Cache and Bandwidth Aware Real-time Subsurface Scattering

UMBC

PhD candidate:

Tiantian Xie

Advisor:

Dr. Marc Olano

Committee Members:

Dr. Adam Bargteil

Dr. David Chapman

Dr. David Hill

Dr. Matthias Gobbert

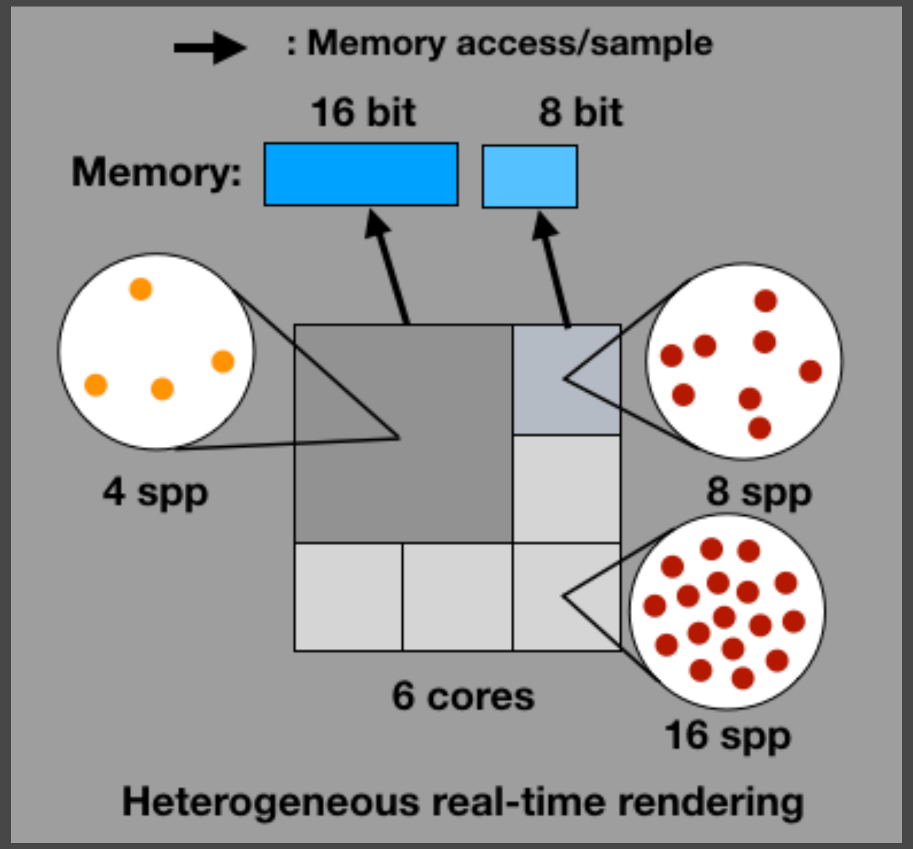
21/06/2021 Tiantian Xie PhD dissertation defense

Metahuman rendered in UE4
with dissertation algorithm

Outline

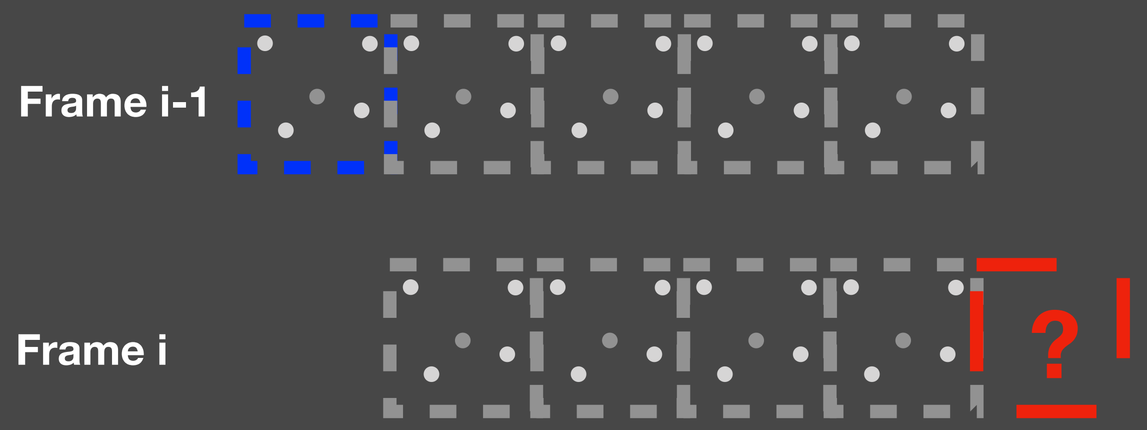
Section I: Chapter 1 ~ Chapter 3

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



Section II: Chapter 4 (I3D’20)

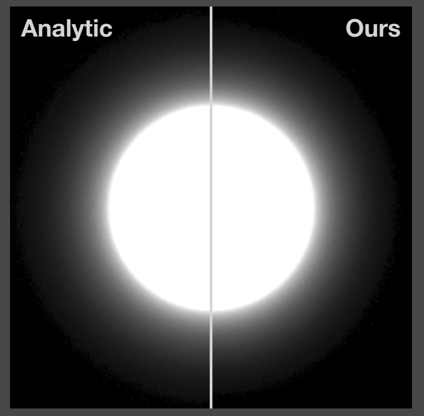
- Real-time Adaptive Sampling $O(1)$



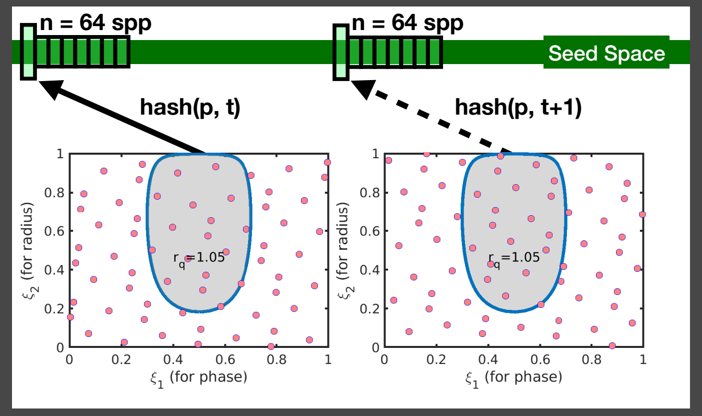
Section III: Chapter 5

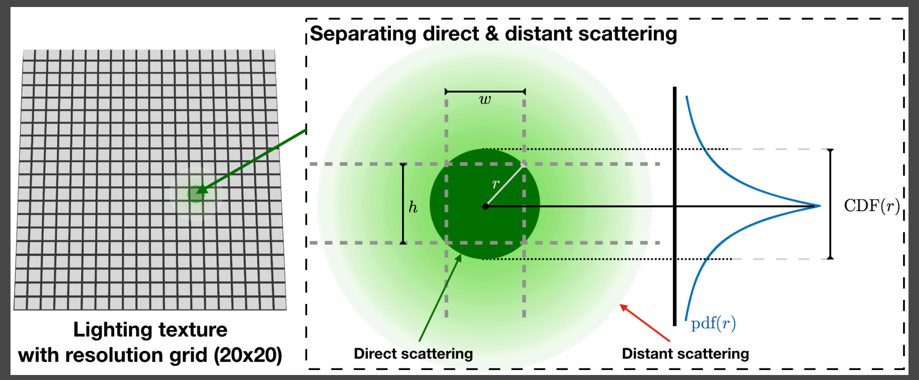
- Subsurface Scattering

Sampling function approximation

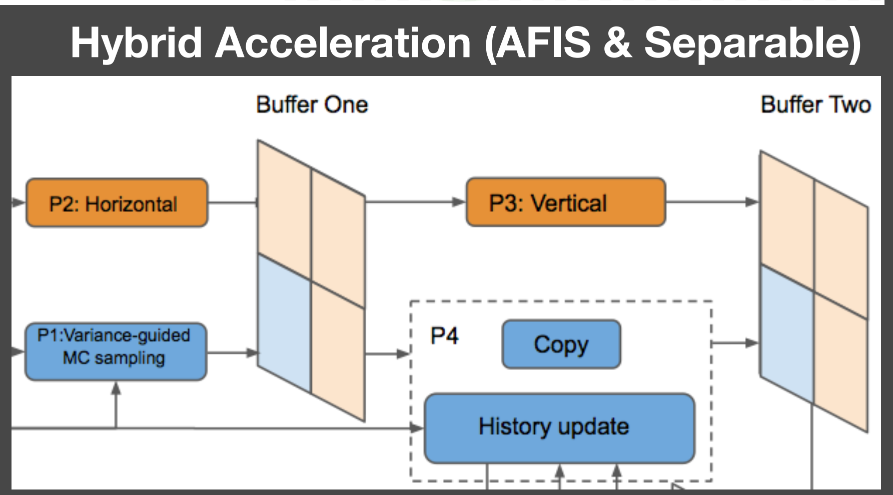


Sampling sequence





Hybrid Acceleration (AFIS & Separable)

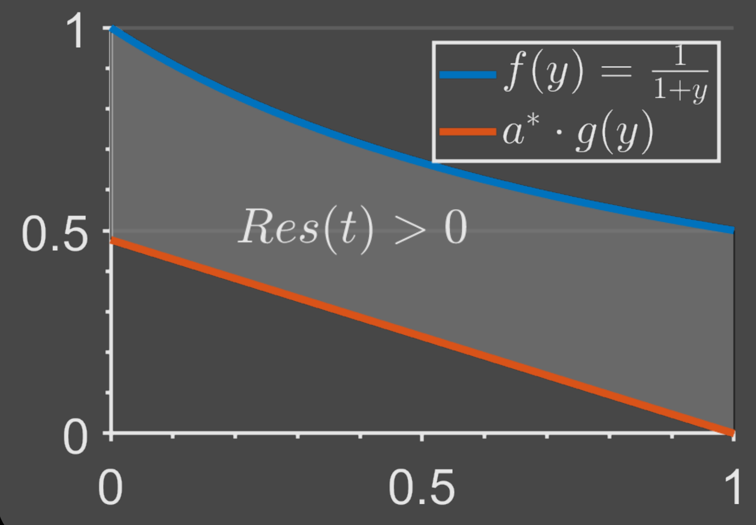


Section IV: Chapter 6 (I3D’21)

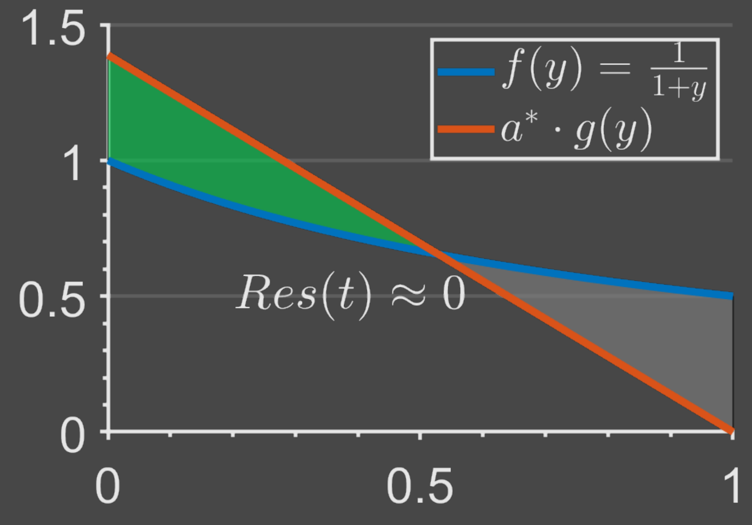
- Real-time Control Variates

with one to even zero additional texture

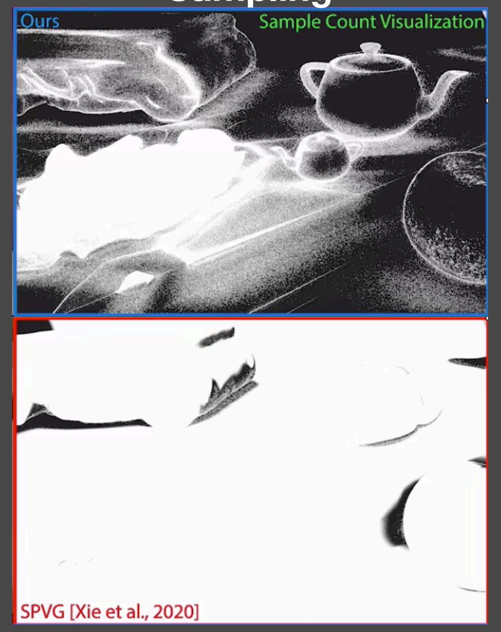
Without temporal info



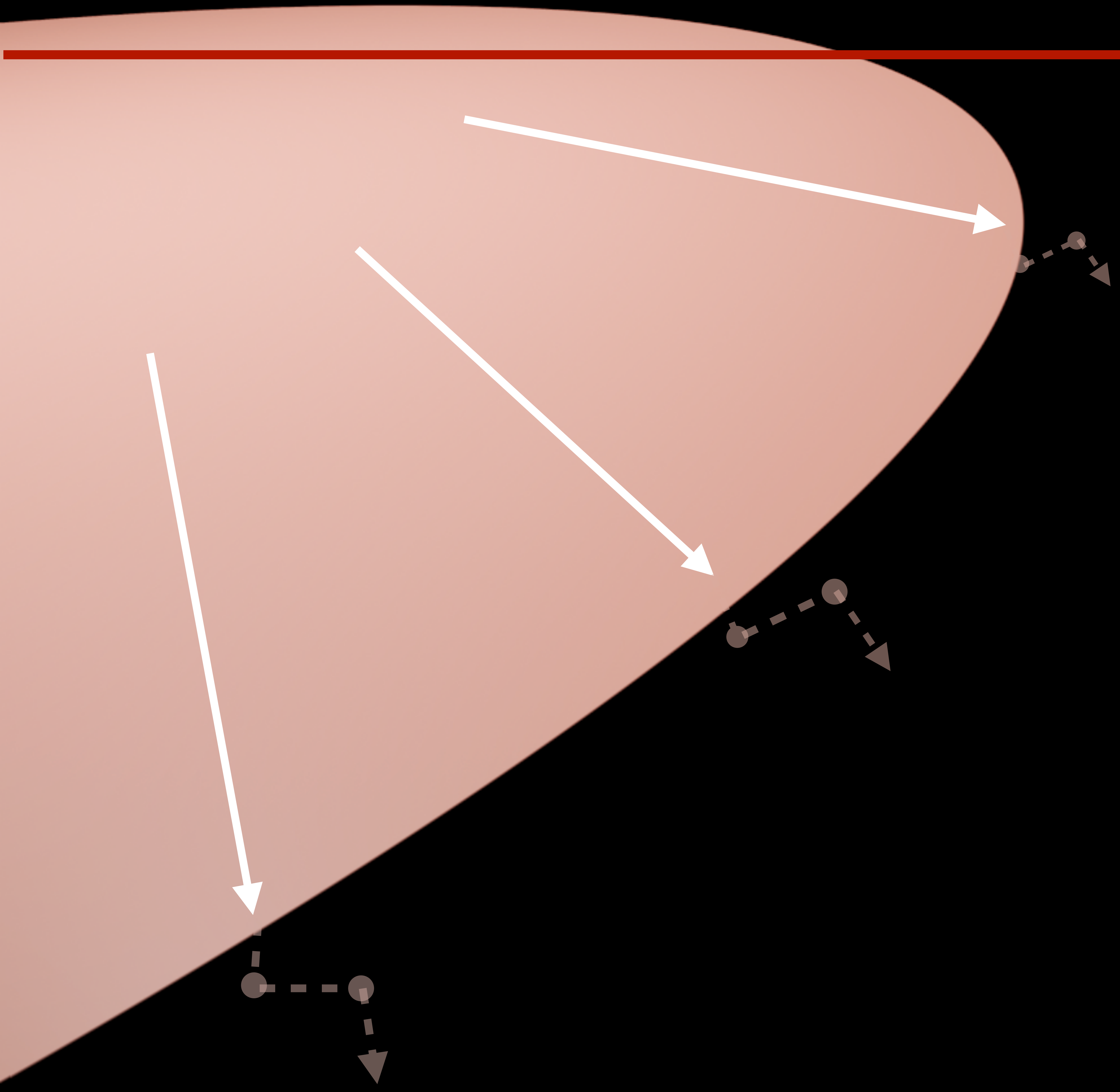
With temporal info



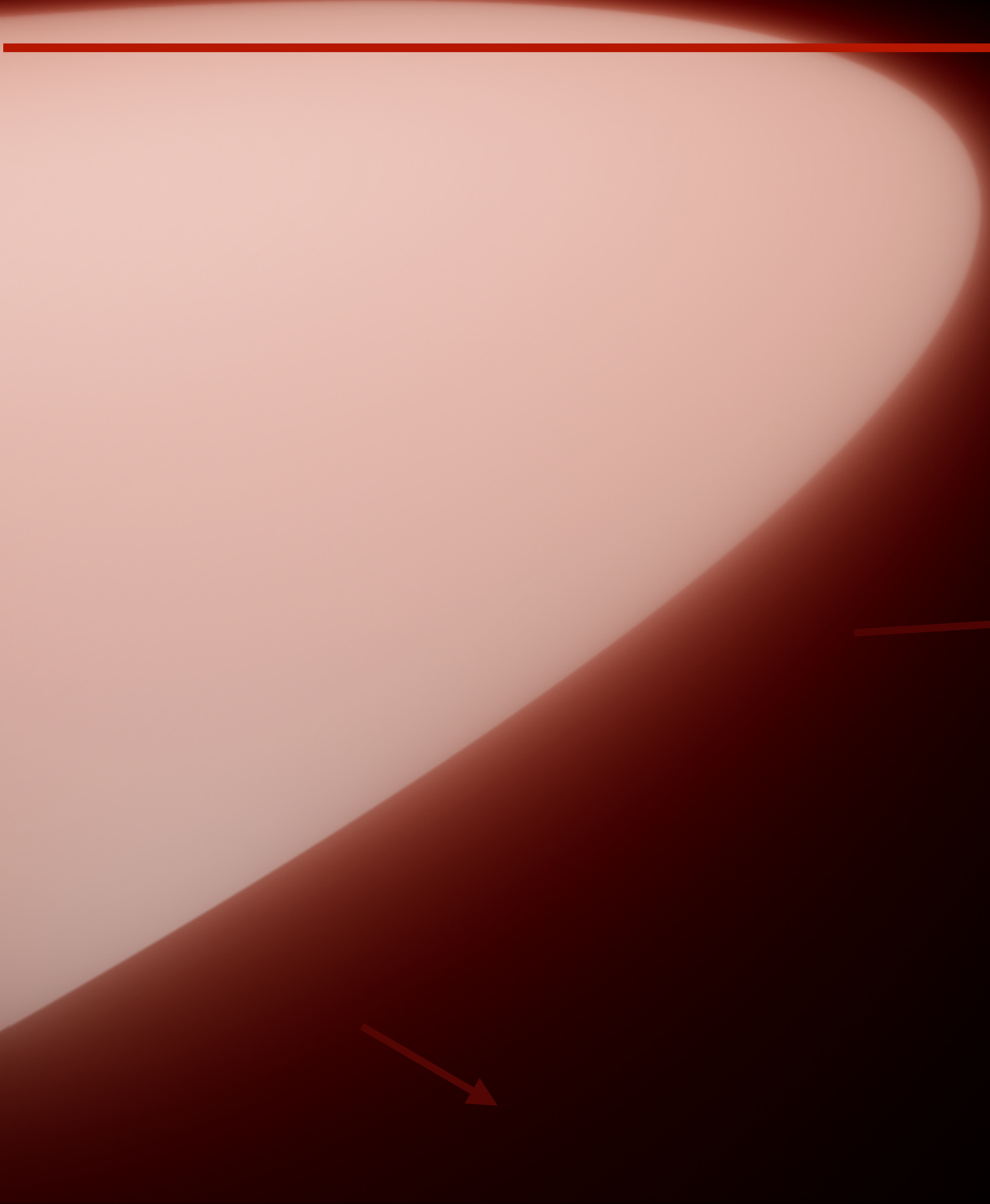
Temporally Stable Adaptive Sampling



Subsurface Scattering



Subsurface Scattering



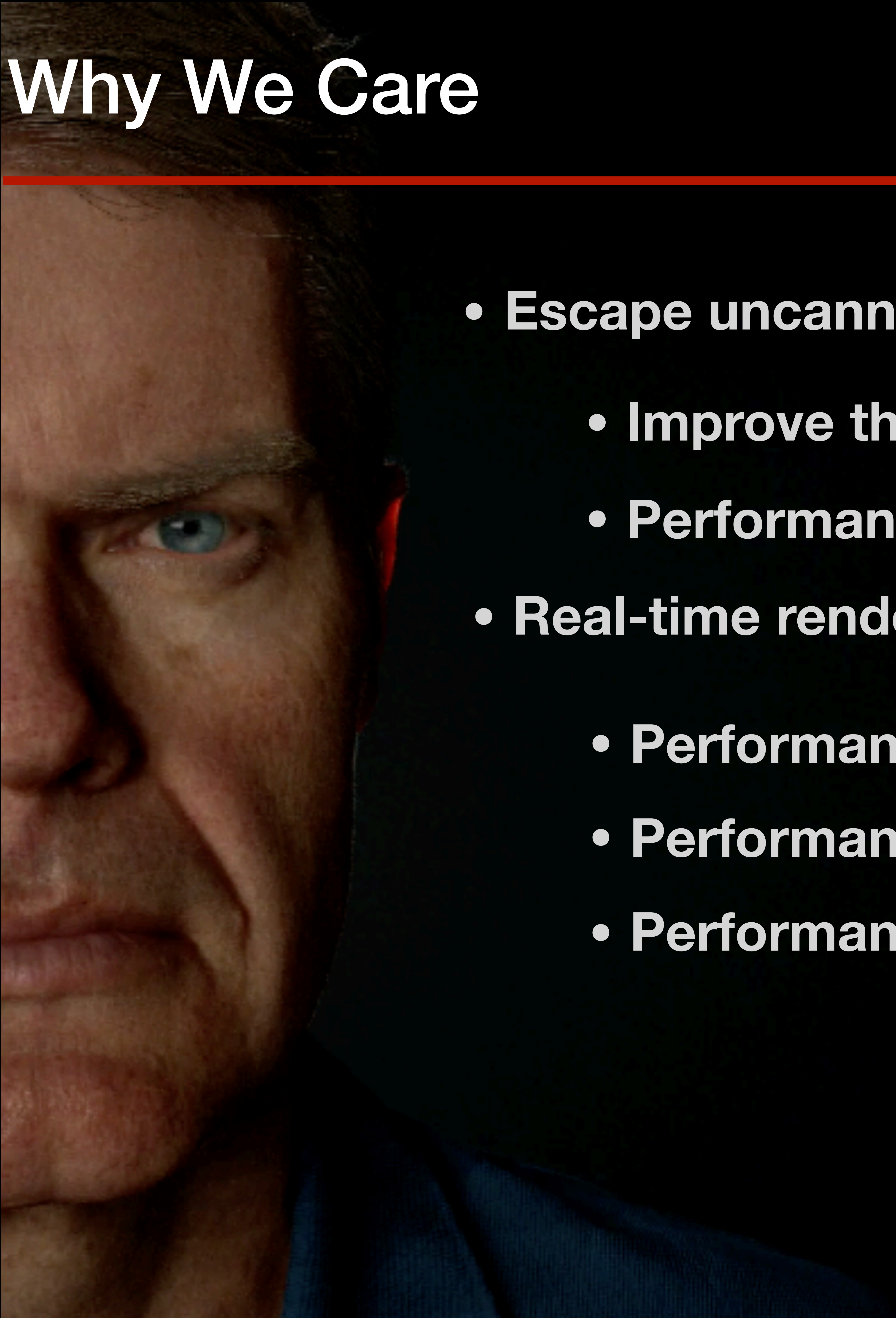
Paul Harrison CC BY 2.0

Scott Feldstein CC BY 2.0

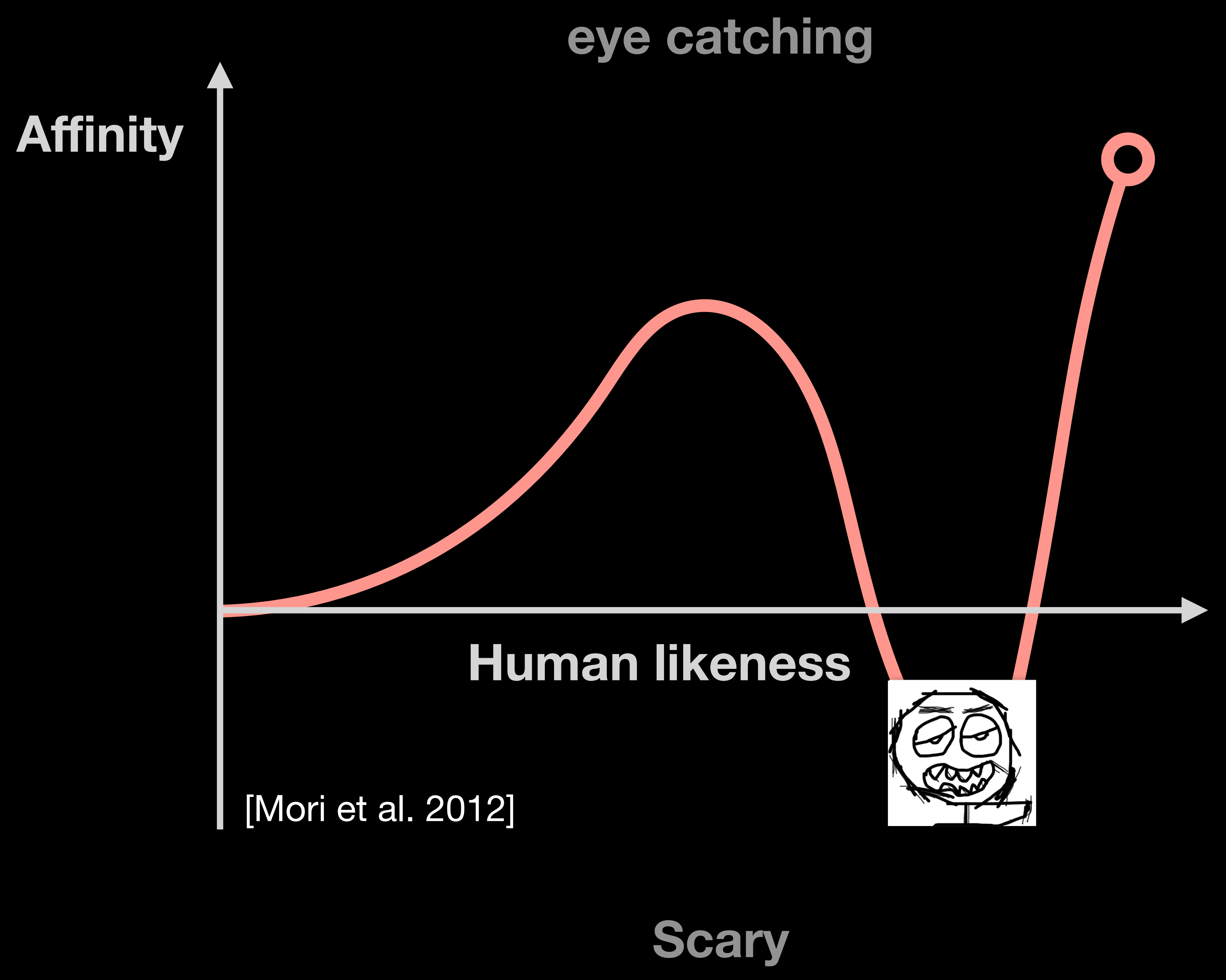
Mike Beauregard CC BY 2.0

Photorealistic Digital Human

Why We Care



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



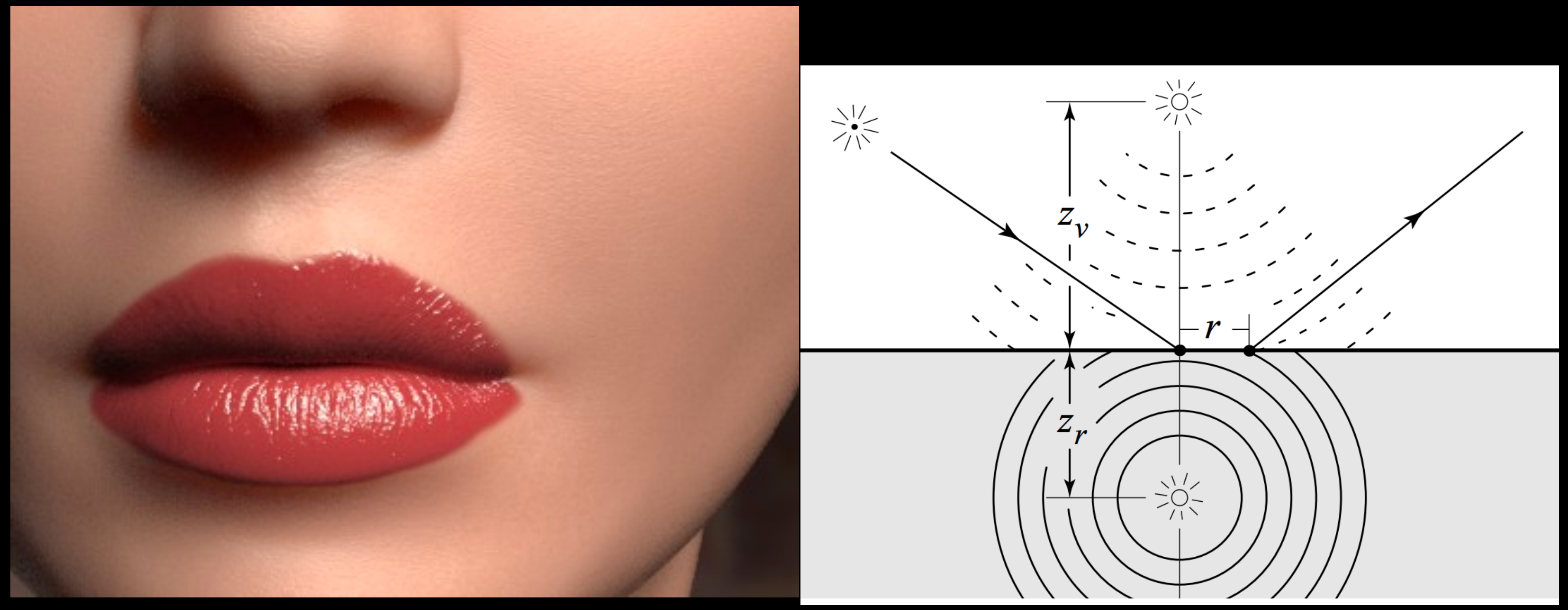
[Mori et al. 2012]

Real-time Subsurface Scattering - Literature Review

2001

Offline

Dipole diffusion profile



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

Real-time Subsurface Scattering - Literature Review

2001

2005

Offline

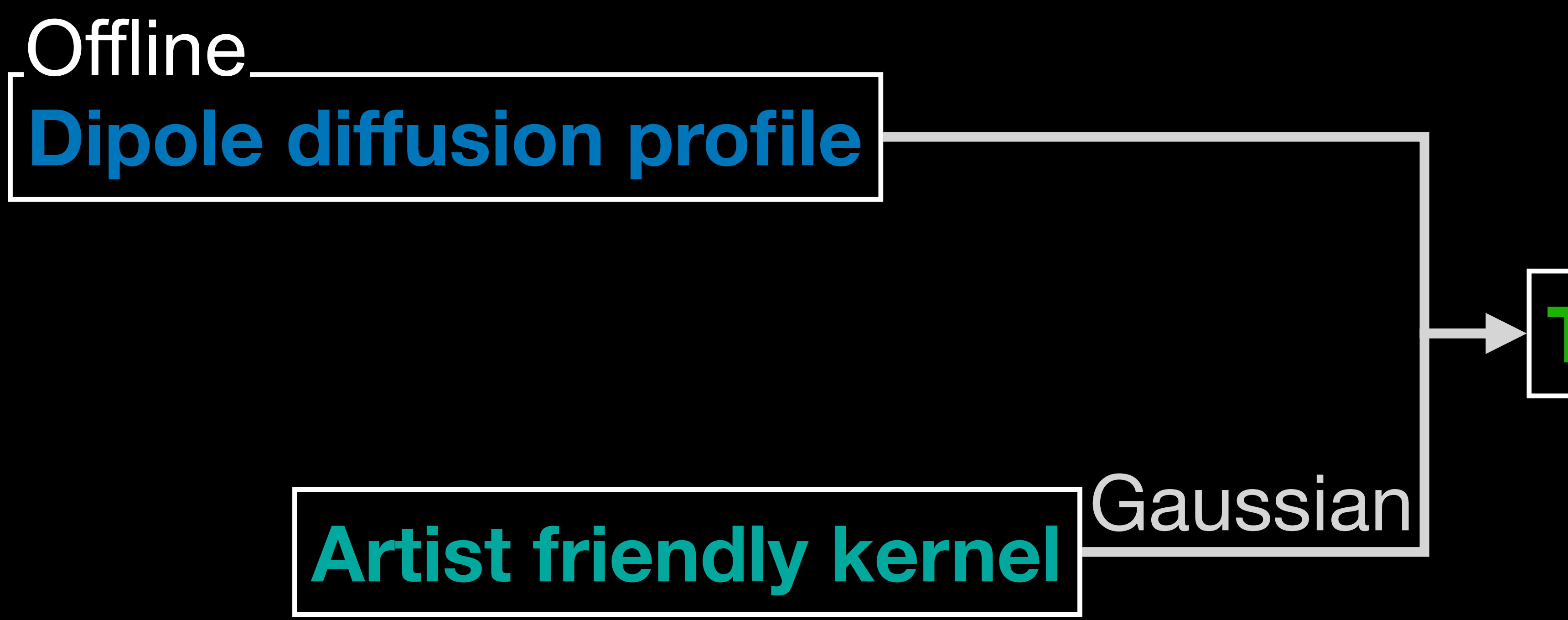
Dipole diffusion profile

Artist friendly kernel

Blur irradiance for human face in “The Matrix Reloaded” [Borshukov and Lewis 2005]

Real-time Subsurface Scattering - Literature Review

2001
2005
2007



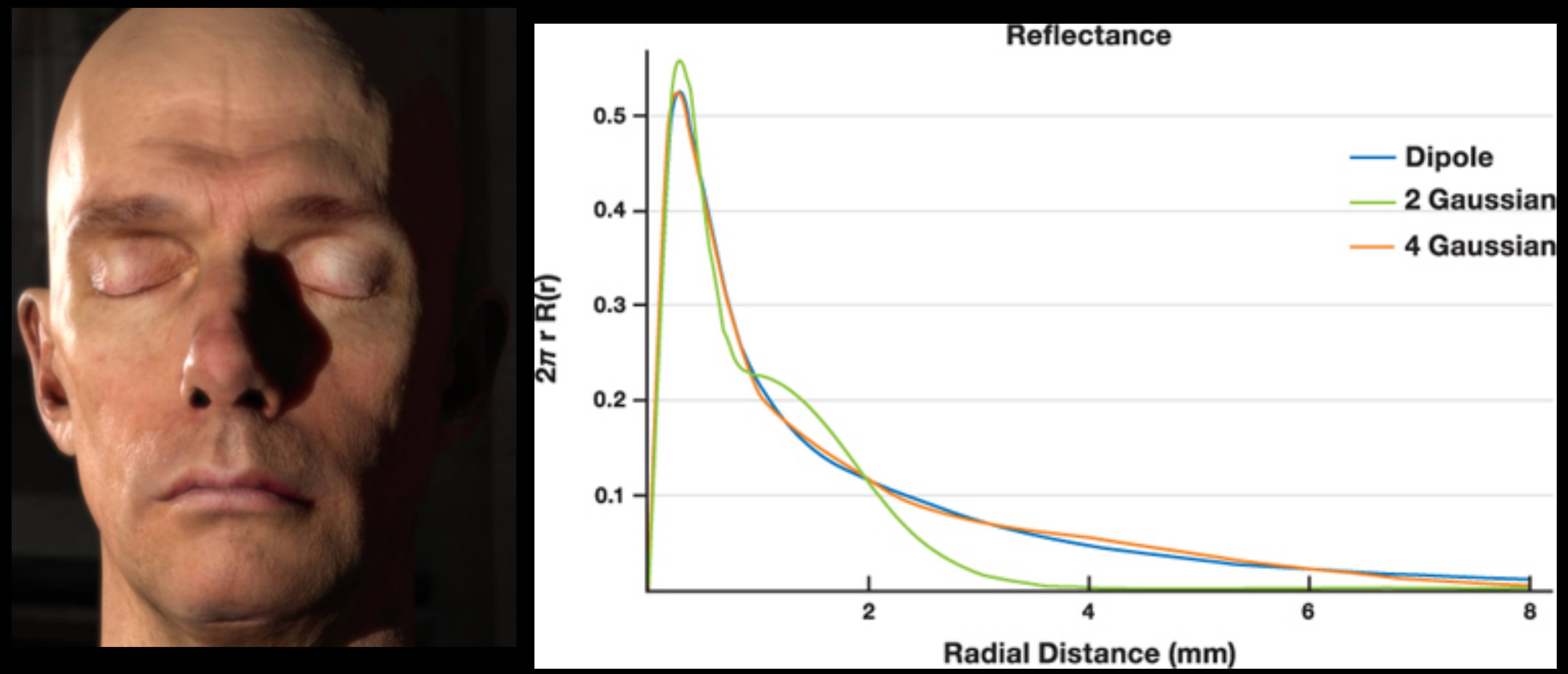
Offline

Dipole diffusion profile

Gaussian

Artist friendly kernel

Texture space



Approximate dipole with Gaussian [d'Eon et al 2007]

Real-time Subsurface Scattering - Literature Review

2001

2005

2007

2011

Pre-Integration

Offline

Dipole diffusion profile

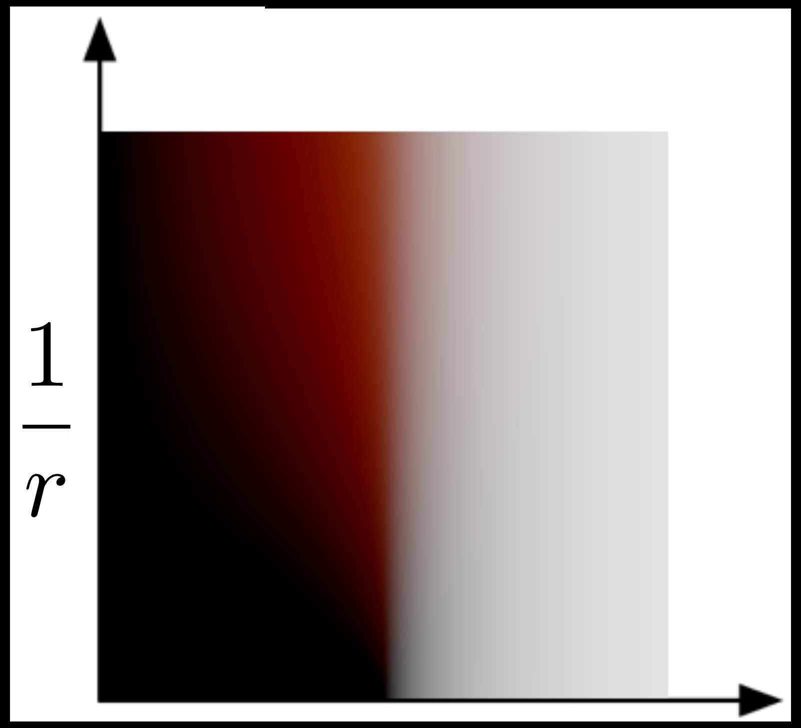
Texture space

Gaussian

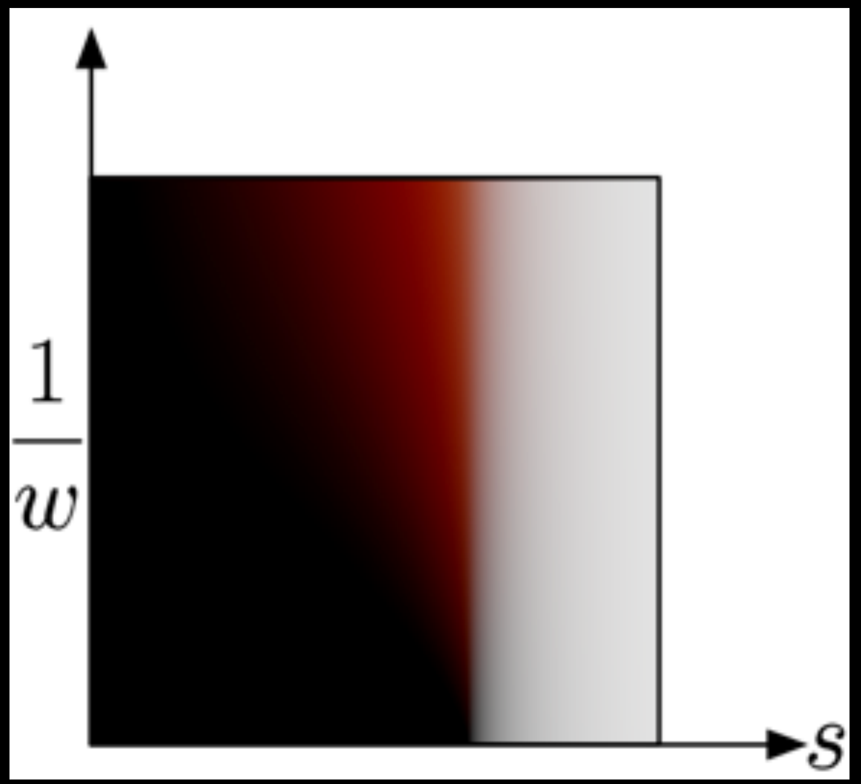
Artist friendly kernel

$N+\theta$
N
$2r\sin(\theta/2))$
L

Pre-integrated skin shading [Penner and Borshukov 2011]

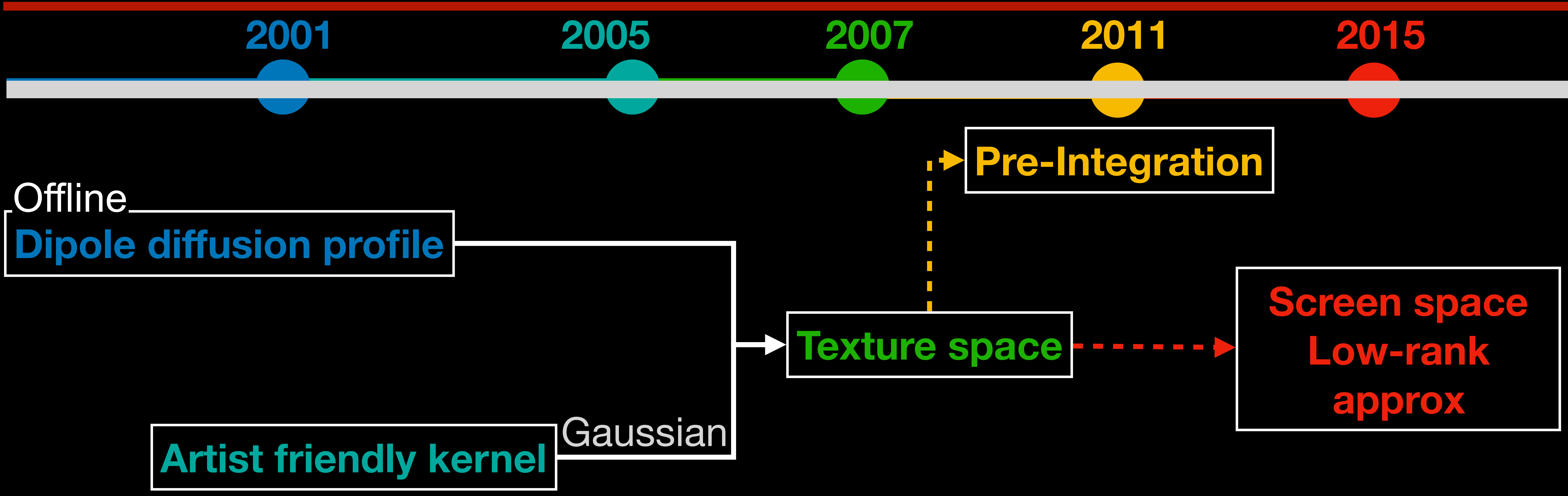


Pre-integrated BRDF



Pre-integrated Penumbrae

Real-time Subsurface Scattering - Literature Review



2001

2005

2007

2011

2015

Pre-Integration

Offline

Dipole diffusion profile

Texture space

Screen space Low-rank approx

Artist friendly kernel

Gaussian

Separable Subsurface Scattering
[Jimenez et al. 2015]
Two 1D instead of one 2D convolution

Offline

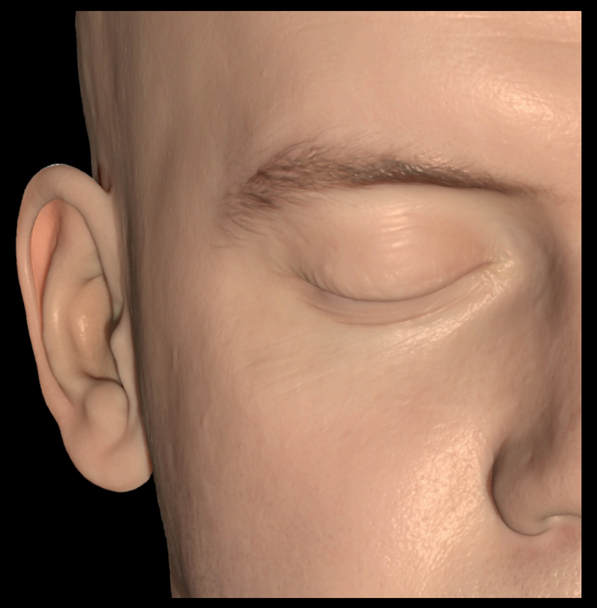
Burley's normalized diffusion profile

Inspired by Schlick's Fresnel Approx.

$$R(\theta) = R_0 + (1 - R_0)(1 - \cos\theta)^5$$

Fit all scattering as a whole

$$R(r) = A\frac{e^{-r/d} + e^{-r/(3d)}}{8\pi d r}$$



[Christensen and Burley 2015]

Real-time Subsurface Scattering - Literature Review

2001

2005

2007

2011

2015

2018

Pre-Integration

Offline

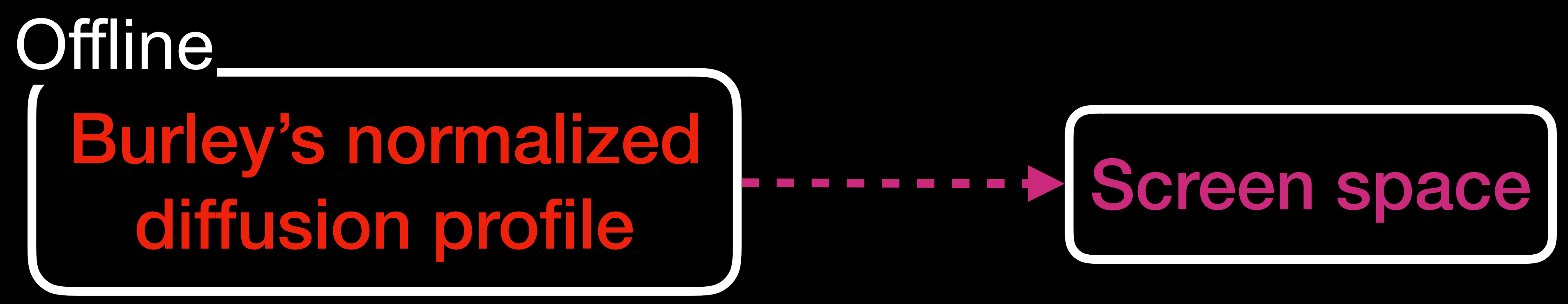
Dipole diffusion profile

Texture space

Screen space
Low-rank
approx

Gaussian

Artist friendly kernel



Offline

Burley's normalized
diffusion profile

Screen space

Real-time subsurface
scattering in Unity
[Golubev 2018]

Motivation

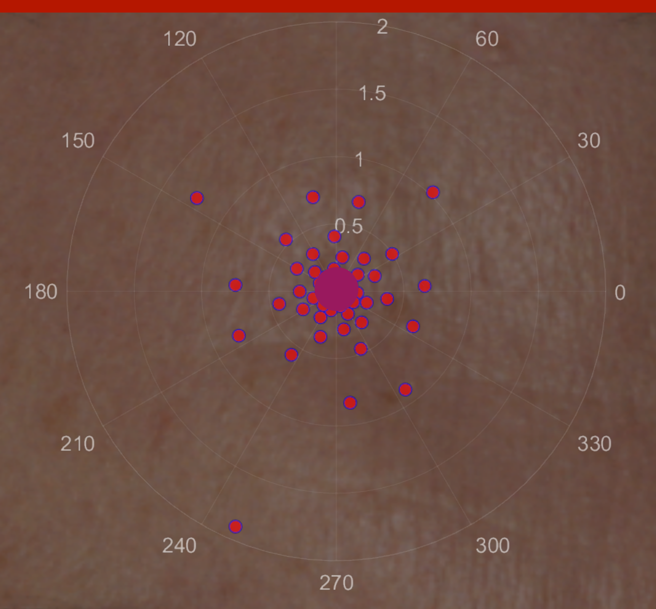
- Photorealistic subsurface scattering without uncanny valley.
  - Monte Carlo estimation
    - Random memory access
    - Cache incoherence
    - High bandwidth demand
- Question: What novel techniques can be used to target on high quality photo-realistic real-time rendering with the contemporary and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

120
2
60
1.5
150
30
1
0.5
180
0
210
330
240
300
270

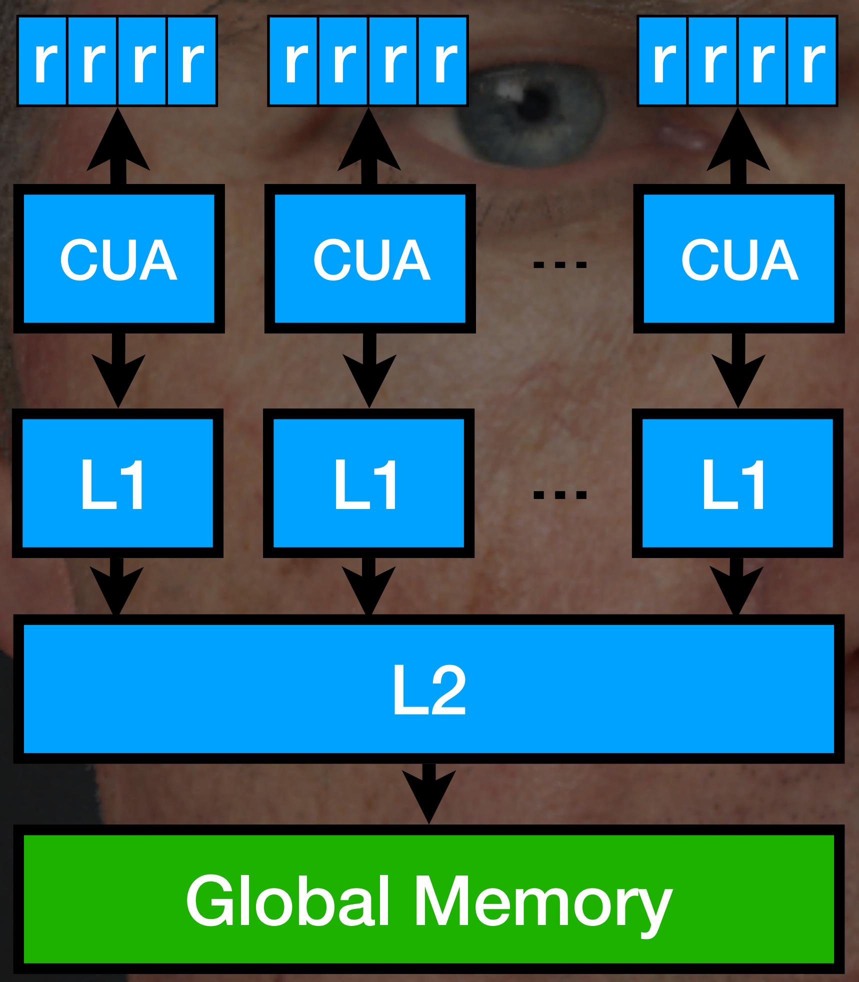
Digital Mike

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?



