Two-history Approach of Spatiotemporal Filtering for Monte Carlo Rendering: Spatiotemporal Variance-Guided Filtering as Example

Speaker: Chia-Ming Tu (涂家銘)

Advisor：Bing-Yu Chen (陳炳宇博士), Shao-Yi Chien (簡韶逸博士)

May 2024

Slides Link

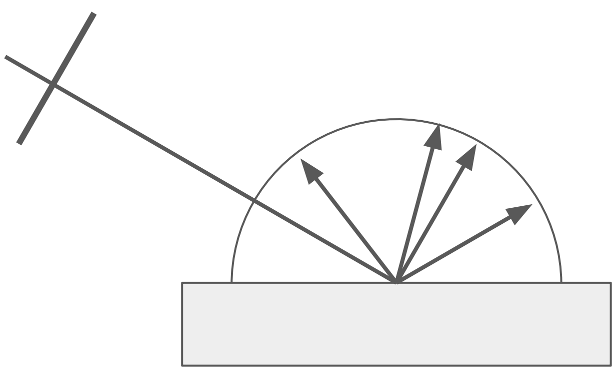
Overview

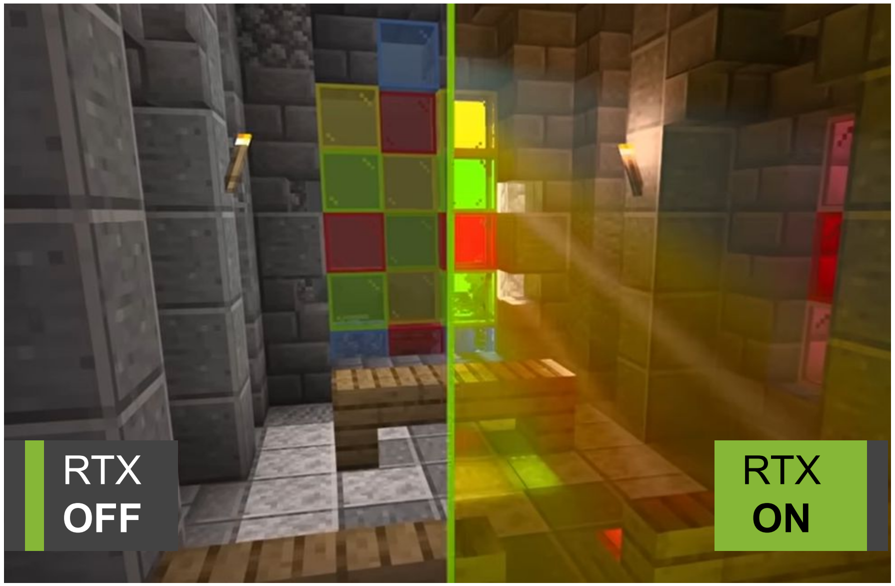
- Introduction
- Related Works
- Proposed Method
- Experiments
- Conclusion
- Limitaions & Future Work

Introduction - Monte Carlo base Path-tracing

Monte Carlo based path-tracing

- Integrate illuminance by sampling random directions
- Realistic results
- **Slow convergence => hard to be realtime**





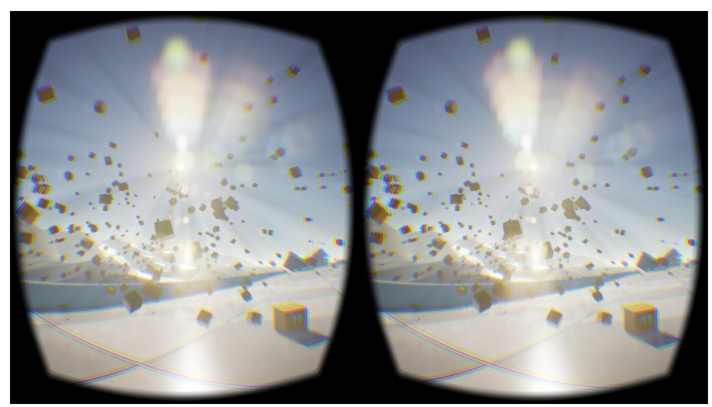
(clipped from video by NVIDIA)

Introduction - Monte Carlo base Path-tracing

For head mounted devices, it's more challenging to achieve realtime.

- Higher resolution is required
- Higher frame rate is required
- Less computing resource compared with desktops

(Image by Evan-Amos. CC0)



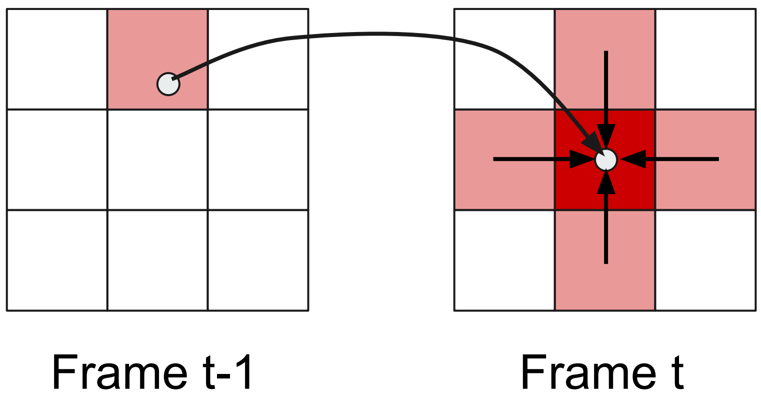
(Image by Ats Kurvet licensed under CC BY 4.0)

Toward Realtime Path-tracing

Common ways to reduce runtime:

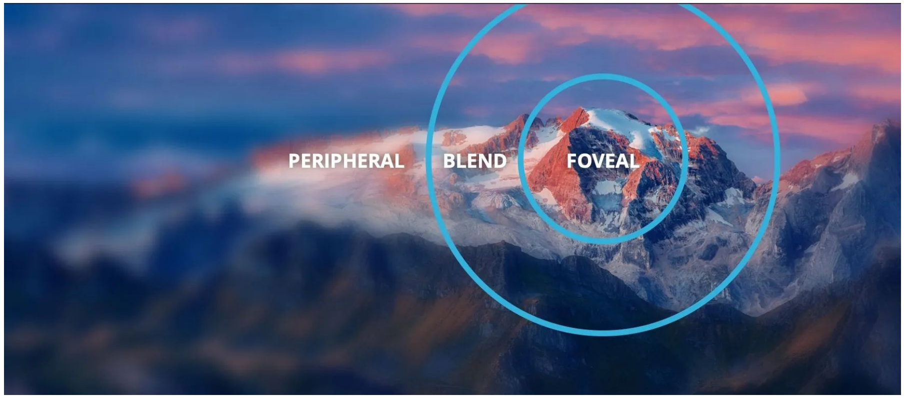
- Increasing effective samples

  e.g. *spatio-temporal filter*



- Distribute more samples to important areas

  e.g. *foveated rendering*

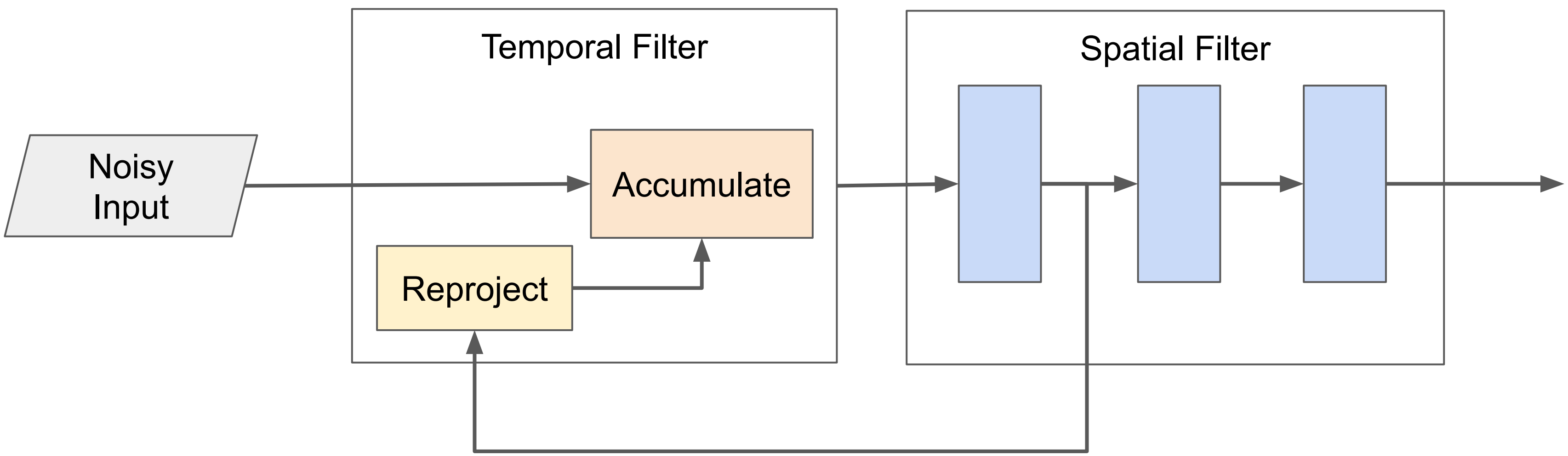


(Image by Tobbi)

Spatio-temporal Filter

Common structure of spatio-temporal filter

- Temporal => Spatial

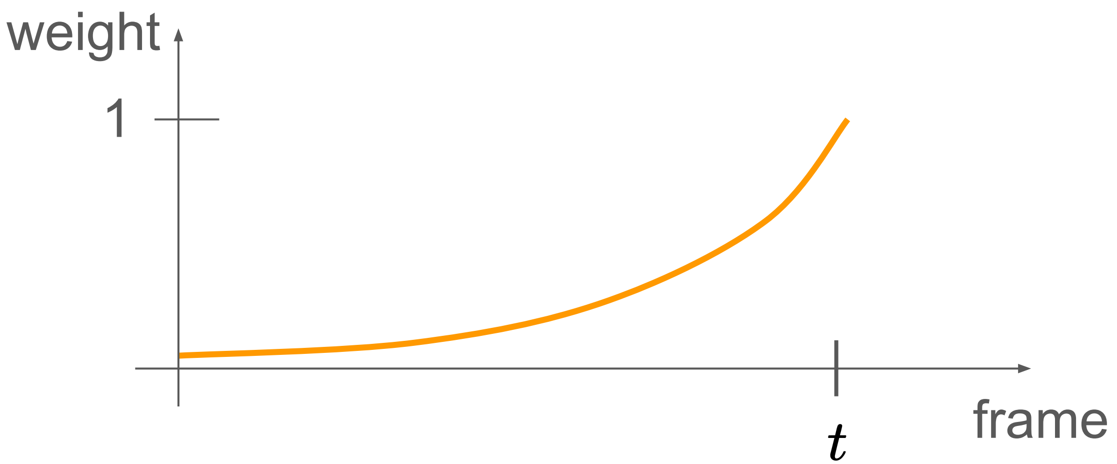


Temporal Filter

- It's common to use exponential moving average in temporal reuse

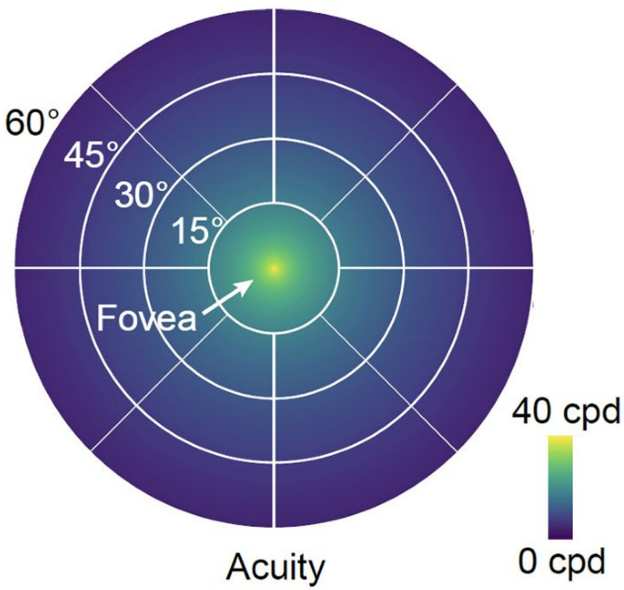
$$x'_t = \beta x'_{t-1} + (1-\beta)x_t = \frac{\sum_{i=1}^{t} \beta^{t-i} x_i}{\sum_{i=1}^{t} \beta^{t-i}}$$

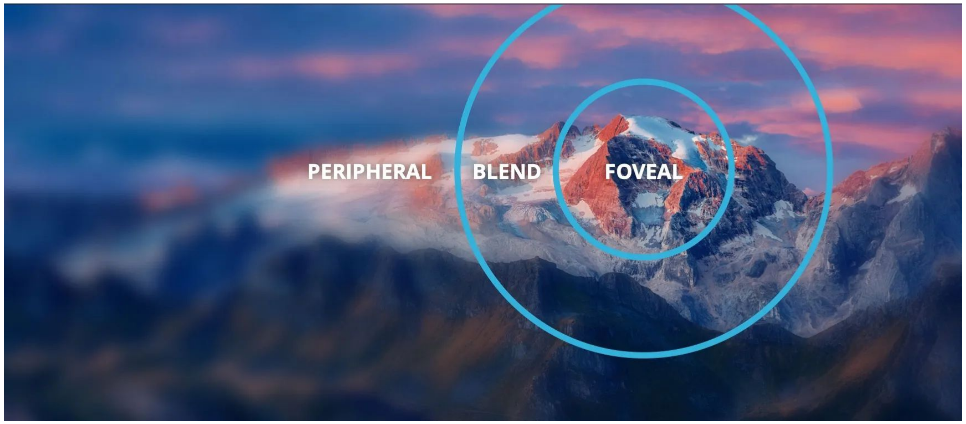
$$\text{where } \beta = \text{decay rate}$$



Foveated Rendering

Foveated rendering exploits properties of human visual system,
renders better quality in foveal and saves computing resource in peripheral.





(Image by Tobbi)

Brooke Krajancich, Petr Kellnhofer, and Gordon Wetzstein. "A perceptual model for eccentricity-dependent spatio-temporal flicker fusion and its applications to foveated graphics". ACM Trans. Graph. 40, 4, Article 47 (August 2021).

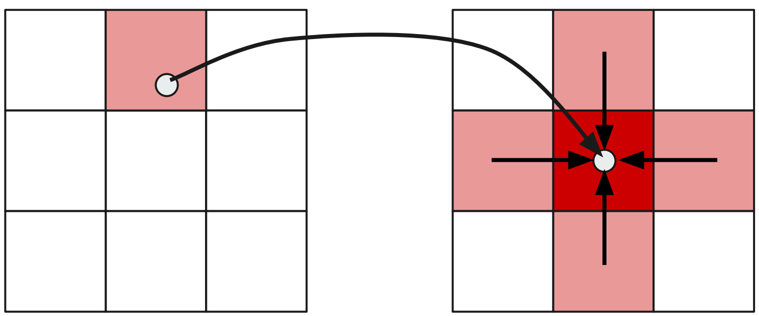
Foveated Rendering

For path-tracing, the easiest way to do foveated rendering is to control number of samples per pixel.

