Blue Noise Distributed MCMC Decorrelation of ReSTIR

By Oscar Fickel

Supervised by Dr. ir. Peter Vangorp

Blue Noise Distributed MCMC Decorrelation of ReSTIR

Ray tracing: computing physically based lighting.

ReSTIR: algorithm for reusing previously computed lighting per pixel.

MCMC decorrelation: slightly adjusting and diversifying the reused computations.

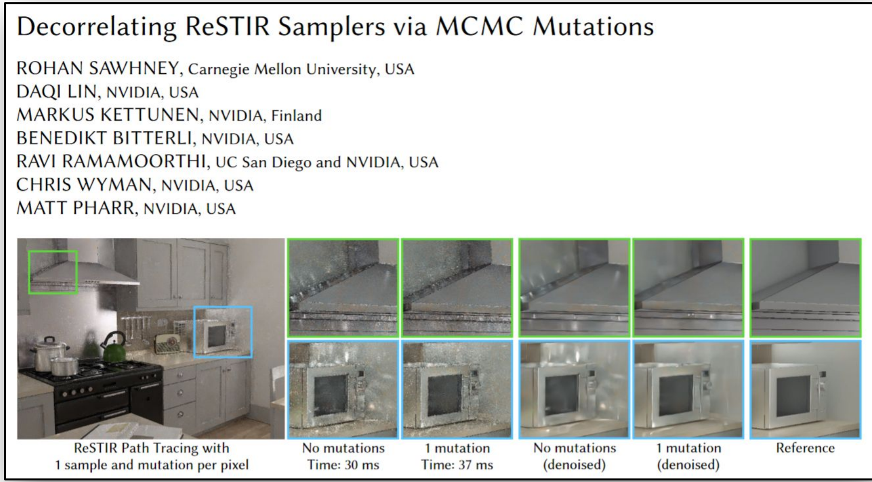
Blue noise: more visually pleasing noise/error in images.

Goal

Reproduce MCMC mutations for ReSTIR.

Adjust mutations to get blue noise.

[1]



Decorrelating ReSTIR Samplers via MCMC Mutations

ROHAN SAWHNEY, Carnegie Mellon University, USA
DAQI LIN, NVIDIA, USA
MARKUS KETTUNEN, NVIDIA, Finland
BENEDIKT BITTERLI, NVIDIA, USA
RAVI RAMAMOORTHI, UC San Diego and NVIDIA, USA
CHRIS WYMAN, NVIDIA, USA
MATT PHARR, NVIDIA, USA

ReSTIR Path Tracing with 1 sample and mutation per pixel | No mutations Time: 30 ms | 1 mutation Time: 37 ms | No mutations (denoised) | 1 mutation (denoised) | Reference

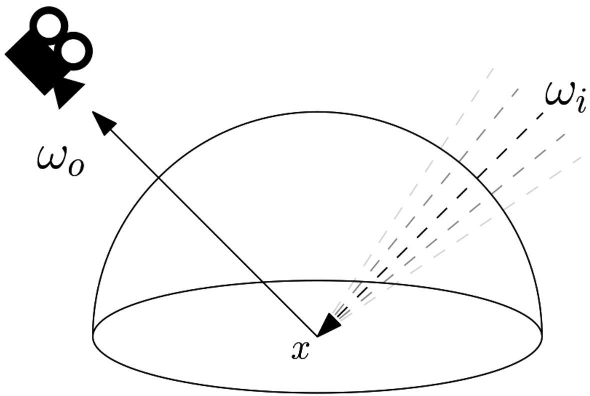
Can blue noise be efficiently incorporated into the decorrelation of ReSTIR algorithms, in such a way that it produces more visually pleasing results using the same number of samples per pixel, and without costing too much additional time?

Content

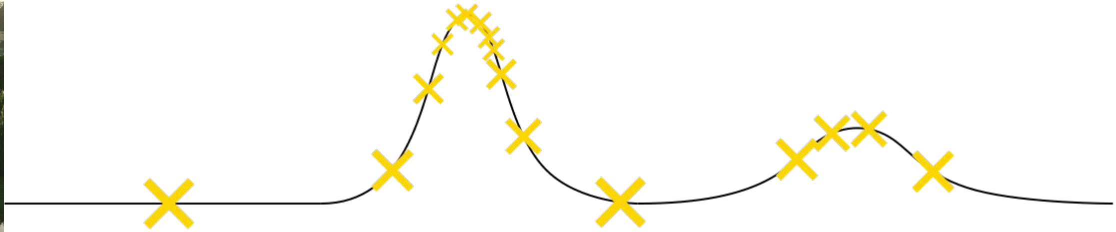
- Background
  - Ray tracing
  - ReSTIR
  - Decorrelating ReSTIR
- (Re)implementation of MCMC decorrelated ReSTIR
- Introducing blue noise
- Evaluation

Ray tracing

Computing intensity of reflected light per pixel.



