High Performance Graphics 2013

19.7.2013

SCREEN-SPACE FAR-FIELD AMBIENT OBSCURANCE

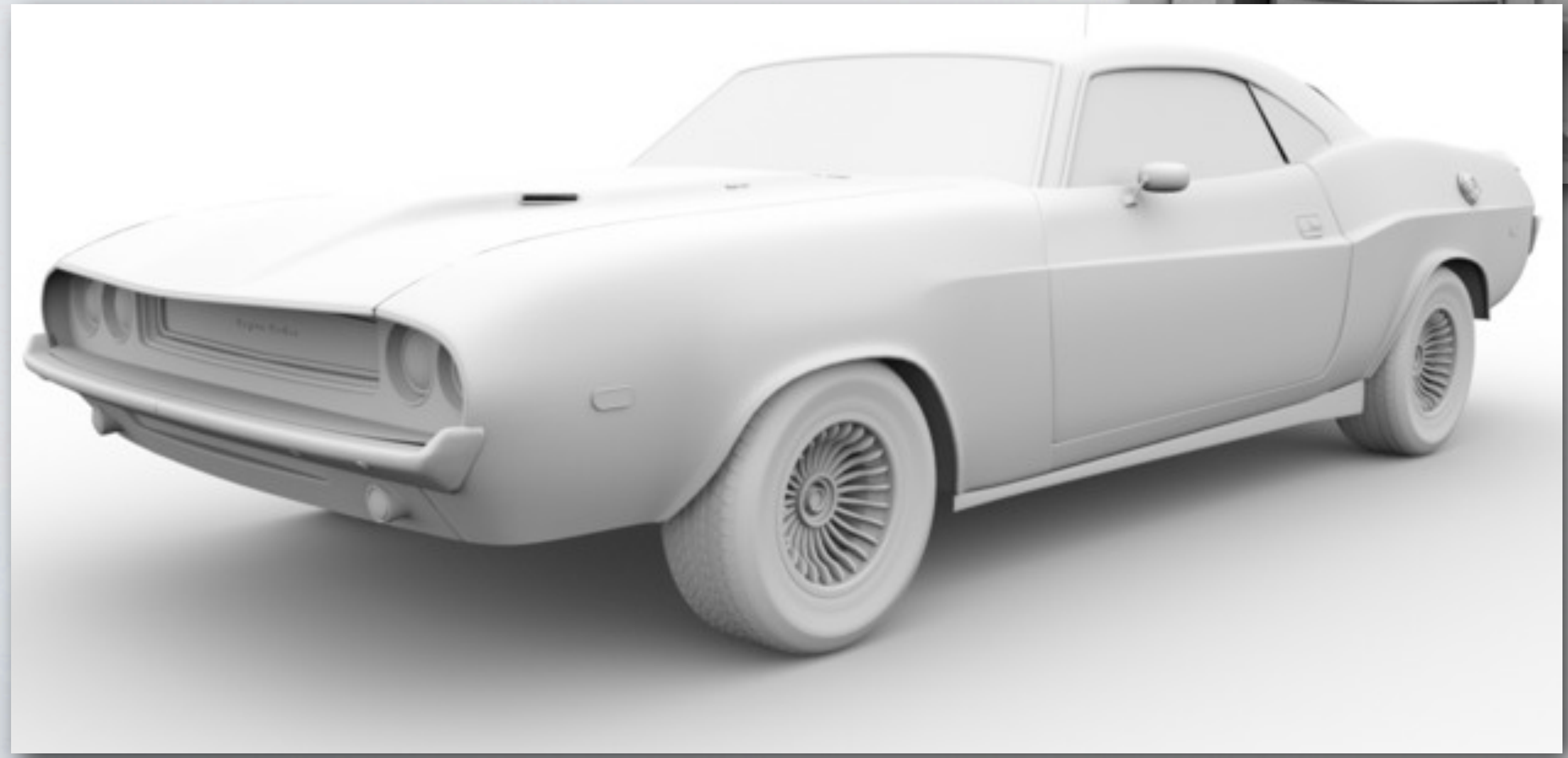
Ville Timonen

Åbo Akademi University

CONTENTS

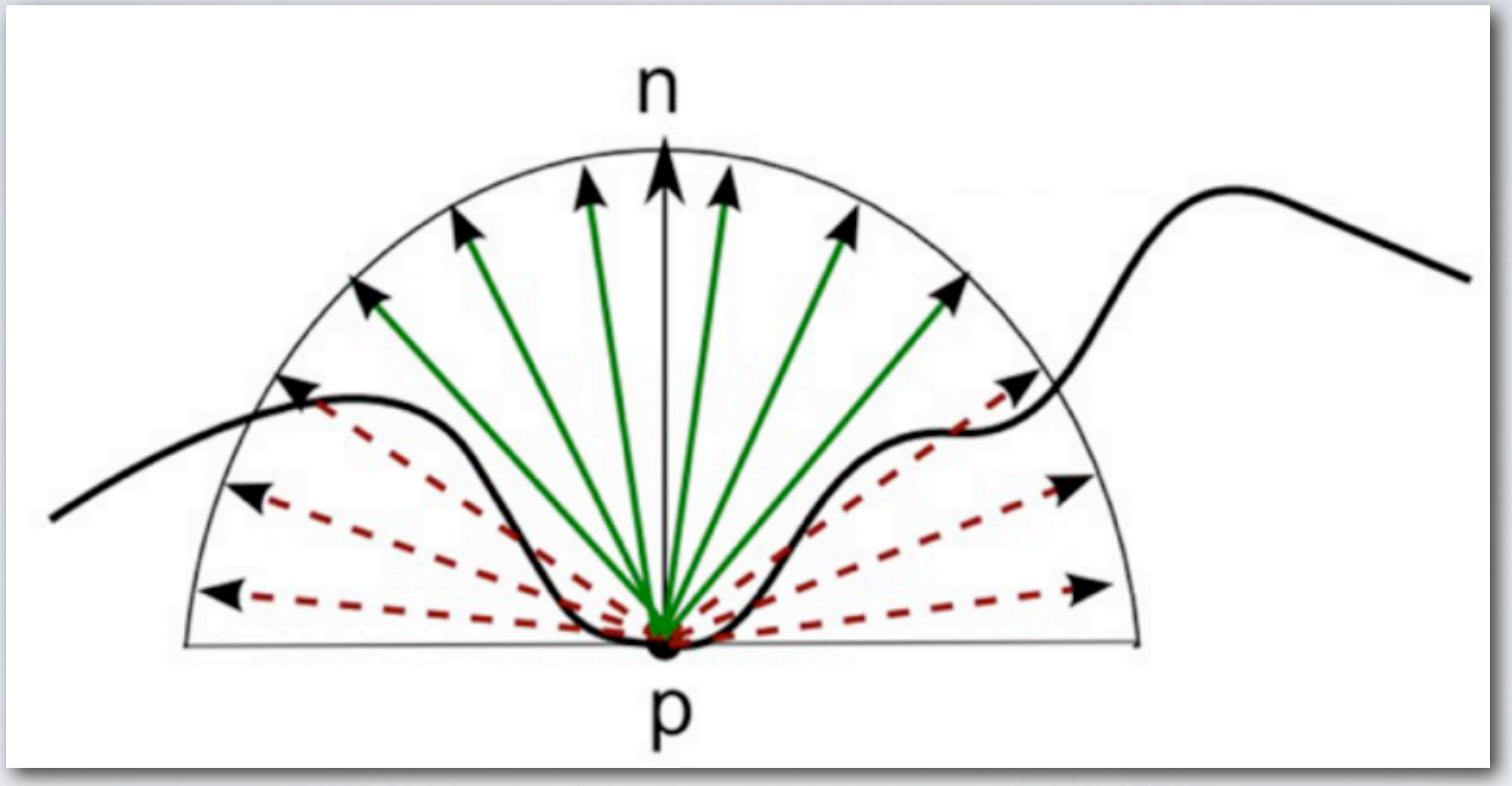
1 AMBIENT OBSCURANCE

Is an approximation for global illumination



1 AMBIENT OBSCURANCE

Defined as the cosine and falloff weighted hemisphere visibility



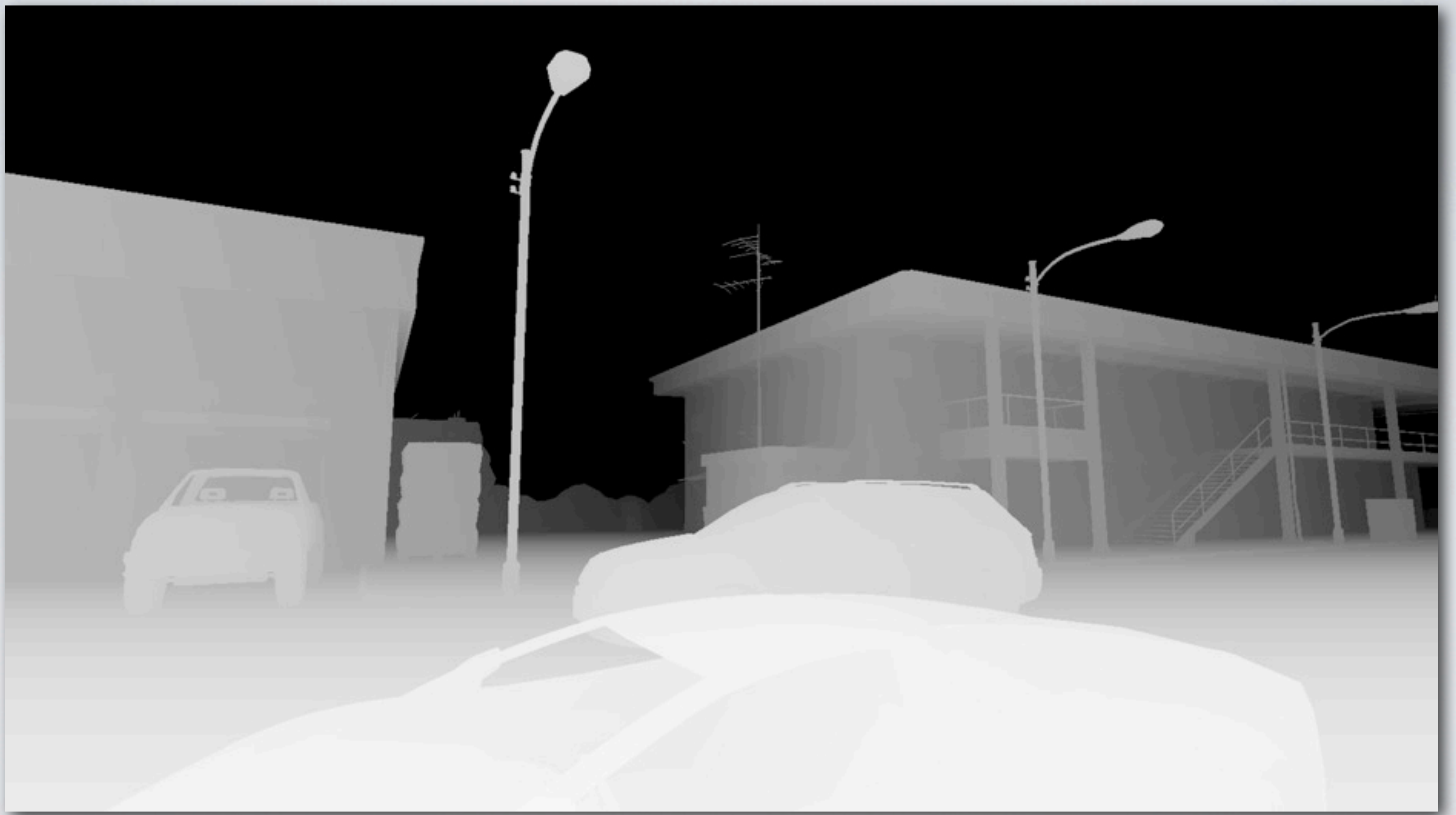
$$A(\mathbf{p},\vec{n}) = \frac{1}{\pi}\int_{\Omega} F(D(\mathbf{p},\vec{\omega}))\,\vec{n}\cdot\vec{\omega}\,d\vec{\omega}$$

