Scalable Virtual Ray Lights Rendering for Participating Media

N. Vibert[1]   A. Gruson[1]   H. Stokholm[2]   T Mortensen[3]   W. Jarosz[4]   T. Hachisuka[5]   D. Nowrouzezahrai[1]

[1]McGill University   [2]Luxion   [3]VIA University College   [4]Dartmouth College   [5]The University of Tokyo

Abstract
Virtual ray lights (VRL) are a powerful representation for multiple-scattered light transport in volumetric participating media. While efficient Monte Carlo estimators can importance sample the contribution of a VRL along an entire sensor subpath, render time still scales linearly in the number of VRLs. We present a new scalable hierarchial VRL method that preferentially samples VRLs according to their image contribution. Similar to Lightcuts-based approaches, we derive a tight upper bound on the potential contribution of a VRL that is efficient to compute. Our bound takes into account the sampling probability densities used when estimating VRL contribution. Ours is the first such upper bound formulation, leading to an efficient and scalable rendering technique with only a few intuitive user parameters. We benchmark our approach in scenes with many VRLs, demonstrating improved scalability compared to existing state-of-the-art techniques.

1. Introduction

Simulating the effects of multiple scattered light in volumetric participating media remains an important problem in realistic image synthesis, as this type of transport is notoriously costly to compute. The challenge compared to, e.g., surface-based indirect illumination, is due in large part to the additional integration dimension over (potentially scattered) light path segments in the volume. As such, several specialized techniques have been developed to render multiple scattering in participating media [BJ17; GKH*13; JNT*11; JZJ08; KGH*14; NNDJ12b].

Many light-based rendering methods rely on a two-step approach to efficiently compute indirect lighting: after tracing light subpaths, starting from emitters and scattering off surfaces or in media, the image contribution is computed by treating these subpath vertices as secondary *virtual point light* (VPL) sources at camera subpath vertices [Kel97]. Such a VPL representation suffers from singularities in its shading formulation, and approaches to reduce these artifacts include clamping and compensation schemes [KK06], approximate bias compensation [ENSD12; NED11; SGH18], spatial regularization of the VPLs (i.e., into *virtual spherical lights*; VSLs) [HKWB09], and attenuation by simply increasing the total number of VPLs.

While increasing the number of VPLs is simple and does not introduce any additional bias, naïvely doing so results in a linear growth in the final rendering cost. Scalable techniques achieve sublinear cost using either hierarchies over the virtual lights [WABG06; WFA*05; WKB12] or low-rank transport matrix completion and sampling [BMB15; HPB07; OP11].

Alternatively, higher-order geometric primitives for the virtual sources have been proposed to address this limitation. Virtual ray lights (VRL; [NNDJ12b]) are derived by "sweeping" virtual points along light subpath edges, increasing the density of secondary lighting in volumes and mathematically reducing the degree of the singularities during shading. Higher-order (hyper-)planar virtual lighting primitives have also been proposed [BJ17], which can be interpreted as a special way to gather VRLs. However, these approaches only handle volume-to-volume computations and do not allow much freedom to perform adaptive sampling.

We propose a method for efficiently rendering multiple scattering, combining the effectiveness of VRL primitives with the scalability of hierarchical light pruning. Motivated by Lightcuts [WFA*05], our scalable solution adapts to variations in illumination complexity. In order to adaptively prune VRLs to an essential subset for shading, we derive a novel upper bound of the maximum contribution of a VRL group. This bound differs significantly from those derived in standard Lightcuts-based methods, since shading with VRLs relies on a more complicated Monte Carlo estimation over eye subpath segments. Our upper bound models this dependence in order to more tightly limit VRL-group contributions (i.e., compared to more naïve bounds), and we demonstrate that the final scalability (and, so, rendering performance) depends fundamentally on the effectiveness of this bound. We reduce equal-quality render times significantly compared to existing baselines, i.e., standard VRL shading and scalable methods that rely on VPLs. Concretely, our contributions are:

- a novel upper bound derivation for VRL cluster contributions,
- a hierarchical method that leverages this bound to efficiently and adaptively select important VRLs, and
- a simple and efficient rendering method implemented atop an existing VRL renderer.

2. Related work

Many methods study efficient rendering in participating media and we refer readers to comprehensive surveys [NGHJ18], focusing on work that is most relevant to our contributions below.

Many-light methods for media. Efficient many-light methods based on instant radiosity (or *virtual point light*; VPLs) exploit light subpath reuse, where extensions to participating media deposit VPLs spatially in the media before accumulating their contributions over eye subpaths. Here, again, we refer readers to a more thorough state-of-the-art report on this deeply investigated area [DKH*14].

VPLs-based methods can also be interpreted as a special case of bidirectional estimators [VG95] that consider only one sampling technique at the cost of introducing singularities during VPL contribution calculations. In practice, the singularity is mitigated by clamping VPL contributions, albeit at the cost of energy loss (i.e., bias). To eliminate this bias, many approaches aim to compensate for the lost energy, often with a secondary unbiased estimator [KK06] or approximation [ENSD12]. Alternatively, virtual sphere lights (VSLs) regularize the singularity by blurring its contribution in space (and/or angles) [HKWB09].

Virtual ray/beam lights [JNT*11; JZJ08; NNDJ12b] apply virtual lighting in participating media with photon beam-like primitives. Here, light subpath edges are interpreted as virtual line lights which, when compared to VPLs, more effectively model the distribution of volumetric distributions of light. Virtual ray lights (VRLs) also reduce the order of shading singularities, thanks in part to an effective importance sampling scheme, but still suffer from a (lower-order) singularity in volume-to-volume transport. Here, visual artifacts are most pronounced as VRLs become more parallel to eye subpath ray segments. The degree of the singularity can also increase for medium-to-surface transport. Virtual Beam Lights (VBLs) generalize the application of spatial kernels from VSLs to VRLs [NNDJ12a], inflating the VRL geometry to a "thicker" line segment, resulting in the elimination of singularities in the contributions, albeit at cost of added bias. We build atop VRLs with a more efficient, scalable rendering solution for participating media. Our method can readily be applied to VBLs by incorporating the additional multiple importance sampling process used when computing shading contributions. We leave this extension to future work as we found the order of complexity gains of our scalable hierarchy outweigh the constant factor cost reduction benefits of moving to VBLs.