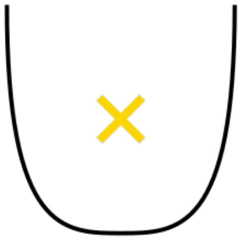
Integrating $\boldsymbol{\omega}_i$ over the hemisphere via sampling



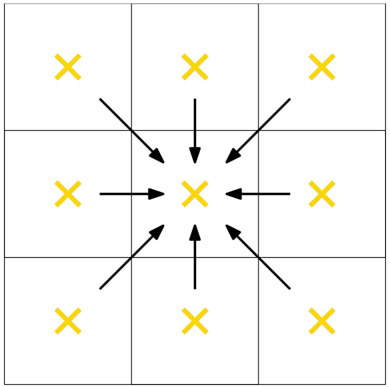
Sampling a using Probability Density Function (PDF)

Spatiotemporal Reservoir Resampling (ReSTIR) [2]

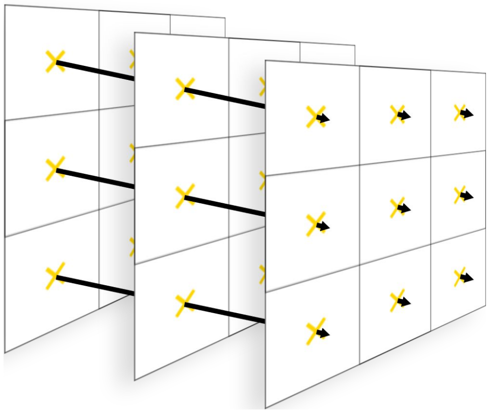
Using reservoirs to efficiently store multiple samples, which get reused from neighbouring pixels.



Weighted reservoir resampling



Spatial resampling



Temporal resampling

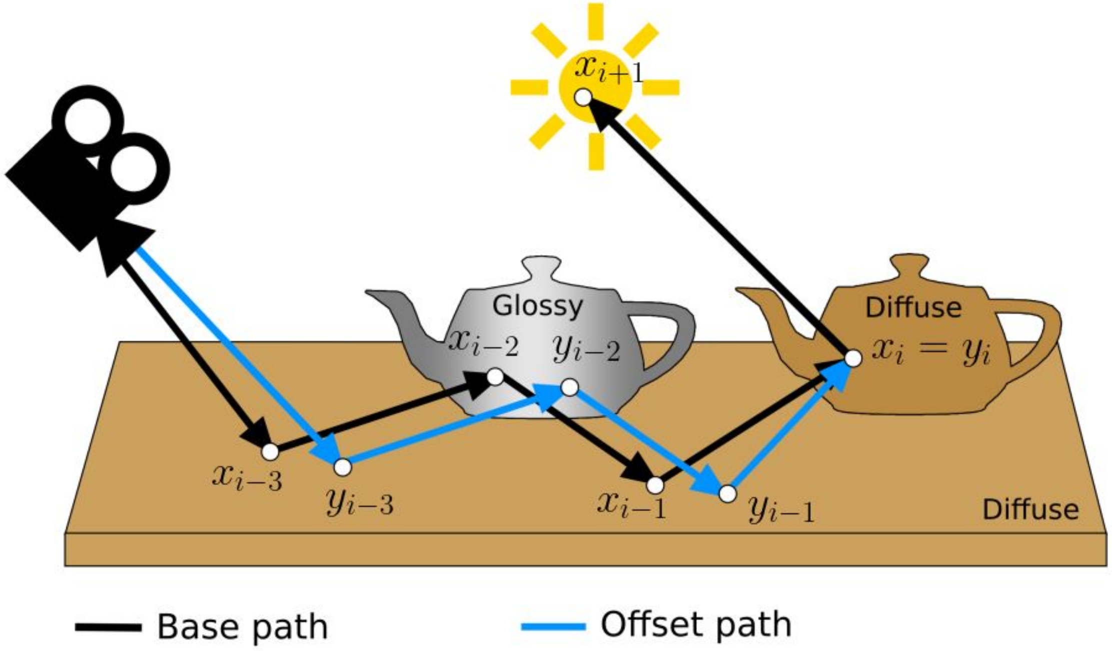
Generalized resampled importance sampling (GRIS) [3]

