

Joint Sampling and Optimisation for Inverse Rendering

Martin Balint

Max Planck Institute for Informatics
Saarbruecken, Germany
mbalint@mpi-inf.mpg.de

Hans-Peter Seidel

Max Planck Institute for Informatics
Saarbruecken, Germany
hpseidel@mpi-inf.mpg.de

Karol Myszkowski

Max Planck Institute for Informatics
Saarbruecken, Germany
karol@mpi-inf.mpg.de

Gurprit Singh

Max Planck Institute for Informatics
Saarbruecken, Germany
gsingh@mpi-inf.mpg.de

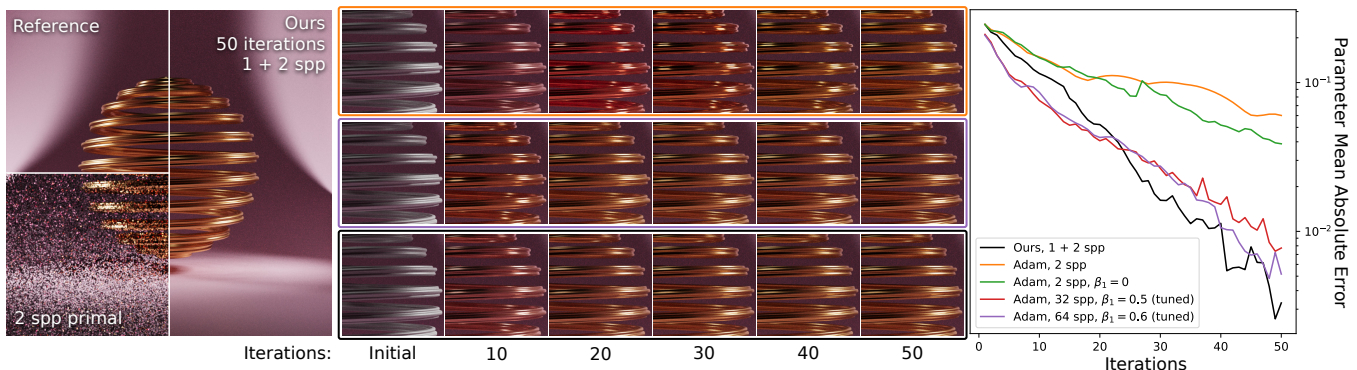


Figure 1: We simultaneously optimise the spiral’s base colour, metalness and roughness. Our gradient *meta-estimator* quickly recovers the gradient with just one finite-difference and two proportional samples per pixel. Adam’s default first moment estimator ($\beta_1 = 0.9$) cannot adapt as quickly and often overshoots. After tuning Adam’s hyperparameters for this specific problem, it approaches our method at 32 spp and matches it only at 64 spp.

ABSTRACT

When dealing with difficult inverse problems such as inverse rendering, using Monte Carlo estimated gradients to optimise parameters can slow down convergence due to variance. Averaging many gradient samples in each iteration reduces this variance trivially. However, for problems that require thousands of optimisation iterations, the computational cost of this approach rises quickly.

We derive a theoretical framework for interleaving sampling and optimisation. We update and reuse past samples with low-variance finite-difference estimators that describe the change in the estimated gradients between each iteration. By combining proportional and finite-difference samples, we continuously reduce the variance of our novel gradient meta-estimators throughout the optimisation process. We investigate how our estimator interlinks with Adam and derive a stable combination.

We implement our method for inverse path tracing and demonstrate how our estimator speeds up convergence on difficult optimisation tasks.

CCS CONCEPTS

• Computing methodologies → Ray tracing.

KEYWORDS

differentiable rendering, inverse rendering, gradient estimation, gradient descent

ACM Reference Format:

Martin Balint, Karol Myszkowski, Hans-Peter Seidel, and Gurprit Singh. 2023. Joint Sampling and Optimisation for Inverse Rendering. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23)*, December 12–15, 2023, Sydney, NSW, Australia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3610548.3618244>

1 INTRODUCTION

Forward rendering involves solving the light transport integrals with given scene parameters, e.g., geometry, materials, textures, by numerically estimating the rendering equation [Kajiya 1986; Pharr et al. 2016]. Inverse rendering reverses this process by estimating the scene parameters starting from a given target image. This process involves inverting the rendering equation [Kajiya 1986].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0315-7/23/12.

<https://doi.org/10.1145/3610548.3618244>

Such inversion tasks are typically solved by gradient descent. Physically-based differentiable renderers [Jakob et al. 2022; Li et al. 2018; Zhang et al. 2020] facilitate these gradient-based optimisation methods. The process involves backpropagating from an underlying loss function, quantifying the disparity between an image generated with the current parameters and the target image, resulting in gradients w.r.t. the scene parameters. These gradient values are approximated from a given set of samples, and subsequently, the scene parameters are adjusted using these gradients to minimise the loss. The ultimate goal is to converge to a parameter set that produces the target image. Due to the nature of Monte Carlo integration, the estimated gradients can be extremely noisy, hampering the performance of gradient-based optimisers. In inverse rendering, gradients are estimated with tens to hundreds of rays traced per pixel [Nimier-David et al. 2020; Zhang et al. 2019] to minimise noise. Usually, inverse rendering requires hundreds to thousands of iterations to converge; recomputing these gradient estimates in every iteration comes at a large cost.

In this paper, we propose a theoretical framework that jointly considers sampling and optimisation by deriving a theoretical framework for interleaving them. We reuse past samples without introducing bias thanks to finite-difference estimators that describe the change in the estimated gradients between each iteration. By combining proportional and finite-difference samples, we continuously reduce the variance of our novel gradient meta-estimators throughout the optimisation process.

First, we introduce our meta-estimation theory and then discuss our variance estimation strategies used to derive coefficients for our meta-estimator. We investigate how our estimator interlinks with Adam and derive a stable combination. We run experiments to evaluate our method in the context of inverse rendering. Finally, we discuss our method concerning future and concurrent work and give our conclusions.

Our contributions include:

- Meta-estimation theory on combining proportional and finite-difference estimators.
- Practical variance approximation techniques to effectively implement meta-estimation.
- Implementation and evaluation of meta-estimation for inverse rendering. (We will release our code upon acceptance.)

2 RELATED WORK

Differentiable Path Tracing. Path tracing accounts for global illumination through physically accurate light transport by Monte Carlo integration of the rendering equation [Kajiya 1986]. Recent works proposed various approaches to differentiate such Monte Carlo integrals and estimate derivatives w.r.t. scene parameters. [Jakob et al. 2022; Li et al. 2018; Nimier-David et al. 2020; Zeltner et al. 2021; Zhang et al. 2020]. While our work applies to any method using gradient descent on Monte Carlo estimated gradients, we mainly experiment with Path Replay Backpropagation [Vicini et al. 2021]; a well-established state-of-the-art method for inverse path tracing, implemented in Mitsuba 3 [Jakob et al. 2022].

