Ray Guiding for Production Lightmap Baking

Ari Silvennoinen
Activision Publishing
ari.silvennoinen@activision.com

Peter-Pike Sloan
Activision Publishing
ppsloan@activision.com

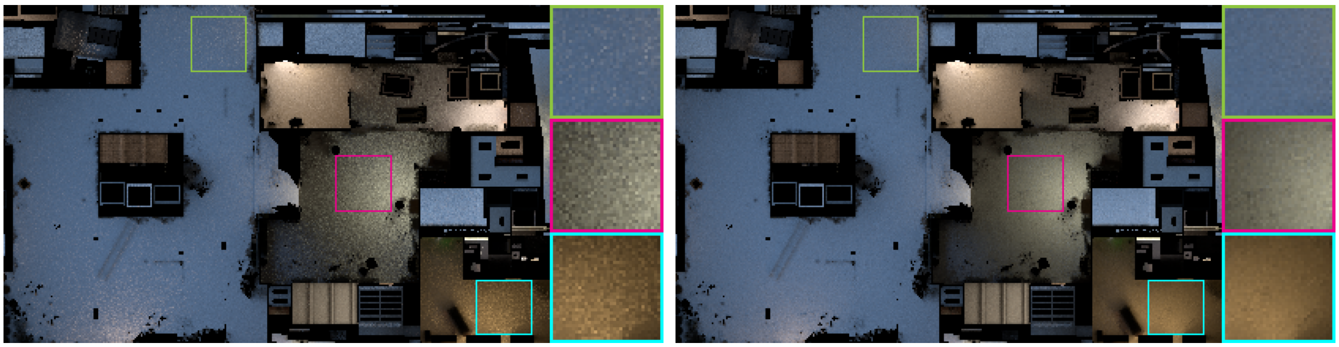


Figure 1: (Left) Noisy lightmap. (Right) In equal time, we obtain a result comparable to using 2x-10x more samples. ©Activision Publishing, Inc.

ABSTRACT

We present a ray guiding technique for improving the computation times in the context of production lightmap baking. Compared to state-of-the-art, our method has better scalability and lower variance.

ACM Reference Format:
Ari Silvennoinen and Peter-Pike Sloan. 2019. Ray Guiding for Production Lightmap Baking. In *SIGGRAPH Asia 2019 Technical Briefs (SA '19 Technical Briefs), November 17–20, 2019, Brisbane, QLD, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3355088.3365167

1 INTRODUCTION

Computing global illumination in large scene is often too costly for a real-time computation, necessitating the need to perform the processing offline. However, this introduces a new problem: the long computation times associated with light baking make fast iteration – highly desirable in production– impossible [O'Donnell 2018]. Computation time is usually dominated by ray tracing in order to tame the long tails of the Monte Carlo (MC) integration of light transport. While efficient methods exist for explicit importance sampling lights and BRDFs, strong secondary light sources together with the complicated visibility term remain a challenge for both sampling and denoising.

Path guiding. To improve the convergence of MC estimators, there has been a recent surge of work related to path guiding [Vorba et al. 2019]. The key idea is to collect samples of light transport and learn a probability distribution for importance sampling [Müller et al. 2017; Reibold et al. 2018; Vorba et al. 2014]. Unlike analytical importance sampling methods, the guiding process is usually designed in a way that it can take into account the light modulated *visibility* term, which is often impractical to tackle analytically in a closed form. Our work is inspired by the recent practical path guiding algorithm by Mueller et al. [2017]. In contrast to their approach, our method runs in fixed memory footprint, *independent of scene complexity*. In addition, we are able to combine information from *all* the learning samples, leading to a more efficient estimator with reduced variance.

Combining multiple estimators. Under normal assumptions, the problem of combining multiple estimators with a common mean but potentially different variances was considered by Graybill and Deal [1959]. In the context of light transport, Rousselle et al. [2016] considered the problem of optimal weighting of estimators with different sampling strategies. However, there has been little work on combining biased estimators – which often have lower variance – with an explicit control on the bias. Finally, we note that similar ideas were considered in a concurrent work by Mueller [2019]: we show comparisons to state-of-the-art.