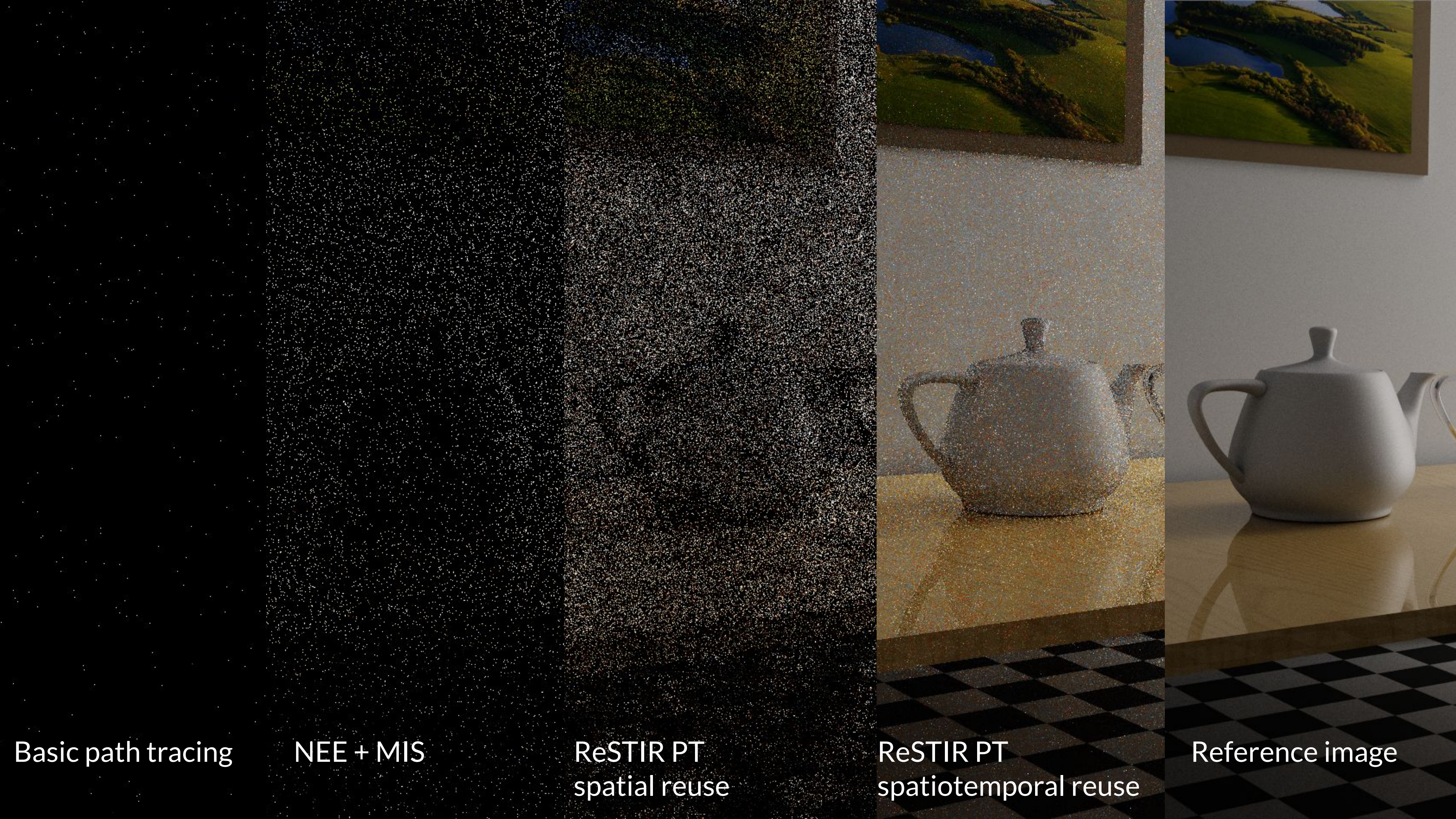
Introduces shift maps to ReSTIR.

