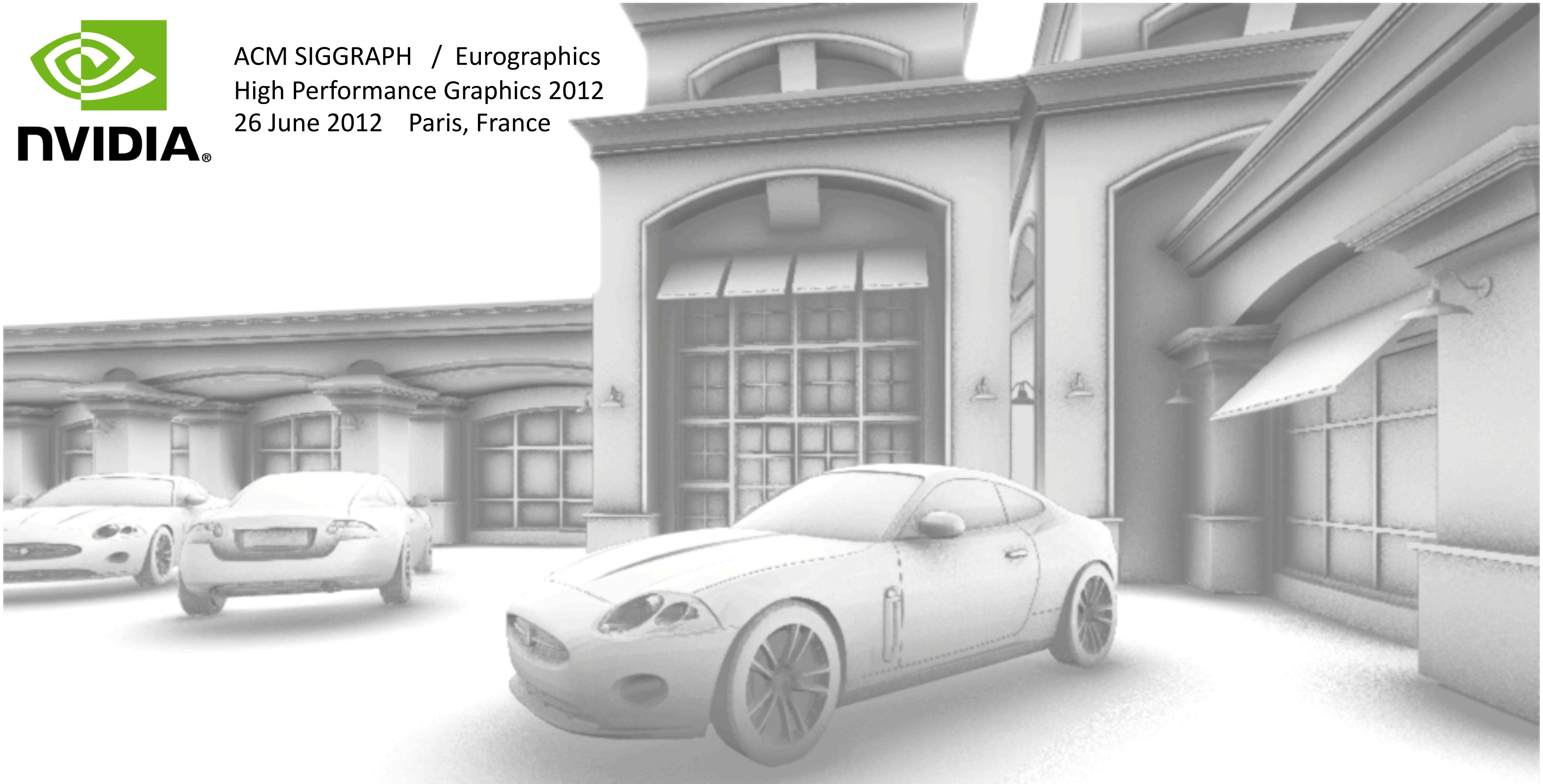
ACM SIGGRAPH / Eurographics
High Performance Graphics 2012
26 June 2012 Paris, France



Scalable Ambient Obscurance

Morgan McGuire
NVIDIA & Williams College

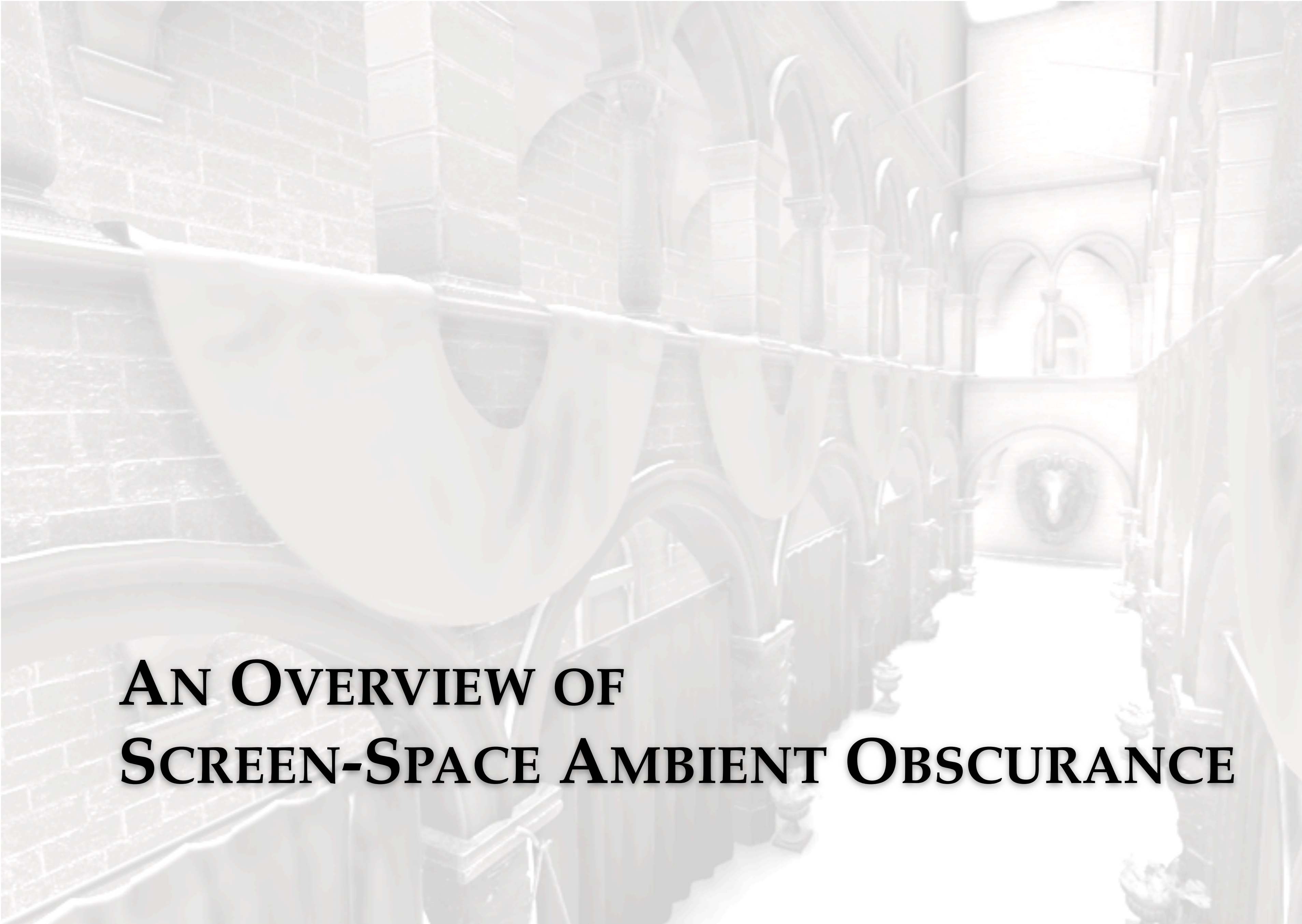
Michael Mara
NVIDIA

David Luebke
NVIDIA

in collaboration with

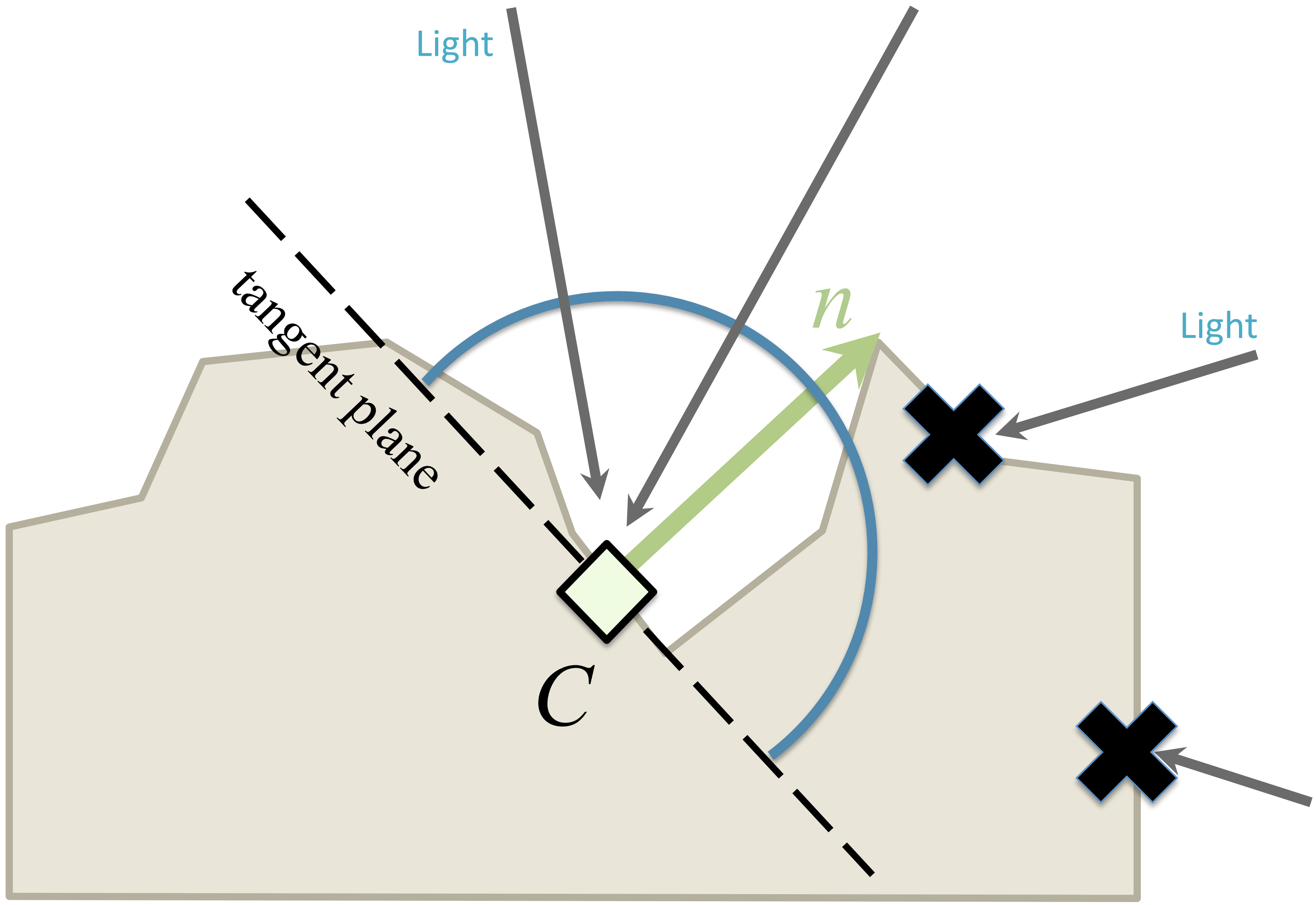
- Leonardo Zide (Treyarch)
- Naty Hoffman (Activision Studio Central)
- Padraic Hennessy, Brian Osman, and Michael Bukowski (Vicarious Visions)
- Louis Bavoil (NVIDIA)

Thanks to Peter-Pike Sloan (NVIDIA), Eric Haines (Autodesk, Inc.), and Guedis Cardenas (Williams)

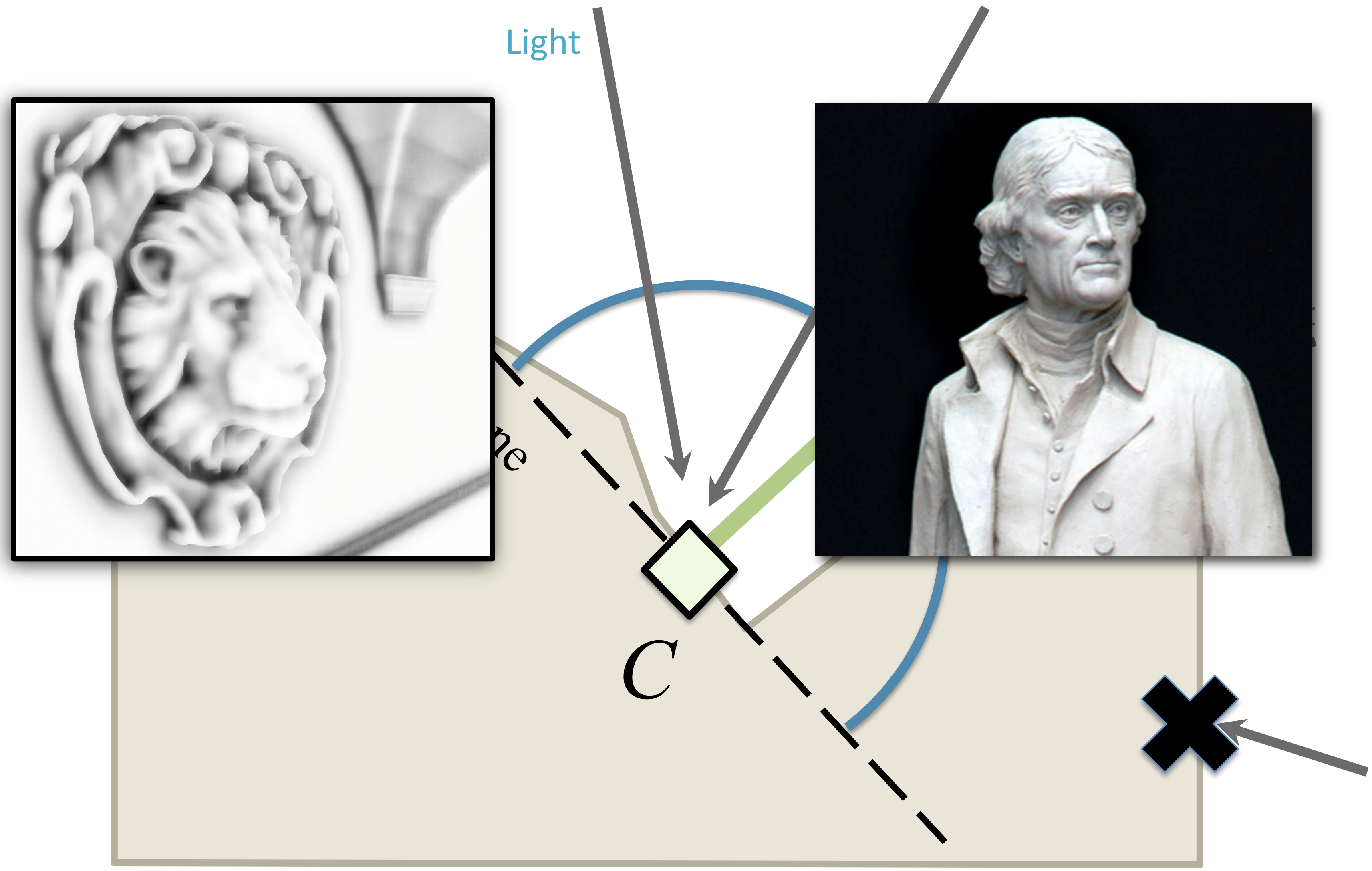


An Overview of Screen-Space Ambient Obscurance

Ambient Obscurance (AO)

