

# Efficient Energy-Compensated VPLs using Photon Splatting

JAMORN SRIWASANSAK, The University of Tokyo

ADRIEN GRUSON, The University of Tokyo and JFLI, CNRS, UMI 3527

TOSHIYA HACHISUKA, The University of Tokyo

Efficient light transport simulation is important for many applications in computer graphics. Many-light methods based on virtual point lights (VPLs) especially work well for scenes consist mainly of diffuse materials. Rendered images with VPLs, however, are known to suffer from energy loss due to clamping of the VPL contributions. Removing this clamping, unfortunately, introduces significant variance in images due to the singularities of VPLs. We propose an efficient and practical algorithm which compensates energy loss of VPLs using yet another efficient rendering method – photon splatting. Recent unified path sampling frameworks show that VPLs are in fact photons. Based on this unified formulation, we propose to reuse VPLs as photons to compensate missing energy of clamped VPLs. This combination is very natural: the singularities of VPLs happen if a shading point is too close to VPLs, but such VPLs can also be seen as neighboring photons that are indeed useful for photon density estimation. We formulate this energy compensation process using multiple importance sampling to combine rendering with VPLs and photon splatting optimally. Our algorithm can be efficiently implemented on GPUs since rendering with VPLs and photon splatting are both suitable for GPUs. The results demonstrate that our method produces more accurate images in the same rendering time than the existing methods. For common use cases of rendering mostly diffuse scenes, our work leads to an efficient, accurate, and practical rendering algorithm based on VPLs.

CCS Concepts: • Computing methodologies → Rendering; Ray tracing;

Additional Key Words and Phrases: global illumination, many lights, density estimation

## ACM Reference Format:

Jamorn Sriwasansak, Adrien Gruson, and Toshiya Hachisuka. 2018. Efficient Energy-Compensated VPLs using Photon Splatting. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 1 (May 2018), 13 pages. <https://doi.org/10.1145/3203189>

## 1 INTRODUCTION

Monte Carlo ray tracing is a common approach for light transport simulation in rendering. This approach probabilistically samples light transport paths and accumulates their contribution to compute the value of each pixel. While Monte Carlo ray tracing produces accurate results, the rendered images will exhibit noise if the number of samples is not sufficiently high.

Many-light methods [Dachsbaecher et al. 2014; Keller 1997] address this issue by reusing parts of sampled paths across all the pixels in the image. In the first step, several paths starting from the light sources (light subpaths) are generated and the vertices of the generated paths are stored as virtual point lights (VPLs). In the second step, a ray is traced from the camera to find the shading point. The last step is to accumulate the contributions from all the VPLs. Since each light subpath is

---

Authors' addresses: Jamorn Sriwasansak, The University of Tokyo, jamorn.sriwasansak@gmail.com; Adrien Gruson, The University of Tokyo, and, JFLI, CNRS, UMI 3527, adrien.gruson@gmail.com; Toshiya Hachisuka, The University of Tokyo, thachisuka@siggraph.org.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2577-6193/2018/5-ART \$15.00

<https://doi.org/10.1145/3203189>

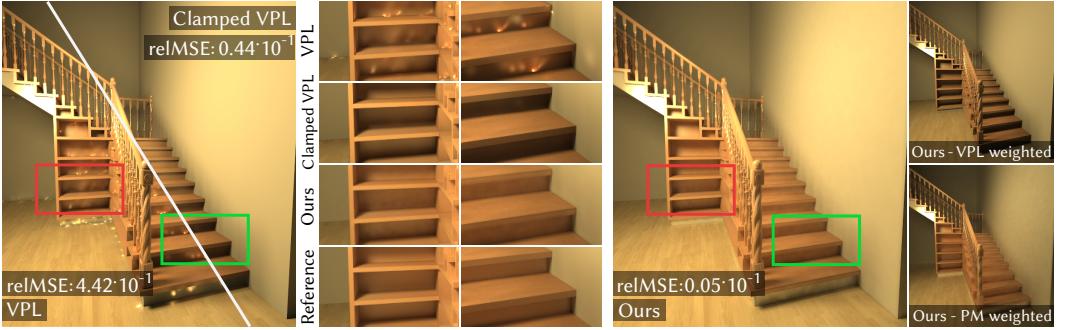


Fig. 1. Comparison between our method and many-light techniques with 15 seconds of rendering. Virtual point lights (VPLs) suffer from the singularities and there are many bright splotches in the image. Clamping (Clamped VPL) can remove such artifacts, but the image becomes darker due to energy loss. Our method avoids the singularities by reusing VPLs as photons in photon mapping and produces a smooth image without energy loss.

reused by all the pixels, this approach significantly increases the number of accumulated samples per path. Many-light methods thus produce smooth images very quickly compared to Monte Carlo ray tracing.

This efficient path reuse in VPLs, unfortunately, comes with the fundamental issue of singularities in VPL contributions. Due to its formulation, the contribution of each VPL is inversely proportional to the squared distance to a shading point. Such singularities can show up as bright splotches in images rendered with VPLs. A common practice to alleviate this issue is to clamp the contribution of each VPL to a user-specified threshold. This common practice, however, leads to energy loss and shows up as dark corners in rendered images. While previous work proposed to compensate lost energy by using Monte Carlo ray tracing [Kollig and Keller 2006] or its approximation in the screen space [Novák et al. 2011], efficient and accurate energy compensation of VPLs remains an open problem.

We propose an efficient approach to compensate lost energy in rendering with VPLs based on a novel combination with photon splatting. Building upon recent unified light transport frameworks [Georgiev et al. 2012; Hachisuka et al. 2012], we explain how VPLs can be seen as photons in photon density estimation. This combination is natural: the singularities of VPLs become problematic when they are too close to a shading point, but such VPLs are useful as photons in photon density estimation and causes no singularities. Our method is computationally very efficient since we can reuse generated VPLs as photons and both approaches are suitable to run on GPUs. To summarize, the main contributions of this paper are:

- Efficient energy compensation of VPLs using photon splatting.
- Application of the unified framework to combine rendering with VPLs and photon splatting.