Previous works have focused on sampling strategies [Bangaru et al. 2020; Yan et al. 2022; Zhang et al. 2021] and improving the optimisation itself [Nimier-David et al. 2020; Vicini et al. 2021]

to reduce noise in the gradients. Particularly relevant, concurrent work by Chang et al. [2023] applies ReSTIR [Bitterli et al. 2020] in parameter-space with the same goal of reducing the variance of the estimated gradients.

Ray Differentials. Igehy [1999] first proposed ray differentials to approximate derivatives for texture interpolation and anisotropic filtering. Kettunen et al. [2015] combine ray differentials with gradient-domain MLT [Lehtinen et al. 2013] to build unbiased image-space gradient estimators for gradient-domain path tracing. Manzi et al. [2016] extend their work to the spatiotemporal domain.

We apply the general idea of finite-difference estimation to temporal gradient averaging on a set of parameters. As we do not assume any structure between individual parameters, we forgo Poisson reconstruction and instead statistically average proportional and integrate finite-difference samples.

Gradient averaging. Iterating with the arithmetic mean of gradient samples is well-understood to improve the convergence of optimisers. Several recursive schemes [Nesterov 1983; Polyak and Juditsky 1992] achieve fast convergence on convex problems [Moulines and Bach 2011], with some proving particularly useful in deep learning [Sutskever et al. 2013]. Kingma and Ba [2014] propose start-up bias-corrected exponential moving averaging on gradients; Adam remains the de-facto optimisation algorithm for deep learning and inverse rendering applications.

In recent work, Gower et al. [2020] analyse gradient averaging methods based on finite sums; they show improvements in convergence analogous to our work, although limited to convex problems. Unfortunately, the finite-sum setting of algorithms like SAGA [Defazio et al. 2014] and SVRG [Johnson and Zhang 2013] does not generalise to Monte Carlo integration [Nicolet et al. 2023].

Reducing the gradient variance is well understood to improve convergence speed and stability. Previous works on optimising neural networks increase the batch size to reduce this variance, which is often preferable over slower learning rates [Smith et al. 2018].

Control Variates. Fieller and Hartley [1954] first propose control variates as a weighted combination of correlated estimators, one of which must be of a closed-form integral. Owen [2013] shows that the optimal control weight is proportional to the covariance of the estimators. Rousselle et al. [2016] generalise control variates to any pair of correlated estimators through two-level Monte Carlo integration; they apply their work to spatiotemporal gradient-domain rendering. Concurrent with our work, Nicolet et al. [2023] further generalise control variates to recursive estimation, applying it to primal renderings in the context of inverse path tracing.

Our work is distinctively different from control variates in that we build on an independent finite-difference estimator rather than a pair of correlated estimators. In particular, this formulation lets us avoid covariance terms in our weighting scheme.

3 DIFFERENTIAL META-ESTIMATORS

Various Monte Carlo methods estimate a sequence of integrals. Often, each integral is a function of the previous one, with the sequence converging to a solution. Optimisation via inverse Monte

Carlo is a prime example; we estimate gradients in each iteration, adjust parameters accordingly, and repeat the process.

Our work is focused on improving the convergence speed and stability of the optimisation process by reducing the variance of the estimated gradients. We draw inspiration from control theory, specifically noise reduction through the combination of proportional and differential signals. These methods assume that samples are drawn from known probability distributions, usually normal distributions with known variances [Kalman 1960]. Unfortunately, we cannot make such assumptions when dealing with Monte Carlo noise.

We combine two estimators: a *proportional estimator* $\langle F_i \rangle$ — any Monte Carlo gradient estimator sampled independently between iterations — and a *finite-difference estimator* $\langle \Delta F_i \rangle$ that estimates the change of a gradient between two consecutive iterations.

Notation. Let F_i denote the integral of function f over the domain \mathcal{X} , given parameters π_i for the current iteration $i \in [0, \infty)$:

$$F_i = \int_{\mathcal{X}} f(\mathbf{x}, \pi_i) d\mathbf{x} . \quad (1)$$

Let $\langle F_i \rangle$ denotes the (proportional) Monte Carlo estimator of F_i , meaning $\mathbb{E}[\langle F_i \rangle] = F_i$. For example, an estimator may sample f given a density p over \mathcal{X} :

$$\langle F_i \rangle = \frac{f(\mathbf{x}, \pi_i)}{p(\mathbf{x}, \pi_i)} . \quad (2)$$

Finite-difference estimation. We write the change of F_i between consecutive steps as:

$$\Delta F_i = F_i - F_{i-1} . \quad (3)$$

A finite-difference estimator ($\langle \Delta F_i \rangle$) estimates this change, ideally with a low variance. For example, we can substitute Equation (1) into Equation (3):

$$\Delta F_i = \int_{\mathcal{X}} f(\mathbf{x}, \pi_i) - f(\mathbf{x}, \pi_{i-1}) d\mathbf{x} , \quad (4)$$

and sample with a density p like in Equation (2):

$$\langle \Delta F_i \rangle = \frac{f(\mathbf{x}, \pi_i) - f(\mathbf{x}, \pi_{i-1})}{p(\mathbf{x}, \pi_i)} . \quad (5)$$

Here we assume that f is continuous w.r.t. π_i . Although our theory may apply to any Monte Carlo integral F_i , we analyse the case where $f(\mathbf{x}, \pi_i) = \partial L / \partial \pi_i$ is the gradient at the i -th iteration, for some objective L .

3.1 Meta-estimation

Our *meta-estimator* aims to optimally combine each proportional $\langle F_i \rangle$ and finite-difference estimator $\langle \Delta F_i \rangle$ available until the current step i . In this subsection, we establish the theoretical conditions required for a variance-optimal, unbiased combination of both estimators.

A finite-difference estimator $\langle \Delta F_i \rangle$, by its definition in Equation (3), lets us update any proportional estimator from the previous step ($\langle F_{i-1} \rangle$) to the current step i . This update can be done simply by addition without introducing any bias. We can easily show this

by expanding the expected value of the sum:

$$\begin{aligned} \mathbb{E}[\langle F_{i-1} \rangle + \langle \Delta F_i \rangle] &= \\ \mathbb{E}[\langle F_{i-1} \rangle] + \mathbb{E}[\langle \Delta F_i \rangle] &= F_{i-1} + F_i - F_{i-1} = F_i . \end{aligned} \quad (6)$$

We define our meta estimator as a weighted sum over the combination of all previous estimators up until step $i-1$ i.e., $\langle F_{i-1} \rangle_M$ and the current proportional and finite-difference estimators:

$$\langle F_i \rangle_M = \alpha_i (\langle F_{i-1} \rangle_M + \langle \Delta F_i \rangle) + (1 - \alpha_i) \langle F_i \rangle . \quad (7)$$

We initialise $\langle F_0 \rangle_M = \langle F_0 \rangle$. As we sample all $\langle F_i \rangle$ and $\langle \Delta F_i \rangle$ independently, the optimal α_i coefficients are given by inverse variance weighting [Sinha et al. 2011]:

$$\alpha_i = \frac{\text{Var}[\langle F_i \rangle]}{\text{Var}[\langle F_i \rangle] + \text{Var}[\langle F_{i-1} \rangle_M] + \text{Var}[\langle \Delta F_i \rangle]} . \quad (8)$$

This simple recurrent relation captures the variance optimal combination of all previously sampled proportional and finite-difference estimators.