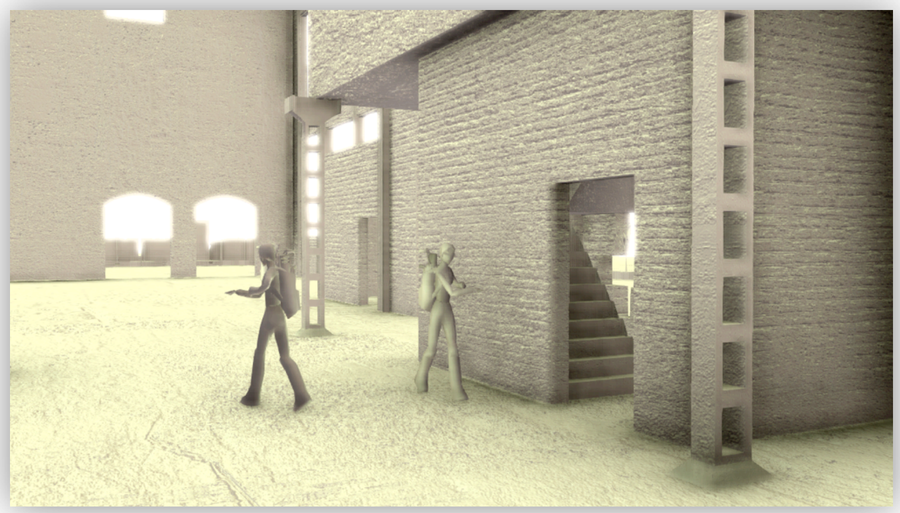


Ambient Obscurance (AO)

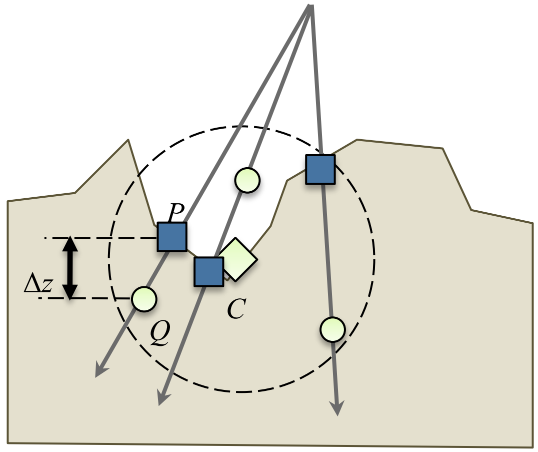


Production Issues

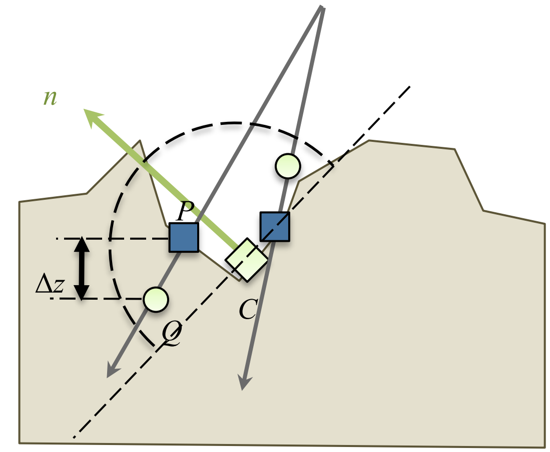
- **AO is a shadowing term ➔ proximity + contact cue**
- Dynamic illumination is cost-effective
- Avoid custom or complex input and output buffers that constrain the effects pipeline
- Real-time constraints: *worst case* performance must be 1-3 ms. *Average case* is not interesting!



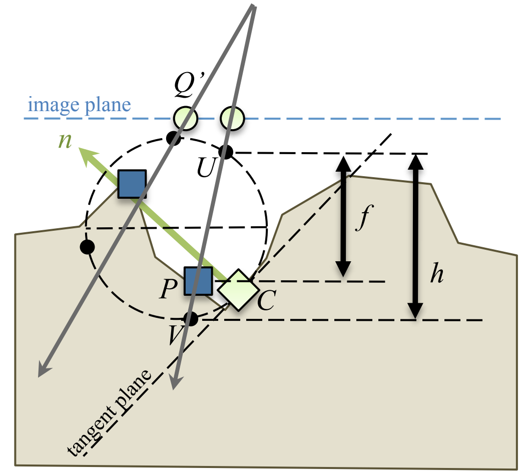
Screen-Space Methods



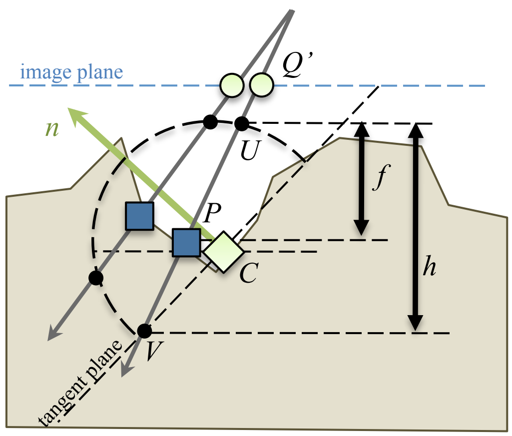
Kajalin [09] & Mittring [06]
(Crytek SSAO)



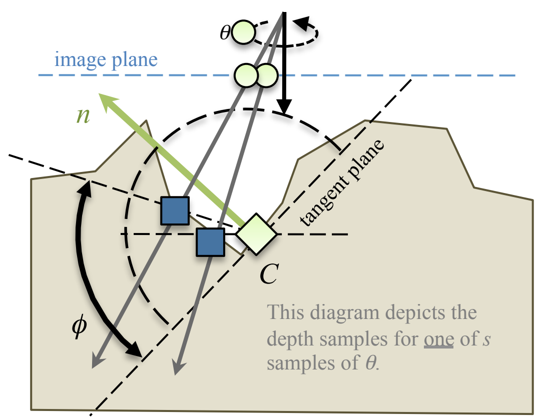
Filion and McNaughton [08]
(StarCraft II AO)



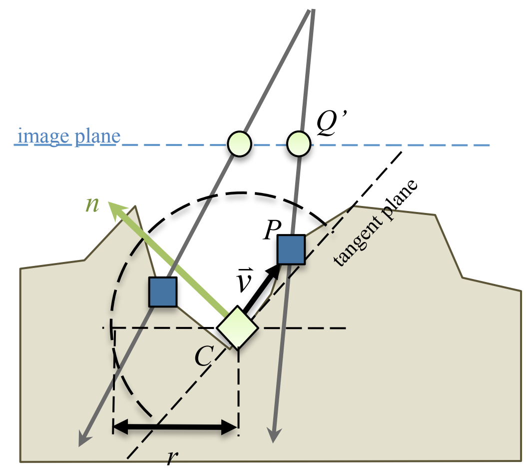
Szirmay-Kalos et al. [09, 10]
(Volumetric AO)



Loos and Sloan [10]
Volumetric Obscurance



Bavoil and Sainz [08,09]
(Horizon-Based AO)



McGuire et al. [11]
(AlchemyAO)

Read from depth buffer
Sample point
Sampled eye ray
Geometric constructions

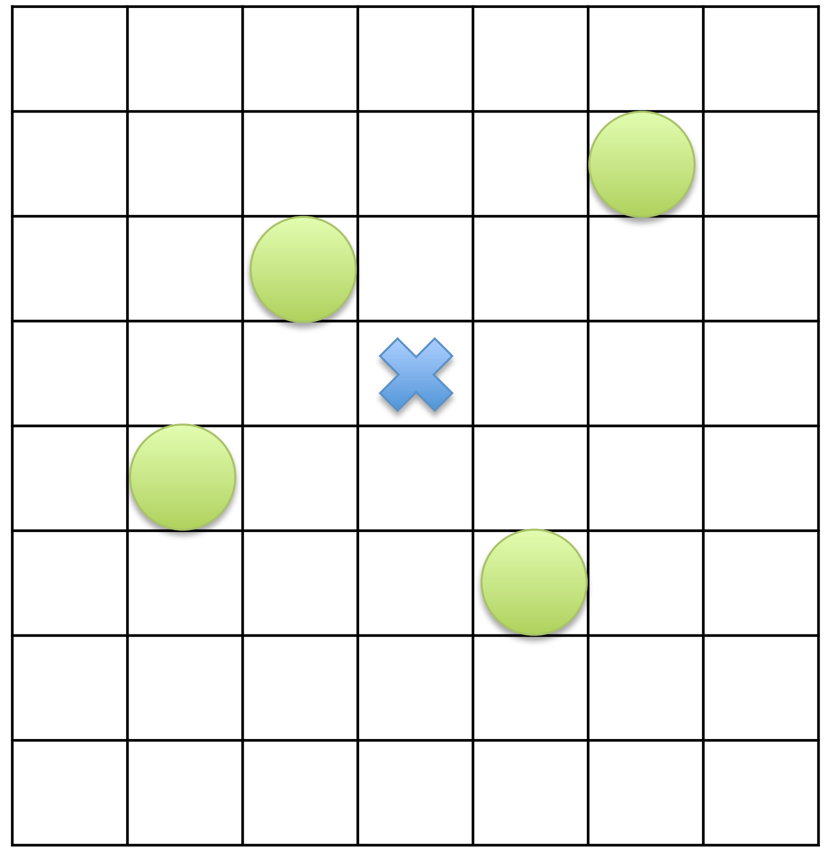
(x_P', y_P') is the screen-space position of camera-space point (x_P, y_P, z_P), which has depth function $z(x_P', y_P') = z_P$

Screen-Space AO Framework

Three full-screen passes:

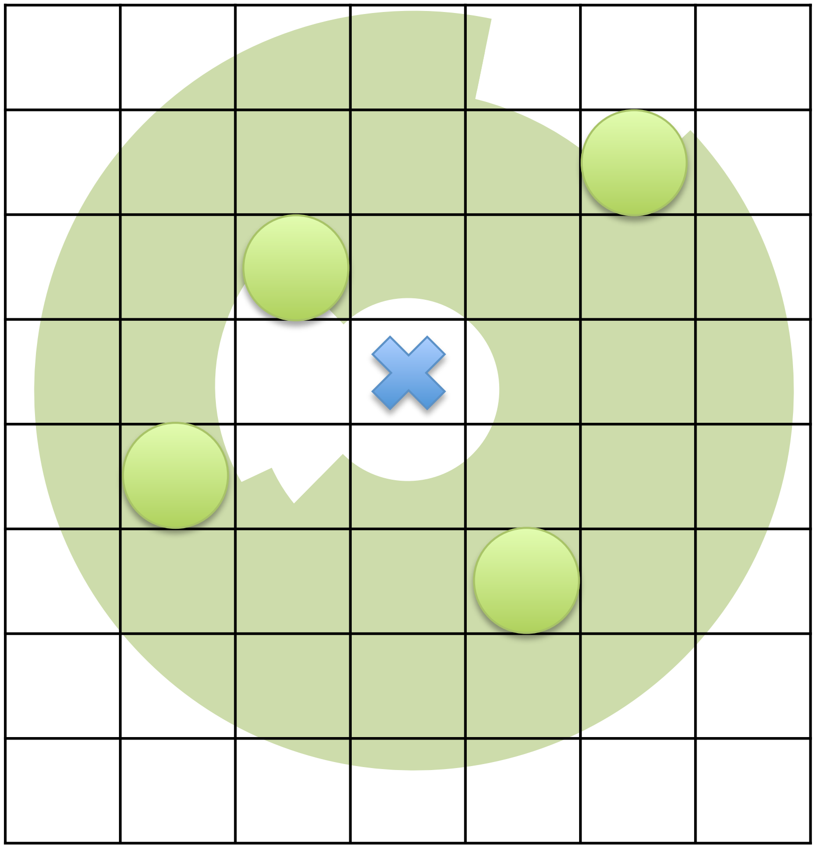
1. **Raw AO** samples:
    1. Sample nearby position or depth, and normals
    2. Estimate occlusion from a patch at the sampled location
2. Horizontal bilateral **blur**
3. Vertical bilateral **blur**

Scale diffuse ambient by the result during shading



Sampling Pattern

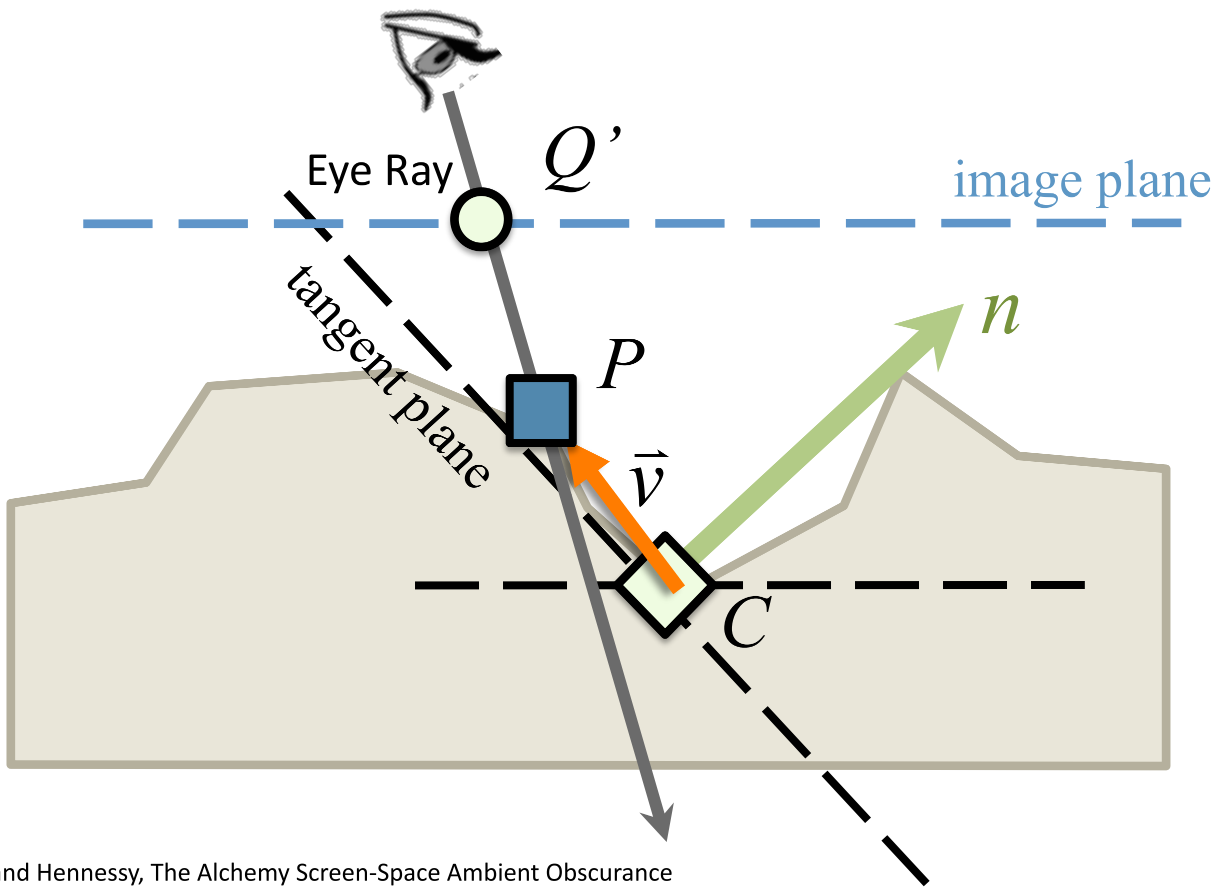
- Fixed screen-space pattern on a disk
- Rotate at each pixel randomly



