Adaptive Image Space Shading for Motion and Defocus Blur

Karthik Vaidyanathan[1] / Robert Toth[1] / Marco Salvi[1] / Solomon Boulos[2] / Aaron Lefohn[1]

[1] Intel Corporation / [2] Stanford University

What is this talk about?

Overview

- Part I: Introduction to 5D
- Part II: Frequency analysis
- Part III: Results

Overview

- Part I: Introduction to 5D
- Part II: Frequency analysis
- Part III: Results

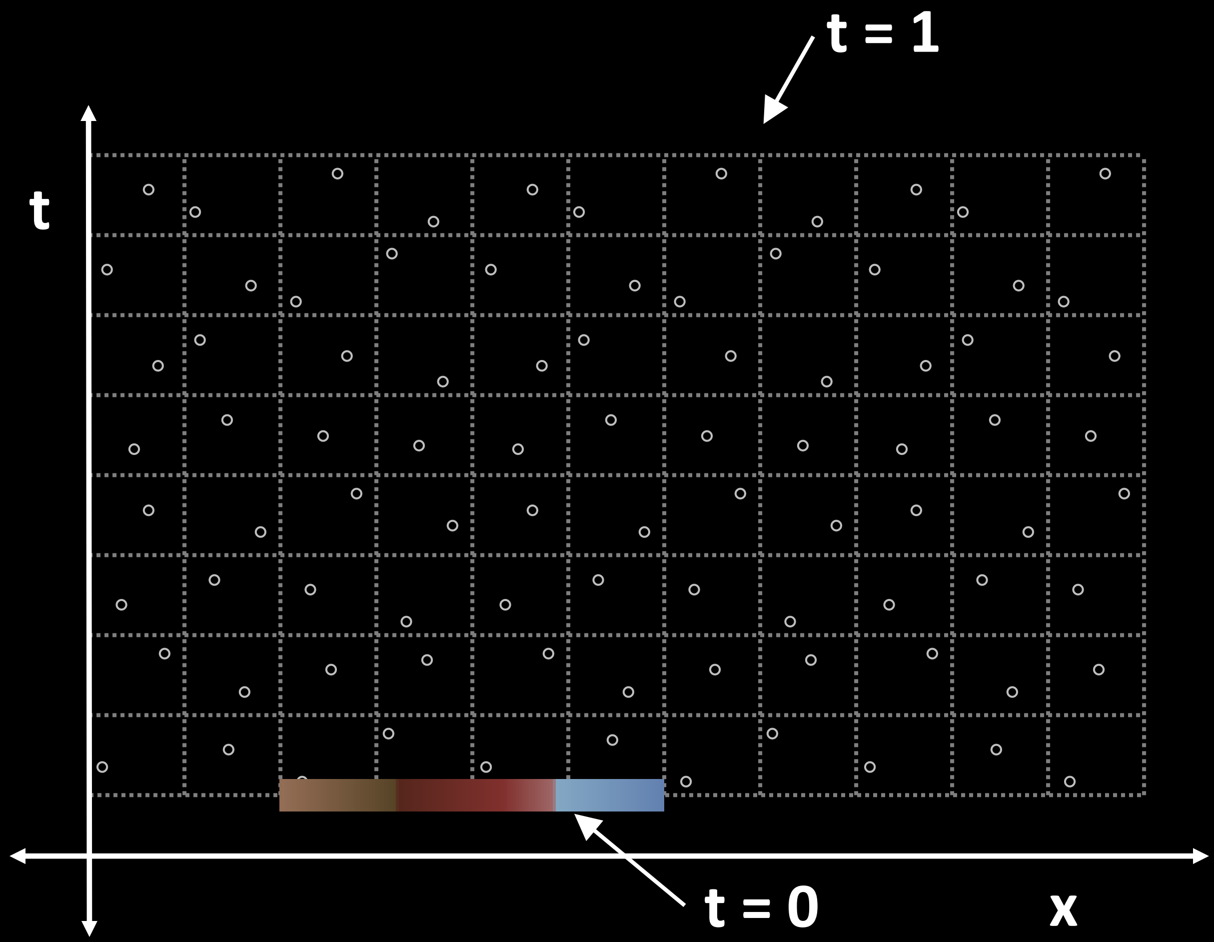
5D stochastic rasterization

- Test if a primitive is covered at:
  - Different points on the screen (x, y)
  - Different points on the aperture (u, v)
  - Different time instants (t)