Our work combines guided sampling and optimal estimator combination, resulting in a practical ray guiding approach. In particular, we present the following contributions:

- A method for reusing information from *all* the learning samples (Section 2.5)
- A technique for filtering the guide distribution to prevent overfitting (Section 2.3)
- A coherent sample warping technique (Section 2.4)

2 METHOD

Our ray guiding technique is a part of a light baking system based on a series expansion interpretation of light transport. The light baker works by computing diffuse global illumination one bounce at a time, making explicit direct light connections on the first and last bounce. We apply our ray guiding in the final gather step to

SA '19 Technical Briefs, November 17–20, 2019, Brisbane, QLD, Australia

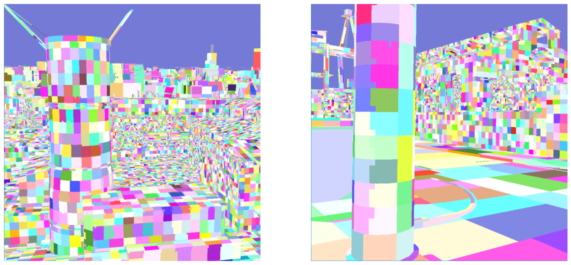


Figure 2: Visualization of texel clusters used for sharing the guide distribution.

better capture strong indirect light sources, which, in contrast to next event estimation used for light sources, there is no explicit sampling strategy available.

To build the guided sampling distribution, we apply the principle of *borrowing strength* by clustering coherent texels together (Section 2.1) and learning an ensemble estimator of the incoming radiance for each cluster over a fixed number of learning iterations (Section 2.2). Similar to Mueller et al. [2017], we represent the spherical directional guide distribution in the primary sample space using a piecewise constant quadtree. However, in contrast to Mueller et al., we do not employ an explicit binary tree over the positions, and, in particular, our solution runs in a scene-independent fixed memory footprint, allowing us to scale the ray guiding method for arbitrarily large worlds (Section 2.2). After building the initial probability density function (PDF), we improve the behavior of the learned distribution under low sample counts by applying a relaxation step to avoid overfitting (Section 2.3). In Section 2.4, we describe a modification to the hierarchical sample warping technique to bring down the cost of using our ray guiding suitable for production environments. Finally, we perform a weighted combination of multiple texel estimators – each corresponding to a learning iteration – to minimize variance (Section 2.5).

2.1 Clustering texels

To reduce variance in the learned ray distribution, we amplify the number of learning rays used to construct the PDF by sharing the PDF between clusters of spatially and geometrically coherent texels. We determine these clusters by constructing a 3D hierarchy over positions of lightmap texels. This 3D tree is built from top-down by splitting the bounding volume over the longest axis until a maximum leaf texel count of 64 texels is reached. In addition, we enforce a constraint that the maximum normal deviation within each leaf node is at most 60 degrees, splitting the node if the constraint is violated. This additional normal cone constraint tends to result in more geometry-aware clusters, and works with both hard edges found in building corners as well as soft edges found in cylindrical objects, as can be seen in Figure 2.

2.2 Building the guiding distribution

We process each cluster independently of others and we discard the guiding PDF as soon as we are done with a cluster. This streaming computation gurantees a small, scene-independent fixed memory footprint and is critical for supporting large scenes. For each cluster, we build the guide PDF by iteratively collecting ray samples. During each iteration, we loop over all the cluster texels and shoot final gather rays, which are importance sampled using the guide distribution from the previous iteration (Section 2.4). After tracing the rays for an entire cluster, we build a directional quadtree over the luminance values of the radiance samples in primary sample space by applying an area-preserving map from the sphere to the unit quad [2017]. We split nodes until they contain fewer than 4 samples or until the node flux is less than 1% of the total flux. For each node we store the flux contained in the samples or a low ambient probability, whichever is higher, resulting in a function that is non-zero over the entire domain. Finally, we normalize the node values, resulting in a piecewise constant probability distribution function.