Random replay: reuse the random seed.

Reconnection shift: connect new path to old path as soon as possible.

Hybrid shift: combines random replay with reconnection shift.

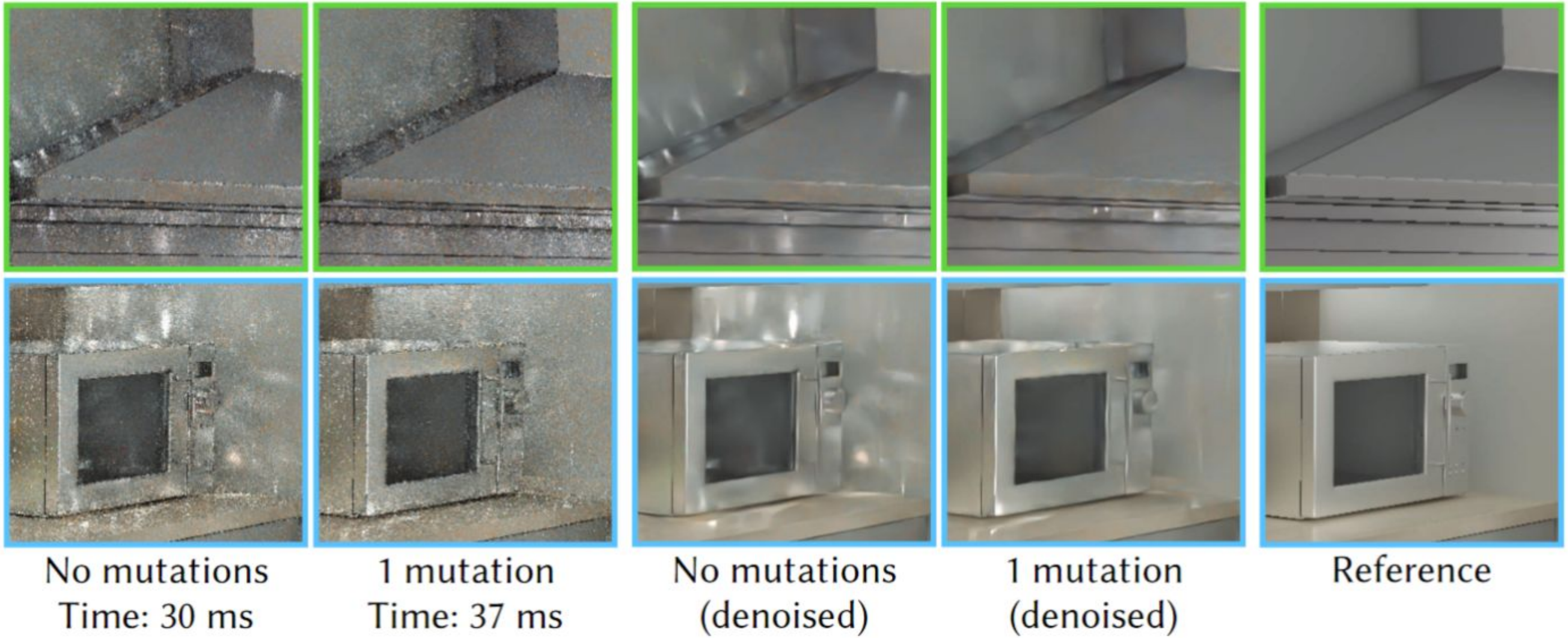



Basic path tracing
NEE + MIS
ReSTIR PT
spatial reuse
ReSTIR PT
spatiotemporal reuse
Reference image

MCMC Decorrelation of ReSTIR

[1]

GRIS is still not perfect, as it suffers from correlation artifacts caused by sample impoverishment and as a result high-contribution samples getting spread out, not averaged out.



Metropolis-Hastings [4]

Used for the MCMC decorrelation.