Screen-Space Ambient Obscurance

- General AO solutions quite not real-time ready
- Screen-Space methods work on the depth buffer geometry only
    - Constant amount of geometry
    - Restricted geometry makes room for optimizations

1 SCREEN-SPACE AMBIENT OBSCURANCE

This is a depth map (dark = far, light = near)

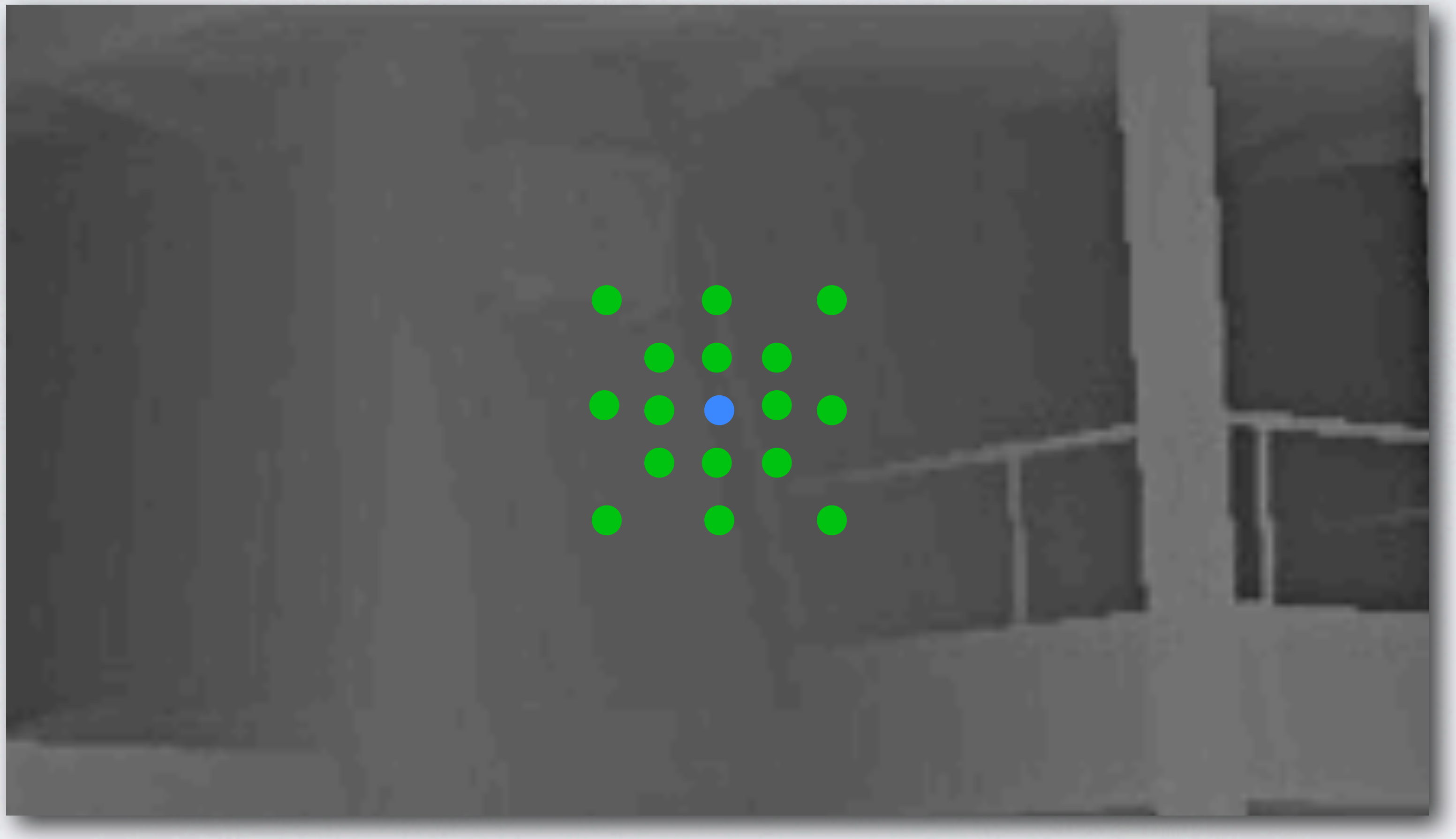


A by-process of most graphics pipelines

1 SCREEN-SPACE AMBIENT OBSCURANCE

So this is what SSAO does:

