Bayesian online regression for adaptive direct illumination sampling

PETR VÉVODA*, Charles University, Prague and Render Legion, a. s.
IVO KONDAPANENI*, Charles University, Prague
JAROSLAV KŘIVÁNEK, Charles University, Prague and Render Legion, a. s.

No learning
RMSE 0.0031 (0.0014 for DI only)

Our method
RMSE 0.0012, *6.7x speedup* (0.000062, *510x* for DI only)

No learning

Our method

Fig. 1. Equal-time comparison (60 s) of path-traced global illumination solutions computed using our learning-based direct illumination sampling method (right) and a baseline sampling method without learning (left). While both methods start off by sampling lights proportionally to rough estimates of their unoccluded contribution, our method progressively incorporates information about their actual contributions, including visibility, dramatically reducing image variance.

Direct illumination calculation is an important component of any physically-based renderer with a substantial impact on the overall performance. We present a novel adaptive solution for unbiased Monte Carlo direct illumination sampling, based on online learning of the light selection probability distributions. Our main contribution is a formulation of the learning process as Bayesian regression, based on a new, specifically designed statistical model of direct illumination. The net result is a set of regularization strategies to prevent over-fitting and ensure robustness even in early stages of calculation, when the observed information is sparse. The regression model captures spatial variation of illumination, which enables aggregating statistics over relatively large scene regions and, in turn, ensures a fast learning rate. We make the method scalable by adopting a light clustering strategy from the Lightcuts method, and further reduce variance through the use of control variates. As a main design feature, the resulting algorithm is virtually free of any preprocessing, which enables its use for interactive progressive rendering, while the online learning still enables super-linear convergence.

CCS Concepts: • **Computing methodologies** → **Rendering**; *Visibility*; *Machine learning*;

Authors' addresses: Petr Vévoda, Charles University, Prague, Malostranské náměstí 25, Prague, 11800, Render Legion, a. s. Karlovo náměstí 288/17, Prague, 12000, petrvevoda@seznam.cz; Ivo Kondapaneni, Charles University, Prague, Malostranské náměstí 25, Prague, 11800, ivo.kondapaneni@gmail.com; Jaroslav Křivánek, Charles University, Prague, Malostranské náměstí 25, Prague, 11800, Render Legion, a. s. Karlovo náměstí 288/17, Prague, 12000, jaroslav.krivanek@mff.cuni.cz.

*Petr Vévoda and Ivo Kondapaneni share the first authorship of this work.

0730-0301/2018/8-ART125 $15.00
https://doi.org/10.1145/3197517.3201340

Additional Key Words and Phrases: direct illumination, adaptive sampling, visibility, learning.

ACM Reference Format:
Petr Vévoda, Ivo Kondapaneni, and Jaroslav Křivánek. 2018. Bayesian online regression for adaptive direct illumination sampling. *ACM Trans. Graph.* 37, 4, Article 125 (August 2018), 12 pages. https://doi.org/10.1145/3197517.3201340

1 INTRODUCTION

Realistic rendering today is almost entirely based on Monte Carlo (MC) methods. The *indirect* illumination component has traditionally been held responsible for the undesirable image noise produced by such algorithms, which is probably why the *direct* illumination has received disproportionately less attention in research. However, many scenes in digital production feature complex lighting setups, and practical experience shows that it is often direct illumination that is responsible for the majority of image noise.

In this paper, we aim at unbiased direct illumination estimation for MC renderers. Specifically, we address the problem of randomly choosing an appropriate light source for a given scene location, so that variance of the direct illumination estimator is minimized. This could be achieved by choosing lights with probability proportional to their respective contributions, but these are unknown at the outset, they are costly to evaluate and difficult to predict. This is true especially due to the visibility, since it can be discontinuous and its evaluation involves expensive ray casting.

One possible solution would involve constructing the light sampling distributions in a preprocessing step [Georgiev et al. 2012]. However, long preprocessing disqualifies any form of *interactive rendering* – a crucial feature of any modern progressive renderer, a feature that we consider a hard constraint in our work. Such preprocessing can be avoided by learning from the observed samples

during rendering, and our work follows this path. This is hardly a new idea in the general MC context and it has been used for direct illumination sampling [Donikian et al. 2006]. Unresolved challenges remain, though, such as how to ensure robustness, especially in the early stages of rendering, when the collected data is sparse.

The above concerns are common to most adaptive MC methods, and we address them through a systematic treatment based on Bayesian modeling. We formulate the learning process as maximum a posteriori (MAP) regression based on a new statistical model of direct illumination that explicitly models the effect of visibility. The prior distribution is modeled using estimates of lights' unoccluded contributions computed at a small cost. The net result of this formulation are regularization strategies that prevent overfitting and enable meaningful use of the collected samples even in early stages of rendering. Our regression model captures spatial variation of illumination, which enables aggregating statistics over relatively large spatial regions, and, in turn, ensures a fast learning rate.

Our second main contribution consists in showing that sampling lights proportionately to their expected contribution can in fact be far from optimal. The reason is the additional variance due to computing illumination from each individual light source, once it has been selected. We derive the optimal sampling strategy for such *nested estimators* and apply it to the light selection problem.

Finally, to achieve a scalable solution we build upon the light clustering strategies from previous work [Walter et al. 2005; Wang and Akerlund 2009], and we further reduce variance by using the gathered statistics to construct a control variate [Kalos and Whitlock 1986]. The resulting algorithm is virtually free of any preprocessing, which enables its use in an interactive progressive renderer, while the online learning enables superlinear convergence, especially in the early stages of rendering. Fig. 1 shows an example result.

2 PREVIOUS WORK

Direct illumination computation. Different ways to improve the performance of direct illumination computation have been explored. One idea is to speed up the evaluation of a single light's contribution, the cost of which is often dominated by determining its visibility. This could be achieved by skipping visibility tests for lights that contribute weakly [Ward 1994], clipping polygonal area lights [Hart et al. 1999], using a visibility oracle based on a photon map [Jensen and Christensen 1995] or learning during rendering [Fernandez et al. 2002]. Wald and Benthin [2003] cull lights based on a path tracing prepass. Random skipping of visibility tests [Billen et al. 2013] or their caching [Popov et al. 2013] have been likewise explored.

Reducing the cost of a single light evaluation cannot reduce the linear complexity of direct illumination computation, which becomes a bottleneck when lights are many. Paquette et al. [1998] and Walter et al. [2005] propose to hierarchically cluster lights into a tree and then use adaptively constructed tree cuts to approximate direct illumination. Both methods scale well but this comes at the expense of some bias. As a follow-up, methods by Walter et al. [2006] and Bus et al. [2015] further reduce the number of scene shading points for which the direct illumination computation is carried out by additionally clustering the shading points.

We address random light selection in a MC renderer. In this context, Shirley et al. [1996] pioneered the idea of designing light selection probabilities based on expected lights' contributions, though they only used a rather crude classification into 'important' and 'unimportant' lights. Wang and Akerlund [2009] sample lights proportionally to the product of a contribution estimate and surface reflectance. The method handles many lights by clustering, an idea we use in our work and extend it with online optimization of sampling distributions. Sampling distributions can also be obtained in a preprocess [Georgiev et al. 2012; Wu and Chuang 2013], but this approach disqualifies any form of interactive rendering. Finally, Donikian et al. [2006] learn a sampling distribution from samples obtained during the rendering, just as we do. The method combines several distributions in an ad hoc manner, which limits its robustness and reliability, as we demonstrate in our results. We show that a theoretically funded Bayesian treatment of adaptive sampling yields substantial improvements in robustness and overall efficiency.

Bayesian modeling in rendering. Bayesian modeling is a widespread methodology in computer vision and graphics, so we only review works closely related to MC rendering. Boughida and Boubekeur [2017] use NL-Bayes image denoising [Lebrun et al. 2013] in the context of MC simulation as a post-processing filter. Brouillat et al. [2009] and Marques et al. [2013] pioneered the use of Bayesian Monte Carlo (BMC) [Rasmussen and Ghahramani 2003] in light transport simulation. While theoretically sound, the BMC methodology comes with some important computational overhead. In contrast, we keep the efficient classic, frequentist MC approach and apply Bayesian modeling to optimize our sampling distributions. This approach was also taken by Vorba et al. [2014], who employ a maximum a posteriori (MAP) formulation to regularize training of parametric mixture models for optimized indirect illumination sampling. Our work uses a MAP formulation of spatial regression so as to obtain robust direct illumination estimates across the scene.

Adaptive sampling. Literature on adaptive sampling in both general MC [Kalos and Whitlock 1986] and in rendering is wide and we only mention some more recent work. One impactful theoretical idea has been population Monte Carlo (PMC) [Cappé et al. 2004], which can, among other, be used to optimize sampling distributions represented by mixture models [Cappé et al. 2008; Douc and Guillin 2007]. Adaptive multiple importance sampling (AMIS) [Cornuet et al. 2009] extends the adaptation idea to multiple importance sampling [Veach 1997], whereas adaptive population importance sampling (APIS) [Martino et al. 2015] attempts to exploit the strong points of PMC or AMIS. PMC has been applied in rendering [Fan et al. 2007; Lai et al. 2007], but the benefits are not large. Our work differs from PMC by the lack of any resampling step which would require storing individual samples.

Path guiding. Methods that build models of incoming illumination specific to a one particular scene and use them for importance sampling have become known as *path guiding*. These methods perform either density estimation from particles obtained in a preprocessing step [Budge et al. 2008; Hey and Purgathofer 2002; Jensen 1995; Vorba et al. 2014] or they derive the importance density through regression modeling [Lafortune and Willems 1995; Müller et al. 2017;

Pegoraro et al. 2008]. Our method is orthogonal to guiding methods since it addresses sampling of *direct* illumination. In fact, it could be incorporated into existing guiding approaches based on regression.