Fig. 1 highlights that our method can produce smooth images without the issue of singularities. Since this scene contains lots of concave corners, the singularities of VPLs show up very frequently. By using our combination, problematic VPLs are now used as neighboring photons without such singularities. The rendered image of our method is close to the reference only within several seconds.

## 2 RELATED WORK

**Singularities in many-light methods.** Many-light methods are prone to cause singularities [Dachsbacher et al. 2014; Keller 1997]. A common approach to address this problem is to clamp the contribution of each VPL to a user-defined threshold. While clamping removes bright splotches, it results in darker images due to missing energy. Hašan et al. [2009] addressed the singularities by blurring each VPL to have non-zero solid angle. A blurred VPL is called a virtual spherical light (VSL). Since each VSL has non-zero solid angle, it is possible to directionally integrate the contributions of VSLs and thus cancels out the singularities. This formulation can be seen as a particular form of final gathering in photon density estimation. While we also use photon density estimation, we do not need to perform directional integration which can be computationally costly.

**Energy compensation.** Another approach is to compensate the missing energy by using another rendering algorithm that does not suffer from the same singularities [Kollig and Keller 2006; Raab et al. 2008]. Since the calculation of the missing energy itself is also costly, an approximation is necessary to achieve interactive performance with this approach [Engelhardt et al. 2012; Novák et al. 2011]. Davidović et al. [2010] proposed to separate the computation into global and local illumination. Their method generates a dense set of local VPLs using path tracing and ignoring the visibility term. While singularities of their local VPLs are less noticeable than those of regular VPLs, clamping is still necessary. Our method does not require any clamping and efficiently reuses VPLs as photons without any approximation except for bias in photon density estimation.

**Scalable methods.** The computational cost of many-light methods is proportional to the number of VPLs. Matrix row-column sampling (MRCS) [Hašan et al. 2007] addressed this problem based on a transfer matrix and exploited the structure of the matrix to approximate the contributions of multiple VPLs. Lightcuts [Walter et al. 2006, 2005, 2012] alleviated this scalability problem by hierarchically clustering VPLs into a tree. Lightslice [Ou and Pellacini 2011] combined the idea of lightcuts and MRCS by clustering shading points into slices and performed MRCS for each slice. These methods approximate the result of VPL rendering such that the computational cost becomes sublinear and allow us to use a significantly large number of VPLs. However, clamping is still necessary. Davidović et al. [2012] thus proposed a modification on lightcuts where the clamping factor is relaxed progressively. While our method performs well without these scalable methods, one can optionally use scalable methods to speed up the VPLs rendering part of our method.

**Photon density estimation.** Photon mapping [Jensen 1996] introduced the concept of density estimation to rendering. We employ a variant of photon mapping called photon splatting [Stürzlinger and Bastos 1997]. Photon splatting can be efficiently implemented on GPUs using rasterization [Lavignotte and Paulin 2003; McGuire and Luebke 2009] to achieve interactive performance. While original photon mapping is biased and has limited accuracy given the memory budget, it can be made into a progressive method [Hachisuka et al. 2008; Knaus and Zwicker 2011], making it asymptotically converge to the correct result.

We build upon the unified light transport frameworks [Georgiev et al. 2012; Hachisuka et al. 2012] to combine photon splatting and rendering with VPLs using multiple importance sampling [Veach and Guibas 1995]. This combination is fundamentally different from bidirectional lightcuts [Walter et al. 2012] which is based on bidirectional path tracing. To our best knowledge, our work is the first to utilize the unified frameworks in many-light methods.

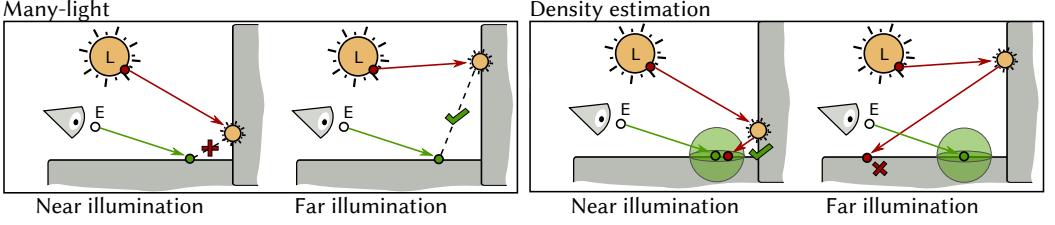


Fig. 2. When a light subpath vertex is close to the shading point, using this vertex as a VPL leads to the issue of the singularity. However, the same vertex is, in fact, useful in photon density estimation since we want this vertex to be close to the shading point. The opposite is true when the vertex is far from the shading point: this vertex is not useful in photon density estimation, but the singularities of VPLs do not happen.

### 3 OVERVIEW

Unified light transport frameworks [Georgiev et al. 2012; Hachisuka et al. 2012] provided a theoretical connection between Monte Carlo path integration and photon density estimation. Since many-light methods are Monte Carlo path integration methods, it is possible to reformulate them under the unified light transport frameworks. One can thus see that VPLs (as they are light subpath vertices) are essentially photons in photon density estimation. This reformulation leads to our key observation: when VPLs are suffering from the singularities due to the geometry term, they are close to a shading point, which are indeed useful as photons in photon density estimation. This combination allows us to avoid singularities of VPLs by switching to photon density estimation when appropriate. Fig. 2 illustrates this observation.

Unlike some existing energy compensation for VPLs, our formulation employs a balance heuristic using multiple importance sampling [Veach and Guibas 1995]. This combination is possible due to our introduction of the unified frameworks into many-light methods. Since VPLs are photons, we can efficiently reuse the same light subpaths for both VPLs and photons. The resulting algorithm is suitable for GPUs since there are many efficient implementations of VPLs and photon density estimation on GPUs.