- Fixed screen-space pattern on a disk
- Rotate at each pixel randomly

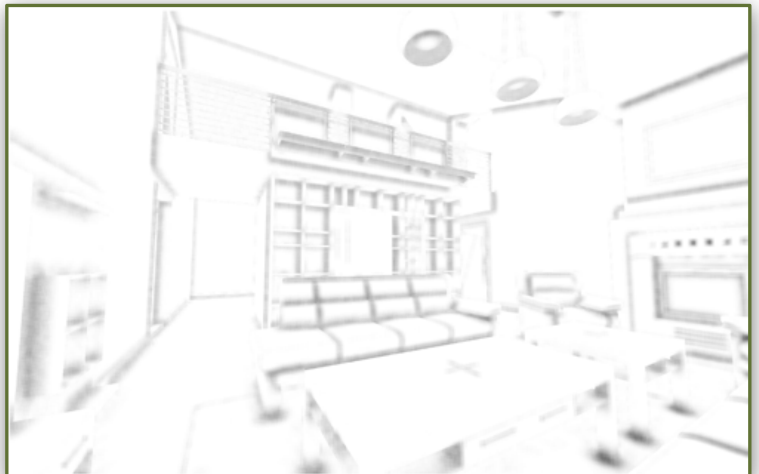
Alchemy AO

$$A \approx \max\left(0, 1 - \frac{2\sigma}{s} \cdot \sum_{i=1}^{s} \frac{\max(0, \vec{v_i} \cdot \hat{n} + z_C \beta)}{\vec{v_i} \cdot \vec{v_i} + \epsilon}\right)^k$$

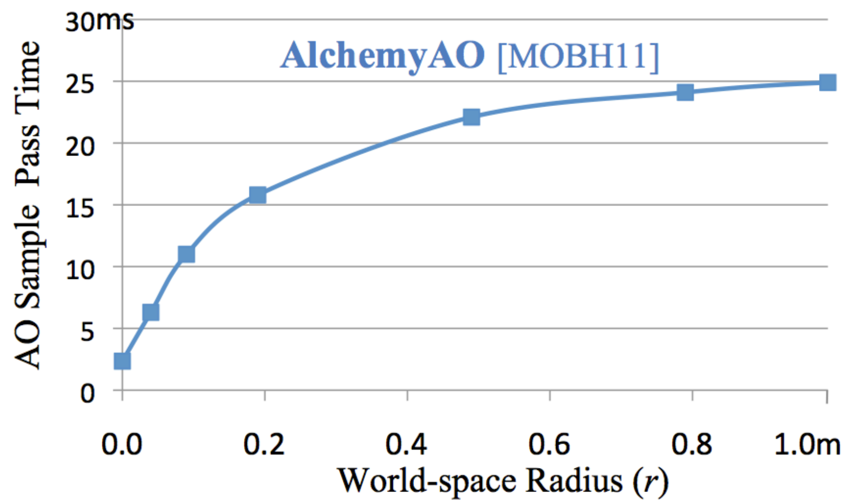


McGuire, Osman, Bukowski, and Hennessy, The Alchemy Screen-Space Ambient Obscurance Algorithm, HPG 12

Alchemy AO Results

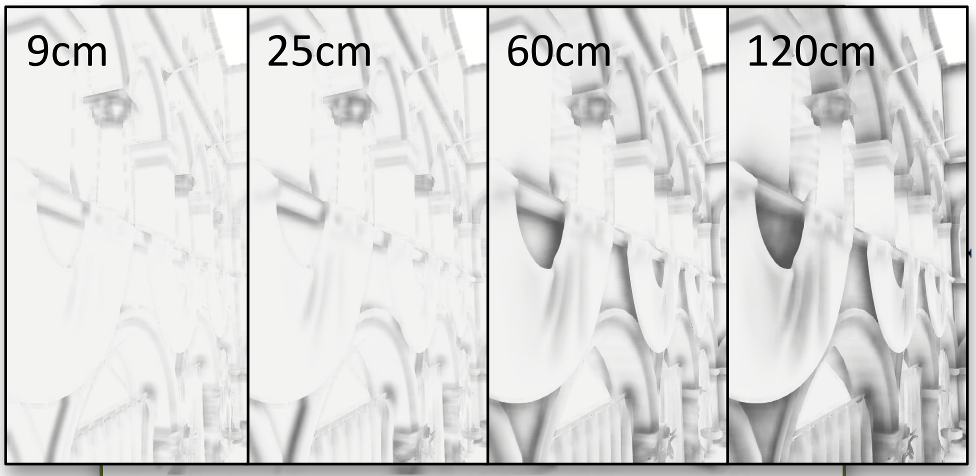


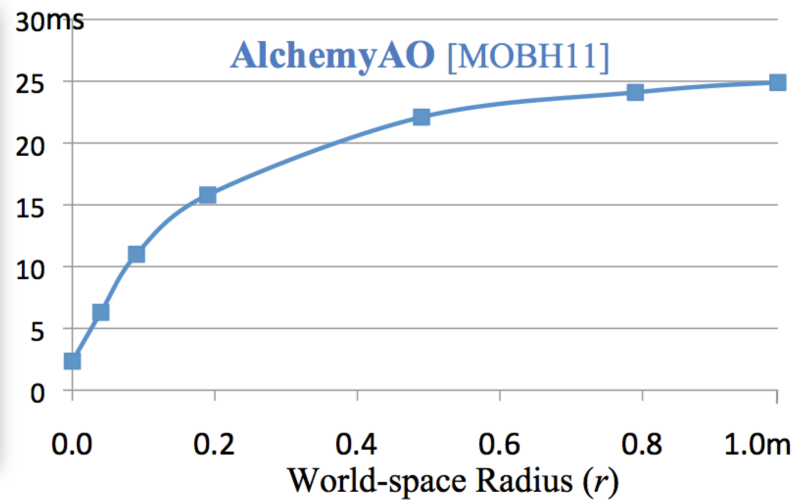
Good Quality
1280 x 720, *r* = 15 cm
3 ms on GeForce GTX 680
(faster with smaller radius)



Poor Scaling
Measured at 2650x1600
GeForce GTX 680

Alchemy AO Results





Good Quality
1280 x 720, *r* = 15 cm
3 ms on GeForce GTX 680
(faster with smaller radius)

Poor Scaling
Measured at 2650x1600
GeForce GTX 680

Scalable Ambient Obscurance (SAO)

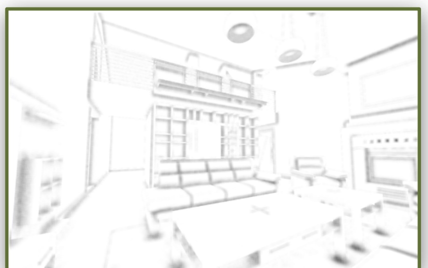
Strategy

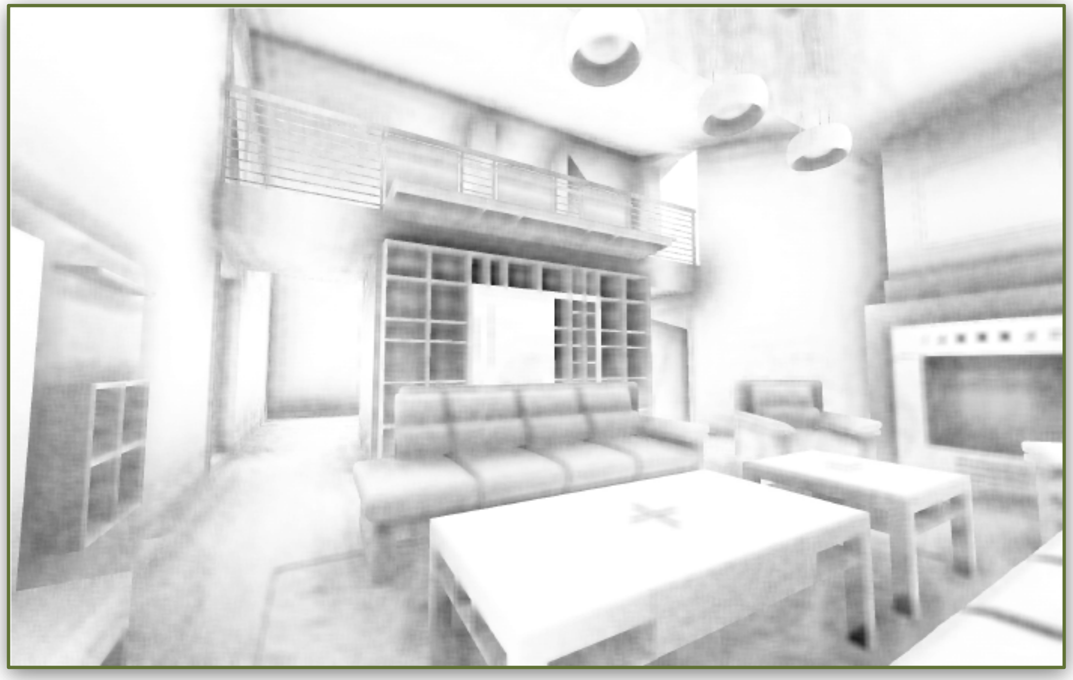
- Start with AlchemyAO from last year
  - Keep the math, change the implementation
- **Integration**:
  - Tune and then hard-code constants
  - Reduce input to a standard depth buffer
- **Performance**:
  - **Low-level** optimizations for constant factor speedup
  - **Algorithmic** optimizations for perfect scaling

SAO Properties

- **Fast**
  - 7x throughput of HPG11 result at comparable quality
  - Tuned for DX11 GPUs (e.g., GeForce GTX 680)
- **Scalable Performance**
  - Independent of pixel density
  - Independent of world-space sampling radius
- **Easy to integrate**
  - Input: projection matrix, sample radius, **depth buffer**
  - Output: occlusion texture map
  - (Combines with screen-space shadow filtering)
  - HLSL, GLSL, and C++ source at http://research.nvidia.com/publication/scalable-ambient-obscurance

Visual Impact





Alchemy AO HPG11
1280 x 720
r = 15 cm
3 ms on GeForce GTX 680

Scalable AO HPG12
2650 x 1600 + guardband
r = 150 cm
3 ms on GeForce GTX 680

Low-Level Optimizations

Reconstructing Position and Normal

depth buffer value on [0, 1]

constants from clip planes

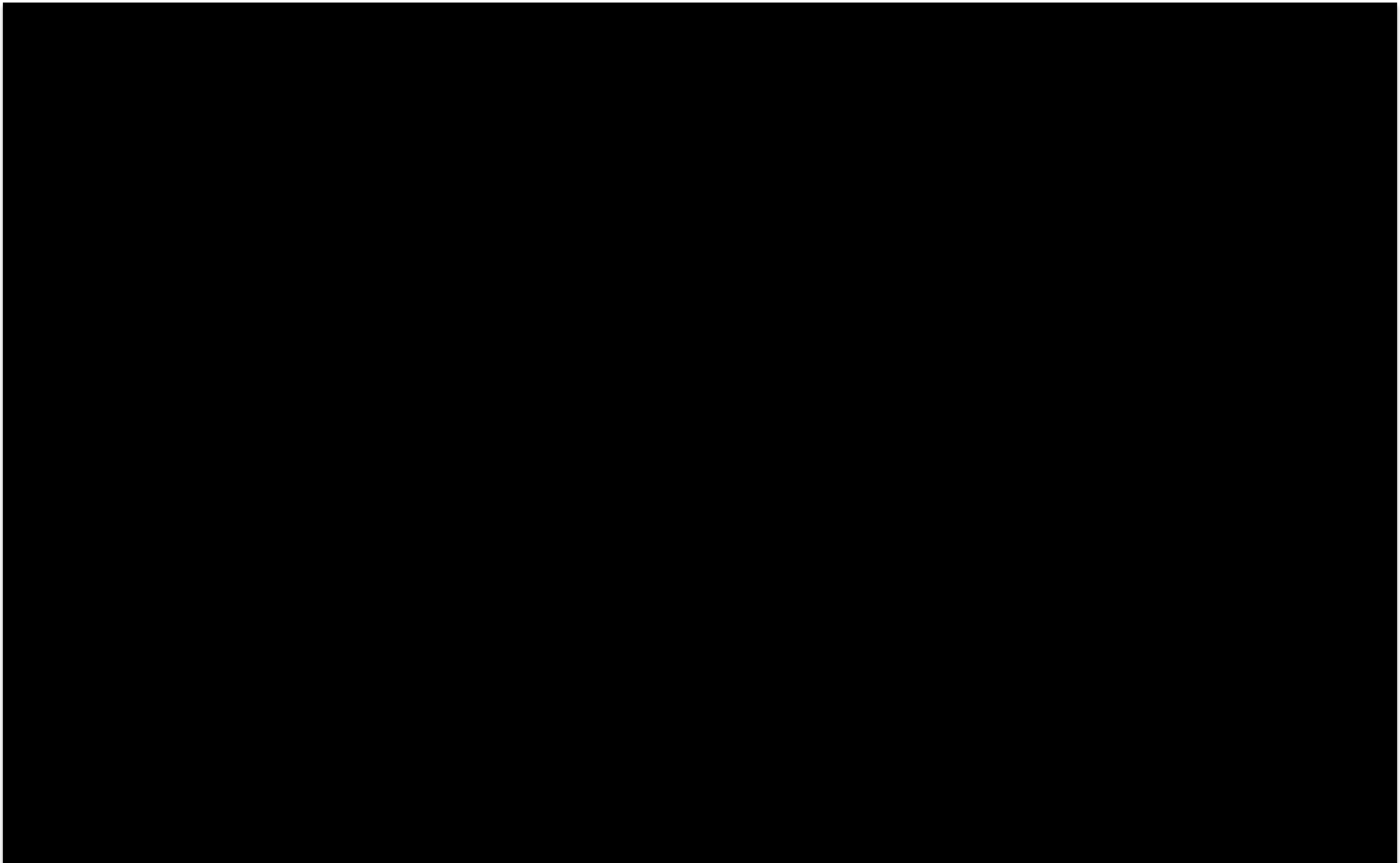
$$z(d) = \frac{\mathbf{c}_0}{d \cdot \mathbf{c}_1 + \mathbf{c}_2}$$

$$(x_C, y_C) = z_C \cdot \left(\frac{1 - \mathbf{P}_{0,2}}{\mathbf{P}_{0,0}} - \frac{2(x' + \frac{1}{2})}{w \cdot \mathbf{P}_{0,0}}, \frac{1 + \mathbf{P}_{1,2}}{\mathbf{P}_{1,1}} - \frac{-2(y' + \frac{1}{2})}{h \cdot \mathbf{P}_{1,1}} \right)$$

$$\hat{n}_C = \text{normalize}\left(\frac{\partial C}{\partial y'} \times \frac{\partial C}{\partial x'} \right)$$

