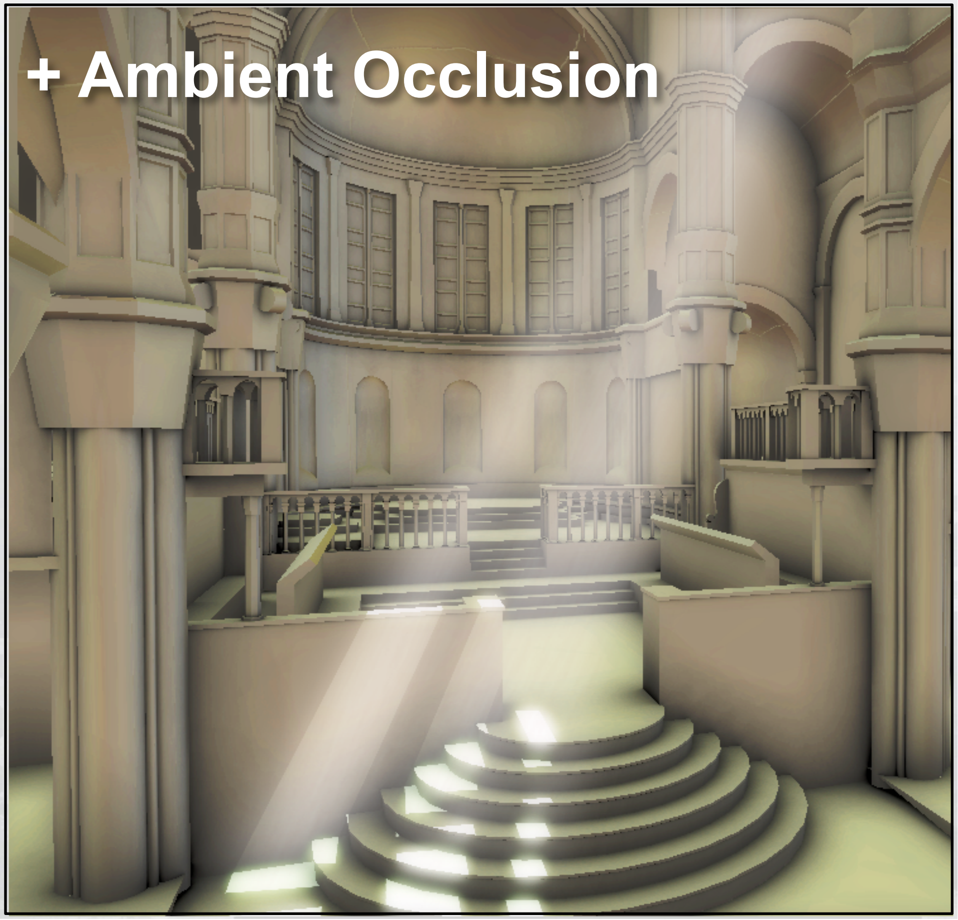
Ambient Occlusion Volumes "AOV"

Morgan McGuire
NVIDIA Research & Williams College

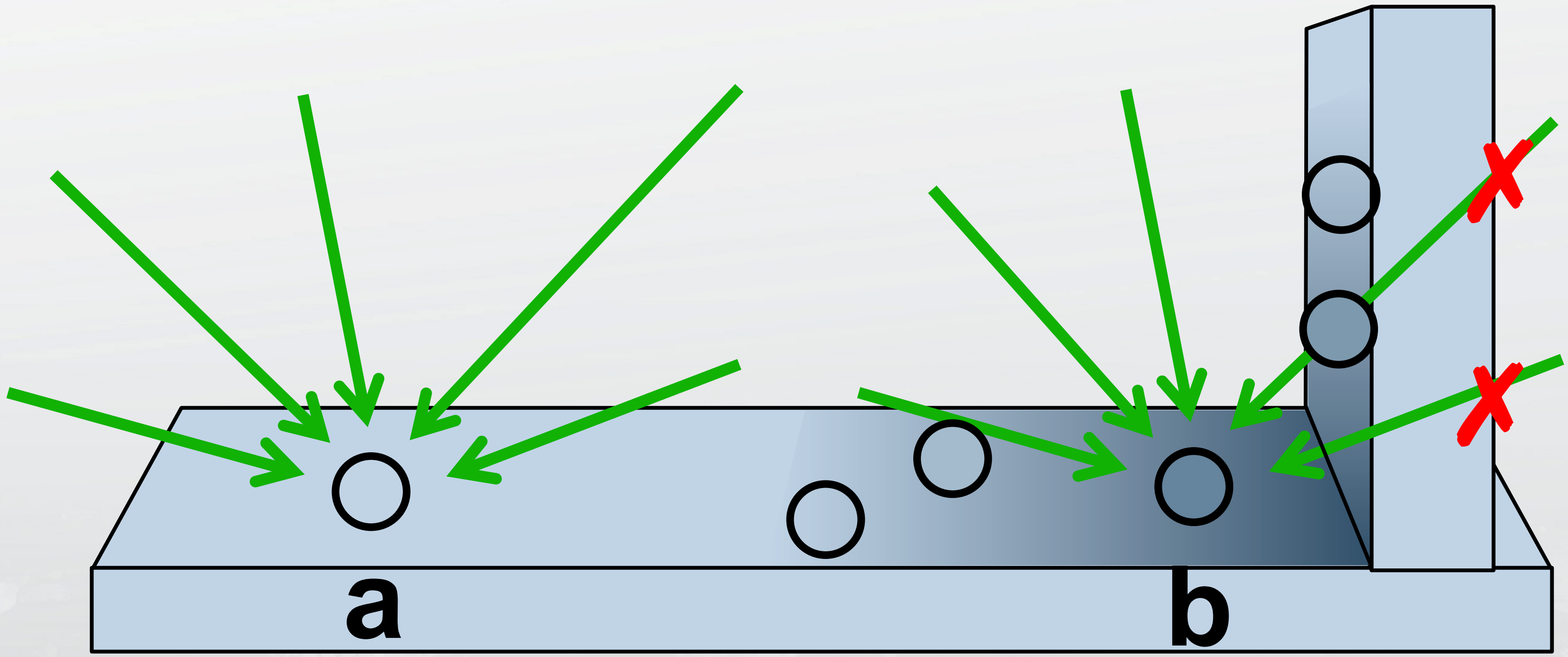
hpg 2010

Artistic Motivation

- Perceived depth
- Proximity
- Lighting contrast
- Curvature and creases



Intuition



100%

60%

Selected Previous Work

- **Analytic**
  - Sphere proxies [Bunnell 05, Hoberock and Jia 07, Shopf et al. 09]
  - Offline [Baum et al. 89, Zhukov et al. 98, Hoberock and Jia 07, Shopf et al. 09]
- **Sampled**
  - Ray tracing [e.g., Dutre et al. 04]
  - Signed distance field [Evans '06]
  - Precomputed [Kontkanen and Laine 05, Malmer
  - Voxel ray trace [Reinbothe et al. 09]
  - Light probe [Sloan et al. 07]
  - Raster bit mask [Laine and Karras '10]
- **Phenomenological**
  - Tree-specific [Hegeman
  - Unsharp masking [Luf
  - Screen space [Mittring '07, Shanmugam and Arikan '07, Bavoil and Sainz '09, Kajalin '09]

Analytic on bounding box

Precomputed volume of AO effect

Tighter bounding volume and LOD

Sparse sampling

AOV Features

- Real-time
- Dynamic polygon soup
- Physically based
- Noise-free

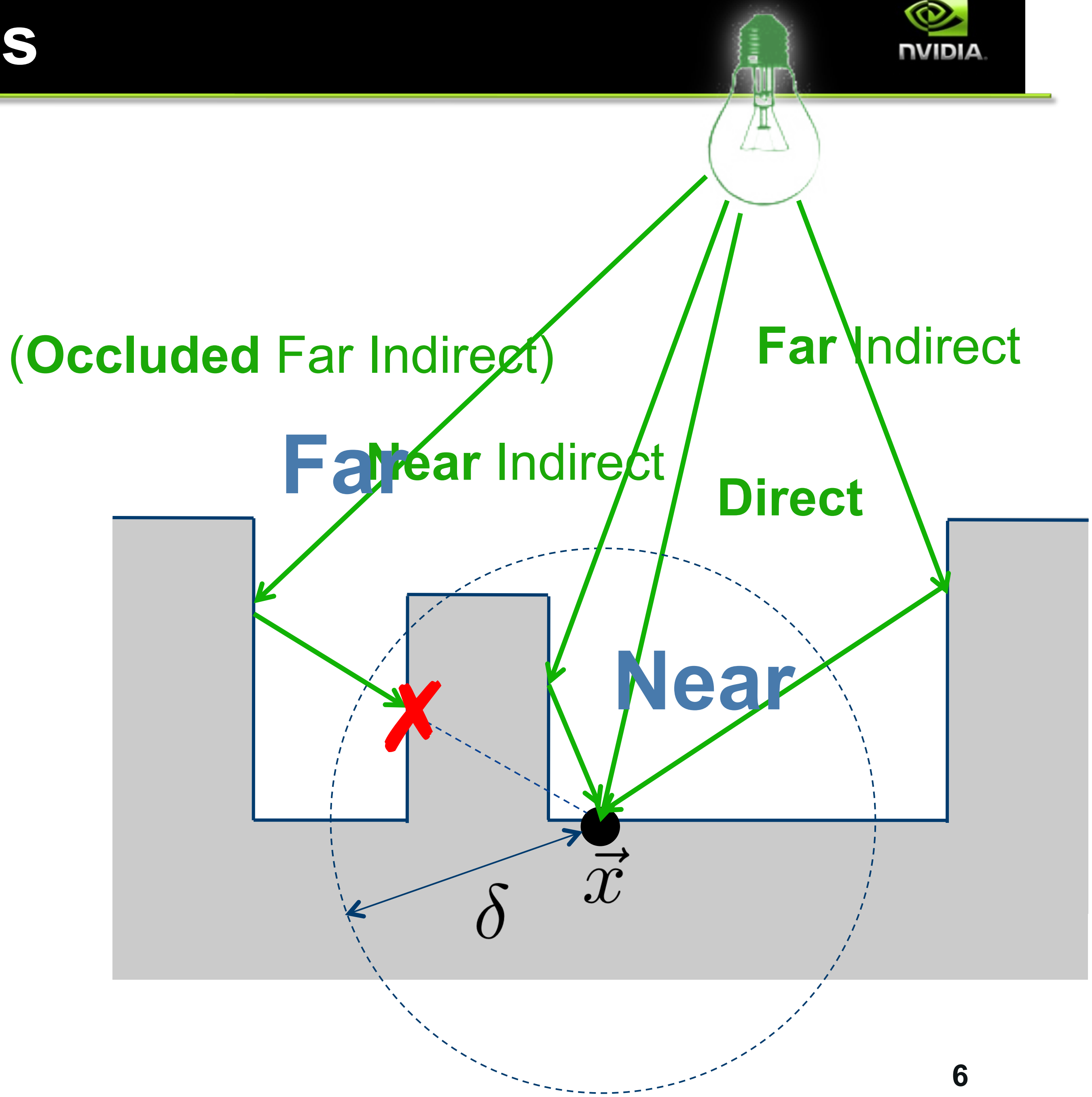
Unlike phenomenological screen-space methods:
- Viewer independent
- Viewport independent
- Designed for integration with real GI

- Approaches the quality of offline ray traced occlusion

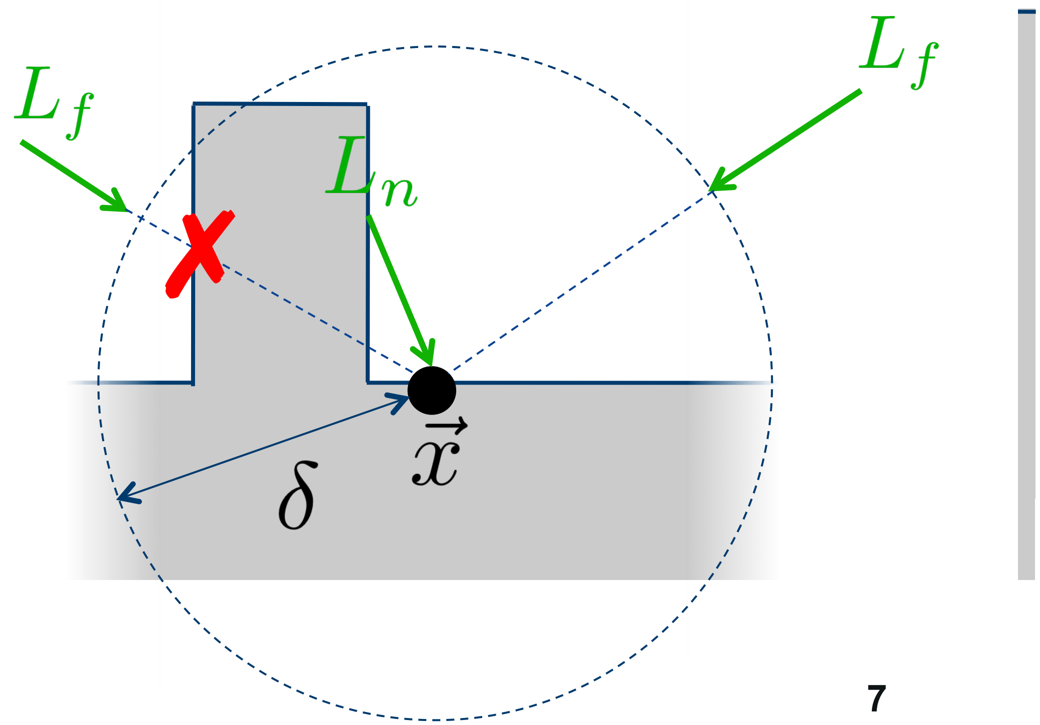
- Requirements:
  - Geometry shader for preferred implementation
  - Normal and depth ("Geometry") buffers
  - High fill rate GPU
  - Minimum thickness to objects; overdarkening when this is violated

