

Cache and Bandwidth Aware Real-time Subsurface Scattering

PhD candidate:

Tiantian Xie

Advisor:

Dr. Marc Olano

Committee Members:

Dr. Adam Bargteil

Dr. David Chapman

Dr. David Hill

Dr. Matthias Gobbert

21/06/2021 Tiantian Xie PhD dissertation defense

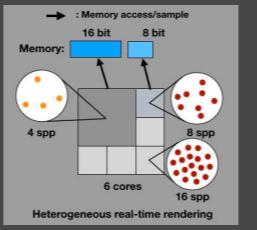
Metahuman rendered in UE4
with dissertation algorithm

Hi everyone. Thanks for coming to my phd defense. I am excited to share what I have found during my PhD study. Thanks to my advisor Dr. Marc Olano and all committee members for providing me feedback and suggestions. And thank Epic Games for supporting my research and allowing me to use part of the content. Like the background rendering from MetaHuman rendered with my dissertation algorithm. In this dissertation, I explore novel algorithms to deal with cache incoherence and bandwidth problem to improve the quality and performance of subsurface scattering.

Outline

Section I: Chapter 1 ~ Chapter 3

- Introduction
- Literature
- Motivation
- Heterogeneous Real-time Rendering



Section II: Chapter 4 (I3D'20)

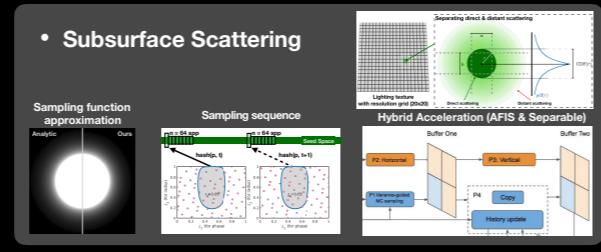
- Real-time Adaptive Sampling $O(1)$

Frame i-1 Frame i



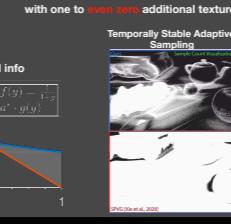
Section III: Chapter 5

- Subsurface Scattering



Section IV: Chapter 6 (I3D'21)

- Real-time Control Variates



2

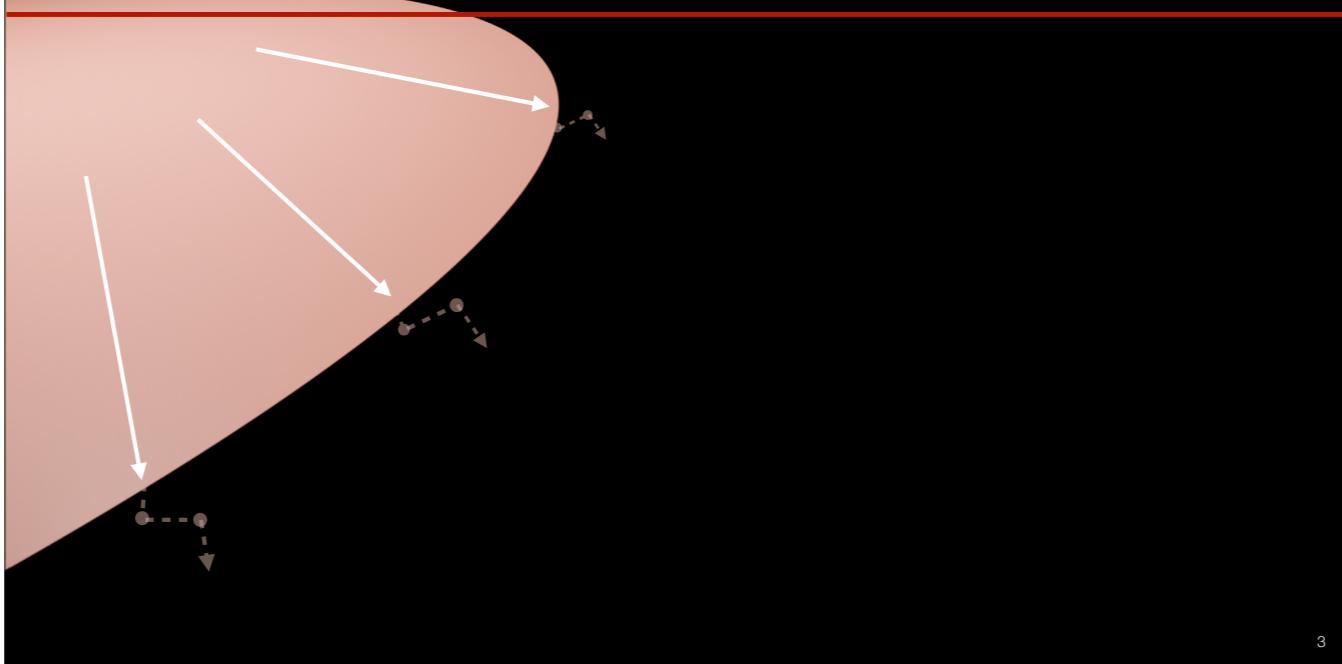
This presentation is organized in four sections. The first section introduces subsurface scattering, the literature review and the motivation for our work. Finally we introduce a taxonomy for heterogeneous real-time rendering algorithms that is cache and bandwidth aware. I hope it is already available to me when I start learning real-time rendering.

In section two, a novel algorithm to deal with bandwidth and cache issue is introduced, real-time adaptive sampling. Moreover, the sample count in each frame can be estimated with an algorithm complexity of $O(1)$. It makes adaptive sampling possible without perceivable noise and little overhead for real-time rendering.

In section three we provide more details about how the subsurface scattering pass is designed and different problems are solved. Like how we approximate the sampling function, how the real-time sampling sequence is designed, how we combine adaptive sampling with filtered importance sampling (adaptive filtered importance sampling, AFIS) to push real-time subsurface scattering to the state-of-the-art for both quality and performance. SPVG is the algorithm name for section II, our I3D'20 publication.

In the last section, we deal with the challenge task of dynamic lighting with a novel usage of control variates for real-time adaptive sampling. We are able to achieve temporally stable adaptive sampling under all conditions.

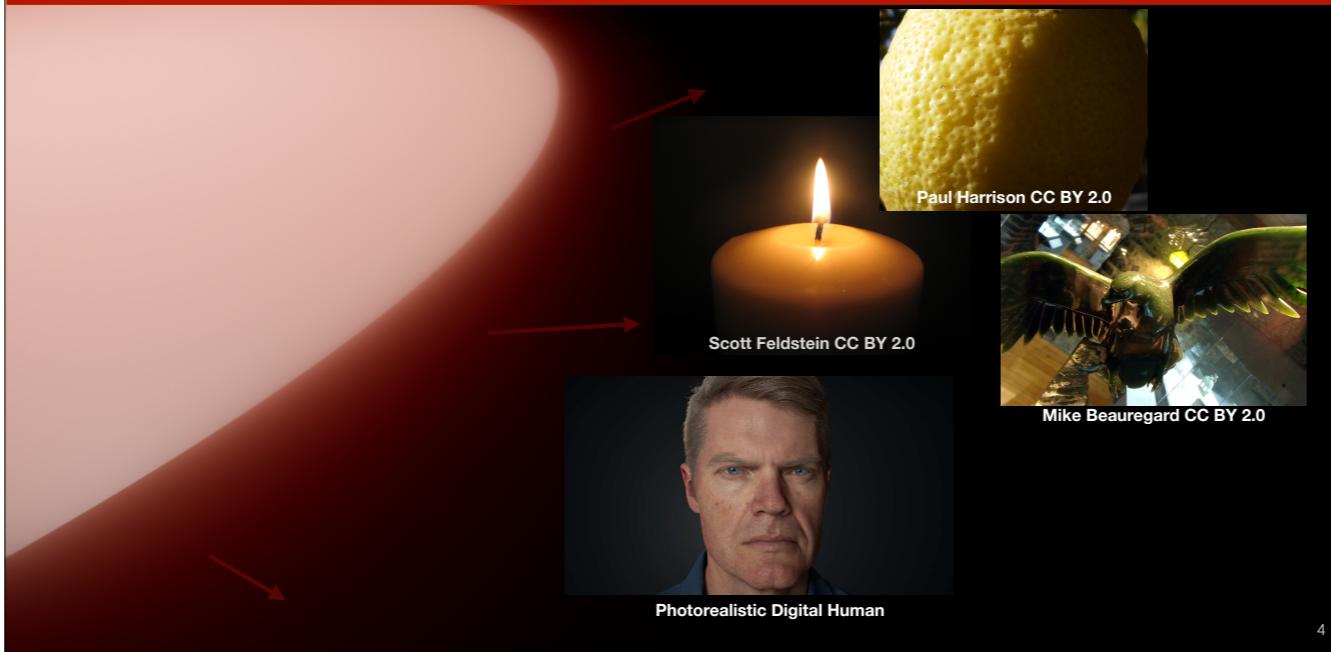
Subsurface scattering



3

When light shades on a surface, it bounces into the surface and bounces out some where else to create a soft look. This effect is subsurface scattering.

Subsurface scattering



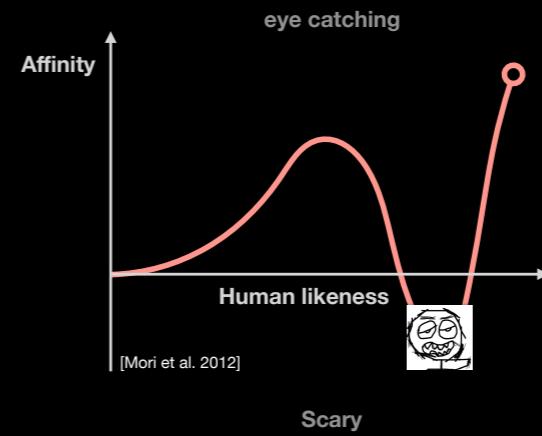
4

There are a lot of beautiful subsurface scattering effects in real world that we enjoy, like candle, fruit, and jade. But what we care most in real-time rendering, especially in game, is photorealistic digital human. We long for such a technique to let those characters not only just live in trailers, but in actual game plays in action!

Why we care



- Escape uncanny valley
- Improve the quality
- Performance drops!
- Real-time rendering (e.g., 60fps)
- Performance
- Performance!
- Performance!!



5

So why we care about 'real' photorealistic digital human. Because we want to escape the uncanny valley. This diagram shows the relationship between the human likeness and affinity.

As an artificial character becomes very close to human, it becomes scary. We need to improve the quality to escape the uncanny valley, but at the cost of performance.

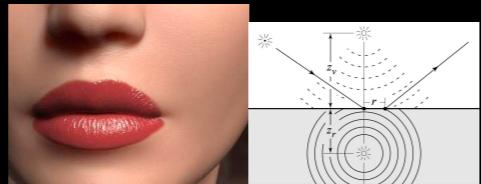
You know, to achieve real-time rendering performance like 60 fps, there are three very important challenges to deal with. Performance, Performance!, Performance!!

Real-time subsurface scattering - Literature review

2001

Offline

Dipole diffusion profile

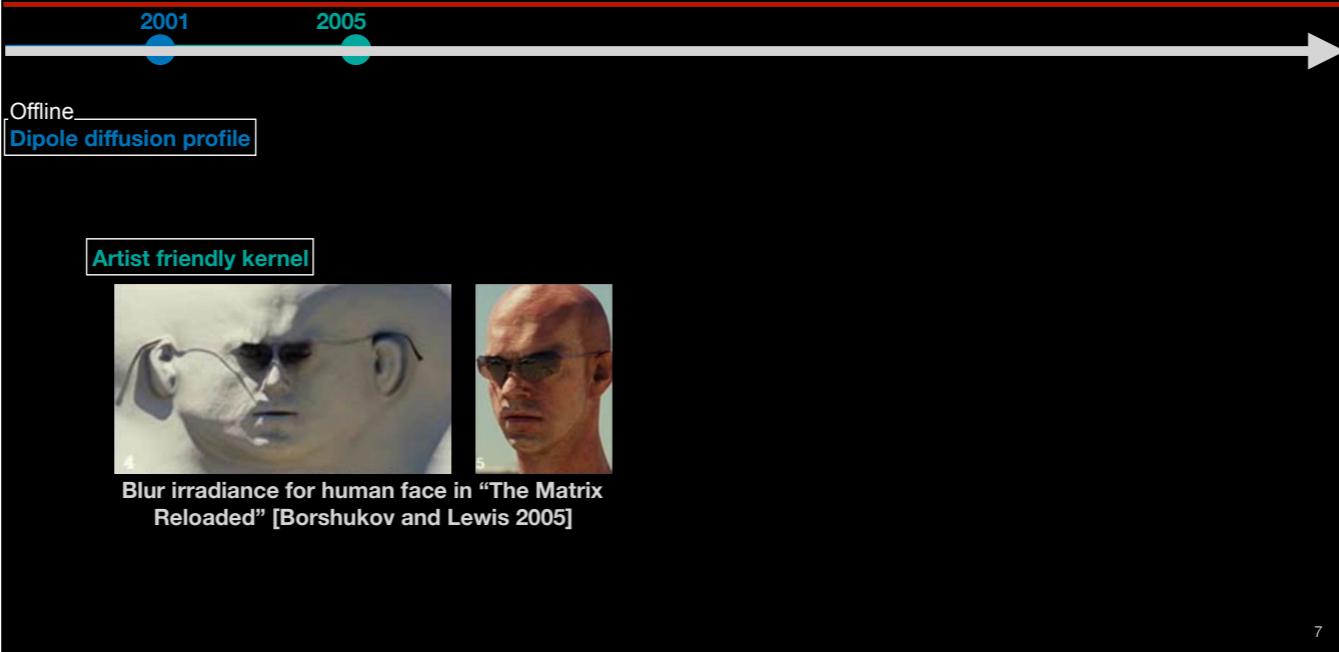


Use dipole to estimate where to scatter out [Jensen et al. 2001]

6

[Jensen et al. 2001] considers isotropic subsurface scattering in highly scattering media, they introduce dipole diffusion profile into the field of computer graphics for subsurface scattering in 2001. The profile allows compact representation and efficient evaluation for realistic subsurface scattering however, for offline renderers only.

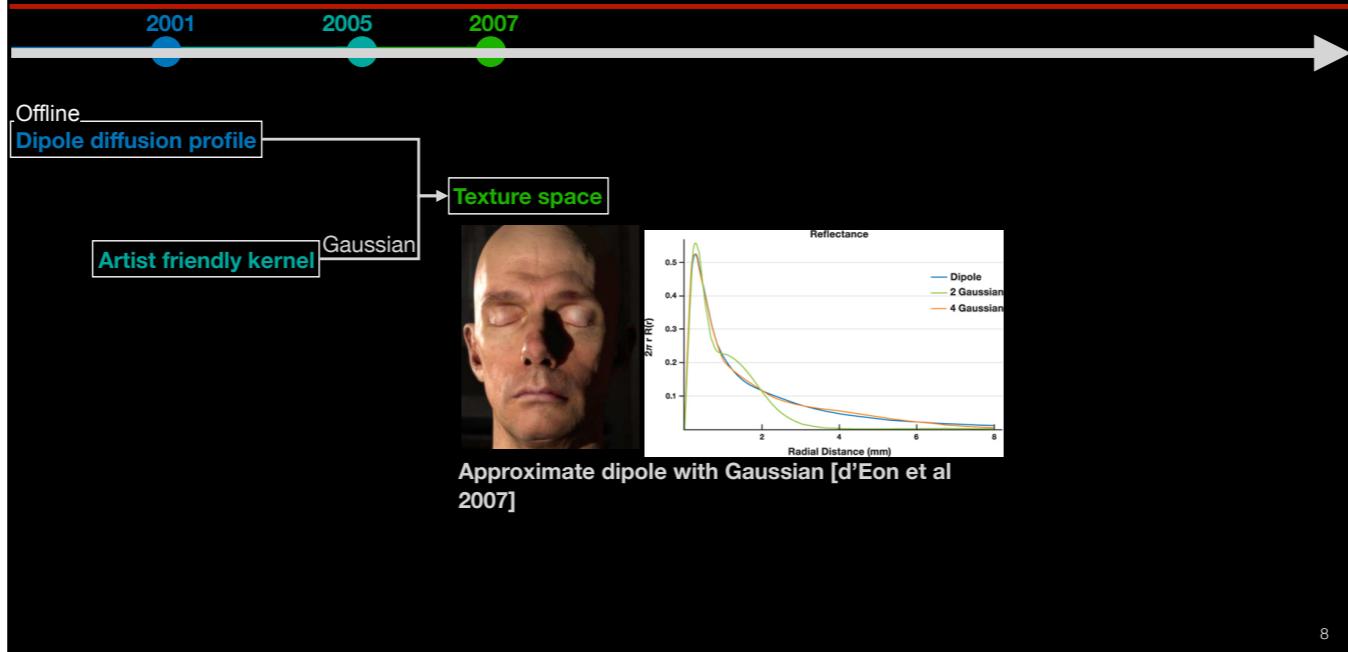
Real-time subsurface scattering - Literature review



What sparks the idea of real-time realistic subsurface scattering is Borshukov and Lewis's work in 2005. Instead of expensive techniques, like path tracer or volumetric scattering simulation. They use artist friendly kernel to blur the gathered irradiance texture to approximate subsurface scattering for human face rendering for the movie "The matrix Reloaded".

(This technique is much simpler and faster)

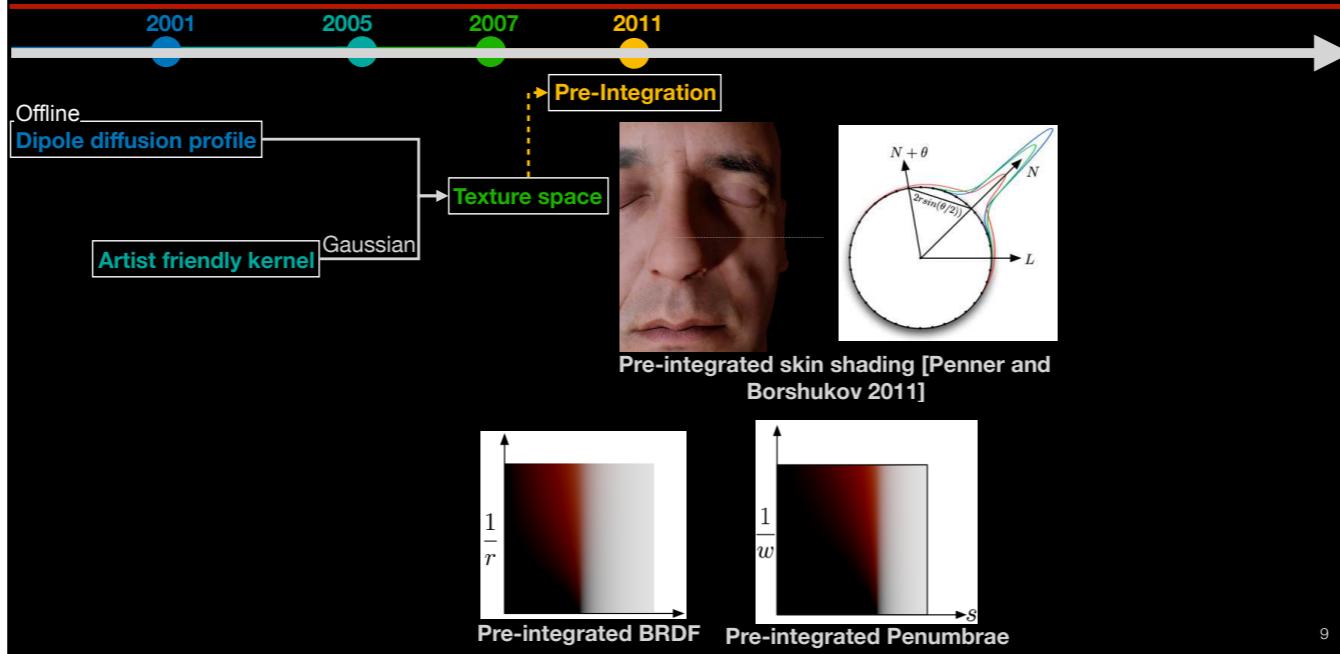
Real-time subsurface scattering - Literature review



8

The work by d'Eon and his colleague [d'Eon et al 2007] brings the realistic part of dipole diffuse profile, and the real-time part of using gaussian filters to approximate dipole, together into texture space. It created the first realistic and real-time subsurface scattering for human skins that matched an offline renderer in the NVIDIA human head demo.