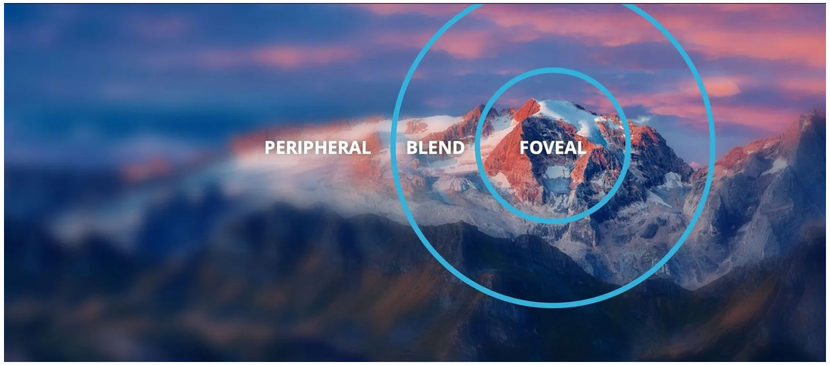
Combining Spatio-temporal Filter and Fovated Rendering

Let try to combine them

Spatio-temproal Filter



Foveated Rendering

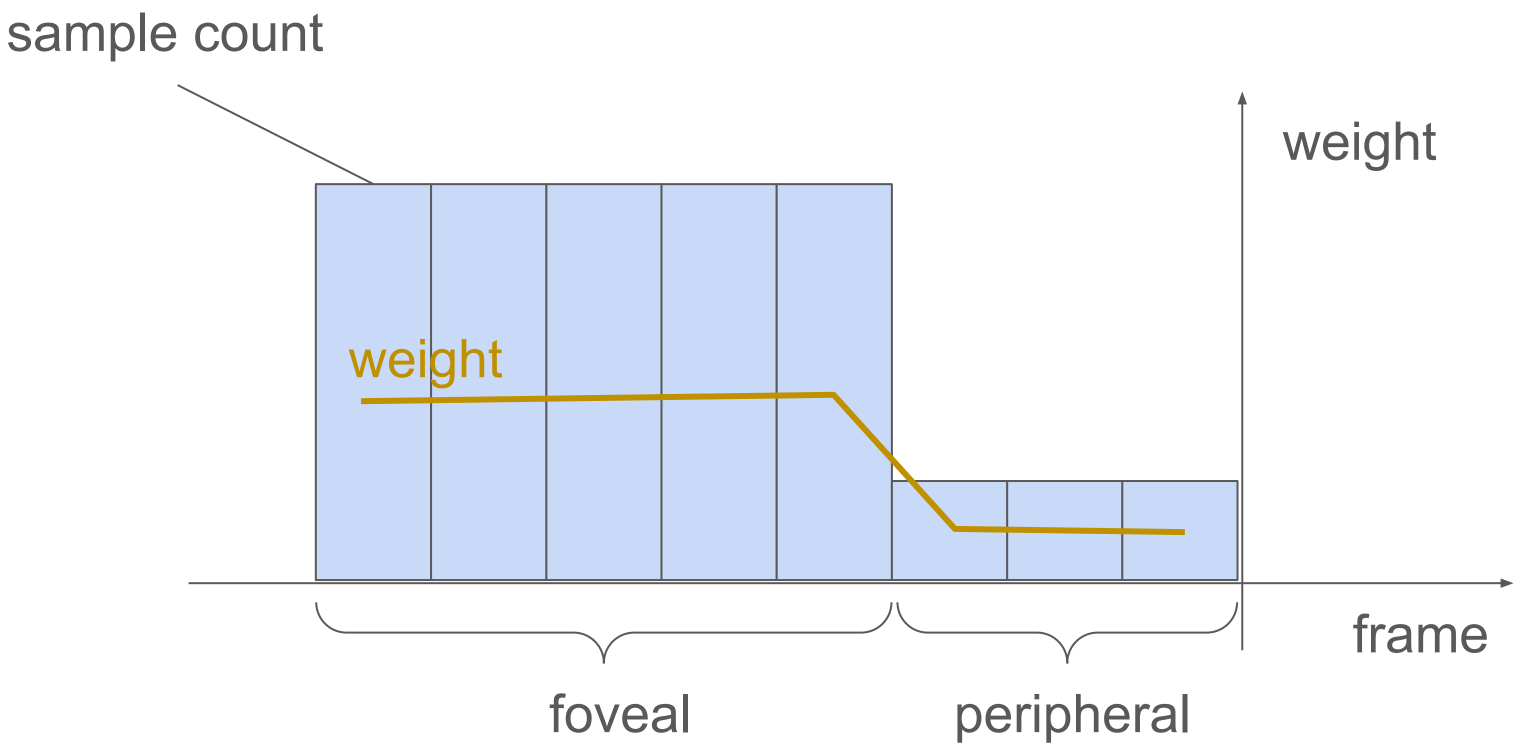


(Image by Tobbi)

Problem of Exponential Decay

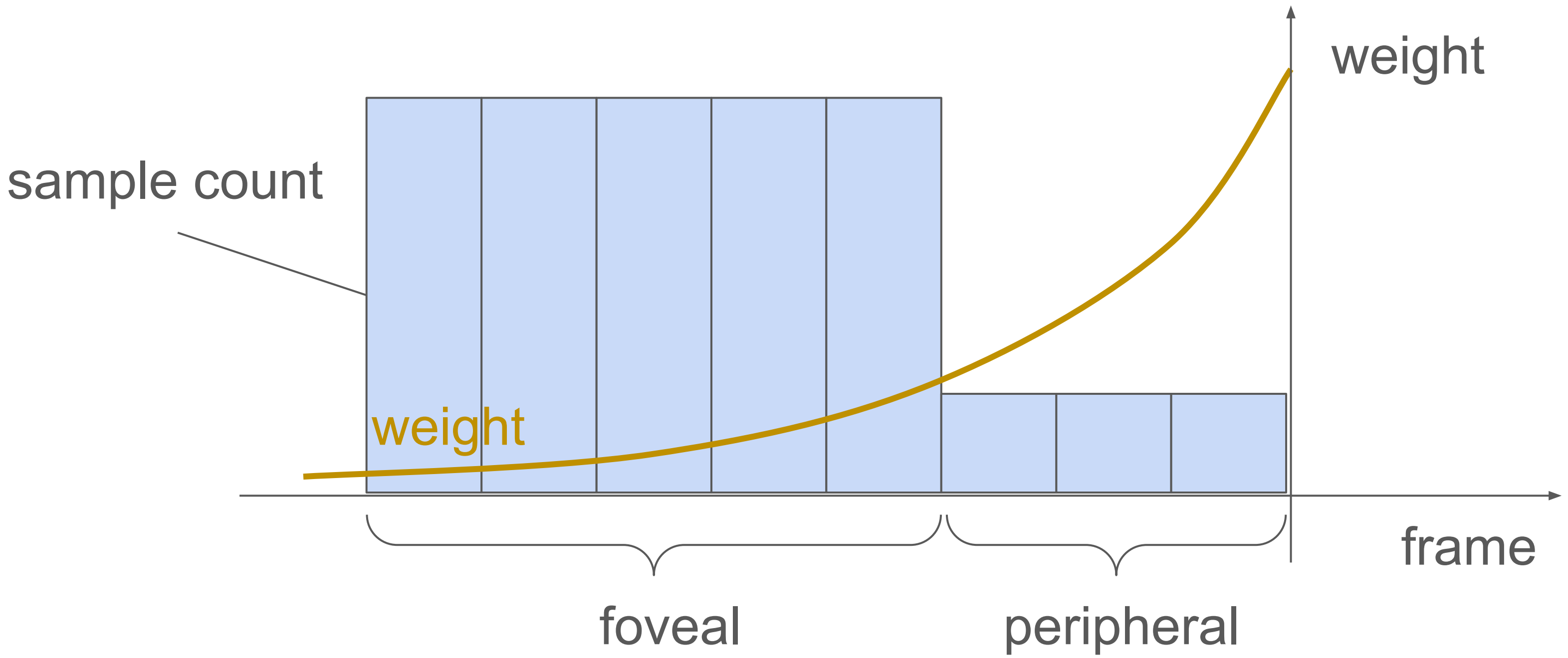
Ideally we should weight the reused colors proportional to sample counts.

However, such method cannot invalidated old samples.



Problem of Exponential Decay

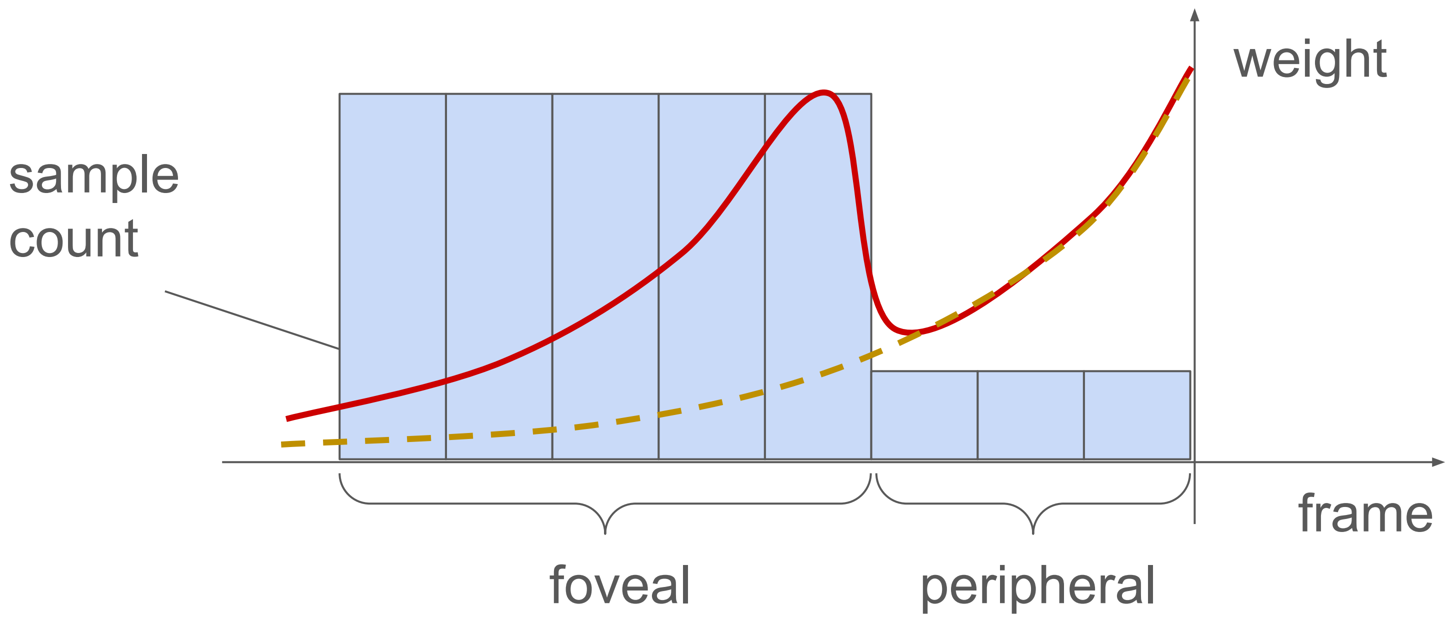
Using exponential decay without considering sample count, **we may put too much weight on pixels generated by smaller number of samples**.



Weighted by Sample Count

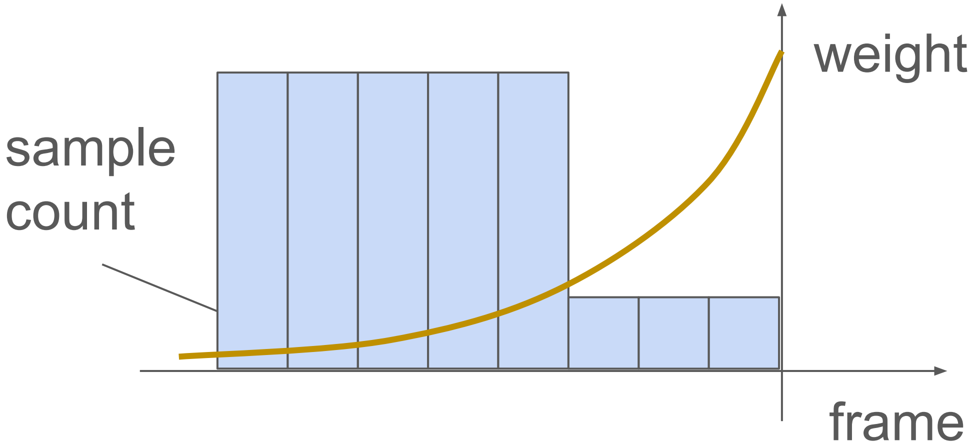
Can we simply **multiply the weights by number of samples**?

In fact, it's not that simple.

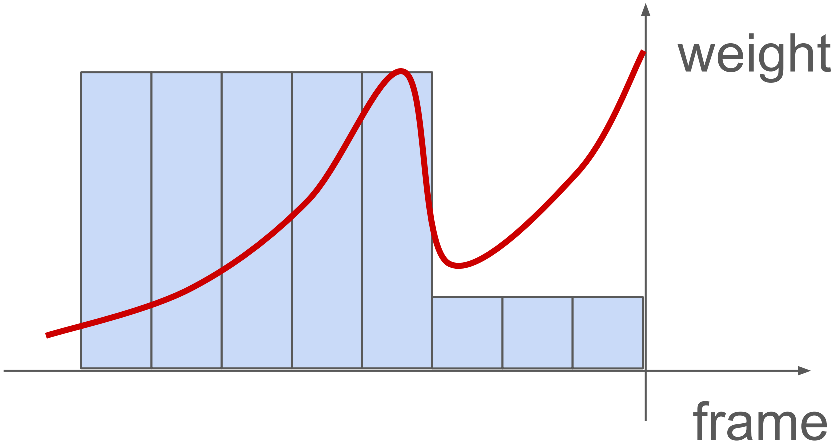


Definitions

Unweighted Method (original method)



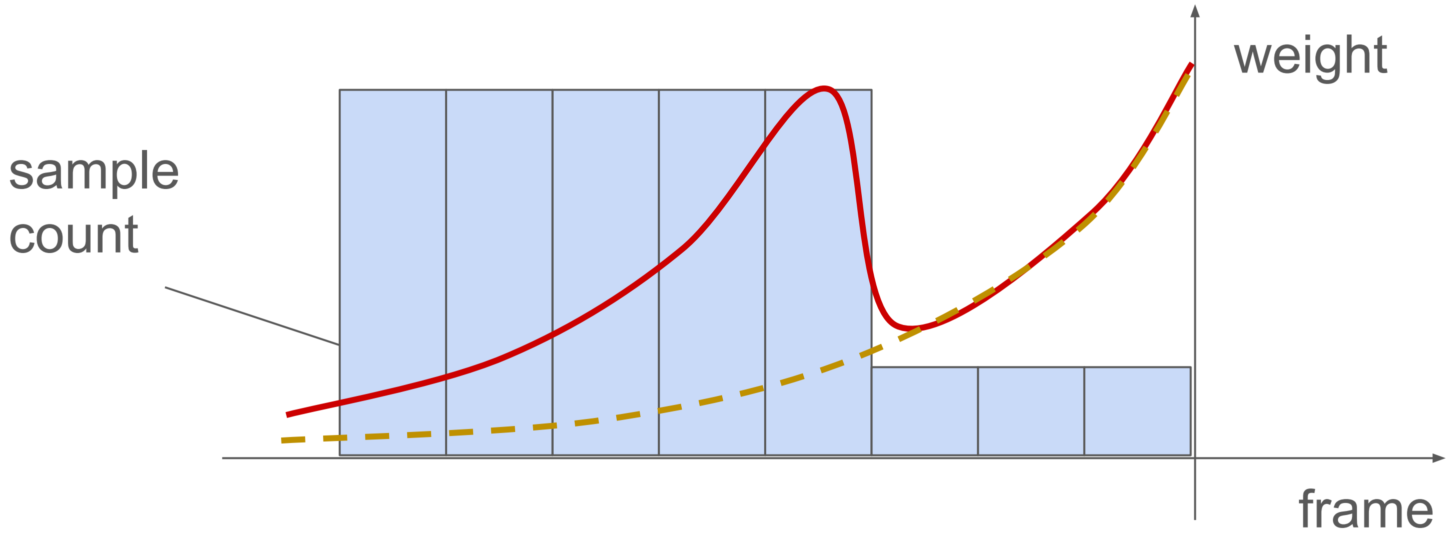
Weighted Method



Problems of Weighted Method

We observe two problems of weighted method:

1. Sometimes weighted method is noisier
2. Temporal bias may be different between pixels



Problems of Weighted Method

1. Sometimes weighted method is noisier

Reference

Unweighted

Weighted

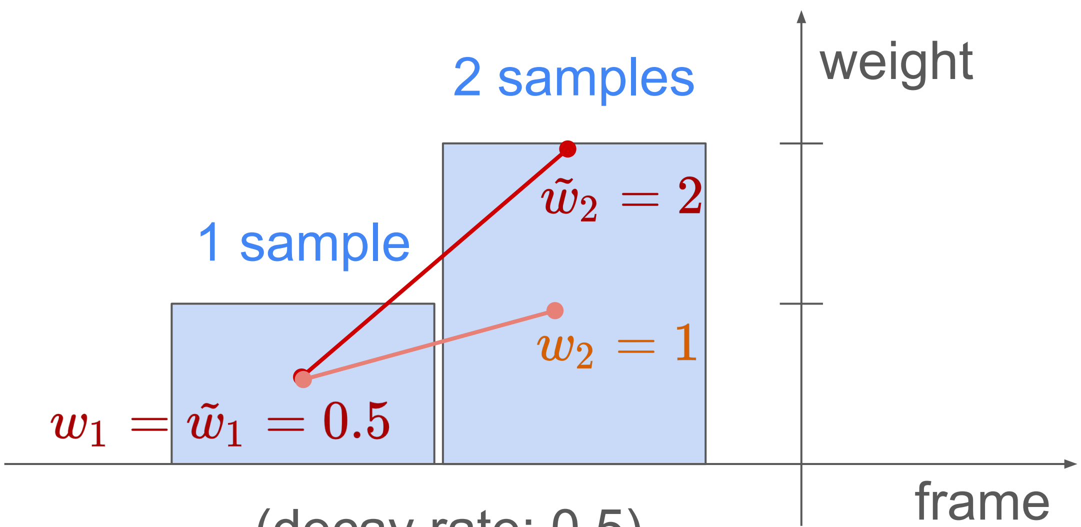
Problems of Weighted Method

1. Sometimes weighted method is noisier

Let w_t and $\tilde{w}_t$ be the weights of color at frame t for unweighted method and weighted method respectively.

To minimize the variance, the weights should be proportional to number of samples.

In the following example, w_t minimizes variance while $\tilde{w}_t$ does not.



(decay rate: 0.5)

Problems of Weighted Method

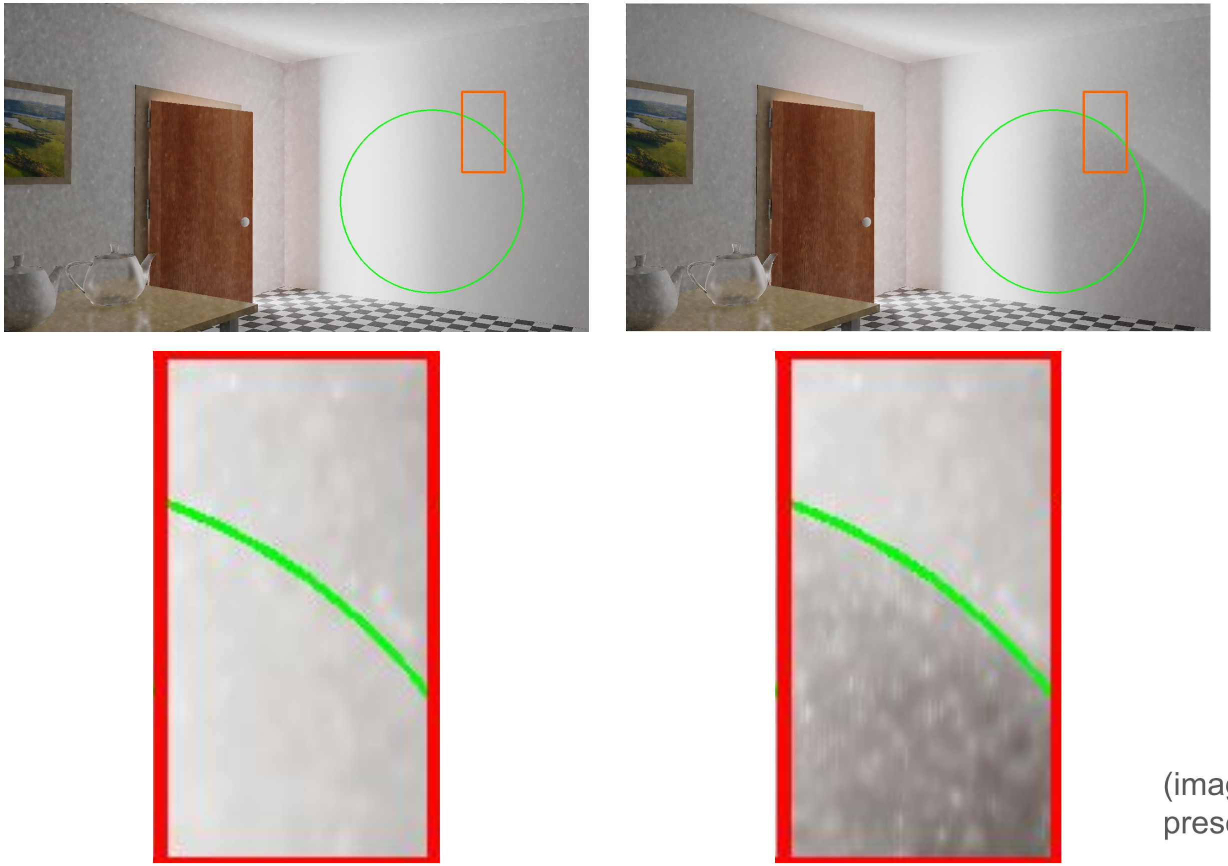
2. Temporal bias may be different between pixels

Problems of Weighted Method

2. Temporal bias may be different between pixels

Unweighted

Weighted

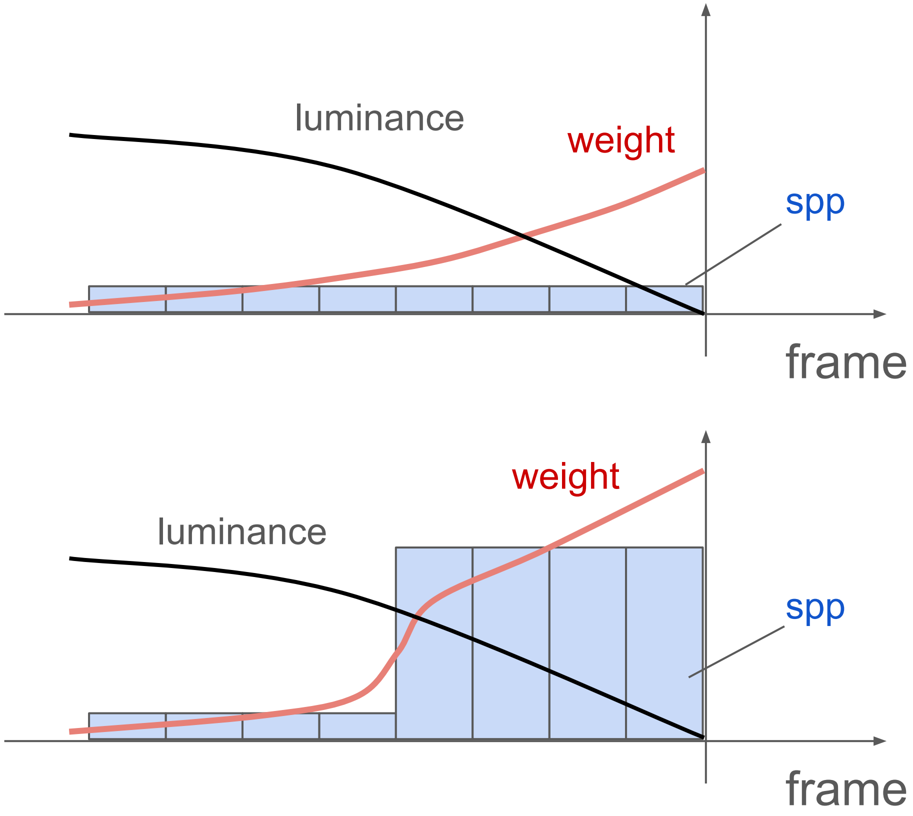


(images adjusted for presentation)

Problems of Weighted Method

Such artifacts appears when both **spp** and **illumination** changes.

(images adjusted for presentation)



Problems of Weighted Method

We observe two problems of weighted method:

1. Sometimes weighted method is noisier
2. Temporal bias may be different between pixels

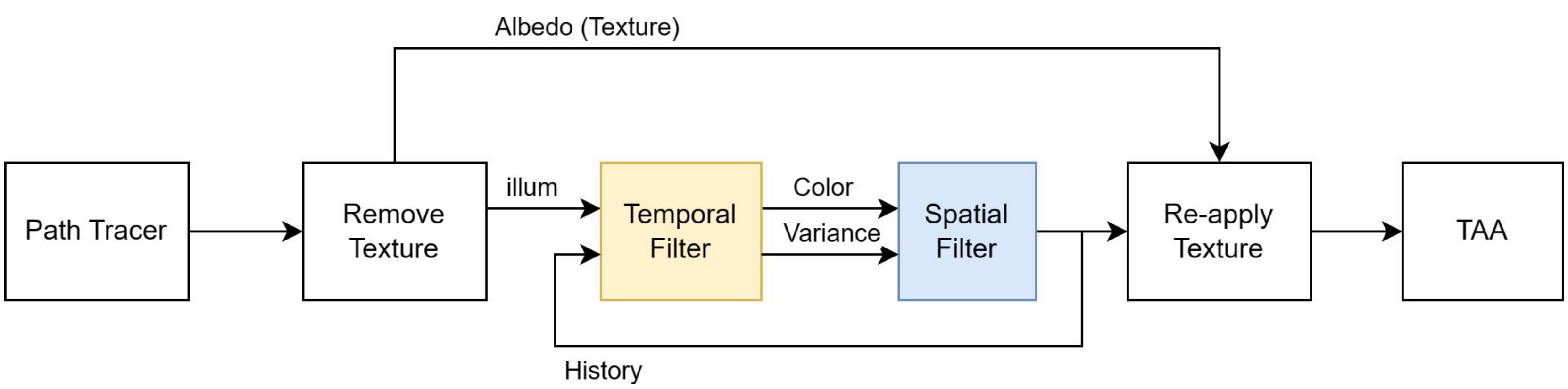
Unweighted Weighted

Overview

- Introduction
- Related Works
  - Spatio-temporal Reuse
  - Foveated Rendering
- Proposed Method
- Experiments
- Conclusion
- Limitaions & Future Work

Spatiotemporal **V**ariance-**G**uided **F**iltering: Real-Time Reconstruction for Path-Traced Global Illumination

- **SVGF** is a spatio-temporal filter for **1 sample per pixel** input.
- They estimate variance in the temporal filter and guide the spatial filter with estimated variance.



Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R. Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. "Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination". In Proceedings of High Performance Graphics, HPG ’17, New York, NY, USA, 2017. Association for Computing Machinery.

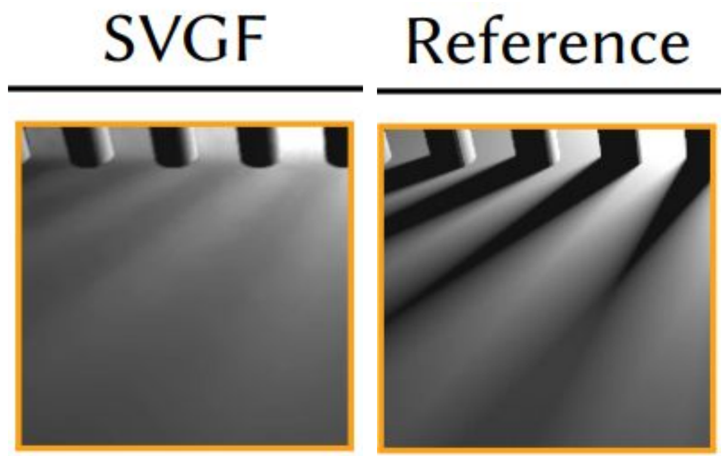
Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination

Temporal Filter

- SVGF uses exponential decay in temporal-filter pass

$$x'_t = \beta x'_{t-1} + (1 - \beta) x_t$$

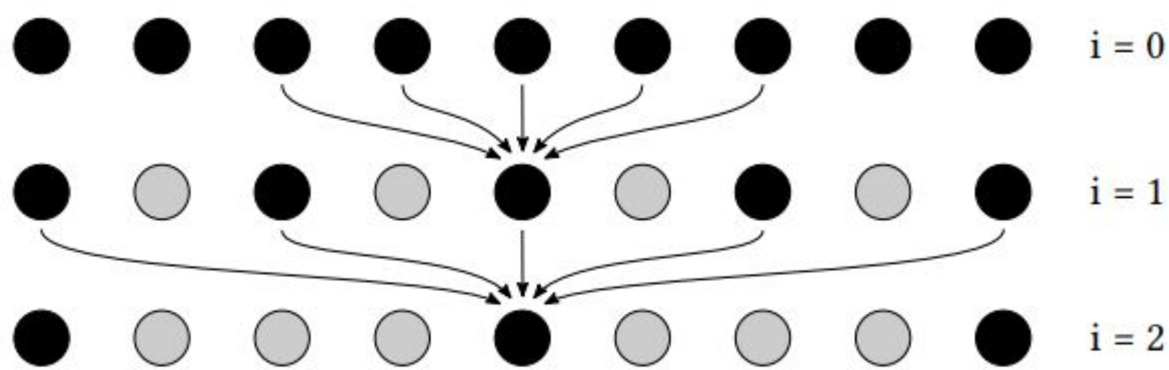
- A large β (0.95) was chosen to produce image with little noise, which yields temporal bias in animated scenes.



Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination

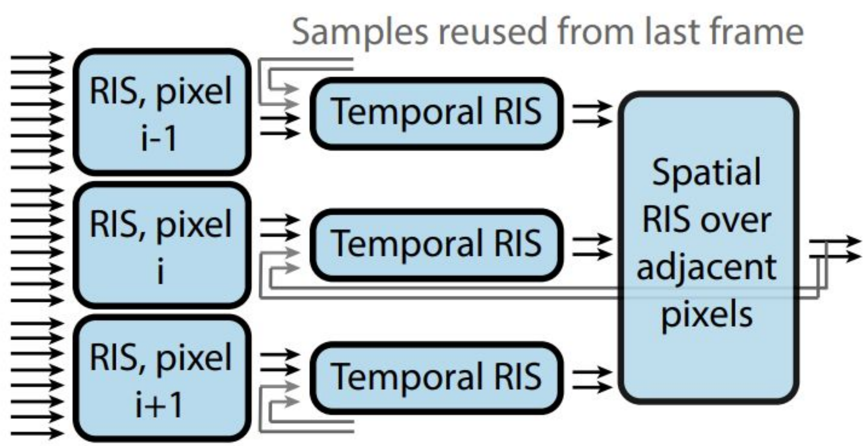
Spatial Filter:

- Hierarchical a-trous wavelet filter is used for spatial filtering.
- Each iteration increases the kernel footprint by a multiple.



