Topic: Filtering methods in Real-Time Rendering

KAI YANG, Sorbonne Université

The path tracing rendering algorithm has long been considered to be unsuitable for real-time rendering, since a large amount of samples is required to produce noise-free renders. In real-times rendering, we are able to sample only several, or even just one or two rays. So one idea is that we render firstly a rough image with many noises, then we use a filtering methods to get a high quality and high resolution image.

Many filtering methods have been proposed, which denoise renders by trading variance for bias. By the inspiration of the article *Gradient Estimation for Real-Time Adaptive Temporal Filtering* [Schied et al. 2018], which talks about an adaptive temporal filtering, I want to explore the evolution of the filtering methods, and discuss the principle.

Additional Key Words and Phrases: A-Trous Wavelet, Temporal anti-aliasing, SVGF, A-SVGF, Filtering

1 INTRODUCTION

When talking about the filtering, the most classic operation that we usually do in computer vision, is a Gaussian convolution. However, a simple Gaussian filtering will blur the whole image. So we distinguish the edges, and apply the smooth filtering method locally. Thus, the first method that I learnt is *A-Trous* [Dammertz et al. 2010].

The previous method runs in just one image, and it has just the information of only one image. Fortunately, a video is a series of images, so we start trying to combine the information of two connected frames. So the second method that I learnt is *Temporal Anti-Aliasing* [Yang et al. 2020].

Then I learnt *Spatiotemporal Variance-Guided Filter* [Schied et al. 2017], which is the basic and mean idea of the article that I chose in this project.

And finally, the topic of the article that I chose: *Adaptive Temporal Filtering* [Schied et al. 2018].

2 A-TROUS FILTER

2.1 Joint Bilateral Filtering

Classic Gaussian blurring averages each pixel indiscriminately, so it causes the entire image to be blurred out, as shown in Figure 1.

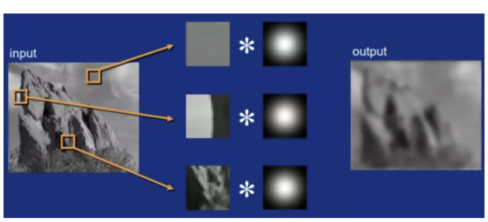


Fig. 1. Same Gaussian kernel everywhere

This report is submitted as a part of project for Advanced 3D Computer Graphics (IMA904/IG3DA), Telecom Paris.

The original work is introduced by [Schied et al. 2018].

Bilateral filtering takes into account the variation of image pixels and performs differential blurring, e.g. blurring lower at the boundaries of image features. The figure 2.

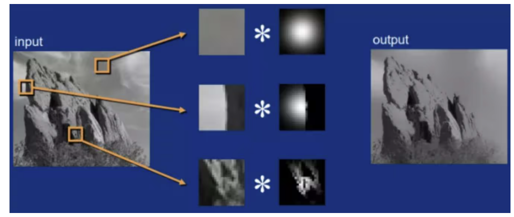


Fig. 2. The kernel shape depends on the image content

Joint bilateral filtering is to apply joint filtering to 3D scenes. She not only considers the mutation of pixels, but also uses the normal vector of 3D space, the 3D coordinate information of pixels, which are recorded in **G-Buffer**, so the filtering is much more accurate. If we look at the figure 3, the boundary between A and B is distinguished by depth, B and C is by normal vector, and D and E is distinguished by color value.

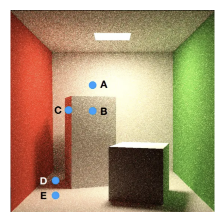


Fig. 3. Scene

2.2 Acceleration based on A-Trous Wavelet

The way that A-Trous use for acceleration is Multiple filtering with a fixed-size filter, then increasing the filter sample interval each time. The figure 4 gives an example.

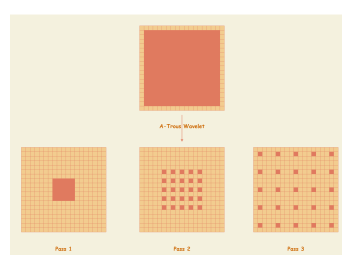


Fig. 4. A-Trous wavelet

2.3 Result

This filtering technique is effective and computationally small, but at the cost of a lossy scene representation and the loss of high-frequency information such as sharp edges. This loss of information is so severe, as shown in the figure 5, that it can even lead to differences in brightness levels in processing highlight or shadow patterns with pretzel noise.