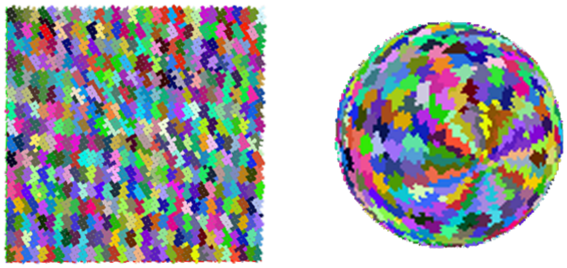
2.3 Filtering the learned distribution

Even though we share all the texel samples within each cluster to learn the PDF, there is a risk of overfitting to the samples that can lead to slower convergence if the learned PDF does not reflect the population distribution well enough. To mitigate this effect, we apply a relaxation step by running a hierarchical smoothing filter over the quadtree representation of the guide distribution. For a given smoothing parameter α, we traverse the hierarchy from top to bottom, visiting each node exactly once and linearly interpolating the flux of each child towards the *average* value of their parent, determined by the filter parameter α (Figure 7). Interestingly, in a concurrent work, Mueller had a similar observation while adapting their path guiding system to a production environment [Vorba et al. 2019]. However, they used a splatting kernel approach, and resorted to a stochastic filter approximation due to the high cost of evaluating their full splatting filter.

2.4 Efficient hierarchical sample warping

To draw samples from the guide PDF, we apply a hierarchical sample warping technique [Clarberg et al. 2005], allowing us to do importance sampling with quasi Monte Carlo (QMC) point sets for further variance reduction due their better stratification properties. To make this process more efficient, we would like to utilize SIMD-vectorization to leverage the available computing power in modern CPUs. However, a naive vectorization by warping, say 8 samples at a time is not effective, since each lane in the vector can potentially have a different tree traversal path during the warping step. Furthermore, the obtained ray packets do not exhibit directional coherence, leaving little opportunity to obtain additional speed ups from ray packet tracing.

To enable vectorized hierarchical sample warping and coherent ray packet traversal, we observe that once the sample count is fixed, any reordering of the samples leads to the same result, i.e., the estimators are permutation invariant. This allows us to sort the samples into directionally coherent packets. Given a fixed sample count and a 2D QMC-point sequence, we build a quad-tree over the directional domain until the leaf nodes contain less than 8 samples. In a subsequent step, we traverse the leafs and output sorted samples in order (Figure 3). This process is done once before the light bake and the presorted sample sequences are reused for



(a) Rays in primary sample space (b) Rays on the sphere

Figure 3: Sorted ray packets (one color per packet) used for sample warping and coherent ray packet tracing.

each texel by decorrelating each sequence using Cranley-Patterson rotation [Cranley and Patterson 1976].

The SIMD-vectorized coherent sample warping brought the total overhead of the ray guiding from 25-30 % down to 3-5% in terms of the total light bake time, thus proving to be a key technique for making the guiding technique production friendly (Figure 5).

2.5 Combining the estimators

For each iteration k, we estimate the mean $\mu_k = E[f(x)/p_k(x)]$ and variance $\sigma_k^2 = \mathrm{Var}[f(x)/p_k(x)]$ using the usual estimators $\widehat{\mu_k}$ and $\widehat{\sigma_k}^2$ given by

$$\widehat{\mu_k} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p_k(x_i)} \tag{1}$$

$$\widehat{\sigma}_k^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \widehat{\mu}_k)^2, \tag{2}$$

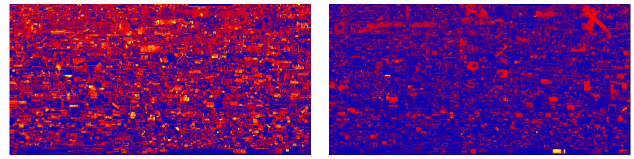
where N is the number of rays, p_k is the guiding density for iteration k, and f is the target integrand, i.e., the irradiance function.