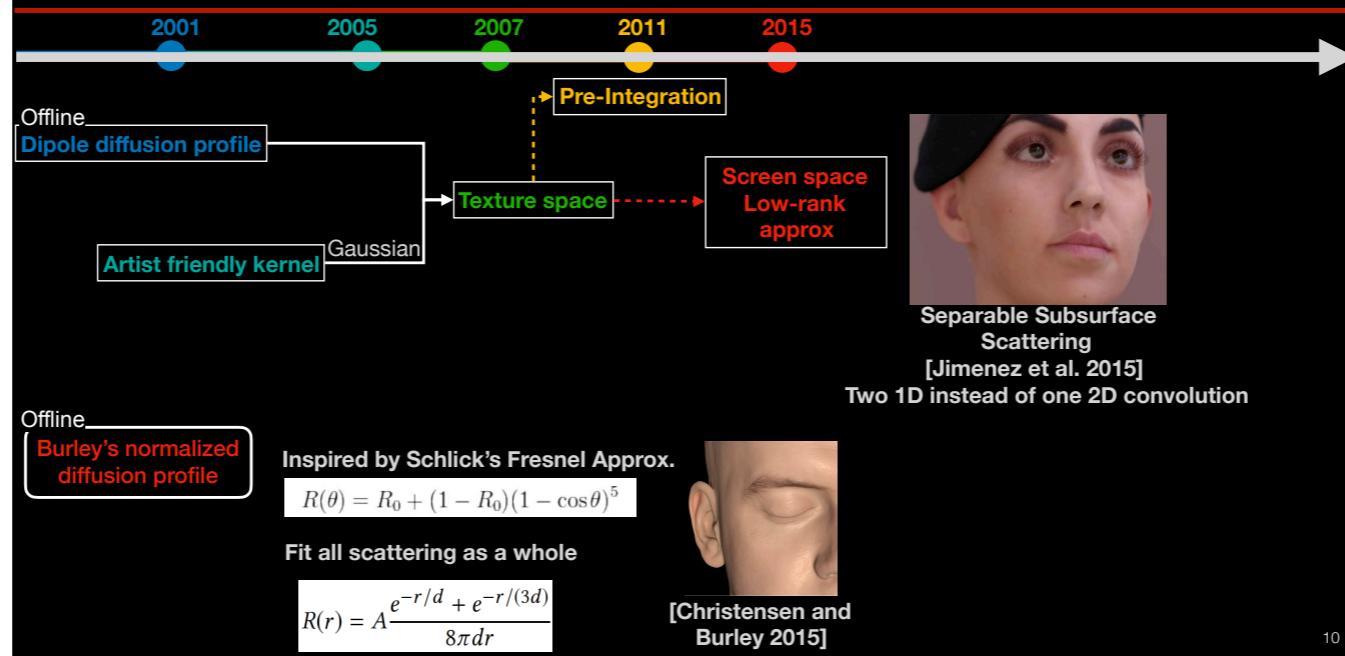
Real-time subsurface scattering - Literature review



9

Eric Penner, and George Borshukov[Penner and Borshukov 2011] moves in another direction by offloading online gaussian sampling into Pre-integrated textures (BRDF for lighting, and Penumbrae /pe'nembre/ for shadowing). During shading, ‘Simple’ texture lookup based on curvature and shadow parameters (shadow value and size of penumbra) is enough

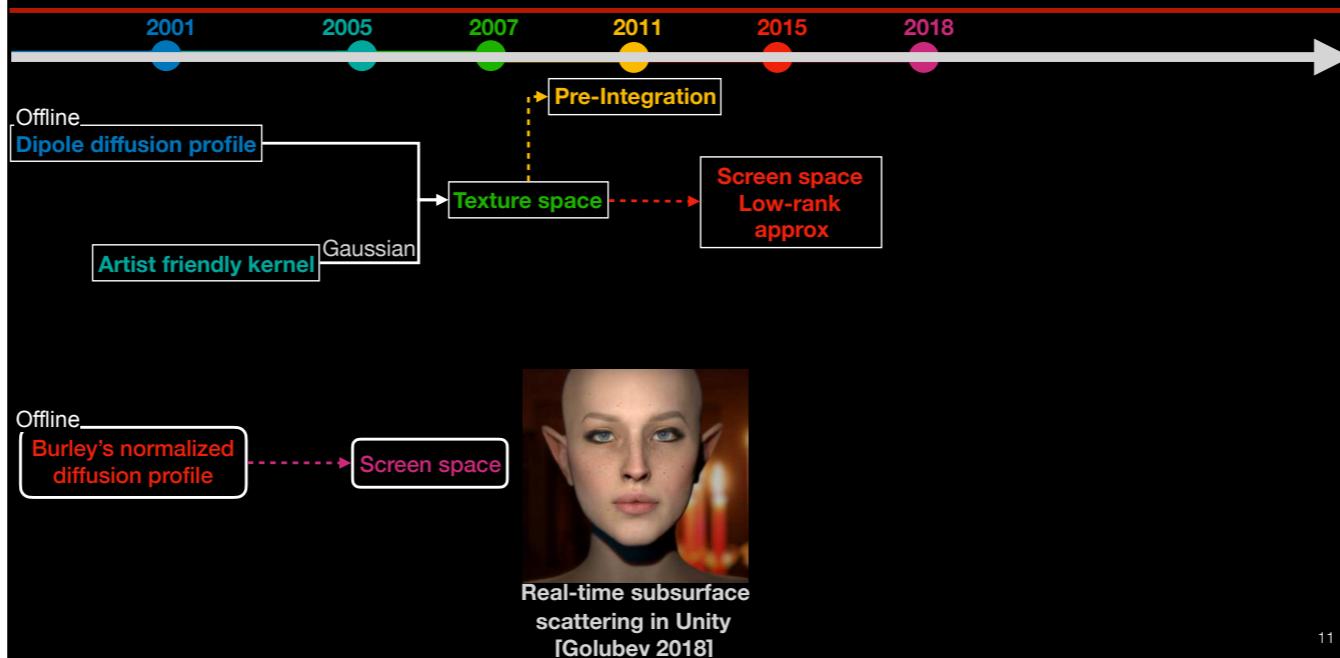
Real-time subsurface scattering - Literature review



In 2015, the work by Jimenez and his colleague [Jimenez et al. 2015] introduces low rank approximation in screen space to further accelerate the realistic real-time subsurface scattering. For example, they apply two 1D convolution instead of one 2D convolution.

In the mean while. Inspired by Schlick's Fresnel approximation, Christensen and Burley proposed the Burley's normalized diffuse profile that fits all scatterings (single and multiple scattering) as a whole. Which is more accurate than gaussian approximation of dipole.

Real-time subsurface scattering - Literature review



In 2018, Golubev brought Burley's normalized diffuse profile into screen space and uses the simple and physically correct control parameters (dmfp) than dipole based techniques. Due to the expense of brutal force MC sampling of the profile, pre-calculated sampling and pattern rotation is applied to avoid real-time sampling. However, it's still not fast enough for real-time applications.

Motivation

- Photorealistic subsurface scattering without uncanny valley.
 - Monte Carlo estimation
 - Random memory access
 - Cache incoherence
 - High bandwidth demand
 - Question: What novel techniques can be used to target on high quality photo-realistic real-time rendering with the contemporary and next-generation hardware?

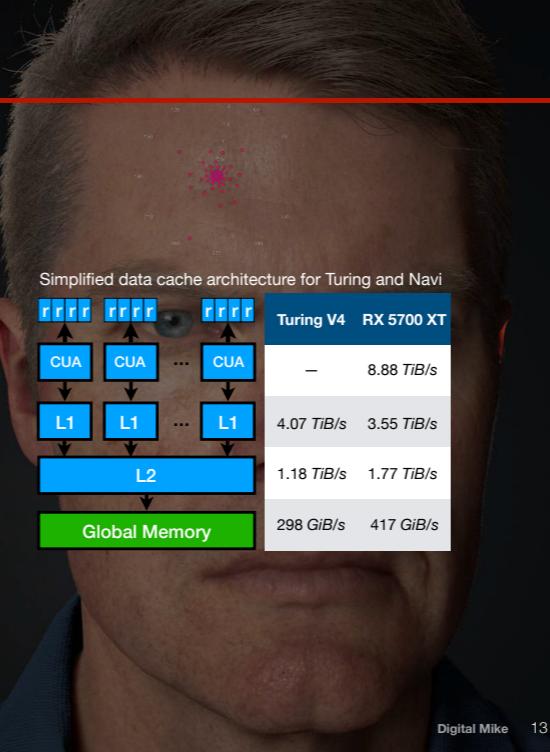


Digital Mike 12

To achieve photorealistic subsurface scattering. It is essential to have Monte Carlo sampling. However, it is extremely expensive. Because it performs random memory access. This is not friendly to the current cache architecture design. It will result in cache incoherence.

Motivation

- Photorealistic subsurface scattering without uncanny valley.
- Monte Carlo estimation
- Random memory access
- Cache incoherence
- High bandwidth demand
- Question: What novel techniques can be used to target on high quality photo-realistic real-time rendering with the contemporary and next-generation hardware?



Here we show the raw bandwidth converted to Byte/s of two GPU and the simplified data cache architecture for Turing and Navi. Cache incoherence access pattern does not have good spatial or temporal patterns. It requires high bandwidth demand to higher cache levels.

Taxonomy

- Cache and bandwidth aware real-time rendering
- Demands
 - Computing
 - Sample
 - Memory
- Heterogeneity

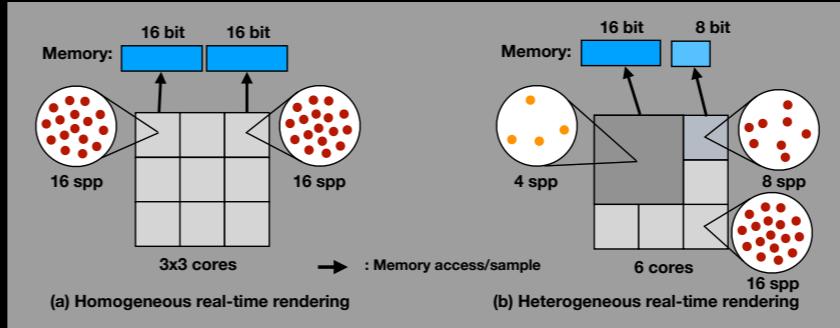


Figure 1: Comparison between (a) homogeneous and (b) heterogeneous real-time rendering. Heterogeneous real-time rendering can dynamically reduce the sampling, shading units, and memory demands without noticeable quality degradation.

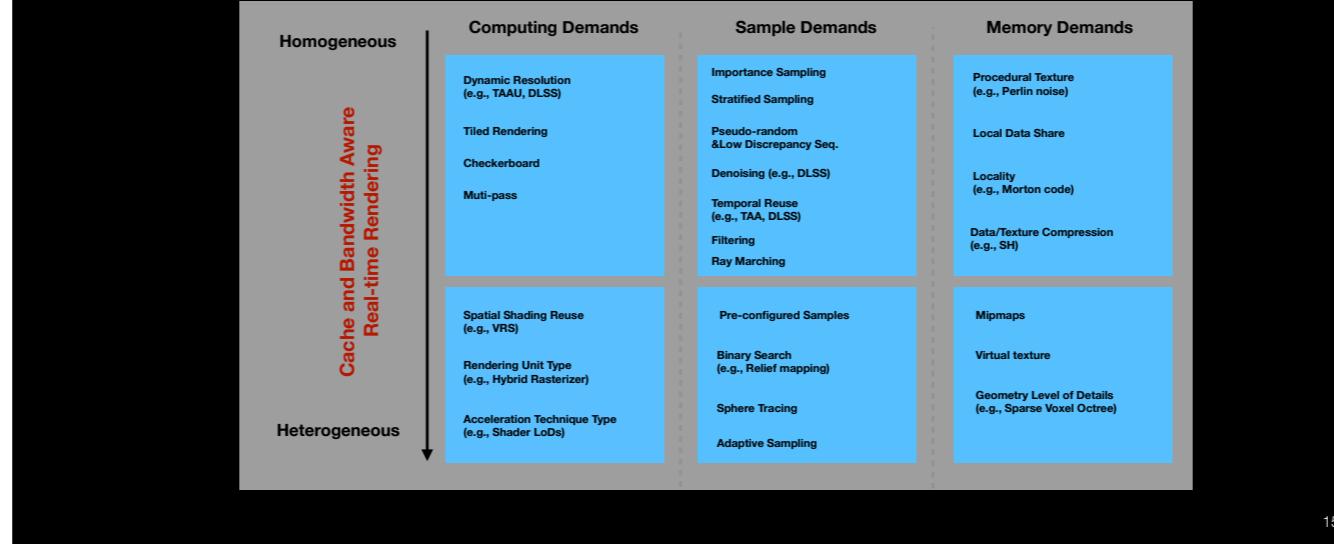
14

Then I started to survey on what I could use. One part is to help the algorithm design, another part is find where my research actually belongs, and how my research fit in academia. Then I find out there is not such a taxonomy to help me organize the techniques that have already been used in real-time rendering. I hope I have already seen such a taxonomy when I start learning real-time rendering.

I got two dimensions for the taxonomy. First is demand, the computing, the sample and the memory demands. The second dimension is heterogeneity, whether we need homogeneous or heterogeneous demands everywhere. As an example, I present a model for homogeneous real-time rendering and heterogeneous real-time rendering. In homogeneous real-time rendering, each pixel uses one core, a fixed 16 spp is used, and each sampling access the same number of bits. For heterogeneous real-time rendering, all computing, sample and memory demands can be variable without noticeable quality degradation.

Taxonomy

- Cache and bandwidth aware real-time rendering



15

Here I fit existing real-time rendering techniques into this taxonomy. For example, Dynamic resolution is a homogeneous computing demands minimization technique, while VRS is a heterogeneous way to reduce computing demands. Importance sampling is a homogeneous way to reduce sample demand, while adaptive sampling can place different samples for different pixels. Procedural texture is a great homogeneous way to reduce memory bandwidth, while accessing different mip levels with mipmaps is a way to reduce demands heterogeneously. Actually, I have done a little bit more for this taxonomy as planned in the proposal.

Taxonomy

- Bandwidth demands cost function

1. Computing demands

$$T(n) = kT'_k(n/k) + O(n)$$

$$T'(n) = \sum_{i=1}^n \sum_{a \in \mathcal{A}, r \in \mathcal{R}} w_{i,a,r} S_{i,a,r} + O(n)$$

Subsurface Scattering

Ch. 5 Acceleration Technique Type

- AFIS
- Separable

$$\mathcal{A} = \{AFIS, Separable\}$$

2. Sample demands

$$S(m) = \sum_{j=1}^{m'} M(d_j) + O(m')$$

$$m' = f(m, \sigma_0^2), m' \leq m$$

Ch. 4 Real-time Adaptive Sampling (sample demands)

Ch. 6 Real-time Control Variates (demands stability)

$$f(\cdot)$$

3. Memory demands

$$M(d) = R(d) + O(d)$$

Ch. 5 Adaptive Filtered Importance Sampling (+ memory demands)

$$R(\cdot)$$

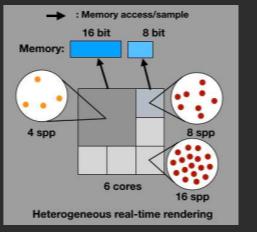
16

Formulation of bandwidth demands cost function for a task. This is also a mathematical way of describing the contribution of different chapters in the dissertation. To reduce computing demands, I proposed a combination of different acceleration technique types, AFIS and Separable. Sampling demands minimization is the major work in this dissertation. $S(m)$ is the total memory demands accumulated based on a sample count m apostrophe, which is determined by a technique function f , based on the sample budget m and a variance metric. Note that the f function is not mathematically determined but designed by the rendering engineer to minimize sample count. In my dissertation, I have proposed a novel real-time adaptive sampling technique to reduce the sample demands, and a novel usage of control variates to make it temporally stable under dynamic lighting conditions, which is very common in games. To reduce the memory demands, I proposed adaptive filtered importance sampling for subsurface scattering

Outline

Section I: Chapter 1 ~ Chapter 3

- Introduction
- Literature
- Motivation
- Heterogeneous Real-time Rendering



Section II: Chapter 4 (I3D'20)

- Real-time Adaptive Sampling

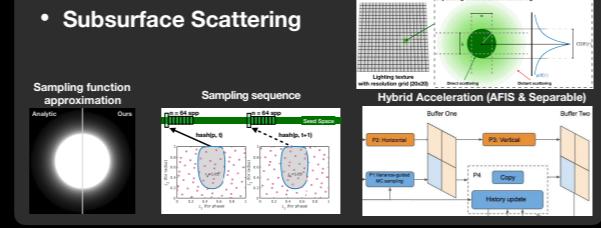
Sampling $O(1)$

Frame i-1
Frame i



Section III: Chapter 5

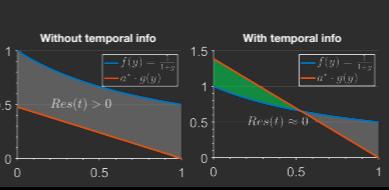
- Subsurface Scattering



Section IV: Chapter 6 (I3D'21)

- Real-time Control Variates

with one to even zero additional texture



17

That's the summary of the dissertation. Next, I will introduce the real-time adaptive sampling technique that minimizes the sample count per pixel.

Note: I3D (ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games) is one of the top conference for real-time rendering.

Real-time Adaptive Sampling

- Adaptively allocate samples to high variation regions.
- Subsurface scattering
 - High variation in lighting gradient change region
- Adaptive sampling procedural
 - Pass 1: Pilots (discarded)
 - Pass 2: Additional samples
- Related work
 - A priori (e.g., frequency or derivative)
 - A Posteriori (Monte Carlo sampling)



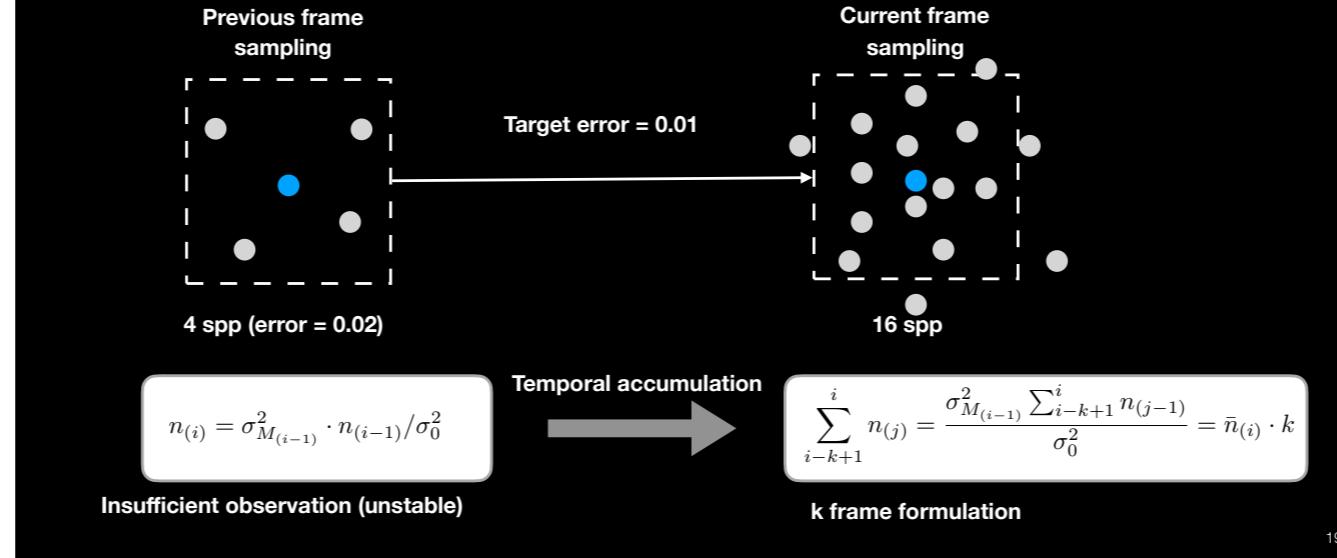
Sample count visualization in
greyscale on MetaHuman character ada 18

Adaptive sampling techniques adaptively allocate samples to regions with high variance. The reason we can apply adaptive sampling to subsurface scattering is that there is only high variance in high gradient change region like the ear, and silhouette region. There are usually two pass to perform adaptive sampling, in the first pass, some pilots are used to survey the variance. In the second pass, more samples are used based on this variance metrics. Usually the sample used in the first pass is discarded to avoid bias.

In the literature, there are two types of adaptive sampling, A priori that surveys the frequency or derivative of neighbor patches. A Posteriori that relies on Monte Carlo sampling itself to estimate the variance. This dissertation focuses on the second one.

Basic Metrics

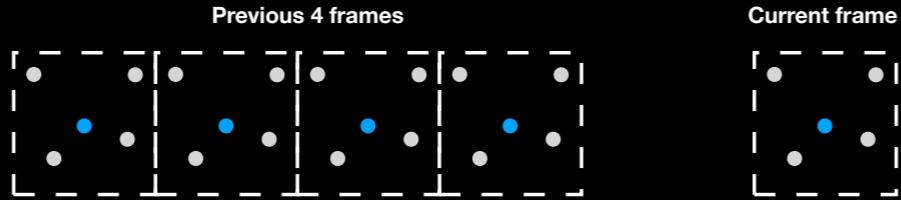
Introduction to basic adaptive sampling



The basic metric is based on the fundamental relationship between sample count and variance that the variance halves when the sample count doubles. Single frame observation is not stable. Instead, we can have temporal accumulation, use k frames instead.

Metrics with Temporal Accumulation

Adaptive sampling with sample history

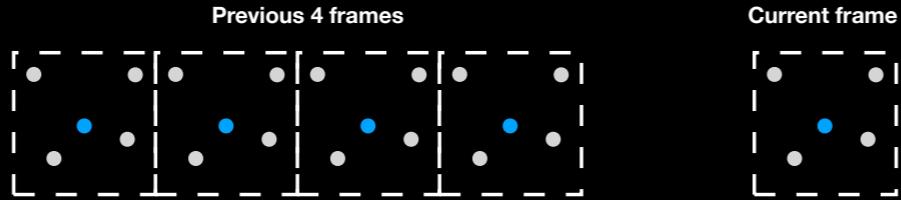


20

To resolve this issue, we could perform adaptive importance sampling with sample history. Here we can distribute the observation in the previous 4 frames. Then, we have sufficient observations to estimate the samples required in the current frame. And iteratively, we can move on for as long as we need it.

Metrics with Temporal Accumulation

Adaptive sampling with sample history



21

To resolve this issue, we could perform adaptive importance sampling with sample history. Here we can distribute the observation in the previous 4 frames. Then, we have sufficient observations to estimate the samples required in the current frame. And iteratively, we can move on for as long as we need it.

