SAARLAND UNIVERSITY

MARS: MULTI-SAMPLE ALLOCATION THROUGH RUSSIAN ROULETTE AND SPLITTING

AUTHOR

JOSHUA MEYER

Student No. 2573323

Master Thesis in Computer Science

Chair for Computer Graphics

Faculty 6.2 – Computer Science

SUPERVISOR

PROF. DR. PHILIPP SLUSALLEK

SAARBRÜCKEN, JUNE 2024

Erklärung

Ich erkläre hiermit, dass ich, Joshua Meyer (2573323), die vorliegende Arbeit, "MARS: Multi-sample Allocation through Russian roulette and Splitting," selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Diese Arbeit entstand unter der Betreung von Professor Philipp Slusallek.

Eine kürzere Version dieser Arbeit wurde als Paper für SIGGRAPH Asia 2024 eingereicht. In diesem Rahmen habe ich mit meinen Co-Autoren Alexander Rath and Ömercan Yazici zusammengearbeitet. Beide haben zum Inhalt dieser Arbeit beigetragen. Eine detaillierte Beitragserklärung ist in Kapitel 8 zu finden.

Declaration of Authorship

I hereby confirm that I, Joshua Meyer (2573323), have written this thesis, entitled "MARS: Multi-sample Allocation through Russian roulette and Splitting," on my own and that I have not used any other media or materials than the ones referred to in this thesis. This work was done under the supervision of Professor Philipp Slusallek.

A shortened version of this thesis was submitted as a paper to SIGGRAPH Asia 2024. In this context, I worked together with my co-authors Alexander Rath and Ömercan Yazici. They both contributed to what is presented in this work. A more detailed contribution declaration can be found in chapter 8.

Saarbrücken, June 2024

Joshua Meyer

Einverständniserklärung

Ich erkläre hiermit, dass die vorliegende Arbeit mit der elektronischen Version übereinstimmt. Außerdem bin ich damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I hereby confirm the congruence of the contents of the printed data and the electronic version of the thesis. Moreover, I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, June 2024

Joshua Meyer

Acknowledgements

This work would not have been possible without the help and contributions of others.

First, I would like to express my deepest gratitude to my advisors, Alexander Rath and Ömercan Yazici. Despite working on their respective PhD theses, they found the time to discuss and guide me through my Master's thesis over the past year. Our weekly meetings and fruitful discussions were invaluable, encouraging and always helpful in determining the next steps. I am thankful for being a welcome, equal member of the computer graphics chair and for the opportunity to gain such great insight into research life. I wish you all the best and hope we stay in contact.

No less, I am grateful to my supervisor, Prof. Dr. Philipp Slusallek, for his continuous support and mentorship. His profound knowledge and vision have been a constant source of inspiration.

Last but not least, I want to thank Dr. Pascal Grittmann for agreeing to be the second reviewer for this thesis.

Abstract

Multiple importance sampling is an indispensable tool in rendering that constructs robust sampling strategies by combining the respective strengths of individual distributions. Its efficiency can be improved greatly by carefully selecting the number of samples drawn from each distribution, but automating this process remains a challenging problem. Existing works are mostly limited to mixture sampling, in which only a single sample is drawn in total, and the works that do investigate multi-sample MIS only optimize the sample counts at a per-pixel level, which cannot account for variations beyond the first bounce. Recent work on Russian roulette and splitting demonstrated how fixed-point schemes can be used to spatially vary sample counts to optimize image efficiency, but is limited to choosing the same number of samples across all sampling strategies. Our work proposes a highly flexible sample allocation strategy that bridges the gap between these areas of work. We show how to iteratively optimize the sample counts to maximize the efficiency of the rendered image using a lightweight data structure, which allows us to make local and individual decisions per technique. We demonstrate the benefits of our approach in two applications, path guiding and bidirectional path tracing, in both of which we achieve consistent and substantial speedups over the respective previous state-of-the-art.

Keywords: Russian roulette, splitting, multi-sample, multiple importance sampling

Contents

Contents ix

List of Figures xi

1 Introduction 1

2 Background & Related Work 3
- 2.1 Rendering Equation . . . 3
- 2.2 Importance Sampling . . . 4
- 2.3 Multiple Importance Sampling . . . 4
- 2.4 Path Tracing . . . 4
  - 2.4.1 Bidirectional Methods . . . 5
  - 2.4.2 Path Guiding . . . 5
  - 2.4.3 Russian Roulette and Splitting . . . 5
- 2.5 MIS Sample Allocation . . . 6
  - 2.5.1 One-Sample / Mixture Sampling . . . 6
  - 2.5.2 Multi-Sample . . . 6

3 Theory 9
- 3.1 Model . . . 9
  - 3.1.1 Continuous Sample Counts . . . 10
  - 3.1.2 Unbiasedness . . . 10
- 3.2 Objective . . . 10
  - 3.2.1 Variance Model . . . 11
  - 3.2.2 Cost Model . . . 12
- 3.3 Optimization & Proxy . . . 12
  - 3.3.1 Variance Derivative . . . 12
  - 3.3.2 Cost Derivative . . . 13
  - 3.3.3 Fixed-Point . . . 15
  - 3.3.4 Proof of Optimality for Budget-Unaware Weights . . . 16
- 3.4 Application to Path Tracing . . . 16

4 Application I: Path Guiding **19**
4.1 Implementation . . . 19
4.1.1 Local Estimates . . . 19
4.1.2 Image Estimates . . . 20
4.1.3 Fixed-Point Iteration . . . 20
4.1.4 Outlier Clamping . . . 20
4.1.5 Handling Colors . . . 20
4.2 Evaluation . . . 20
4.2.1 Setup . . . 21
4.2.2 Mixture Optimization . . . 21
4.2.3 Results . . . 22
4.2.4 Convergence . . . 22
4.2.5 Rounding . . . 24

5 Application II: Bidirectional Path Tracing **27**
5.1 Implementation . . . 27
5.2 Evaluation . . . 28
5.2.1 Setup . . . 28
5.2.2 Results . . . 28
5.2.3 Convergence . . . 30
5.2.4 Rounding . . . 30

6 Limitations and Future Work **31**

7 Conclusion **33**

8 Co-Authors' Contributions **35**

Appendices

A Appendix A: Additional Guiding Results **43**

B Appendix B: Additional BDPT Results **45**

List of Figures

1.1 BDPT teaser figure of a modified Kitchenette 2

2.1 Overview comparing our approach to previous ones 7

3.1 Evaluation of our method on various 1D examples 14
3.2 Convergence of our method on a 1D examples 15

4.1 Comparison of previous mixture sampling approaches 21
4.2 Convergence plots of primary hits and efficiency in guiding 22
4.3 Evaluation matrix comparing our method with state-of-the-art 23
4.4 Impact of stochastic rounding 24

5.1 Evaluation matrix comparing our method with state-of-the-art 29
5.2 A closer look at our sample allocation (Bedroom) 30
5.3 Convergence plots of primary hits and efficiency in BDPT 30

6.1 Failure case example in BDPT 32

A.1 Evaluation matrix of the remaining tested scenes for guiding 43

B.1 Evaluation matrix of the remaining tested scenes for BDPT 45

1 Introduction

Synthetically generating realistic images from virtual, mathematical scene descriptions is an essential aspect in various fields such as multimedia, gaming, and advertisement. This is accomplished by reconstructing the image from a sample of admissible light paths, which are commonly obtained through unidirectional or bidirectional *path tracing*.

Path tracing is a powerful, yet computationally highly expensive, algorithm for generating photorealistic images utilizing Monte-Carlo integration. Since Kajiya formulated the rendering equation in 1986, improving its efficiency and accuracy has been extensively researched. Despite decades of research, no single sampling distribution can robustly capture all visual effects. Instead, contemporary works employ a blend of multiple strategies using a technique known as *multiple importance sampling* (MIS).

Many works have explored how to weight strategies in an MIS combination, but surprisingly little attention has been devoted to determining the optimal number of samples to assign to each technique in a (bidirectional) path tracing context. Although there have been attempts to address this issue, none have provided entirely satisfactory solutions. In practice, sample counts are often steered by manual user intervention while automatic approaches are limited to only using image space as guidance, leading to non-granular, suboptimal decisions.

We present a novel theory for MIS sample allocation. By considering the spatially local variances and costs of samples, our approach sensibly distributes samples among an arbitrary number of strategies, significantly increasing rendering efficiency. Furthermore, we validate our theory by applying it to two rendering algorithms, demonstrating its practical viability. In summary, our contributions are:

- We derive a novel theory for multi-sample MIS budget allocation, based on the theory of fixed-point iterations (chapter 3)
- We show how to apply our theory in the popular context of path guiding, achieving noticeable speedups over previous works (chapter 4)
- Additionally, we accelerate bidirectional path tracing with our approach and demonstrate equally substantial speedups (chapter 5) (as teasered in figure 1.1)

A shortened version was submitted to SIGGRAPH as a joint work with Alexander Rath and Ömercan Yazici. A detailed declaration of contributions can be found in chapter 8.

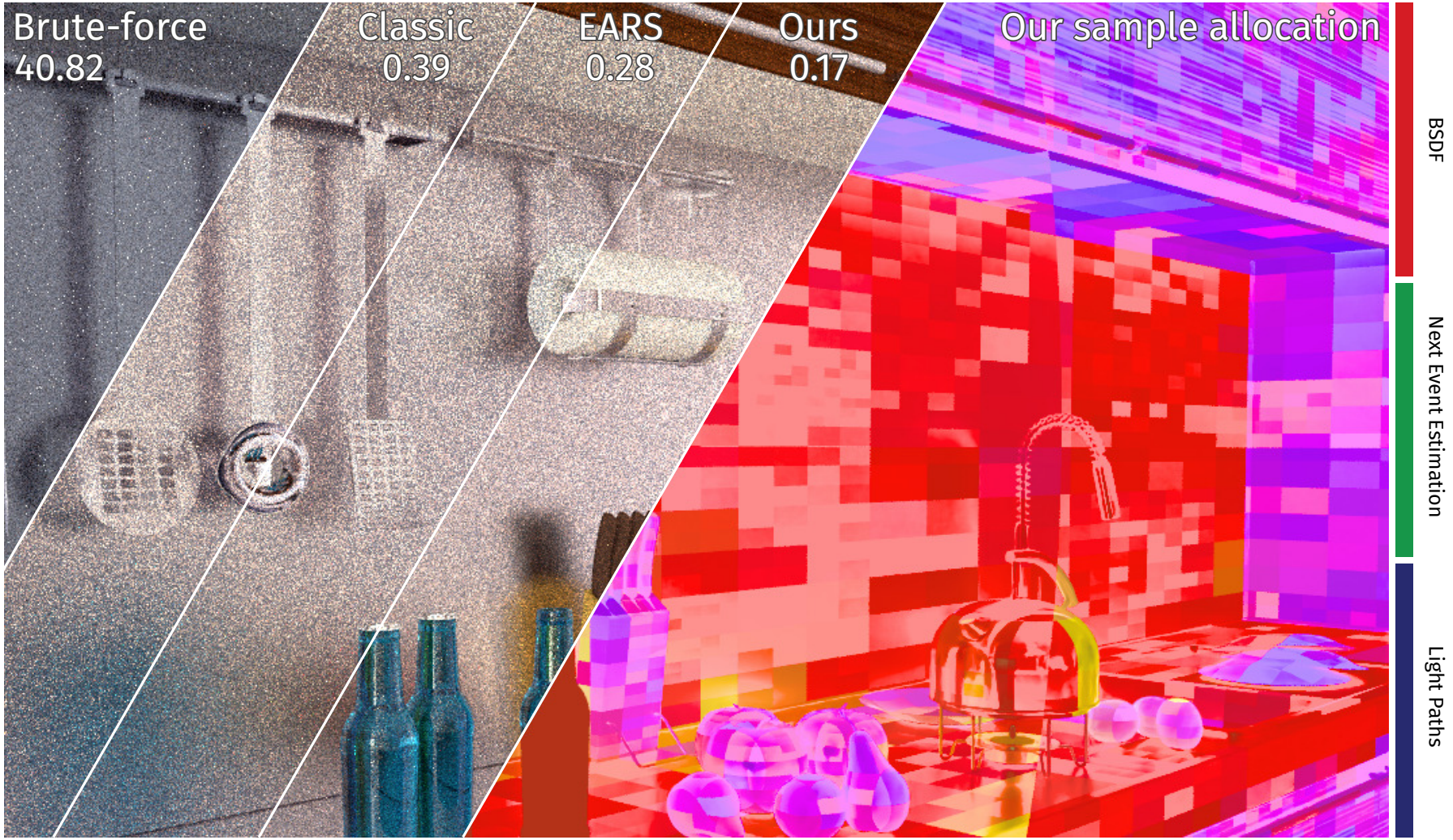


Figure 1.1: *The performance of multiple importance sampling can be greatly improved by choosing appropriate sample counts for each sampling strategy. Here, we compare the performance of our fixed-point approach applied to bidirectional path tracing (10 minute equal-time renders). Existing methods are either limited to per-pixel decisions ("brute-force" [Grittmann et al., 2022]) or to choosing the same amount of samples for each strategy ("EARS" [Rath et al., 2022]). Our method learns spatially varying sample counts individually for each strategy, visualized on the right, which increases performance by 65% over previous works.*

2

Background & Related Work

With technological advancements, the expensive technique of *path tracing* has become a viable rendering algorithm, even for some real-time applications. Previously, rasterization was the primary method, based on projecting primitives, such as triangles, onto image space. Although computationally inexpensive, rasterization lacks the ability to render certain effects physically accurately, e.g., reflections and caustics. To address these limitations, various tricks are implemented, often requiring additional rasterization steps or rendering passes. Yet these measures cannot fully overcome the inherent constraints. In contrast, nowadays, path tracing can be employed when physically accurate simulations are necessary or desired. Due to its nature of directly simulating light and its distribution throughout a scene, path tracing naturally generates physically accurate reflections and incorporates global illumination.

2.1 Rendering Equation

The idea of following light rays and collecting their interactions dates back hundreds of years, with artists using it to accurately capture perspective and distortions. Almost 40 years ago, Kajiya [1986] formulated the rendering equation

$$L_{\mathrm{o}}(x, \omega_o) = L_{\mathrm{e}}(x, \omega_o) + \int_{\Omega} f_{\mathrm{r}}(\omega_i, x, \omega_o) L_{\mathrm{i}}(x, \omega_i) \cos \omega_i \, \mathrm{d}\omega_i$$

shaping the direction of future research in light simulation. It describes the outgoing radiance L_{o} at position x in direction ω_o as the sum of emitted radiance L_{e} and an integral over all incident directions ω_i to collect the reflected radiance. L_{i} is the incident radiance that is weighted by the incident angle ω_i and the *bidirectional scattering distribution function* (BSDF) f_{r} that defines the material's properties. Due to the high dimensionality, many discontinuities, and overall complexity, obtaining a general analytical solution is impractical outside of trivial examples. Therefore, Monte Carlo integration is typically employed to compute this integral.

