi_1)$ for the light's surface and $\gamma = (\gamma_0, \gamma_1)$ for the direction. Since we assume the light size for caustic effects will typically be small, we apply our guiding to γ_0 and γ_1 which start in the $[0, 1)^2$ unit square.

The first step is to remap γ using the importance table M , sampling a row then a column (though hierarchical warping could also be used). This discrete decision, leaves remaining random bits from γ in γ'_0 and γ'_1 . For a table of m rows and n columns:

$$\begin{aligned} i, \gamma'_0 &= \text{samplerow}(M, \gamma_0) \\ j, \gamma'_1 &= \text{samplecolumn}(M_i, \gamma_1) \\ \beta &= \left(\frac{i + \gamma'_0}{m}, \frac{j + \gamma'_1}{n} \right), \end{aligned} \quad (3)$$

where β is the remapped pair of random numbers that fall in the cell

M_{ij} and have a new probability $m \cdot n \cdot M_{ij}$. From i, j we also locate the parent atlas region A_{lo} .

Finally we can sample a direction from the light in the cone towards the target's bounding sphere using ξ and β' , where β' is β rescaled to the rectangle A_{lo} :

$$\beta' = \left(\frac{m\beta_0 - x_{lo}}{\text{width}_{lo}}, \frac{n\beta_1 - y_{lo}}{\text{width}_{lo}} \right) \quad (4)$$

This further scales the PDF by $\frac{\text{width}_{lo}^2}{m \cdot n}$ leaving a scale of $\text{width}_{lo}^2 \cdot M_{ij}$ for the probability of the photon.

As we will see in Section 6, having a notion of radius and spread for the photon is important for estimating our guiding signal. We want these to be a function of the photon probability and solid angle of the sampled direction cone. If a particular cell has high probability, photons emitted from that cell should have a small radius to compensate for the number of photons that will be traced from that cell. Every cell M_{ij} in the sampling table maps to a light-object pair where the number of allocated cells is $|A_{lo}|$. If N_p is the number of photons per cell shot at a given render pass, we can define the photon density at a cell as:

$$D_{ij} = N_p |A_{lo}| M_{ij}. \quad (5)$$

Then we use the solid angle of the light-object emission cone Ω to assign a per photon disk of Ω/D_{ij} steradians and finally compute spread from the radius of that disk. An analogous heuristic can be used to compute the radius for infinite lights when spread is zero and an orthographic projection applies.

In the following section we will detail how we make use of this photon radius and spread to integrate a visibility signal that drives the update of the guiding table. It is important to note that while we lookup photons with a pixel footprint sized radius to minimize bias, we want our visibility signal to reflect the region of influence of the photon cone so that we can robustly capture its influence in the guiding map. Photons from low density cells affect a proportionally larger area that needs to be noticed for the learning process to correct the density deficiency.

5. Guiding Table Update

Photon visibility is the signal we use to drive the guiding table probabilities. It aims to capture which photons were useful to the path tracer, so we can increase photon density along those directions in subsequent passes.

During path tracing, when a photon is used for lighting, visibility is accumulated back into the guiding table proportionally to its throughput. An important property of production rendering systems is that the rendering system must be strictly deterministic. Rendering the same scene several times should produce bit-wise identical results. This can be tricky to achieve in a guiding context as tiny numerical differences can be amplified by the learning process to yield dramatically different outcomes. Therefore we implement the photon visibility accumulation with fixed-point arithmetic rather than floating point to prevent the non-associativity of floating point addition from influencing the guiding process during multi-threaded accumulation. This wouldn't be necessary if the path-tracing stage

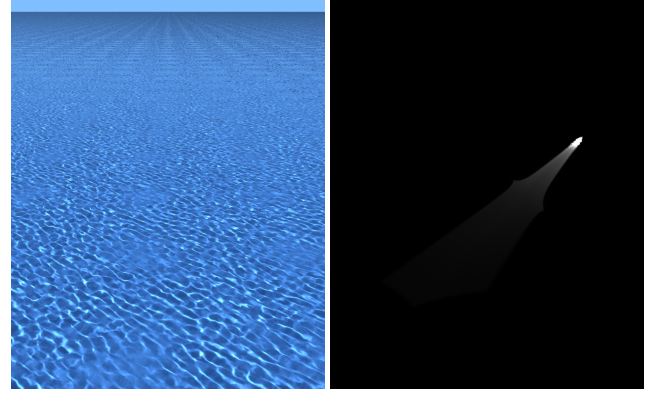


Figure 9: When photons need to be distributed from very close distances to very far, our visibility signal becomes more intense near the camera (right) where there are more pixels using the photons.

could guarantee a deterministic pixel ordering, but this is not practical when the computation cost of a pixel is not known ahead of time. Once a rendering pass is completed, the results can be merged back into a floating point CDF safely for consumption by the next pass.