Scalable methods. The seminal Lightcuts approach [WFA*05] pioneered a scalable solution for virtual lighting, using a hierarchical clustering of VPLs that can be adaptively refined for each shading query, based on conservative perceptual error bounds: cuts in the VPL tree hierarchy are chosen so as to control error in the final image. This technique is well suited to the local variations in illumination complexity over an image. A multi-dimensional extension of Lightcuts [WABG06] improves performance in shading scenarios where additional distribution integral effects are employed (e.g., participating media, motion blur, depth of field). Here, a Cartesian product of hierarchical gather trees is traversed to form an appropriate multi-dimensional cuts. Note that VRLs with Lightcuts could be interpreted as a form of Multi-dimensional Lightcuts – VRLs implicitly model an infinite collection of media VPLs. Bidirectional Lightcuts [WKB12] use multi-bounce and distributed shading points to treat a variety of shading effect. This formulation relies on point primitives and does not support more complicated queries, such as those necessary for line integration.

Matrix row and column sampling (MRCS) [HPB07] formulates the many light problem as a matrix completion and sampling problem. This technique retains a global set of VPLs and can fail to capture local lighting variation. To better adapt to these effects, Lightslice [OP11] adds a refinement step based on clustered camera queries. Huo et al. [HWJ*15] improve the performance of these techniques by relying on matrix predictors and matrix factorization.

IlluminationCut [BMB15] combines the matrix and lightcut approaches by constructing a global gather point tree. This approach is more efficient than Multi-dimensional Lightcuts which constructs an independent gather point tree at each pixel. Moreover, a revised upper bound is used to refine gather point nodes. This approximate bound, combined with tailored heuristics, are used to control the error introduced by terminating the refinement of a light or gather point tree.

Some previous work have explored a combination of VRLs with scalable techniques [FBD15; HWH*16], but are motivated to avoid Lightcuts-based approaches due to the difficulty of deriving sufficiently tight bounds on the VRL contributions. Instead, these approaches rely on manual user parameters to control the error. We derive such a bound for a cluster of VRLs that is efficient to evaluate, allowing us to refine shading estimates automatically. Our approach does not rely on hand-tuned user parameters nor heuristics, and so is simple to implement atop existing VRL renderers.

3. Background

Radiative transport. The radiance arriving at a location x from direction $\vec{\omega}$ can be expressed as the sum of two terms,

$$L(x,\vec{\omega}) = T_r(s)L(x_s,\vec{\omega}) + L_m(x,\vec{\omega}) \tag{1}$$

where the surface radiance $L(x_s,\vec{\omega})$ from a location x_s attenuated by transmittance T_r, and L_m is in-scattering due to participating media. The L_m term can be costly to evaluate as it involves computing the integral of the radiance from every point between x and x_s:

$$L_m(x,\vec{\omega}) = \int_0^s T_r(u)L_i(x_u,\vec{\omega})\,\mathrm{d}u \tag{2}$$

where $s = |x - x_s|$ and L_i is the scattered radiance from the point $x_u = x_s + \vec{\omega} \cdot u$ along the ray in direction $\vec{\omega}$:

$$L_i(x_u,\vec{\omega}) = \int_{\mathcal{S}^2} f(\vec{\omega},\vec{\omega}')L(x_u,\vec{\omega}')\mathrm{d}\vec{\omega}' \tag{3}$$

and f is the phase function pre-multiplied with the scattering coefficient. For brevity, we will focus exclusively on estimating Equation 2 with many-light techniques.

Virtual point lights. One approach to estimate transport in the medium (Equation 2) relies on sampling a distance u along a sensor ray (according to probability density $p(u)$, i.e., proportional to the transmittance) before estimating the incoming radiance L_i at this position by deterministically gathering the contribution from VPLs deposited using light-tracing [Kel97] (Figure 1a).

$$L_m(x,\vec{\omega}) = \mathrm{E}\left[\sum_k \frac{I_k f(x_u,x_k)\,T_r(w_k)\,V(x_u,x_k)}{w_k^2\,p(u)}\right] = \mathrm{E}\left[\sum_k L_{m,k}^{\mathrm{VPL}}\right], \tag{4}$$

where I_k, x_k and $L_{m,k}^{\mathrm{VPL}}$ are the kth VPL's intensity, position and contribution, $f(x_u,x_k)$ is the phase function if x_k lies in the volume
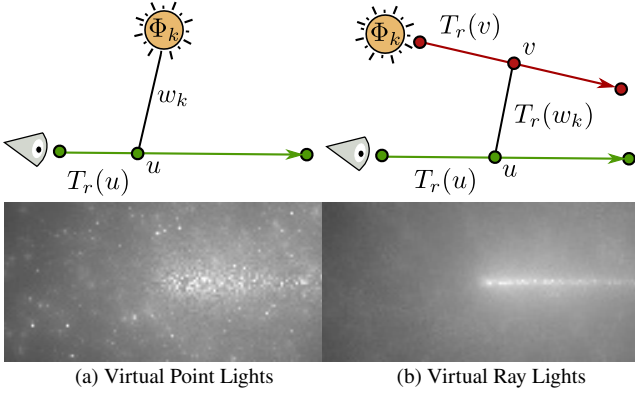
(a) Virtual Point Lights   (b) Virtual Ray Lights

Figure 1: *Compared to VPL, VRL integrates an additional dimensionality over the light subpath segment. Using this extra dimension leads to an excellent reduction in noise in the final result at equal render time.*

(otherwise it is the cosine-weighted BSDF at the surface point x_k), $w_k = \|x_u - x_k\|$ is the distance between the VPL and the gather point location x_u, and $V(x_u, x_k)$ is the binary line-of-sight visibility.

Note that the distance u is sampled independently from the VPL position, which has the effect of increasing the variance of the final rendering estimator (Figure 1). Another source of variance compared to VRL estimates is due to the inverse squared distance in the denominator of the VPL contribution (Equation 4). This term can lead to arbitarily large VPL contributions as w_k tends to 0.