2.2 Importance Sampling

In Monte Carlo integration, uniformly sampling the integrand along all dimensions leads to a convergence rate of $\frac{1}{\sqrt{N}}$ toward the unbiased ground truth where N is the number of samples, i.e., quadrupling the number of samples halves the estimate's error. *Importance sampling* [Kloek et al., 1978] refers to the idea of drawing samples from a distribution different than the one that is approximated to reduce variance. The more similar the sampling distribution is to the integrand, the lower the variance. However, knowing the integral's value and its normalization factor to turn it into a probability density requires us to have solved the problem already.

In the context of estimating the rendering equation, the unknown and problematic term is the incident radiance L_i. By discarding this term, and only considering the angle and BSDF terms, it is possible to find an approximation of the real integrand that is good enough to decrease the estimator's variance. This is particularly helpful in the presence of strong peaks in small parts of the domain [Veach, 1997] which often is the case in complex scenes with intricate lighting effects. To remain unbiased while using importance sampling, each sample is weighted by its probability of being chosen.

2.3 Multiple Importance Sampling

Often, there is not a single distribution that estimates the target sufficiently well over the whole domain. *Multiple importance sampling* (MIS) [Veach et al., 1995b] can be utilized to combine several sampling distributions to leverage their respective strengths. To this end, a *weighting heuristic* approximates which strategy performs well within which subdomain to weigh them against each other, which has been explored in great detail by previous works. For instance, Veach et al. [1995b] introduced the power heuristics, which weights each technique by its relative probability of generating the acquired sample raised to a certain power. Due to their simplicity, yet good results, they are frequently used and allow for mathematical arguments about optimality. More sophisticated approaches describe how to achieve optimal MIS weights [Kondapaneni et al., 2019].

2.4 Path Tracing

Forward path tracing [Kajiya, 1986] is the primary choice for accurately simulating light transport due to its flexibility and extensibility. Starting from the camera, a random walk through the scene is performed to construct a path sample. At each intersection, the reflection properties (BSDF) are sampled to steer the random walk. Additional samples, such as direct connections to light sources (*next event estimation*, NEE), are commonly incorporated through MIS. However, even with sophisticated NEE enhancements, like *manifold next event estimation* [Hanika et al., 2015; Zeltner et al., 2020], forward path tracing alone does not suffice to render many visual effects efficiently and robustly.

While forward path tracing effectively handles many scenarios, it struggles with certain complex lighting effects, such as caustics and scenes with significant specular-diffuse-specular interactions. These limitations necessitate combining path tracing with additional techniques to achieve more robust rendering results.

2.4.1 Bidirectional Methods

In *bidirectional path tracing* (BDPT) [Veach et al., 1995a], an additional random walk from light sources is performed. Using MIS, it is possible to connect all pairs of intersections between the camera and light paths to obtain many complete path samples [Lafortune et al., 1993]. Instead of connecting the subpaths, they can also be merged at nearby intersections [Jensen, 1996; Shirley et al., 1995; Walter et al., 1997]. These methods can be combined and used simultaneously [Hachisuka et al., 2012; Georgiev et al., 2012]. It has been shown that connecting each camera path vertex to a fixed number of light subpath vertices can benefit the rendering performance [Popov et al., 2015; Nabata et al., 2020].

In this work, we only consider bidirectional path tracing without merging. Nonetheless, the proposed method could be extended to handle merging as well.

2.4.2 Path Guiding

Another popular approach to improve upon classical path tracing is *path guiding* [Jensen, 1995; Vorba et al., 2014; Müller et al., 2017]. By using previously acquired samples, important directions can be identified, allowing for the construction of adaptive sampling densities. To increase robustness, guiding is jointly used with standard forward path tracing and NEE in an MIS combination. Subsequent studies explored further improvements to enhance robustness for common effects [Ruppert et al., 2020; Rath et al., 2023]. Its effectiveness made path guiding an indispensable tool for achieving high-quality renderings in reasonable time frames.

2.4.3 Russian Roulette and Splitting

An integral component of path tracing, is Russian roulette and splitting (RRS) [Arvo et al., 1990]. RR entails prematurely terminating paths based on their current throughput, reducing computational effort for paths contributing little to the final image. Splitting, on the other hand, increases the number of subpaths at a given intersection, enhancing the exploration of important light paths to reduce variance. More advanced approaches consider additional metrics. For example, Vorba et al. [2016] propose a method that factors in expectations of future contributions. Furthermore, *efficiency-aware RRS* (EARS) [Rath et al., 2022] employs a fixed-point iteration to optimize RRS factors for efficiency, based on the theoretical optimal derivations of Bolin et al. [1997].

While these methods can optimize for global efficiency, they only determine the total number of samples taken at each intersection and have no control over how they are allocated to individual strategies. This limitation highlights the need for more granular and educated sample allocation methods to further improve rendering performance.

2.5 MIS Sample Allocation

MIS can be performed in two ways to address the issue of assigning an individual budget to each strategy. On the one hand, there is *one-sample MIS* in which a mixture ratio is applied that stochastically determines from which distribution each sample is drawn. This means that the random walk in path tracing is only ever continued with a single sample at each step. On the other hand, there is the *multi-sample* approach in which each technique is allocated a separate sampling budget.

2.5.1 One-Sample / Mixture Sampling

The ratios with which techniques are picked have a profound impact on the efficiency of MIS, but existing works on automating those share many limitations. Most works in this category focus on direct illumination only, optimizing the ratio between BSDF and NEE samples on a per-pixel basis, for example through second-order approximations [Lu et al., 2013] or iterative Newton-Raphson root finding [Sbert et al., 2019; Szirmay-Kalos et al., 2022]. Murray et al. [2020] introduce a simple heuristic to extend these works to global illumination and up to three strategies. A more powerful alternative to learning ratios per pixel is to learn them per region of space. Vorba et al. [2019] use gradient descent to compute sampling ratios between BSDF and path guiding, allowing for optimization of variance or KL divergence. Rath et al. [2020] improve this method by considering image variance instead of local variance alone.

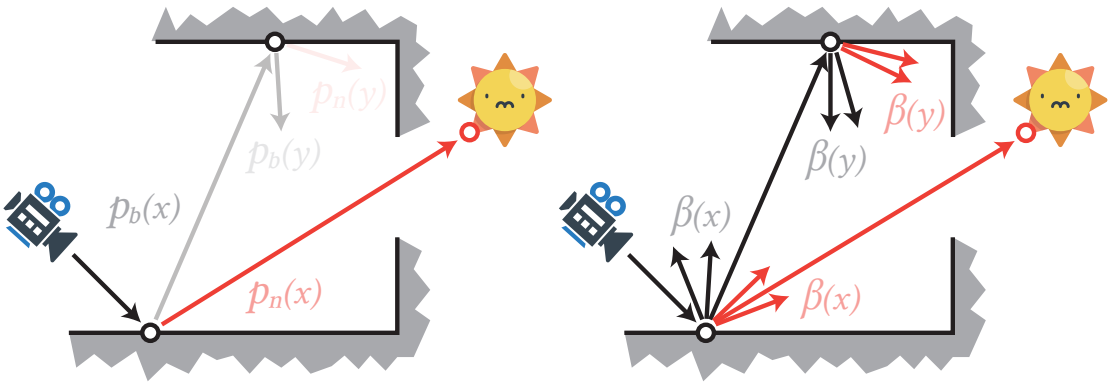
An important limitation shared by these approaches is that they only consider mixture sampling, meaning they only continue the random walk with a single sample. Additionally, they are limited to optimizing efficiency locally since the ratios are optimized independently, which is globally suboptimal.

2.5.2 Multi-Sample

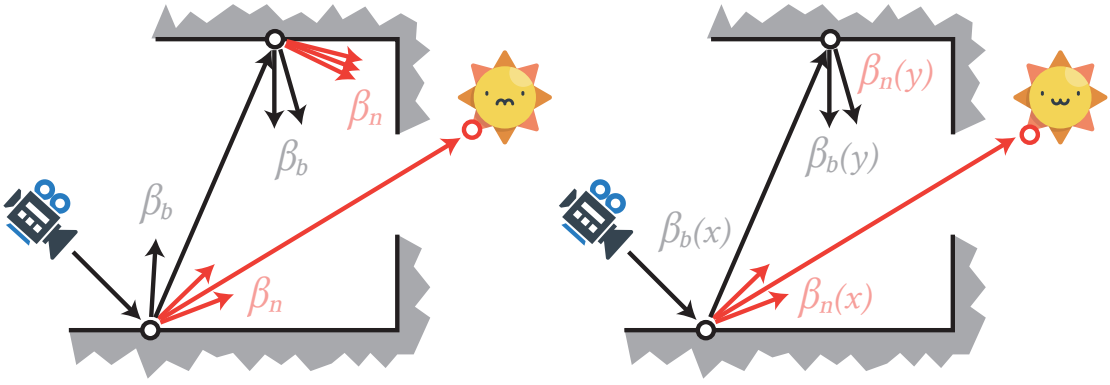
In multi-sample MIS, each distribution gets to draw an arbitrary amount of samples. This combines the strengths of mixture sampling (i.e., determining the ratios across techniques) and RRS (i.e., determining the total number of samples). However, we show that combining existing works from both domains, even if they are individually optimal, is insufficient to achieve optimal performance. Hence, they must be optimized jointly.

By modeling the expected computational cost of an MIS combination, Grittmann et al. [2022] globally optimize BDPT efficiency using a brute-force search through a set of candidate sample counts. Due to the curse of dimensionality, this technique is limited to optimizing global parameters (such as light path counts) or boolean per-pixel decisions.

Our approach (figure 2.1) similarly optimizes the efficiency of multi-sample MIS combinations but does so with much higher granularity by optimizing individual sampling strategies with spatially varying sample counts. This is enabled through a fixed-point scheme similar to EARS [Rath et al., 2022], which we generalize for MIS.



(a) *One-sample MIS* **(b)** *Russian roulette and splitting*



(c) *Per-pixel multi-sample MIS* **(d)** ***Our approach***

Figure 2.1: *Existing sample allocation works share three common limitations.* ***(a)*** *One-sample MIS randomly picks a single strategy to continue the path, which can result in poor efficiency since path prefixes are not reused.* ***(b)*** *RRS can reuse path prefixes but chooses the same sample count for each sampling strategy independent of whether the strategy contributes or not.* ***(c)*** *Existing multi-sample MIS approaches are limited to using the same locally-unaware sample accounts along the entire path.* ***(d)*** *Our method generalizes all of these methods and chooses spatially varying sample counts for each strategy individually.*

3

Theory

In this chapter, we first formulate the model underlying our theory, followed by the derivation of our efficiency-aware multi-sample MIS budget allocation strategy. We finalize by expressing the fixed-point iteration and briefly looking at its convergence.

3.1 Model

In multi-sample MIS, we are interested in finding an integral value $I = \int_\chi f(x)\,\mathrm{d}x$ by combining estimates from n_t sampling strategies with varying sample counts β_t

$$\langle I \rangle = \sum_{t=1}^{n_t} \frac{1}{\beta_t} \sum_{s=1}^{\beta_t} \langle I_t(x_{t,s}) \rangle = \sum_{t=1}^{n_t} \langle I_t(x_{t,\cdot}); \beta_t \rangle, \tag{3.1}$$

where $\langle I; \beta \rangle$ denotes a secondary estimator for I with β samples, and $\langle I_t \rangle$ denotes the primary estimator of technique t.

An important aspect is that estimators often capture different, possibly disjoint parts of the integrand. In path tracing, for example, we commonly have two subintegrands: direct light (estimated by NEE and BSDF sampling), and indirect light (only estimated by BSDF sampling). We address this by expressing the integrand as a sum of subintegrands $f = \sum_{i=1}^{n_i} f_i$ and perform MIS weighting thereof within the primary estimators $\langle I_t \rangle$

$$\langle I_t(x) \rangle = \sum_{i=1}^{n_i} \frac{f_i(x)}{p_t(x)} w_{it}(x),$$

where p_t is the probability density function of technique t and w_{it} is the MIS weight with respect to all other methods that estimate integrand f_i. Its dependency on i is due to the characteristic function $\mathbb{1}_{it}$ in the MIS weights, which is 1 whenever technique t estimates integrand f_i and 0 otherwise. This ensures correctness and $\sum_{t=1}^{n_t} w_{it}(x) = 1$ for all i. For example, for the budget-unaware balance heuristic we obtain

$$w_{it}(x) = \frac{\mathbb{1}_{it} p_t(x)}{\sum_{k=1}^{n_t} \mathbb{1}_{ik} p_k(x)}.$$

For brevity, we will drop the argument of I_t in the following whenever it is not the subject of interest. Note, that we can also drop the argument with explicit subscript s for the s-th sample since the samples are pairwise independent.

3.1.1 Continuous Sample Counts

Instead of restricting ourselves to integer sample counts $\beta_t \in \mathbb{N}$, we shift the domain to $\mathbb{R}^+$ to make the problem continuous. Since, in reality, we can only perform an integer number of samples, we employ a stochastic rounding function

$$r(\beta) = \begin{cases} \lfloor \beta \rfloor + 1 & \text{with probability } \beta - \lfloor \beta \rfloor, \\ \lfloor \beta \rfloor & \text{otherwise,} \end{cases}$$

where $\lfloor \cdot \rfloor$ is the floor function. Accounting for the stochastic rounding, the estimator still divides by the real-valued β. This is analogous to how previous works determine sample counts for RRS [Vorba et al., 2016; Rath et al., 2022], but generalizes the concept to individual techniques instead of using the same splitting factor across all of them. In section 4.2.5, we discuss the impact of this decision.