DERIVATION

Physical Basis



Physical Basis



Physical Basis

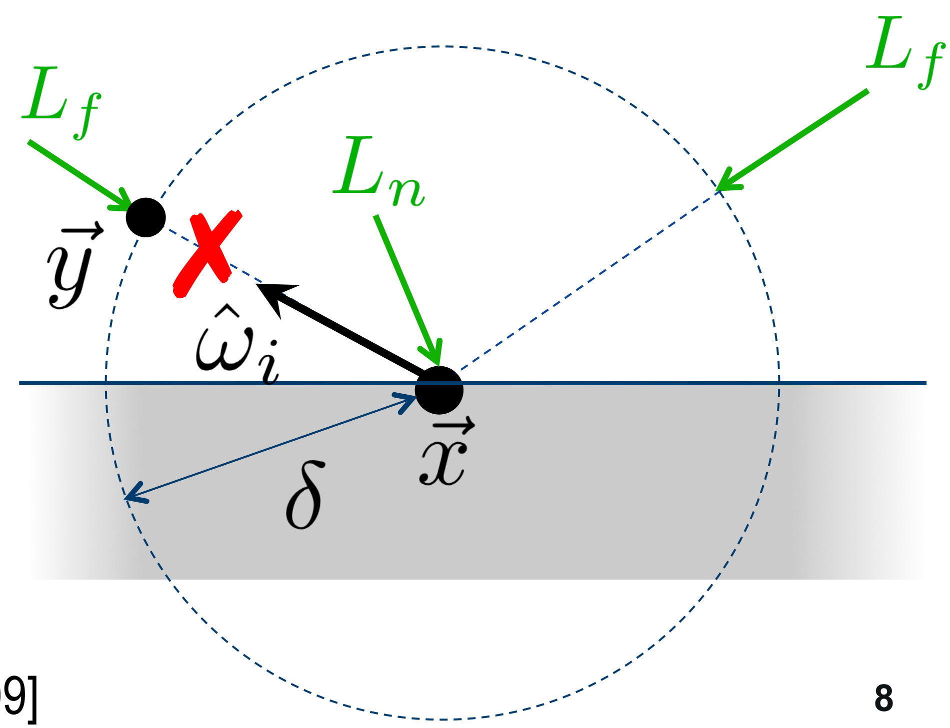
Let $\vec{y} = \vec{x} + \delta\hat{\omega}_i$

$$\mathcal{V}(\vec{x}, \vec{y}) = 1 \text{ iff } \vec{y} \text{ is visible to } \vec{x}, 0 \text{ otherwise}$$

$$L_i(\vec{x}, \hat{\omega}_i) = L_n(\vec{x}, \hat{\omega}_i) \cdot (1 - \mathcal{V}(\vec{x}, \vec{y})) + L_f(\vec{y}, \hat{\omega}_i) \cdot \mathcal{V}(\vec{x}, \vec{y})$$

Set δ to your global illumination algorithm's sampling resolution (or choose artistically)

$\delta = \infty$ is undesirable: makes everything 100% occluded for indoor scenes [Shanmugam and Arikan 07]



[Arikan et al. 05; Szirmay-Kalos et al. 09]

Incident Light

[KAJIYA 86]

$$L_o = L_e + \int L_i \cdot f \cdot (\hat{\omega}_i \cdot \hat{n}) \; d\hat{\omega}_i$$

[MCGUIRE 09, 10]

$$L_o = L_e +$$ "**emitted**" (e.g., emissive map)

$$\sum_{\hat{\omega}_i} L_s \cdot \mathcal{V} \cdot f \cdot (\hat{\omega}_i \cdot \hat{n}) +$$ "**direct**" (BRDF + shadow map)

$$\int L_n \cdot (1 - \mathcal{V}) \cdot f \cdot (\hat{\omega}_i \cdot \hat{n}) \; d\hat{\omega}_i +$$ "**local indirect**" (often ignored)

$$\int L_f \cdot \mathcal{V} \cdot f \cdot (\hat{\omega}_i \cdot \hat{n}) \; d\hat{\omega}_i$$ "**ambient**" (e.g., S.H. evt. light)

Ambient term:

"**ambient light**" (e.g., precompute, VPL, or ISPM)

"**accessibility**" (via AO estimation)

$$\int L_f \cdot \mathcal{V} \cdot f \cdot (\hat{\omega}_i \cdot \hat{n}) d\hat{\omega}_i \approx \left[\int L_f \cdot f \cdot (\hat{\omega}_i \cdot \vec{n}) d\hat{\omega}_i \right] \cdot \left[\frac{1}{\pi} \int \mathcal{V} \cdot \max(\hat{\omega}_i \cdot \vec{n}, 0) d\hat{\omega}_i \right]$$

low frequency

medium frequency

best when L_f and f are smooth (ideally, constant)

Ambient Occlusion

"accessibility" "ambient occlusion"

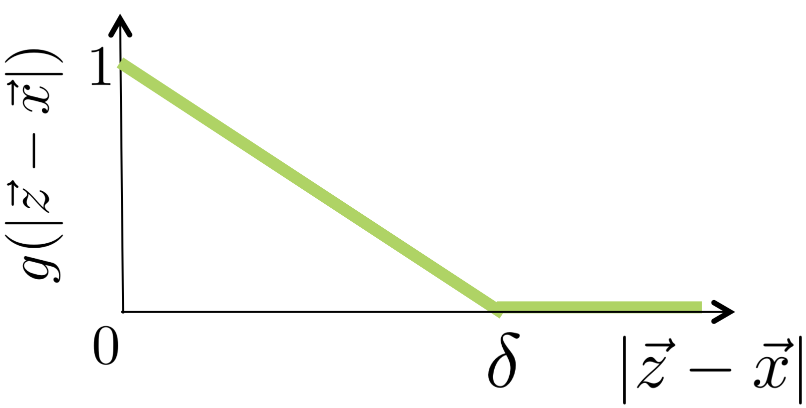
$$\left[\frac{1}{\pi}\int \mathcal{V}\cdot\max(\hat{\omega}_i\cdot\vec{n},0)\,d\hat{\omega}_i\right] = 1 - \mathrm{AO}(\vec{x})$$

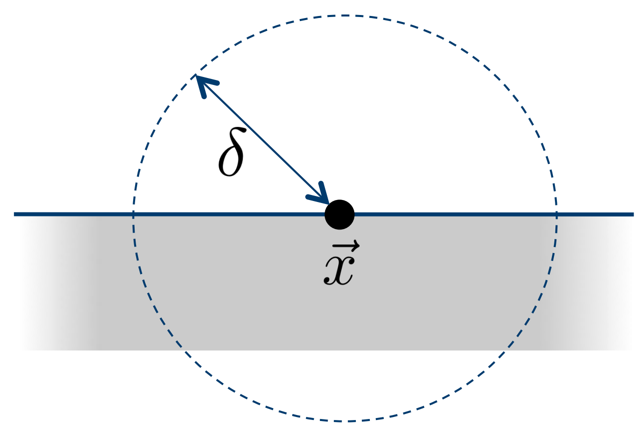
obscurance

Ambient ~~occlusion~~ of $\vec{x}$ by one polygon, P:

$$\mathrm{AO}_P(\vec{x}) = 1 - \frac{\alpha_P}{\pi}\int_P \mathcal{V}(\vec{x},\vec{z})\cdot\max\left(\frac{\vec{z}-\vec{x}}{|\vec{z}-\vec{x}|}\cdot\hat{n},0\right)\cdot g(|\vec{z}-\vec{x}|)\;d\vec{z}$$

[Zhukov et al. 98]





AOV Fundamental Operation

Clip P to tangent plane

For minimum scene thickness δ, no *other* polygon obscures P, so $\mathcal{V}$=1

Choose linear falloff in point-poly distance

$$\mathrm{AO}_P(\vec{x}) = 1 - \frac{\alpha_P}{\pi}\int_P \mathcal{V}(\vec{x},\vec{z})\cdot \max\left(\frac{\vec{z}-\vec{x}}{|\vec{z}-\vec{x}|}\cdot\hat{n},0\right)\cdot g(|\vec{z}-\vec{x}|)\, d\vec{z}$$

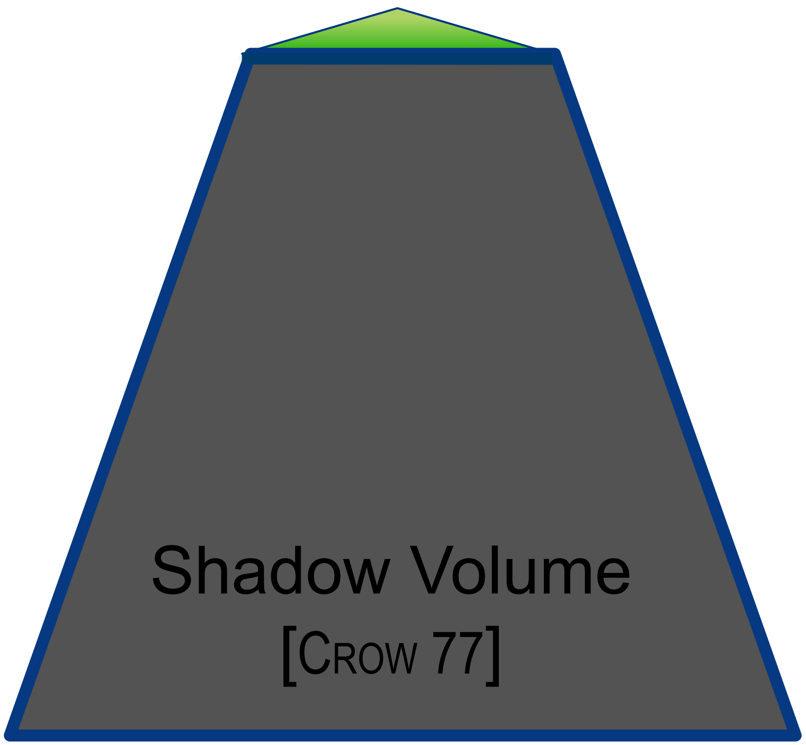
$$\approx 1 - \frac{g\cdot\alpha_P}{\pi}\int_{P'} \frac{\vec{z}-\vec{x}}{|\vec{z}-\vec{x}|}\cdot\hat{n}\, d\vec{z}$$

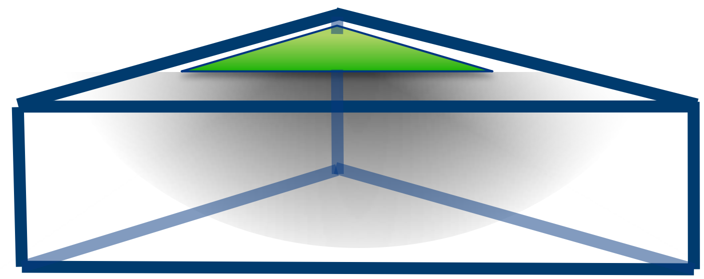
k = 4 for clipped triangle
store $\cos^{-1}$ LUT in a texture

$$= \frac{g\cdot\alpha_P}{2\pi}\sum_{i=0}^{k-1}\cos^{-1}\left(\frac{\vec{p}_i\cdot\vec{p}_j}{||\vec{p}_i||\ ||\vec{p}_j||}\right)\hat{n}\cdot\frac{\vec{p}_i\times\vec{p}_j}{||\vec{p}_i\times\vec{p}_j||}$$