Simplified data cache architecture for Turing and Navi



Turing V4	RX 5700 XT
–	8.88 *TiB/s*
4.07 *TiB/s*	3.55 *TiB/s*
1.18 *TiB/s*	1.77 *TiB/s*
298 *GiB/s*	417 *GiB/s*

Taxonomy

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

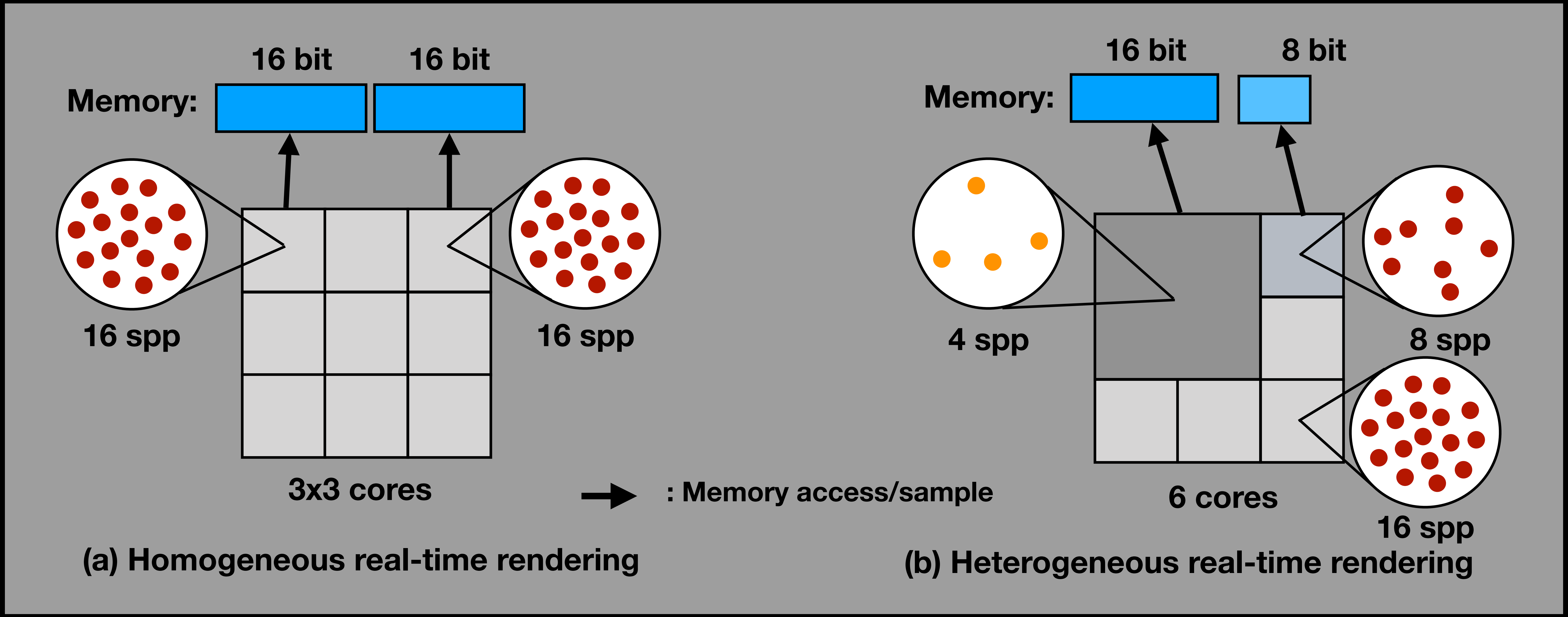
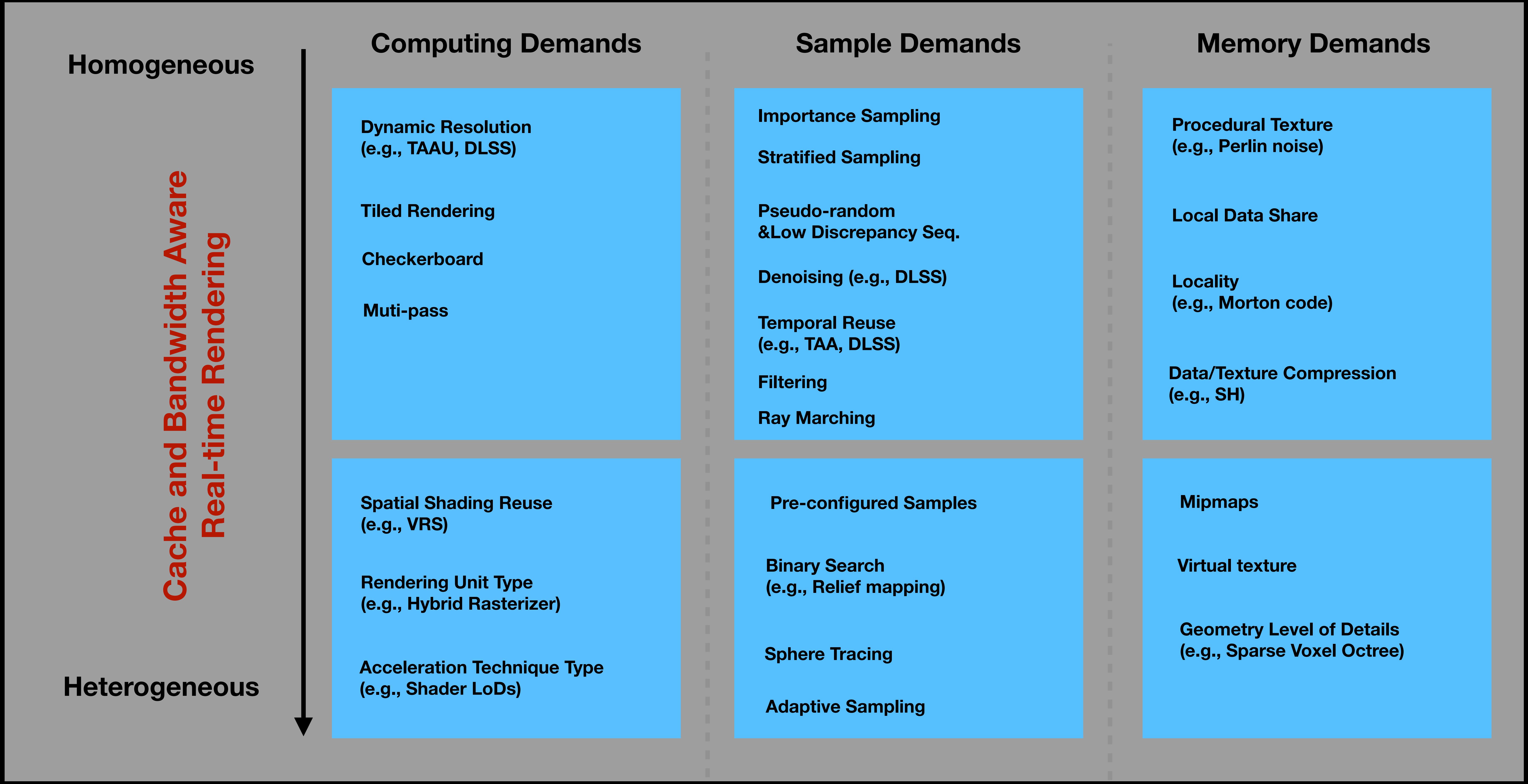


Figure 1: Comparison between (a) homogeneous and (b) heterogeneous real-time rendering. Heterogeneous real-time rendering can dynamically reduce the sampling, shading units, and memory demands without noticeable quality degradation.

Taxonomy

- Cache and bandwidth aware real-time rendering



	Computing Demands	Sample Demands	Memory Demands
Homogeneous	Dynamic Resolution (e.g., TAAU, DLSS) Tiled Rendering Checkerboard Muti-pass	Importance Sampling Stratified Sampling Pseudo-random &Low Discrepancy Seq. Denoising (e.g., DLSS) Temporal Reuse (e.g., TAA, DLSS) Filtering Ray Marching	Procedural Texture (e.g., Perlin noise) Local Data Share Locality (e.g., Morton code) Data/Texture Compression (e.g., SH)
Heterogeneous	Spatial Shading Reuse (e.g., VRS) Rendering Unit Type (e.g., Hybrid Rasterizer) Acceleration Technique Type (e.g., Shader LoDs)	Pre-configured Samples Binary Search (e.g., Relief mapping) Sphere Tracing Adaptive Sampling	Mipmaps Virtual texture Geometry Level of Details (e.g., Sparse Voxel Octree)

Taxonomy

- Bandwidth demands cost function

1. Computing demands

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

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

2. Sample demands

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

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

3. Memory demands

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

Subsurface Scattering

Ch. 5 Acceleration Technique Type
- **AFIS**
- **Separable**

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

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

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

$$f(\cdot)$$

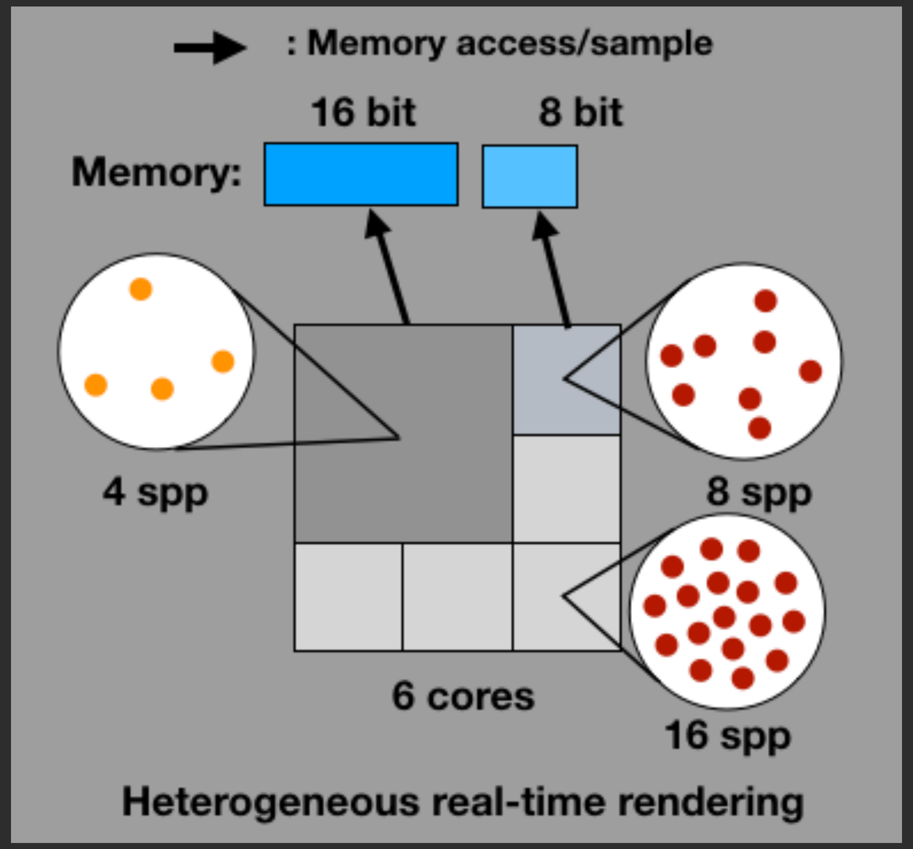
Ch.5 Adaptive Filtered Importance Sampling (+ memory demands)

$$R(\cdot)$$

Outline

Section I: Chapter 1 ~ Chapter 3

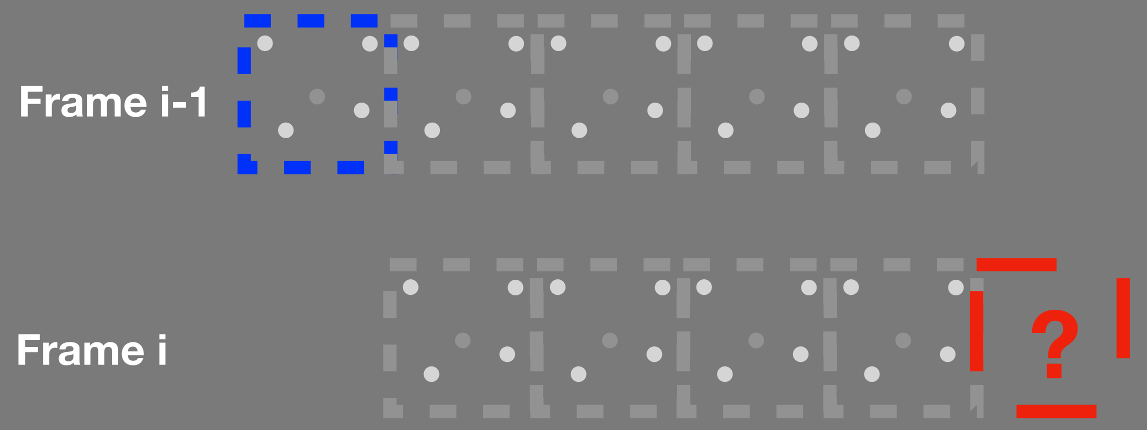
- Introduction
- Literature
- Motivation
- Heterogeneous Real-time Rendering



Heterogeneous real-time rendering

Section II: Chapter 4 (I3D'20)

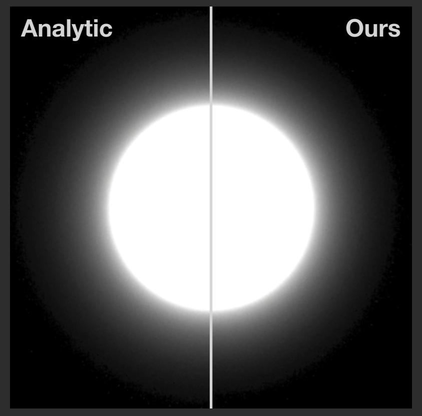
- Real-time Adaptive Sampling $O(1)$



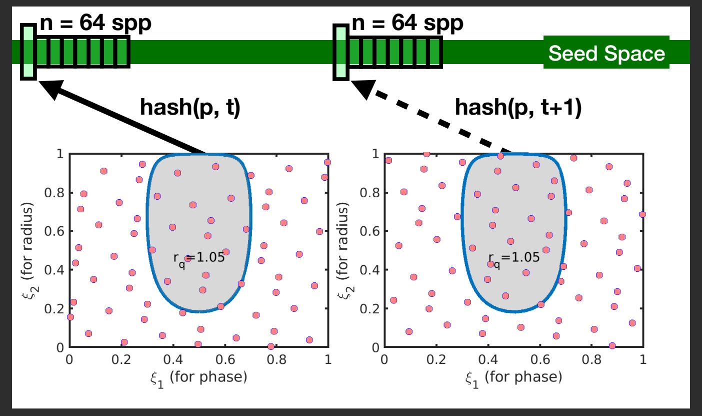
Section III: Chapter 5

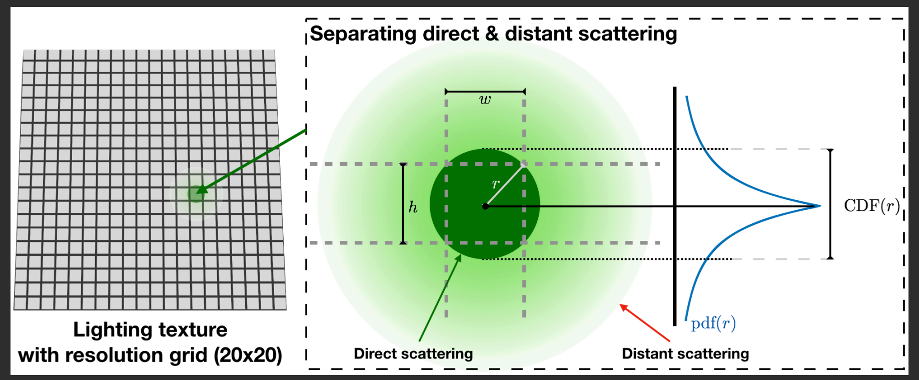
- Subsurface Scattering

Sampling function approximation

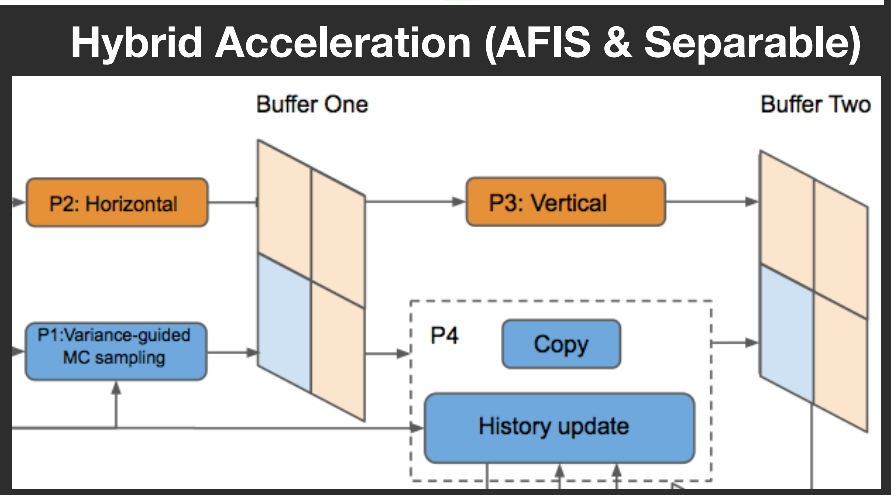


Sampling sequence




Separating direct & distant scattering
Lighting texture with resolution grid (20x20)
w
h
r
CDF(r)
pdf(r)
Direct scattering
Distant scattering

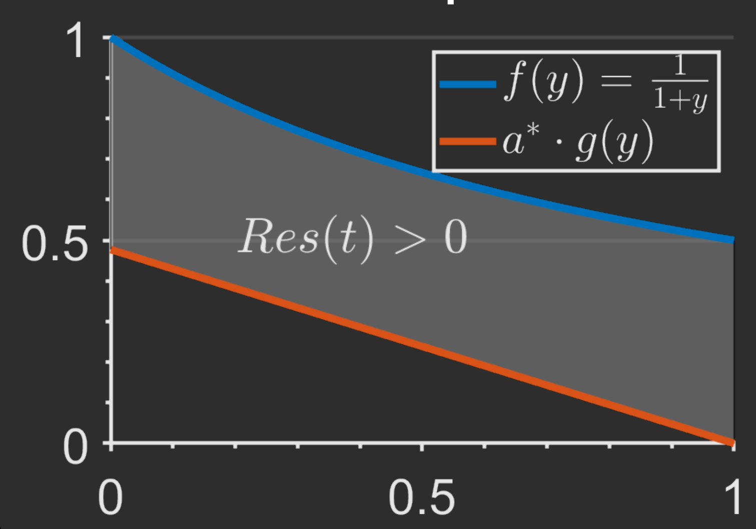
Hybrid Acceleration (AFIS & Separable)



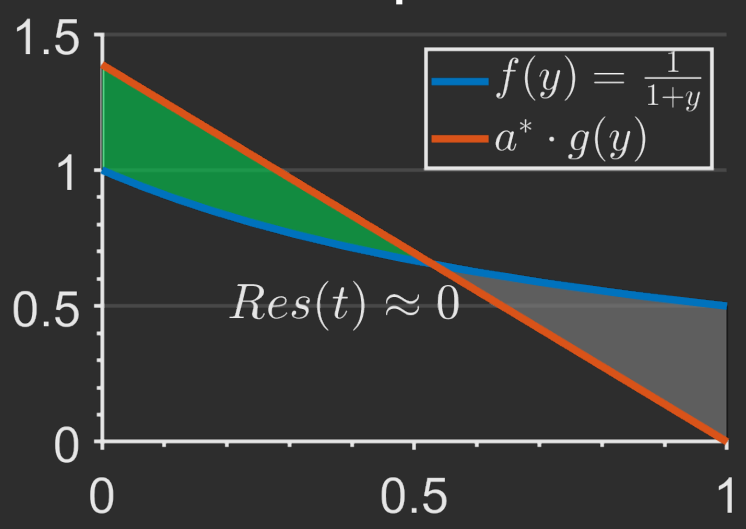
Section IV: Chapter 6 (I3D'21)

- Real-time Control Variates with one to even zero additional texture

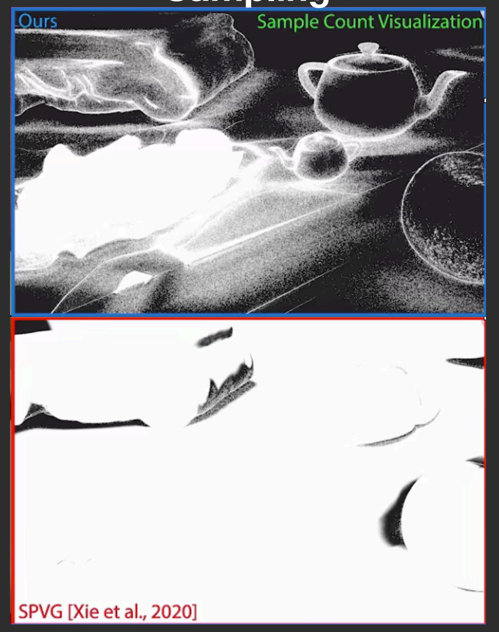
Without temporal info



With temporal info

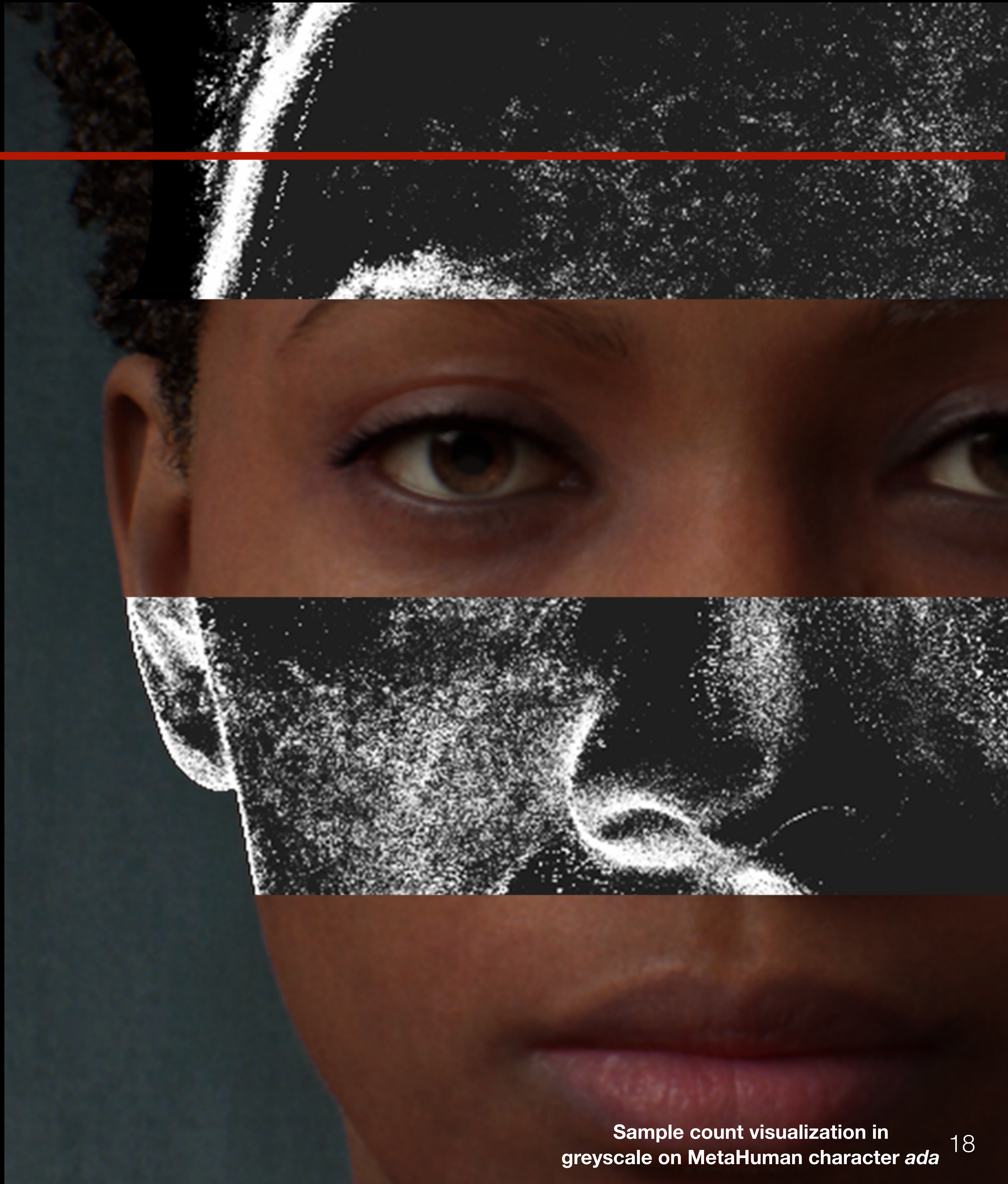


Temporally Stable Adaptive Sampling



Real-time Adaptive Sampling

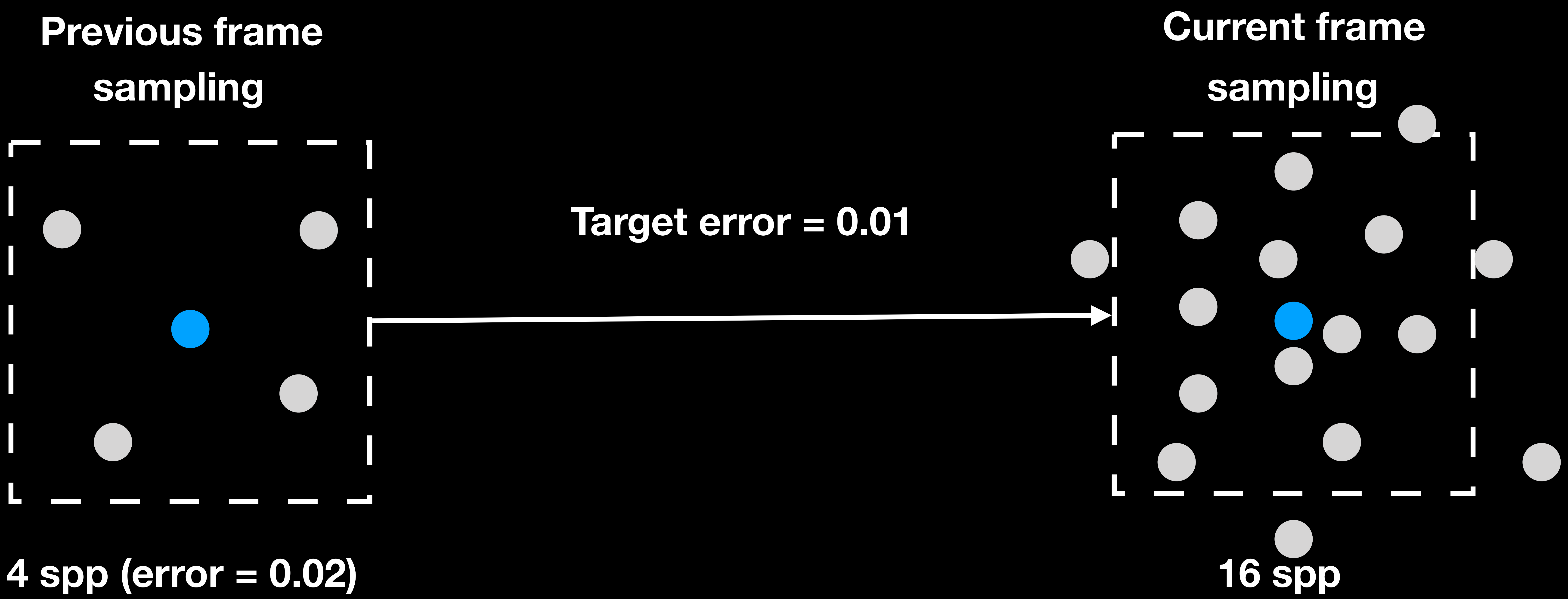
- Adaptively allocate samples to high variation regions.
- Subsurface scattering
  - High variation in lighting gradient change region
- Adaptive sampling procedural
  - Pass 1: Pilots (discarded)
  - Pass 2: Additional samples
- Related work
  - A priori (e.g., frequency or derivative)
  - A Posteriori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



Sample count visualization in greyscale on MetaHuman character *ada*

Basic Metrics

Introduction to basic adaptive sampling



$$n_{(i)} = \sigma^2_{M_{(i-1)}} \cdot n_{(i-1)}/\sigma^2_0$$

Insufficient observation (unstable)

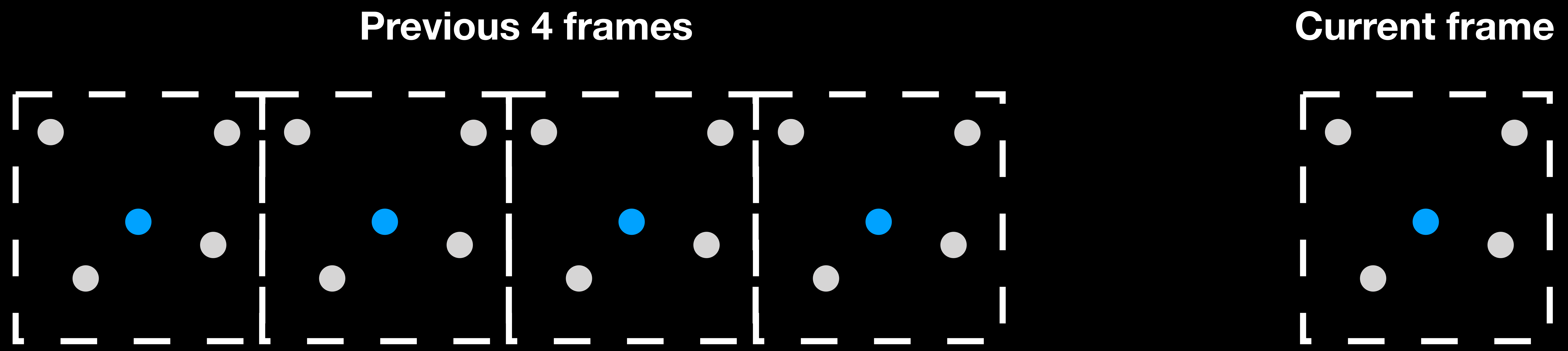
Temporal accumulation

$$\sum_{i-k+1}^{i} n_{(j)} = \frac{\sigma^2_{M_{(i-1)}} \sum_{i-k+1}^{i} n_{(j-1)}}{\sigma^2_0} = \bar{n}_{(i)} \cdot k$$

k frame formulation

Metrics with Temporal Accumulation

Adaptive sampling with sample history

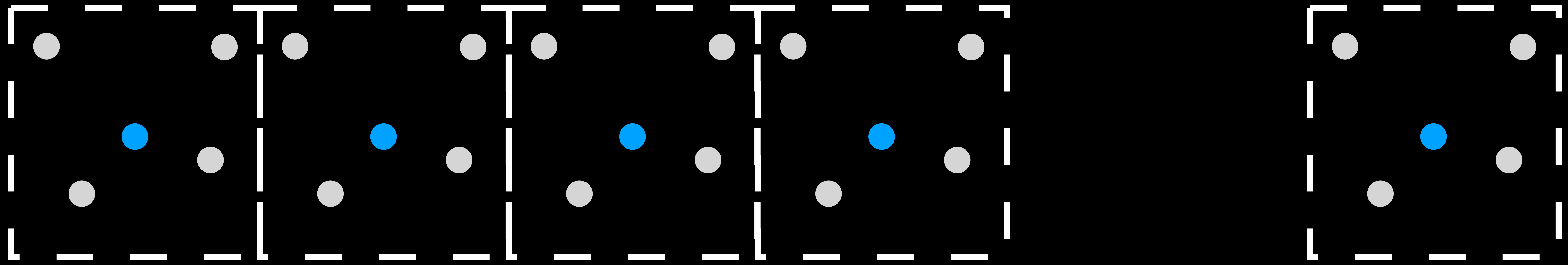


Metrics with Temporal Accumulation

Adaptive sampling with sample history

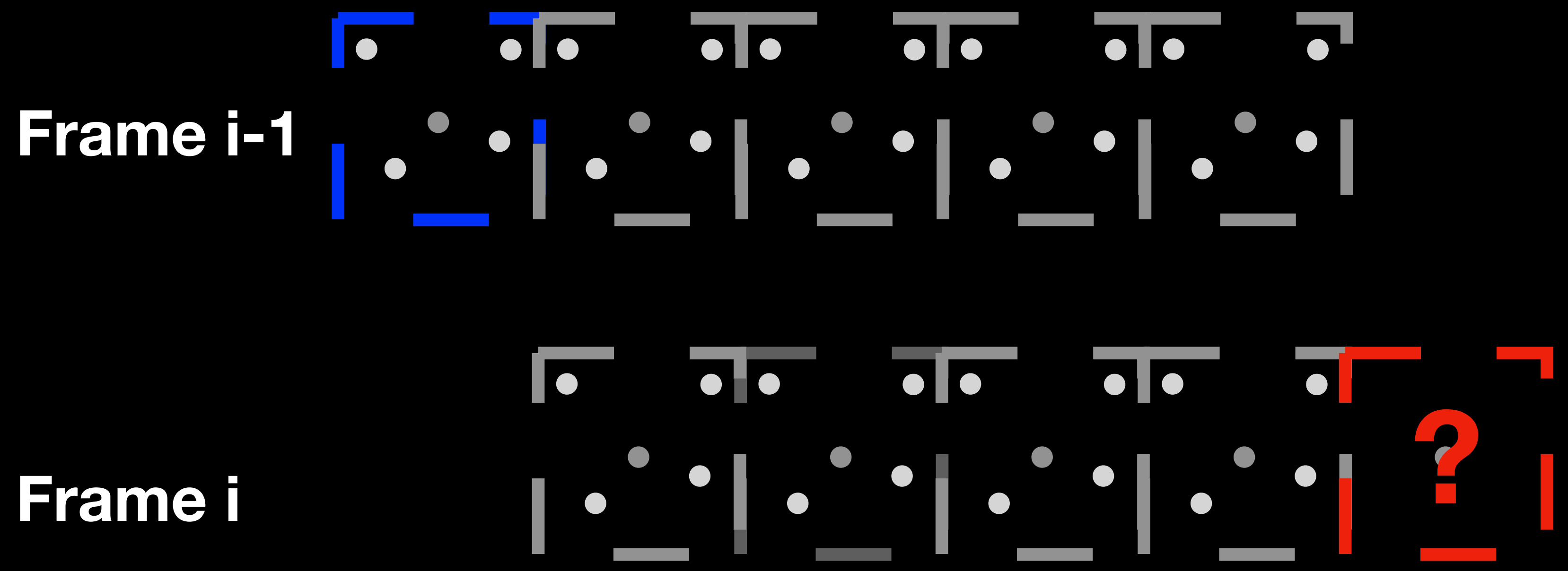
Previous 4 frames

Current frame

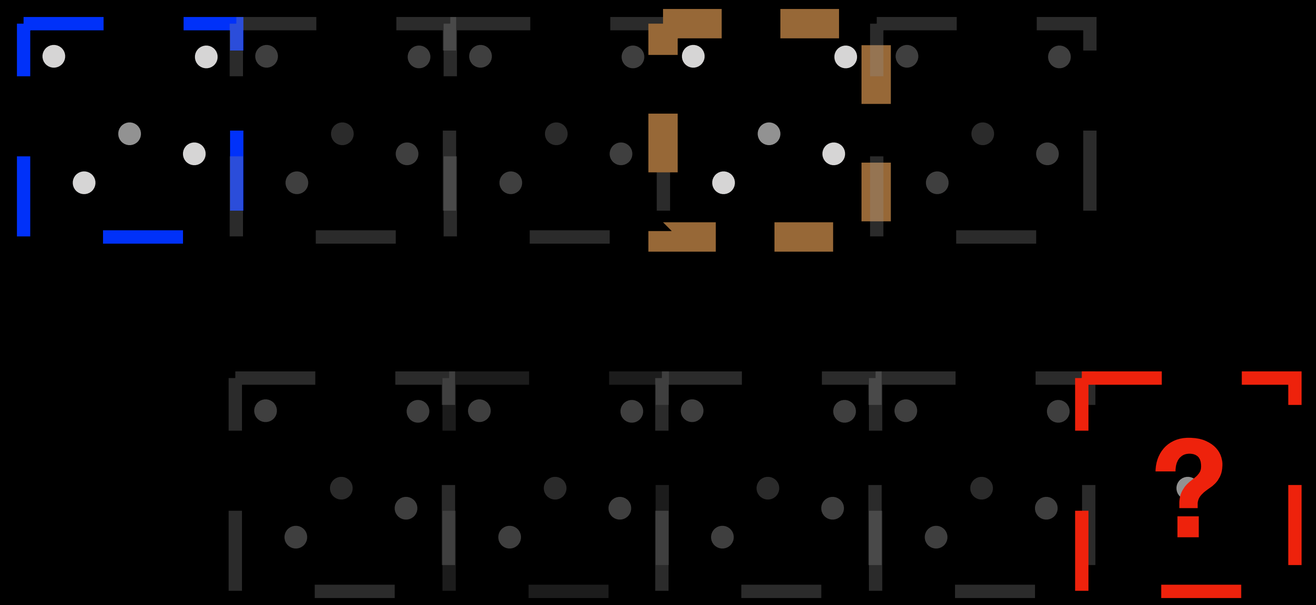


Metrics with Temporal Accumulation

- Use difference of temporal accumulation



$$\sum_{i-k+1}^{i} n_{(j)} = \frac{\sigma^2_{M_{(i-1)}} \sum_{i-k+1}^{i} n_{(j-1)}}{\sigma_0^2} = \bar{n}_{(i)} \cdot k$$

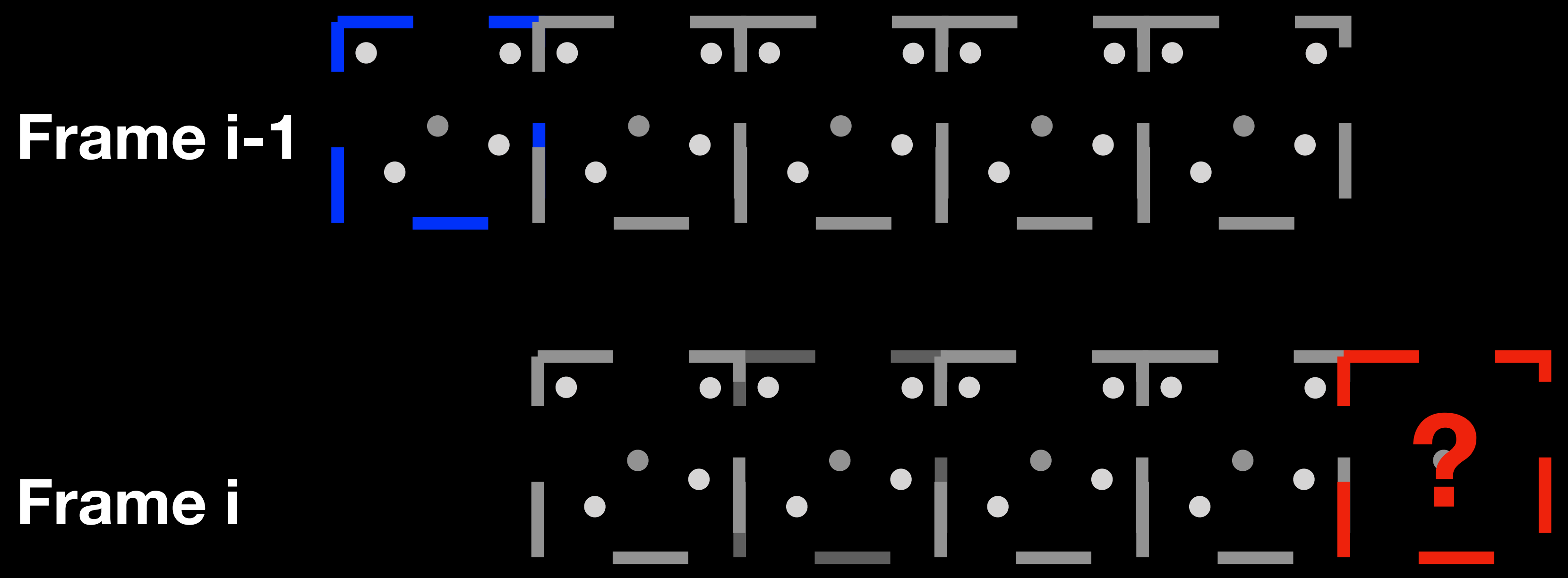


$$\hat{n}_{(i)} = \sum_{i-k+1}^{i} n_{(j)} - \sum_{i-k+1}^{i} n_{(j-1)} + n_{(i-k)}$$

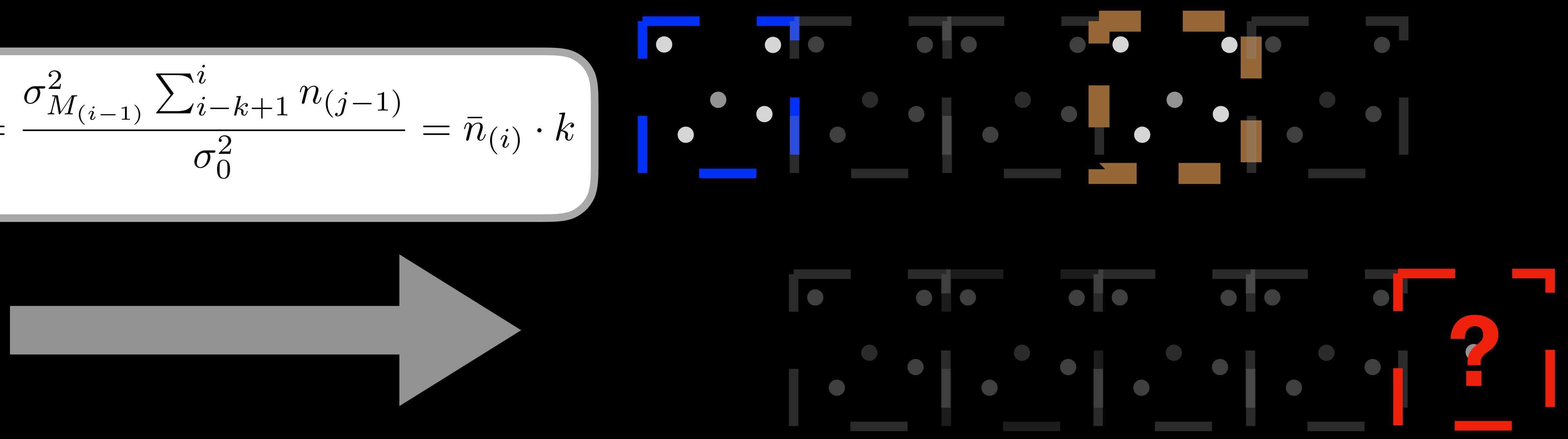
$$\hat{n}_{(i)} = \frac{(\sigma^2_{M_{(i-1)}} - \sigma_0^2)}{\sigma_0^2} \cdot \bar{n}_{(i-1)} \cdot k + n_{(i-k)}$$

Metrics with Temporal Accumulation

- Use difference of temporal accumulation



$$\sum_{i-k+1}^{i} n_{(j)} = \frac{\sigma^2_{M_{(i-1)}} \sum_{i-k+1}^{i} n_{(j-1)}}{\sigma_0^2} = \bar{n}_{(i)} \cdot k$$



$$\hat{n}_{(i)} = \sum_{i-k+1}^{i} n_{(j)} - \sum_{i-k+1}^{i} n_{(j-1)} + n_{(i-k)}$$

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

In stable regions with $\sigma^2_{M_{(i-1)}} = \sigma_0^2$

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

We have an approximation as:

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

Algorithm complexity: O(1)

Metrics with Temporal Accumulation

- Use temporal accumulation to reduce cached sample counts

$$\hat{n}_{(i)} \approx \frac{(\sigma^2_{M_{(i-1)}} - \sigma^2_0)}{\sigma^2_0} \cdot \bar{n}_{(i-1)} \cdot k + \bar{n}_{(i-1)}$$

