

Decorrelating ReSTIR Samplers via MCMC Mutations - Supplemental

ROHAN SAWHNEY, Carnegie Mellon University and NVIDIA, USA

DAQI LIN, NVIDIA, USA

MARKUS KETTUNEN, NVIDIA, Finland

BENEDIKT BITTERLI, NVIDIA, USA

RAVI RAMAMOORTHY, NVIDIA and UC San Diego, USA

CHRIS WYMAN, NVIDIA, USA

MATT PHARR, NVIDIA, USA

CCS Concepts: • **Computing methodologies** → **Ray Tracing**.

Additional Key Words and Phrases: real-time rendering, resampled importance sampling, weighted reservoir sampling, Markov chain Monte Carlo

ACM Reference Format:

Rohan Sawhney, Daqi Lin, Markus Kettunen, Benedikt Bitterli, Ravi Ramamoorthi, Chris Wyman, and Matt Pharr. 2023. Decorrelating ReSTIR Samplers via MCMC Mutations - Supplemental. *ACM Trans. Graph.* 1, 1, Article 1 (October 2023), 3 pages. <https://doi.org/10.1145/3629166>

1 WHY MUTATIONS DECREASE COVARIANCE

We present an argument for why mutating samples after temporal reuse reduces covariance in the final image. We study the problem in a simplified setting, which we believe captures the essence.

In particular, we study the covariance between two pixels with target functions \hat{p}^1 and \hat{p}^2 , with pixels 1 and 2 spatially reusing samples from random neighbors. We model random neighbor selection as follows: the input samples for pixel $i \in \{1, 2\}$ are X_j^i where j ranges from 1 to M , and their contribution weights are W_j^i . Superscripts denote the target pixel and subscripts the index of its j 'th input; input samples for the pixels are assumed to be distinct. As in ReSTIR DI, we assume samples lie in the same domain Ω and share the support (i.e., agree on visibility), with shift mappings not needed for reuse between pixels (i.e., the identity shift mapping is used with light vertices as they are). We also use constant MIS weights $1/M$. With this setup, we show a reduction in covariance in the limit case as the number of mutations approaches infinity (assuming good importance sampling), and argue that benefits in the finite case arise from approximating the limit case.

We interpret the input samples X_j^i as the results of temporal resampling, and denote the final mutation results before spatial reuse Y_j^i and their contribution weights W_j^i . The assumption of good importance sampling means that $W_j^i \approx 1/\hat{p}_j^i(X_j^i)$, where \hat{p}_j^i is the

Authors' addresses: Rohan Sawhney, rsawhney@nvidia.com, Carnegie Mellon University and NVIDIA, USA; Daqi Lin, daqil@nvidia.com, NVIDIA, USA; Markus Kettunen, mktettunen@nvidia.com, NVIDIA, Finland; Benedikt Bitterli, bbitterli@nvidia.com, NVIDIA, USA; Ravi Ramamoorthi, ravir@cs.ucsd.edu, NVIDIA and UC San Diego, USA; Chris Wyman, cwyman@nvidia.com, NVIDIA, USA; Matt Pharr, mpharr@nvidia.com, NVIDIA, USA.

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).

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

0730-0301/2023/10-ART1

<https://doi.org/10.1145/3629166>

Table 1. Summary of notation

| | |
|----------------------------|---|
| X_j^i, W_j^i | Pixel i 's original input sample j and its contribution weight |
| Y_j^i, W_j^i | Pixel i 's mutated input sample j and its contribution weight |
| Z^i, W^i | Sample chosen for pixel i from the Y_j^i and its contribution weight |
| w_j^i | Resampling weight for choosing Y_j^i as the new Z^i |
| \hat{p}^i, \hat{p}^i | Target function of pixel i and its normalized target PDF |
| \hat{p}_j^i, \hat{p}_j^i | Target function of pixel i 's input j and its target PDF |
| f_j^i | The integrand in pixel i , here $f^i = \hat{p}^i$ |
| Z_j^i, W_j^i | Sample chosen for pixel i from the X_j^i and its contribution weight |
| w_j^i | Resampling weight for choosing X_j^i as the new Z^i |
| $\ \cdot\ $ | The 1-norm, $\ g\ = \int_{\Omega} g(x) dx$, e.g., $\bar{p} = \hat{p}/\ \hat{p}\ $ |