where $j = (i+1) \bmod k$

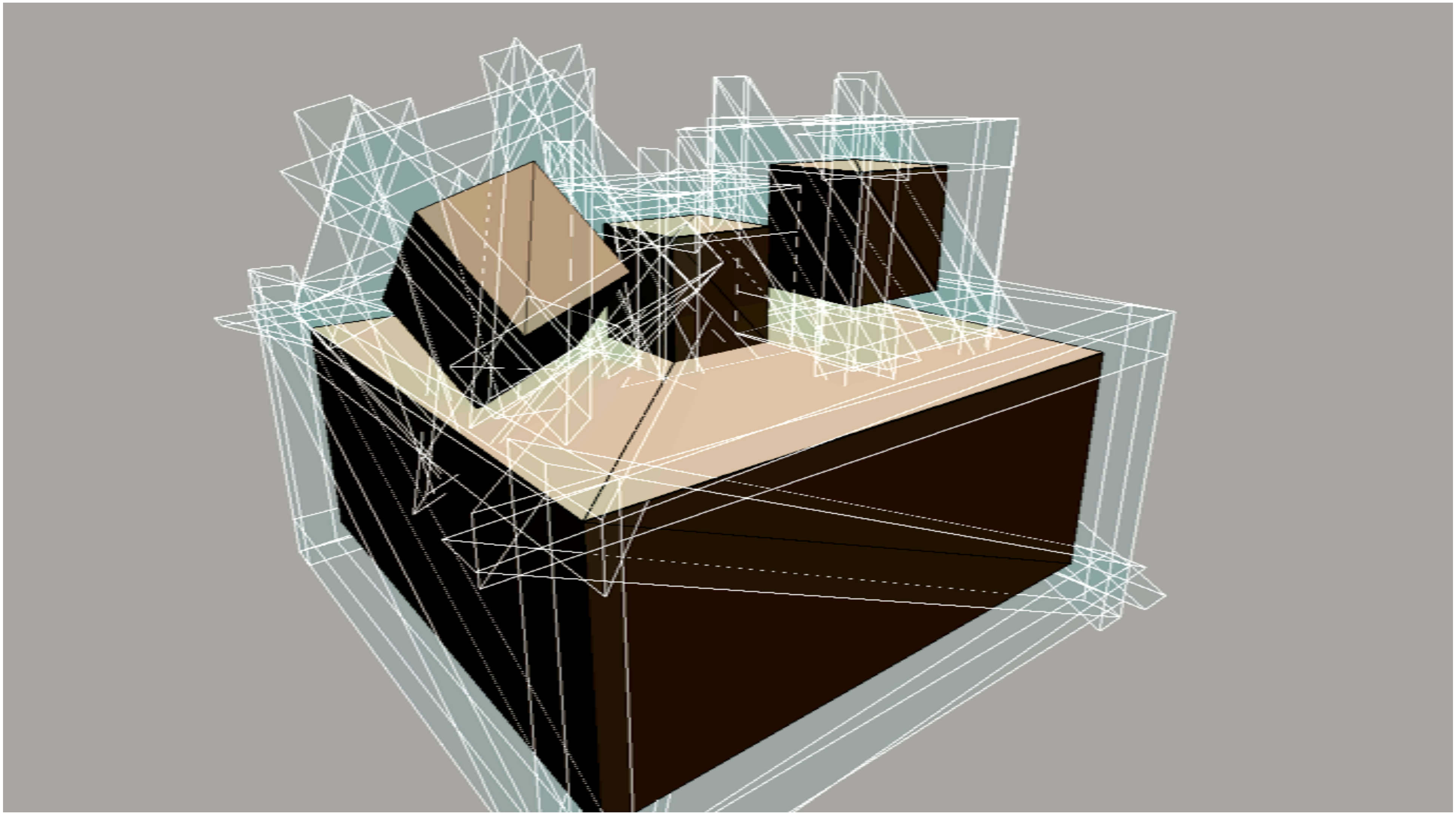
Iterating over Polygons Near a Point



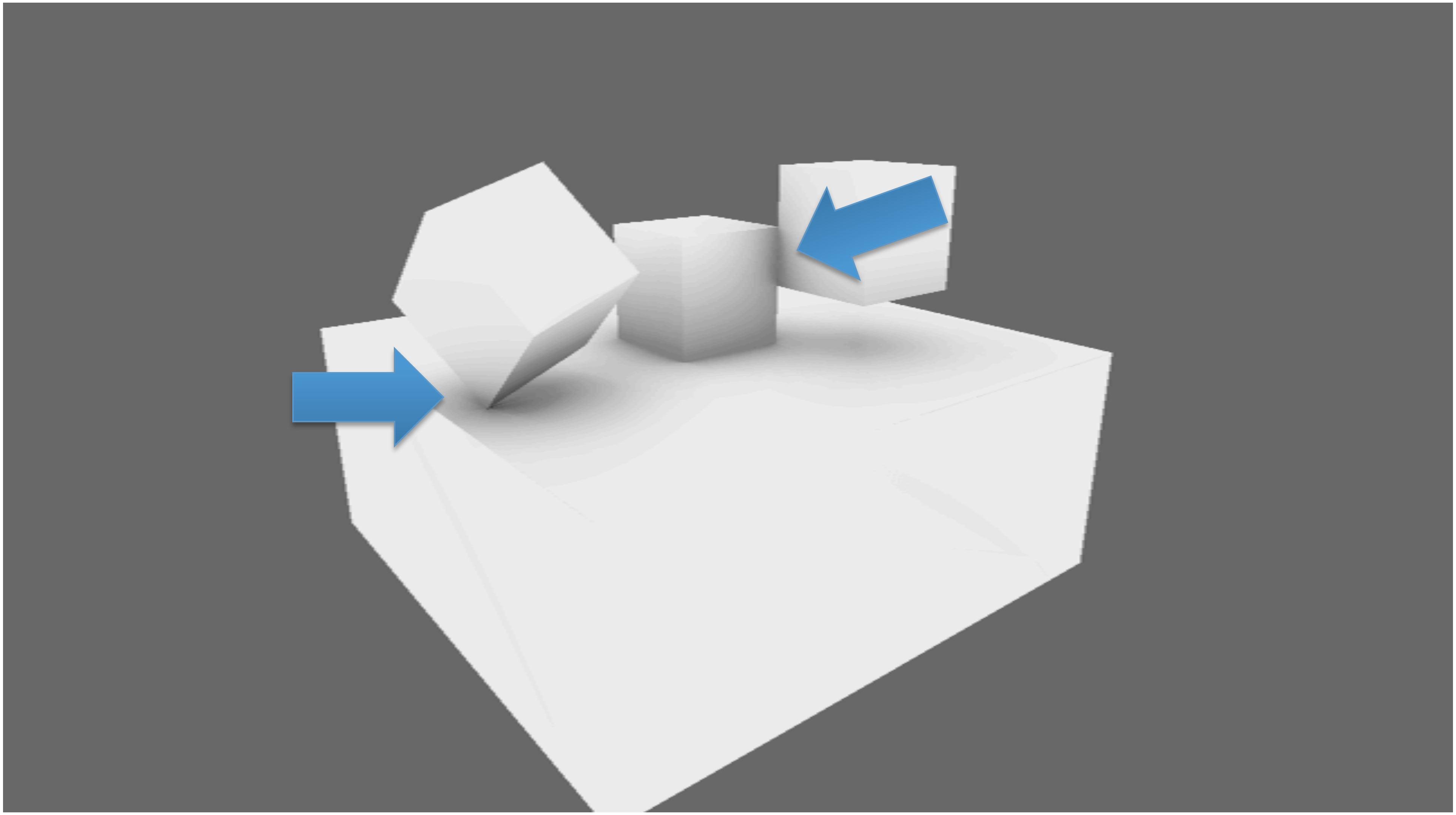


New Ambient Occlusion Volume

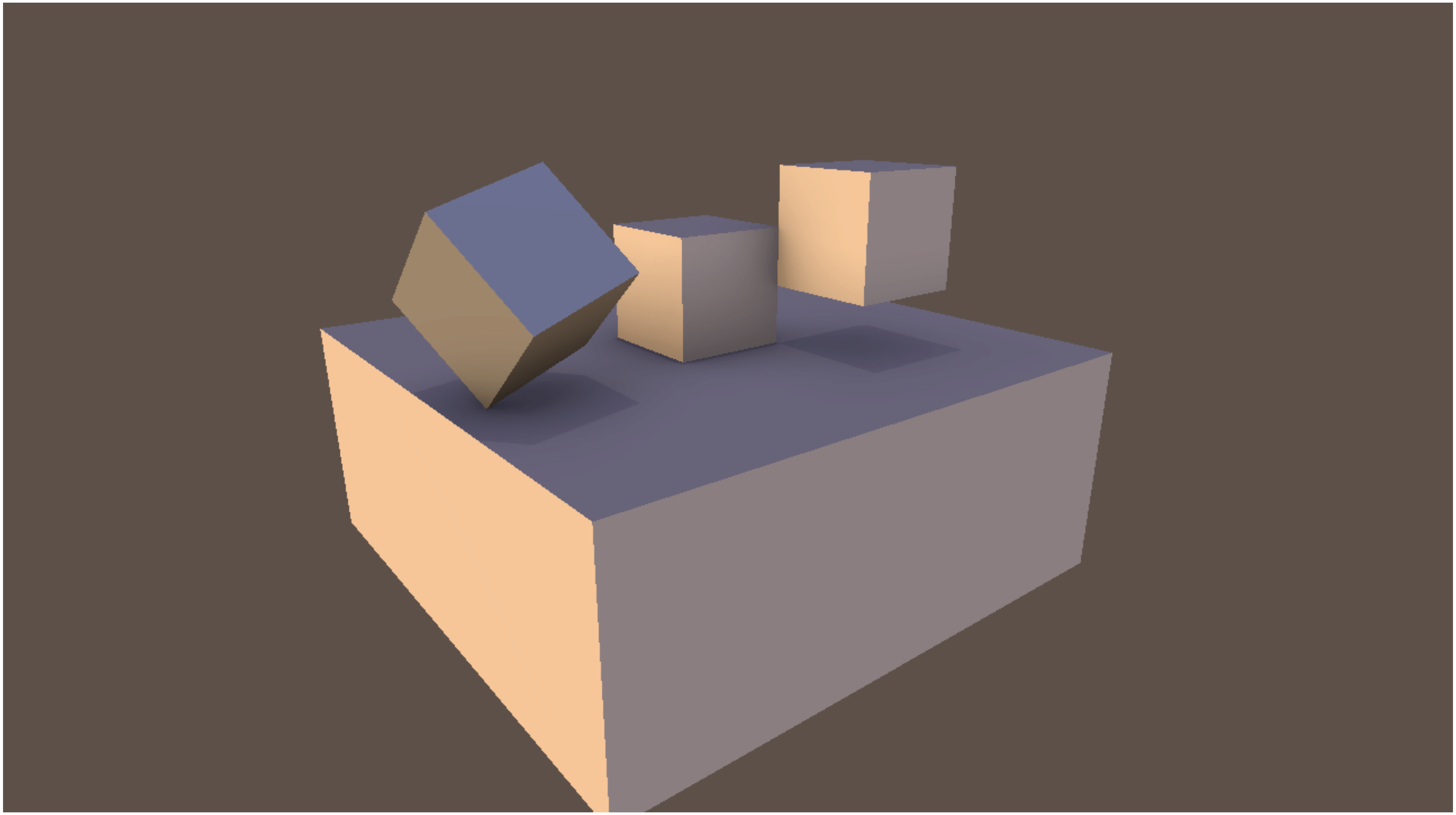
Geometry + AOVs



Accessibility Factor



Final Shading Result



Details

Quality

- Partial coverage
- Disable near plane clipping

  Use “z-fail” depth test or full-screen quad if camera position is inside AOV
- Compensation map

Performance

- Precompute volumes for static geometry
- Sparse sampling

RESULTS

Ray traced AO

- "Ground truth" baseline
- 8-cores, BVH

Crytek SSAO [Mittring '07, Kajalin '09]

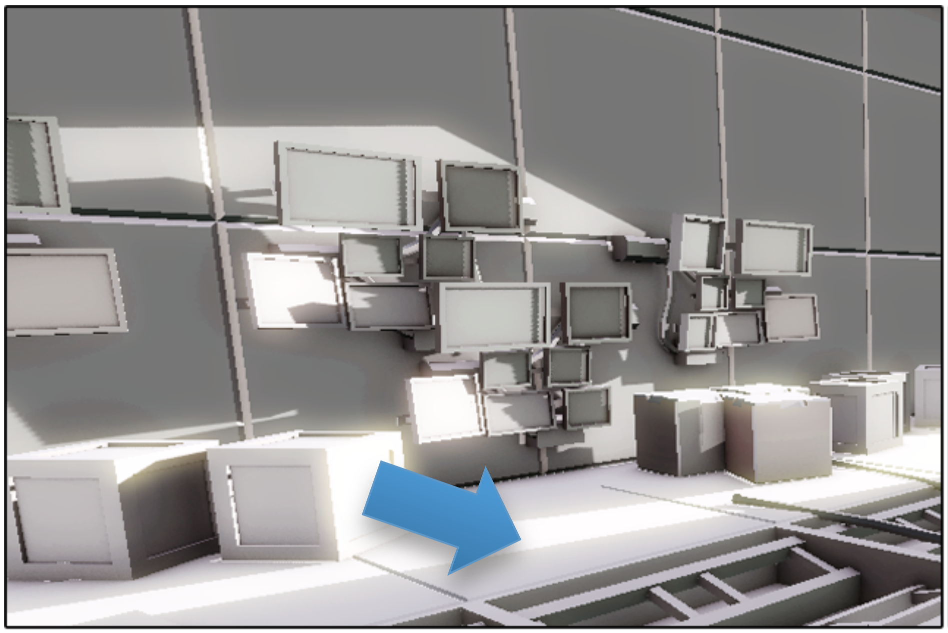
- Popular industry technique
- Artistically driven

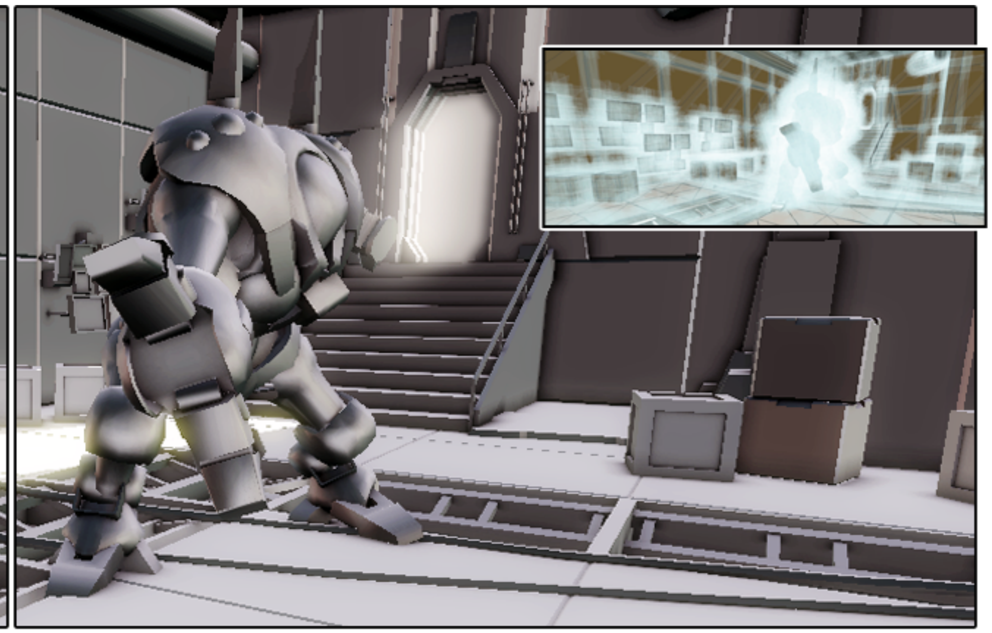
Volumetric AO [Szirmay-Kalos et al. '09]

- Fastest screen space method
- Physical based

AOV (This paper)

Qualitative





Ray traced AO

124309 ms = 2 minutes

Real time AOV

23 ms

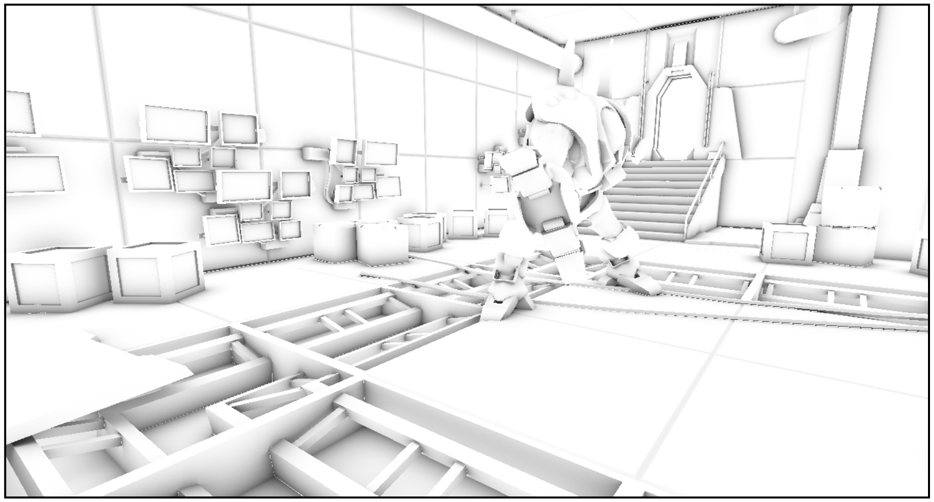
(or 6 ms with 3x3 subsample)