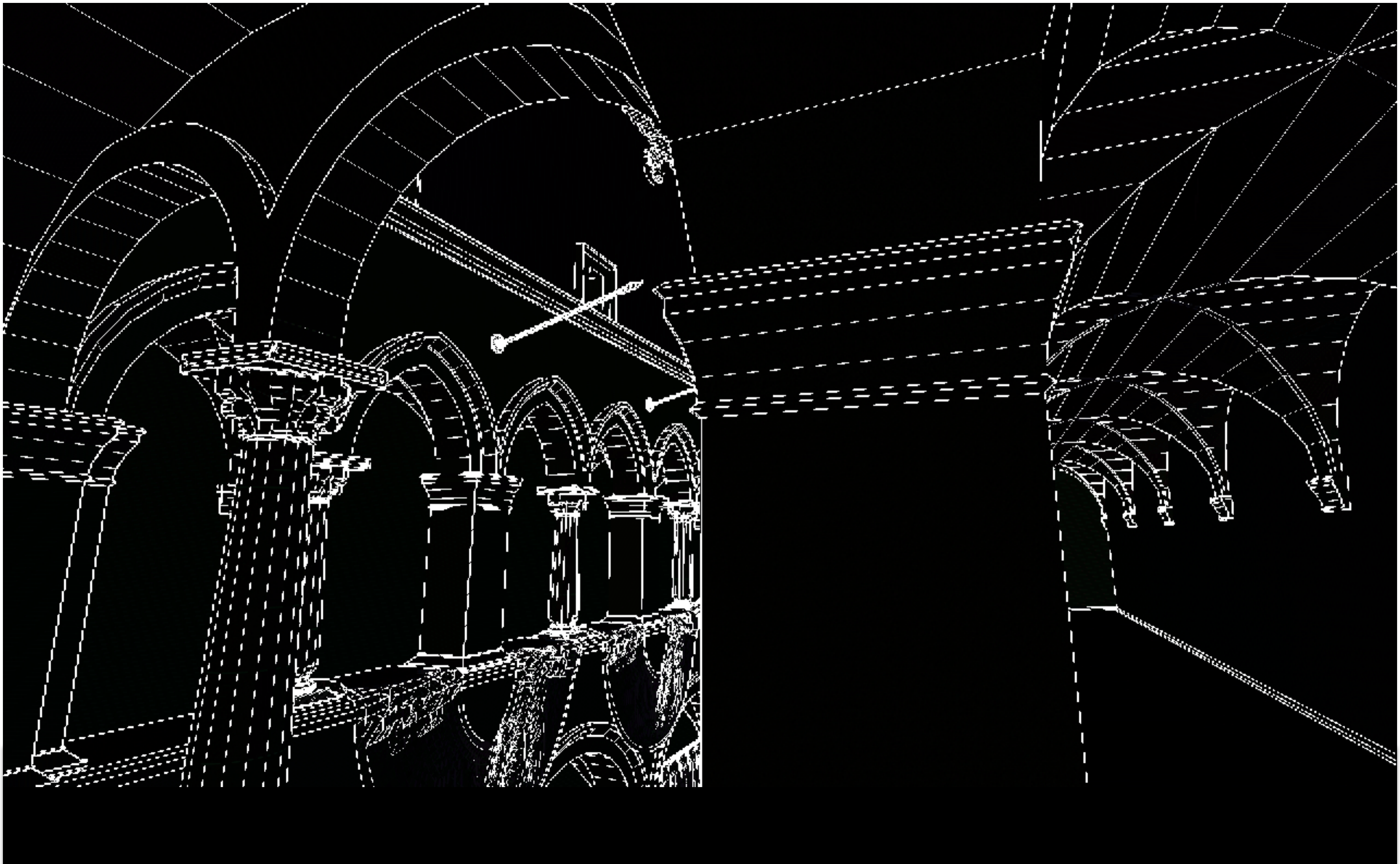
See the paper for details on maintaining precision.

Depth Error x 10

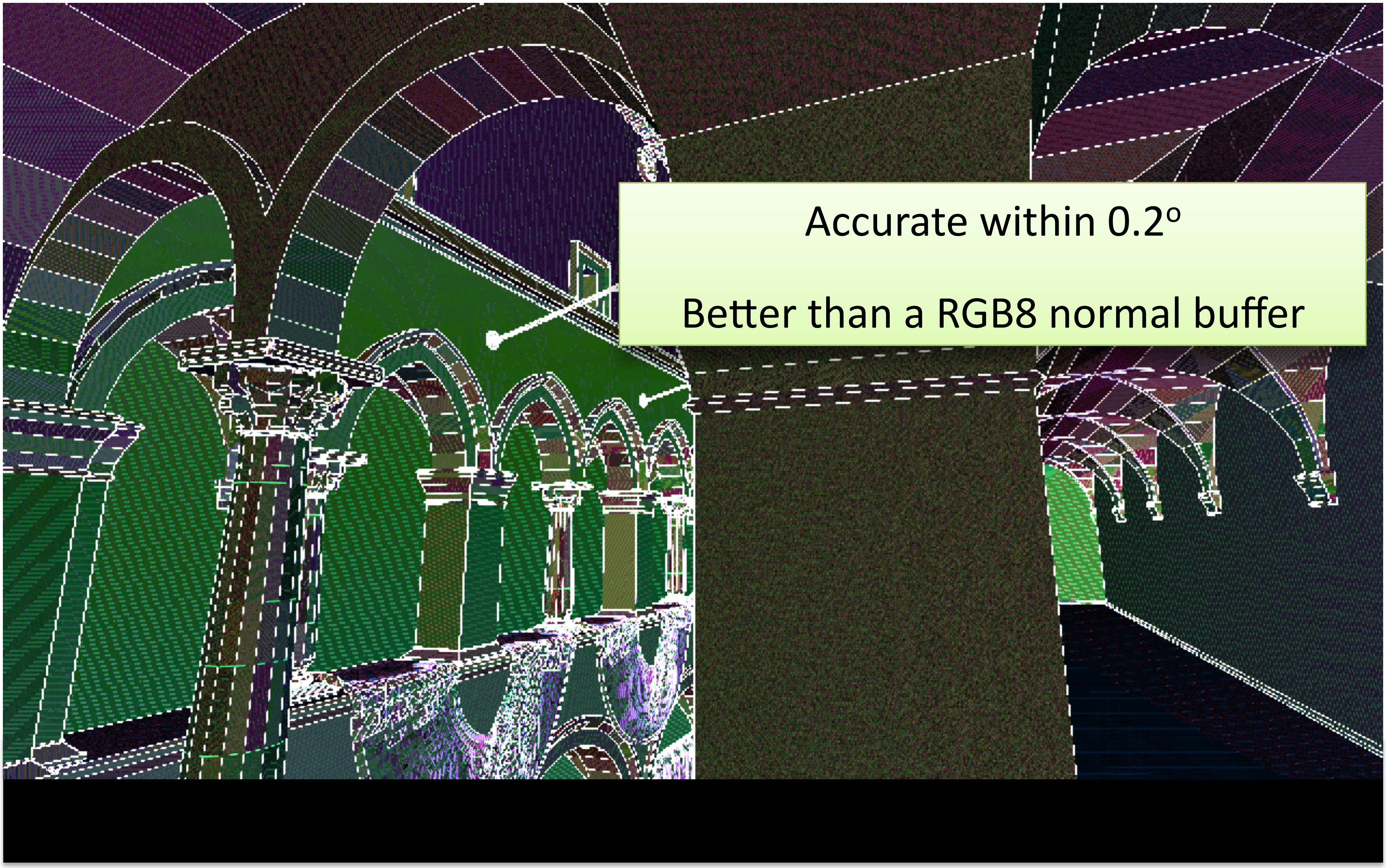


Depth Error x 1000

Normal Error x 10



Normal Error x 100



58% Memory Traffic Reduction

Alchemy AO HPG11

AO	Format	Bytes/pix
12 positions	RGB32F	12 x 12
1 normal	RGB16F	6
1 AO out	R8	1
H + V Blur		
26 depths	R32F	26 x 4
26 AO values	R8	26 x 1
2 AO out	R8	2
	TOTAL	**283**

Scalable AO HPG12

AO	Format	Bytes/pix
9 depths	R32F	9 x 4
1 AO + Z out	RGBA8	4
H + V Blur		
18 AO + Z	RGBA8	18 x 4
1 AO + Z out	RGBA8	4
1 AO out	R8	1
	TOTAL	**117**

58% Memory Traffic Reduction

Reconstructing position reduces per-sample bandwidth by 66%

Alchemy

AO	Format	Bytes/pix
12 positions	RGB32F	12 x 12
1 normal	RGB16F	6
1 AO out	R8	1
H + V Blur		
26 depths	R32F	26 x 4
26 AO values	R8	26 x 1
2 AO out	R8	[illegible]
	TOTAL	[illegible]

PG12

AO	Format	Bytes/pix
9 depths	R32F	9 x 4
1 AO + Z out	RGBA8	4
H + V Blur		
18 AO + Z	RGBA8	18 x 4
1 AO + Z out	RGBA8	4
1 AO out	R8	1
	TOTAL	**117**

Packing AO + Z reduces fetch instruction count by 50% and bandwidth by 20%.

58% Memory Traffic Reduction

Alchemy AO HPG11

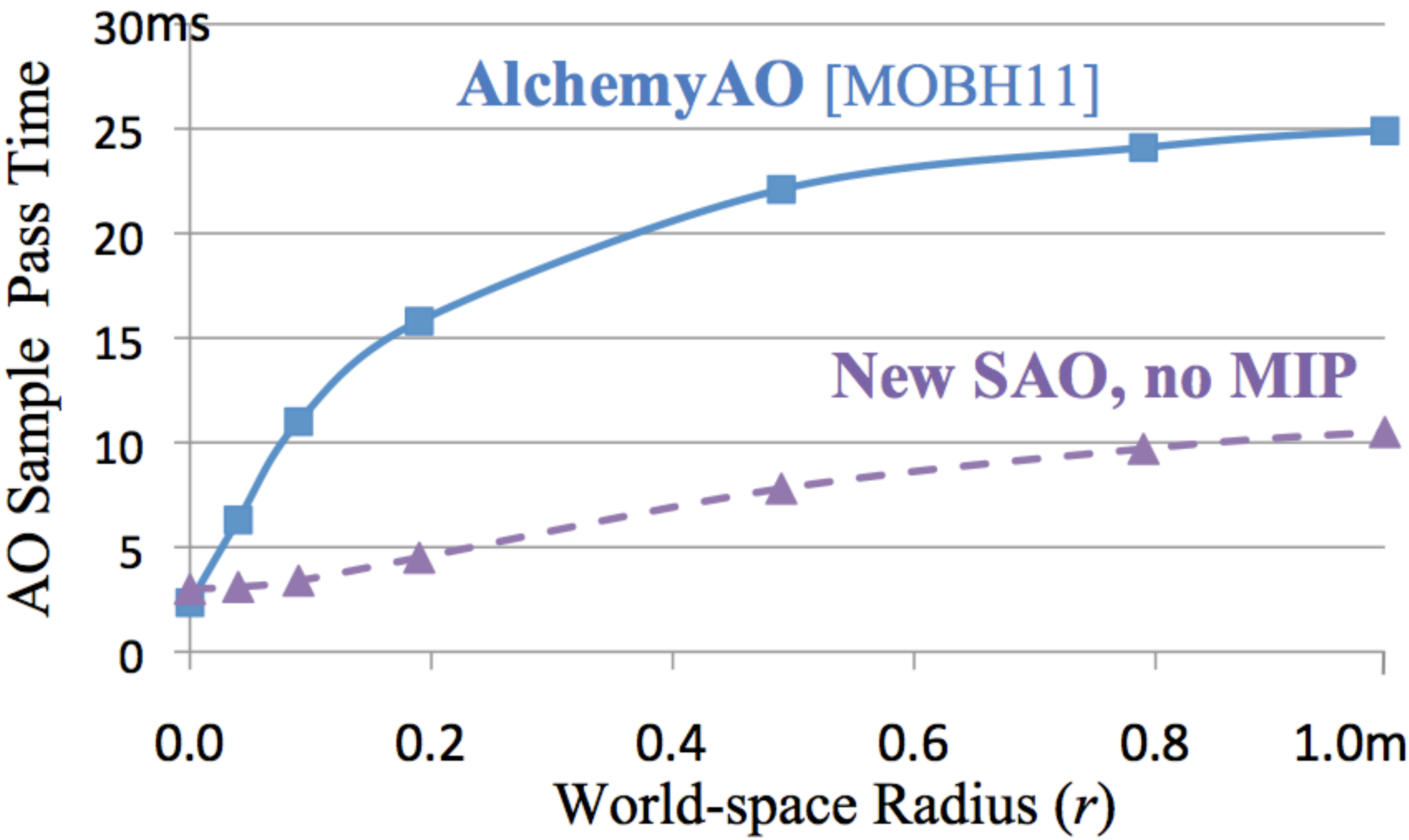
AO	Format	Bytes/pix
12 positions	RGB32F	12 x 12
1 normal	RGB16F	6
1 AO out	R8	1
H + V Blur		
26 depths	R32F	26 x 4
26 AO values	R8	26 x 1
2 AO out	R8	2
	TOTAL	**283**

Scalable AO HPG12

AO	Format	Bytes/pix
9 depths	R32F	9 x 4
1 AO + Z out	RGBA8	4
18 AO + Z	RGBA8	18 x 4
1 AO + Z out	RGBA8	4
1 AO out	R8	1
	TOTAL	**117**

Derivative instructions provide extra samples without main memory traffic.