normalized version of the target function \hat{p}_j^i , i.e., $\bar{p}_j^i = \hat{p}_j^i/\|\hat{p}_j^i\|$. We again denote the target pixel in the superscript and the index of its j 'th input pixel in the subscript. Assuming a large number of mutations implies that we can treat the mutation results Y_j^i as independent of each other, the input samples X_j^i , and their contribution weights W_j^i . The contribution weight of a mutated sample Y_j^i is

$$W_j^i = \frac{\hat{p}_j^i(X_j^i)}{\hat{p}_j^i(Y_j^i)} W_j^i, \quad (1)$$

while the resampling weight for choosing sample Y_j^i for pixel i is

$$w_j^i = \frac{1}{M} \hat{p}^i(Y_j^i) W_j^i \quad (2)$$

$$= \frac{1}{M} \frac{\hat{p}^i(Y_j^i)}{\hat{p}_j^i(Y_j^i)} \hat{p}_j^i(X_j^i) W_j^i. \quad (3)$$

The sample Z^i selected for pixel i by resampling proportionally to w_j^i has the contribution weight

$$W^i = \frac{1}{\hat{p}^i(Z^i)} \sum_{j=1}^M w_j^i \quad (4)$$

$$= \frac{1}{\hat{p}^i(Z^i)} \sum_{j=1}^M \frac{1}{M} \frac{\hat{p}^i(Y_j^i)}{\hat{p}_j^i(Y_j^i)} \hat{p}_j^i(X_j^i) W_j^i. \quad (5)$$

Using the integrand of the rendering equation f^i as our target function \hat{p}^i , the pixel estimate $f^i(Z^i)W^i$ is

$$\hat{p}^i(Z^i)W^i = \frac{1}{M} \sum_{j=1}^M \frac{\hat{p}^i(Y_j^i)}{\hat{p}_j^i(Y_j^i)} \hat{p}_j^i(X_j^i) W_j^i, \quad (6)$$

and the covariance between estimates for pixels 1 and 2 becomes

$$\begin{aligned} & \text{Cov}\left(\hat{p}^1(Z^1)W^1, \hat{p}^2(Z^1)W^2\right) \\ &= \frac{1}{M^2} \sum_{j=1}^M \sum_{k=1}^M \text{Cov}\left(\frac{\hat{p}^1(Y_j^1)}{\hat{p}_j^1(Y_j^1)} \hat{p}_j^1(X_j^1)W_j^1, \frac{\hat{p}^2(Y_k^2)}{\hat{p}_k^2(Y_k^2)} \hat{p}_k^2(X_k^2)W_k^2\right). \end{aligned} \quad (7)$$

Since we study the limit case in which the mutated samples Y_j^i are independent of other random variables, we can rewrite this expression using the relation $\text{Cov}(XY, Z) = \mathbb{E}[X] \text{Cov}(Y, Z)$, which assumes that X is independent of Y and Z . Applying this relation to both parameters of the covariance yields

$$= \frac{1}{M^2} \sum_{j=1}^M \sum_{k=1}^M \mathbb{E}\left[\frac{\hat{p}^1(Y_j^1)}{\hat{p}_j^1(Y_j^1)}\right] \mathbb{E}\left[\frac{\hat{p}^2(Y_k^2)}{\hat{p}_k^2(Y_k^2)}\right] \text{Cov}\left(\hat{p}_j^1(X_j^1)W_j^1, \hat{p}_k^2(X_k^2)W_k^2\right). \quad (8)$$