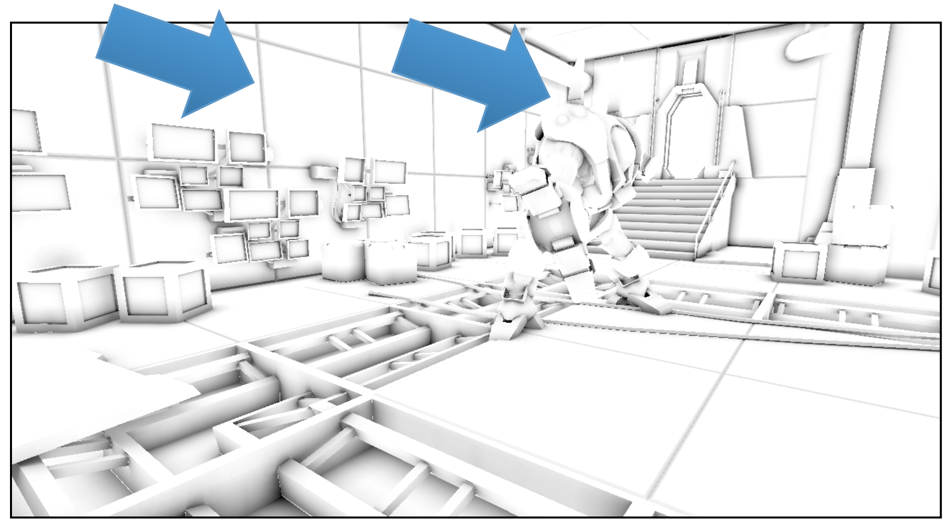
1.4 million triangles, 1280x720 pixels, δ= 0.42m

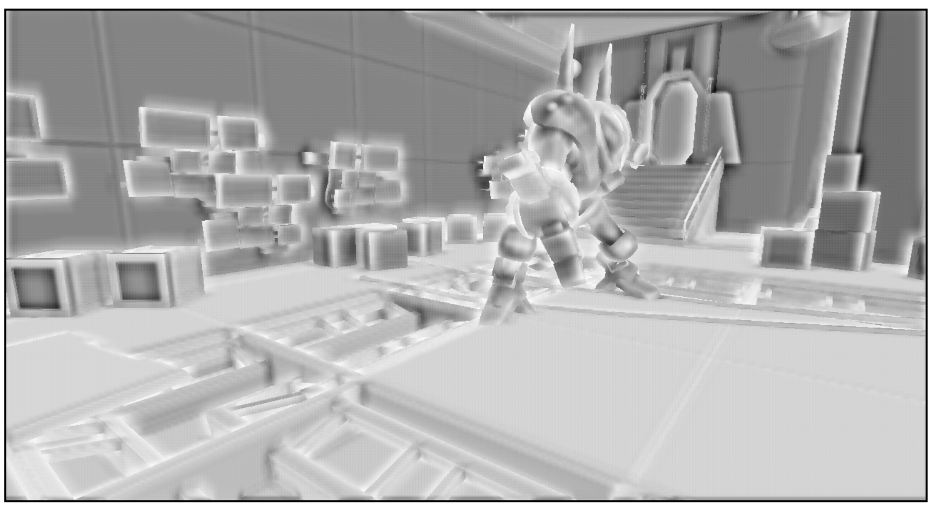
“Secret War” scene from *Marvel Ultimate Alliance 2* courtesy Vicarious Visions