History buffer O(1):

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

- Exponential moving average (EMA):

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

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

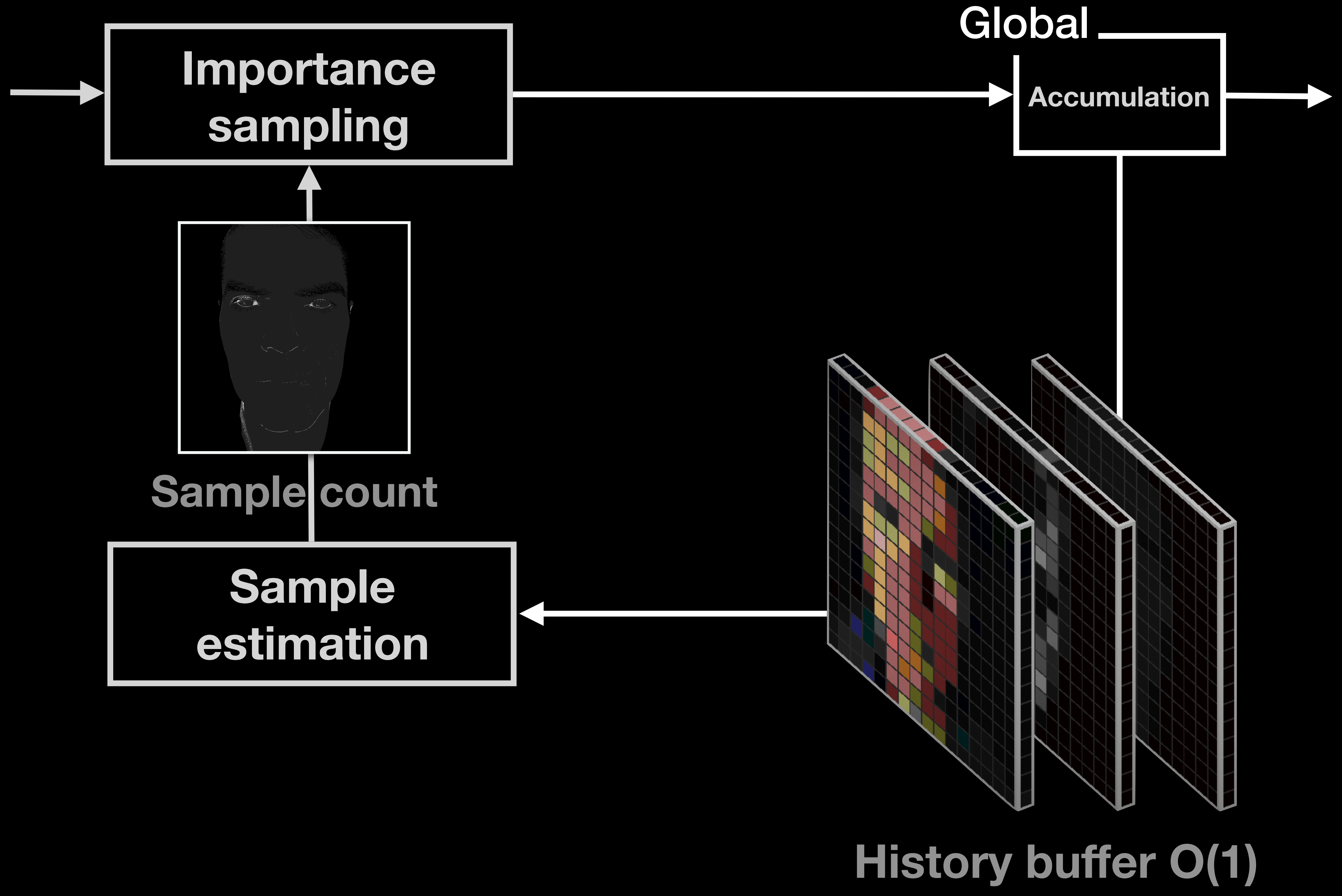
$$\sigma^2_{M_{(i)}} = (1-\alpha)\sigma^2_{M_{(i-1)}} + \alpha(1-\alpha)(\mathcal{S}(p_i) - \mathcal{C}(x_i, \Lambda))^2$$

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

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

Local Guiding

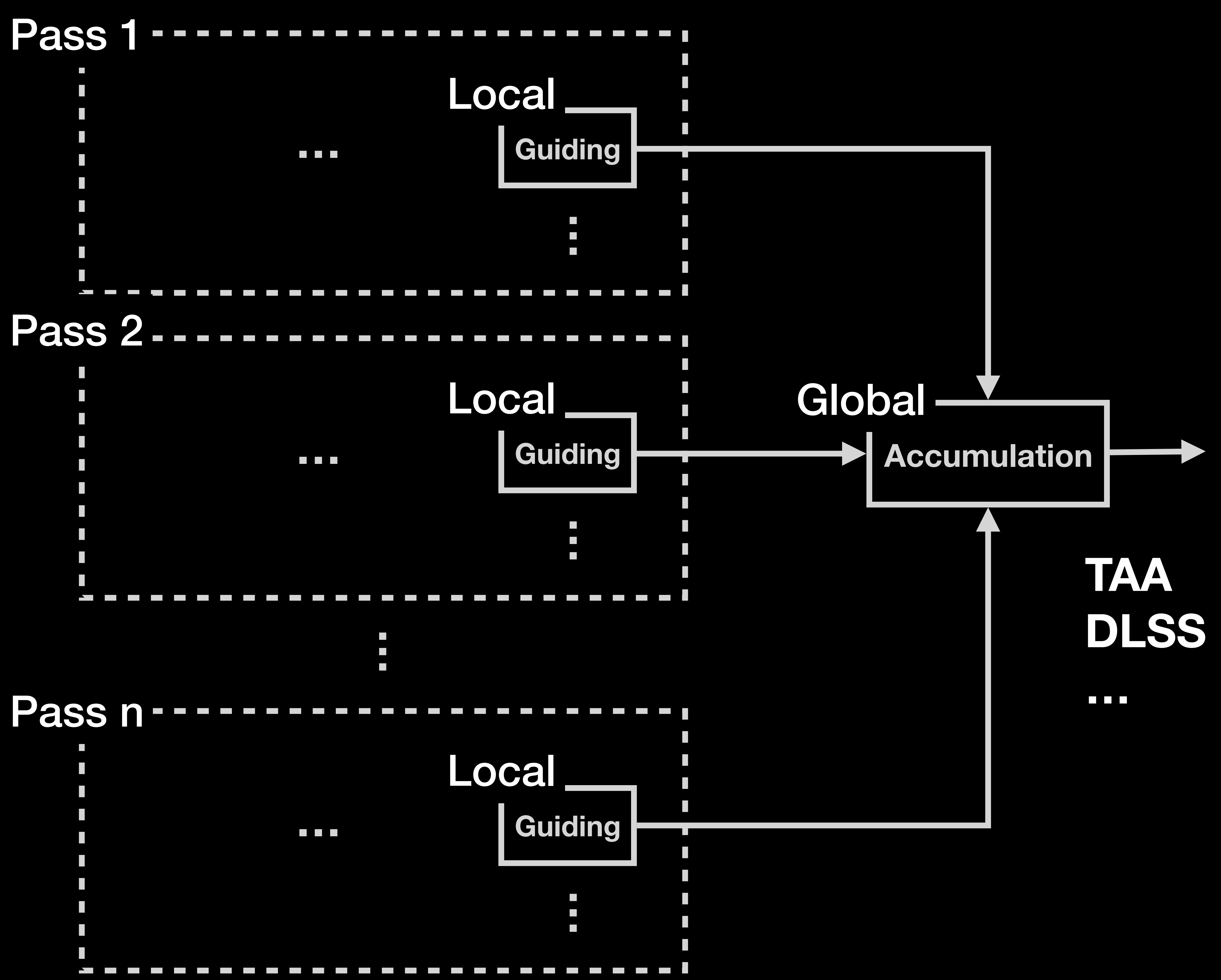
Current single pass adaptive sampling architecture:



- 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.

Local Guiding

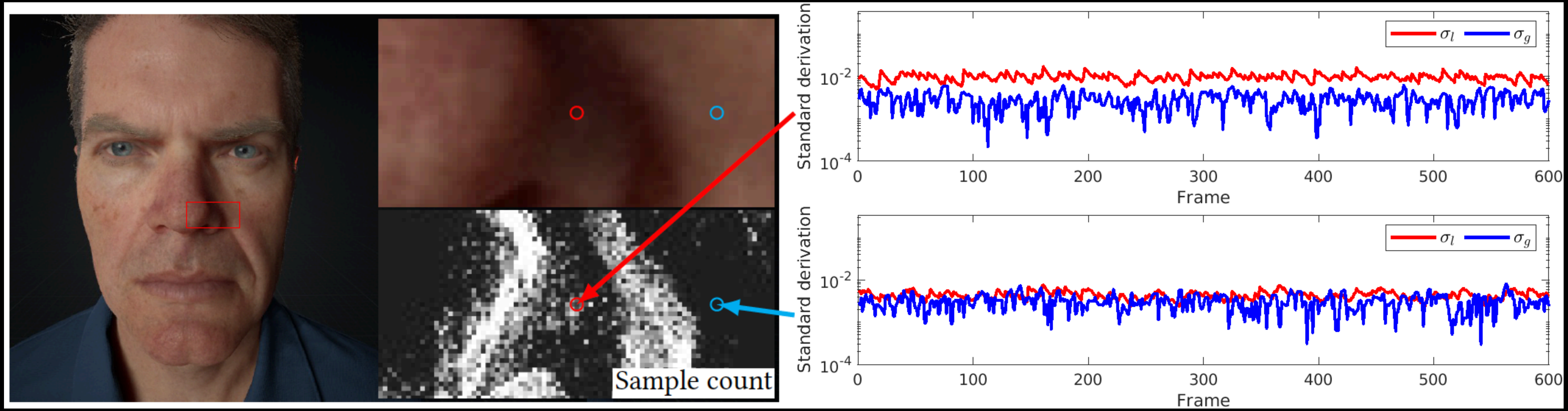
Local pass guided adaptive sampling



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

Local Guiding with Global TAA

Shadow region global quality: 11x



Flat region global quality: 2x

Operator set: target a lower bound on local pass quality

	Local Guiding	Global TAA [Karis 2014]
Resample & Reject	Nearest neighbor sampler without history rejection	Velocity & Box clamping
Weight update	Max weight used by global TAA	Velocity & Box clamping
Neighbor function	Same global jitter with box filter	Jitter with gaussian filter

Local Guiding with Deep Learning Super Sampling

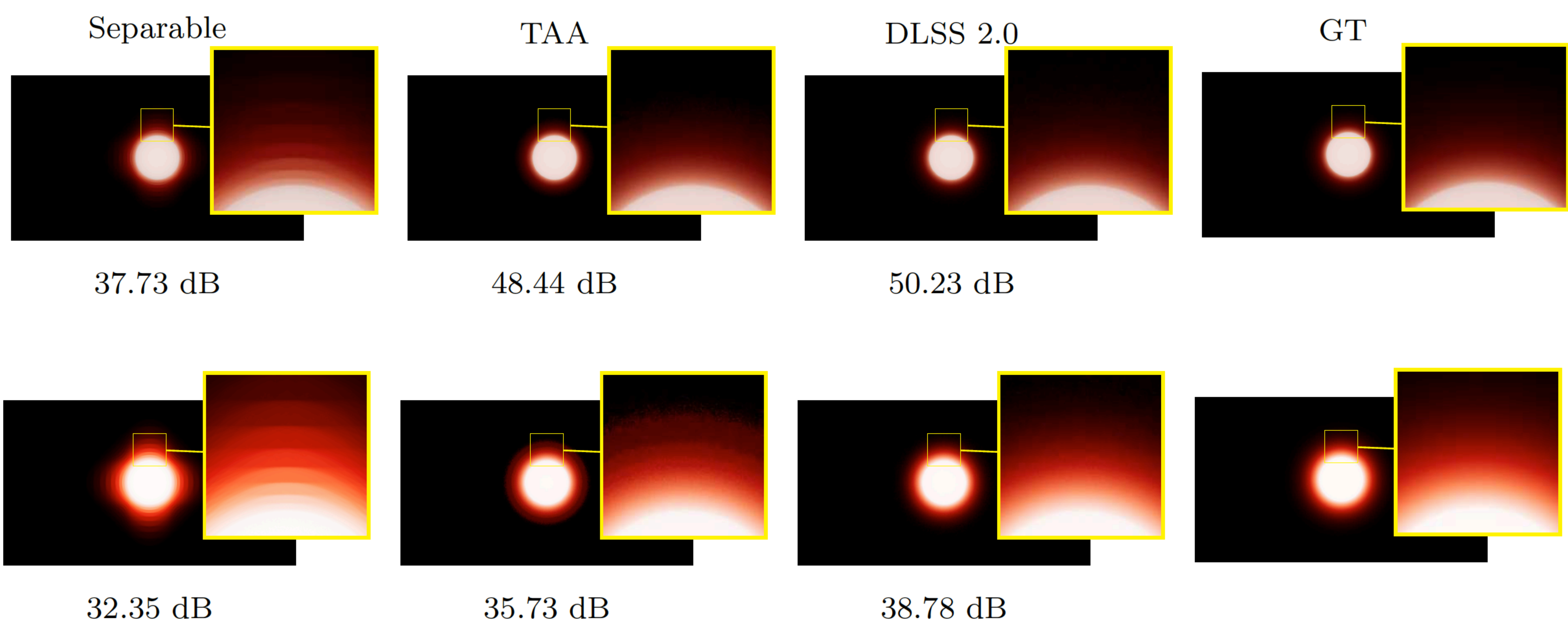
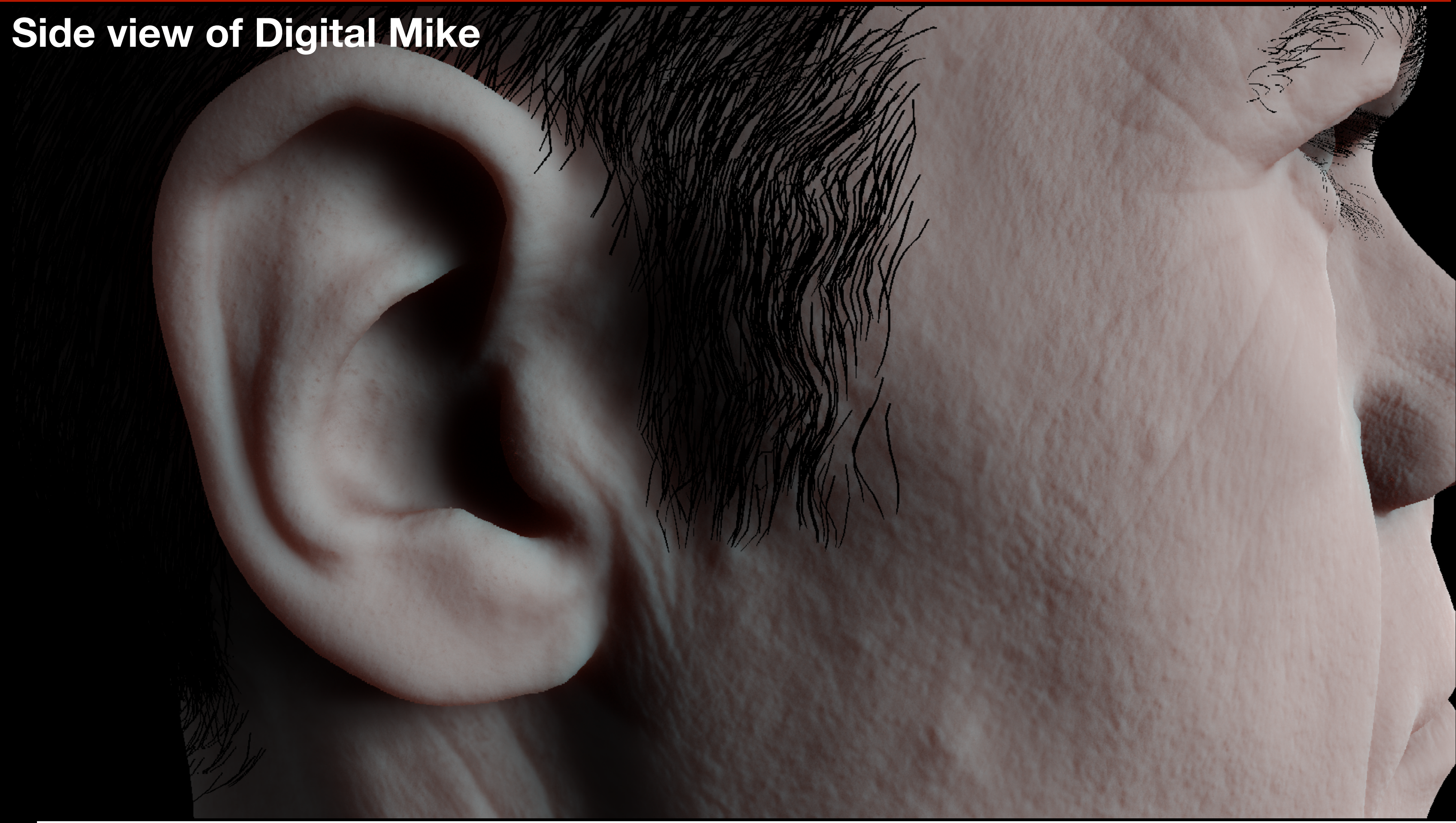


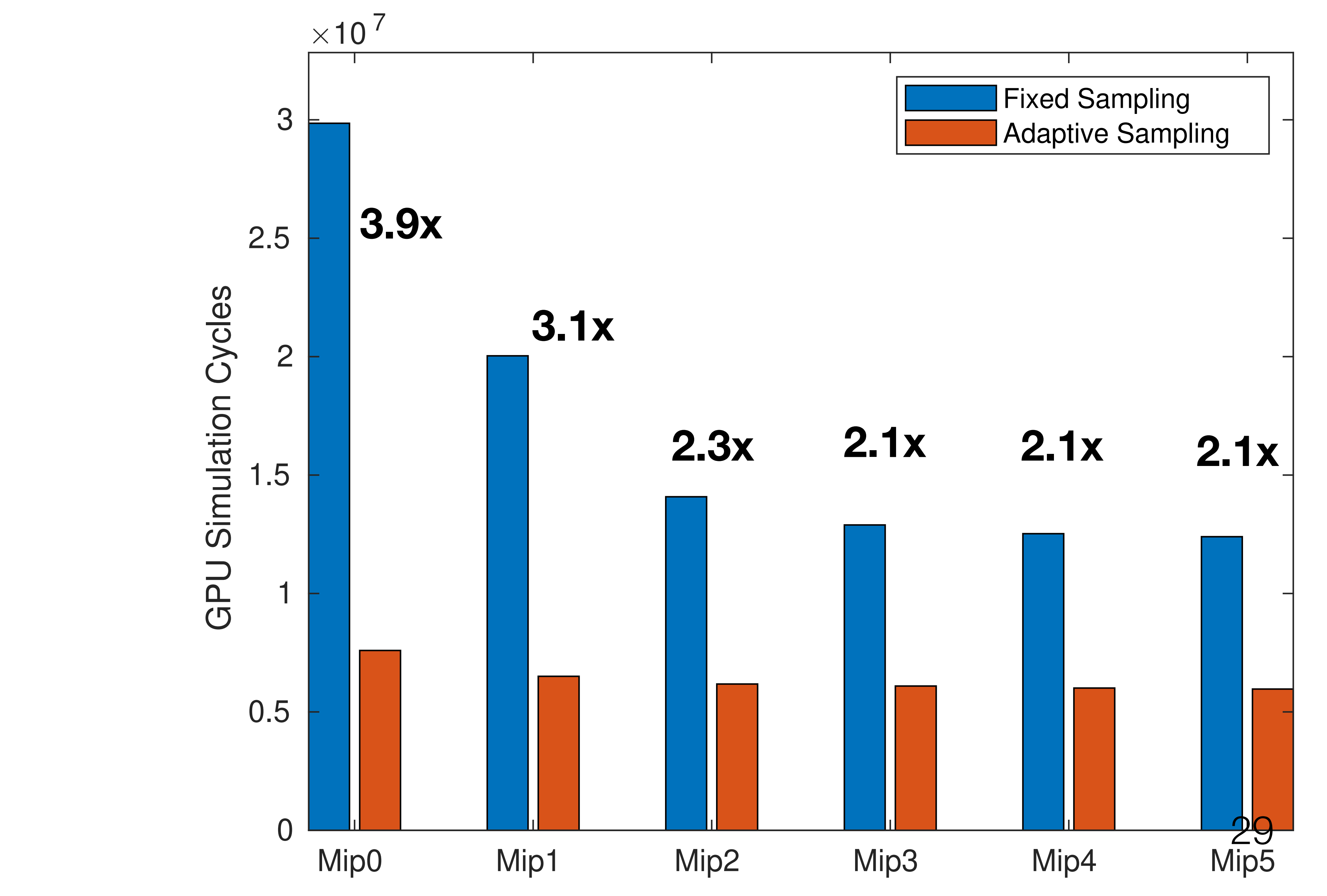
Figure 2: Quality of different global temporal accumulation methods (TAA and DLSS 2.0) under two different intensities. The quality of Separable is also presented. The PSNR is tested against the ground truth with 1024 spp after converting to luminance.

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

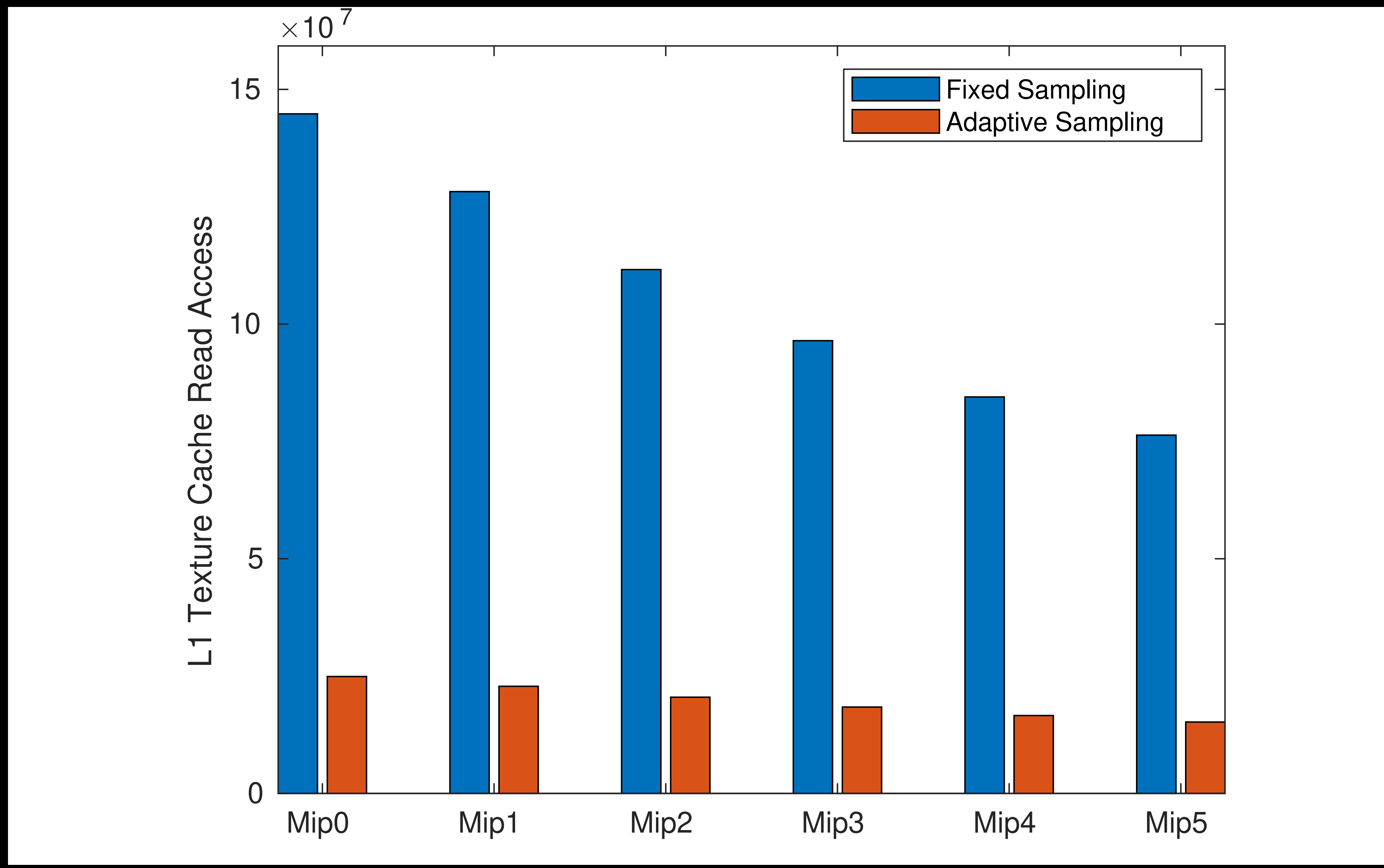
Side view of Digital Mike

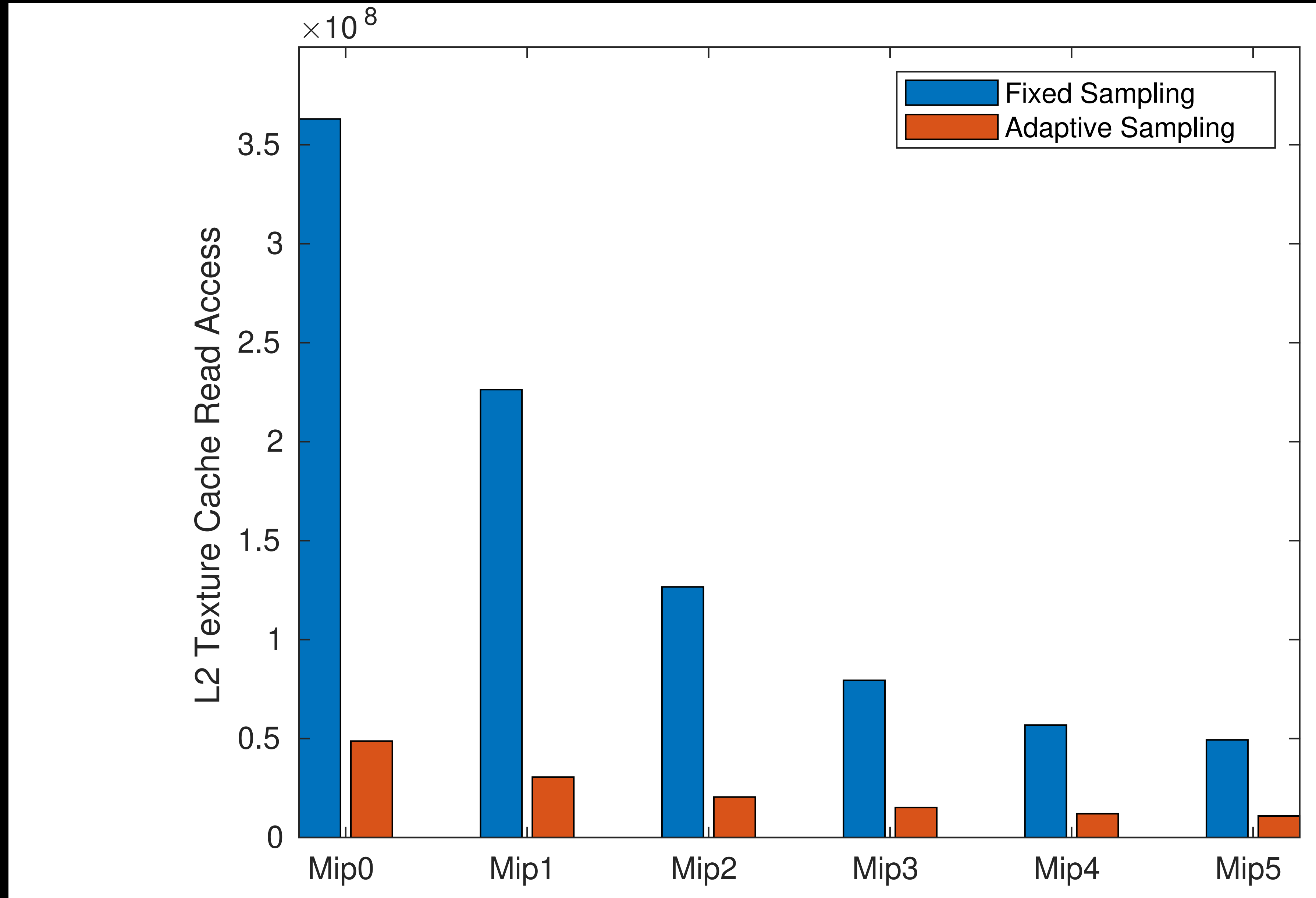




Cycle Level Analysis of Bandwidth Demands

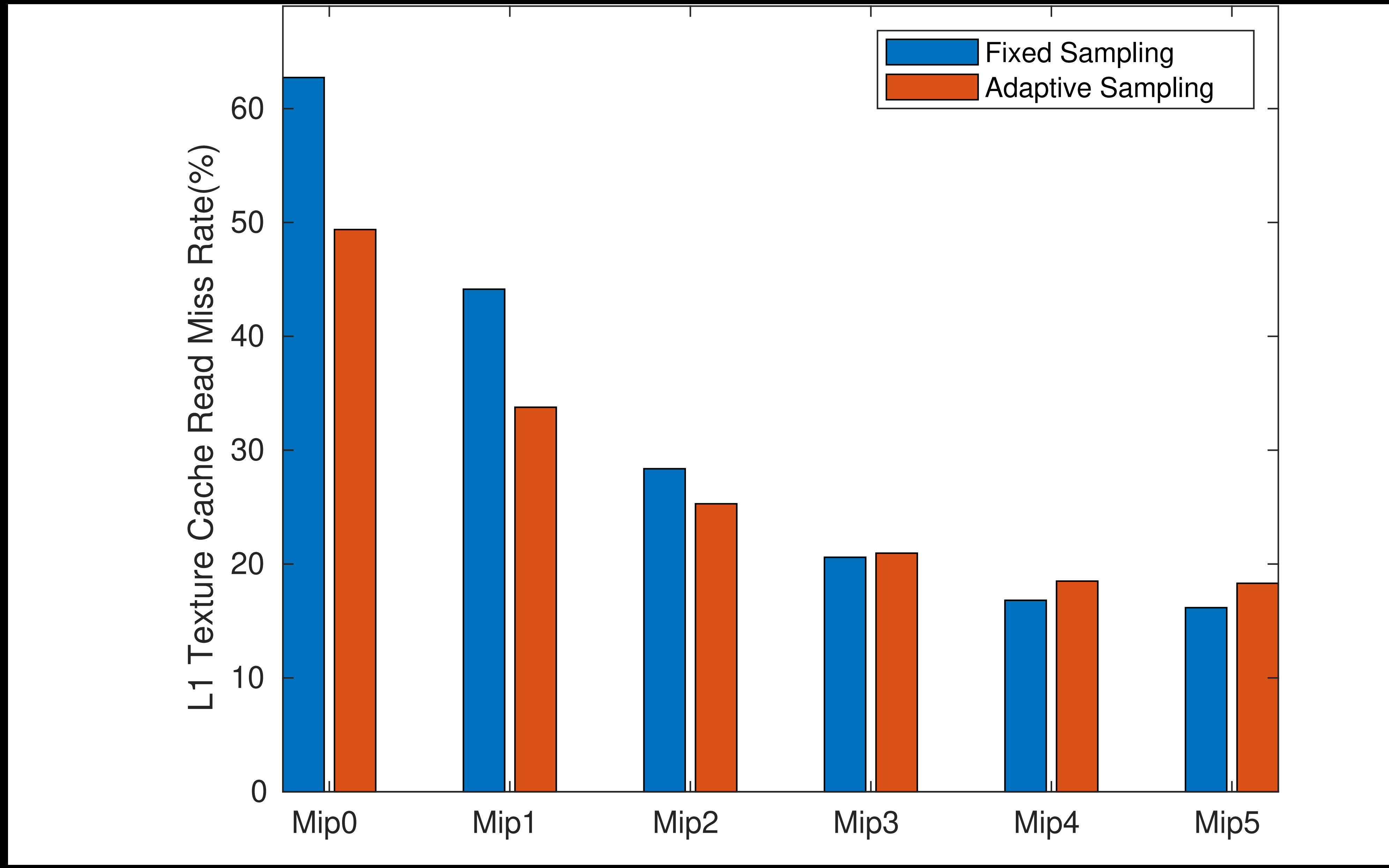
- GPGPU-Sim [Bakhoda et al., 2009, Khairy et al., 2020]
  - Cycle-level analysis of parallel computing
  - CUDA
  - Hardware config:
    - NVIDIA TITAN X Pascal (compute shader)
- Adaptive vs. Fixed sampling
  - Simulation cycles
  - L1 & L2 cache demands

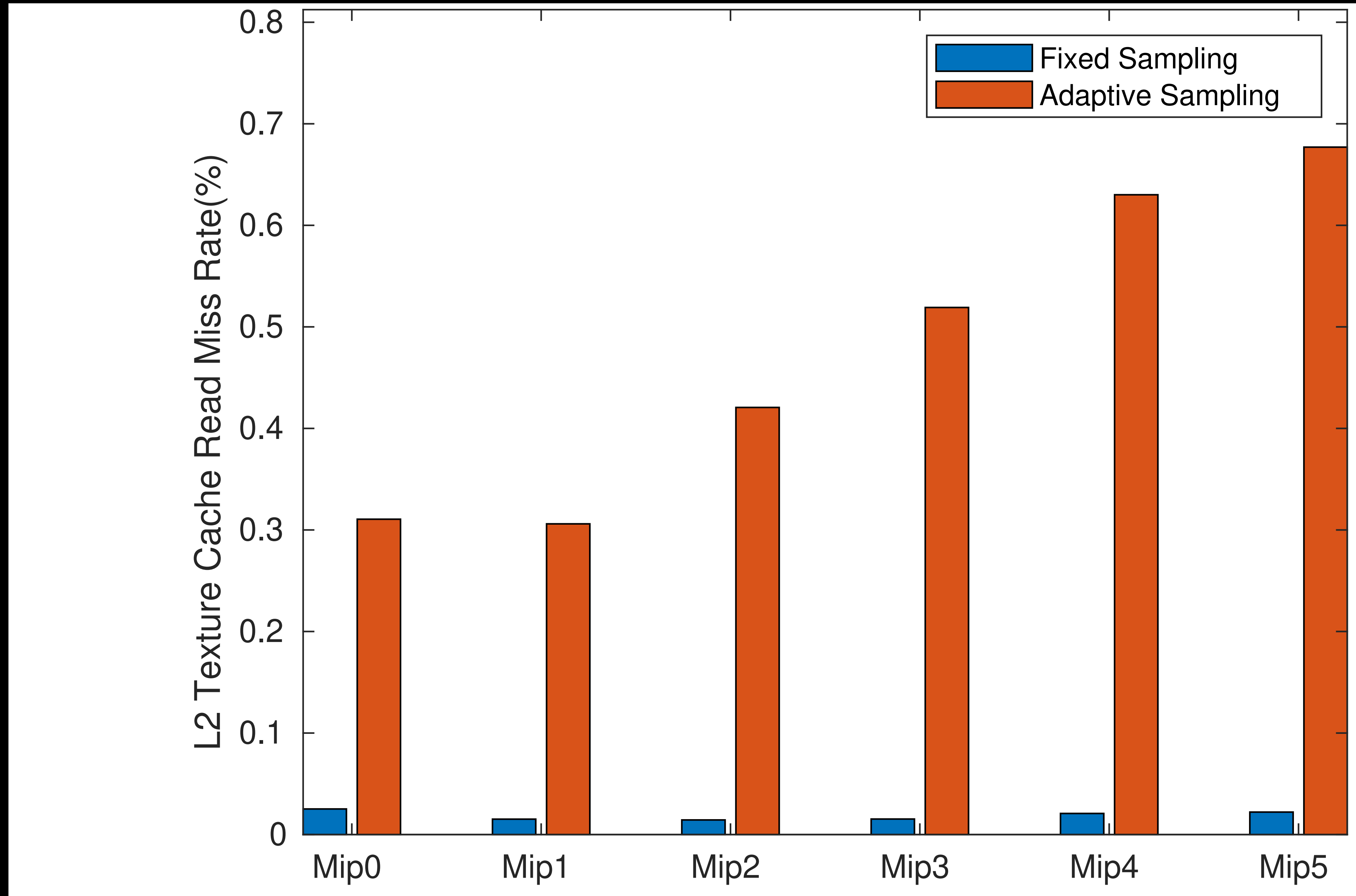




Cycle Level Analysis of Bandwidth Demands

- 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



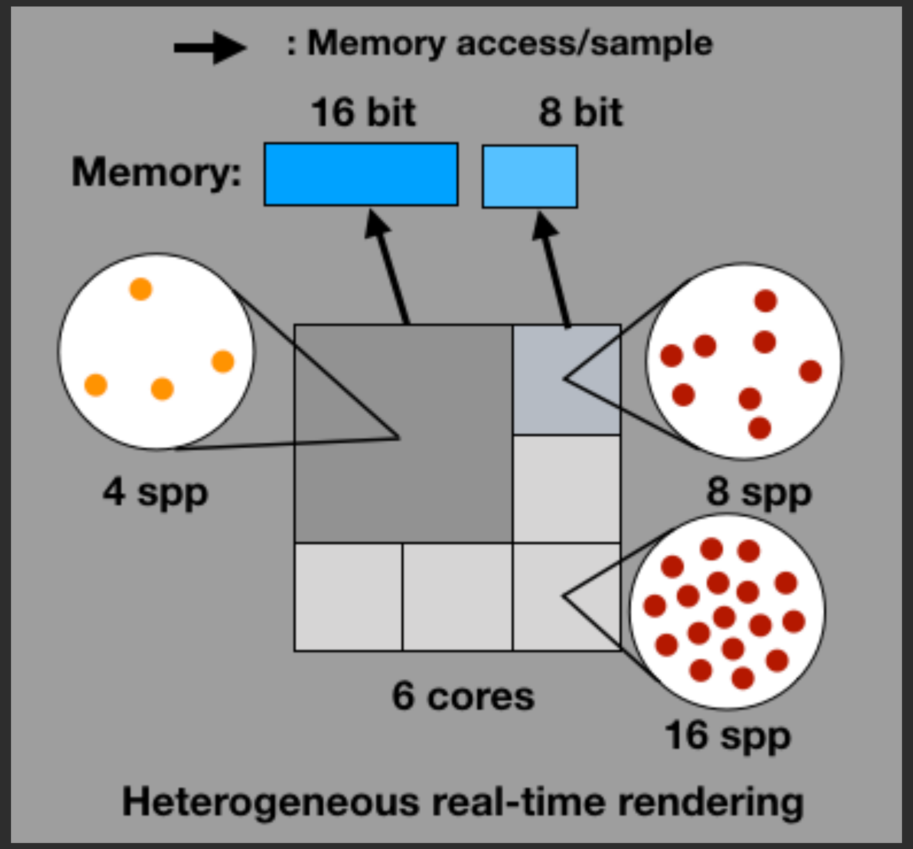


Mip0: 91,541 vs 151,356

Outline

Section I: Chapter 1 ~ Chapter 3

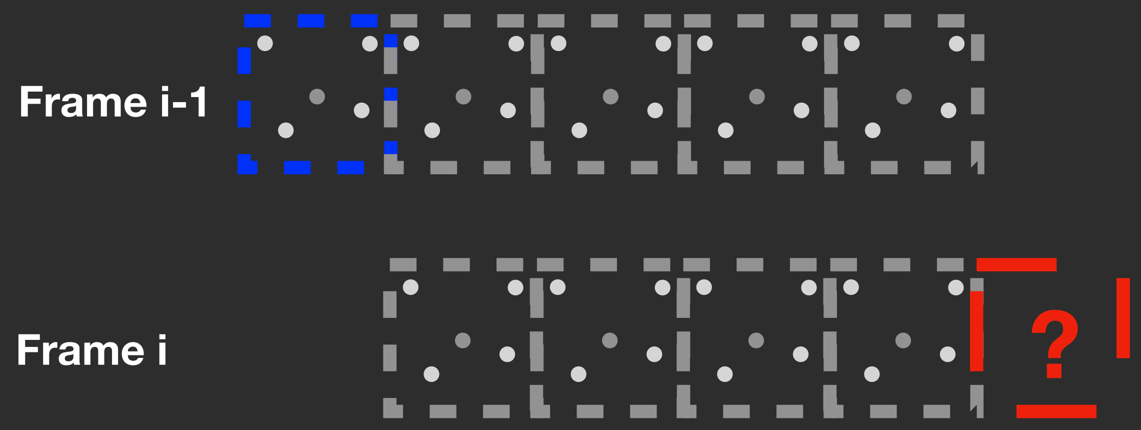
- Introduction
- Literature
- Motivation
- Heterogeneous Real-time Rendering



Heterogeneous real-time rendering

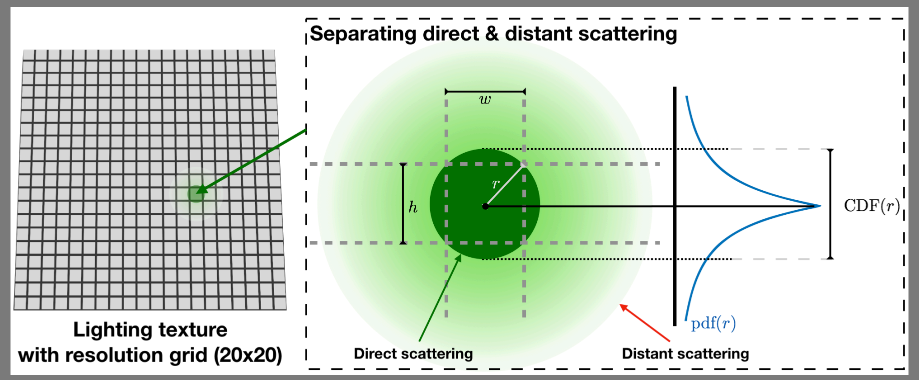
Section II: Chapter 4 (I3D’20)

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

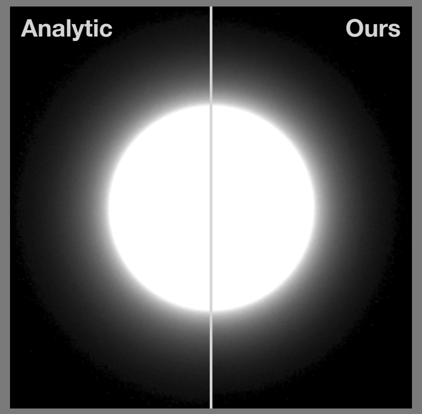


Section III: Chapter 5

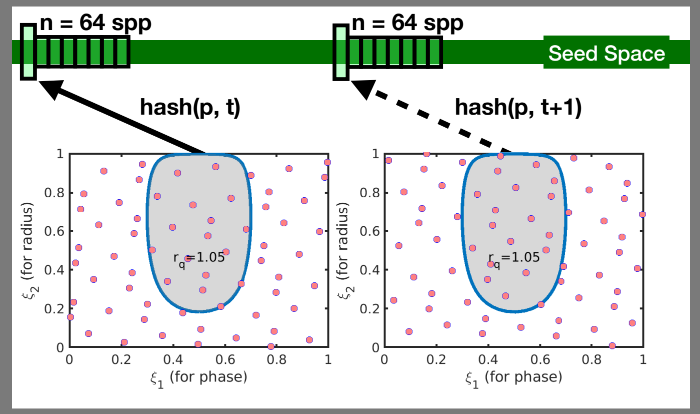
- Subsurface Scattering



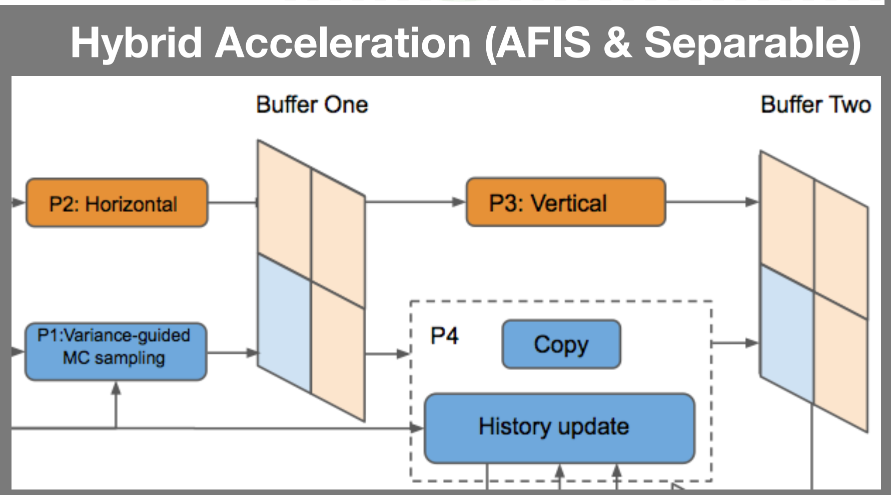
Sampling function approximation



Sampling sequence



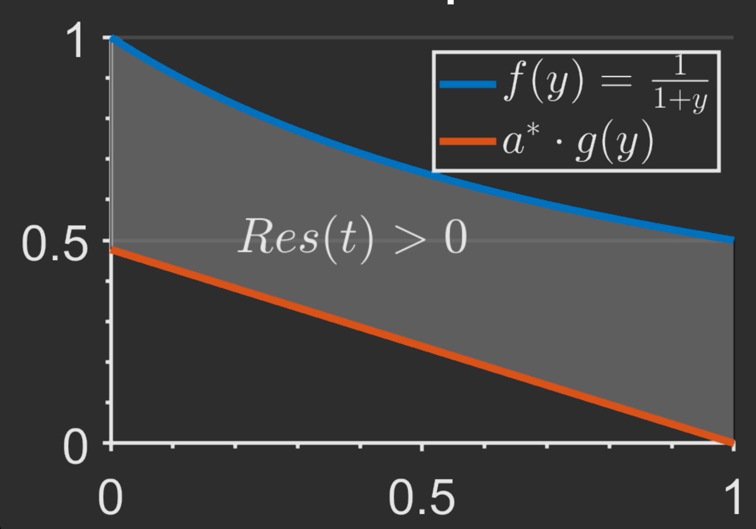
Hybrid Acceleration (AFIS & Separable)



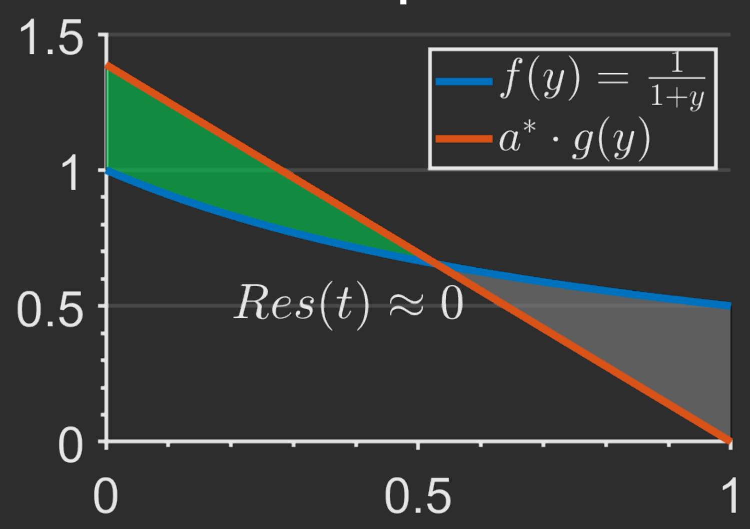
Section IV: Chapter 6 (I3D’21)