### 4 UNIFYING VPLS AND PHOTONS

We explain how to unify rendering with VPLs and photon splatting under the vertex merging formulation [Georgiev et al. 2012]. Table 1 summarizes our notations. The path integral formulation [Veach 1998] models the intensity  $I_p$  of a pixel  $p$  as

$$I_p = \int_{\Omega} f_p(\bar{x}) d\mu(\bar{x}), \quad (1)$$

where  $\bar{x} = \mathbf{x}_0 \dots \mathbf{x}_k$  is a path,  $f_p(\bar{x})$  is the measurement contribution function for the pixel  $p$ , and  $\Omega$  is the set of all possible paths. The measurement contribution function is defined as

$$f_p(\bar{x}) = W_p(\mathbf{x}_0, \mathbf{x}_1) \prod_{i=1}^{k-1} f_r(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}) \prod_{j=0}^{k-1} G(\mathbf{x}_j, \mathbf{x}_{j+1}) L_e(\mathbf{x}_{k-1}, \mathbf{x}_k). \quad (2)$$

for given BRDFs  $f_r$ , the importance function  $W_p$ , and the emission function  $L_e$ . We use the notation  $f_r(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1})$  to define the value of a BRDF  $f_r$  for given directions  $\mathbf{x}_{i+1}$  to  $\mathbf{x}_i$  and  $\mathbf{x}_i$  to  $\mathbf{x}_{i-1}$ . The geometry term  $G(\mathbf{x}_j, \mathbf{x}_{j+1})$  is defined using the visibility term  $V$  as  $G(\mathbf{x}_j, \mathbf{x}_{j+1}) = V(\mathbf{x}_j, \mathbf{x}_{j+1}) \frac{\cos(\theta_j) \cos(\theta_{j+1})}{||\mathbf{x}_j - \mathbf{x}_{j+1}||^2}$  where  $\theta_j$  and  $\theta_{j+1}$  are angles of the segment  $\mathbf{x}_j \mathbf{x}_{j+1}$  to the surface normals

Table 1. Our notations.

Symbol	Description
$\bar{x}$	a path
$I_p(\bar{x}), f_p(\bar{x})$	pixel intensity and a measurement contribution function of the pixel $p$
$G(\mathbf{x}, \mathbf{y})$	a geometry term
$f_r(\mathbf{x}, \mathbf{y}, \mathbf{z})$	a BSDF function at $\mathbf{y}$ location
$\bar{y}, \bar{z}$	a light and an eye subpath
$\mathbf{y}_i, \mathbf{z}_i$	$i$ th vertex of a light and an eye subpath
$f_l(\bar{y}), f_e(\bar{z})$	a measurement contribution function of a light and an eye subpath
$p_l(\bar{y}), p_e(\bar{z})$	PDF of a light and an eye subpath
$C_l(\bar{y}), C_e(\bar{z})$	contribution of light and eye subpath
$\bar{x}'$	an extended path
$\bar{y}'$	an extended light subpath
$\mathbf{y}'_i$	an extended vertex of light subpath
$p(\mathbf{x} \mathbf{y}, \mathbf{z})$	PDF of BSDF sampling $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{z}$
$C_{vpl}(\bar{y}), C_{pm}(\bar{y})$	contribution using VPL and photon
$w_{vpl}(\bar{x}), w_{pm}(\bar{x}')$	a weighting function of VPL and photon
$N_{vpl}, N_{pm}$	a number of VPL and photon light subpaths

at  $\mathbf{x}_j$  and  $\mathbf{x}_{j+1}$ . In the following explanation, we consider paths of a fixed length to simplify the equations, but our method is not limited to this variant.

Suppose that we construct a path  $\bar{x}$  based a length-one eye subpath  $\bar{z} = \mathbf{z}_0 \mathbf{z}_1$  and a light subpath  $\bar{y} = \mathbf{y}_s \dots \mathbf{y}_0$  as  $\bar{x} = \bar{z}\bar{y}$ . The measurement contribution function can be factorized into the eye subpath throughput  $f_e(\bar{z})$ , the light subpath throughput  $f_l(\bar{y})$  as

$$\begin{aligned} f_p(\bar{x}) &= f_e(\bar{z})f_r(\mathbf{z}_0, \mathbf{z}_1, \mathbf{y}_s)G(\mathbf{z}_1, \mathbf{y}_s)f_r(\mathbf{z}_1, \mathbf{y}_s, \mathbf{y}_{s-1})f_l(\bar{y}) \\ f_e(\bar{z}) &= W_p(\mathbf{z}_0, \mathbf{z}_1)G(\mathbf{z}_0, \mathbf{z}_1) \\ f_l(\bar{y}) &= \prod_{i=1}^{s-1} f_r(\mathbf{y}_{i+1}, \mathbf{y}_i, \mathbf{y}_{i-1}) \prod_{j=0}^{s-1} G(\mathbf{y}_j, \mathbf{y}_{j+1})L_e(\mathbf{y}_1, \mathbf{y}_0), \end{aligned} \quad (3)$$

By sampling  $\bar{z}$  and  $\bar{y}$  according to PDFs  $p_e(\bar{z})$  and  $p_l(\bar{y})$ , the Monte Carlo integration of Equation 1 becomes

$$\begin{aligned} I_p &= \mathbb{E} \left[ \frac{f_e(\bar{z})f_r(\mathbf{z}_0, \mathbf{z}_1, \mathbf{y}_s)G(\mathbf{z}_1, \mathbf{y}_s)f_r(\mathbf{z}_1, \mathbf{y}_s, \mathbf{y}_{s-1})f_l(\bar{y})}{p_e(\bar{z})p_l(\bar{y})} \right] \\ &= \mathbb{E} [C_e(\bar{z})f_r(\mathbf{z}_0, \mathbf{z}_1, \mathbf{y}_s)G(\mathbf{z}_1, \mathbf{y}_s)f_r(\mathbf{z}_1, \mathbf{y}_s, \mathbf{y}_{s-1})C_l(\bar{y})] \end{aligned} \quad (4)$$

where we define  $C_e(\bar{z}) \equiv \frac{f_e(\bar{z})}{p_e(\bar{z})}$  and  $C_l(\bar{y}) \equiv \frac{f_l(\bar{y})}{p_l(\bar{y})}$  to denote the contributions from subpaths. A common approach is to use PDFs  $p_e(\bar{z})$  and  $p_l(\bar{y})$  that are proportional to  $f_e(\bar{z})$  and  $f_l(\bar{y})$  respectively. Many-light methods and photon density estimation rely on the same approaches for generating  $\bar{z}$  and  $\bar{y}$ . Thus the contributions  $C_e(\bar{z})$  and  $C_l(\bar{y})$  do not change between two approaches. The difference between these two approaches exists only in the rest of the terms in Equation 4.