Figure 8 shows our visibility signal for the scene from Figure 7. Because photons that do not hit the target object are immediately discarded, we can clearly distinguish the outline of the target Buddha models in the guiding table. The black areas are cells whose directions did not hit the target object at all, or where the photons land out of camera reach.

Figure 9 shows another example of a visibility signal. Here a tiny portion of a very large ocean plane is visible to the camera. Starting from a uniform density of photons across the ocean floor, the camera lookups to visible photons drive a quick increase in the visibility signal close to the camera which tapers off with distance. This in turn causes more photons to be traced toward these areas in subsequent passes which further reinforces which photons are important.

As will be detailed in Section 6, we support both surface and volume caustics via photon beams. It is expected that a beam will be visible by many more pixels than a surface photon. To deal with this unbalance we accumulate visibility to two different buffers depending on the surface/volume origin, and then we combine both after dividing by the maximum visibility on both buffers independently. The resulting importance table is the per-pixel maximum of both buffers after this normalization. This allows surface photons and beams to influence the guiding step in an equitable manner.

The very first photon tracing pass will use uniform probabilities, as such, it frequently leads to a low quality photon distribution. For this reason, we do not allow the very first photon tracing pass to contribute to the image at all. However we do still use the lookups to update the visibility signal. When rendering the second pass, we slightly increase the contribution of these photons to compensate for the missing energy in the first pass. The scale is simply $\frac{N_1 + N_2}{N_2}$, where N_i is the number of pixel samples in pass i .

The 2D visibility signal V^n at pass n is nothing more than a Monte-Carlo rendering with an unusual camera projection based on the photon M_{ij} origin. Therefore it refines with every pass according

to a sample average like the actual output. It can be expressed as

$$I_{ij} = \frac{1}{N_n} \sum_{k=1}^{N_n} \text{cell}(p_k, i, j) \frac{f(p_k)}{M_{ij}^n}, \quad (6)$$

$$V^n = \frac{T_{n-1} V^{n-1} + N_n I}{T_n}$$

where M_{ij}^n is the probability of emitting a photon from the cell i, j , $f(p_k)$ is the throughput of the photon, $\text{cell}()$ is a per-cell box filter, and $T_n = \sum^n N_i$. All constrained by the superscript n to the current pass. But while tracing photons, the radius spread is already proportional to N_n and M_{ij}^n , meaning low probability photons have bigger radii and are visible by more pixels than lower probability ones. This makes the $\frac{1}{N_n M_{ij}^n}$ Monte-Carlo factor implicit. We adjust the accumulation to avoid double counting it.

Finally let $K_n = 1/\sum V_{ij}^n$ be the visibility normalization constant and U be the initial uniform distribution, then we choose an arbitrary blending factor t for defining

$$M^1 = U, \quad (7)$$

$$M^n = \text{normalize} \left(t U + (1-t) K_{n-1} V^{n-1} \right).$$

This blending factor was chosen from empirical testing as $1/n$. This way the uniform distribution fades away slow enough for the sampling to be reliable in finding visible features. Also note this is tailored to our renderer using a number of samples per pixel which grows linearly from pass to pass as $\text{spp} = 2n - 1$, so the total at any given pass n amounts to n^2 spp.

Most path guiding solutions to date have attempted to build a structure to capture the global radiance of the scene, which requires a complex multi-dimensional structure (such as a 5D SD-tree [MGN17]) which is mainly beneficial to steer paths over many bounces. In the specific case of caustic paths traced from the light, the first sampling decision is the most critical one. The caustics which are most difficult to capture with pure path tracing are from small lights hitting smooth surfaces. Therefore, once we choose the initial ray direction, there is little value in trying to further control the direction of the photons. This allows us to focus solely on the initial photon direction and get away with learning this much simpler 2D signal.

6. Stochastic Photon Lookups

During the main path tracing pass, we gather photons following the progressive photon mapping approach. However we modify the process slightly to prevent photon and beams from being recognized as features by the image space adaptive sampler. As mentioned earlier, the photon visibility accumulation also happens during this stage and uses slightly different rules.