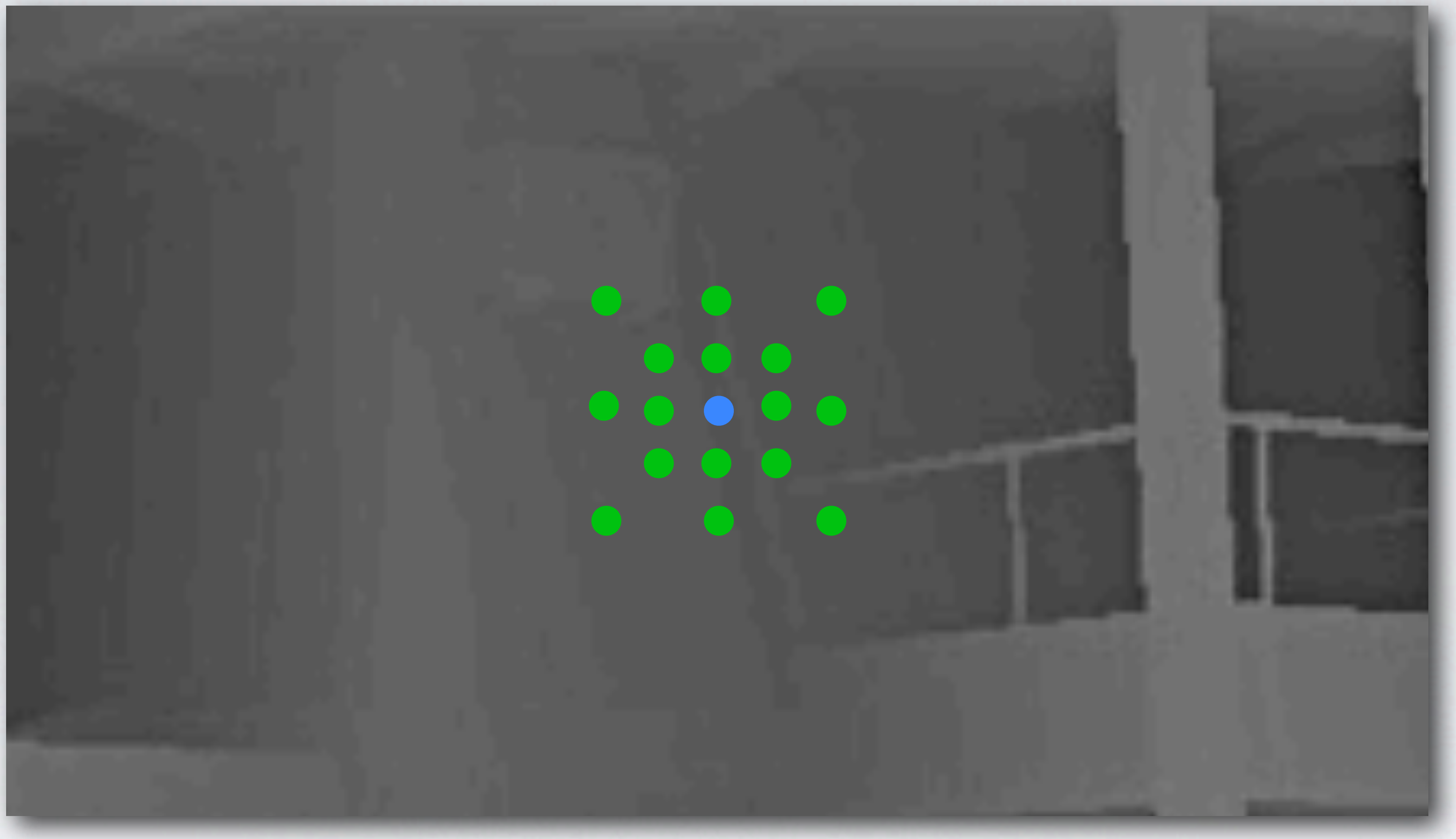
For the blue point...



Sample the surroundings...

1 SCREEN-SPACE AMBIENT OBSCURANCE

So this is what SSAO does:



Deproject points to world space and evaluate AO:

$$A(\mathbf{p}, \vec{n}) = \frac{1}{\pi} \int_{\Omega} F(D(\mathbf{p}, \vec{\omega}))\, \vec{n} \cdot \vec{\omega}\, d\vec{\omega}$$

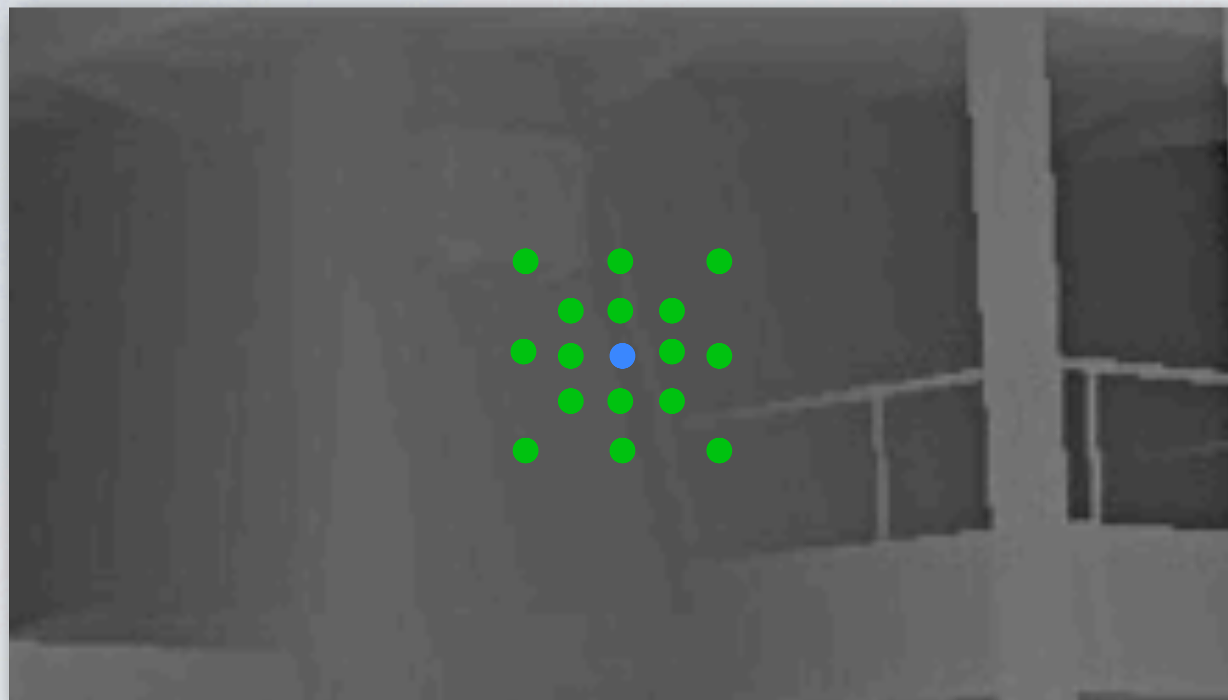
1 SCREEN-SPACE AMBIENT OBSCURANCE

Two components of SSAO

1. Input geometry (i.e. samples of the depth field)
   - This is our main contribution, presentation's topic
2. Obscurance estimator (i.e. how to integrate AO from the samples)
   - Our secondary contribution; supports any falloff function, efficient to evaluate and converges to ray traced reference, beyond presentation's scope