3.1.2 Unbiasedness

Having adapted the standard multi-sample MIS model and introduced stochastic rounding, we have to verify that our model still yields an unbiased estimate of I. This is a straightforward computation in which we use that the sum of the MIS weights over all techniques equals 1:

$$\begin{aligned} \mathbb{E}[\langle I \rangle] &= \sum_{t=1}^{n_t} \mathbb{E}[\langle I_t; \beta_t \rangle] = \frac{1}{\beta_t} \sum_{t=1}^{n_t} \mathbb{E}\left[\sum_{s=1}^{r(\beta_t)} \langle I_t \rangle \right] \\ &= \left(\frac{(\beta_t - \lfloor \beta_t \rfloor)(\lfloor \beta_t \rfloor + 1)}{\beta_t} + \frac{(1 - \beta_t + \lfloor \beta_t \rfloor)\lfloor \beta_t \rfloor}{\beta_t} \right) \sum_{t=1}^{n_t} \mathbb{E}[\langle I_t \rangle] \\ &= \sum_{t=1}^{n_t} \mathbb{E}[\langle I_t \rangle] = \sum_{t=1}^{n_t} \sum_{i=1}^{n_i} \mathbb{E}\left[\frac{f_i(x)}{p_t(x)} w_{it}(x) \right] = \int_{\mathcal{X}} \sum_{i=1}^{n_i} f_i(x) \underbrace{\left(\sum_{t=1}^{n_t} w_{it}(x) \right)}_{=1} \mathrm{d}x = I. \end{aligned}$$

3.2 Objective

Our goal is to assign an optimal number of samples to each strategy, i.e., to find optimal β_t. Previous works mostly focused on optimizing variance, which works well as long as the computational cost is largely unaffected by the choice of β_t, such as when optimizing one-sample MIS combinations. In multi-sample MIS, the cost highly depends on the number of samples. Consequently, we have to consider the *efficiency* $\mathcal{E}$ [Pharr et al., 2016]

$$\mathcal{E}[\langle I \rangle] = \frac{1}{\mathbb{V}[\langle I \rangle] \mathbb{C}[\langle I \rangle]}.$$

To optimize the efficiency, we therefore need to predict the variance and cost for any choice of β_t.

3.2.1 Variance Model

Assuming that the techniques are uncorrelated, we can express the variance of the MIS estimator neatly as the sum of the variances of all techniques in use. To derive this, we utilize that $\langle I \rangle$ is unbiased and start with the general formulation for variance given by

$$\mathbb{V}\left[\langle I \rangle\right] = \mathbb{E}\left[\langle I \rangle^2\right] - \mathbb{E}^2\left[\langle I \rangle\right] = \mathbb{E}\left[\langle I \rangle^2\right] - I^2.$$

Observe that the only unknown in this equation is the second moment of the main estimator. We can reformulate it as a sum dependent on moments of the secondary and primary estimators

$$\begin{aligned}\mathbb{E}\left[\langle I \rangle^2\right] = \mathbb{E}\left[\left(\sum_{t=1}^{n_t} \langle I_t; \beta_t \rangle\right)^2\right] &= \sum_{t=1}^{n_t} \mathbb{E}\left[\langle I_t; \beta_t \rangle^2\right] + \sum_{k \neq l}^{n_t} \mathbb{E}\left[\langle I_k; \beta_k \rangle \langle I_l; \beta_l \rangle\right] \\ &= \sum_{t=1}^{n_t} \mathbb{E}\left[\langle I_t; \beta_t \rangle^2\right] + \sum_{k \neq l}^{n_t} \mathbb{E}\left[\langle I_k \rangle\right] \mathbb{E}\left[\langle I_l \rangle\right],\end{aligned}$$

where for the last equality we assume independence of the techniques, which usually is a reasonable assumption, and the fact that $\mathbb{E}\left[\langle I_t; \beta_t \rangle\right] = \mathbb{E}\left[\langle I_t \rangle\right]$, as can easily be seen in the previous unbiasedness computation (see section 3.1.2).

Using the independence of the acquired samples within a technique, and $\beta_t \in \mathbb{N}_{>0}$, take note of the identity resulting from distributivity

$$\mathbb{E}\left[\left(\sum_{s=1}^{\beta_t} \langle I_t \rangle\right)^2\right] = \beta_t \mathbb{E}\left[\langle I_t \rangle^2\right] + \beta_t(\beta_t - 1)\mathbb{E}^2\left[\langle I_t \rangle\right].$$

With it, we can simplify the secondary estimator's second moment by eliminating its inherent dependency on β_t. For any fixed t, it becomes

$$\begin{aligned}\mathbb{E}\left[\langle I_t; \beta_t \rangle^2\right] &= \frac{1}{\beta_t^2} \mathbb{E}\left[\left(\sum_{s=1}^{r(\beta_t)} \langle I_t \rangle\right)^2\right] = \frac{\beta_t - \lfloor \beta_t \rfloor}{\beta_t^2} \mathbb{E}\left[\left(\sum_{s=1}^{\lfloor \beta_t \rfloor + 1} \langle I_t \rangle\right)^2\right] + \frac{1 - \beta_t + \lfloor \beta_t \rfloor}{\beta_t^2} \mathbb{E}\left[\left(\sum_{s=1}^{\lfloor \beta_t \rfloor} \langle I_t \rangle\right)^2\right] \\ &= \frac{1}{\beta_t} \mathbb{E}\left[\langle I_t \rangle^2\right] + \frac{2\beta_t \lfloor \beta_t \rfloor - \lfloor \beta_t \rfloor^2 - \lfloor \beta_t \rfloor}{\beta_t^2} \mathbb{E}^2\left[\langle I_t \rangle\right].\end{aligned}$$

One last trick is to express the first moment squared, I^2, as a sum of the pairwise products of the primary estimators as follows

$$I^2 = \left(\sum_{t=1}^{n_t} \mathbb{E}\left[\langle I_t \rangle\right]\right)^2 = \sum_{t=1}^{n_t} \mathbb{E}^2\left[\langle I_t \rangle\right] + \sum_{k \neq l}^{n_t} \mathbb{E}\left[\langle I_k \rangle\right] \mathbb{E}\left[\langle I_l \rangle\right].$$

Putting these formulations for the first and second moments together, the formula for the variance of our estimator simply becomes the sum of the variance of all techniques

$$\mathbb{V}\left[I\right] = \sum_{t=1}^{n_t} \mathbb{V}\left[\langle I_t; \beta_t \rangle\right] = \sum_{t=1}^{n_t} \left(\frac{1}{\beta_t} \mathbb{V}\left[\langle I_t \rangle\right] + \rho(\beta_t) \mathbb{E}^2\left[\langle I_t \rangle\right]\right),$$

where $\rho(\beta) = \frac{(\beta - \lfloor \beta \rfloor)(\lceil \beta \rceil - \beta)}{\beta^2} = \frac{2\beta\lfloor \beta \rfloor - \lfloor \beta \rfloor^2 - \lfloor \beta \rfloor - \beta^2 + \beta}{\beta^2}$ arises due to stochastic rounding.

3.2.2 Cost Model

The cost of our multi-sample MIS estimator is given by the sum of all techniques' real-valued sample counts multiplied by the cost of taking a sample for this technique, i.e., $\mathbb{C}[\langle I \rangle] = \sum_{t=1}^{n_t} \beta_t \mathbb{C}_t + \mathbb{C}_\Delta$, where $\mathbb{C}_t$ denotes the cost of taking a sample for technique t and $\mathbb{C}_\Delta$ is a constant to incorporate overhead costs.

3.3 Optimization & Proxy

We want to find the optimal β_t that maximize efficiency. To reduce clutter, our derivation pursues the equivalent goal of minimizing inverse efficiency, i.e., $\mathcal{E}^{-1}[\langle I \rangle]$. A necessary condition for a minimum with respect to β_t is that the partial derivatives of inverse efficiency equal 0. Using the product formula, we obtain the expression

$$\frac{\mathrm{d}\mathcal{E}^{-1}[\langle I \rangle]}{\mathrm{d}\beta_t} = \frac{\mathrm{d}\mathbb{V}[\langle I \rangle]}{\mathrm{d}\beta_t}\mathbb{C}[\langle I \rangle] + \mathbb{V}[\langle I \rangle]\frac{\mathrm{d}\mathbb{C}[\langle I \rangle]}{\mathrm{d}\beta_t},$$

revealing that we need the derivatives of the estimator's cost and variance.

3.3.1 Variance Derivative

The derivative of variance is given by

$$\begin{aligned}\frac{\mathrm{d}\mathbb{V}[\langle I \rangle]}{\mathrm{d}\beta_t} &= \sum_{k=1}^{n_t}\left(\frac{\mathrm{d}}{\mathrm{d}\beta_t}\frac{1}{\beta_k}\mathbb{V}[\langle I_k \rangle] + \frac{\mathrm{d}}{\mathrm{d}\beta_t}\rho(\beta_k)\mathbb{E}^2[\langle I_k \rangle]\right) \\ &= -\frac{1}{\beta_t^2}\mathbb{V}[\langle I_t \rangle] + \frac{\mathrm{d}\rho(\beta_t)}{\mathrm{d}\beta_t}\mathbb{E}^2[\langle I_t \rangle] + \sum_{k=1}^{n_t}\underbrace{\left(\frac{1}{\beta_k}\frac{\mathrm{d}\mathbb{V}[\langle I_k \rangle]}{\mathrm{d}\beta_t} + \rho(\beta_k)\frac{\mathrm{d}\mathbb{E}^2[\langle I_k \rangle]}{\mathrm{d}\beta_t}\right)}_{=0,\text{ for budget-unaware MIS weights}}.\end{aligned}$$

Note that, at this point, the sum over all n_t techniques is 0 if the MIS weights are budget-unaware, i.e., do not contain any β.

Moreover, the complex shape of $\rho(\beta_t)$ makes an analysis of its derivative challenging. Therefore, we resort to the same simplification employed by previous works in similar situations [Rath et al., 2022], i.e., we assume $\rho(\beta) = \max(1, \frac{1}{\beta}) - 1$. It is easy to verify that this simplification only affects the $\beta > 1$ case, where it drops the noise introduced by stochastic rounding to arrive at a convex expression for variance

$$\mathbb{V}[\langle I_t; \beta_t \rangle] = \begin{cases}\frac{1}{\beta_t}\mathbb{E}[\langle I_t \rangle^2] - \mathbb{E}^2[\langle I_t \rangle] & \text{if } \beta_t < 1, \\ \frac{1}{\beta_t}\mathbb{V}[\langle I_t \rangle] & \text{otherwise.}\end{cases} \tag{3.2}$$

For the derivative, this yields

$$\frac{\mathrm{d}\mathbb{V}[\langle I \rangle]}{\mathrm{d}\beta_t} = \begin{cases}-\frac{1}{\beta_t^2}\mathbb{E}[\langle I_t \rangle] + \sum_{k=1}^{n_t}\frac{1}{\beta_k}\frac{\mathrm{d}\mathbb{E}[\langle I_k \rangle^2]}{\mathrm{d}\beta_t} & \text{if } \beta_t \leq 1, \\ -\frac{1}{\beta_t^2}\mathbb{V}[\langle I_t \rangle] + \underbrace{\sum_{k=1}^{n_t}\frac{1}{\beta_k}\frac{\mathrm{d}\mathbb{V}[\langle I_k \rangle]}{\mathrm{d}\beta_t}}_{=0,\text{ if budget-unaware}} & \text{otherwise.}\end{cases}$$

The proxy we just introduced is optimal for budget-unaware MIS weights (i.e., weights independent of β_t), but for budget-aware weights, its optimum can deviate from the true optimal solution. We perform two evaluations to verify the effectiveness of this proxy: **First**, we compare its derivatives and optima to the true efficiency for simple one-dimensional functions in figure 3.1. We observe that in many cases our proxy's optimum is reasonably close to the real one, and, in particular, closer than the ones previous work finds. As can be seen in the last two examples, however, there are also cases where our proxy does not perform best which seems to be especially in those cases where one technique is (almost) perfect. **Second**, in our two rendering applications, we demonstrate that budget-aware weights with this proxy robustly outperform theoretically optimal allocations of budget-unaware weights.

Budget-Unaware Expression Putting the derivatives' equations together, we obtain

$$\frac{\mathrm{d}\mathcal{E}^{-1}[\langle I\rangle]}{\mathrm{d}\beta_t} = \begin{cases} -\frac{1}{\beta_t^2}\mathbb{E}[\langle I_t\rangle^2]\,\mathbb{C}[\langle I\rangle] + \mathbb{V}[\langle I\rangle]\,\mathbb{C}_t & \text{if } \beta_t \le 1, \\ -\frac{1}{\beta_t^2}\mathbb{V}[\langle I_t\rangle]\;\mathbb{C}[\langle I\rangle] + \mathbb{V}[\langle I\rangle]\,\mathbb{C}_t & \text{otherwise.} \end{cases} \tag{3.3}$$

Budget-Aware Expression In the budget-aware scenario, the exact expression strongly depends on the MIS weight heuristic but is not generally convex anymore even for simple heuristics like the balance heuristic (see figure 3.1; the last row visualizes this particularly well). Hence, we will focus on the budget-unaware case and treat it as a simplification for the budget-aware case. With this, we lose optimality, but we show empirically that it performs well both in simple 1D examples, and complex rendering applications (see evaluations in chapters 4 and 5).