Spatiotemporal Reservoir Resampling for Real-time Ray Tracing with Dynamic Direct Lighting (ReSTIR)

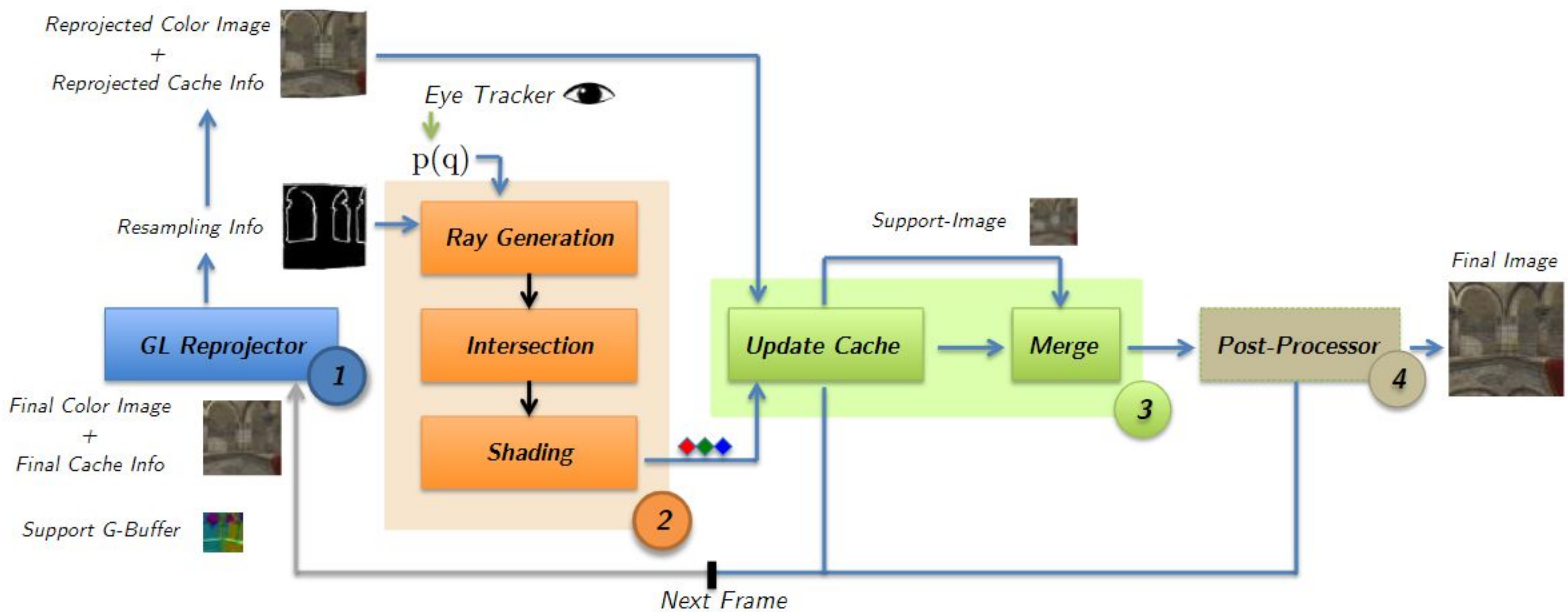
- Resampling samples instead of filtering colors.
- Implicitly assume constant number of samples per pixel.



Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting". ACM Trans. Graph., 39(4), aug 2020.

Foveated Real-Time Ray Tracing for Head-Mounted Displays

- Use cached colors and gaze point together to reduce required samples.

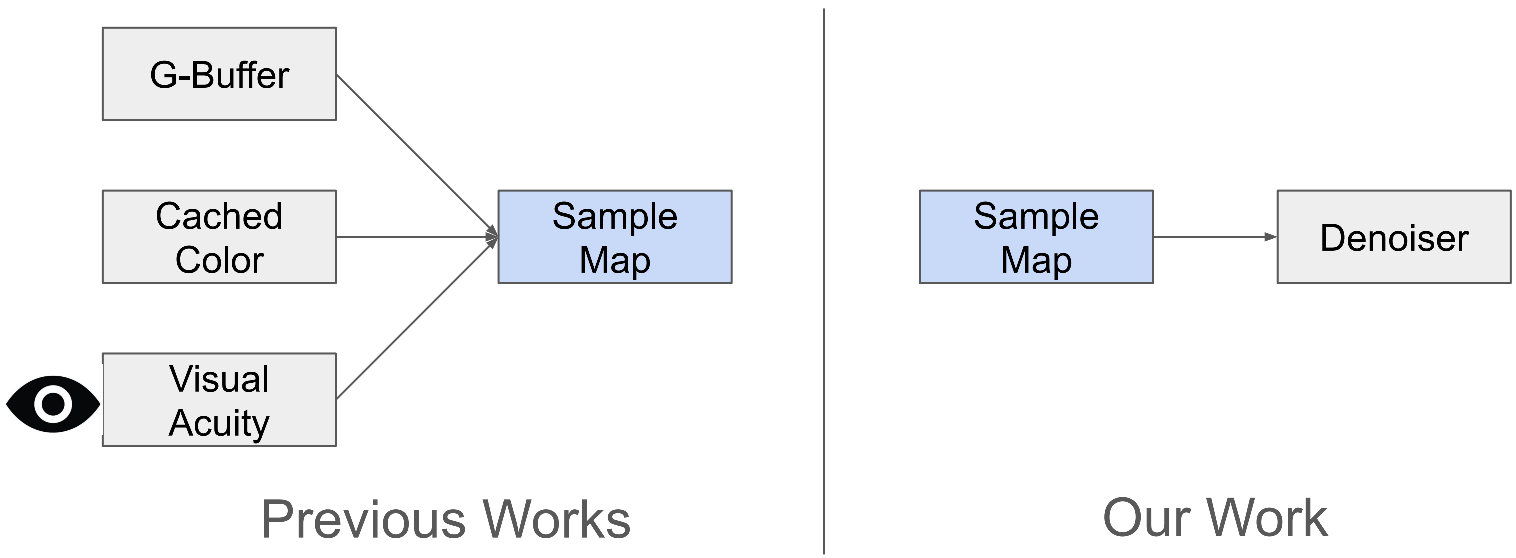


Martin Weier, Thorsten Roth, Ernst Kruijff, Andr´e Hinkenjann, Ars`ene P´erard-Gayot, Philipp Slusallek, and Yongmin Li. "Foveated real-time ray tracing for head-mounted displays". Comput. Graph. Forum, 35(7):289–298, oct 2016

Focus of Existing Foveated Raytracing vs. Ours

Existing foveated ray-tracing algorithms focus on **reducing number of samples** to take by various informations.

While our work focus on further **reducing noise** base on existing sample map.



Overview

- Introduction
- Related Works
- Proposed Method
  - Two-history Temporal Filter
  - Blending
  - Two-history Spatio-temporal Filter
- Experiments
- Conclusion
- Limitaions & Future Work

Recap: Problems of Weighted Method

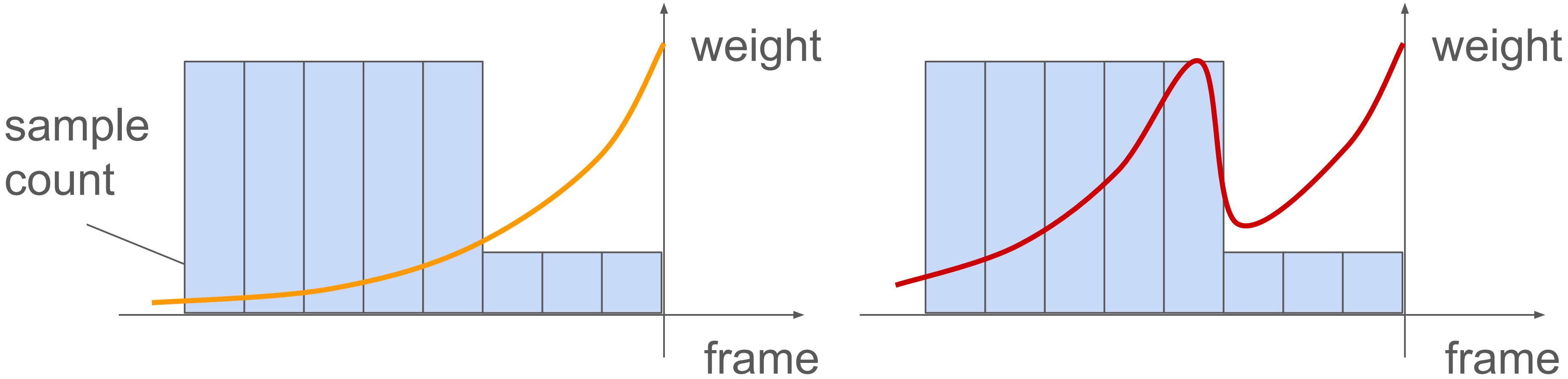
We want to make use of sample count while avoiding these problems:

1. Sometimes weighted method is noisier
2. Temporal bias may be different between pixels

Unweighted Weighted

Concept of Our Work

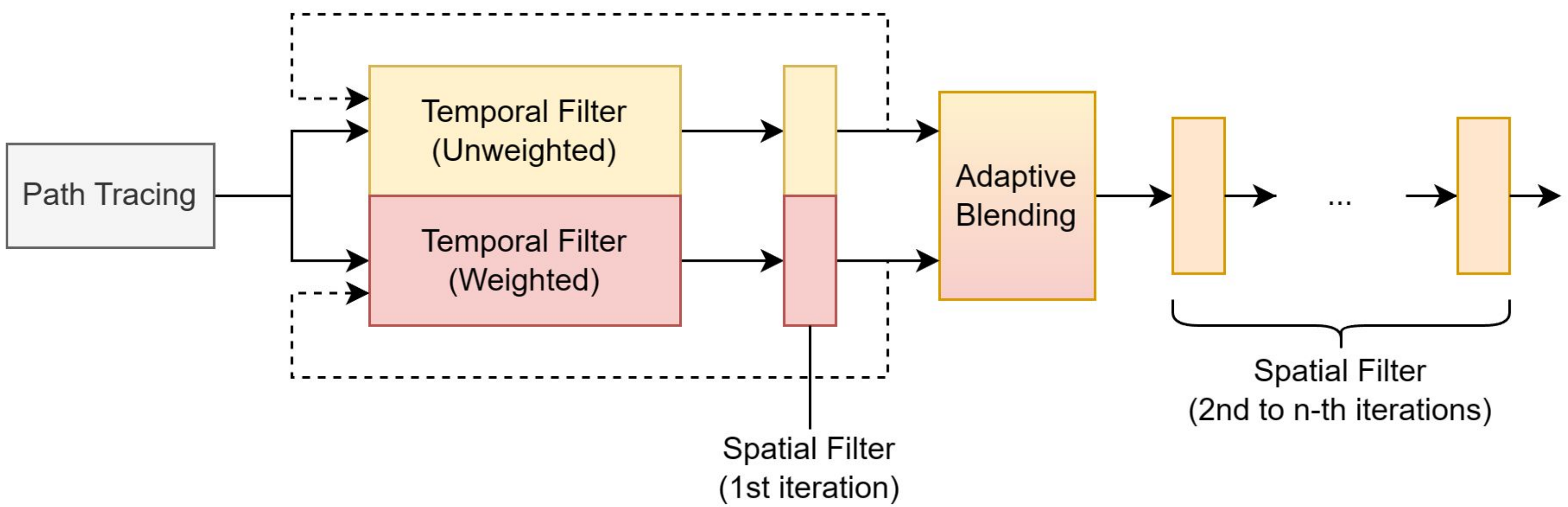
- When weighted is better => use weighted
- When weighted is worse => use unweighted



Two-history Method Overview

We do both unweighted method and weighted method

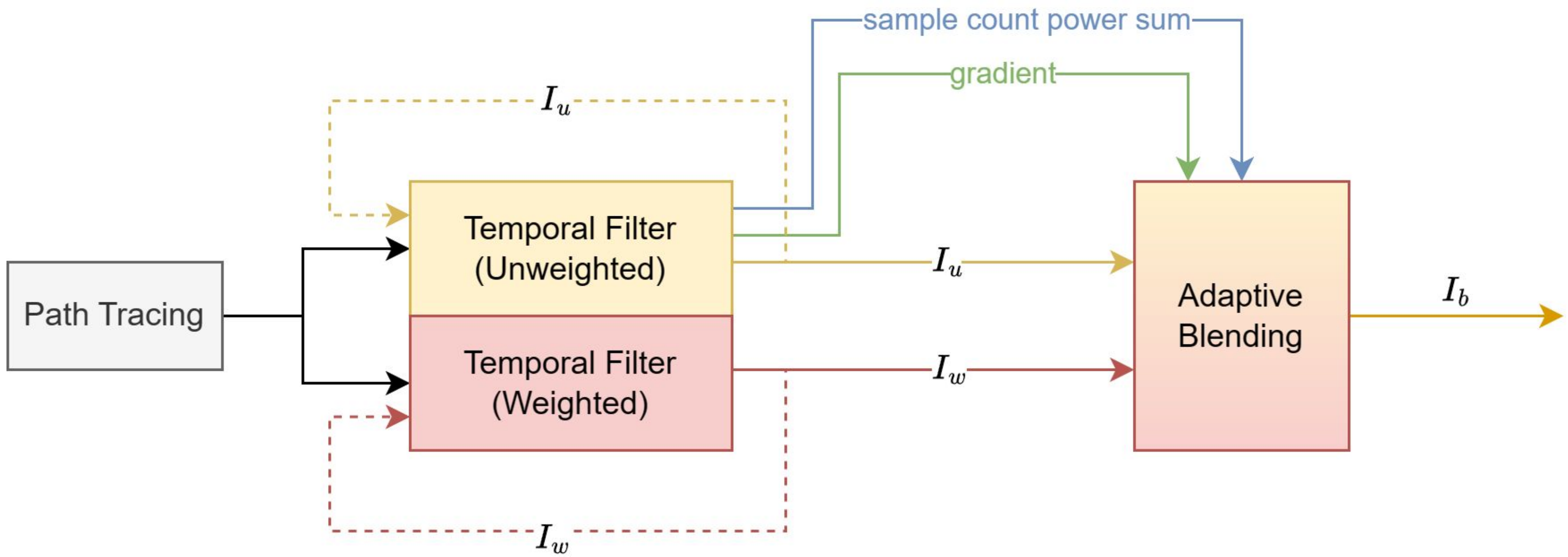
- When weighted is better => rely more on weighted
- When weighted is worse => rely more on unweighted



Two-history Temporal Filter

For simplicity, lets start with temporal filter only.

- We maintain two illumination histories, I_u and I_w, produced by weighted and unweighted method respectively.
- Then two outputs are blended adaptively guided by sample counts and gradients informatoins.