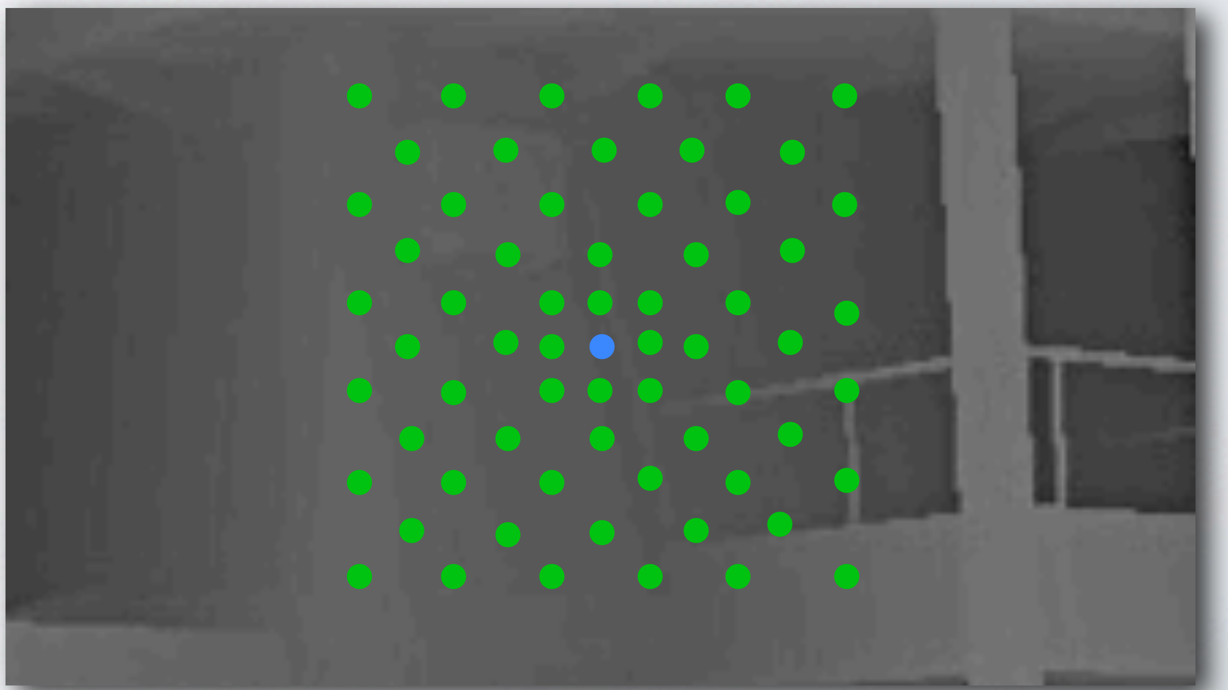
1 SCREEN-SPACE AMBIENT OBSCURANCE

Key problem of SSAO



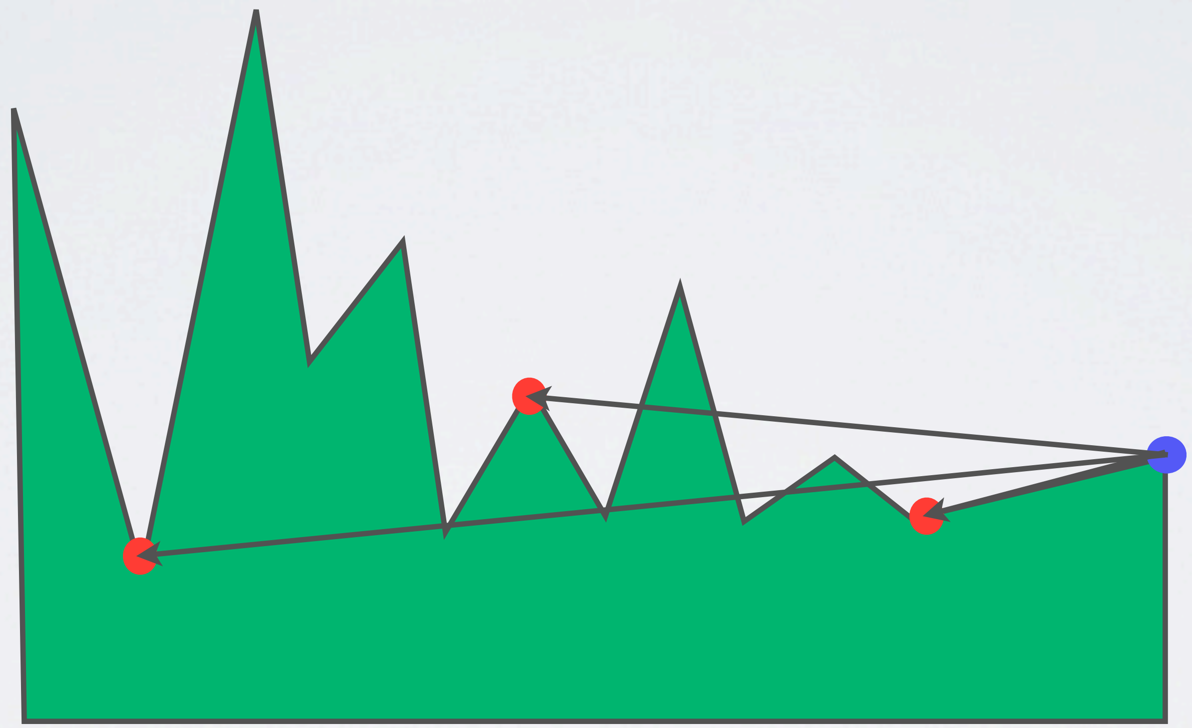
Sampling the near field
is not a problem

But far field easily becomes
prohibitively expensive:
Have to seriously undersample



1 SCREEN-SPACE AMBIENT OBSCURANCE

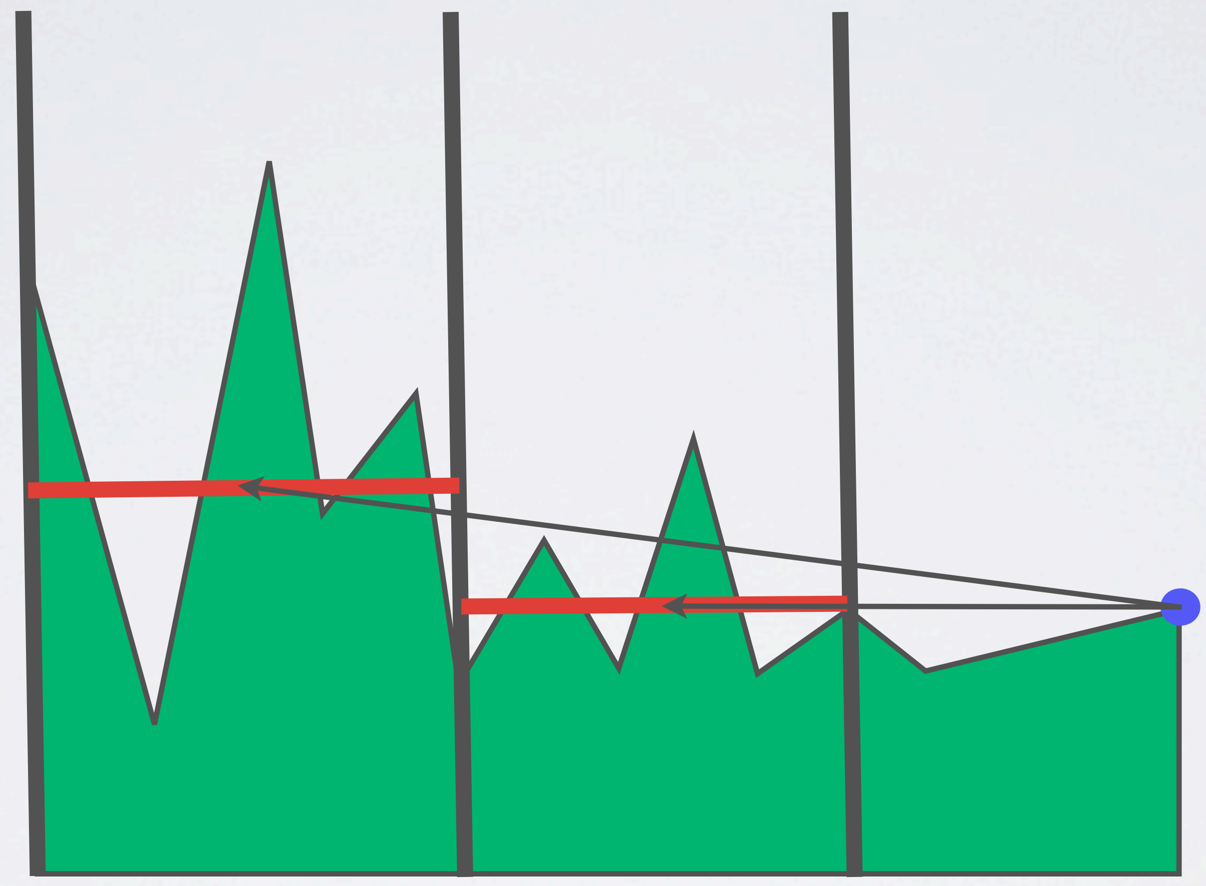
Previous 2 approaches:



Direct depth buffer samples
easily miss important occluders

I SCREEN-SPACE AMBIENT OBSCURANCE

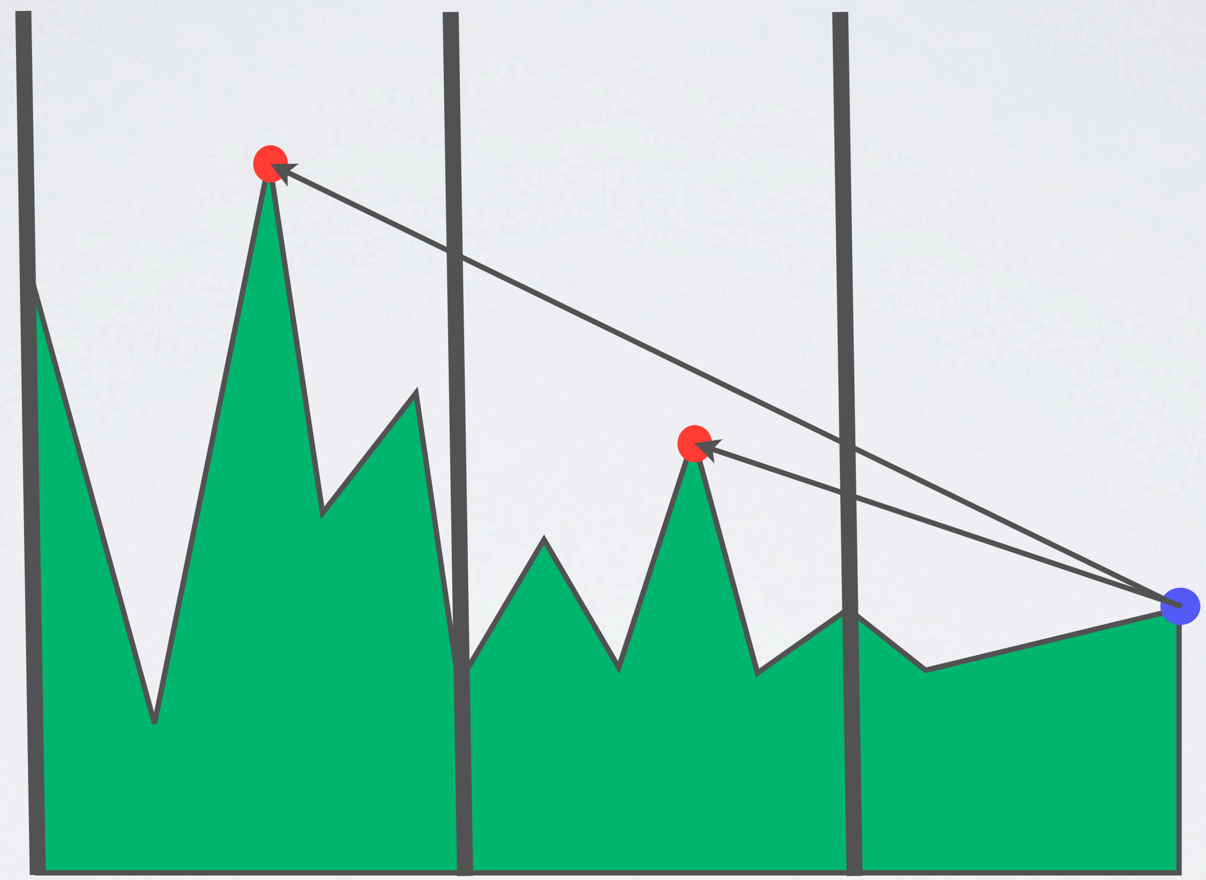
Previous 2 approaches:



Mip-mapping flattens the geometry

2 OUR METHOD

2 OUR METHOD

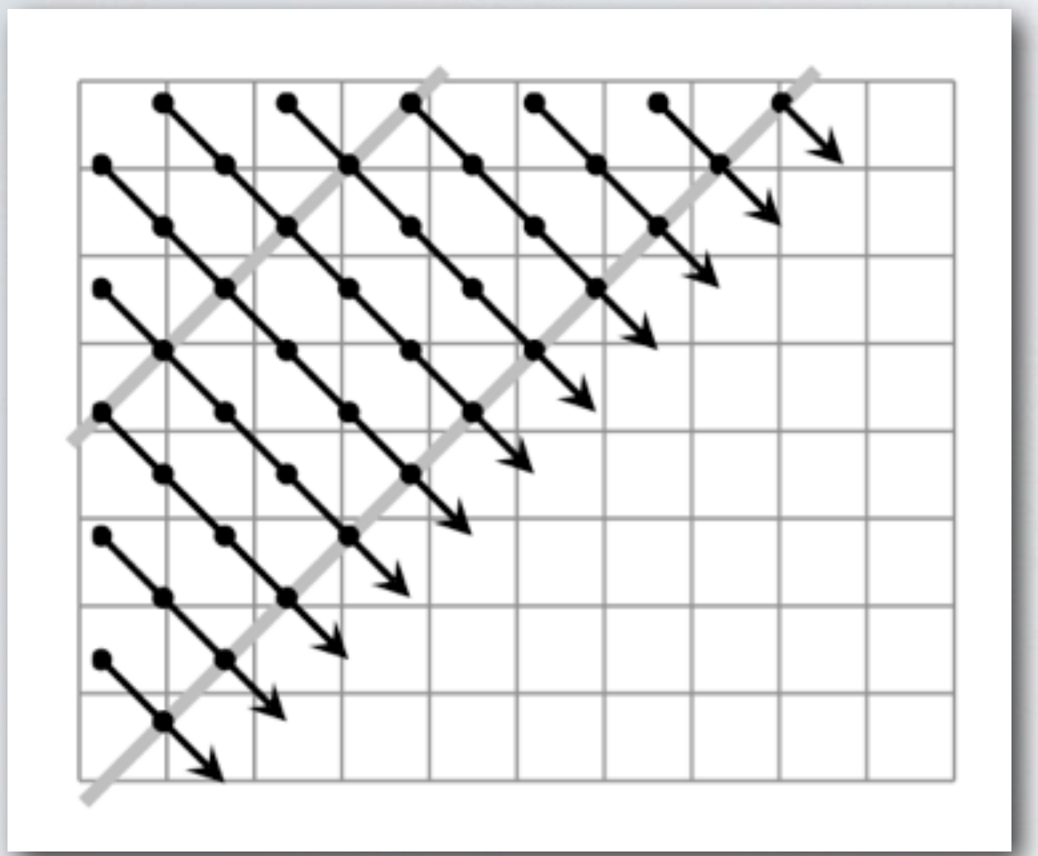


We capture points important for AO

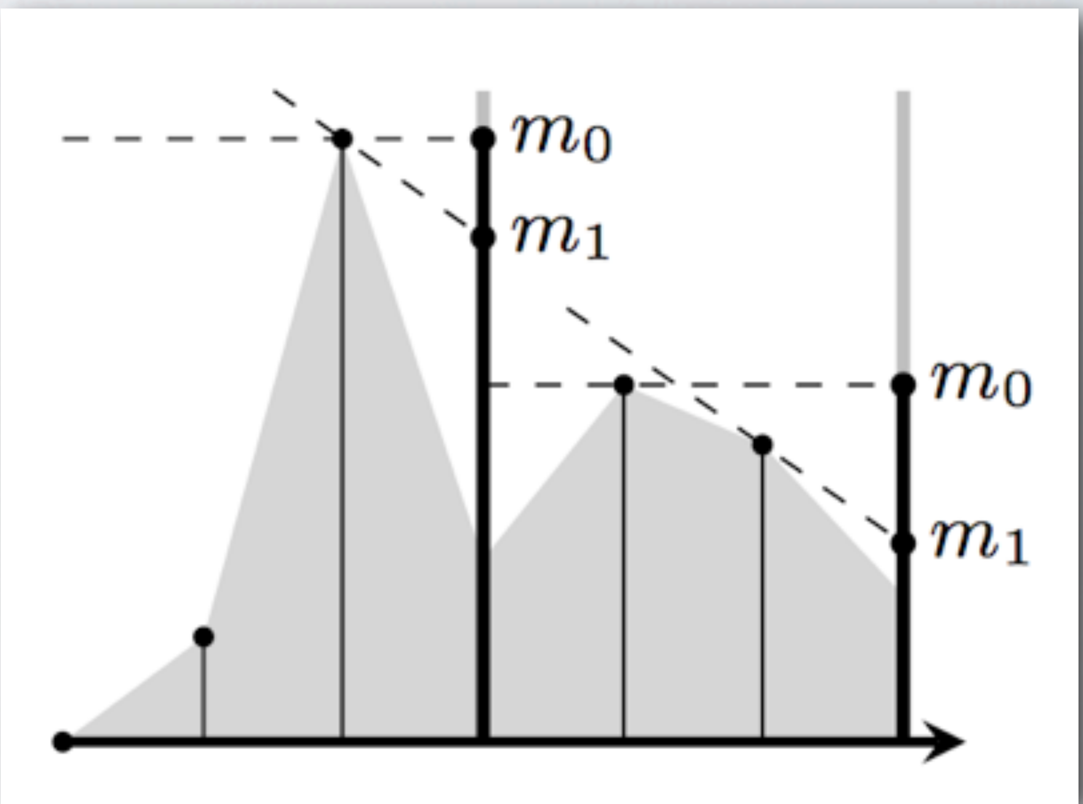
2 OUR METHOD

We capture points important for AO

Scans in multiple directions (one direction below)