The sample means for each batch are asymptotically normal, and the variance minimizing weighted combination of the batch estimators $\widehat{\mu}_k$, also known as the Greybill-Deal estimator $\widehat{\mu}_{GD} = \sum_k w_k \widehat{\mu}_k$, is given by using weights w_k proportional to the inverse variance [Graybill and Deal 1959]:

$$w_k = \frac{\widehat{\sigma}_k^{-2}}{\sum_j \widehat{\sigma}_j^{-2}} \tag{3}$$

From a Bayesian perspective, this is equivalent with the MAP-estimator for the expected radiance under normal assumptions of the batch means. Intuitively, this makes sense: each iteration gives new information and it is incorporated to the result according to the confidence we have in the batch, characterized by the batch variance.

To obtain unbiased estimator, we could use sample splitting or other techniques to decorrelate the mean and variance estimates at the cost of inflated variance [Nelson 1990]. Instead, we choose to estimate the batch mean μ_k and batch variance σ_k^2 using the *same* samples, and, in doing so, trade variance for bias. To correct for this bias, we estimate the expected luminance over the entire lightmap in two ways: (1) using the Graybill-Deal weighted average luminance, giving us a biased estimate $\widehat{L}_{GD}$ and (2) using a non-weighted average of the estimators, giving as a non-biased estimate $\widehat{L}$. This allows us to estimate a global, bias correcting scaling factor ρ as the ratio of the unbiased and biased expected luminances, i.e., $\rho = \widehat{L}/\widehat{L}_{GD}$. Finally, as a post-processing step, we apply this luminance preserving bias correction by scaling the RGB values by the scaling factor ρ (Figure 4).



(a) Without bias correction (b) With bias correction using ρ

Figure 4: Visualization of 8x absolute error over the lightmap. (a) Using same samples to estimate both mean and variance causes bias. (b) The bias is significantly reduced after applying our bias correcting scale.

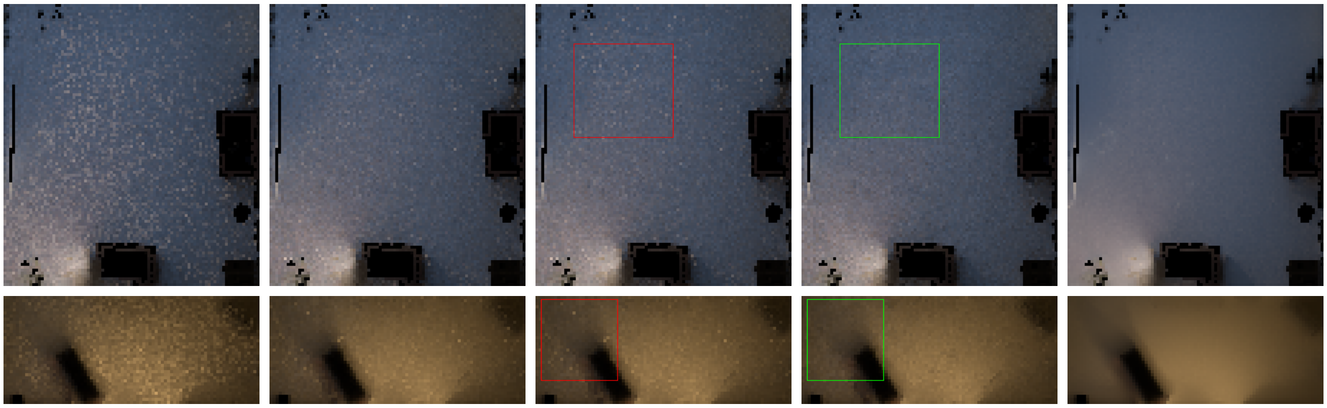
Guided	Guided and coherent	Baseline
2301s (1.28x)	1876s (**1.05x**)	1791s (1x)

Figure 5: Total computation times compared to baseline for a shipping quality light bake.

3 RESULTS

Our ray guiding approach has been integrated into a production light baking pipeline. To evaluate our method, we show comparisons against a baseline method using equal number of samples, guided sampling with equal weights, using a weighting scheme from a concurrent work [Vorba et al. 2019], and our bias corrected variance weighting scheme (Figure 6). Using a guide distribution (Figure 6b) reduces variance compared to the baseline method (Figure 6a), but the result can still contain fireflies, i.e., samples with high energy and low probability. The global variance weighting scheme proposed by Mueller in a concurrent work [2019] yields an image which is visually indistinguishable from the simple average weighted guided sampling (Figure 6c). This is not surprising, since the weights are global per image, leaving little to no capacity for local adaptivity. In contrast, our local, per-texel variance weights are effective in downweighting estimators with high variance, and, consequently, in removing the remaining fireflies (Figure 6d) while remaining close to the ground truth (Figure 6e).

