Out-of-the-loop Autotuning of Metropolis Light Transport with Reciprocal Probability Binning

K. Herveau[1], H. Otsu[1,2], and C. Dachsbacher[1]

[1]Karlsruhe Institute of Technology, Institute for Visualization and Data, Germany
[2]McGill University, Centre for Intelligent Machines, Canada

Abstract

The performance of Markov Chain Monte Carlo (MCMC) rendering methods depends heavily on the mutation strategies and their parameters. We treat the underlying mutation strategies as black-boxes and focus on their parameters. This avoids the need for tedious manual parameter tuning and enables automatic adaptation to the actual scene. We propose a framework for out-of-the-loop autotuning of these parameters. As a pilot example, we demonstrate our tuning strategy for small-step mutations in Primary Sample Space Metropolis Light Transport. Our σ-binning strategy introduces a set of mutation parameters chosen by a heuristic: the inverse probability of the local direction sampling, which captures some characteristics of the local sampling. We show that our approach can successfully control the parameters and achieve better performance compared to non-adaptive mutation strategies.

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing;***

1. Introduction

Markov chain Monte Carlo (MCMC) rendering methods have been an active area of research in recent years. MCMC rendering uses a correlated sequence of samples: transport paths representing the state of the Markov chain, where the next state is iteratively generated based on the current one. Rendering efficiency depends on how well the path space is explored, and MCMC methods differ in how new paths are obtained using *mutation strategies*. Each strategy has one or more parameters which control the perturbation of the samples. A poor choice of parameter values results in unfavorable exploration and higher variance, e.g. due to highly-correlated Markov chains.

In this paper, we propose a framework for *out-of-the-loop autotuning* of the mutation kernel parameter, which is agnostic to the underlying mutation strategy. Our framework decouples the tuning process from rendering, allowing us to also tune strategies with complex parameterizations, and to use generic optimization techniques.

We showcase our novel σ-*binning* strategy for parameter tuning using small-step mutations of Primary Sample Space Metropolis Light Transport [KSAC02]. Instead of only tuning a single (global) scaling parameter, our σ-binning maintains a set of scaling parameters for mutations which are chosen depending on the probability density function (`pdf`) of local direction sampling. The intuition is that the scaling parameter should depend on surface properties. Note that this `pdf` can be evaluated independently of the actual mutation strategy, i.e. the latter can still be considered as a black-box. Our experiments confirm that autotuning can successfully control the parameters of the σ-binning strategy.

2. Background and Related Work

Path Integral. The path integral [Vea98] determines the intensity of a pixel: $I = \int_{\mathcal{P}} f(\bar{x}) d\mu(\bar{x})$, where $\mathcal{P}$ is the *path space*, the set of all transport paths of all possible lengths. An individual *path* $\bar{x} \in \mathcal{P}$ is defined by a sequence of (surface) points; $d\mu$ is the product area measure. The measurement contribution function f is the contribution of a single path. For brevity, we omit the pixel index and assume the reconstruction filter being integrated into f.

Primary Sample Space. We present our approach for Primary Sample Space MLT (PSSMLT) [KSAC02]. Here each path $\bar{x}$ is defined by a vector of random numbers $\bar{u} \in \mathcal{U} = [0,1)^n$. A path sampling strategy, e.g. path tracing, maps elements $\bar{u}$ of the *primary sample space* $\mathcal{U}$ to actual paths, and consequently n is the number of required random numbers. Using $\mathcal{U}$ as the state space simplifies MCMC rendering compared to the original MLT [Vea98] where the path space serves as the state space.