Metrics with Temporal Accumulation

- Use difference of temporal accumulation

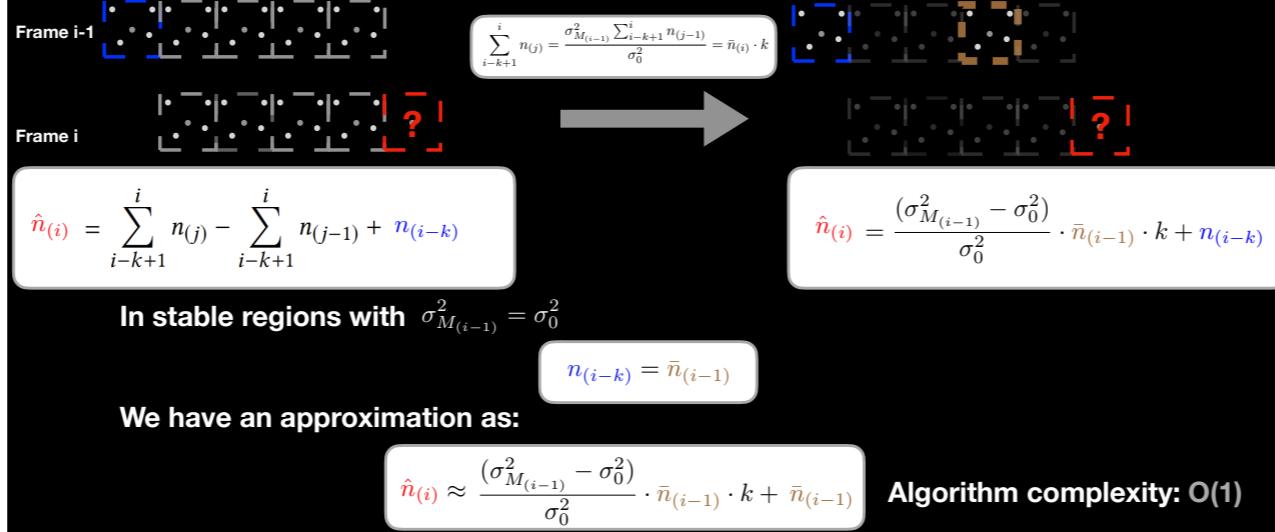
The diagram illustrates the temporal accumulation process. It shows two frames, "Frame i-1" and "Frame i". In "Frame i-1", there are blue dashed boxes representing accumulated sample counts. A mathematical equation calculates the total accumulated count $\sum_{i-k+1}^i n(j) = \frac{\sigma_{M_{(i-1)}}^2 \sum_{i-k+1}^i n(j-1)}{\sigma_0^2} = \bar{n}_{(i)} \cdot k$. In "Frame i", the accumulated counts from "Frame i-1" are shown, along with new red dashed boxes representing new samples. A large arrow points from "Frame i-1" to "Frame i". Below "Frame i", a formula for estimating the current sample count $\hat{n}_{(i)} = \sum_{i-k+1}^i n(j) - \sum_{i-k+1}^i n(j-1) + n_{(i-k)}$ is shown. To the right, another formula is provided: $\hat{n}_{(i)} = \frac{(\sigma_{M_{(i-1)}}^2 - \sigma_0^2)}{\sigma_0^2} \cdot \bar{n}_{(i-1)} \cdot k + n_{(i-k)}$.

22

To achieve this, we can use the difference of temporal accumulation to estimate the sample count with the metrics. However, we still have $n(i-k)$ in the formulation, which still requires us to keep k frame in the memory.

Metrics with Temporal Accumulation

- Use difference of temporal accumulation



Actually we have $n(i-k)$ =mean of the previous temporal accumulation when the variance is stable. We can use this approximation to reduce the algorithm complexity from $O(k)$ to $O(1)$, this is what affordable in a real-time rendering engine.

Metrics with Temporal Accumulation

- Use temporal accumulation to reduce cached sample counts

$$\hat{n}_{(i)} \approx \frac{(\sigma_{M_{(i-1)}}^2 - \sigma_0^2)}{\sigma_0^2} \cdot \bar{n}_{(i-1)} \cdot k + \bar{n}_{(i-1)}$$

History buffer O(1):

- Exponential moving average (EMA):

$$\bar{n}_{(i)} = (1 - \alpha)\bar{n}_{(i-1)} + \alpha n_{(i)}$$

$$\mathcal{H}_i = (\bar{n}_{(i)}, \mu_i, \sigma_{M_i}^2)$$

- Exponential moving variance (EMV) [Finch 2009]:

$$\sigma_{M_{(i)}}^2 = (1 - \alpha)\sigma_{M_{(i-1)}}^2 + \alpha(1 - \alpha)(\mathcal{S}(p_i) - \mathcal{C}(x_i, \Lambda))^2$$

- k-day EMA and simple moving average conversion [Bauer and Dahlquist 1998]

$$k = 2/\alpha - 1$$

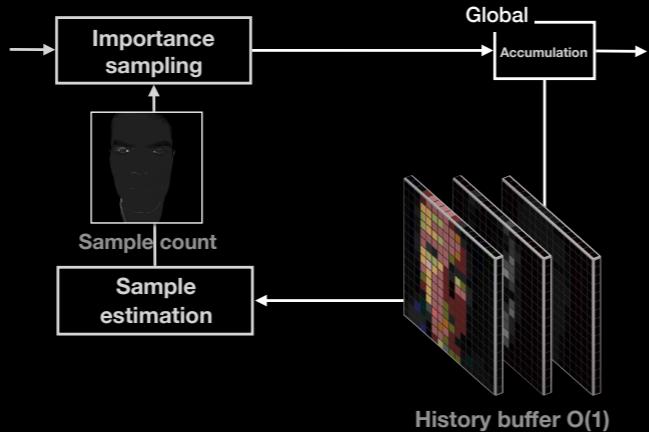
24

To achieve that mathematically, I list the most important formulas here. We can use EMA and EMV. Moreover, there is a commonly used conversion between k and the coefficient that's typically used in financial data analysis. In our formulation, we only need to maintain one history buffer.

Local Guiding

25

Current single pass adaptive sampling architecture:



- Depend on the existing global TAA
 - Requires a modification to output history
 - Affected by other passes using TAA. E.g., transparency overlay on subsurface.
- Global TAA parameter sets improve overall quality instead of a single pass.

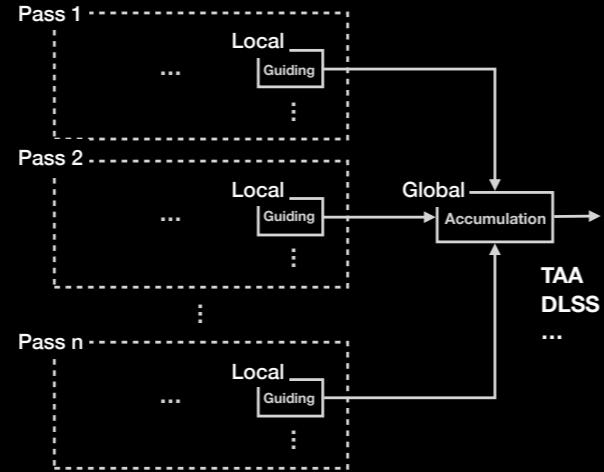
25

Current single pass adaptive importance sampling architecture is as below. It relies on the global TAA (Temporal Anti-Aliasing) pass to update the history buffer. This design makes it depend on the existing global TAA, which has three drawbacks. It requires a modification of global TAA to output history. Second, the subsurface pass might be affected by other passes like the overplayed transparency. At last, the global TAA parameter sets might improve overall quality instead of a single pass.

Local Guiding

26

Local pass guided adaptive sampling



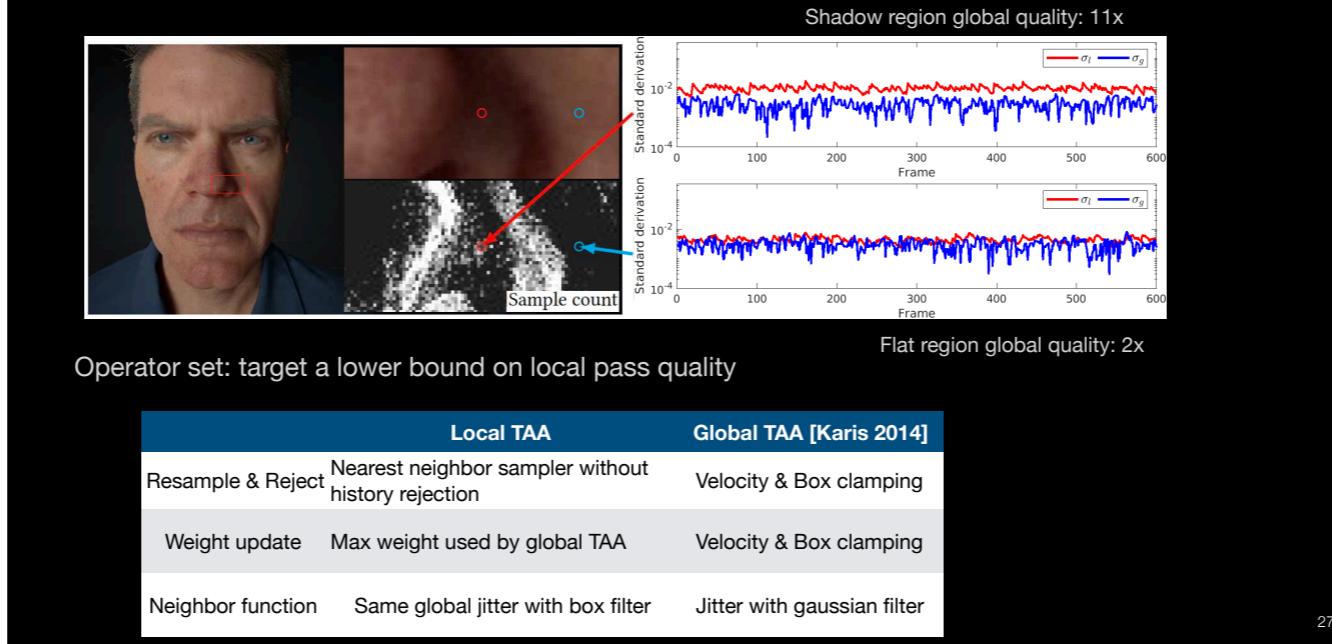
- Decouple the global accumulation
 - No global modification is required
 - Custom local quality control
 - Better quality input to global accumulation.

26

To resolve this issue, we propose a local pass guided adaptive importance sampling. This design decouples the global TAA. Which means, no global TAA modification is required, and we have custom local quality control, and we could have better quality input to global TAA.

Local Guiding with Global TAA

27



27

Let's take a look at a concrete example. For two points in the zoomed in region, one is in the shadow region, and the other is in the flat region. We show the red local error and blue global error for these two pixels over 600 frames. For shadow region, the global quality is 11x better, and the flat region global quality is 2x better.

Our local pass does not constrain the global TAA implementation. Because we target a lower bound on local pass quality. For example, as shown in this table, we use max weight of global TAA for weight update, while the weight in Global TAA is determined by velocity and box clamping. Because of this, people might be worry that this could cause ghosting to the rendering.

Local Guiding with Deep Learning Super Sampling

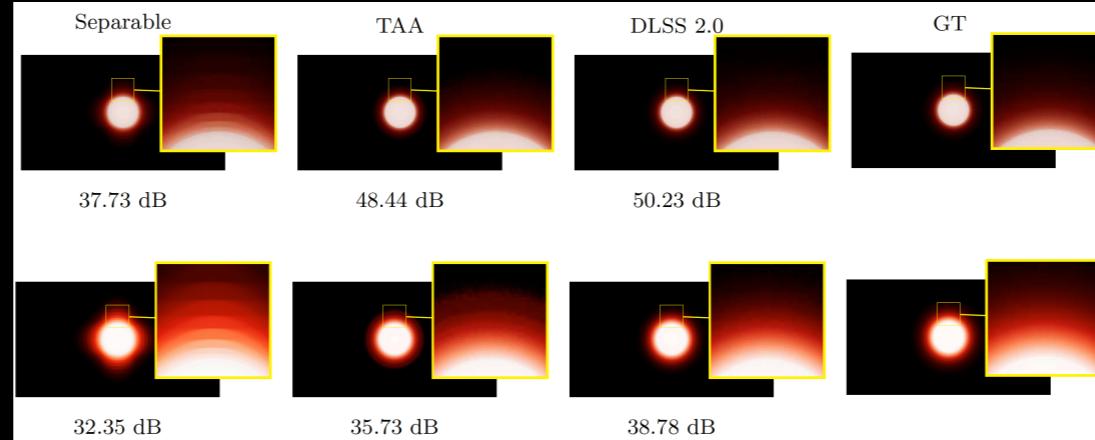


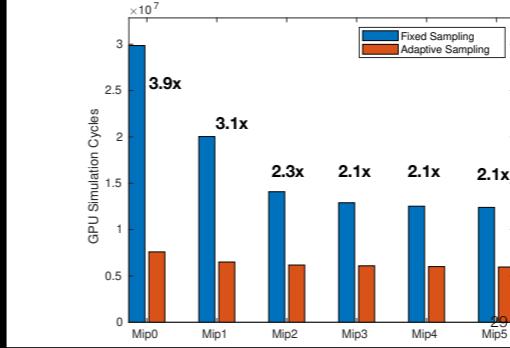
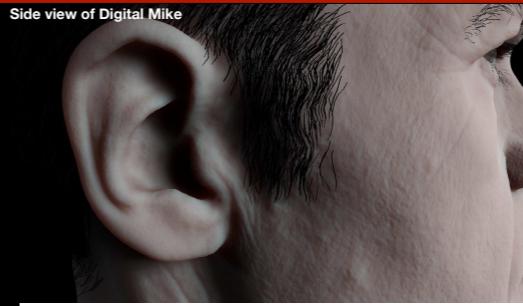
Figure 2: Quality of different global temporal accumulation methods (TAA and DLSS 2.0) under two different intensities. The quality of Separable is also presented. The PSNR is tested against the ground truth with 1024 spp after converting to luminance.

28

What happens if we replace the temporal accumulation method TAA by Deep Learning Super Sampling (DLSS) a technique developed by NVIDIA, which has the temporal accumulation capability. I have measured the quality in PSNR (peak signal-to-noise ratio, the higher the better) for a beam light that strikes the surface vertically. DLSS has higher quality. Note that DLSS is designed for general accumulation, not designed for subsurface scattering. Just imagine what would happen if we have a deep learning neural network that's designed for subsurface scattering.

Cycle Level Analysis of Bandwidth Demands

- GPGPU-Sim [Bakhoda et al., 2009, Khairy et al., 2020]
- Cycle-level analysis of parallel computing
- CUDA
- Hardware config:
 - NVIDIA TITAN X Pascal (compute shader)
- Adaptive vs. Fixed sampling
 - Simulation cycles



Now it's working, let's see how it improves the bandwidth and cache coherence. To achieve this, a cycle level analysis framework GPGPU-sim is used. The shading code is manually ported and we run both adaptive and fixed sampling for a side view of digital mike. The diagram shows the cycles saved with different mip level.

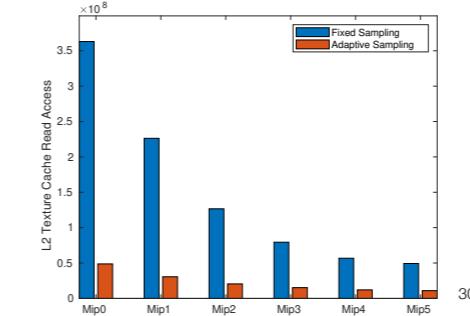
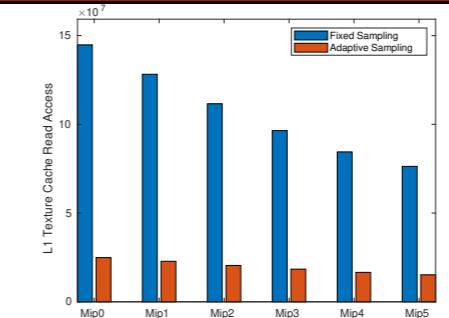
The available tested configurations have:

SM2_GTX480, SM3_KEPLER_TITAN, SM6_TITANX, SM7_TITANV, SM7_QV100, SM75_RTX2060

SM5 introduces compute shader. Therefore the minimal configuration SM6_TITANX is selected for the analysis.

Cycle Level Analysis of Bandwidth Demands

- GPGPU-Sim [Bakhoda et al., 2009, Khairy et al., 2020]
 - Cycle-level analysis of parallel computing
 - CUDA
 - Hardware config:
 - NVIDIA TITAN X Pascal (compute shader)
- Adaptive vs. Fixed sampling
 - Simulation cycles
 - L1 & L2 cache demands

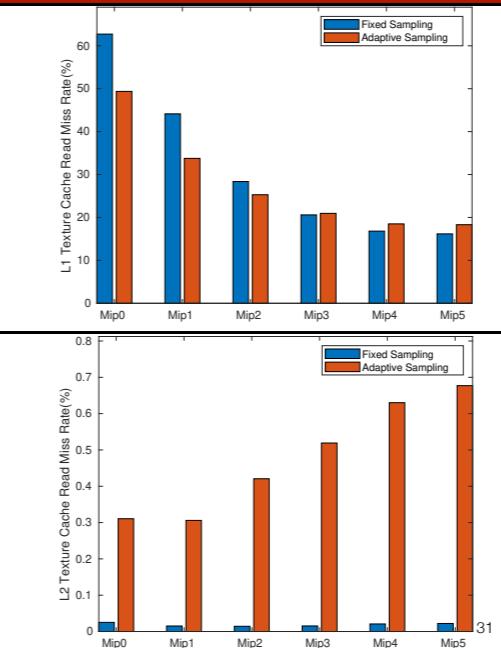


We can reduce both cache demands

Cycle Level Analysis of Bandwidth Demands

- GPGPU-Sim [Bakhoda et al., 2009, Khairy et al., 2020]
 - Cycle-level analysis of parallel computing
 - CUDA
 - Hardware config:
 - NVIDIA TITAN X Pascal (compute shader)
- Adaptive vs. Fixed sampling
 - Simulation cycles
 - L1 & L2 cache demands
 - L1 & L2 cache miss rate

Mip0: 91,541 vs 151,356

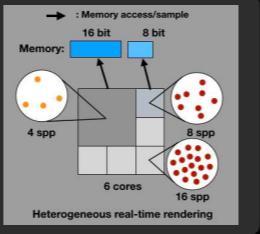


Moreover, we can reduce the cache read miss a lot for L1. And keep L2 small and similar. Our adaptive sampling strategy performs good to deal with the bandwidth and cache problem.

Outline

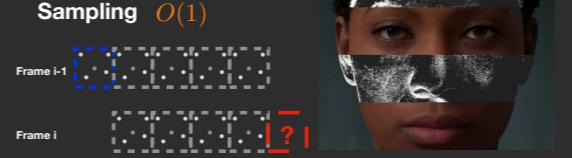
Section I: Chapter 1 ~ Chapter 3

- Introduction
- Literature
- Motivation
- Heterogeneous Real-time Rendering



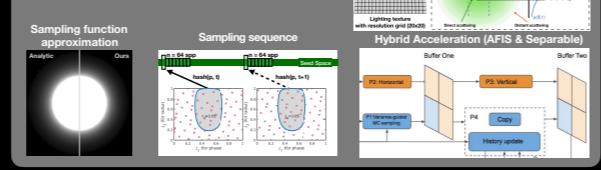
Section II: Chapter 4 (I3D'20)

- Real-time Adaptive Sampling $O(1)$



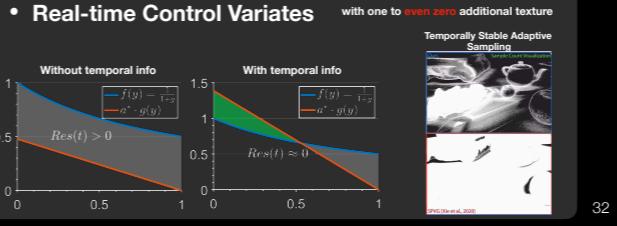
Section III: Chapter 5

- Subsurface Scattering



Section IV: Chapter 6 (I3D'21)

- Real-time Control Variates



32

Section 3 is dedicated for the actual design of efficient subsurface scattering.

Introduction

Scene



33

Subsurface scattering is a demanding feature in real-time graphics. For this whole scene rendered in Unreal Engine 4, if we apply screen-space subsurface scattering with Burley's diffusion profile, it looks much better but at high rendering cost.

Introduction

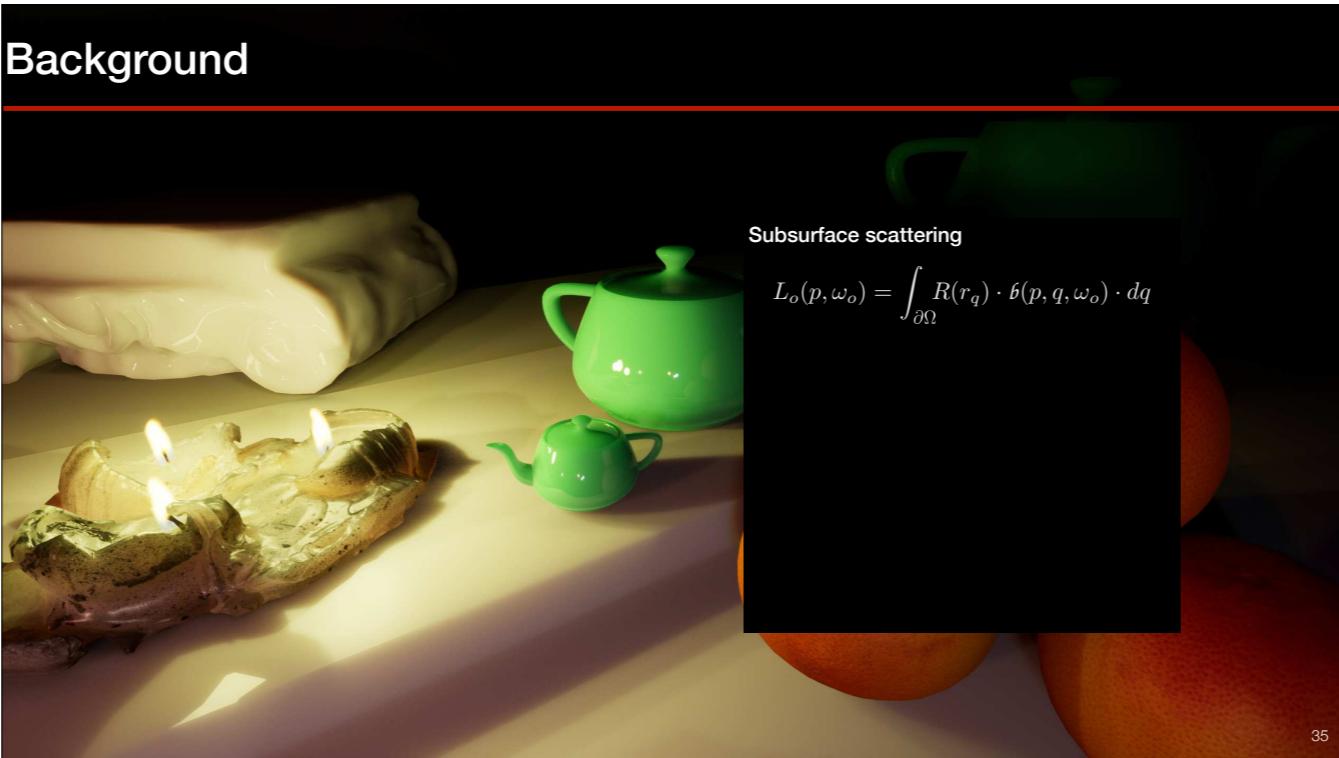
Scene + Subsurface scattering with Burley's diffusion profile [Christensen and Burley 2015]



34

Subsurface scattering is a demanding feature in real-time graphics. For this whole scene rendered in Unreal Engine 4, if we apply screen-space subsurface scattering with Burley's diffusion profile, it looks much better but at high rendering cost.

