

# Rich-VPLs for Improving the Versatility of Many-Light Methods

Florian Simon, Johannes Hanika and Carsten Dachsbaecher

Karlsruhe Institute of Technology, Germany

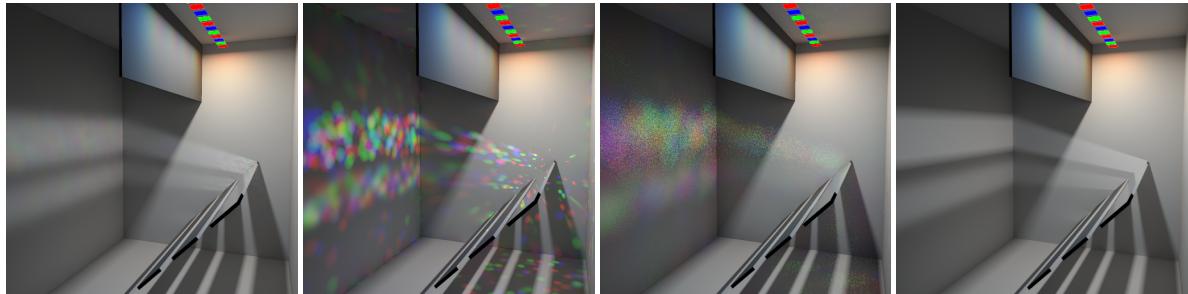


Figure 1: The Disco-scene with multiple colored light sources and reflected highly-glossy caustics where colors add to whitish light. Rendered with our new virtual light type “Rich-VPL” (left), standard VPLs (middle left), virtual spherical lights [HKWB09] (VSL, middle right), and path tracing reference (right). All many-light methods use 25k virtual lights; our method introduces an overhead of about 50% (less in more complex scenes) on top of standard VPL rendering, but significantly improves the ability to capture glossy and near-specular light transport. VSL require about 20% additional rendering time on top of standard VPLs due to stochastic sampling.

## Abstract

Many-light methods approximate the light transport in a scene by computing the direct illumination from many virtual point light sources (VPLs), and render low-noise images covering a wide range of performance and quality goals. However, they are very inefficient at representing glossy light transport. This is because a VPL on a glossy surface illuminates a small fraction of the scene only, and a tremendous number of VPLs might be necessary to render acceptable images. In this paper, we introduce Rich-VPLs which, in contrast to standard VPLs, represent a multitude of light paths and thus have a more widespread emission profile on glossy surfaces and in scenes with multiple primary light sources. By this, a single Rich-VPL contributes to larger portions of a scene with negligible additional shading cost. Our second contribution is a placement strategy for (Rich-)VPLs proportional to sensor importance times radiance. Although both Rich-VPLs and improved placement can be used individually, they complement each other ideally and share interim computation. Furthermore, both complement existing many-light methods, e.g. Lightcuts or the Virtual Spherical Lights method, and can improve their efficiency as well as their application for scenes with glossy materials and many primary light sources.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

## 1. Introduction

Rendering photorealistic images usually entails a physically-based simulation of light transport, which is nowadays almost exclusively computed using Monte Carlo methods, such as path tracing [Kaj86], photon mapping [Jen96], Metropolis light transport [VG97] and related approaches and variants. These methods capture all phenomena, but often produce noisy images even with very long computation times. Many-light methods have been prominent in computing (near) artifact-free images with low noise levels within

predictable render times and offer a simple solution to many, but not all, rendering problems. The underlying observation [Kel97] is that the light transport in a scene can be approximated by computing the direct illumination from many virtual point lights (VPLs). This explains two main benefits of many-light methods: first, they are scalable and can control the accuracy of the approximation by adjusting the number of VPLs, and second, the render times usually mainly depend on the number of VPLs (sub-linearly, e.g. when using Lightcuts [WFA<sup>\*</sup>05]).

However, many existing many-light methods struggle with glossy surfaces: generated VPLs often represent diffuse reflection only [WKB12], or their contributions are clamped to avoid spikes in the illumination due to a narrow emission profile that only illuminates a small portion of a scene. Clamping, however, has a negative impact on the perception of glossy materials [KFB10]. Virtual spherical lights [HKWB09] (VSLs) reduce artifacts and avoid clamping (for BRDF and geometry term) at the expense of higher shading cost. But they do not solve the problem that a huge number of virtual lights is required to faithfully capture glossy transport (Fig. 2).

The contributions of this paper expand the range of effects that can be rendered with many-light methods (note that we focus on surface transport and do not consider participating media in this work). First we introduce a novel virtual light type, the Rich-VPL, with the following beneficial properties:

- Rich-VPLs represent many incident light paths which increases their efficiency compared to standard VPLs. Emission profiles are, for example, stored in textures.
- The emission profiles can be prefiltered and mollified in the angular domain. This can be used to reduce artifacts and enables the rendering of near-specular light transport.
- Rich-VPLs integrate well into state-of-the-art many-light methods, such as Virtual Spherical Lights [HKWB09] and all variants of Lightcuts [WFA\*05, WABG06, WKB12].

Our second contribution is a technique for importance sampling and improving the locations of (Rich-)VPLs:

- (Rich-)VPLs should be created where they contribute considerably to the image. We propose a method to sample their locations proportional to the densely sampled product of sensor importance [Vea98] and radiance.
- We further introduce an (optional) cheap-to-compute, iterative relaxation scheme to obtain (Rich-)VPL locations with blue noise characteristics.

We demonstrate that many-light methods benefit from Rich-VPLs and the improved placement. We also show how the emission profiles of Rich-VPLs can be directly obtained from the radiance sampled during the placement of VPLs.

## 2. Related Work

The entire evolution of many-light methods began with Instant Radiosity [Kel97] where the concept of VPLs has been introduced. It has been recently portrayed in a comprehensive STAR on this topic [DKH\*13]. To this end, we concentrate on the most closely related work from this field.