Virtual ray lights. To address these issues, Novák et al. [NNDJ12b] proposed to compute the contribution over entire sensor ray and light path segments, the latter of which are represented with VRLs, as an direct estimate of the underlying double integral (Figure 1). Formally, given a camera ray of length s, we can evaluate the medium contribution (Equation 2) by gathering over many VRL contributions, as with VPLs. The individual contribution of the kth VRL is

$$L_{m,k}^{\text{VRL}} = \Phi_k \int_0^s \int_0^{t_k} \frac{f_k(u,v) T_r(u) T_r(v) T_r(w_k(u,v)) V(u,v)}{w_k(u,v)^2} \, dv \, du,$$

where Φ_k and t_k are the kth VRL's flux and segment length, $w_k(u,v)$ is the distance between pairs of sensor (u) and VRL (v) points, and f_k is the product of scattering terms. Moving forward, we omit explicit (u,v) parameters, for brevity. Note that $E\left[\sum_k L_{m,k}^{\text{VRL}}\right]$ is equal to $L_m(x,\vec{\omega})$.

The double integral in the equation above is evaluated with a Monte Carlo estimator where u and v are sampled according to $p(u,v)$ for each kth VRL:

$$\bar{L}_{m,k}^{\text{VRL}} = \frac{\Phi_k f_k(u,v) T_r(u) T_r(v) T_r(w_k(u,v))}{w_k(u,v)^2 \, p(u,v)} \quad (5)$$

where $E\left[\bar{L}_{m,k}^{\text{VRL}}\right]$ is equal to $L_{m,k}^{\text{VRL}}$ and $E\left[\sum_k \bar{L}_{m,k}^{\text{VRL}}\right]$ is equal to $L_m(x,\vec{\omega})$. Novák et al. [NNDJ12b] proposed an importance sampling scheme that leverages a 2D analytic probability density function (pdf) to jointly sample VRL and sensor points. This pdf

$p(u,v) \propto w_k(u,v)^{-2}$ eliminates the variance due to the geometric term in the integrand of the above equation. The remainder of our document will focus on deriving and analyzing an error bound for VRLs and VRL-cluster contributions, for isotropic phase function. As in Novák et al.'s work [NNDJ12b], we also represent anisotropic phase functions with a piece-wise approximation when constructing $p(u,v) \propto w_k(u,v)^{-2} f(u,v)$; however, the piecewise approximative nature of this pdf precludes it from completely eliminating variance from these terms. We explain later how to handle this scenario when deriving our upper-bound on VRL-cluster contributions.

4. Scalable Virtual Ray Lights

For each pixel, we wish to automatically pick a subset of the most important VRLs to use for shading, according to their relative image contributions, from a much larger set of VRLs deposited during light tracing. Motivated by Lightcuts [WFA*05], we first build a hierarchy over VRLs where a node represents a group of spatially-coherent VRLs. We traverse this hierarchy per-pixel when shading, selecting a set of nodes (referred to as a *cut*) through the hierarchy that is guaranteed to maintain a desired error threshold.

When traversing the hierarchy, we decide to terminate our traversal based on an estimate of the maximum image error we can expect to incur if we skip the associated subtree. This estimate must model the expected contribution for *all* the VRLs in the node, and it must be efficient to evaluate. Moreover, this error estimate is necessarily conservative since the precise distribution of VRLs in a node is unknown (unless we traverse to the leaves, defeating the purpose of the hiearchy). As such, the performance of any such hierarchical cut-based method relies on avoiding unnecessary subtree refinement or, equivalently, on having upper bound for the error estimate that is as tight as possible.

After a quick overview of so-called *long* vs. *short* VRLs, we describe the construction of our VRL light tree before motivating and deriving an error bound that takes into account the stochastic nature of VRL contribution evaluation. Indeed, we need to express the bound B on the following Monte Carlo estimator:

$$\left(\sum_{k \in \mathcal{C}} \bar{L}_{m,k}^{\text{VRL}}\right) = \left(\sum_{k \in \mathcal{C}} \frac{\Phi_k f_k(u,v) T_r(u) T_r(v) T_r(w_k(u,v))}{w_k(u,v)^2 \, p(u,v)}\right) < B \quad (6)$$

where k indexes VRLs in a cluster $\mathcal{C}$ (i.e., node in the hierarchy) and points (u,v) are drawn proportional to the joint density $p(u,v)$.

Long and short VRLs. Clustering line segments (VRLs) is more challenging than clustering points (VPLs), due to their extent and directionality. The expected length for a given VRL depends primarily on the optical thickness of the medium and how its transmittance is evaluated during light tracing. An *expected transmittance* formulation leads to so-called long VRLs that extend to the boundaries of the scene/media. It is more challenging to hierarchically cluster longer beams according to spatial and/or angular similarity. Another disadvantage of long VRLs is that their exponential intensity fall-off complicates the evaluation of their contribution.

Alternatively, a *null collision transmittance* formulation leads to so-called short VRL segments typically (much) shorter that long VRLs, and whose intensities are constant. Short VRLs are easier to

cluster and we rely on this formulation when constructing our VRL light tree. Note that, since we only cluster light subpath segments (VRLs) and not eye subpath segments, we opt to use the dual of long camera rays to shade with our short VRLs.

Building the light tree. We construct our light tree by assigning a node to each VRL and merging them with a bottom-up approach. Each node contains an axis-aligned bounding box and a representative VRL. The latter is chosen randomly from the two children nodes that get merged. We compute the resulting cluster flux Φ_C after merging children nodes as:

$$\Phi_C = \Phi_A + \left(|B_{rep}|/|A_{rep}| \right) \Phi_b, \qquad (7)$$

where Φ_A and $|A_{rep}|$ are the flux and length of the representative VRL in cluster A, and this formulation ensures energy conservation.