We find highest "projections" (m_0, m_1) every *n* steps, and write them out

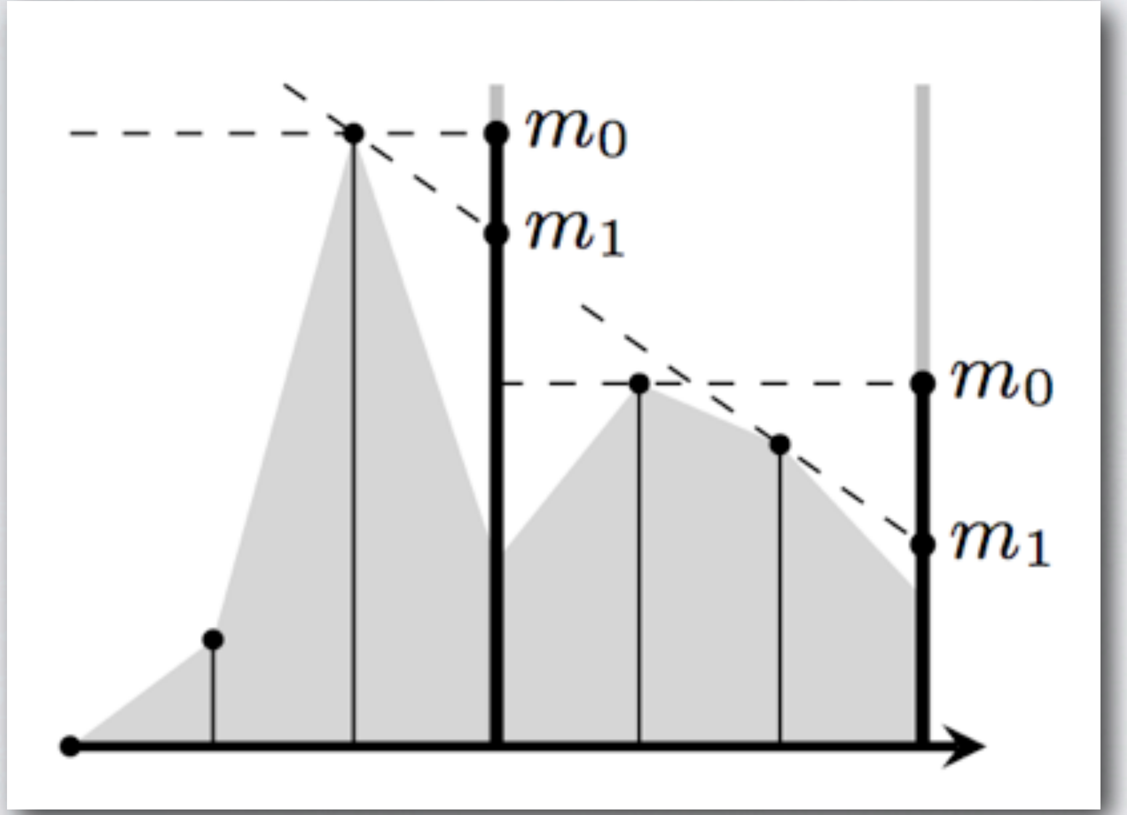


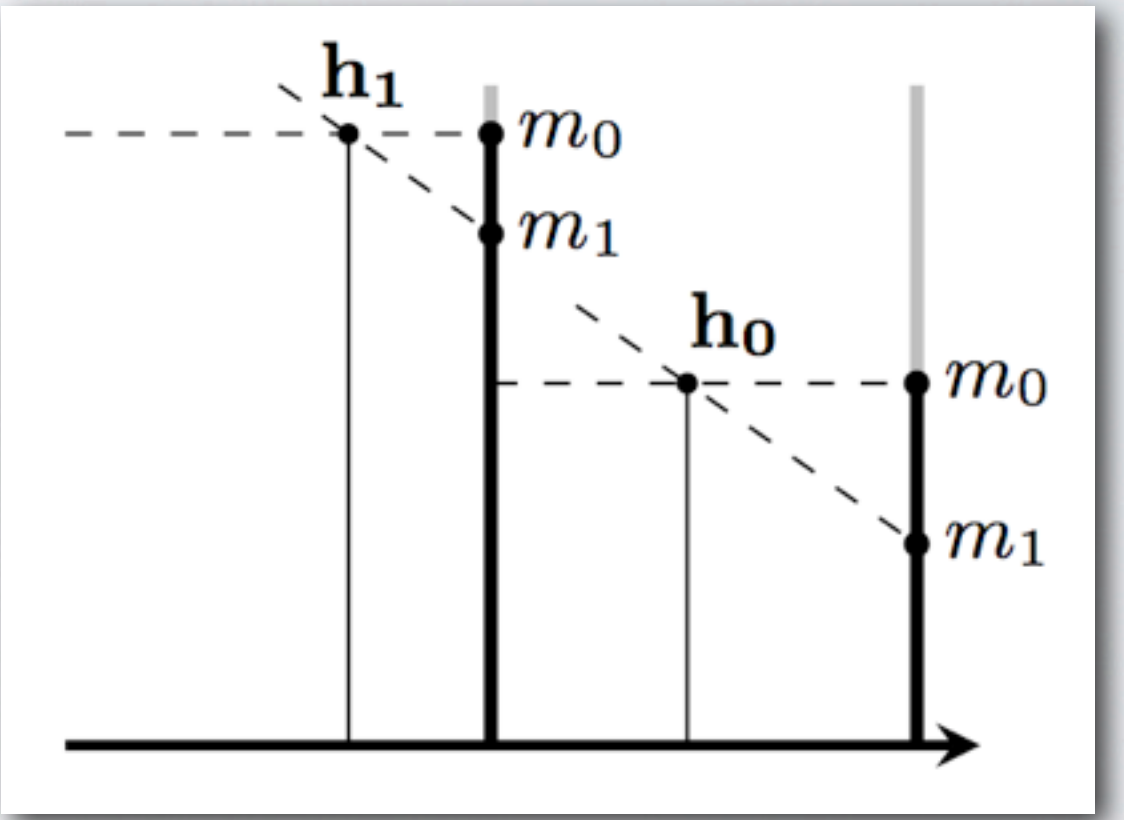
Time complexity of this phase is small w.r.t. to full SSAO

2 OUR METHOD

Final sample points, h_i

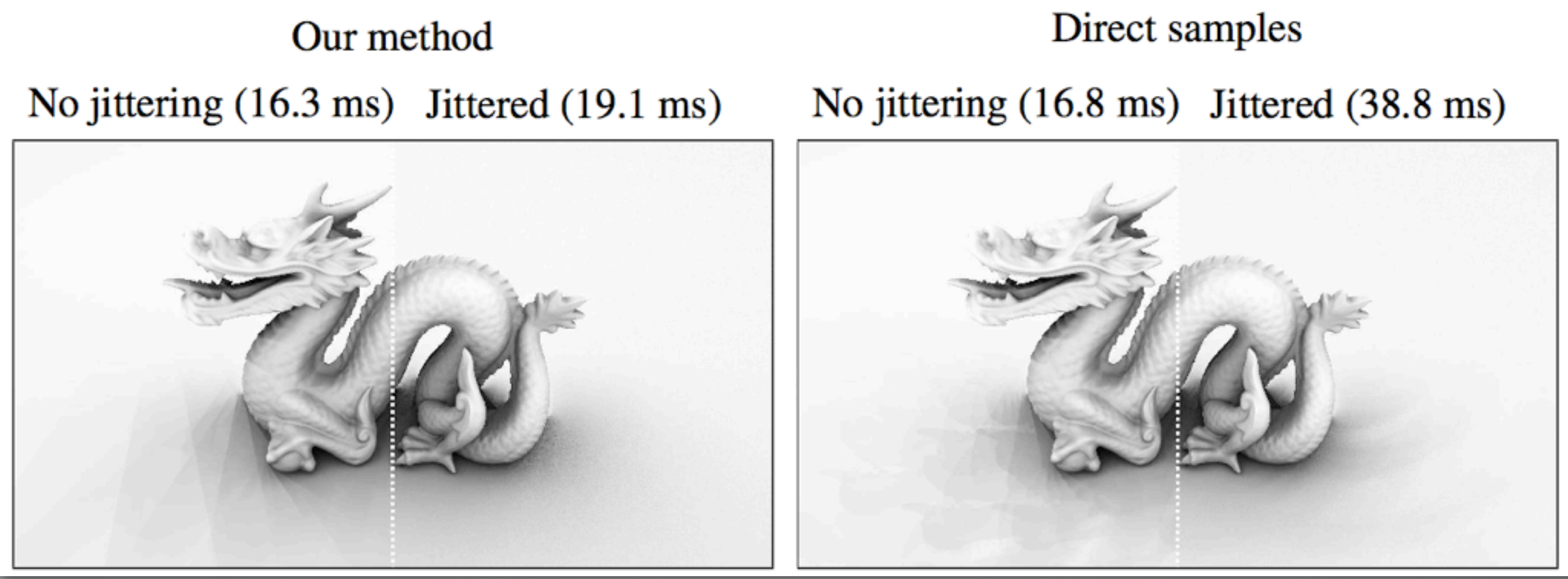
Out of these values, we construct final sample points h_i at the intersections





2 OUR METHOD

Results



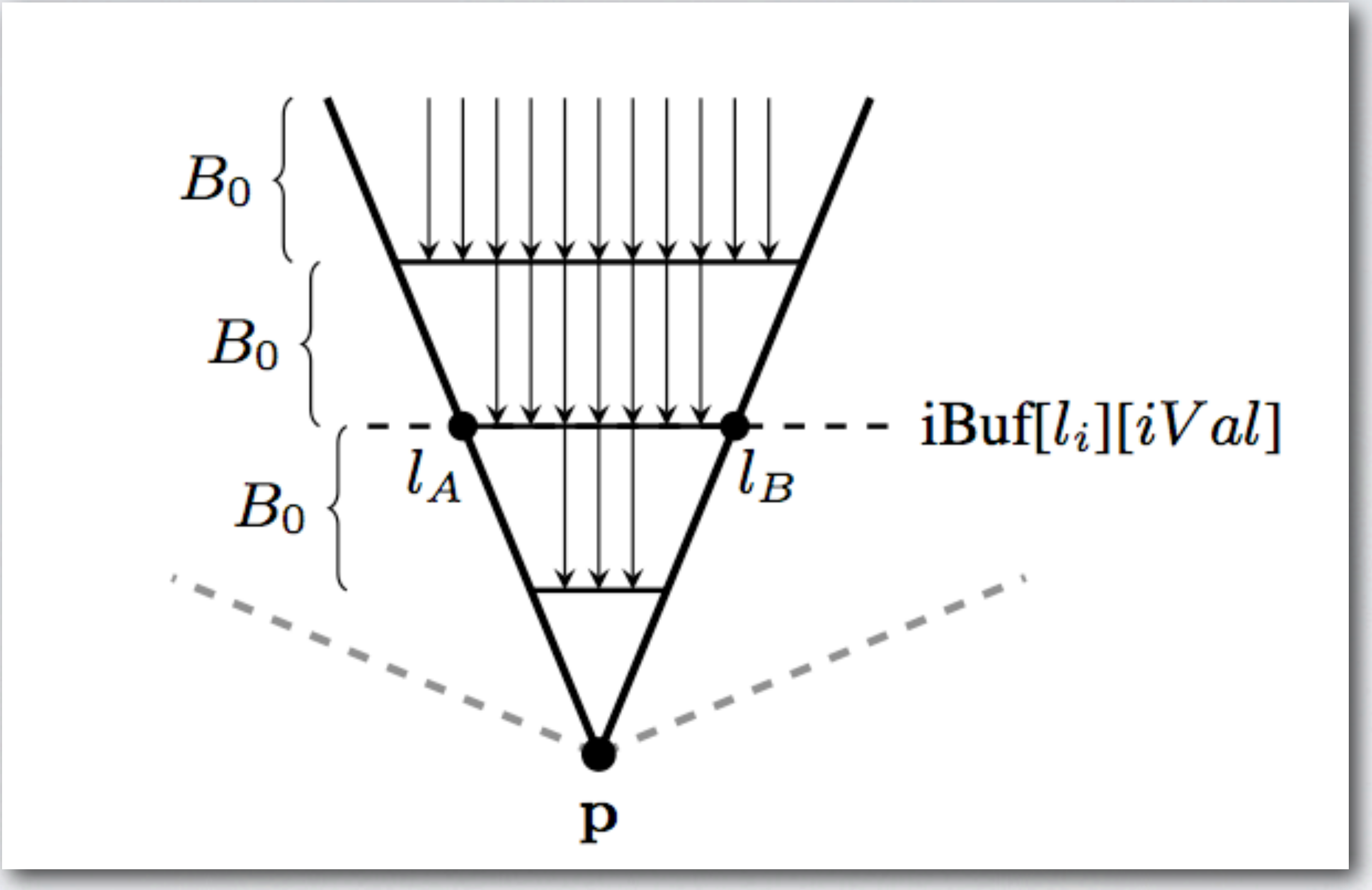
Along each azimuthal sampling direction, results are good