Problems of weighted method:

1. Sometimes weighted method is noisier
2. Temporal bias may be different between pixels

Blending

We conceptually do a two-phase blending

Each blending step avoids one problem of weighted method

$$I_{opt} = \mathrm{lerp}(I_w, I_u, p)$$

=> for avoiding first problem

$$I_b = \mathrm{lerp}(I_{opt}, I_u, q)$$

=> for avoiding second problem

Combinin them we get:

$$I_b = \mathrm{lerp}(I_w, I_u, \mathrm{lerp}(q, 1, p))$$

First Phase of Blending

We observe two issues of weighted method:

1. **Sometimes weighted method is noisier**
2. Temporal bias may be different between pixels

Unweighted Weighted

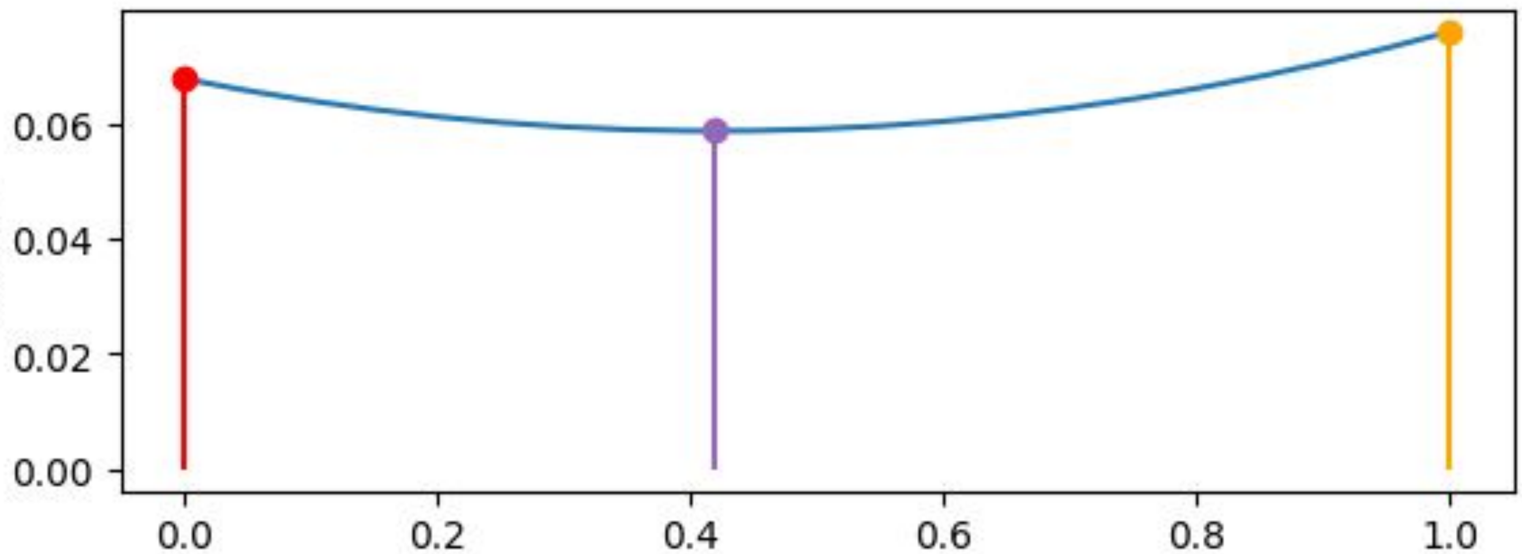
First Phase of Blending

To deal with that sometimes weighted method is noisier

- We find the optimal linear combination between two histories

$$I_{opt} = \mathrm{lerp}(I_w, I_u, p)$$

which minimize the variance



$$I_{opt} = \mathrm{lerp}(I_w, I_u, p)$$

First Phase of Blending

To find the best blending factor p, we solving the optimization problem:

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & \mathrm{Var}\left[\mathrm{lerp}(I_w, I_u, p)\right] \\ \text{subject to} \quad & 0 \le p \le 1 \end{aligned}$$

(assuming constant ground truth illumination to calculate variance from noise only)

The result would be:

$$p = \mathrm{clamp}\left(\frac{W_u^2 S_0 - W_u W_w S_1}{W_u^2 S_1 - 2 W_u W_w S_1 + W_w^2 S_{-1}}, 0, 1\right)$$

W_u = total weight of unweighted history

W_w = total weight of weighted method

$S_k = \sum_{i=1}^{t} n^k \beta^{2(t-i)}$ = weighted k-th power sum of of sample count

Prove of $p = \mathrm{clamp}\left(\frac{W_u^2 S_0 - W_u W_w S_1}{W_u^2 S_1 - 2W_u W_w S_1 + W_w^2 S_{-1}}, 0, 1\right)$

Definition of the histories:

$$I_u = \frac{\sum_{i=1}^{t} \beta^{t-i} x_i}{\sum_{i=1}^{t} \beta^{t-i}} = \frac{\sum_{i=1}^{t} \beta^{t-i} x_i}{W_u}$$

$$I_w = \frac{\sum_{i=1}^{t} \beta^{t-i} n_i x_i}{\sum_{i=1}^{t} \beta^{t-i} n_i} = \frac{\sum_{i=1}^{t} \beta^{t-i} n_i x_i}{W_w}$$

Assume the ground truth color is the same over time,

i.e. :

$$x_i = \sum_{j=1}^{n_i} x_j$$

$$\text{where } x_j \sim X$$

Prove of $p = \text{clamp}\left(\frac{W_u^2 S_0 - W_u W_w S_1}{W_u^2 S_1 - 2 W_u W_w S_1 + W_w^2 S_{-1}}, 0, 1\right)$

To find the extreme value:

$$
\begin{aligned}
&\frac{d}{dp}\operatorname{Var}[\operatorname{lerp}(I_w, I_u, p)] = 0 \\
\Rightarrow &\frac{d}{dp}\sum_{i=1}^{t}\left((1-p)\frac{\beta^{t-i} n_i}{W_w} + p\frac{\beta^{t-i}}{W_u}\right)^2 \frac{1}{n_i}\operatorname{Var}(X) = 0 \\
\Rightarrow &2\sum_{i=1}^{t}\left((1-p)\frac{\beta^{t-i} n_i}{W_w} + p\frac{\beta^{t-i}}{W_u}\right)\left(\frac{\beta^{t-i}}{W_u} - \frac{\beta^{t-i} n_i}{W_w}\right)\frac{1}{n_i} = 0 \\
\Rightarrow &p = \frac{\frac{1}{W_w^2}\sum_{i=1}^{t}\beta^{2(t-i)} n_i - \frac{1}{W_w W_u}\sum_{i=1}^{t}\beta^{2(t-i)}}{\frac{1}{W_w^2}\sum_{i=1}^{t}\beta^{2(t-i)} n_i - 2\frac{1}{W_w W_u}\sum_{i=1}^{t}\beta^{2(t-i)} + \frac{1}{W_u^2}\sum_{i=1}^{t}\beta^{2(t-i)}\frac{1}{n_i}}
\end{aligned}
$$

$$\text{Let } S_k = \sum_{i=1}^{t}\beta^{2(t-i)} n_i^k$$

Then: $$p = \frac{W_u^2 S_1 - W_w W_u S_0}{W_u^2 S_1 - 2 W_w W_u S_0 + W_w^2 S_{-1}}$$

Prove of $p = \text{clamp}\left(\frac{W_u^2 S_0 - W_u W_w S_1}{W_u^2 S_1 - 2W_u W_w S_1 + W_w^2 S_{-1}}, 0, 1\right)$

Finally, to prove the extreme value is indeed a minimum, we show that $\frac{d^2 \text{Var}[\text{lerp}(I_w, I_u, p)]}{dp^2} \geq 0$

$$\frac{d^2 \text{Var}[\text{lerp}(I_w, I_u, p)]}{dp^2}$$
$$= \frac{d \sum_{i=0}^{t} 2\left((1-p)\frac{\beta^{t-i} n_i}{W_w} + p\frac{\beta^{t-i}}{W_u}\right) \frac{\text{Var}(X)}{n_i} \left(\frac{\beta^{t-i}}{W_u} - \frac{\beta^{t-i} n_i}{W_w}\right)}{dp}$$
$$= 2 \cdot \text{Var}(X) \cdot \sum_{i=1}^{t} \frac{1}{n_i} \left(\frac{\beta^{t-i}}{W_u} - \frac{\beta^{t-i} n_i}{W_w}\right)^2 \geq 0$$

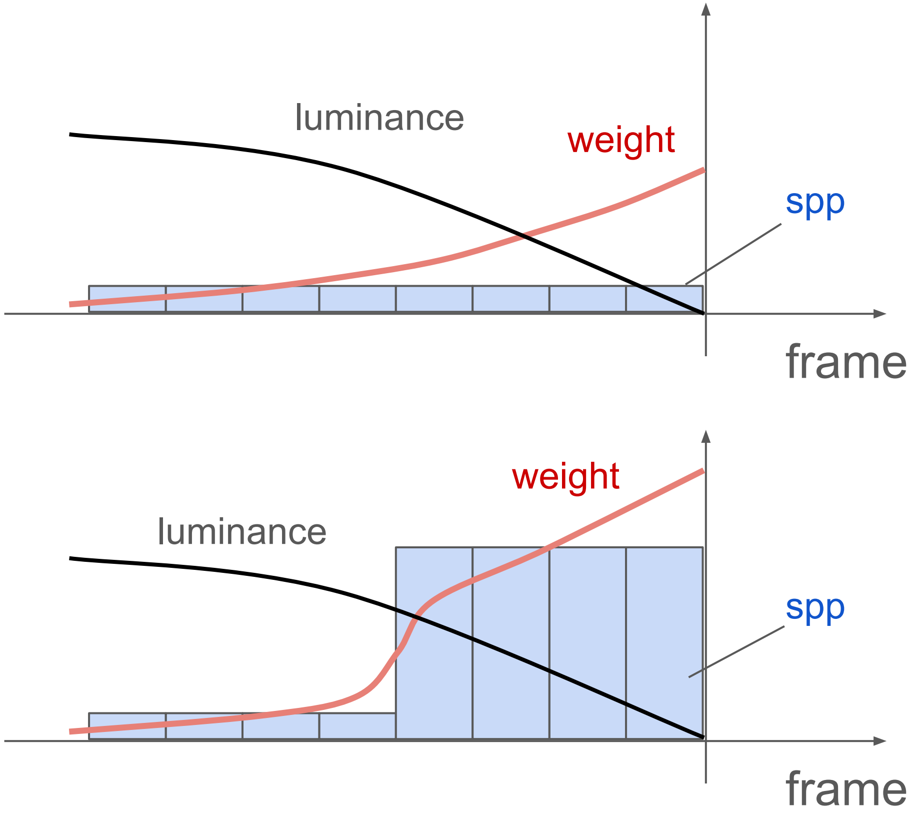
How Does Two-History Method Resolve The Issues?

We observe two issues of weighted method:

1. It's not always better to weight by sample count
2. **Temporal bias may be different between pixels**

When Will Temporal Bias be Perceived?

Temporal bias is only perceptible when luminance changes drastically.



(images adjusted for presentation)

Gradient Guided Blending

Artifacts may appear when the luminances change fast.
So the larger the absolute value of gradient is, the more we rely on unweighted history.

$$I_b = (1 - q)I_{opt} + qI_u$$

$$q = \text{clamp}\left(S \cdot \frac{\text{lum}(|g|)}{STD(\text{lum}(I_u))}, 0, 1\right)$$

where:

I_u = unweighted illumination history

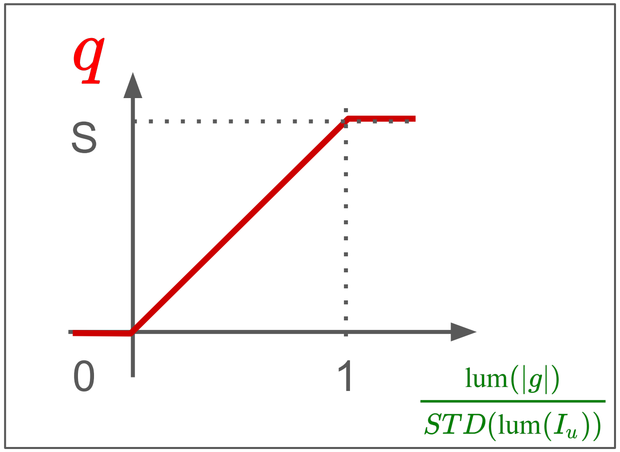
$I_{opt} = \text{lerp}(I_w, I_u, p)$

I_b = blended illumination

g = estimated temporal gradient

S = scaling parameter

lum = luminance



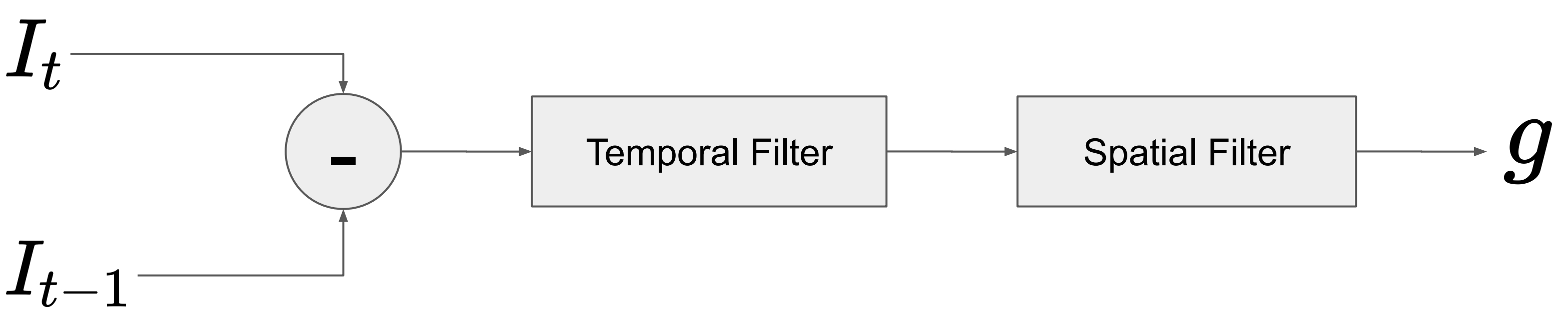
We found that **S=1.0** works well in most of the scenes we have tested.

Gradient Estimation

We estimate temporal gradient by temporally accumulate luminance difference using exponential moving average:

$$g_t = \beta g_{t-1} + (1-\beta)\left(I_t - I_{t-1}\right)$$

Since gradient estimated by such method is noisy, we filter it by an edge-aware spatial filter.



Gradient Guided Blending

where:

I_u = unweighted illumination history

$I_{opt} = \text{lerp}(I_w, I_u, p)$

I_b = blended illumination

g = estimated temporal gradient

S = slope parameter

lum = luminance

$$I_b = (1 - q)I_{opt} + qI_u$$

$$q = \text{clamp}\left(S \cdot \boxed{\frac{\text{lum}(|g|)}{STD(\text{lum}(I_u))}}, 0, 1\right)$$

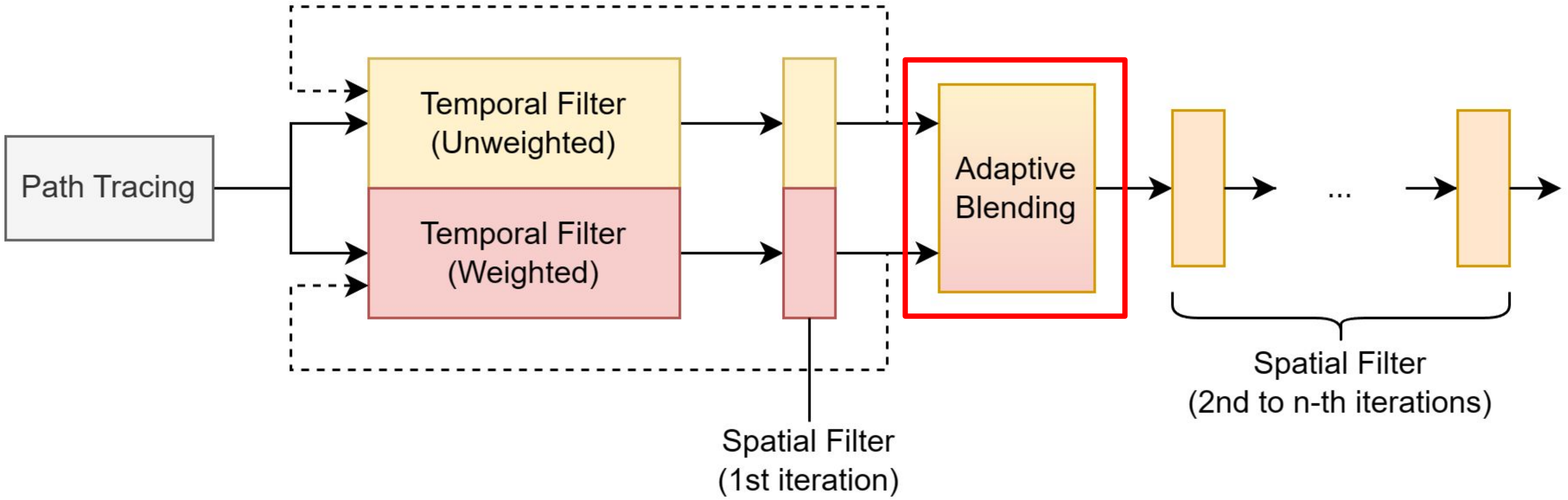
We normalize the luminance change by **dividing the standard deviation** of luminance.