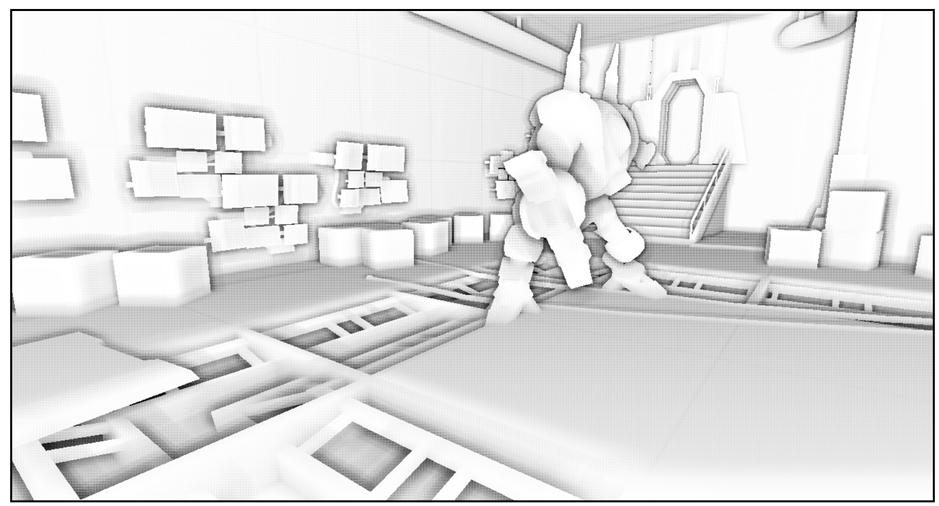
Bounding Volume Wireframe

Quality Comparison

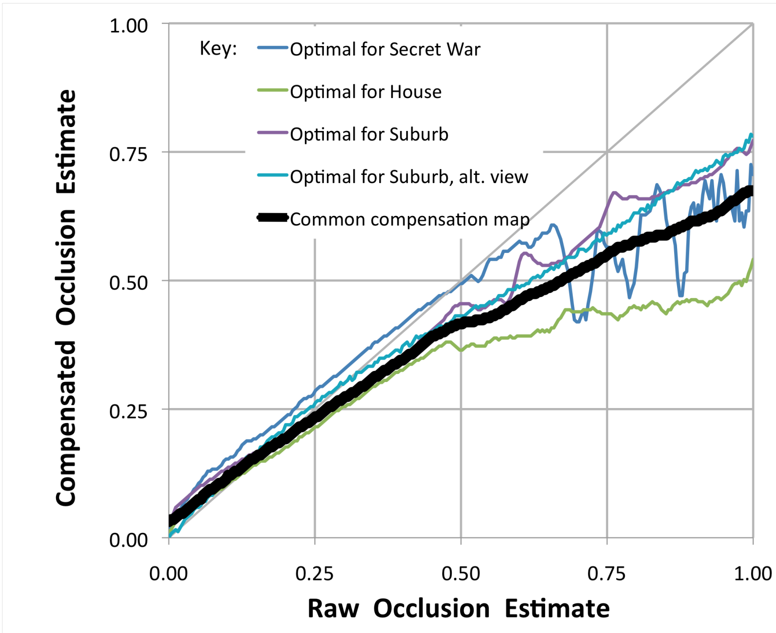

Ray trace

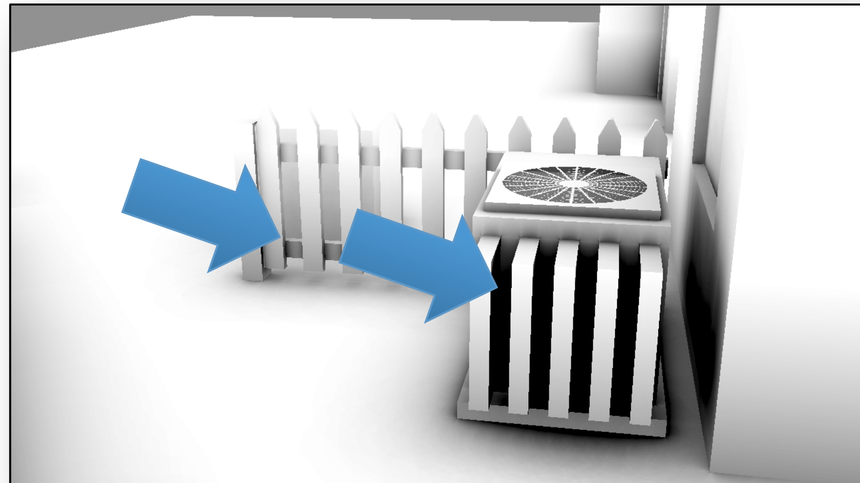

AOV


Crytek SSAO

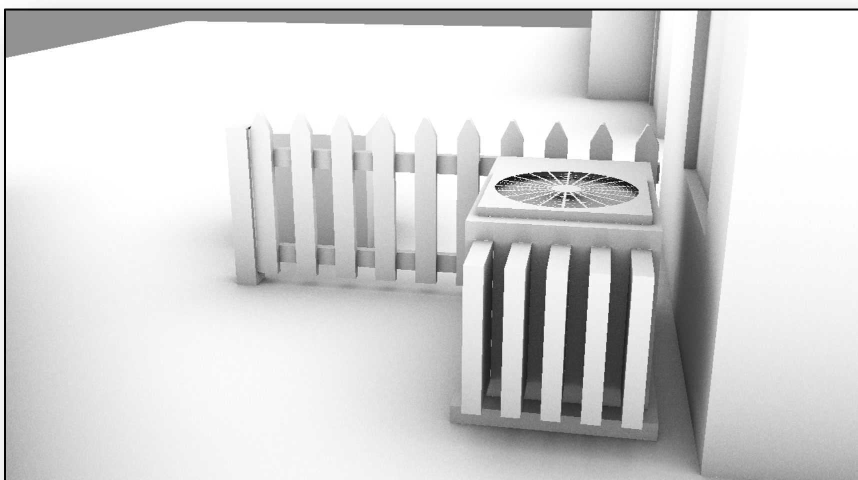

Volumetric

Compensation Map

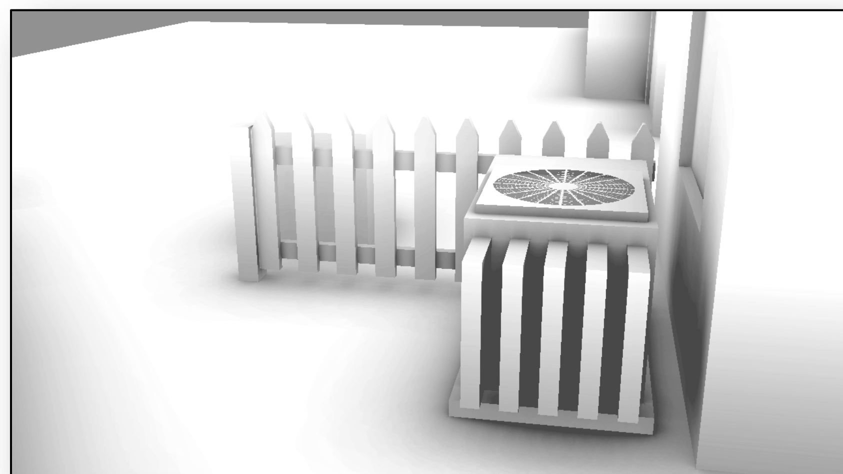




Raw AOV Result

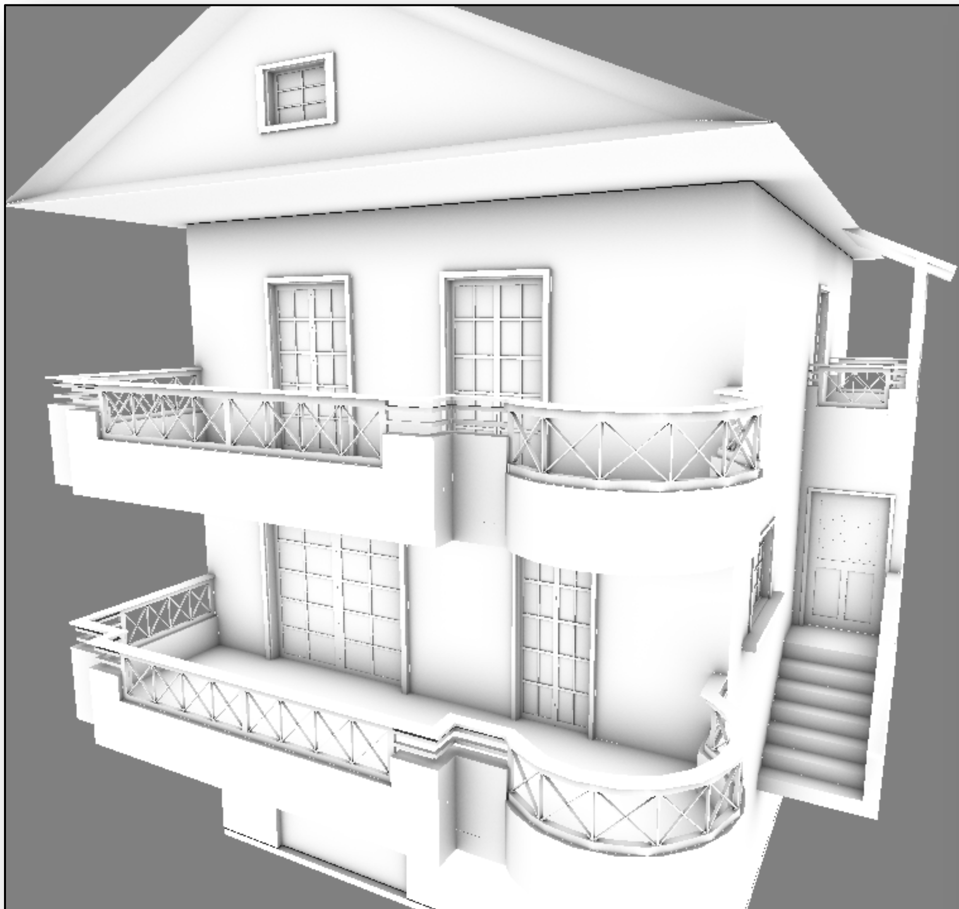


Ray trace

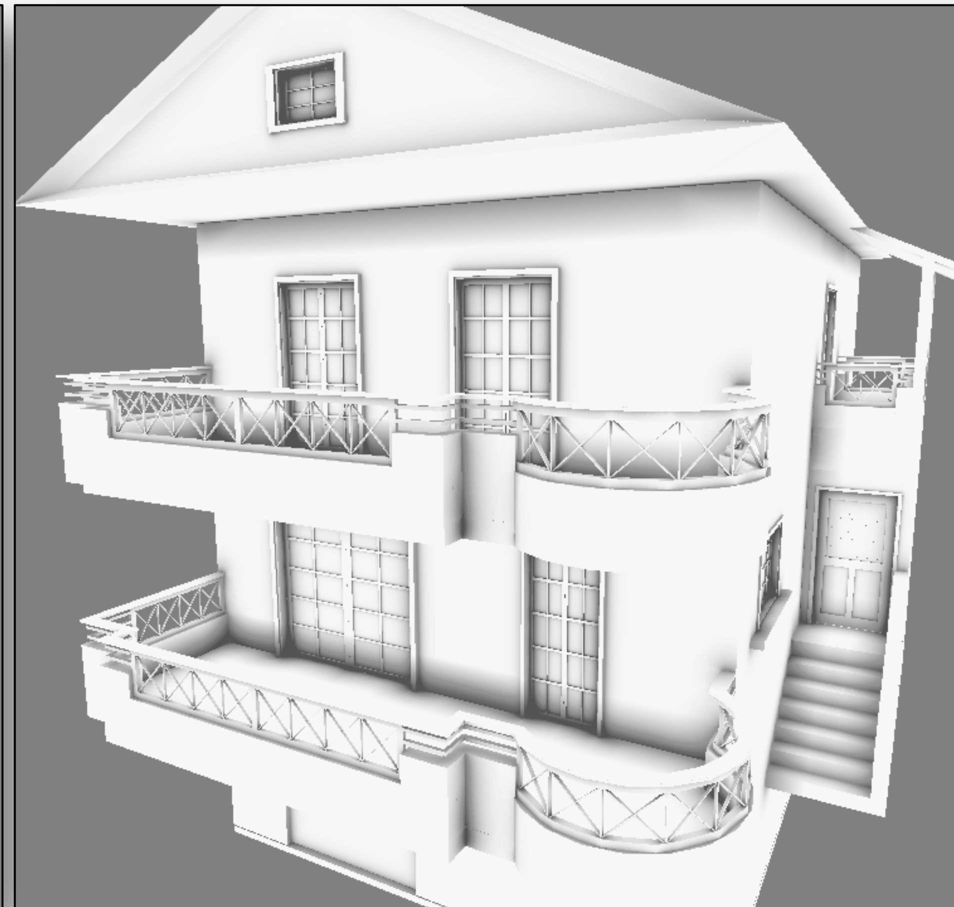


AOV + Compensation

Overdarkening Artifact

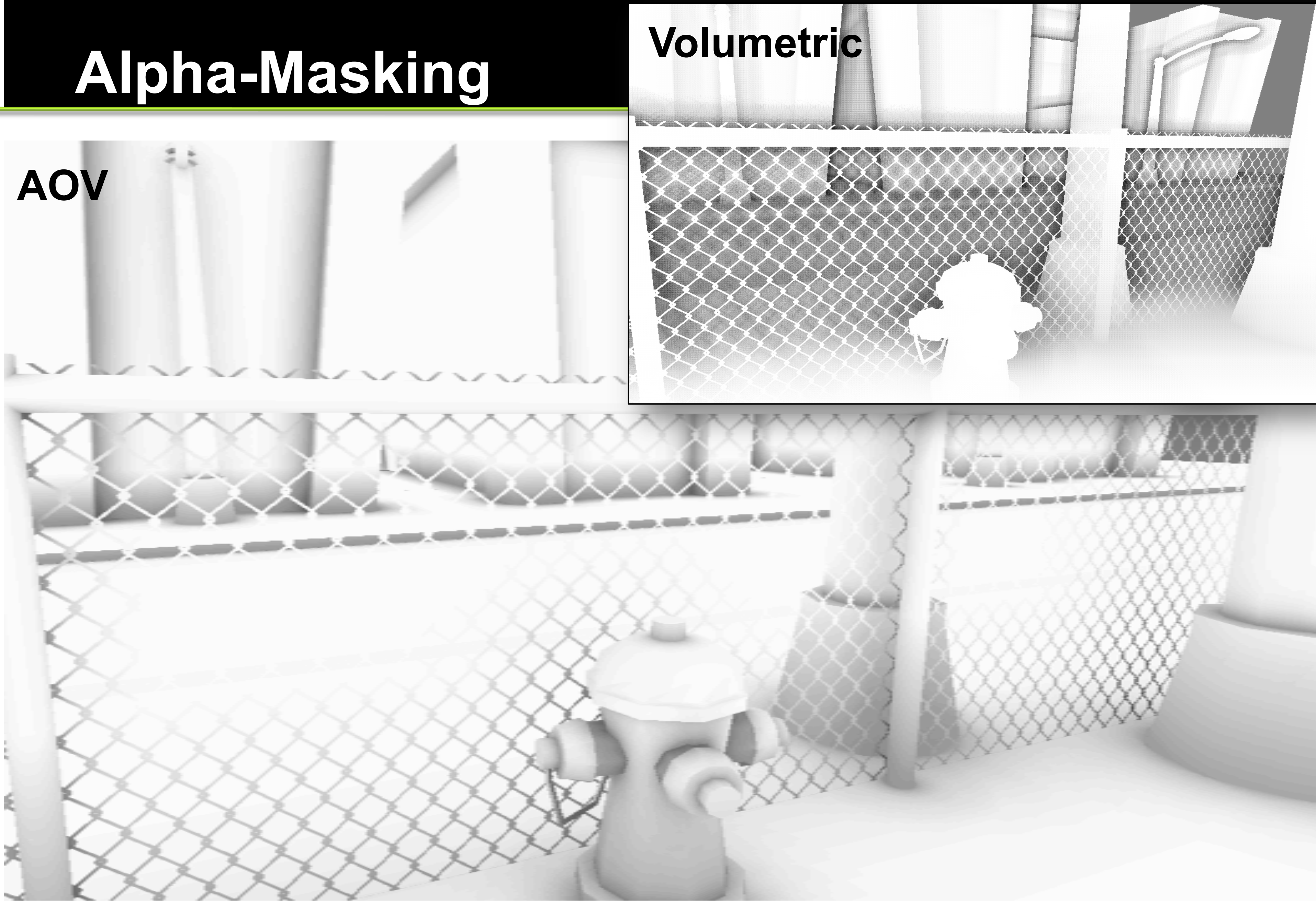


Ray trace

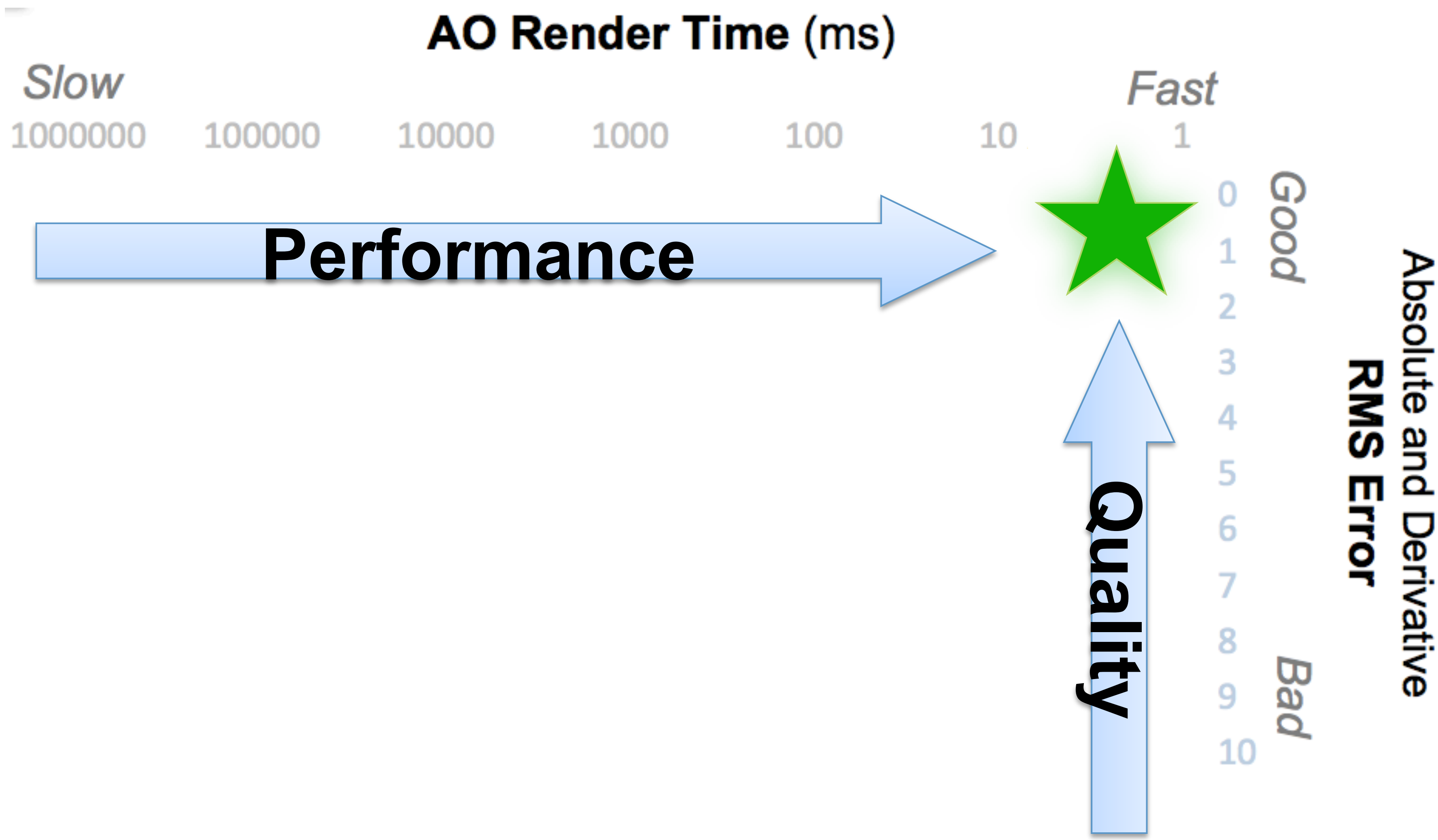


AOV

Alpha-Masking

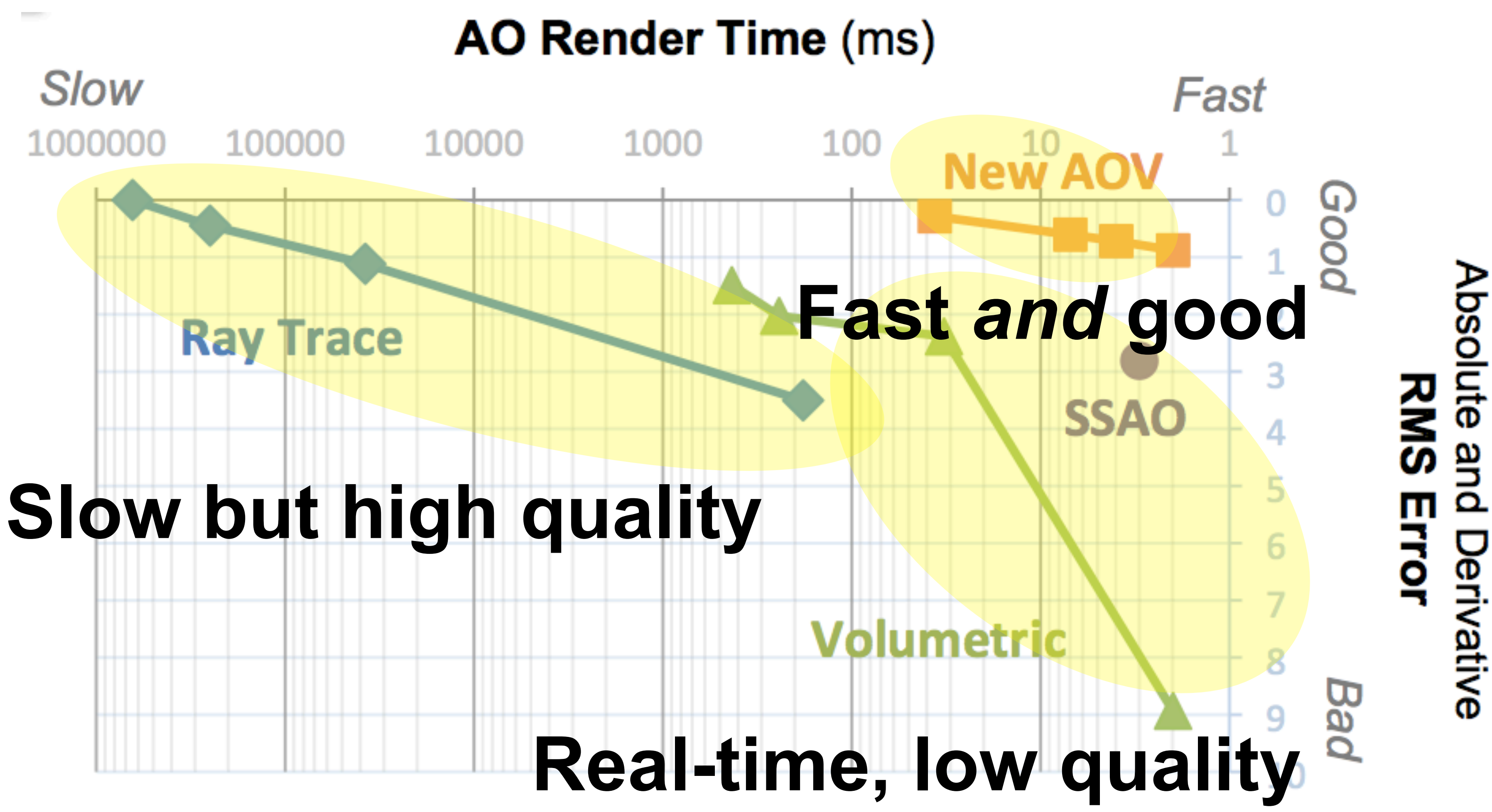


Quality vs. Performance (varying sample rate)