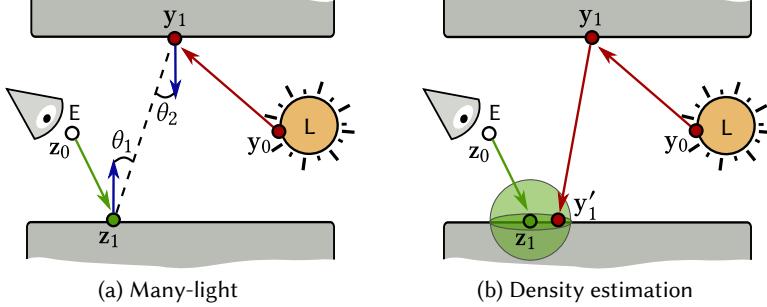


Fig. 3. Light transport for one bounce of indirect lighting. Compare to many-light methods, photon density estimation need an additional vertex  $y'_1$ .

**VPLs.** Rendering with VPLs is based on a two-pass algorithm. In the first pass, we generate a set of light subpath vertices and store them as VPLs. For brevity of explanation, we consider only the last vertex  $y_s$  of each light subpath as a VPL. In the second pass, we trace an eye subpath for each pixel. The set of light subpaths is shared across all the pixels to compute the terms in Equation 4. The contribution of each VPL  $C_{vpl}(\bar{y})$  is

$$C_{vpl}(\bar{y}) = f_r(z_0, z_1, y_s)G(z_1, y_s)f_r(z_1, y_s, y_{s-1})C_l(\bar{y}). \quad (5)$$

While a VPL enables efficient path reuse across all the pixels, Equation 5 has a singularity due to  $G(z_1, y_s)$ . As the eye subpath vertex  $z_1$  gets close to the VPL location  $y_s$ , the geometry term  $G(z_1, y_s)$  becomes infinitely large. This causes a bright splotch in the rendered image. This issue is known as weak singularities of VPLs [Kollig and Keller 2006]. We propose to address this issue by using photon splatting.

**Photons.** Our method is built on the unified light transport frameworks [Georgiev et al. 2012; Hachisuka et al. 2012] which point out that a set of light subpaths can also be used as photons. To see this connection, consider extending a light subpath  $\bar{y}$  by sampling one more vertex  $\bar{y}' = y'_s \bar{y}$  according to the PDF  $p(y'_s|y_s, y_{s-1}) \sim G(y'_s, y_s)f_r(y'_s, y_s, y_{s-1})$  (see Fig. 3). The vertex merging formulation [Georgiev et al. 2012] considers this additional light subpath vertex  $y'_s$  in photon density estimation as a sample for Russian roulette, and merges it to the last eye subpath vertex  $z_1$ .

Under this unified formulation, the measurement contribution of an extended path  $\bar{x}' = \bar{z}\bar{y}'$  remains the same as Equation 3 as

$$f_p(\bar{x}') = f_e(\bar{z})f_r(z_0, z_1, y_s)G(z_1, y_s)f_r(z_1, y_s, y_{s-1})f_l(\bar{y}), \quad (6)$$

because  $y'_s$  is merged to  $z_1$ . The probability density function  $p'_l(\bar{y}')$  becomes

$$p'_l(\bar{y}') \approx p_l(\bar{y})p(y'_s|y_s, y_{s-1})\pi r^2 \quad (7)$$

where  $r$  is the radius of the density estimation kernel and  $p(y'_s|y_s, y_{s+1})\pi r^2$  is the Russian roulette probability under a few assumptions using a uniform kernel [Georgiev et al. 2012]. The contribution of each photon  $C_{pm}(\bar{y}')$  thus becomes

$$C_{pm}(\bar{y}') \approx \frac{f_r(z_0, z_1, y_s)G(z_1, y_s)f_r(z_1, y_s, y_{s-1})}{p(y'_s|y_s, y_{s-1})\pi r^2} C_l(\bar{y}). \quad (8)$$

Since  $p(\mathbf{y}'_s | \mathbf{y}_s, \mathbf{y}_{s-1}) \sim G(\mathbf{y}'_s, \mathbf{y}_s) f_r(\mathbf{y}'_s, \mathbf{y}_s, \mathbf{y}_{s-1})$ , by assuming that  $\mathbf{z}_1 \approx \mathbf{y}'_s$  and a uniform kernel  $K(\mathbf{z}_1, \mathbf{y}'_s) = 1/\pi r^2$  if  $\|\mathbf{z}_1 - \mathbf{y}'_s\| < r$  else 0, we have

$$C_{pm}(\bar{y}') \approx f_r(\mathbf{z}_0, \mathbf{z}_1, \mathbf{y}_s) K(\mathbf{z}_1, \mathbf{y}'_s) \Phi C_l(\bar{y}), \quad (9)$$

where  $\Phi$  is the normalization constant of  $p(\mathbf{y}'_s | \mathbf{y}_s, \mathbf{y}_{s-1})$  which is obtained by solving the integrand  $\Phi = \int_S G(\mathbf{y}_s, \mathbf{y}'_s) f_r(\mathbf{y}'_s, \mathbf{y}_s, \mathbf{y}_{s-1}) d\mathbf{y}'_s$ . Because the uniform kernel could be evaluated to zero, naively summing the contribution of all the photons results in  $C_{pm}(\bar{y}') = 0$  most of the time. A common approach is to gather neighboring photons using nearest neighbor search such that  $\|\mathbf{z}_1 - \mathbf{y}'_s\| < r$ .