Background



In order to get this image in soft look, we need to solve this simplified equation in screen space.

Background



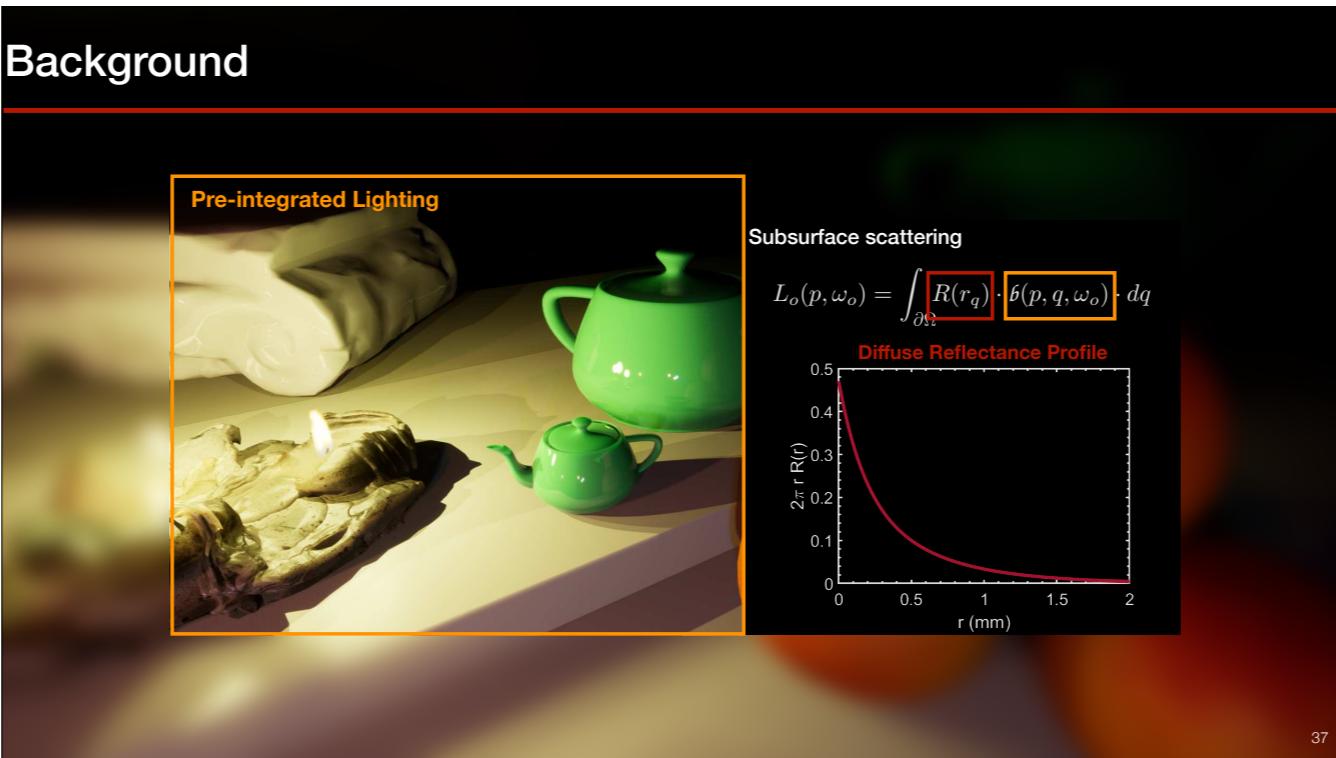
Subsurface scattering

$$L_o(p, \omega_o) = \int_{\partial\Omega} R(r_q) \cdot [b(p, q, \omega_o)] \cdot dq$$

36

where b is the screen-space pre-integrated lighting.

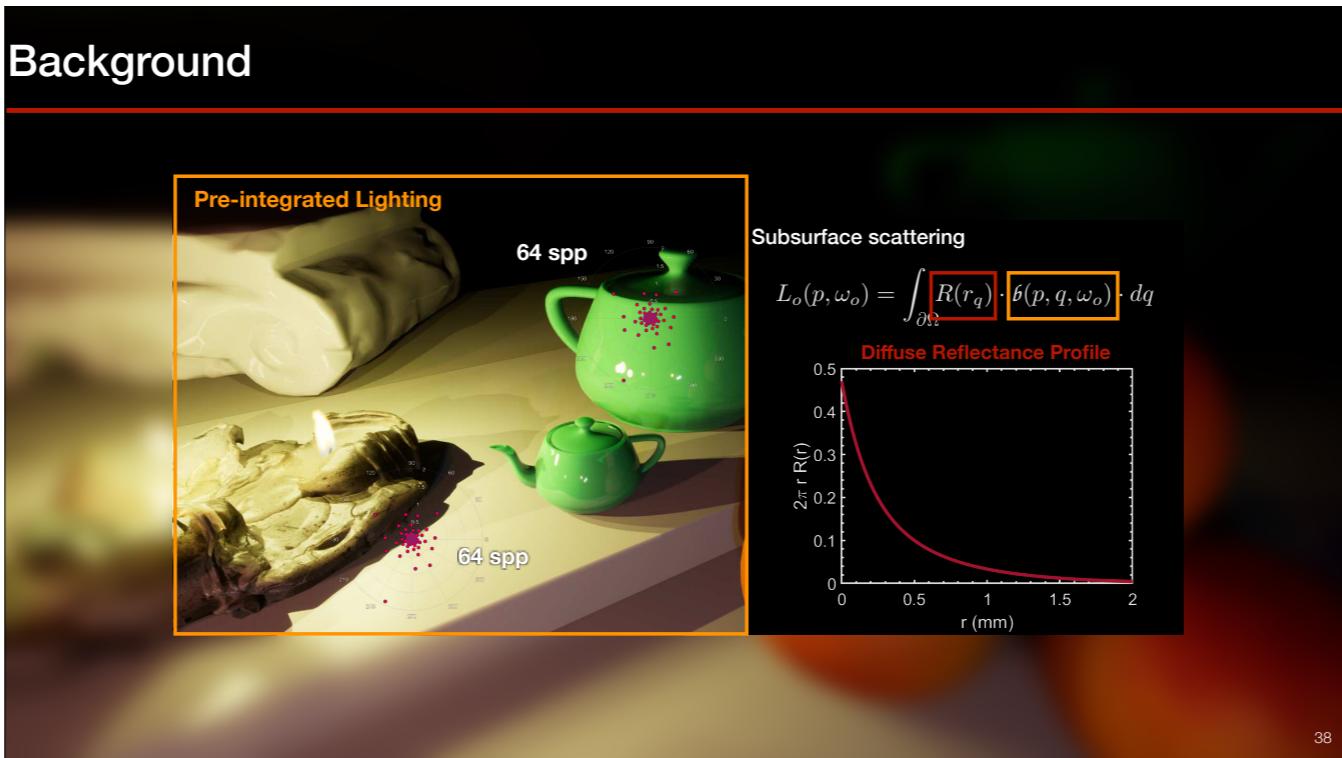
Background



37

R is the `diffuse reflectance profile`. To solve this problem in rendering, we can use Monte Carlo sampling for each pixel,

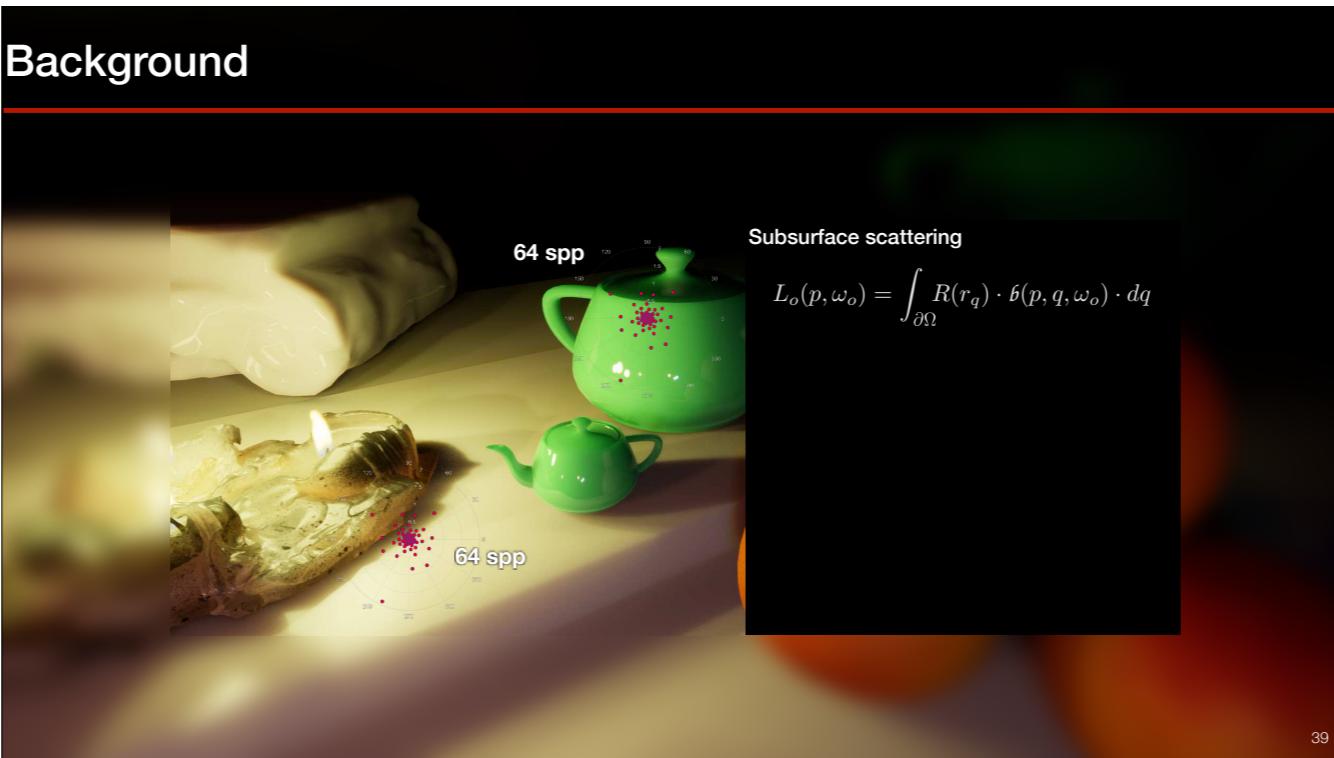
Background



38

like this with uniform 64 samples per pixel everywhere (either around shadow regions or over flat regions).

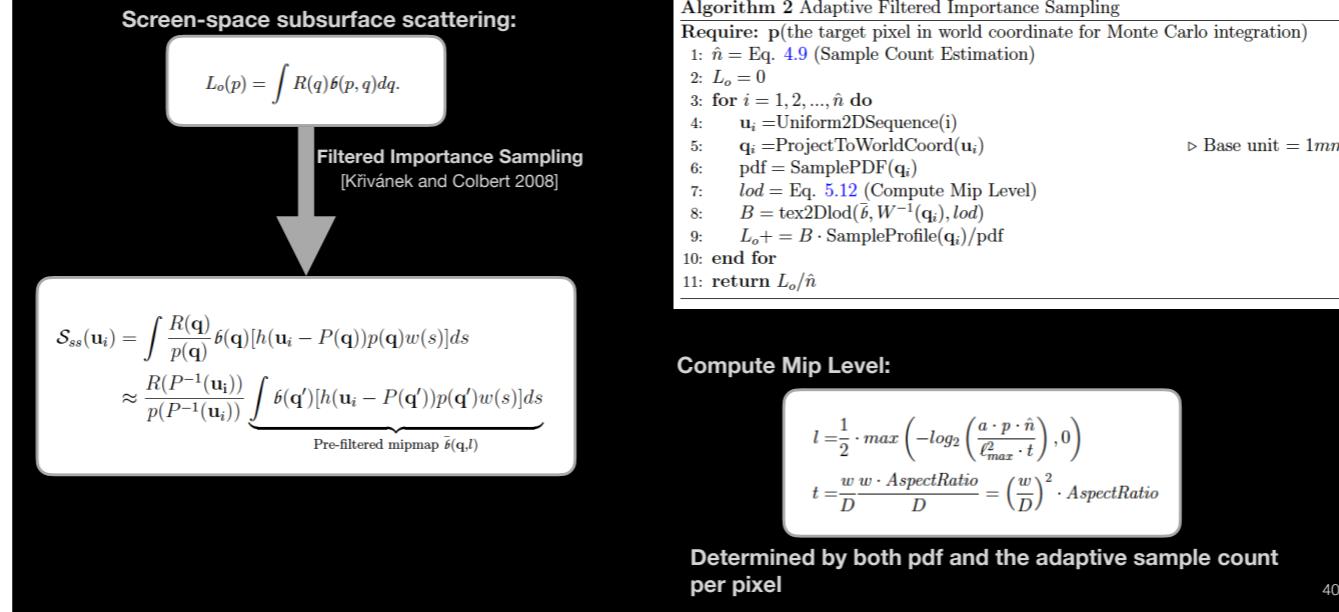
Background



39

However, a uniform sampling of 64 spp creates high bandwidth demands and thus is expensive in real-time rendering. To reduce the bandwidth demands, real-time adaptive sampling can be used.

Adaptive Filtered Importance Sampling



Since we have demonstrated that mipmaps can increase the performance. We use filtered importance sampling to make use of this feature. Then we combine it with our adaptive sampling technique as adaptive filtered importance sampling (AFIS) for efficient subsurface scattering. Note that the mip level is not only determined by the pdf but the adaptive sampling count per pixel.

Sampling function

CDF for Burley's Normalized Diffusion Profile [Christensen and Burley 2015]:

$$cdf(r) = 1 - \frac{1}{4}e^{-r/d} - \frac{3}{4}e^{-r/(3d)}$$

Analytic inverse [Golubev 2019]:

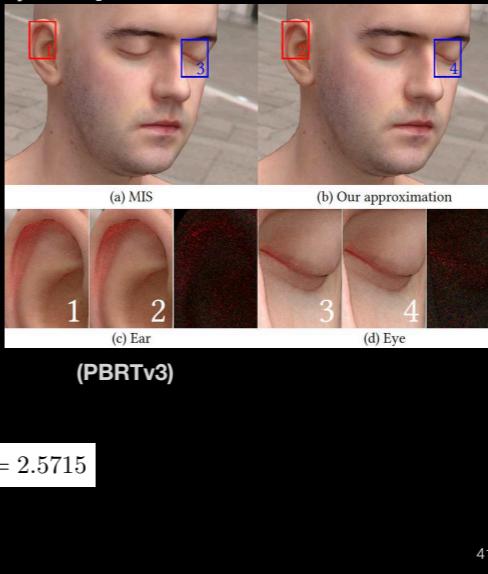
$$cdf^{-1}(\xi) = 3d \log\left(\frac{1 + G(\xi)^{-1/3} + G(\xi)^{1/3}}{4\xi}\right)$$

$$G(\xi) = 1 + 4\xi(2\xi + \sqrt{1 + 4\xi^2})$$

Analytic

Ours

Our approximation:



$$cdf^{-1}(\xi) = d((2 - c)\xi - 2)\log(1 - \xi) \quad c = 2.5715$$

41

In order to perform importance sampling, we need to know the cumulated distribution function for Burley's Normalized Diffusion profile. And find the analytic inverse, and use it for radius sampling in subsurface scattering. However, this inverse is a little complicated. Instead we propose an approximation, which is much simpler and faster. The profile also looks similar. We added our sampling function into PBRT (an offline renderer accompanied with the book Physically Based Rendering: From Theory to Implementation), and compared it with the default implementation, we could observe a close match for 2k spp in ear and eye region.

Sampling Sequence

- Low-discrepancy sequence

- 2D R₂ sequence [Roberts, 2018]

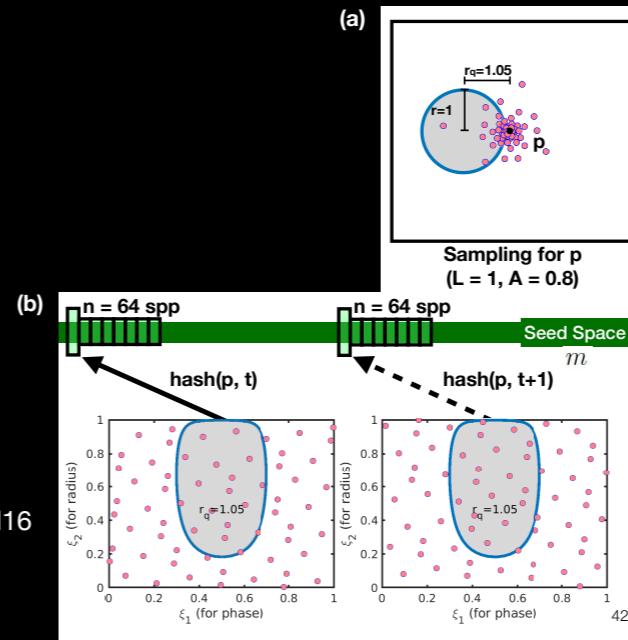
$$t_m = m\psi, m = 1, 2, 3, \dots$$

$$\psi = \left(\frac{1}{\phi_d}, \frac{1}{\phi_d^2}, \dots \frac{1}{\phi_d^2} \right)$$

$$x^{d+1} = x + 1 \quad \phi_2 \approx 1.324718$$

- Random number generator [Jarzynski and Olano, 2020]

- Hash the location and frame index with pcg3d16



For offline rendering, a low-discrepancy sequence is enough. We choose to use R2 sequence which computes fast online. For real-time rendering. We also need a random number generator so that the sampling sequences between frames are randomized while in-frame has low discrepancy. The best generator is selected based on the work by Jarzynski and Dr. Olano. Figure (a) shows the sampling for a point p . Figure (b) show two consecutive sampling in the 2D sampling sequence domain.

Side story: runs into an issue not related to algorithm, but the monitor. There is a vertical banding in the monitor I use, ViewSonic. XG350R-C for the sample count map visualization.

Adaptive Filtered Importance Sampling



a) Close patch



b) Ear



c) Front

Fixed and Adaptive (Adt.) 64spp sampling time (ms):

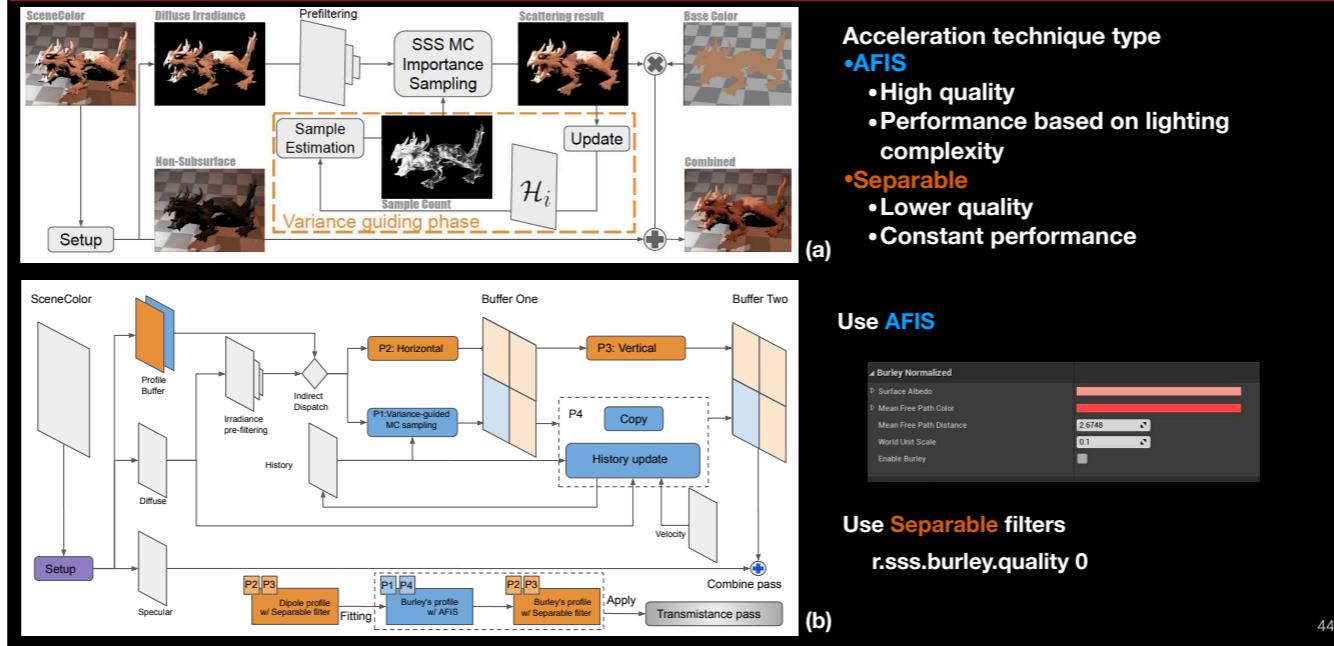
Scenario	Setup	Pre-filtering	Sampling	Update	Combine	Total
a)+fixed	0.38	0.16	10.73	N/A	0.20	11.47
a)+adt.	0.38	0.16	1.50	0.54	0.20	2.78
b)+fixed	0.41	0.17	9.72	N/A	0.27	10.22
b)+adt.	0.41	0.17	2.72	0.46	0.27	4.03
c)+fixed	0.35	0.17	1.45	N/A	0.11	2.08
c)+adt.	0.35	0.17	0.38	0.14	0.11	1.15

Configuration: 1080p with NVIDIA Quadro P4000 (SM:14).