Let $S : \mathcal{U} \to \mathcal{P}$ be the path sampling function mapping a primary sample $\bar{u}$ to a path $\bar{x}$. By the change of variable, the path integral can be written as

$$I = \int_{\mathcal{U}} f(S(\bar{u})) \left| \frac{dS(\bar{u})}{d\bar{u}} \right| d\bar{u} = \int_{\mathcal{U}} \frac{f(S(\bar{u}))}{p(S(\bar{u}))} d\bar{u} = \int_{\mathcal{U}} \tilde{C}(\bar{u}) d\bar{u}, \quad (1)$$

DOI: 10.2312/egs.20231005

where $p(\bar{x})$ is the pdf (with respect to $d\mu$) of the path $\bar{x}$ generated by the function S. Here, $\left|\frac{dS(\bar{u})}{d\bar{u}}\right| = \frac{1}{p(S(\bar{u}))}$ and $\tilde{C}(\bar{u}) = \frac{f(S(\bar{u}))}{p(S(\bar{u}))}$.

MH Update. PSSMLT uses the Metropolis-Hastings (MH) algorithm with the *scalar* contribution function $\tilde{C}^*(\bar{u})$ as target density; typically using the luminance of $C(\bar{u})$. It iteratively updates the current sample to the next. Given the current $\bar{u}$, the algorithm first proposes a tentative sample $\bar{v}$, which is obtained using the *transition kernel* $T(\cdot \mid \bar{u})$. The proposed sample $\bar{v}$ is either accepted or rejected according to the *acceptance probability*

$$a(\bar{v} \mid \bar{u}) = \min\left(1, \frac{\tilde{C}^*(\bar{v})T(\bar{u} \mid \bar{v})}{\tilde{C}^*(\bar{u})T(\bar{v} \mid \bar{u})}\right). \tag{2}$$

If accepted, $\bar{v}$ becomes the next state; otherwise, $\bar{u}$ is also kept as the next state. The N-sample estimate of the path integral then becomes $\hat{I} = \frac{b}{N}\sum_{i=1}^{N}\frac{\tilde{C}(\bar{u}_i)}{\tilde{C}^*(\bar{u}_i)}$, where $b := \int_{\mathcal{U}}\tilde{C}^*(\bar{u})d\bar{u}$ is the normalization constant. The constant b is typically estimated once using ordinary MC techniques.

Mutation Techniques. PSSMLT uses two mutation strategies: large-step and small-step mutations. The large-step mutation reinitializes the state with uniform random numbers for global exploration. The small-step mutation explores locally by perturbing each random number of the current state $\bar{u}$ independently with $\mathcal{N}(u_i, \sigma)$.

Adaptive Mutations. Several approaches strive to control the perturbation (parameters) for MCMC rendering. Zsolnai et al. [ZS13] propose to control the selection probability for the large-step mutations based on gathered statistics. The approach by Li et al. [LLR*15] adapt to the local structure of the state space by using the derivatives of the integrand. Hachisuka et al. [HJ11] applied adaptive MCMC in the context of photon tracing. Otsu et al. [OHHD18] adapted the mutation to the surrounding geometry around a path. These approaches rely on domain knowledge obtained from either the target function or the generated samples, thus they are tightly coupled to the specific mutation strategy. In contrast, we aim at not being dependent on the mutation strategy.

Autotuning and Optimization Autotuning and machine learning have proven to be useful for various applications in computer graphics. Currently, optimizing program parameters is often done ahead of time with tools such as OpenTuner [AKV*14], which uses a variety of methods, including Nelder-Mead [NM65], of which we use a more recent variation [Cha12] in our work.

3. Method Overview

Motivation. In this section we present our method which treats the rendering process as a black-box whose performance depends on a set of input parameters, i.e. our method is independent of the underlying technique. The autotuning process only modifies these inputs and assesses the output without using domain knowledge of the rendering process. Since the optimization takes place outside of this black-box, we name it out-of-the-loop autotuning.

To go beyond demonstrating our framework for a single, global parameter, we introduce more fine-grained tuning capabilities with our novel σ-binning (Section 4). This increased granularity (more tuning parameters) would be more difficult for a user to control and thus can be considered a good application for autotuning.