6.1. Estimation on Surfaces

We use standard density estimation to connect photons with eye-paths according to a smooth kernel [Jen96]. However, for guiding purposes, our photons should not be treated as simple points, rather they carry a radius as described in Section 4 which accounts for the photon spacing between the emitted photons. High probability

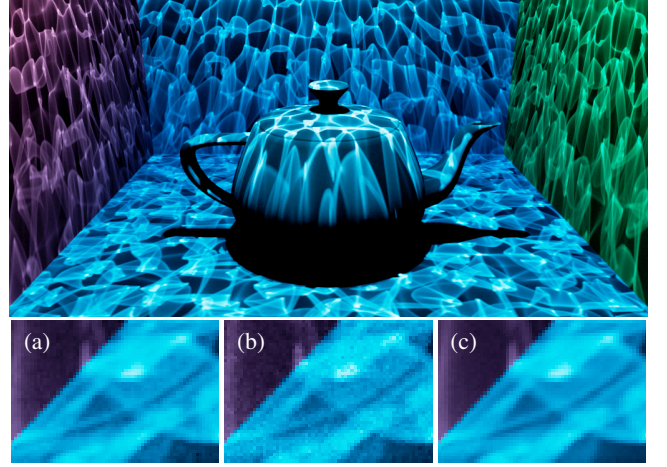


Figure 10: An underwater Cornell box. In the insets we compare density estimation using all photons within the pixel radius (a), to our stochastic technique using only one (b). The former appears converged, but is not when compared to letting the stochastic approach refine adaptively (c).

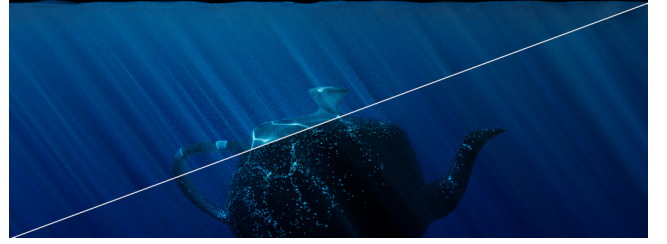


Figure 11: A comparison of our guiding against naive photon emission on a scene where only a small portion of the light's lit area is visible from the camera. The render uses 64spp and one photon path per eye path.

photons will be small, while low probability photons will cover a larger area.

Using these large radii during rendering would lead to very high bias that would be hard to recover from even as the radius shrinks due to our learning step. Instead, we only use the large radius for accumulating the visibility signal and clip the rendering radius to the pixel footprint.

This still leaves us with two issues: a potentially high number of BSDF evaluations in high density areas, and low frequency noise. The low frequency noise from density estimation will confuse the image space adaptive rendering (which tracks pixel variance), as well as later denoising post-processes which are also based on pixel variance. To mitigate both problems at once we build a simple CDF over the gathered photons proportional to their energy and the inverse of the radius squared. Then we stochastically select a single photon for shading.

As shown in Figure 10, this turns the low frequency noise into high frequency, allowing image adaptive refinement (or denoising)

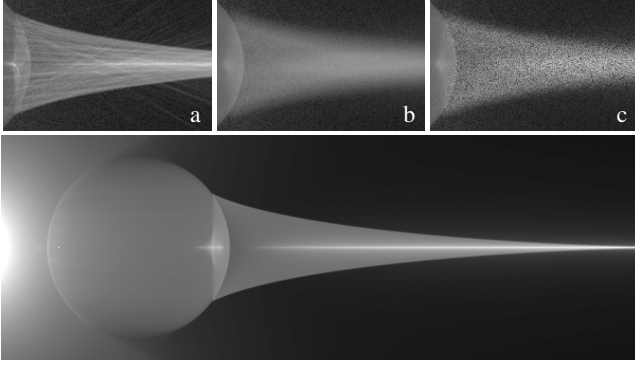


Figure 12: Volumetric caustics from a glass sphere. Using the pixel size width gives a sharp result (a) in 6m2s, but the line-shaped noise will be confused for detail. Using all beams with the width from sampling density gives a blurry result (b) and takes 7m36s to render. Neither render will be refined. Our method (c) reduces the time to 3m26s. The more natural noise allows the render can be refined adaptively to obtain the bottom render in 19m18s.

to keep targeting those pixels. In production scenes with complex BSDFs, the savings can be substantial.

6.2. Estimation in Volumes

For participating media we use photon beams [JNSJ11]. We chose the 1D estimator for its simplicity. We use the photon ray’s spread factor to assign a width to the beam ribbons similarly to surface photons. Once again, this implies a fairly large bias in early passes when the beams are still wide. In the volume context, estimating volumetric radiance is even more expensive since every beam the camera ray intersects needs to traverse a BVH over the beams, plus additional operations for computing the contribution.