**VPL Placement** It is obvious that VPLs that do not contribute (significantly) to an image only waste computation. To this end, Segovia et al. [SIMP06] trace paths from the camera to place VPLs on surfaces one bounce after the surfaces visible in the image. This idea has been further improved by employing the Metropolis-Hastings algorithm to

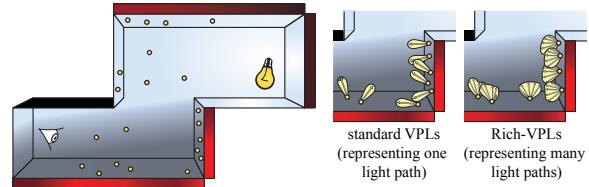


Figure 2: Left: a scene with one primary light and several VPLs. The red bars visualize the sensor importance reaching the surfaces (in this example surfaces are moderately glossy). We describe a method to importance sample VPL locations according to the product of importance and radiance (surface brightness), and introduce Rich-VPLs which represent many light paths and have a more efficient emission profile than standard VPLs (right).

create VPLs with roughly the same contribution [SIP07]. Georgiev and Slusallek [GS10] achieve the same goal by stochastically rejecting VPLs that do not significantly contribute to the image, which is evaluated for a sub-sampled image. However, while being simple to implement, this approach typically generates many candidate VPLs before one VPL is accepted. All three algorithms drive the distribution of VPLs by their contribution to the image (as our method), however, we use a significantly denser sampling of sensor importance (contribution to the image) and light paths.

Davidovic et al. [DKH\*10] also generate VPLs from the camera, denoted as local VPLs, which are used to locally increase the density of VPLs and to capture short-range interreflections. Local VPLs only contribute to a small portion of the image, and it is assumed that there is no occlusion for these VPLs. The global (long-range) transport is still computed with conventionally generated VPLs and could benefit from our VPL generation method.

**Scalability** The general idea of scalable methods is that within a set of VPLs, not every VPL contributes equally to a *shading point*. Typically VPLs are clustered and hierarchically organized such that the aggregate effect of a group of VPLs can be approximated by evaluating a single, brighter, representative VPL. Lightcuts [WABG06] determines a cut (the set of clusters) for every shading point using an analytic error bound and a perceptual metric. Lightcuts scales excellently for large numbers of VPLs, however, the error bounds ignore occlusion which results in wasteful cuts in scenes with complex visibility. Multi-dimensional Lightcuts [WABG06] further introduces a hierarchy of shading points to achieve scalable performance for effects such as depth of field or motion blur. Bidirectional Lightcuts [WKB12] extends the previous approaches to avoid clamping artifacts and handle a wider range of materials, such as glossy reflections, subsurface scattering and short-range indirect illumination. In all Lightcuts variants, VPLs only represent diffuse reflection.

Matrix Row-Column Sampling [HPB07] (MRCS) com-

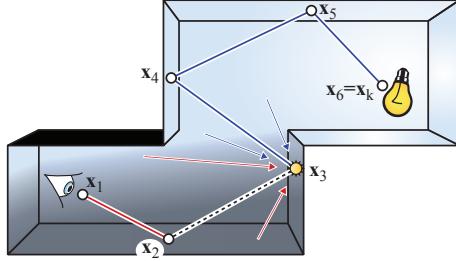


Figure 3: Notation for a transport path connecting the camera  $x_1$  to the light source  $x_k$ . A VPL is placed at  $x_3$  illuminating a visible surface  $x_2$ . Note that the probability of placing a VPL should depend on all importance arriving a surface point (indicated by the red arrows) and it should ideally represent not just one incident light direction in its emission (blue arrows).

putes only one global cut for the whole image. Again, the contribution of representatives is estimated using a sub-sampled image. Ou et al. [OP11] combine the idea of locally adapted cuts and MRCS. The underlying observation is that while adaptation is important, many cuts share a certain set of VPLs which can be reused for local cluster refinements. Lastly, Georgiev et al. [GKPS12] propose to choose the most relevant VPLs for a given shading point based on importance caching. In a pre-process they compute the contribution of all VPLs to the sub-sampled image, and use this cached contribution nearby a shading point as a discrete probability distribution to sample VPLs.

Rich-VPLs and our placement can be used with any of the above algorithms for improving the scalability of many-lights rendering, which we demonstrate exemplarily for Lightcuts.

**Avoiding the Singularity** A constant companion of many-light methods are singularities in the geometry term due to shading with point lights and with glossy BRDFs. The simplest workaround is to clamp the VPLs' contributions; however, this removes short-distance light transport and alters the appearance of materials [KFB10]. The clamped residual energy can be recovered with bias compensation [KK04] which is very costly, or approximate [NED11].

Virtual spherical lights (VSLs) [HKWB09] address the source of the singularity directly: they replace the point-to-point shading evaluation by an integration over the solid angle subtended by a spherical light, essentially averaging the BRDFs over the solid angle and distributing the energy over nearby surfaces (visibility is still evaluated point-to-point). VSLs can be combined with scalable methods, such as Lightcuts or MRCS, and also with our Rich-VPLs.

### 3. Theory and Motivation

In this section, we start from the path integral formulation of light transport and derive the motivation of Rich-

VPLs. When computing light transport, we strive to sample light paths  $\mathbf{X}$  with vertices  $\mathbf{x}_1$  (at the camera) to  $\mathbf{x}_k$  (on the light source) proportional to the measurement contribution [Vea98]:

$$f(\mathbf{X}) = W(\mathbf{x}_1) \left( \prod_{i=1}^{k-1} G_{\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}} \right) \left( \prod_{i=2}^{k-1} f_r(\mathbf{x}_i) \right) L(\mathbf{x}_k), \quad (1)$$

where  $W(\mathbf{x}_1)$  is the importance off the sensor, and  $L(\mathbf{x}_k)$  the emitted radiance. For the following considerations, we assume a path length of 3 vertices or greater and write  $f(\mathbf{X})$  as a product of the sensor importance  $W(\mathbf{x}_3)$  reaching  $\mathbf{x}_3$ , the BRDF at  $\mathbf{x}_3$ , and the radiance  $L(\mathbf{x}_3)$  reaching  $\mathbf{x}_3$ :

$$f(\mathbf{X}) = W(\mathbf{x}_3) \cdot f_r(\mathbf{x}_3) \cdot L(\mathbf{x}_3), \text{ with}$$

$$W(\mathbf{x}_3) = W(\mathbf{x}_1) \cdot G_{\mathbf{x}_1 \leftrightarrow \mathbf{x}_2} \cdot f_r(\mathbf{x}_2) \cdot G_{\mathbf{x}_2 \leftrightarrow \mathbf{x}_3}$$

$$L(\mathbf{x}_3) = L(\mathbf{x}_k) \left( \prod_{i=3}^{k-1} G_{\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}} \right) \left( \prod_{i=4}^{k-1} f_r(\mathbf{x}_i) \right) \quad (2)$$

Note how this seems similar to having created a VPL at  $\mathbf{x}_3$  with emission  $f_r(\mathbf{x}_3)L(\mathbf{x}_3)$ , illuminating a surface point  $\mathbf{x}_2$  visible to the camera (see Fig. 3). Instant radiosity [Kel97] creates VPLs proportional to  $L(\mathbf{x}_3)$  by particle tracing from the light sources, while more elaborate methods, e.g. [SIP07, GS10], consider the product of  $W(\mathbf{x}_3)$ ,  $L(\mathbf{x}_3)$  and  $f_r(\mathbf{x}_3)$  when creating VPLs. However, they do so by *sparsely sampling paths or VPL contributions to the image*.

