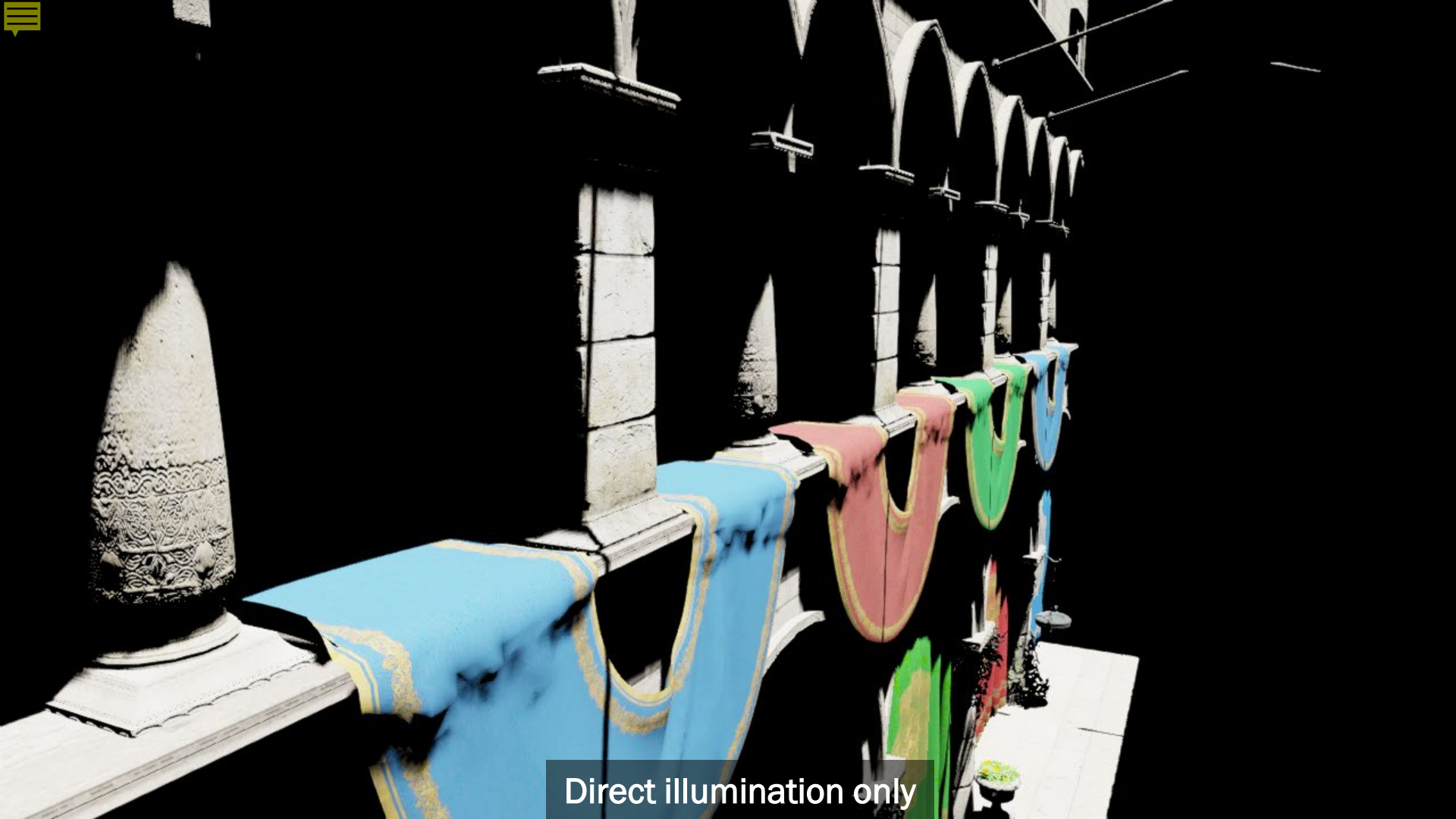
Real-Time Rendering with Lighting Grid Hierarchy

Daqi Lin & Cem Yuksel

University of Utah

I3D 2019


Direct illumination only


Our method (estimate global illumation from 1 million VPLs)
37 ms


Our method (estimate global illumation from 1 million VPLs, interleaved sampling)
12 ms

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]

Many-lights Problem

- Rendering with many lights

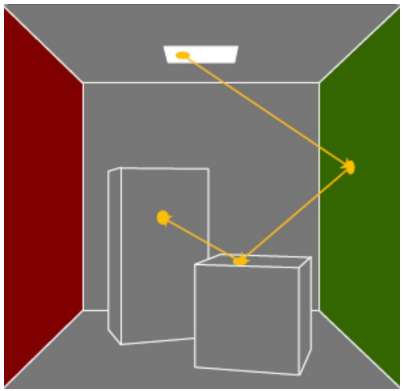
[Image source: https://80.lv/articles/the-future-of-real-time-rendering-with-lumberyard/]

Many-lights Problem

- Rendering with many lights
- Instant Radiosity [Keller 1997]

Many-lights Problem

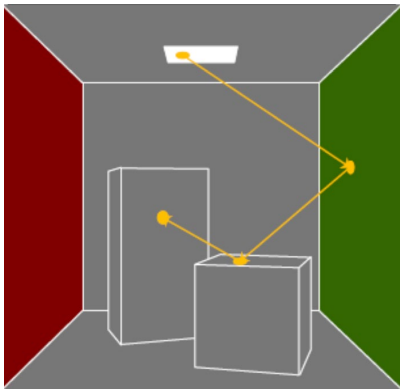
- Rendering with many lights
- Instant Radiosity [Keller 1997]

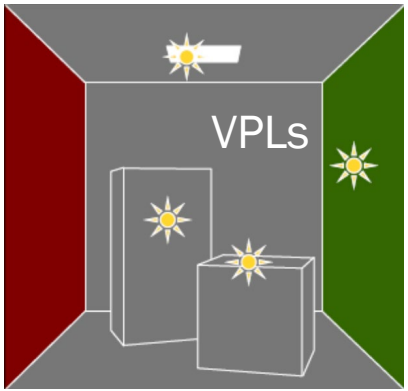


[Image source: Ivo Boyadzhiev and Kevin Matzen Presentation of Instant Radiosity at Cornell CS6630 Spring 2012]

Many-lights Problem

- Rendering with many lights
- Instant Radiosity [Keller 1997]

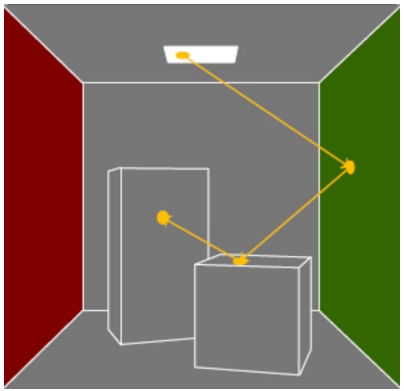


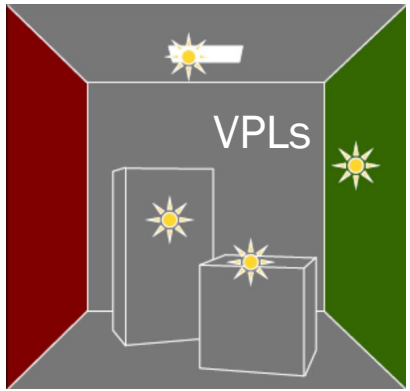


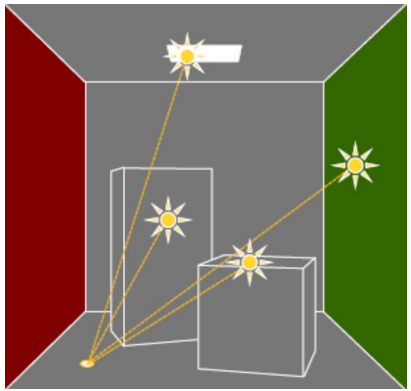
[Image source: Ivo Boyadzhiev and Kevin Matzen Presentation of Instant Radiosity at Cornell CS6630 Spring 2012]

Many-lights Problem

- Rendering with many lights
- Instant Radiosity [Keller 1997]



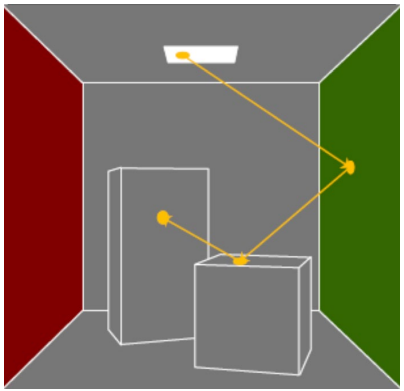


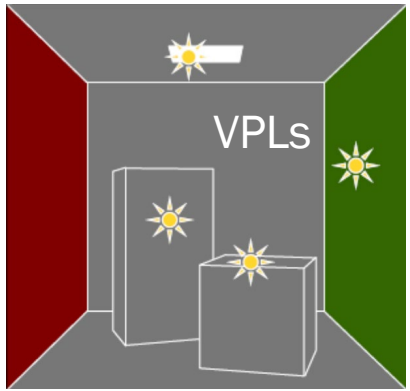


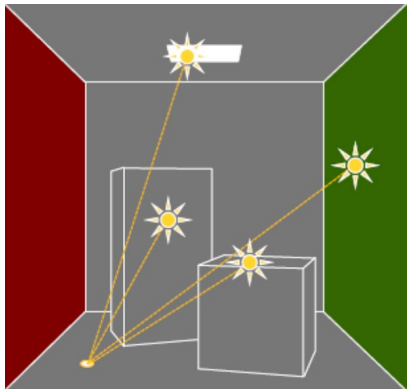
[Image source: Ivo Boyadzhiev and Kevin Matzen Presentation of Instant Radiosity at Cornell CS6630 Spring 2012]

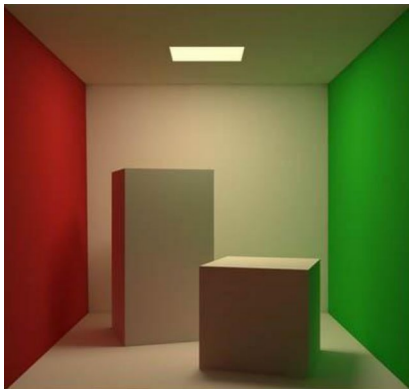
Many-lights Problem

- Rendering with many lights
- Instant Radiosity [Keller 1997]









[Image source: Ivo Boyadzhiev and Kevin Matzen Presentation of Instant Radiosity at Cornell CS6630 Spring 2012]

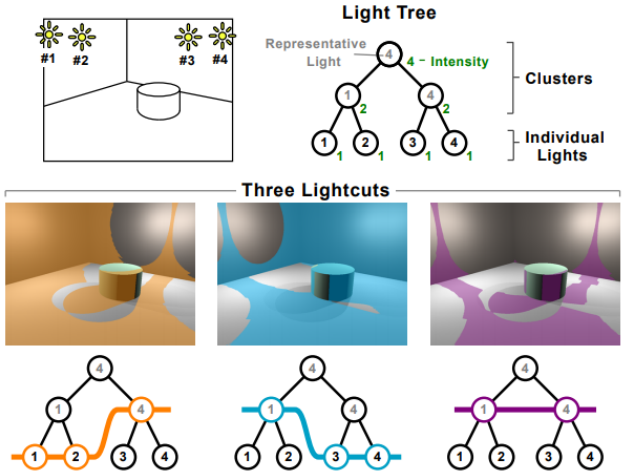
Many-lights Problem

- Rendering with many lights
- Instant Radiosity [Keller 1997]
  - Only efficient if VPL count is small

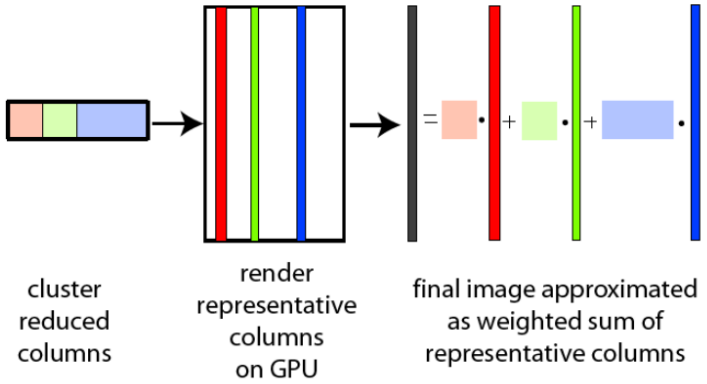
Many-lights Problem

- Offline Methods

Lightcuts
[Walter et al. 2005]



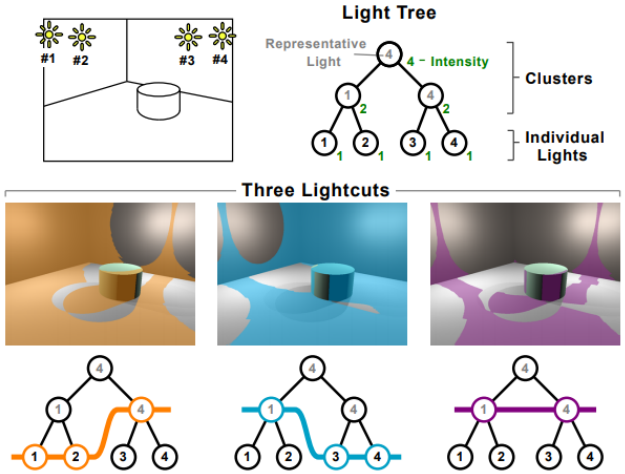
Matrix Row-Column Sampling
[Hašan et al. 2007]



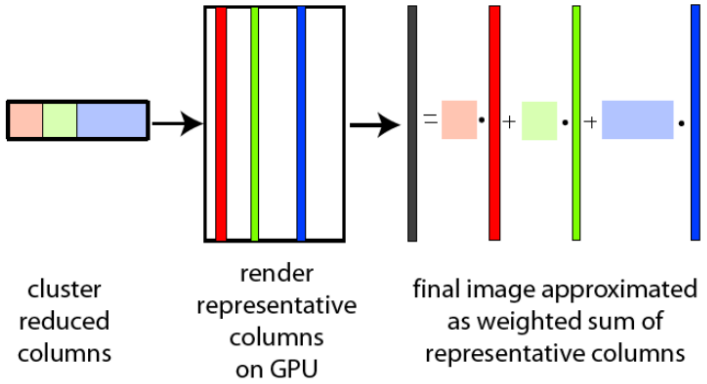
Many-lights Problem

- Offline Methods (temporally unstable)

Lightcuts
[Walter et al. 2005]



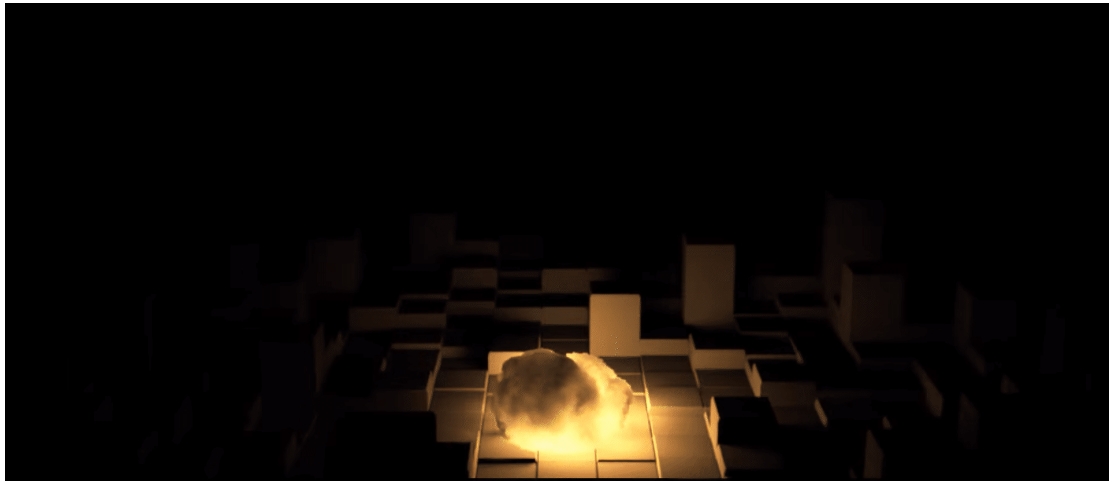
Matrix Row-Column Sampling
[Hašan et al. 2007]