But banding persists between directions

2 OUR METHOD

Averaging the sectors

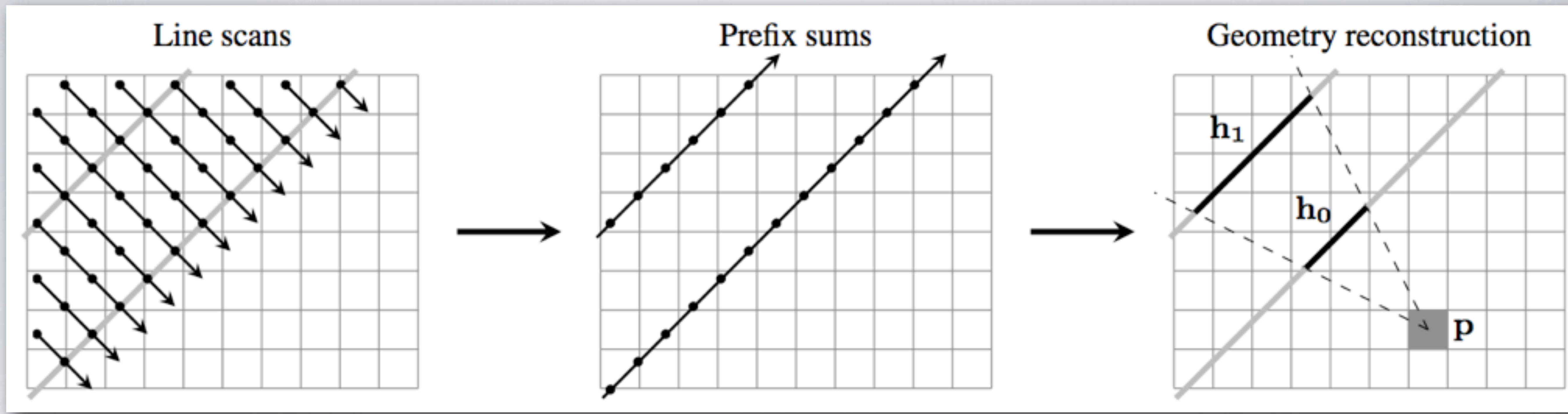
Instead of sampling along straight lines, we can average m_0 and m_1 across the azimuthal sector width



For fast averaging, we turn arrays of m_0 and m_1 into prefix sums

2 OUR METHOD

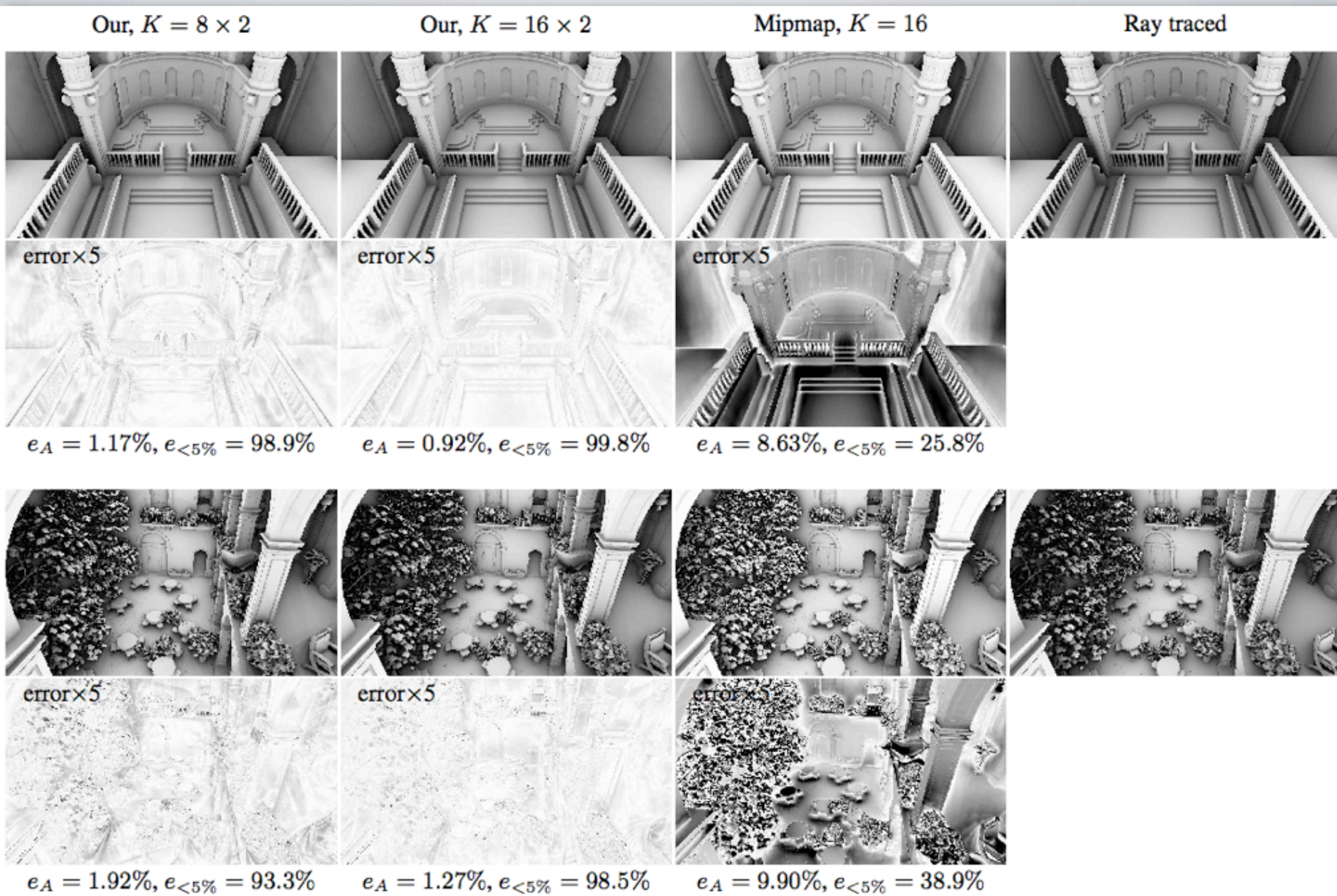
Overview of the 3 phases



This way we get scene points h_i that represent the entire sector, and therefore the full depth buffer

3 RESULTS

3 RESULTS



Our, $K = 8 \times 2$
Our, $K = 16 \times 2$
Mipmap, $K = 16$
Ray traced
error×5
error×5
error×5
$e_A = 1.17\%, e_{<5\%} = 98.9\%$
$e_A = 0.92\%, e_{<5\%} = 99.8\%$
$e_A = 8.63\%, e_{<5\%} = 25.8\%$
error×5
error×5
$e_A = 1.92\%, e_{<5\%} = 93.3\%$
$e_A = 1.27\%, e_{<5\%} = 98.5\%$
$e_A = 9.90\%, e_{<5\%} = 38.9\%$

3 RESULTS

Table 1: *Total render times of the far-field AO component.*

Method	Radeon 7970	GTX 580
$1280(+256) \times 720(+144)$, $B_0 = 10$:		
Our, $K = 8 \times 2$	7.26 ms	12.0 ms
Our, $K = 16 \times 2$	13.3 ms	23.6 ms
Mipmap, $K = 16$	19.2 ms	17.7 ms
$1920(+384) \times 1080(+216)$, $B_0 = 10$:		
Our, $K = 8 \times 2$	16.7 ms	29.4 ms
Our, $K = 16 \times 2$	31.6 ms	58.1 ms
Mipmap, $K = 16$	31.5 ms	37.9 ms