We observe three fundamental aspects motivating our work:

- VPLs on glossy surfaces have a narrow emission profile due to  $f_r(\mathbf{x}_3)$ , while VPLs illuminating glossy surfaces have a spatially limited contribution to the image due to  $f_r(\mathbf{x}_2)$  (both simultaneously aggravates the problem). In these cases evaluating VPL contributions by sub-sampling in image space often misses important features.
- The emission of a VPL should ideally account for incident radiance from all possible light sub-paths reaching  $\mathbf{x}_3$ .
- Then VPL positions should be chosen with a probability density reflecting the importance for *all* shading points.

To this end, we propose to sample VPL locations proportional to the product of the *total importance*  $\hat{W}(\mathbf{x}_3)$  reaching a surface point  $\mathbf{x}_3$ ,  $\hat{W}(\mathbf{x}_3) = \int_{\mathbf{x}_2} W(\mathbf{x}_3) d\mathbf{x}$ , and the *total incident radiance*  $\hat{L}(\mathbf{x}_3)$  reaching  $\mathbf{x}_3$ . Note that we omit the BRDF at  $\mathbf{x}_3$  here: as our Rich-VPLs account for incident radiance from all possible directions, the emission profile (even with Rich-VPLs on highly glossy surfaces) is typically not narrow. We could, however, trivially include the maximum BRDF-value into the sampling to prevent creating VPLs on very dark surfaces.

Note that in practice, VPLs are not used for shading (near-)specular surfaces as a VPL's contribution can easily be (or will be) missed. Instead the camera sub-path is continued in this case until it reaches a moderately glossy or diffuse surface ( $LS^*(G|D)$  sub-path) which is then lit by VPLs. Consequently, we let  $\hat{W}$  also include importance via longer paths if these begin with consecutive specular interactions.

#### 4. Placement of Virtual Lights

In this section we detail our method for determining the VPL locations according to the aforementioned criteria. We also propose an optional relaxation step that improves the distribution in a post-process.

##### 4.1. Importance Sampling of VPL Locations

In order to sample VPL locations, we need to provide means to compute the incident radiance  $\hat{L}(\mathbf{x})$  and importance  $\hat{W}(\mathbf{x})$  at a surface point  $\mathbf{x}$ . To this end, we compute a photon map [Jen96] and an importance map [PP98] and estimate the two quantities by performing  $K$ -nearest neighbor density estimations with a 2D Epanechnikov kernel:

$$\hat{L}(\mathbf{x}) = \frac{2}{\pi d_K^2} \sum_{i=1}^K \Phi_i \cdot w_i(\mathbf{x}) \text{ and } \hat{W}(\mathbf{x}) = \frac{2}{\pi d_K^2} \sum_{i=1}^K \Psi_i \cdot w_i(\mathbf{x}),$$

with  $w(\mathbf{x}) = 1 - d_i^2/d_K^2$ , where  $d_i$  is the distance of  $i$ -th photon (importton) to  $\mathbf{x}_3$  ( $d_1 < d_2 < \dots < d_K$ ) and  $\Phi_i$  ( $\Psi_i$ ) is the incident flux (importance) of the  $i$ -th photon (importton). The incident flux of a randomly sampled photon with a path  $X = (\mathbf{x}_k, \dots, \mathbf{x}_3)$  where  $\mathbf{x}_k$  is on a light source is

$$\Phi = \left( \prod_{i=3}^{k-2} \frac{G_{\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}} f_r(\mathbf{x}_{i+1})}{p_A(\mathbf{x}_i)} \right) \frac{G_{\mathbf{x}_k \leftrightarrow \mathbf{x}_{k-1}}}{p_A(\mathbf{x}_{k-1}) N_P p_A(\mathbf{x}_k)} L(\mathbf{x}_k).$$

Here,  $N_P$  is the number of sampled photon paths and  $p_A(\mathbf{x}_i)$  is the probability of sampling  $\mathbf{x}_i$  in area measure. The incident importance  $\Psi$  of an importton with path  $X = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$  where  $\mathbf{x}_1$  is on the sensor is evaluated similarly.

In contrast to the original work [PP98], we store importons only at the interaction after the first non-specular bounce (after  $LS^*(G|D)$  interactions). On these surfaces we want to create VPLs, even if they are glossy or specular.

A naive solution to obtain the VPL locations is to perform uniform area sampling of the scene's surfaces, and use rejection sampling according to  $\hat{L}(\mathbf{x})\hat{W}(\mathbf{x})$ . However, this might require generating a huge number of samples to create VPLs, as many candidate locations might lie in unlit regions or regions not, or weakly, illuminating visible surfaces.

To avoid excessive rejection we take a different approach: we randomly choose one photon and take its position  $\mathbf{x}_p$  as a candidate location. Photons are distributed proportional to  $L(\mathbf{x})$ , which is only an estimator for  $\hat{L}(\mathbf{x})$ , however we did not observe any difference in our experiments (note that we will take the other light paths (photons) into account when creating a Rich-VPL). Next, we evaluate  $\hat{W}(\mathbf{x}_p)$  and create a VPL with an acceptance probability of:

$$P_{RS}(\mathbf{x}_p) = \min \left( 1, \hat{W}(\mathbf{x}_p) / (q \cdot \bar{W}) \right), \quad (3)$$

where  $\bar{W}$  is the average importance of all photons locations, and  $q$  is the ratio of the number of photons to the number of VPLs to be generated.

This sampling strategy can be used with any many-light method. In our case, we will reuse the photon map in the later stages of our algorithm.

Fig. 4 shows the differences of importon and photon densities and the target product distribution. Fig. 5 demonstrates the impact of the VPL placement strategy.

**Discussion** The selection of photons as VPL locations is somewhat similar to the importance driven photon map of Peter and Pietrek [PP98]. However, the construction of two independent maps for importons and photons is simpler to implement. Furthermore, their photon shooting is based on a discrete directional probability distribution of importance at every photon interaction. This is costly and requires an excessive number of importons for glossy BRDFs.

Metropolis Instant Radiosity [SIP07] creates VPL location at positions proportional to the full measurement contribution of *one individual path*. In contrast, we make sure the location is equally good for all shading points by using the total importance  $\hat{W}$ .