- Real-time Control Variates with one to even zero additional texture

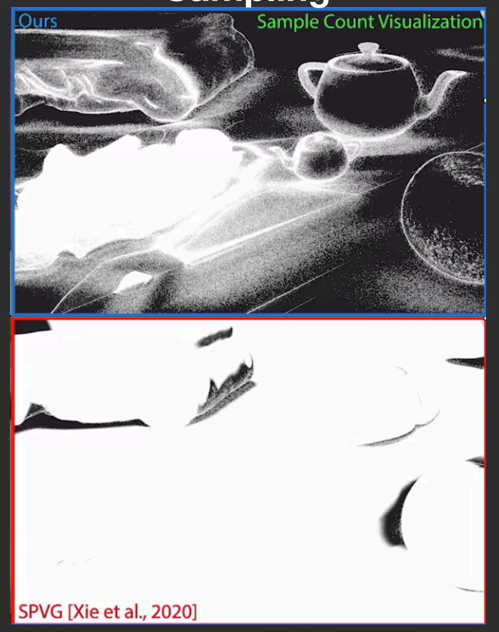
Without temporal info



With temporal info



Temporally Stable Adaptive Sampling



Introduction

Scene

Introduction

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

Background

Subsurface scattering

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

Background

Pre-integrated Lighting

Subsurface scattering

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

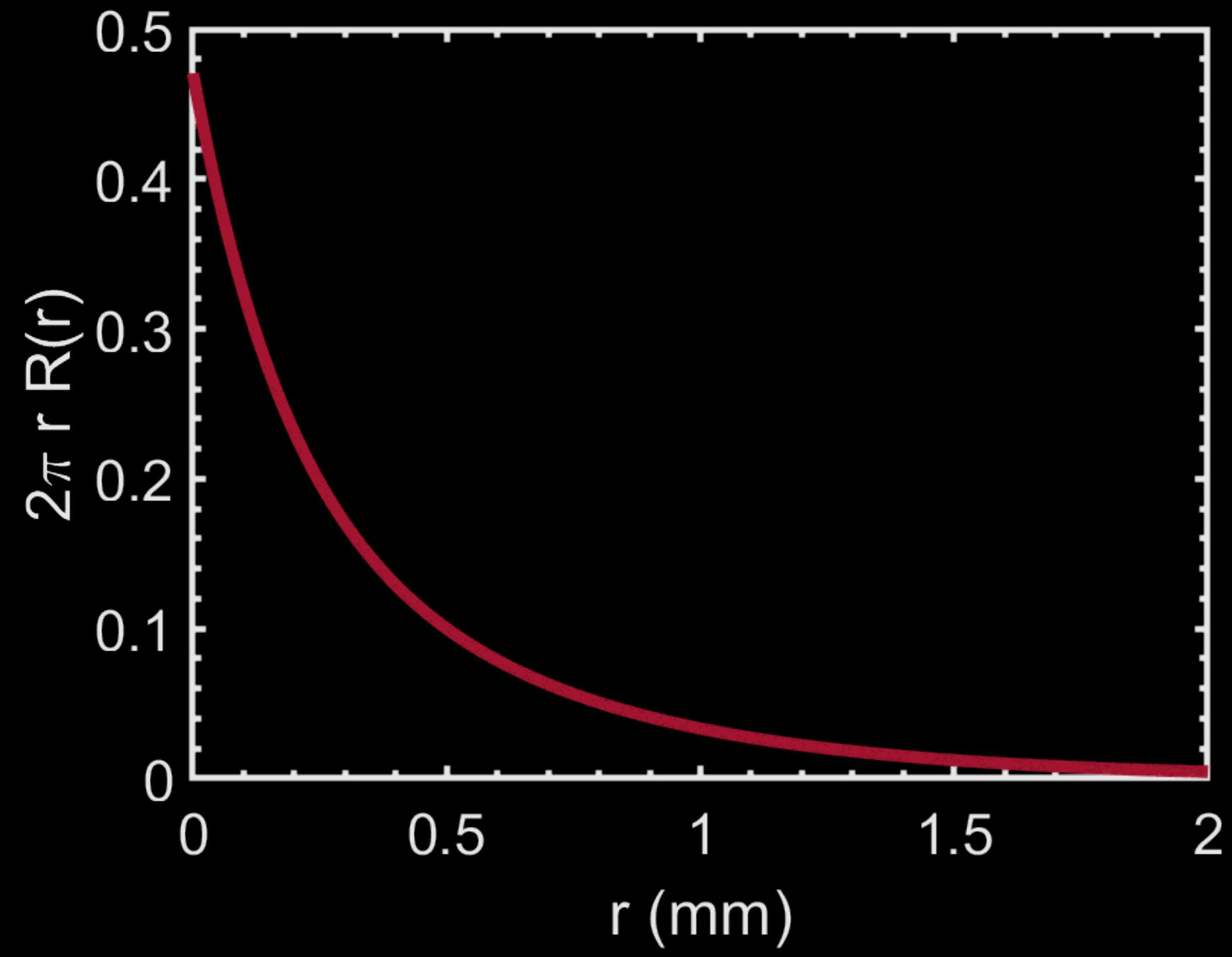
Background

Pre-integrated Lighting

Subsurface scattering

$$L_o(p,\omega_o) = \int_{\partial\Omega} \boxed{R(r_q)} \cdot \boxed{\not{b}(p,q,\omega_o)} \cdot dq$$

Diffuse Reflectance Profile



Background

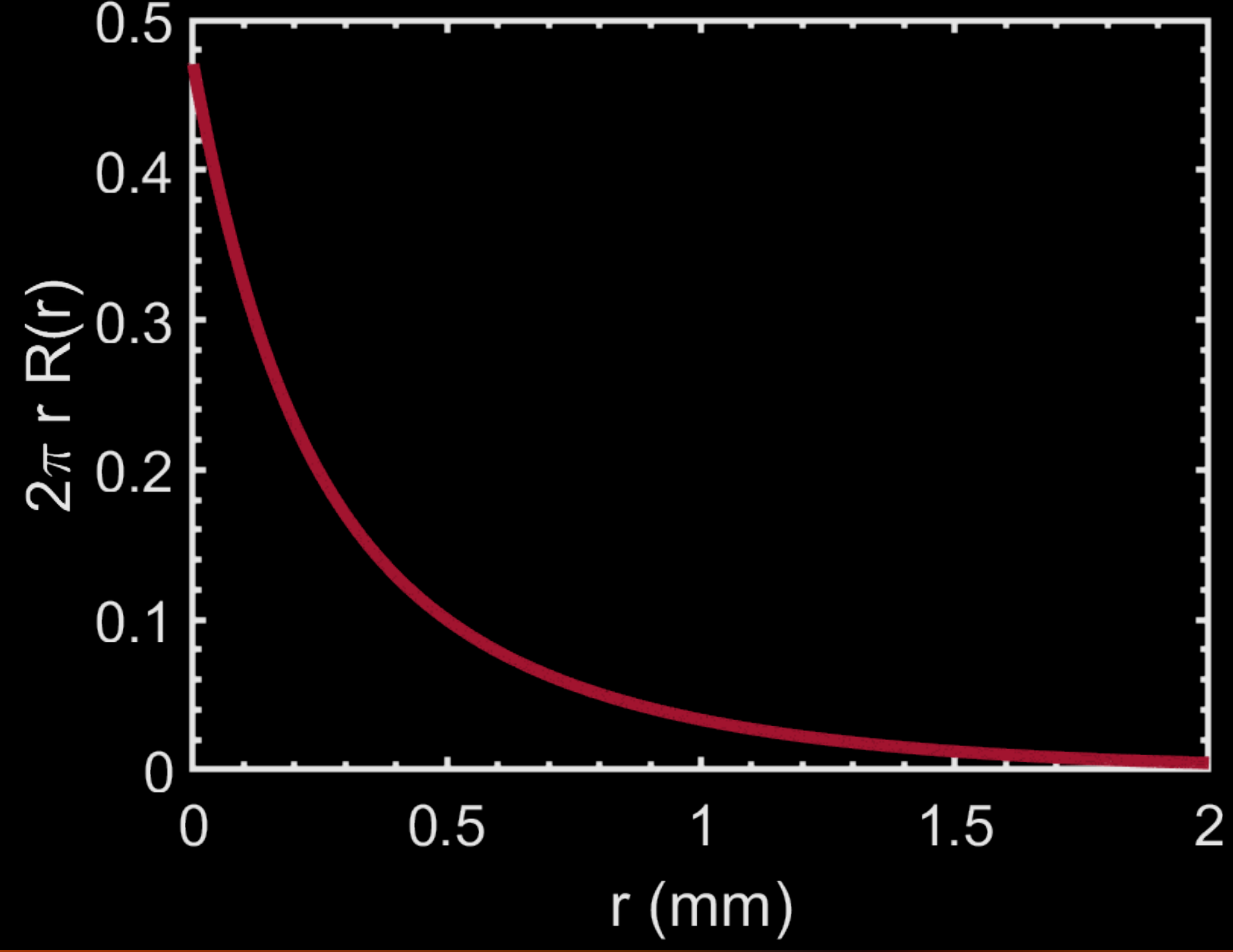
Pre-integrated Lighting

64 spp
90
2
120
60
1.5
150
1
30
0.5
180
0
210
330
240
300
270
64 spp

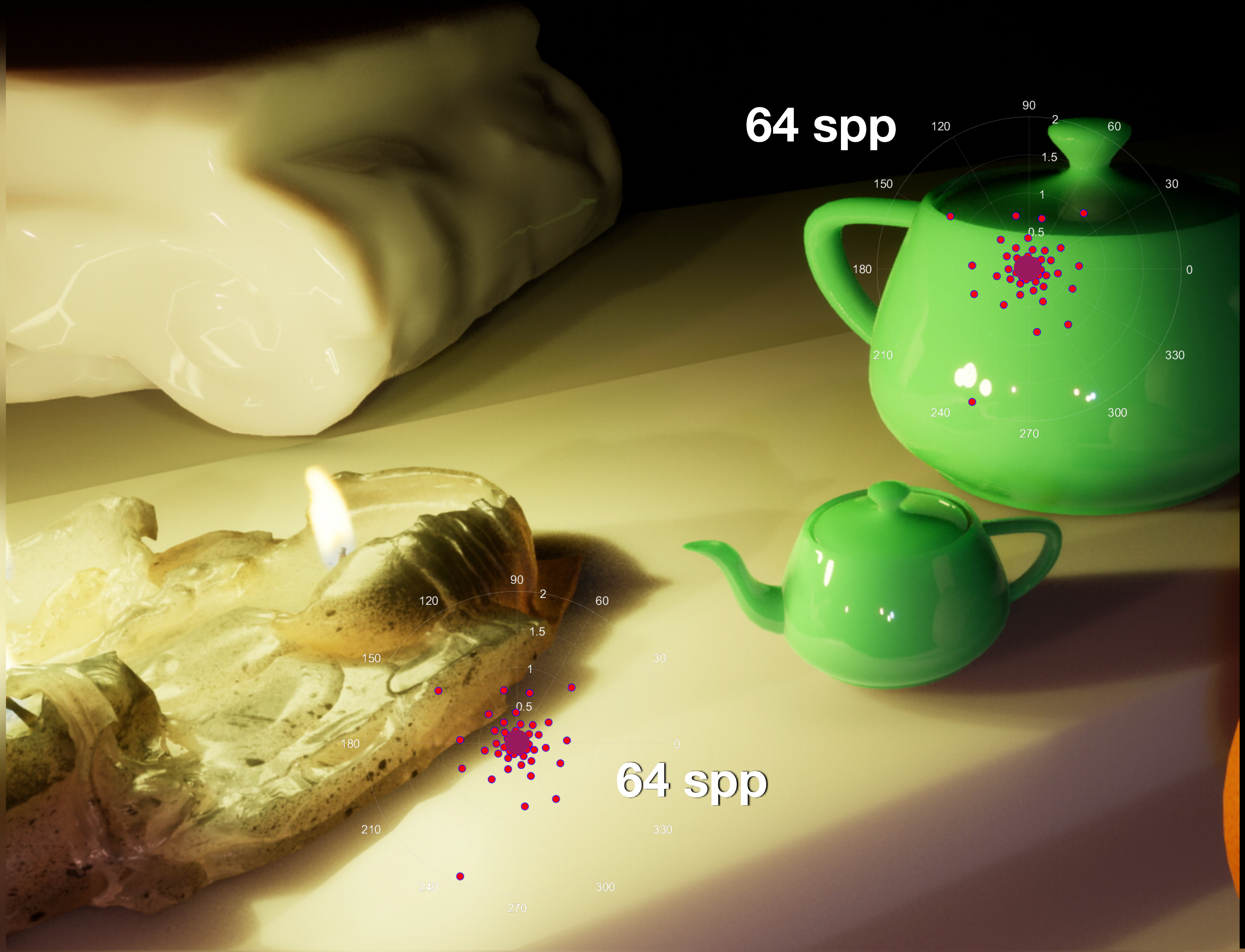
Subsurface scattering

$$L_o(p,\omega_o) = \int_{\partial\Omega} \boxed{R(r_q)} \cdot \boxed{\not{b}(p,q,\omega_o)} \cdot dq$$

Diffuse Reflectance Profile



Background

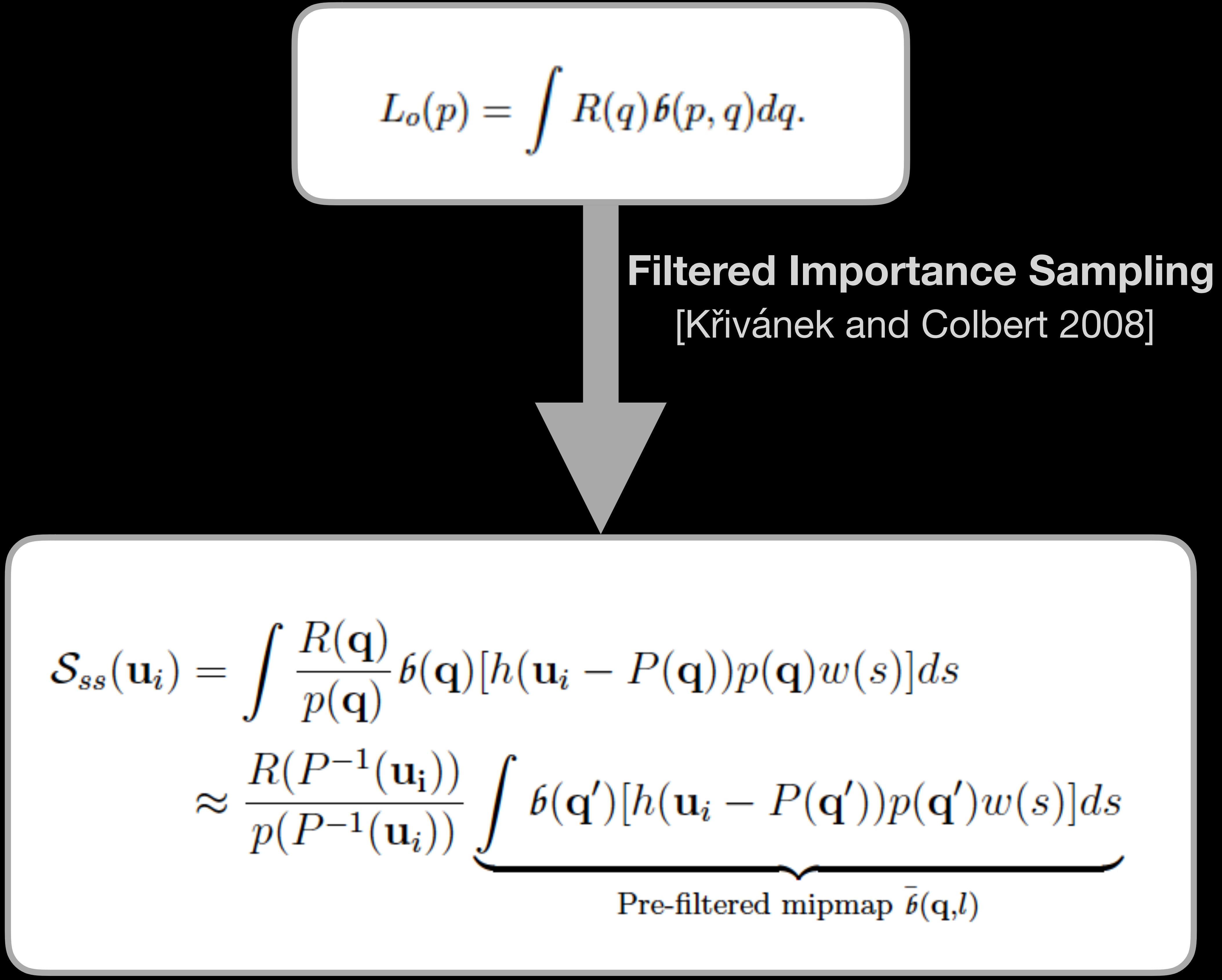


Subsurface scattering

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

Adaptive Filtered Importance Sampling

Screen-space subsurface scattering:

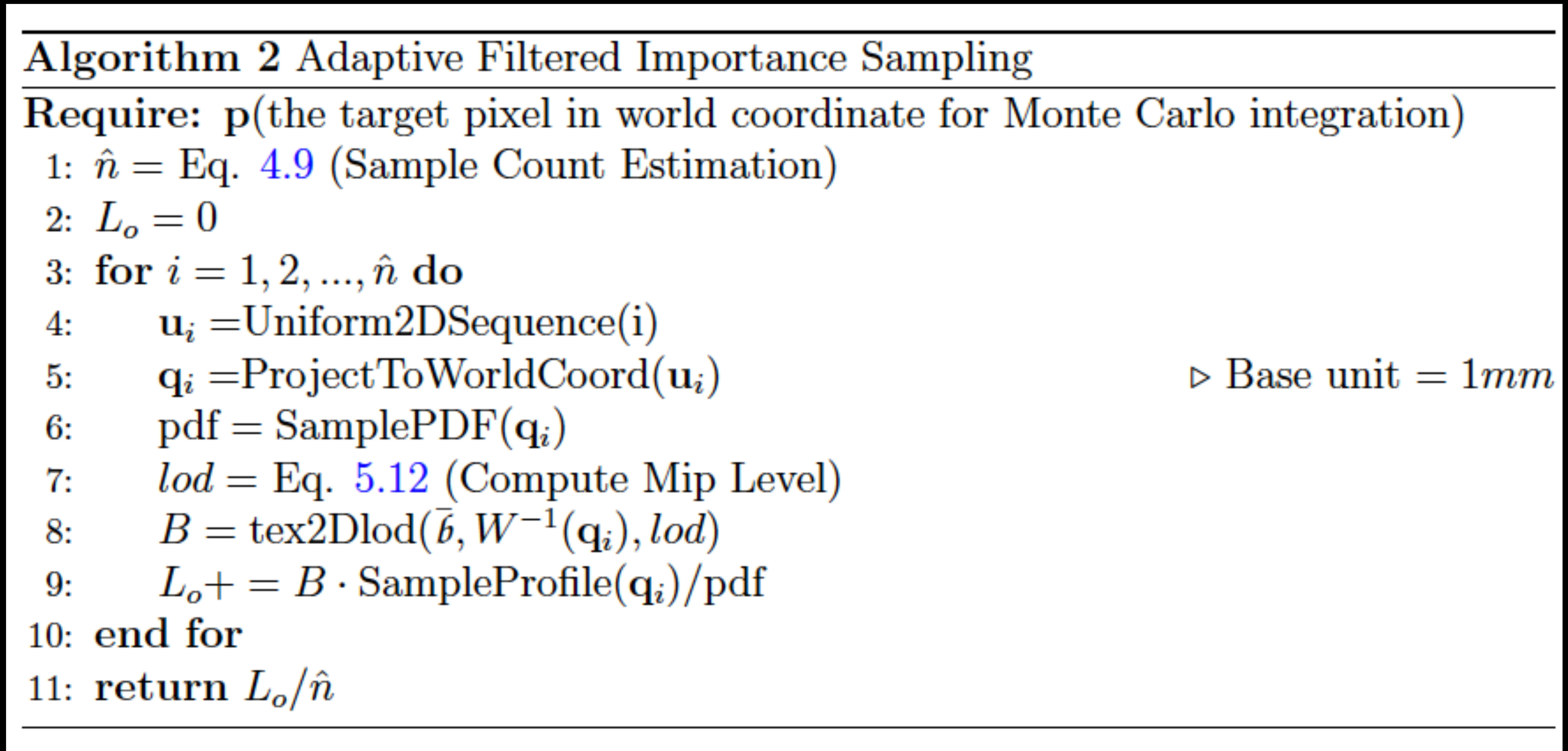


$$L_o(p) = \int R(q)\mathscr{b}(p,q)dq.$$

Filtered Importance Sampling

[Křivánek and Colbert 2008]

$$\begin{aligned}\mathcal{S}_{ss}(\mathbf{u}_i) &= \int \frac{R(\mathbf{q})}{p(\mathbf{q})}\mathscr{b}(\mathbf{q})[h(\mathbf{u}_i - P(\mathbf{q}))p(\mathbf{q})w(s)]ds \\ &\approx \frac{R(P^{-1}(\mathbf{u_i}))}{p(P^{-1}(\mathbf{u}_i))}\underbrace{\int \mathscr{b}(\mathbf{q}')[h(\mathbf{u}_i - P(\mathbf{q}'))p(\mathbf{q}')w(s)]ds}_{\text{Pre-filtered mipmap } \bar{\mathscr{b}}(\mathbf{q},l)}\end{aligned}$$



Algorithm 2 Adaptive Filtered Importance Sampling

```
Require: p(the target pixel in world coordinate for Monte Carlo integration)
$\hat{n}$ = Eq. 4.9 (Sample Count Estimation)
$L_o = 0$
for $i = 1, 2, ..., \hat{n}$ do
    $\mathbf{u}_i$ =Uniform2DSequence(i)
    $\mathbf{q}_i$ =ProjectToWorldCoord($\mathbf{u}_i$)        ▷ Base unit = 1$mm$
    pdf = SamplePDF($\mathbf{q}_i$)
    $lod$ = Eq. 5.12 (Compute Mip Level)
    $B$ = tex2Dlod($\bar{\mathscr{b}}, W^{-1}(\mathbf{q}_i), lod$)
    $L_o$+ = $B$ · SampleProfile($\mathbf{q}_i$)/pdf
end for
return $L_o/\hat{n}$
```

Compute Mip Level:

$$l = \frac{1}{2}\cdot max\left(-log_2\left(\frac{a\cdot p\cdot \hat{n}}{\ell^2_{max}\cdot t}\right),0\right)$$

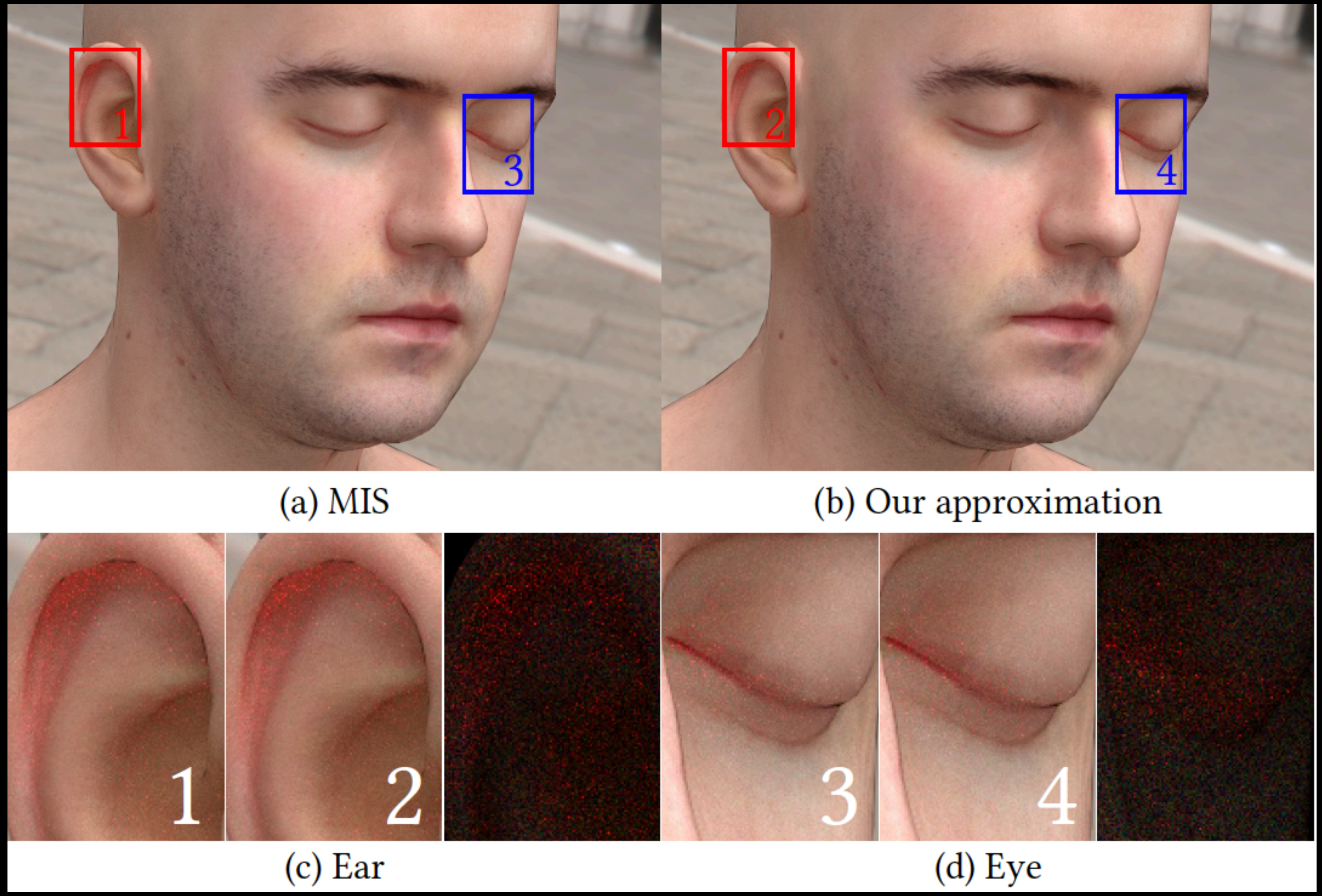
$$t = \frac{w}{D}\frac{w\cdot AspectRatio}{D} = \left(\frac{w}{D}\right)^2\cdot AspectRatio$$

Determined by both pdf and the adaptive sample count per pixel

Sampling Function

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

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



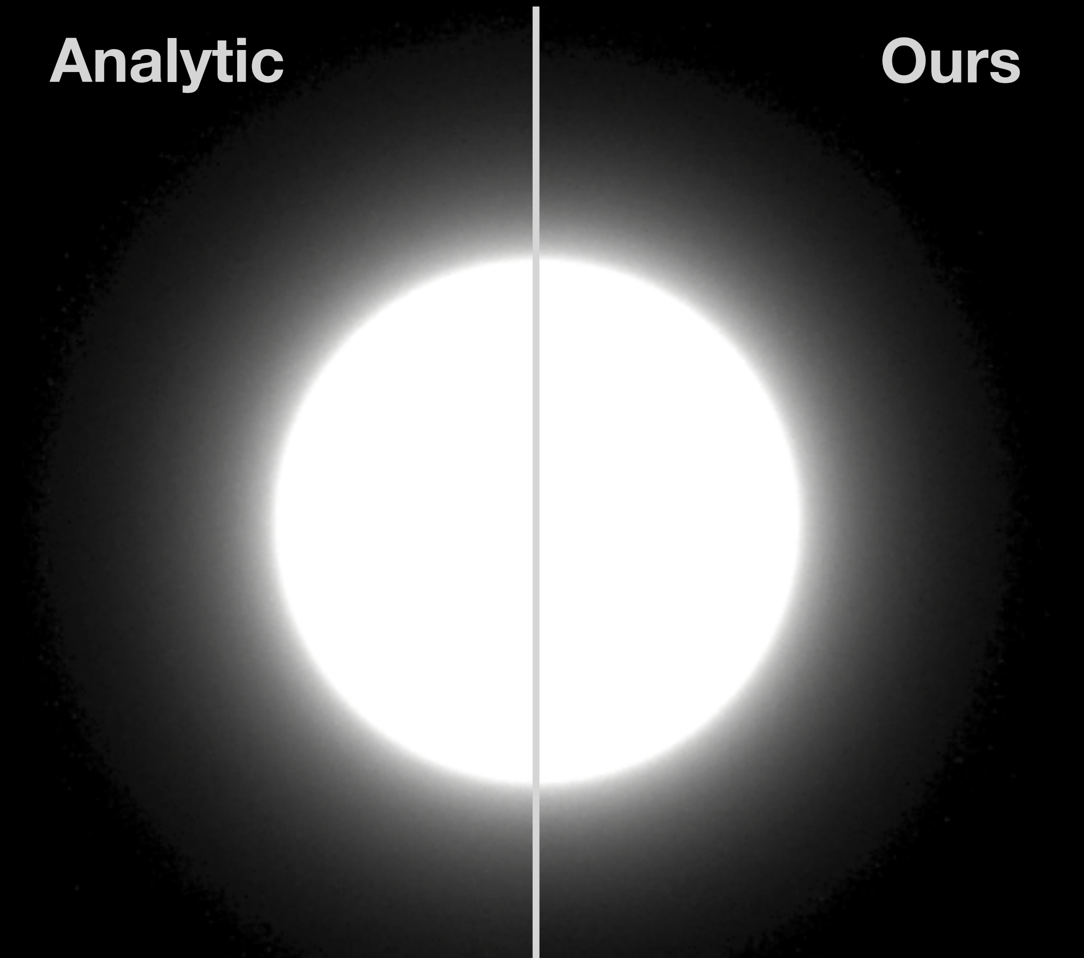
(a) MIS (b) Our approximation

(c) Ear (d) Eye

(PBRTv3)

Analytic inverse [Golubev 2019]:

$$cdf^{-1}(\xi) = 3d\log(\frac{1 + G(\xi)^{-1/3} + G(\xi)^{1/3}}{4\xi})$$
$$G(\xi) = 1 + 4\xi(2\xi + \sqrt{1 + 4\xi^2})$$



Our approximation:

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

Sampling Sequence

- Low-discrepancy sequence
  - 2D R2 sequence [Roberts, 2018]

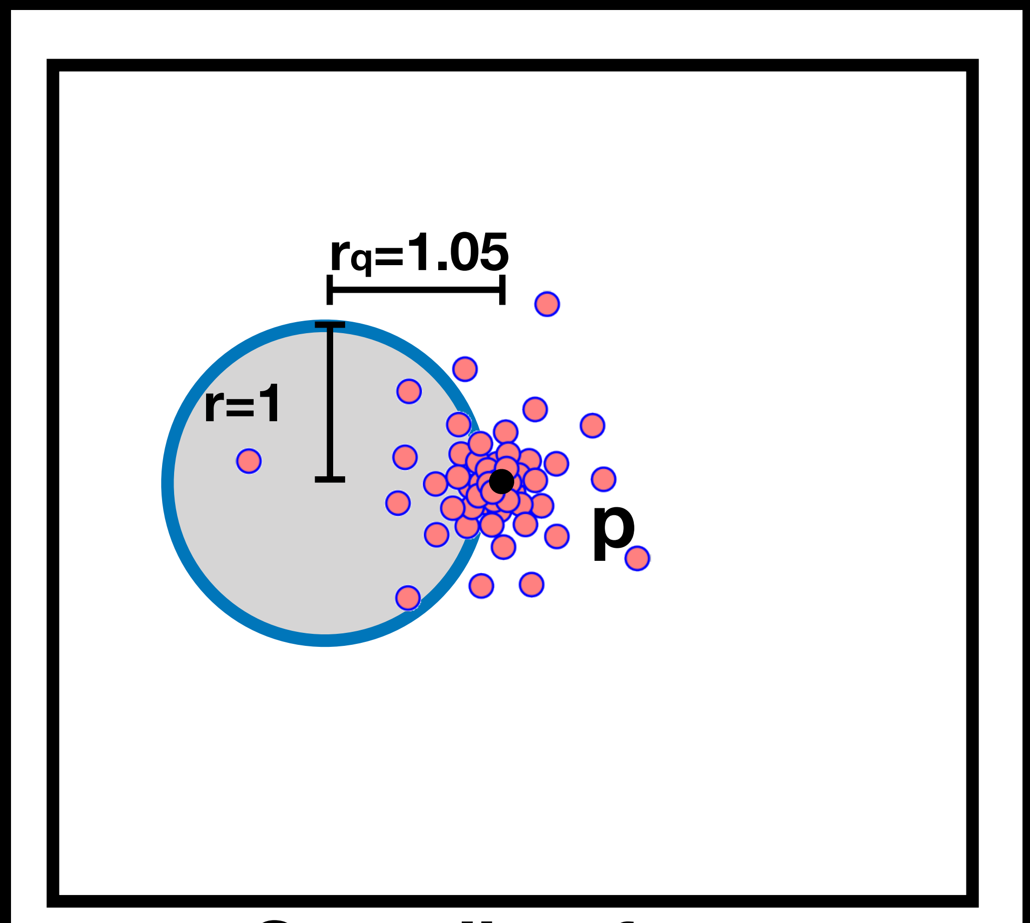
$$t_m = m\psi, m = 1, 2, 3, \ldots$$

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

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

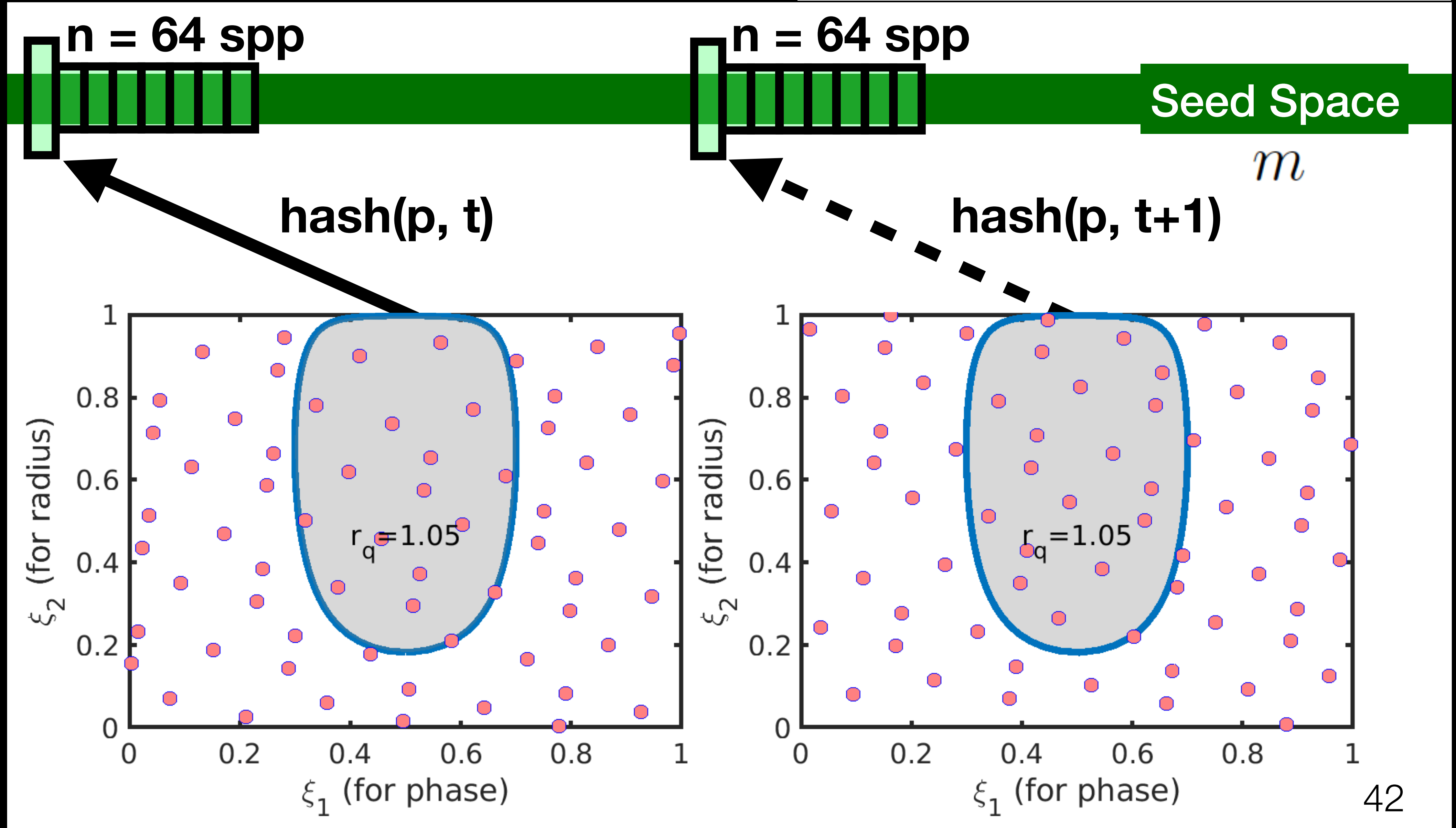
- Random number generator [Jarzynski and Olano, 2020]
  - Hash the location and frame index with pcg3d16

(a)

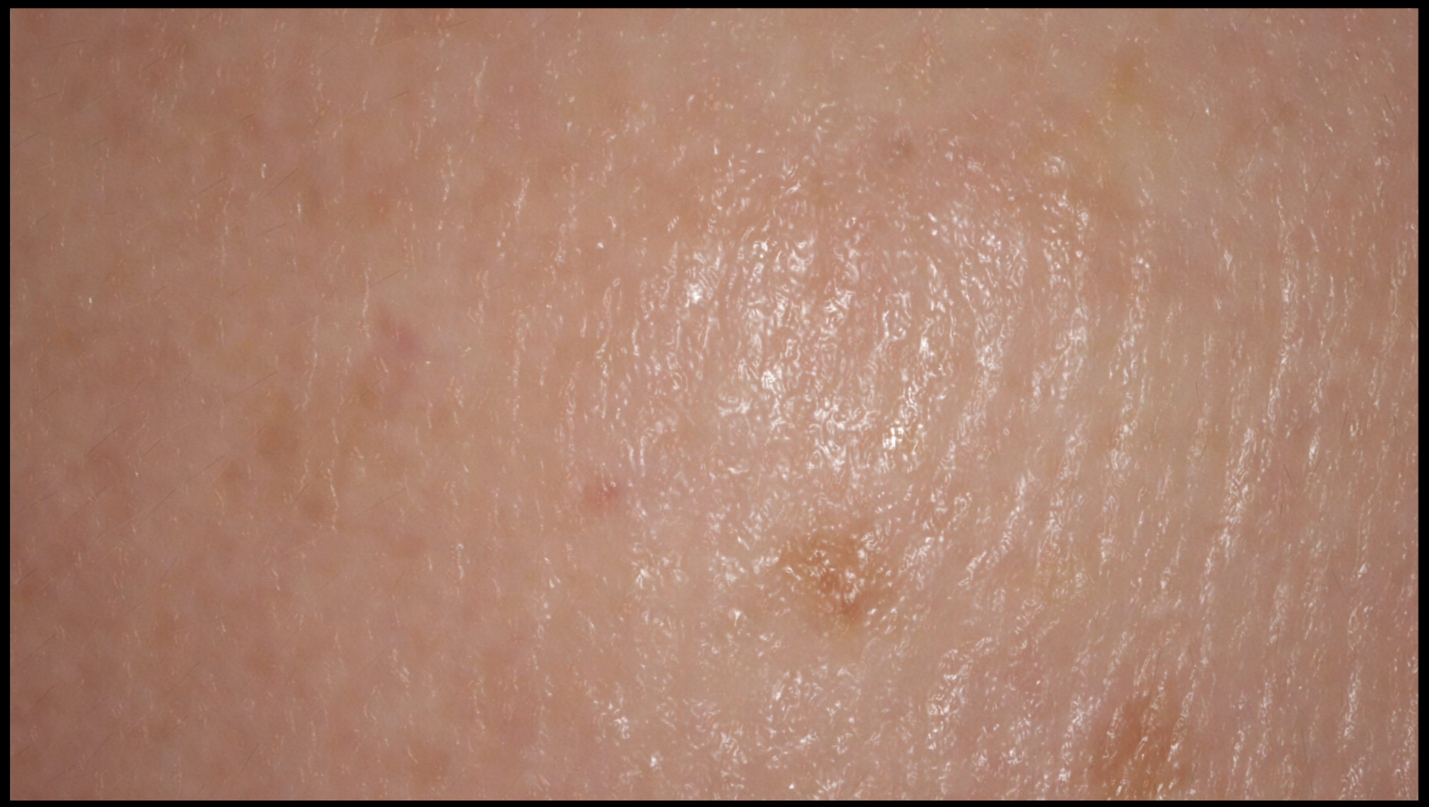


Sampling for p
(L = 1, A = 0.8)

(b)



Adaptive Filtered Importance Sampling



a) Close patch

b) Ear

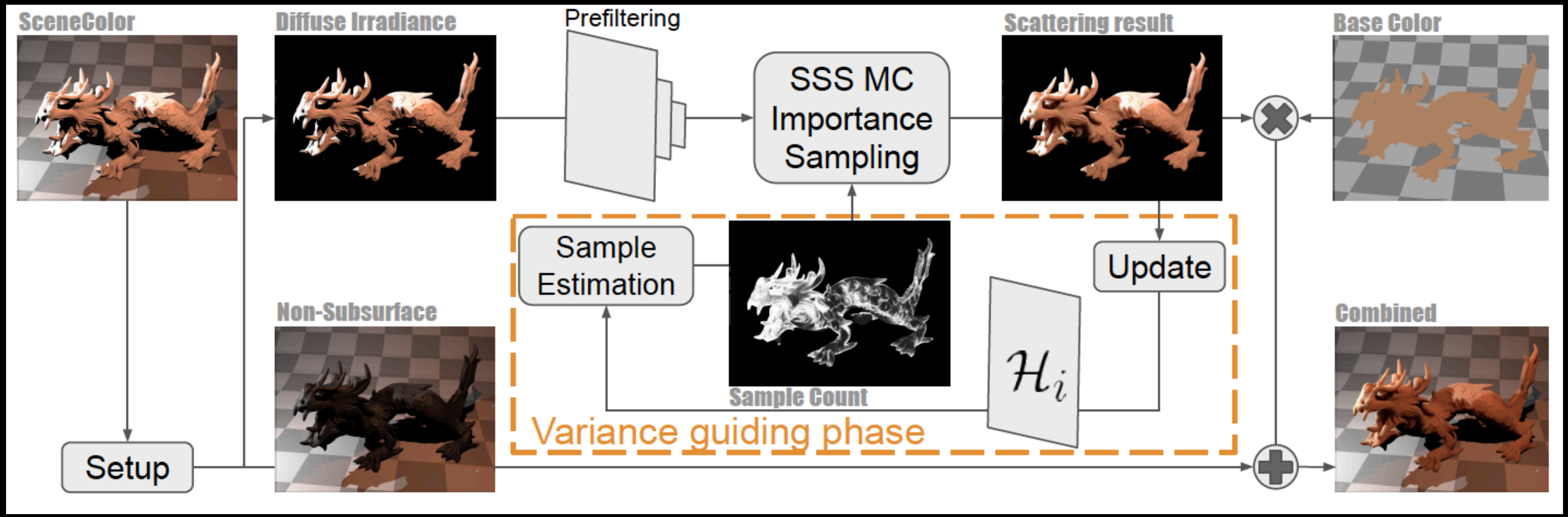
c) Front

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

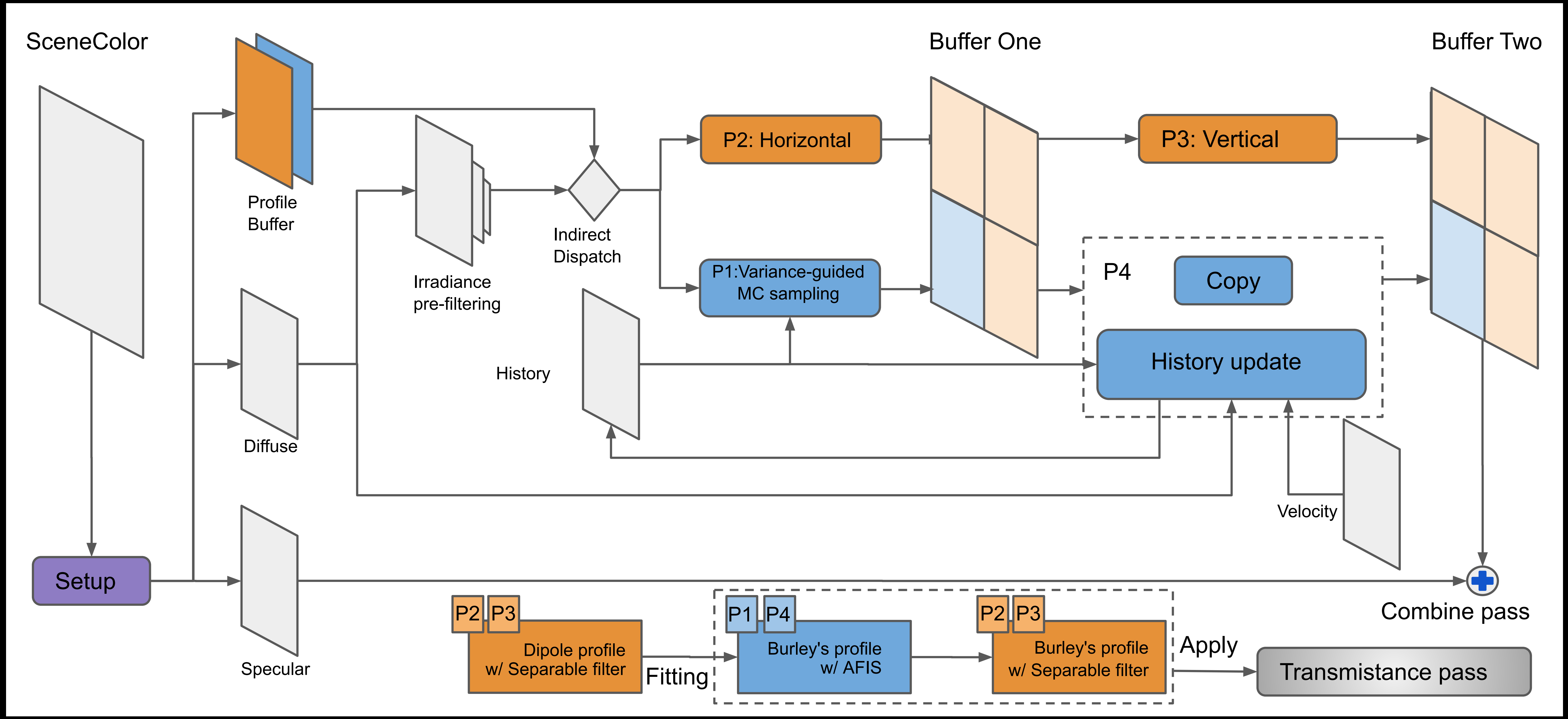
Scenario	Setup	Pre-filtering	Sampling	Update	Combine	Total
a)+fixed	0.38	0.16	10.73	N/A	0.20	11.47
a)+adt.	0.38	0.16	1.50	0.54	0.20	2.78
b)+fixed	0.41	0.17	9.72	N/A	0.27	10.22
b)+adt.	0.41	0.17	2.72	0.46	0.27	4.03
c)+fixed	0.35	0.17	1.45	N/A	0.11	2.08
c)+adt.	0.35	0.17	0.38	0.14	0.11	1.15

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

Importance-guided Acceleration



(a)

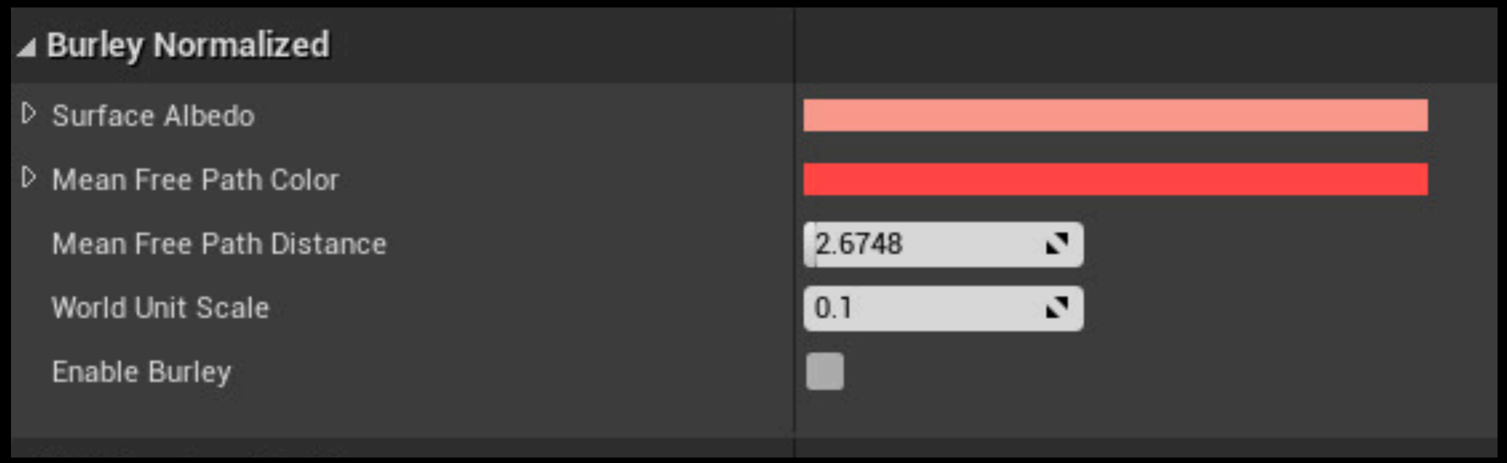


(b)