Many-lights Problem

- Offline Methods (temporally stable)

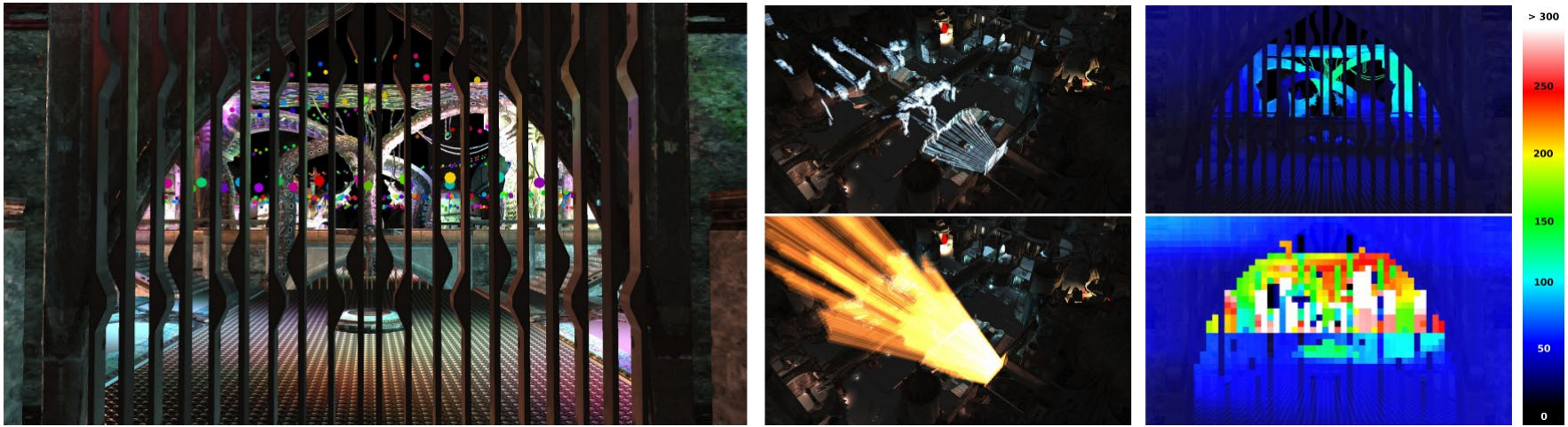
Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



Many-lights Problem

- Real-Time Methods

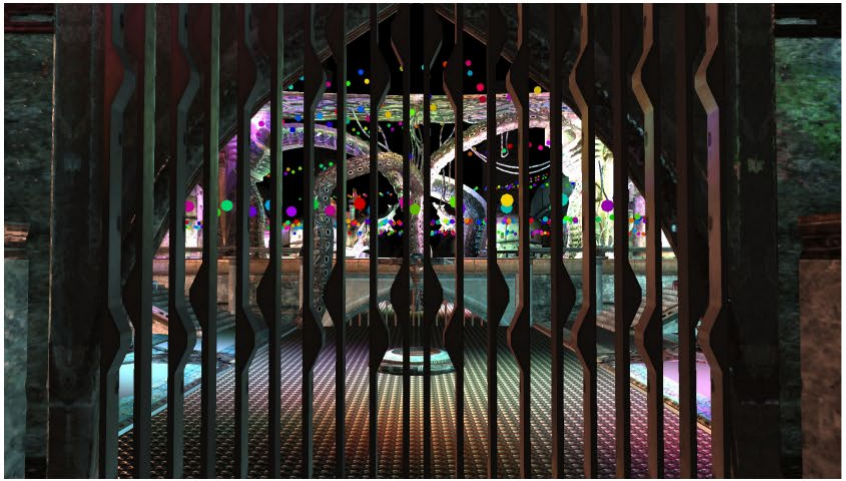
Clustered Shading [Olsson et al. 2012]



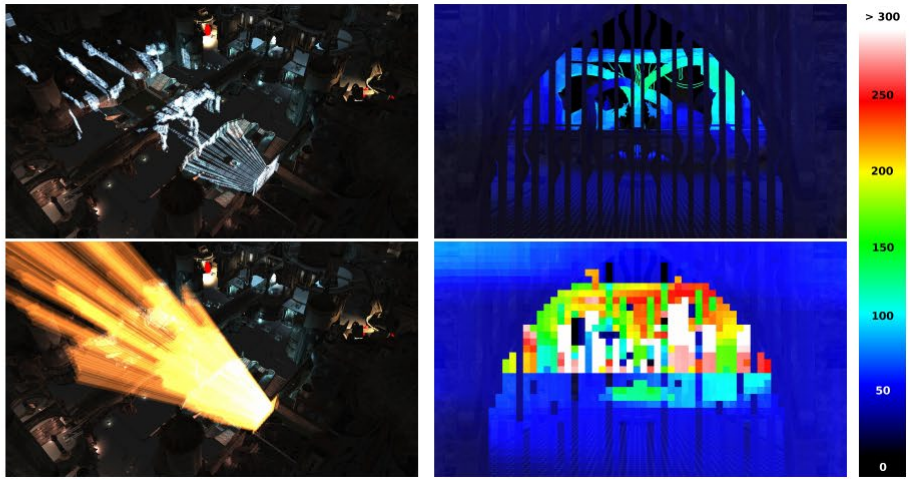
Many-lights Problem

- Real-Time Methods

Clustered Shading [Olsson et al. 2012]



VPL (with global influence) ✗



Many-lights Problem

- Real-Time Methods

Forward light cuts [Laurent et al. 2016]

Many-lights Problem

- Real-Time Methods

Forward light cuts [Laurent et al. 2016]

VPL Shadows ✗

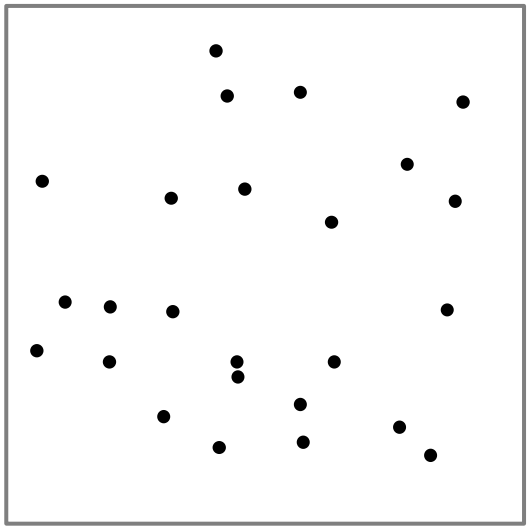
Real Time Rendering With Lighting Grid Hierarchy

- Ours
  - A large number of lights ✓
  - VPL (with global Influence) ✓
  - VPL Shadows ✓

➡ Real-time solution to many-lights

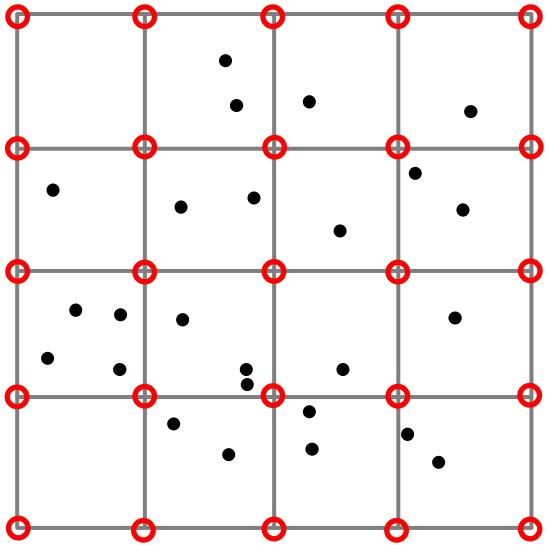
➡ Real time global illumination with VPLs

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]

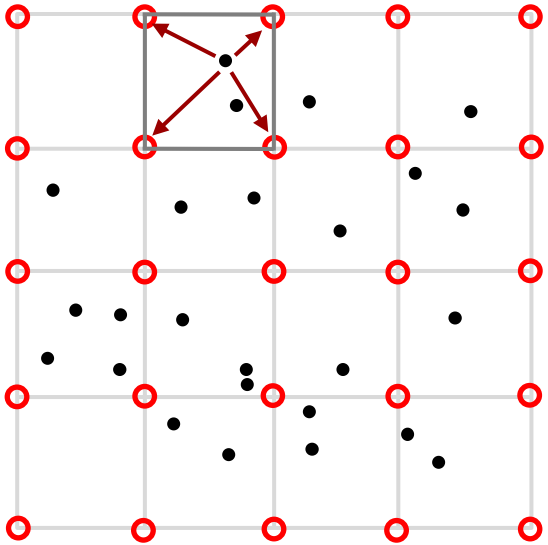


$\mathbb{S}_0$ (VPLs)

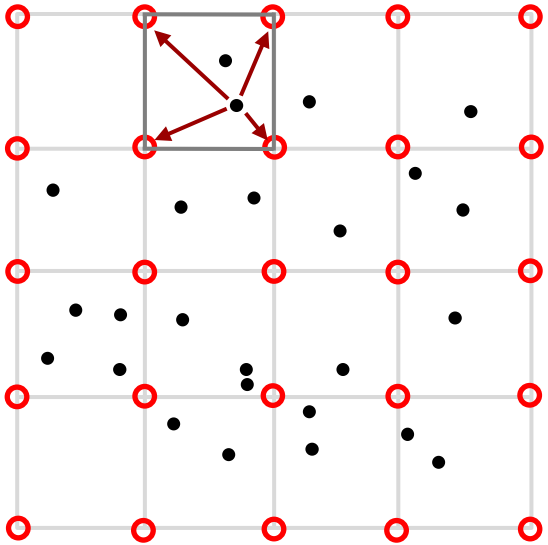
Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



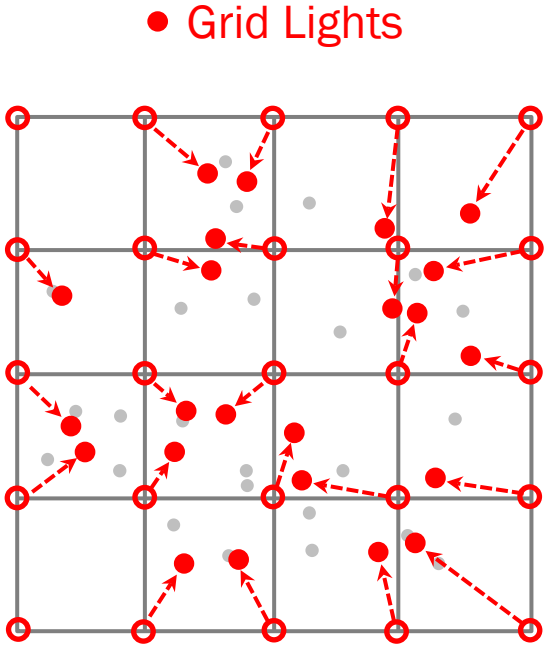
Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



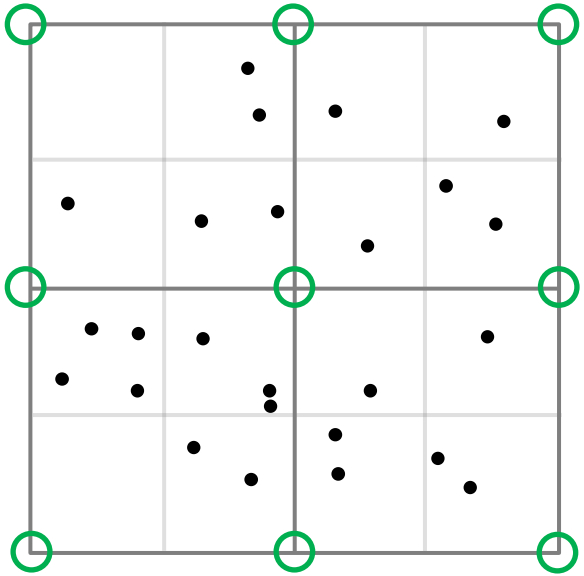
Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



Lighting Grid Hierarchy [Yuksel & Yuksel 2017]