43

Next, we show the actual performance breakdown of digital mike at three different distance for 1080p with NVIDIA Quadro P4000, which has 14 Streaming multiprocessors. We compared the contribution of adaptive importance sampling to fixed 64 spp. Our adaptive sampling method contributes an additional overall acceleration from 2x to 4x even considering all the overhead. We got a max of 6x acceleration in all of our test cases. It's orthogonal to other applied acceleration technique like pre-filtering.

Importance-guided Acceleration



Instead of using the AFIS framework in (a), I have also considered the computing demands minimization by combining multiple techniques, AFIS and Separable to make full use of their pros. The new framework is shown in figure (b).

Importance-guided Acceleration



a) Close patch



b) Ear



c) Front

AFIS (ms):

Scenario	Setup	Pre-filtering	Sampling	Update	Combine	Total	PSNR
a)+adt.	0.38	0.16	1.50	0.54	0.20	2.78	37.58 dB
b)+adt.	0.41	0.17	2.72	0.46	0.27	4.03	45.70 dB
c)+adt.	0.35	0.17	0.38	0.14	0.11	1.15	44.28 dB

Separable (ms)

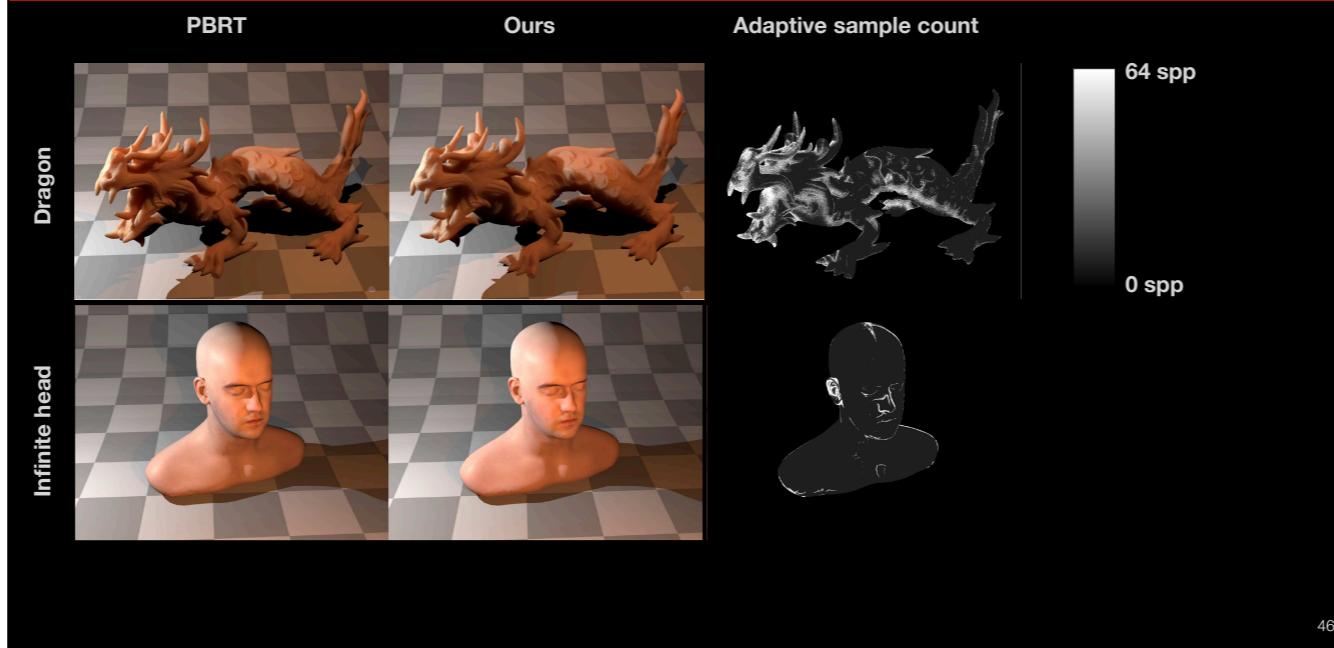
Total	PSNR
4.13	37.46 dB
4.00	40.28 dB
0.95	42.95 dB

Configuration: 1080p with NVIDIA Quadro P4000 (SM:14).

45

Next, we show the actual performance breakdown of digital mike at three different distance for 1080p with NVIDIA Quadro P4000, which has 14 Streaming multiprocessors. We compare the quality and performance of AFIS and Separable. For better quality we can use Separable, but for performance reason, we can use separable or a hybrid usage.

Result - Ground truth comparison

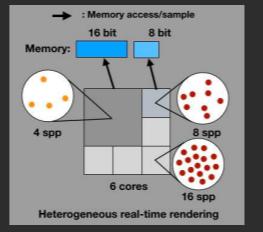


We show the ground truth comparison between ours implemented in UE4 and PBRT for asia dragon and infinite head model. The adaptive sample count is also shown here. We can see a close match between ours and the ground truth. Our adaptive sample count can capture sample count demands, e.g., during the shadow nose region of infinite head.

Outline

Section I: Chapter 1 ~ Chapter 3

- Introduction
- Literature
- Motivation
- Heterogeneous Real-time Rendering



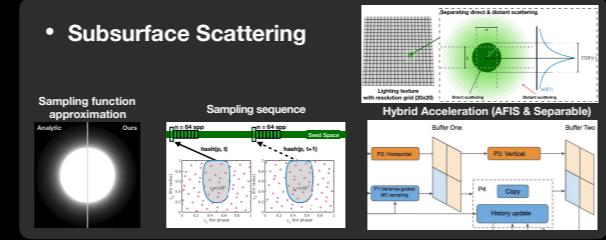
Section II: Chapter 4 (I3D'20)

- Real-time Adaptive Sampling $O(1)$



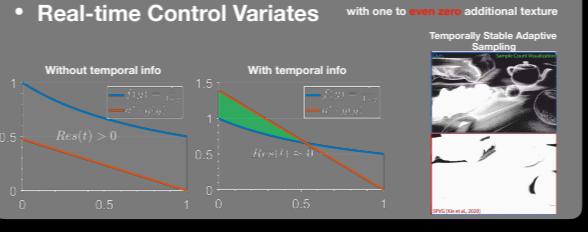
Section III: Chapter 5

- Subsurface Scattering



Section IV: Chapter 6 (I3D'21)

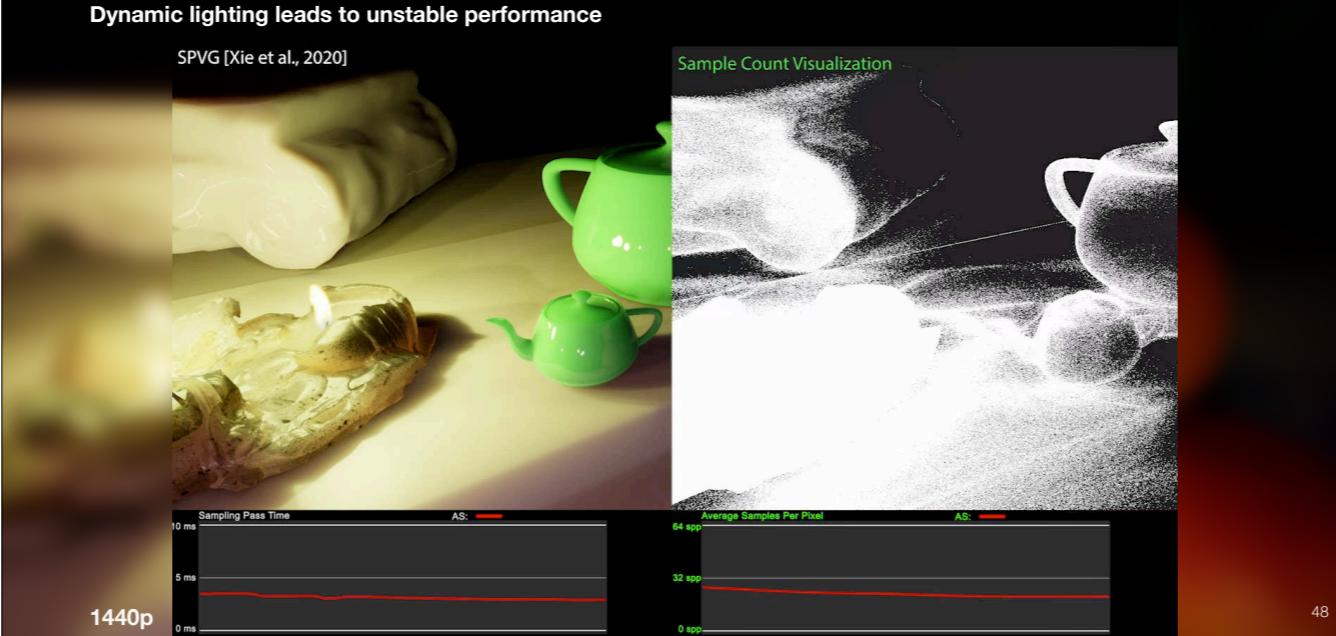
- Real-time Control Variates



47

In the last section, we talk about real-time control variates to create temporally stable adaptive sampling.

Motivation Example

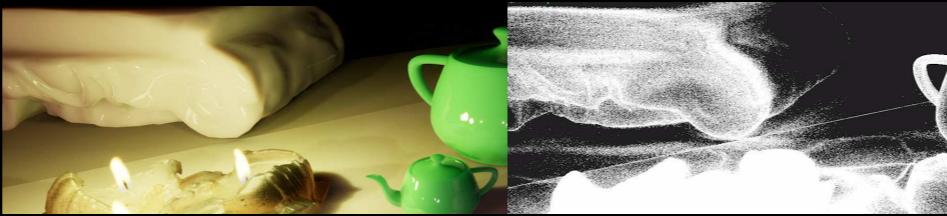


The algorithm proposed in section II has good performance for stable scenes. However, dynamic lighting leads to unstable performance. Here we show the scene and the corresponding sample count. The bottom shows the sampling pass time and the average samples per pixel for the whole frame. As you can see, turning on/off the light adds in temporal variance of the lighting that increases the sample count and the corresponding sampling pass time.

Motivation Example

Static scene:

- Use spatial variance



Dynamic scene:

- Reduce temporal variance



How?

49

So how can we use the spatial variance in static scene previously in real-time adaptive sampling, and reduce the temporal variance during dynamic scene at the same time, to achieve temporally stable adaptive sampling?

Temporally Stable Adaptive Sampling

- Reduce temporal variance with Control Variates

50

To achieve this, we seek to use Control Variates. Before running into the detailed design to use subsurface control variates, let's first analyze the variance and then solve the problem on a simplified example in temporal domain.

Variance Analysis

Variance of the product of a spatial F and temporal T variable:

$$Var(\langle TF \rangle) = (Var(T) + E(T)^2) Var(F) + Var(T)E(F)^2 \quad (T \text{ and } F \text{ are independent})$$

For a spatial variable F and another temporal variable T. The Variance of the Monte Carlo sampling of their product can be decomposed into two terms where T and F are independent.

Variance Analysis

Variance of the product of a spatial F and temporal T variable:

$$Var(\langle TF \rangle) = (Var(T) + E(T)^2) Var(F) + Var(T)E(F)^2 \quad (\text{T and F are independent})$$

Sample Count ↑ $Var(F)$ ↓

52

As we know, when the total sample count increases at each sampling time, the variance of F decreases.

Variance Analysis

Variance of the product of a spatial F and temporal T variable:

$$Var(\langle TF \rangle) = (Var(T) + E(T)^2) Var(F) + Var(T)E(F)^2 \quad (\text{T and F are independent})$$

Sample Count ↑ $Var(F)$ ↓

$$Var(\langle TF \rangle) = \boxed{(Var(T) + E(T)^2)Var(F)} + Var(T)E(F)^2$$

↓ **Unaffected**

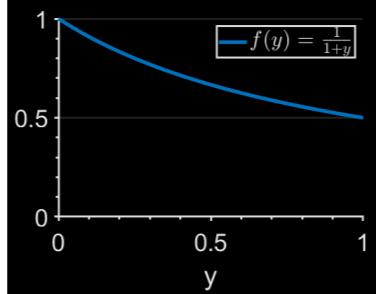
How to get rid of $Var(T)E(F)^2$ for variance tracking ?

What is $(Var(T) + E(T)^2)Var(F)$? Variance of CV residual, Res(t)

53

It reduces the total variance of the green term. However, no matter how many samples are used, the second term is not affected. However, during dynamic scene, the second variance term can increase suddenly, causes the overestimation of sample count. So how to get rid of the second term for variance tracking, and what is the first term. If you have read the paper, it is the variance of CV residual. We can track that directly.

Standard CV



$$F = \int_{y \in \mathcal{D}} f(y) dy$$

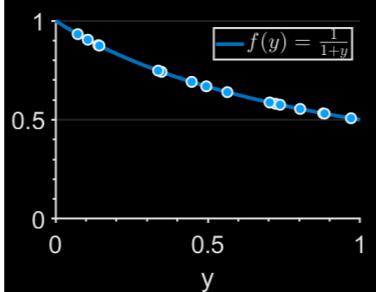
16 spp 0.6841

$F = \ln(2) \approx 0.6931$

54

Let's review the standard CV usage. If we want to integrate a function over the domain shown here numerically for each pixel, like 16 spp, we have an estimation, which has a large error when compared to the ground truth.

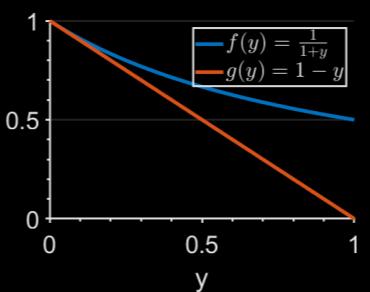
Standard CV



$$F = \int_{y \in \mathcal{D}} f(y) dy$$

16 spp

0.6841



$$F = G + \int_{y \in \mathcal{D}} f(y) - g(y) dy$$

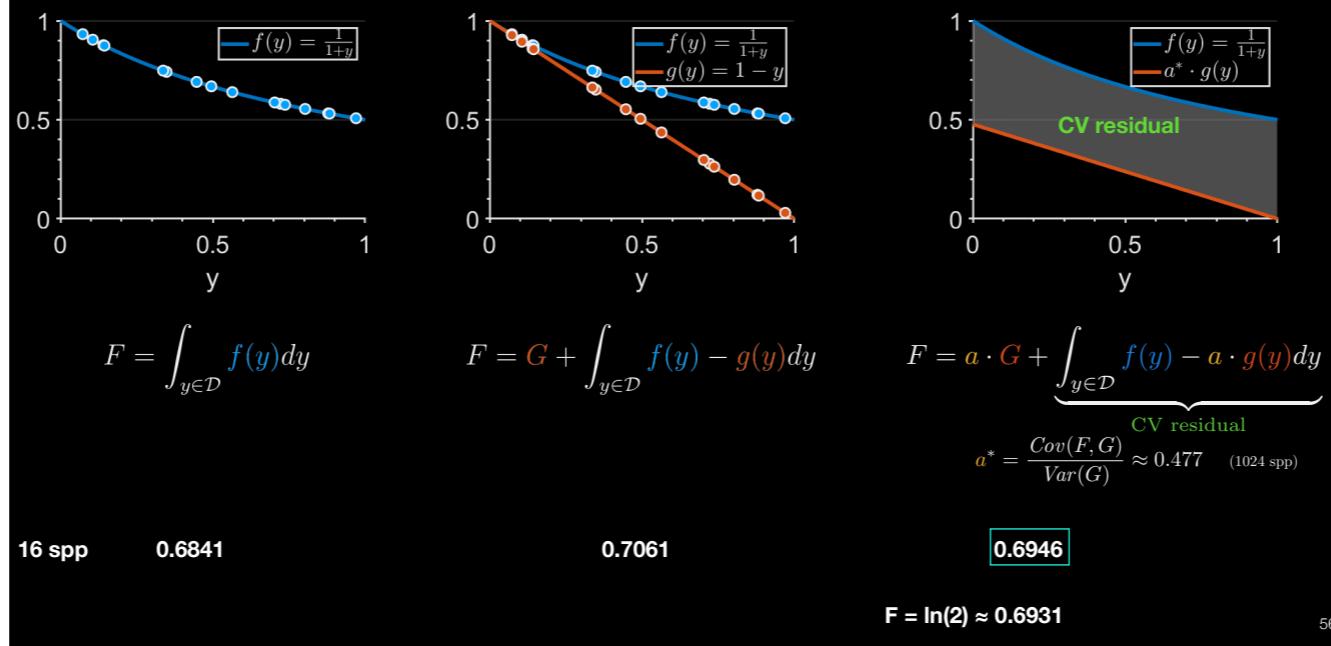
0.7061

$$F = \ln(2) \approx 0.6931$$

55

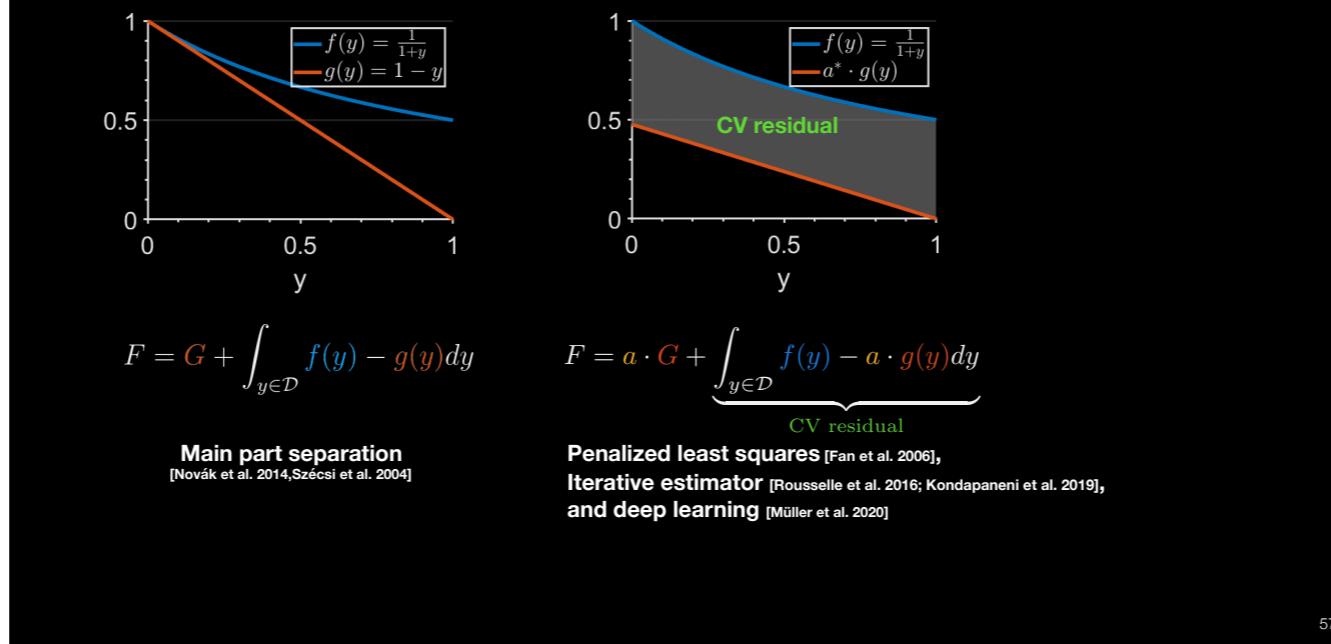
To improve the estimation, we can add a control variable directly with a known integration to reduce the error. However, it can lead to higher error if not well selected.