Comparing Equation 5 and Equation 9 reveals the benefit of using photons over VPLs: the contribution of each photon does not contain any geometry term. If  $\mathbf{z}_1$  and  $\mathbf{y}_s$  are close, a VPL suffers from the singularity due to  $G(\mathbf{z}_1, \mathbf{y}_s)$ . In photon density estimation, we not only remove  $G(\mathbf{z}_1, \mathbf{y}_s)$  and  $f_r(\mathbf{z}_1, \mathbf{y}_s, \mathbf{y}_{s-1})$ , but also are likely to have  $\mathbf{y}'_s$  with  $\|\mathbf{z}_1 - \mathbf{y}'_s\| < r$ , if  $\mathbf{z}_1$  and  $\mathbf{y}_s$  are already close. On the other hand, if  $\mathbf{z}_1$  and  $\mathbf{y}_s$  are far, a VPL works fine, but it is unlikely to generate  $\mathbf{y}'_s$  near  $\mathbf{z}_1$  by tracing a ray from  $\mathbf{y}_s$ . VPLs and photons thus have complementary properties, making its combination meaningful. Another important aspect is that they share a lot of terms such as  $C_e(\bar{z})$ ,  $C_l(\bar{y})$ ,  $p_e(\bar{z})$ , and  $p_l(\bar{y})$ . We build upon this observation and combine VPLs and photons using multiple importance sampling.

**Combining VPLs and Photons.** Now that VPLs and photons are formulated under the same framework, it is straightforward to combine them using multiple importance sampling [Veach and Guibas 1995]. We consider two paths  $\bar{x}_i = \bar{z}\bar{y}_i$  and  $\bar{x}'_i = \bar{z}\bar{y}'_i$  that corresponds to the same light transport path using VPLs and photons. The PDFs in this case are

$$p_{vpl}(\bar{x}_i) = p_e(\bar{z}) p_l(\bar{y}_i) \quad (10)$$

$$p_{pm}(\bar{x}'_i) \approx p_e(\bar{z}) p_l(\bar{y}_i) p(\mathbf{y}'_{s,i} | \mathbf{y}_{s,i}, \mathbf{y}_{s-1,i}) \pi r^2 \quad (11)$$

Using the balance heuristic, we can compute weight for photon splatting as

$$w_{pm}(\bar{x}'_i) = \frac{p(\mathbf{y}'_{s,i} | \mathbf{y}_{s,i}, \mathbf{y}_{s-1,i}) \pi r^2 N_{pm}}{p(\mathbf{y}'_{s,i} | \mathbf{y}_{s,i}, \mathbf{y}_{s-1,i}) \pi r^2 N_{pm} + N_{vpl}} \quad (12)$$

where  $N_{pm}$  and  $N_{vpl}$  are the numbers of light paths used in photon splatting and rendering with VPLs respectively. We thus have a weighted contribution for photon density estimation

$$I_{p,pm} = \mathbb{E} \left[ \frac{1}{N_{pm}} \sum_{i=1}^{N_{pm}} w_{pm}(\bar{x}'_i) C_e(\bar{z}) C_{pm}(\bar{y}'_i) \right]. \quad (13)$$

Since we only have two sampling strategy for a given path, the weight for VPLs is simply  $w_{vpl}(\bar{x}_i) = 1 - w_{pm}(\bar{x}'_i)$ . For rendering with VPLs, we have

$$I_{p,vpl} = \mathbb{E} \left[ \frac{1}{N_{vpl}} \sum_{i=1}^{N_{vpl}} w_{vpl}(\bar{x}_i) C_e(\bar{z}) C_{vpl}(\bar{y}_i) \right]. \quad (14)$$

**Discussion.** Kollig and Keller [2006] reformulated geometry clamping as another heuristic of multiple importance sampling [Veach and Guibas 1995]. Their heuristic is essentially  $w_{vpl}(\bar{x}) = \min(1, b/G(\mathbf{z}_1, \mathbf{y}_s))$  where  $b$  is a user-defined bounding value. Their heuristic can efficiently reduce the amount of computation. In our method, we do not need to discard the samples as photon splatting is not expensive. While bidirectional lightcuts [Walter et al. 2012] also uses multiple importance sampling to avoid the singularities of VPLs, due to the use of bidirectional estimators, it requires additional path sampling from the eye. In the results, we show that photon splatting in our method can be used to compensate lost energy without additional path sampling.

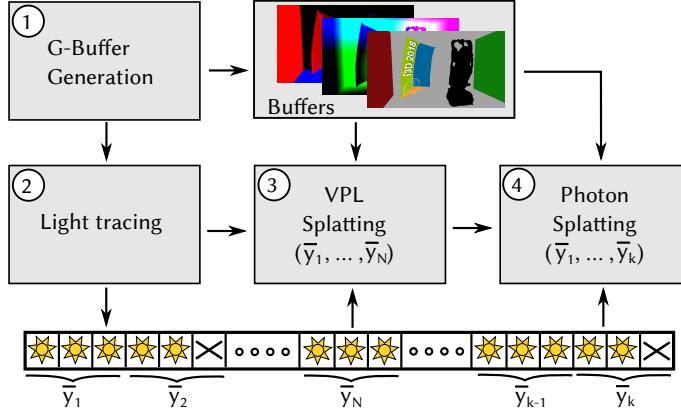


Fig. 4. Pipeline overview of our implementation.