Although the formula is imperical, is has the following good properties:

- $\frac{\text{lum}(|g|)}{STD(\text{lum}(I_u))}$ is limited to [0,C] for some constant C
- Since larger noise yields larger STD, the more noisy the input, the more I_{opt} (which is less noisy) would be adopted.

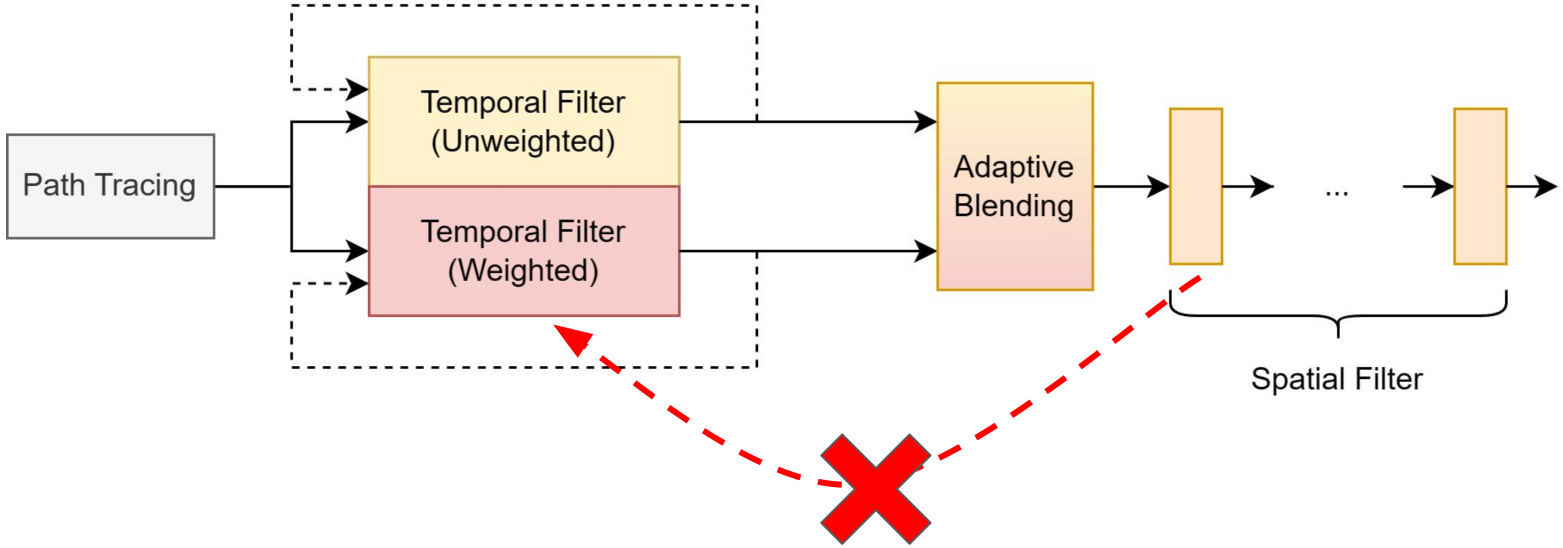
Two-history Spatio-temporal Filter

- To work with spatio-temporal filter, we put adaptive blending pass **between 1st and 2nd iterations** of spatial filter.
- While it's possible to apply adaptive blending earlier or later, both ways has their drawbacks.



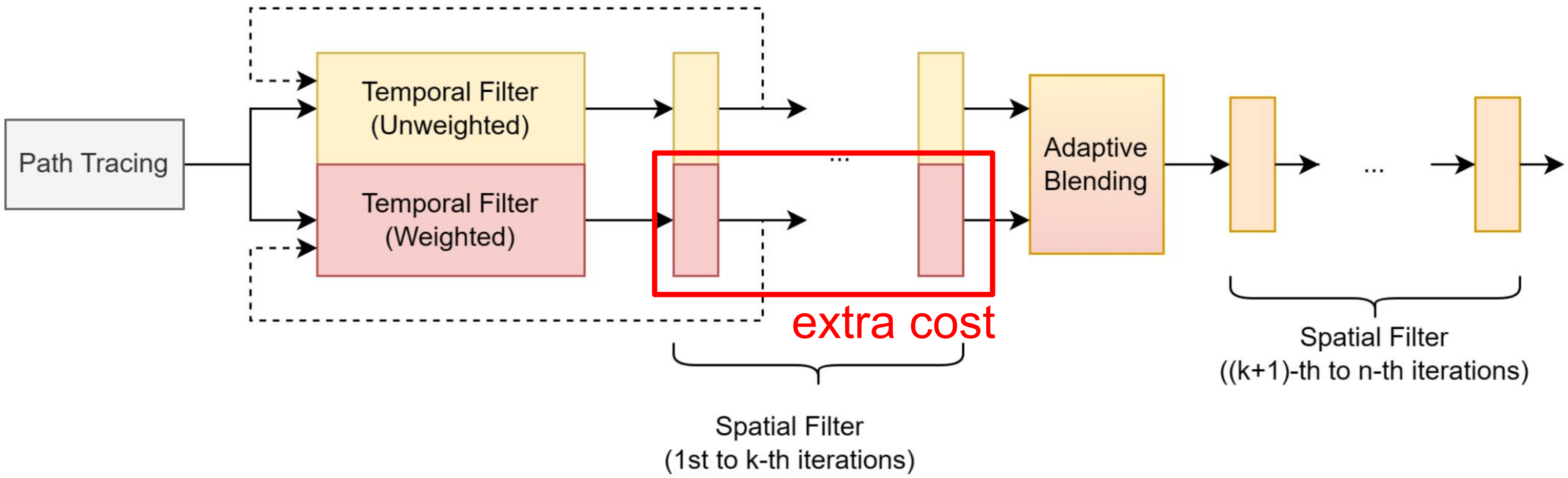
Why Not Blend Earlier

- Spatio-temporal filters usually require feeding back spatial filtered output into the temporal filter, so as to reuse pixels from broader area as history growth.
- If blending happens before all spatial filter iterations, we can't feedback spatially filtered illuminations into temporal filters because blended history will break the accumulation policies.

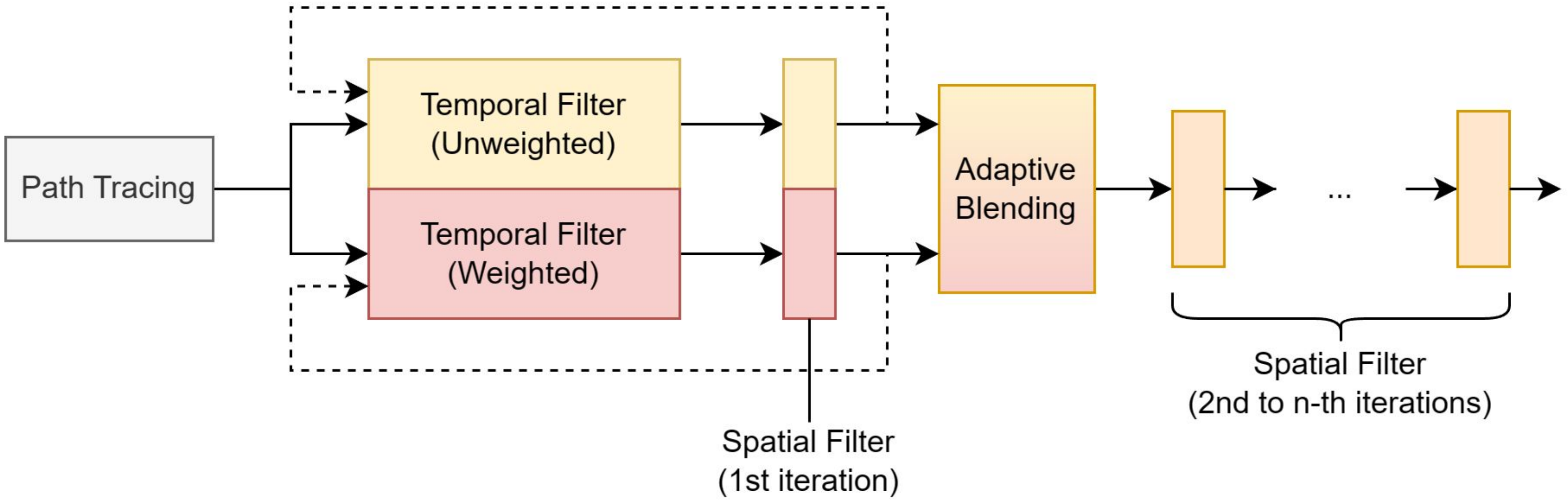


Why Not Blend Later

- Spatial filter for unweighted and weighted history are calculated separately before blending.
- If we apply blending after more spatial filter iterations, the computation cost will increase.



Two-history Spatio-temporal Filter

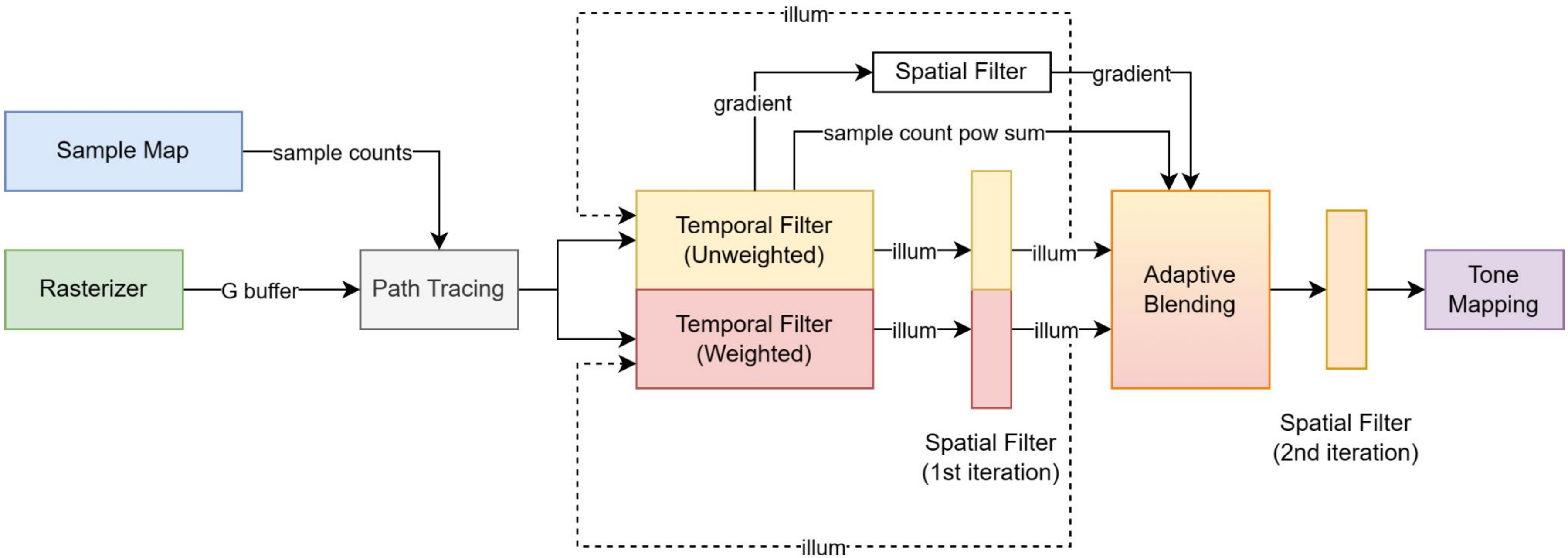


Overview

- Introduction
- Related Works
- Proposed Method
- Experiments
- Conclusion
- Limitaions & Future Work

Experiment - Implementaiton

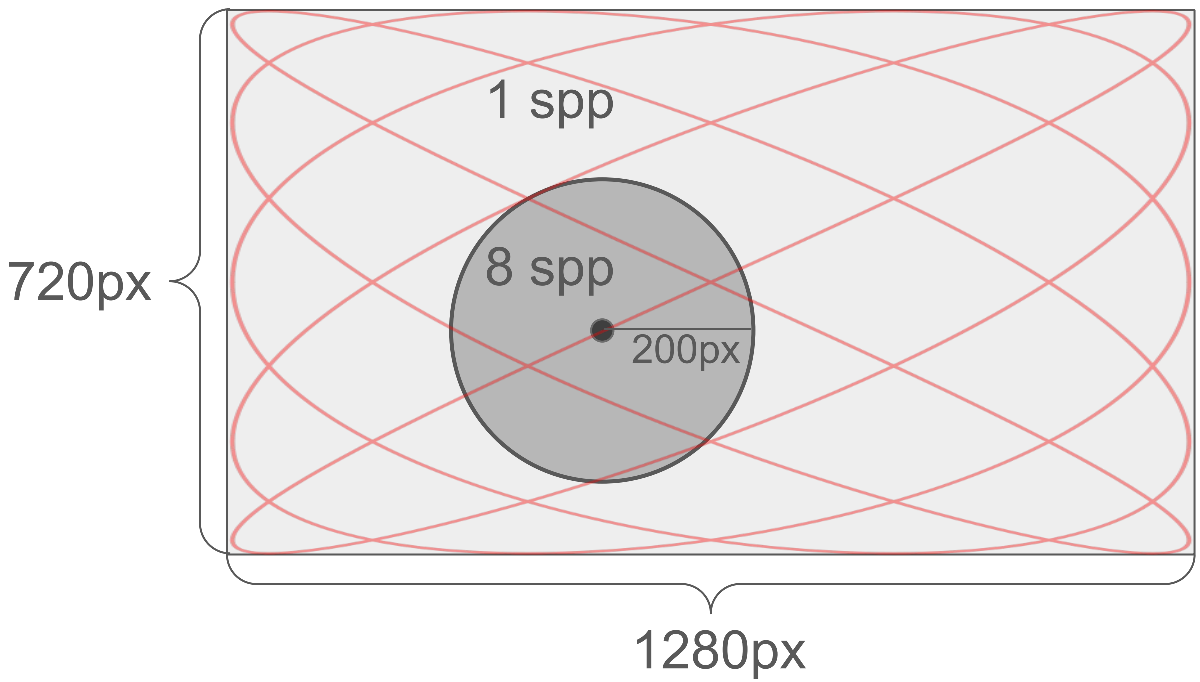
- We modify SVGF by our Two-history Approach
- Since any improvement in temporal filter would be less significant after spatial-filtering due to blurring, we use only 2 iterations instead of 4 that recommended by the original paper.



Experiment - Foveated Rendering

We test our algorithm with a simple foveated rendering algorithm:

- 8 spp for foveal area (a circle of 200px radius)
- 1 spp for peripheral area
- We use synthetic gaze points that move along a Lissajous curve.