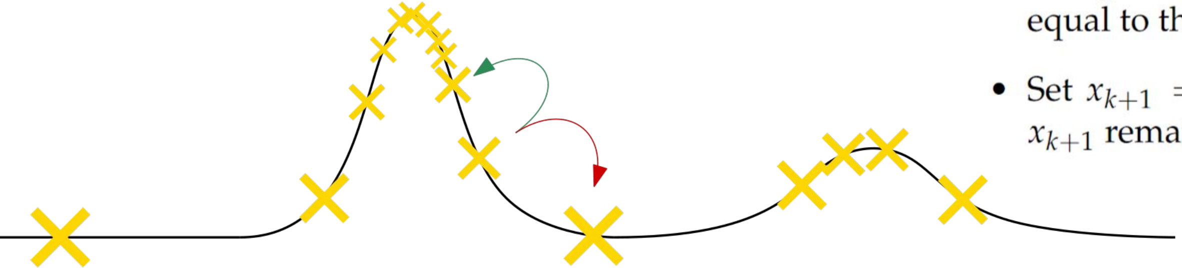
Mutating samples while following a target distribution.

- Generate a candidate sample x'_k from the previous sample, using a proposal distribution $T(x_k \to x'_k)$. Here the transition kernel $T(x_k \to x'_k)$ represents the probability of generating the x'_k sample given the x_k sample.
- Compute the acceptance probability:

$$a(x_k \to x'_k) = \frac{C(x'_k)T(x'_k \to x_k)}{C(x_k)T(x_k \to x'_k)}.$$

In regular MH we set the contribution function C to be equal to the target PDF $\hat{p}$.

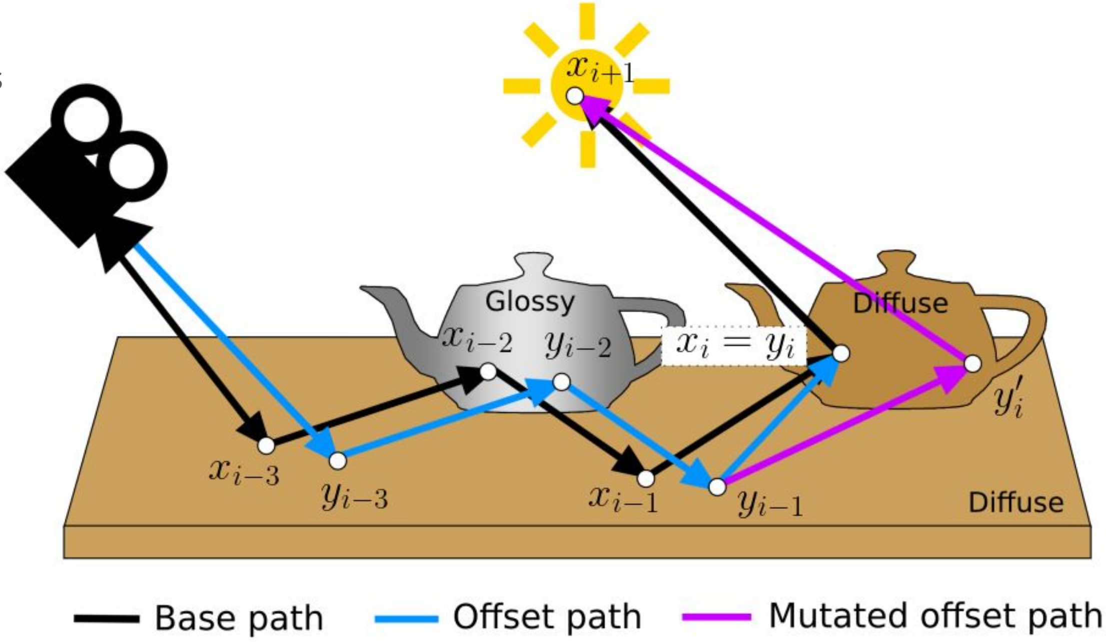
- Set $x_{k+1} = x'_k$ based on the probability a. Otherwise x_{k+1} remains x_k.



Mutating the hybrid shift map [1]

To get the mutated candidate samples used by Metropolis Hastings.