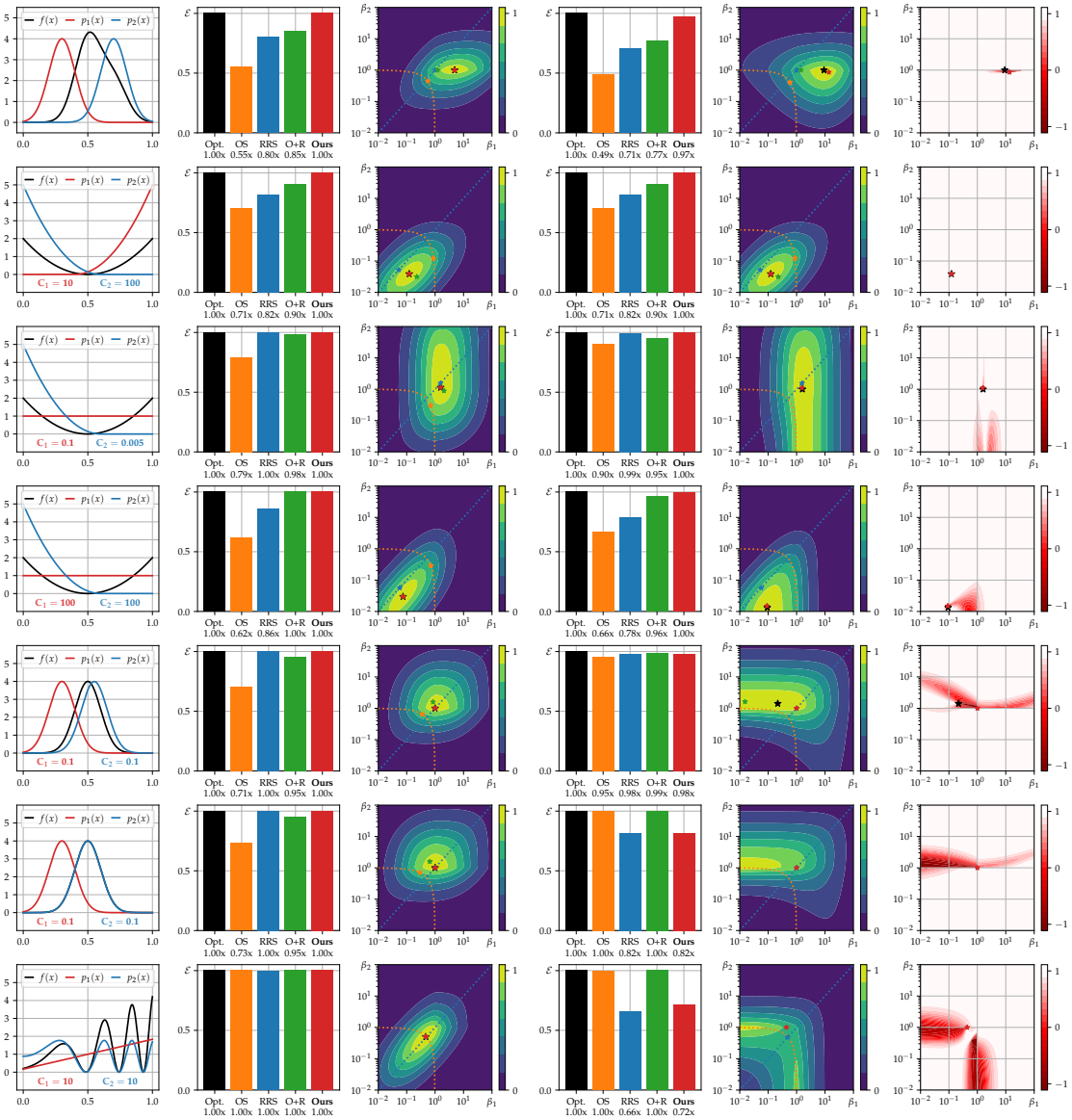
The variance's derivative includes some extra terms that are generated by the derivative of the MIS weights. For any fixed k, these extra terms take the form

$$\begin{aligned} &\frac{1}{\beta_k}\frac{\mathrm{d}\mathbb{V}[\langle I_k\rangle]}{\mathrm{d}\beta_t} + \rho(\beta_k)\frac{\mathrm{d}\mathbb{E}^2[\langle I_k\rangle]}{\mathrm{d}\beta_t} \\ &= \frac{2}{\beta_k}\mathbb{E}\left[\langle I_k\rangle\frac{\mathrm{d}\langle I_k\rangle}{\mathrm{d}\beta_t}\right] + \frac{2}{\beta_k}\mathbb{E}[\langle I_k\rangle]\,\mathbb{E}^2\left[\frac{\mathrm{d}\langle I_k\rangle}{\mathrm{d}\beta_t}\right] + 2\rho(\beta_k)\mathbb{E}[\langle I_k\rangle]\,\mathbb{E}^2\left[\frac{\mathrm{d}\langle I_k\rangle}{\mathrm{d}\beta_t}\right] \\ &= \frac{2}{\beta_k}\mathbb{E}\left[\langle I_k\rangle\sum_{i=1}^{n_i}\frac{f_i(x)}{p_k(x)}\frac{\mathrm{d}w_{ik}(x)}{\mathrm{d}\beta_t}\right] + \frac{2}{\beta_k}\mathbb{E}[\langle I_k\rangle]\,\mathbb{E}^2\left[\sum_{i=1}^{n_i}\frac{f_i(x)}{p_k(x)}\frac{\mathrm{d}w_{ik}(x)}{\mathrm{d}\beta_t}\right] \\ &\quad + 2\rho(\beta_k)\mathbb{E}[\langle I_k\rangle]\,\mathbb{E}^2\left[\sum_{i=1}^{n_i}\frac{f_i(x)}{p_k(x)}\frac{\mathrm{d}w_{ik}(x)}{\mathrm{d}\beta_t}\right]. \end{aligned}$$

3.3.2 Cost Derivative

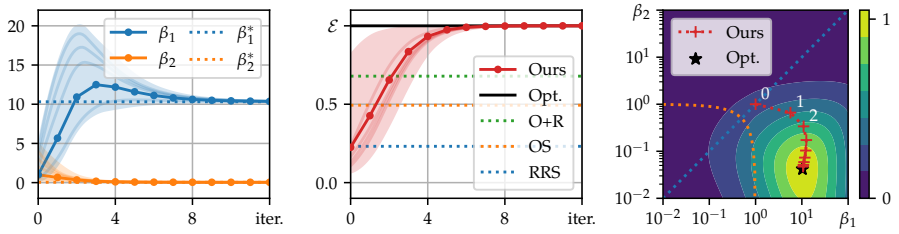
Only one summand of cost depends on β_t. Therefore, the derivative is simply given by

$$\frac{\mathrm{d}\mathbb{C}[\langle I\rangle]}{\mathrm{d}\beta_t} = \mathbb{C}_t.$$



(a) *Integrand* **(b)** *Budget-unaware balance heuristic* **(c)** *Budget-aware balance heuristic* **(d)** *Proxy error*

Figure 3.1: ***(a)*** *We evaluate the performance of our model on a simple 1D example, in which a single integrand f is estimated by two techniques p_1 and p_2, the latter of which matches slightly better but is 30× as expensive as p_1.* ***(b)*** *We compare the efficiency of four approaches: Optimal mixture sampling ("OS"), optimal Russian roulette and splitting ("RRS"), combining the two ("O+R"), and our proxy model ("Ours"). Note that combining optimal mixture sampling and RRS does not yield optimal performance. Our model optimizes ratios and total sample counts jointly, resulting in optimal performance for budget-unaware MIS weights.* ***(c)*** *For budget-aware MIS weights, the optimum predicted by our model can deviate from the true efficiency optimum.* ***(d)*** *We investigate the error of our proxy by plotting the dot product of the true gradients and our proxy gradients ("1" indicates a perfect match, "-1" indicates opposing directions).*



(a) *Sample budgets* **(b)** *Efficiency* **(c)** *Landscape*

Figure 3.2: ***(a)*** *We visualize the quick convergence of the sample allocations β_1 and β_2 towards their optimal values for a simple 1D example. The shaded area corresponds to trajectories of various initializations.* ***(b)*** *After 5 iterations we are close to the optimal efficiency. While performing more steps brings us closer, it yields diminishing returns.* ***(c)*** *We show the path our fixed-point iteration takes along the efficiency landscape.*

3.3.3 Fixed-Point

To devise a fixed-point scheme, we need to find the inverse efficiency's derivative's roots, which we achieve by rearranging equation (3.3). For brevity, let

$$M_t = \begin{cases} \mathbb{E}\left[\langle I_t \rangle\right] & \text{if } \beta_t \leq 1, \\ \mathbb{V}\left[\langle I_t \rangle\right] & \text{otherwise.} \end{cases}$$

Then, since $\beta_t > 0$,

$$\frac{\mathrm{d}\mathcal{E}^{-1}\left[\langle I \rangle\right]}{\mathrm{d}\beta_t} = -\frac{1}{\beta_t^2} M_t \mathbb{C}\left[\langle I \rangle\right] + \mathbb{V}\left[\langle I \rangle\right] \mathbb{C}_t \stackrel{!}{=} 0 \iff \beta_t = \sqrt{\frac{M_t}{\mathbb{V}\left[\langle I \rangle\right]} \frac{\mathbb{C}\left[\langle I \rangle\right]}{\mathbb{C}_t}}.$$

With this equation, there is to note that β_t appears on both sides — and non-linearly so on the right-hand side as a part of $\mathbb{V}\left[\langle I \rangle\right]$ and $\mathbb{C}\left[\langle I \rangle\right]$. Hence, an analytical solution becomes impractical and we have to resort to numerical methods. The equation is of a similar form to what previous work [Rath et al., 2022] has solved with a fixed-point iteration scheme before. Inspired by them, we can employ a fixed-point iteration using the equation above as an update function and compute the new sample allocations based on the variances and costs of using the previous iterations' allocations, as given by

$$\beta_{t,i+1} = \begin{cases} \beta_{t,i}^{\mathrm{RR}} = \sqrt{\frac{\mathbb{E}\left[\langle I_t ; \beta_{t,i} \rangle^2\right]}{\mathbb{V}[\langle I ; \boldsymbol{\beta}_i \rangle]} \frac{\mathbb{C}[\langle I ; \boldsymbol{\beta}_i \rangle]}{\mathbb{C}_t}} & \text{if } \beta_{t,i}^{\mathrm{RR}} < 1, \\ \beta_{t,i}^{\mathrm{S}} = \sqrt{\frac{\mathbb{V}[\langle I_t ; \beta_{t,i} \rangle]}{\mathbb{V}[\langle I ; \boldsymbol{\beta}_i \rangle]} \frac{\mathbb{C}[\langle I ; \boldsymbol{\beta}_i \rangle]}{\mathbb{C}_t}} & \text{if } \beta_{t,i}^{\mathrm{S}} > 1, \\ 1 & \text{otherwise,} \end{cases}$$

where $\boldsymbol{\beta}_i$ is the vector of all $\beta_{t,i}$ of the i-th iteration.

The third case arises because, with our proxy, the problem is split into two cases, one where $\beta_t \leq 1$ and one where $\beta_t > 1$. In some scenarios, an update step using the first one would lead to $\beta_{t,i+1} \geq 1$, whereas an update step in the other would lead to $\beta_{t,i+1} \leq 1$. We resort to following neither decision by allocating a single sample. In practice, this process converges quickly after only a few iterations as depicted in figure 3.2.

3.3.4 Proof of Optimality for Budget-Unaware Weights

We know that the efficiency is of the form

$$\mathcal{E}^{-1}\left[\langle I;\boldsymbol{\beta}\rangle\right] = \underbrace{\left(\sum_t^{n_t} \frac{1}{\beta_t}\mathbb{V}[\langle I_t\rangle] + \mathbb{V}_\Delta\right)}_{\mathbb{V}_\Sigma}\underbrace{\left(\sum_t^{n_t}\beta_t\mathbb{C}_t + \mathbb{C}_\Delta\right)}_{\mathbb{C}_\Sigma},$$

and hence optimal β_t^* must satisfy

$$\beta_t^* = \sqrt{\frac{\mathbb{V}[\langle I_t\rangle]}{\mathbb{C}_t}\frac{\mathbb{C}_\Sigma}{\mathbb{V}_\Sigma}} = \lambda\sqrt{\frac{\mathbb{V}[\langle I_t\rangle]}{\mathbb{C}_t}}.$$

For budget-unaware weights, the variance $\mathbb{V}[\langle I_t\rangle]$ is independent of β_t, and hence all that remains to be found is λ. Rewriting the inverse efficiency in terms of λ yields

$$\mathcal{E}^{-1}\left[\langle I;\lambda\rangle\right] = \left(\frac{1}{\lambda}M + \mathbb{V}_\Delta\right)\left(\lambda M + \mathbb{C}_\Delta\right),$$

where $M = n_t\sqrt{\mathbb{V}[\langle I_t\rangle]\mathbb{C}_t}$. Since our update formula enforces that we move on the line defined by λ, our fixed-point scheme degenerates to the EARS fixed-point scheme after a single step, which has been proven to be optimal [Rath et al., 2022].

3.4 Application to Path Tracing

As a last step, we discuss how to transfer and apply our theory to path tracing, for which we extend our theory to multiple spatially varying estimators.

Primary Estimator In path tracing, we combine multiple strategies for estimating reflected radiance L_r at each intersection

$$\langle L_{\mathrm{r},t}(x,\omega_\mathrm{o})\rangle = \frac{\langle L_\mathrm{i}(x,\omega_\mathrm{i})\rangle\, B(x,\omega_\mathrm{i},\omega_o)\,|\cos\theta_\mathrm{i}|}{p_t(\omega_\mathrm{i}\mid x_j,\omega_\mathrm{o})}\, w_t(x_j,\omega_\mathrm{i},\omega_\mathrm{o}),$$

where technique t samples $\omega_\mathrm{i}\sim p_t$ and B denotes the BSDF.

Secondary Estimator Analogous to equation (3.1), we define the multi-sample estimator

$$\langle L_\mathrm{r}(x,\omega_\mathrm{o})\rangle = \sum_{t=1}^{n_t}\langle L_{\mathrm{r},t}(x,\omega_\mathrm{o});\beta_t(\bar{x})\rangle,$$

where the sample budgets β_t are optimized for each individual technique t and path prefix $\bar{x}$. Instead of budgets that are constant across the image, we thereby adapt our sample budgets locally.

Random Walks Our random walk for a pixel px begins with a ray sampled by the camera and then recursively applies the multi-sample estimator of L_r at each intersection

$$\langle I_\mathrm{px}\rangle = \frac{W_\mathrm{px}(x_0,\omega_{\mathrm{i},0})}{p(x_0,\omega_{\mathrm{i},0})}\langle L_\mathrm{r}(x_1,\omega_{\mathrm{o},1})\rangle.$$

Multiple Estimators Optimizing each pixel individually is insufficient to achieve a global optimum, hence we maximize the efficiency built from the average variance $\mathbb{V}_\mathrm{I}$ and average cost $\mathbb{C}_\mathrm{I}$ over the entire image instead [Rath et al., 2022]. To not assign too much weight to bright pixels, we similarly use the *relative* variance of each pixel, i.e., we divide by the pixel's ground truth I_px

$$\mathcal{E}^{-1} = \underbrace{\left(\frac{1}{N_\mathrm{px}}\sum_\mathrm{px}^{N_\mathrm{px}} \frac{\mathbb{V}\left[\langle I_\mathrm{px}\rangle\right]}{I_\mathrm{px}^2}\right)}_{\mathbb{V}_\mathrm{I}}\underbrace{\left(\frac{1}{N_\mathrm{px}}\sum_\mathrm{px}^{N_\mathrm{px}} \mathbb{C}\left[\langle I_\mathrm{px}\rangle\right]\right)}_{\mathbb{C}_\mathrm{I}}.$$

Optimization Extended to multiple spatially varying estimators, our update function becomes

$$\beta_t(\bar{x}) = \begin{cases} \beta_t^\mathrm{RR} = \frac{T(\bar{x})}{I_\mathrm{px}}\sqrt{\frac{\mathbb{C}_\mathrm{I}}{\mathbb{C}_t}\frac{\mathbb{E}\left[\langle L_{\mathrm{r},t}\rangle^2\right]}{\mathbb{V}_\mathrm{I}}} & \text{if } \beta_t^\mathrm{RR} < 1, \\ \beta_t^\mathrm{S} = \frac{T(\bar{x})}{I_\mathrm{px}}\sqrt{\frac{\mathbb{C}_\mathrm{I}}{\mathbb{C}_t}\frac{\mathbb{V}[\langle L_{\mathrm{r},t}\rangle]}{\mathbb{V}_\mathrm{I}}} & \text{if } \beta_t^\mathrm{S} > 1, \\ 1 & \text{else.} \end{cases}$$

Here, $T(\bar{x}_k)$ denotes the throughput weight of the path prefix

$$T(\bar{x}_k) = \frac{W_\mathrm{px}(x_0, \omega_{\mathrm{i},0})}{p(x_0, \omega_{\mathrm{i},0})}\prod_{j=1}^{k-1}\frac{B(x_j, \omega_{\mathrm{i},j}, \omega_{\mathrm{o},j})\,|\cos(\theta_{\mathrm{i},j})|}{p_{t_j}(\omega_{\mathrm{i},j} \mid x_j, \omega_{\mathrm{o},j})}\frac{w_{t_j}(\bar{x}_j)}{\beta_{t_j}(\bar{x}_j)},$$

where t_j denotes the technique that was used at the j-th bounce, and w_ind is the MIS weight over all techniques that participate in estimating indirect light (i.e., lead to a secondary estimator being performed at x_j).