3 OVERVIEW

Direct illumination estimator. Our goal is to compute the reflected radiance L due to direct illumination at a shading point $\mathbf{x}$ as seen from a direction ω. It is defined as an integral over all points $\mathbf{y}$ on the surface A of all scene light sources

$$L(\mathbf{x}, \omega) = \int_A F(\mathbf{y} \to \mathbf{x} \to \omega)\, \mathrm{d}\mathbf{y}, \tag{1}$$

where the integrand equals

$$F(\mathbf{y} \to \mathbf{x} \to \omega) = L_\mathrm{e}(\mathbf{y} \to \mathbf{x}) B(\mathbf{y} \to \mathbf{x} \to \omega) V(\mathbf{y} \leftrightarrow \mathbf{x}) G(\mathbf{y} \leftrightarrow \mathbf{x}). \tag{2}$$

$L_\mathrm{e}(\mathbf{y} \to \mathbf{x})$ is the radiance emitted from $\mathbf{y}$ toward $\mathbf{x}$, $B(\mathbf{y} \to \mathbf{x} \to \omega)$ is the BRDF describing the surface reflectance at $\mathbf{x}$, and $V(\mathbf{y} \leftrightarrow \mathbf{x})$ is the binary visibility function returning 1 if $\mathbf{y}$ is visible from $\mathbf{x}$ and 0 otherwise. The geometry factor $G(\mathbf{y} \leftrightarrow \mathbf{x})$ equals to $\frac{\cos\theta_\mathbf{y} \cos\theta_\mathbf{x}}{d^2(\mathbf{y},\mathbf{x})}$, where $d(\mathbf{y}, \mathbf{x})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ and $\cos\theta_\mathbf{y} = \mathbf{n}_\mathbf{y} \cdot \frac{\mathbf{x}-\mathbf{y}}{d(\mathbf{y},\mathbf{x})}$, $\cos\theta_\mathbf{x} = \mathbf{n}_\mathbf{x} \cdot \frac{\mathbf{y}-\mathbf{x}}{d(\mathbf{y},\mathbf{x})}$ with $\mathbf{n}_\mathbf{y}$, $\mathbf{n}_\mathbf{x}$ being the unit surface normal at $\mathbf{y}$ and $\mathbf{x}$, respectively.

A Monte Carlo estimator for the integral (1) is given by

$$\langle L(\mathbf{x}, \omega) \rangle = \frac{F(\mathbf{y} \to \mathbf{x} \to \omega)}{p(\mathbf{y}|\mathbf{x}, \omega)}, \tag{3}$$

where $p(\mathbf{y}|\mathbf{x}, \omega)$ denotes the pdf of sampling the light point $\mathbf{y}$ from the shading point $\mathbf{x}$ given the viewing direction ω. The better the pdf approximates the integrand, the lower the variance, with the pdf directly proportional to the integrand yielding zero variance.

Light sampling. We seek a practical approximation to the ideal pdf described above. We follow a standard approach for generating a light sample $\mathbf{y}$, where one first selects a light source, and then samples a point on that light [Pharr et al. 2016]. To ensure good scalability with many lights, we additionally employ adaptive light clustering: each point $\mathbf{x}$ in the scene has an associated set C of light clusters c. In this setup, sampling the light point $\mathbf{y}$ in the estimator (3) breaks down into the following three steps:

(1) Select a light cluster $c \in C$ with the probability $P(c|\mathbf{x})$,[1]
(2) Select a light $l \in c$ with the probability $P(l|c)$ proportional to its flux, i.e. $P(l|c) = \Phi_l / \sum_{l' \in c} \Phi_{l'}$,
(3) Select a point $\mathbf{y} \in l$ with the pdf $p(\mathbf{y}|l, \omega)$ using standard techniques [Pharr et al. 2016; Shirley et al. 1996].

The resulting pdf $p(\mathbf{y}|\mathbf{x}, \omega)$ is then obtained as $P(c|\mathbf{x})\, P(l|c)\, p(\mathbf{y}|l, \omega)$.

Adaptive cluster sampling. Our main contribution consists in a new adaptive method for constructing the cluster sampling distribution $P(c|\mathbf{x})$ used in Step (1). To this end, we first derive, in Sec. 4, the optimal distribution for cluster selection in presence of *variance due to nested MC estimation*, i.e. illumination evaluation within each cluster corresponding to Steps (2) and (3). Second, we devise a Bayesian methodology to learn such a distribution in a progressive manner (Sec. 5). For that purpose, we design a statistical MAP regression model of cluster contribution and visibility. The model is initialized by conservative cluster contribution estimates, which embody our prior knowledge. It is then updated on the fly during rendering using the calculated (observed) light contributions.

We do not use learning for sampling the point $\mathbf{y}$ on an individual light in Step (3), since techniques tailored to different kinds of light geometries provide close-to-optimal solutions [Gamito 2016; Shirley et al. 1996]. Furthermore, we design our cluster sampling distributions to be view independent: we omit the BRDF factor and we drop the dependency on the view direction ω in most equations. This is motivated by practical considerations of a production renderer, where reflectance can be defined by arbitrarily complex shaders, often given as a black-box. We discuss the above decisions in Sec. 8.

Light clustering and scene partitioning. Our light clustering approach is inspired by Lightcuts [Walter et al. 2005]. Similar to Wang and Akerlund [2009], we use the clusters for light selection, as opposed to using them directly as illumination estimates. As a result, the clustering affects the estimator variance, not a systematic image error, and hence it can be rather coarse.

In a preprocessing step, we first hierarchically cluster the lights into a binary *light tree* in a similar way to Lightcuts. During rendering, the light tree then serves for finding light clusterings C, represented as a cut in the light tree. Unlike in the original Lightcuts algorithm, where lights are clustered for each shading point on-the-fly, we generate and cache light clusterings for entire scene regions. Such persistent clusterings are necessary to keep the statistics for updating the cluster sampling distributions. The scene is therefore divided into disjoint spatial regions, and each region has an associated light clustering, represented as a light cut. The light cut for a scene region is created on demand, the first time direct illumination calculation is carried out in that region. In scenes with a moderate light count, the clusters usually correspond to the individual lights, and our adaptive algorithm then samples the lights themselves.

As in Lighcuts, the cut construction starts at the root and repeatedly replaces the cluster with the highest estimated contribution by its two children, until the estimated cluster contribution falls below ϵ-fraction of the estimated contribution of the entire cut (we use $\epsilon = 0.1$ and limit the cut size to 100 in all our results). Calculation of the cluster contribution estimates is described in Appendix A.

Baseline scalable method. An algorithm based on the above light clustering, where cluster sampling probability $P(c|\mathbf{x})$ is proportional to the cluster contribution estimates (Appendix A) and is *not* adapted during calculation, serves as a baseline for comparisons in Sec. 7. We call it the *Scalable* method.

4 WHAT WE LEARN: OPTIMAL CLUSTER SELECTION

We now discuss the optimal cluster selection probabilities $P(c|\mathbf{x})$ in Step (1) of our three-step light sampling procedure (Sec. 3). The conventional way to shape $P(c|\mathbf{x})$ would be to select cluster c proportionally to its true expected contribution, denoted $L_c(\mathbf{x})$. However, as we show below, this choice would be optimal only if the cluster contributions could be evaluated with no variance. This is rarely the case in practice, since the *nested MC estimator* $\langle L_c(\mathbf{x}) \rangle$ of the cluster contribution is itself subject to additional variance. Intuitively, one would want to sample more frequently clusters that contribute more

[1] Probabilities are denoted by the capital P while probability *densities* are lower-case p.

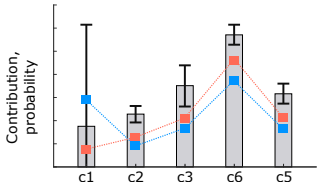

Fig. 2. Illustration of optimal sampling probabilities on a synthetic dataset. The gray bars represent the expected cluster contributions and the error bars show standard deviation of the nested cluster contribution estimators. The orange distribution shows the conventional cluster selection probabilities directly proportional to the cluster contributions, while the blue one correspond to our provably optimal sampling probabilities promoting sampling of clusters that contribute more variance.

variance to the overall result, but the simple selection proportional to the contribution does not do this (Fig. 2). We now derive the optimal cluster selection probabilities conforming to this intuition.

We seek optimal cluster sampling probabilities $P_{\text{opt}}(c|\mathbf{x})$ minimizing the overall variance of estimator (3). Given our three-step sampling, we have $p(\mathbf{y}|\mathbf{x}) = P(c|\mathbf{x})P(l|c)p(\mathbf{y}|l)$, and the variance can be written as:

$$\text{Var}[\langle L(\mathbf{x})\rangle] = -L(\mathbf{x})^2 + \sum_{c\in C} \frac{1}{P(c|\mathbf{x})} \underbrace{\int_{A_c} \frac{(F(\mathbf{y}\to\mathbf{x}))^2}{P(l|c)p(\mathbf{y}|l)}\,\mathrm{d}y}_{m_{2,c}}. \quad (4)$$

Note that $m_{2,c}$ is the second moment of the *nested MC estimator* $\langle L_c(\mathbf{x})\rangle = \frac{F(\mathbf{y}\to\mathbf{x})}{P(l|c)p(\mathbf{y}|l)}$ of the cluster contribution.

We find $P_{\text{opt}}(c|\mathbf{x})$ as a solution to a constrained optimization problem, as described in Appendix B, with the result that the *optimal cluster selection probabilities are proportional to the square root of the second moment* $m_{2,c}$. Given that $\text{Var}[\langle L_c(\mathbf{x})\rangle] = m_{2,c} - L_c^2(\mathbf{x})$, we obtain the final result:

$$\boxed{P_{\text{opt}}(c|\mathbf{x}) \propto \sqrt{L_c^2(\mathbf{x}) + \text{Var}[\langle L_c(\mathbf{x})\rangle]}.} \quad (5)$$

Note that $P_{\text{opt}}(c|\mathbf{x})$ is not proportional just to $L_c(\mathbf{x})$, but it takes into account also the variance of the nested estimator, i.e., variance due to sampling of light areas and complex visibility. This is crucial for the robustness of our method as it prevents excessive noise by focusing on problematic areas in the cases when the nested sampling according to the pdf $P(l|c)p(\mathbf{y}|l)$ is far from ideal (see Fig. 5).

A derivation similar to ours appears in the work by Pantaleoni and Heitz [2017], but in a different context: seeking an optimal piecewise constant approximation to a given sampling probability density.