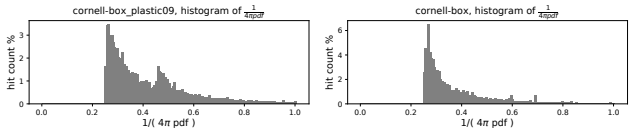

Figure 1: *Bin landscapes with two identical scenes except for one of the boxes having a plastic material. The bump corresponding to the different materials is visible at 0.45 on the x-axis.*

Top-Level Optimization Loop. Starting from an initial set of parameters, out-of-the-loop autotuning executes the rendering process with a fixed number of samples. Next, the metric to be optimized (e.g., a perceptual metric, variance, rendering time, acceptance ratio, etc.) is computed from the outputs of the rendering process. Based on the history of input parameters and the outputs, the autotuner determines the next set of candidate parameter values. The convergence time in our experiments was below 50 iterations, as motivated by the leftmost Figure 2, after which we continue rendering using the best configuration found.

4. Parameterization and Autotuning with σ-binning

Variable Scaling for Local Direction Sampling. The original PSSMLT employs the single global parameter for the perturbation, which specifies the standard deviation of the per-dimensional normal distributions. When mapped through the path sampler, the perturbation of an element of the primary sample corresponds to the perturbation of the sampled local directions. Thus the directional perturbation occurs irrespective of the surface properties. Our idea for the new mutation technique is based on the observation that the scaling parameter should depend on the surface properties.

Parameterization. The `pdf` for BSDF sampling obtained during path creation will also depend on the different incoming directions and materials. Based on this insight, we classify the scaling parameters according to the reciprocal `pdf` of the local direction sampling. Figure 1 illustrates the influence of surface properties on the reciprocal of the `pdf` Let u be the element of the primary sample corresponding to the direction sampling. We define a function $\Theta(u) = \frac{1}{\alpha \cdot p_\sigma(\omega(u))}$, where $\omega(u)$ is the mapped direction using the path sampler and $p_\sigma(\omega(u))$ is the `pdf` of the local direction sampling with respect to the solid angle measure. Here, the parameter α controls the spread of the reciprocal `pdf`. Let M be the size of bins. We split the interval $[0,1)$ into partitions $A_1, A_2, \ldots, A_{M-1}$, where $0 < A_1 < \cdots < A_{M-1} < 1$. For convenience, $A_0 := 0$ and $A_M := 1$. Here, the interval $[A_{i-1}, A_i)$ corresponds to the i-th bin. We define the scaling parameters $\sigma_1, \sigma_2, \ldots, \sigma_M$ for each bin. Then the scaling parameter corresponding to the current state u is defined by $\sigma(u) = \sigma_i$ if $\Theta(u) \in [A_{i-1}, A_i)$.

We use the same mapping Θ for all local direction samplings, regardless of the number of bounces. The partition $A_1, \ldots, A_{M-1}$ and the scaling parameters $\sigma_1, \sigma_2, \ldots, \sigma_M$ are exposed as a set of parameters for the mutation strategy. To evaluate how impactful the bins will be, we used $M = 200$ equally spaced bins and counted how many samples fell in each to produce histograms (see Figure 1).

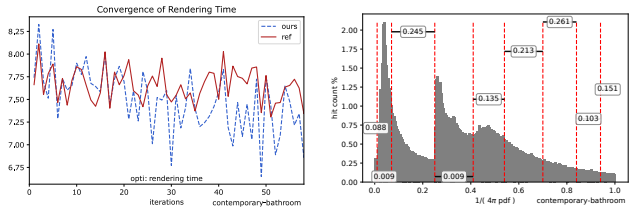

Figure 2: ***Left:*** *Convergence during tuning. Red line is a non-tuning run (always same configuration) to show the variance between iterations.* ***Right:*** *Different bin positions and σ-values have been optimized. Bins locate features and the values are in a sensibe range.*

The number of different prominent features is usually low in all the scenes we have tested, i.e. there are not many small, sharp peaks. The effect of tuning smaller features (those having a small area on the histogram) is easily drowned in the rendering noise, so having only a few bins is sufficient for our optimization. Higher bin count usually do not converge on more difficult scenes. The parameter α is configured to capture the spread of the reciprocal `pdf` in the predefined range $[0,1)$ (values outside are clamped). We found that this parametrization is consistent and stable accross scenes, so we use $\alpha = 4\pi$ for all scenes.