Roughly as fast as mipmap samples, but higher quality results and converges to ground truth faster

Table 2: *Render time breakdown of our method per kernel.*

Phase	Radeon 7970	GTX 580
$1280(+256) \times 720(+144)$, $K = 8 \times 2$, $B_0 = 10$:		
Scan	0.537 ms	0.489 ms
Prefix sum	0.945 ms	0.617 ms
Obscurance	5.77 ms	10.9 ms

3 RESULTS

1280(+256)x720(+144), 4.6ms/frame, Radeon 7970

Screen-Space Far-Field Ambient Obscurance

4 LEFT OUT FROM THE PRESENTATION

4 LEFT OUT FROM THE PRESENTATION

- The obscurance estimator
- Fixed thickness depth fields
- Multiple levels of detail for the projection intervals
- Interleaved sampling (quick preview next)
- Combining with a near field search (quick preview next)

4 LEFT OUT FROM THE PRESENTATION

Preview: *Interleaved sampling*

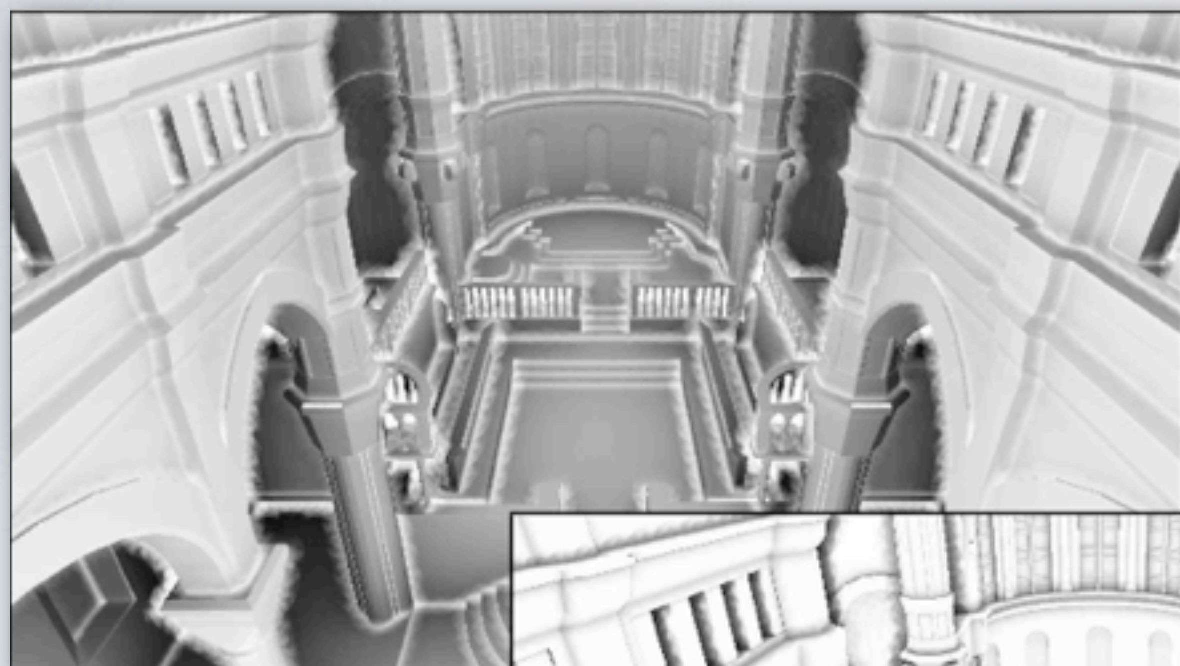
4.6 ms/frame, interleaved

ray traced

- Evaluate a subset of sectors per pixel, gather in a selective box filter pass
- Used in the video as well

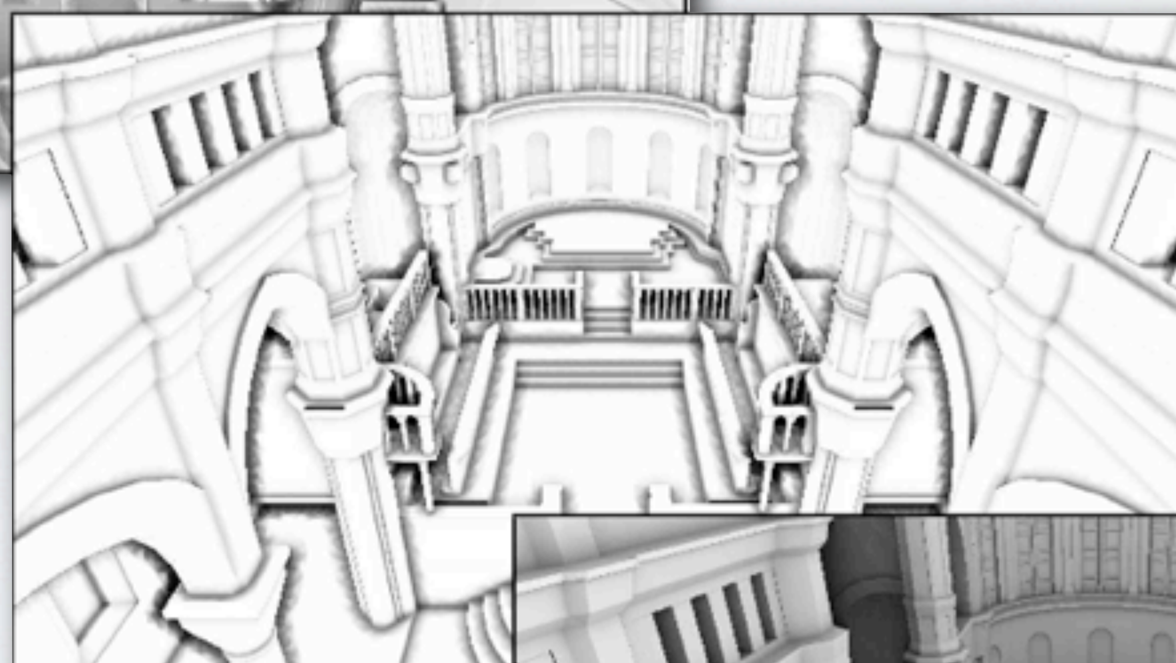
4 LEFT OUT FROM THE PRESENTATION

Preview: *Combining with a near field search*



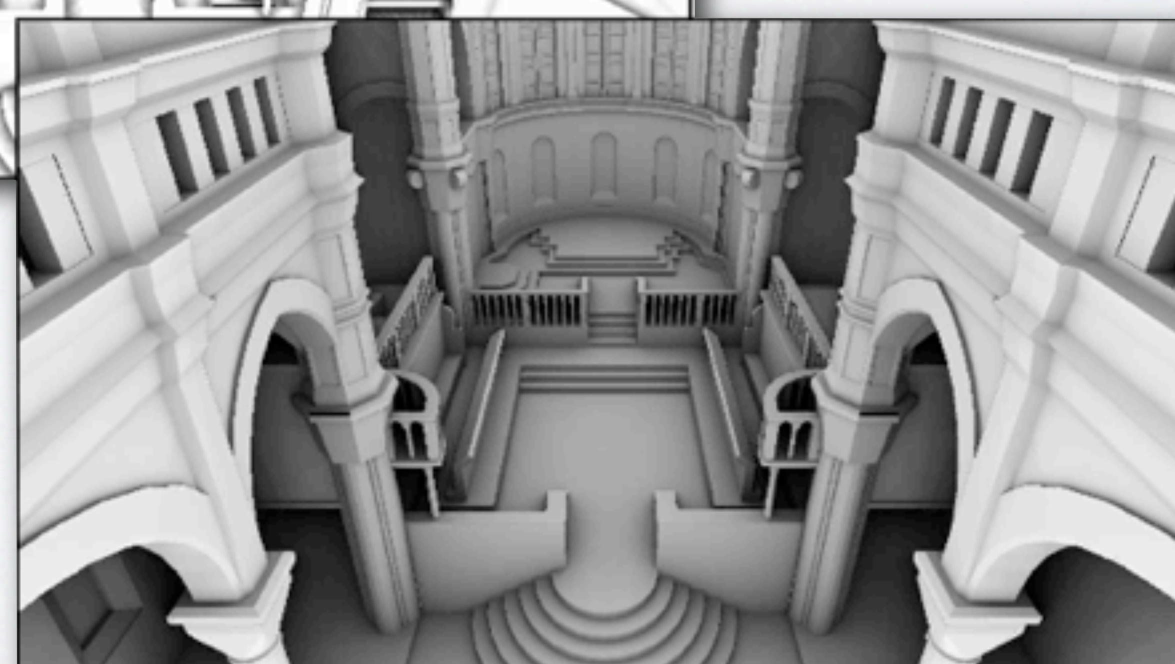
Far field ≥15px (our method)

+



Near field <15px

=



Final result

5 QUESTIONS

Or comments...

?

Sources available online, under the BSD license

http://wili.cc/research/ffao/