We now simplify the expression above by first writing $\mathbb{E}\left[\frac{\hat{p}^i(Y_j^i)}{\hat{p}_j^i(Y_j^i)}\right]$ as $\frac{\|\hat{p}^i\|}{\|\hat{p}_j^i\|} \mathbb{E}\left[\frac{\hat{p}^i(Y_j^i)}{\hat{p}_j^i(Y_j^i)}\right]$, where we move the norms of the target functions outside the expectation. Since we study the limit case of near-infinite mutations, Y_j^i has PDF $\bar{p}_j^i(Y_j^i)$, and $\mathbb{E}\left[\frac{\hat{p}^i(Y_j^i)}{\hat{p}_j^i(Y_j^i)}\right] = \int_{\Omega} \bar{p}^i(y) dy = 1$, which simplifies the expression for covariance to

$$= \frac{1}{M^2} \sum_{j=1}^M \sum_{k=1}^M \frac{\|\hat{p}^1\| \|\hat{p}^2\|}{\|\hat{p}_j^1\| \|\hat{p}_k^2\|} \text{Cov}\left(\hat{p}_j^1(X_j^1)W_j^1, \hat{p}_k^2(X_k^2)W_k^2\right). \quad (9)$$

Here the norms of \hat{p}_j^1 and \hat{p}_k^2 normalize \hat{p} into \bar{p} inside the covariance, yielding

$$= \frac{\|\hat{p}^1\| \|\hat{p}^2\|}{M^2} \sum_{j=1}^M \sum_{k=1}^M \text{Cov}\left(\bar{p}_j^1(X_j^1)W_j^1, \bar{p}_k^2(X_k^2)W_k^2\right). \quad (10)$$

Finally, we use the definition of covariance, $\text{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$, and the relation $\mathbb{E}[\bar{p}_j^i(X_j^i)W_j^i] = \int_{\Omega} \bar{p}_j^i(x) dx = 1$ (by the way unbiased contribution weights transform expectations into integrals) to evaluate μ_X and μ_Y and reach the final form for the covariance of the pixel estimates,

$$= \frac{\|\hat{p}^1\| \|\hat{p}^2\|}{M^2} \sum_{j=1}^M \sum_{k=1}^M \mathbb{E}\left[(\bar{p}_j^1(X_j^1)W_j^1 - 1)(\bar{p}_k^2(X_k^2)W_k^2 - 1)\right]. \quad (11)$$

This final expression for the covariance shows that when the input samples X_j^i are importance sampled well at their original pixels, i.e., $W_j^i \approx 1/\bar{p}_j^i(X_j^i)$, then both factors in the expectation tend to be small, yielding a small covariance as well.

We now consider the case without mutations, deriving covariance when \tilde{Z}^1 and \tilde{Z}^2 are resampled directly from the samples X_j^i without mutations. The resampling weights are

$$\tilde{w}_j^i = \frac{1}{M} \hat{p}^i(X_j^i)W_j^i, \quad (12)$$

while the chosen sample \tilde{Z}^i has contribution weight

$$\tilde{W}^i = \frac{1}{\hat{p}^i(\tilde{Z}^i)} \sum_{j=1}^M \tilde{w}_j^i = \frac{1}{\hat{p}^i(\tilde{Z}^i)} \sum_{j=1}^M \frac{1}{M} \hat{p}^i(X_j^i)W_j^i. \quad (13)$$

We again set $\hat{p}^i = f^i$, which yields the pixel contribution

$$\hat{p}^i(\tilde{Z}^i)\tilde{W}^i = \frac{1}{M} \sum_{j=1}^M \hat{p}^i(X_j^i)W_j^i. \quad (14)$$

The pixel covariance then is

$$\begin{aligned} & \text{Cov}\left(\hat{p}^1(\tilde{Z}^1)\tilde{W}^1, \hat{p}^2(\tilde{Z}^2)\tilde{W}^2\right) \\ &= \frac{1}{M^2} \sum_{j=1}^M \sum_{k=1}^M \text{Cov}\left(\hat{p}^1(X_j^1)W_j^1, \hat{p}^2(X_k^2)W_k^2\right), \end{aligned} \quad (15)$$

which we simplify to

$$= \frac{\|\hat{p}^1\| \|\hat{p}^2\|}{M^2} \sum_{j=1}^M \sum_{k=1}^M \text{Cov}\left(\bar{p}^1(X_j^1)W_j^1, \bar{p}^2(X_k^2)W_k^2\right). \quad (16)$$