Figure 4: U-Shape Scene: qualitative visualization of the distribution of incident radiance  $\hat{L}$  (left) and incident importance  $\hat{W}$  (middle) and their product (right); individually tone-mapped for better presentation.

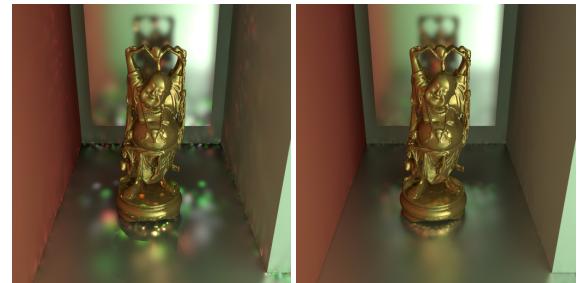


Figure 5: U-Shape Scene: even the impractical number of 1.4 million VPLs (for demonstration) with standard sampling [Kel97] does not yield acceptable results (left). The same number of VPLs sampled according to the product distribution (right). Note how the reflection of the Buddha is better captured (render time 3h 40min in both cases).

##### 4.2. Iterative Relaxation of VPL Locations

A low discrepancy distribution of VPL locations is desirable as this leads to less clumping artifacts in the shading. Of course we can stratify light sub-paths or use quasi-random

sequences (akin to Keller [Kel97]) when generating VPLs, however, some noise in the distribution remains [SJ09].

Following the idea of photon relaxation [SJ09], we can optionally relax the VPL locations after sampling them. Note that this does not move around energy in the scene in our approach as *the emission of VPLs is determined afterwards*.

We use the iterative relaxation scheme developed by Spencer and Jones from Turk's point repulsion method [Tur91]. In every iteration we search the  $K$ -nearest neighbors of a VPL and modify its location by:

$$\Delta \mathbf{x} = \frac{1}{K} \sum_{k=1}^K (\mathbf{x} - \mathbf{x}_k) \left( \frac{d}{\|\mathbf{x} - \mathbf{x}_k\| + \epsilon} - 1 \right),$$

where  $d$  is the distance to the  $K + 1$ -nearest neighbor and  $\mathbf{x}_k$  is the position of the  $k$ -th neighbor.

However, there is one important difference to photon relaxation [SJ09]: we must prevent VPLs from moving off the surfaces (esp. on concave surfaces) or under the surface (esp. in convex regions). In our case, we can achieve this with negligible cost: we observe that the photon map (Sect. 4.1) provides a dense sampling of the scene's surfaces with orders of magnitudes more photons than VPLs. After each relaxation step, we simply snap the position of each VPL to the location of the closest photon. This ensures that VPLs always reside on surfaces. The snapping did not negatively impact the relaxation according to our experiments.

Note that although we reuse the photon map as a dense surface sampling in this step, the relaxation itself is cheap to compute as the number of VPLs is significantly lower than the number of photons. In all our experiments we used  $K = 6$  and performed 20 relaxation iterations which took less than 1 second to compute in all our experiments; Fig. 6 shows the VPL locations and renderings without and with relaxation.

## 5. Generating Rich-VPLs

After having determined the (possibly relaxed) location  $\mathbf{p}$  of a Rich-VPL we need to determine and store its emission. For this we have to convolve the incident radiance at  $\mathbf{p}$ ,  $L_{in}(\mathbf{p}, \omega)$ , with the BRDF to obtain the exitant radiance:  $L_{\mathbf{p}}(\omega) = \int f_r(\mathbf{p}, \omega_i \rightarrow \omega) L_{in}(\mathbf{p}, \omega_i) d\omega_i$ . In principle it would be possible to sample  $L_{in}(\mathbf{p}, \omega)$  using path tracing, or estimating it from another set of conventionally created VPLs [SIMP06]. However, this is either costly, or again subsamples the light transport leading to problems with glossy surfaces and many primary lights.

Instead we can reuse the same photon map which we created to determine the VPL locations and take photons in the proximity of  $\mathbf{p}$  as an estimate of the incident radiance.

To this end, we query the photon map for the  $K$ -nearest photons at a position  $\mathbf{p}$  and estimate the exitant radiance for

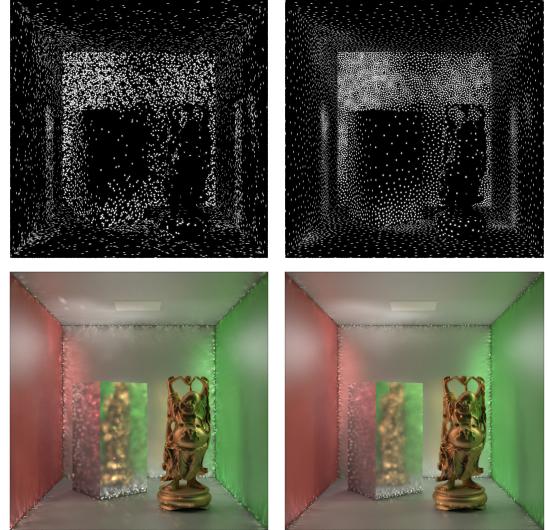


Figure 6: Top row: 10k Rich-VPLs sampled according to the product distribution before (left) and after relaxation (right). After 20 iterations the blue noise characteristic of the Rich-VPL distribution is clearly visible while the “global” distribution is well preserved. Bottom: renderings (indirect only) using the Rich-VPL distributions shown above. We intentionally disabled clamping to highlight the improvements.

an outgoing direction  $\omega$  as

$$L_{\mathbf{p}}(\omega) = \frac{1}{P_{RS}} \sum_{j=1}^K w(d_j, d_K) f_r(\mathbf{p}, \omega_j, \omega) \Phi_j$$

where  $\Phi_j$  is the incident flux,  $\omega_j$  is the incident direction of photon  $j$  and  $w(\cdot, \cdot)$  is a filter kernel with

$$\sum_{j=1}^K w(d_j, d_K) = 1.$$

Note that this function approximation is different from density estimation as we do not divide by surface area. This enables us to resample the photon locations for determining VPL positions and still reuse the photon map information to estimate the exitant radiance at a VPL. This estimation introduces bias similar to photon mapping which depends on  $K$  and the total number of photons.

As we want a Rich-VPL to represent many light paths, the number of photons  $N$  is typically large and it would be costly to query the incident radiance and evaluate the reflected radiance during shading. Instead we propose to tabulate  $L_{\mathbf{p}}(\omega)$  and store it as a small “environment map” which can be used during shading with fixed look-up cost, independent of the number of photons or incident light paths.