Using an narrow beam width, for example sized by the query ray’s differentials, is not as helpful as it was with surface photons. The resulting line noise can take a long time to converge and is very distracting visually. We address these issues in two ways:

1. Randomize the beam width between the query ray width (pixel sized) and the photon beam width (which is density driven) on a per-ray basis.
2. Sample a maximum merge distance $t_{\max}$ along the query ray based on the medium attenuation and ignore all beams beyond this value.

The second measure has previously been suggested in the literature [JNSJ11]. If the medium is uniform and monochromatic, the exponential terms will cancel and save computation. Even if this does not hold in more general case of colored heterogeneous media, it is still interesting to consider because it can greatly reduce the amount of computation without introducing additional bias (see Figure 12).

6.2.1. Randomized Beam Width

We want to limit the minimum width of the beams to the approximate pixel size, but during the initial passes, when the density is still low,

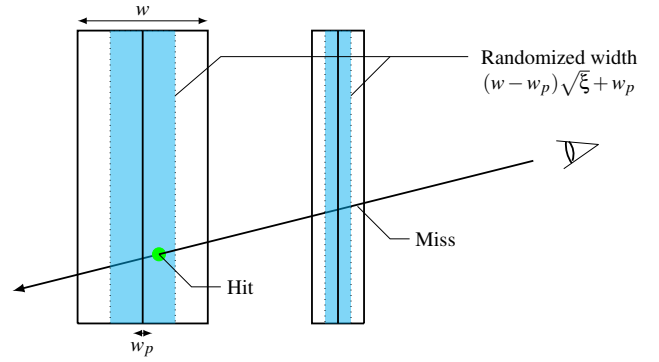


Figure 13: Randomly scaling beams between the tiny pixel width w_p and the density based width w per ray.

the widths are bigger to avoid *laser-beam* artifacts and to help with the guiding process. This can produce substantial bias that could be mistaken to be a smooth converged feature. We again convert this bias into noise by randomizing the width from ray to ray in the camera path.

Beam width can go from a pixel sized minimum w_p (via the query ray’s differentials) to the original width w computed during photon tracing. We use a random number to sample the final width between these two numbers. If this value turns out to be smaller than the distance from the ray to the beam we skip it and continue tracing.

Only one random number ξ is generated per ray to scale widths for all intersected beams. This random scaling is computed proportionally to $\sqrt{\xi}$ so the implicit kernel is a triangle function. This avoids creating a singularity near the center which would make beam unnecessarily sharper (see Appendix A for a derivation of the implicit kernel).

6.2.2. Unbiased Beam Culling

In scenes where the participating media extends over a long distance, it is possible for distant beams to be present but have relatively low overall contribution to the image. We account for this by only considering beams closer than a randomly chosen distance. This distance is sampled proportionally to extinction along the eye ray.

When weighing the contribution of a beam, we must take into account the stochastic clip distance by dividing the beam contribution by the probability of not clipping the beam. In an infinite homogeneous medium, this probability is exactly equal to the transmittance, which conveniently cancels out the transmittance term leading up to the beam itself. Achieving the same cancellation for finite homogeneous media requires a bit more care however. The probability of shading the beam tends toward 0 towards the end of the integration segment, which makes the overall weight raise sharply above 1 near the end of the segment which can lead to fireflies. To avoid this problem, we mix in a non-zero probability of not clipping the segment at all. Setting the probability of this defensive sampling to the transmittance through the segment restores an overall weight of 1. This process can be similarly extended to heterogeneous media (see Figure 14) and spectrally varying media (see Figure 15) in a similar way. The aggressiveness of beam culling can also be tuned

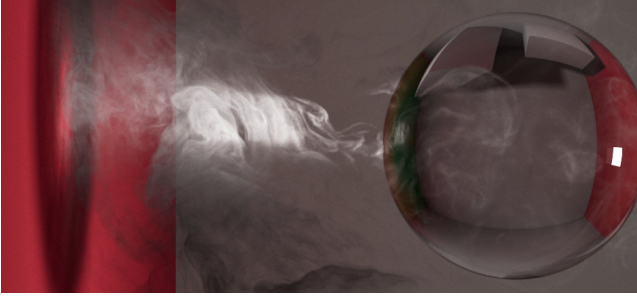


Figure 14: Our renderer uses biased ray-marching for heterogeneous media. Sampling from a tabulated CDF, followed by analytic sampling within each step. Our unbiased culling is applicable to this case as well.