To summarise, we introduce the optimal and unbiased meta-estimator in Equation (7). However, in practice, we use a more efficient implementation which suffers from some start-up bias. We describe this version in the following section. To visualise the estimators mentioned above and to motivate the design of our optimiser, we show a simple example in Figure 2.

4 VARIANCE ESTIMATION

Implementing Equation (7) in practice presents a challenge due to the unknown variances of our estimators. In this section, we describe how we approximate each variance term. We must balance three main objectives: the efficiency of our variance approximation methods, the optimality of our approximated α_i coefficients, and any bias potentially introduced to $\langle F_i \rangle_M$.

4.1 Proportional estimator variance

We approximate $\text{Var}[\langle F_i \rangle]$ as a zero-centred raw moment [Papoulis and Pillai 1984], computed using an *exponential moving average* (EMA) with coefficient β_F :

$$\text{Var}[\langle F_i \rangle] = \beta_F \text{Var}[\langle F_{i-1} \rangle] + (1 - \beta_F) \langle F_i \rangle^2 . \quad (9)$$

This formulation is similar to Adam's second moment estimate [Kingma and Ba 2014]. Here, $\text{Var}[\langle F_i \rangle]$ is a large, stable value that only varies in the initial stage of optimisation, when parameter changes can notably affect the problem's overall noise characteristics. A large β_F coefficient minimises the correlation of the approximate variance to any singular $\langle F_i \rangle$, resulting in an overall stable variance approximation.

4.2 Finite-difference estimator variance

As the proportional estimator reaches steady-state, we can safely assume that $\langle F_i \rangle$ are identically distributed over consecutive iterations. Unfortunately, the same observation does not apply to finite-difference estimation as this finite-difference depends on the optimisation step we take in the previous iteration. For example, a larger step will cause a larger shift in the per-parameter gradients.

To resolve this issue we propose to decouple the optimisation step size ($|\Delta \pi_i|$) from the approximated finite-difference estimator variance ($\text{Var}[\langle \Delta F_i \rangle]$). We begin the derivation of this decoupling

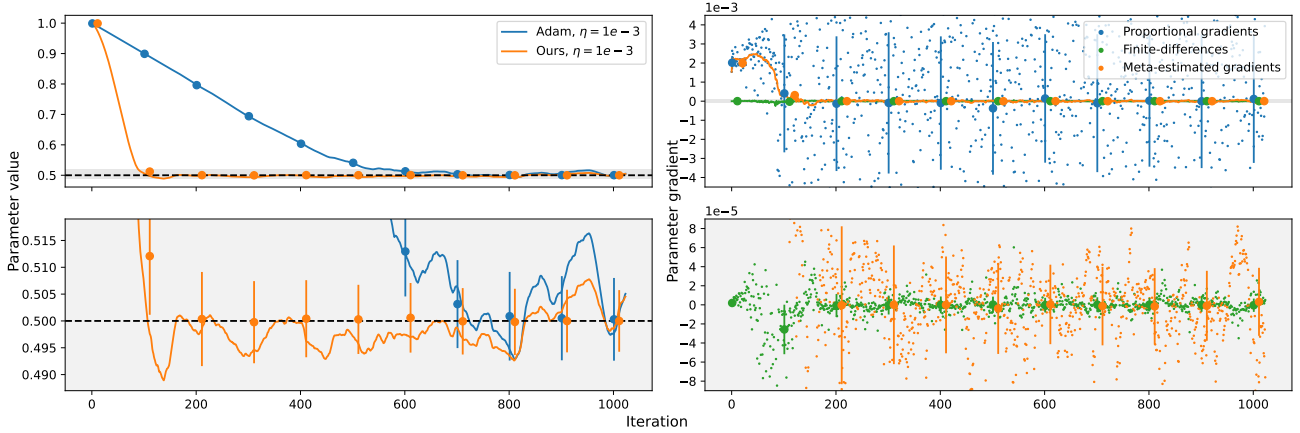


Figure 2: We optimise the rate parameter of an exponential distribution such that the mean of the distribution matches our target value of 2.0. We take 32 samples of the distribution in each iteration and compute an L2 loss between their mean and the target value. The bottom row shows insets of the graphs in the top row, indicated by grey regions. On the left, we show how Adam and our method reach the ground truth rate parameter 0.5. Error bars show the run-to-run variation of the optimised parameter. Our method converges significantly faster and is more stable than Adam. On the right, we show the estimators we use for our method; the proportional estimator has a large variance, while the finite-difference estimator is much less noisy. Our meta-estimator combines both, with its variance reducing over time.

by expanding the fraction in Equation (5) by the Euclidean step size $\|\Delta\pi_i\|_2$ of the previous iteration:

$$\langle \Delta F_i \rangle = \frac{f(\mathbf{x}, \pi_i) - f(\mathbf{x}, \pi_{i-1})}{p(\mathbf{x}, \pi_i)} = \frac{f(\mathbf{x}, \pi_i) - f(\mathbf{x}, \pi_{i-1})}{\|\Delta\pi_i\|_2 p(\mathbf{x}, \pi_i)} \|\Delta\pi_i\|_2. \quad (10)$$

For sufficiently small step sizes, we can rearrange the terms in Equation (10) to approximate the finite-difference of gradients f as the second-order gradient ($\partial f / \partial \pi$), times a unit-directional vector, times the left over terms:

$$\langle \Delta F_i \rangle \approx \left(\frac{\partial f}{\partial \pi}(\mathbf{x}, \pi_i) \cdot \frac{\Delta\pi_i}{\|\Delta\pi_i\|_2} \right) \frac{\|\Delta\pi_i\|_2}{p(\mathbf{x}, \pi_i)}. \quad (11)$$

Applying the variance operator to Equation (11) gives us the decoupled finite-difference variance ($\text{Var}[\langle \Delta F_i \rangle]_D$):

$$\begin{aligned} \text{Var}[\langle \Delta F_i \rangle] &\approx \text{Var} \left[\left(\frac{\partial f}{\partial \pi}(\mathbf{x}, \pi_i) \cdot \frac{\Delta\pi_i}{\|\Delta\pi_i\|_2} \right) \frac{1}{p(\mathbf{x}, \pi_i)} \right] \|\Delta\pi\|_2^2 \\ &= \text{Var}[\langle \Delta F_i \rangle]_D \|\Delta\pi\|_2^2. \end{aligned} \quad (12)$$

We use a zero-centred EMA, with a coefficient β_Δ , to approximate this decoupled variance as:

$$\text{Var}[\langle \Delta F_i \rangle]_D = \beta_\Delta \text{Var}[\langle \Delta F_{i-1} \rangle]_D + (1 - \beta_\Delta) \frac{\langle \Delta F_i \rangle^2}{\|\Delta\pi\|_2^2}. \quad (13)$$

We generally use a small β_Δ coefficient as $\text{Var}[\langle \Delta F_i \rangle]_D$ can change quickly and is typically less noisy than $\text{Var}[\langle F_i \rangle]$. Finally, we can rescale $\text{Var}[\langle \Delta F_i \rangle]_D$ to estimate $\text{Var}[\langle \Delta F_i \rangle]$:

$$\text{Var}[\langle \Delta F_i \rangle] = \text{Var}[\langle \Delta F_i \rangle]_D \|\Delta\pi\|_2^2. \quad (14)$$

We found this decoupled variance more closely distributed between iterations, better suited for approximation via moving averages.