Metrics

We use the following metrics to measure image quality

- relMSE (for HDR image)

$$\frac{1}{|P|}\sum_{t=1}^{|P|}\frac{(y_i - r_i)}{(\epsilon + r_i^2)}$$

- SSIM[WBSS04] (for tone mapped image)

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\,\mu_x\,\mu_y + C_1)\,(2\,\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)\,(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004

Experiment - Foveated Rendering

Average relMSE and SSIM:

	relMSE			SSIM		
Scene	Unweighted	Weighted	Two-history	Unweighted	Weighted	Two-history
VeachAjar	4.435e-2	**4.137e-2**	4.213e-2	0.9321	**0.9453**	0.9422
VeachAjarAnimated	3.559e-2	3.716e-2	**3.457e-2**	0.9280	**0.9371**	0.9352
BistroExterior	1.507e-1	1.505e-1	**1.492e-1**	0.9316	0.9343	**0.9359**
BistroInterior	2.128e+2	**2.098e+2**	2.111e+2	0.8528	0.8534	**0.8548**
BistroInterior_Wine	1.683e+0	**1.678e+0**	1.678e+0	0.9311	0.9340	**0.9354**
SunTemple	1.246e+0	**1.199e+0**	1.212e+0	0.8788	0.8893	**0.8903**
EmeraldSquare_Day	1.690e+0	**1.642e+0**	1.668e+0	0.8632	0.8653	**0.8667**
EmeraldSquare_Dusk	1.580e-4	1.617e-4	**1.558e-4**	0.9805	0.9811	**0.9812**
ZeroDay_1	1.491e-2	1.473e-2	**1.311e-2**	0.9543	0.9540	**0.9563**
ZeroDay_7	**9.159e-3**	1.250e-2	9.181e-3	0.9547	0.9537	**0.9561**
ZeroDay_7_Colored_Lights	1.542e-2	2.161e-2	**1.540e-2**	0.9415	0.9411	**0.9436**

Unweighted:	unmodified SVGF
Weighted:	SVGF with weights multiplied by sample count
Ours:	SVGF modified by two-history approach

Experiment - Foveated Rendering

Average relMSE and SSIM relative to unweighted method:

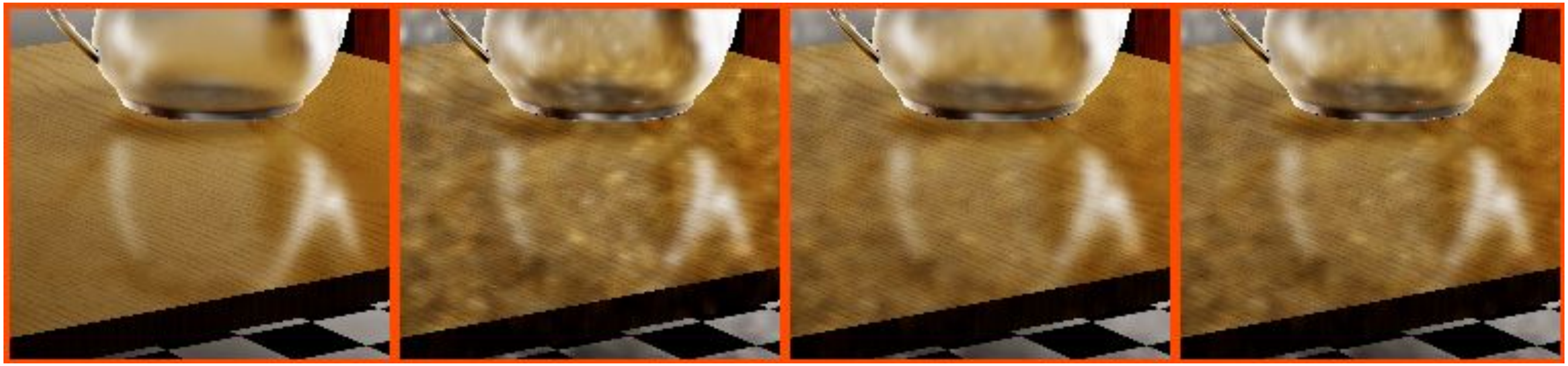
	relMSE		SSIM	
Scene	Weighted	Two-history	Weighted	Two-history
VeachAjar	-6.722%	-5.015%	+1.4142%	+1.0833%
VeachAjarAnimated	+4.409%	-2.862%	+0.9828%	+0.7765%
BistroExterior	-0.196%	-1.014%	+0.2973%	+0.4657%
BistroInterior	-1.419%	-0.797%	+0.0722%	+0.2420%
BistroInterior_Wine	-0.333%	-0.308%	+0.3100%	+0.4571%
SunTemple	-3.759%	-2.724%	+1.1955%	+1.3103%
EmeraldSquare_Day	-2.813%	-1.266%	+0.2354%	+0.3996%
EmeraldSquare_Dusk	+2.397%	-1.379%	+0.0589%	+0.0747%
ZeroDay_1	-1.196%	-12.114%	-0.0284%	+0.2162%
ZeroDay_7	+36.487%	+0.238%	-0.1050%	+0.1457%
ZeroDay_7_Colored_Lights	+40.137%	-0.125%	-0.0336%	+0.2280%

- Improved
- Worsen

Experiment - Foveated Rendering

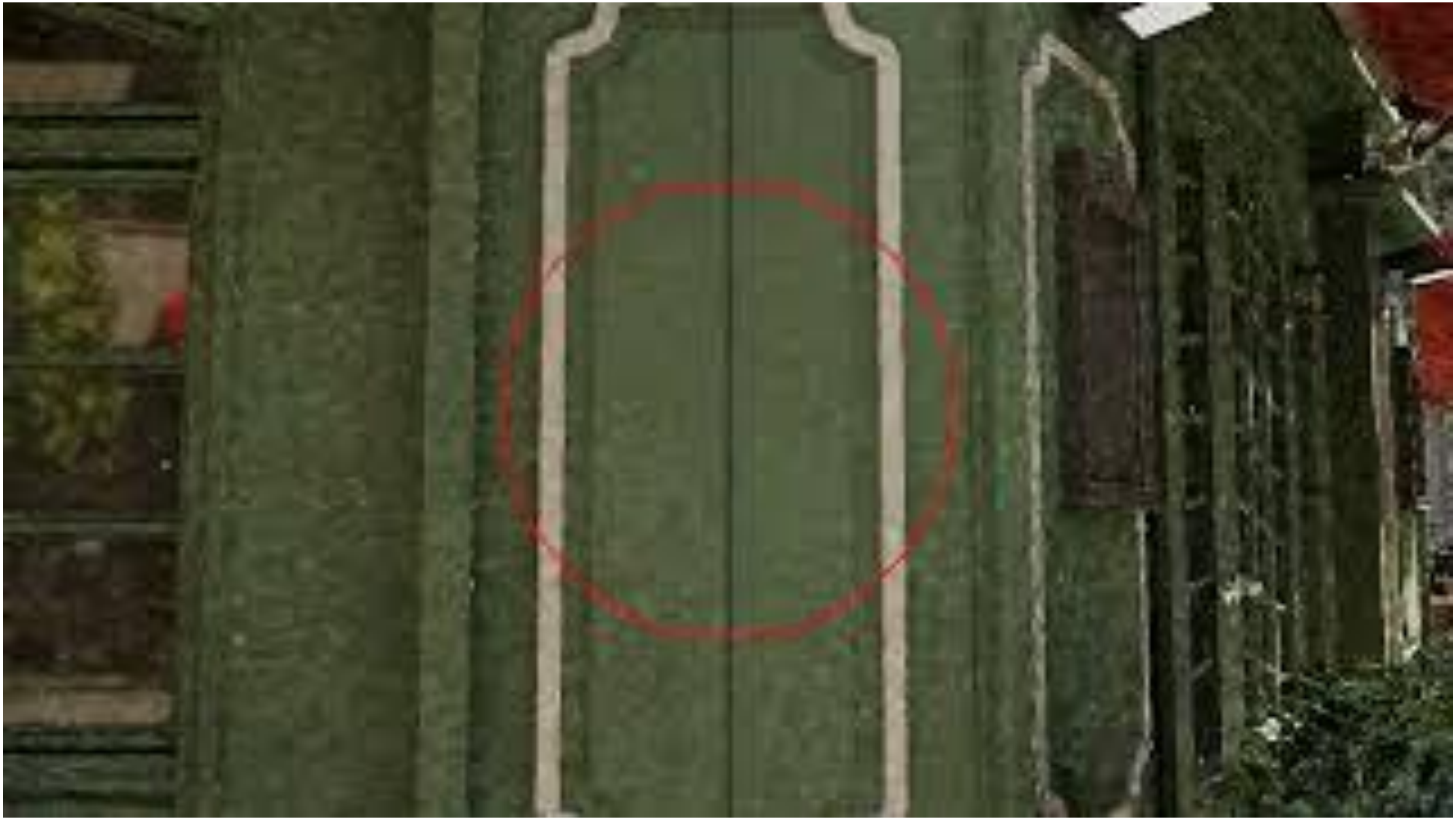
When weighted method works well:

Reference | Unweighted | Weighted | Two-history



(images are enhanced for presentation)

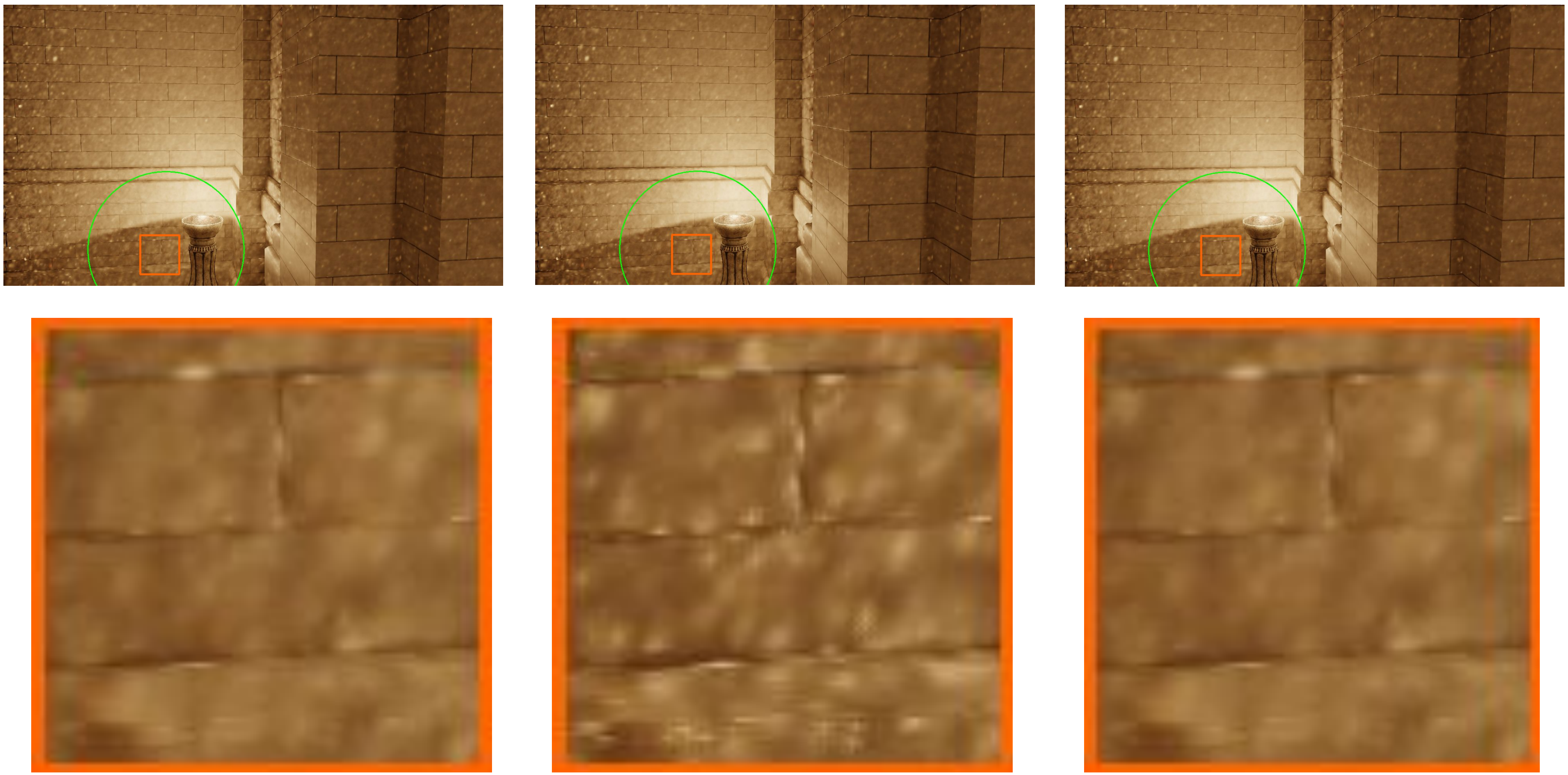
Experiment - Foveated Rendering



Experiment - Foveated Rendering

When weighted method is noisy:

Unweighted | Weighted | Two-history



Experiment - Foveated Rendering

When inconsistent temporal bias is significant:

Unweighted

Weighted

Two-history

Weighted Two-history

Experiment - Foveated Rendering

Error in foveal area:

	relMSE			SSIM		
Scene	Unweighted	Weighted	Two-history	Unweighted	Weighted	Two-history
VeachAjar	7.639e-3	1.076e-2	**7.540e-3**	0.9659	0.9312	**0.9660**
VeachAjarAnimated	6.272e-3	2.378e-2	**6.223e-3**	0.9624	0.9247	**0.9625**
BistroExterior	2.618e-2	3.532e-2	**2.591e-2**	**0.9693**	0.9422	0.9692
BistroInterior	7.221e+1	**4.989e+1**	6.156e+1	**0.8603**	0.8354	0.8601
BistroInterior_Wine	2.438e-1	2.461e-1	**2.385e-1**	**0.9582**	0.9319	0.9582
SunTemple	2.181e-1	2.798e-1	**2.180e-1**	**0.9125**	0.8633	0.9124
EmeraldSquare_Day	6.582e-1	**2.088e-1**	4.815e-1	0.9344	0.9120	**0.9345**
EmeraldSquare_Dusk	3.576e-5	6.161e-5	**3.484e-5**	0.9895	0.9879	**0.9896**
ZeroDay_1	4.377e-3	9.542e-3	**4.290e-3**	**0.9744**	0.9542	0.9743
ZeroDay_7	2.919e-3	8.988e-3	**2.870e-3**	**0.9784**	0.9613	0.9783
ZeroDay_7_Colored_Lights	5.900e-3	1.563e-2	**5.720e-3**	**0.9701**	0.9499	0.9701

Unweighted: unmodified SVGF
Weighted: SVGF with weights multiplied by sample count
Ours: SVGF modified by two-history approach

Experiment - Foveated Rendering

Error in foveal area relative to unweighted method

	relMSE		SSIM	
Scene	Weighted	Two-history	Weighted	Two-history
VeachAjar	+40.837%	**-1.287%**	-3.5940%	**+0.0116%**
VeachAjarAnimated	+279.064%	**-0.790%**	-3.9232%	**+0.0078%**
BistroExterior	+34.906%	**-1.025%**	-2.7954%	-0.0034%
BistroInterior	**-30.906%**	**-14.741%**	-2.8883%	-0.0147%
BistroInterior_Wine	+0.955%	**-2.143%**	-2.7475%	-0.0054%
SunTemple	+28.294%	**-0.012%**	-5.3940%	-0.0060%
EmeraldSquare_Day	**-68.272%**	**-26.852%**	-2.3976%	**+0.0084%**
EmeraldSquare_Dusk	+72.301%	**-2.559%**	-0.1615%	**+0.0145%**
ZeroDay_1	+118.000%	**-1.987%**	-2.0680%	-0.0025%
ZeroDay_7	+207.912%	**-1.670%**	-1.7441%	-0.0024%
ZeroDay_7_Colored_Lights	+164.914%	**-3.043%**	-2.0783%	-0.0017%

- Improved
- Worsen

In foveal area, weighted method usually doesn't work well, while our method can still reduce some error.