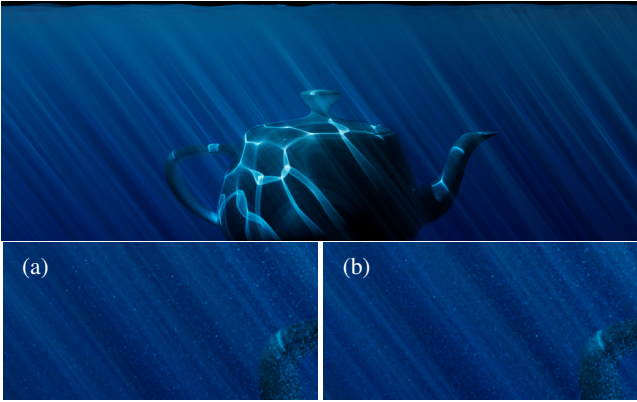


Figure 15: An example of an expansive water surface where most beams in the distance contribute very little. The unconverged insets show gathering all beams (a) 6m47s against our unbiased culling (b) 2m30s. We are able to produce the converged picture on the top in 1h.

by artificially thinning out the medium density used for sampling the distance. We find that sampling a distance with half the density of the volume gives a good balance between speed and quality.

In Figure 15 we show how this optimization reduces the render time dramatically in scenes with deep volumes. The contribution of the distant beams is small enough that the variance increase is negligible.

7. Multiple Importance Sampling

7.1. Approximate Path Weights

The last implementation detail to cover is how to weight the photon mapping path construction with the path tracer. Paths can be constructed by any of these techniques:

- Pure path tracing from camera to light (good for large lights).
- Path tracing with next-event estimation (good for small lights and rough surfaces).
- Photon mapping (good for small lights and smooth surfaces).

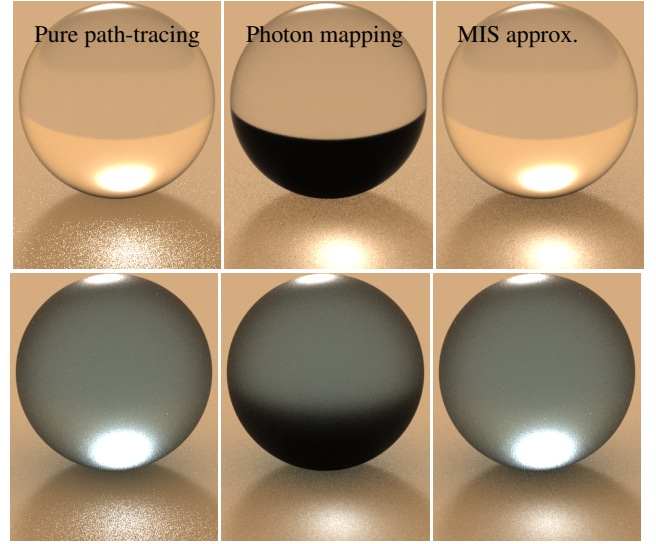


Figure 16: Our mixing heuristic between path tracing and photon estimation. This scene has a smooth skydome with a very intense sun. Photons capture the sun caustic well, but are not useful for capturing the smooth caustic of the sky. Since we do not store photons on objects casting caustics, the sky renders black through the sphere with photon mapping alone. Bottom row shows same comparison with higher roughness.

Computing the exact PDF would require to track extra data over the paths involving backwards PDF evaluations. Keeping in mind an eventual GPU implementation of our code, we wanted to minimize the amount of extra state required per path, and minimize the number of PDF evaluations.

MIS between BSDF and light sampling for next-event-estimation (NEE) is well understood and can be computed using only the information local to the last segment of the path. Since our renderer already uses this technique, we decided to simplify the MIS for photons and compute a weight between the regular lighting estimate and photon mapping only.

Our approximate MIS blends the combined direct lighting estimate (including both BSDF and light sampling) against the photon lookups. This way we do not sacrifice the accuracy (or need to modify the code) of the regular path tracer’s next event estimation logic. We simply compute two new weights W_l and W_p (for regular lighting and photon mapping respectively). We compute these weights from two quantities:

- P_b is the last BSDF PDF for the outgoing ray before a caustic surface is hit.
- P_l is the minimum of the light sampling probability and the BSDF PDF for the last ray of the path.

From these factors, weights are defined as:

$$W_l = \frac{1}{1 + \frac{N_p P_l}{P_b}} \quad (8)$$

$$W_p = \frac{1}{1 + \frac{P_b}{N_p P_l}}, \quad (9)$$

where N_p would be the expected number of photons to be merged per lookup. This could be based on the number of photons shot for each pixel at each pass, but to keep the mix constant across passes we hardcode $N_p = 16$ as a magic constant. Higher values of N_p favor photon mapping. Naturally, this weighting only applies when shading a light and an object tagged for caustics.