4.3 Meta-estimator variance

We approximate the variance of our meta estimator in Equation (7) by recurrently applying the variance operator:

$$\begin{aligned} \text{Var}[\langle F_i \rangle_M] &= \alpha_i^2 (\text{Var}[\langle F_{i-1} \rangle_M] + \text{Var}[\langle \Delta F_i \rangle]) \\ &\quad + (1 - \alpha_i)^2 \text{Var}[\langle F_i \rangle]. \end{aligned} \quad (15)$$

Here we assume α_i to be a non-random value to simplify our mathematical derivation. Later in Section 6, we show the choice of α_i is less significant as long as its correlation with the gradient samples diminishes.

4.4 Alpha clipping

Meta-estimation is most vulnerable to underestimated $\text{Var}[\langle F_i \rangle_M]$; unless a significant $\text{Var}[\langle \Delta F_i \rangle]$ indicates a shift, $\langle F_i \rangle_M$ will only slowly correct its overconfidently estimated value by averaging $\langle F_i \rangle$ over many iterations. The risk of underestimation is the greatest while our exponential moving averages accumulate their initial samples. Clipping alpha based on the iteration resolves this risk:

$$\alpha_i = \min(\alpha_i, 1 - 1/(i + 1)). \quad (16)$$

Intuitively, Equation (16) constrains alpha by the perfect average of all previous estimates; any value above this must be overestimated. We generalise this observation to the entire optimisation process, assuming that $\text{Var}[\langle F_i \rangle]$ is similar in subsequent steps:

$$\alpha_i = \min(\alpha_i, 1/(2 - \alpha_{i-1})). \quad (17)$$

5 OPTIMISATION

Combining meta-estimation and optimisation creates a complex feedback loop; the meta-estimated gradients $\langle F_i \rangle_M$ depend on their finite-difference estimates $\langle \Delta F_i \rangle$, which depend on the optimiser's steps $\Delta\pi_i$, which, in turn, depend on the gradients estimated in the

previous step $\langle F_{i-1} \rangle_M$. It becomes crucial that the meta-estimator provides the optimiser with reliable gradients and that the optimiser makes steps that let the meta-estimator converge. We aim to combine Adam with meta-estimated gradients. We explain Adam’s variance approximation and update rule to show where we can integrate meta-estimation.

Adam [Kingma and Ba 2014] is well known for its robustness to outlier gradient samples; upon encountering an outlier, the estimated second moments adjust in the same step, swiftly pulling down the step size. This mechanism works because Adam first updates its second-moment estimate:

$$v_i = \beta_2 v_{i-1} + (1 - \beta_2) \langle F_i \rangle^2, \quad (18)$$

corrects the EMA startup bias:

$$\hat{v}_i = v_i / (1 - \beta_2^i), \quad (19)$$

and then divides the step size by its square root:

$$\Delta \pi_{i+1} = -\eta \frac{\hat{m}_i}{\sqrt{\hat{v}_i + \epsilon}}, \quad (20)$$

where m_i and v_i refer to Adam’s moment estimates, η to the learning rate, β_2 to Adam’s second moment coefficient, and ϵ to a small value to ensure numerical stability. α_i and $\text{Var}[\langle F_i \rangle]$ behave similarly in our case; first, we update $\text{Var}[\langle F_i \rangle]$ for the current step (Equation (9)), compute α_i (Equation (8)), and add the outlier gradient $\langle F_i \rangle$ to our meta-estimator $\langle F_i \rangle_M$ weighted by $(1 - \alpha_i)$ (Equation (7)). Therefore, just like β_2 for Adam, β_F offers a tradeoff between outlier robustness and estimation bias.

When using optimisers like RMSProp [Graves 2014] and Adam [Kingma and Ba 2014], lower variance gradients naturally accelerate convergence since these optimisers divide their step size by the standard deviation of the gradients (Equation (20)). Additionally, Momentum [Sutskever et al. 2013] helps these optimisers handle tricky non-linear, multivariate curvatures such as ravines. The optimisers’ effectiveness is greatly reduced if the noise in the gradients overpowers the variance arising from non-linearities in the estimated moments required for these mechanisms.

Naively feeding the meta-estimated gradients to Adam is problematic; Adam computes its moment estimates, assuming the input gradients in each iteration to be independent. Meanwhile, our meta-estimator outputs an already averaged gradient (Equation (7)) with a strong positive correlation to previous averages. Adam’s moment estimates are also redundant since we already estimate the variance of our meta-estimator $\text{Var}[\langle F_i \rangle_M]$. Therefore, we formulate the update step in terms of our estimates:

$$\Delta \pi_{i+1} = -\eta \frac{\langle F_i \rangle_M}{\sqrt{\text{Var}[\langle F_i \rangle_M] + \epsilon}}. \quad (21)$$

Dividing by $\sqrt{\text{Var}[\langle F_i \rangle_M]}$ sets the step size based on our meta-estimator. As $\text{Var}[\langle F_i \rangle_M]$ responds to changes in the estimated gradients much more quickly than Adam’s second-moment estimate with the suggested $\beta_2 = 0.999$ parameter, the stability of our method may seem uncertain. We observe that the responsivity of our method actually improves convergence, especially when combined with the decoupled estimation of $\text{Var}[\langle \Delta F_i \rangle]$. Optimisation speeds up quickly when low-noise gradients are available and slows down naturally when approaching a minimum.

6 EXPERIMENTS

We run several experiments to confirm our method’s behaviour and verify its theory. We also compare our method against Adam, as it is used in state-of-the-art inverse rendering pipelines. We implement our method in Mitsuba 3 [Jakob et al. 2022] and use Path Replay Backpropagation [Vicini et al. 2021] to sample gradients computed with the unbiased Mean Relative Squared Error loss [Deng et al. 2022; Pidhorskyi et al. 2022]. For texture optimisation tasks, we use gradient preconditioning as proposed by Nicolet et al. [2021]. We compute $\langle \Delta F_i \rangle$ with a simplified form of the shift mapping proposed by Kettunen et al. [2015], only accounting for the BRDF sampling. While this implementation is sufficient for our proof-of-concept demonstrations, a full implementation of shift mapping can also account for changes in geometry at an insignificant cost compared to proportional samples. Unless mentioned otherwise, we tune learning rates of each method in each experiment.