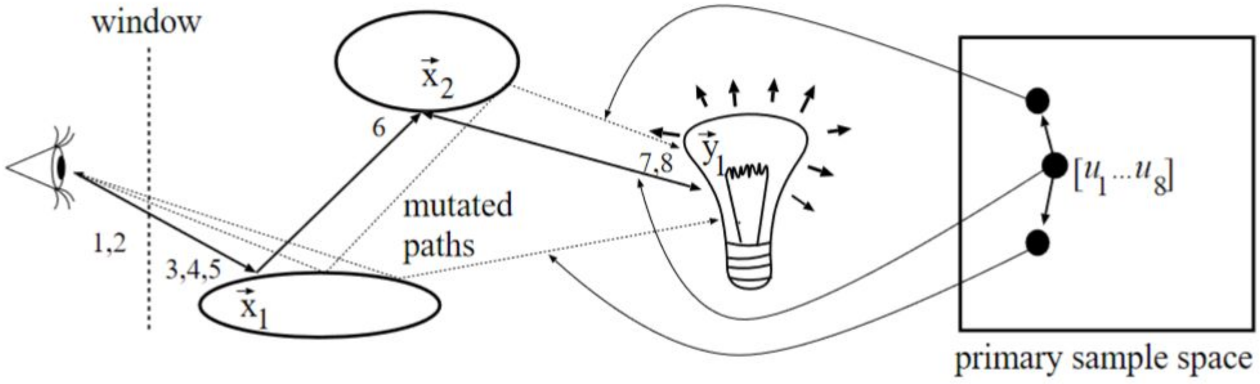
Perturb the reconnection vertex.



Primary sample space (PSS) [5]

To obtain the perturbed reconnection vertex.

Rather than the path itself, we can mutate the vector of random numbers used to generate the path.

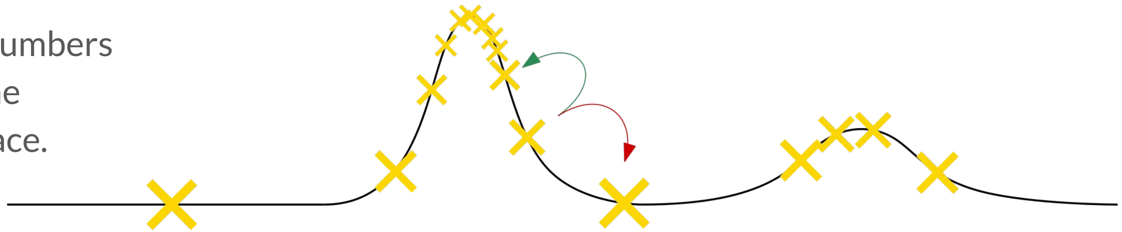


Decorrelating ReSTIR

Contribution function C changes due to use of PSS:

$$C(\bar{u}) := \frac{\hat{p}(\bar{y}(\bar{u}))}{q(\bar{y}(\bar{u}))}$$

A constant size mutation on the random numbers will result in a mutation proportional to the sampling PDF when converted to path space.



Reservoir contribution weight also needs to be updated:

$$W(x_k) = \frac{\hat{p}(x_0)}{\hat{p}(x_k)} W(x_0)$$

Implementation

Build upon implementation of ReSTIR PT in Falcor.

Temporal retrace: retrace random replay up to reconnection vertex.

We put the decorrelation pass between temporal and spatial resampling passes, as it diversifies the samples before they get spread out over the image.

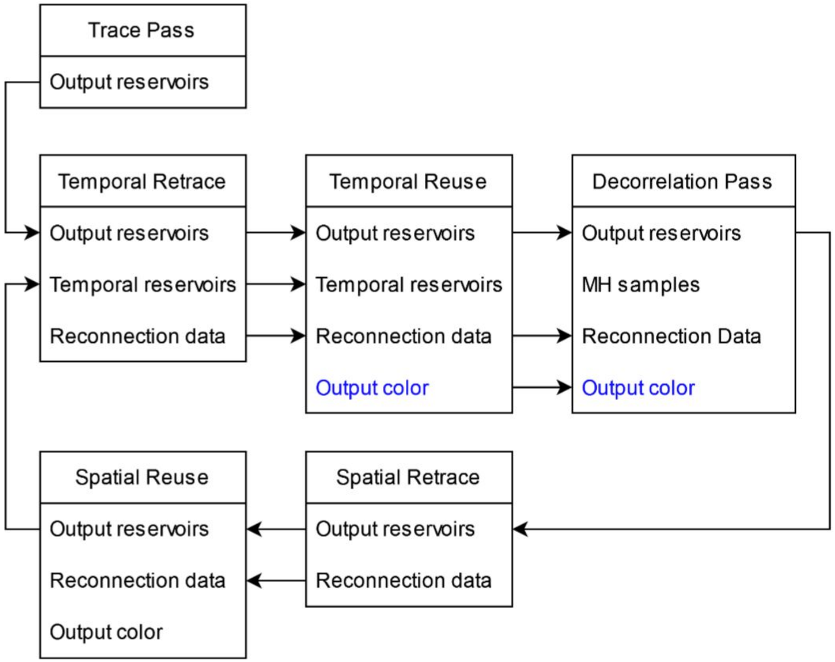


Figure 6: An overview of how the decorrelation pass fits in between the ReSTIR PT passes. The output color in blue is only used for blue noise.