4

Application I
Path Guiding

In this chapter, we present the first application of our method. We begin by demonstrating how our technique can be used in unidirectional rendering to combine three common strategies: BSDF sampling, next event estimation (NEE) and guided sampling.

4.1 Implementation

We test our method in a guided path tracer on top of the implementation of Müller et al. [2017] in Mitsuba [Jakob, 2010]. Each completed training iteration for guiding marks one step of our fixed-point scheme. To compute the necessary variance and cost estimates, we build Monte Carlo estimates from the samples obtained throughout the iteration.

4.1.1 Local Estimates

We share the spatio-directional tree used by guiding to store the local estimates needed by our method, such as costs, variances, and second moments for each region and each technique. For every cache c and technique t, the first and second moments are estimated from the iteration's training samples that landed in the cache and were produced by the technique itself

$$\mathbb{E}\left[\langle L_{\mathrm{r},t}\rangle\right] \approx \frac{1}{n_{t,c}} \sum_{s=1}^{n_{t,c}} \langle L_{\mathrm{r},t_s}(x_s)\rangle \quad \text{and} \quad \mathbb{E}\left[\langle L_{\mathrm{r},t}\rangle^2\right] \approx \frac{1}{n_{t,c}} \sum_{s=1}^{n_{t,c}} \langle L_{\mathrm{r},t_s}(x_s)\rangle^2,$$

from which we can compute the estimate for the variance.

To estimate the cost of each technique, we employ the well-established heuristic of using the average path length generated by the technique as a proxy for cost [Veach, 1997; Rath et al., 2022].

4.1.2 Image Estimates

In lieu of the real ground truth, we apply a denoiser [Áfra, 2019] to determine the pixel estimates I_{px}. The pixel estimates are also used to estimate the image variance

$$\mathbb{V}[\langle I \rangle] \approx \frac{1}{N_{\text{px}}} \sum_{\text{px}} \frac{1}{N_{\text{spp}}} \sum_{s=1}^{N_{\text{spp}}} \left(\frac{\langle I_{\text{px}} \rangle_s - \hat{I}_{\text{px}}}{\hat{I}_{\text{px}}} \right)^2 .$$

The image's cost is similarly estimated by counting the total number of rays traced.

4.1.3 Fixed-Point Iteration

Implementing our fixed-point iteration algorithm requires an iterative rendering process, in which each iteration performs some number of samples to estimate the variances and costs, thus allowing us to perform steps toward the fixed-point. Path guiding already uses an iterative scheme to learn its guided sampling distribution in a training phase, into which we integrate the fixed-point iteration. We use ordinary throughput-based Russian roulette without splitting for the first three training iterations to give the data structure time to adapt, reducing estimate noise, before we switch to our sample allocation.

4.1.4 Outlier Clamping

Outliers can drastically reduce the quality of the variance estimates. To combat this, we clamp the contribution of paths to a fixed multiple of the pixel estimate (50). Note that this is only used for our variance estimates; the render itself is unaffected. Similar to previous works, we clamp the maximal number of samples per technique (to $[0.05, 20]$).

4.1.5 Handling Colors

To extend our theory to color, we perform computations channel-wise and use the L_2 norm to compute the final sample counts analogous to previous works [Rath et al., 2022].

4.2 Evaluation

We compare our approach against the state-of-the-art in path guiding, based on mixture optimization using gradient descent [Vorba et al., 2019] in conjunction with EARS [Rath et al., 2022] to steer sample counts ("EARS"). We use the variant that minimizes KL divergence of image contributions [Rath et al., 2020], which performs best across all our scenes (see section 4.2.2). As a baseline, we include vanilla path guiding with throughput-based RR ("classic RR") starting at the 5th bounce, while all other methods start performing RR with the first bounce. We provide results for our strategy in both budget-unaware and budget-aware MIS weight scenarios. An additional ablation ("Ours + Grad. Descent") combines gradient descent (mixing BSDF and guiding) and our approach (sample counts of NEE and the BSDF/guiding mixture) to demonstrate that our approach chooses better mixtures than the previous state-of-the-art optimizer.

4.2.1 Setup

All images are rendered for 5 minutes (2.5min training, 2.5min rendering) on an AMD Ryzen Threadripper 2950X yielding an equal-time comparison since an equal-sample comparison is difficult when splitting is involved. The maximum path length is set to 40 and we perform 9 training iterations whose time increases exponentially [Müller et al., 2017]. The noise is quantified using relative mean squared error (relMSE), with 0.01% of the brightest outliers discarded to arrive at a robust error estimate.

4.2.2 Mixture Optimization

In path guiding, we optimize the BSDF/guiding ratio of previous work using the KL divergence extended to image contributions [Rath et al., 2020]. This cannot directly maximize efficiency because **(1)** it does not contain information about cost, **(2)** only optimizes decisions locally rather than balancing variance across pixels and regions, and **(3)** optimizes KL divergence instead of variance. While **(1)** and **(2)** are fundamental limitations, **(3)** is, in theory, addressed by the variance optimizing variant [Vorba et al., 2019], but performed poorly across our tests scenes – likely due to the Adam optimizer drastically reducing its step-size in the presence of strong outliers. Hence, as suggested by the original authors, we use the KL divergence instead. A brief selection of scenes rendered for 5 minutes showcasing minimizing for **(a)** KL divergence, **(b)** KL divergence with image contributions, **(c)** variance, and **(d)** variance with image contributions can be found in figure 4.1.

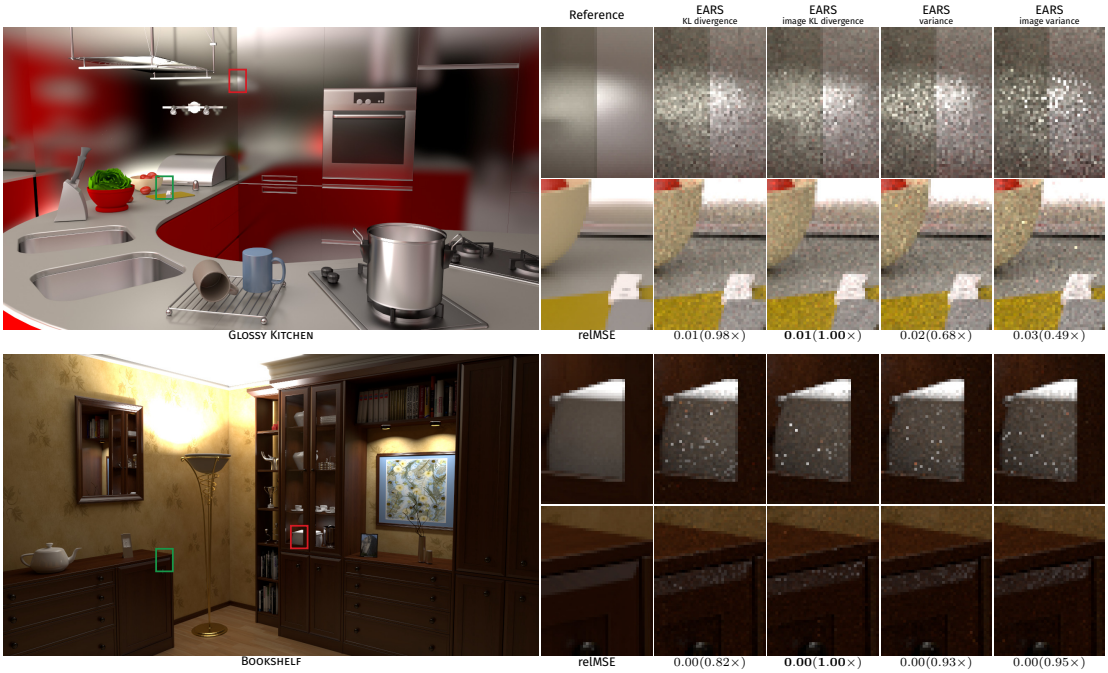


Figure 4.1: *We compare variants of gradient descent for the BSDF and guiding sample mixture on top of EARS in guiding:* ***(a)*** *KL divergence,* ***(b)*** *KL divergence of image contributions,* ***(c)*** *variance, and* ***(d)*** *variance of image contributions. Across our set of test scenes, KL divergence with image contributions performs best.*

4.2.3 Results

We tested our method on a set of 10 common benchmark scenes, in which it achieves an average speedup of 1.30× over the previous state-of-the-art. We find that the budget-aware variant almost always yields superior performance. A selection of interesting scenes can be found in figure 4.3.

In Living Room, our method automatically detects that BSDF sampling performs best to estimate the incident light on the floor. On diffuse surfaces, guiding is used instead. NEE is left mostly disabled, saving computational cost and yielding a 33% speedup.

The Glossy Kitchen contains various glossy surfaces, which make guiding and NEE poor choices. By disabling NEE almost entirely and using guiding only on the counter in the lower part where it is beneficial, we save precious time that can be spent elsewhere.

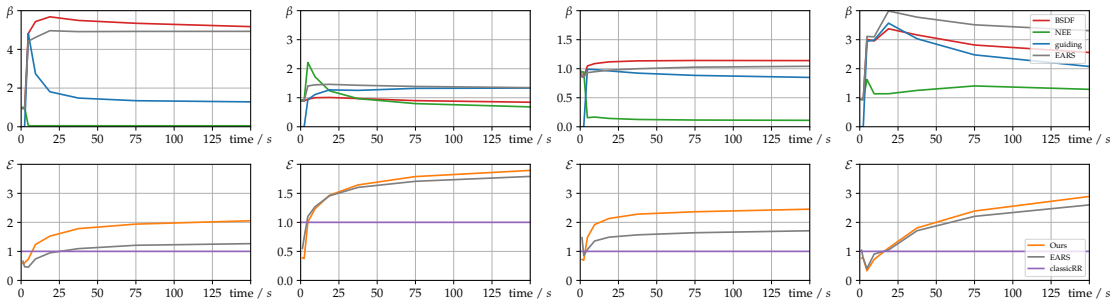
Corona Benchmark features strong indirect illumination from a directly illuminated spot on the sofa. Our method correctly identifies that any path arriving at the sofa should attempt a high number of NEE samples. Note that this would not be possible with per-pixel-based approaches, which are limited to using the same sampling ratios independent of where the path lands at later bounces.

The Bedroom is a scene where all three available techniques are useful in different areas. The left wall benefits greatly from performing NEE and every path encountering it at any bounce can utilize this to reduce variance.

In the Pool, an outdoor scene with many caustics, our method achieves a 37% speedup. It correctly identifies the necessity of guiding to estimate the caustic cast by the windows on the right of the pool.

4.2.4 Convergence

We verified that our method converges in practice by ensuring repeated rendering yields the same results and that the curves of average budget allocation per technique as well as overall efficiency flatten over time. A selection of convergence graphs is shown in figure 4.2. We tested the impact of the number of training iterations (7,9,11,13) but found the algorithm is mostly insensitive to this parameter, yielding no noticeable difference.



(a) *Glossy Kitchen* **(b)** *Dining Room* **(c)** *Glossy Bathroom* **(d)** *Bookshelf*

Figure 4.2: *We show convergence of the average primary hit point sample allocation (first row) and overall efficiency (second row, higher is better) in four selected scenes for path guiding. The setup is the same as in the main text. Our fixed-point scheme quickly converges towards solutions that are better than the baseline and previous state-of-the-art.*