Despite our mix not being as precise as the true weight from path probabilities would be, it does result in plausible renders as shown in Figure 16. The alternative is to just ignore the regular lighting through caustic and smooth objects' indirect, but this would make these lights invisible through refraction.

7.2. Limitations derived from MIS

With the above simplification of MIS we are limiting the use of photon mapping and the number of effects we can render. The main consequences are:

- Photons have to stop at the first non-caustic surface they hit. Their benefit ends there and regular path tracing will be responsible for further bounces. We lookup photons for secondary rays to get indirect lighting from caustics, in particular SDS paths. This appears to achieve good results and we have not found a motivation from production to address this limitation.
- Objects tagged as caustic cannot themselves receive caustics. While we have not encountered this case in production yet, we expect that an object with a mix of rough and smooth parts could require us to support it.

8. Results

Our technique is currently in use in the production of several animated feature films. Due to the timing of the film releases, we were unable to obtain production images to incorporate in this document. The most common use of caustics revolve around water simulations as in Figure 17, either underwater or reflected. The render times are short for surface caustics, ranging from 5 to 20 core-hours.

Volumetric caustics, on the other hand can be more costly. The range extends up to 30 – 40 core-hours, and also have higher memory usage due to the photon beams taking up to 4GB. Thankfully our bias-as-noise trick lets us rely on image space denoising to obtain visually stable results even on renders that are not fully converged. In comparison to our previous implementation of Metropolized VCM, these render times have been much more acceptable to artists which have embraced the effect and started to enable it in more and more scenes.

From a usability standpoint, the two clicks required to enable caustics in our renderer have been acceptable. While a global solution might be more elegant, we feel that the desire for artist control makes our more targeted approach the right one for now. It also simplifies debugging as we can be sure that stray shiny objects cannot interfere with the simulation at all.

From an engineering point of view, this new algorithm has also been much easier to fit into our renderer. The core photon mapping logic and guiding update is mostly self-contained and only interacts with the existing path tracer at well-defined points. This is in contrast

to the sprawling infrastructure required to augment the renderer with pluggable integrators, re-playable random number streams for Markov chain methods, and the extra layer of complexity in BSDF APIs required by reverse pdf evaluations. The fact that our guiding data is simply a 2D image also greatly simplifies debugging and checkpointing of computations.

9. Future work and limitations

The most pressing issue to address in our system is the high memory usage consumed by photon beams. They must be split into small segments for intersection by a BVH to be efficient. We believe an oriented bounding box hierarchy could provide significant speedups.

The learning process for the guiding map is a simple mixing based on sample count for the visibility signal. While our approach seems to work well, we believe a more principled solution likely exists. Some successful improvements have been made in the context of other guiding methods [VHH*19] by using variance to combine successive passes. However the fact that photon guiding interacts with image space adaptive sampling at the same time makes the problem more complex.

When the caustic producing objects are bigger than the visible area by several orders of magnitude, it is possible for the learning process to have difficulty bootstrapping itself. So far we have been able to work around this by narrowing the region of influence of the lights, but further improving robustness to extreme cases would be desirable. This bootstrapping problem is common to all methods that try to adapt to prior samples and is an important open problem in rendering.

Acknowledgements

We want to thank Clifford Stein for reviewing this article, the rest of the Arnold team and Imageworks for making this work possible.