Acceleration technique type

- **AFIS**
  - High quality
  - Performance based on lighting complexity
- **Separable**
  - Lower quality
  - Constant performance

Use **AFIS**



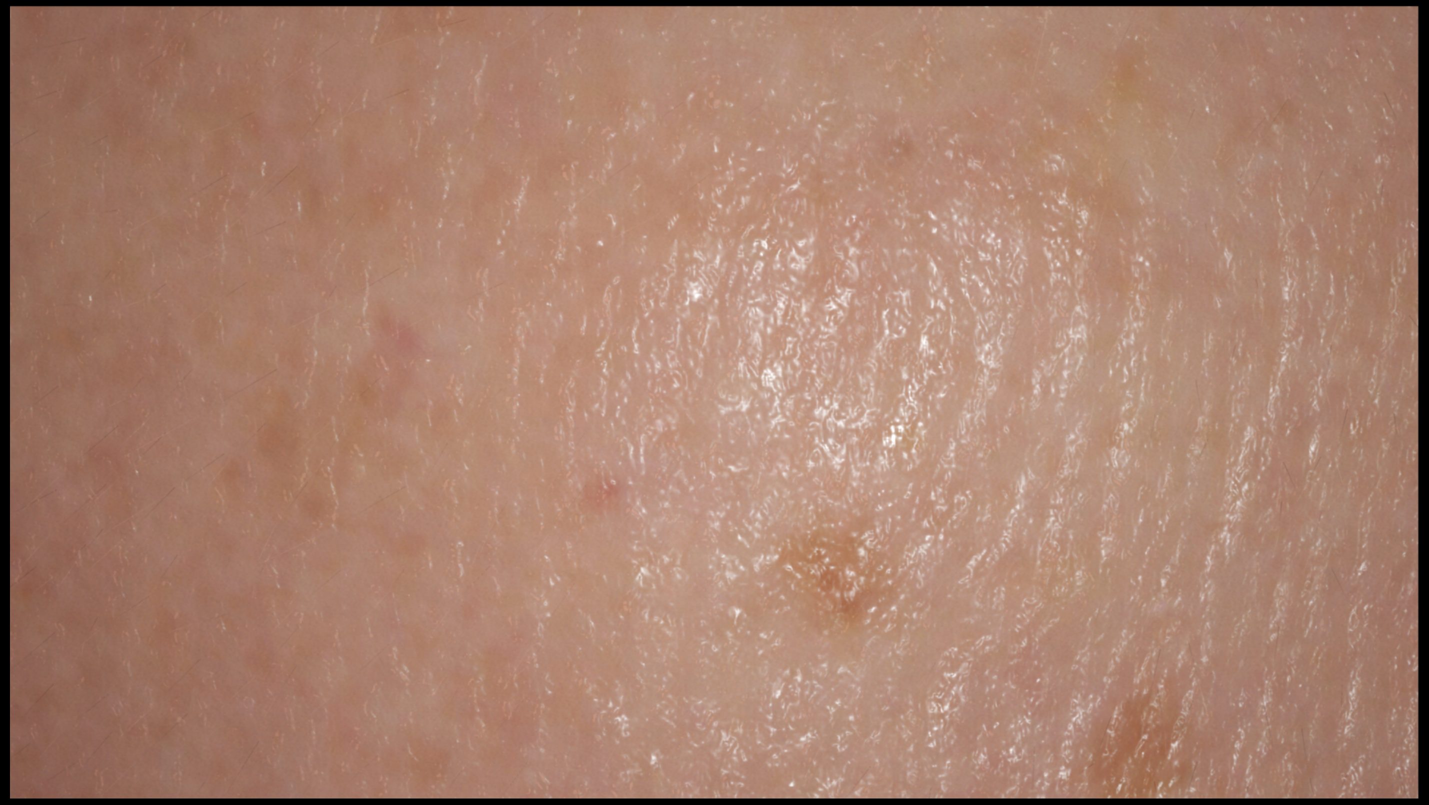
Burley Normalized

Surface Albedo	
Mean Free Path Color	
Mean Free Path Distance	2.6748
World Unit Scale	0.1
Enable Burley	□

Use **Separable** filters

r.sss.burley.quality 0

Importance-guided Acceleration



a) Close patch

b) Ear

c) Front

AFIS (ms):

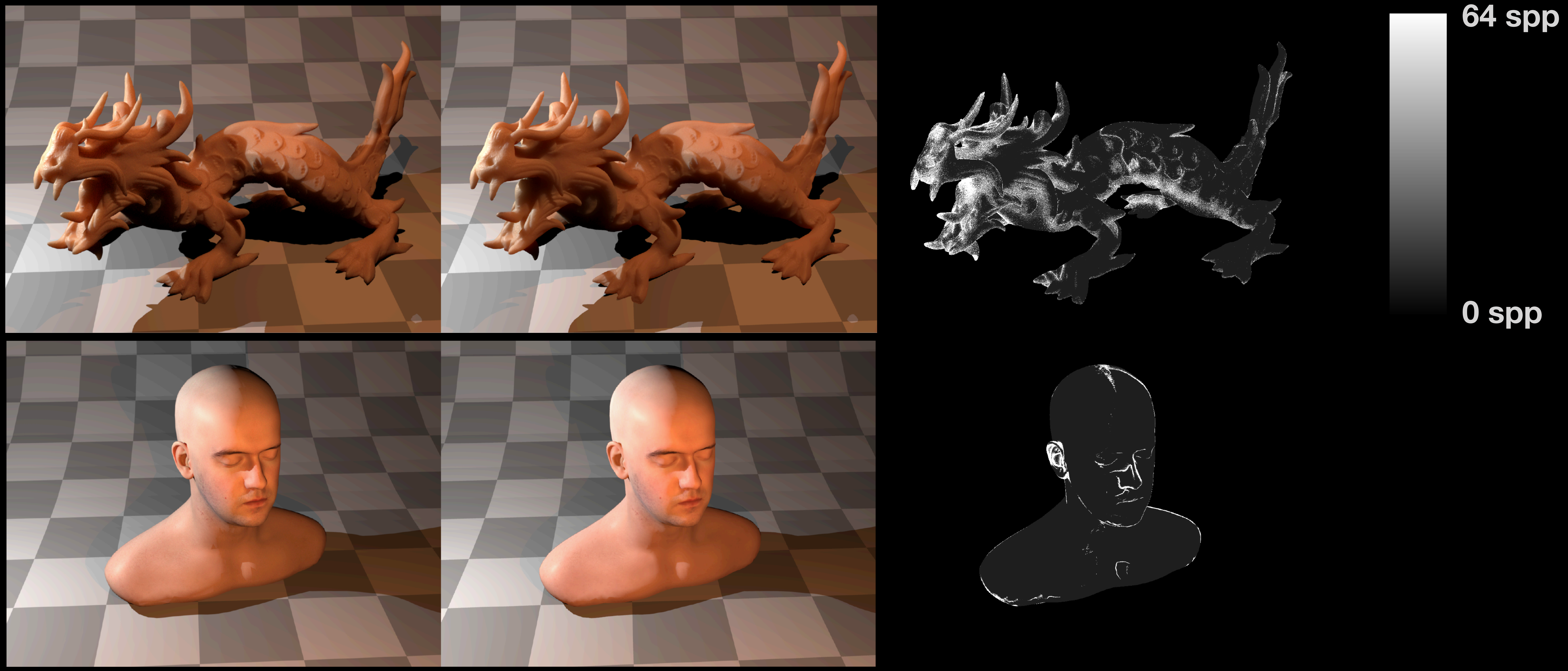
Scenario	Setup	Pre-filtering	Sampling	Update	Combine	Total	PSNR
a)+adt.	0.38	0.16	1.50	0.54	0.20	2.78	37.58 dB
b)+adt.	0.41	0.17	2.72	0.46	0.27	4.03	45.70 dB
c)+adt.	0.35	0.17	0.38	0.14	0.11	1.15	44.28 dB

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).

Result - Ground Truth Comparison



PBRT

Ours

Adaptive sample count

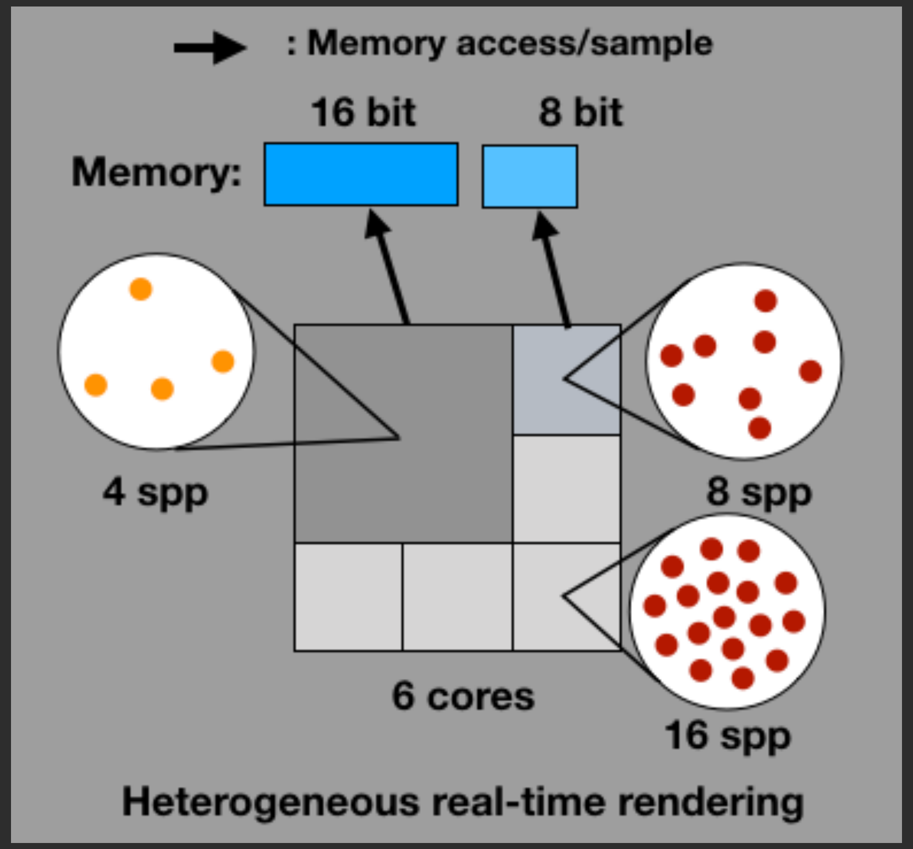
Dragon

Infinite head

Outline

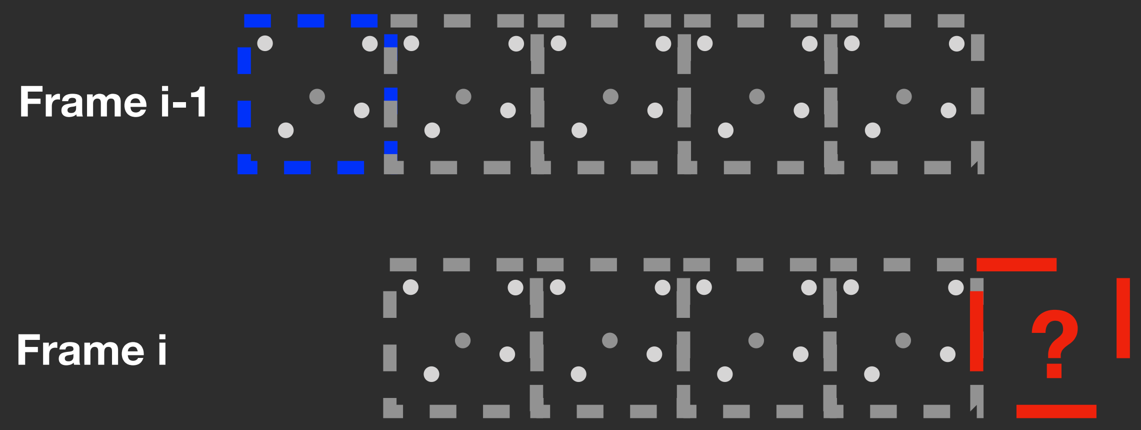
Section I: Chapter 1 ~ Chapter 3

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



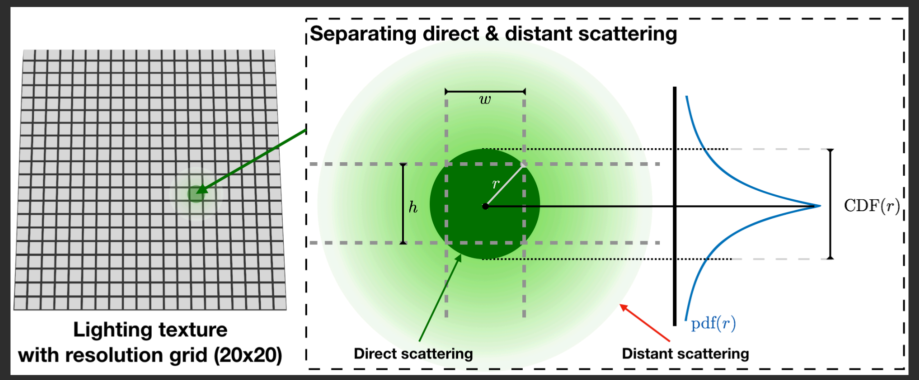
Section II: Chapter 4 (I3D'20)

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



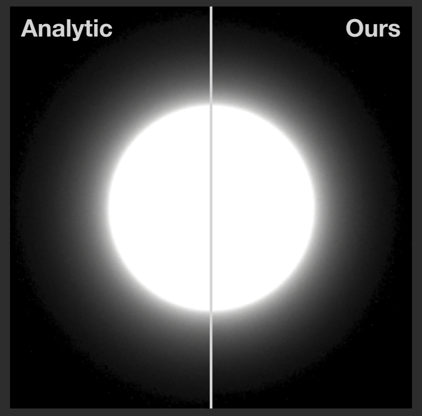
Section III: Chapter 5

- Subsurface Scattering

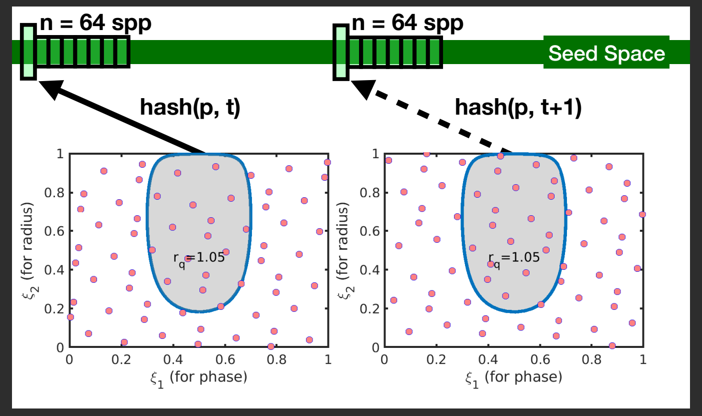


Separating direct & distant scattering
Lighting texture
with resolution grid (20x20)
Direct scattering
Distant scattering
CDF(r)
pdf(r)

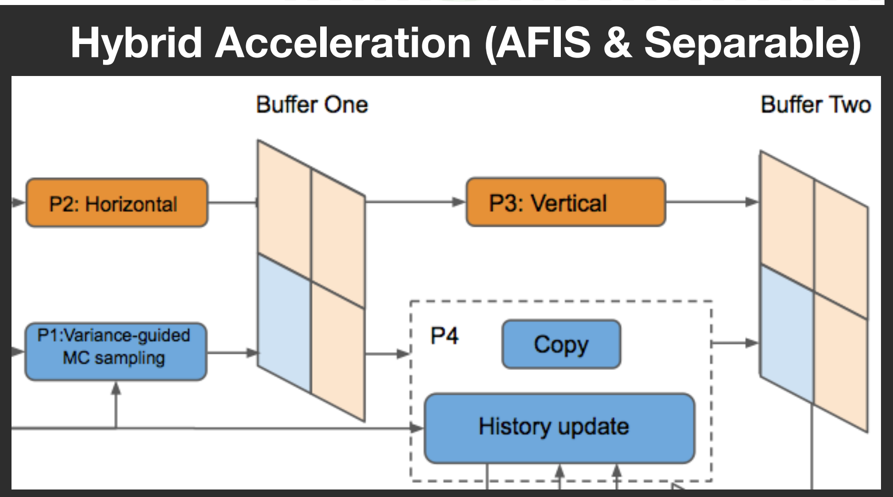
Sampling function approximation



Sampling sequence



Hybrid Acceleration (AFIS & Separable)

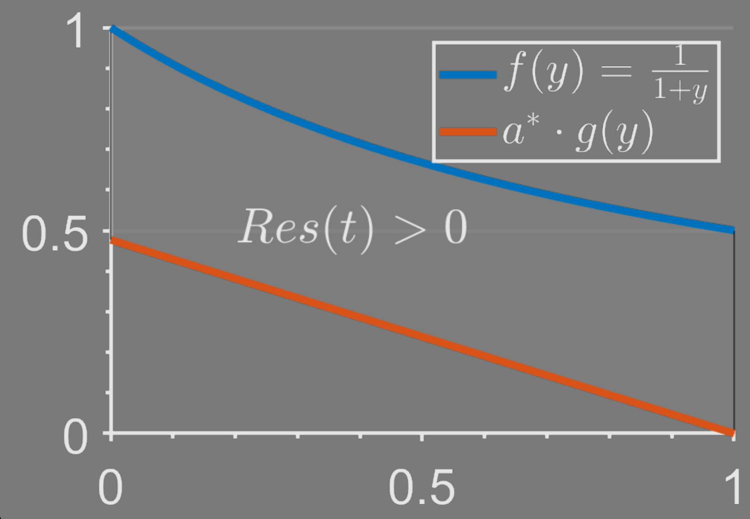


Section IV: Chapter 6 (I3D'21)

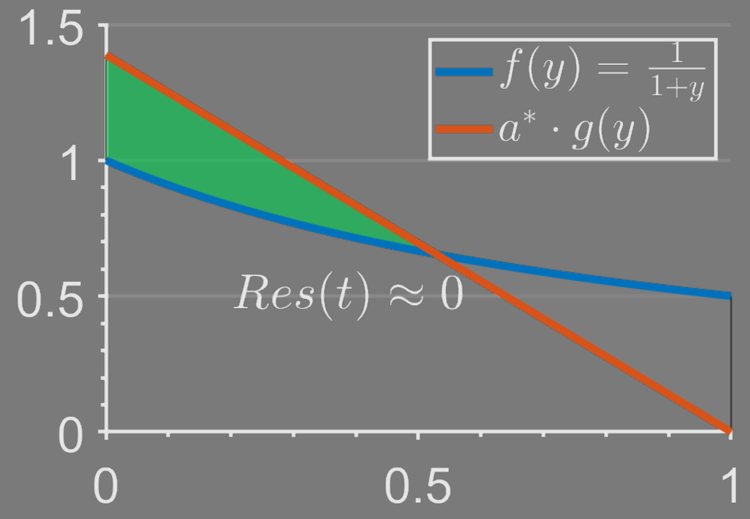
- Real-time Control Variates

with one to even zero additional texture

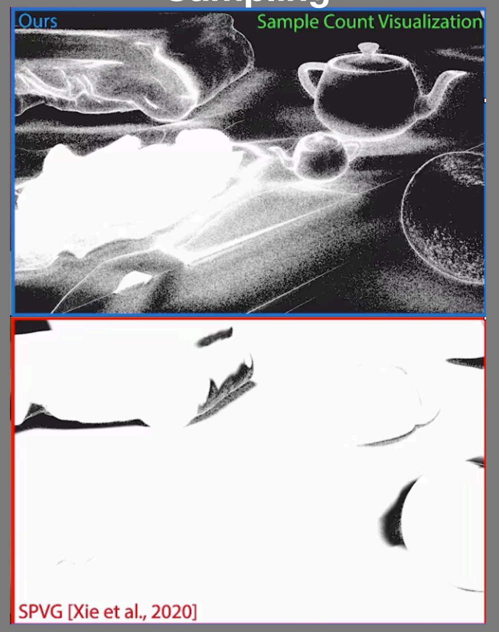
Without temporal info



With temporal info

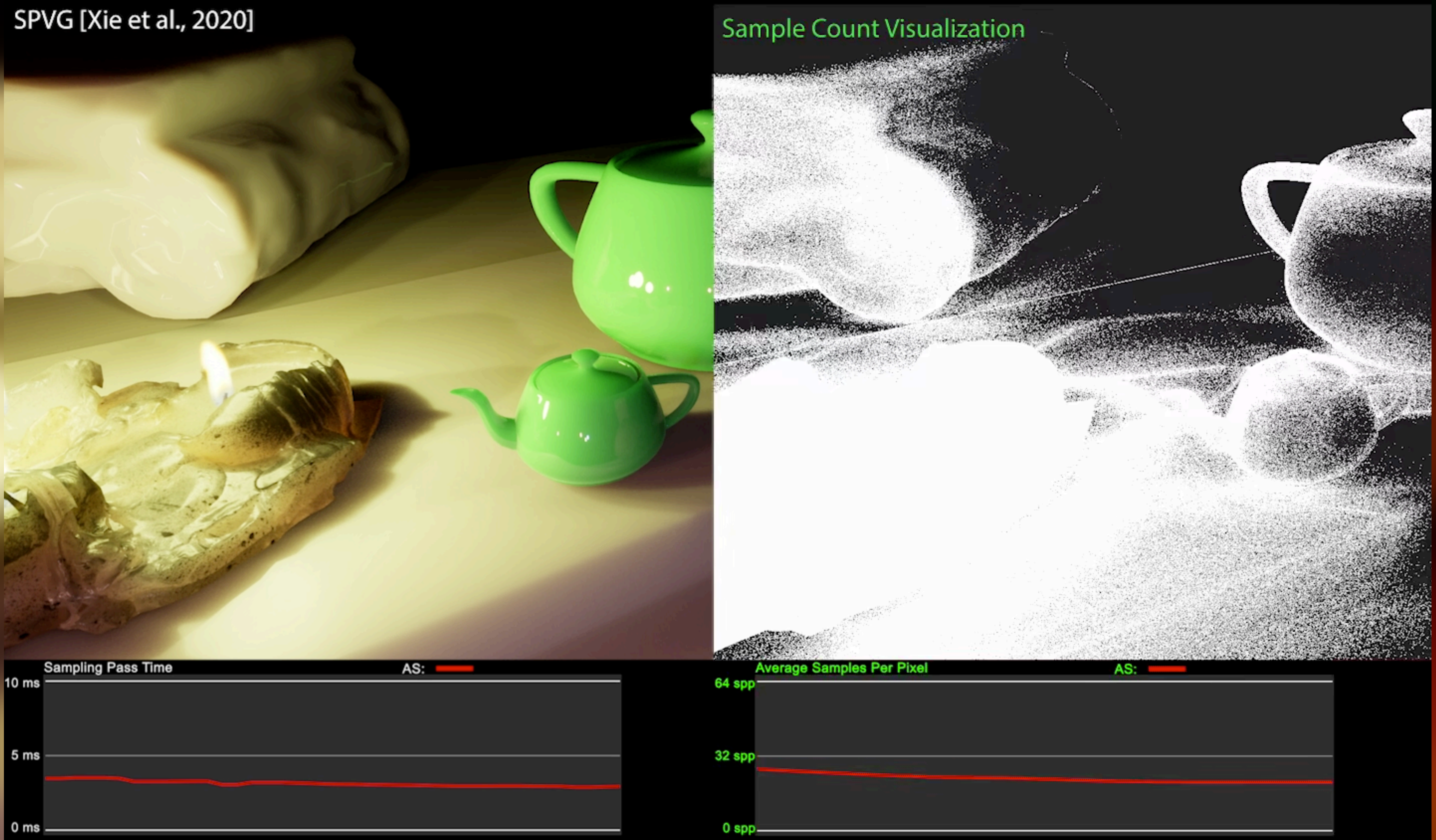


Temporally Stable Adaptive Sampling



Motivation Example

Dynamic lighting leads to unstable performance

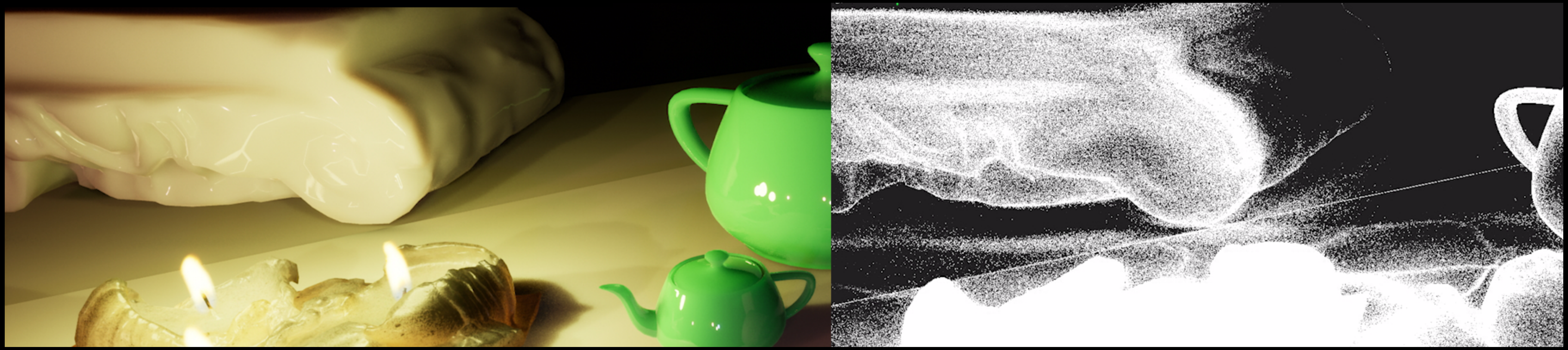


1440p

Motivation Example

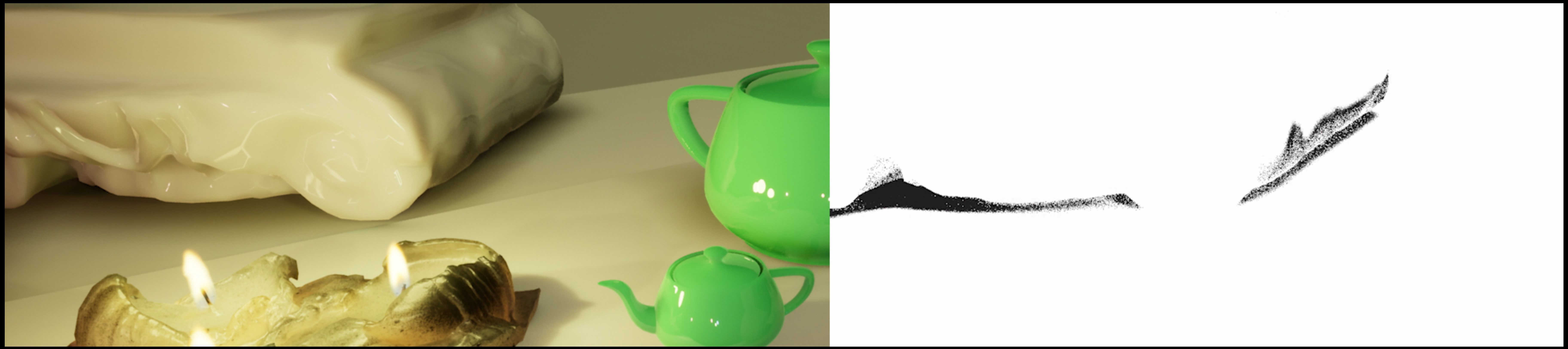
Static scene:

- Use spatial variance



Dynamic scene:

- Reduce temporal variance



How?

Temporally Stable Adaptive Sampling

- Reduce temporal variance with Control Variates

Variance Analysis

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

$$Var(\langle TF \rangle) = (Var(T) + E[T]^2)\,Var(F) + Var(T)\,E[F]^2$$

(*T* and *F* are independent)

Variance Analysis

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

$$Var(\langle TF \rangle) = (Var(T) + E[T]^2)\,Var(F) + Var(T)\,E[F]^2$$

(T and F are independent)

Sample Count ↑ $Var(F) \downarrow$

Variance Analysis

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

$$Var(\langle TF \rangle) = \left(Var(T) + E[T]^2 \right) Var(F) + Var(T)E[F]^2$$

(T and F are independent)

Sample Count ↑ $Var(F) \downarrow$

$$Var(\langle TF \rangle) = \boxed{\left(Var(T) + E[T]^2 \right) \; Var(F)} + Var(T)E[F]^2$$

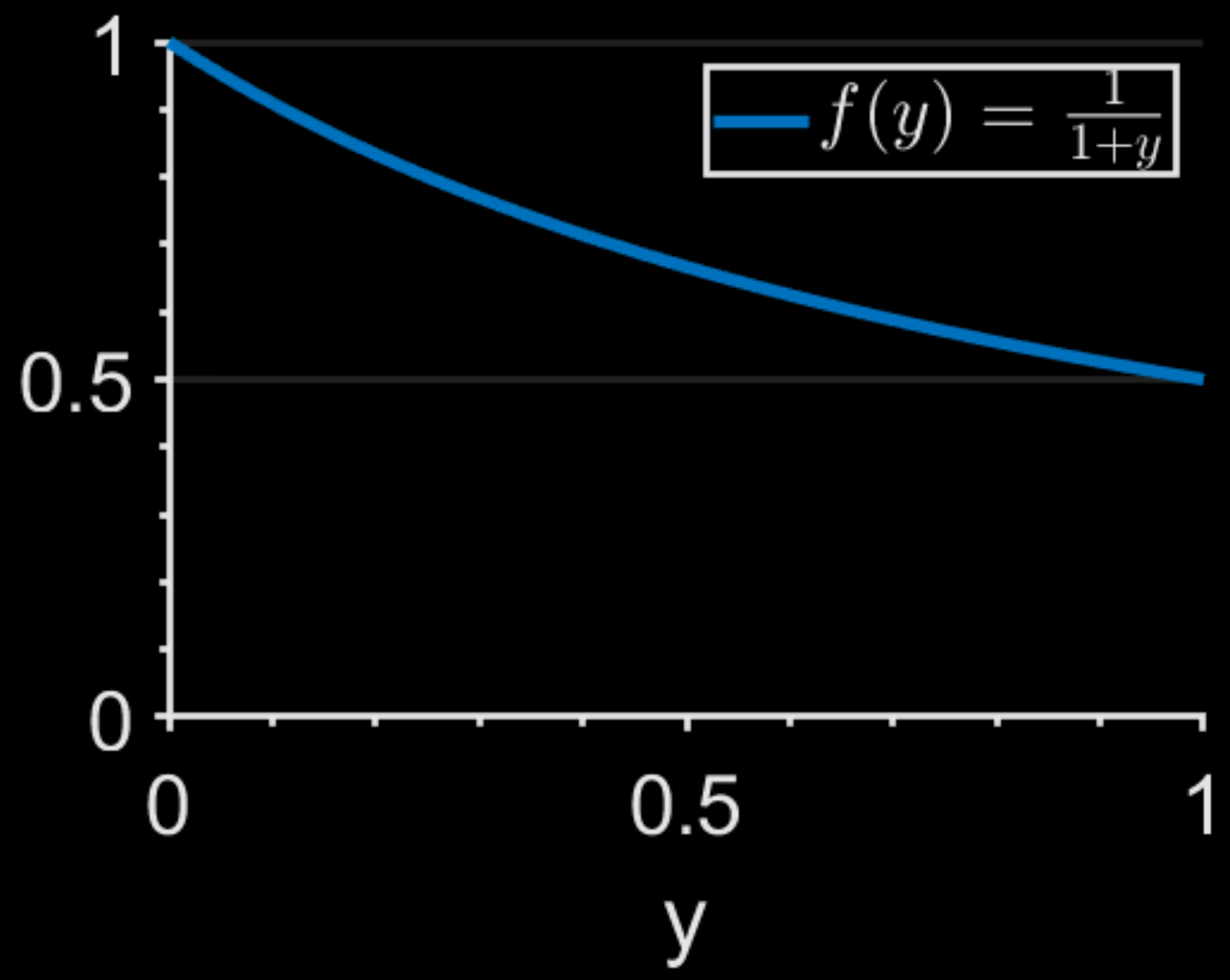
↓

Unaffected

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

What is $\left(Var(T) + E[T]^2 \right) Var(F)$ **?** **Variance of CV residual, Res(t)**

Standard CV

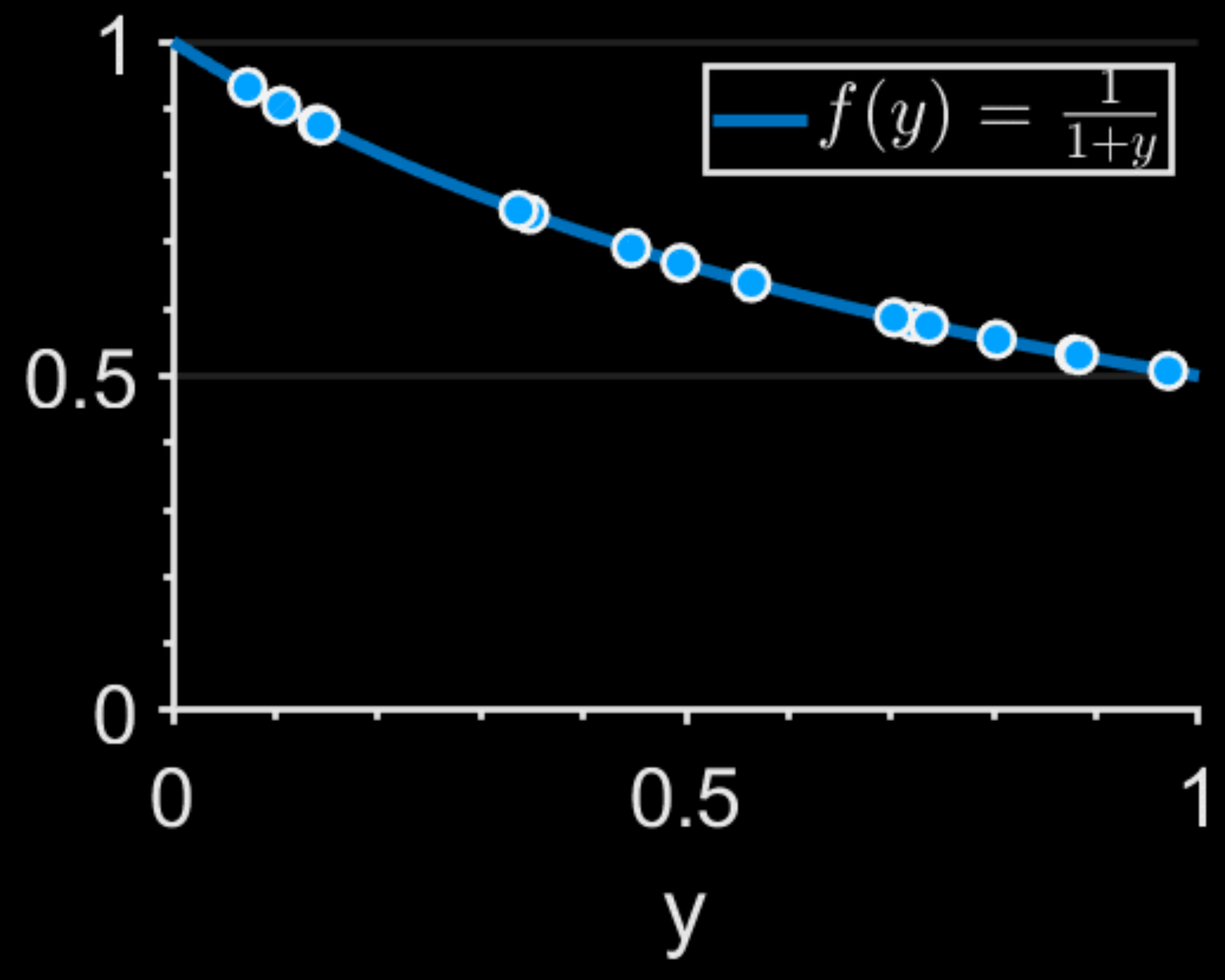


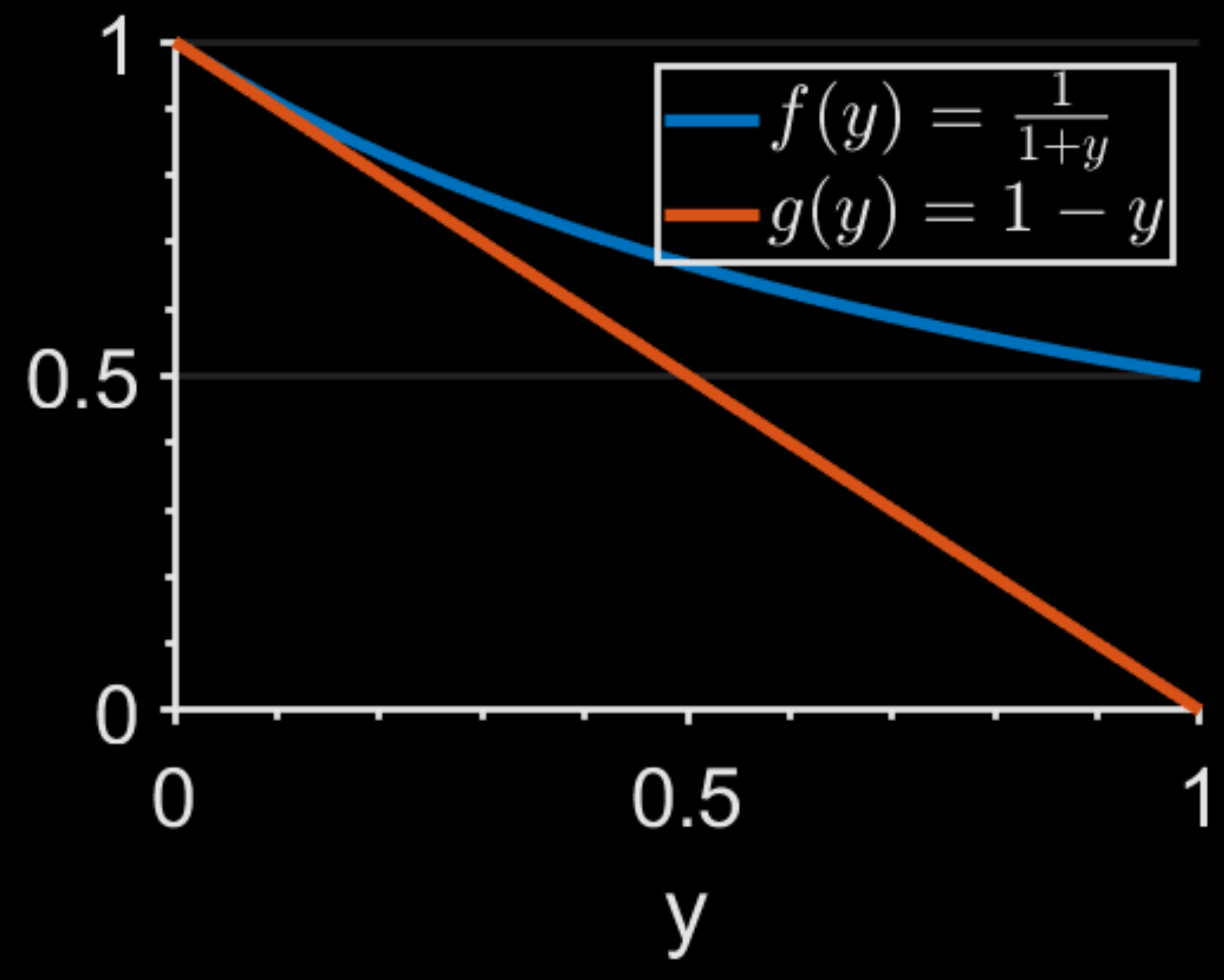
$$F = \int_{y \in \mathcal{D}} f(y) dy$$

16 spp **0.6841**

F = ln(2) ≈ 0.6931

Standard CV





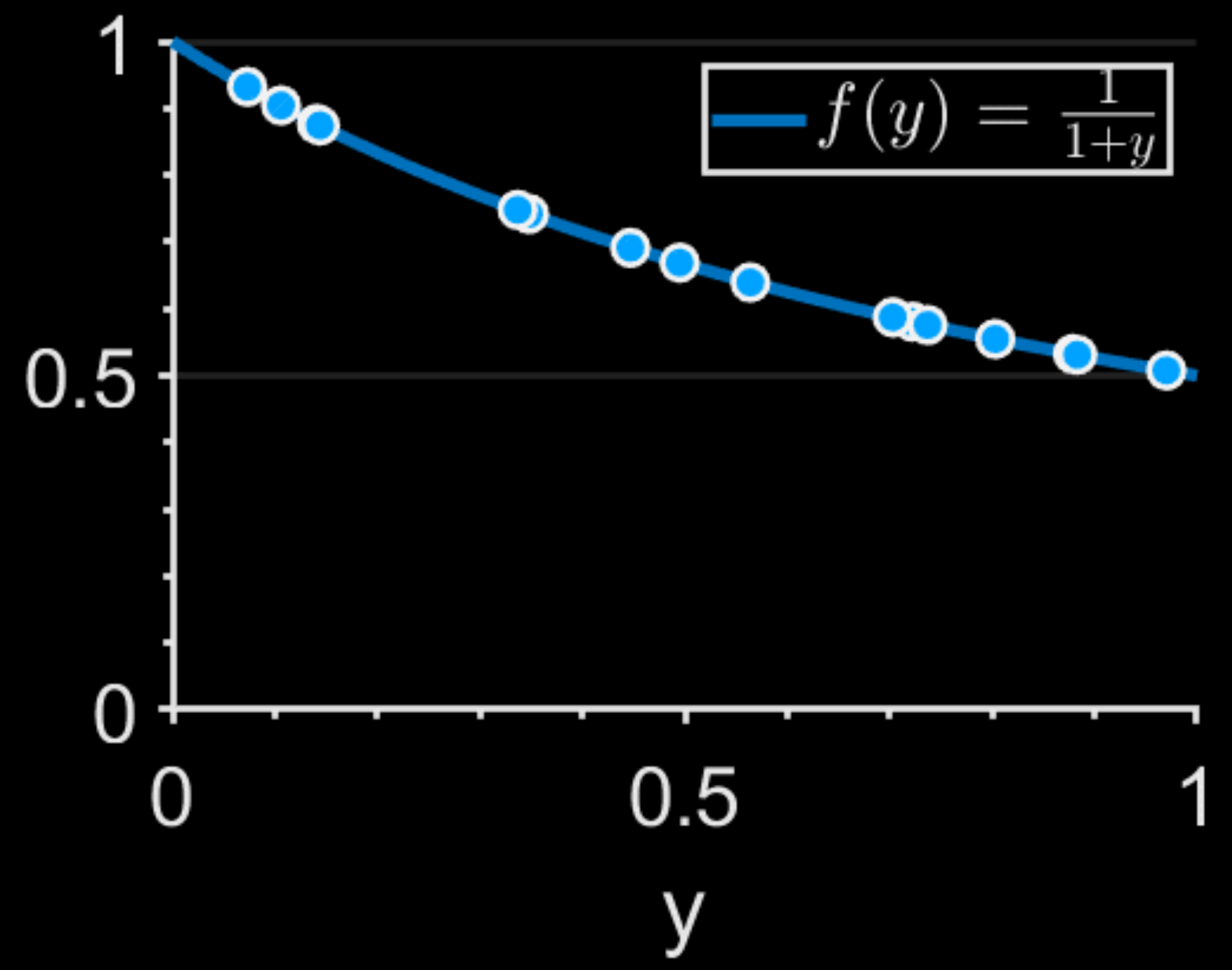
$$F = \int_{y \in \mathcal{D}} f(y) dy$$

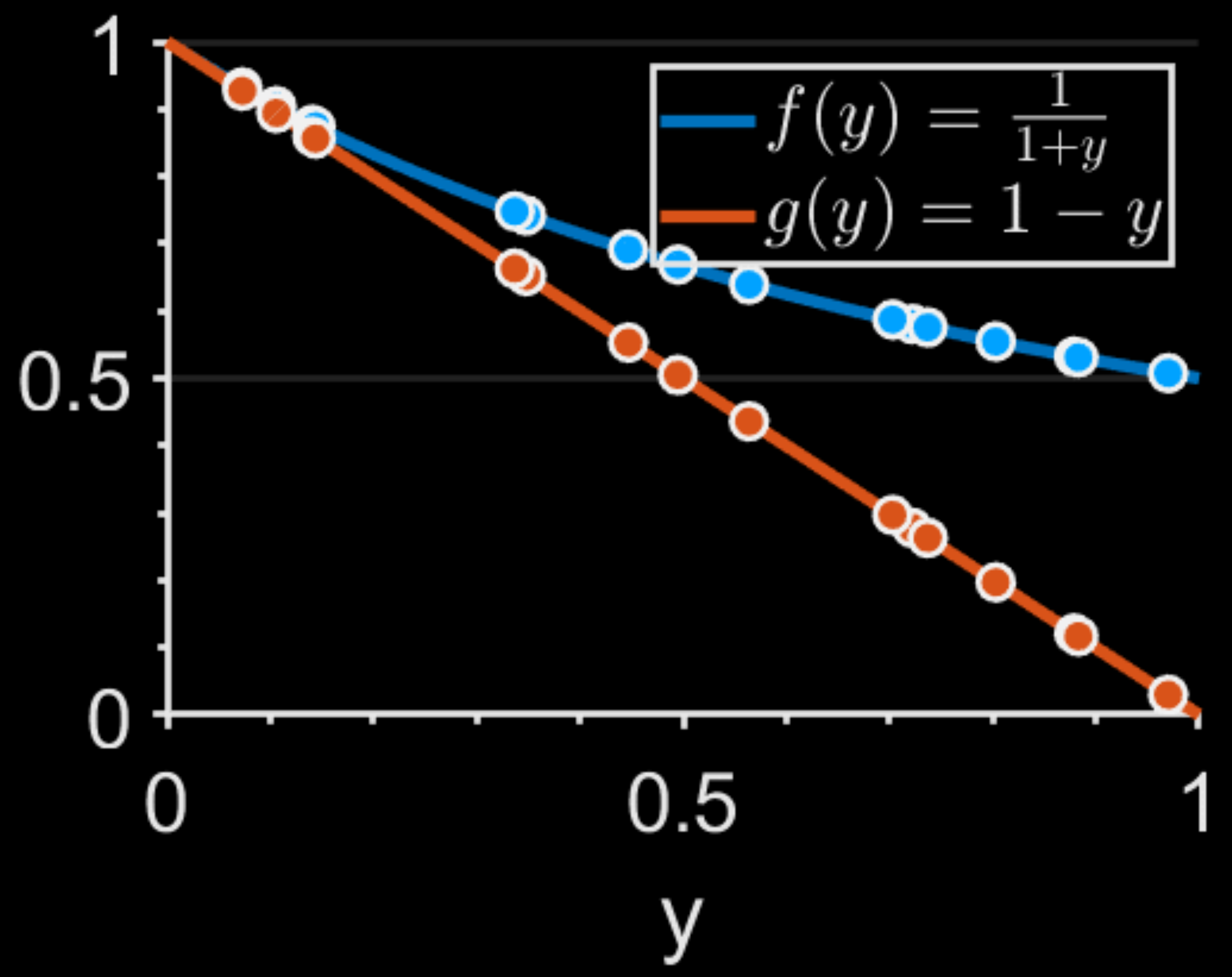
$$F = G + \int_{y \in \mathcal{D}} f(y) - g(y) dy$$