5 HOW WE LEARN: BAYESIAN ONLINE REGRESSION

In the previous section we have shown that optimal cluster selection probability $P(c|\mathbf{x})$, given by Eq. (5), depends both on the expected cluster contribution $L_c(\mathbf{x})$ and the variance of the nested cluster contribution estimator $\text{Var}[\langle L_c(\mathbf{x})\rangle]$. These quantities are, however, unknown up front, and have to be approximated.

We have two types of information available for that: a) Unbiased, but noisy MC direct illumination samples taken during rendering. b) Noise-free, but biased, estimates of unoccluded cluster contribution (see Appendix A). Both are useful, but insufficient by themselves: The MC samples converge to the exact solution, but are extremely unreliable in early stages of computation. The contribution estimates are more reliable early on, but they do not get any more accurate over time and provide no information on visibility or the nested estimator's variance. A principled approach to *exploiting such uncertain information* and *fusing different sources of information* for adaptive MC sampling is the primary contribution of this paper.

Intuitively, we understand the contribution estimates as our prior knowledge and the MC samples as observations. This view naturally leads to Bayesian modeling. While MC quadrature has traditionally served as a tool for Bayesian inference [Bishop 2006], *we employ Bayesian inference as a tool for robust adaptive MC sampling*. The general idea of the Bayesian approach is to create a probability model describing the *likelihood* (occurrence probability density) of observed data, impose some *prior* probability over parameters of that model and then, infer the *posterior* probability of the model parameters after seeing the data. From the posterior, we can determine the quantity of interest. In our case, by modeling the likelihood of the MC samples and constructing the prior distribution using the contribution estimates, we can find the most probable approximations to $L_c(\mathbf{x})$ and $\text{Var}[\langle L_c(\mathbf{x})\rangle]$ given both these sources of information.

5.1 Model

We start with a standard statistical learning setup. First, we define our training data $\mathcal{D}$ based on the MC samples observed during rendering. Second, we derive a model $p(\mathcal{D}|\theta)$ describing the likelihood of the data given parameters θ. Mean and variance of this model provide the $L_c(\mathbf{x})$ and $\text{Var}[\langle L_c(\mathbf{x})\rangle]$ we are looking for.

These statistics depend on the parameters θ that are initially unknown. We could find them by direct maximization of $p(\mathcal{D}|\theta)$, i.e., use the *maximum likelihood* (ML) estimate. However, ML is prone to overfitting when data is scarce and provides poor approximations in early stages of rendering as shown in Fig. 5. Since robustness is a major concern in adaptive MC, we employ the Bayesian treatment: Impose *prior* probability $p(\theta)$, and infer the *posterior* probability $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$ after seeing the data. By its maximization we get a robust *maximum a posteriori* (MAP) estimate of the parameters.

Data. Each scene region is associated with a set of light clusters (the light cut). We collect the data and learn the model *independently for each region–cluster pair*. Consider one such pair. Sampling of lights in the cluster yields MC illumination samples, $\langle L_c(\mathbf{x})\rangle = \frac{F(\mathbf{y}\to\mathbf{x})}{P(l|c)p(\mathbf{y}|l)}$, where $\mathbf{y}$ is a sampled point on light l, and $\mathbf{x}$ is a shading point inside the region. Our goal is to use the MC samples collected for the region-cluster pair to build a model that accurately predicts $L_c(\mathbf{x})$ and $\text{Var}[\langle L_c(\mathbf{x})\rangle]$ over the different positions $\mathbf{x}$ in the region.

A major cause of spatial variations of illumination is the cosine term $\cos\theta_{\mathbf{x}}$ changing due to varying surface normal. While this effect would be difficult to capture by statistical modeling, it is trivial to compute when needed, so we drop it from our model. We therefore define two quantities

$$\hat{e} = \frac{L_e(\mathbf{y}\to\mathbf{x})V(\mathbf{y}\leftrightarrow\mathbf{x})\cos\theta_{\mathbf{y}}/d^2(\mathbf{x},\mathbf{y})}{P(l|c)\,p(\mathbf{y}|l)} \quad \text{and} \quad \hat{e}_{\mathbf{x}} = \hat{e}\,\overline{\cos}\theta_{\mathbf{x}}. \quad (6)$$

The former quantity, $\hat{e}$, represents the MC sample of the cluster contribution, $\langle L_c(\mathbf{x})\rangle = \frac{F(\mathbf{y}\to\mathbf{x})}{P(l|c)p(\mathbf{y}|l)}$, with the surface cosine term $\cos\theta_{\mathbf{x}}$ dropped. In the latter quantity, $\hat{e}_{\mathbf{x}}$, we replace the cosine term

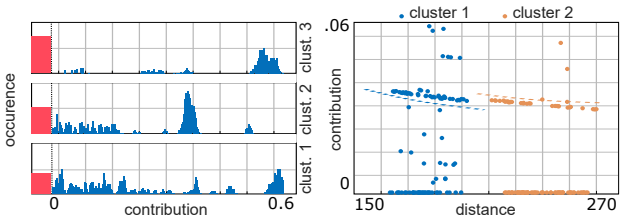

Fig. 3. Left: Histogram of direct illumination samples for three region-cluster pairs. The area of each column corresponds to the overall occurrence in the dataset. Note that zeros (in red) are frequent due to complex occlusion. Right: Scatter plot of sample contribution $\hat{e}_{\mathbf{x}}$ vs sample distance $\hat{d}$ for two clusters distinguished by colors. Note the inverse-squared-distance falloff.

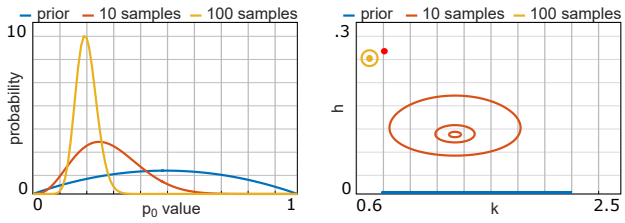

Fig. 4. Evolution of the posterior distribution for the parameters p_o (beta distribution, left), and k and h (normal-inverse-gamma, right) after seeing 0 (prior), 10 and 100 (synthetic) samples, with hyperparameters set as described in the main text, and the hyperparameter μ_0 set to 1.5. The visible samples' true mean and variance was 0.8 and 0.25, respectively, and is marked by the red dot (right). Note that the $\mathcal{N}$-Γ^{-1} prior is almost zero everywhere except near the x-axis, due to $\beta = 1\text{e}{-6}$.

by its upper bound over the entire cluster $\overline{\cos}\theta_{\mathbf{x}}$ (see Appendix A). Our region statistics are based on $\hat{e}$, while $\hat{e}_{\mathbf{x}}$ is used at a specific shading point $\mathbf{x}$ to inject surface normal dependency into our model.

After the surface normal, the second important factor in illumination variation across a region is the inverse-squared-falloff with the distance $\hat{d} = d(\mathbf{x}, \mathbf{y})$, as confirmed by the empirical data (Fig. 3). To capture this dependency, we explicitly model the relation between illumination samples and the distance $\hat{d}$ using a regression model. Therefore, our training data $\mathcal{D}$ consists of tuples $(\hat{e}_{\mathbf{x},i}, \hat{d}_i)$.

Model and its parameters. The next step is to define a statistical regression model $p(\mathcal{D}|\theta)$ expressing the data likelihood, i.e., *probability of MC samples of direct illumination*. The general form of the likelihood used to model the relation between $\hat{d}$ and $\hat{e}_{\mathbf{x}}$ is:

$$p(\mathcal{D}|\theta) = \prod_i^N p(\hat{e}_{\mathbf{x},i}|\hat{d}_i, \theta). \tag{7}$$

where $p(\hat{e}_{\mathbf{x}}|\hat{d}, \theta)$ represents a regression model, N is the total number of samples (for a region-cluster pair), and the model parameters θ are discussed below.

Our regression model of direct illumination has the following two important features:

(1) Approximation of the inverse-squared-distance falloff.
(2) Explicit modeling of occluded contributions.

Motivation for the former property has been given above and follows naturally from the form of the sample contribution $\hat{e}_{\mathbf{x}}$, Eq. (6). The second feature arises from the all-or-nothing nature of the visibility function, which is difficult to model by any common smooth distribution (Fig. 3). We, therefore, design our regression model as a mixture of a delta function δ (describing zero, i.e., occluded contributions) and a Gaussian $\mathcal{N}$ with mean and variance decreasing with the second and fourth power of the distance term (describing non-zero, i.e., visible contributions):

$$p(\hat{e}_{\mathbf{x}}|\hat{d}, \theta) = \delta(\hat{e}_{\mathbf{x}})p_o + (1 - p_o)\mathcal{N}\left(\hat{e}_{\mathbf{x}} \,\middle|\, \frac{k}{\hat{d}^2}, \frac{h}{\hat{d}^4}\right). \tag{8}$$

The model parameters $\theta = (p_o, k, h)$ are respectively the probability of occlusion, average visible contribution coming from a cluster omitting the distance, and the variance of this contribution. As each sample $\hat{e}_{\mathbf{x},i}$ shows inverse-squared-distance falloff of its mean, sample's variance changes as well, but with $1/\hat{d}^4$. The benefit of explicit visibility modeling is illustrated in Fig. 5.

Prior distribution. To make the inference step tractable, we seek a *conjugate prior*, i.e., prior distribution which yields a posterior of the same function type. The conjugate prior for our model, derived in Appendix C, has p_o distributed according to the beta distribution B and the pair (k, h) according to the normal-inverse-gamma distribution $\mathcal{N}$-Γ^{-1}. Our prior distribution for parameters θ is then:

$$p(\theta) = \mathrm{B}(p_o|\hat{N}_o, \hat{N}_v)\ \mathcal{N}\text{-}\Gamma^{-1}(k, h \,|\, \mu_0, \hat{N}, \hat{N}_\alpha, \beta). \tag{9}$$

The various hyperparameters in the above equation can be understood as statistics of hypothetical prior observations before the first actual sample has been taken. $\hat{N}_o$ and $\hat{N}_v$ denote the number of occluded and visible prior observations, μ_0 is the mean of $\hat{N}$ prior visible observations and β is the sum of squares of $2\hat{N}_\alpha$ prior visible observations. Note that these hyperparameters do not necessarily describe a consisitent set of virtual prior observations (i.e., in general $\hat{N} \neq \hat{N}_v$ and $2\hat{N}_\alpha \neq \hat{N}$). Intuitively, $\hat{N}_o$, $\hat{N}_v$, $\hat{N}$ and $\hat{N}_\alpha$ express the strength of the priors and larger values will cause slower, but potentially more robust learning.