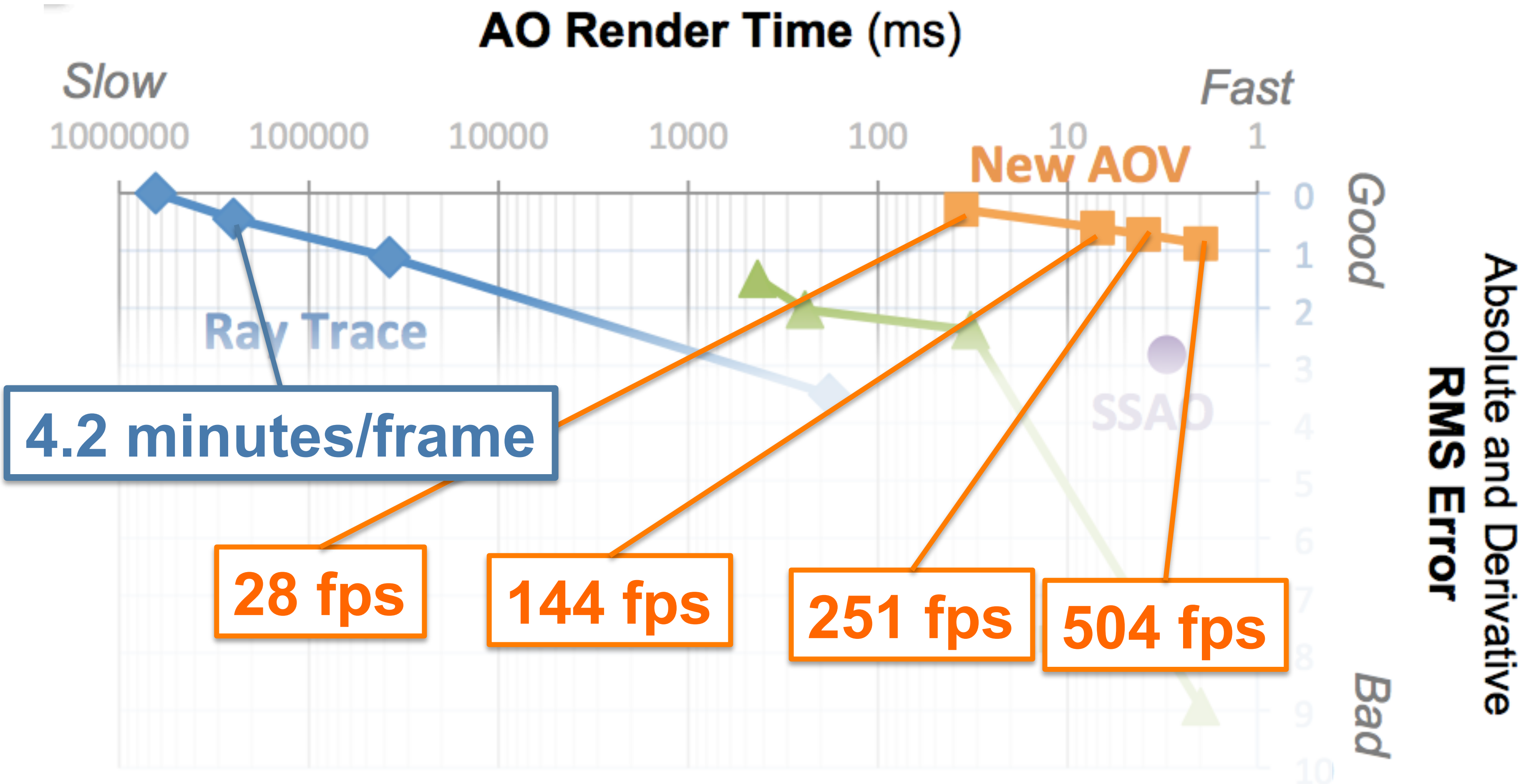
NVIDIA GeForce GTX 480, Dual Quad-core Intel CPU

Quality vs. Performance (varying sample rate)



NVIDIA GeForce GTX 480, Dual Quad-core Intel CPU

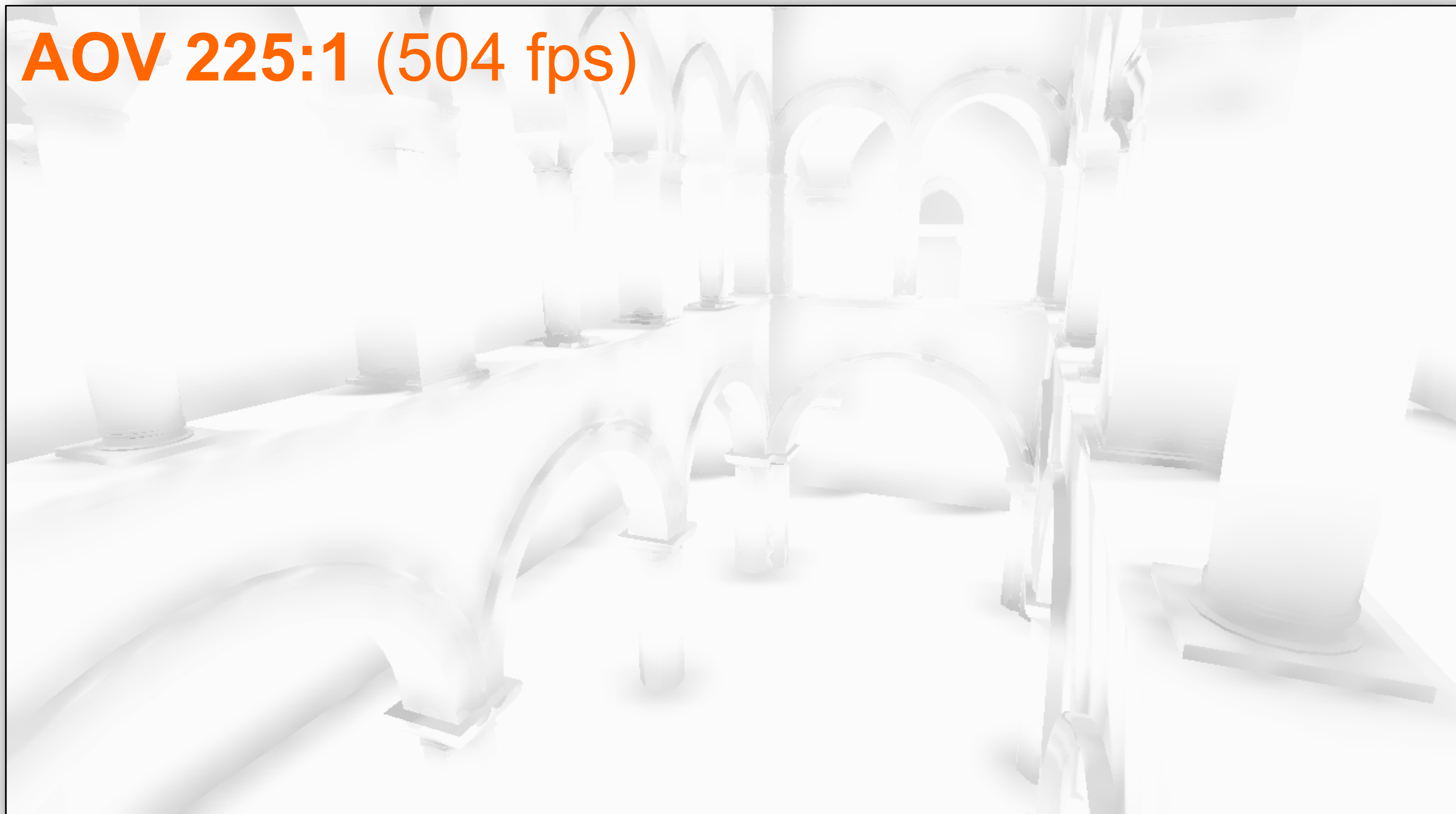
Quality vs. Performance (varying sample rate)



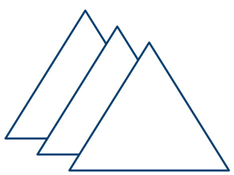
AOV Performance at Ray Trace Quality Level

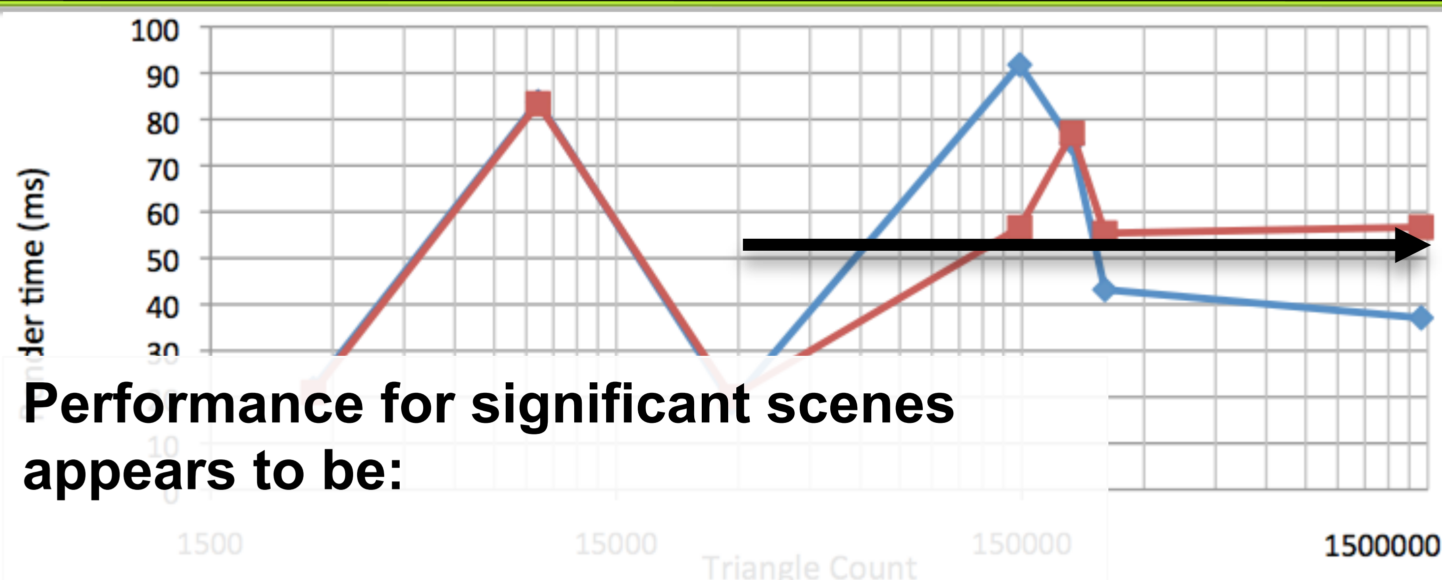
NVIDIA GeForce GTX 480, Dual Quad-core Intel CPU

Quality vs. Performance



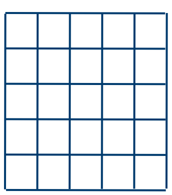
Render Time vs. Triangles, vs. Pixels

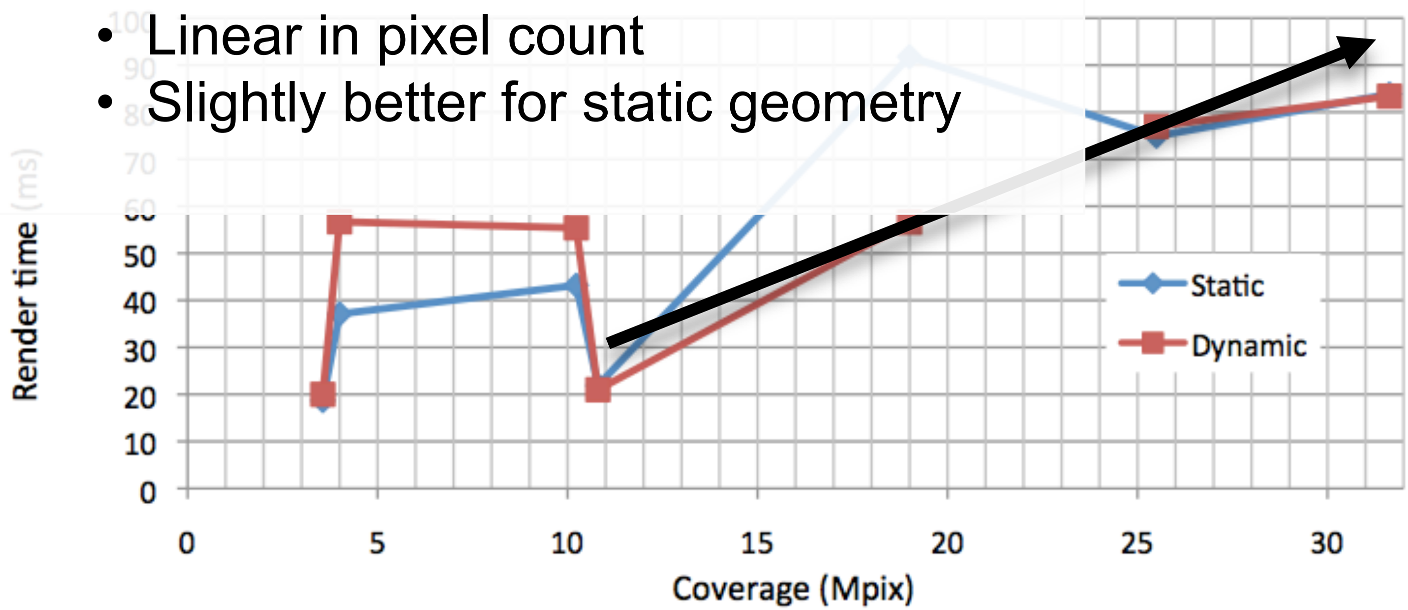




Performance for significant scenes appears to be:

- Independent of triangle count
- Linear in pixel count
- Slightly better for static geometry