Bin Positions as Parameters. Including the bin positions into the optimization is critical for satisfactory results. Using between 5 and 10 bins produces similar results and tuning their number only makes convergence longer. We therefore decided to not include the number of bins as a parameter. We report a run with 8 different bins in Figure 2. This figure shows the bins correctly encircling interesting features. The first and second bin separate the leftmost sharp feature from the flat portion preceding the next sharp peak. The large (resp. small) σ values show where the samples in this remapped space require larger (resp. smaller) mutation size. Note that each bin has an effect proportional to the area of the histogram they contain. It is clear from this distribution that less bins could have been employed as the tail of the distribution seems to prefer uniformly higher values of σ. Separating the features is not our primary goal, it is a by-product of the autotuning process which informs us that using different σs for different features of this remapped space does improve the error metric.

Handling Constraints in Autotuning. The autotuner can only tune parameters between fixed bounds and cannot handle constraints between parameters. In order to impose the constraint $0 < A_1 < \cdots < A_{M-1} < 1$, we introduce a mapping of the partition $A_1, A_2, \ldots, A_M$ by a sequence of numbers $a_1, \ldots, a_{M+1}$ in $[0,1)$, defined by $A_i = (\sum_{j=1}^{i} a_j)/S$ for $i \in \{1, \ldots, M-1\}$, where $S = \sum_{i=1}^{M+1} a_i$. This mapping guarantees that the numbers $A_1, \ldots, A_{M-1}$ satisfy the desired constraint. Eventually, the autotuner optimizes the parameters $a_1, \ldots, a_{M+1}$ and $\sigma_1, \ldots, \sigma_M$. This corresponds to two tables of values. These values are optimized within predefined bounds: $[0,1)$ for each number of $a_1, \ldots, a_{M+1}$ and $[0.0001, 0.3]$ for the σ-values.

Optimization Criteria. The optimization criteria can be a variety of measurable quantities, such as rendering time, (r)RMSE, acceptance ratio etc. We choose to optimize with respect to the acceptance ratio. Indeed, there is an optimal value of the acceptance ratio, 0.234, that can be derived mathematically under certain assumptions [AT08]. We found it to be satisfactory in our case. Optimizing for rendering time yielded similar results.

5. Small-Step Mutation with σ-binning

Non-Symmetric Proposal. The original PSSMLT uses the fixed parameter as the standard deviation. Thus the transition probability between the current state $\bar{u}$ and the proposed state $\bar{v}$ can be considered symmetric, that is, $T(\bar{v} \mid \bar{u}) = T(\bar{u} \mid \bar{v})$. This simplifies the computation of the acceptance probability since the terms cancel out: $a(\bar{v} \mid \bar{u}) = \min\left(1, \tilde{C}^*(\bar{v})/\tilde{C}^*(\bar{u})\right)$. Our method uses state dependent mutation parameters, thus the symmetric property no longer holds, which necessitates the evaluation of the transition probabilities.

Truncated Normal Distribution. In the original PSSMLT, if the perturbed value is outside of the interval $[0,1)$, the value is *wrapped* to $[0,1)$. Thus the normal distribution on the domain is technically a *wrapped* normal distribution. The `pdf` of the wrapped normal distribution involves an infinite sum since the normal distribution has an infinite support. In order to avoid the complication with respect to the wrapping, our approach uses the *truncated* normal distribution. We configure the truncation interval to be $[-1/2, 1/2)$, so that for all proposal in $[0,1)$, there is only one way to perturb the current state to the proposal.