Variance reduction without lag. We investigate the variance reduction our method can achieve while the scene parameters are changing. We run a fixed linear interpolation of the parameters without an optimiser to prevent any effects from the feedback of the gradients.

Forward gradients of several pixels in Figure 3 show that our method avoids the lag in gradients typical of EMAs. Our meta-estimate’s actual variances and estimate variances are much tighter than the estimates computed by Adam. Furthermore, our method remains more stable upon encountering outliers.

Approximation accuracy. We repeat the previous setup in Figure 4, only now we test an exponentially decaying change in the gradients. Again, our meta-estimator stays within 0.5 to 2 times its predicted standard deviation. As the gradients settle, our method provides a consistent variance reduction (Row 1), averaging a large number of samples wherever possible. Meanwhile, Adam struggles with high-variance gradients (Row 2) and is thrown off by outliers.

We also show the approximated variances compared to ground truth variances computed over 1000 independent runs. Our approximation methods perform reasonably, only overestimating $\text{Var}[\langle F_i \rangle]$ (Row 3). This overestimation results in generally conservative α_i values, erring on the side of robustness rather than maximising variance reduction (Row 5). On the other hand, we approximate $\text{Var}[\langle \Delta F_i \rangle]$ (Row 4) with little bias, although with often a large run-to-run variance.

Multivariate optimisation. We simultaneously optimise an object’s colour, metalness, and roughness, as shown in Figure 1. Thanks to our meta-estimator, our method can traverse the loss surface without losing past samples. Furthermore, our finite-difference estimates let our meta-estimator adjust rapidly, avoiding the overshoots typical of Momentum-based methods. Even when tuning Adam’s hyperparameters for the specific problem, it can only match our method at an over 20 times increase in computational cost, not counting the time spent on hyperparameter tuning.

Texture optimisation. We show a difficult texture optimisation case in Figure 6. Texture optimisation requires disentangling global illumination with very few gradient samples per texel. Adam can only take a few steps within a fixed budget at a high sample count,

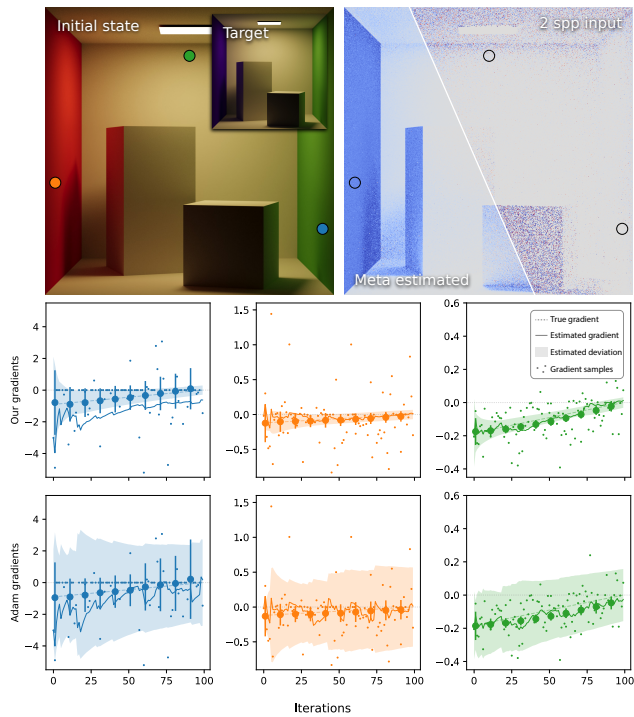


Figure 3: We estimate forward gradients of the left wall’s colour’s blue channel while linearly changing the scene from the initial state (top left) to the target state over 100 iterations. The dashed line represents the actual gradient, dots the gradient samples, the solid line the estimated gradient, and the shaded area the estimated standard deviation. Error bars every 10 iterations show the run-to-run variation of the estimated gradient. Meta-estimation eliminates lag, improves robustness to outliers, and offers lower variance while more accurately estimating this variance. We select the three pixels w.r.t. the actual gradient variance; blue is the noisiest, orange is at 75’th percentile, and green is the median.

requiring a high learning rate that skips over the intricate loss surface necessary to navigate for disentangling various effects. At a lower sample count, however, Adam struggles to progress as steps devolve into a random walk as the scale of the gradients shrinks close to minima.

High-dimensional optimisation. In Figure 7, we optimise an emissive-absorptive volume of size $256 \times 256 \times 256$ voxels, totalling 70 million parameters. Perfectly fitting such a non-physical volume to rendered images is impossible. Thus, the optimiser needs to balance per-pixel losses for a good approximation, further needing to disentangle the small subset of parameters visible through any given pixel. Previous works avoid convergence to local minima by upsampling the optimised volume in several stages; our method does not need this workaround. On the other hand, Figure 8 shows that our method provides less benefit when our finite-difference estimator’s sampling is too sparse across the volume.

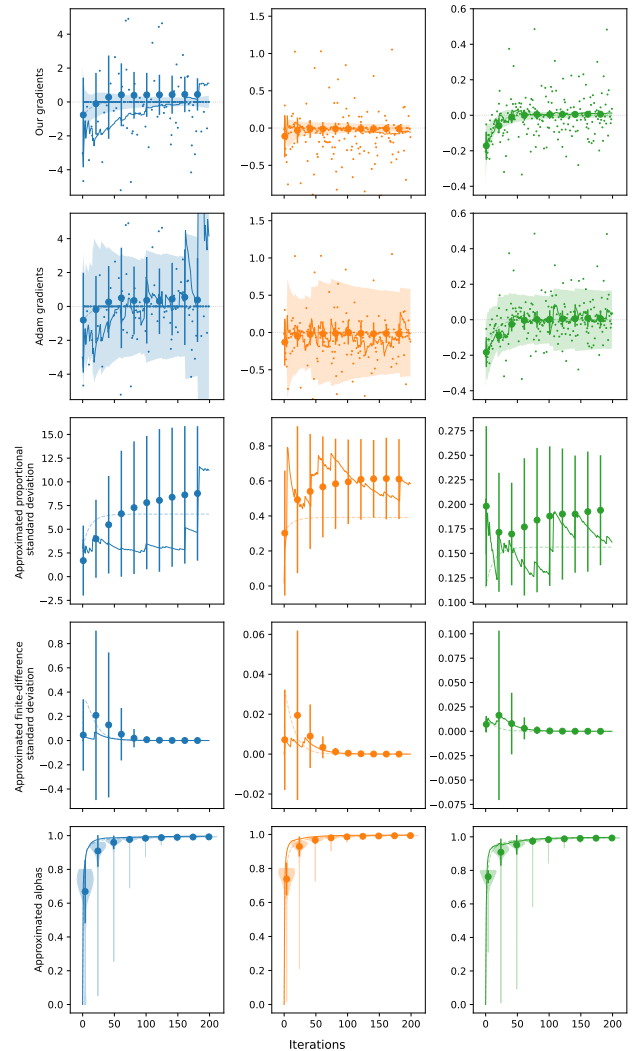


Figure 4: Following the same setup as Figure 3, we show the estimated gradients and standard deviations of our meta-estimator and Adam. Our meta-estimators achieve significantly lower actual variance in each case, while also providing much more accurate approximations. Dashed lines represent actual or optimal values, solid lines a randomly sampled run, while error bars show run-to-run variation. In addition, we show violin plots for alpha, demonstrating how our method is more likely to be conservative and not to be swayed by outliers.