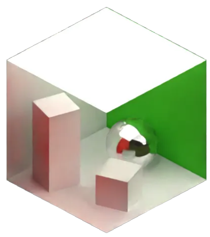


Fig. 5. Result of A-Trous

3 TEMPORAL ANTI-ALIASING

To solve the problem of the loss of high-frequency information, and avoid the aliasing. In fact, this is a up-sample method, and we can increase sampling rate to get a better image, however real-time rendering do not allow to do so.

The core idea of Temporal anti-aliasing technique is to apportion the operation to multiple frames, as shown in Fig. 6, only one pixel point is computed in any frame, and as t frames accumulate, there are t sampled pixels, and then mixing the results is equivalent to doing t sub-pixels of oversampling. By spreading the oversampling of pixels over multiple frames, the sampling efficiency is improved to achieve the purpose of mitigating jaggies.

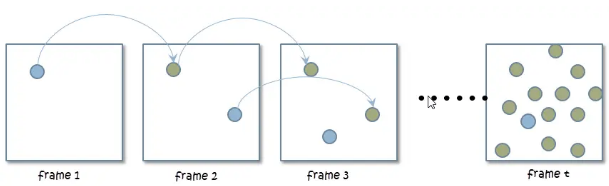


Fig. 6. apportion the operation to multiple frames

In fact, the algorithm will use only two frame to save time. If the pixel value of the current frame is $s_t(p)$, and the historical frame pixels of the previous t-1 frames are $f_{t-1}(\pi_{t-1}(p))$, weighted mixing of them gives the result obtained by apportioning multi-frame oversampling as shown in Equation 1.

$$f_t(p) = \alpha\ s_t(p) + (1-\alpha)\ f_{t-1}(\pi_{t-1}(p)) \tag{1}$$

In practice, the frame is not completely static, so we need to calculate the position of the pixels in the current frame in the history frame, this process is called Re-projection. If only the lens is moving, then the position of the history frame can be calculated from the transformation matrix of the lens. But if the model itself is also moving, we need to calculate the difference between the positions of the model vertices in the current frame and the history frame, which is called Motion Vector. Re-projection is the process of deriving the position of pixels p in the current frame on the screen in the corresponding position of the history frame $\pi_{t-1}(p)$ based on the view projection matrix of the current frame and the history frame and the Motion Vector.

3.1 Result

This method works well in anti-aliasing, however, when the object moves, some pixels in history frame will be hidden, so the information of these sample is useless information, and should be ignored. TAA can't Detect invalid samples, and Ghosting can occur (figure 7) if history frame pixels are not corrected.

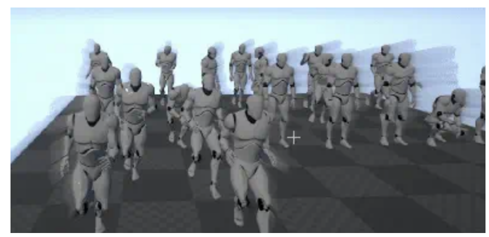


Fig. 7. Ghoasting

4 SPATIOTEMPORAL VARIANCE-GUIDED FILTERING

The core idea of SVGF to deal with very low number of samples is to combine temporal and spatial information together to do Filter, because each frame has only one coloring sample per pixel, we can only distribute the samples of each pixel uniformly in time and try to use the information of the past few frames in the Filter of the current frame. At the same time, the information from past frames can also provide an estimate of the variance of the sample distribution in the current pixel region, and this variance information affects the spatial Filter.

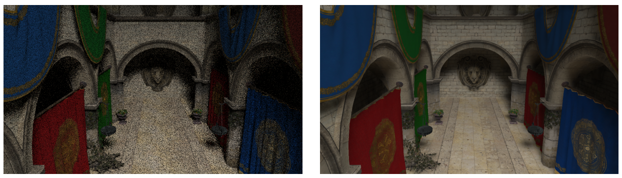
In fact, the filter in temporal information is similar to Temporal Anti-Aliasing, which will use the equation 1 to combine two frames. The filter in spatial information is similar to Joint Bilateral Filtering, which will use the variance to detect the edges. If the accumulating Luminance's First and Second Moments in time is μ'_1 and μ'_2, the variance of luminance is computed as equation 2.

$$\sigma_i'^2 = \mu'_{2i} - \mu_{1i}'^2 \tag{2}$$

So the basic process for reconstruction filtering is to use Exponential Moving Average to accumulate the color information of pixel Irrdiance to increase the number of valid samples per pixel at 1SPP, and at the same time to accumulate the First and Second Moment of Luminance to estimate the variance of each pixel in time. The variance size is then used as an input to the Edge Stopping function in the subsequent Spatial Filter to better guide the strength of the Spatial Filter in different noise regions.