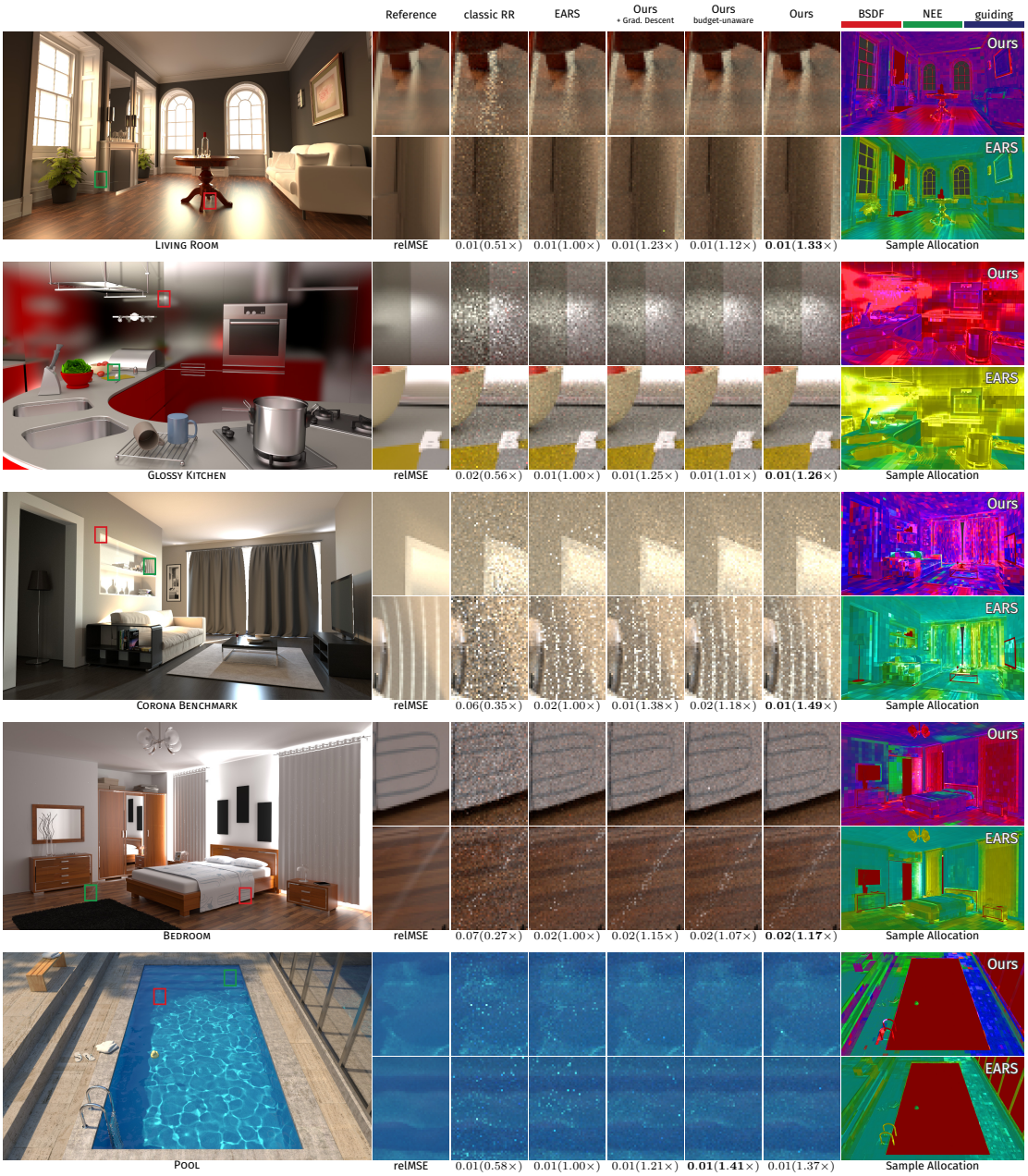
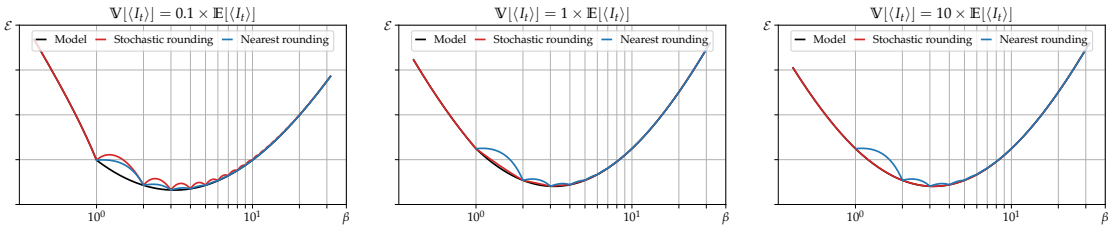


Figure 4.3: *We compare our method against state-of-the-art EARS applied to guiding in 5 scenes in a 5 minute equal-time comparison. The numbers below the crops correspond to the relative mean squared error (relMSE, lower is better), with the speedup compared against EARS in parentheses (higher is better). To the right, we visualize the allocation decisions EARS and our method take. The color channels red, green and blue correspond to BSDF, NEE, and guiding samples, respectively, with brighter colors indicating higher sample counts. The other 5 scenes we tested are provided in appendix A. Our "+ Grad. Descent" test uses gradient descent for the BSDF/guiding mixture and our method to combine it with NEE using budget-aware weights.*



(a) *Low variance* **(b)** *Medium variance* **(c)** *High variance*

Figure 4.4: *We compare the efficiency of stochastic rounding, rounding to nearest integers, and the simplified model for the secondary estimator variance our work has in common with previous works. The performance of stochastic rounding depends on the ratio of estimator variance to expected value. All models perform the same for RR ($\beta < 1$) and at integer splitting factors.* ***(a)*** *For low variance estimators, stochastic rounding can introduce more noise than nearest rounding.* ***(b)*** *When the estimator variance is on the same order of magnitude as its expected value, stochastic rounding becomes the best option.* ***(c)*** *Stochastic rounding and the simplified model begin to agree perfectly when the variance is high.*

4.2.5 Rounding

Rounding Variants Rounding fractional sample counts to integers, which is required since we can only take an integer amount of samples, always introduces additional noise in the secondary estimator. The shape and magnitude of noise depend on how the secondary estimator is normalized after rounding. We evaluate two unbiased options: **(1)** dividing by the real-valued sample count as done by previous works [Vorba et al., 2016; Rath et al., 2022], and **(2)** dividing by the rounded sample count in the case of splitting and real-valued count for RR. The theoretical difference in noise is demonstrated in figure 4.4. In practice, we have found that both techniques perform equally well, and, hence for consistency with previous works, we normalize by real-valued sample counts in all of our evaluations.

Low Discrepancy Rounding In practice, we apply a small trick to reduce the variance introduced by stochastic rounding by roughly 2%. With naïve stochastic rounding, every technique draws an independent random number to decide whether to round the number of samples up or down. Looking at the sampling process as a whole, the decision is not whether 1 sample is taken or not, but rather whether n_t samples are taken, each individually. With an unlucky draw none of the allocations are rounded up, whereas with a lucky draw, all of them are rounded up. We instead only draw a *single* random number to decide the rounding of *all* techniques (see alg. 1). After a technique is rounded, the probability of rounding up the next sample increases or decreases depending on whether the previous ones were rounded down or up, respectively. Overall, this leads to the random process only deciding between taking 1 sample more or less, instead of n_t samples, which reduces variance slightly.

ALGORITHM 1: Low Discrepancy Stochastic Rounding

```
/* Given stochastic numbers of samples β_t ∈ ℝ_0^+ for techniques 1 ≤ t ≤ n_t  */
r ← random() ;                          /* Obtain uniform random number in [0,1) */
for t = 1 to n_t do                          /* Round all n_t sample allocations */
    γ_t ← ⌊β_t + r⌋ ;                  /* Stochastically obtain γ_t ∈ {⌊β_t⌋,⌈β_t⌉} */
    r ← r + β_t − γ_t ;      /* Remaining fractional random part for next iter */
end
/* Use each γ_t ∈ ℕ as number of samples for technique t.                       */
```

5

Application II
Bidirectional Path Tracing

We discuss the second application, bidirectional path tracing, in this chapter. Analogous to the previous one, we first clarify the implementation before turning to the evaluation.

Our bidirectional application considers three techniques: path tracing, connections with light paths, and NEE. Although NEE is a special case of light paths, its different variance and cost properties (it only estimates direct light) make it beneficial to assign it its own budget. In theory, performance could further be improved by assigning separate budgets of light paths for any length, but we leave exploring this to future work.

It might make sense to perform splitting for both the camera and light paths, but doing so comes with its own set of challenges. Most importantly, splitting paths on both ends and combining all acquired intersections leads to an exponential increase in connections between the paths, which is computationally expensive. Thus, we resort to throughput-based RR for the light path and only perform splitting on the camera path.

We leave exploring more sophisticated variants of bidirectional rendering, such as vertex merging, light path caches, or stochastic connections as future work.

5.1 Implementation

The implementation of our second application is analogous to the first. Local estimates are recorded in the octree data structure proposed by EARS [Rath et al., 2022]. To execute our fixed-point scheme, we rewrite Mitsuba's BDPT implementation to become progressive: Rendering occurs in short iterations that grow in duration over time, with the final image being the inverse-variance weighted combination of all iterations [Vorba et al., 2019]. We apply a similar outlier rejection to EARS [Rath et al., 2022], which eliminates the 10 brightest pixels in each pass for the image variance estimate. Similar to guiding, we additionally tried applying an outlier rejection to the local samples recorded in the octree but found that this results, on average, in a 6% performance decrease on the evaluated scenes: Since in this application sampling is not learned adaptively, it seems beneficial to perform excessive splitting in the presence of outliers.

5.2 Evaluation

We compare against EARS applied to camera paths ("EARS") and provide the results of a vanilla BDPT baseline ("classic RR"). Sample allocation/RRS is only performed on the camera path side, and throughput-based RR is used to construct light paths. Additionally, we implemented a variation of efficiency-aware MIS [Grittmann et al., 2022] ("brute-force"), which is applied to pure bidirectional path tracing without merging and makes decisions on a per-pixel basis instead of a single global one to allocate the budget.

5.2.1 Setup

All images are rendered for 5 minutes on an AMD Ryzen Threadripper 2950X with 16 cores with a maximum path length of 40. The first 3 iterations use classic RR also for the camera path to reduce the noise of the estimates. As in guiding, we evaluate a version with budget-unaware MIS weights which, on average, performs 13% worse. Averaged, we achieve a speedup of 1.6× over EARS, and a speedup of 2.6× over classic BDPT.

5.2.2 Results

Figure 5.1 showcases a variety of interesting scenes. For example, in the Glossy Kitchen, the glossy features neither benefit greatly from light connections nor NEE. Our method can automatically detect and tune down these techniques locally, while still using connections on diffuse surfaces, yielding a 74% speedup. Plain RRS cannot separate the contributions of different techniques, wasting compute time on negligible contributions.

The Glossy Bathroom similarly avoids NEE, but for a different reason: no light is directly visible in this scene. Instead, our method relies on BSDF sampling for glossy surfaces and light paths for rougher surfaces. Similar to EARS, our method uses less splitting overall (indicated by the darker colors). EARS exhibits the same issue as in its original paper due to issues with variance estimates on glossy surfaces. While our method fundamentally suffers from the same issue, this is offset by an overall better sample allocation resulting in a 90% speedup over EARS.

In contrast to the previous scenes, the Living Room is a scene where NEE is invaluable, given that guiding is unavailable. The sofa on the right is directly illuminated by the light sources and well estimated by NEE. Random BSDF bounces from diffuse elements in the scene ending at this location can exploit this effective technique. Light paths are avoided entirely, as they struggle to penetrate the room.

The Bedroom (figure 5.2) and Bookshelf both feature a difficult indirect illumination, with light only induced from small areas: from behind the curtains and from small light sources, respectively. For BSDF samples, it is hard to find a contributing path without the assistance of NEE or connections. Hence, at intersections without occluders, our method increases the NEE sampling budget. With occluders present, light paths can help avoid them. Deciding this locally in space generates superior estimations of indirect illumination, whereas previous work is limited to a single decision for the whole path.

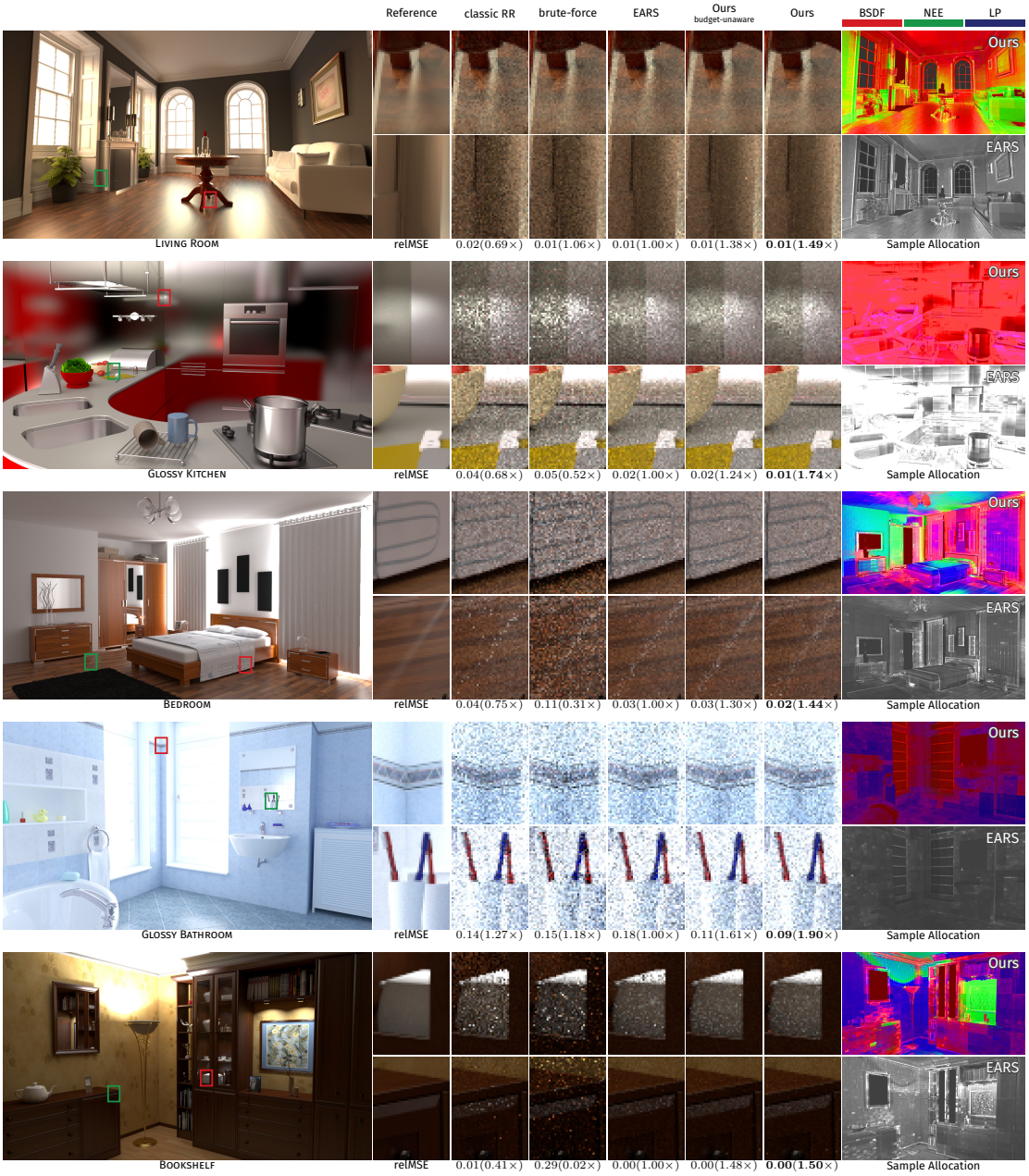
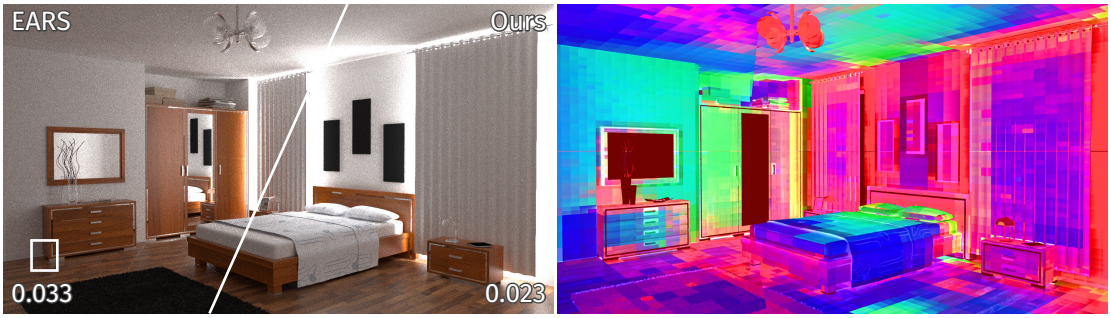