Performance Impact



GeForce GTX 680, 2650x1600

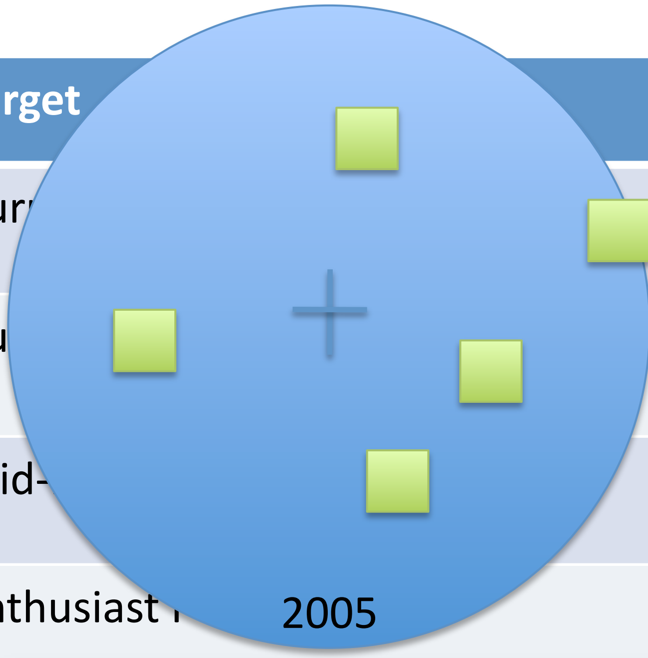
ALGORITHMIC OPTIMIZATION

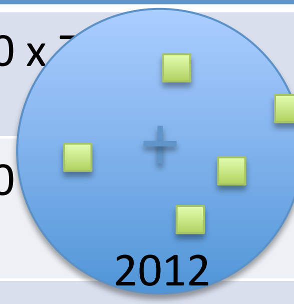
Single-Display Resolutions

Target	Resolution
Current Console (2005), 720p	1280 x 720
Current Console (2005), 1080p	1920 x 1080
Mid-range PC (2010)	1920 x 1200
Enthusiast PC (2012)	2650 x 1600
New iPad (2012)	2048 x 1536
MacBook with Retina Display (2012)	2880 x 1800
27" LCD with iPhone4S (326ppi) pixel density	7000 x 4800

Single-Display Resolutions

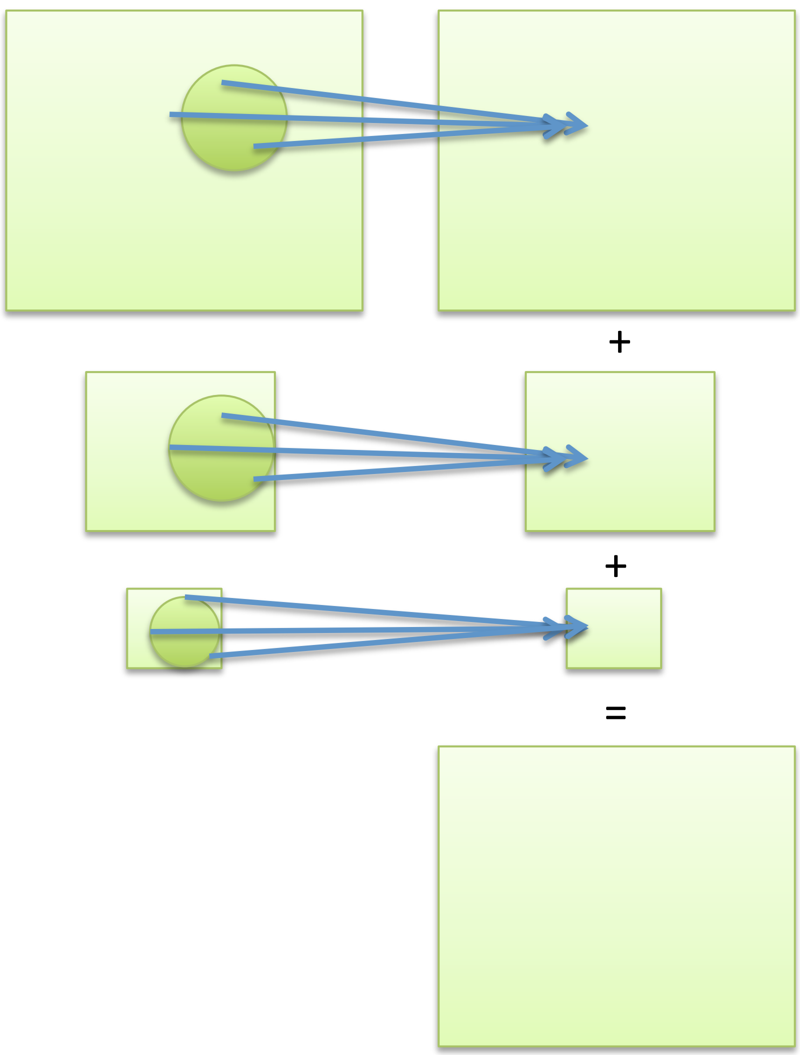
Target	Resolution
Curr	1280 x 7
Cu	1920
Mid-	1920 x 1200
Enthusiast	2650 x 1600
Ne	
Ma	
27” LCD with iPhone4S (326ppi) pixel density	7000 x 4800



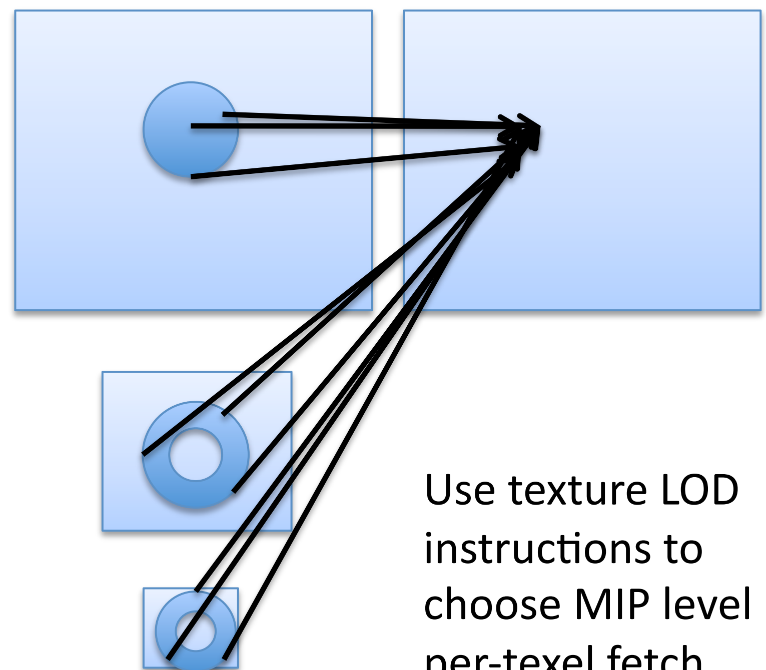


The screen-space disk of samples that fit in a fixed-size cache is rapidly shrinking.

Multi-Scale Strategies



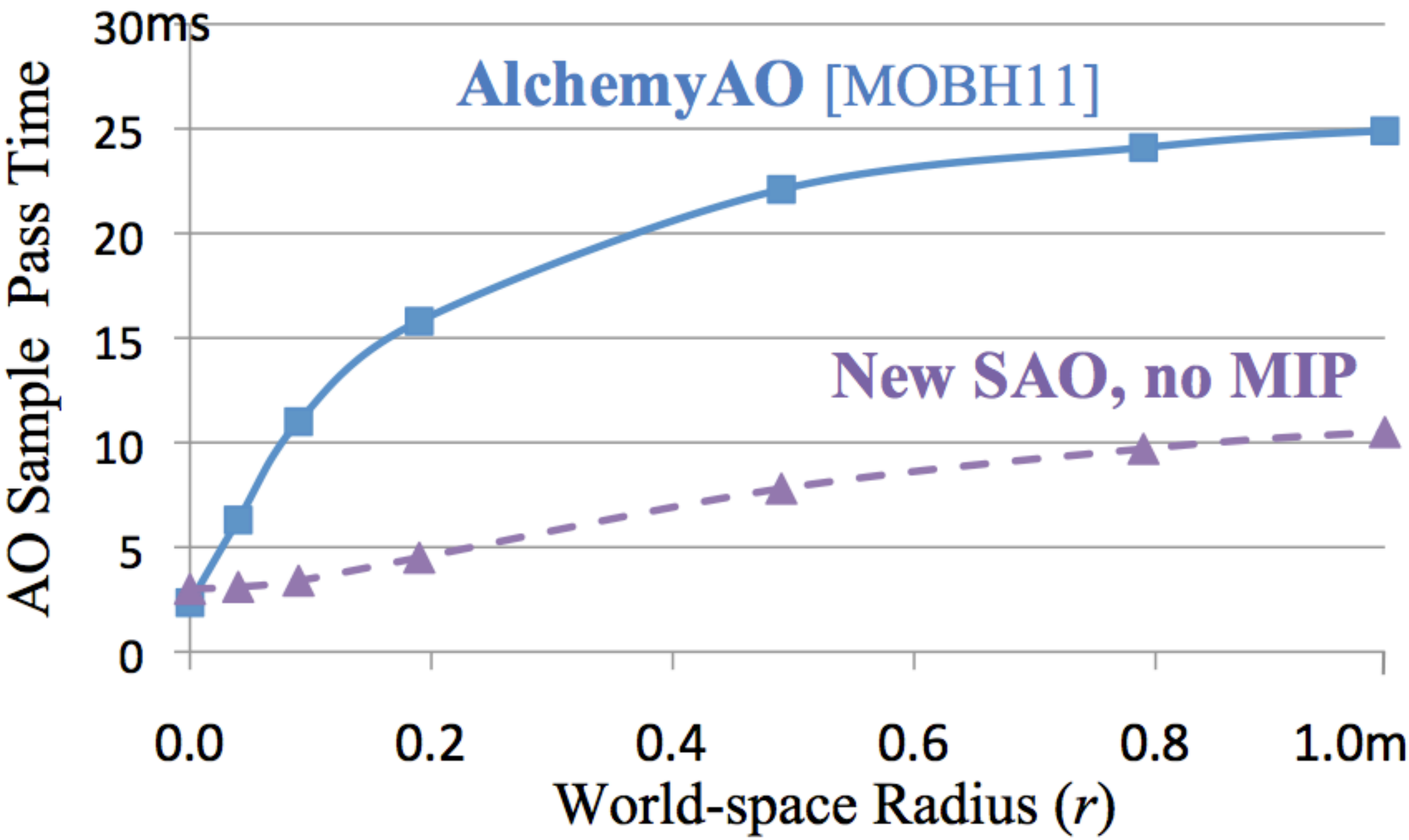
Multi-res AO [Hoang & Low 12]



Use texture LOD instructions to choose MIP level per-texel fetch

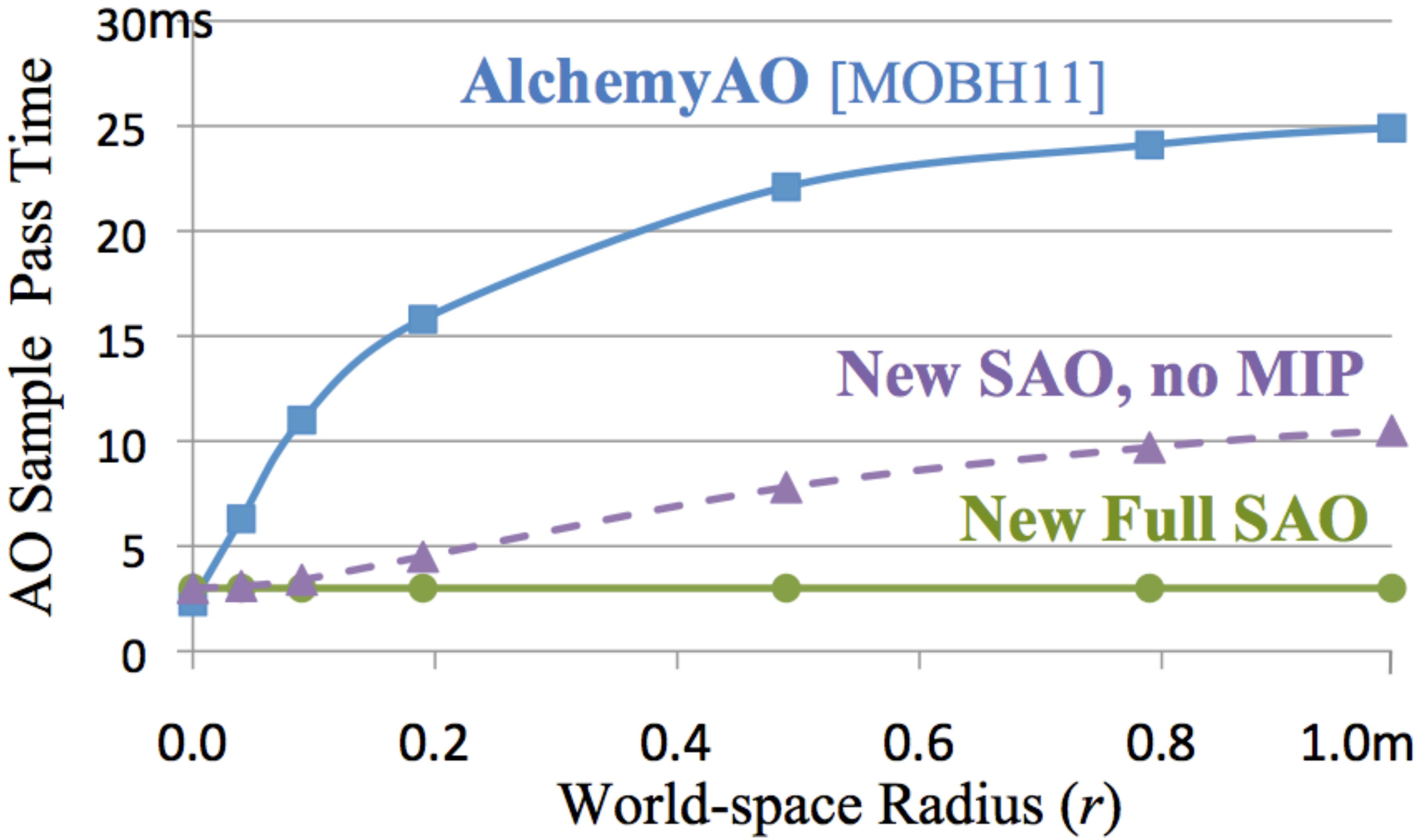
New: Scalable AO [McGuire et al. 12]