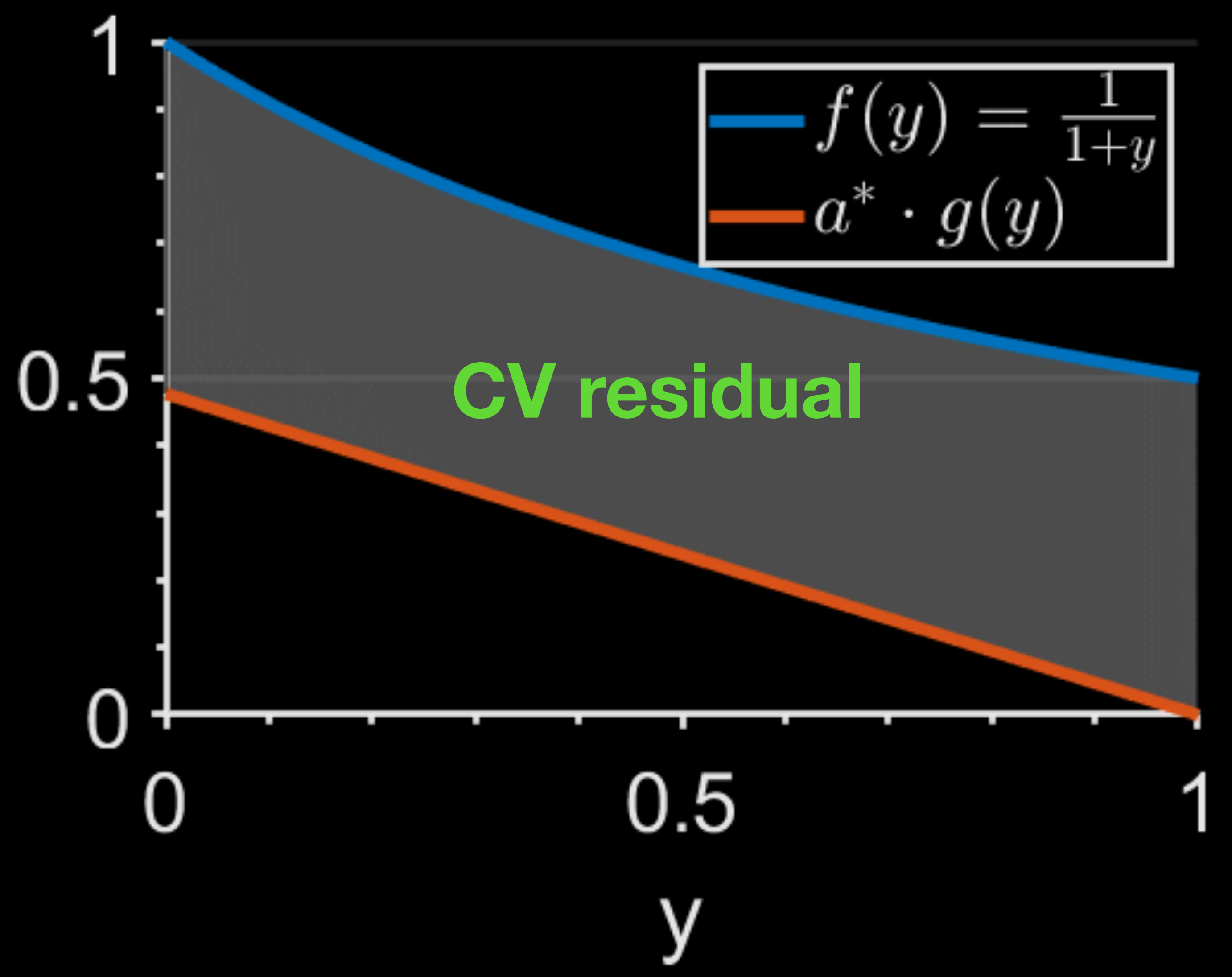
16 spp **0.6841** **0.7061**

F = ln(2) ≈ 0.6931

Standard CV







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

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

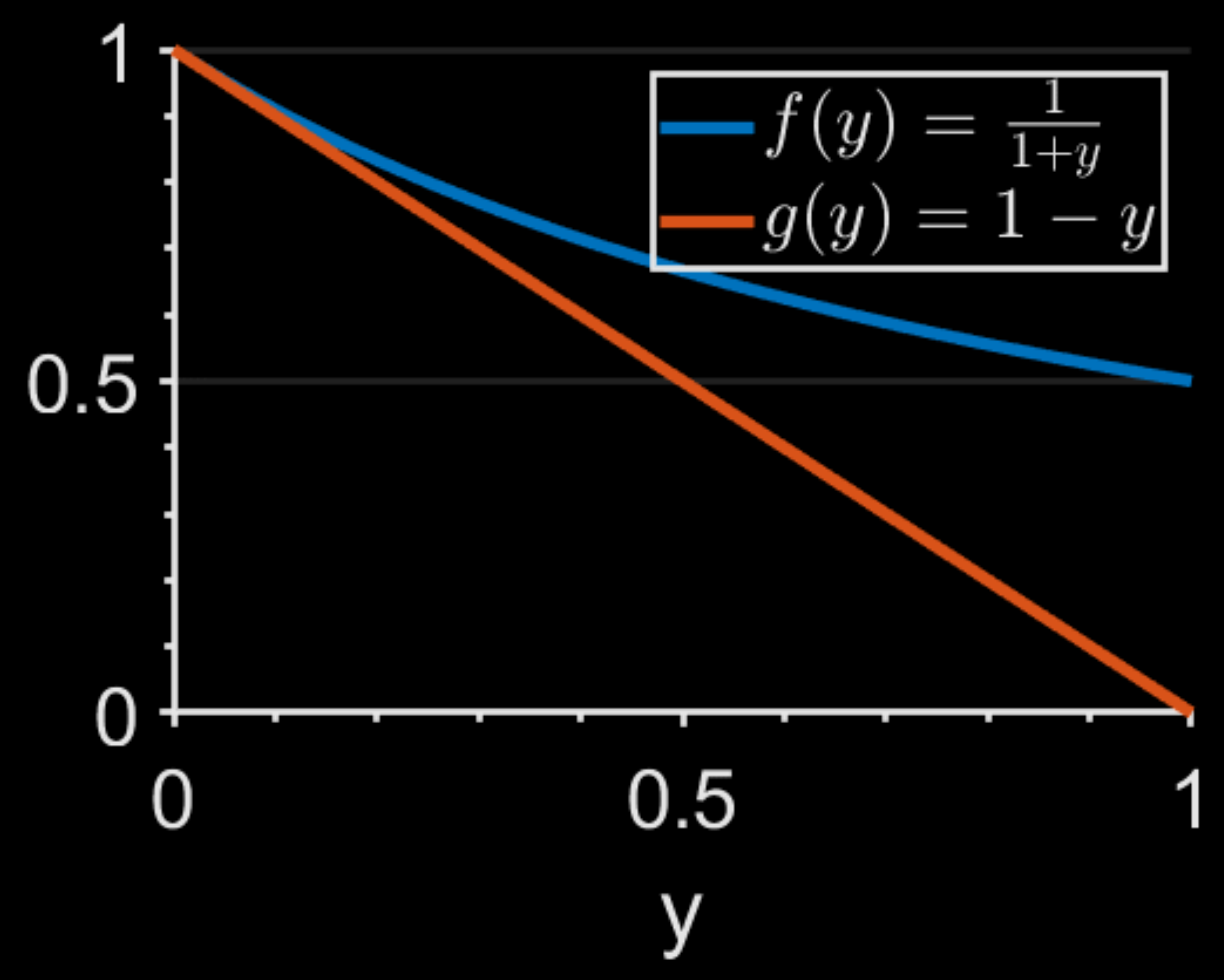
$$F = a \cdot G + \underbrace{\int_{y \in \mathcal{D}} f(y) - a \cdot g(y) dy}_{\text{CV residual}}$$

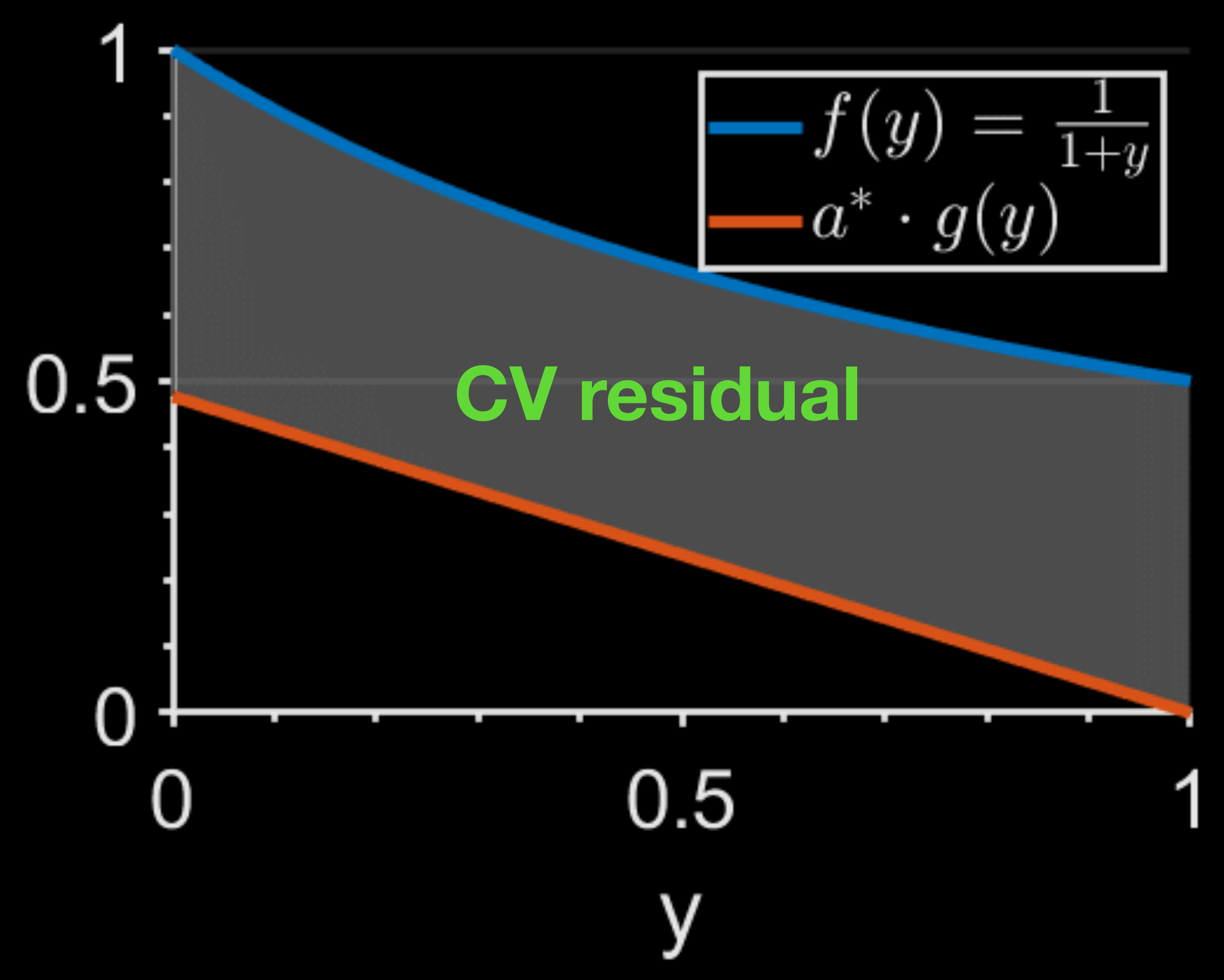
$$a^* = \frac{Cov(F, G)}{Var(G)} \approx 0.477 \quad \text{(1024 spp)}$$

16 spp **0.6841** **0.7061** $\boxed{\textbf{0.6946}}$

F = ln(2) ≈ 0.6931

Standard CV in Computer Graphics





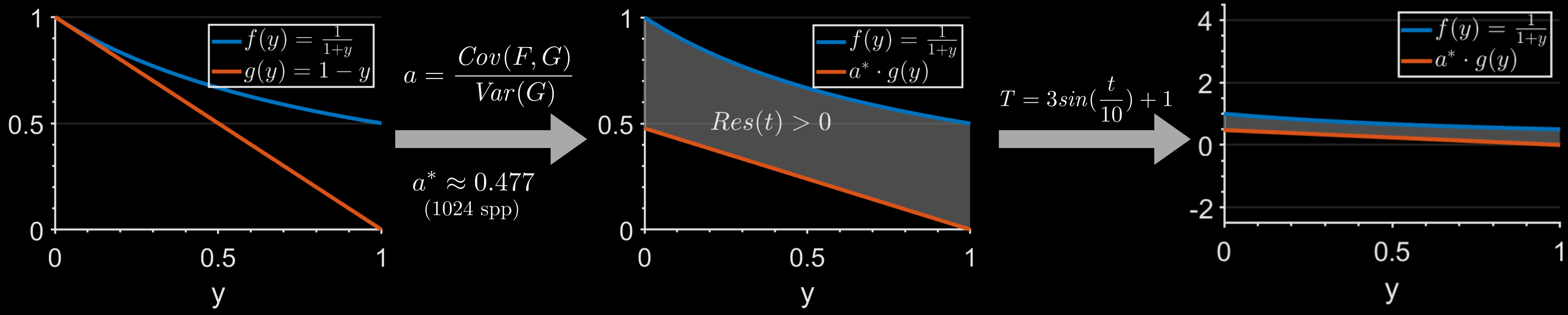
$$F = G + \int_{y \in \mathcal{D}} f(y) - g(y) dy$$

$$F = a \cdot G + \underbrace{\int_{y \in \mathcal{D}} f(y) - a \cdot g(y) dy}_{\text{CV residual}}$$

Main part separation
[Novák et al. 2014,Szécsi et al. 2004]

Penalized least squares [Fan et al. 2006],
Iterative estimator [Rousselle et al. 2016; Kondapaneni et al. 2019],
and deep learning [Müller et al. 2020]

In-frame Standard CV

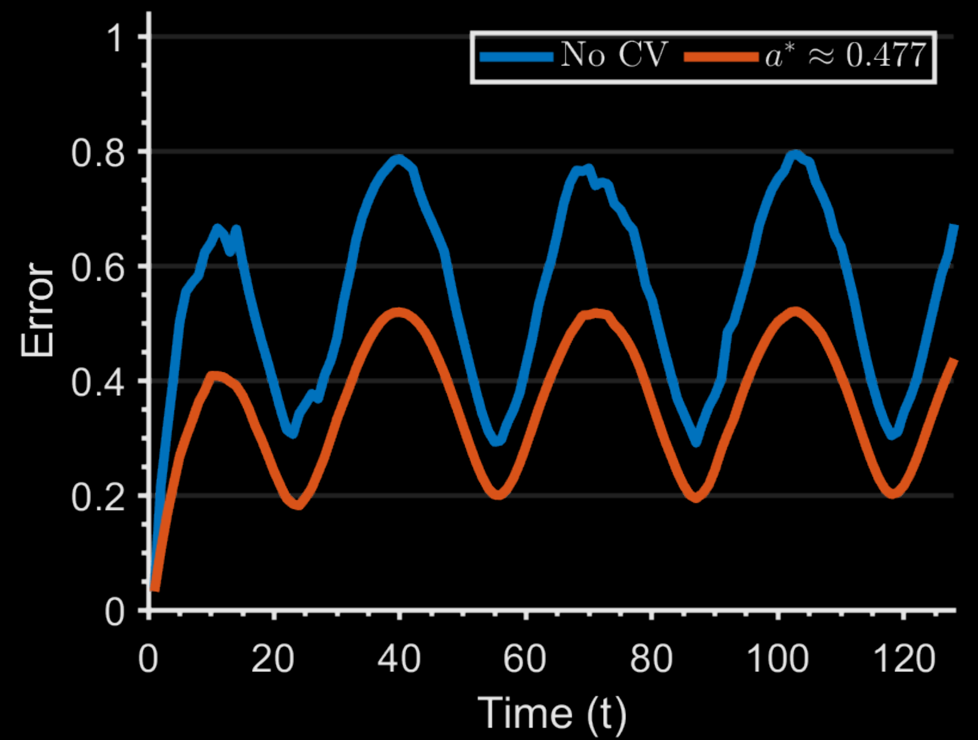


$a = \frac{Cov(F,G)}{Var(G)}$

$a^* \approx 0.477$
(1024 spp)

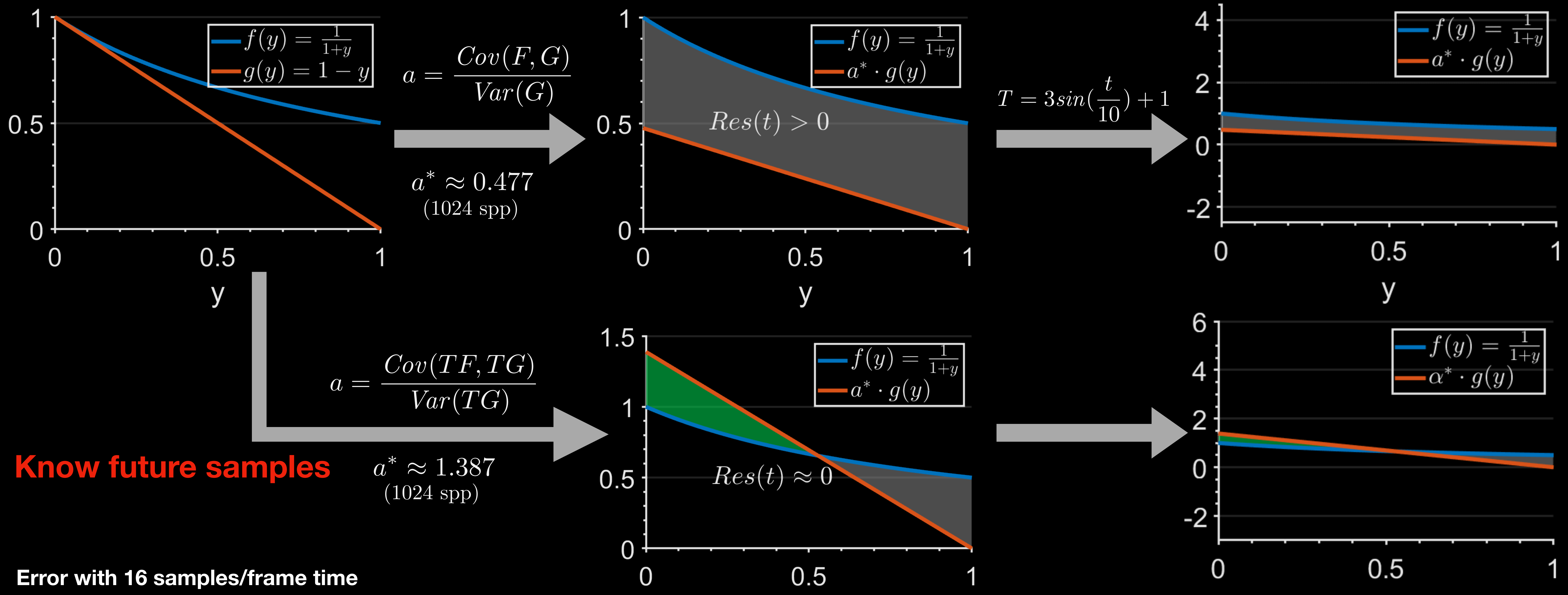
$T = 3sin(\frac{t}{10}) + 1$

Error with 16 samples/frame time



Error: The square root of exponential moving variance

In-frame Standard CV



$$a = \frac{Cov(F,G)}{Var(G)}$$

$$a^* \approx 0.477$$
(1024 spp)

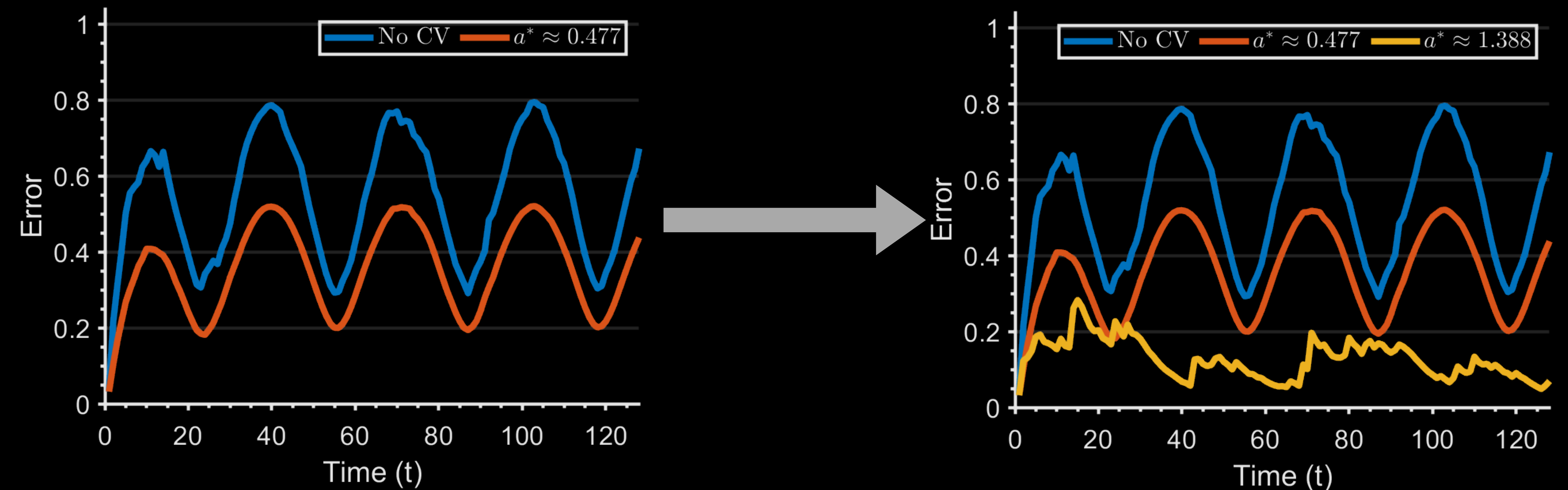
$$T = 3sin(\frac{t}{10}) + 1$$

$$a = \frac{Cov(TF,TG)}{Var(TG)}$$

Know future samples

$$a^* \approx 1.387$$
(1024 spp)

Error with 16 samples/frame time



Issue from sampling G:

- **Bandwidth**
- **Sampling error**

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

$$a = \frac{E[F]}{E[G]} = \frac{E[TF]}{E[TG]}$$

G is a constant control variable

$$Var(G) = 0$$

(2) No independence assumption

Exponential Moving Covariance Matrix (EMCM):

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

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

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

$\mathbf{\Sigma}_t$ **Covariance of TF and TG.**

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

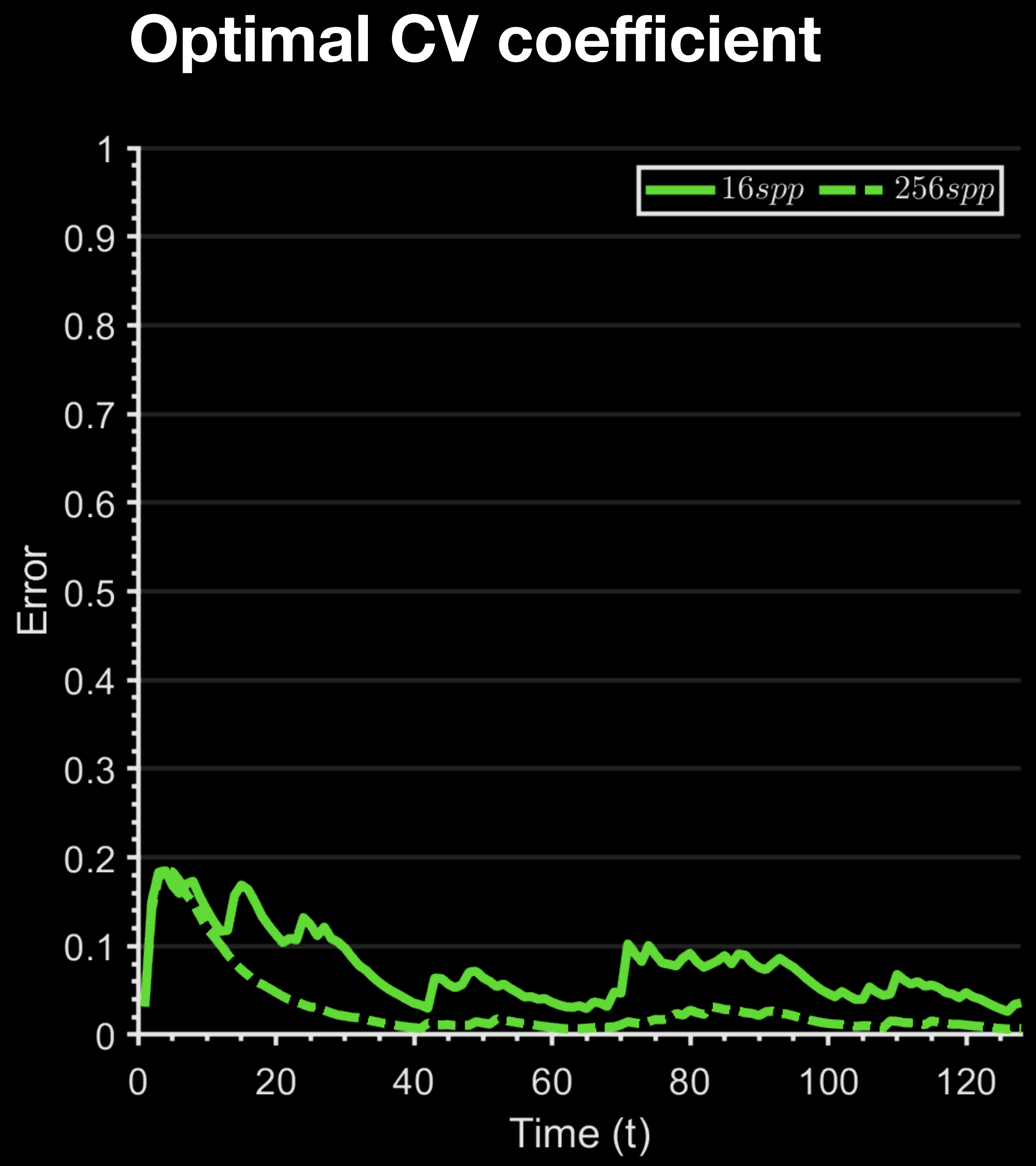
CV Coefficient estimation:

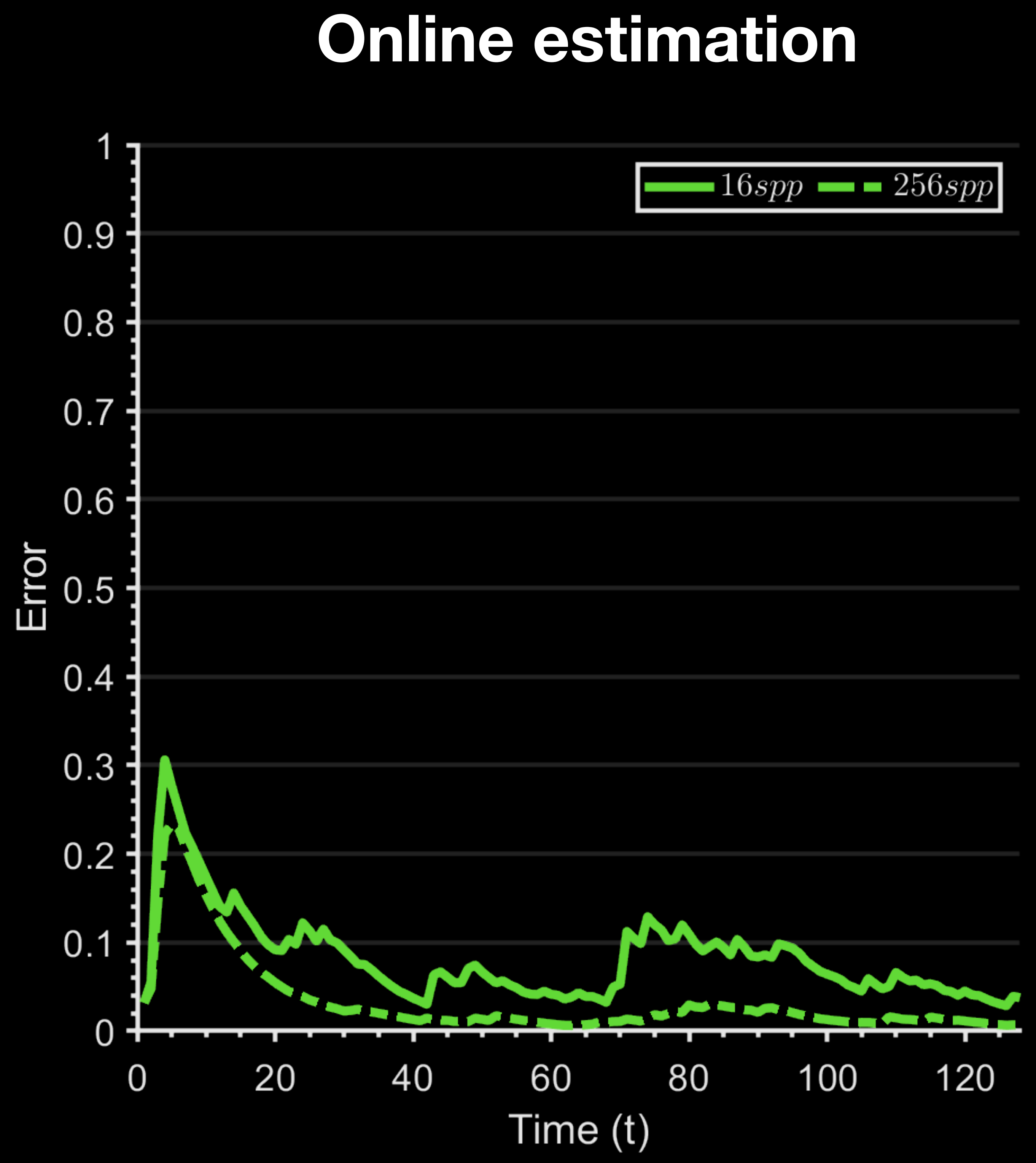
$$a_t = \frac{\mathbf{\Sigma}_t.xy}{\mathbf{\Sigma}_t.yy}$$

Online CV Estimation

Track variance of CV residual:

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


Optimal CV coefficient


Online estimation

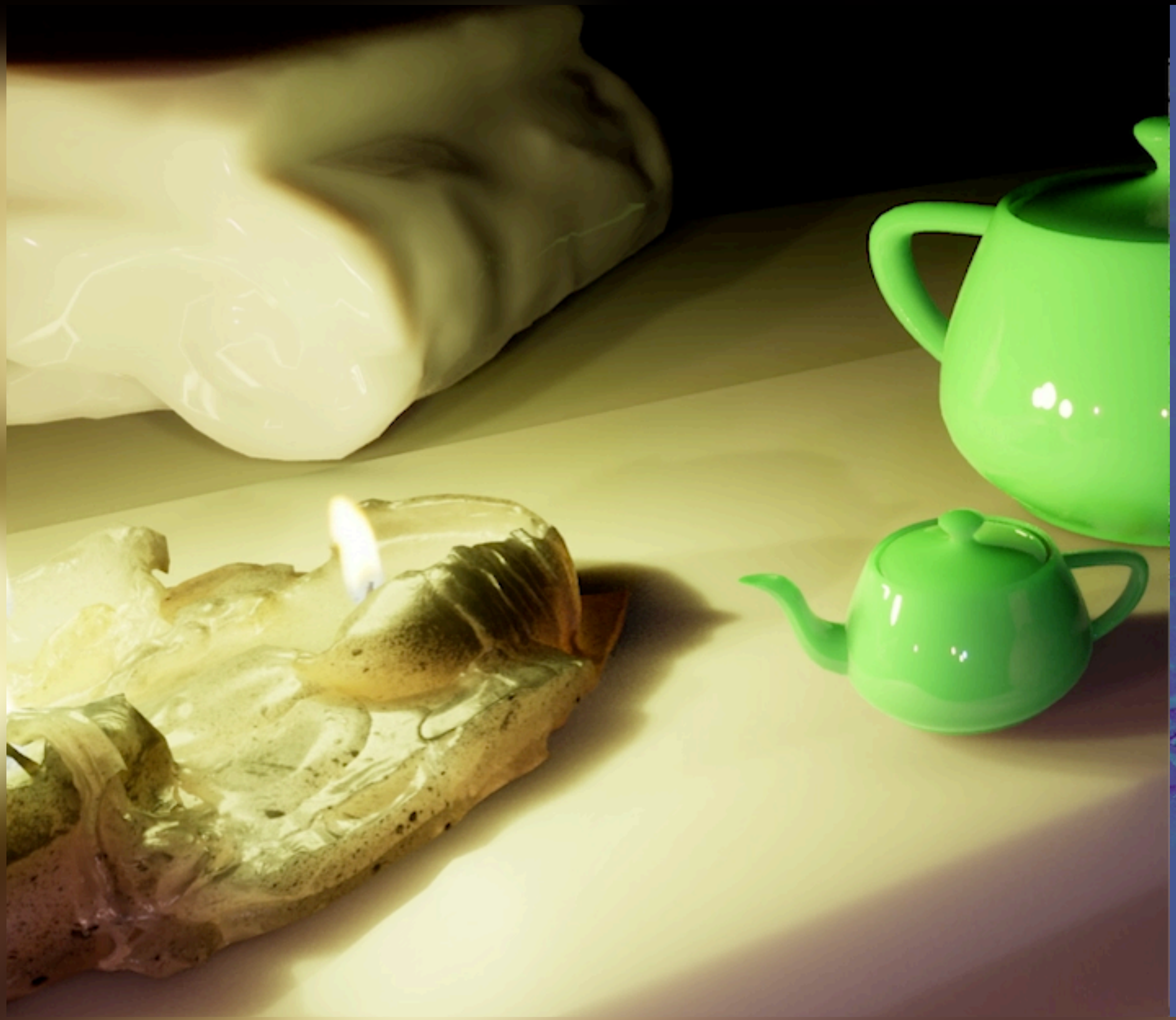
Temporally Stable Adaptive Sampling

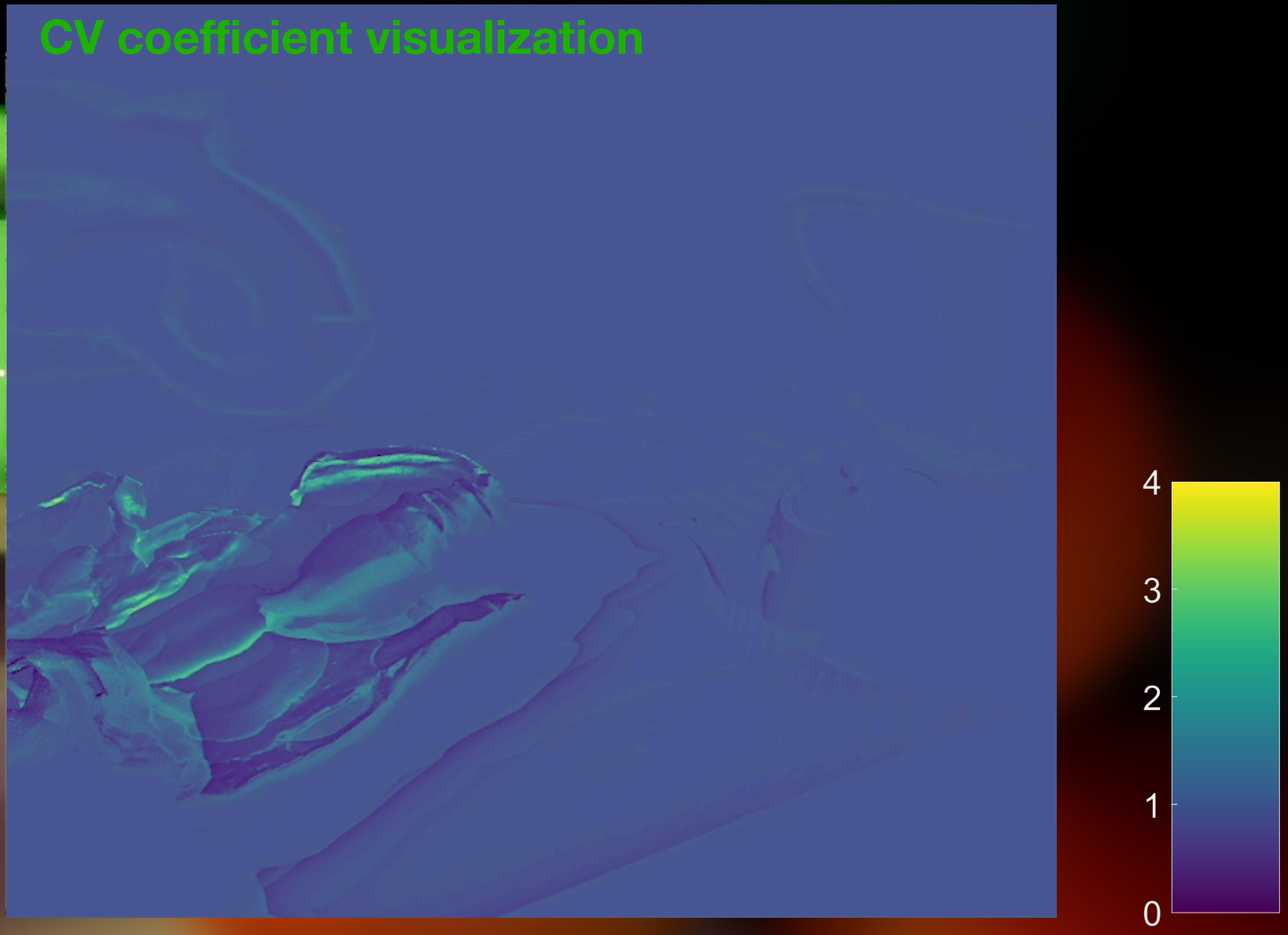
- Reduce temporal variance with Control Variates

- Application to real-time subsurface scattering

Application to Subsurface Scattering

Online CV coefficient for subsurface scattering





For large flat lighting region:

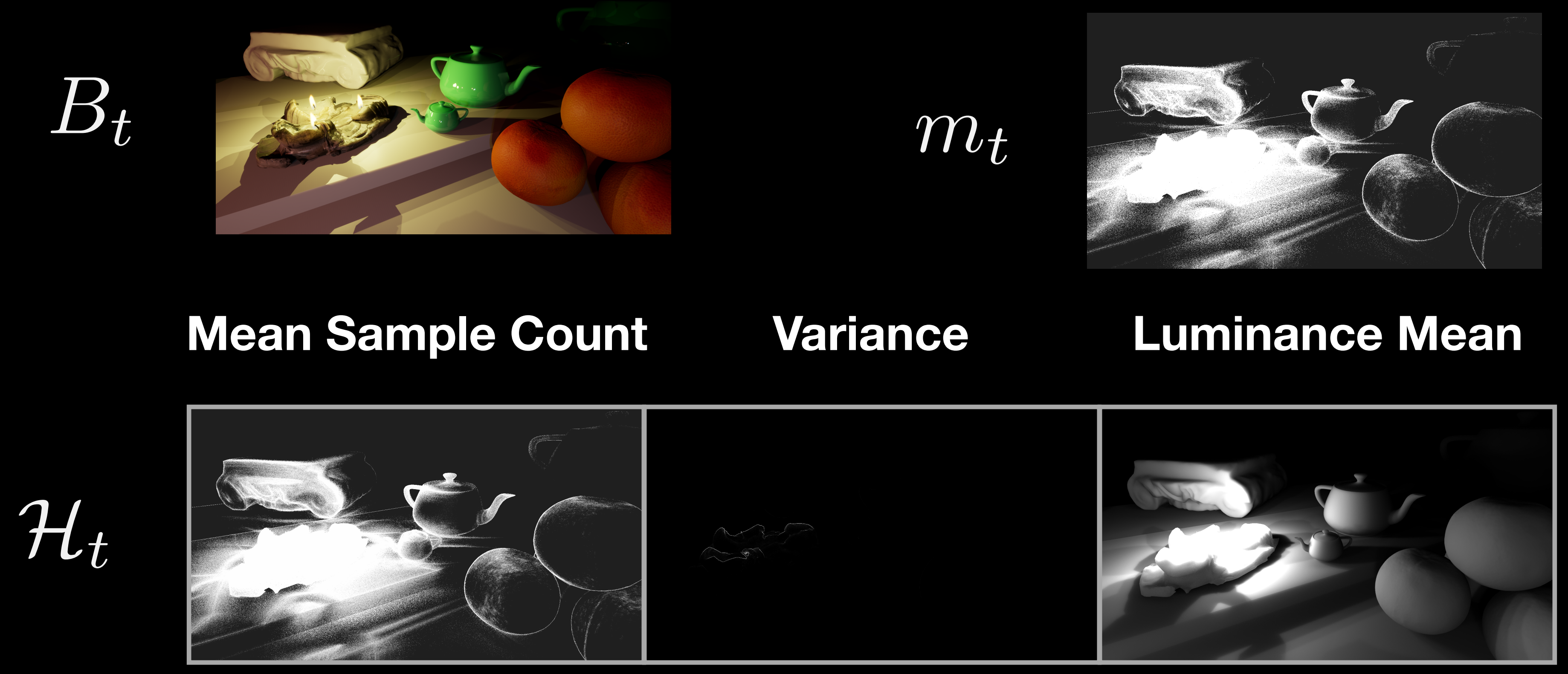
$$a_t^* = \frac{E[L_s(t)]}{G(t)} = 1$$

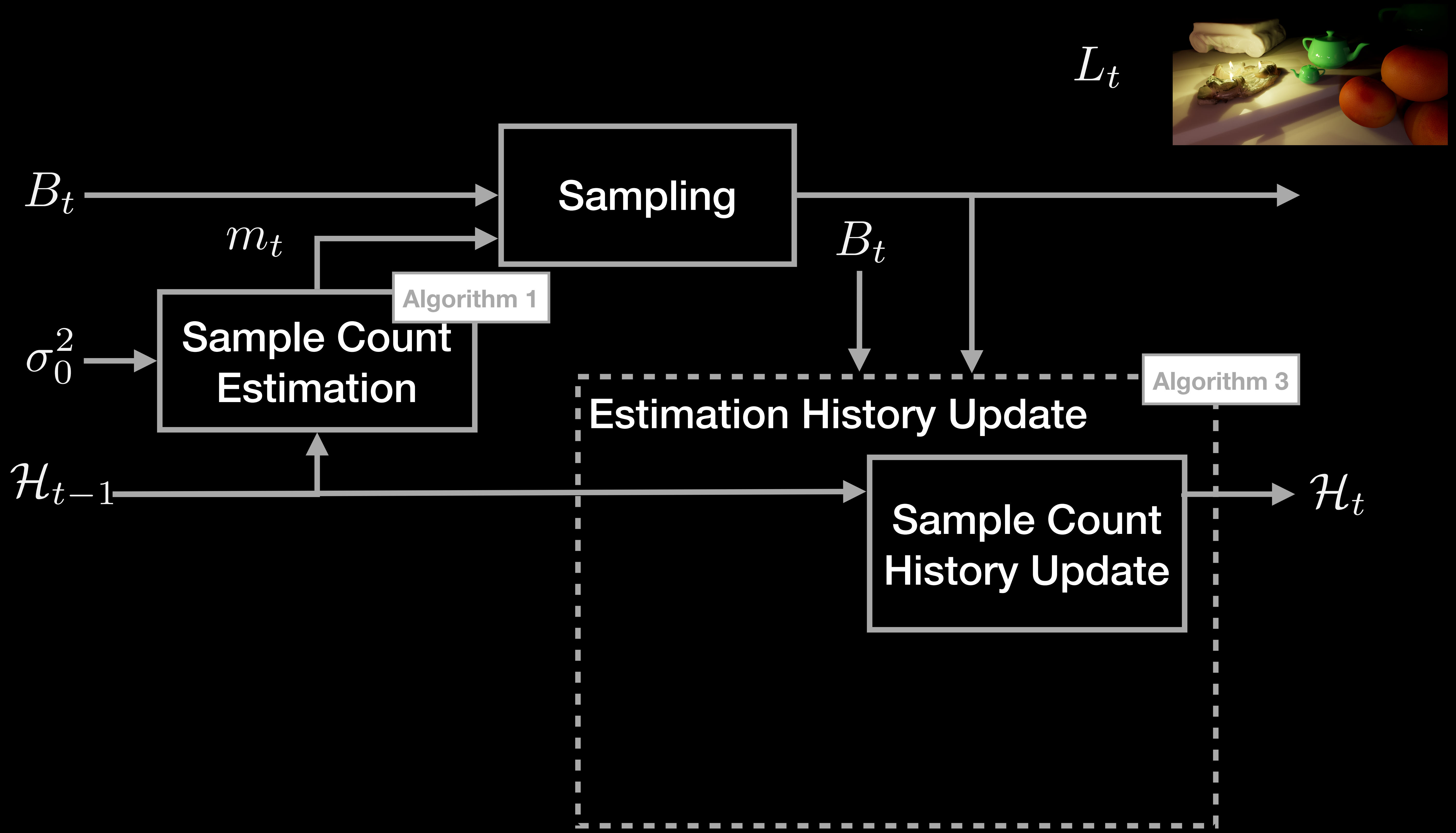
Temporally Stable Adaptive Sampling

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

Temporally Stable Adaptive Sampling Algorithm Overview

Real-time Adaptive Sampling



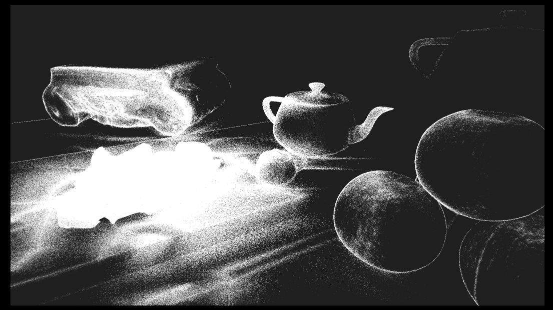


Temporally Stable Adaptive Sampling Algorithm Overview

Real-time Adaptive Sampling

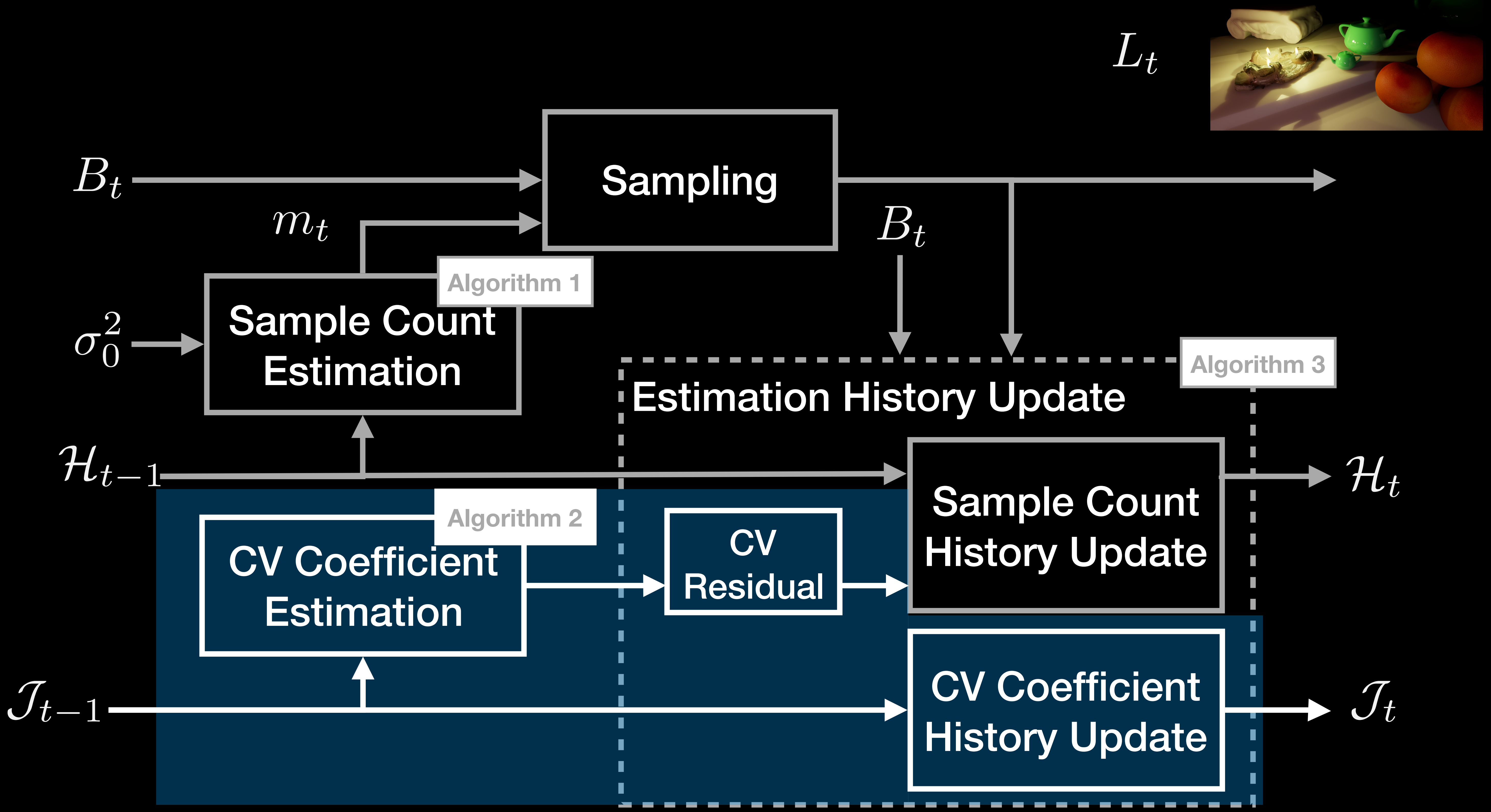
B_t

m_t



Mean Sample Count | Variance | Luminance Mean

$\mathcal{H}_t$

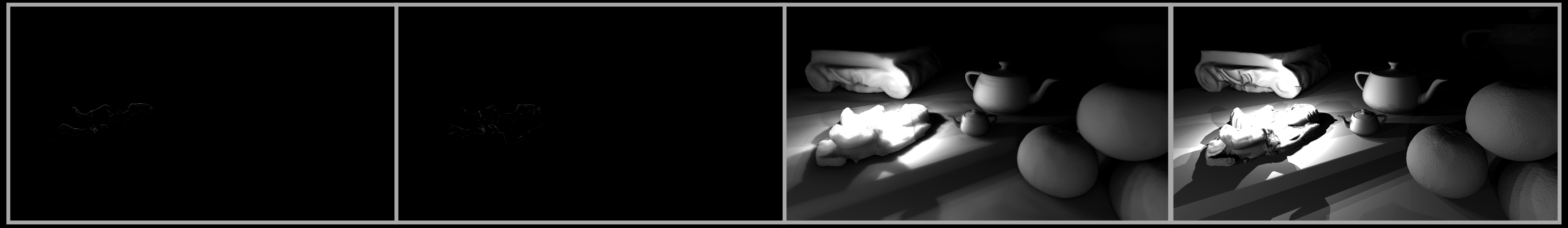


Real-time Control Variates

a_t^*

Covariance | Cv Variance | Mean Luminance | Mean Cv

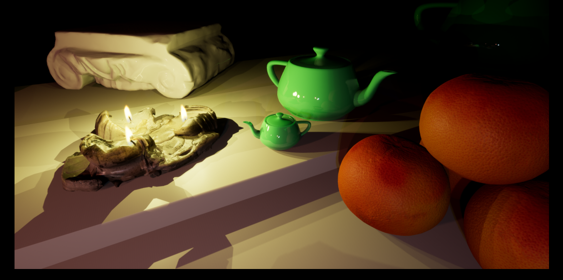
$\mathcal{J}_t$



Temporally Stable Adaptive Sampling Algorithm Overview

Real-time Adaptive Sampling

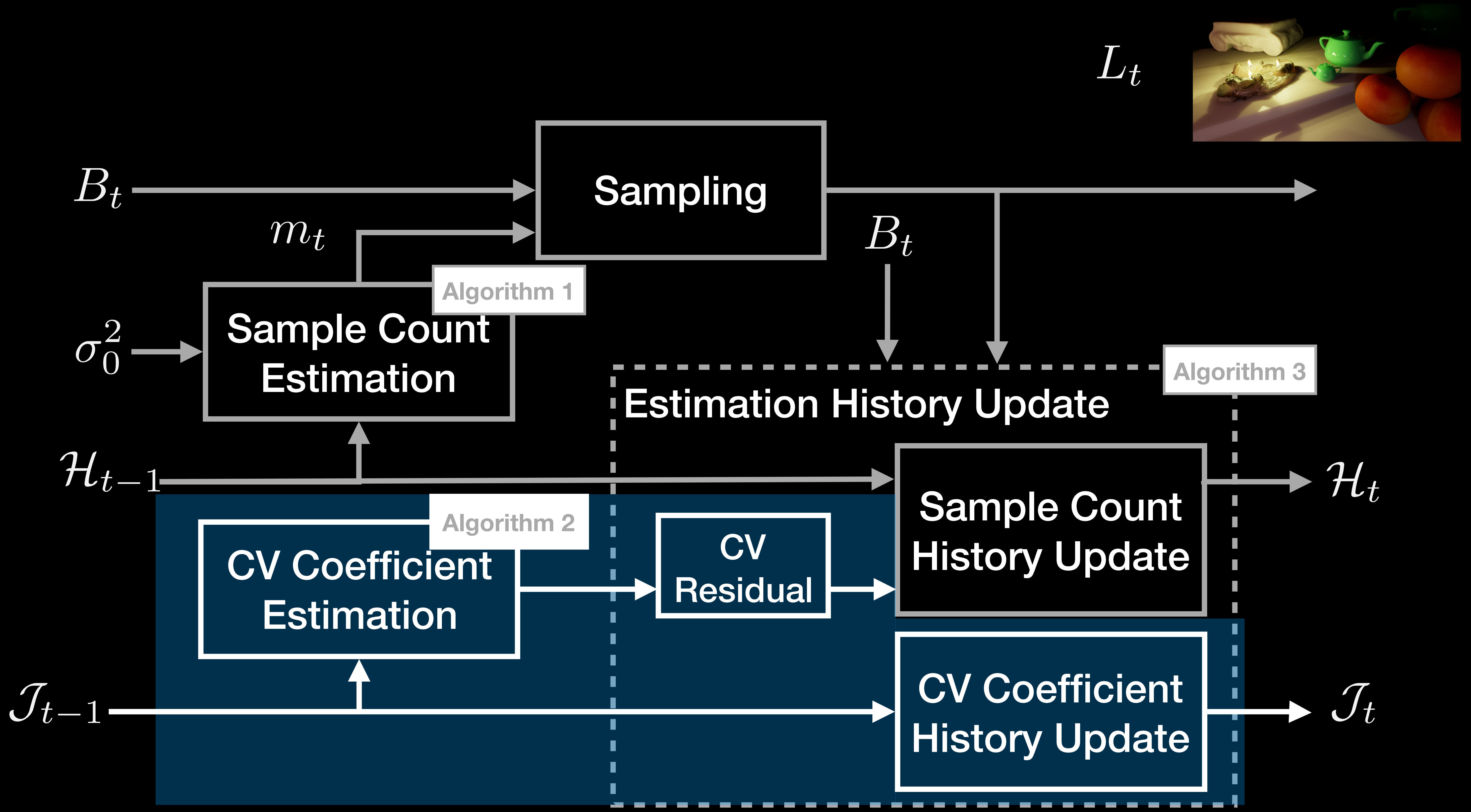
B_t



m_t

$\mathcal{H}_t$

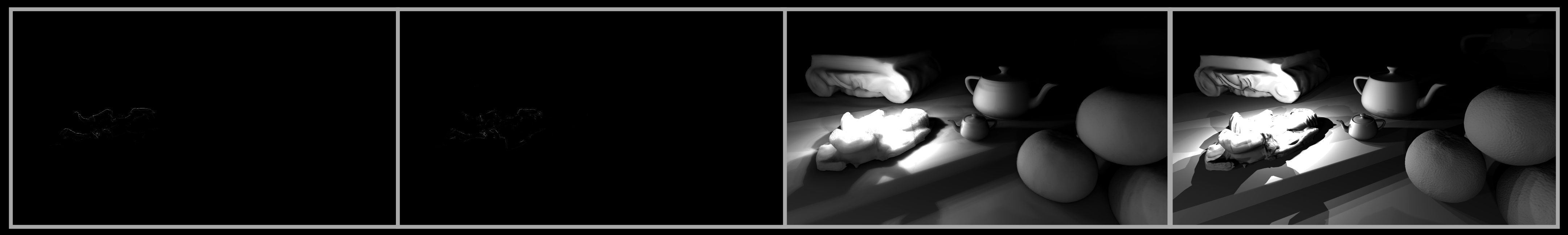
Mean Sample Count
Variance
Luminance Mean



Real-time Control Variates

Online CV a_t^*

$\mathcal{J}_t$



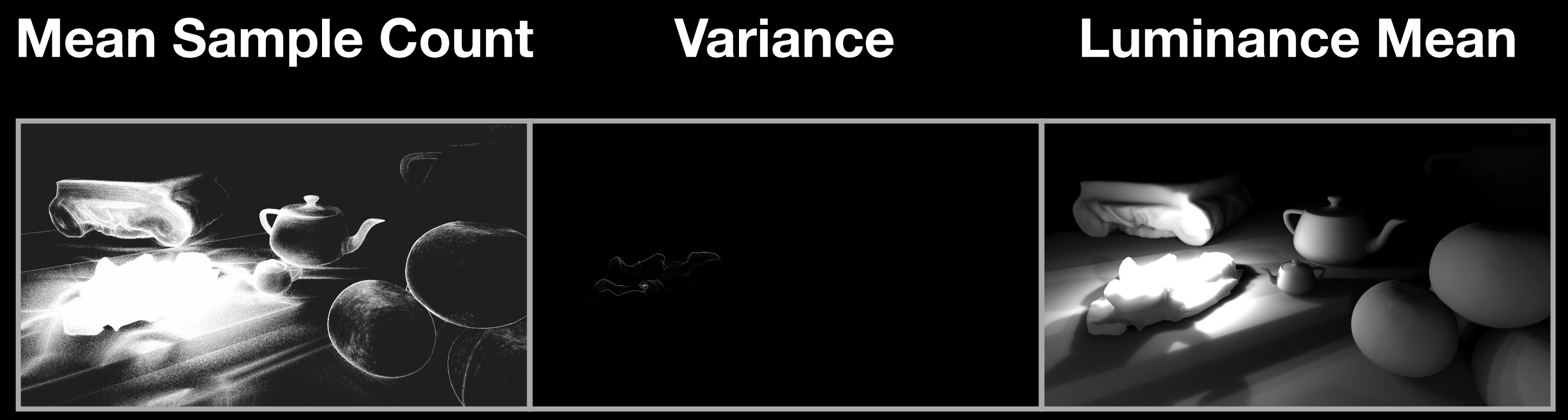
Temporally Stable Adaptive Sampling Algorithm Overview

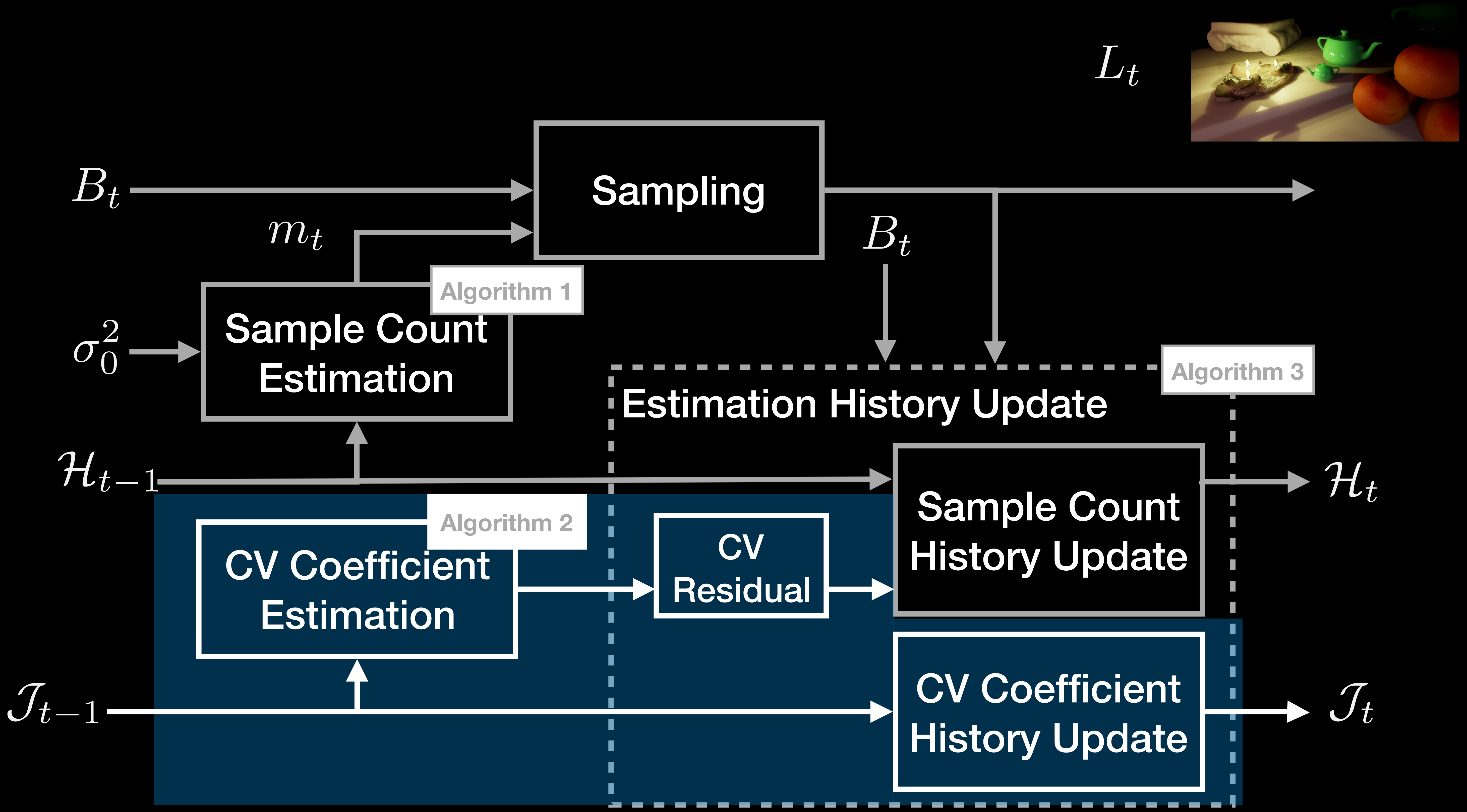
Real-time Adaptive Sampling

B_t

m_t

$\mathcal{H}_t$

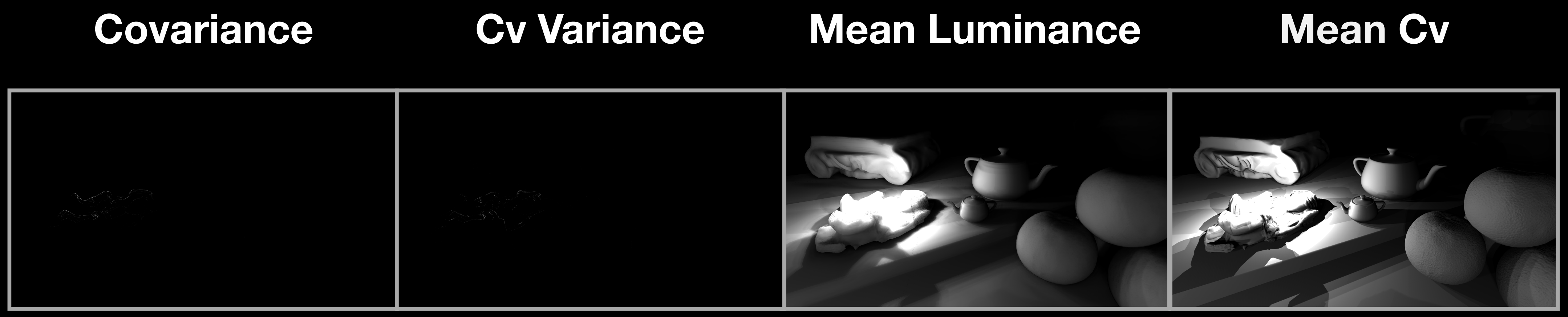




Real-time Control Variates

Online CV a_t^*

$\mathcal{J}_t$

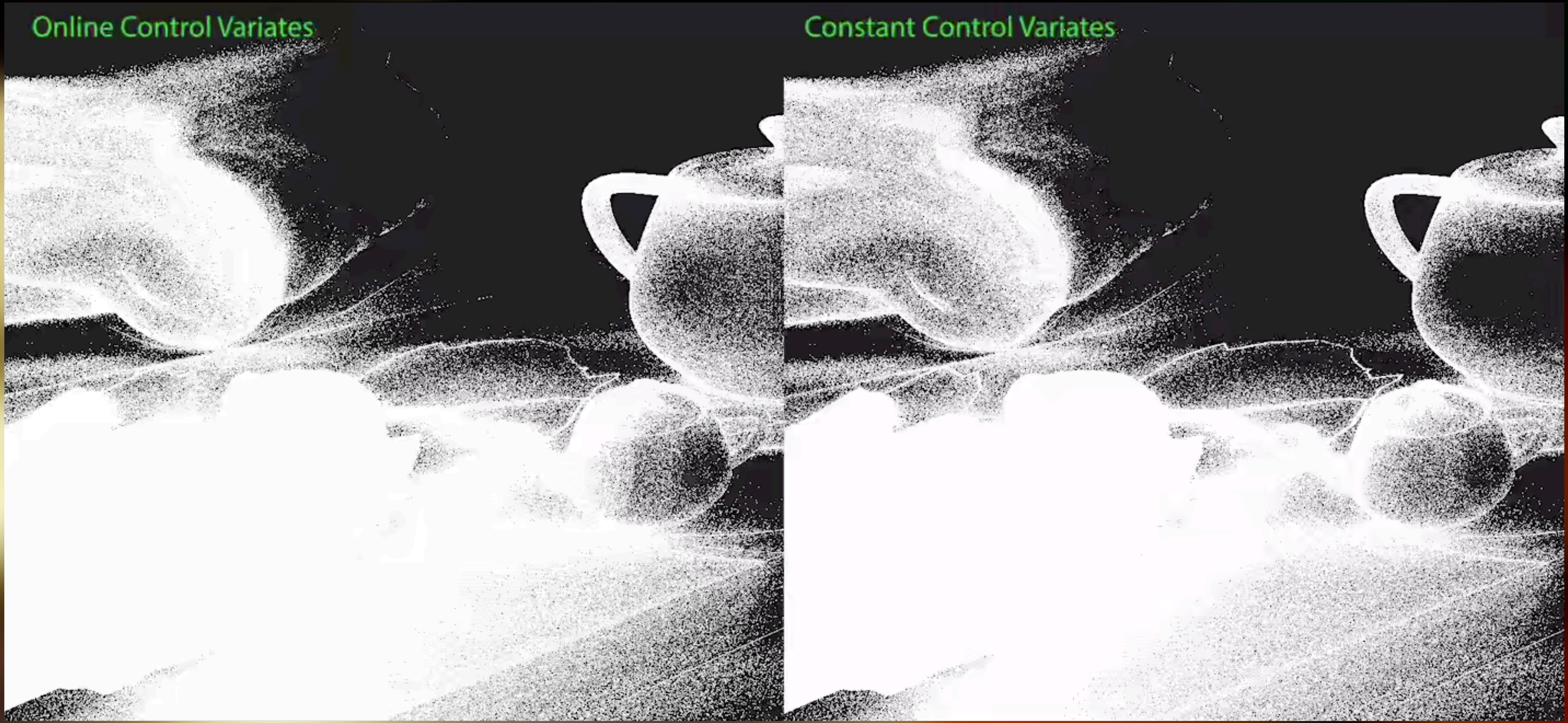


Constant CV $a_t^* = \dfrac{E[L_s(t)]}{G(t)} = 1$

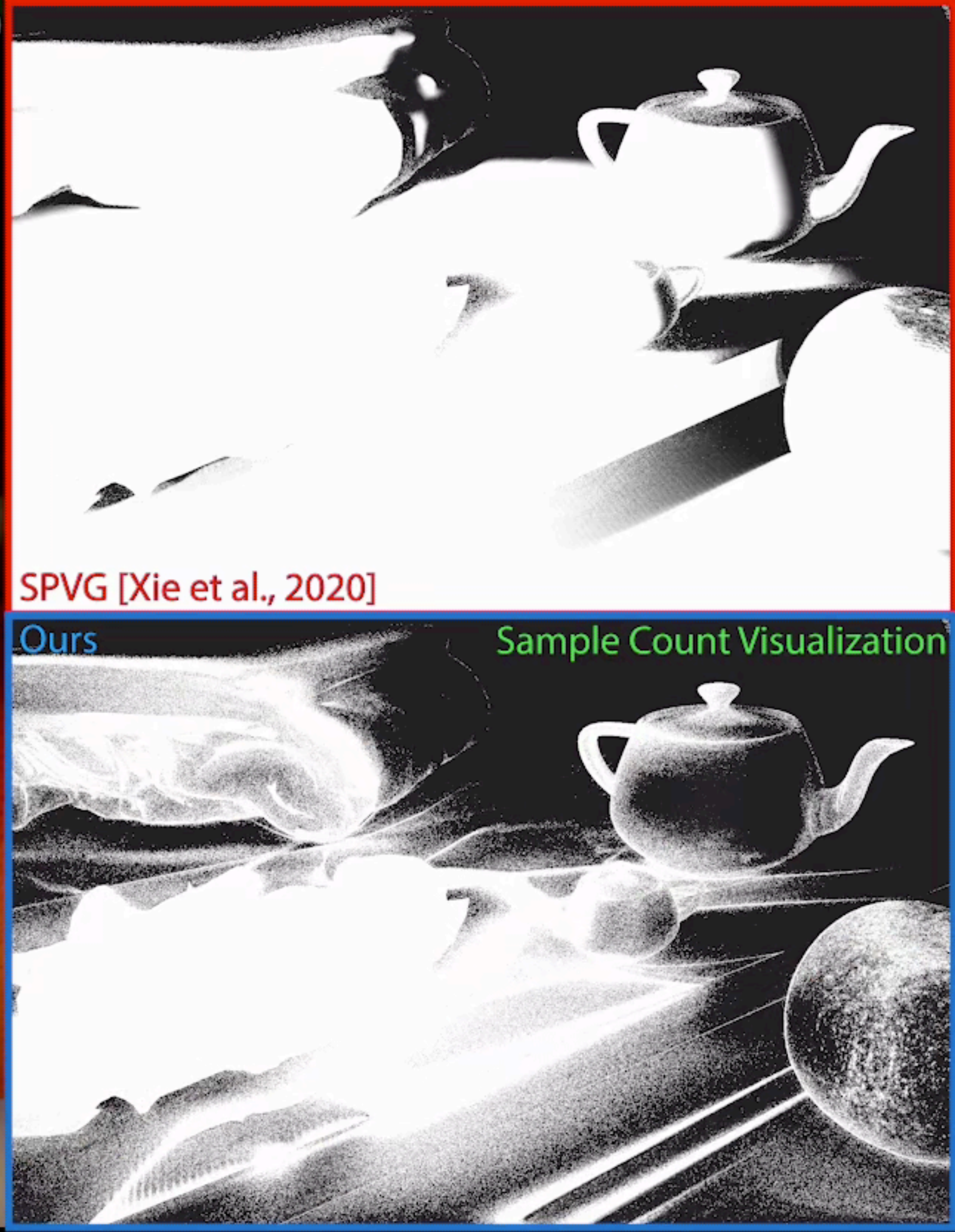
No more textures!

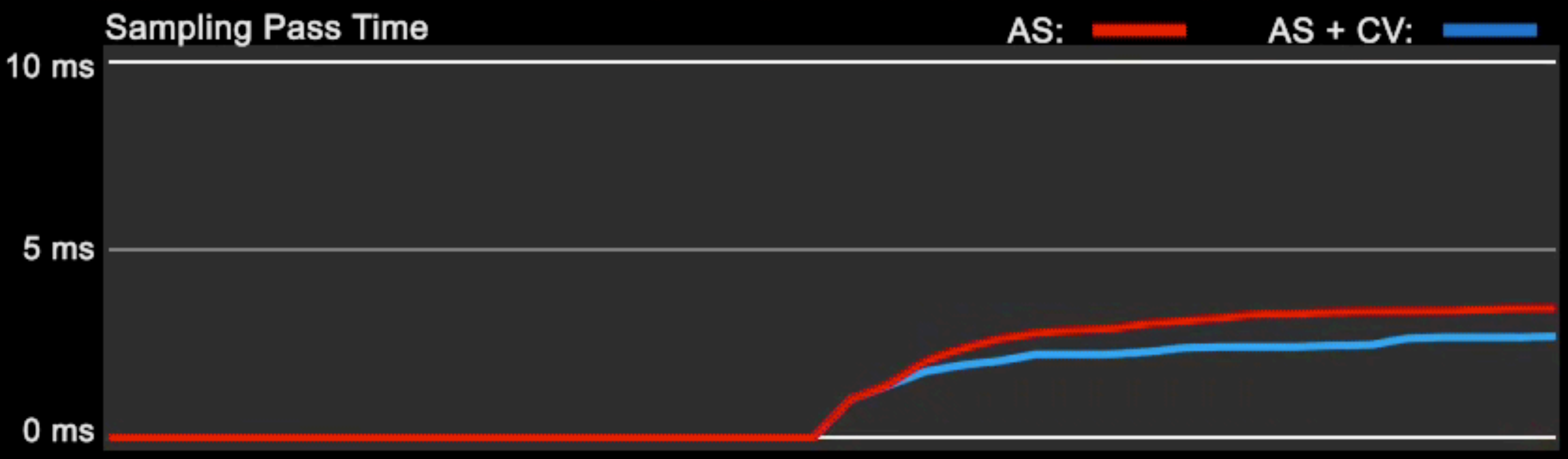
Online vs Constant Control Variates

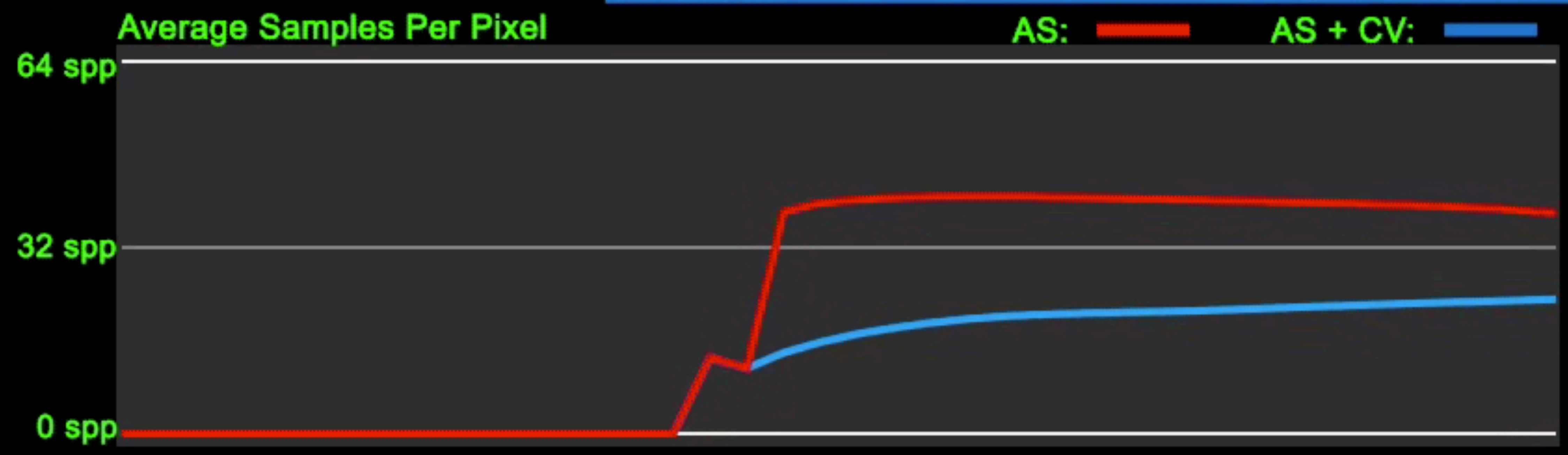
Estimated Sample Count:



Regular Lighting



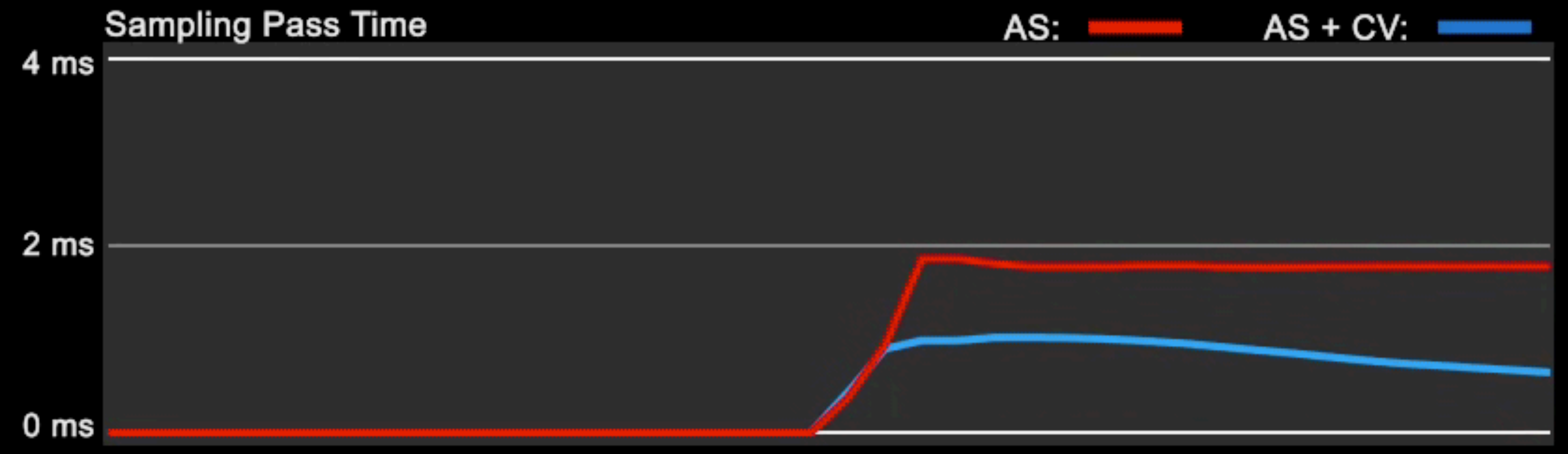


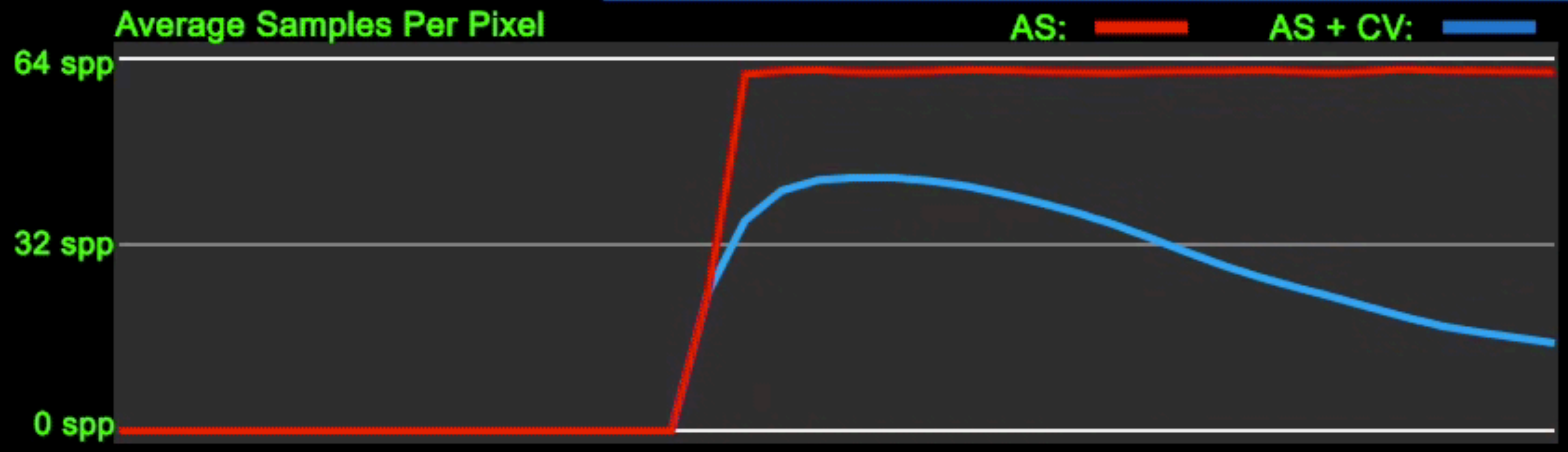


Light Flash

2560x1440
SPVG [Xie et al., 2020]
Ours
Sample Count Visualization

Lt. Belica, Paragon Character

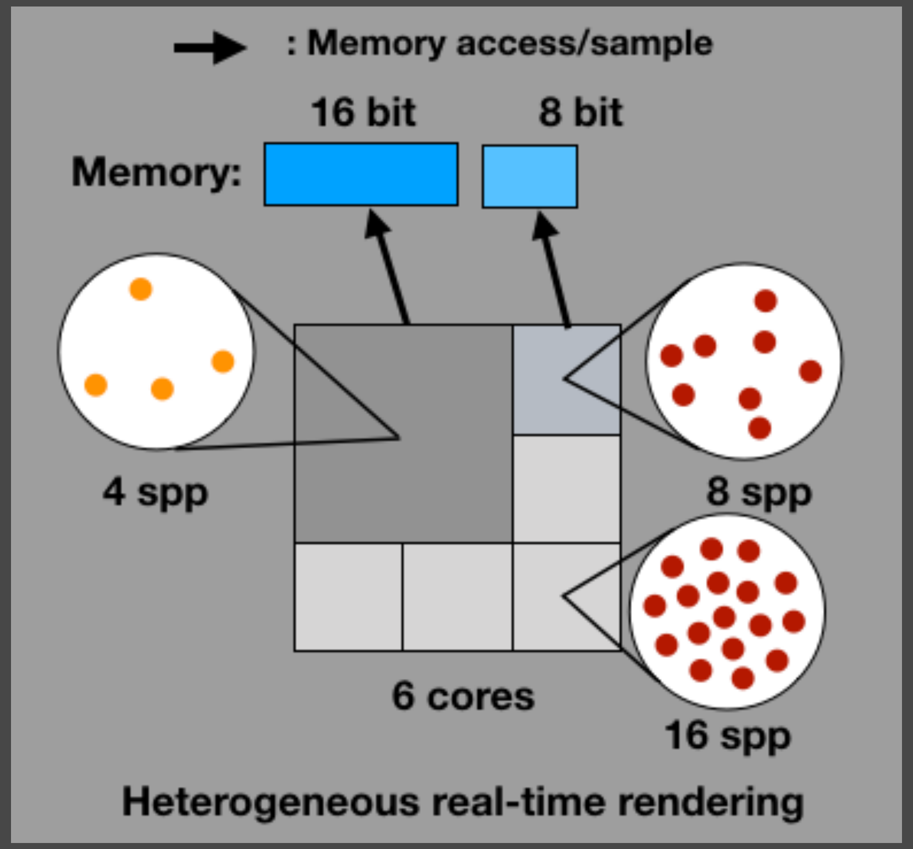




Outline

Section I: Chapter 1 ~ Chapter 3

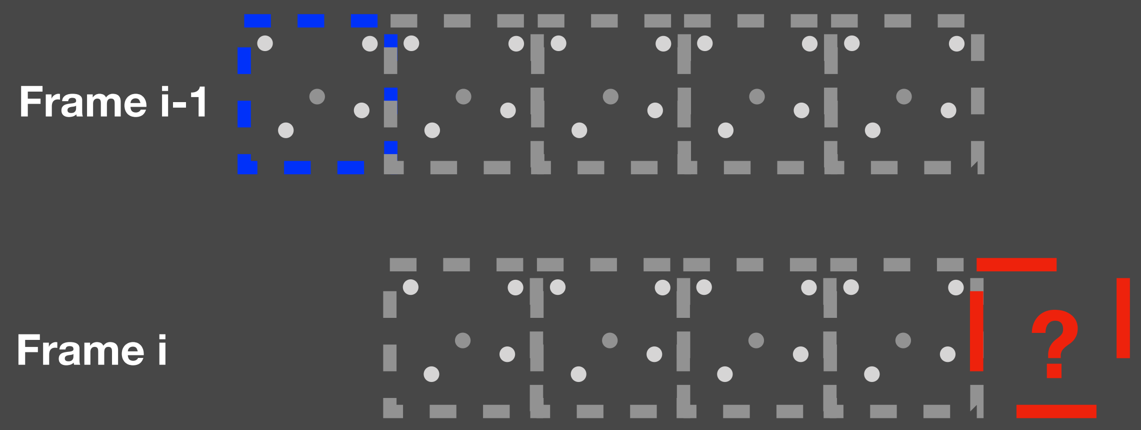
- Introduction
- Literature
- Motivation
- Heterogeneous Real-time Rendering



Heterogeneous real-time rendering

Section II: Chapter 4 (I3D'20)

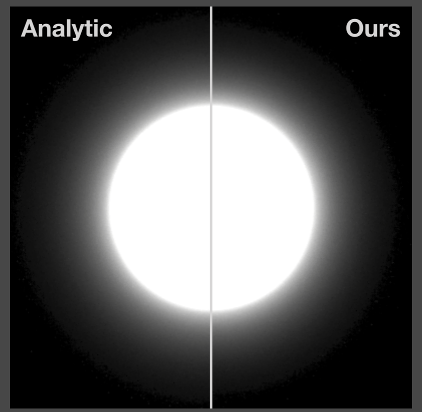
- Real-time Adaptive Sampling $O(1)$



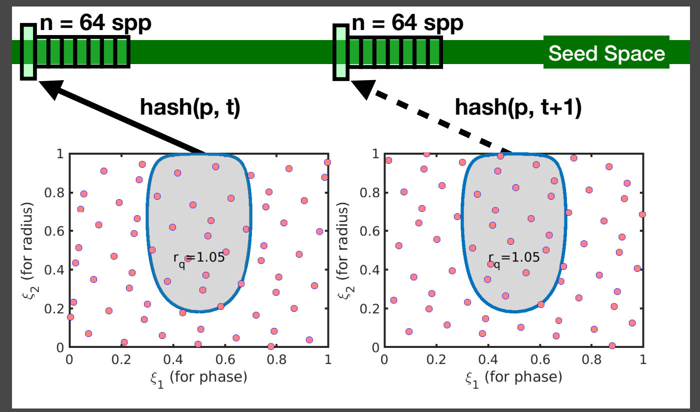
Section III: Chapter 5

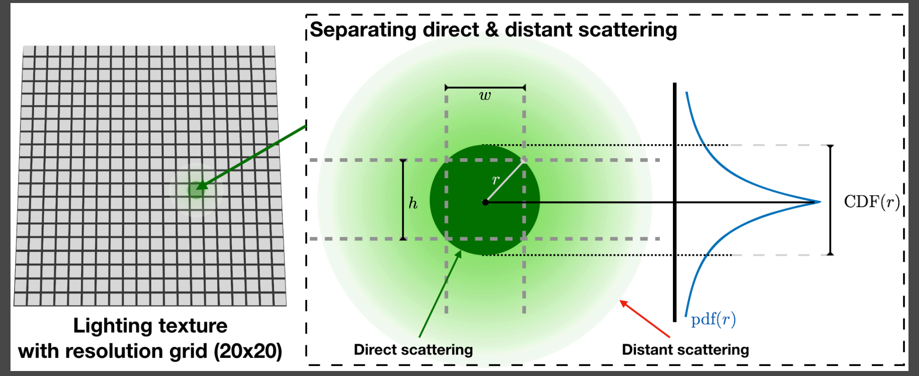
- Subsurface Scattering

Sampling function approximation

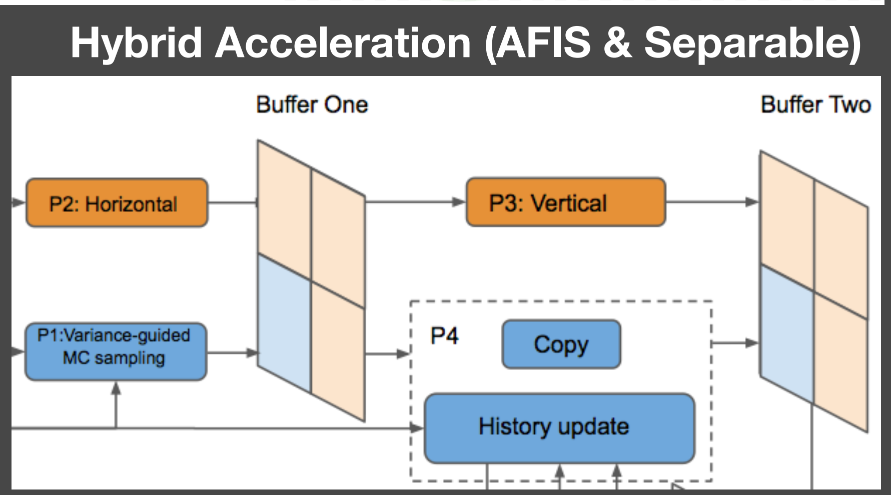


Sampling sequence





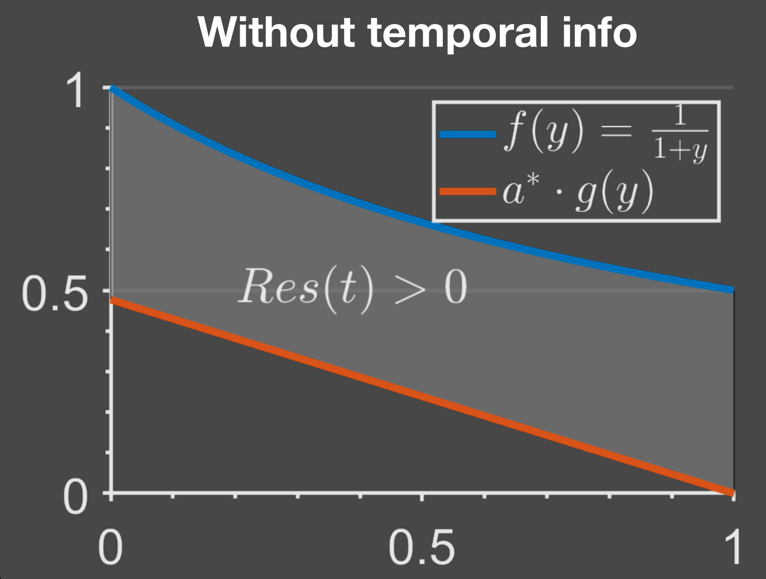
Hybrid Acceleration (AFIS & Separable)

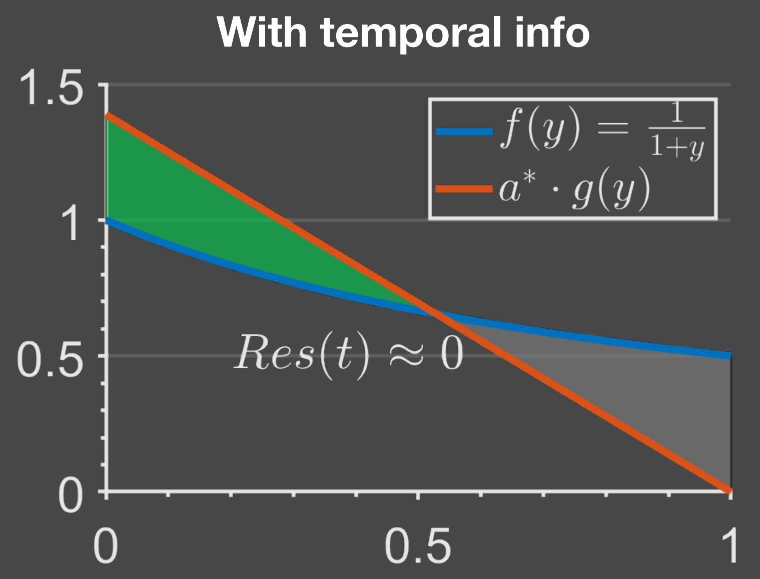


Section IV: Chapter 6 (I3D'21)

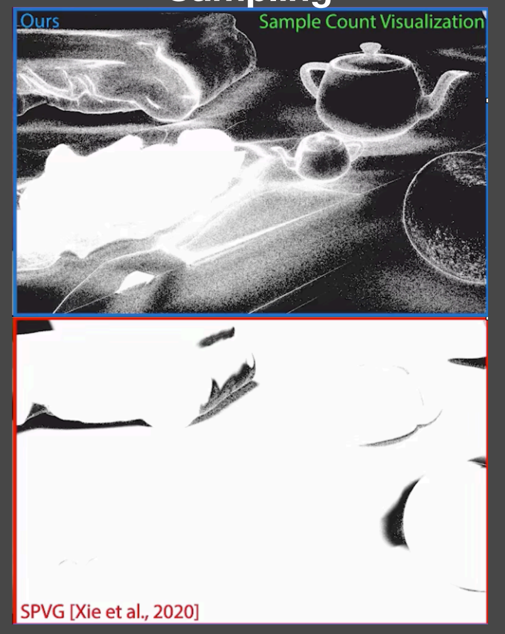
- Real-time Control Variates

with one to even zero additional texture





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!

UMBC

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.

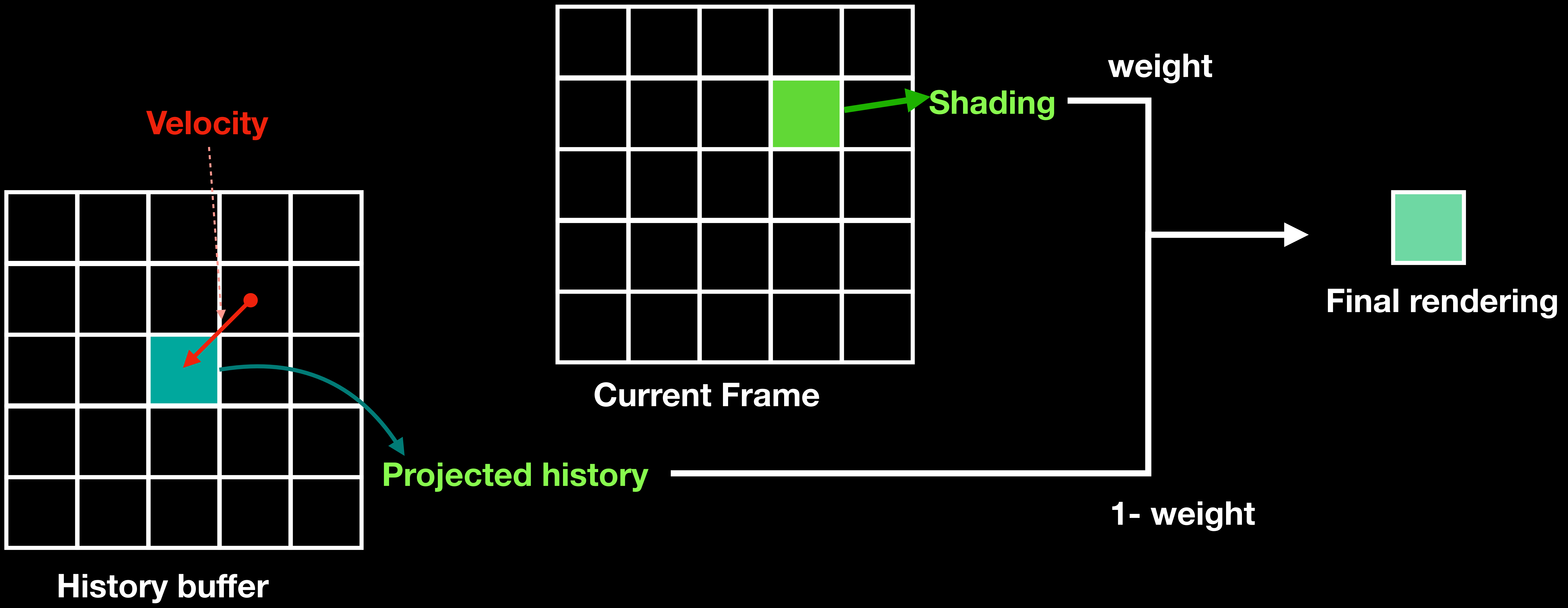
9. Chen, R., **Xie, T.**, Xie, Y., Lin, T., & Tang, N. (2016, October). Do speech features for detecting cognitive load depend on specific languages?. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (pp. 76-83).