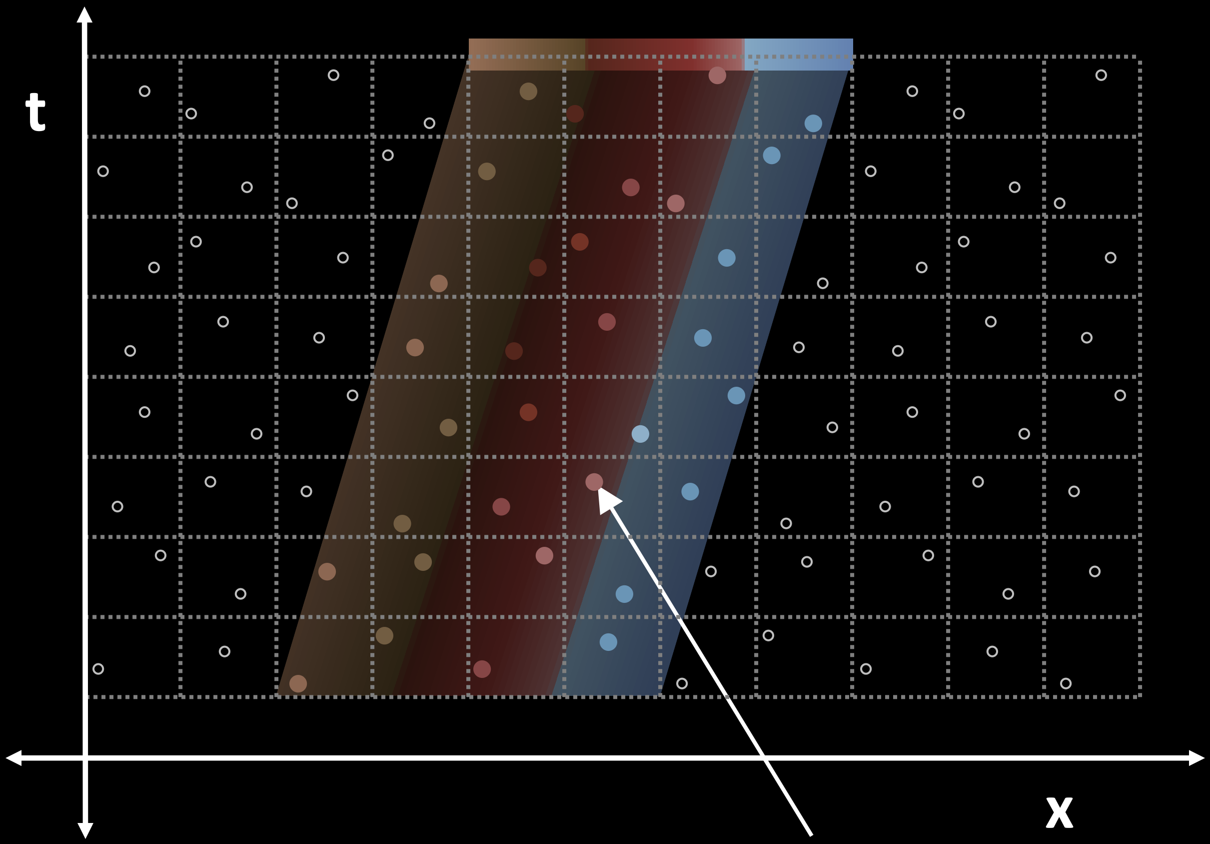
5D stochastic rasterization

- Lots of samples per pixel are needed to eliminate noise
  - We do not want to shade all of them individually (super sampling)
  - We base our work on decoupled sampling [Ragan-Kelley et al. 2011]