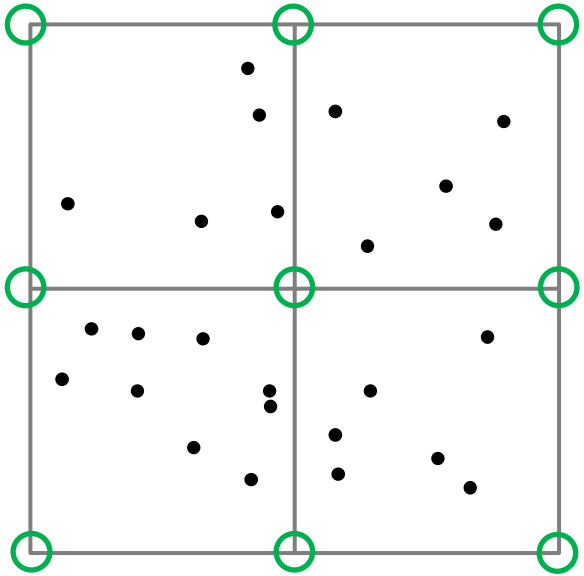


Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



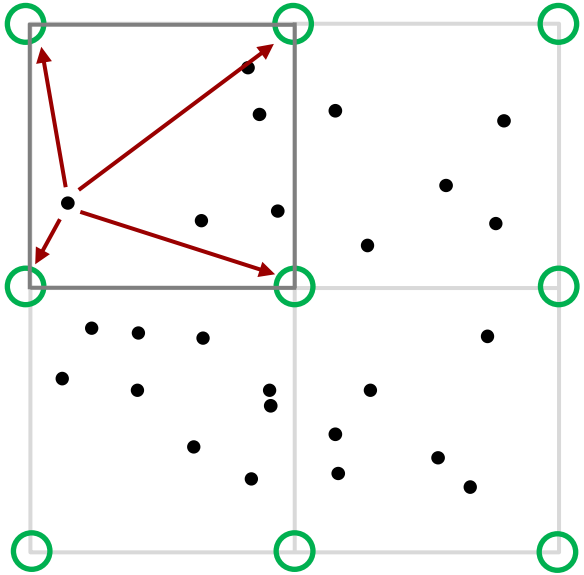
$\mathbb{S}_2$

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



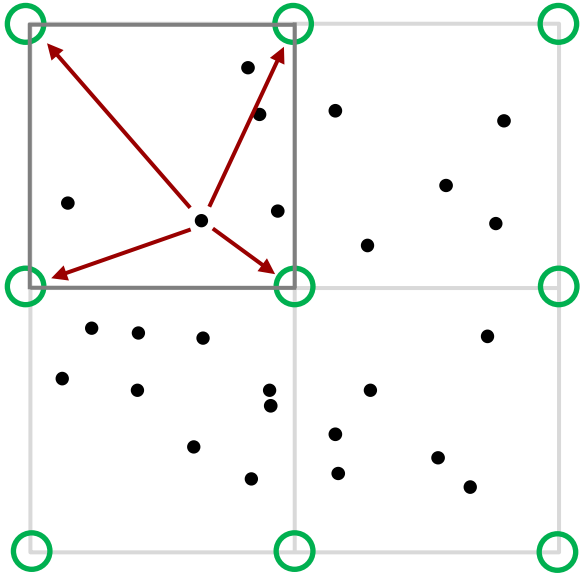
$\mathbb{S}_2$

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



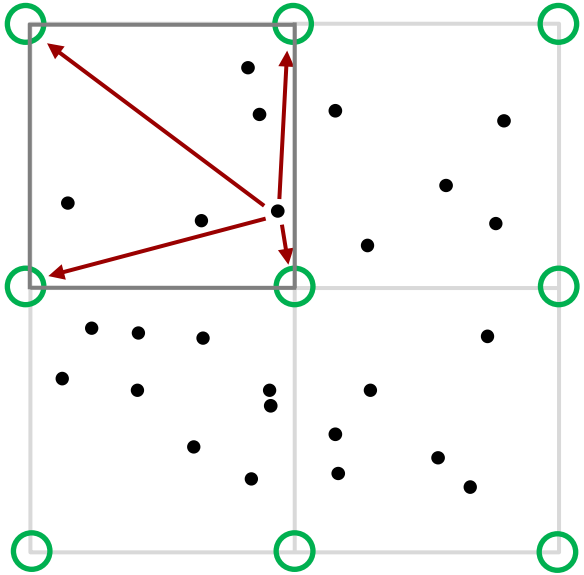
$\mathbb{S}_2$

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



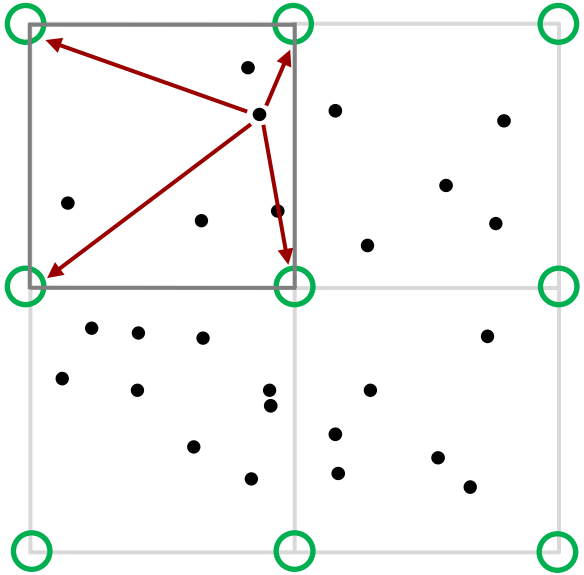
$\mathbb{S}_2$

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



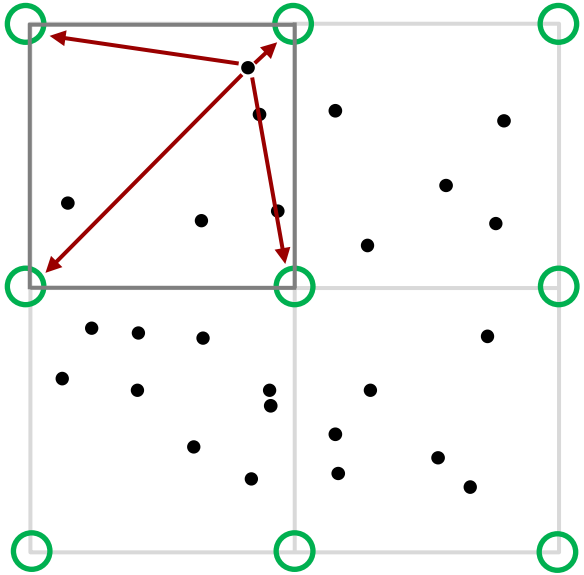
$\mathbb{S}_2$

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



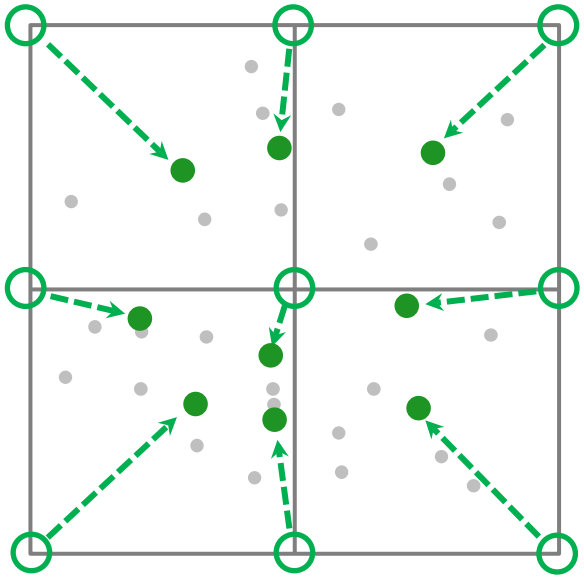
$\mathbb{S}_2$

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]

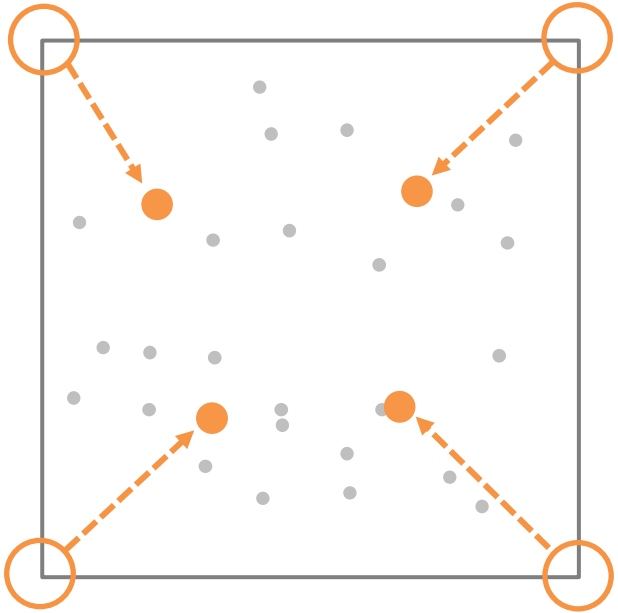


$\mathbb{S}_2$

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]



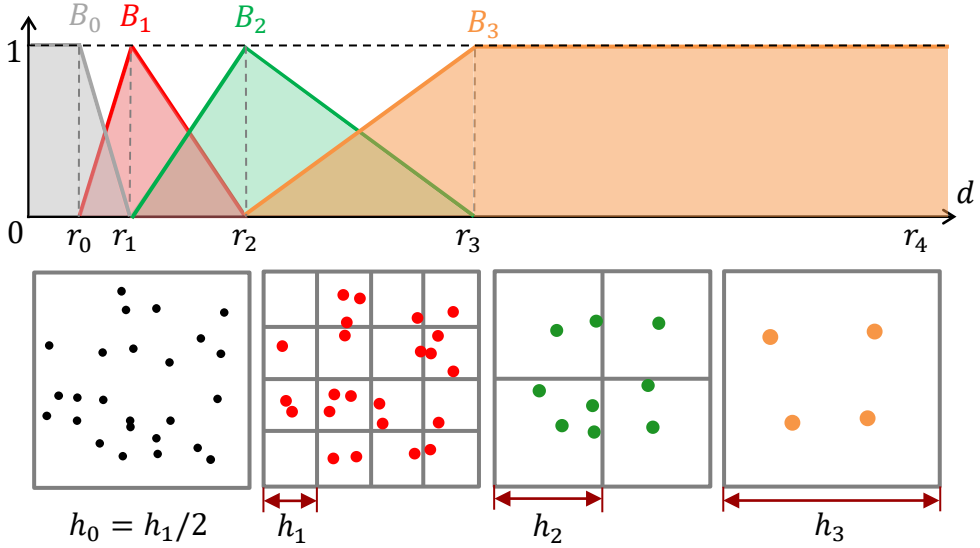
$\mathbb{S}_2$



$\mathbb{S}_3$

Lighting Grid Hierarchy [Yuksel & Yuksel 2017]

- Blending Functions (larger α improves accuracy)

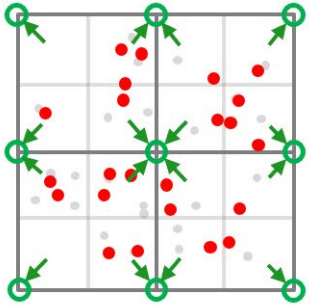


$$\underline{r_\ell} = \alpha \underline{h_\ell}$$

Influence Radius — Grid Spacing

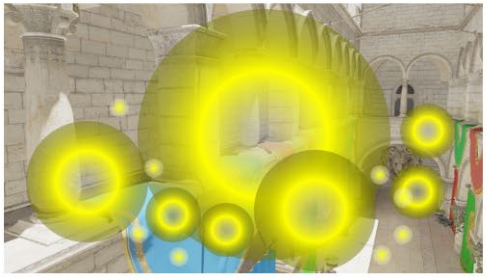
Our Algorithm

1



GPU Construction of LGH

2



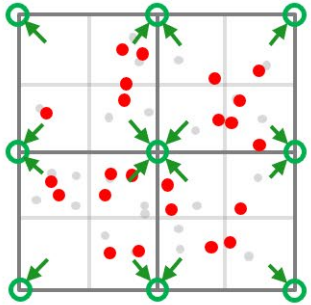
Lighting Computation

3

Shadow Sampling

Our Algorithm

1



GPU Construction of LGH

2

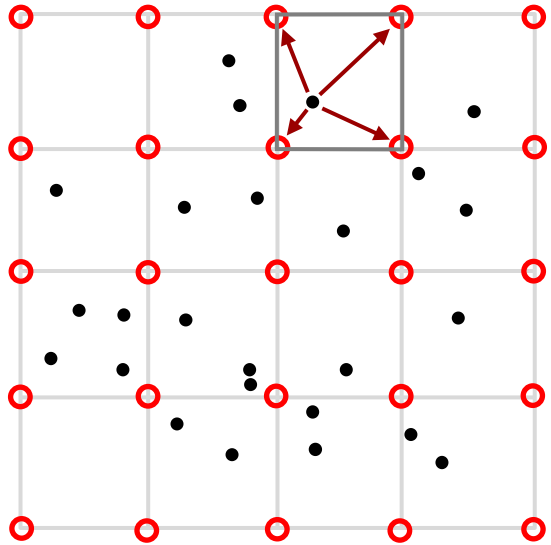
Lighting Computation

3

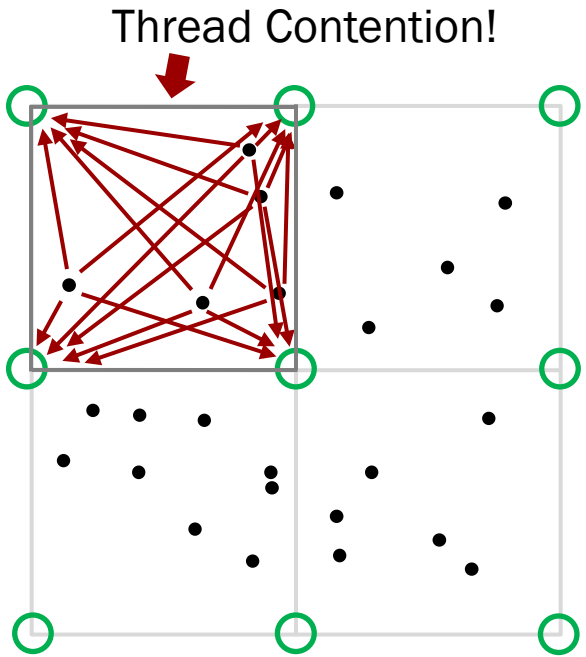
Shadow Sampling

GPU Construction of LGH

- Scatter VPLs to all levels ✕



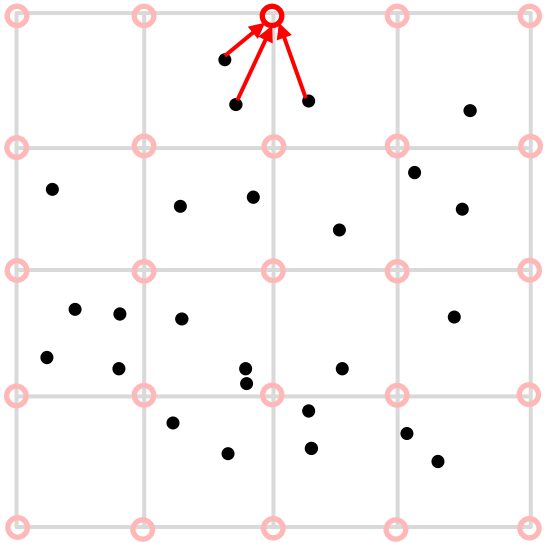
$\mathbb{S}_1$



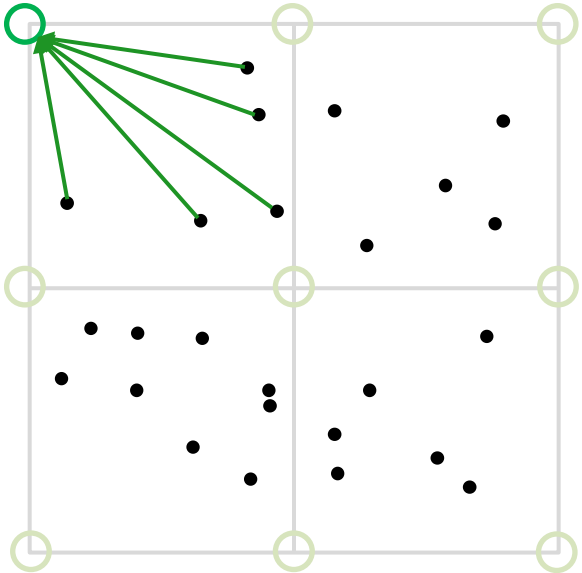
$\mathbb{S}_2$

GPU Construction of LGH

- Gather ✗ Need to search for surrounding VPLs!



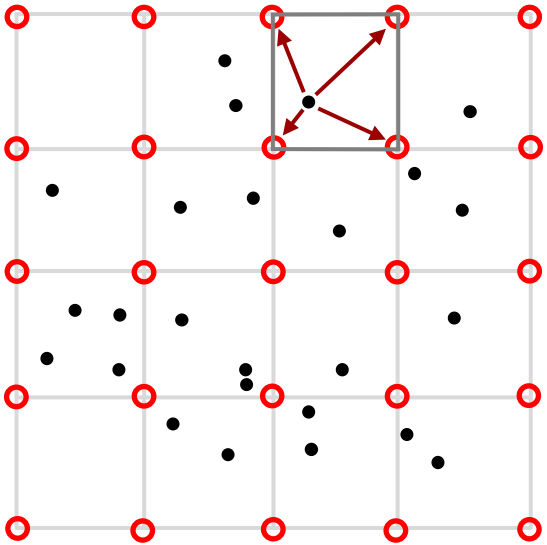
$\mathbb{S}_1$



$\mathbb{S}_2$

GPU Construction of LGH

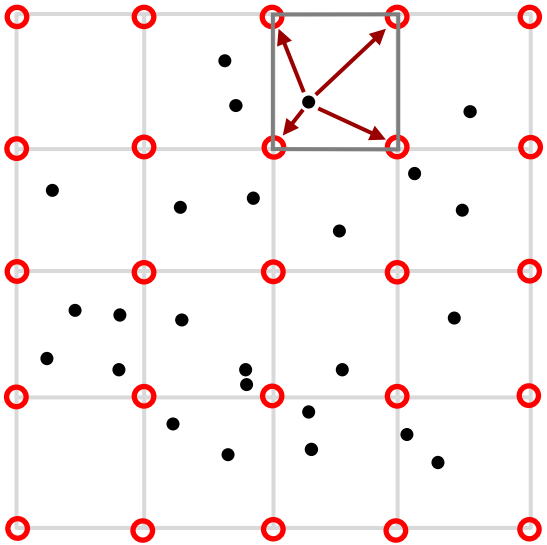
- Scatter VPLs to $\mathbb{S}_1$



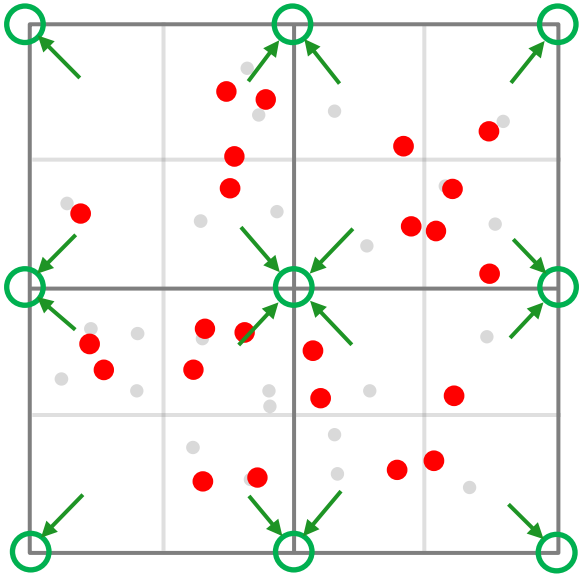
$\mathbb{S}_1$

GPU Construction of LGH

- Scatter VPLs to $\mathbb{S}_1$, gather from $\mathbb{S}_1$ for upper levels ✓



$\mathbb{S}_1$



$\mathbb{S}_2$

GPU Construction of LGH

- Lighting grid hierarchy build time for 100k VPLs

	Scatter VPLs	Gather from $\mathbb{S}_1$
Compute bounds	1.7 ms	1.7 ms
Compute $\mathbb{S}_1$	2.3 ms	2.3 ms
Compute $\mathbb{S}_2$	2.3 ms	1.0 ms
Compute $\mathbb{S}_3$	4.7 ms	1.0 ms
Compute $\mathbb{S}_4$	23 ms	2.0 ms
Compute $\mathbb{S}_5$	107 ms	1.0 ms
Compute $\mathbb{S}_6$	405 ms	1.1 ms
Compute $\mathbb{S}_7$	1,563 ms	1.5 ms
Merge levels	0.5 ms	0.5 ms
Total	**2,110 ms**	**12.1 ms**