To be of practical use, light tree construction must be efficient and must also produce a tree that (ideally) minimizes the number of cuts performend during refinement. Building a light tree for VRLs is much more challenging than for VPLs due to the increased likelihood of overlapping bounding boxes. We initially apply a fast agglomerative clustering approach [WBKP08] that builds an initial kd-tree to bootstrap light tree construction. We found that this approach works well at the beginning of the merging operations, when bounding boxes do not overlap much; however, as we progress through construction and merging, the algorithm unfortunately degenerates into an $O(n^2)$ search over all pairs of nodes in the kd-tree. Indeed, the increased bounding box overlaps offset the benefits of the guided kd-tree when accelerating the nearest neighbor search. Moreover, it is challenging to parallelize this agglomerative approach, motivating our alternative approach.

We instead rely on a fast, readily parallelizable hybrid tree-building technique that first aggressively partitions the space by sorting VRLs along all principal directions before selecting according to a median split. We greedily choose the split that minimizes the bounding box overlap in the resulting children nodes, recursing with this subdivision scheme until we reach a target number of VRLs in each partition (2048 in our implementation). After partitioning, we build a local light tree in each partition using the heap-based agglomerative clustering [WBKP08] that minimizes the metric $\Phi_c E_c^2$, where Φ_c is the resulting cluster flux and E_c is the AABB extent. Once all the local light trees are built, we re-optimize the main tree using the same building approach.

Our scheme is capable of building a well-balanced VRL tree for 10^6 VRLs in the order of seconds. We observe experimentally that our construction strategy generates trees that have a gathering overhead of 25% compared to a brute-force optimized tree. We believe that this tree construction strategy reaches a good trade-off between the construction time and the resulting cluster structure. We could further improve this structure by, e.g., re-optimizing the local trees using the Bonzai approach [GBDA15] while the main tree is refined, but leave this exploration for future work.

Upper bound derivation. Given a node in our tree, we need to evaluate the error bound in order to decide whether to stop the refinement process at shade-time. Such an upper bound should model a worst-case scenario, since iterating over the exact distribution of VRLs would be too costly. As this bound is conservative, we allow the user to control the maximum allowable integration error over any single camera query.

A naïve derivation of the upper bound would proceed by bounding the individual terms in the VRL estimate (Equation 6). This approach, however, does not consider that variance due to many of these terms are compensated for during importance sampling for VRL shading. Of particular interest is the case of isotropic phase function where we can derive a particularly tight upper bound by considering multiple terms at once:

$$\frac{1}{w(u,v)^2 p(u,v)} = \frac{1}{w(u,v)^2 p(u|v) p(v)} < B, \qquad (8)$$

where $p(v)$ is the marginalized pdf of sampling a particular position on the VRL and $p(u|v)$ is the conditional pdf of sampling a point on the sensor ray given that we first sampled v. As we consider the worst-case scenario, i.e. when the ray query intersects the node's bounding box, we will force the refinement of this node. Indeed, as the distribution of VRL is unknown, we can potentially have an infinite amount of variance for our Monte Carlo estimate. Note that this infinite variance scenario occurs only when the ray query perfectly overlaps a VRL's geometry.

When the camera ray does not intersect the node's bounding box, we can derive a meaningful bound for B. The worst case for $p(v)$ is when the VRL and ray query are coplanar: here, $p(v)$ degenerates to uniform sampling over the VRL and we therefore have $p(v) < 1/L_{max}$, where L_{max} is the maximum VRL length inside the node.

For the remaining terms $p(u|v)w(u,v)^2$, we perform a change of variables (as shown in Figure 6), where $\hat{h}$ is the shortest distance between the potentially sampled VRL point v and the sensor ray, and $\hat{u}$ is the distance of the projected point onto the sensor segment. With this reparameterization [KF11] , we can now write:

$$p(u|v)w(u,v)^2 = \frac{h(h^2+u^2)}{(\theta_A-\theta_B)(h^2+u^2)} = \frac{h}{(\theta_A-\theta_B)}, \qquad (9)$$

where θ_A and θ_B are the maximum opening angles for the segment extremities. To minimize this equation, we can set:

$$h/(\theta_A-\theta_B) > h_{min}/\theta_{max}, \qquad (10)$$

where h_{min} is the minimum distance between the node bounding box and the query ray, and θ_{max} is the maximum opening angle. The opening angle is maximized when the projected point v is on the middle of the sensor segment and when v is at minimum distance. This point might be out of the node's bounding box, but considering the most general case in 3D is nontrivial. With these assumptions, we can derive a maximum angle $\theta_{max} = 2 \cdot \tan^{-1}(2 \cdot h_{min}/s)$, where s is the total length of the sensor ray. Note that this θ_{max} approaches π for an infinitely long camera ray.

Other terms in Equation 6 can be bounded individually similar to Lightcuts. For example, we assume the visibility term $V = 1$ as it is difficult and costly to ensure that the whole node bounding box cannot be visible at all possible locations on the sensor ray. As we use short VRLs, the VRL transmittance cancel-out. As any point on the sensor ray can be sampled, we use a conservative bound of 1 for its transmittance. Finally, in the case of an isotropic phase
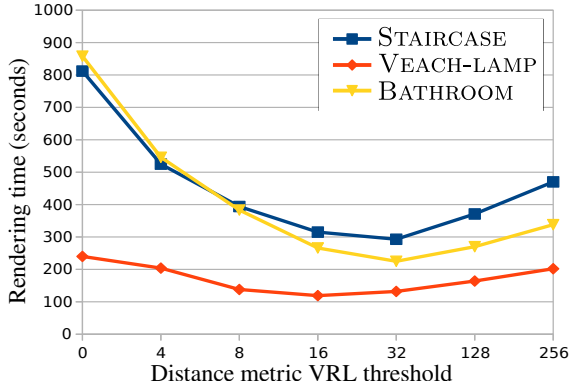
Figure 2: *We explore the trade-off between the cost of computing accurate minimum distances and the render time. Using more VRLs to compute accurate minimal distances is beneficial for nodes with few VRLs as this favors avoiding refinement for these nodes. This experiment uses* 100k *VRLs for each scene to compare scenarios with similar tree node counts.*

function, we can evaluate the phase function terms directly as they are directionally invariant.

To compute the bound of the transmittance $T_r(w)$, we use the minimal distance $h_{\min}$ as we are in the homogeneous participating media context. This bound is only valid for convex media shapes. For concave media shapes, we bound its value to 1. The final formula for our bound is:

$$B = \frac{\Phi_k\, f_k(u,v)\, T_r(h_{\min})\, \theta_{\max}\, L_{\max}}{h_{\min}} \qquad (11)$$

where Φ is the total flux of all the VRLs inside the cluster.