Performance Impact



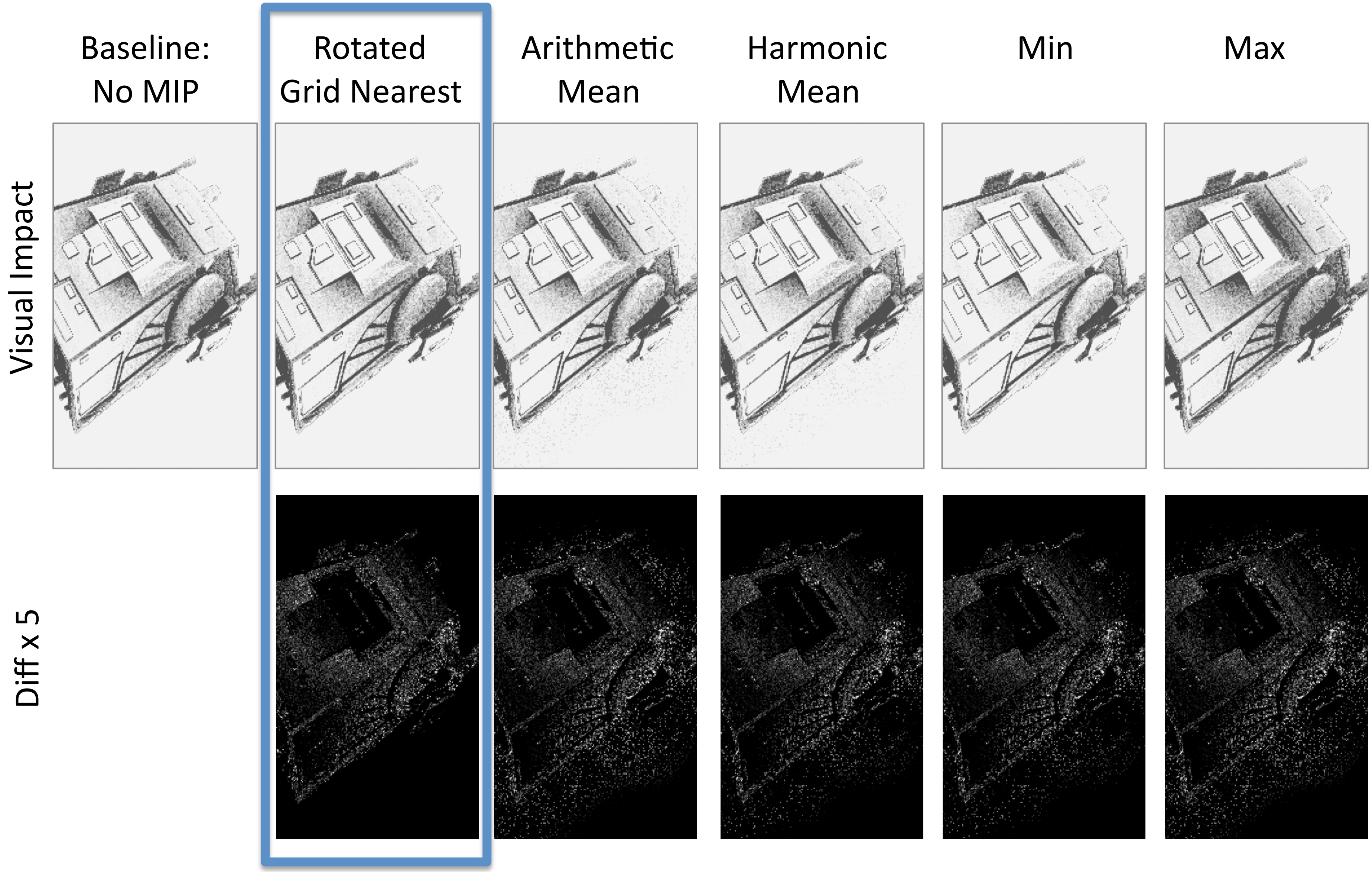
GeForce GTX 680, 2650x1600

Performance

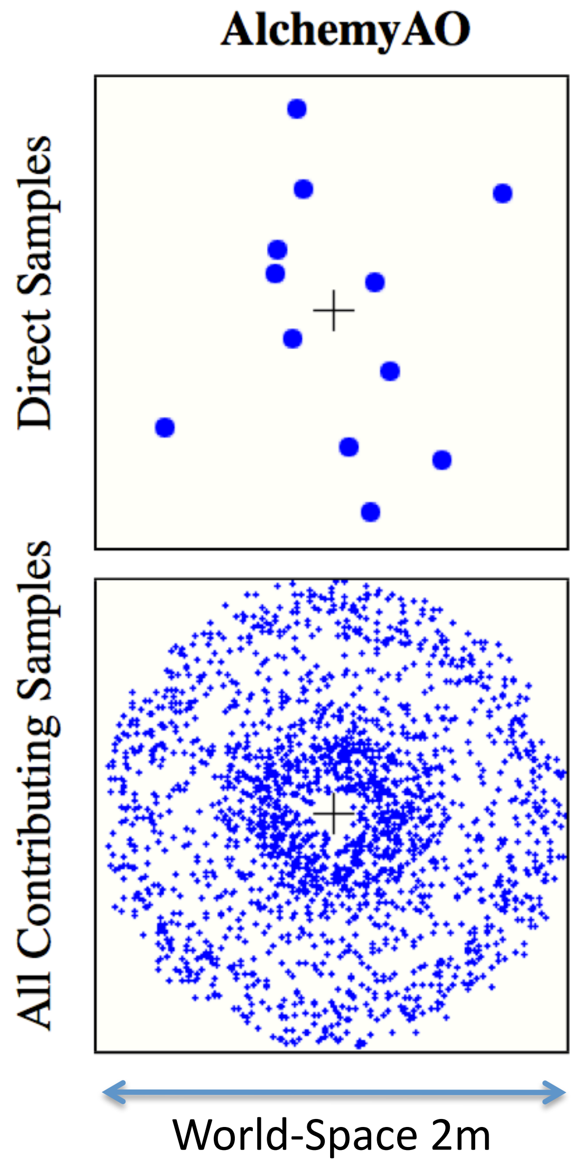


GeForce GTX 680, 2650x1600

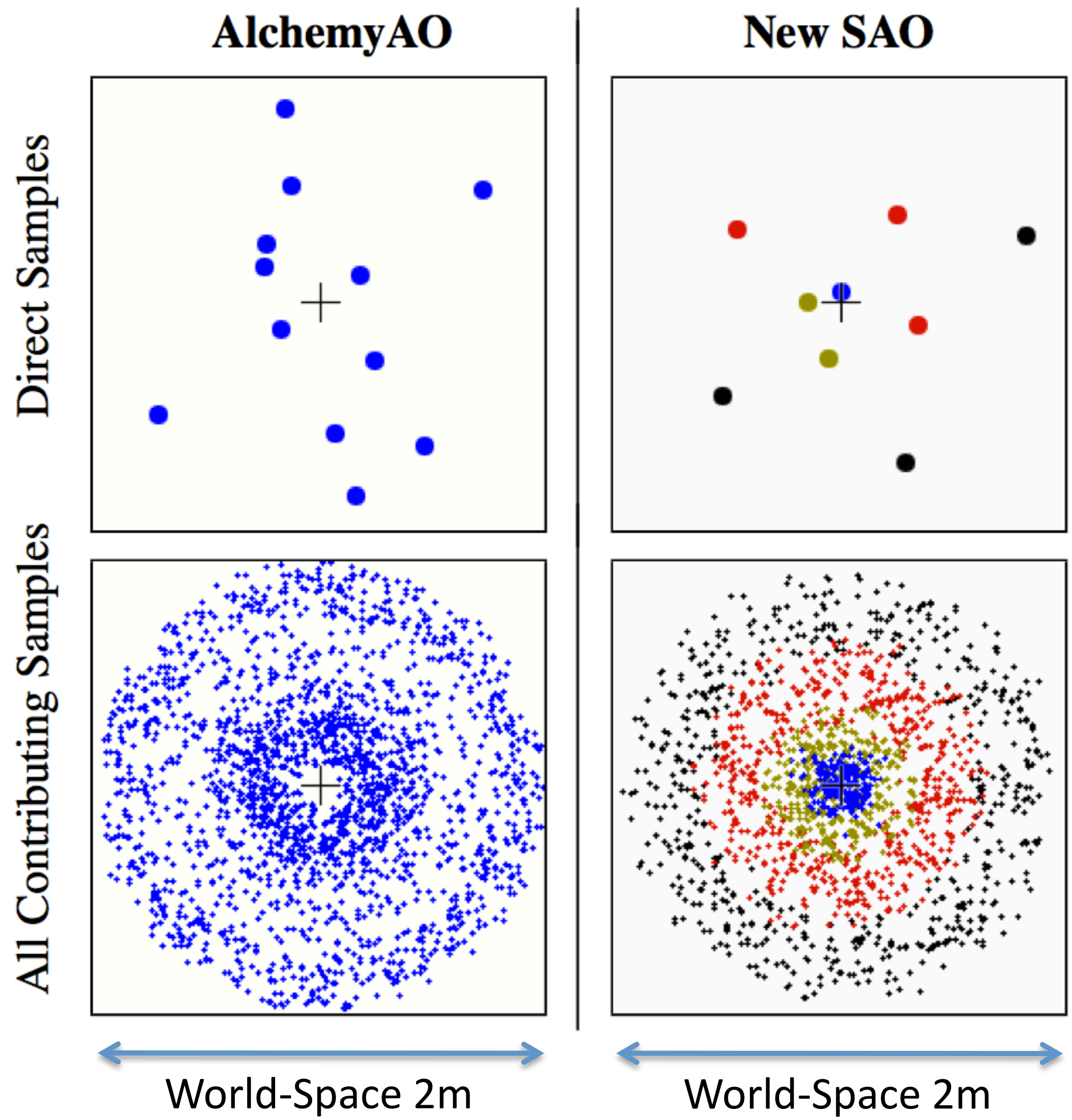
z-MIP Generation Methods



Better Samples Converge Faster



Better Samples Converge Faster



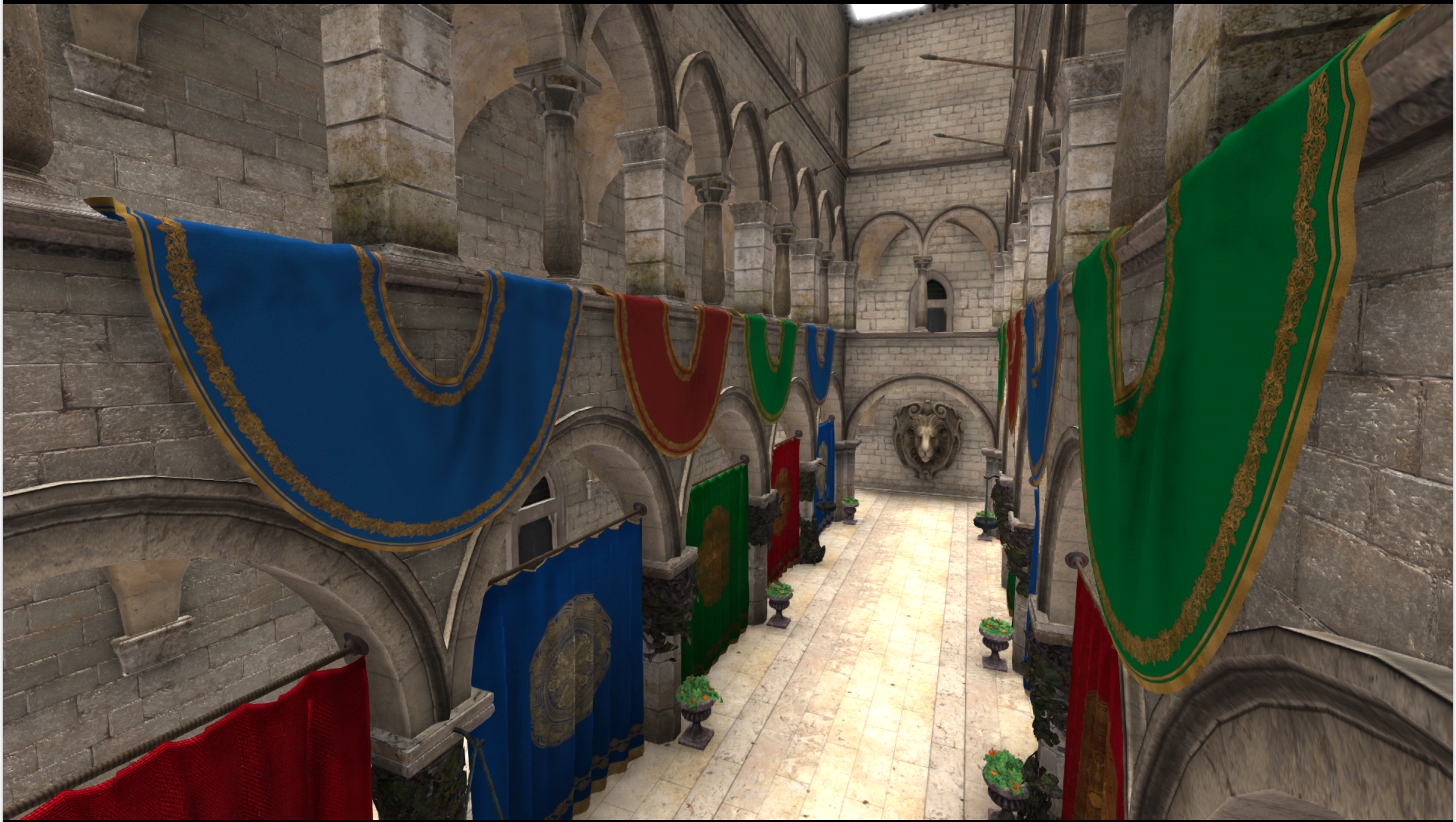
Result Images

Some flicker is due to video compression—download our demo for accurate results.

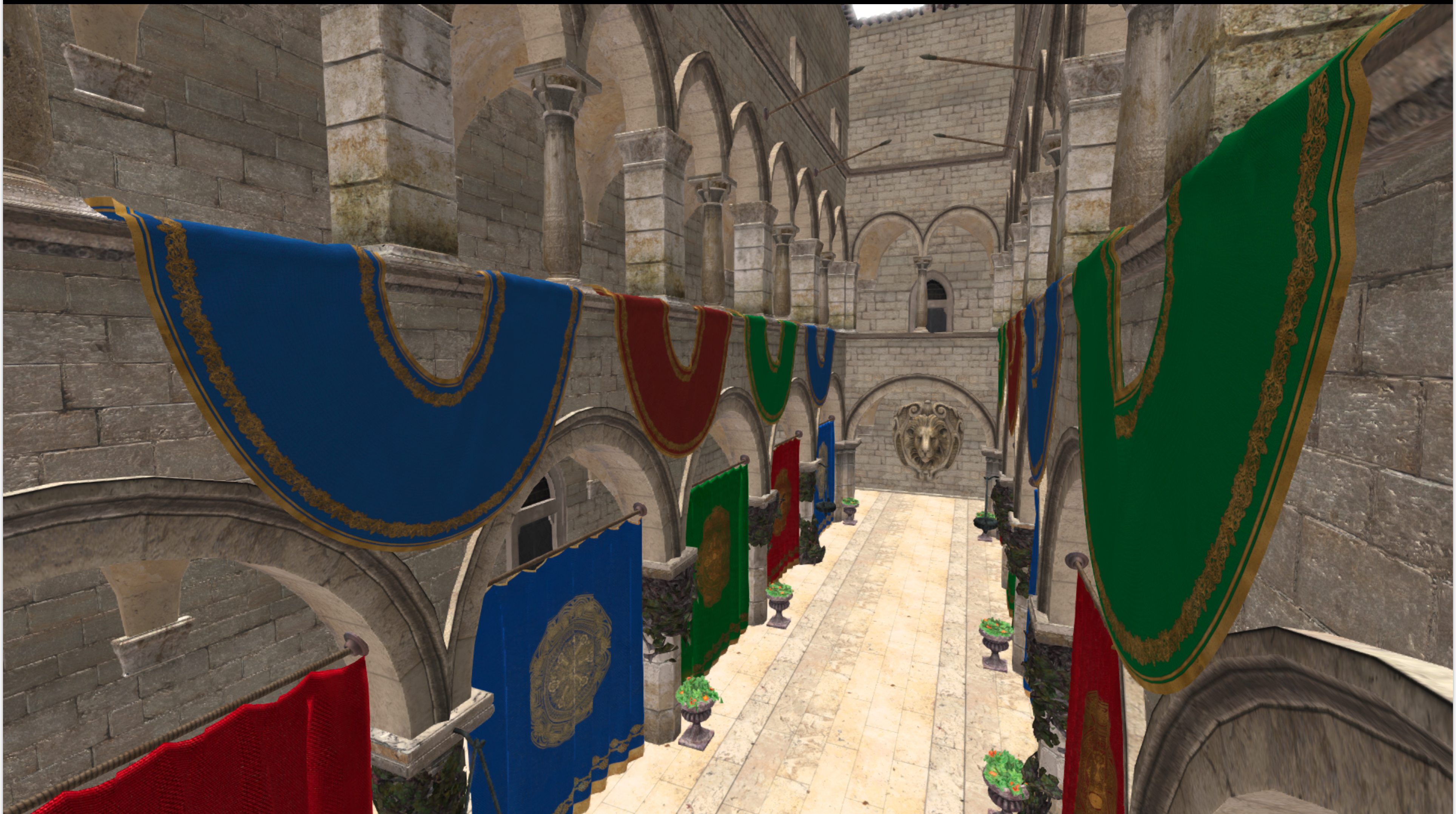
11 samples per pixel + bilateral blur reconstruction