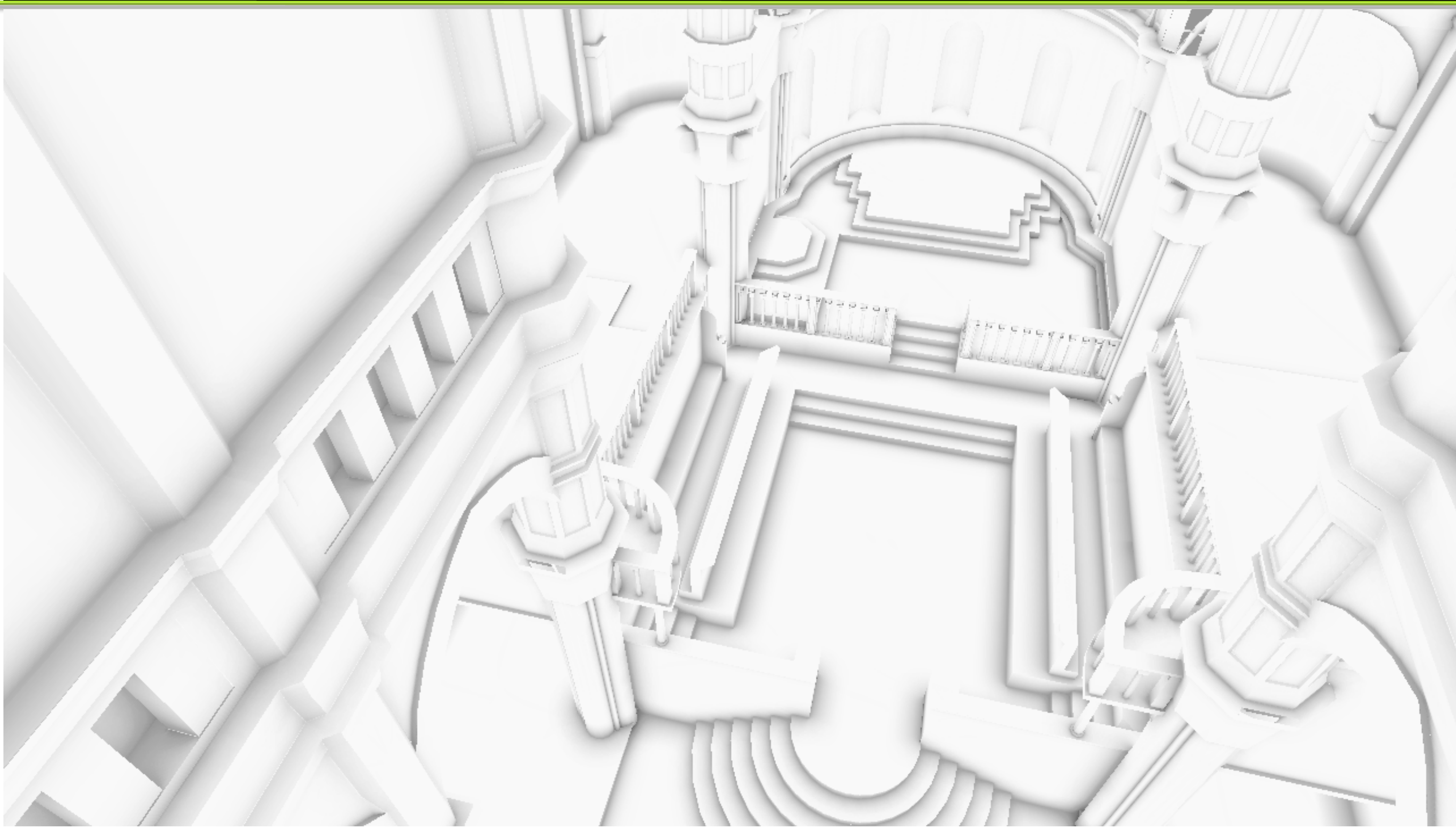


δ = 6.0m

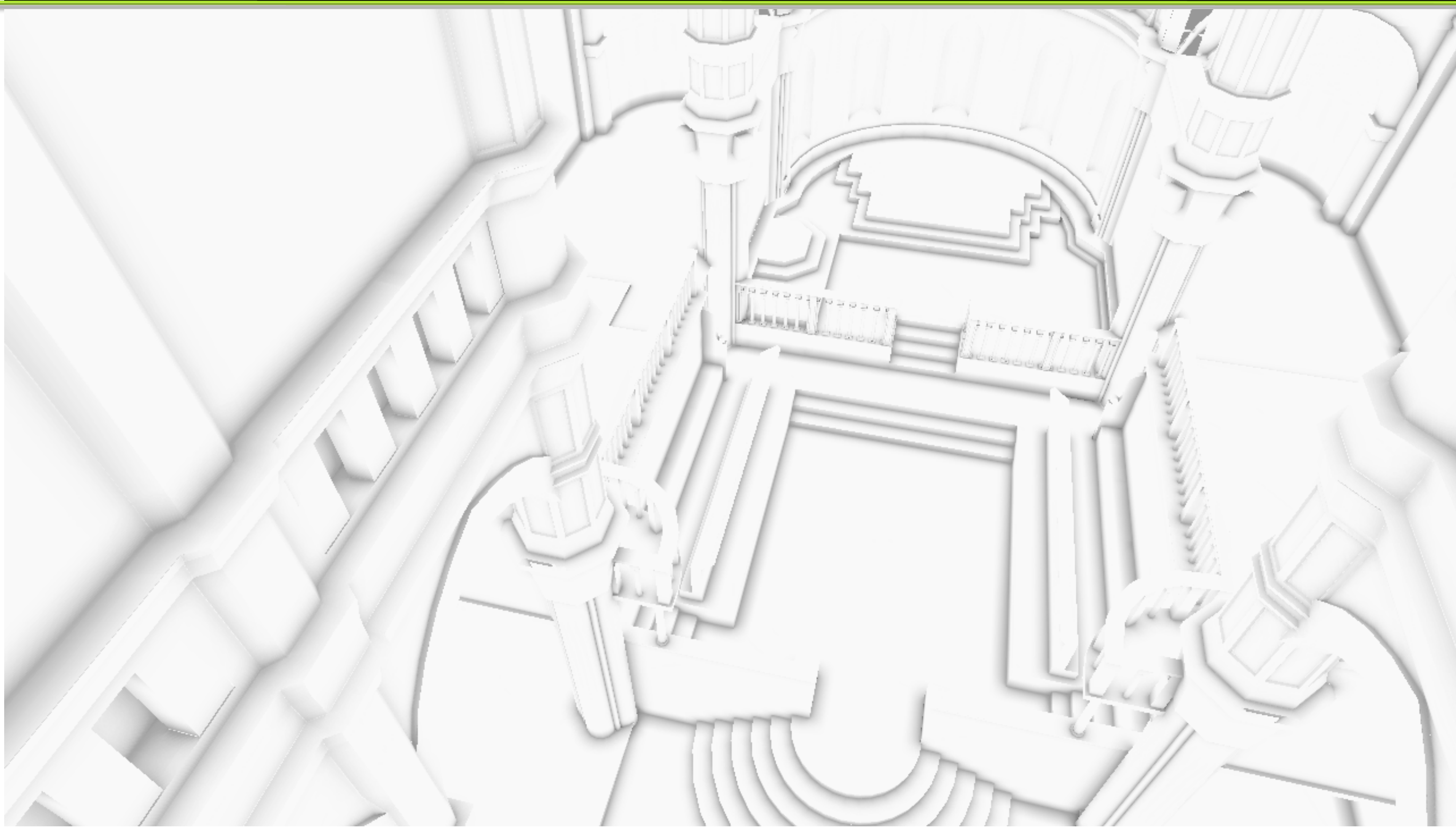
δ = 4.0m

δ = 2.0m

δ = 1.0m

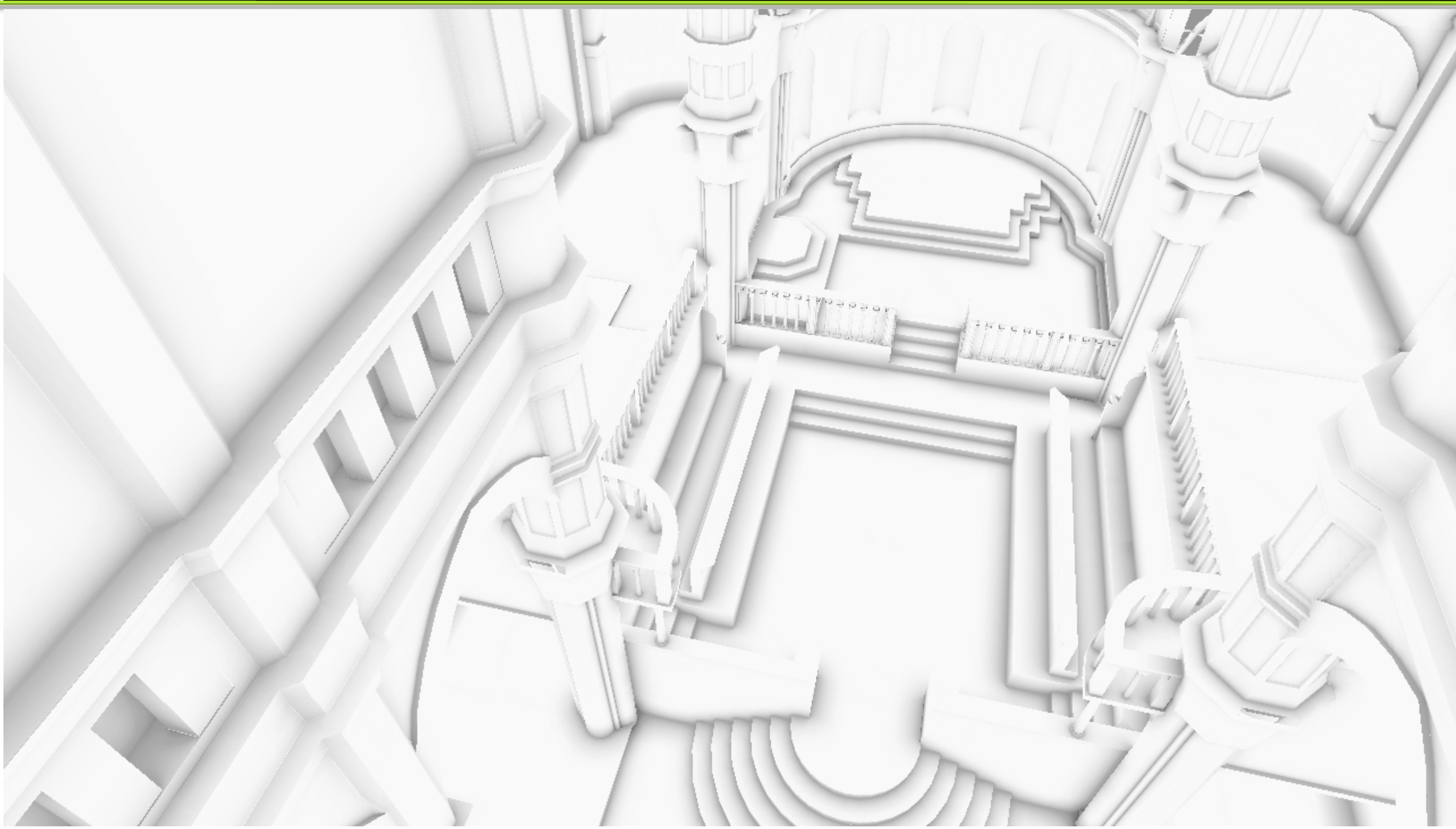


δ = 0.5m



δ = 0.1m

δ = 1.0m

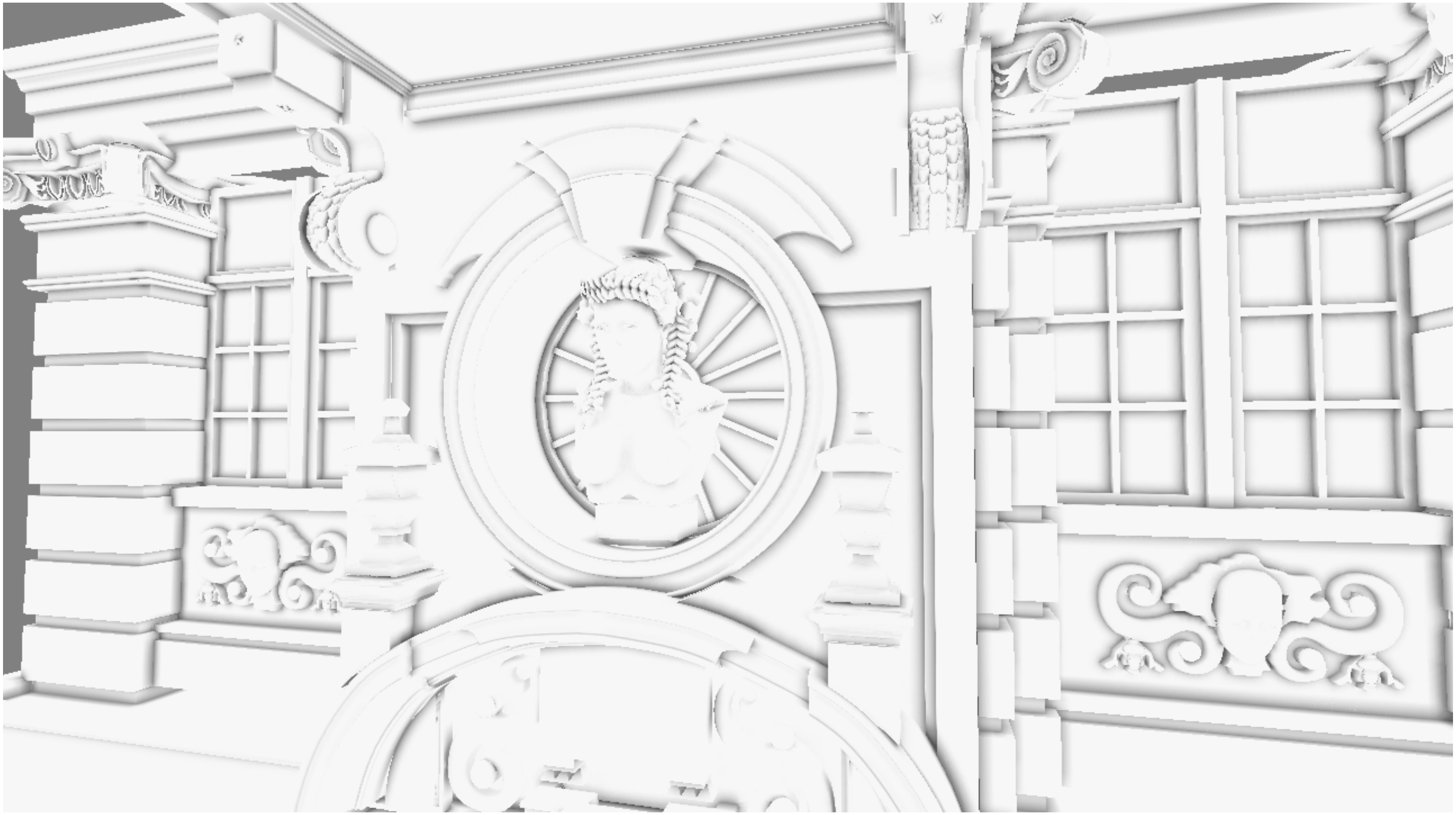


2-24 ms

Foliage (150k triangles, with α)

3-20 ms

Architectural Detail (690k sliver triangles)



3-17 ms

Summary

Presented physical basis for AO and other real-time terms

- Made clear previously implicit approximations and assumptions
- How to combine with real-time GI methods without overlap

Two tricks (AO ... s

- Compute tig...
- Compensati...
- Includes α f...
- Upsampling...

High-quality AO...