5. Results

We implemented our VRL Lightcuts method in Mitsuba [Jak13]. All our results were generated on an Intel E5-2683 v4 CPU at 2.10 GHz with 32-cores and 32 GB of memory. We only compute medium-to-medium light transport to facilitate comparisons between the different methods, and we disable antialiasing by forcing camera ray generation through pixel centers.

We benchmarked our technique (VRL LC) against standard VRLs [NNDJ12b] and against an optimized Lightcuts implementation with VPLs (VPL LC) [WFA*05]. We opted to not clamp VPL contribution, and we avoid excessive weighting of singularities inside the metric using a symmetric mean absolute percentage error given by $\text{SMAPE} = (2/n)\sum_j |R_j - I_j|/(R_j + I_j + \varepsilon)$, where n is the number of pixels, R_j is the reference and I_j is the rendered pixel luminance value with $\varepsilon = 0.01$. We also used the root mean square error (RMSE) as an alternative metric to assess numerical accuracy. The maximum path length is 12 for all the scenes. We compute the reference using VRLs without Lightcuts, requiring many hours of computation.

To collect VPLs along an eye ray, we use stratified sampling

with 32 samples with its distribution proportional to the transmittance. We ensure that this sampling always succeeds on the eye ray by correctly normalizing the distribution. We use the same light tree construction algorithm for VPLs and VRLs as we observed no computational overhead compared to Walter et al.'s [WBKP08] approach on VPLs. A representative VPL/VRL is selected similarly to the original Lightcut paper: we randomly choose one of the child representatives based on their total flux. Note that, for VRLs, the representative flux is scaled by the length. Our refinement strategy is the same as Lightcuts: we track keep track of a node that produces the highest error in a heap, and continue to refine it until the node with the highest error falls below a threshold. Our threshold is based on the current VPL/VRL contribution estimate, scaled by a user parameter.

Improving the minimal distance. Computing the minimal distance based solely on bounding boxes can yield an overly conservative minimal value. To obtain a more precise minimal distance, we iterate over all of the node's VRLs after a certain number of VRLs are reached in a node. Figure 2 shows how rendering time varies when we use different threshold values. Having a more accurate minimal distance at an higher computational cost prevents an unnecessary refinement of the nodes. Note that this accurate minimal distance also works when we intersect a node. The derivation of the bound remains unchanged when using a more accurate minimal distance. As this parameter is not very scene sensitive, we used a threshold value of 32 across all scenes.

Equal-time comparison. We used three different scenes: STAIR-CASE, VEACH-LAMP and BATHROOM to benchmark our technique. Every scene has an isotropic phase function. Figure 3 summarizes the equal-time comparison between our technique (VRL LC), VRL [NNDJ12b] and hierarchical VPL (VPL LC).

STAIRCASE scene (Figure 3, first row) contains a unique area light placed at the top of the stairs. As the smoke is dense, the VRL density decreases vertically towards the floor. As this scene does not contain sharp volumetric lighting, VPL-based techniques can render this scene efficiently. However, as our technique does not exhibit large singularities, it results in the smallest RMSE value.

VEACH-LAMP scene (Figure 3, middle row) contains two spot-lights that generate a sharp lighting effect. These lighting effects are difficult to capture with VPL-based techniques as the sampling on the sensor ray is independent of the light distribution. VRLs can capture this change of distribution effectively. Thanks to our scalable technique, we can process a large number of VRLs and output a more converged image compared to using VRLs without our scalable technique.

BATHROOM scene (Figure 3, last row) has two spotlights that concentrate the lighting but also generate light directly towards the sensor. These types of light transport paths are difficult to handle by VRL-based techniques due to higher variance. Our technique generates images that are free from any singularities (best RMSE score), but the VPL-based approach gets a more accurate light distribution (best SMAPE score).