As before, we finally express covariance with expectations,

$$= \frac{\|\hat{p}^1\| \|\hat{p}^2\|}{M^2} \sum_{j=1}^M \sum_{k=1}^M \mathbb{E}\left[(\bar{p}^1(X_j^1)W_j^1 - 1)(\bar{p}^2(X_k^2)W_k^2 - 1)\right]. \quad (17)$$

We immediately observe the critical difference between Equation 11 and Equation 17: the inputs X_j^i are in both cases distributed approximately proportionally to \bar{p}_j^i , where with good importance sampling we have $W_j^i \approx 1/\bar{p}_j^i(X_j^i)$; without mutations, the expressions inside the expectations approximately equal

$$\left(\frac{\bar{p}^1(X_j^1)}{\bar{p}_j^1(X_j^1)} - 1\right) \left(\frac{\bar{p}^2(X_k^2)}{\bar{p}_k^2(X_k^2)} - 1\right), \quad (18)$$

which have no guarantees of being small if the target functions of the pixels and their inputs are not similar. In fact, these ratios may be arbitrarily large. However, by applying many mutations, these expressions typically become much smaller, yielding

$$\left(\frac{\bar{p}_j^1(X_j^1)}{\bar{p}_j^1(X_j^1)} - 1\right) \left(\frac{\bar{p}_k^2(X_k^2)}{\bar{p}_k^2(X_k^2)} - 1\right) \approx 0. \quad (19)$$

This implies that mutations help bring covariance closer to zero even when the target functions are different.

We have demonstrated that with well-distributed input samples, a large number of mutations help decorrelate the inputs to reservoir sampling during spatial reuse, effectively making input paths independent. The resulting independence of the mutated paths suggests that covariance results merely from bad importance sampling of the original pixels, not from incompatibility between close-by pixels.

Pixels whose covariance is minimized should share as few input samples as possible—the proof requires a complete separation. In practice, we only apply a small number of mutations, which does not completely decorrelate the input samples. We also randomize the input pixels for each target pixel, leading to a small probability of overlap between input samples. While we cannot realize the ideal in practice, we aim for it as much as possible—every mutation is a step closer to the limit, and heuristically we expect some decrease in covariance with a smaller number of mutations.

Covariance does not go to zero simply by increasing the number of mutations. We are still bounded by random overlaps between

input pixels and the quality of importance sampling as seen in Equation 11.

In practice, proper MIS weights also protect the renderer from the most terrible correlations and can be used to guarantee eventual convergence (see Lin et al. [2022]). A performance-optimized implementation that greedily neglects MIS weights is however much more prone to high covariance—this analysis predicts that mutations are especially effective at removing correlation artifacts in such use cases. It also predicts that covariance will be present especially between close-by pixels with very different target functions. ReSTIR implementations (such as ours) try to defend against such cases by using expensive MIS weights and/or careful neighbor selection; this analysis, together with our empirical results, suggests that mutations are a useful addition for further robustness.

2 WHY MUTATIONS DO NOT DECREASE VARIANCE (MUCH)

Variance can be studied as a pixel’s covariance with itself, $\text{Var}(X) = \text{Cov}(X, X)$. We do not prove that variance reduction cannot happen when mutations are used—in some cases, it can. However, as we noted earlier, the two pixels’ input samples should be different to minimize covariance. This is not true in the case of variance: a pixel, tautologically, has the same input samples as itself, and the mechanism to reduce covariance does not apply to variance—the corresponding inputs have 100% correlation. This is in line with our empirical findings: mutations have little impact on variance.

REFERENCES

Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuskel, and Chris Wyman. 2022. Generalized Resampled Importance Sampling: Foundations of ReSTIR. *ACM Transactions on Graphics (TOG)* 41, 75.