To obtain the prior parameter μ_0, we use our *second source of information*, unoccluded cluster contribution estimate $\tilde{L}_c(\mathbf{x})$ (Appendix A). To make the prior more robust to occasional gross errors in these estimates, we blend the $\tilde{L}_c(\mathbf{x})$-proportional distribution with a defensive uniform distribution over the clusters [Veach 1997]. Finally, $\tilde{L}_c(\mathbf{x})$ contains a division by the squared-distance $d^2(\mathrm{ctr}(c), \mathbf{x})$ to the cluster center $\mathrm{ctr}(c)$. But μ_0 is a prior on the parameter k, which gets divided by the distance in our model, Eq. (8). We counter double division by the distance by pre-multiplying by $d^2(\mathrm{ctr}(c), \mathbf{x})$. In summary, our informed prior mean reads

$$\mu_0 = \frac{1}{2}\left(\tilde{L}_c(\mathbf{x}) + \frac{\sum_{c' \in C} \tilde{L}_{c'}(\mathbf{x})}{|C|}\right) d^2(\mathrm{ctr}(c), \mathbf{x}). \tag{10}$$

Good hyperparameter values should strike a good tradeoff between the learning rate and robustness to noisy samples. We found the following values to work robustly across all our tests: $\hat{N}_o = 2, \hat{N}_v = 2, \hat{N} = 1, \hat{N}_\alpha = 1, \beta = 1\text{e}{-6}$. Refer to Fig. 4 for an illustration how posterior distributions of parameters θ evolve with the number of observed samples.

5.2 Inference

With both the likelihood and prior defined, we now infer the most probable parameters' values after seeing the data. We maximize the

logarithm of the posterior distribution with respect to the parameters to obtain the MAP point estimate for θ. That boils down to finding the solution to $\nabla_\theta \log(\, p(\mathcal{D}|\theta)p(\theta)\,) = 0$, which expands to:

$$\frac{\nabla_\theta p(\theta)}{p(\theta)} + \sum_i^N \frac{\nabla_\theta p(\hat{e}_{\mathbf{x},i}|\hat{d}_i,\theta)}{p(\hat{e}_{\mathbf{x},i}|\hat{d}_i,\theta)} = 0. \tag{11}$$

Plugging our model, Eq. (8) and (9), into Eq. (11) we get the following MAP estimate of the θ parameters (see Appendix D):

$$p_o = \frac{-1 + \hat{N}_o + N_o}{-2 + \hat{N}_o + \hat{N}_v + N}, \tag{12}$$

$$k = \frac{s_{1,\mathbf{x}} + \hat{N}\mu_0}{\hat{N} + N_v}, \tag{13}$$

$$h = \frac{-2\hat{N}\mu_0 s_{1,\mathbf{x}} - s_{1,\mathbf{x}}^2 + (s_{2,\mathbf{x}} + 2\beta)(\hat{N} + N_v) + \hat{N}N_v\mu_0^2}{(2\hat{N}_\alpha + N_v - 1)(\hat{N} + N_v)} \tag{14}$$

where $s_1 = \sum_i^{N_v} \hat{d}_i^2\hat{e}_i$, $s_{1,\mathbf{x}} = s_1\overline{\cos}\theta_{\mathbf{x}}$ and $s_2 = \sum_i^{N_v} \hat{d}_i^4\hat{e}_i^2$, $s_{2,\mathbf{x}} = s_2\overline{\cos}\theta_{\mathbf{x}}^2$ represent statistics over *visible* samples, N_o and N_v are the number of occluded and visible samples, and $N = N_o + N_v$ is the overall number of samples (for the considered region-cluster pair).

With these parameters, the expectation and variance of our model in Eq. (8), approximating $L_c(\mathbf{x})$ and $\mathrm{Var}[\langle L_c(\mathbf{x})\rangle]$, respectively, are:

$$L_c(\mathbf{x}) \approx (1 - p_o)k/\hat{d}^2, \tag{15}$$

$$\mathrm{Var}[\langle L_c(\mathbf{x})\rangle] \approx (1 - p_o)(p_o k^2 + h)/\hat{d}^4. \tag{16}$$

We set $\hat{d} = d(\mathrm{ctr}(c), \mathbf{x})$ to approximate the not yet known distance for $\mathbf{x}$, where $\mathrm{ctr}(c)$ denotes the cluster center.

5.3 Summary

Let us now summarize the steps involved in direct illumination computation at a shading point $\mathbf{x}$. We take the cut C stored in region R containing $\mathbf{x}$ and for each of its clusters c we compute the unoccluded contribution estimates $\tilde{L}_c(\mathbf{x})$ and $\overline{\cos}\theta_{\mathbf{x}}$ (Appendix A), and we set $\hat{d} = d(\mathrm{ctr}(c), \mathbf{x})$. We cull clusters with $\tilde{L}_c(\mathbf{x}) = 0$, i.e., which have provably zero contribution to $\mathbf{x}$, from any further processing.

For the remaining clusters, we compute μ_0 using Eq. (10), retrieve the region-cluster statistics s_1, s_2, N_o, N_v, and compute the MAP parameters (p_o, k, h). Finally, we get the sampling probability $P^*(c|\mathbf{x})$ by plugging Equations (15) and (16) into (5):

$$P^*(c|\mathbf{x}) \propto \frac{1}{\hat{d}^2}\sqrt{(1 - p_o)^2k^2 + (1 - p_o)(p_o k^2 + h)}. \tag{17}$$

Using these probabilities we select a cluster c^*, then we select a light $l^* \in c^*$ with probability $P(l^*|c^*) = \Phi_{l^*}/\sum_{l' \in c^*}\Phi_{l'}$ and finally, sample a point $\mathbf{y}^* \in A_{l^*}$ using the standard techniques [Pharr et al. 2016; Shirley et al. 1996]. Contribution of this sample is then used to update statistics s_1, s_2, N_o, N_v stored for the cluster c^* in the region.

6 CONTROL VARIATES

Inspired by the successes of control variates (CV) documented in previous work [Clarberg and Akenine-Möller 2008; Owen and Zhou 2000; Pegoraro et al. 2008; Rousselle et al. 2016], we exploit our accumulated statistics as a CV for further variance reduction. We keep the nested MC estimator $\langle L_c(\mathbf{x})\rangle$ of cluster contribution as before (i.e. Steps (2) and (3) in Sec. 3), and apply the CV to the MC estimator of the sum over clusters:

$$\langle L(\mathbf{x})\rangle_{\mathrm{CV}} = \frac{\langle L_c(\mathbf{x})\rangle + H(c,\mathbf{x})}{P(c|\mathbf{x})} - \sum_{c' \in C} H(c', \mathbf{x}). \tag{18}$$

The better the control variate $H(c, \mathbf{x})$ approximates the true cluster contribution $L_c(\mathbf{x})$, the more the variance is reduced. Since this is precisely the purpose of our Bayesian model (Eq. (15)), it would seem natural to also use it directly as the CV. However, while we strongly prefer overestimation to underestimation for the sampling distribution, this is not the case for the CV. We, therefore, omit the conservative prior in its definition, and the CV reads

$$H(c,\mathbf{x}) = \frac{1}{N}\frac{s_{1,\mathbf{x}}}{d^2(\mathrm{ctr}(c),\mathbf{x})}. \tag{19}$$

Despite the CV acting as a mere empirical improvement over the theory presented so far, it yields noticeable variance reduction at a negligible cost (Fig. 5).

7 RESULTS

Implementation. We have implemented our method in a production path tracer and deployed it among users. Our path tracer combines light sampling and BRDF importance sampling using MIS [Veach 1997] alleviating the fact that our sampling distributions do not take BRDF into account. When used in this setting, the direct illumination samples we use for training are pre-weighted by MIS weights. This heuristic approach works well in practice (see Fig. 11), and a more principled analysis is left for future work.

Test setup. We show results of our tests on three different scenes: Living room, City and Door (see Fig. 1 and 10). Living room is a typical scene in the architectural visualization featuring a living room lit by the sun and a few area lights on the ceiling. In contrast, the City scene shows a street at night and contains more than 5000 light sources. Finally, Door is a rather simple scene featuring complex shadowing.

In addition to these three main scenes, we use two another scenes for specific comparisons: Wedge and Hall (see Fig. 9 and 11). Wedge is a simple synthetic scene illuminated by three area lights and an environment map. Hall features complex glossy materials illuminated by the sun, an environment map and tens of area lights of various sizes.

Exact light counts along with other statistics are summarized in Table 1. All scenes were rendered at the resolution 1080×720 on a single machine with the Intel Core i7-5820K CPU (6 cores, 12 threads) and 32 GB of RAM.

Method components. We first demonstrate individual components of our method in the City scene in Fig. 5. We start by sampling proportionally to an estimate of each light's unoccluded contribution (a). At every shading point, this method estimates the contribution of *all* scene lights (using $\tilde{L}_c(\mathbf{x})$ from Appendix A), and uses these estimates to construct the sampling distribution. This procedure becomes prohibitively expensive for the many lights as in this scene.