4.1 Result

For a stable scene, this method can reconstruct a clear image from a very low number of samples as shown in fig 8. However, when the movement is fast, it is faced to the same problem like TAA: ghosting. And also a temperal lag, as shown in 9(a).



(a) With noising (b) denoised

Fig. 8. Result of SVGF

5 ADAPTIVE SPATIAL-TEMPORAL VARIANCE GUIDED FILTERING

A-SVGF improves SVGF by adaptively re-projecting previous samples spatially based on temporal features (e.g., changes in variance, viewpoint, etc.), encoding them in a Moment Buffer, and filtering them by a fast bilateral filter. Thus, unlike accumulating samples based on the length of history, changes in variance are used to determine the ratio of old samples to new samples, thus reducing ghosting ghosting.SVGF uses only the moment buffer for blurring, whereas A-SVGF performs both filtering and accumulation.

The core idea is that we adjust the time accumulation factor (α) in equation 1 pixel by pixel, the adjustment rule follows the equation 3, where $\lambda(p)$ is a value computed from the dense normalized history weights of pixels p, which is defined by combining the temporal gradient of the reconstruction $\delta_i(p)$ and an additional normalization factor $\Delta_i(p)$, as shown in equation 4. Here $\Delta_i(p)$ represents the maximum value of the absolute value of the signal change, and is used to control the time filter's focus on the relative rate of change.

In this way, the time accumulation factor is able to respond quickly when sudden changes occur, while using more aggressive temporal filtering in regions where the signal is stable.

$$\alpha_i(p) = (1 - \lambda(p)) \cdot \alpha + \lambda(p) \tag{3}$$

$$\lambda(p) = \min\left(1, \frac{\delta_i(p)}{\hat{\Delta}_i(p)}\right) \tag{4}$$

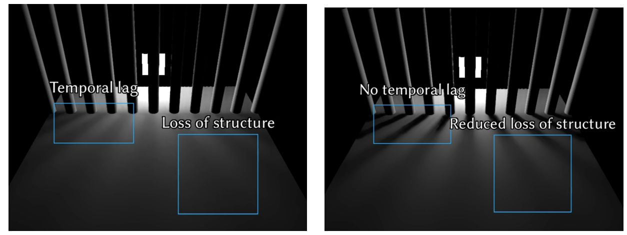
This adaptive approach allows for better handling of abrupt changes in the scene, providing a balance between reducing temporal lag and maintaining stability.

Figure 9 shows the comparison of SVGF and A-SVGF.

6 FURTHER WORK

6.1 ReSTIR

Spatiotemporal Importance Resampling for Many-Light Ray Tracing (ReSTIR) [Bitterli et al. 2020] attempts to bring forward the spatio-temporal reprojection step of the real-time denoiser at rendering time, reusing statistical information about the probability of neighboring samples. Essentially a combination of earlier papers discussing Resampled Importance Sampling and adding the idea of spatio-temporal denoising.



(a) Temperal lag and Ghosting in SVGF (b) Result of A-SVGF

Fig. 9. Comparison between SVGF and A-SVGF

6.2 Mahine Learning

Machine learning techniques such as noise-reducing autoencoders, sample map estimator-driven sample counting or significant sampling, neural bilateral lattice filtering, and oversampling, although these techniques are slower than other algorithms such as A-SVGF, they offer the most significant improvements in image quality.

7 CONCLUSION

In this project of IG3DA, I learnt the mean idea of real-time rendering for Ray Tracing Denoising, and a lot of methods to achieve Ray Tracing Denoising.

From traditional Monte Carlo methods to brand new applications of deep learning, each step is inseparable from a deeper understanding of the theory and algorithms, as well as creative solutions to practical problems.

We're moving towards real-time ray tracing gradually.

REFERENCES

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 (TOG)* 39, 4 (2020), 148–1.

Holger Dammertz, Daniel Sewtz, Johannes Hanika, and Hendrik PA Lensch. 2010. Edge-avoiding a-trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics*. 67–75.

Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. 2017. Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics*. 1–12.

Christoph Schied, Christoph Peters, and Carsten Dachsbacher. 2018. Gradient estimation for real-time adaptive temporal filtering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 2 (2018), 1–16.

Lei Yang, Shiqiu Liu, and Marco Salvi. 2020. A survey of temporal antialiasing techniques. In *Computer graphics forum*, Vol. 39. Wiley Online Library, 607–621.