Zero-centred EMAs. We chose to use zero-centred moving averages so that we do not need to approximate the mean of our estimators directly. This approach is generally more robust and memory efficient, though it overestimates variance at large signal-to-noise ratios. However, gradients generally have a low signal-to-noise ratio, so this tradeoff works in our favour. Figure 5 (top-left) demonstrates how non-zero-centred variance approximation is unstable, providing unreliable alphas, thus causing optimisation to diverge.



Figure 5: We ablate our design choices, demonstrating their importance for robust gradient estimation and optimisation. Each modification of our method is less robust than our results in Figure 6.

Alpha clipping. Alpha clipping helps resolve cases when our approximated variances are inaccurate, improving robustness. It may hinder the variance reduction of our method in the special case when $\text{Var}[\langle F_i \rangle]$ is sharply decreasing over iterations. However, we have not encountered this behaviour with our tested proportional estimators. Figure 5 (bottom-left) shows an ablation without alpha clipping for a scene from Figure 6, demonstrating rapid divergence as the initial variance approximations are unreliable.

Sample reuse. We use the same samples for rendering and variance approximation. This correlation introduces some bias at the start of the optimisation process, which diminishes over time. Figure 5 (top-right) shows an unbiased ablation using uncorrelated samples. Although this independently approximated variance eliminates bias, it misses outliers in the samples used for gradient estimation, causing the parameters receiving these outliers to diverge.

7 LIMITATIONS

Estimators $\langle \Delta F_i \rangle$ are not generally available for many problems. Kettunen et al. [2015] propose shift mapping for path tracing, which we use in our work. Our meta-estimators rely heavily on $\langle \Delta F_i \rangle$; as we recurrently sum $\text{Var}[\langle \Delta F_i \rangle]$ in Equation (15), it inherently bounds the variance of our meta-estimator. Doing so is fine as long as $\text{Var}[\langle \Delta F_i \rangle]$ is quadratic w.r.t. the step size (Equation (14)). Thus, we need to ensure this property when building finite-difference estimators while also aiming for the lowest variance to achieve the best stability and convergence with meta-estimation.

Zeltner et al. [2021] show that gradient estimators benefit from specialised differential sampling strategies. The same is true of

finite-difference estimators; our naive toy formulation in Equation (5) glosses over this problem where $p(\mathbf{x}, \pi_i)$ is usually only optimised by importance sampling $f(\mathbf{x}, \pi_i)$, not the difference between $f(\mathbf{x}, \pi_i)$ and $f(\mathbf{x}, \pi_{i-1})$.

Suboptimal sampling strategies of $\langle F_i \rangle$ compound the issue. As our work focuses on gradient estimation, meaning F_i are gradients, sampling of $\langle F_i \rangle$ is not yet well established. For example, Zeltner et al. [2021] show the poor performance of roughness gradient estimators. We experience these issues first-hand, as we show in Figure 9.

8 CONCLUSION

Our proposed meta-estimation technique and corresponding adaptation of the Adam update rule can substantially improve convergence when descending on noisy gradients, reducing computation costs by several orders of magnitude. We solve cases where low-sample-count gradients are too noisy for fast convergence while high-sample-count gradients are prohibitively expensive to compute for the required number of iterations on difficult non-linear, multivariate problems.

Future work. We look forward to applications of meta-estimation to various inverse Monte Carlo problems, especially as MC gradient estimators become prominent in machine learning [Mohamed et al. 2020]. Building good gradient and finite-difference estimators may seem challenging — and are the main limitation of our method — but it is undoubtedly a fruitful direction for future work. We did not investigate training deep neural networks in this work but see it as the next step once low-variance finite-difference estimators become available.

ACKNOWLEDGMENTS

This work is supported by an academic gift from Meta. We thank the anonymous reviewers for their valuable feedback.

REFERENCES

- Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased Warped-Area Sampling for Differentiable Rendering. *ACM Trans. Graph.* 39, 6, Article 245 (nov 2020), 18 pages. <https://doi.org/10.1145/3414685.3417833>
- Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020). <https://doi.org/10/gg8xc7>
- Wesley Chang, Venkataram Sivaram, Derek Nowrouzezahrai, Toshiya Hachisuka, Ravi Ramamoorthi, and Tzu-Mao Li. 2023. Parameter-Space ReSTIR for Differentiable and Inverse Rendering. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) (SIGGRAPH '23). Association for Computing Machinery, New York, NY, USA, Article 18, 10 pages. <https://doi.org/10.1145/3588432.3591512>
- Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. 2014. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/ede7e2b6d13a41ddf9f4bdef84fdc737-Paper.pdf
- Xi Deng, Fujun Luan, Bruce Walter, Kavita Bala, and Steve Marschner. 2022. Reconstructing Translucent Objects Using Differentiable Rendering. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) (SIGGRAPH '22). Association for Computing Machinery, New York, NY, USA, Article 38, 10 pages. <https://doi.org/10.1145/3528233.3530714>
- EC Fieller and HO Hartley. 1954. Sampling with control variables. *Biometrika* 41, 3/4 (1954), 494–501.
- Robert M. Gower, Mark Schmidt, Francis Bach, and Peter Richtárik. 2020. Variance-Reduced Methods for Machine Learning. *Proc. IEEE* 108, 11 (2020), 1968–1983. <https://doi.org/10.1109/JPROC.2020.3028013>