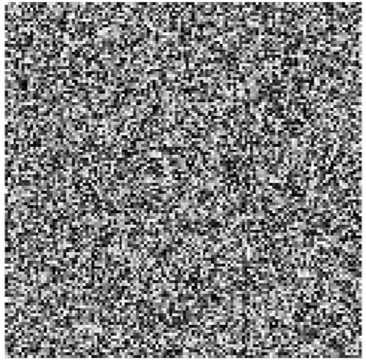
Blue Noise

Only high frequency noise: differing values close together, similar values further apart, which results in less clumps.

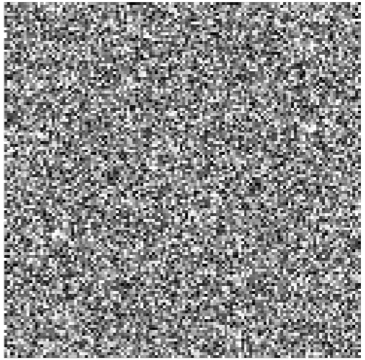
Discrete Fourier Transform is used for evaluation.

Georgiev et al. [6] use a blue noise energy function via simulated annealing to swap pixels and create a sample mask.

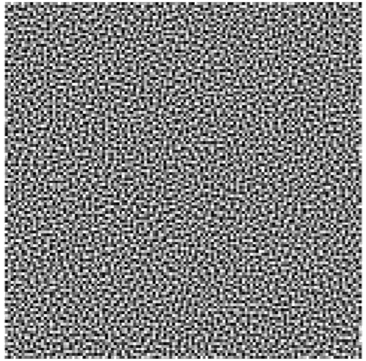
[7]



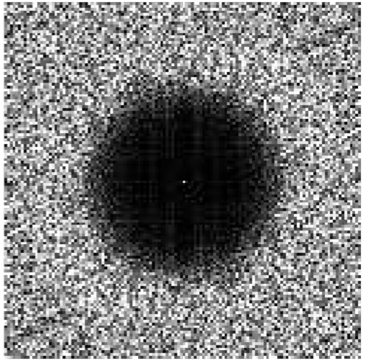
(a) A white noise mask



(b) A DFT of a white noise mask



(c) A blue noise mask



(d) A DFT of a blue noise mask

Blue Noise Mutations

Mutating sample instead of swapping pixels, so we compute energy for number of neighbours.

Blue noise acceptance probability derived from simulated annealing to minimise energy function, but with T already set to near 0.

Replace MH's contribution function ratio with weighted average of blue noise acceptance.

$$\boxed{a_{bn} = \begin{cases} 1, & \text{if } e' < e \\ exp\left(-\frac{(e'-e)}{T}\right), & \text{otherwise} \end{cases}}$$ [8]

$$a(x_k \to x'_k) = \left(a_{bn} w_{bn} + \frac{C(x'_k)}{C(x_k)}(1 - w_{bn})\right) \frac{T(x'_k \to x_k)}{T(x_k \to x'_k)}$$

Evaluation

Rendered 10 frames, each after 50 frames of temporal history.

Root Mean Square Error: how much do the pixel values deviate from the reference image?

Relative pixel covariance: compared to an average image, how much do pixel values deviate in the same way as other pixels in their neighbourhood?

[1]

$$c_{ij} = \frac{1}{K-1} \sum_{m=1}^{K} (I_{mi} - \bar{I}_i)(I_{mj} - \bar{I}_j)$$


ReSTIR
Our Decorrelated ReSTIR
Our Blue Noise
Decorrelated ReSTIR
Reference

Evaluation

Improvement in covariance.

Minimal increase in error.

Scene	RMSE 0 mutations	RMSE 5 mutations	RMSE 5 mutations (blue)
Veach Ajar (ceiling)	0.1019	0.1057	0.1026

Table 1: Measured root-mean-square error.