Figure 5.1: *We compare our method against state-of-the-art EARS applied to BDPT in 5 scenes in a 5 minute equal-time comparison. The numbers below the crops correspond to the relative mean squared error (relMSE, lower is better), with the speedup compared against EARS in parentheses (higher is better). To the right, we visualize the allocation decisions EARS and our method take. The color channels red, green and blue correspond to BSDF, NEE, and light path samples, respectively, with brighter colors indicating higher sample counts. The other 5 scenes we tested are provided in appendix B.*

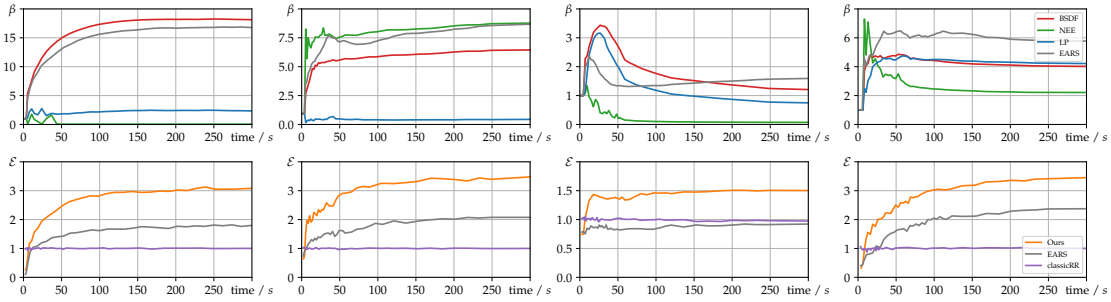


(a) *Comparison against previous work* **(b)** *Our sample allocation*

Figure 5.2: *A closer look at our sample allocation at the example of the* Bedroom *scene. In* ***(a)*** *we show a comparison with previous work, EARS, with our technique and give the relative mean squared errors at the bottom. Each color channel of* ***(b)*** *represents one technique: BSDF samples (red), next event estimation (green), and light paths (blue). The intensity is proportional to the number of samples allocated to each technique (higher intensity for more samples). The vibrant colors in* ***(b)*** *demonstrate clearly that not all techniques are equally useful at every point in space, making our method, which can separate the contributions of different techniques, superior. For instance, on the ceiling (and on the bed) close to the windows, where the light is close, NEE proves to be indispensable. Without direct light connections possible at the first hitpoint, BSDF sampling and path guiding take over. In particular, path guiding is the technique of choice for difficult regions, far away from the light, in which it is unlikely to find contributions in the next few bounces with only BSDF samples because of occluders, e.g., the bottom left part next to the sideboard.*

5.2.3 Convergence

As in path guiding, we have verified that repeated runs result in the same performance. We similarly provide plots for the convergence of average sample allocations across techniques and overall efficiency on the same scenes in figure 5.3.



(a) *Glossy Kitchen* **(b)** *Dining Room* **(c)** *Glossy Bathroom* **(d)** *Bookshelf*

Figure 5.3: *We show convergence of the average primary hit point sample allocation (first row) and overall efficiency (second row, higher is better) in four selected scenes for BDPT. The setup is the same as in the main text. Our fixed-point scheme quickly converges towards solutions that are better than the baseline and previous state-of-the-art.*

5.2.4 Rounding

The same stochastic, low discrepancy rounding as described in the path guiding chapter in section 4.2.5 was analyzed, implemented and used in the bidirectional implementation.

6 Limitations and Future Work

Our method efficiently allocates sampling budgets while remaining flexible and extensible, but relies heavily on the accuracy and robustness of local estimates for variance.

Overhead Our method introduces the overhead of collecting local estimates which does not always pay off. In the presence of very short rendering times, the collected estimates have not converged enough yet when the rendering finishes. In figure 6.1, we present the Country Kitchen where a rendering time of 30 seconds does not suffice, even though we outperform BDPT quickly afterward with a rendering time of less than a minute. In trivial scenes, e.g., the Cornell Box depicted in the same figure, the overhead is not detrimental in the sense that we are much worse, but we also do not surpass the baseline's performance either. With a low rendering time of 10 seconds, as depicted, we are visibly worse still. Apart from the trivialness of the scene and, thus, the irrelevance of sophisticated techniques, rendering slightly longer makes the difference vanish.

Local Estimates Outliers in the local estimates can reduce performance due to excessive splitting. We used a simple outlier rejection scheme in the guiding application, in which the problem was most pronounced, but future work could look into analyzing this further and applying more sophisticated techniques [Zirr et al., 2018].

Discontinuity Artefacts Noise in the spatial caches can lead to noticeable discontinuities due to the discretization through the cache, which is an existing problem in previous work already [Rath et al., 2022]. It is especially pronounced with shorter render times. Possible mitigations could be to apply spatial denoising to the local estimates or interpolate them to arrive at smoothly varying sample counts. Additionally, using all the samples for each technique's estimates regardless of which technique produced them could improve the general estimation quality, thus reducing the noise in the caches.

Proxy Accuracy Our proxy is only optimal for budget-unaware MIS weights. While it performs robustly for budget-aware weights across all our tested scenes, a more sophisticated proxy can likely increase efficiency even further. A brief analysis of the proxy's error is presented in figure 3.1 at several 1-dimensional examples.

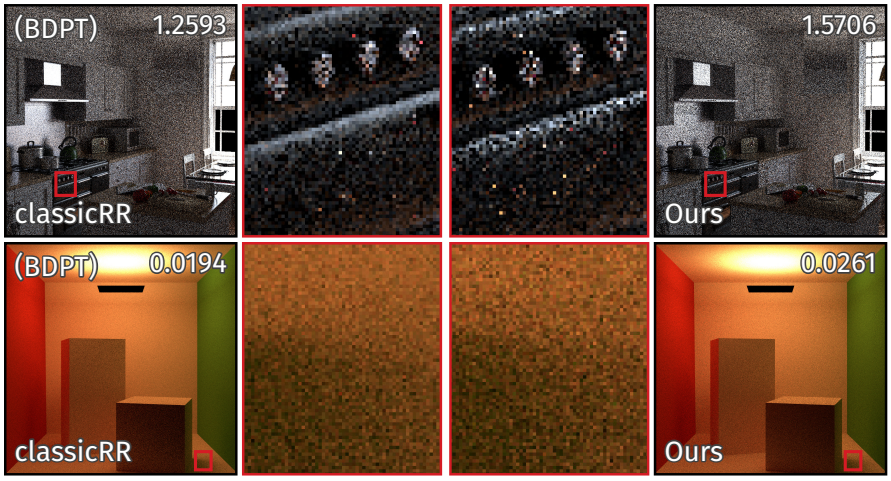


Figure 6.1: *We show two failure cases. The* Country Kitchen *is a noisy scene in which our method requires time to converge and convert the overhead into an advantage. The 30 seconds it was rendered for does not suffice, but by rendering only slightly longer we surpass BDPT's performance. In the second case, we present a* Cornell Box *rendered for 10 seconds. In trivial scenes, the overhead we introduce dominates the advantage – especially in the presence of short render times. In this scene, rendering longer closes the gap. We give the relMSE in the upper right.*

Combinations With Other Strategies As a general MIS budget allocation technique, our work is orthogonal to other commonly used methods allowing combinations thereof to leverage their respective strengths. For instance, our bidirectional application always traces new light paths for every sample that is performed. Introducing a light cache to reuse previously sampled paths could save precious time by only having to trace shadow rays. Moreover, we only showed the viability using 3 techniques in practice. However, the theory holds for an arbitrary number. Therefore, exploring combinations of more techniques might provide additional insights. For example, one could consider connections to light paths of different fixed lengths, similar to what we did for NEE.

Dynamic Scenes We only considered static scenes. Future work could look into applying our method to dynamic scenes, such as real-time rendering or animations, in which local estimates or sample allocations from earlier frames could be re-used to kickstart the fixed-point scheme.

MIS Weights We limited our analysis to the most common MIS weighting function, the balance heuristic [Veach et al., 1995b]. We briefly tested the power heuristic with different exponents without finding noticeable differences. Future work could investigate the impact of more sophisticated budget-aware weights on our proxy model, such as the optimal weights proposed by Kondapaneni et al. [2019].

Applications Outside of Rendering We presented and tested our approach in the context of rendering, but our theory is general and could be applied to any Monte Carlo integration problem that relies on multi-sample MIS estimation.

7

Conclusion

Based on the insights from previous work on Russian roulette and splitting, we present a novel theory on a multi-sample MIS allocation strategy that combines the strengths of both fields. We demonstrate that independently optimizing the one-sample MIS mixture and RRS factors leads to suboptimal decisions. Therefore, we join the research areas of MIS sample allocation and RRS by jointly optimizing the budget allocation in a multi-sample combination.

Our theory considers the possibility of combining techniques that estimate different parts of the integrand – for instance, next event estimation can only estimate direct light, whereas BSDF samples estimate both direct and indirect light.

Our approach is based on a fixed-point scheme and is proven to yield optimal performance for budget-unaware MIS weights provided that variances and costs are accurately known. We demonstrate that our method achieves consistent speedups over previous approaches, even with noisy estimates and budget-aware weights, in both path guiding and bidirectional path tracing.

Although we focused on rendering as our application, the presented theory can be applied to any Monte Carlo integration problem where multiple sampling techniques are combined through MIS – even outside of rendering.

8

Co-Authors' Contributions

A shortened version of this thesis was submitted as a paper to SIGGRAPH Asia 2024. In this context, I worked together with Alexander Rath and Ömercan Yazici to achieve what is presented in this thesis. In this chapter, I state, in detail, which parts are my work and which are not.

First and foremost, all implementations and evaluations that are presented were done by me. The only exception to this is the implementation of the efficiency-aware MIS variant [Grittmann et al., 2022], denoted as "brute-force" throughout this thesis, which was done by Ömercan Yazici.

The theory and derivation of our approach were performed by me. Again, there is a small exception: The proof of optimality for the budget-unaware case, section 3.3.4, was produced by Alexander Rath since the time to the SIGGRAPH deadline was short.

Regarding the figures, figures 2.1 and 3.1 are the only figures not created by me, but by Alexander Rath instead. The remaining figures are my original work.

The contents of this thesis were written by me. For the paper we submitted, my co-authors also performed polishing passes and reformulated sentences. Some of these reformulations will undoubtedly have found their way into this thesis.

Bibliography

Áfra, Attila T. (2019). *Intel® Open Image Denoise*. URL: https://www.openimagedenoise.org/.

Arvo, James and David Kirk (Sept. 1990). "Particle Transport and Image Synthesis". In: *ACM SIGGRAPH Computer Graphics* 24.4, pp. 63–66. ISSN: 0097-8930. DOI: 10.1145/97880.97886.

Bolin, Mark R. and Gary W. Meyer (1997). "An Error Metric for Monte Carlo Ray Tracing". In: *Rendering Techniques '97*. Ed. by Julie Dorsey and Philipp Slusallek. Eurographics. Vienna: Springer, pp. 57–68. ISBN: 978-3-7091-6858-5. DOI: 10.1007/978-3-7091-6858-5_6.

Georgiev, Iliyan et al. (Nov. 2012). "Light Transport Simulation with Vertex Connection and Merging". In: *ACM Transactions on Graphics* 31.6, pp. 1–10. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/2366145.2366211.

Grittmann, Pascal et al. (July 2022). "Efficiency-Aware Multiple Importance Sampling for Bidirectional Rendering Algorithms". In: *ACM Transactions on Graphics* 41.4, 80:1–80:12. ISSN: 0730-0301. DOI: 10.1145/3528223.3530126.

Hachisuka, Toshiya, Jacopo Pantaleoni, and Henrik Wann Jensen (Nov. 2012). "A Path Space Extension for Robust Light Transport Simulation". In: *ACM Transactions on Graphics* 31.6, 191:1–191:10. ISSN: 0730-0301. DOI: 10.1145/2366145.2366210.

Hanika, Johannes, Marc Droske, and Luca Fascione (July 2015). "Manifold Next Event Estimation". In: *Computer Graphics Forum* 34.4, pp. 87–97. ISSN: 0167-7055, 1467-8659. DOI: 10.1111/cgf.12681.

Jakob, Wenzel (2010). *Mitsuba Renderer*. URL: https://mitsuba-renderer.org/.

Jensen, Henrik Wann (1995). "Importance Driven Path Tracing Using the Photon Map". In: *Rendering Techniques '95*. Ed. by Patrick M. Hanrahan and Werner Purgathofer. Vienna: Springer, pp. 326–335. ISBN: 978-3-7091-9430-0. DOI: 10.1007/978-3-7091-9430-0_31.

Jensen, Henrik Wann (1996). "Global Illumination Using Photon Maps". In: *Rendering Techniques '96*. Ed. by Xavier Pueyo and Peter Schröder. Vienna: Springer Vienna, pp. 21–30. ISBN: 978-3-211-82883-0 978-3-7091-7484-5. DOI: 10.1007/978-3-7091-7484-5_3.

Kajiya, James T. (Aug. 1986). "The Rendering Equation". In: *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '86. New York, NY, USA: Association for Computing Machinery, pp. 143–150. ISBN: 978-0-89791-196-2. DOI: 10.1145/15922.15902.

Kloek, T. and H. K. van Dijk (1978). "Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo". In: *Econometrica* 46.1, pp. 1–19. ISSN: 0012-9682. DOI: 10.2307/1913641. JSTOR: 1913641.

Kondapaneni, Ivo et al. (July 2019). "Optimal Multiple Importance Sampling". In: *ACM Transactions on Graphics* 38.4, 37:1–37:14. ISSN: 0730-0301. DOI: 10.1145/3306346.3323009.

Lafortune, Eric P. and Y. Willems (Dec. 1993). *Bi-Directional Path Tracing*.