- Alex Graves. 2014. Generating Sequences With Recurrent Neural Networks. arXiv:1308.0850 [cs.NE]
- Homan Igehy. 1999. Tracing Ray Differentials. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 179–186. <https://doi.org/10.1145/311535.311555>
- Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. 2022. *Mitsuba 3 renderer*. <https://mitsuba-renderer.org>.
- Rie Johnson and Tong Zhang. 2013. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In *Advances in Neural Information Processing Systems*, C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2013/file/ac1dd209cbcc5e5d1c6e28598e8cbb8-Paper.pdf
- James T Kajiya. 1986. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 143–150.
- R. E. Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* 82, 1 (03 1960), 35–45. <https://doi.org/10.1115/1.3662552> arXiv:https://asmédigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf
- Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-Domain Path Tracing. *ACM Trans. Graph.* 34, 4, Article 123 (jul 2015), 13 pages. <https://doi.org/10.1145/2766997>
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-Domain Metropolis Light Transport. 32, 4, Article 95 (jul 2013), 12 pages. <https://doi.org/10.1145/2461912.2461943>
- Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Trans. Graph.* 37, 6, Article 222 (dec 2018), 11 pages. <https://doi.org/10.1145/3272127.3275109>
- Marco Manzi, Markus Kettunen, Frédo Durand, Matthias Zwicker, and Jaakko Lehtinen. 2016. Temporal Gradient-Domain Path Tracing. *ACM Trans. Graph.* 35, 6, Article 246 (dec 2016), 9 pages. <https://doi.org/10.1145/2980179.2980256>
- Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. 2020. Monte Carlo Gradient Estimation in Machine Learning. *J. Mach. Learn. Res.* 21, 1, Article 132 (jan 2020), 62 pages.
- Eric Moulines and Francis Bach. 2011. Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning. In *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (Eds.), Vol. 24. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2011/file/40008b9a5380fcacce3976bf7c08af5b-Paper.pdf
- Yurii Evgen'evich Nesterov. 1983. A method of solving a convex programming problem with convergence rate $O(\frac{1}{k^2})$. In *Doklady Akademii Nauk*, Vol. 269. Russian Academy of Sciences, 543–547.
- Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. 2021. Large Steps in Inverse Rendering of Geometry. *ACM Trans. Graph.* 40, 6, Article 248 (dec 2021), 13 pages. <https://doi.org/10.1145/3478513.3480501>
- Baptiste Nicolet, Fabrice Rousselle, Jan Novák, Alexander Keller, Wenzel Jakob, and Thomas Müller. 2023. Recursive Control Variates for Inverse Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 42, 4 (Aug. 2023). <https://doi.org/10.1145/3592139>
- Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020). <https://doi.org/10.1145/3386569.3392406>
- Art B Owen. 2013. Monte Carlo theory, methods and examples. (2013).
- Athanasios Papoulis and S Unnikrishna Pillai. 1984. *Probability, random variables, and stochastic processes*. McGraw-Hill, 145–149.
- Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation (3rd ed.)* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Stanislav Pidhorskyi, Timur Bagautdinov, Shugao Ma, Jason Saragih, Gabriel Schwartz, Yaser Sheikh, and Tomas Simon. 2022. Depth of Field Aware Differentiable Rendering. 41, 6, Article 190 (nov 2022), 18 pages. <https://doi.org/10.1145/3550454.3555521>
- Boris T Polyak and Anatoli B Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization* 30, 4 (1992), 838–855.
- Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-Space Control Variates for Rendering. *ACM Trans. Graph.* 35, 6, Article 169 (dec 2016), 12 pages. <https://doi.org/10.1145/2980179.2982443>
- Bimal K Sinha, Joachim Hartung, and Guido Knapp. 2011. *Statistical meta-analysis with applications*. John Wiley & Sons.
- Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. 2018. Don't Decay the Learning Rate, Increase the Batch Size. arXiv:1711.00489 [cs.LG]
- Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*. PMLR, 1139–1147.
- Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021), 108:1–108:14. <https://doi.org/10.1145/3450626.3459804>
- Kai Yan, Christoph Lassner, Brian Budge, Zhao Dong, and Shuang Zhao. 2022. Efficient Estimation of Boundary Integrals for Path-Space Differentiable Rendering. *ACM Trans. Graph.* 41, 4, Article 123 (jul 2022), 13 pages. <https://doi.org/10.1145/3528223.3530080>
- Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo Estimators for Differential Light Transport. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021). <https://doi.org/10.1145/3450626.3459807>
- Cheng Zhang, Zhao Dong, Michael Doggett, and Shuang Zhao. 2021. Antithetic Sampling for Monte Carlo Differentiable Rendering. *ACM Trans. Graph.* 40, 4 (2021), 77:1–77:12.
- Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-Space Differentiable Rendering. *ACM Trans. Graph.* 39, 4, Article 143 (aug 2020), 19 pages. <https://doi.org/10.1145/3386569.3392383>
- Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. 2019. A Differential Theory of Radiative Transfer. *ACM Trans. Graph.* 38, 6, Article 227 (nov 2019), 16 pages. <https://doi.org/10.1145/3355089.3356522>