Scene	Covariance 0 mutations	Covariance 5 mutations	Covariance 5 mutations (blue)	Time (ms) 0 mutations	Time (ms) 5 mutations	Time (ms) 5 mutations (blue)
Veach Ajar (ceiling)	3.5208e-04	3.0106e-04	2.938e-04	17	22	24

Table 2: Measured relative covariance with pixel radius 8, based on ReSTIR PT with a hybrid shift.

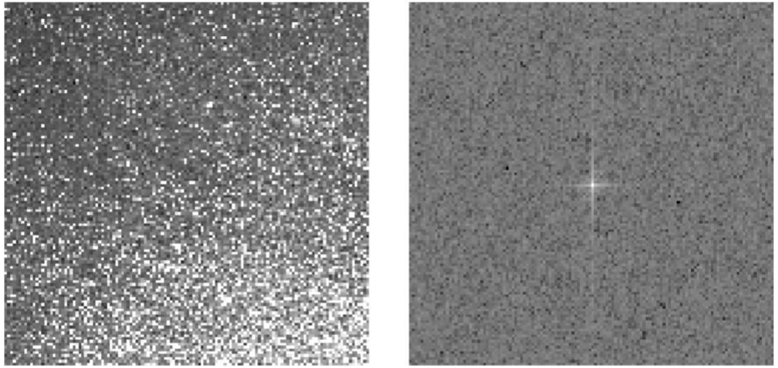
Evaluation - Blue Noise

No visible effect, possible explanations:

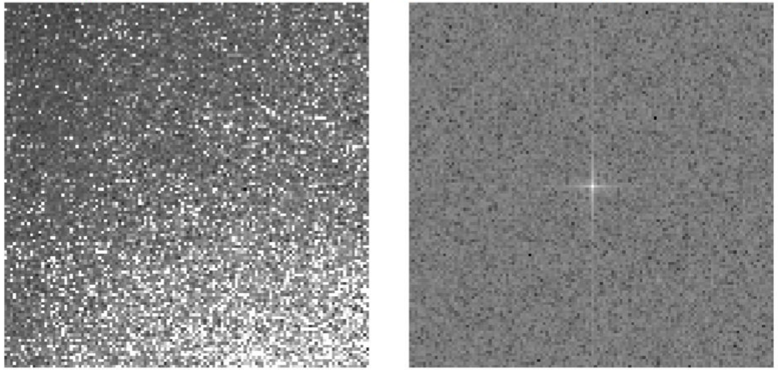
-No pixel luminance updates between mutations.

-Gets cancelled out by spatial resampling pass.

-Mutations don’t decrease error, indicating that they are not key to finding light bringing paths, and thus don’t often increase pixel value, which is not an issue for diversifying, but is for negatively correlating. Since in this case we can only decrease light intensity for pixels in a blue noise pattern, but not increase it for the neighbouring pixels.



(a) Cropped decorrelated ReSTIR (b) DFT of the regular mutations



(c) Cropped blue noise frame (d) DFT of blue noise mutations

Bibliography

1. Sawhney, R., Lin, D., Kettunen, M., Bitterli, B., Ramamoorthi, R., Wyman, C., and Pharr, M. (2022). Decorrelating restir samplers via mcmc mutations.
2. Bitterli, B., Wyman, C., Pharr, M., Shirley, P., Lefohn, A., and Jarosz, W. (2020). Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. ACM Transactions on Graphics (Proceedings of SIGGRAPH), 39(4).
3. Lin, D., Kettunen, M., Bitterli, B., Pantaleoni, J., Yuksel, C., and Wyman, C. (2022). Generalized resampled importance sampling: Foundations of restir. ACM Trans. Graph., 41(4).
4. Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. Biometrika, 57(1):97–109.
5. Kelemen, C., Szirmay-Kalos, L., Antal, G., and Csonka, F. (2002). A simple and robust mutation strategy for the metropolis light transport algorithm. Comput. Graph. Forum, 21.
6. Georgiev, I. and Fajardo, M. (2016). Blue-noise dithered sampling. In ACM SIGGRAPH 2016 Talks, SIGGRAPH '16, New York, NY, USA. Association for Computing Machinery.
7. Chizhov, V., Georgiev, I., Myszkowski, K., and Singh, G. (2022). Perceptual error optimization for monte carlo rendering. ACM Trans. Graph., 41(3).
8. Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220(4598):671–680.