Standard CV



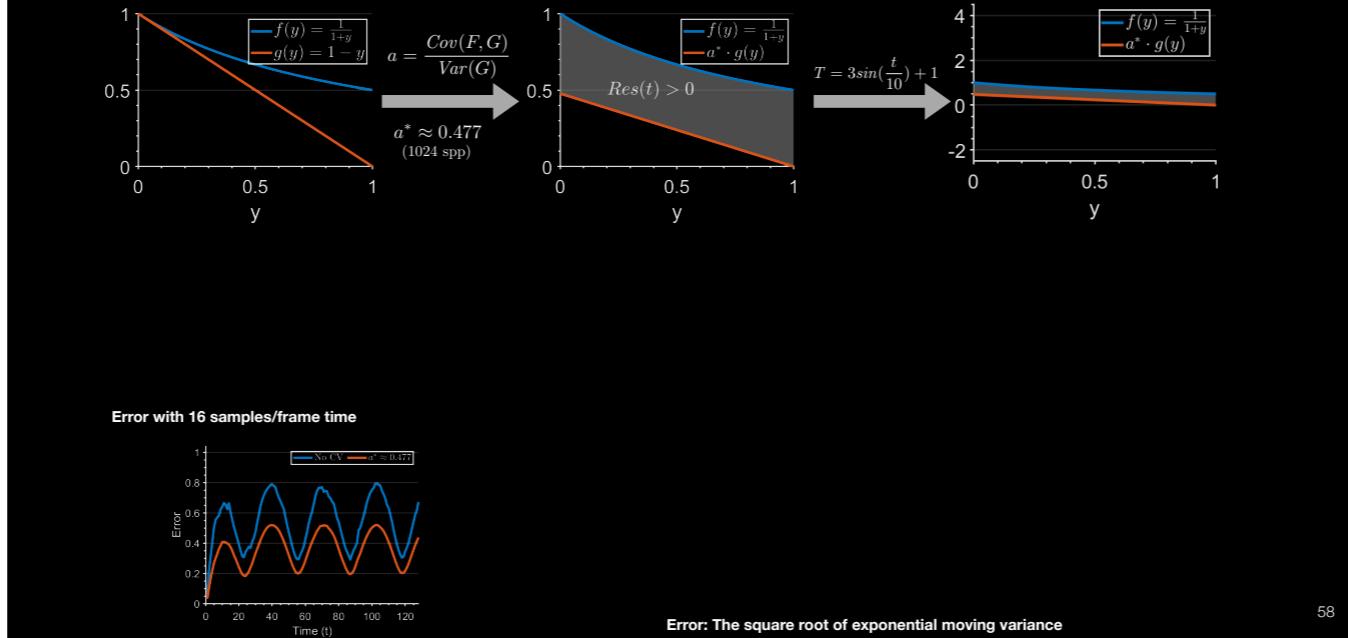
Even with this control variable, we can still find an optimal scale factor numerically beforehand. In the same sampling configuration, it leads to less error when compared to the ground truth. This scale factor is the CV coefficient, and the integration part is the CV Residual.

Standard CV in Computer Graphics



In computer graphics, when CV coefficient is a predefined constant like 1. it is usually called main part separation. Otherwise, different methods have been used to get the optimal CV coefficient like penalized least squares, iterative estimator and deep learning.

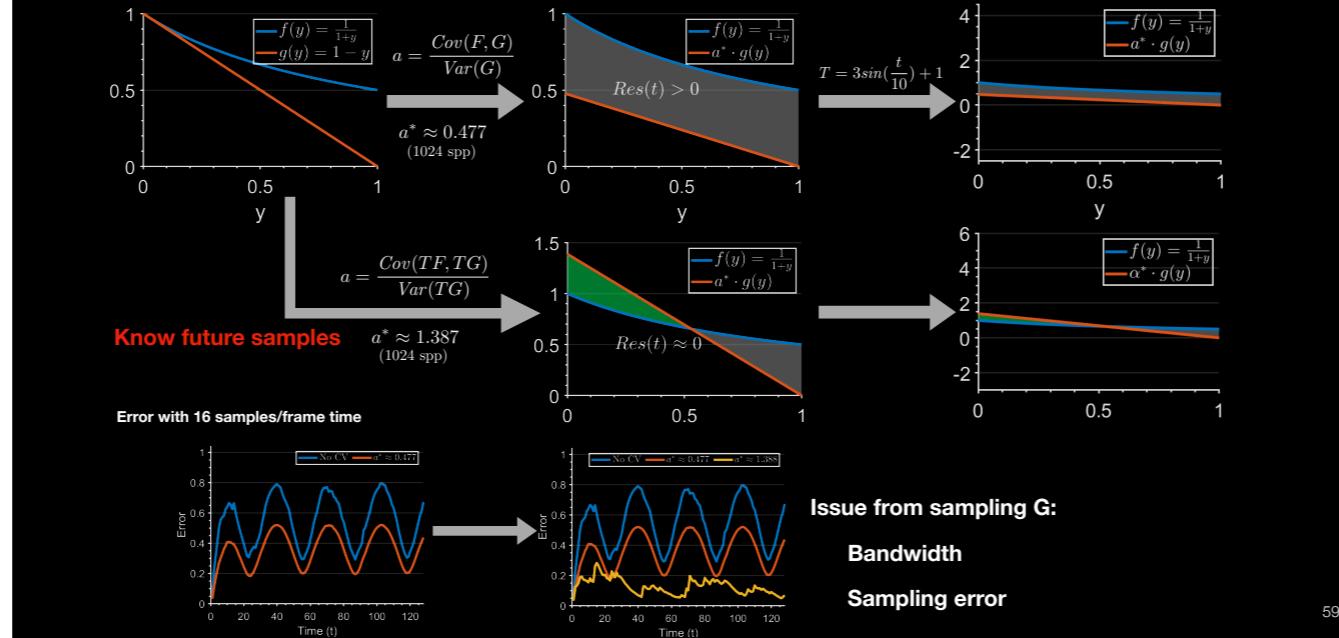
In-frame Standard CV



58

However, if we have a temporal component, this temporal dynamics still impacts the error a lot with in-frame standard CV.

In-frame Standard CV



59

If we take time into the formula and we know future samples, we can arrive at a CV coefficient ($a^* = 1.387$), that leads to a residual close to zero. The periodic large error can be removed. However, in real-time rendering, we do not know the future. Moreover, we have another issue from sampling G. Because sampling G could double the bandwidth. And it could add in new sampling errors.

Online CV estimation algorithm

How to solve CV coefficient online: $a = \frac{Cov(TF, TG)}{Var(TG)}$

(1) T is independent from F and G (2) No independence assumption

Exponential Moving Covariance Matrix (EMCM):

$$\Sigma_t = (1 - \alpha)\Sigma_{t-1} + \alpha(1 - \alpha)(\mathbf{Z}_t - \zeta_{t-1})(\mathbf{Z}_t - \zeta_{t-1})^T$$

\mathbf{Z}_t **Observation of TF and TG at t**

ζ_{t-1} **Exponential moving average of TF and TG at t-1.**

Σ_t **Covariance of TF and TG.**

$$\Sigma_t = \begin{pmatrix} Var_t(\bar{f}) & Cov_t(\bar{f}, \bar{g}) \\ Cov_t(\bar{g}, \bar{f}) & Var_t(\bar{g}) \end{pmatrix}$$

G is a constant control variable

$$Var(G) = 0$$

CV Coefficient estimation:

$$a_t = \frac{\Sigma_t \cdot xy}{\Sigma_t \cdot yy}$$

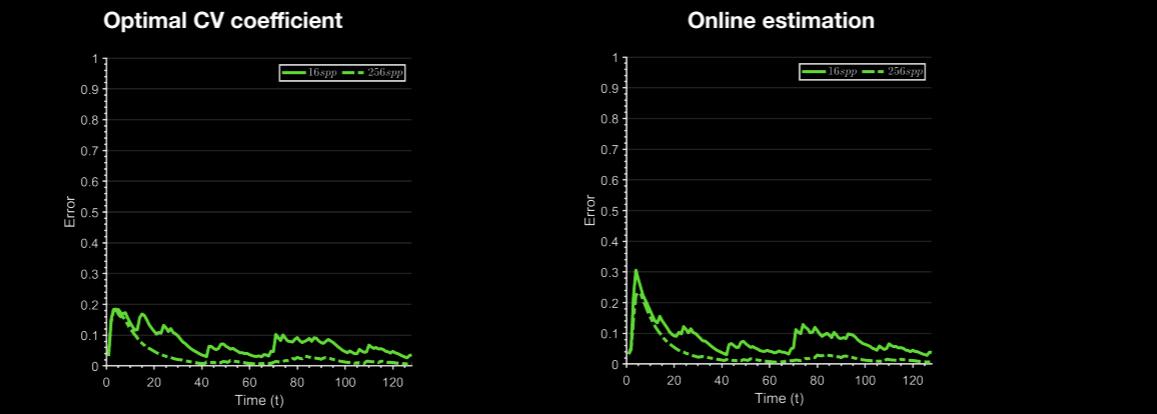
60

So how to solve the CV coefficient formula online. Actually, if T is independent from F and G, we can have a simplified formula if G is a constant control variable. When we do not have this independence assumption, we proposed exponential moving covariance matrix, which uses more recent observations to update the covariance. Then the CV coefficient can be estimated online with this formula.

Online CV estimation

Track variance of CV residual:

$$\text{Var}\left(\int_{y \in D} f(x, y, t) - a(x, t) \cdot g(x, y, t) dy\right)$$



61

Here we illustrate the CV residual error between the optimal CV coefficient and our online estimation. Our online estimation reduces the periodic error similar to optimal CV coefficient.

Temporally Stable Adaptive Sampling

- Reduce temporal variance with Control Variates
- Application to real-time subsurface scattering

62

Now that we know how to reduce the temporal variance with control variates, let's apply the online CV coefficient estimation to real-time subsurface scattering.

Application to Subsurface Scattering



Let's go back to the example demonstrated in the motivation example. Let's see how the CV coefficient changes according to time. Note that in the visualization, most regions are actually blue. That is because for large flat lighting region, we have a theoretical optimal CV coefficient, which is 1 for subsurface scattering. The CV coefficient visualized here is the new changes. The accumulated one has slight change only, and is not shown here.

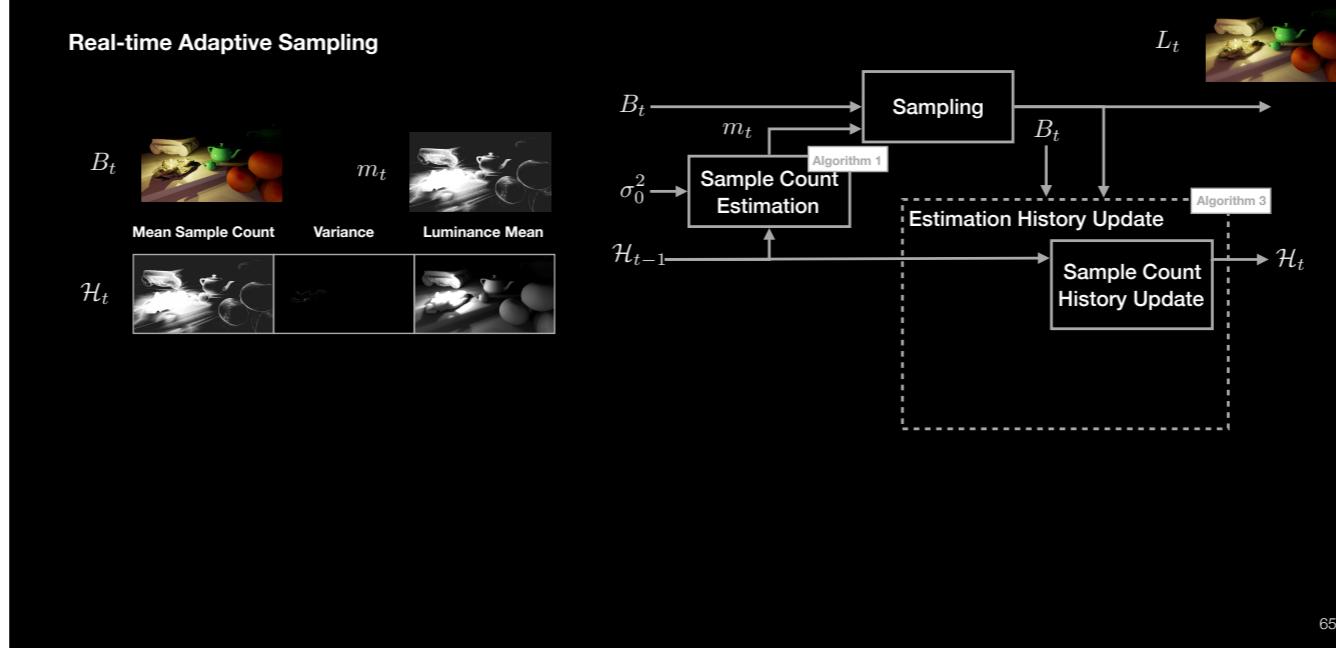
Temporally Stable Adaptive Sampling

- Reduce temporal variance with Control Variates
- Application to real-time subsurface scattering
- Temporally stable adaptive sampling algorithm

64

Now that we have the subsurface control variates, let's integrate it into real-time adaptive sampling to estimate sample count. Note that our application of CV is different from the typical use where CVs are applied in shading domain. Instead, we apply it in the sample domain, to reduce overestimated sample count. Our application will not create bias in rendering.

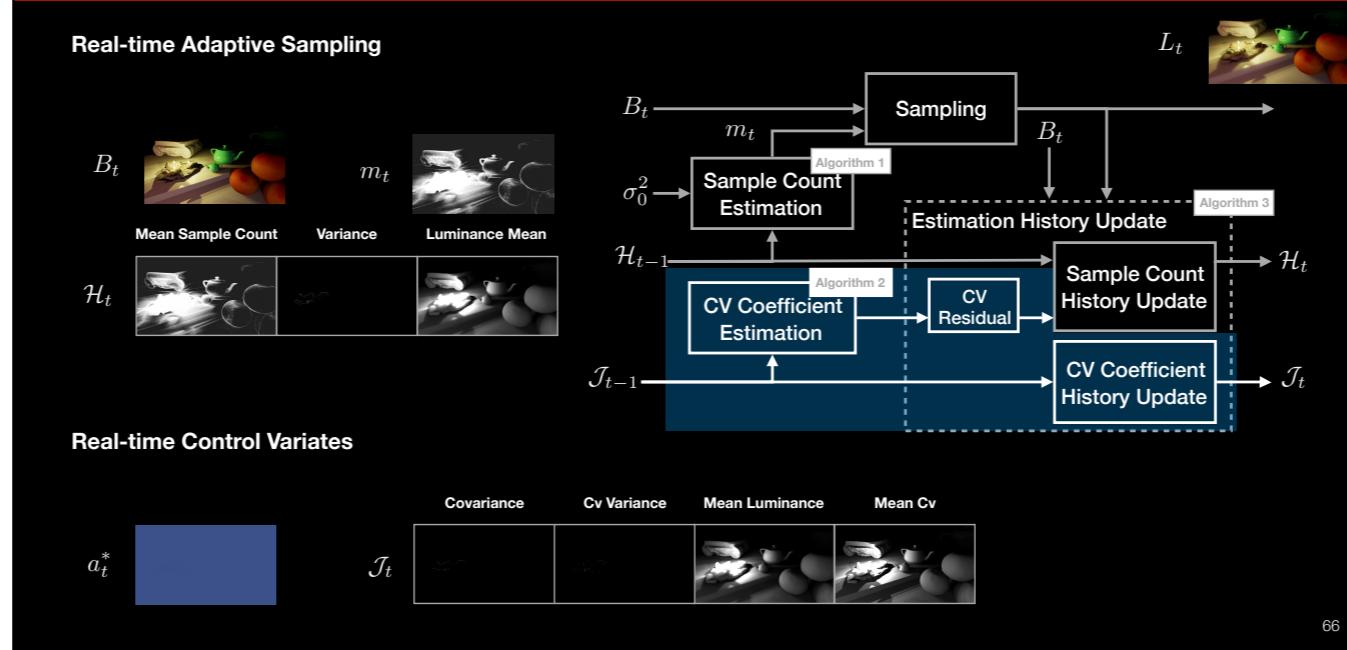
Temporally Stable Adaptive Sampling Algorithm Overview



65

What we see here is the real-time adaptive sampling algorithm. Each time when we get the pre-integrated lighting, a sample count map m_t is estimated for each pixel based on history in the previous frame. After sampling, three history channels are updated for sample count estimation: the mean sample count, the variance and mean of luminance.

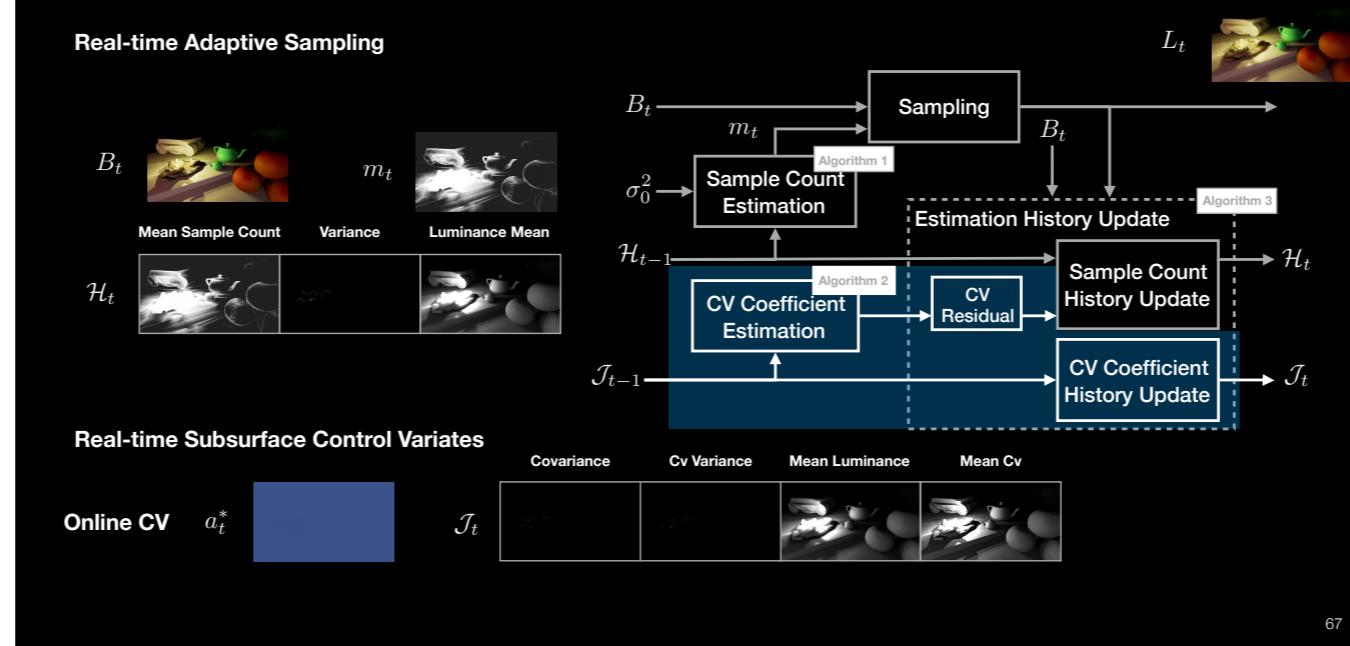
Temporally Stable Adaptive Sampling Algorithm Overview



66

For our paper, real-time subsurface control variates, we added another history texture to store the compacted history for CV coefficient estimation. The four channels are Covariance, Control variable variance, mean luminance and mean Control variable. The CV residual replaces the Luminance in real-time adaptive sampling algorithm.

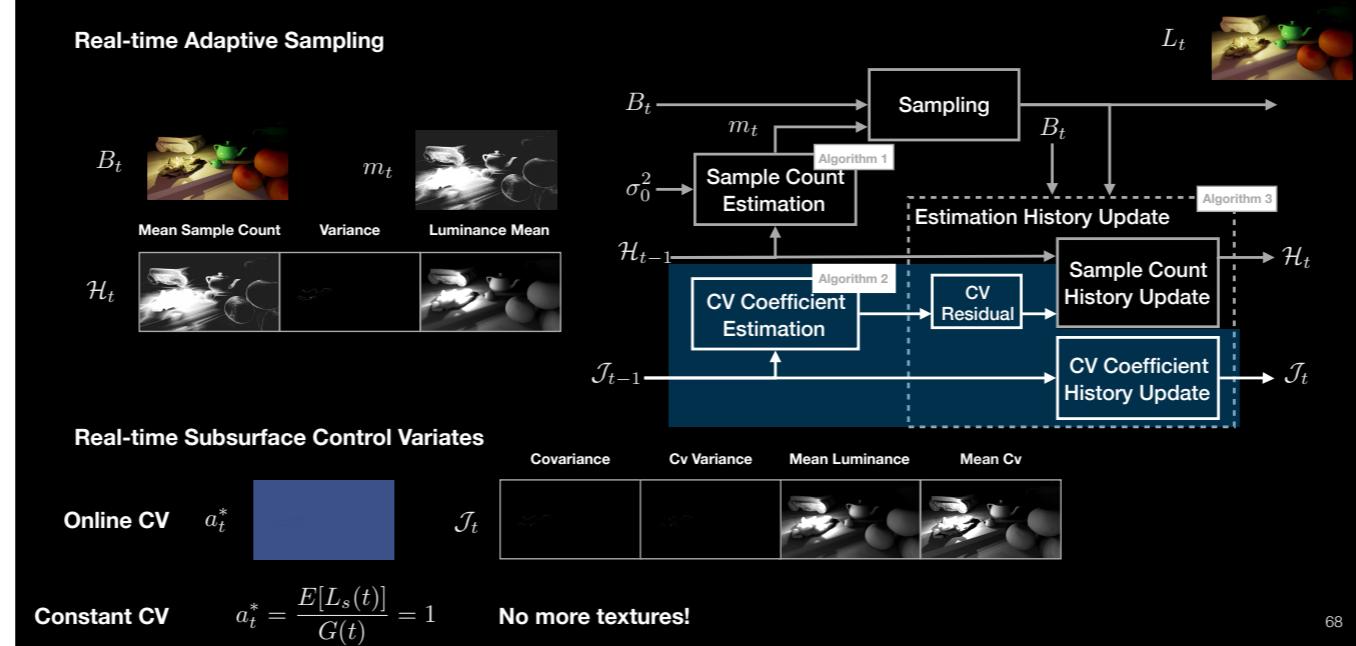
Temporally Stable Adaptive Sampling Algorithm Overview



67

This is our online CV method for subsurface scattering.