References

- [BAC*18] BURLEY B., ADLER D., CHIANG M. J.-Y., DRISKILL H., HABEL R., KELLY P., KUTZ P., LI Y. K., TEECE D.: The design and evolution of disney's hyperion renderer. *ACM Transactions on Graphics* 37, 3 (Aug. 2018). doi:10.1145/3182159. 2
- [Boo86] BOOTH T. E.: A monte carlo learning/biasing experiment with intelligent random numbers. *Nuclear Science and Engineering* 92, 3 (1986), 465–481. URL: <https://doi.org/10.13182/NSE86-A17534>, arXiv:<https://doi.org/10.13182/NSE86-A17534>, doi:10.13182/NSE86-A17534. 3
- [CFS*18] CHRISTENSEN P., FONG J., SHADE J., WOOTEN W., SCHUBERT B., KENSLER A., FRIEDMAN S., KILPATRICK C., RAMSHAW C., BANNISTER M., RAYNER B., BROUILLAT J., LIANI M.: Renderman: An advanced path-tracing architecture for movie rendering. *ACM Trans. Graph.* 37, 3 (Aug. 2018). URL: <https://doi.org/10.1145/3182162>, doi:10.1145/3182162. 2
- [FHL*18] FASCIONE L., HANIK J., LEONE M., DROSKE M., SCHWARZHAUPT J., DAVIDOVIĆ T., WEIDLICH A., MENG J.: Manuka: A batch-shading architecture for spectral path tracing in movie production. *ACM Trans. Graph.* 37, 3 (Aug. 2018). URL: <https://doi.org/10.1145/3182161>, doi:10.1145/3182161. 2
- [GIF*18] GEORGIEV I., IZE T., FARNSWORTH M., MONTROYA-VOZMEDIANO R., KING A., LOMMEL B. V., JIMENEZ A., ANSON O., OGAKI S., JOHNSTON E., HERUBEL A., RUSSELL D., SERVANT F.,



Figure 17: An underwater shot where the production hero asset has been replaced by a public model. All the light comes from volumetric and surface caustics. The sampling rate varies from pixel to pixel due to the adaptive nature of our system and it goes from a minimum 16 to 256 samples per pixel. Photons are traced in half that amount. The full 2k render takes 1h23m on a 12 core workstation and consumes 3GB in photon and beam data. No denoising was applied.

- FAJARDO M.: Arnold: A brute-force production path tracer. *ACM Trans. Graph.* 37, 3 (Aug. 2018). URL: <https://doi.org/10.1145/3182160>, doi:10.1145/3182160. 2
- [GKDS12] GEORGIEV I., KRIVÁNEK J., DAVIDOVIČ T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Trans. Graph.* 31, 6 (Nov. 2012), 192:1–192:10. URL: <http://doi.acm.org/10.1145/2366145.2366211>, doi:10.1145/2366145.2366211. 2
- [GPGSK18] GRITTMANN P., PÉRARD-GAYOT A., SLUSALLEK P., KRIVÁNEK J.: Efficient caustic rendering with lightweight photon mapping. *Comput. Graph. Forum* 37 (2018), 133–142. 3
- [HDF15] HANIKA J., DROSKE M., FASCIONE L.: Manifold next event estimation. *Comput. Graph. Forum* 34, 4 (July 2015), 87–97. 1, 2
- [HJ11] HACHISUKA T., JENSEN H. W.: Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph.* 30, 5 (Oct. 2011). URL: <https://doi.org/10.1145/2019627.2019633>, doi:10.1145/2019627.2019633. 2
- [HOJ08] HACHISUKA T., OGAKI S., JENSEN H. W.: Progressive photon mapping. *ACM Trans. Graph.* 27, 5 (Dec. 2008), 130:1–130:8. URL: <http://doi.acm.org/10.1145/1409060.1409083>, doi:10.1145/1409060.1409083. 2
- [HPJ12] HACHISUKA T., PANTALEONI J., JENSEN H. W.: A path space extension for robust light transport simulation. *ACM Trans. Graph.* 31, 6 (Nov. 2012). URL: <https://doi.org/10.1145/2366145.2366210>, doi:10.1145/2366145.2366210. 2
- [HZE*19] HERHOLZ S., ZHAO Y., ELEK O., NOWROUZEZAHRAI D., LENSCH H. P. A., KRIVÁNEK J.: Volume path guiding based on zero-variance random walk theory. *ACM Trans. Graph.* 38, 3 (June 2019), 25:1–25:19. URL: <http://doi.acm.org/10.1145/3230635>, doi:10.1145/3230635. 2
- [Jen96] JENSEN H. W.: Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96* (Berlin, Heidelberg, 1996), Eurographics, Springer-Verlag, p. 21–30. 2, 6
- [JNSJ11] JAROSZ W., NOWROUZEZAHRAI D., SADEGHI I., JENSEN H. W.: A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics (Presented at SIGGRAPH)* 30, 1 (Jan. 2011), 5:1–5:19. doi:10/fcdh2f. 2, 7
- [KCSG18] KULLA C., CONTY A., STEIN C., GRITZ L.: Sony pictures imageworks arnold. *ACM Trans. Graph.* 37, 3 (Aug. 2018). URL: <https://doi.org/10.1145/3180495>, doi:10.1145/3180495. 2
- [KGH*14] KRIVÁNEK J., GEORGIEV I., HACHISUKA T., VÉVODA P., ŠIK M., NOWROUZEZAHRAI D., JAROSZ W.: Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 33, 4 (July 2014). doi:10/f6cz72. 2
- [MGN17] MÜLLER T., GROSS M., NOVÁK J.: Practical path guiding for efficient light-transport simulation. *Computer Graphics Forum* 36, 4 (June 2017), 91–100. URL: <http://dx.doi.org/10.1111/cgf.13227>, doi:10.1111/cgf.13227. 2, 6
- [Pha18] PHARR M.: Guest editor's introduction: Special issue on production rendering. *ACM Trans. Graph.* 37, 3 (July 2018). URL: <https://doi.org/10.1145/3212511>, doi:10.1145/3212511. 1
- [SHJD18] SIMON F., HANIKA J., JUNG A., DACHSBACHER C.: Selective guided sampling with complete light transport paths. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 37, 6 (Dec. 2018). doi:10.1145/3272127.3275030. 2
- [ŠK19] ŠIK M., KRIVÁNEK J.: Implementing one-click caustics in corona renderer. In *Eurographics Symposium on Rendering - DL-only and Industry Track* (2019), Boubekeur T., Sen P., (Eds.), The Eurographics Association, pp. 61–67. doi:10.2312/sr.20191221. 1, 2
- [VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1995), SIGGRAPH '95, Association for Computing Machinery, p. 419–428. URL: <https://doi.org/10.1145/218380.218498>, doi:10.1145/218380.218498. 2
- [VHH*19] VORBA J., HANIKA J., HERHOLZ S., MÜLLER T., KRIVÁNEK J., KELLER A.: Path guiding in production. In *ACM SIGGRAPH 2019 Courses* (New York, NY, USA, 2019), SIGGRAPH '19, ACM, pp. 18:1–18:77. URL: <http://doi.acm.org/10.1145/3305366.3328091>, doi:10.1145/3305366.3328091. 2, 9
- [vOHK16] ŠIK M., OTSU H., HACHISUKA T., KRIVÁNEK J.: Robust light transport simulation via metropolised bidirectional estimators. *ACM Trans. Graph.* 35, 6 (Nov. 2016), 245:1–245:12. URL: <http://doi.acm.org/10.1145/2980179.2982411>, doi:10.1145/2980179.2982411. 2