- Analytic: vie...
- Comparable...
- Comparable to ray traced AO quality

Acknowledgements

Peter Shirley, NVIDIA

David Luebke, NVIDIA

Tom Garrity, Williams College

Corey Taylor, Electronic Arts

Joakim Carlsson, Chalmers

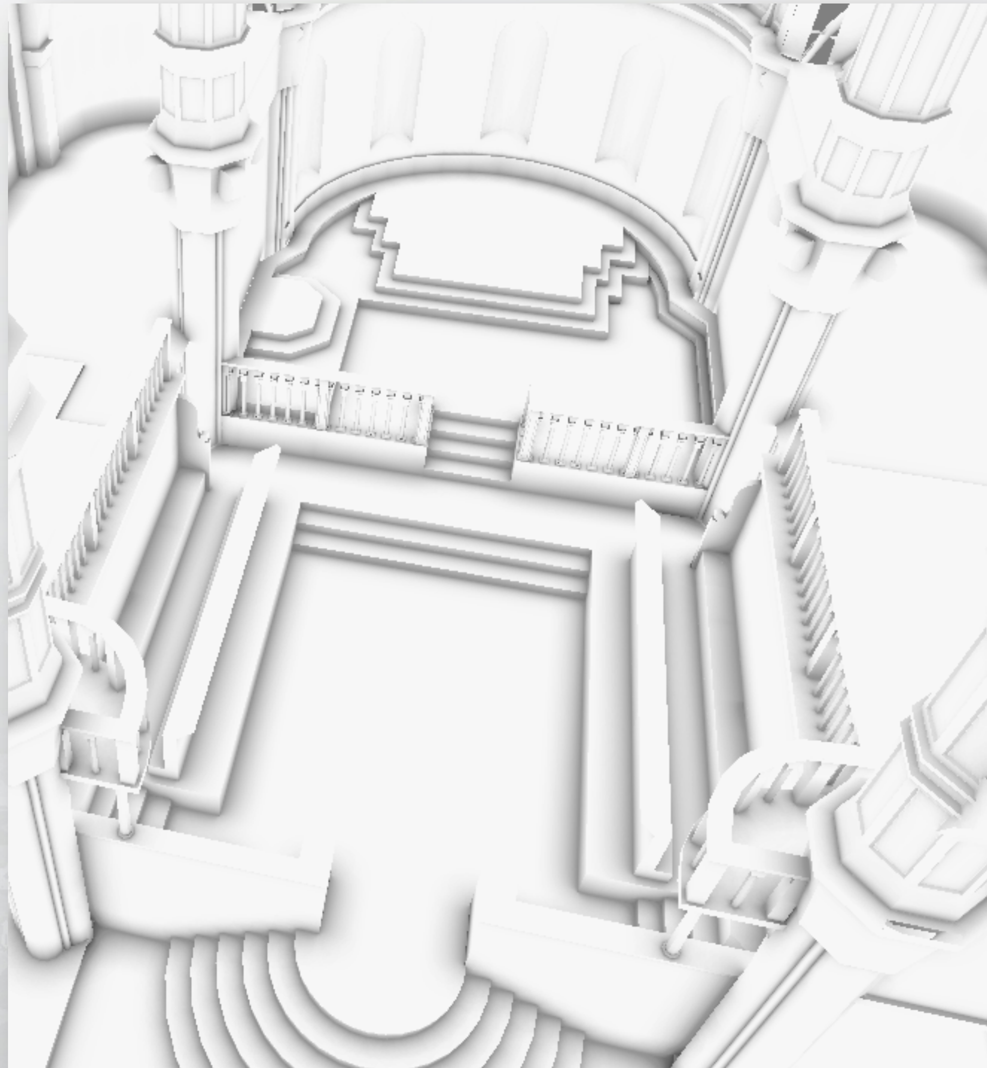
Patrik Sjölin, Chalmers

Louis Bavoil, NVIDIA

Chris Wassum, Vicarious Visions

Max McGuire, Unknown Worlds

Marko Dabrovic



ADDITIONAL MATERIAL

Future Directions

Integrate with real-time global illumination

- Corrects missing high frequencies in ISPM and VPL approaches
- Natural extension to *near-field* indirect illumination: accumulate both light and occlusion

Performance improvements

- Ongoing work with Louis Bavoil at NVIDIA
- LOD and hexaprism from Laine and Karras EGSR ’10

Real-world artist controls

- AO caster/receiver flags
- Per-surface falloff function
- Delta as a function of distance from camera

Algorithm

1. Render x, n geometry buffers [Saito and Takahashi 91]
2. Bind **Accessibility** buffer, clear to 1
3. Set subtractive blending mode
4. **Draw Call + Vertex Shader**: *as if making a regular pass*
5. **Geometry Shader**:

   For each polygon P with vertices $\{p_0, \ldots, p_{k-1}\}$:

   1. Compute prism B bounding $g_P(x) > 0$
   2. If camera is inside B, then emit full screen quad, else emit B
6. **Pixel Shader**:
   1. Read position x and normal n
   2. Compute $g = g_P(x)$; discard if $g \leq 0$
   3. Clip P to the local tangent plane, generating quad $P' = \{p_0', \ldots, p_3'\}$
   4. Output to $\mathrm{AO}_{P'}(x)$ to blender

Quality vs. Performance (varying sample rate)

Method	Samples	Belgium, *Architecture*, 687,054 tris			Boxes, *Correctness*, 144 tris		City, *Alpha*, 9,624 tris			House, *Architecture*, 28,866 tris			Trees, *Foliage*, 148,101 tris		
		GF 280 (ms)	GF 480 (ms)	Error (σ^2)			GF 280 (ms)	GF 480 (ms)	Error (σ^2)	GF 280 (ms)	GF 480 (ms)	Error (σ^2)	GF 280 (ms)	GF 480 (ms)	Error (σ^2)
Ray Trace	5000	490803.0	490803.0	0.00	76103.3	0.00	603202.0	603202.0	0.00	283226.0	283226.0	0.00	691334.0	691334.0	0.00
	1941	190687.0	190687.0	0.35	29712.1	0.12	234515.0	234515.0	0.65	109923.0	109923.0	0.23	270322.0	270322.0	0.28
	292	28779.9	28779.9	1.02	4528.6	0.27	35372.4	35372.4	1.47	16574.2	16574.2	0.60	40803.0	40803.0	0.80
	1	156.4	156.4	6.83	79.5	1.06	167.9	167.9	6.06	87.7	87.7	2.88	195.0	195.0	6.06
AOV (new)	1	77.7	16.95	0.59	1.9	0.06	137.3	46.343	0.46	25.9	9.530999	0.27	100.3	19.435	2.03
	$1/3^2 \cdot 5^2$	41.6	5.744	1.03	0.3	0.06	20.5	7.213	0.51	7.2	2.318	0.69	38.0	6.881	2.40
	$1/5^2 \cdot 5^2$	28.1	2.15	1.55	0.1	0.06	8.9	3.432	0.72	5.6	1.538	0.78	28.7	4.729	2.74
	$1/15^2 \cdot 5^2$	11.9	2.62	2.28	0.1	0.08	2.9	1.231	1.10	2.9	1.031	0.90	12.8	2.821	3.14
Volumetric	1024	895.4	347.277	2.19	371.0	0.20	1035.3	496.336	3.96	473.7	401.605	1.20	742.8	386.223	4.32
	256	224.3	223.38	2.50	93.6	0.36	259.3	258.587	4.75	119.0	118.88	1.47	186.8	186.032	4.98
	32	29.3	29.334	4.11	12.4	1.03	33.6	33.723	7.03	15.6	16.572	2.42	24.4	24.343	6.35
	1	3.1	1.966	6.65	1.4	2.26	3.2	2.009	12.89	1.7	1.955	4.58	2.4	1.908	12.26
Crytek	$16 \cdot 4^2$	15.6	?	4.34	11.8	0.52	15.6	?	3.82	12.8	?	1.68	14.3	?	2.98

Method	Samples	Secret War, *Video Game*, 1,445,620 tris			Sibenik, *Architecture*, 240,090 tris			Sponza, *Architecture*, 199,362 tris			Suburb, *Worst Case*, 2,688 tris		
		GF 280 (ms)	GF 480 (ms)	Error (σ^2)	GF 280 (ms)	GF 480 (ms)	Error (σ^2)	GF 280 (ms)	GF 480 (ms)	Error (σ^2)	GF 280 (ms)	GF 480 (ms)	Error (σ^2)
Ray Trace	5000	556228.0	556228.0	0.00	514959.4	514959.4	0.00	642571.0	642571.0	0.00	401222.0	401222.0	0.00
	1941	216142.0	216142.0	0.89	200163.7	200163.7	0.57048	249890.0	249890.0	0.45	155504.0	155504.0	1.05
	292	32564.8	32564.8	2.07	30220.5	30220.5	1.49404	37627.1	37627.1	1.12	23439.2	23439.2	2.19
	1	161.1	161.1	9.38	155.3	155.3	6.40968	180.9	180.9	3.50	130.7	130.7	8.98
AOV (new)	1	31.2	30.99	0.43	?	23.568	0.51684	110.1	36.337	0.28	31.7	22.178	0.65
	$1/3^2 \cdot 5^2$	21.3	6.379	0.73	?	4.64	0.93845	32.2	6.96	0.61	5.2	4.72	0.68
	$1/5^2 \cdot 5^2$	18.9	5.598	0.88	?	3.152	1.16105	21.8	3.977	0.72	2.8	2.781	0.74
	$1/15^2 \cdot 5^2$	4.6	5.044	1.16	?	1.875	1.59577	10.2	1.988	0.88	1.4	1.465	0.85
Volumetric	1024	954.8	395.758	1.62	?	431.225	6.79809	967.9	431.225	1.49	1050.1	1049.041	1.34
	256	252.1	238.379	2.55	?	242.471	2.37686	242.9	242.471	2.03	265.2	263.368	2.28
	32	31.0	31.79	4.74	?	32.474	5.6291	238.6	32.474	2.39	34.4	34.58	4.29
	1	3.1	1.932	9.94	?	1.99	10.1938	3.1	1.99	8.93	3.2	3.149	11.36
Crytek	$16 \cdot 4^2$	15.6	?	2.85	?	?	2.91351	15.7	?	2.81	15.5	?	2.76

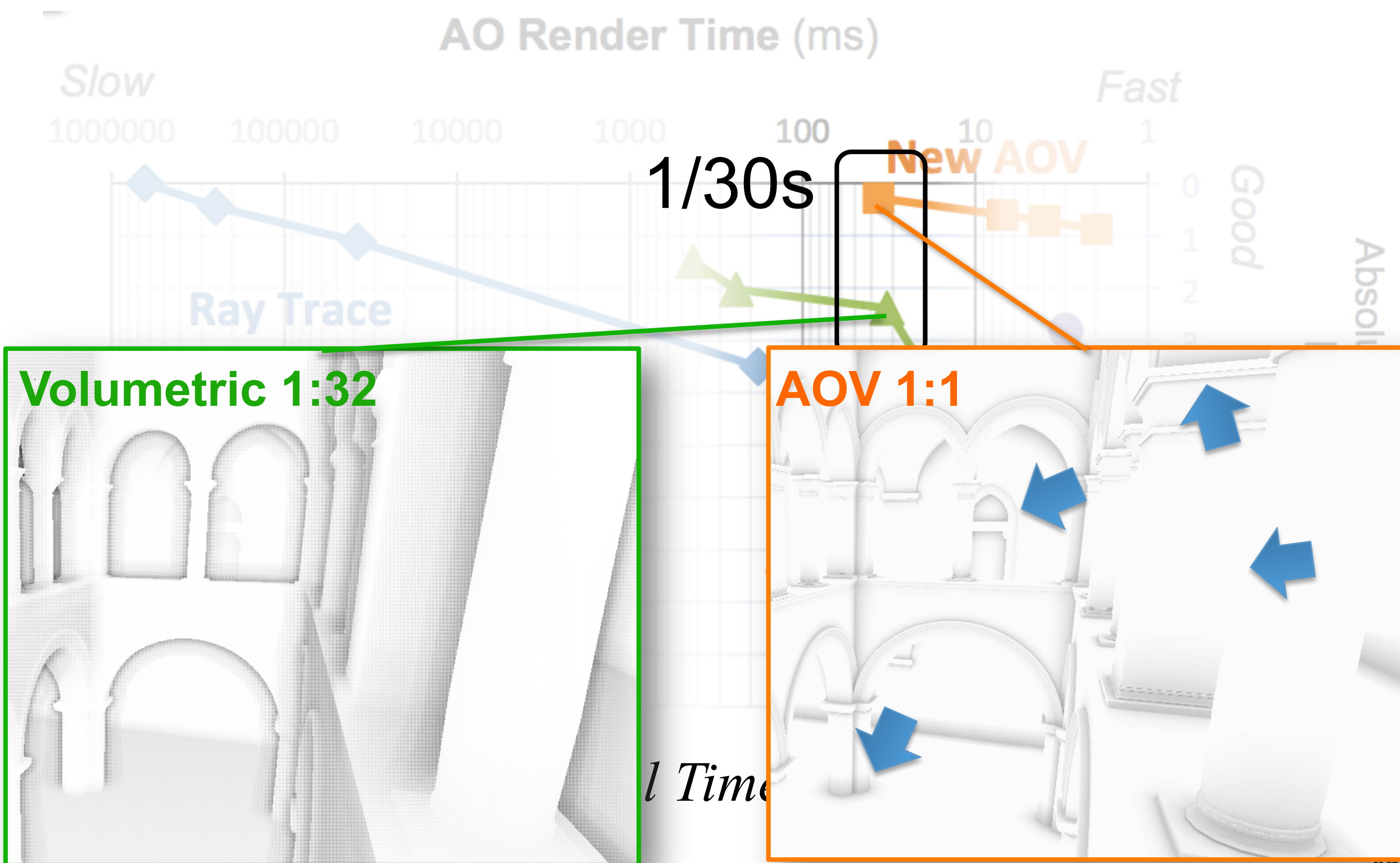
Color Key: Fast 10ms 33ms 100ms > 200ms ... Slow

Exact Inaccurate

Images, 3D models, and data at
http://graphics.cs.williams.edu

500 fps

Quality vs. Performance (varying sample rate)



Geometry Shader

$$\vec{e}_{0\leq i<k} = \begin{bmatrix} \hat{m}_i \\ \hat{m}_{(i-1+k) \bmod k} \\ \hat{m}_k \end{bmatrix}^{-1} \begin{bmatrix} -\delta \\ -\delta \\ 0 \end{bmatrix}$$

$$\vec{e}_{k\leq i<2k} = \vec{e}_{i-k} - \delta\hat{m}_k$$

Operation Cost: Pixel Shader

Read x, n

2 fetch

Clip P

Maximum: 3 dot, 4 if, 11 move, 2 divide, 2 vec3 lerp = 21 arith, 2 div, 4 if

Falloff function

3 vec3 sub, 4 dot, 1 clamp, 1 vec4 madd, 4 scalar mul = 29 arith, 1 clamp4

Form factor

Maximum 4 vertices * (1 cross, 5 dot, 2 rsqrt, 4 scalar mul, 1 fetch)
100 scalar arith, 8 rsqrt, 4 fetch

significant scalar ops: 150 arith, 2 divide, 6 fetch, 4 if