Let $\bar{u}$ and $\bar{v}$ be the current and proposed states, where the elements of each state are indexed by i. Since the mutation happens independently per dimension of the primary sample, we can write the acceptance probability as $a(\bar{v} \mid \bar{u}) = \min\left(1, \frac{\tilde{C}^*(\bar{v})}{\tilde{C}^*(\bar{u})} \cdot \prod_i R(u_i, v_i)\right)$, where $R(u_i, v_i) := T(u_i \mid v_i)/T(v_i \mid u_i)$ denotes the ratio of the transition probabilities for the i-th element. Let σ_u and σ_v be the standard deviations associated to the states u and v. We can derive the R term for u and v:

$$R(u,v) = \frac{\sigma_u}{\sigma_v} \cdot \exp\left(\frac{(v-u)^2}{2} \cdot \left(\frac{1}{\sigma_u^2} - \frac{1}{\sigma_v^2}\right)\right) \cdot \frac{\operatorname{erf}\left(\frac{\sqrt{2}}{2\sigma_u}\right)}{\operatorname{erf}\left(\frac{\sqrt{2}}{2\sigma_v}\right)}. \quad (3)$$

Since the number of bins is predefined, we can optimize the computation of the acceptance probability by caching the terms related to the function erf, which drastically reduces the number of costly operations down to a simple read.

Sampling Proposal. Let u be an element of the current primary sample. We use the standard inverse transform method to generate a sample from the truncated normal distribution. Let ξ be a random number uniformly distributed in $[0,1)$. Then the random variable $\delta := F^{-1}(F(-1/2) + \xi \cdot (F(1/2) - F(-1/2)))$ is distributed according to the truncated normal distribution on $[-1/2, 1/2)$. Here, $F(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma_u}\right)\right)$ is the cumulative distribution function (cdf) of the standard normal distribution. Then the proposed state v can be obtained by taking the fractional part of $u + \delta$, that is, $v = (u+\delta) - \lfloor u + \delta \rfloor$.

6. Evaluation

We have implemented our method in pbrt version 3 [pha17]. We compared our method (*ours*) against the original small-step muta-

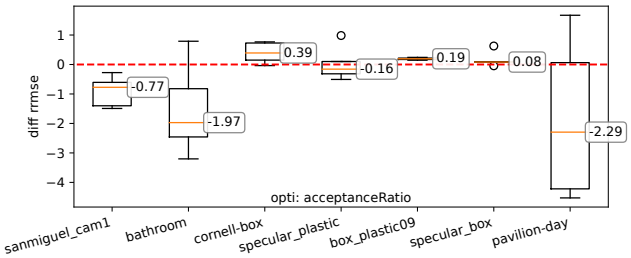

Figure 3: *The distribution of the quality difference between* ours *and* original *on different scenes (lower is better). We observe that more complex scenes benefit more from our method.*

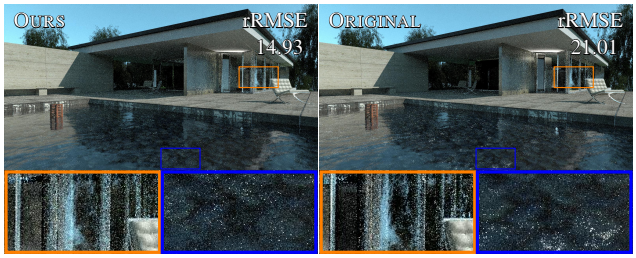

Figure 4: *Equal-sample (100 mpp) rendering of* pavilion-day. *Our method achieves 1.3 seconds (4%) faster rendering time. The water surface has much fewer fireflies, the windows on the right are correctly resolved as well. This behavior is consistent with Figure 3.*

tion (*original*), which uses the fixed scaling parameter $\sigma = 0.01$. We treat the large-step probability as a hyperparameter, and chose a value that produces better results for each technique; *ours*: 0.1, *original*: 0.3. All experiments are conducted on a machine with 8x Intel Xeon E7-8867 CPUs with 4TB RAM with 256 threads. For the experiments, we chose scenes with diverse characteristics: *sanmiguel*, *contemporary-bathroom*, *pavilion-day* and *cornell-box*.