Lu, H., R. Pacanowski, and X. Granier (2013). "Second-Order Approximation for Variance Reduction in Multiple Importance Sampling". In: *Computer Graphics Forum* 32.7, pp. 131–136. ISSN: 1467-8659. DOI: 10.1111/cgf.12220.

Müller, Thomas, Markus Gross, and Jan Novák (2017). "Practical Path Guiding for Efficient Light-Transport Simulation". In: *Computer Graphics Forum* 36.4, pp. 91–100. ISSN: 1467-8659. DOI: 10.1111/cgf.13227.

Murray, David et al. (May 2020). "On Learning the Best Local Balancing Strategy". In: *Eurographics 2020 - Annual Conference of the European Association for Computer Graphics*. Vol. xx, p. 1. DOI: 10.2312/egs.20201009.

Nabata, Kosuke, Kei Iwasaki, and Yoshinori Dobashi (Mar. 2020). "Resampling-Aware Weighting Functions for Bidirectional Path Tracing Using Multiple Light Sub-Paths". In: *ACM Transactions on Graphics* 39.2, 15:1–15:11. ISSN: 0730-0301. DOI: 10.1145/3338994.

Pharr, Matt, Wenzel Jakob, and Greg Humphreys, eds. (2016). *Physically Based Rendering: From Theory to Implementation (3rd Ed.)* Morgan Kaufmann Publishers Inc. ISBN: 978-0-12-800645-0.

Popov, Stefan et al. (2015). "Probabilistic Connections for Bidirectional Path Tracing". In: *Computer Graphics Forum* 34.4, pp. 75–86. ISSN: 1467-8659. DOI: 10.1111/cgf.12680.

Rath, Alexander, Ömercan Yazici, and Philipp Slusallek (July 2023). "Focal Path Guiding for Light Transport Simulation". In: *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH '23. New York, NY, USA: Association for Computing Machinery, pp. 1–10. ISBN: 9798400701597. DOI: 10.1145/3588432.3591543.

Rath, Alexander et al. (Aug. 2020). "Variance-Aware Path Guiding". In: *ACM Transactions on Graphics (TOG)* 39.4, 151:151:1–151:151:12. ISSN: 0730-0301. DOI: 10.1145/3386569.3392441.

Rath, Alexander et al. (July 2022). "EARS: Efficiency-Aware Russian Roulette and Splitting". In: *ACM Transactions on Graphics* 41.4, 81:1–81:14. ISSN: 0730-0301. DOI: 10.1145/3528223.3530168.

Ruppert, Lukas, Sebastian Herholz, and Hendrik P. A. Lensch (Aug. 2020). "Robust Fitting of Parallax-Aware Mixtures for Path Guiding". In: *ACM Transactions on Graphics* 39.4, 147:147:1–147:147:15. ISSN: 0730-0301. DOI: 10.1145/3386569.3392421.

Sbert, Mateu, Vlastimil Havran, and László Szirmay-Kalos (2019). "Optimal Deterministic Mixture Sampling". In: *Eurographics 2019 - Short Papers*, 4 pages. ISSN: 1017-4656. DOI: 10.2312/EGS.20191018.

Shirley, Peter et al. (1995). "Global Illumination via Density-Estimation". In: *Rendering Techniques '95*. Ed. by Patrick M. Hanrahan and Werner Purgathofer. Vienna: Springer Vienna, pp. 219–230. ISBN: 978-3-211-82733-8 978-3-7091-9430-0. DOI: 10.1007/978-3-7091-9430-0_21.

Szirmay-Kalos, Laszlo and Mateu Sbert (2022). "Robust Sample Budget Allocation for MIS". In: *Eurographics 2022 - Short Papers*, pp. 17–20. DOI: 10.2312/EGS.20221022.

Veach, Eric (Dec. 1997). "Robust Monte Carlo Methods for Light Transport Simulation". PhD thesis.

Veach, Eric and Leonidas Guibas (1995a). "Bidirectional Estimators for Light Transport". In: *Photorealistic Rendering Techniques*. Ed. by Georgios Sakas, Stefan Müller, and Peter Shirley. Focus on Computer Graphics. Berlin, Heidelberg: Springer, pp. 145–167. ISBN: 978-3-642-87825-1. DOI: 10.1007/978-3-642-87825-1_11.

Veach, Eric and Leonidas J. Guibas (Sept. 1995b). "Optimally Combining Sampling Techniques for Monte Carlo Rendering". In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. New York, NY, USA: Association for Computing Machinery, pp. 419–428. ISBN: 978-0-89791-701-8. DOI: 10.1145/218380.218498.

Vorba, Jiří and Jaroslav Křivánek (July 2016). "Adjoint-Driven Russian Roulette and Splitting in Light Transport Simulation". In: *ACM Transactions on Graphics* 35.4, 42:1–42:11. ISSN: 0730-0301. DOI: 10.1145/2897824.2925912.

Vorba, Jiří et al. (July 2014). "On-Line Learning of Parametric Mixture Models for Light Transport Simulation". In: *ACM Transactions on Graphics* 33.4, 101:1–101:11. ISSN: 0730-0301. DOI: 10.1145/2601097.2601203.

Vorba, Jiří et al. (July 2019). "Path Guiding in Production". In: *ACM SIGGRAPH 2019 Courses*. SIGGRAPH '19. New York, NY, USA: Association for Computing Machinery, pp. 1–77. ISBN: 978-1-4503-6307-5. DOI: 10.1145/3305366.3328091.

Walter, Bruce et al. (July 1997). "Global Illumination Using Local Linear Density Estimation". In: *ACM Transactions on Graphics* 16.3, pp. 217–259. ISSN: 0730-0301. DOI: 10.1145/256157.256158.

Zeltner, Tizian, Iliyan Georgiev, and Wenzel Jakob (Aug. 2020). "Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints". In: *ACM Transactions on Graphics* 39.4, 149:149:1–149:149:15. ISSN: 0730-0301. DOI: 10.1145/3386569.3392408.

Zirr, Tobias, Johannes Hanika, and Carsten Dachsbacher (2018). "Re-Weighting Firefly Samples for Improved Finite-Sample Monte Carlo Estimates". In: *Computer Graphics Forum* 37.6, pp. 410–421. ISSN: 1467-8659. DOI: 10.1111/cgf.13335.

Appendices

A

Additional Guiding Results

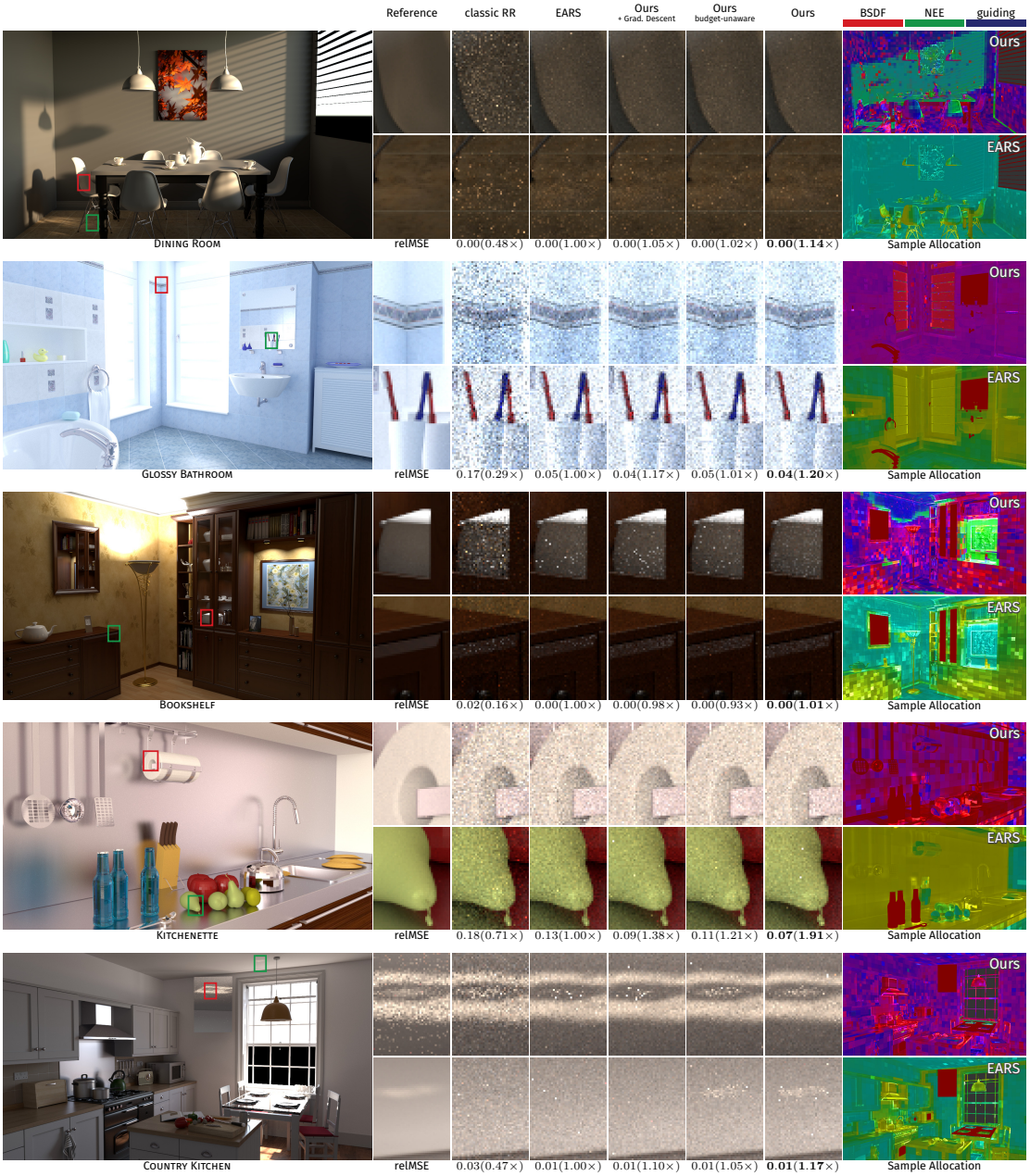


Figure A.1: *We compare our method against state-of-the-art EARS applied to guiding in the 5 scenes not provided in the main text in a 5 minute equal-time comparison. The numbers below the crops correspond to the relative mean squared error (relMSE, lower is better), with the speedup compared against EARS in parentheses (higher is better).*

B

Additional BDPT Results

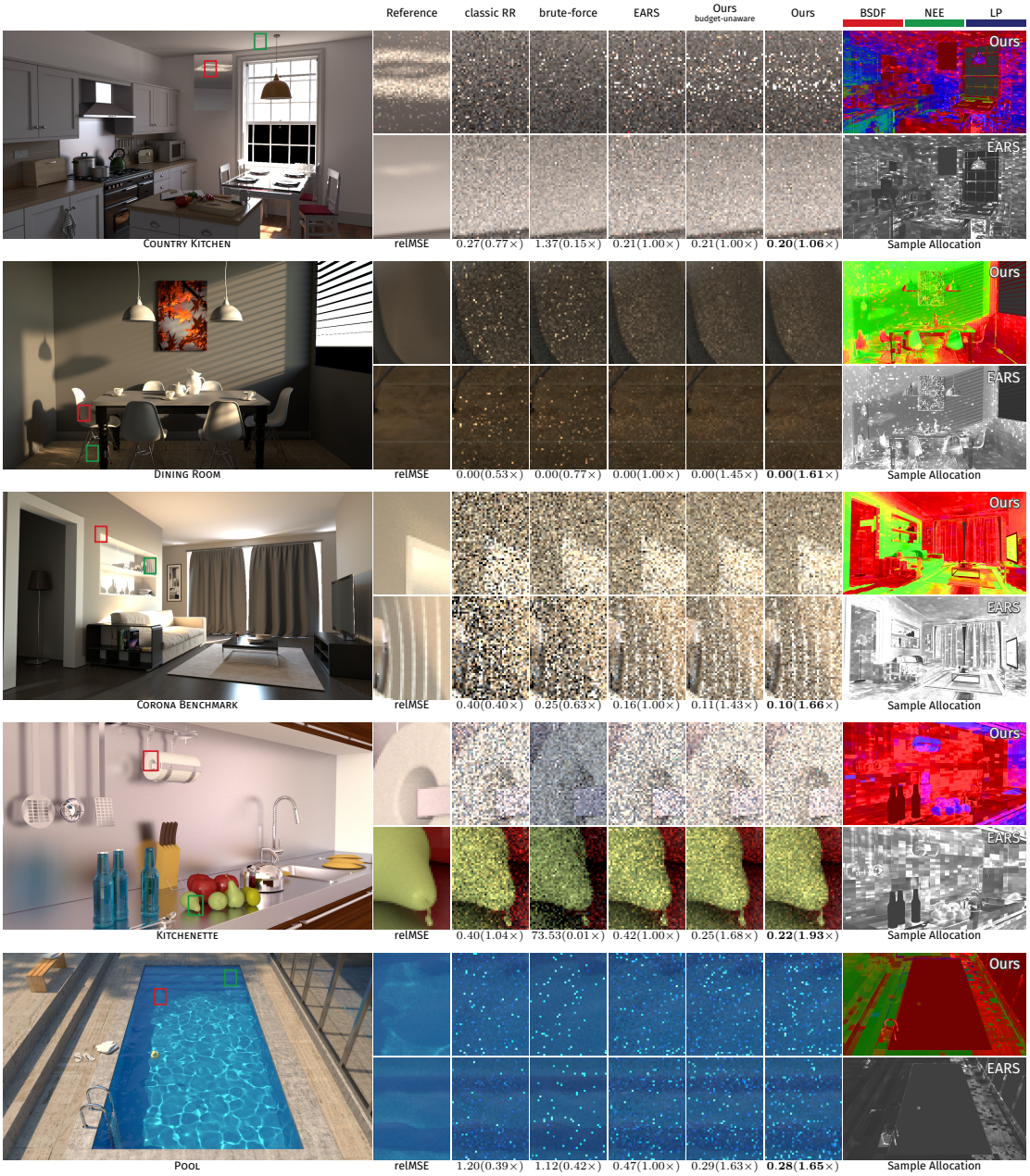


Figure B.1: *We compare our method against state-of-the-art EARS applied to BDPT in the 5 scenes not provided in the main text in a 5 minute equal-time comparison. The numbers below the crops correspond to the relative mean squared error (relMSE, lower is better), with the speedup compared against EARS in parentheses (higher is better).*

Saarland University

MARS: Multi-sample Allocation through Russian roulette and Splitting

Joshua Meyer
Student No. 2573323

Saarbrücken, June 2024