174x faster

GPU Construction of LGH

- Lighting grid hierarchy build time for 100k VPLs

	Scatter VPLs	Gather from $\mathbb{S}_1$
Compute bounds	1.7 ms	1.7 ms
Compute $\mathbb{S}_1$	2.3 ms	2.3 ms
Compute $\mathbb{S}_2$	2.3 ms	1.0 ms
Compute $\mathbb{S}_3$	4.7 ms	1.0 ms
Compute $\mathbb{S}_4$	23 ms	2.0 ms
Compute $\mathbb{S}_5$	107 ms	1.0 ms
Compute $\mathbb{S}_6$	405 ms	1.1 ms
Compute $\mathbb{S}_7$	1,563 ms	1.5 ms
Merge levels	0.5 ms	0.5 ms
Total	**2,110 ms**	**12.1 ms**

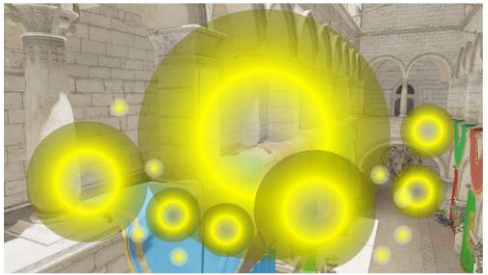
1042x faster

Our Algorithm

1

GPU Construction of LGH

2



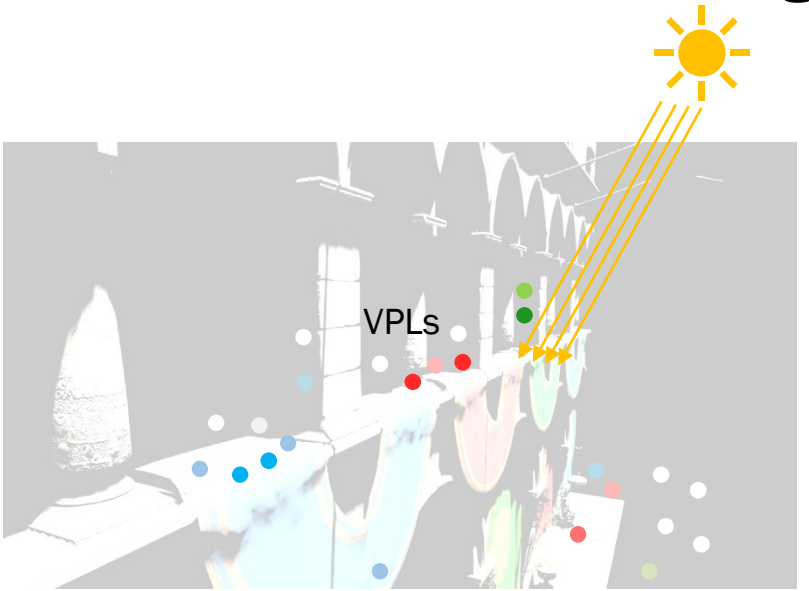
Lighting Computation

3

Shadow Sampling

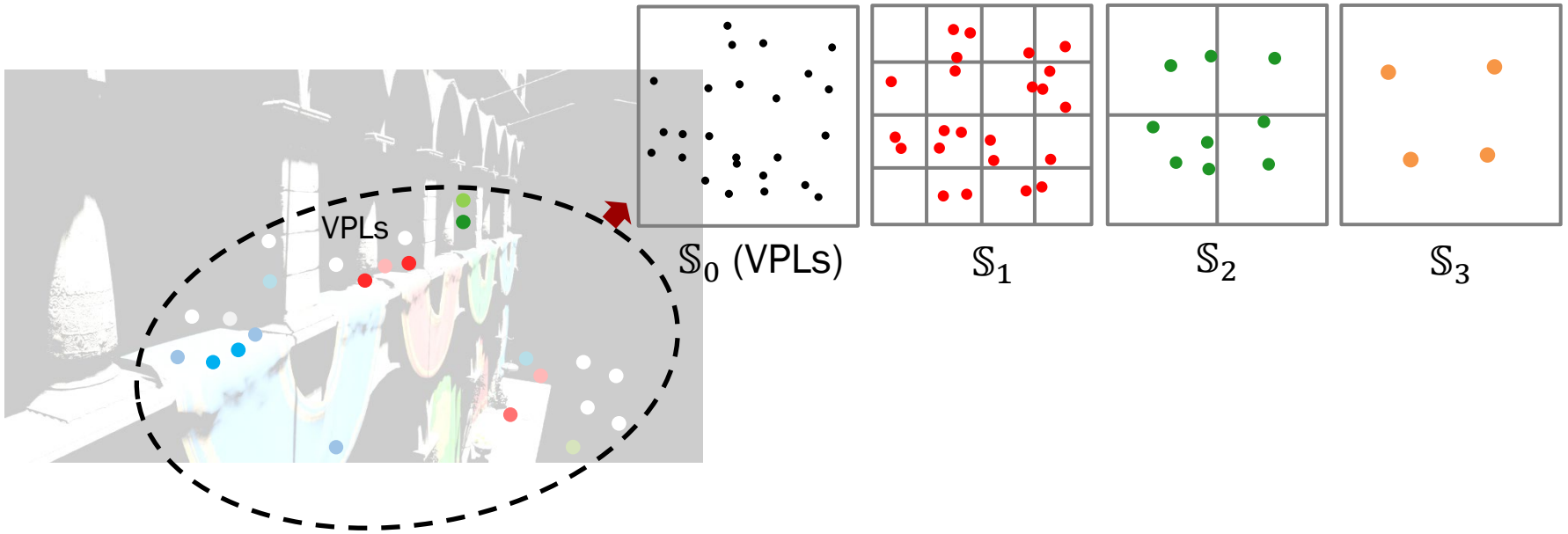
Lighting Computation

- Generate VPLs from light sources



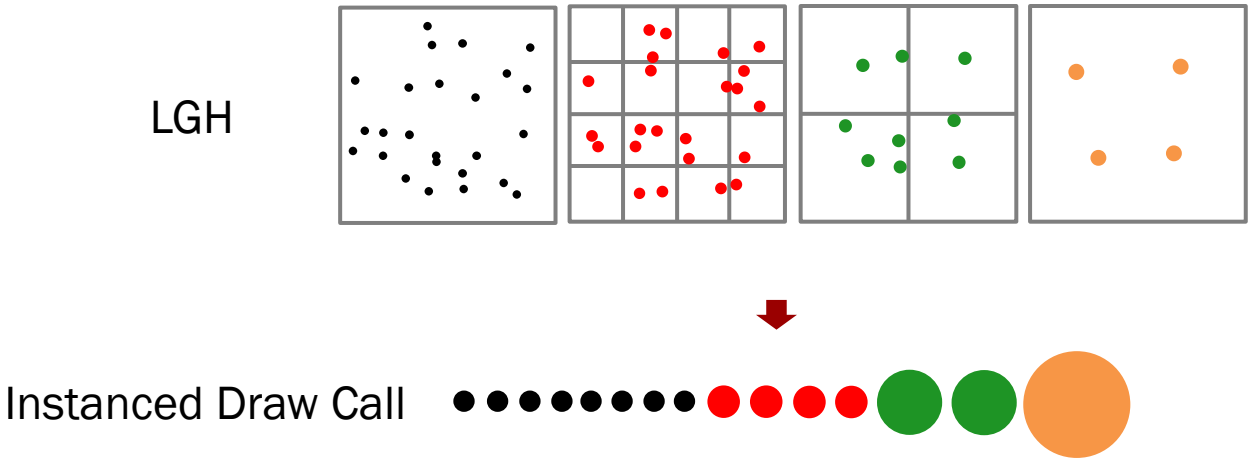
Lighting Computation

- Build Lighting Grid Hierarchy



Lighting Computation

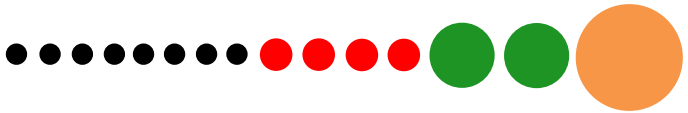
- Rasterize lights as (coarse) spheres in a deferred renderer



Lighting Computation

- Splat illumination attenuated by the blending function

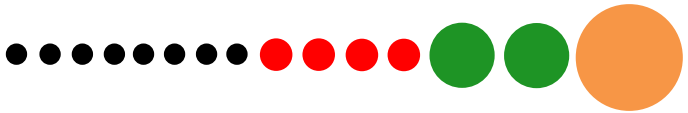
Instanced Draw Call



Lighting Computation

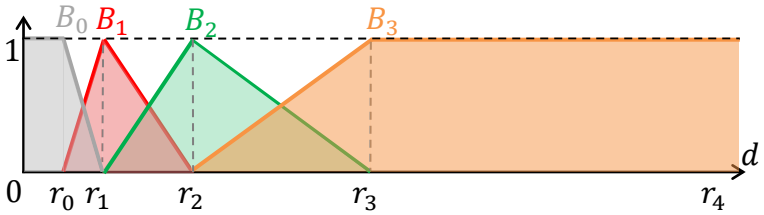
- Splat illumination attenuated by the blending function

Instanced Draw Call

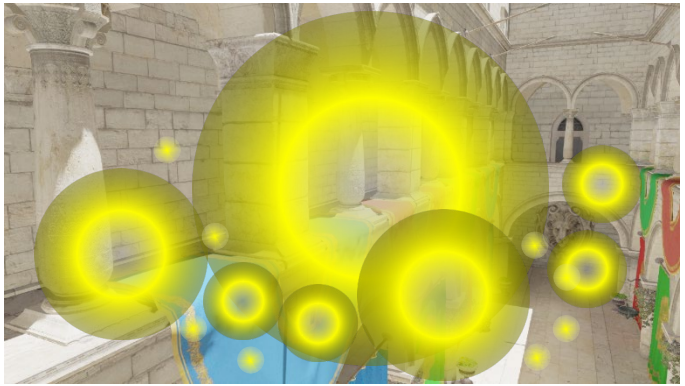


+

Blending function



Unshadowed indirect illumination!

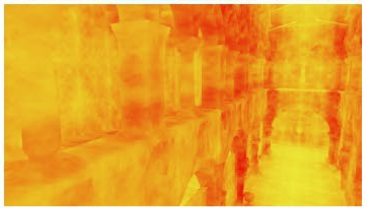


Lighting Computation

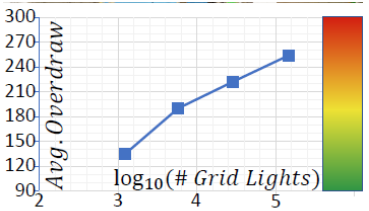
1K grid lights
Avrg. Overdraw: 135

5K grid lights
Avrg. Overdraw: 190

28K grid lights
Avrg. Overdraw: 222



144K grid lights
Avrg. Overdraw: 254



# of Grid Lights vs.
Avrg. Overdraw

Our Algorithm

1

GPU Construction of LGH

2

Lighting Computation

3

Shadow Sampling

Shadow Sampling

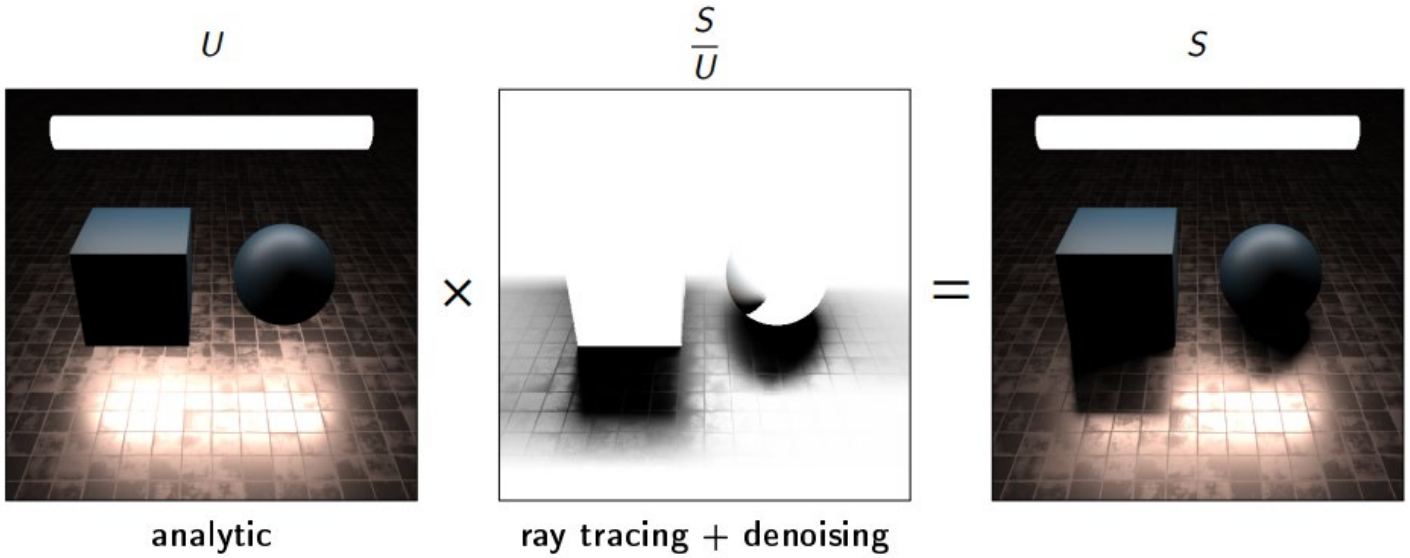
- Send shadow rays to hundreds of lights / pixel ! ✗

Shadow Sampling

- Send shadow rays to hundreds of lights / pixel ! ✗
- Small number of shadow samples & Importance sampling ✓

Shadow Sampling

- Shadow Ratio Estimator [Heitz et al. 2018]



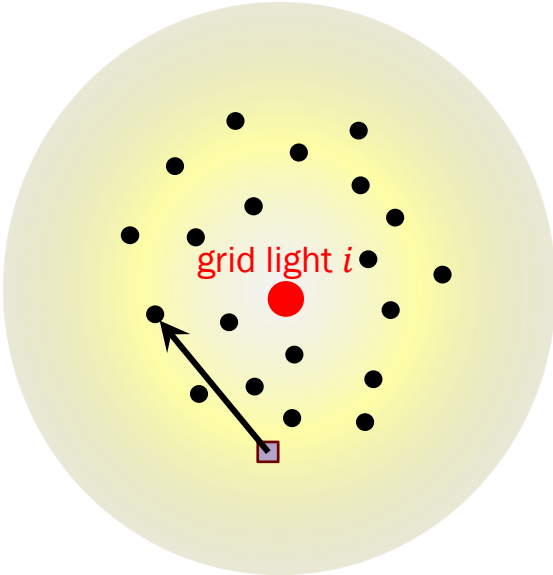
Shadow Sampling

- Our method: pick k shadow samples with desired probabilities using a fixed memory footprint

Shadow Sampling

- Our method: pick k shadow samples with desired probabilities using a fixed memory footprint

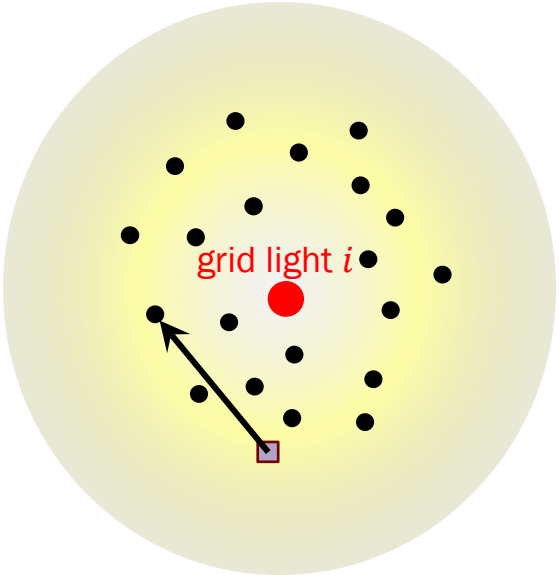
Each fragment samples
a position in the point cloud of
grid light i



Shadow Sampling

- Our method: pick k shadow samples with desired probabilities using a fixed memory footprint

The sampled light overwrites the shadow sample with a probability related to f_i (its P.D.F.) and $\sum_{j=0}^{i} f_j$ (the sum of previous P.D.F.)