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.

BACKUP SLIDES SINCE THIS PAGE

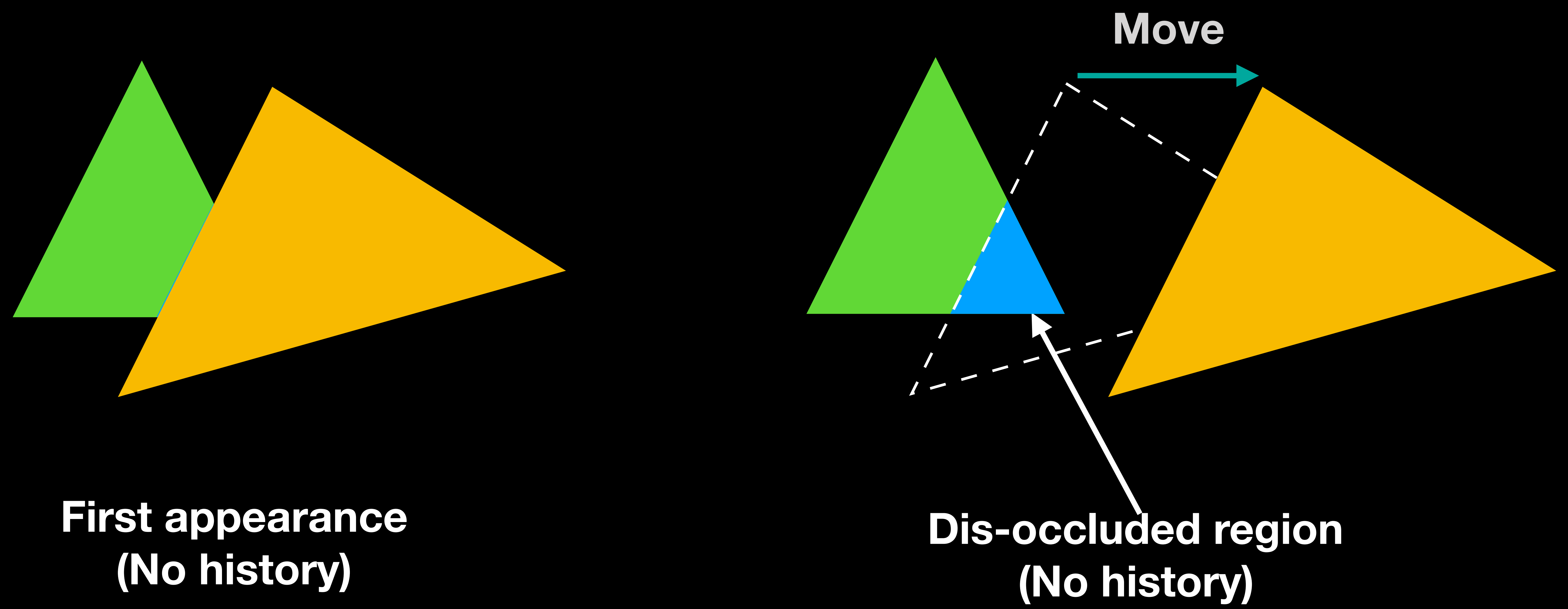
Temporal accumulation

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



Disocclusion

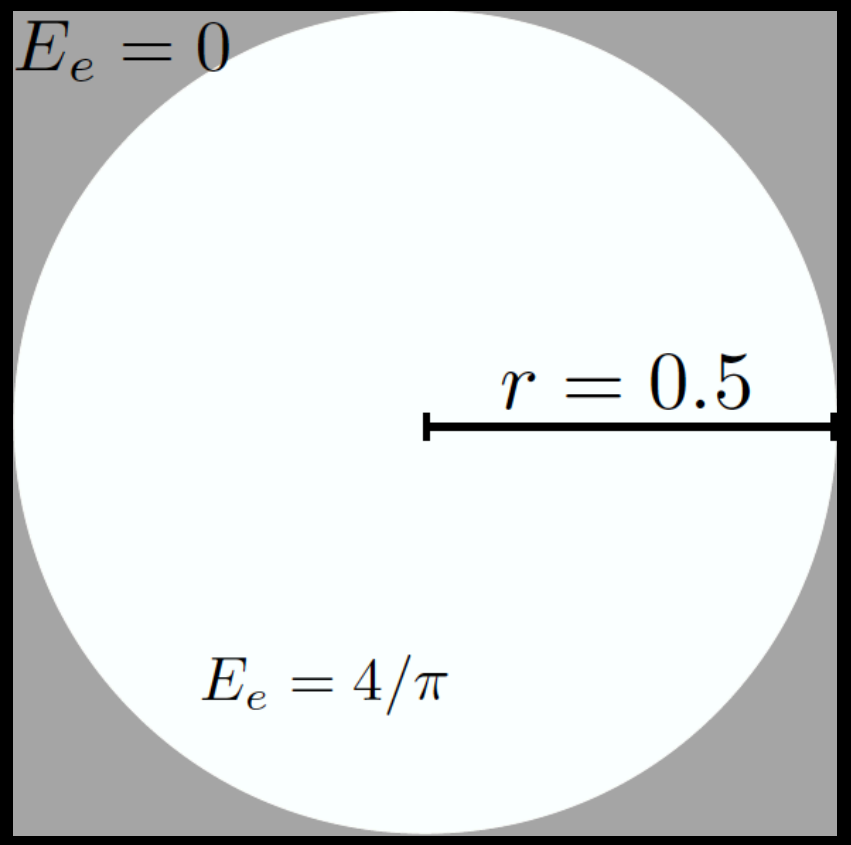
Disocclusion. How to sample efficiently without history.

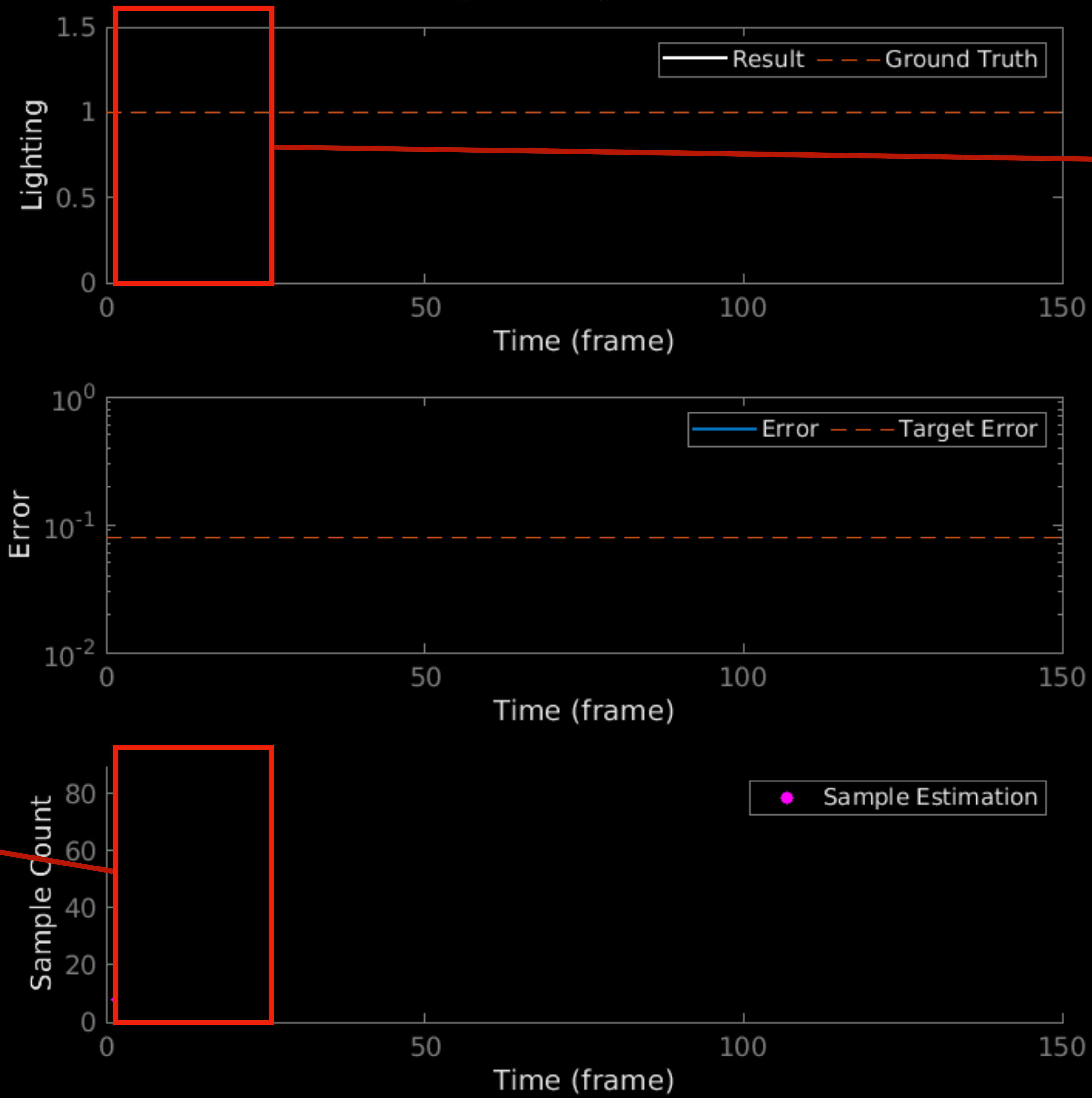


Disocclusion

Circle scenario: Sample center point lighting

Irradiance texture





Lighting converges very slow

Overestimation of sample count

Disocclusion

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

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

$$\hat{\sigma}_i^2(\alpha_0, \Lambda) = \begin{cases} \sigma_0^2 & C_s(x_i, \Lambda) = 0 \\ \sigma_i^2 & \text{otherwise} \end{cases}$$

no history

target variance:

$$\sigma_0^2 = 0.08^2$$

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

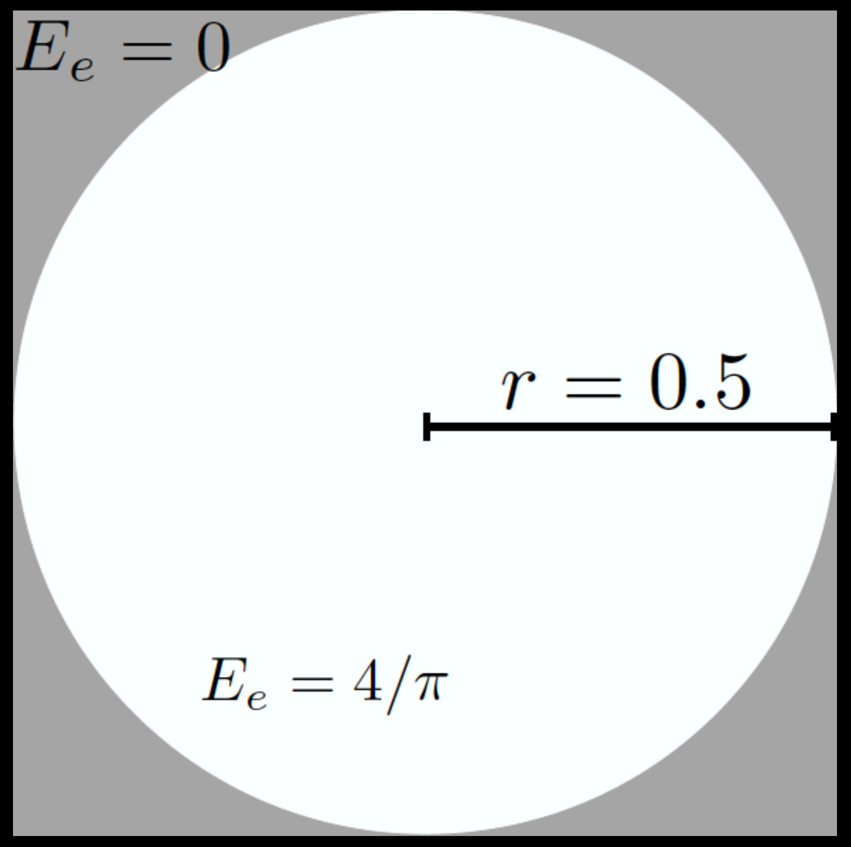
$$\mathcal{M}'(\alpha_0, \Lambda) = \begin{cases} 1 & C_s(x_i, \Lambda) = 0 \\ \alpha_0 & \text{otherwise} \end{cases}$$

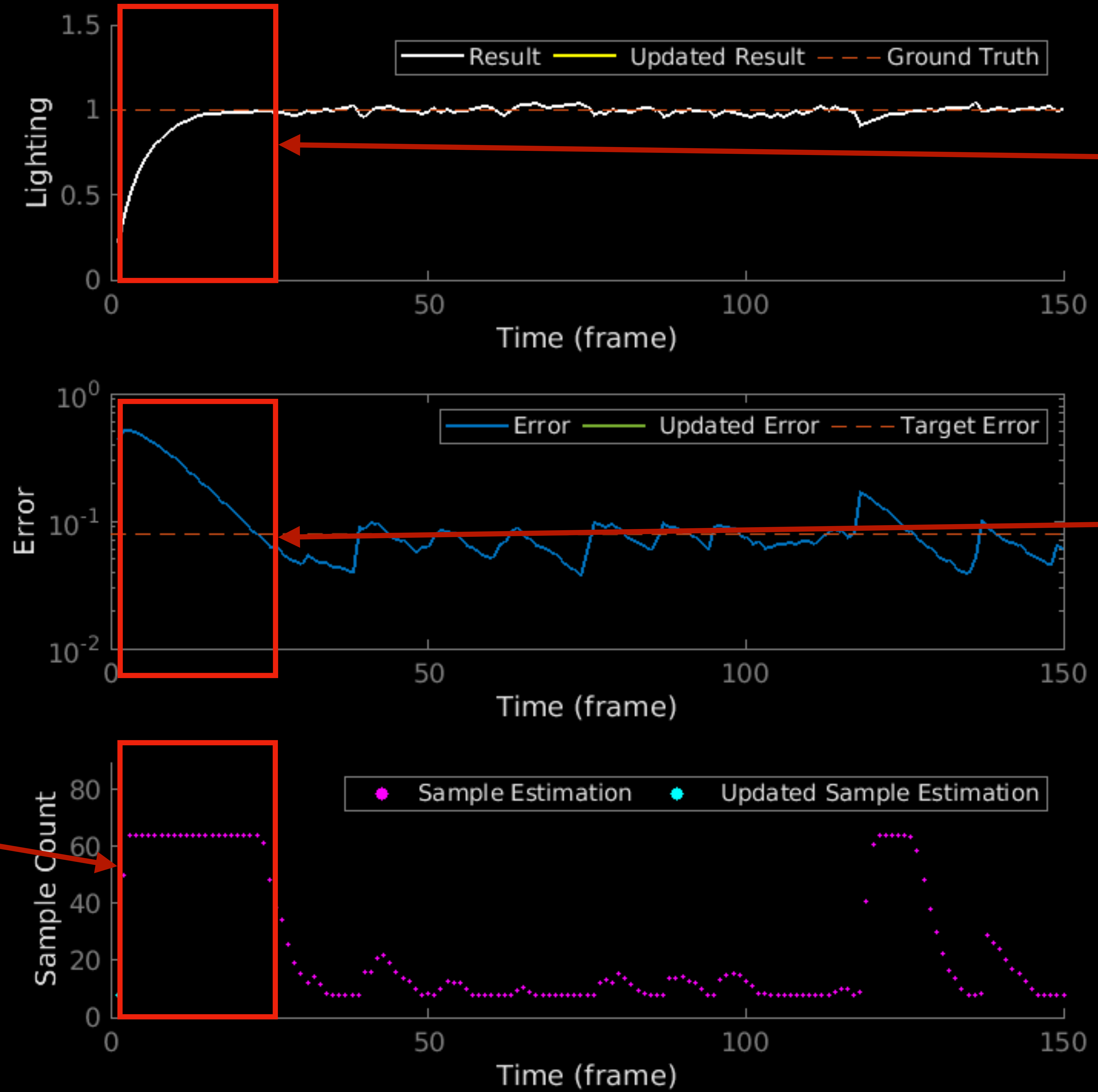
no history

Disocclusion

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

Irradiance texture



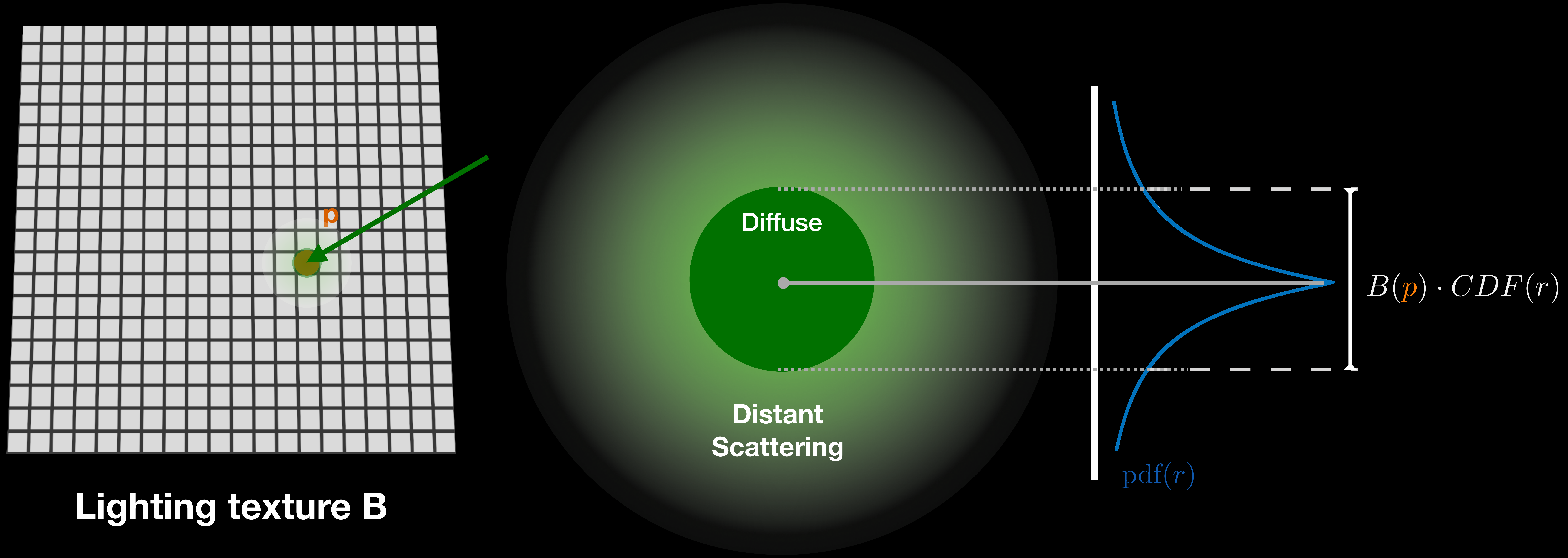


Lighting converges faster

Better error estimation

Overestimation resolved

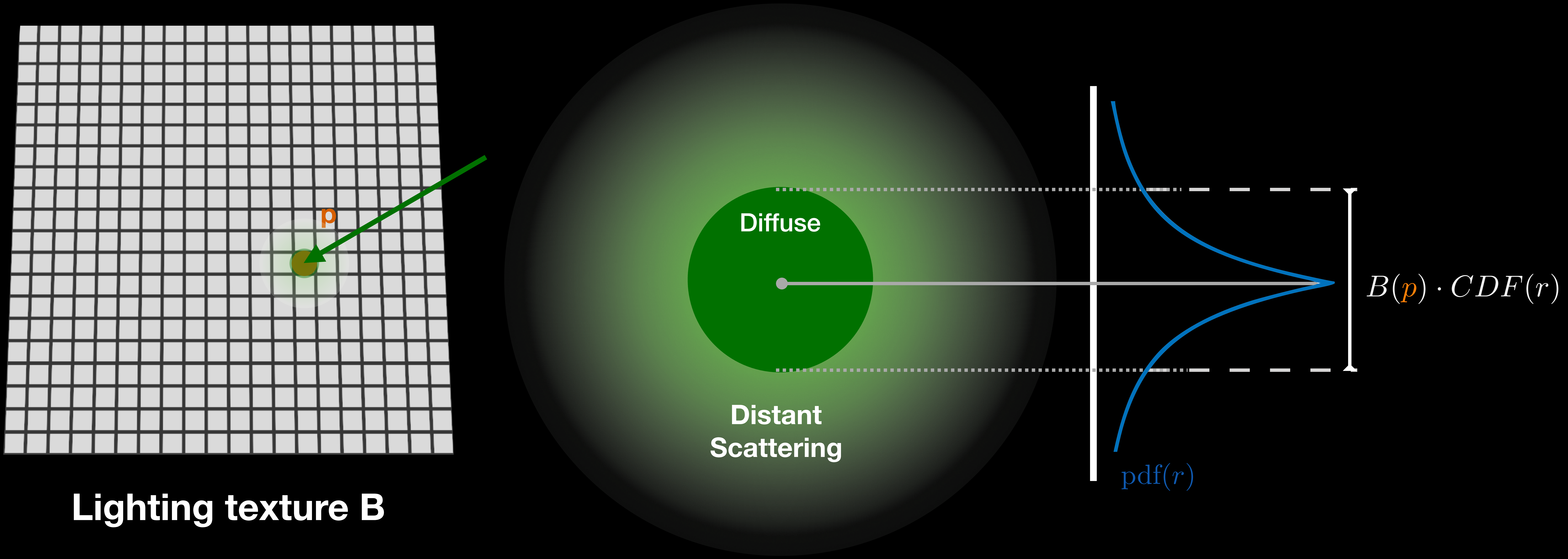
Separating Direct and Distant Scattering



Lighting texture B

Track the variance of distant scattering only

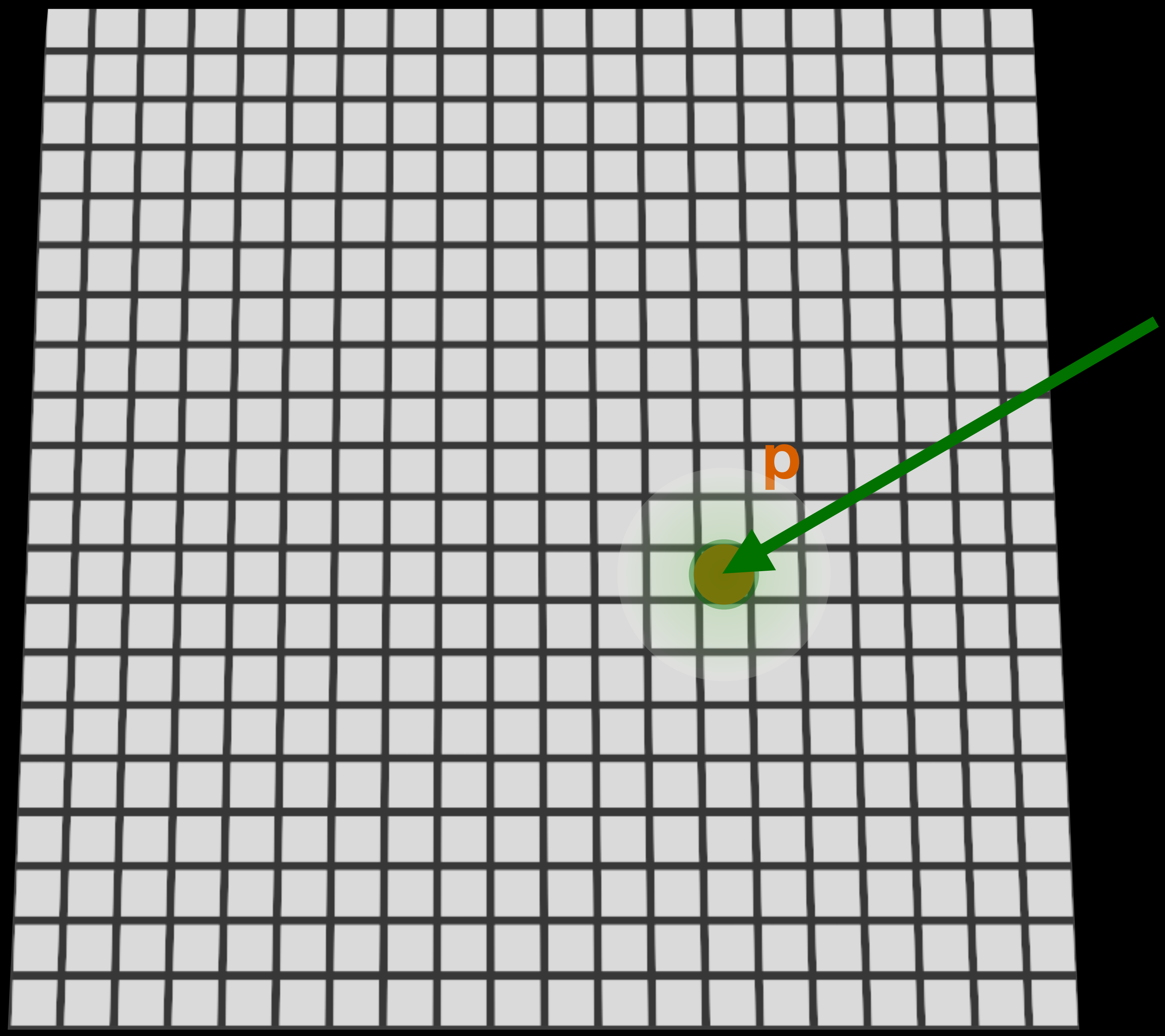
Separating Direct and Distant Scattering



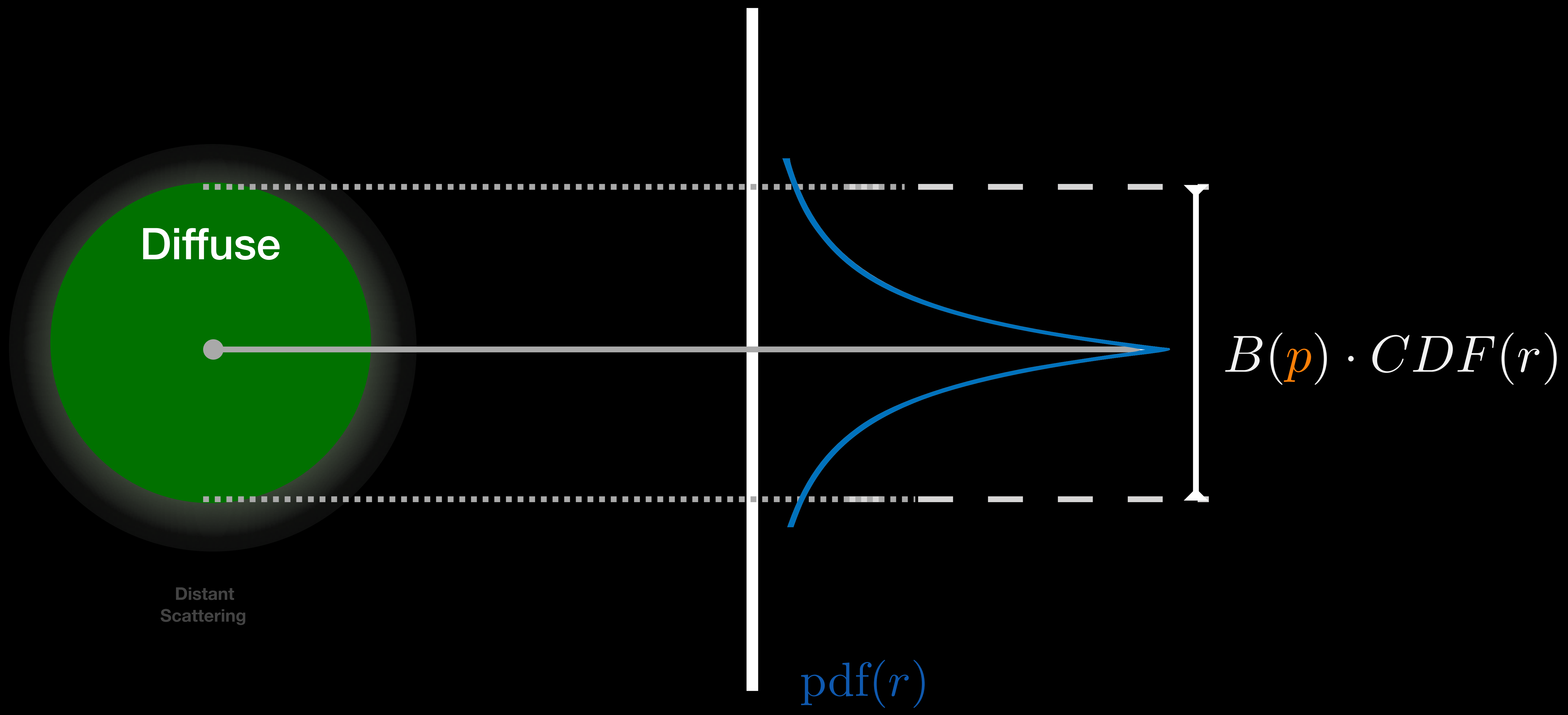
Lighting texture B

Track the variance of distant scattering only

Separating Direct and Distant Scattering

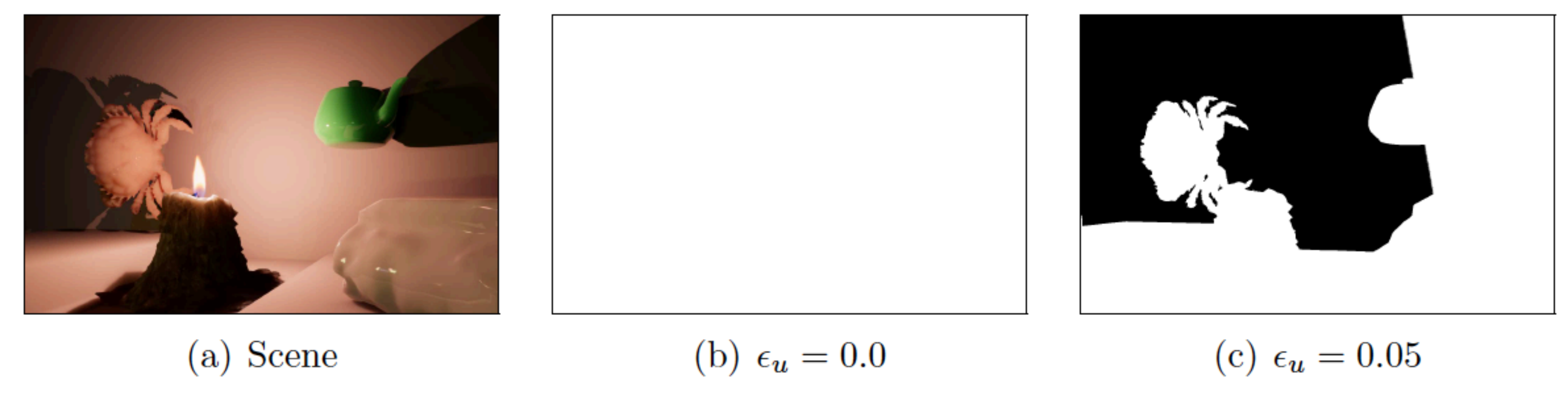


Lighting texture B



Ignore distant scattering: 0 spp

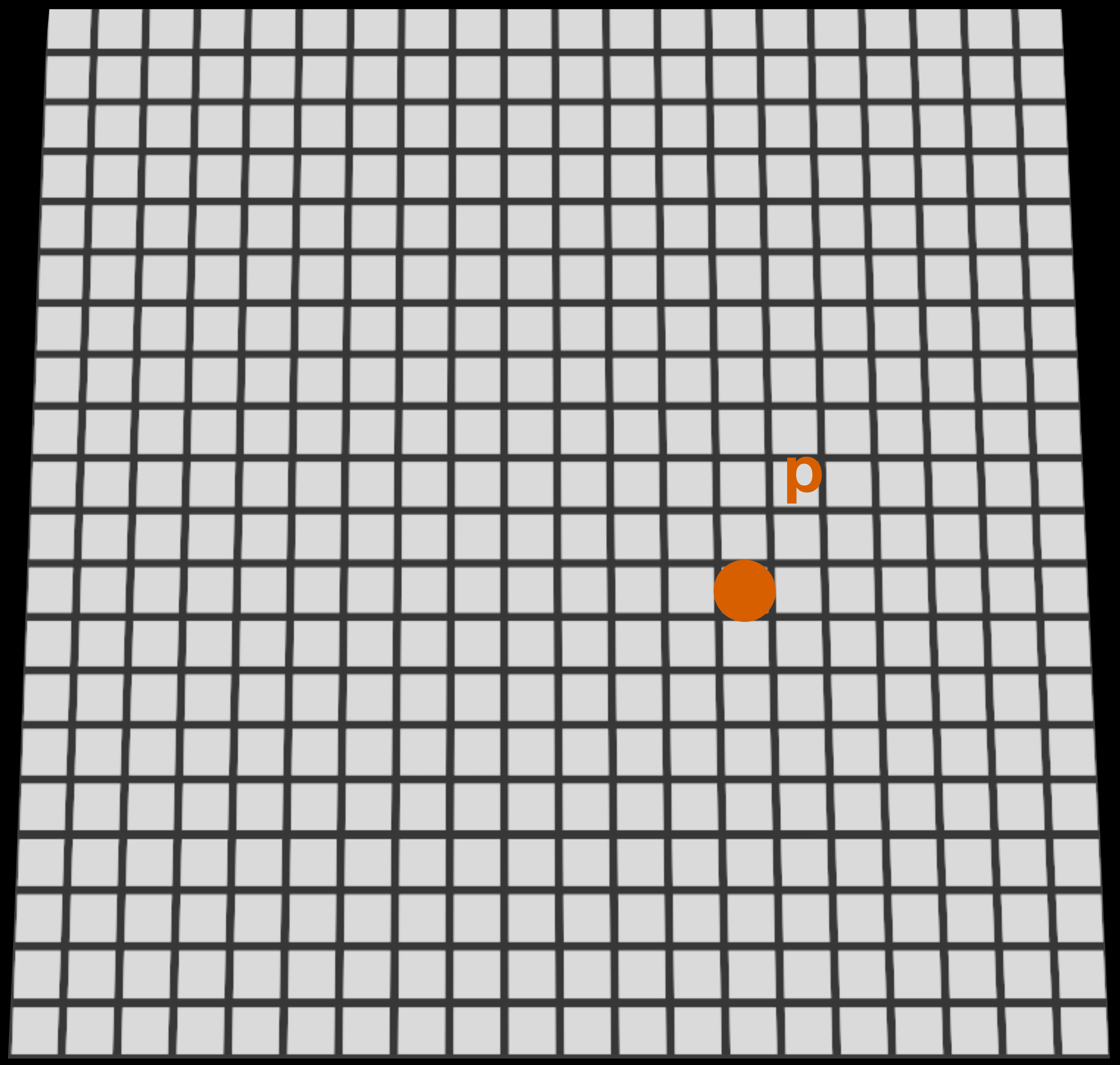
Separating Direct and Distant Scattering



(a) Scene (b) $\epsilon_u = 0.0$ (c) $\epsilon_u = 0.05$

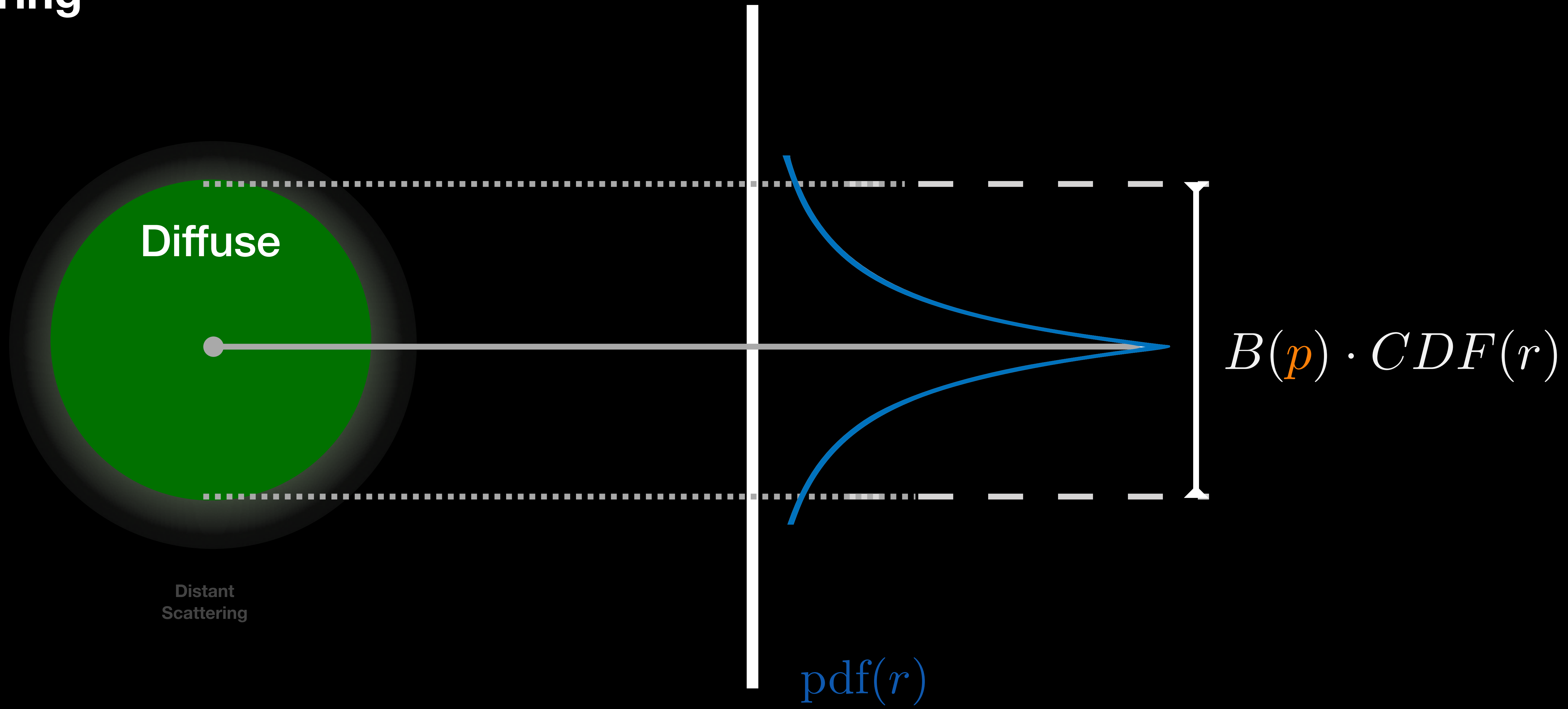
Direct/diffuse region (black) and direct+distant (white) (b,c) for scene (a). The vertical line on the wall (c) is the boundary where only 5% of scattering energy is from distant scattering.

Separating Direct and Distant Scattering



Lighting texture B

TAA jittering



Remove jittering variance of diffuse

In-frame Constant CV

When T is independent from F, G

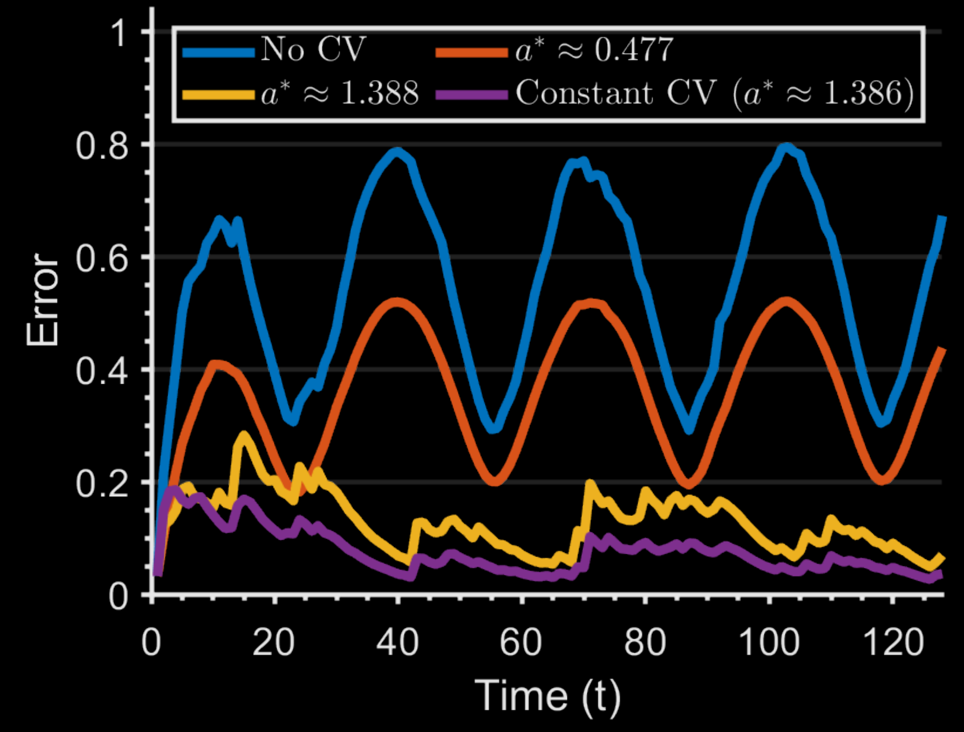
$$\begin{aligned} Var(\langle TF \rangle) &= Var(\langle TF - aTG \rangle + aTG) \\ &= Var(T(F - aG)) + Var(aTG) + 2Cov(T(F - aG), aTG) \end{aligned}$$

$$Var(T(F - aG)) = E[T]^2 Var(F - aG) + Var(T) Var(F - aG) + Var(T) E[F - aG]^2$$

In-frame constant CV:

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

$a^* \approx 1.388$
(1024 spp)



Simplification:

(Variance of CV residual)

$$Var(\langle TF \rangle) = Var(T(F - aG)) + Var(T) E[F]^2$$

$$Var(T(F - aG)) = E[T]^2 Var(F) + Var(T) Var(F)$$

Sample Estimation - monitor variance of CV residual:

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

Sample Count ↑

Decrease ↓

Unaffected

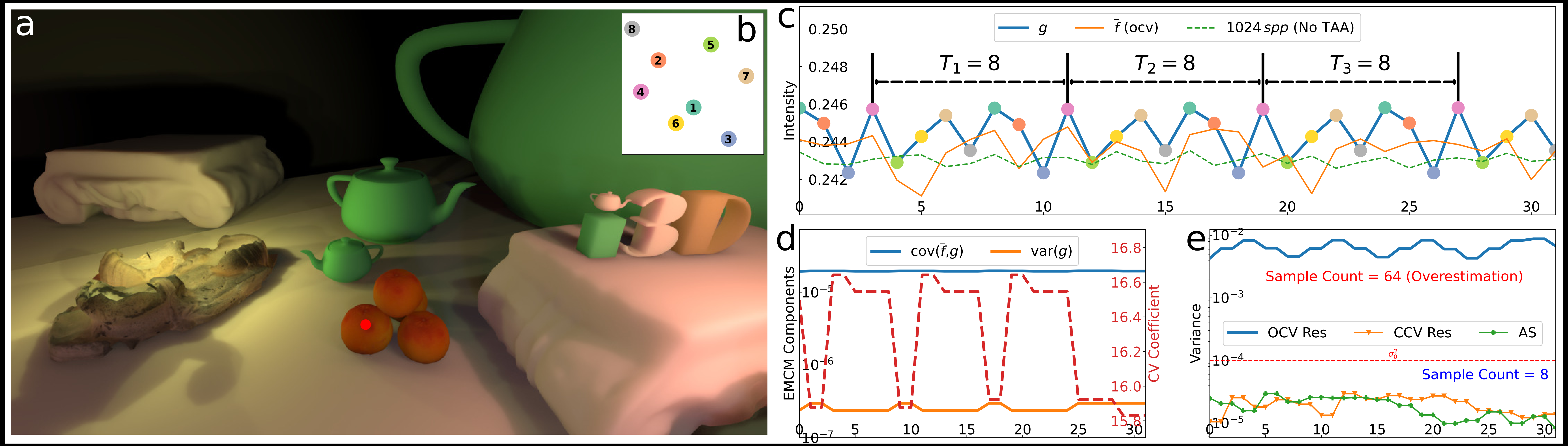
Static Scene with Jittering

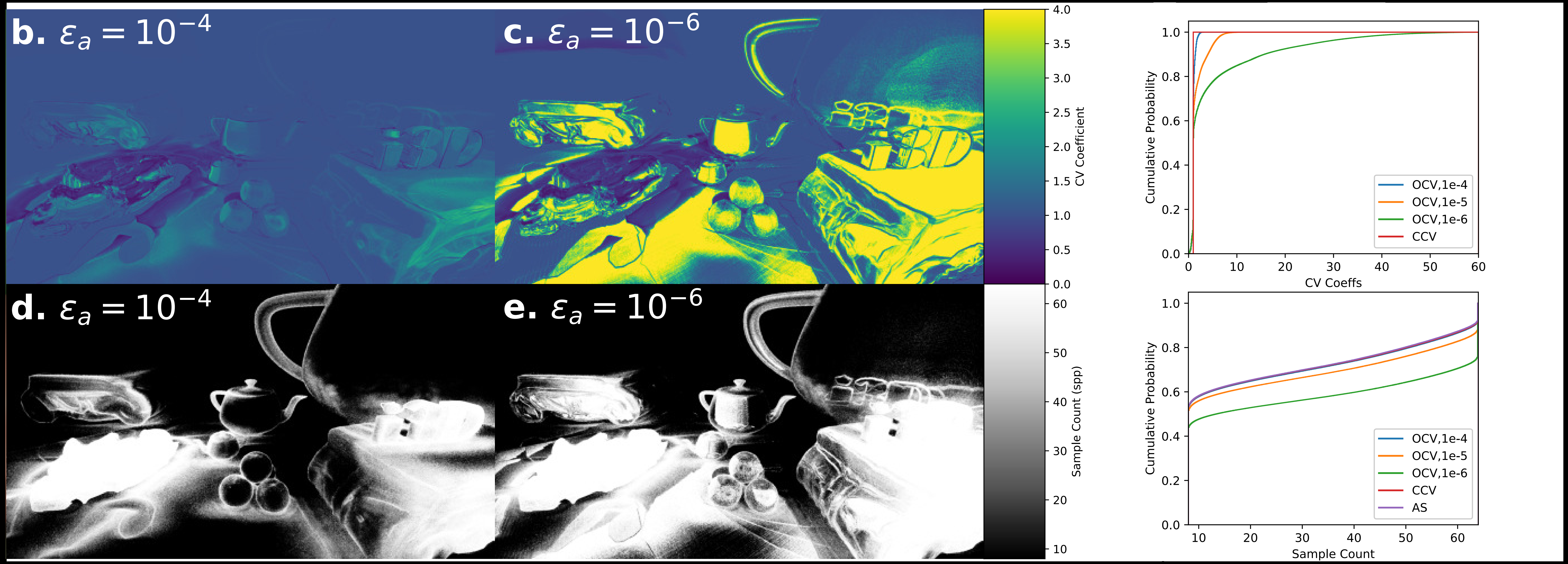
CV Coefficient estimation:

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

With suppressing factor:

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





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 In- ternational 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

- [Křivánek and Colbert 2008] Jaroslav Křivánek 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://jcgt.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.realtimerendering.com/s2014/epic/TemporalAA.ppt