**Computing the Emission of Rich-VPLs** We use two different ways to compute the tabulated emission depending on the glossiness of the surface at  $\mathbf{p}$ :

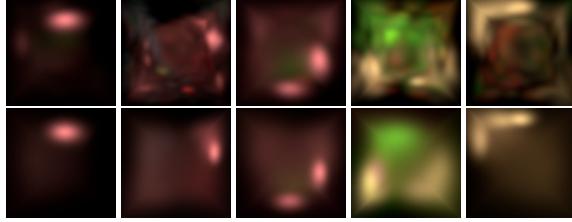


Figure 7: Top row: emission of Rich-VPLs tabulated and stored as octahedron environment maps (upper hemisphere shown only). Bottom row: approximation of the emission using 5 von Mises-Fisher-lobes per color channel (see Sect. 8).

- **highly glossy/specular** (the reflection is focused to a narrow solid angle): for each photon we importance sample outgoing directions  $\omega$  according to the BRDF  $f_r(\mathbf{x}_k, \omega_k \rightarrow \omega)$  ( $\mathbf{x}_k$  is the position, and  $\omega_k$  the incident direction of the  $k$ -th photon) and accumulate the reflected radiance.
- **diffuse/moderately glossy**: first tabulate the accumulated incident radiance of all  $N$  photons before convolving it with the BRDF. However, if we accumulate first and then convolve, we cannot use the BRDFs at the photons' locations and instead use the BRDF at the VPL position.

For VPLs on Lambertian surfaces, we simply store the flux as there is no need to tabulate  $L_p(\omega)$ . Note that we can handle singular BRDFs (e.g. perfect mirrors) as the energy is spread over at least the solid angle of one texel in the environment map. Unless otherwise noted, we used an octahedron map [ED08] with  $32 \times 32$  texels for the hemisphere of outgoing directions (Fig. 7); the angular resolution then is approximately 4 to 5 degrees. We used a bilinear filter during the accumulation steps. Note that other filters can be used to increase the mollification (see also Sec. 8).

**Shading with Rich-VPLs** Naive many-light rendering with Rich-VPLs is almost identical to traditional VPL shading, except that Rich-VPLs directly store outgoing radiance and the BRDF is not evaluated on the fly (which is of course trivial for VPLs on Lambertian surfaces assumed in many methods). When accessing the environment map during shading, we use bilinear filtering to obtain a smooth radiance field.

## 6. Many-Light Scalability and Fighting Artifacts

Naive shading with Rich-VPLs, computing the illumination of every virtual light to every shading point, benefits from our improved placement and the richer emission profiles. However, other typical problems of many-light methods remain: scalability, as not every (Rich-)VPL is equally important for every shading point, and singularities and clamping.

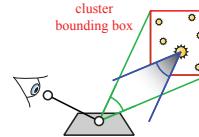
**Lightcuts with Rich-VPLs** Lightcuts [WFA<sup>\*</sup>05] clusters VPLs according to spatial proximity and orientation. For each cluster a representative VPL is chosen and its emission



Figure 8: Rich-VPL Lightcuts with an error threshold of 0.5% (left) and naive Rich-VPLs (right). Shading times (without Rich-VPL sampling and generation) were 2:30 min and 13 min for 42k virtual lights.

equals that of all VPLs in the cluster together. In its original form, it does not support glossy VPLs as the clustering metric is not well-suited for strongly varying emission direction.

Rich-VPLs can be easily clustered for spatial proximity and we can directly obtain the emission of a representative by summing up the environment maps of all Rich-VPLs in a cluster (for representatives we need to store both hemispheres in the environment map instead of only one).



To estimate the error bound during cut refinement, we need to compute the maximum incident radiance from a cluster onto a shading point. Therefore we compute the solid angle subtended by the cluster's bounding box and then determine the maximum exitant radiance at the representative within the same solid angle centered around the direction to the shading point [WFA<sup>\*</sup>05, Sec.4.1].

As we store the emission in an environment map, we can easily compute a max-mip map hierarchy for the emission

Scene	#VPLs	VPLs	VPLs+IS	Rich-VPLs
Fig. 8 (Box)	25k	436	456	589
Fig. 4,5 (U-Shape)	35k	340	550	555
Fig. 12 (Garage)	13k	372	565	576
Fig. 11 (Kitchen)	42k	814	833	837

Scene	#VPLs	IS	relax.	enrich	shading
Fig. 8 (Box)	25k	11	< 1	11	567
Fig. 4,5 (U-Shape)	35k	13	< 1	8	533
Fig. 12 (Garage)	13k	18	< 1	2	554
Fig. 11 (Kitchen)	42k	12	< 1	10	808

Figure 9: Top table: total runtimes in seconds for VPLs, VPLs using our proposed importance sampling (IS, Sec. 4), and Rich-VPLs with IS. Timing breakdown (bottom table): individual timings in seconds for IS, relaxation, Rich-VPL creation (enrich), and shading. All renders used  $q = 100 \times$  more photons than (Rich-)VPLs and 20 relaxation iterations.

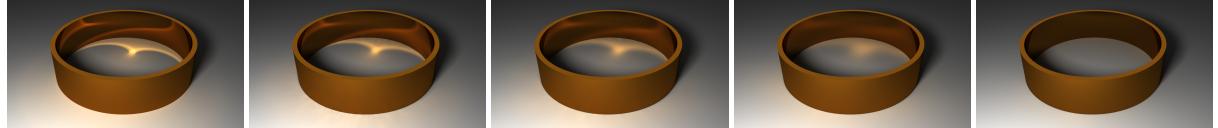


Figure 10: A metal ring rendered with different resolutions for the tabulated exitant radiance. On the far left is the path tracing reference and on the far right a VPL rendering with diffuse VPLs. In between are resolutions for the exitant radiance of  $32^2$ ,  $16^2$  and  $8^2$ . The mean squared error compared to the reference is  $0.59 \cdot 10^{-3}$ ,  $0.69 \cdot 10^{-3}$ ,  $1.01 \cdot 10^{-3}$  and  $3.39 \cdot 10^{-3}$ , respectively.



Figure 11: The Kitchen scene rendered with 42k (Rich-)VPLs. From left to right: standard VPLs (13 min), VPL with our importance sampling (14 min), and Rich-VPLs with importance sampling (14 min). In this scene the overhead for Rich-VPLs is small as the render time is dominated by shadow connections.

and directly access the appropriate mip-level to obtain a conservative estimate for a cone of directions.