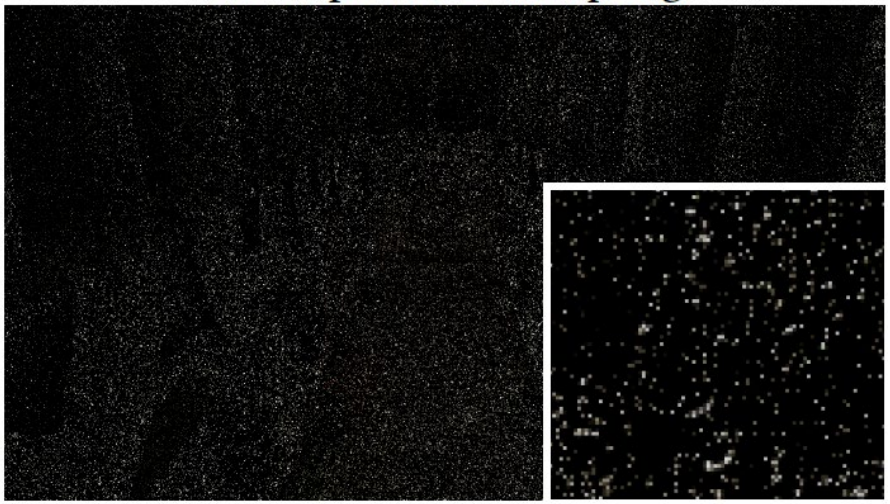
Shadow Sampling

- Our method: pick k shadow samples with desired probabilities using a fixed memory footprint

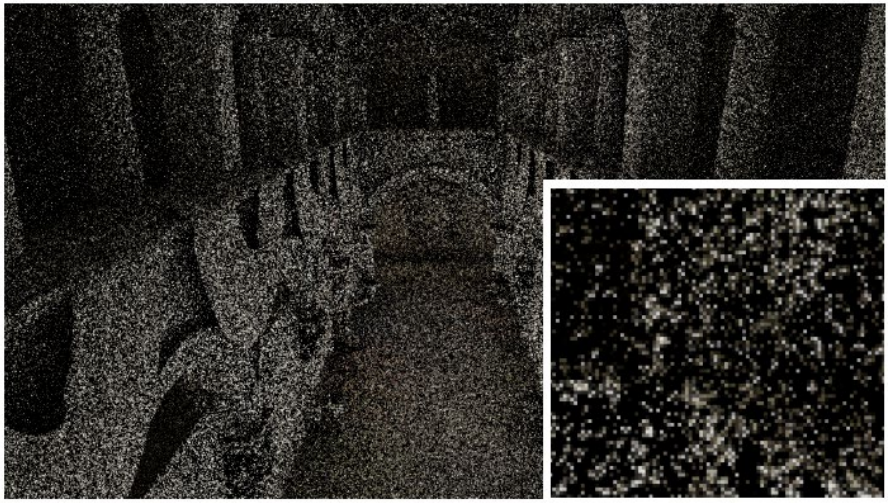
Result: Light i (P.D.F. f_i) picked with desired probability $\frac{f_i}{\sum_{j=0}^{n} f_j}$

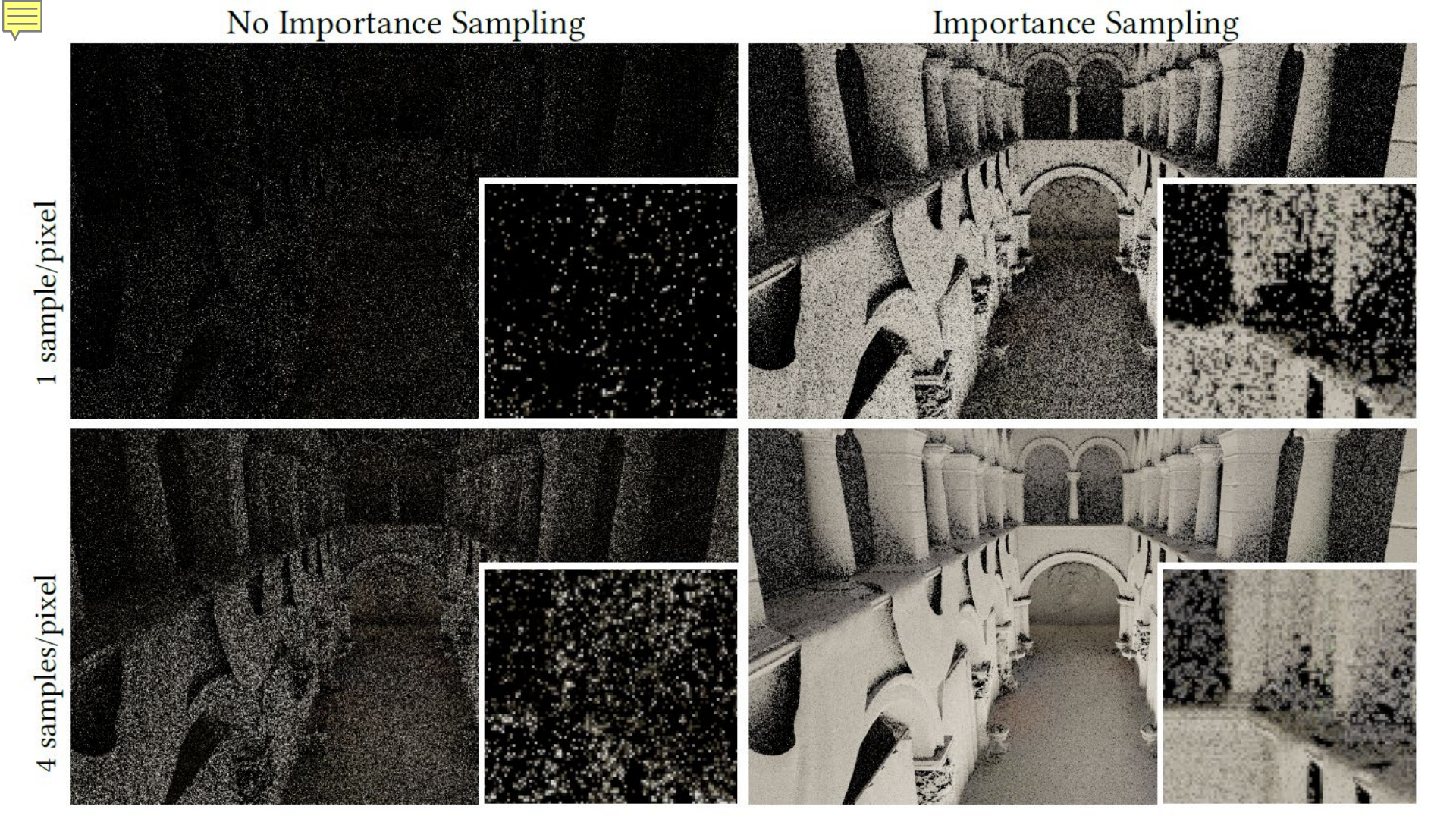
No Importance Sampling

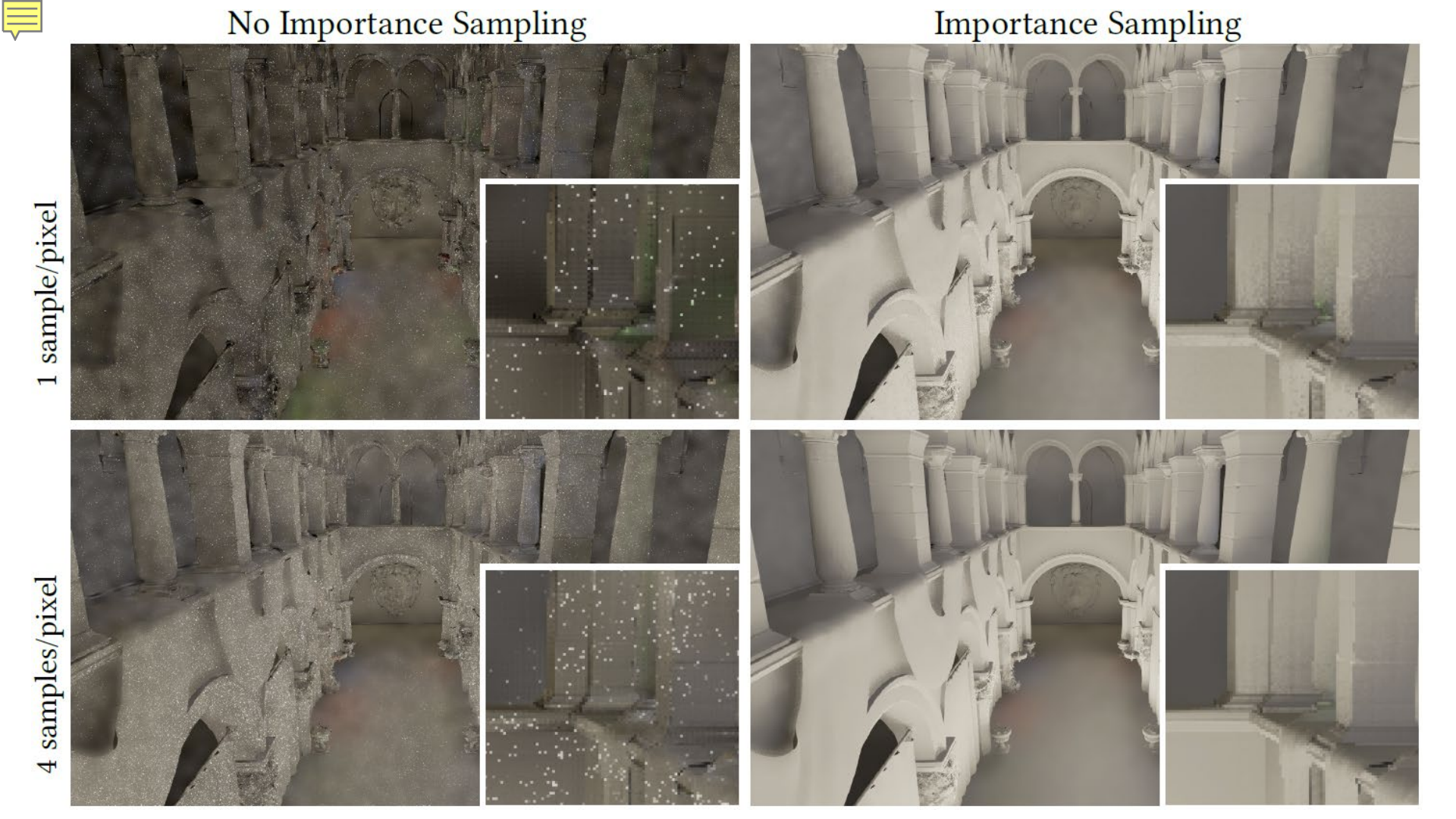
1 sample/pixel



4 samples/pixel

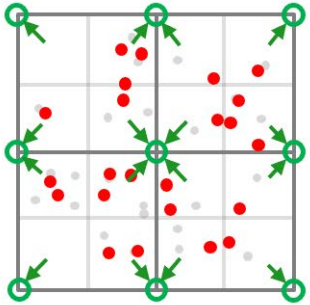



No Importance Sampling
Importance Sampling
1 sample/pixel
4 samples/pixel


No Importance Sampling
Importance Sampling
1 sample/pixel
4 samples/pixel

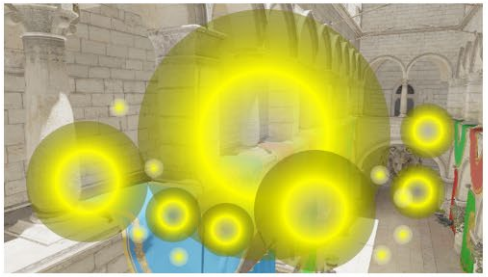
Our Algorithm

1



GPU Construction of LGH

2



Lighting Computation

3

Shadow Sampling

Results

Settings:
Crytek Sponza at 1280 x 720 ($\alpha = 1$)

Graphics Card:
RTX 2080

Number of VPLs	**1K**	**10K**	**100K**	**1M**
VPL Generation	0.1 ms	0.2 ms	0.4 ms	2.5 ms
Hierarchy Construction	7.0 ms	9.2 ms	12.1 ms	32.0 ms

Results

Settings:
Crytek Sponza at 1280 x 720 ($\alpha = 1$)

Graphics Card:
RTX 2080

Number of VPLs	**1K**	**1M**	1000x
VPL Generation	0.1 ms	2.5 ms	25x
Hierarchy Construction	7.0 ms	32.0 ms	4.5x

Results

Settings:
Crytek Sponza at 1280 x 720 ($\alpha = 1$)

Graphics Card:
RTX 2080

Number of VPLs	**1K**	**10K**	**100K**	**1M**
Render time (1 shadow/pixel)	10.1 ms	14.3 ms	18.4 ms	24.3 ms
Render time (4 shadows/pixel)	15.5 ms	22.4 ms	28.5 ms	37.0 ms