Appendix A: Implicit kernel from random beam width

Randomizing the beam width $w_r = w_b \sqrt{\xi}$ determines the apparent kernel in the resulting picture. We are assuming $w_p = 0$ for clarity in

the equations. After randomization a simple explicit uniform kernel $1/w_r$ is used, which would render a sharp ribbon without smoothing. But the expectance of the contribution $f(b, h)$ for a given ray passing by a beam b at a particular offset $h < w$ from the axis is:

$$E[f(b, h)] = \int_h^{w_b} \frac{\text{pdf}(w_r)}{w_r} dw_r, \quad (10)$$

where $1/w_r$ comes for our explicit kernel and w is the original width of the beam. For $w_r = w_b \sqrt{\xi}$ the PDF is $2w_r^2/w_b^2$ and the resulting expectance is:

$$E[f(b, h)] = \frac{2}{w_b^2} \int_h^{w_b} dw_r = \frac{2}{w_b^2} (w_b - w_r), \quad (11)$$

which is a triangle kernel. If, instead, we were to use uniform width sampling $w_r = w_b \xi$, the resulting implicit kernel would be:

$$E[f(b, h)] = \frac{1}{w_b} \int_h^{w_b} \frac{1}{w_r} dw_r = \frac{1}{w_b} \log\left(\frac{w_b}{h}\right)$$

which has a singularity at the center and is sharper as shown in Figure 18. We found the triangle kernel gave more visually pleasing results.

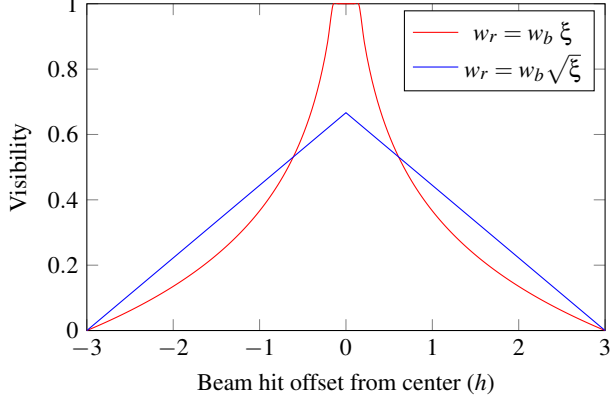


Figure 18: Implicit kernel from uniform beam width sampling and square root version (linear).