90 samples per pixel

SAO modulating environment lighting

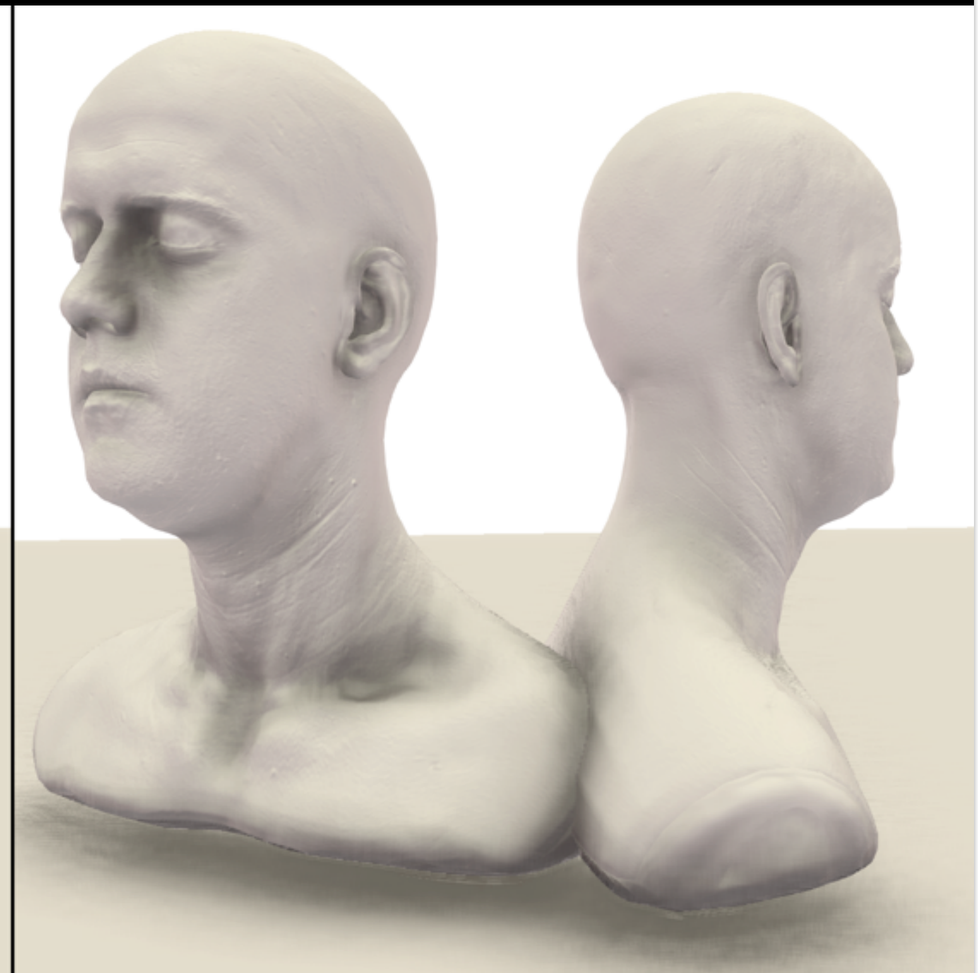


with texture

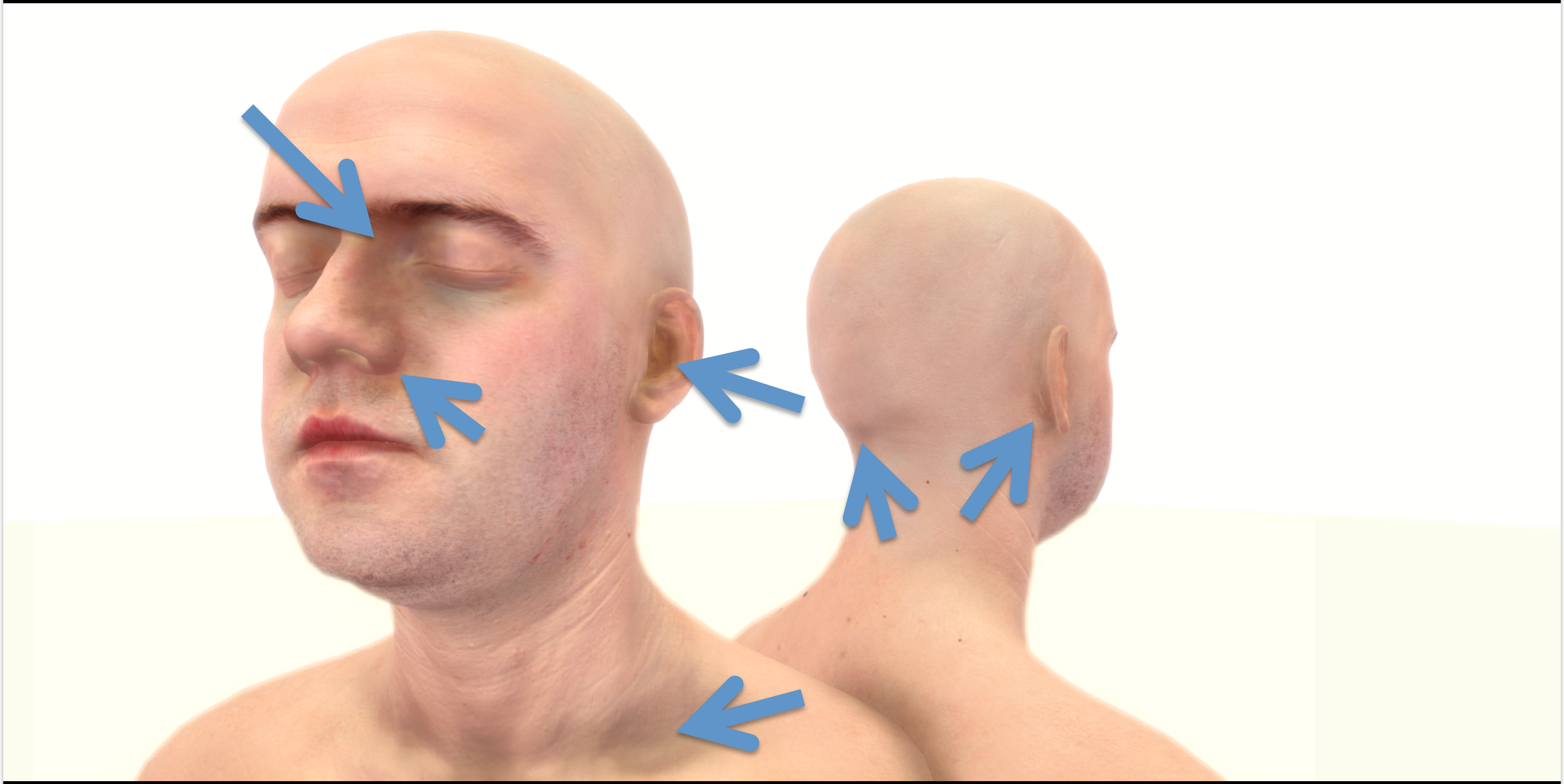


(no AO)

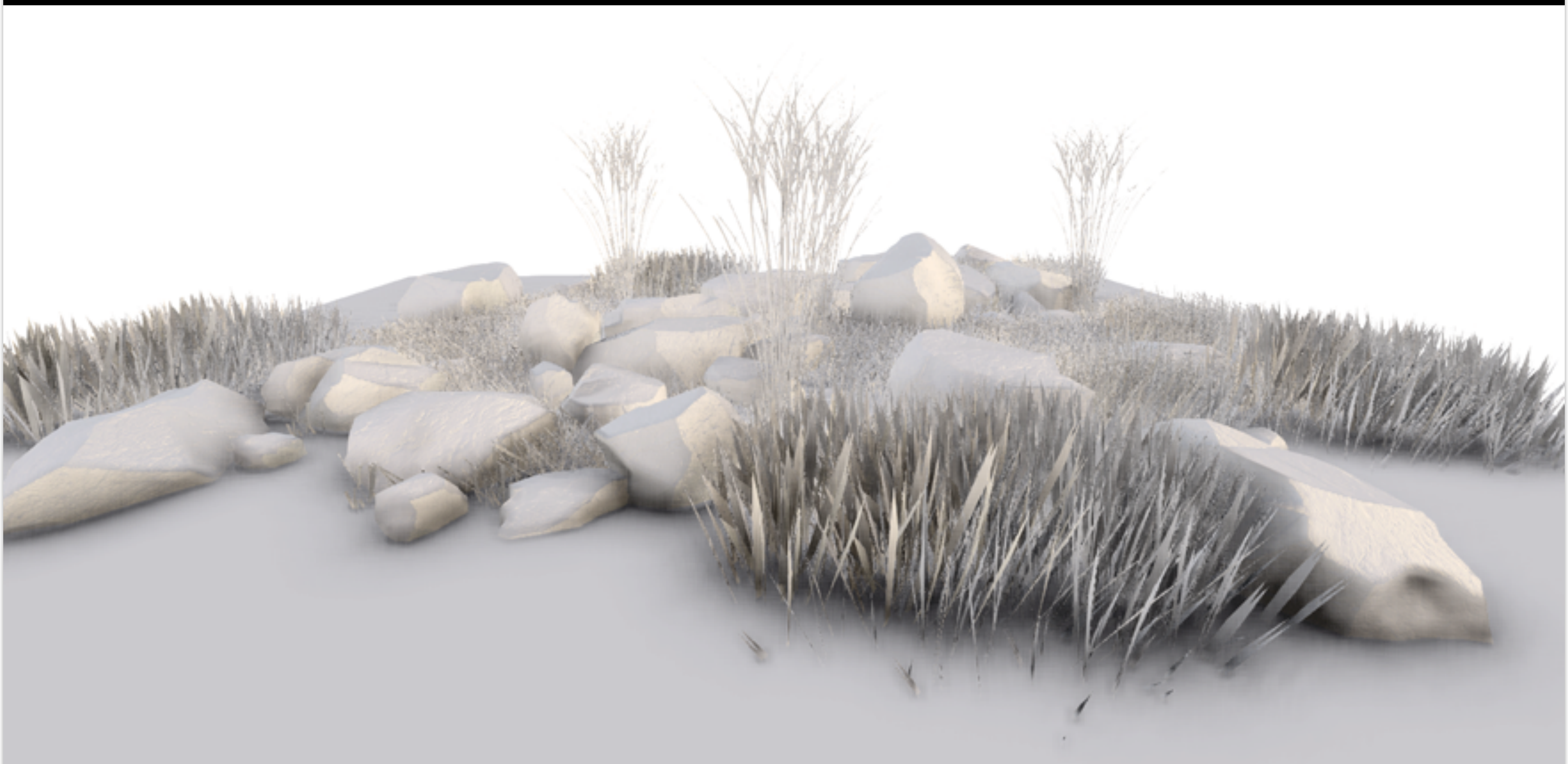
Environment Lighting



+ SAO



+ Texture and Color Grading



CONCLUSIONS

Screen-Space Effects

- Motion Blur
- Depth of Field
- Vignetting
- Bloom
- Atmospheric Attenuation
- Antialiasing
- Color Grading
- (Local) Glossy Reflection
- **(Local) Ambient Occlusion**

Road Map

- **Today**:
  - SAO for radiosity-like AO effects, 1cm < r < 200 cm
  - 5-25 samples

- **Next few years**:
  - SAO for radiosity-like AO effects, 1cm < r < 400 cm
  - 20-100 samples

- **Long term**:
  - SAO for local AO effects 1cm < r < 25 cm
  - Geometric techniques for large-scale GI

More Information

- **Downloads**
  - http://research.nvidia.com/publication/scalable-ambient-obscurance
  - http://graphics.cs.williams.edu/papers/SAOHPG12
  - OpenGL / C++ reference implementation
  - DX11 shader port by Lenardo Zide, Treyarch
  - Full-resolution result images

- Vicarious Visions presentation in SIGGRAPH 2012 *Advances in Real-Time Rendering* course

Open Problems

- Subpixel aliasing
  - Temporal post-processed AA
  - Apply SAO to MSAA depth buffer
  - LOD & dynamic tessellation
- Microscale: Bump-map AO
- Megascale: Efficient geometric AO/GI
  - e.g., AO fields, ray casting, AOV, VPL, ISPM
  - Combining with SAO

Alchemy AO

$$A \approx \max\left(0, 1 - \frac{2\sigma}{s} \cdot \sum_{i=1}^{s} \frac{\max(0, \vec{v_i} \cdot \hat{n} + z_C \beta)}{\vec{v_i} \cdot \vec{v_i} + \epsilon}\right)^k$$

Our new hard-coded constants:

s = Number of samples (console: 6, DX11 PC: 10-20, future up to 80)
σ = 2.25
k = 1
ε = 1cm
Fix $z_C \beta$ = -1cm

McGuire, Osman, Bukowski, and Hennessy, The Alchemy Screen-Space Ambient Obscurance Algorithm, HPG 12