By subdividing the scene into regions and sampling proportionally to the unoccluded contribution of light clusters in the associated

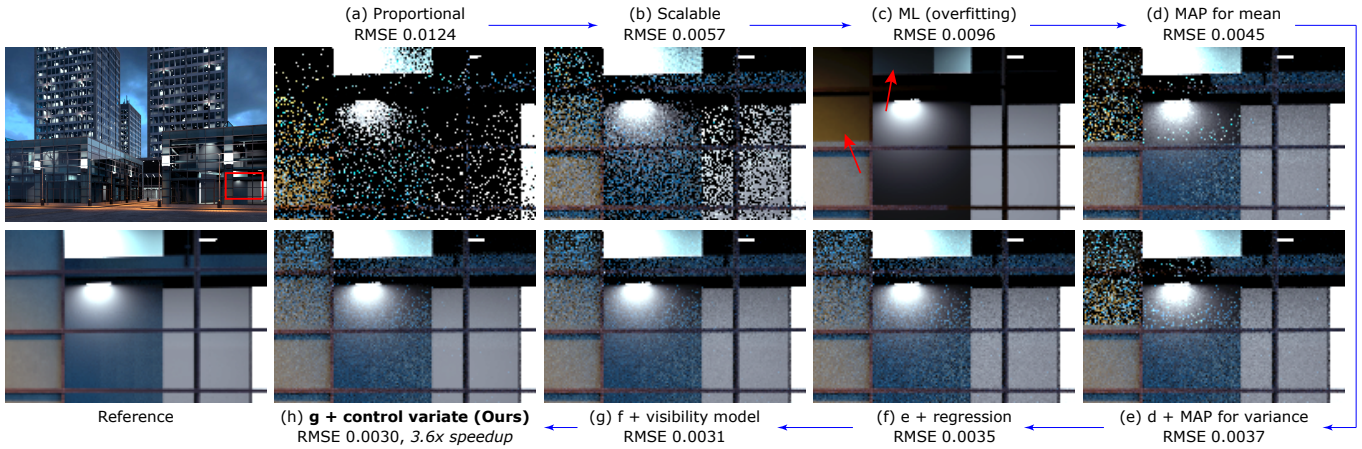

Fig. 5. Equal-time comparison (60 s) of different components of our direct illumination sampling method in a scene with more than 5000 lights and high occlusion. We compare sampling proportional to (a) unoccluded light contribution computed separately for each shading point and light, (b) unoccluded light cluster contribution incorporating our scalable solution, (c) maximum likelihood (ML) estimate of the mean cluster contribution (dark artifacts are a consequence of overfitting), (d) maximum a posteriori (MAP) estimate of the mean cluster contribution. The remaining variants gradually add the following components: (e) MAP estimate for variance, (f) regression to model the distance falloff, (g) explicit modeling of occluded samples, (h) control variate. The last result corresponds to our final solution. The numbers below the method names denote the RMSE to a reference solution. The speedup is with respect to (b).

cuts, we obtain the *Scalable* method (b) which scales much better with the number of lights but still neglects visibility.

Learning light sampling probabilities using a simple maximum likelihood (ML) estimate, i.e., the mean of MC samples, (c) can easily lead to bias: If the first observed sample is occluded (zero), the cluster will not receive any further samples, yielding dark artifacts highlighted by red arrows in the figure.

Such artifacts can be avoided by using a MAP estimate of the mean (d). However, as we show in Sec. 4, optimal cluster sampling distribution should take into account the variance of sampling inside each cluster. Indeed, adding a MAP estimate for this nested estimator's variance significantly reduces noise (e). Incorporating regression modeling of the distance falloff (f) eliminates noise most noticeable near region boundaries. Finally, explicit modeling of occluded samples and the use of control variates further reduces noise. This is the complete method we use in all our further tests, and we denote it *Ours*. Version (b), denoted *Scalable*, serves as a baseline for the comparisons. In this scene, Ours is 3.6× faster than Scalable.

Grid resolution. Our spatial regression model makes the performance of our algorithm rather insensitive to the division of scene into regions. As shown in Fig. 6, a trade-off exists between the model accuracy (the smaller the regions, the more accurate the models) and the learning rate (the larger the regions the more samples are available) in the City scene (though the dependence is weak) while almost no difference is visible in the other scenes. For this reason, all our results use a fixed-resolution uniform grid with cubical regions with 64 regions along the shortest scene dimension.

Robustness and DI-only performance. We now demonstrate superior robustness of our method over the work by Donikian et al. [2006] (details of our reimplementation are given in the Supplemental). While Donikian et al.'s method also relies on learning, it is based on heuristics that eventually fail to deliver a robust solution. The method gathers statistics in image space and cannot be easily integrated in a global illumination solution. For this reason, we compare on direct illumination (DI), and take this opportunity to provide a DI-only comparison to the Scalable method, see Fig. 7.

The sun in the *Living room* scene is significantly stronger than other lights. Since the Scalable method has no notion of visibility, it prefers sampling the sun while undersampling the other lights, even in sun's shadow. Our method quickly learns the sun occlusion and avoids the excessive noise of Scalable. It converges more evenly and more than 500× faster. Donikian et al.'s method also shows improvement over Scalable but struggles with sampling an area light covered by a shade letting only a small portion of the light through. The method overfits and introduces spiky noise.

The *Door* scene aims at testing robustness with complex shadow and light patterns. While Scalable struggles in shadows as before, Donikian et al.'s method learns light occlusion quickly and it may even outperform our method in uniformly lit areas. However, this aggressive adaptation comes at the cost of overfitting, which is then manifested as spiky noise and artifacts around shadow boundaries. Notice the square holes in the penumbra of the plant in the first inset and at intersections of the net of shadows in the second one. Our method robustly handles all these situations while being more than 9× faster than Scalable.

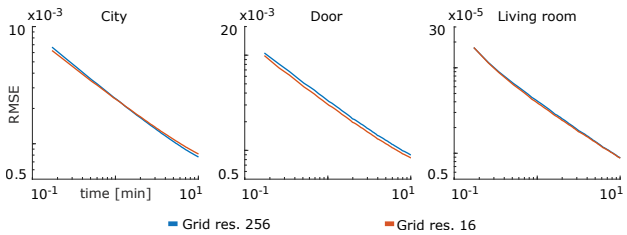

Fig. 6. RMSE evolution (10 min) for different grid resolutions. With a finer resolution our model might learn more slowly but achieve better accuracy (and thus lower RMSE). Nonetheless, the differences are very small.

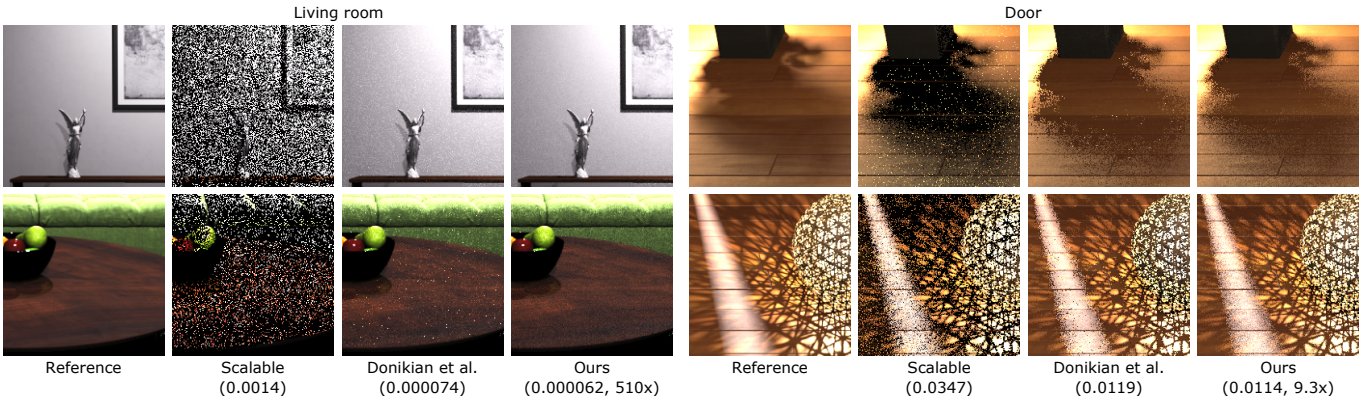

Fig. 7. Equal-time comparison (60 s) of our method against Scalable and Donikian et al.'s methods in a direct illumination setting. See the main text for details.

We compared Scalable and our method in the *City* scene (Fig. 5) but we had to omit Donikian et al.'s because of its vague description of dealing with many lights. RMSE evolution plots in Fig. 8 show that in the City scene our method maintains a stable speedup over Scalable, while in the other two we can observe a higher empirical convergence rate.

We want to underline that our improvement over Donikian et al. lies mainly in the robustness, not the speed. In fact, their method can outperform ours in uniformly lit areas, but introduces unacceptable artifacts at shadow boundaries (Fig. 7 and 9). This lack of robustness is an inherent property of their static strategy to prevent overfitting (weighting distributions based on the iteration step) and cannot be avoided by any parameter tweaking. Addressing this deficiency is the very purpose of our Bayesian approach.

Discussion of other competing work. The method of Wang and Akerlund [2009] is similar to the Scalable method. Unlike Wang and Akerlund, Scalable omits the BRDF from light sampling distribution, but that does not introduce any disadvantage on diffuse surfaces. Furthermore, Scalable gains some performance gain by caching of light cuts for scene regions. As a result, comparison against the Scalable baseline can serve as a fairly good approximation to a comparison against Wang and Akerlund.

We do not compare against methods that involve substantial preprocessing [Georgiev et al. 2012; Wu and Chuang 2013] since these methods address a different use case than ours. In a typical commercial rendering workflow a vast majority of renders are in fact short tests, not the final images. In this context, a preprocessing step is an obstacle that would prevent the method from being used in the pipeline that our users rely on in their daily work.

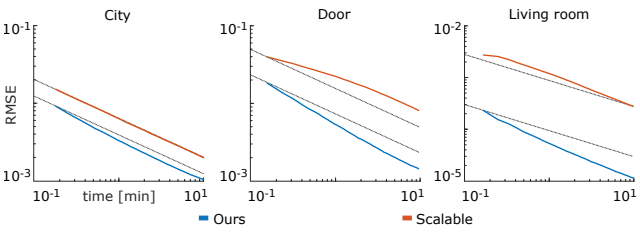

Fig. 8. RMSE evolution (10 min) for the direct illumination only. Our method is compared against the Scalable method. The plots start at 10 seconds to ensure all pixels were sampled at least once.

Global illumination integration. When integrated in a global illumination (GI) solution, the relative performance improvement of our method naturally depends on the variance contribution due to the direct and indirect components. While our DI method yields an almost noise-free GI result in all three scenes, in the *City* and *Door* scenes (Fig. 10) roughly half of the speedup of the DI-only solution is retained (speedup 3.6× from Fig. 5 and 9.3× from Fig. 7 decreases to 2.0× and 4.3× respectively). On the other hand, our 510× speedup in the DI-only comparison in the *Living room* scene (Fig. 7) reduces to 6.7× in GI (Fig. 1). This indicates that variance contribution of the direct component in this scene is small in comparison to the total illumination.