During our experiments we found that the error threshold of the original heuristic for cut refinement has to be lowered for highly glossy VPLs as otherwise distracting artifacts appear. This results in larger cuts and somewhat reduces the efficiency of Lightcuts, but still provides improved scalability (see Fig. 8). Note that Lightcuts requires storing twice as many emission profiles due to inner nodes; moreover for these we need to store full spherical emission.

**Rich-VSLs** We can combine Rich-VPLs with the idea of VSLs [HKWB09] which address the problem of shading singularities by distributing the energy of a VPL over nearby surfaces. During VSL shading, the BRDFs at the shading point and the VSL are importance-sampled. Rich-VSLs work almost identical, except that we do not importance sample the BRDF (and thus the exitant radiance) at the VSL location. First of all, this would require the (costly) sampling according to a discrete probability density when storing the emission as environment map (which is our default case). Second, even on glossy surfaces the emission of a Rich-VPLs/Rich-VSLs is spread more evenly and not spiky.

## 7. Implementation and Results

We implemented our VPL placement and the different many-light methods (standard VPLs, Rich-VPLs, Lightcuts, VSLs) in our own rendering framework, which also supports path tracing for reference images and photon mapping. In our (Rich-)VPL shading we are only clamping the geometric term, while the BSDF values are never clamped.

For our VPL location sampling (Sec. 4) we cast 32 importon paths per pixel, and use a photon map which contains  $q = 100$ -times as many photons as we want to create VPLs. When querying the photon map to compute the emission of a Rich-VPL, we collect photons within a spherical proximity with a radius equal to the distance to the second closest Rich-VPL. In our renderings with Rich-VSLs we use the same parameters as Hašan et al. [HKWB09] (radii  $M$ -times the distance to the 10th nearest neighbor VPL, with  $M = 4..10$ ).

To compute the exitant radiance of Rich-VPLs (Sec. 5) we execute the BRDF importance sampling on the CPU, and the convolution of the accumulated incident radiance on the GPU using CUDA. For all results, Rich-VPLs use a hemispherical environment map of  $32 \times 32$  texels. During shading we use interleaved sampling with a  $3 \times 3$  pixel sub-sampling scheme just as [HKWB09, DKH\*10].

All steps of our algorithm besides the aforementioned CUDA kernel are implemented on the CPU. All components except for the kD-tree construction for photon and importon maps are multi-threaded.

We evaluate and compare Rich-VPLs running on an Intel Core i7-3770 CPU with 3.40GHz and 16GB ram, using eight threads. Fig. 9 shows timings of our method. For the same number of virtual lights, Rich-VPLs with our location sampling require approximately 50% more computation time (slightly increased cost of emission computation, shading, and the importance sampling).

The impact of our VPL placement can be best observed in the U-Shape scene (Fig. 5). Due to difficult visibility, the benefit of good VPL importance sampling is tremendous. To demonstrate the effectiveness of the Rich-VPLs, we show the disco scene in Fig. 1. This example intentionally has very

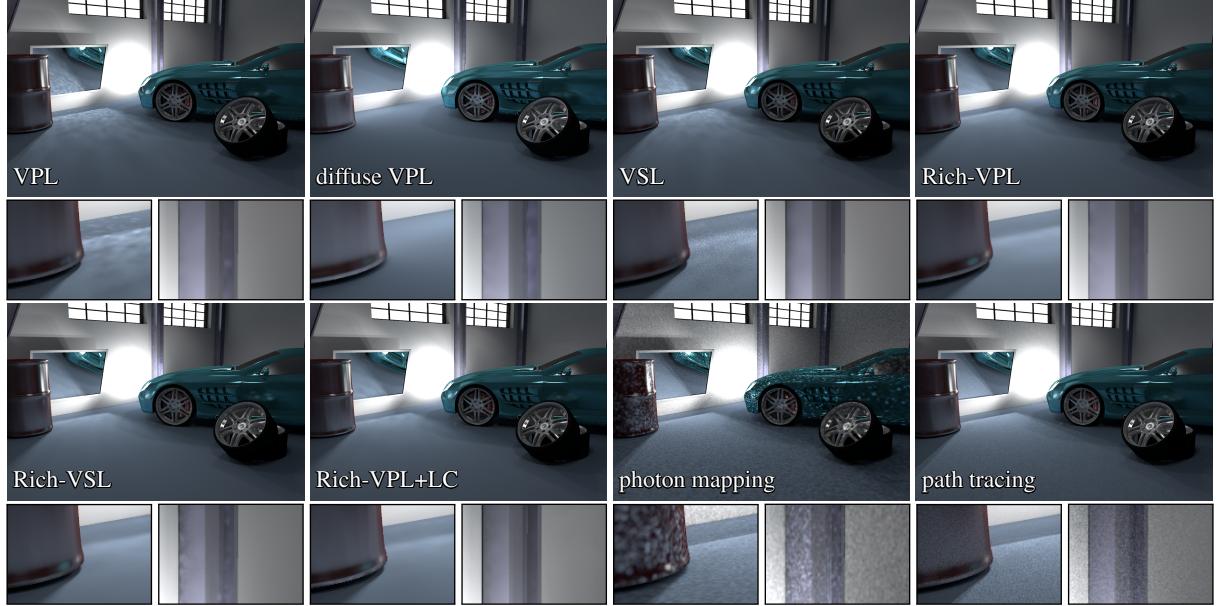


Figure 12: The Garage-scene rendered with 250k virtual lights (created from 11 primary light sources). Rich-VPLs can capture highly glossy light transport much better than standard VPLs. In many methods, VPLs are assumed to be diffuse which results in wrong or missing light transport when compared to the path tracing reference. VSLs handle glossy reflections better than VPLs, but are still more prone to artifacts than Rich-VPLs. Combined with Lightcuts (Rich-VPL+LC) render times with Rich-VPLs are reduced by about 20% (1% threshold). All VPL-based methods used our importance sampling for VPL placement (Sec. 4).

simple visibility to highlight the benefits for glossy reflections and multiple primary light sources.

We can trade visual accuracy for memory by modifying the resolution of the tabulated exitant radiance. To demonstrate the impact of varying angular resolutions we rendered a scene with a metal-like ring (Fig. 10) and a modified CornellBox (see supplemental material) with path tracing, diffuse VPLs, and Rich-VPLs with different resolutions.