## 5 RESULTS

**Implementation.** We utilized OpenGL rasterization to generate a shading point and its information such as position, normal, and material parameters, and store them in a G-Buffer [Deering et al. 1988]. OptiX [Parker et al. 2010] was used for light subpaths sampling. We preallocated  $depth + 1$  light vertices for every  $\max(N_{pm}, N_{vpl})$  light paths to store all information needed. Some preallocated vertices might not contain any information due to early path termination. These vertices are discarded during rendering process and might cause some inefficiency. One can potentially use atomic counter similar to light vertex cache [Davidović et al. 2014] technique to solve the problem.

For all the results in the paper, we used ray tracing to compute the visibility of each VPL. One could alternatively use imperfect shadow maps [Ritschel et al. 2008] to improve the performance of the VPL part if the approximation is acceptable. We used image space photon mapping (ISPM) for photon splatting [McGuire and Luebke 2009]. The results were generated with an Intel i7-6850K and an NVIDIA GeForce GTX 980 Ti. The error metrics that were used are sMAPE =  $(2/n) \sum |I - R| / (I + R)$  and relative MSE (relMSE) =  $(1/n) \sum (I - R)^2 / (R^2 + 0.001)$ . We only use the first metric to compare against images rendered with VPLs without clamping because the sMAPE metric is less sensitive to singularities caused by VPLs. The reference images have been rendered by Mitsuba [Jakob 2010] with several days computation using regular path tracing. We compare our method with VPL without clamping, VSL, ISPM, and GPU path tracing with OptiX. Our GPU path tracing uses next event estimation with the balance heuristic to combines samples from BSDF sampling and light sampling. To achieve fair comparisons, we used the same number of photons per iteration for the comparison between ours and ISPM, and we used the same number of VPLs per iteration for the comparison between ours, VPL, and VSL.

Table 2. Time spent for path sampling, gathering VPLs, and splatting photons per light subpath in our method.

Scene	Light tracing	VPL	Splatting photon
STAIRS	$6.32 \times 10^{-5}$ ms	2.83 ms	$1.71 \times 10^{-4}$ ms
LIVING ROOM	$6.54 \times 10^{-5}$ ms	2.66 ms	$1.65 \times 10^{-4}$ ms
CONFERENCE	$9.21 \times 10^{-5}$ ms	5.26 ms	$1.87 \times 10^{-4}$ ms
BUDDHA	$8.39 \times 10^{-5}$ ms	8.39 ms	$2.09 \times 10^{-4}$ ms

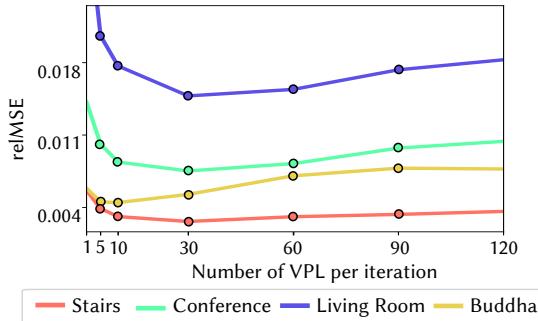


Fig. 5. Relative MSE for different scenes with different numbers of VPLs for 15 seconds of rendering. The plot shows that the efficiency of our method does not vary across different scenes, except when the number of VPLs is too small or too large.

**Influence of parameters.** The main parameter of our method is the ratio between the number of VPLs  $N_{vpl}$  and the number of photons  $N_{pm}$  per iteration. One might want to simply reuse all the light subpaths by setting  $N_{vpl} = N_{pm}$ . However, this choice is not necessarily optimal since rendering with VPLs and photon splatting have significantly different performance per light subpath. Table 2 shows that splatting of photons is multiple of magnitudes faster than gathering VPLs per light subpath. It is thus generally a good idea to set  $N_{vpl}$  significantly smaller than  $N_{pm}$  to balance the rendering times of algorithms better. Based on the numerical experiment shown in Fig. 5, we set  $N_{pm} = 300000$  and  $N_{vpl} = 30$  regardless of the scene.

In order to reuse as much computation as possible between VPLs and photons, we splat all of sampled light subpath vertices as photons and reuse a fraction of them as VPLs. Since  $N_{vpl}$  is significantly smaller than  $N_{pm}$ , correlation between VPLs and photon splatting is usually negligible unlike bidirectional estimators in bidirectional lightcuts [Walter et al. 2012].

We set the kernel radius of photon splatting to be 0.3% of the radius of the scene bounding sphere. For VSL, we set its radius to 5% of the radius of the scene bounding sphere. We carefully chose radius for all the methods so that each method provides the best relative mean square error.

**Scenes.** The *Stairs* scene (Fig. 1) has edges and corners that can easily cause problems for many-light methods. Clamping the geometry term reduces the singularities issue at the cost of a darker image. Our method does not have this problem and produces an image that is close to the reference. This scene highlights a typical use case of our method such as light transport simulation for architectures.

The *Conference* scene (Fig. 6, left column) is challenging for many-light methods due to the complex geometry and the glossy floor. It consists of 16 light sources which make path tracing inefficient due to explicit light connections with a large number of light sources. Photon splatting also has a similar level of noise to path tracing in this scene. Rendering with VPLs again has a problem in the curtain area due to the singularities. While using VSLs reduces the singularities in this area, but also produces the darker result around the corners due to its approximation. Compare to other methods, ours produces a more accurate result on the curtain at the cost of little additional noise. In the pillar region, the visible noise is due to indirect VPLs and photons have a hard time reaching this scene location. Fig. 7 shows the same scene with shorter rendering time.

The *Living Room* scene (Fig. 6, middle column) showcases strong indirect illumination from the ceiling. Path tracing produces a noisy result because next-event estimation does not work well in this scene. While this scene mainly uses diffuse materials, which is suitable for rendering with



Fig. 6. Equal-time comparisons in 15 seconds between GPU path tracing, image space photon mapping (ISPM) [McGuire and Luebke 2009], virtual point light (VPL) [Keller 1997], virtual spherical light (VSL) [Hašan et al. 2009], and our method. Path tracing and ISPM usually exhibit a visible level of noise. In comparison, VPL produces smoother images but suffers from the singularities. VSL solves this problem but introduces a visible level of bias. Our method provides results close to the reference.