Memory consumption and overhead. At our grid resolution (64 regions along the shortest scene dimension), memory consumed

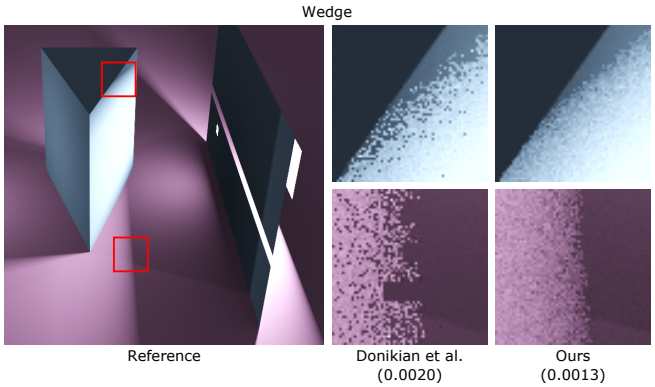

Fig. 9. Equal-time time comparison (10 s) of our method against Donikian et al.'s method in a direct illumination setting. The large area light on the right that illuminates the scene only through a narrow gap presents a difficult situation for the Donikian et al.'s method. Many of its samples are blocked, which increases the danger of overfitting, manifested as block artifacts along shadow boundaries where the algorithm incorrectly decided to stop sampling the light.

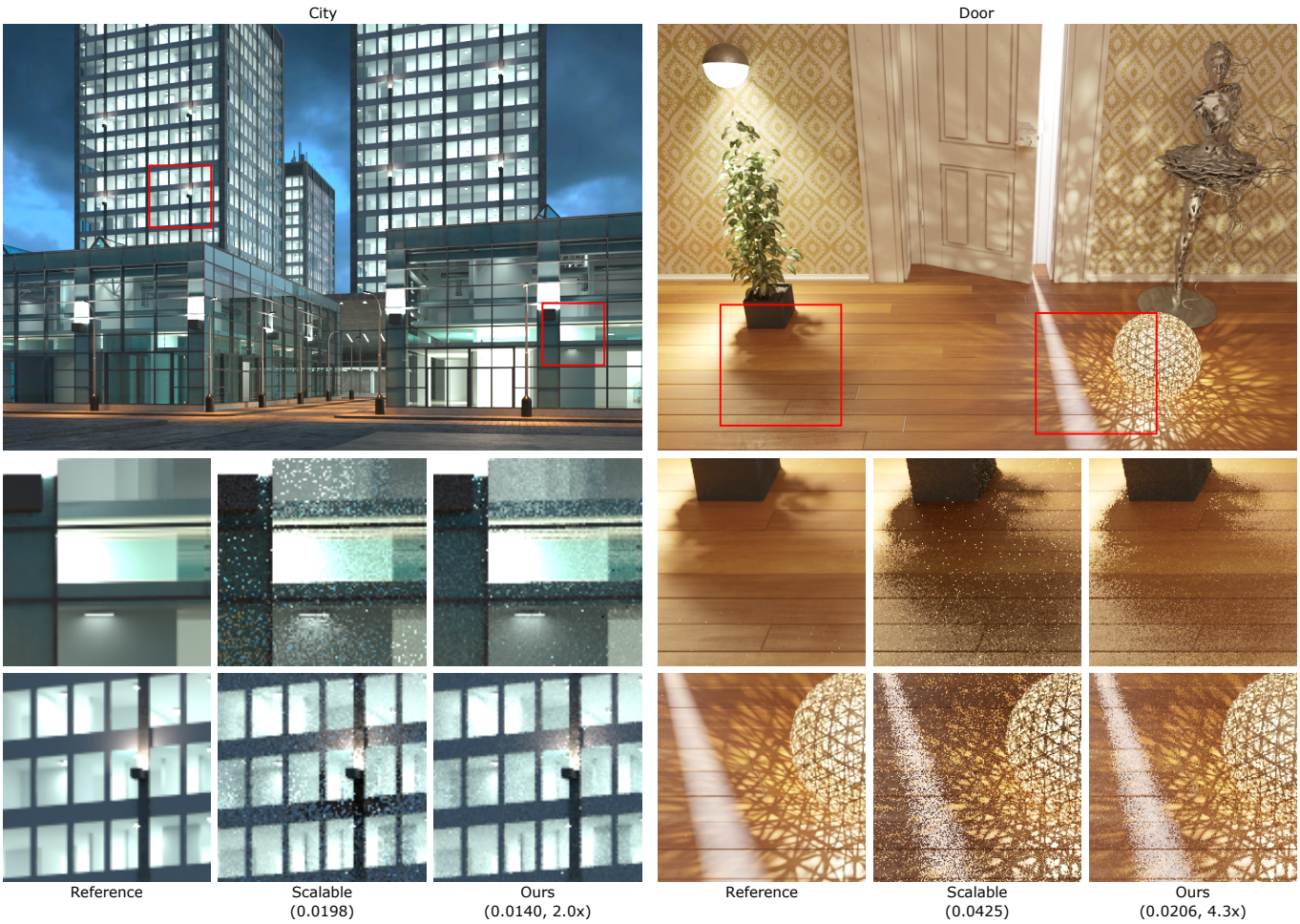

Fig. 10. Equal-time time comparison (60 s) of our method against Scalable in a global illumination setting. See the main text for more details.

by the stored light cuts and model statistics is moderate, as shown in Table 1. These numbers are for a GI setting and less memory is consumed when computing only DI. An empty scene region occupies 40 B of memory. Every cluster inside a region consumes additional 48 B in order to store: 2× 64-bit double for statistics s_1, s_2; 2× 32-bit integer for statistics N_o, N_v; 64-bit pointer to cluster tree node; 32-bit integer for flags; 3× 32-bit float for RGB channels of s_1 for the control variate.

Regarding computation overhead, number of pixel samples per second decreased in our method in comparison to Scalable by no more than 10% in all our test scenes (see Table 1). The learning compensates for this by better sampling yielding much improved overall result.

MIS combination. We tested our method both with and without MIS combination with BRDF sampling. While there is almost no difference in the Living room, City and Door scenes, in scenes with large area lights and glossy materials, the MIS combination proves beneficial as shown in the Hall scene in Fig. 11. Even in this scene containing complex illumination and glossy materials, our method performs well even though our light sampling distribution does not take the BRDF into account nor it addresses sampling of individual lights.

Unbiasedness. Although we use past samples to update sampling distributions, we do not modify sample values based on the past

Table 1. Statistics gathered after 120 s of rendering of our test scenes with global illumination. The average cut size (i.e., number of clusters per region) is taken over non-empty regions only. Total memory consumed by the regions and clusters is reported. The overhead expresses relative decrease of pixel samples per second with respect to Scalable.

	Light count	Non-empty regions	Average cut size	Memory (MB)	Over-head
City	5022	39666 (4.1%)	33	101	7.2%
Door	5	24526 (1.1%)	5	97	3.6%
Living room	5	57304 (2.3%)	5	113	7.9%
Hall	78	31304 (6.6%)	39	78	9.8%
Wedge	4	10871 (0.4%)	4	101	9.0%

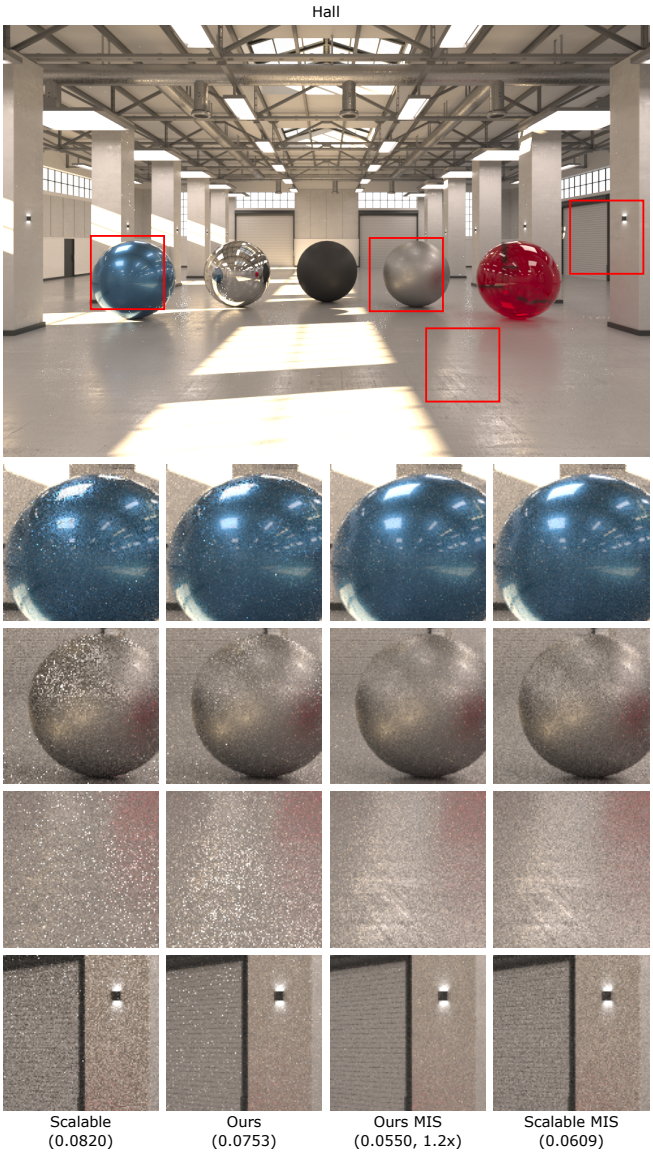

Fig. 11. Equal-time time comparison (60 s) of our method against Scalable with and without MIS in a global illumination setting.

observations and our method is therefore unbiased. In Fig. 12 we empirically demonstrate a steady convergence of our method to the result of the (non-adaptive) Scalable method.

8 LIMITATIONS AND FUTURE WORK

Multiple Importance Sampling (MIS). We have discussed in Sec. 7 the heuristic nature of the integration of our method with MIS. While our approach works well in practice and successfully handles large area light sources and complex materials (Fig. 11), a more in-depth analysis could yield further improvements.