To show the performance of the algorithm in more realistic scenes, we show the Kitchen-scene (Fig. 11) and the Garage-scene (Fig. 12). The kitchen shows roughly the same increase in quality from VPL importance sampling as well as from using Rich-VPLs. The garage was rendered with all diffuse VPLs (as used in most many-light methods, e.g. Bidirectional Lightcuts [WKB12]), a direct rendering of a photon map (122 million photons) and a path tracing reference (250k samples per pixel). We compare these to VPLs, Rich-VPLs, VSLs and Rich-VSLs for all of which we used our VPL importance sampling. While diffuse VPLs produce a very smooth image, they fail to capture features such as the shadow boundary at the barrel completely. For glossy virtual lights, VSLs should ameliorate the remaining problems with blotches, however, the original heuristic was difficult to adjust (the parameter  $M$  which controls the VSL radii) for the very high VSL density on the mirror without overblurring the other parts (see for example the reflection of the tires in the door of the car). We believe that a better heuristic can

easily be designed. The insets in Fig. 12 show how Rich-VPLs faithfully maintain the shadow boundaries, with just a little blur due to the environment map quantization; also the typical VPL blotchiness is reduced significantly. When using Lightcuts on top of Rich-VPLs we gain an overall speedup of about 20% (compared to brute force Rich-VPLs). The cut sizes in this scene varied strongly due to the glossy surfaces between 900 and 70k virtual lights.

In Fig. 13 we show equal-time comparisons of the Rich-VPLs to path tracing and photon mapping, and the resulting mean squared error (MSE) compared to a path tracing reference. We also show the relative error  $|r - v|/r$  where  $r$  is the averaged (over RGB) pixel value of the path tracing reference. In Fig. 14 and the supplemental material we also show equal-time comparisons of Rich-VPLs to standard VPLs. A comparison of Rich-VPLs to rendering the original set of photons as VPLs can also be found in the supplemental material.

## 8. Discussion and Future Work

**Richness and Mollification** Rich-VPLs represent an arbitrary number of incident light paths and are thus better-suited for glossy transport than traditional VPLs/VSLs, and they efficiently handle scenes with many primary lights (Fig. 1). They inherently mollify reflections off (near-)specular surfaces, but only by this VPLs also can render this transport.

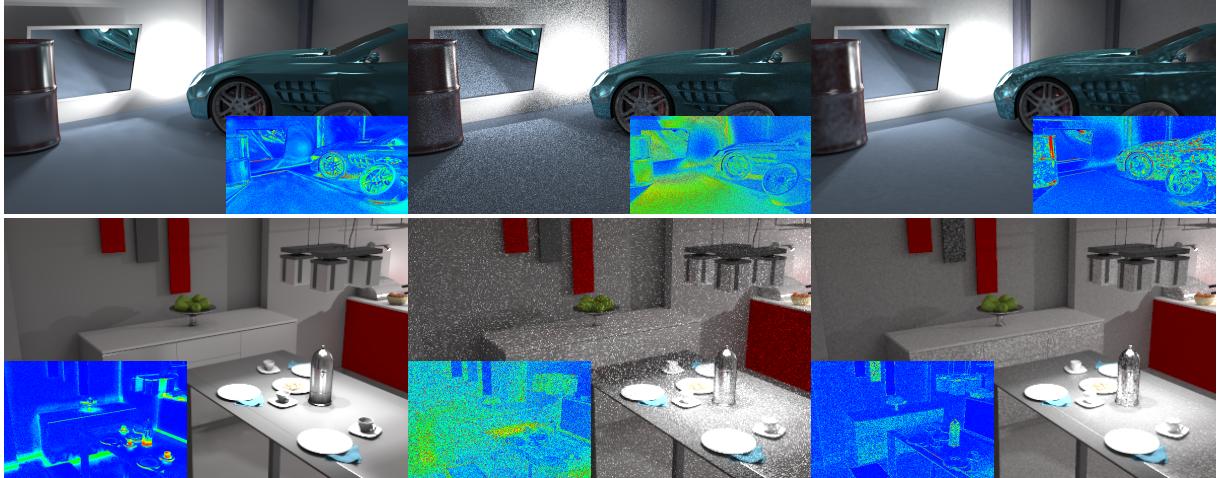


Figure 13: Equal-time rendering of the Garage (5h) and Kitchen (1h) scenes with Rich-VPLs (left), path tracing (middle) and photon mapping (right). The MSE compared to a path tracing reference is  $4.70 \cdot 10^{-3}$ ,  $12.35 \cdot 10^{-3}$  and  $4.73 \cdot 10^{-3}$  in the Garage, and  $0.24 \cdot 10^{-3}$ ,  $0.59 \cdot 10^{-3}$  and  $0.29 \cdot 10^{-3}$  in the Kitchen, respectively. The insets show the relative error compared to the path tracing reference.

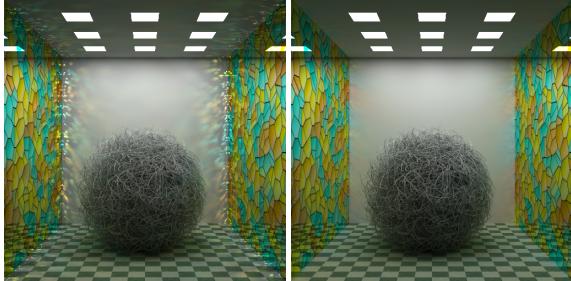


Figure 14: Equal-time comparison (72 min) of standard VPLs (left) and Rich-VPLs (right) in a scene with textured mirror walls.

An interesting future work would be to explicitly control the mollification for progressive rendering [KD13] or to render with fewer Rich-VPLs trading accuracy for speed.

**Photon Mapping** A legitimate question is how Rich-VPLs position themselves compared to photon mapping, in particular as we compute a photon map. Many-light methods in general can be interpreted as a “final shooting” algorithm. A Rich-VPL can furthermore be seen as a cache storing the result of a radiance kernel estimation (of many photons). During final gathering for photon mapping this kernel estimation would be evaluated much more often. In the end it depends on the particular application whether noise due to shooting gather rays, or smooth shading yet correlated light transport with many-light rendering is better suited.

**Clamping and Bias** Rich-VPLs do not address clamping or bias compensation problems. As demonstrated, artifacts can be reduced with VSLs, but not eliminated completely. We believe that Rich-VPLs are a good complement to the Bidirectional Lightcuts method [WKB12] which significantly re-

duces bias, enabling it to handle more transport using virtual lights (e.g. caustics as in Fig. 10 and Fig. 12 are not handled at all by [WKB12]).