FALLOFF KERNEL

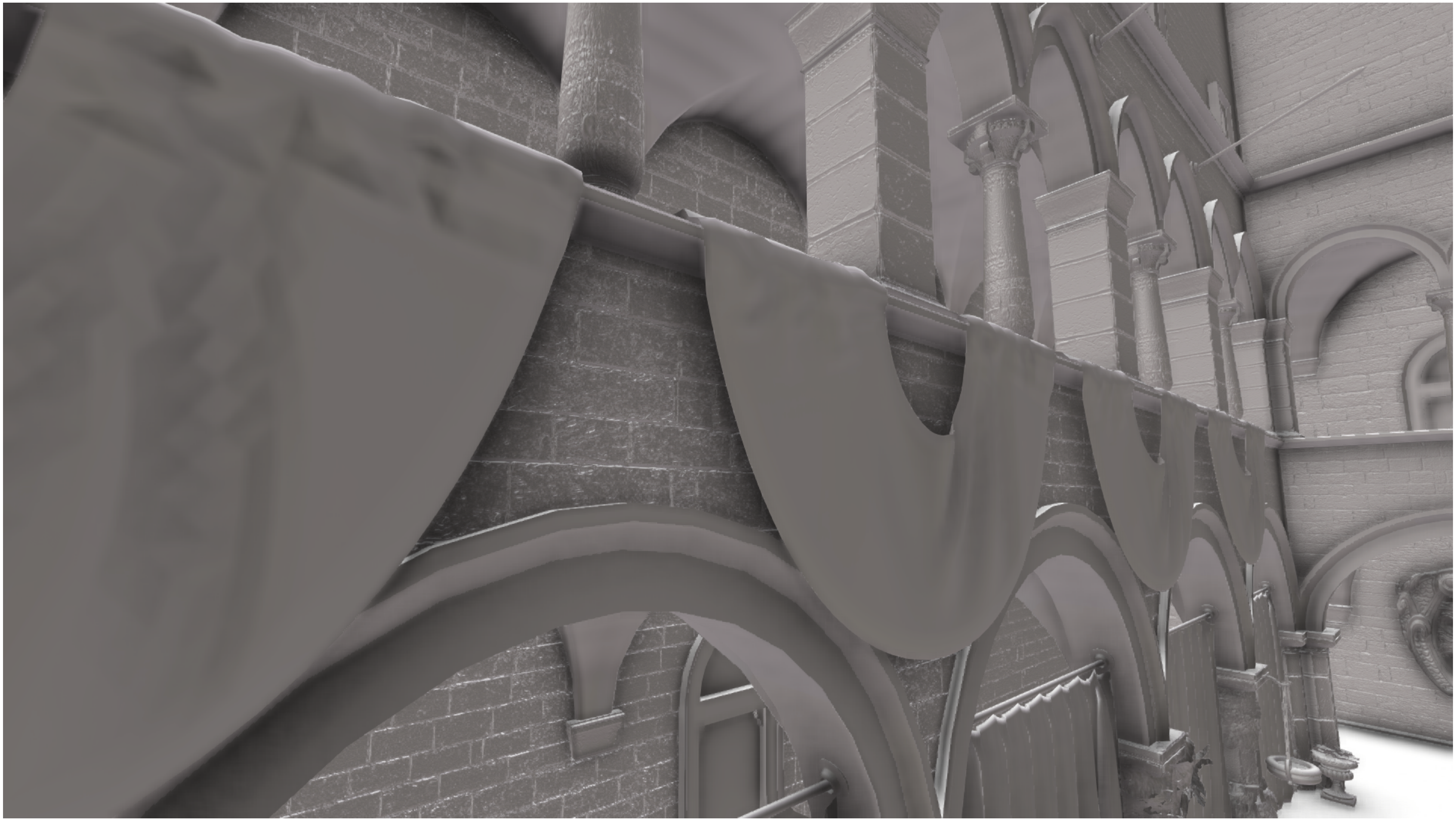
Falloff Kernel

- Lots of options, e.g., from
  - VO, Crease shading, AlchemyAO, HBAO
- Issues:
  - Angular view dependence
  - Spatial and temporal variance
  - Falloff rate from corners
  - Computational efficiency
- All of the following have comparable run-time and have been normalized to the same intensity

A

Published in paper

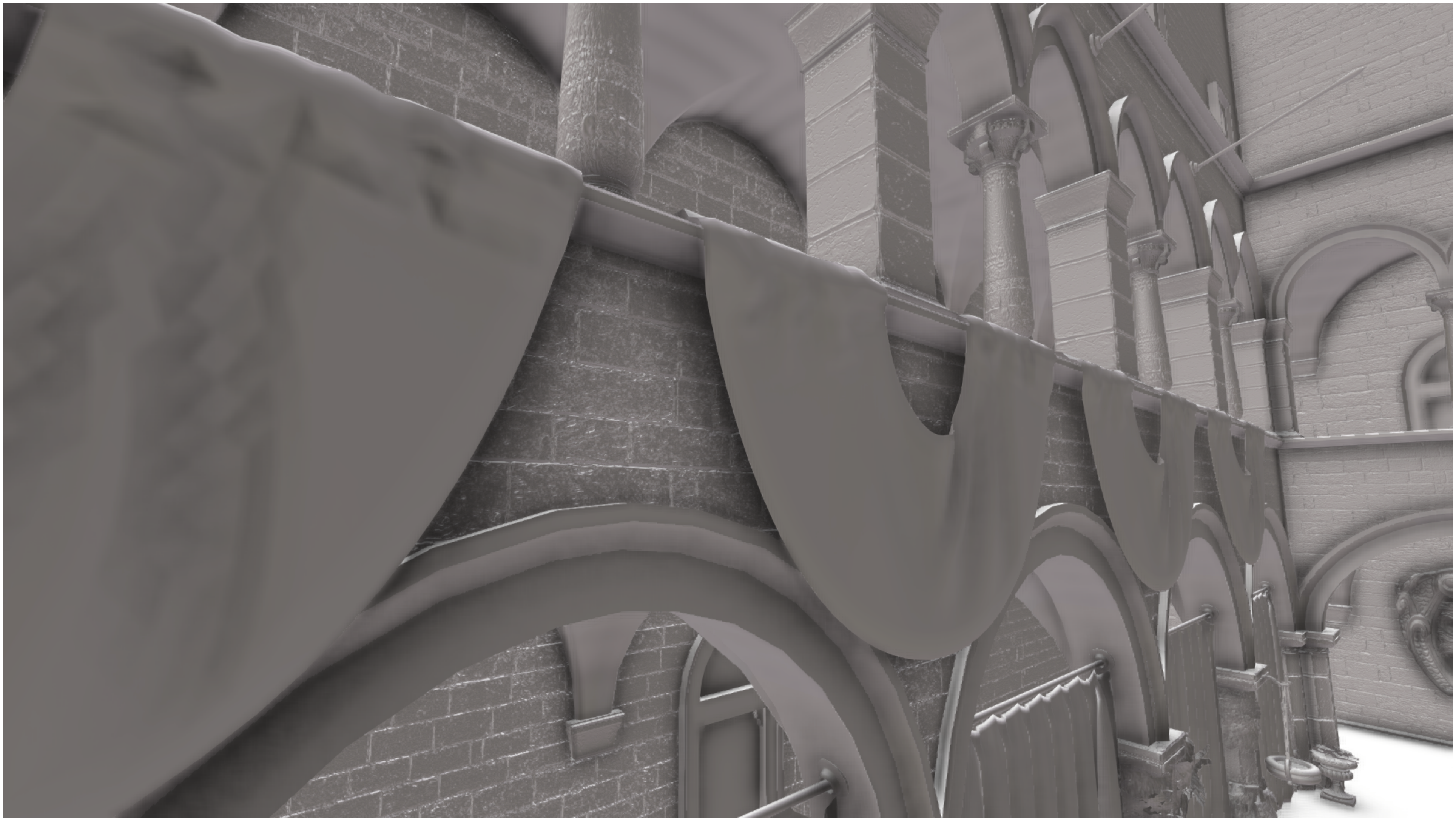
```
const float epsilon = 0.01;
return float(vv < radius2) * max((vn - bias) /
    (epsilon + vv), 0.0) * radius2 * 0.6;
```



B

Currently recommended

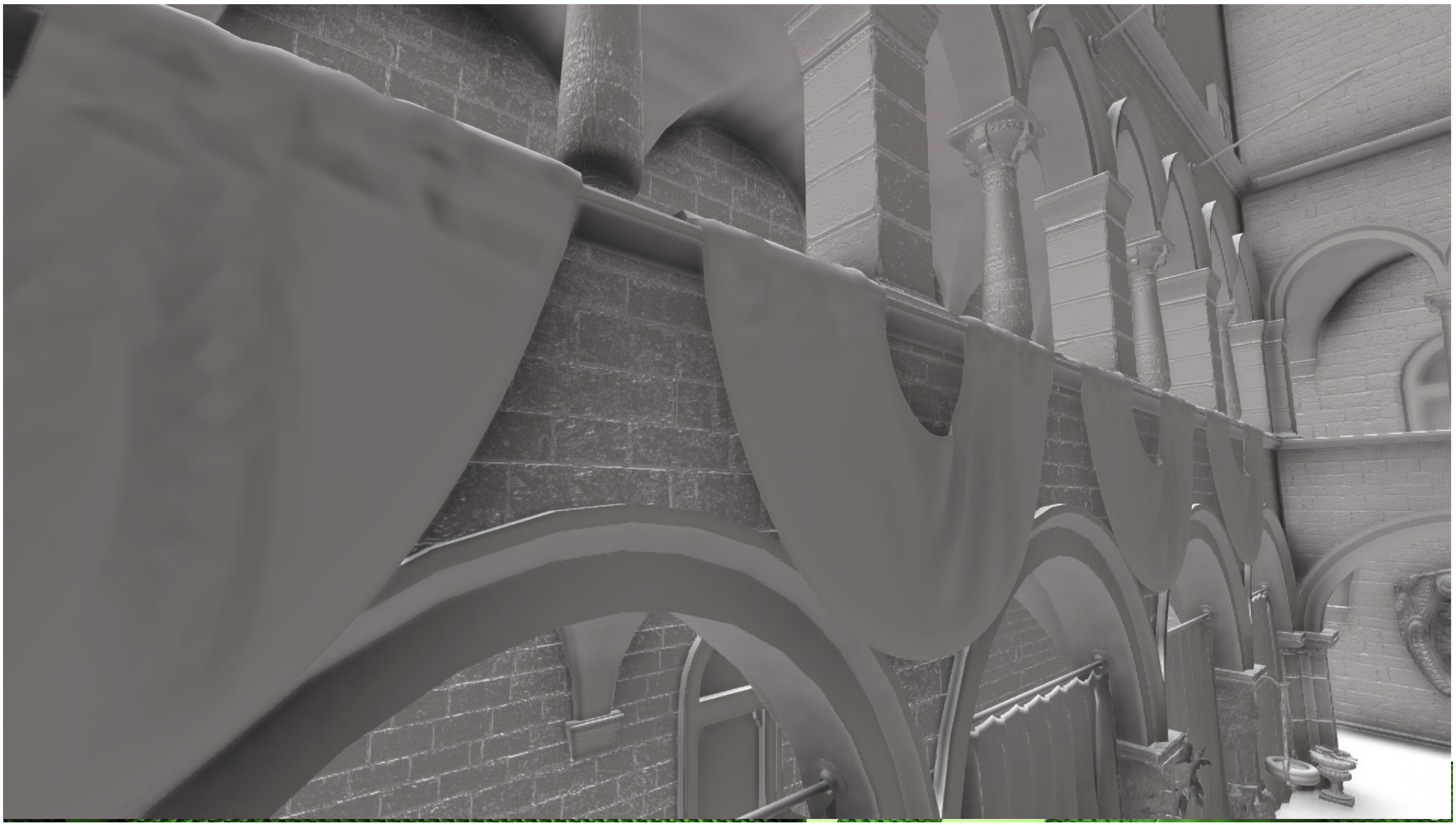
```
const float epsilon = 0.01;
float f = max(radius2 - vv, 0.0);
return f * f * f * max((vn - bias) /
    (epsilon + vv), 0.0);
```



C

```
return 4.0 * max(1.0 - vv * invRadius2, 0.0) *
        max(vn - bias, 0.0);
```

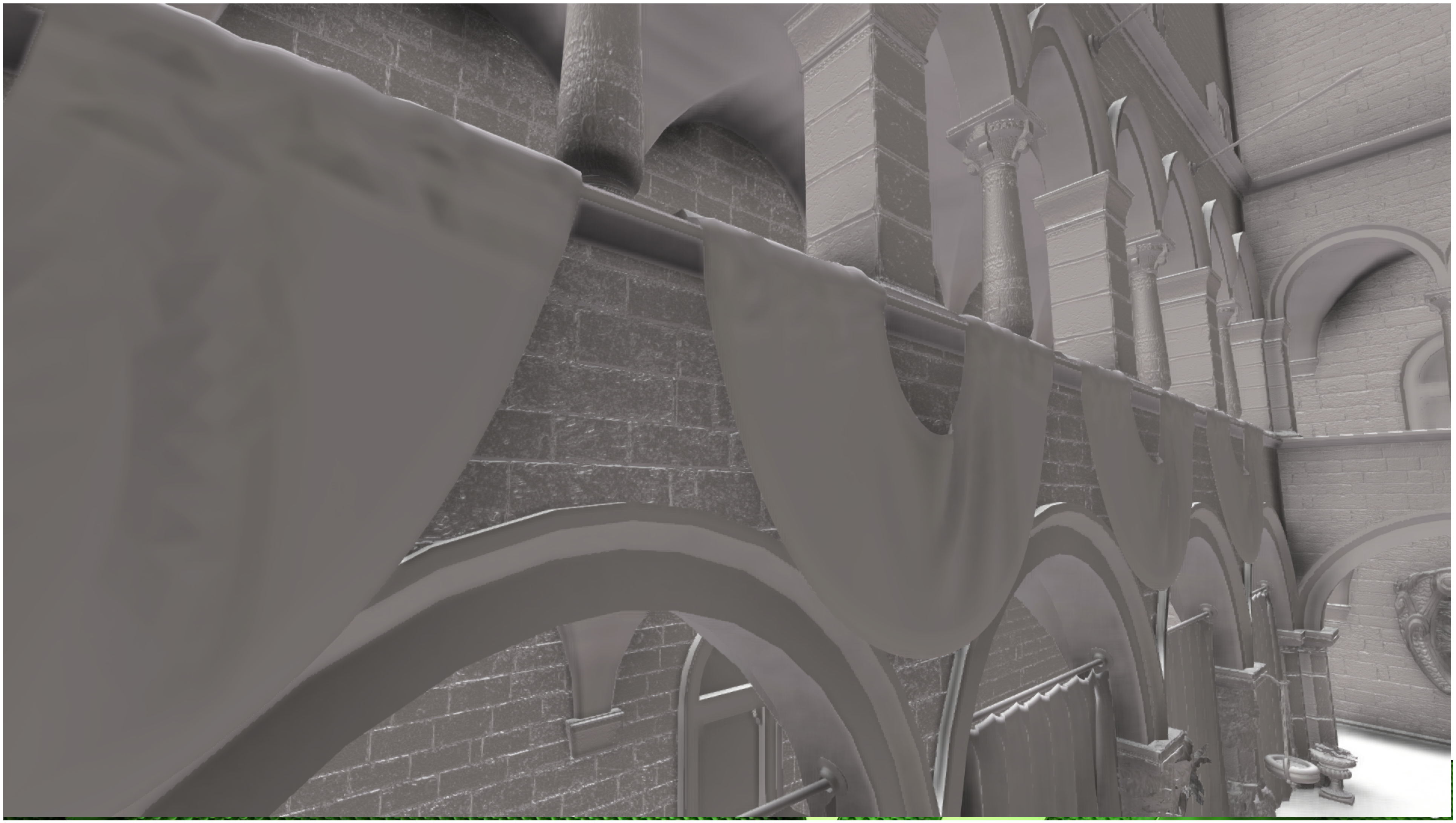
No division



D

```
return 2.0 * float(vv < radius * radius) *
    max(vn - bias, 0.0);
```

No division



Supplemental Images