We show the effect of our global bias reduction scheme in Figure 4. The heatmap visualizes the 8x magnified absolute error distribution over the lightmap from a single production map. Our expected luminance preserving scheme reduces bias in most areas of the map. However, since it is based on global estimate, it can also increase bias in areas that would have remained bias free. We leave the investigation of more locally adaptive luminance preserving bias reduction as future work.



(a) Baseline (b) Guided sampling (c) Global variance (Mueller) (d) Ours (e) Reference

Figure 6: (a) Baseline method using 256 final gather rays per texel. (b) Guided sampling with estimator averaging. (c) Guided sampling with a weighting scheme proposed in a concurrent work [Vorba et al. 2019]. (d) Guided sampling with our bias-corrected optimal variance weights. (e) Ground truth.

(a) $\alpha = 0$ (b) $\alpha = 0.25$ (c) $\alpha = 0.5$ (d) $\alpha = 0.75$ (e) $\alpha = 1$

Figure 7: (a) Without smoothing, the learned distribution can overfit data. (b-d) With moderate smoothing, the unwanted artifacts disappear. (e) For high smoothing values, the variance increases due to less effective guidance.

The optimal estimator combination can help to reduce variance even in the cases where guided sampling is ineffective (Figure 8). This firefly reduction effect is explained by the usage of multiple estimators: the probability that *every* estimator contains an outlier decreases as the number of estimators increases.

We show the effect of applying a smoothing filter to the guide distribution with various filter parameters in Figure 7. Without any smoothing, the guiding distribution is overfitted to the samples and this can lead to clustering artifacts (Figure 7a). With moderate smoothing, the artifacts disappear (Figure 7b-d). With heavy smoothing, the PDF will approach a uniform one and the results will be more noisy due to less effective ray guiding. In practice, we use a fixed filter value of 0.25 for all our maps.

4 CONCLUSION

We described a ray guiding technique for production lightmap baking based on two key ideas: (1) guided sampling and (2) minimum variance based estimator combination.

(a) Baseline (b) Guided + average (c) Guided + optimal

Figure 8: Variance minimizing combination of estimators helps even when the guide distribution is ineffective.

REFERENCES

P. Clarberg, W. Jarosz, T. Akenine-Möller, and H.W. Jensen. 2005. Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 24, 3 (2005).

R Cranley and T.N.R. Patterson. 1976. Randomization of Number Theoretic Methods for Multiple Integration. *SIAM Journal of Numberical Analysis* 13, 6 (1976), 904–914.

Franklin A. Graybill and R.B. Deal. 1959. Combining Unbiased Estimators. *Biometrics* 15, 4 (1959), 543–550. https://doi.org/10.2307/2527652

T. Müller, M. Gross, and J. Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Computer Graphics Forum (Proceedings of EGSR)* 36, 4 (2017).

Barry L. Nelson. 1990. Control Variate Remedies. *Oper. Res.* 38, 6 (Dec. 1990), 974–992.

Y. O'Donnell. 2018. Precomputed Global Illumination in Frostbite. *GDC Talks* (2018).

F. Reibold, J. Hanika, A. Jung, and C. Dachsbacher. 2018. Selective Guided Sampling with Complete Light Transport Paths. *ACM Trans. Graph.* 37, 6 (Dec. 2018).

F. Rousselle, W. Jarosz, and J. Novák. 2016. Image-space Control Variates for Rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 35, 6 (2016).

Jiří Vorba, Johannes Hanika, Sebastian Herholz, Thomas Müller, Jaroslav Křivánek, and Alexander Keller. 2019. Path Guiding in Production. In *ACM SIGGRAPH 2019 Courses (SIGGRAPH '19)*. ACM, New York, NY, USA, Article 18, 77 pages.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014)* 33, 4 (aug 2014).