Speedup. Figure 4 shows the speedup difference between the scalable techniques for VPLs/VRLs. Our technique's speedup in-
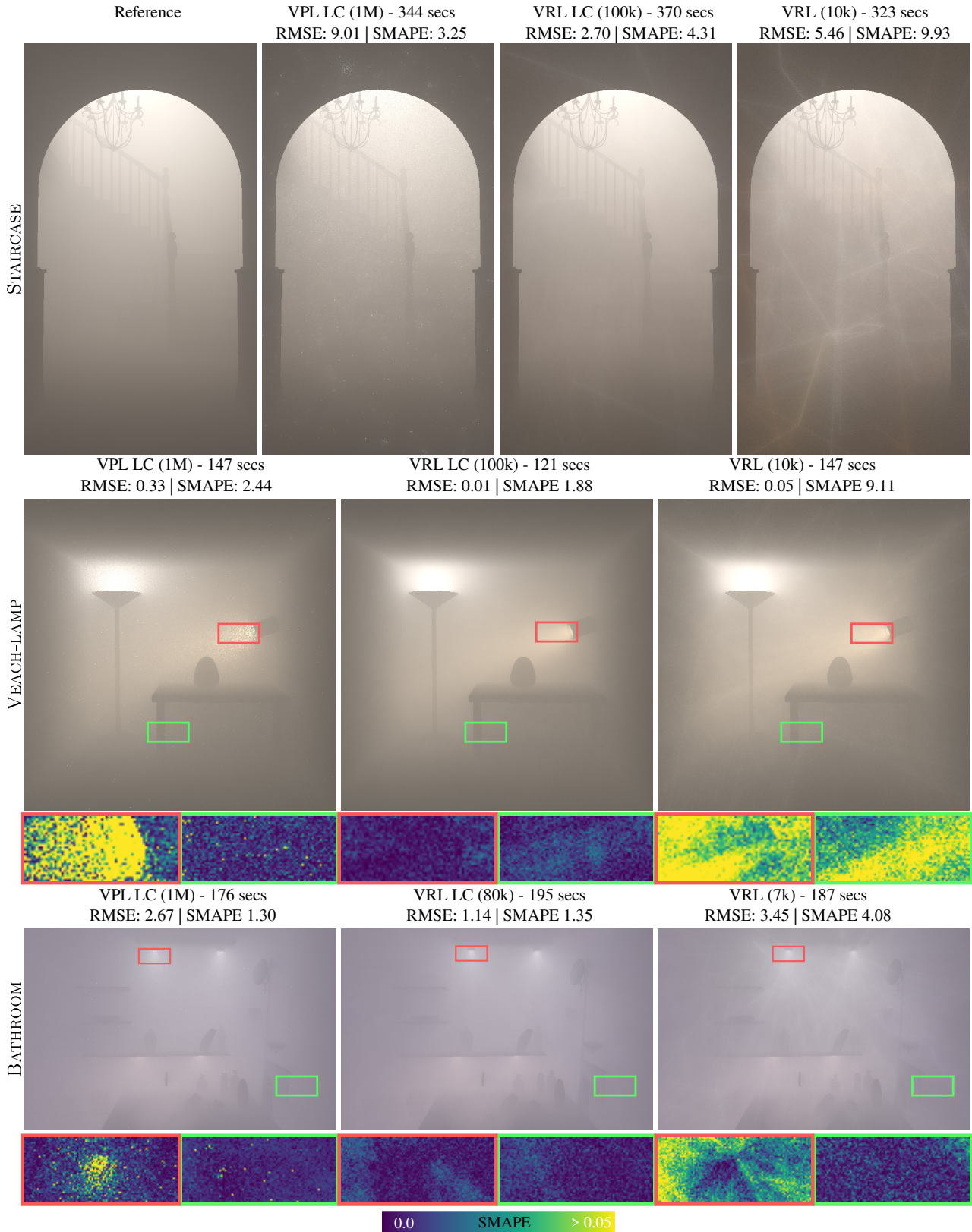
Figure 3: *Equal-time comparison between our technique, VRL without lightcut, and VPL with lightcut. Our technique achieves the smoothest results as the VPL technique cannot properly sample a point along each sensor ray while considering the contribution from VPLs. The metric values are scaled by* 100.
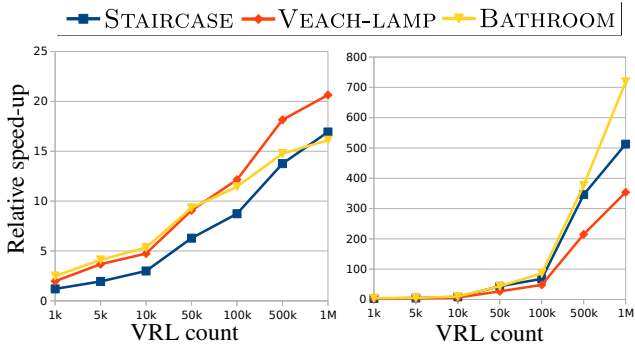
Figure 4: *The different speedup factor when changing the number of VRL or VPL with lightcut approach. Note that our technique does not scale very well due to the large number of intersection within the different nodes.*

VRL (20k) - 1203 sec
SMAPE = 9.54×10^{-2}

VRL LC (500k) - 1291 sec
SMAPE = 3.67×10^{-2}

Figure 5: *Comparison between VRL and our approach. In the anisotropic case, the error is also lower as we can process more VRLs.*

creases when more VRLs are contained in the light tree, but we cannot achieve the same sublinear scaling as VPL Lightcuts.

The first reason for this lower scalability is that our technique uses the camera ray as a line segment query into the light tree while VPL-based technique instead uses a point query. Using a line segment gives lower variance on the sensor subpath but adds an overhead for intersecting more nodes within the light tree. Similarly to Lightcut, intersecting a light tree node forces us to refine this node as the error can be arbitrarily large. The more accurate distance (Figure 2) mitigates this issue but does not completely solve it.

The second reason is that VRLs span more space than VPLs. The bounding boxes for VRLs overlap more which increases the excepted number of ray intersections with the camera query. This bounding box overlapping was already observed when we applied the Walter et al. [WBKP08] construction scheme. Still, with a reasonable amount of VRL, our technique can achieve more than 10 times speedup.

We empirically validate this reasoning by visualizing evaluated and intersected nodes (Figure 6 and Table 1). We observe that these two statistics are more correlated in our method, and the number of node intersections is also larger compared to VPLs (as expected, due to ray queries and wider VRL spatial extents).

6. Future work

Medium-to-surface light transport VRL can be used to compute medium-to-surface light transport with bigger singularity compared to medium-to-medium light transport. Note that this singularity that is less with VPL-based methods and it might be possible to blur VRL contribution to remove this singularities [NNDJ12a].

Similarly to VPL-based Lightcuts methods, we should bound the BSDF term based on the VRL cluster geometry. In previous work, it exists a finite set of analytical BSDF model where this bound is derived. However, in our knowledge, it did not exists bounds for arbitrary BSDF models.

Heterogeneous or concave participating media. Querying a minimum transmittance density in order to bound the transmittance in heterogeneous media, or more general media shapes, requires an additional spatial data structure. Designing such a structure, however, poses an interesting challenge as queries need to be extremely fast. Balancing the construction and query efficiencies is not straightforward, and without such a supplemental data structure, our method needs to conservatively assume $T_r(h_{\min}) = 1$. Exploring design options and trade-offs here is left for future work.

Anisotropic phase functions. Compared to other methods i.e. based on cone clustering, supporting an anisotropic phase function is straightforward in our approach. Indeed, for VRLs, we can build a piecewise pdf $p(u,v) \propto f(u,v)w(u,v)^{-2}$ for each evaluation. Here, similar to the isotropic case when importance sampling is used to shade, the geometry factor and the phase function terms are cancelled out (and, so, they do not contribute any variance to the estimator). Our upper bound assumes isotropic scattering, and so to account for the added variance due to the piecewise phase function approximation, we multiply our bound B by a user-controlled factor $(1+\varepsilon)$ for some $\varepsilon > 0$. Note that, in Equation 11, the phase function term will corresponds to the media "color" at the scattering event as all terms other cancel out.