VPLs, photon splatting yields lower relative error since the image rendered with VPLs suffers from the singularities. The rendered image with VSL shows visible artifacts due to bias. With the combination of VPLs and photons, our method alleviates the issues of singularities and bias at the same time.

The *Buddha* scene (Fig. 6, right column) features complex glossy light transport. This scene contains multiple spotlights for lighting up the Buddha status and the flag. Path tracing again results in a noisy image due to the presence of glossy light transport. Photon splatting performs well in this scene, except in the area where diffuse light transport dominates. Rendering with VPLs and VSLs are not efficient in computing glossy to glossy light transport on the statue. Our method provides the best result as it combines density estimation and VPL efficiently.

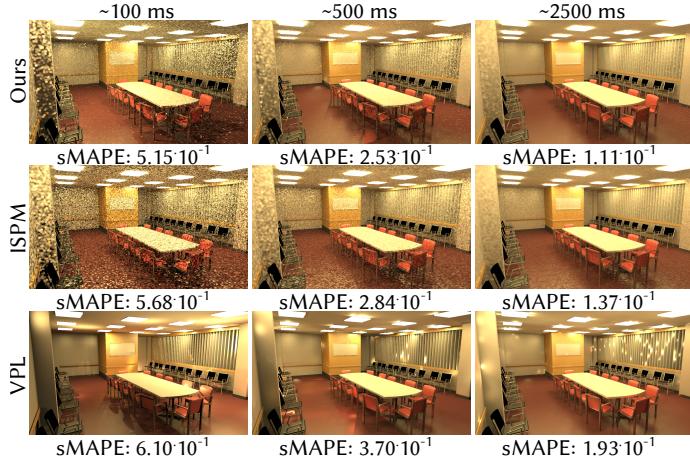


Fig. 7. After only 100 ms of rendering time, rendering with VPLs exhibits low frequency artifacts due to a low sample count. Compared to photon density estimation, our method shows much less noise since VPLs efficiently render far illumination. Having longer rendering time makes our method converges to a smooth result overall.

**Progressive rendering.** One feature of our method is that the photon splatting part can be turned into a consistent algorithm by the progressive formulation [Hachisuka et al. 2008; Knaus and Zwicker 2011]. This makes the entire algorithm consistent since it is the same approach as unified light transport frameworks [Georgiev et al. 2012; Hachisuka et al. 2012]. We only need to reduce the kernel size at every iteration progressively.

We compared the progressive variant of our method with progressive VPL clamping [Davidovič et al. 2012]. For each scene, we searched the relaxation parameter to minimize relative MSE. We also compared with a progressive variant of VSL [Novák et al. 2012a]. Fig. 8 shows the relative mean square error evolution of rendering technique in the test scenes. VSL and clamped VPL have a similar amount of error. Our method starts with a lower error at the beginning and provides more accurate results in all the case.

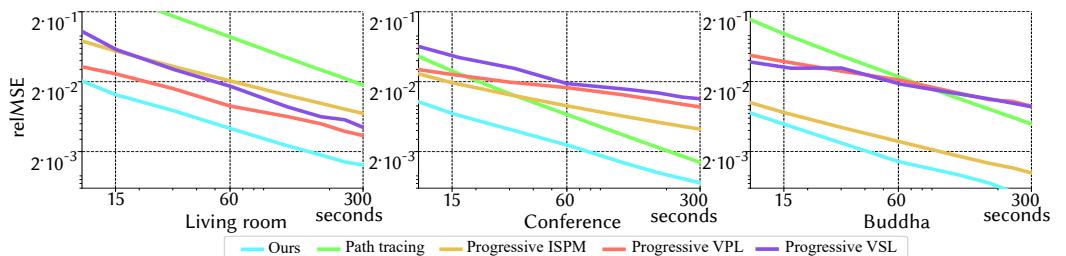


Fig. 8. Relative MSE for our method, GPU path tracing, Progressive ISPM [McGuire and Luebke 2009], Progressive VPL [Davidovič et al. 2012], and Progressive VSL [Hašan et al. 2009]. For all the scenes shown in Fig. 6, our method provides the most accurate results.

## 6 LIMITATIONS AND FUTURE WORK

**Glossy surfaces.** Since our method does not continue tracing secondary rays from the eye, there is no importance sampling of BRDFs at the shading point. Previous works [Hachisuka and Jensen 2009; Kollig and Keller 2006; Walter et al. 2012] show that tracing such rays are beneficial for scenes with glossy surfaces. More general algorithms for photon density estimation on GPUs [Hachisuka and Jensen 2010] is necessary to implement such a variant of our method on GPUs.

**Scalability.** While our current implementation gathers all the VPLs without any clustering, it is potentially useful to employ scalable algorithms such as lightcuts [Walter et al. 2005] to amortize the cost of gathering VPLs. However, since we weight the contribution of each VPL via multiple importance sampling, finding a tight error bound is no longer trivial. Clustering of VPLs also introduces another source of error, which would require special care in the weighting strategy. While our method generally performs well without such clustering, this direction is interesting and surely leads to more efficient path reuse.

**Volumes.** Many-light methods have been successfully applied to volumes [Novák et al. 2012a,b]. Likewise, photon density estimation with higher dimensional primitives [Jarosz et al. 2011] also has excellent performance in volumes. We believe that our approach of combining many-light methods and photon density estimation is promising for volumes as well.

## 7 CONCLUSION

We presented an efficient rendering method that combines rendering with VPLs and photon splatting under the unified formulation. The intuition behind this combination is that, when a VPL is too close to a shading point and suffers from singularities, this exact same VPL can be used as a neighboring photon in photon density estimation. We showed that this natural combination with photon splatting indeed efficiently alleviates the singularities of VPLs. Moreover, since both rendering with VPLs and rendering with photon splatting run efficiently on GPUs, our combined method is also suitable for GPUs. The results demonstrate that our method achieves generally better performance compared to other methods based only on many-lights or photons. We believe that our method is particularly useful to quickly render scenes with mostly diffuse materials, which should open up many practical applications in the industry.

## ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for thoroughly reviewing our submission. Furthermore, We thank Anat Grynberg and Greg Ward (the conference room), Stanford Computer Graphics Laboratory (happy buddha), blendswap.com artists "Wayne"(the stairs) and "cenobi"(the living room). This project is partially funded by Japanese Government (Monbukagakusho - MEXT) Scholarship and JSPS KAKENHI Grant Number 15H05308.

## REFERENCES

- Carsten Dachsbacher, Jaroslav Krivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák. 2014. Scalable Realistic Rendering with Many-Light Methods. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 88–104.
- Tomáš Davidovič, Iliyan Georgiev, and Philipp Slusallek. 2012. Progressive lightcuts for GPU. In *ACM SIGGRAPH 2012 Talks*. ACM, 1.
- Tomáš Davidovič, Jaroslav Krivánek, Miloš Hašan, and Philipp Slusallek. 2014. Progressive light transport simulation on the GPU: Survey and improvements. *ACM Transactions on Graphics (TOG)* 33, 3 (2014), 29.
- Tomáš Davidovič, Jaroslav Krivánek, Miloš Hašan, Philipp Slusallek, and Kavita Bala. 2010. Combining global and local virtual lights for detailed glossy illumination. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 143.
- Michael Deering, Stephanie Winner, Bic Schediwy, Chris Duffy, and Neil Hunt. 1988. The Triangle Processor and Normal Vector Shader: A VLSI System for High Performance Graphics. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 21–30.

- Thomas Engelhardt, Jan Novák, Thorsten-W Schmidt, and Carsten Dachsbacher. 2012. Approximate bias compensation for rendering scenes with heterogeneous participating media. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 2145–2154.
- Iliyan Georgiev, Jaroslav Krivánek, Tomas Davidovic, and Philipp Slusallek. 2012. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.* 31, 6 (2012), 192–1.
- Toshiya Hachisuka and Henrik Wann Jensen. 2009. Stochastic Progressive Photon Mapping. *ACM Trans. Graph.* 28, 5, Article 141 (Dec. 2009), 8 pages.
- Toshiya Hachisuka and Henrik Wann Jensen. 2010. Parallel progressive photon mapping on GPUs. In *ACM SIGGRAPH ASIA 2010 Sketches*. ACM, 54.
- Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. 2008. Progressive photon mapping. *ACM Transactions on Graphics (TOG)* 27, 5 (2008), 130.
- Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A path space extension for robust light transport simulation. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 191.
- Miloš Hašan, Jaroslav Krivánek, Bruce Walter, and Kavita Bala. 2009. Virtual spherical lights for many-light rendering of glossy scenes. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 143.
- Miloš Hašan, Fabio Pellacini, and Kavita Bala. 2007. Matrix row-column sampling for the many-light problem. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 26.
- Wenzel Jakob. 2010. Mitsuba renderer. (2010).
- Wojciech Jarosz, Derek Nowrouzezahrai, Robert Thomas, Peter-Pike Sloan, and Matthias Zwicker. 2011. Progressive photon beams. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 181.
- Henrik Wann Jensen. 1996. Global illumination using photon maps. *Rendering techniques* 96 (1996), 21–30.
- Alexander Keller. 1997. Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 49–56.
- Claude Knous and Matthias Zwicker. 2011. Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics (TOG)* 30, 3 (2011), 25.
- Thomas Kollig and Alexander Keller. 2006. Illumination in the presence of weak singularities. In *Monte Carlo and Quasi-Monte Carlo Methods 2004*. Springer, 245–257.
- Fabien Lavingotte and Mathias Paulin. 2003. Scalable photon splatting for global illumination. In *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*. ACM, 203–ff.
- Morgan McGuire and David Luebke. 2009. Hardware-accelerated global illumination by image space photon mapping. In *Proceedings of the Conference on High Performance Graphics 2009*. ACM, 77–89.
- Jan Novák, Thomas Engelhardt, and Carsten Dachsbacher. 2011. Screen-space bias compensation for interactive high-quality global illumination with virtual point lights. In *Symposium on Interactive 3D Graphics and Games*. ACM, 119–124.
- Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. 2012a. Progressive virtual beam lights. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1407–1413.
- Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. 2012b. Virtual ray lights for rendering scenes with participating media. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 60.
- Jiawei Ou and Fabio Pellacini. 2011. LightSlice: Matrix Slice Sampling for the Many-lights Problem. In *Proceedings of the 2011 SIGGRAPH Asia Conference (SA '11)*. ACM, New York, NY, USA, Article 179, 8 pages.
- Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, et al. 2010. Optix: a general purpose ray tracing engine. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 66.
- Matthias Raab, Daniel Seibert, and Alexander Keller. 2008. Unbiased global illumination with participating media. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*. Springer, 591–605.
- T. Ritschel, T. Grosch, M. H. Kim, H.-P. Seidel, C. Dachsbacher, and J. Kautz. 2008. Imperfect Shadow Maps for Efficient Computation of Indirect Illumination. *ACM Trans. Graph.* 27, 5, Article 129 (Dec. 2008), 8 pages.
- Wolfgang Stürzlinger and Rui Bastos. 1997. Interactive rendering of globally illuminated glossy scenes. In *Rendering Techniques'97*. Springer, 93–102.
- Eric Veach. 1998. *Robust monte carlo methods for light transport simulation*. Stanford University Stanford.
- Eric Veach and Leonidas J Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 419–428.
- Bruce Walter, Adam Arbree, Kavita Bala, and Donald P Greenberg. 2006. Multidimensional lightcuts. In *ACM Transactions on Graphics (TOG)*, Vol. 25. ACM, 1081–1088.
- Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P Greenberg. 2005. Lightcuts: a scalable approach to illumination. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 1098–1107.
- Bruce Walter, Pramook Khungurn, and Kavita Bala. 2012. Bidirectional lightcuts. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 59.