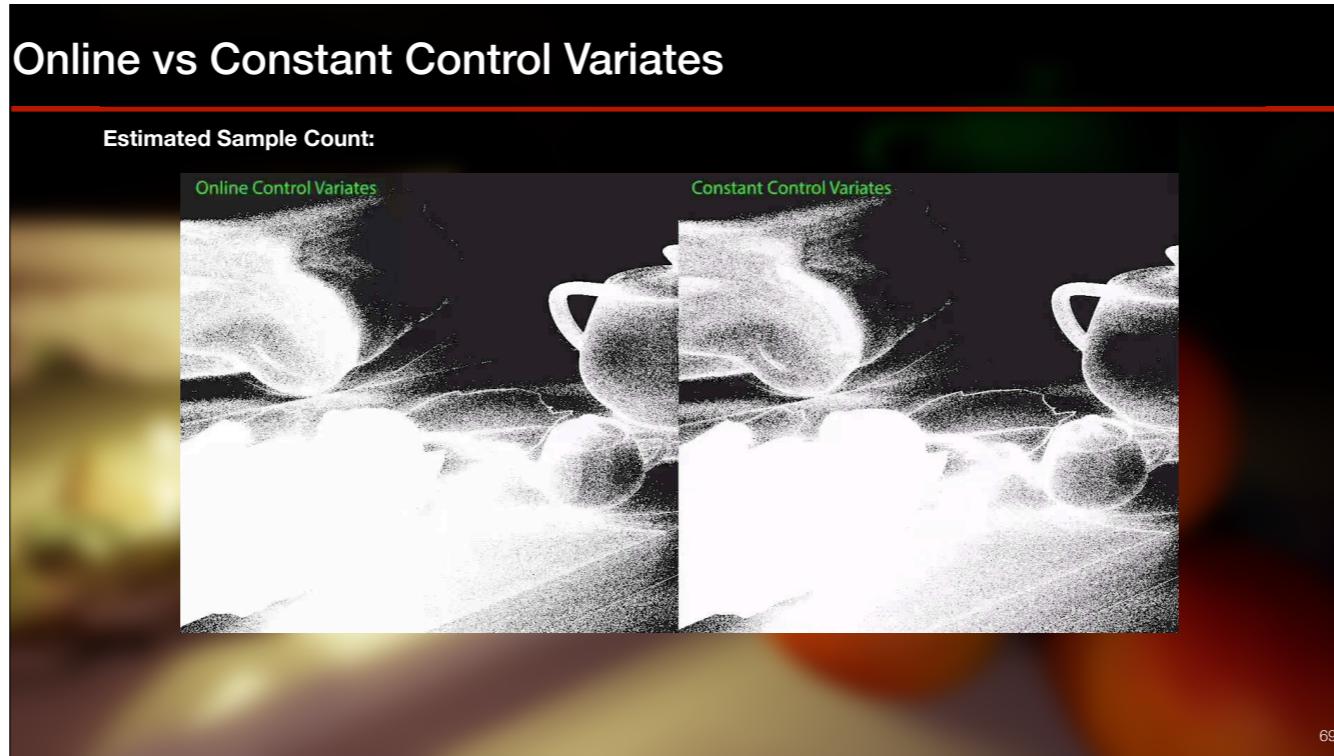
Temporally Stable Adaptive Sampling Algorithm Overview



68

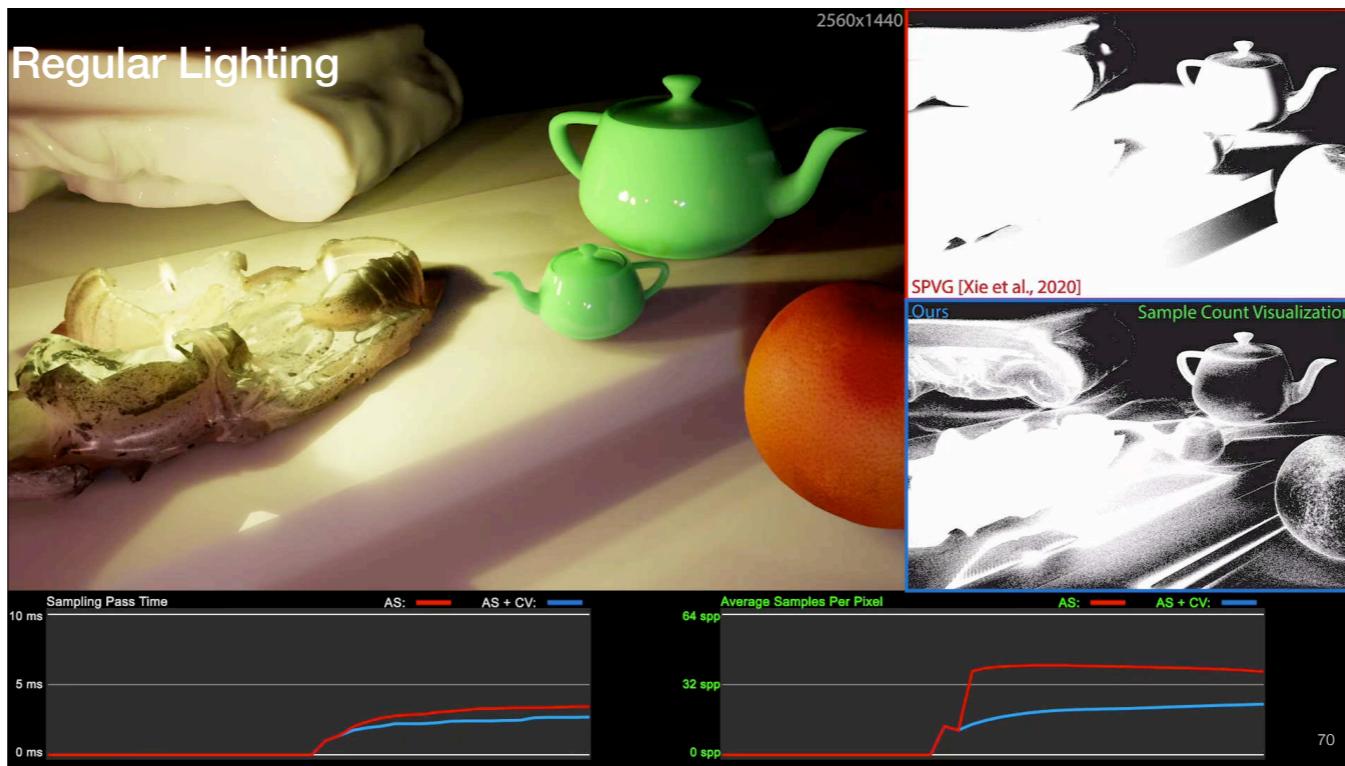
In fact, we can have an estimation of applying constant CV with a coefficient of 1, then no more textures are added. Sample domain instead of shading domain.

Online vs Constant Control Variates

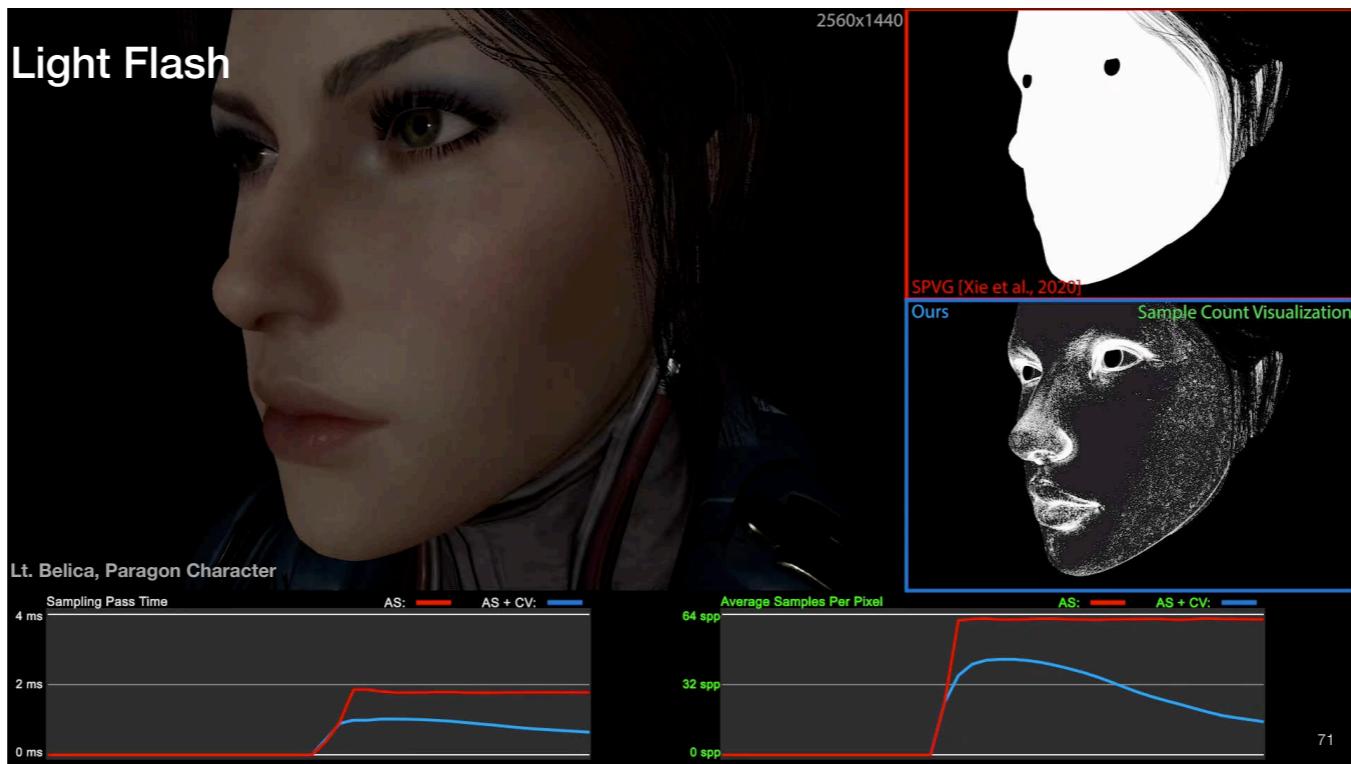


Let's go back to the motivation example again. Here we show the sample count differences between Online and Constant Control Variates. We could observe less samples in Constant CV.

Note: the exponential moving coefficient is 0.000005f for cv coefficient estimation.



In this regular lighting scene, we simulate occasional light change. The sample count visualization of the SPVG 2020 and this paper is also presented. The bottom shows the corresponding sampling pass time and the average samples per pixel.

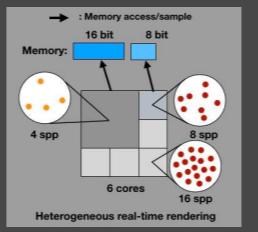


In this light flash scene, we simulate gunshots and lightning with Lt. Belica, a paragon character.

Outline

Section I: Chapter 1 ~ Chapter 3

- Introduction
- Literature
- Motivation
- Heterogeneous Real-time Rendering



Section II: Chapter 4 (I3D'20)

- Real-time Adaptive Sampling

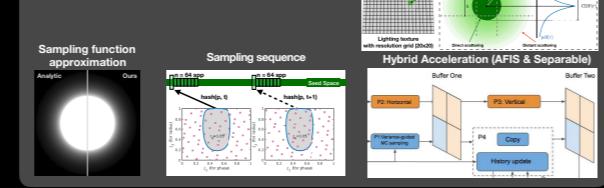
Sampling $O(1)$

Frame i-1
Frame i



Section III: Chapter 5

- Subsurface Scattering



Section IV: Chapter 6 (I3D'21)

- Real-time Control Variates

with one to even zero additional texture



72

This is the summary of the dissertation, I propose novel techniques to make subsurface scattering running efficient for real-time engines. A taxonomy is proposed to help design and illustrate the contribution of this dissertation. Section II introduces a real-time adaptive sampling technique that is in a single pass and runs in $O(1)$ complexity. The local guiding enables different temporal accumulation algorithms including TAA, and DLSS. We have also demonstrated the bandwidth and cache improvements.

Section III introduces the detail of subsurface scattering design. How we approximate the sampling function, what is the sampling sequence, the adaptive filtered information sampling, and the importance-guided acceleration framework.

Section IV details the control variates that enables temporally stable adaptive sampling.

Future Work



Deep Learning

- Global subsurface scattering accumulation



New applications

- Other rendering passes
 - Glossy, ambient occlusion, & PCSS
- Offline rendering



Misc

- Real-time adaptive multiple importance sampling
- In-frame standard control variates
- Reduce memory demands with hybrid representations
- Underestimation study

My Papers from Dissertation

- **Tiantian Xie**, Marc Olano, Brian Karis, and Krzysztof Narkowicz. (2020) Real-time subsurface scattering with single pass variance-guided adaptive importance sampling. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(1): 1–21.
- **Tiantian Xie**, and Marc Olano. (2021). Real-time Subsurface Control Variates: Temporally Stable Adaptive Sampling. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 4(1), 1-18.

Acknowledgements

I own all my gratitude to all the people who made this dissertation possible:

My advisor: Dr. Marc Olano

My Committee members: Dr. Adam Bargteil, Dr. David Chapman, Dr. David Hill and Dr. Matthias Gobbert.

My parents: Zhiqiang Xie, and Birong Zhao

My pet: Eevee (伊布)

Other faculty and staff members, my lab mates and friends

People in the industrial: John Hable, Brian Karis, Patrick Kelly, Krzysztof Narkowicz, Peter Sumanaseni, Mike Seymour etc.

everyone else who helped me throughout the journey!

and Epic Games!



Thank you all :)

Other Publications

1. Wang, W., **Xie, T.**, Liu, X., Yao, Y., & Zhu, T. (2019). ECT: Exploiting cross-technology transmission for reducing packet delivery delay in IoT networks. *ACM Transactions on Sensor Networks (TOSN)*, 15(2), 1-28.
2. Wang, W., **Xie, T.**, Liu, X., & Zhu, T. (2018, April). Ect: Exploiting cross-technology concurrent transmission for reducing packet delivery delay in iot networks. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications* (pp. 369-377). IEEE.
3. Chi, Z., Yao, Y., **Xie, T.**, Liu, X., Huang, Z., Wang, W., & Zhu, T. (2018, November). EAR: Exploiting uncontrollable ambient RF signals in heterogeneous networks for gesture recognition. In *Proceedings of the 16th ACM conference on embedded networked sensor systems* (pp. 237-249).
4. Yao, Y., Li, Y., Liu, X., Chi, Z., Wang, W., **Xie, T.**, & Zhu, T. (2018, April). Aegis: An interference-negligible RF sensing shield. In *IEEE INFOCOM 2018-IEEE conference on computer communications* (pp. 1718-1726). IEEE.
5. Chi, Z., Huang, Z., Yao, Y., **Xie, T.**, Sun, H., & Zhu, T. (2017, May). EMF: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous IoT devices. In *IEEE INFOCOM 2017-IEEE conference on computer communications* (pp. 1-9). IEEE.
6. **Xie, T.**, Huang, Z., Chi, Z., & Zhu, T. (2017, April). Minimizing amortized cost of the on-demand irrigation system in smart farms. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks* (pp. 43-46).
7. Huang, Z., **Xie, T.**, Zhu, T., Wang, J., & Zhang, Q. (2016, December). Application-driven sensing data reconstruction and selection based on correlation mining and dynamic feedback. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 1322-1327). IEEE.
8. Chi, Z., Yao, Y., **Xie, T.**, Huang, Z., Hammond, M., & Zhu, T. (2016, November). Harmony: Exploiting coarse-grained received signal strength from IoT devices for human activity recognition. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)* (pp. 1-10). IEEE.
9. Chen, R., **Xie, T.**, Xie, Y., Lin, T., & Tang, N. (2016, October). Do speech features for detecting cognitive load depend on specific languages?. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (pp. 76-83).
10. Li, S., Yi, P., Huang, Z., **Xie, T.**, & Zhu, T. (2016, September). Energy scheduling and allocation in electric vehicles energy internet. In *2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)* (pp. 1-5). IEEE.

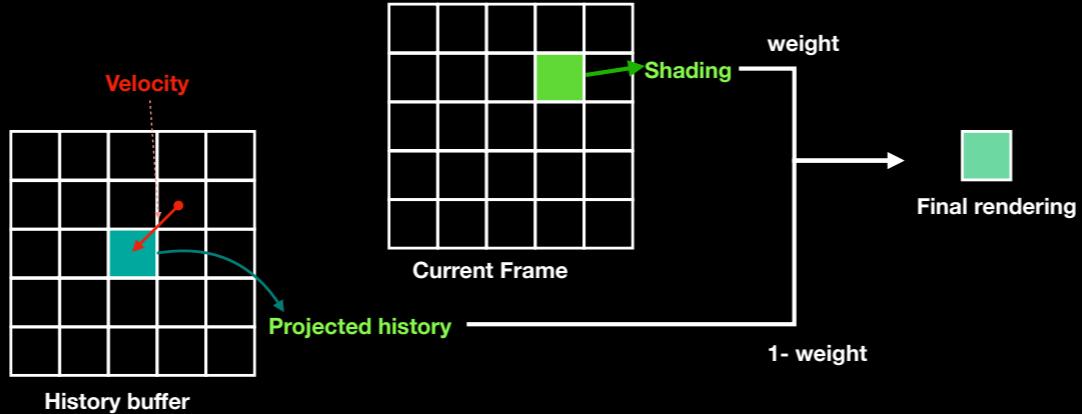
76

This publication list is done during PhD study, but before I come back to real-time rendering, where I am eager to develop my career!

**BACKUP SLIDES SINCE
THIS PAGE**

Temporal accumulation

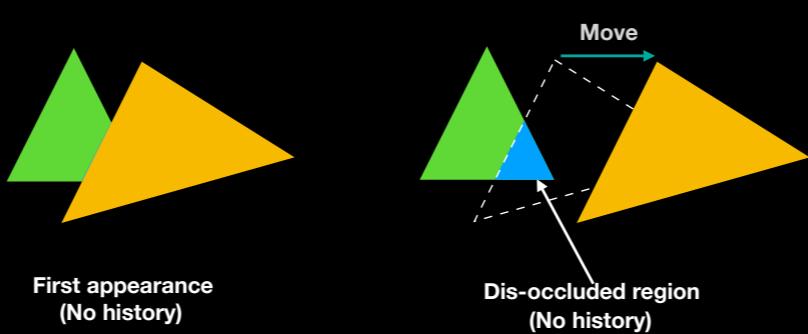
Temporal accumulation similar to Temporal anti-aliasing (TAA) [Karis 2014] for game.



To resolve this issue, we use temporal accumulation similar to temporal anti-aliasing for game. The basic idea for TAA is if in the current frame we have a shading, then we can use per pixel velocity to get the projected history in the history buffer of previous frame. After a weighted sum, we can get the final rendering. In this way, we only need to maintain one history buffer instead of n.

Disocclusion

Disocclusion. How to sample efficiently without history.



79

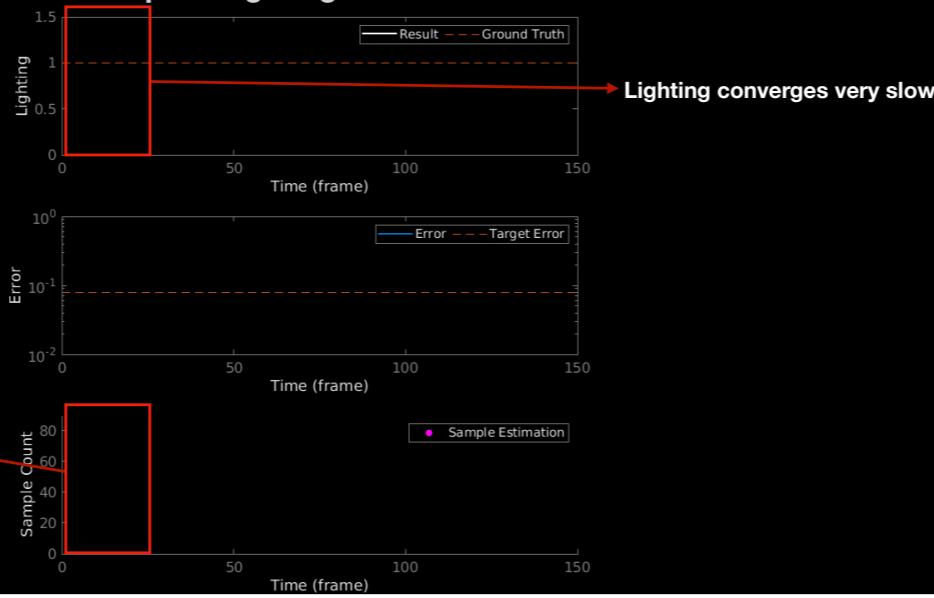
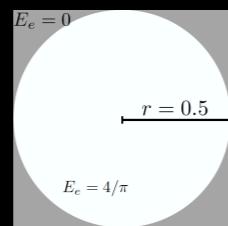
The second challenge we need to handle is dis-occlusion due to first appearance, and motion, where we do not know the history. So how to sample efficiently without history. Let's see what happens when there is no history.

Disocclusion

80

Circle scenario: Sample center point lighting

Irradiance texture



80

Here we sample the lighting of the center point of an irradiance texture. We show the lighting, error, and the estimated sample count over 150 frames. In the first frame, we do not have history.

We could observe first the lighting converges very slow, second, and most importantly, we over estimated the samples. To solve this issue.

Disocclusion

81

Use the target quality (variance) level when there is no history.

1. Update to target variance if no history. Otherwise, use temporal variance.

$$\hat{\sigma}_i^2(\alpha_0, \Lambda) = \begin{cases} \sigma_0^2 & C_s(x_i, \Lambda) = 0 \text{ no history} \\ \sigma_i^2 & \text{otherwise} \end{cases}$$

target variance:
 $\sigma_0^2 = 0.08^2$

2. Use the new frame rendering when there is no history. Otherwise use temporal accumulation.

$$\mathcal{M}'(\alpha_0, \Lambda) = \begin{cases} 1 & C_s(x_i, \Lambda) = 0 \text{ no history} \\ \alpha_0 & \text{otherwise} \end{cases}$$

81

We use the target quality level when there is no history. It composes of two steps.

First, use the target variance if no history. Otherwise, use temporal variance.