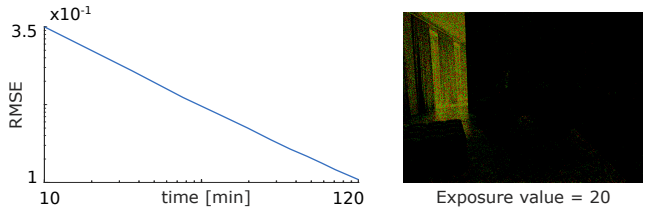

Fig. 12. Steady convergence of our method (RMSE plot, left) to reference solution in the Living room scene suggests that our adaptive method accumulates no bias. A $2^{20}\times$ amplified color-coded difference image (right), taken at the end of the measurement, shows that any remaining differences are due to a random noise (red=positive and green=negative difference).

BRDF. Our method does not consider the BRDF factor in learning the sampling distributions. This makes the learning more tractable (less detailed function to learn) and practical in a production setting (the BRDF can be a black box). But it limits the adaptability of the sampling. Though this has not been an issue in practice thanks to the MIS combination with BRDF sampling, incorporating the BRDF in the learning process could still be beneficial.

Scene subdivision. Another interesting point is the trade-off between model accuracy and learning rate due to the scene division. The graphs in Fig. 6 suggest such a trade-off exists, although the differences are small. However, the graphs show aggregate statistics over the entire scene, which can obscure the fact that adaptive scene subdivision could still have an important positive *local* impact.

Hyperparameters. As yet another area of research we see a more rigorous approach for hyperparameter selection. Our default choice yields an uninformed prior distribution over the parameters, which fits all scenes, but it might deliver suboptimal performance. Full Bayesian treatment could yield further performance gains. See the Supplemental for additional analysis of different hyperparameter values.

Sampling of individual lights. Our method focuses on light selection and leaves sampling of the final point on the light unaddressed. This is motivated by the fact that the light selection is usually responsible for most of the variance in direct illumination. But this may not always be the case, especially when the individual lights are large (e.g., environment maps). This is partially alleviated by the integration with MIS (Fig. 11) but there is certainly some potential for improvement.

Overhead. Probably the thorniest practical issue, shared with the Scalable method, is the overhead associated with constructing the sampling distribution at each shading point. This is amortized in our implementation by a relatively large splitting factor (16 samples taken from one distribution) but it could be an issue in a simple path tracer without splitting.

Relation to path guiding. As mentioned in Sec. 2, path guiding and our method share the idea of sampling according to a priori unknown illumination estimates. But while path guiding usually focuses on indirect illumination, we address specifically light source selection for direct illumination computation. In fact, our work is a component that could be integrated into a path guiding solution.

9 CONCLUSION

We proposed an unbiased adaptive direct illumination algorithm with online learning of light sampling distributions. The distributions are continually improved based on the contribution of the direct illumination samples taken during rendering, including the visibility factor. As in any other adaptive Monte Carlo sampling scheme, issues associated with limited reliability of the available information threaten the robustness of the resulting algorithm. As our main contribution, we propose a Bayesian treatment of the learning process based on a statistical model developed specifically for the direct illumination sampling process. This treatment results in a robust and efficient algorithm, and we hope that the presented methodology will find its use in other adaptive Monte Carlo schemes both in image synthesis and other application domains.

ACKNOWLEDGMENTS

Many thanks to Ludvík Koutný (a.k.a. rawalanche) for modeling the test scenes. The work was supported by the Charles University Grant Agency project GAUK 1172416, by the grant SVV-2017-260452, and by the Czech Science Foundation grant 16-18964S.

REFERENCES

Niels Billen, Björn Engelen, Ares Lagae, and Philip Dutré. 2013. Probabilistic Visibility Evaluation for Direct Illumination. *Computer Graphics Forum* 32, 4 (2013), 39–47.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag New York Inc.

Malik Boughida and Tamy Boubekeur. 2017. Bayesian collaborative denoising for Monte Carlo rendering. *Computer Graphics Forum* 36, 4 (2017), 137–153.

Jonathan Brouillat, Christian Bouville, Brad Loos, Charles Hansen, and Kadi Bouatouch. 2009. A Bayesian Monte Carlo approach to global illumination. *Computer Graphics Forum* 28, 8 (2009), 2315–2329.

Brian C. Budge, John C. Anderson, and Kenneth I. Joy. 2008. Caustic forecasting: Unbiased estimation of caustic lighting for global illumination. *Computer Graphics Forum* 27, 7 (2008), 1963–1970.

Norbert Bus, Nabil H. Mustafa, and Venceslas Biri. 2015. IlluminationCut. *Computer Graphics Forum (Proceedings of Eurographics 2015)* 34, 2 (2015), 561–573.

Olivier Cappé, Randal Douc, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. 2008. Adaptive importance sampling in general mixture classes. *Statistics and Computing* 18, 4 (2008), 447–459. arXiv:0710.4242

Olivier Cappé, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. 2004. Population Monte Carlo. *Journal of Computational and Graphical Statistics* 13, 4 (2004), 907–929.

Petrik Clarberg and Tomas Akenine-Möller. 2008. Exploiting visibility correlation in direct illumination. *Computer Graphics Forum* 27, 4 (2008), 1125–1136.

Jean-Marie Cornuet, Jean-Michel Marin, Antonietta Mira, and Christian P. Robert. 2009. Adaptive Multiple Importance Sampling. *Scandinavian Journal of Statistics* 39, 4 (2009), 798–812.

Michael Donikian, Bruce Walter, Kavita Bala, Sebastian Fernandez, and Donald P. Greenberg. 2006. Accurate direct illumination using iterative adaptive sampling. *IEEE Transactions on Visualization and Computer Graphics* 12, 3 (2006), 353–363.

Randal Douc and Arnaud Guillin. 2007. Minimum variance importance sampling via population Monte Carlo. *ESAIM: Probability and Statistics* 11 (2007), 427–447.

Shaohua Fan, Yu-Chi Lai, Stephen Chenney, and Charles Dyer. 2007. *Population Monte Carlo Sampler for Rendering*. Technical Report 1613. Department of Computer Sciences, University of Wisconsin-Madison.

Sebastian Fernandez, Kavita Bala, and Donald P. Greenberg. 2002. Local Illumination Environments for Direct Lighting Acceleration. *Proceedings of the 13th Eurographics workshop on Rendering* (2002), 7–14.

Manuel N. Gamito. 2016. Solid Angle Sampling of Disk and Cylinder Lights. *Comput. Graph. Forum* 35, 4 (July 2016), 25–36. https://doi.org/10.1111/cgf.12946

Iliyan Georgiev, Jaroslav Křivánek, Stefan Popov, and Philipp Slusallek. 2012. Importance Caching for Complex Illumination. *Computer Graphics Forum* 31 (2012).

David Hart, Philip Dutré, and Donald P. Greenberg. 1999. Direct illumination with lazy visibility evaluation. *SIGGRAPH '99* (1999), 147–154.

Heinrich Hey and Werner Purgathofer. 2002. Importance sampling with hemispherical particle footprints. *Proceedings of the 18th spring conference on Computer graphics - SCCG '02* (2002), 107.

Henrik Wann Jensen. 1995. Importance driven path tracing using the photon map. *Rendering Techniques* 95 (1995), 326–335.

Henrik Wann Jensen and Niels Jørgen Christensen. 1995. Efficiently rendering shadows using the photon map. *Compugraphics '95* (1995), 285–291.

Malvin H. Kalos and Paula A. Whitlock. 1986. *Monte Carlo Methods*. Wiley-Interscience.

Eric P. Lafortune and Yves D. Willems. 1995. A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing. (1995), 11–20.

Yu-Chi Lai, Shao Hua Fan, Stephen Chenney, and Charcle Dyer. 2007. Photorealistic Image Rendering with Population Monte Carlo Energy Redistribution. In *Proceedings of Eurographics Symposium on Rendering (EGSR'07)*.

Marc Lebrun, Antoni Buades, and Jean-Michel Morel. 2013. A Nonlocal Bayesian Image Denoising Algorithm. *SIAM Journal on Imaging Sciences* 6, 3 (2013), 1665–1688.

Ricardo Marques, Christian Bouville, Mickaël Ribardiere, Luís Paulo Santos, and Kadi Bouatouch. 2013. A spherical gaussian framework for Bayesian Monte Carlo rendering of glossy surfaces. *IEEE Trans. Vis. Comput. Graph.* 19, 10 (2013), 1619–1632.

Luca Martino, Víctor Elvira, David Luengo, and Jukka Corander. 2015. An Adaptive Population Importance Sampler: Learning from Uncertainty. *IEEE Transactions on Signal Processing* 63, 16 (2015), 4422–4437.

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Eurographics Symposium on Rendering* 36, 4 (2017).

Art Owen and Yi Zhou. 2000. Safe and Effective Importance Sampling. *J. Amer. Statist. Assoc.* 95, 449 (2000), 135–143.

Jacopo Pantaleoni and Eric Heitz. 2017. Notes on optimal approximations for importance sampling. 2, 5 (2017). arXiv:1707.08358

Eric Paquette, Pierre Poulin, and George Drettakis. 1998. A light hierarchy for fast rendering of scenes with many lights. *Computer Graphics Forum* 17, 3 (1998), 63–74.

Vincent Pegoraro, Carson Brownlee, Peter S. Shirley, and Steven G. Parker. 2008. Towards interactive global illumination effects via sequential Monte Carlo adaptation. *IEEE/EG Symposium on Interactive Ray Tracing 2008* (2008), 107–114.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering, From Theory to Implementation* (3rd ed.). Morgan Kaufmann Publishers Inc.

Stefan Popov, Iliyan Georgiev, Philipp Slusallek, and Carsten Dachsbacher. 2013. Adaptive quantization visibility caching. *Computer Graphics Forum* 32, 2 (2013), 399–408.

Carl Edward Rasmussen and Zoubin Ghahramani. 2003. Bayesian Monte Carlo. *Advances in Neural Information Processing Systems 15* 1 (2003), 489–496.

Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-space Control Variates for Rendering. *ACM Transactions on Graphics* 35, 6 (2016).

Peter Shirley, Changyaw Wang, and Kurt Zimmerman. 1996. Monte Carlo techniques for direct lighting calculations. *ACM Transactions on Graphics* 15, 1 (1996), 1–36.

Eric Veach. 1997. Robust Monte Carlo Methods for Light Transport Simulation. *Dissertation at the Department of Computer Science of Stanford University* (1997).