**Memory** For very large numbers of Rich-VPLs the memory required for storing their emission might become an issue (e.g. for  $32 \times 32$  texels, each holding a RGB-float triple, we need 12 kilobytes). Therefore we evaluated whether the emission can be sufficiently well represented using Gaussian-like distributions. To this end, we computed a fit of a von Mises-Fisher (vMF) distribution with 5 lobes per color channel and 32 iterations using expectation-maximization. To reduce memory consumption during rendering, we compute the fit for each completed Rich-VPL and only keep the coefficients for subsequent shading. For each lobe we store 4 floats (2 for direction, inverse width and amplitude), i.e. 240 bytes per Rich-VPL. Fig. 7 shows several original emission profiles and their vMF-fits. Fig. 15 shows that vMF-fits result in very similar renderings. In all other examples we used tabulated emission instead of our unoptimized proof-of-concept vMF-implementation. Another simple way to reduce memory requirements is to adapt the environment map resolution to surface glossiness. We can also compute smaller photon maps and compute the Rich-VPLs’ emission progressively (compute emission from a smaller photon map, discard photon map and repeat with a new one).

**Placement Heuristics** We believe that our VPL placement can flexibly incorporate heuristics for more fine-grained control. For example, we can artificially increase the importance (deviating from physics) when an importon travelled along a short path segment. This would enforce more VPLs in cavities which are often undersampled with many-light methods.



Figure 15: Comparison (only indirect illumination shown) using tabulated emission for Rich-VPLs (left), von Mises-Fisher approximation (5 lobes per color channel, center), and path tracing reference (right). Bottom: the respective absolute difference images scaled by a factor of 4.

## 9. Conclusions

In this paper we introduced Rich-VPLs to increase the efficiency of many-light rendering. This new lighting primitive accounts for the contribution of many, instead of one, incident light paths. Rich-VPLs enable angular filtering of light transport and extend the set of lighting features that can be handled well with many-light methods. Along with the new light type, we propose a VPL placement strategy that accounts for the total importance (contribution to image) as well as the total incident radiance of potential VPL locations. We demonstrated the benefits of our method and its ability to complement many-light methods addressing orthogonal problems such as scalability and bias compensation.

## References

- [DKH\*10] DAVIDOVIČ T., KŘIVÁNEK J., HAŠAN M., SLUSALLEK P., BALA K.: Combining global and local virtual lights for detailed glossy illumination. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 29, 6 (2010), 143:1–143:8. [2](#), [7](#)
- [DKH\*13] DACHSBACHER C., KŘIVÁNEK J., HAŠAN M., ARBREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. *Computer Graphics Forum* 33, 1 (2013), 88–104. [2](#)
- [ED08] ENGELHARDT T., DACHSBACHER C.: Octahedron environment maps. In *Proceedings of Vision, Modeling and Visualization* (2008), pp. 383–388. [6](#)
- [GKPS12] GEORGIEV I., KŘIVÁNEK J., POPOV S., SLUSALLEK P.: Importance caching for complex illumination. *Computer Graphics Forum (Proc. of Eurographics)* 31, 3 (2012), 701–710. [3](#)
- [GS10] GEORGIEV I., SLUSALLEK P.: Simple and robust iterative importance sampling of virtual point lights. In *Eurographics short papers* (2010). [2](#), [3](#)
- [HKWB09] HAŠAN M., KŘIVÁNEK J., WALTER B., BALA K.: Virtual spherical lights for many-light rendering of glossy scenes. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 28, 5 (2009), 143:1–143:6. [1](#), [2](#), [3](#), [7](#)
- [HPB07] HAŠAN M., PELLACINI F., BALA K.: Matrix row-column sampling for the many-light problem. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 26, 3 (2007), 26. [2](#)
- [Jen96] JENSEN H. W.: Global illumination using photon maps. In *Proc. Eurographics Workshop on Rendering* (1996), pp. 21–30. [1](#), [4](#)
- [Kaj86] KAJIYA J. T.: The rendering equation. *Computer Graphics (Proc. SIGGRAPH)* (1986), 143–150. [1](#)
- [KD13] KAPLANYAN A., DACHSBACHER C.: Path space regularization for holistic and robust light transport. *Computer Graphics Forum (Proc. of Eurographics)* 32, 3 (2013). [9](#)
- [Kel97] KELLER A.: Instant radiosity. In *SIGGRAPH '97* (1997), pp. 49–56. [1](#), [2](#), [3](#), [4](#), [5](#)
- [KFB10] KŘIVÁNEK J., FERWERDA J. A., BALA K.: Effects of global illumination approximations on material appearance. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 29, 4 (2010), 112:1–112:10. [2](#), [3](#)
- [KK04] KOLLIG T., KELLER A.: Illumination in the presence of weak singularities. *Monte Carlo and Quasi-Monte Carlo methods* (2004), 245–257. [3](#)
- [NED11] NOVÁK J., ENGELHARDT T., DACHSBACHER C.: Screen-space bias compensation for interactive high-quality global illumination with virtual point lights. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2011), pp. 119–124. [3](#)
- [OP11] OU J., PELLACINI F.: Lightslice: Matrix slice sampling for the many-lights problem. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* (2011). [3](#)
- [PP98] PETER I., PIETREK G.: Importance driven construction of photon maps. In *Proc. Eurographics Workshop on Rendering* (1998), pp. 269–280. [4](#)
- [SIMP06] SEGOVIA B., IEHL J. C., MITANCHEY R., PÉROCHE B.: Bidirectional instant radiosity. In *Proc. Eurographics Symposium on Rendering* (2006), pp. 389–398. [2](#), [5](#)
- [SIP07] SEGOVIA B., IEHL J. C., PÉROCHE B.: Metropolis instant radiosity. *Computer Graphics Forum* 26, 3 (2007), 425–434. [2](#), [3](#), [4](#)
- [SJ09] SPENCER B., JONES M.: Into the blue: Better caustics through photon relaxation. *Computer Graphics Forum* 28, 2 (2009), 319–328. [5](#)
- [Tur91] TURK G.: Generating textures on arbitrary surfaces using reaction-diffusion. *Computer Graphics (Proc. SIGGRAPH)* (1991), 289–298. [5](#)
- [Vea98] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1998. AAI9837162. [2](#), [3](#)
- [VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. *Proc. ACM SIGGRAPH* (1997), 65–76. [1](#)
- [WABG06] WALTER B., ARBREE A., BALA K., GREENBERG D. P.: Multidimensional lightcuts. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 25, 3 (2006), 1081–1088. [2](#)
- [WFA\*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: a scalable approach to illumination. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 24, 3 (2005), 1098–1107. [1](#), [2](#), [6](#)
- [WKB12] WALTER B., KHUNGURN P., BALA K.: Bidirectional lightcuts. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 31, 4 (2012), 59:1–59:11. [2](#), [8](#), [9](#)