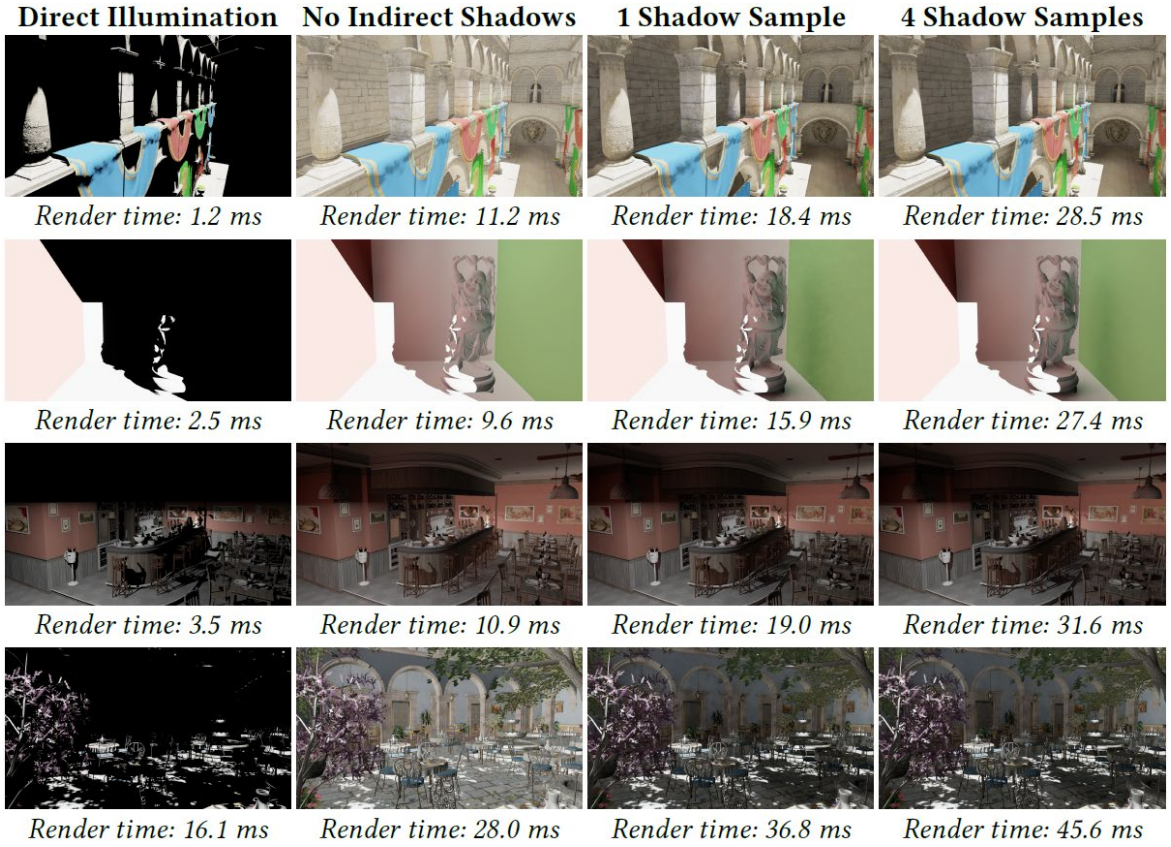
Results

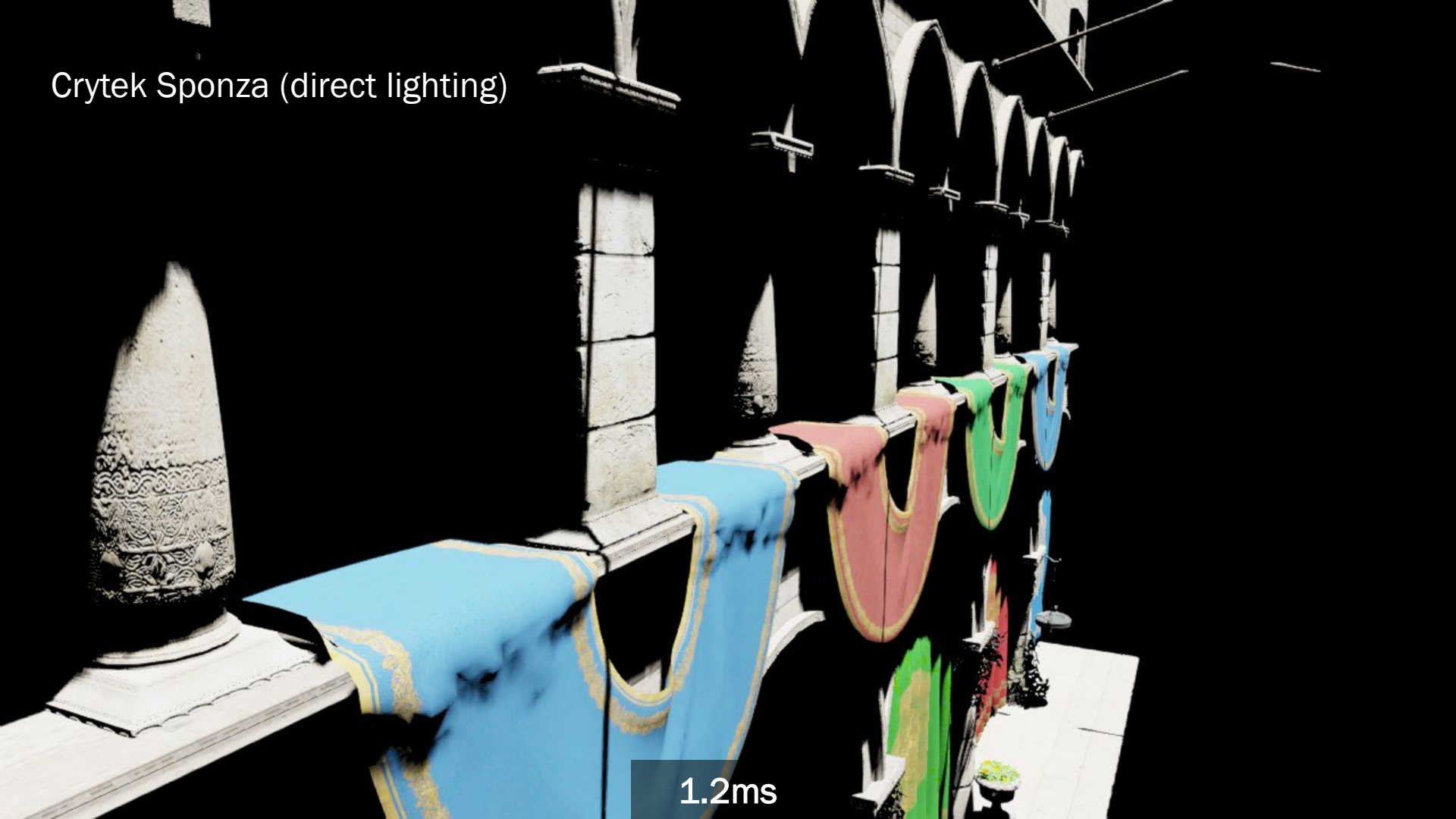
Settings:
Crytek Sponza at 1280 x 720 ($\alpha = 1$)

Graphics Card:
RTX 2080

Number of VPLs	**1K**	**1M**	1000x
Render time (1 shadow/pixel)	10.1 ms	24.3 ms	2.4x
Render time (4 shadows/pixel)	15.5 ms	37.0 ms	2.4x
	1.5x	1.5x	

Result (thumbnail)