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Trans. Graph.* 33, 4 (2014), 101:1–101:11.

Ingo Wald and Carsten Benthin. 2003. Interactive Global Illumination in Complex and Highly Occluded Environments. *Eurograhics Symposium on Rendering* (2003), 1–9.

Bruce Walter, Adam Arbree, Kavita Bala, and Donald P. Greenberg. 2006. Multidimensional lightcuts. *ACM Transactions on Graphics* 25, 3 (2006), 1081.

Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P Greenberg. 2005. Lightcuts: a scalable approach to illumination. *ACM Transactions on Graphics* 24, 3 (2005), 1098–1107.

Rui Wang and Oskar Akerlund. 2009. Bidirectional Importance Sampling for Unstructured Direct Illumination. *Computer Graphics Forum* 28, 2 (2009), 269–278.

Gregory J. Ward. 1994. Adaptive Shadow Testing for Ray Tracing. *Proceedings of the Second Eurographics Workshop on Rendering* (1994), 11–20.

Yu Ting Wu and Yung Yu Chuang. 2013. VisibilityCluster: Average directional visibility for many-light rendering. *IEEE Trans. Vis. Comput. Graph.* 19, 9 (2013), 1566–1578.

A CLUSTER CONTRIBUTION ESTIMATES

Our scalable method differs from Lightcuts mainly in the way the cluster contribution estimates are calculated. We use two kinds of estimates. First, $\tilde{L}_c(\mathbf{x})$ denotes an estimate of the contribution of cluster c to a particular shading point $\mathbf{x}$. It is used as a prior distribution in our Bayesian learning model. Second, since we construct one cut per entire scene region, the cut construction needs an estimate $\tilde{L}_c(R)$ valid for all points in the respective region R.

We first discuss the *point estimate* $\tilde{L}_c(\mathbf{x})$. Unlike Lightcuts, we do not desire an upper bound, since it often drastically overestimates the actual contribution. Instead, we use less conservative estimates, so that our prior better matches actual contributions. We seek to

estimate the radiance due to direct illumination from cluster c:

$$L_c(\mathbf{x}) = \int_{A_c} \frac{L_e(\mathbf{y}\to\mathbf{x})V(\mathbf{y}\leftrightarrow\mathbf{x})\cos\theta_\mathbf{y}\cos\theta_\mathbf{x}}{d^2(\mathbf{y},\mathbf{x})}\,\mathrm{d}\mathbf{y}. \tag{20}$$

As in Lightcuts we use the same trivial bound for visibility $V = 1$ and upper bound $\overline{\cos}\theta_\mathbf{x}$ on the cosine at surface. We use also the original upper bound for the cosine at the light cluster, but only if the cluster center is further than 1.5 times the cluster diameter. For nearby clusters this bound would become too conservative and yield poor priors, so we average it with the cosine at the cluster center $\mathrm{ctr}(c)$, i.e., the cosine between the direction $\mathbf{x}-\mathrm{ctr}(c)$ and the axis of the cluster's normal cone. We denote the resulting cosine estimate as $\overline{\cos}\theta_c$. For the distance factor, we use a distance to the cluster center $d(\mathrm{ctr}(c),\mathbf{x})$. And finally for each light $l \in c$ we conservatively estimate radiance L_e it can emit to $\mathbf{x}$ and denote it $\overline{L}_{e,l}$. For instance, for cosine lights with emission defined as $I_0(\cos\theta_\mathbf{y})^\alpha$ this estimate can be obtained as $I_0(\overline{\cos}\theta_c)^\alpha$. Together we have:

$$\tilde{L}_c(\mathbf{x}) = \frac{\overline{\cos}\theta_c\overline{\cos}\theta_\mathbf{x}}{d^2(\mathrm{ctr}(c),\mathbf{x})}\sum_{l\in c}|A_l|\overline{L}_{e,l}. \tag{21}$$

On the other hand, the *region-wide estimate* $\tilde{L}_c(R)$ is more conservative so as to produce better cuts (it is less prone to a premature stop of the cut construction because of underestimating parent clusters). We construct it as an upper bound of $\tilde{L}_c(\mathbf{x})$ over all points in region R by bounding its individual factors. A trivial bound is used for the cosine at surface since the surface normal in the region may be arbitrary. To bound the cluster cosine with respect to the entire region, we enlarge the cluster bounding box by the region box [Walter et al. 2006]. The distance between the cluster and the region is bounded from below. Finally, emitted radiance is bounded using maximum radiance a cluster light can contribute to any point in the region (similarly as in $\tilde{L}_c(\mathbf{x})$ but using the region-wide bound on the cluster cosine).

See the Supplemental for discussion of importance of $\tilde{L}_c(\mathbf{x})$ accuracy and analysis of impact of the clustering on the method performance.

B OPTIMAL CLUSTER SAMPLING PROBABILITIES

In this section we derive the optimal cluster sampling distribution $P_{\mathrm{opt}}(c|\mathbf{x})$ from Sec. 4. To achieve that, we minimize the variance $\mathrm{Var}[\langle L(\mathbf{x})\rangle]$ given by Eq. (4) with respect to the cluster sampling probabilities $P(c|\mathbf{x})$, subject to $\sum_{c\in C}P(c|\mathbf{x}) = 1$.

Let us denote $w_c = P(c|\mathbf{x}), c \in C$, where C is the set of clusters. We further define $\mathbf{w} = (w_{c_1},\ldots,w_{c_{|C|}})$ and $m_{2,c}$ as in Eq. (4). Next, we set up a lagrangian $\mathbb{L}(\mathbf{w},\lambda)$

$$\mathbb{L}(\mathbf{w},\lambda) = -L(\mathbf{x})^2 + \left(\sum_{c\in C}\frac{1}{w_c}m_{2,c}\right) + \lambda\left(\sum_{c\in C}w_c - 1\right), \tag{22}$$

where $\lambda\in\mathbb{R}$ and we seek a solution $\mathbf{w},\lambda$ of the equation $\nabla\mathbb{L}|_{\mathbf{w},\lambda} = \mathbf{0}$, yielding the following set of equations:

$$\begin{aligned}\frac{d}{dw_c}\mathbb{L}(\mathbf{w},\lambda) &= -\frac{1}{w_c^2}m_{2,c} + \lambda = 0,\\ \frac{d}{d\lambda}\mathbb{L}(\mathbf{w},\lambda) &= \sum_{c\in C}w_c - 1 = 0.\end{aligned} \tag{23}$$

The solution is $w_c = \sqrt{\frac{1}{\lambda}m_{2,c}}$ and $\lambda = \left(\sum_{c\in C}\sqrt{m_{2,c}}\right)^2$, where λ serves as a normalization factor making the w_c sum up to one. In other words, the optimal cluster sampling probability $P_{\mathrm{opt}}(c|\mathbf{x})$ is proportional to the square root of the second moment $m_{2,c}$.

C CONJUGATE PRIORS FOR OUR MODEL

Setting $p(\theta) = p(p_o)p(k,h)$ in the relation $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$, the posterior $p(\theta|\mathcal{D})$ will be proportional to:

$$p(p_o)p(k,h)\left(\prod_i^{N_o}\delta(\hat{e}_{\mathbf{x},i})p_o\right)\left(\prod_i^{N_v}(1-p_o)\mathcal{N}\left(\hat{e}_{\mathbf{x},i}\;\middle|\;\frac{k}{\hat{d}_i^2},\frac{h}{\hat{d}_i^4}\right)\right). \tag{24}$$

Beta prior. To get the posterior distribution of p_o, we need to divide the above expression (24) by the marginal distribution $p(\mathcal{D},k,h)$ which we get by integrating out p_o from (24). By doing so we get the posterior in the form:

$$p(p_o|\mathcal{D},k,h) = K\,p(p_o)\,(1-p_o)^{N_v}p_o^{N_o}, \tag{25}$$

where K is some normalization factor depending only on the data $\mathcal{D}$ and our choice of the prior $p(p_o)$. We see that $(1-p_o)^{N_v}p_o^{N_o}$ is of the same form as the Beta distribution. Therefore by setting $p(p_o) = \mathrm{B}(p_o|\hat{N}_o,\hat{N}_v)$ we are now able to evaluate K from (25) and we get the posterior distribution

$$p(p_o|\mathcal{D},k,h) = \mathrm{B}(p_o|\hat{N}_o+N_o,\hat{N}_v+N_v). \tag{26}$$

We see that the Beta distribution is indeed a conjugate prior of our model from Eq. (8).

Normal-inverse-gamma prior. To find a conjugate prior for the k and h parameters, we proceed similarly as before with p_o. We get the posterior distribution of the form:

$$p(k,h|\mathcal{D},p_o) = K\,\hat{d}_i^2\,p(k,h)\prod_i^{N_v}\mathcal{N}\left(\hat{e}_{\mathbf{x},i}\hat{d}_i^2|k,h\right), \tag{27}$$

where K is again some normalization constant and we used the relation $\mathcal{N}(\hat{e}_{\mathbf{x},i}|k/\hat{d}_i^2,h/\hat{d}_i^4) = \hat{d}_i^2\mathcal{N}(\hat{e}_{\mathbf{x},i}\hat{d}_i^2|k,h)$. Normal-inverse-gamma $\mathcal{N}$-Γ^{-1} distribution is a conjugate prior for such a case [Bishop 2006]. Therefore it is a conjugate prior for our model (8).

D MAP FOR DIRECT ILLUMINATION SAMPLING

Plugging Eq. (8) and (9) into Eq. (11) yields the following system of equations.

$$\begin{aligned}\frac{(\hat{N}_o-1)(1-p_o)-p_o(\hat{N}_v-1)}{p_o(1-p_o)} - \frac{N_o}{p_o} - \frac{N_v}{1-p_o} &= 0\\ \frac{s_{1,\mathbf{x}} - k(\hat{N}+N_v)+\hat{N}\mu_0}{h} &= 0\\ \frac{-2ks_{1,\mathbf{x}}+s_{2,\mathbf{x}}-2\hat{N}_\alpha h+2\beta+\hat{N}(\mu_0-k)^2+N_v(k^2-h)+h}{h} &= 0.\end{aligned}$$

The solution to this system gives us the MAP solution in Eq. (12), (13) and (14).