For comparisons, we use relative root mean square error (rRMSE). We optimize for acceptance ratio and error values are computed with respect to reference images, computed with 16384 spp with bidirectional path tracing. We repeated the measurements between 5 and 10 times to display the error spread. For tuning, we perform 50 tuning iterations, each with 5 mutations per pixel (mpp). We select the best configuration explored. The tuning step is a Nelder-Mead update. This update is extremely fast for our number of parameters (<1ms) and is negligible compared to rendering time. The rendering outputs generated during the entire tuning process are still useful. We created several variations of the *cornell-box* to highlight that different materials produce a different binning landscape (Figure 1), by changing the materials of the two boxes; *specular-plastic*: specular and plastic (0.9 roughness), *box-plastic*: diffuse and plastic, and *specular-box*: specular and diffuse.

We compare two approaches using the same number of samples: 100 mpp or 200 mpp depending on the scenes. Figure 3 summarizes the relative differences of rRMSE between the two approaches. Our method performs well on the scenes with more specular paths, as shown in *pavilion-day* (Figure 4) and *contemporary-bathroom* (Figure 5). We observe faster rendering for the former, and far fewer artifacts for the latter.

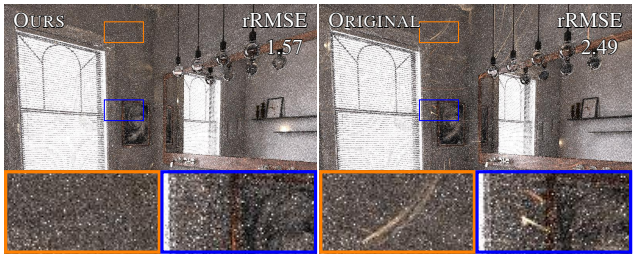

Figure 5: *Equal-sample (200 mpp) rendering of* contemporary-bathroom. *The original implementation produces sharp features that are notoriously difficult to denoise whereas our approach presents far fewer of these artifacts, and a better rRMSE.*

7. Conclusion

We have presented a framework for automatically tuning mutation kernel parameters for MCMC rendering. We propose an adaptive small-step mutation parameterization for PSSMLT called σ-binning, based on a heuristic to categorize the mutation size parameter according to the reciprocal pdf of the local sampling. We demonstrated that the approach can result better rendering performance, showing the potential benefit of using autotuning in the context of MCMC rendering.

References

[AKV*14] ANSEL J., KAMIL S., VEERAMACHANENI K., RAGAN-KELLEY J., BOSBOOM J., O'REILLY U.-M., AMARASINGHE S.: OpenTuner: An extensible framework for program autotuning. ACM Press, pp. 303–316. 2

[AT08] ANDRIEU C., THOMS J.: A tutorial on adaptive MCMC. *Statistics and Computing 18* (Dec. 2008). 3

[Cha12] CHANG K.-H.: Stochastic Nelder-Mead Simplex Method. *European Journal of Operational Research 220*, 3 (Aug. 2012), 684–694. 2

[HJ11] HACHISUKA T., JENSEN H. W.: Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph. 30*, 5 (Oct. 2011), 114:1–114:11. 2

[KSAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. *Computer Graphics Forum* (2002). 1

[LLR*15] LI T.-M., LEHTINEN J., RAMAMOORTHI R., JAKOB W., DURAND F.: Anisotropic Gaussian mutations for metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Trans. Graph. 34*, 6 (Nov. 2015), 209:1–209:13. 2

[NM65] NELDER J. A., MEAD R.: A Simplex Method for Function Minimization. *The Computer Journal 7*, 4 (Jan. 1965). 2

[OHHD18] OTSU H., HANIKA J., HACHISUKA T., DACHSBACHER C.: Geometry-aware metropolis light transport. *ACM Trans. Graph. 37*, 6 (Dec. 2018), 278:1–278:11. 2

[pha17] Physically Based Rendering. In *Physically Based Rendering (Third Edition)*, Pharr M., Jakob W., Humphreys G., (Eds.). Morgan Kaufmann, Boston, Jan. 2017, p. 1235. 3

[Vea98] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, Stanford, USA, 1998. 1

[ZS13] ZSOLNAI K., SZIRMAY-KALOS L.: Automatic parameter control for metropolis light transport. In *Eurographics 2013 - Short Papers* (2013), Eurographics Association, pp. 53–56. 2