Figure 5 shows the early experimental results of our technique with an anisotropic phase function ($g = 0.7$). To be conservative, we use $\varepsilon = 1.0$ as the variance of VRLs can be significant in this scene as some VRLs scatter towards the sensor. With this factor, we found that the upper bound is wrong for less than 0.02% cases when we evaluate the light tree. While it is a heuristic, we observe good results using this approach.

Reducing node intersections. Constructing an cost-efficient hierarchy over line segments is not trivial due to varying orientations and lengths. We experimented replacing axis-aligned bounding boxes (AABBs) with oriented bounding boxes (OBBs) to reduce overlap. In our experiments, it turned out that OBBs do not
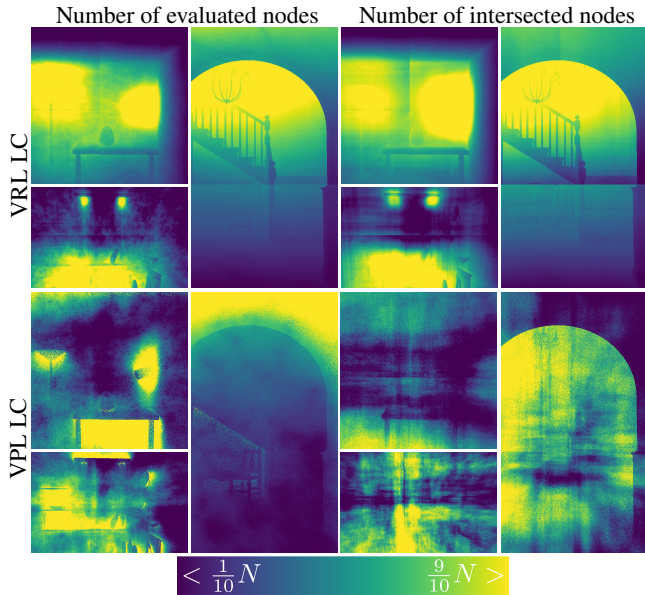
Number of evaluated nodes　　Number of intersected nodes

Figure 6: *Node evaluation and intersection statistics for VPL Lightcuts and VRL Lightcuts. False color visualizations are auto-exposed by sorting the N values and clipping them past a certain index. VRL node evaluation correlates with node intersection. VRLs generate more intersections due to their wider spatial extent and the use of sensor ray queries. We use a similar number of VRLs and VPLs as in Figure 3. Table 1 lists the associated statistics, averaged over the image.*

Table 1: *Average number of evaluated (# Eval) and intersected (# Inter) nodes per method. We observe that the number of intersected nodes is larger for VRLs than VPLs. See Figure 6 for associated false color visualizations.*

Scene	VPL Lightcuts		VRL Lightcuts	
	# Eval	# Inter	# Eval	# Inter
STAIRCASE	20.7 k	0.54 k	10.6 k	2.91 k
VEACH-LAMP	24.0 k	1.76 k	7.82 k	2.72 k
BATHROOM	16.7 k	0.87 k	7.68 k	3.16 k

a novel formulation of the upper bound of the contribution of many VRLs, solving an open problem that has precluded Lightcuts-based solutions for VRLs in the past. Furthermore, we devised an efficient VRL light-tree construction algorithm that is parallelizable and generates high-quality trees (i.e., compared to brute-force construction) in only a fraction of the time. Our rendering method is simple to integrate atop existing VRL methods, improves the rendering quality compared to VRL and hierarchical VPL techniques. At equal-time, we typically outperform VPL Lightcuts and scale much better than standard VRLs. Our renderings result in fewer visible singularities and does not requires to have clamping. We observe sub-optimal scaling compared to VPL Lightcuts, and we show that this behavior is intrinsic to the nature of VRLs: larger node overlaps lead to sub-optimal tree traversals. We are motivated by these initial results, and believe that our work paves many interesting avenues of future work.

Acknowledgement

reduce overlap over AABBs since we were not able to find an efficient and effective approach to merge OBBs. Simple and approximate merging approaches may generate OBBs larger than the associated AABB for the node. More elaborate OBB merging methods mitigate this issue, but at the cost of overhead during tree construction.

Building camera subtree. We decided to build our hierarchy only over light subpath segments, however, one can imagine building a dual structure over eye subpath segments in order to, that is, efficiently and adaptively splat VRL contributions. This approach may seem promising, as sensor subpaths are usually more coherent than light subpaths. It is not clear however how to efficiently and accurately splat VRLs across a clustered sensor node.

Multidimensional Lightcuts. Our comparison targets techniques with similar technical complexity. For this reason we did not compare to Multidimensional Lightcuts [WABG06]. Note that increasing the sampling rate on eye rays will certainly reduce variance in VPL renderings, but it will not eliminate VPL singularities. Still, extending our suite of comparisons with Multidimensional Lightcuts [WABG06] is a study we plan on pursuing.

7. Conclusion

We proposed a scalable VRL rendering approach for efficient multiple-scattering in participating media. Our method leverages

References

[BJ17] BITTERLI, B. and JAROSZ, W. "Beyond points and beams: higher-dimensional photon samples for volumetric light transport". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 36.4 (July 2017). DOI: 10/gfznbr 1.

[BMB15] BUS, N., MUSTAFA, N. H., and BIRI, V. "IlluminationCut". *Computer Graphics Forum (Proceedings of Eurographics)* 34.2 (May 1, 2015). DOI: 10/gfzv7p 1, 2.

[DKH*14] DACHSBACHER, C., KŘIVÁNEK, J., HAŠAN, M., ARBREE, A., WALTER, B., and NOVÁK, J. "Scalable realistic rendering with many-light methods". *Computer Graphics Forum* 33.1 (Feb. 1, 2014). DOI: 10/f5twgd 2.

[ENSD12] ENGELHARDT, T., NOVÁK, J., SCHMIDT, T.-W., and DACHSBACHER, C. "Approximate bias compensation for rendering scenes with heterogeneous participating media". *Computer Graphics Forum (Proceedings of Pacific Graphics)* 31.7 (Sept. 2012). DOI: 10/f4bhz5 1, 2.