Crytek Sponza (direct lighting)
1.2ms

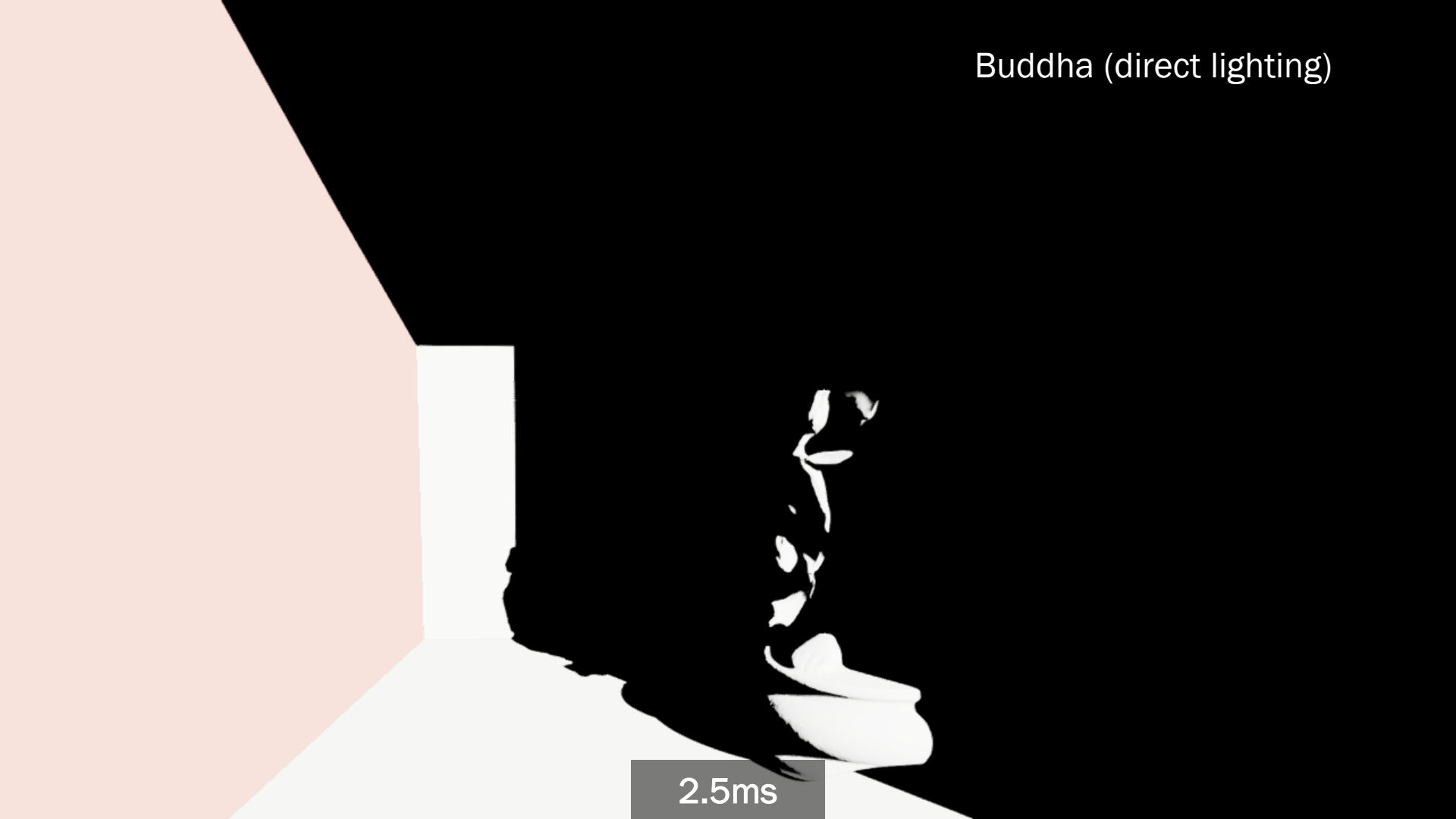

Crytek Sponza (no indirect shadows)
11.2ms

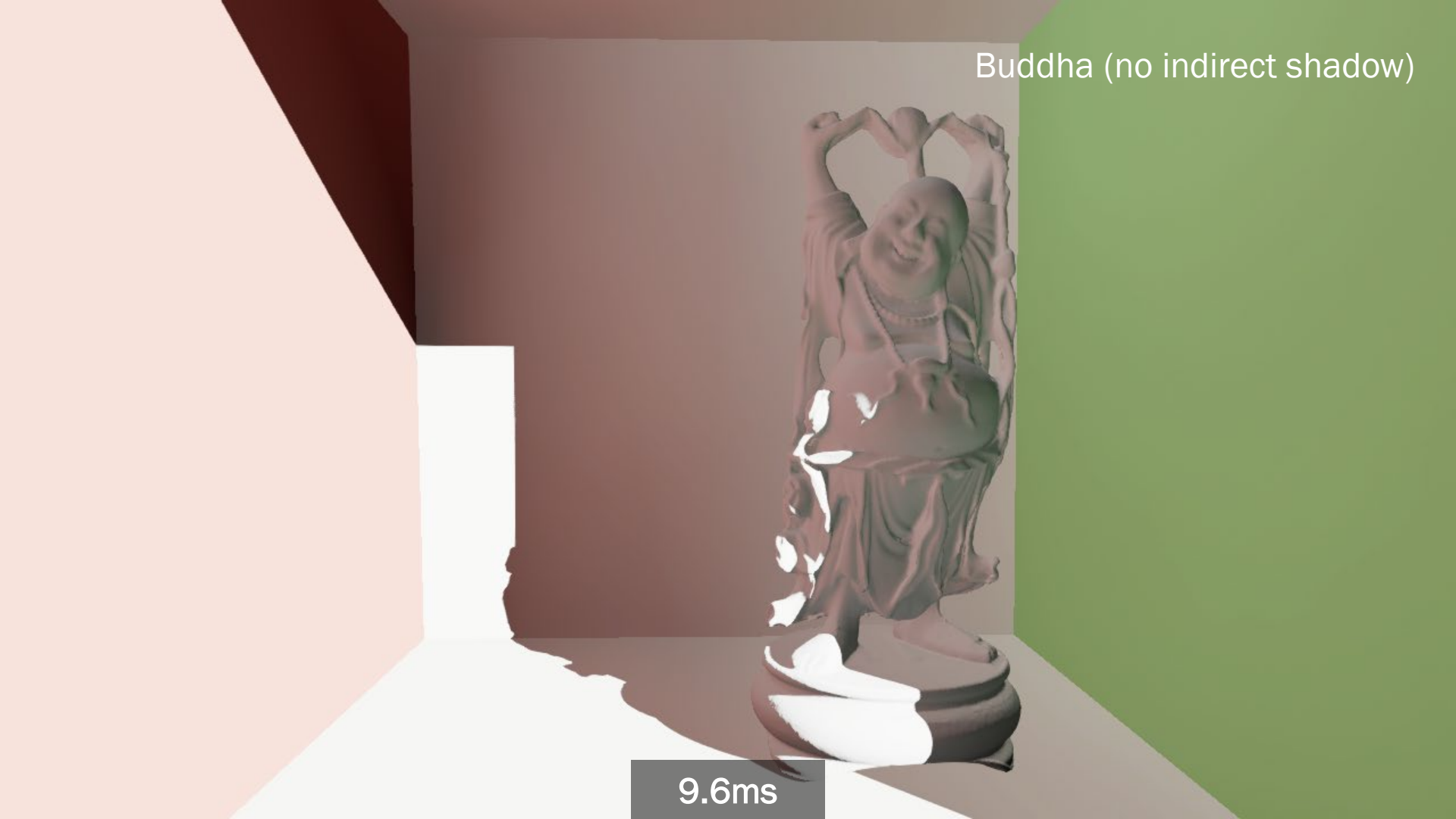

Crytek Sponza (1 shadow sample)
18.4ms

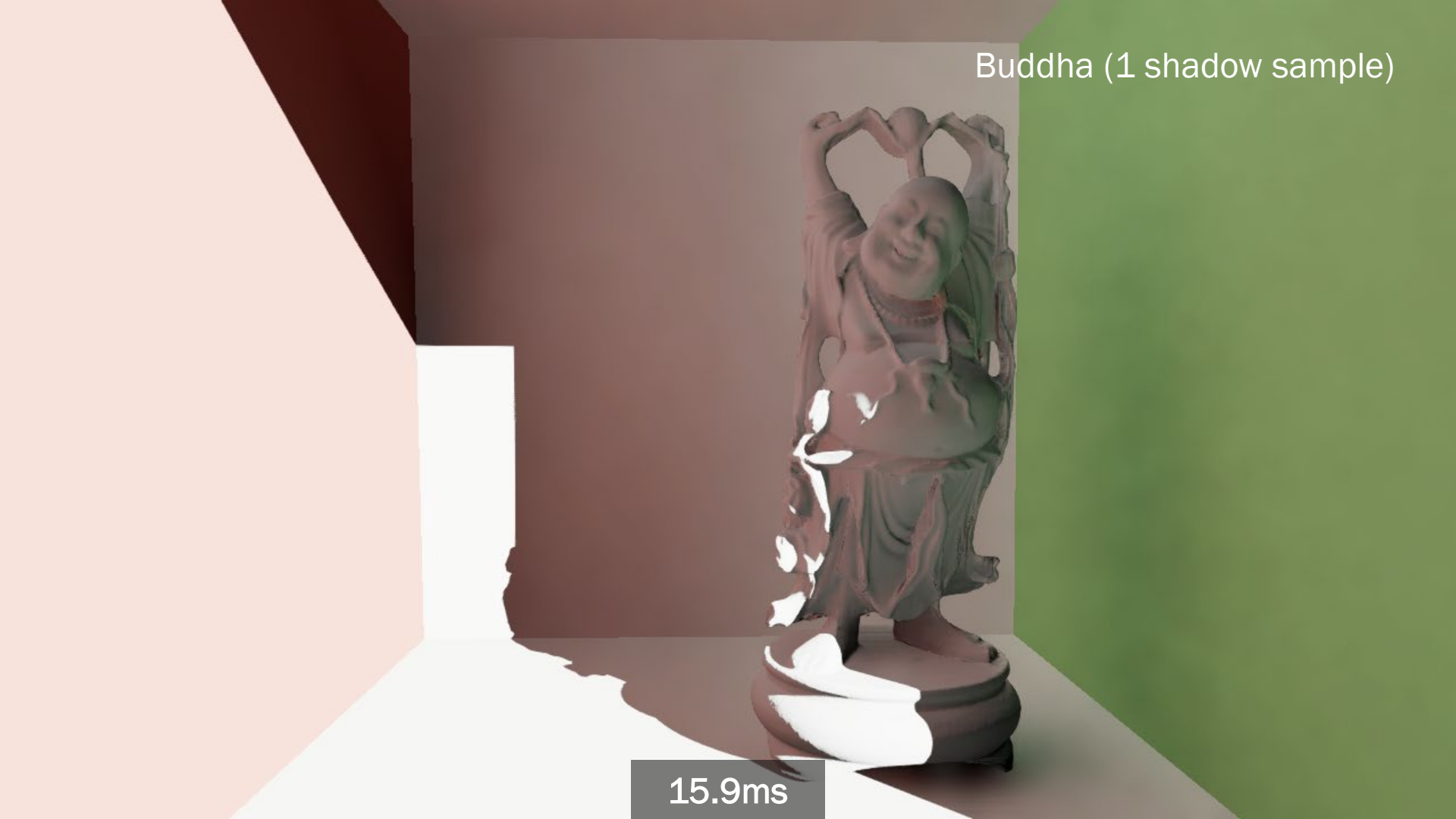

Crytek Sponza (4 shadow samples)
28.5ms

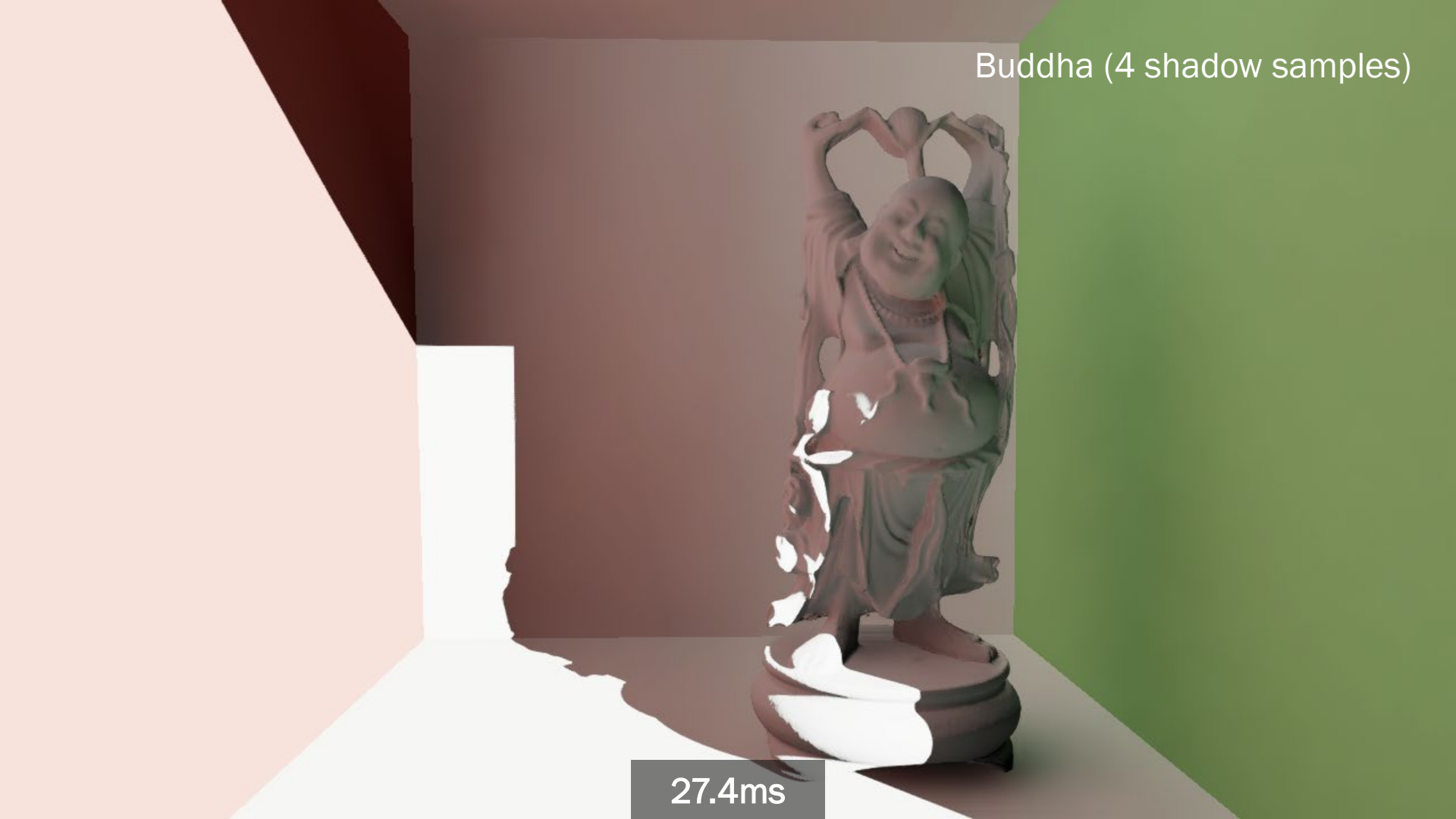

Crytek Sponza (1 shadow sample)
18.4ms

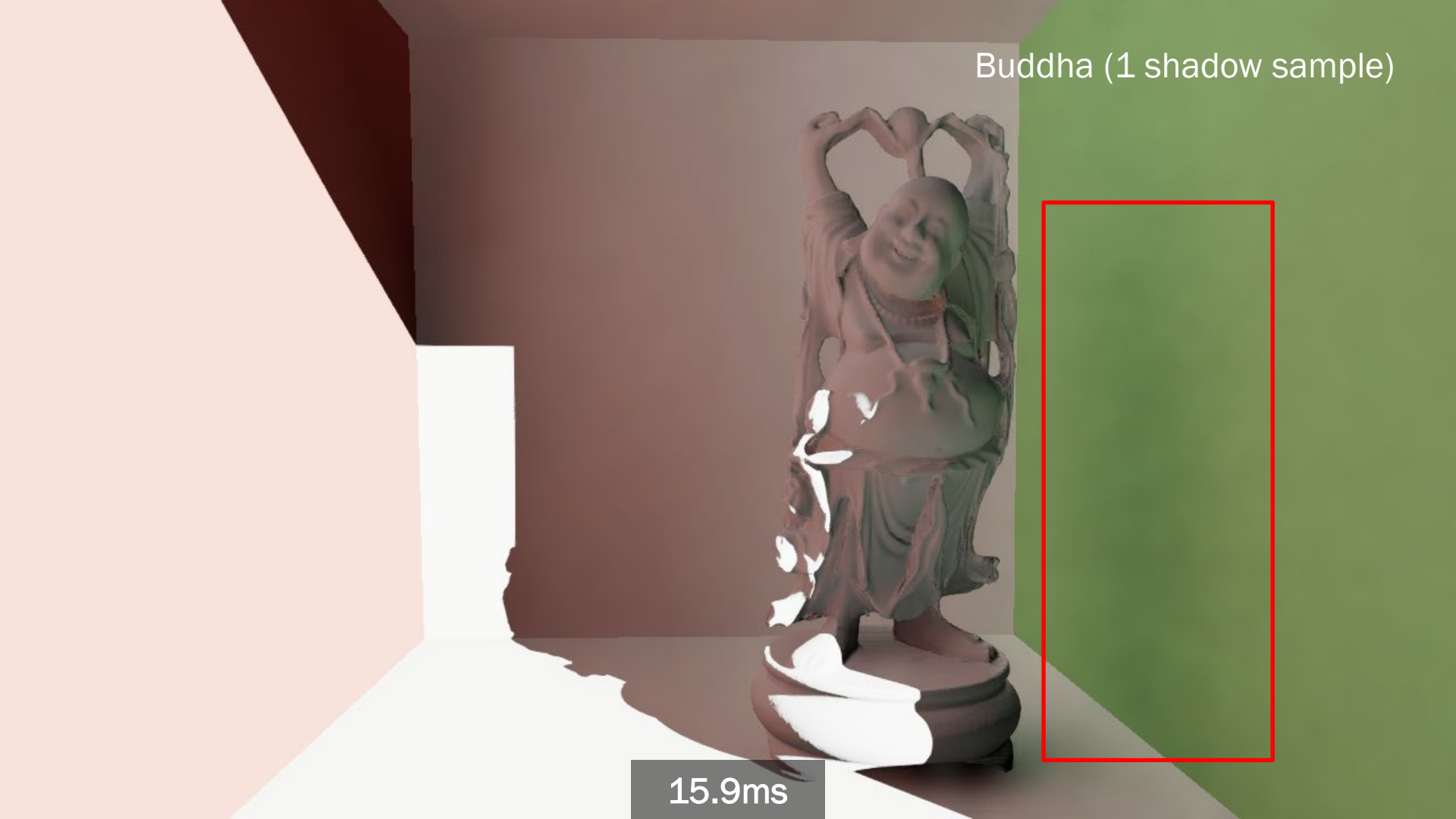

Crytek Sponza (4 shadow samples)
28.5ms

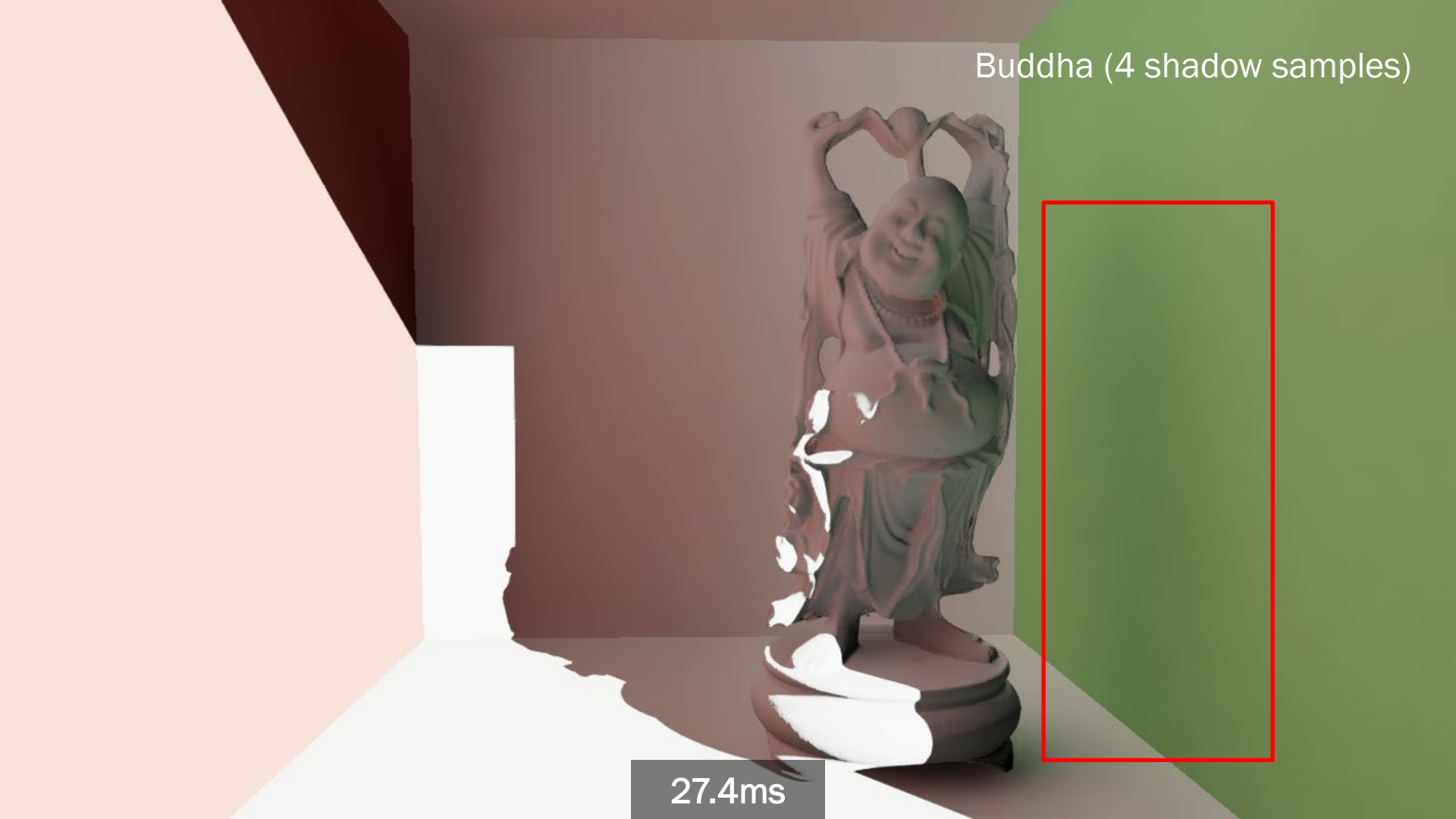

Buddha (direct lighting)
2.5ms


Buddha (no indirect shadow)
9.6ms


Buddha (1 shadow sample)
15.9ms


Buddha (4 shadow samples)
27.4ms

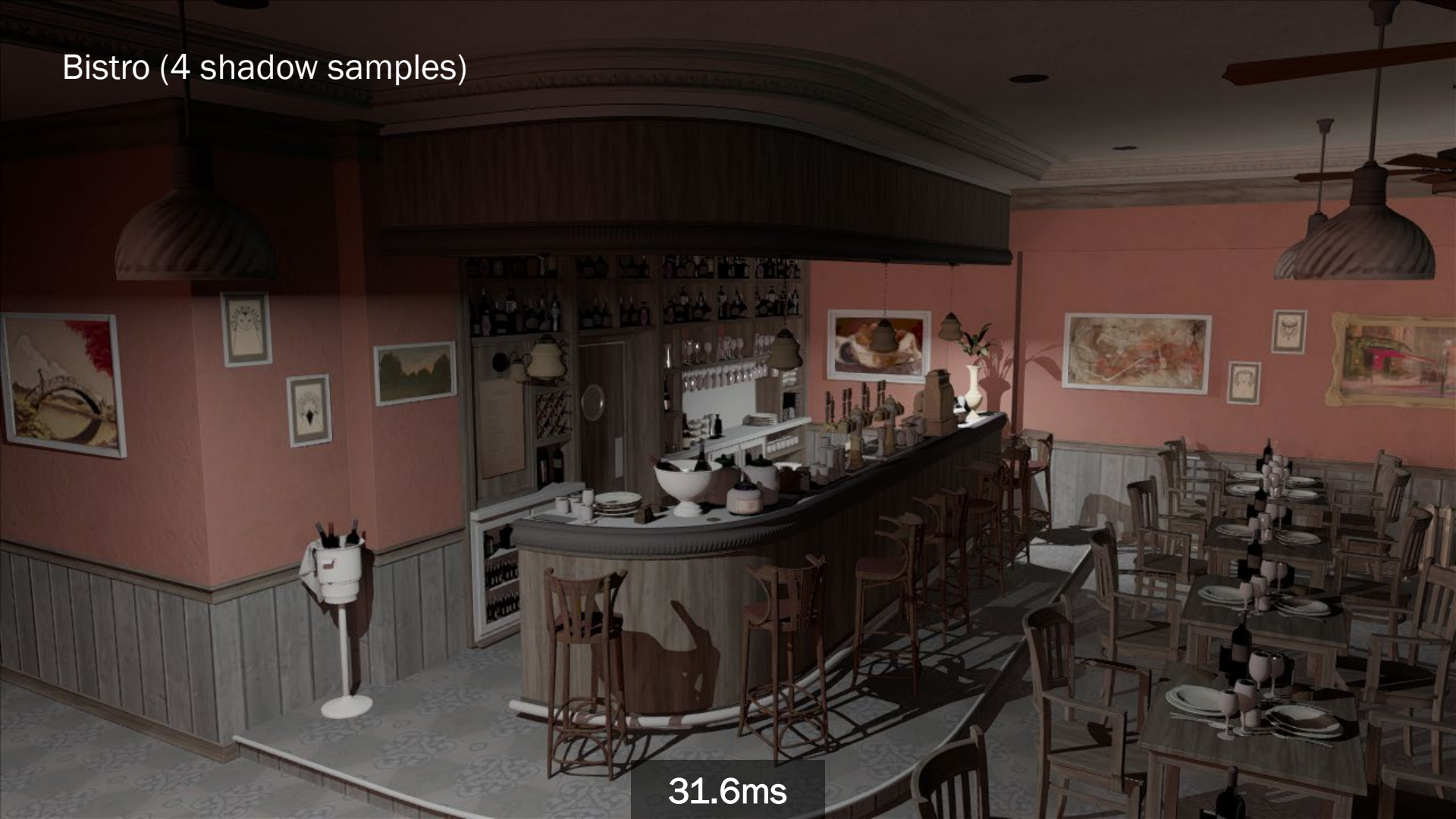

Buddha (1 shadow sample)
15.9ms


Buddha (4 shadow samples)
27.4ms

Bistro (direct lighting)