Experiment - Foveated Rendering

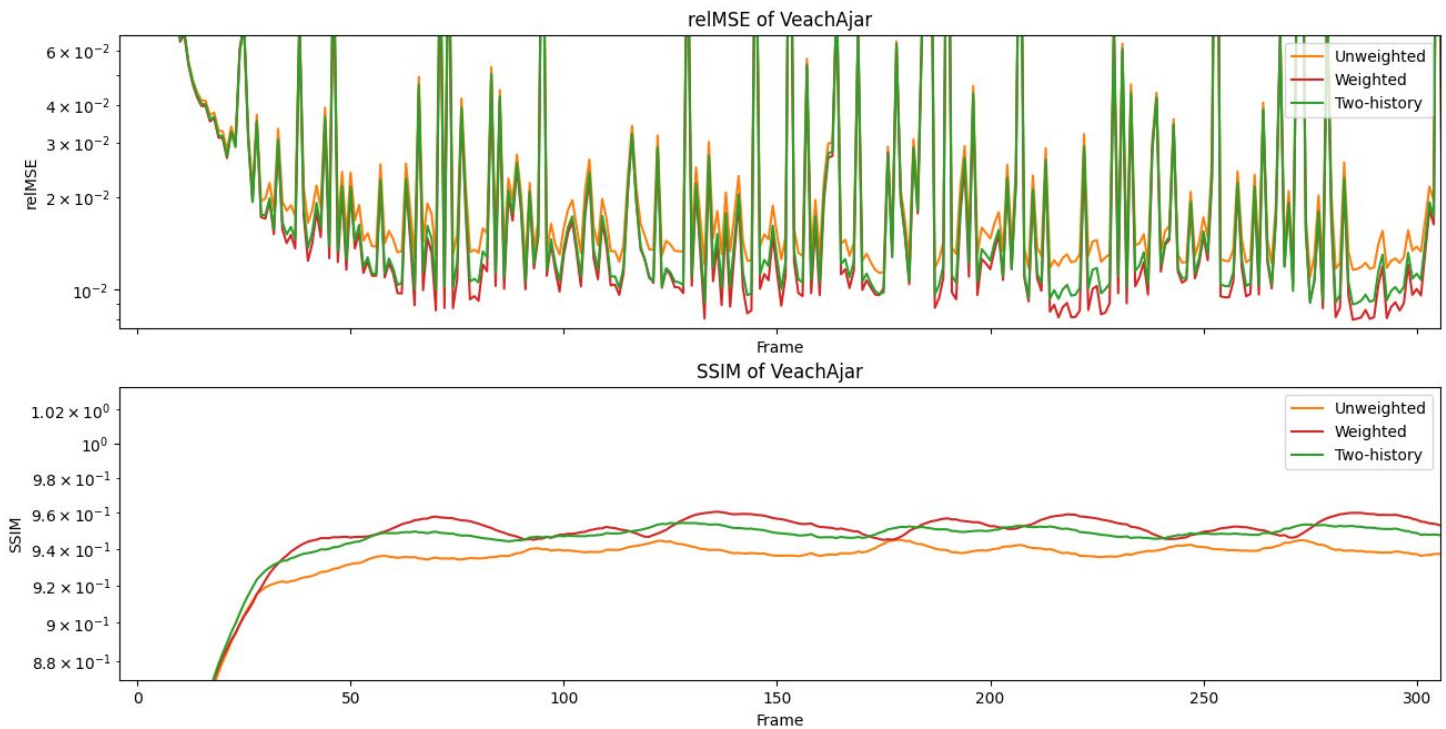
	relMSE		SSIM	
Scene	Weighted	Two-history	Weighted	Two-history
VeachAjar	-6.722%	-5.015%	+1.4142%	+1.0833%
VeachAjarAnimated	+4.409%	-2.862%	+0.9828%	+0.7765%

VeachAjar

VeachAjarAnimated

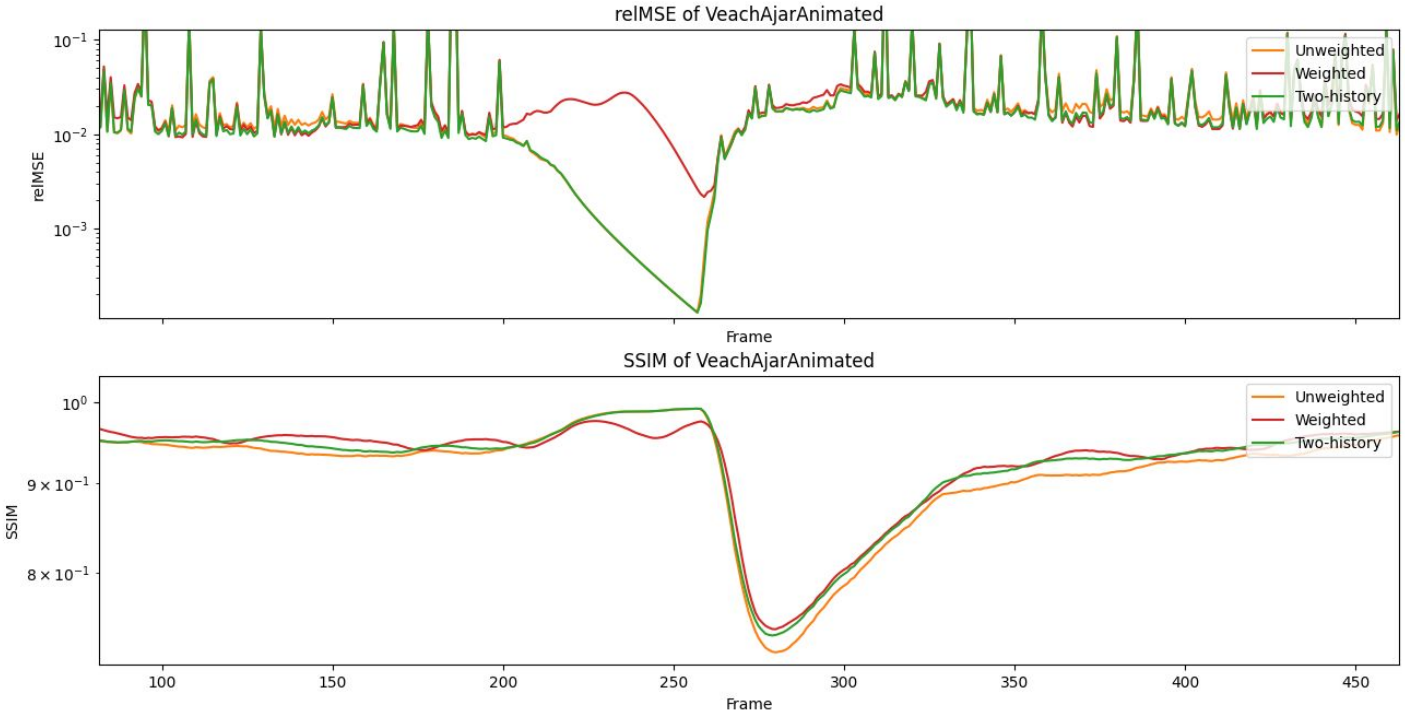
Experiment - Foveated Rendering

	relMSE		SSIM	
Scene	Weighted	Two-history	Weighted	Two-history
VeachAjar	-6.722%	-5.015%	+1.4142%	+1.0833%



Experiment - Foveated Rendering

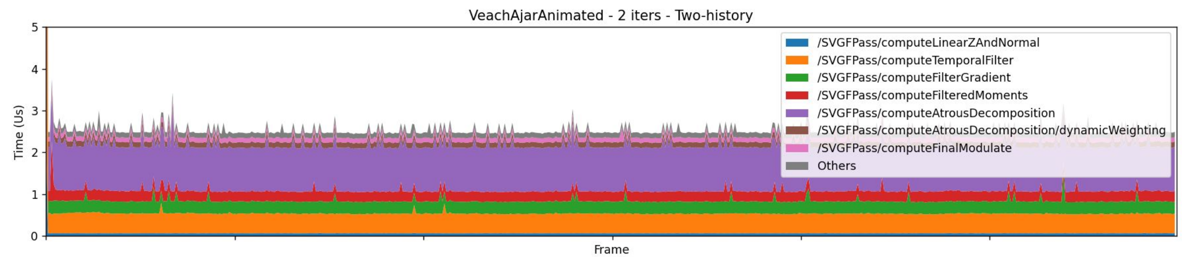
	relMSE		SSIM	
Scene	Weighted	Two-history	Weighted	Two-history
VeachAjarAnimated	+4.409%	-2.862%	+0.9828%	+0.7765%



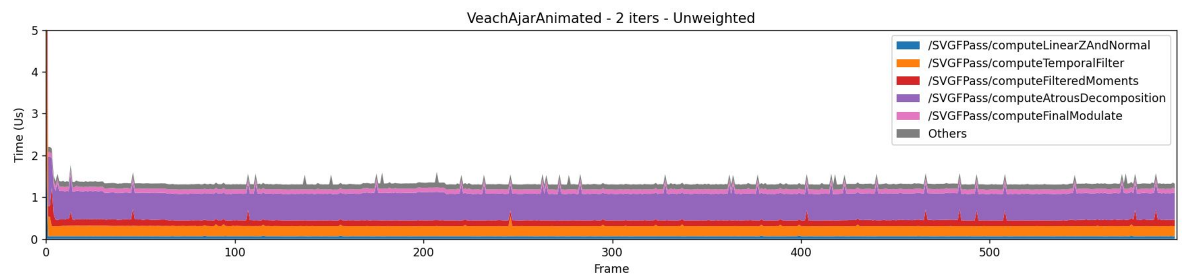
Execution Time

The two-history SVGF takes about 1.2ms longer than the original SVGF at 720p resolution on an RTX 3070 Ti GPU, which is small amount of time with comparison to those time can be saved by foveated rendering.

Two-history
2.528 ms
(+1.192 ms)



Unweighted
1.336 ms



(2 spatial-filter iterations, feedback/blending after 1st iteration)

Execution Time

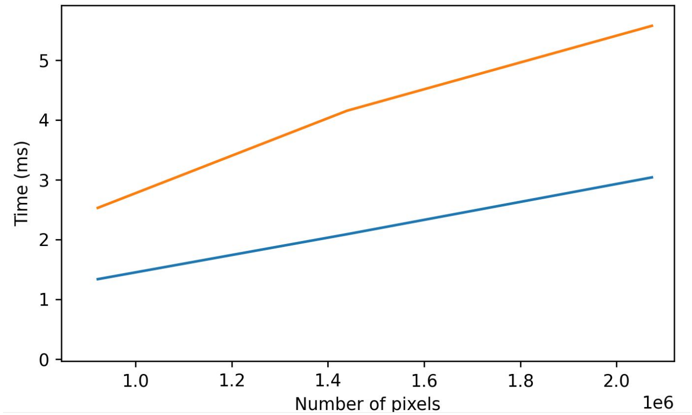
Process	Original SVGF	Two-history SVGF
LinearZAndNormal	0.065 ms	0.065 ms
TemporalFilter	0.245 ms	0.471 ms
FilterGradient	-	0.299 ms
FilteredMoments	0.145 ms	0.250 ms
AtrousDecomposition	0.650 ms	1.079 ms
DynamicWeighting	-	0.130 ms
FinalModulate	0.105 ms	0.106 ms
Others	0.126 ms	0.129 ms
Total time	1.336 ms	2.528 ms

Execution Time

Usually our method cost no more than twice the time cost by the unweighted method.

Resolution	Pixel Count	Time (unweighted)	Time (Two-history)
1280 x 720	921,600	1.336 ms	2.528 ms
1600 x 900	1,440,000	2.088 ms	4.153 ms
1920 x 1080	2,073,600	3.038 ms	5.572 ms

(running on RTX 3070 Ti GPU)



Extra Memory Usage

By our modifications, the following extra buffers are needed:

- `float3   curWeightedIllumination, prevWeightedIllumination`
- `float2   curTotalWeights, prevTotalWeights`
- `float3   curGradient, prevGradient`
- `float3   curSampleCountPowerSums, prevSampleCountPowerSums`

Total cost = W x H x 22 x sizeof(float) = W x H x 88 bytes

Resolution	**Memory**
1280 x 720	81.1 MB
1920 x 1080	182.5 MB
3840 x 2160	729.9 MB
2064 x 2208 x 2 (Quest 3)	802.1 MB

This amount of memory is affordable to most of the morden GPUs on desktop.

However it might be a little too costly to VR devices.

- Introduction
- Related Works
- Proposed Method
- Experiments
- Conclusion
- Limitaions & Future Work

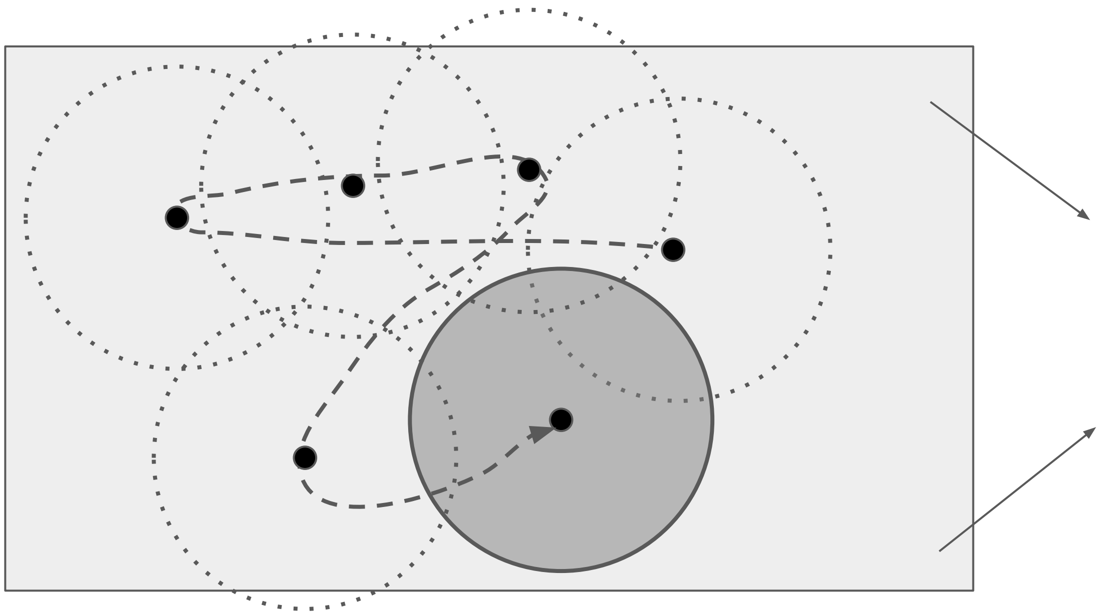
Conclusions

- We observed that previous spatio-temporal filters, which use exponential moving average for temporal accumulation, failed to put weight in colors generated by high sample count.
- To utilize colors generated by more samples, we introduced the "*Two-history Approach*" to adaptively blend between weighted and unweighted histories.
- When artifacts from weighted method are less significant, our method utilize it to reduce noise.
- When artifacts from weighted method are more significant, our method performs similar to unweighted method.
- Our aproach requires 88 bytes per pixel additional memory and usually cost no more than twice the time cost by the unweighted method.

- Introduction
- Related Works
- Proposed Method
- Experiments
- Conclusion
- Limitaions & Future Works

Limitations

- For areas that spp rarely change, our method has nearly no effect but still consume extra computing power.



spp
rarely
change

Limitations

- The extra memory cost can be a little costly to VR devices.
  (For example, Quest 3 has only 6GB GPU memory)

Resolution	Memory
1280 x 720	81.1 MB
1920 x 1080	182.5 MB
3840 x 2160	729.9 MB
2064 x 2208 x 2 (Quest 3)	802.1 MB

Future Works

Dealing with limitations

- Save computing resource when spp rarely change.
- Reduce extra memory cost.

More Applications

- Apply our method to other algorithms such as ReSTIR.
- Maybe we can reduce temporal overblurring by adaptively blending between histories with different decay rates.

Thanks for listening