[FBD15] FREDERICKX, R., BARTELS, P., and DUTRÉ, P. "Adaptive light-slice for virtual ray lights". *Proceedings of Eurographics Short Papers*. The Eurographics Association, 2015. DOI: 10/gfzq7m 2.

[GBDA15] GANESTAM, P., BARRINGER, R., DOGGETT, M., and AKENINE-MÖLLER, T. "Bonsai: rapid bounding volume hierarchy generation using mini trees". *Journal of Computer Graphics Techniques (JCGT)* 4.3 (Sept. 22, 2015). URL: http://jcgt.org/published/0004/03/02/4.

[GKH*13] GEORGIEV, I., KŘIVÁNEK, J., HACHISUKA, T., NOWROUZEZAHRAI, D., and JAROSZ, W. "Joint importance sampling of low-order volumetric scattering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 32.6 (Nov. 2013). DOI: 10/gbd5qs 1.

[HKWB09] HAŠAN, M., KŘIVÁNEK, J., WALTER, B., and BALA, K. "Virtual spherical lights for many-light rendering of glossy scenes". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 28.5 (2009). DOI: 10/bb9r45 1, 2.

[HPB07] HAŠAN, M., PELLACINI, F., and BALA, K. "Matrix row-column sampling for the many-light problem". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 26.3 (July 29, 2007). DOI: 10/djv68s 1, 2.

[HWH*16] HUO, Y., WANG, R., HU, T., HUA, W., and BAO, H. "Adaptive matrix column sampling and completion for rendering participating media". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 35.6 (Nov. 2016). DOI: 10/f9cm6s 2.

[HWJ*15] HUO, Y., WANG, R., JIN, S., LIU, X., and BAO, H. "A matrix sampling-and-recovery approach for many-lights rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 34.6 (Oct. 26, 2015). DOI: 10/f7wrx7 2.

[Jak13] JAKOB, W. *Mitsuba Renderer*. 2013. URL: http://www.mitsuba-renderer.org 5.

[JNT*11] JAROSZ, W., NOWROUZEZAHRAI, D., THOMAS, R., SLOAN, P.-P., and ZWICKER, M. "Progressive photon beams". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 30.6 (Dec. 2011). DOI: 10/fn5xzj 1, 2.

[JZJ08] JAROSZ, W., ZWICKER, M., and JENSEN, H. W. "The beam radiance estimate for volumetric photon mapping". *Computer Graphics Forum (Proceedings of Eurographics)* 27.2 (Apr. 2008). DOI: 10/bjsfsx 1, 2.

[Kel97] KELLER, A. "Instant radiosity". *Annual Conference Series (Proceedings of SIGGRAPH)*. ACM Press, Aug. 1997. DOI: 10/fqch2z 1, 2.

[KF11] KULLA, C. and FAJARDO, M. "Importance sampling of area lights in participating media". *ACM SIGGRAPH Talks*. (Vancouver, Canada). ACM Press, 2011. DOI: 10/ccbw9x 4.

[KGH*14] KŘIVÁNEK, J., GEORGIEV, I., HACHISUKA, T., VÉVODA, P., ŠIK, M., NOWROUZEZAHRAI, D., and 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 1.

[KK06] KOLLIG, T. and KELLER, A. "Illumination in the presence of weak singularities". *Monte Carlo and Quasi-Monte Carlo Methods*. Ed. by NIEDERREITER, H. and TALAY, D. Springer-Verlag, 2006. DOI: 10/d42nk6 1, 2.

[NED11] NOVÁK, J., ENGELHARDT, T., and DACHSBACHER, C. "Screen-space bias compensation for interactive high-quality global illumination with virtual point lights". *Proceedings of the Symposium on Interactive 3D Graphics and Games*. (San Francisco, California). ACM Press, 2011. DOI: 10/ftsmbp 1.

[NGHJ18] NOVÁK, J., GEORGIEV, I., HANIKA, J., and JAROSZ, W. "Monte Carlo methods for volumetric light transport simulation". *Computer Graphics Forum (Proceedings of Eurographics State of the Art Reports)* 37.2 (May 1, 2018). DOI: 10/gd2jqq 1.

[NNDJ12a] NOVÁK, J., NOWROUZEZAHRAI, D., DACHSBACHER, C., and JAROSZ, W. "Progressive virtual beam lights". *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 31.4 (June 2012). DOI: 10/gfzndw 2, 7.

[NNDJ12b] NOVÁK, J., NOWROUZEZAHRAI, D., DACHSBACHER, C., and JAROSZ, W. "Virtual ray lights for rendering scenes with participating media". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 31.4 (July 2012). DOI: 10/gbbwk2 1–3, 5.

[OP11] OU, J. and PELLACINI, F. "LightSlice: matrix slice sampling for the many-lights problem". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 30.6 (Dec. 2011). DOI: 10/gfzm95 1, 2.

[SGH18] SRIWASANSAK, J., GRUSON, A., and HACHISUKA, T. "Efficient energy-compensated vpls using photon splatting". *Proc. ACM Comput. Graph. Interact. Tech.* 1.1 (July 2018). DOI: 10.1145/3203189 1.

[VG95] VEACH, E. and GUIBAS, L. J. "Bidirectional estimators for light transport". *Photorealistic Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*. Springer-Verlag, 1995. DOI: 10/gfznbh 2.

[WABG06] WALTER, B., ARBREE, A., BALA, K., and GREENBERG, D. P. "Multidimensional lightcuts". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 25.3 (July 2006). DOI: 10/dzgsz7 1, 2, 8.

[WBKP08] WALTER, B., BALA, K., KULKARNI, M., and PINGALI, K. "Fast agglomerative clustering for rendering". *Proceedings of IEEE Symposium on Interactive Ray Tracing*. Aug. 2008 4, 5, 7.

[WFA*05] WALTER, B., FERNANDEZ, S., ARBREE, A., BALA, K., DONIKIAN, M., and GREENBERG, D. P. "Lightcuts: a scalable approach to illumination". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 24.3 (Aug. 2005). DOI: 10/dhp5d3 1–3, 5.

[WKB12] WALTER, B., KHUNGURN, P., and BALA, K. "Bidirectional lightcuts". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 31.4 (July 2012). DOI: 10/gfzrcx 1, 2.