Bistro (no indirect shadow)
10.9ms


Bistro (1 shadow sample)
19.0ms


Bistro (4 shadow samples)
31.6ms

Bistro (1 shadow sample)

19.0ms

Bistro (4 shadow samples)

31.6ms


San Miguel (direct lighting)
16.1ms


San Miguel (no indirect shadow)
28.0ms


San Miguel (1 shadow sample)
36.8ms

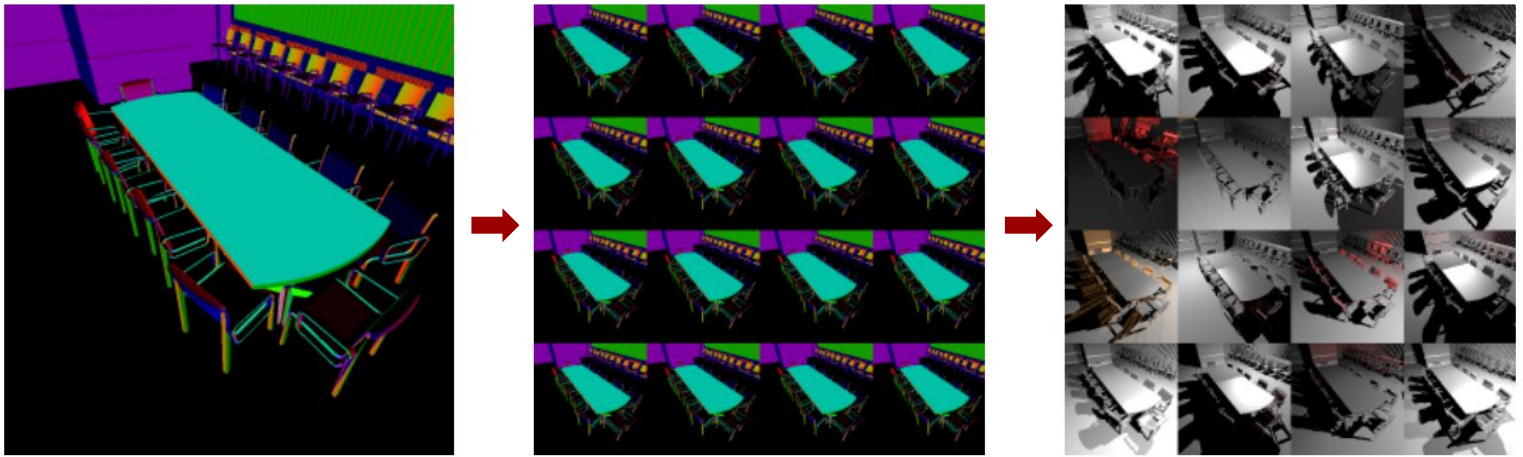

San Miguel (4 shadow samples)
45.6ms


San Miguel (1 shadow sample)
36.8ms


San Miguel (4 shadow samples)
45.6ms

Interleaved Sampling

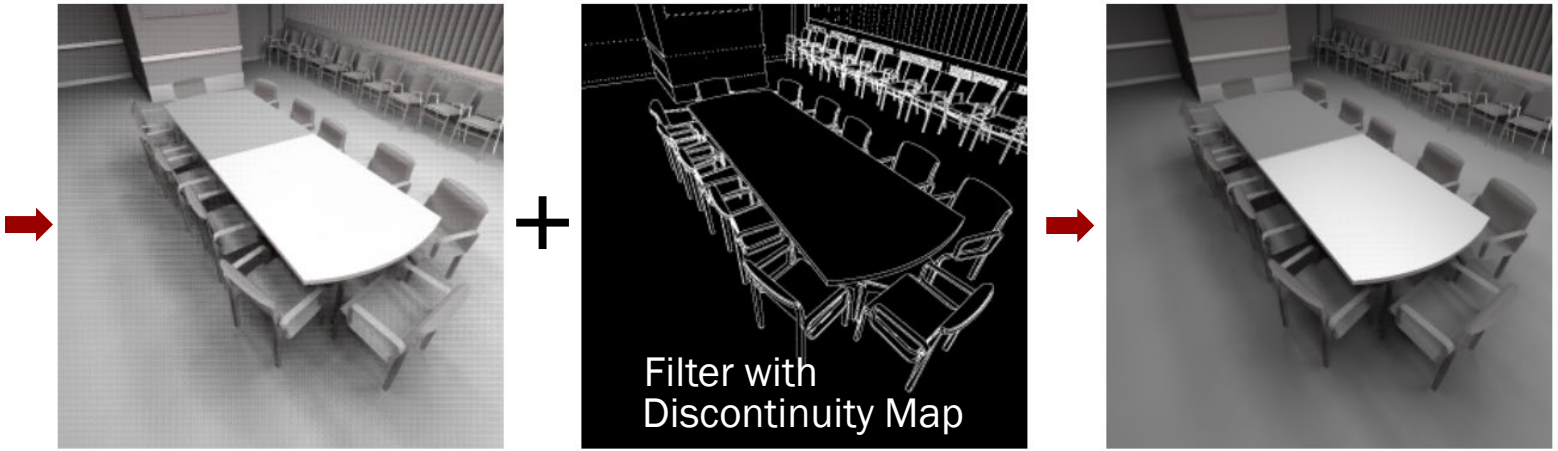
[Keller and Heidrich 2001, Wald et al. 2002, Segovia et al. 2006b]



[Segovia et al. 2006b]

Interleaved Sampling

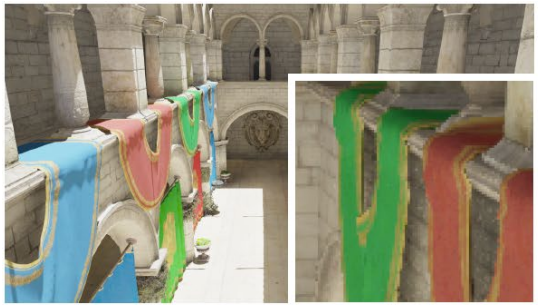
[Keller and Heidrich 2001, Wald et al. 2002, Segovia et al. 2006b]



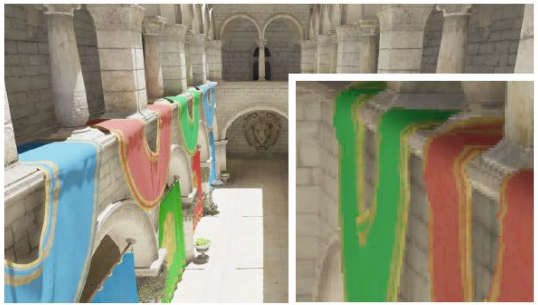
[Segovia et al. 2006b]

Interleaved Sampling

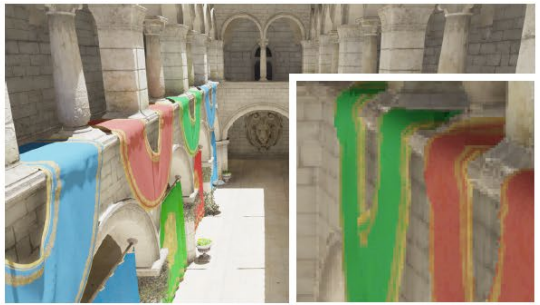
[Keller and Heidrich 2001, Wald et al. 2002, Segovia et al. 2006b]



(a) *No interleaved sampling*
Render time: 30.2 ms



(b) 2×2 *interleaved sampling*
Render time: 15.2 ms

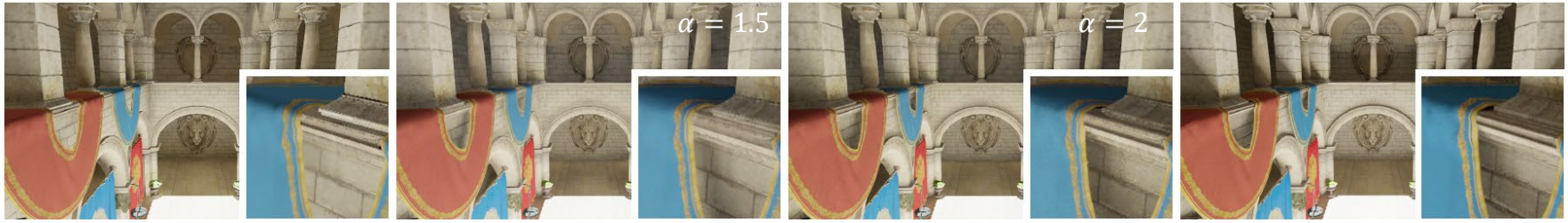


(c) 4×4 *interleaved sampling*
Render time: 10.1 ms

Settings:
Crytek Sponza at 1280 x 720 ($\alpha = 1$), 100K VPLs

Comparison

- We compare the results of our method to the NVIDIA VXGI implementation of voxel cone tracing [Crassin et al. 2011]



(a) *VXGI*
Render time: 21 ms

(b) 4×4 *interleaved samp.*
Render time: 15 ms

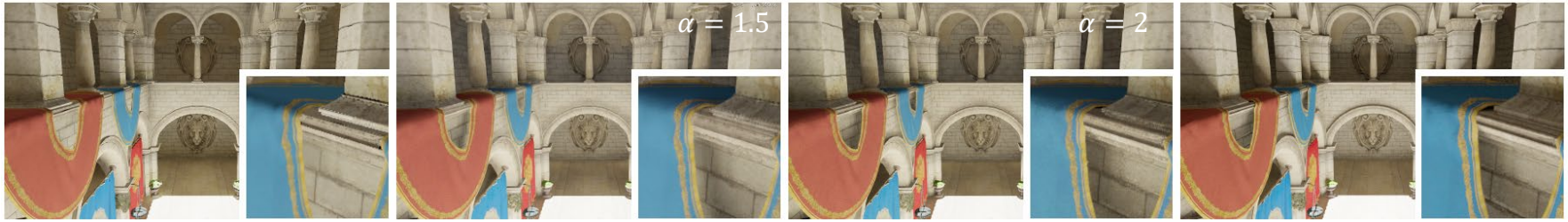
(c) *No interleaved samp.*
Render time: 93 ms

(d) *Path Tracing Reference*
Render time: 2 hours

Our Method

Comparison

- We compare the results of our method to the NVIDIA VXGI implementation of voxel cone tracing [Crassin et al. 2011]



(a) *VXGI*
Render time: 21 ms

(b) 4 × 4 *interleaved samp.*
Render time: 15 ms

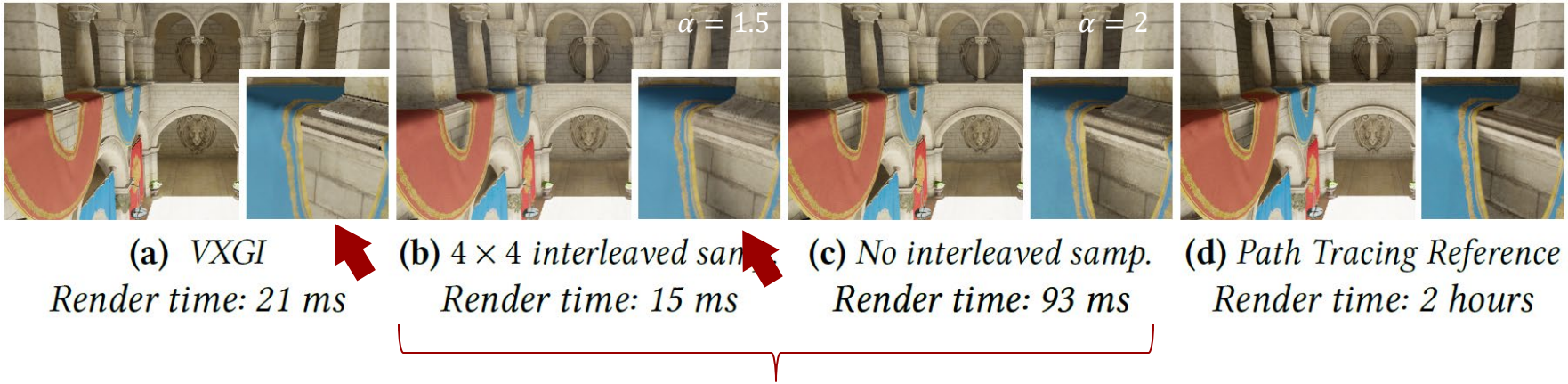
(c) *No interleaved samp.*
Render time: 93 ms

(d) *Path Tracing Reference*
Render time: 2 hours

Our Method

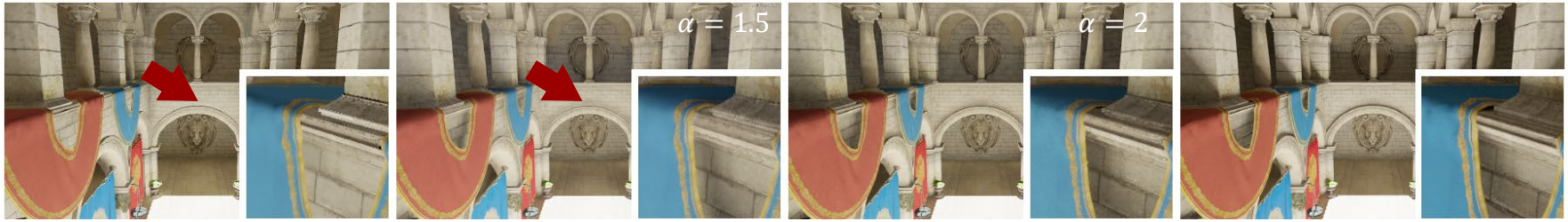
Comparison

- We compare the results of our method to the NVIDIA VXGI implementation of voxel cone tracing [Crassin et al. 2011]



Comparison

- We compare the results of our method to the NVIDIA VXGI implementation of voxel cone tracing [Crassin et al. 2011]



Conclusion

- We provide a real-time many-lights solution
- Our method can be used for real-time global illumination with VPLs

- Future Work: dynamic hierarchy update

Thank you

Paper, code and video available on:

- https://dqlin.xyz/pubs/2019-i3d-LGH/

Email: daqi@cs.utah.edu

Real-Time Rendering with Lighting Grid Hierarchy

I3D 2019