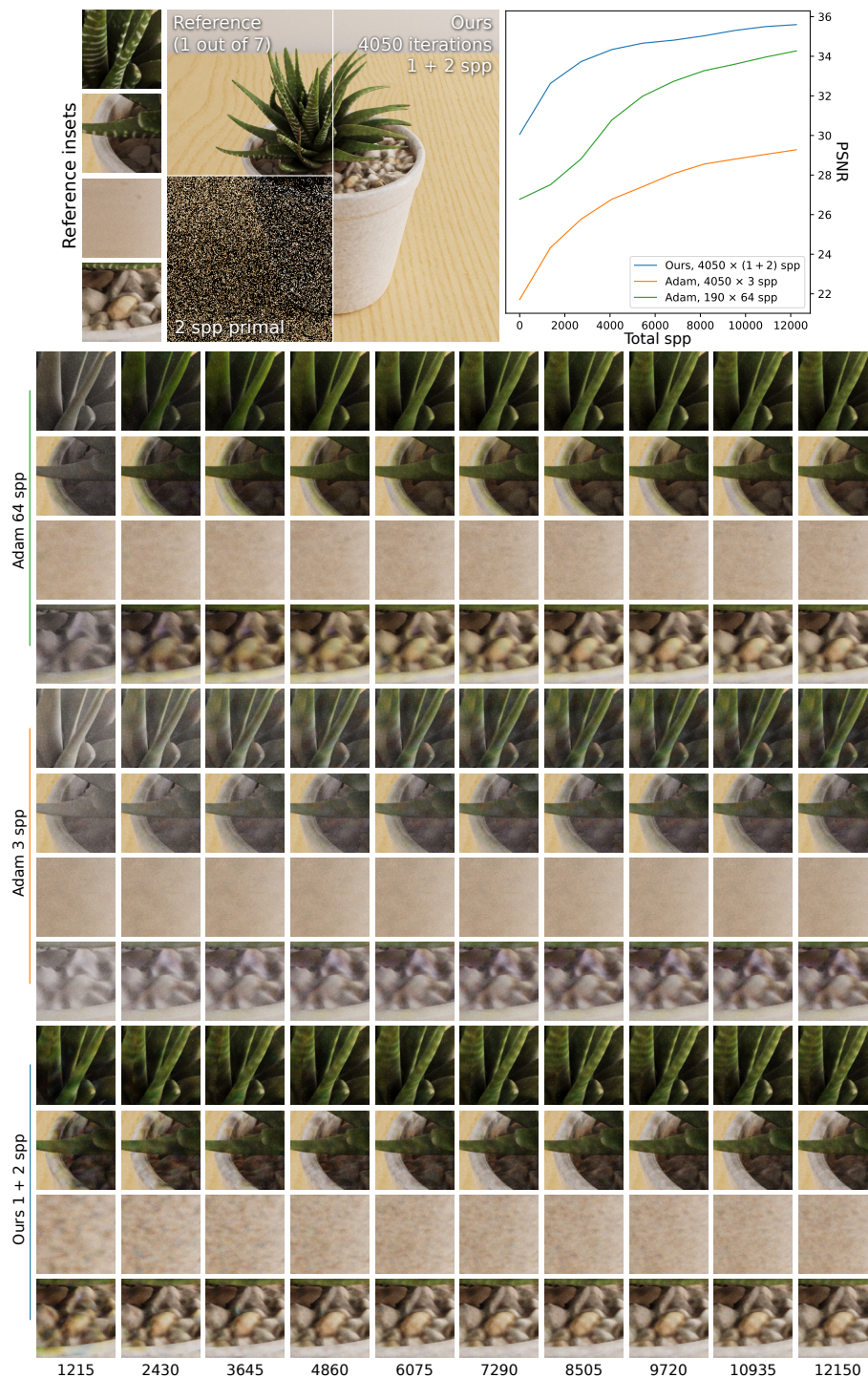


Figure 6: We optimise diffuse textures, placing a potted plant in the extremely noisy Veach Ajar scene. At 64 spp, Adam can only afford to take a few steps at a large learning rate and cannot match small details, further failing on complex interactions such as the leaf’s reflection on the pot’s side, where navigating a difficult loss surface is necessary. At 3 spp, Adam reduces to a random walk after reaching vaguely accurate parameters where gradients vanish against the high noise level. Our estimator shows good convergence. Textures remain a little blurry as a known consequence of regularisation [Nicolet et al. 2021]. The texture of the glossy pot (third inset) shows relatively worse convergence due to the concerns discussed in Section 7.

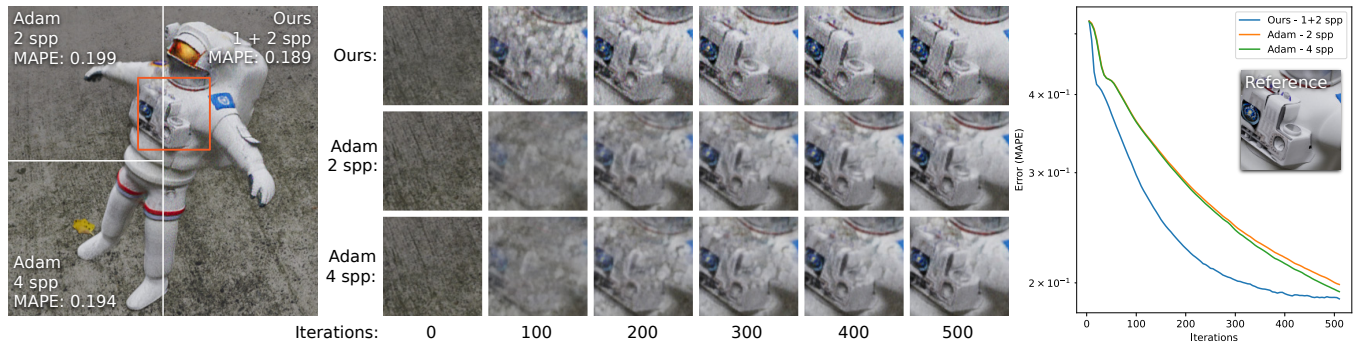


Figure 7: We optimise emission and absorption volumes in a NeRF-like experiment. Regarding computational cost, our method lies between 2 spp and 4 spp Adam at one finite-difference and two proportional samples. However, we greatly outperform both regarding convergence and final quality. Although the final quantitative difference is smaller, we resolve jarring artefacts such as holes in the volume and overall blur.

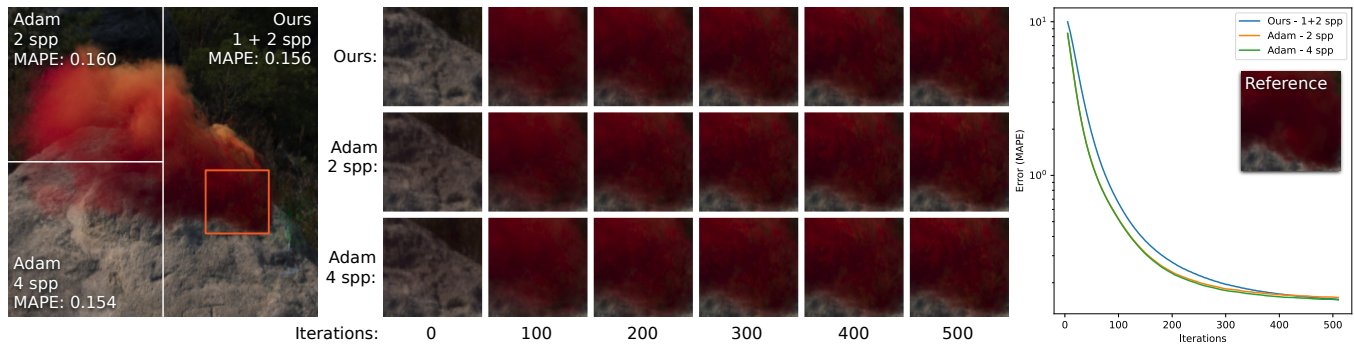


Figure 8: Same setup as Figure 7. Our method performs worse here as the volume is sparser, concentrating our finite-difference samples less. Thus, parameters at individual voxels are samples relatively sparsely, making finite-difference estimation for the whole volume difficult.

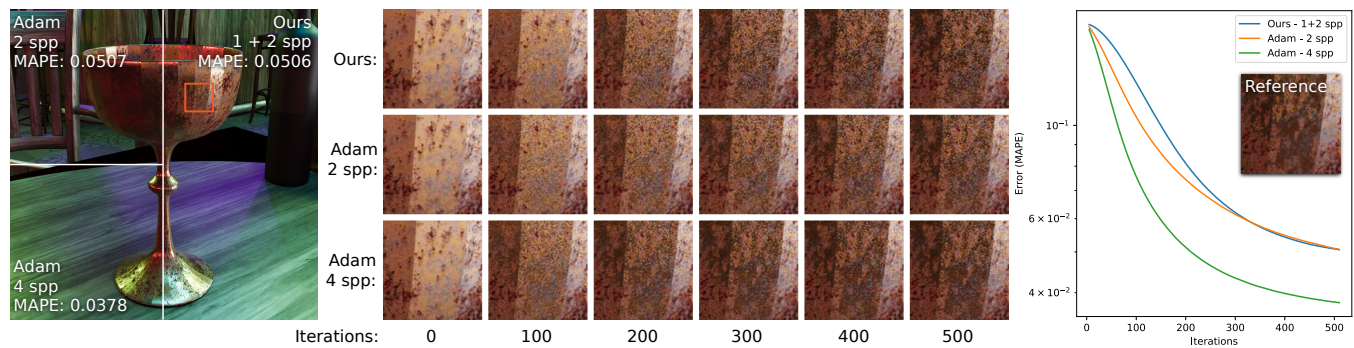


Figure 9: We optimise the 2K roughness texture of the Chalice scene from Chang et al. [2023]. Using only basic shift mapping, our finite-difference estimator struggles to cope with the changes in roughness over sharp lighting. As the roughness parameter influences our sampling strategy, this is a challenging case for finite-difference estimation.