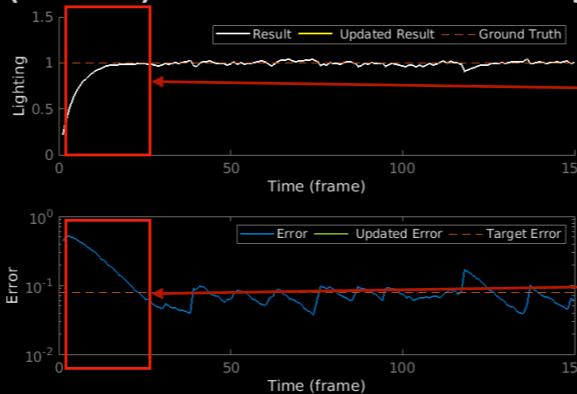
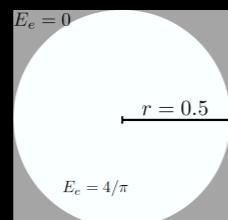
Second, use the new shading when there is no history. Otherwise use temporal accumulation.

Disocclusion

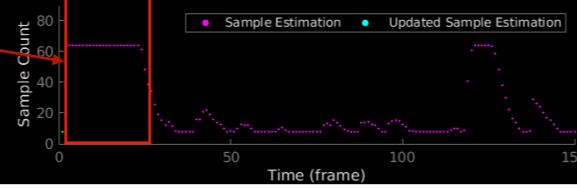
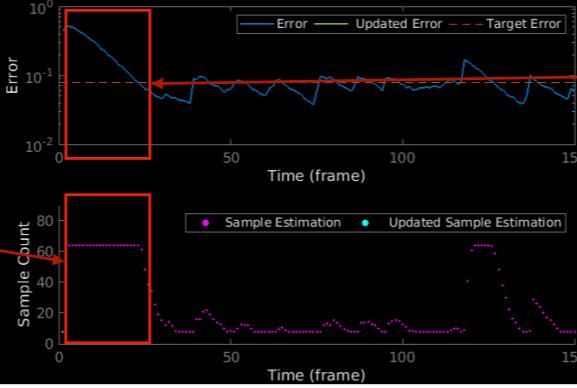
82

Use the target quality (variance) level when there is no history.

Irradiance texture



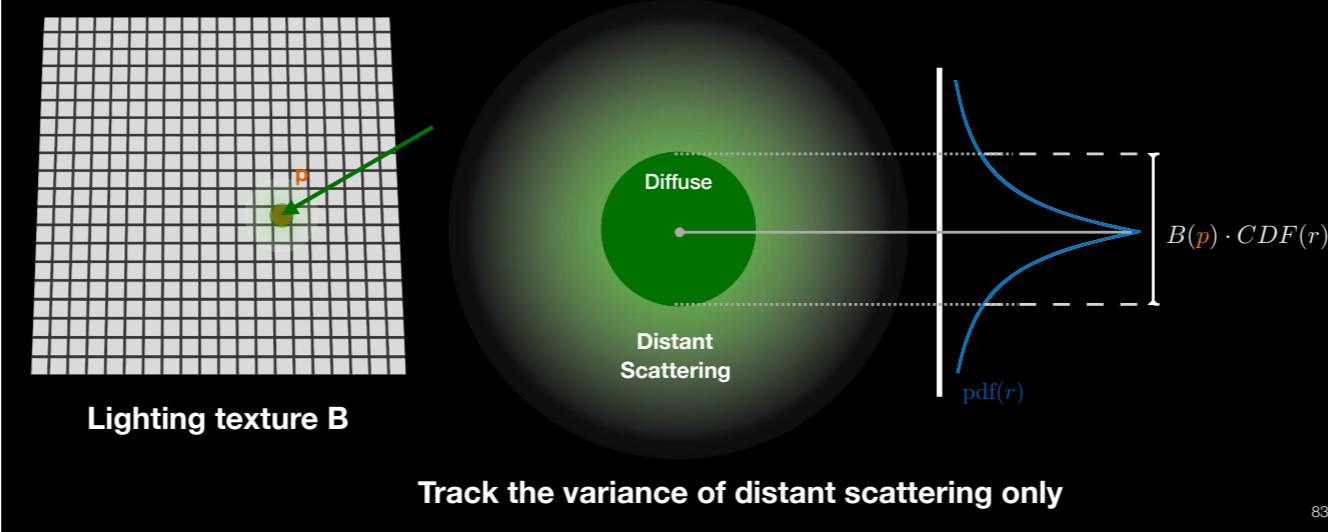
Overestimation resolved



82

With this modification. Let's see what changes for the lighting, error, and sample count. Lighting converges faster. We also have better error estimation. Most importantly, the overestimation has been resolved.

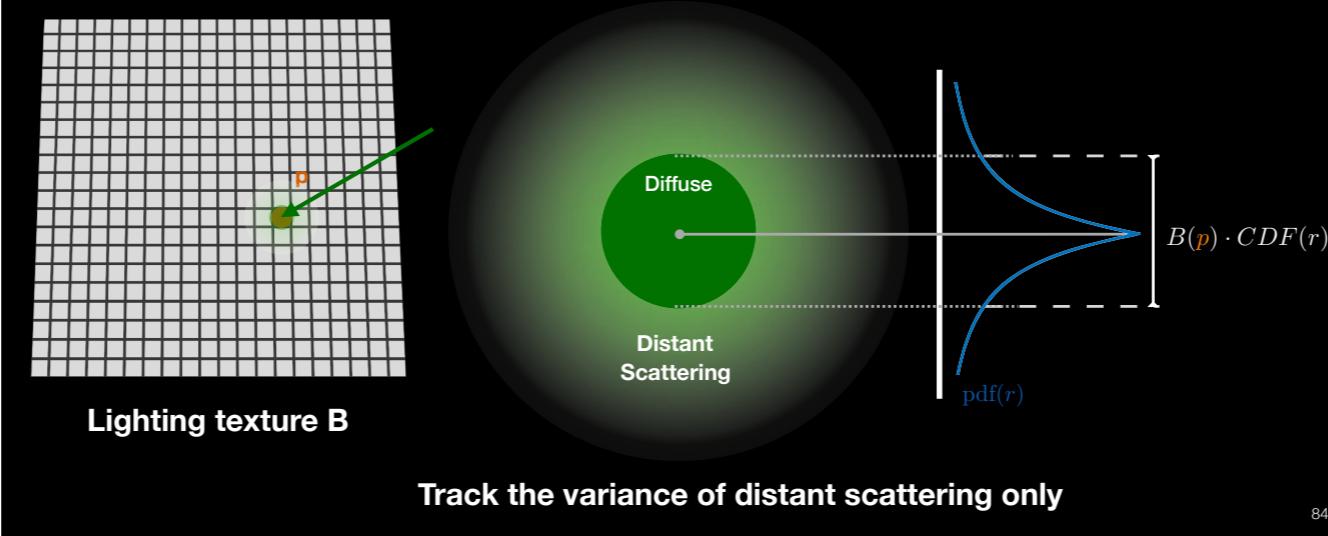
Separating Direct and Distant Scattering



83

In order to shade the subsurface scattering for a point p , we apply a filter kernel onto the pre-integrated lighting texture, where we can divide the applied region into two parts: the direct scattering region, and distant scattering. Since the pre-integrated lighting texture in reality is actually discrete, the direct lighting region can be approximated by a pixel value as constant diffuse. More specifically in the importance sampling framework, the scattering result is the discrete lighting at that pixel by the CDF, which is a constant. Then we only need to track the variance of distant scattering and apply CV there. and estimate sample counts for distant scattering.

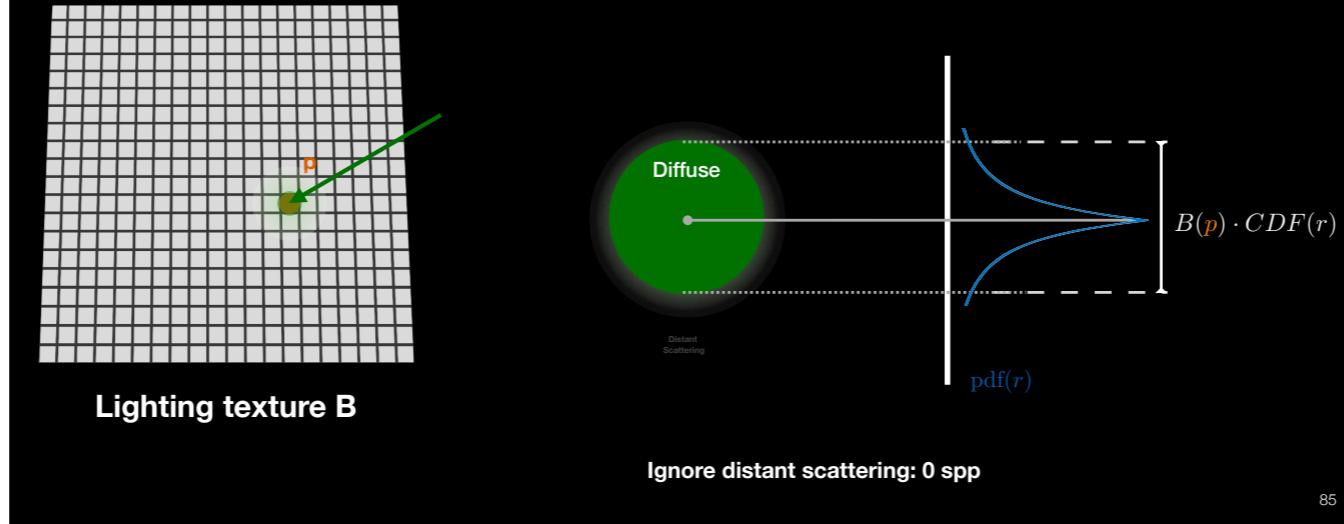
Separating Direct and Distant Scattering



84

In order to shade the subsurface scattering for a point p , we apply a filter kernel onto the pre-integrated lighting texture, where we can divide the applied region into two parts: the direct scattering region, and distant scattering. Since the pre-integrated lighting texture in reality is actually discrete, the diffuse lighting region can be approximated by a pixel value as constant. More specifically in the importance sampling framework, the scattering result is the discrete lighting at that pixel by the CDF, which is a constant. Then we only need to track the variance of distant scattering, apply CV there, and estimate sample counts for distant scattering. This change of variance tracking has two implications. First if the distant scattering energy is very small,

Separating Direct and Distant Scattering



85

We could actually ignore it and achieve zero distant scattering samples in the same model.

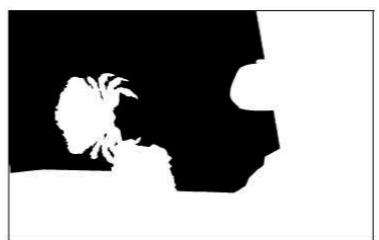
Separating Direct and Distant Scattering



(a) Scene



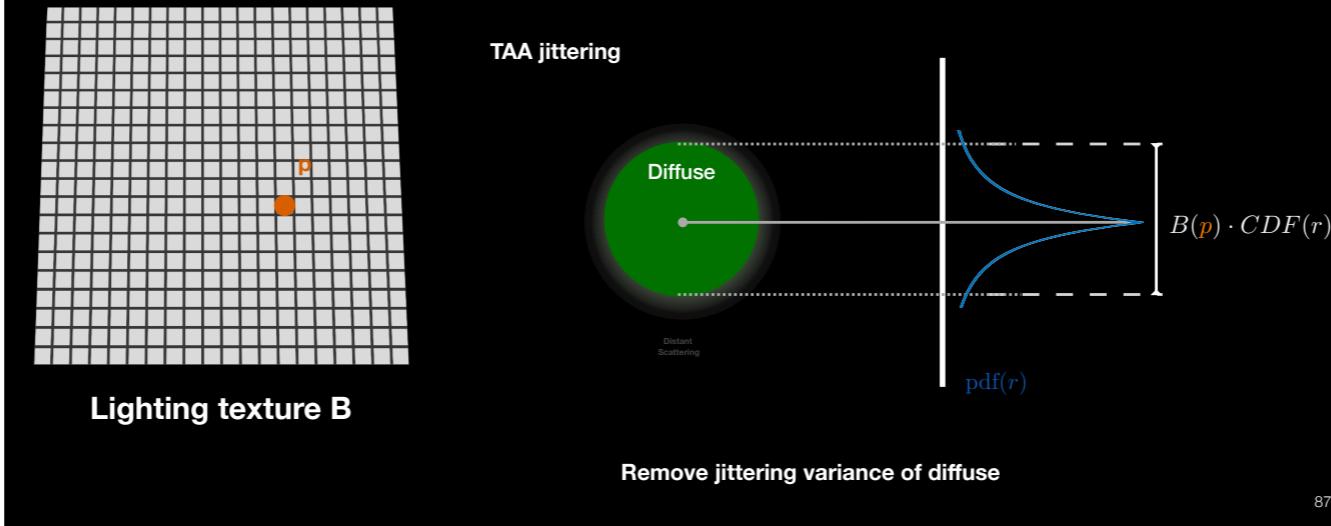
(b) $\epsilon_u = 0.0$



(c) $\epsilon_u = 0.05$

Direct/diffuse region (black) and direct+distant (white) (b,c) for scene (a). The vertical line on the wall (c) is the boundary where only 5% of scattering energy is from distant scattering.

Separating Direct and Distant Scattering



Second, TAA creates jittered samplings, which could bring variance even for diffuse. Separating the scattering and only monitor the variance for distant scattering helps to remove sample counts due to jittering, like at the silhouette edge during stable lighting.

In-frame Constant CV

When T is independent from F, G

$$\begin{aligned} \text{Var}(\langle TF \rangle) &= \text{Var}(\langle TF - aTG \rangle + aTG) \\ &= \text{Var}(T(F - aG)) + \text{Var}(aTG) + 2\text{Cov}(T(F - aG), aTG) \\ \text{Var}(T(F - aG)) &= E[T]^2 \text{Var}(F - aG) + \text{Var}(T) \text{Var}(F - aG) + \text{Var}(T)E[F - aG]^2 \end{aligned}$$

In-frame constant CV:

$$a = \frac{\text{Cov}(TF, TG)}{\text{Var}(TG)} \xrightarrow{\text{Var}(G) = 0} a = \frac{E[F]}{E[G]} = \frac{\int_0^1 \frac{1}{1+y} dy}{\int_0^1 1-y dy} = \ln(4) \approx 1.386 \quad a^* \approx 1.388 \quad (1024 \text{ spp})$$

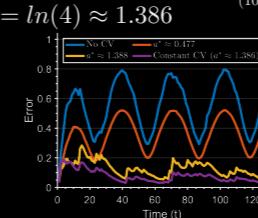
Simplification:

(Variance of CV residual)

$$\begin{aligned} \text{Var}(\langle TF \rangle) &= \text{Var}(T(F - aG)) + \text{Var}(T)E[F]^2 \\ \text{Var}(T(F - aG)) &= E[T]^2 \text{Var}(F) + \text{Var}(T) \text{Var}(F) \end{aligned}$$

Sample Estimation - monitor variance of CV residual:

$$\text{Var}(\langle TF \rangle) = [\text{Var}(T) + E[T]^2] \text{Var}(F) + \text{Var}(T)E[F]^2 \quad \begin{matrix} \text{Sample Count} \uparrow \\ \text{Decrease} \downarrow \quad \text{Unaffected} \end{matrix}$$



88

So why we can use residual variance to monitor the spatial variance. First, we can decompose the variance with the three random variable : T, F, and G into the variance of CV redidual and other terms. With in-frame constant CV, we have the coefficient $a = E[F]/E[G]$. After simplification, with this formula, we could arrive at the formula by the green and red, where the green term is the CV coefficient, where we could ignore the variance of tempo in the read term, which will not change even when the sample count increases.

Static Scene with Jittering

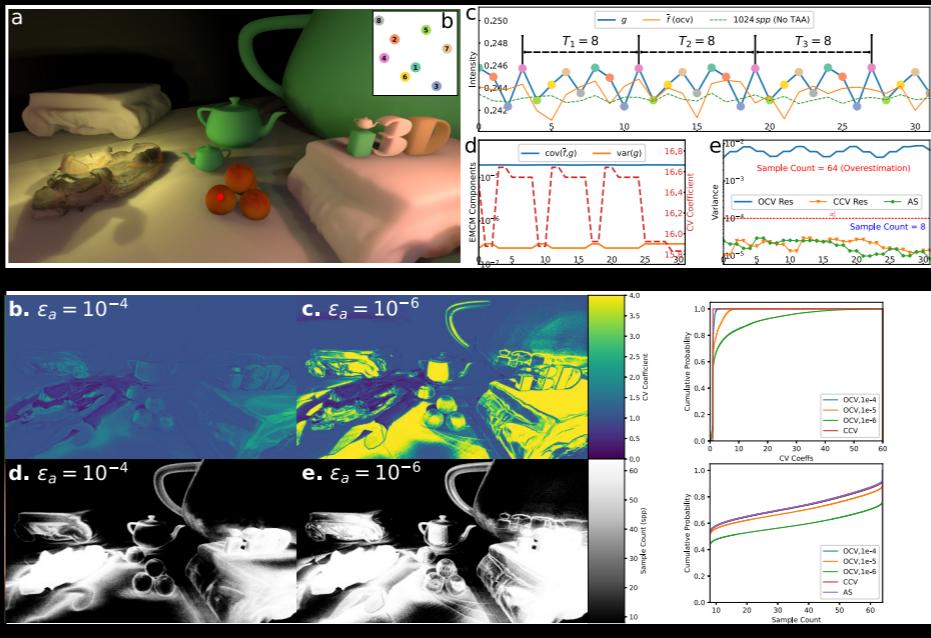
89

CV Coefficient estimation:

$$a_t = \frac{\Sigma_t \cdot xy}{\Sigma_t \cdot yy}$$

With suppressing factor:

$$a_t = \frac{\Sigma_t \cdot xy + \epsilon_a}{\Sigma_t \cdot yy + \epsilon_a}$$



There is one special consideration for static scene with jittering. There jittering will create a periodic changing control variable, which is small. The EMCM could estimate a large CV coefficient based on this small change, which will lead to overestimation. We can use the suppressing factor to deal with this oversampling.

Reference

- [Mori et al. 2012] Mori, Masahiro, Karl F. MacDorman, and Norri Kageki. "The uncanny valley [from the field]." *IEEE Robotics & Automation Magazine* 19.2 (2012): 98-100.
- [Jensen et al. 2001] Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan. 2001. A practical model for subsurface light transport. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, New York, NY, 511-518.
- [d'Eon et al 2007] Eugene d'Eon, David Luebke, and Eric Enderton. 2007. Efficient rendering of human skin. In Proceedings of the 18th Eurographics conference on Rendering Techniques. Eurographics Association, Aire-la-Ville, Switzerland, 147-157.
- [Penner and Borshukov 2011] Eric Penner and George Borshukov. 2011. Pre-integrated skin shading. *Gpu Pro* 2 (2011), 41-55.
- [Jimenez et al. 2015] Jorge Jimenez, Károly Zsolnai, Adrian Jarabo, Christian Freude, Thomas Auzinger, Xian-Chun Wu, Javier von der Pahlen, Michael Wimmer, and Diego Gutierrez. 2015. Separable Subsurface Scattering. *Computer Graphics Forum* 34, 6 (Sept. 2015), 188-197.
- [Christensen and Burley 2015] Per H. Christensen and Brent Burley. 2015. Approximate Reflectance Profiles for Efficient Subsurface Scattering. Technical Report. Pixar.
- [Golubev 2018] Evgenii Golubev. 2018. Efficient screen-space subsurface scattering using Burley's normalized diffusion in real-time. Retrieved Aug 29, 2019 from <http://advances.realtimerendering.com/s2018/Efficient%20screen%20space%20subsurface%20scattering%20Siggraph%202018.pdf>
- [Finch 2009] Tony Finch. 2009. Incremental calculation of weighted mean and variance. University of Cambridge 4, 11-5 (2009), 41-42.
- [Bauer and Dahlquist 1998] Richard J Bauer and Julie R Dahlquist. 1998. Technical Markets Indicators: Analysis & Performance. Vol. 64. John Wiley & Sons.
- [Bakhoda et al., 2009] Ali Bakhoda, George L Yuan, Wilson WL Fung, Henry Wong, and Tor M Aamodt. Analyzing CUDA workloads using a detailed GPU simulator. In 2009 IEEE International Symposium on Performance Analysis of Systems and Software, pages 163-174. IEEE, 2009.
- [Golubev 2019] Evgenii Golubev. 2019. Sampling Burley's Normalized Diffusion Profiles. Retrieved Nov 17, 2019 from <https://zero-radiance.github.io/post/sampling-diffusion/>

Reference

- [Krivánek and Colbert 2008] Jaroslav Krivánek and Mark Colbert. 2008. Real-time shading with filtered importance sampling. Computer Graphics Forum 27, 4 (2008), 1147–1154.
- [Khairy et al., 2020] Mahmoud Khairy, Zhesheng Shen, Tor M Aamodt, and Timothy G Rogers. Accel-Sim: An extensible simulation framework for validated GPU modeling. In 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), pages 473–486. IEEE, 2020.
- [Roberts, 2018] Martin Roberts. The Unreasonable Effectiveness of Quasirandom Sequences, 2018. URL <http://extremelearning.com.au/unreasonable-effectiveness-of-quasirandom-sequences/>.
- [Jarzynski and Olano, 2020] Mark Jarzynski and Marc Olano. Hash Functions for GPU Rendering. Journal of Computer Graphics Techniques (JCGT), 9(3):20–38, October 2020. ISSN 2331- 7418. URL <http://jcgtr.org/published/0009/03/02/>.
- [Xie et al., 2020] Tiantian Xie, Marc Olano, Brian Karis, and Krzysztof Narkowicz. Real-time subsurface scattering with single pass variance-guided adaptive importance sampling. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 3(1): 1–21, 2020.
- [Karis 2014] Brian Karis. 2014. High Quality Temporal Supersampling. Retrieved Aug 29, 2019 from <http://advances.reallimerendering.com/s2014/epic/TemporalAA.ppt>