Example: motion blur

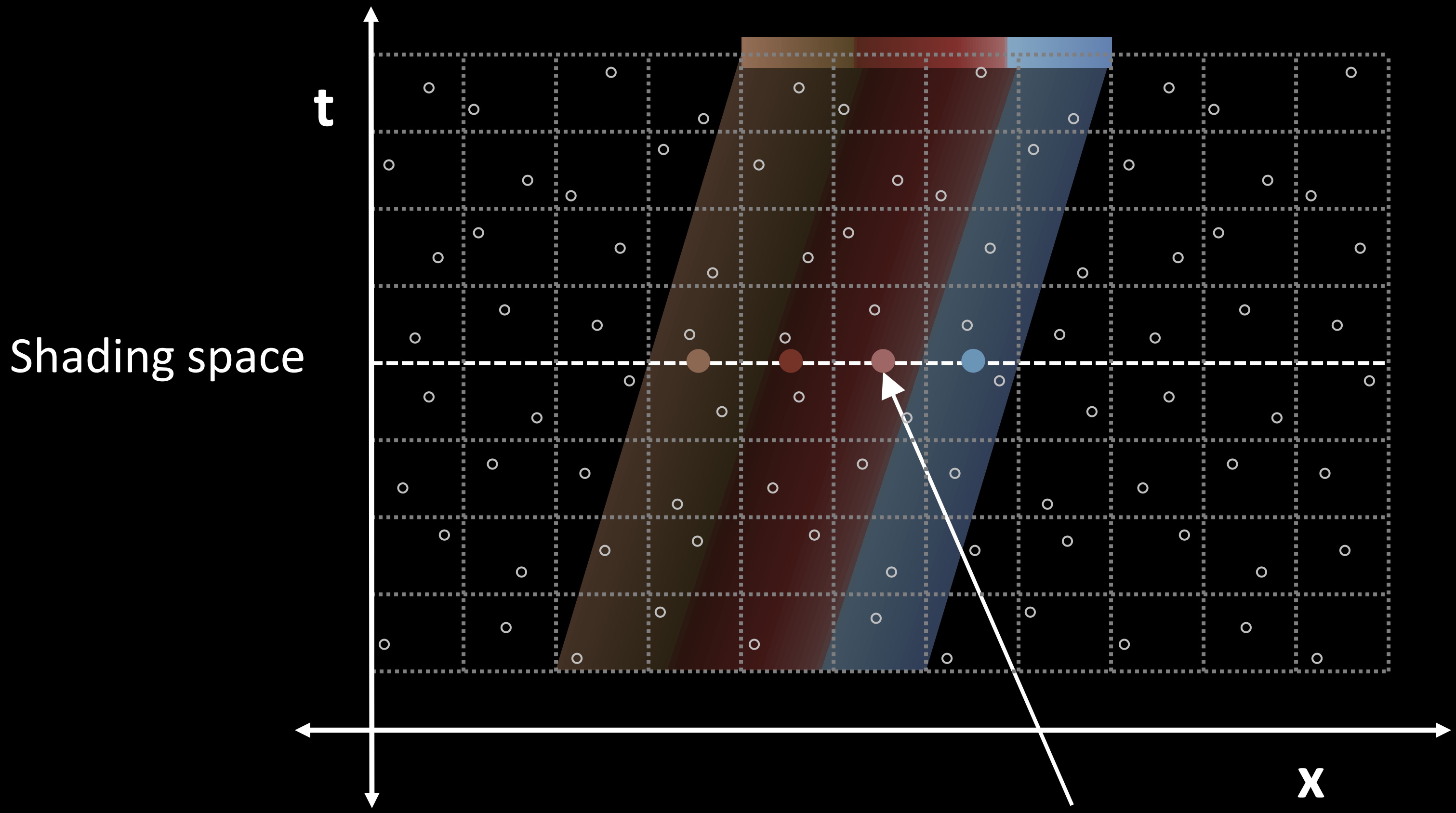


Super sampled shading



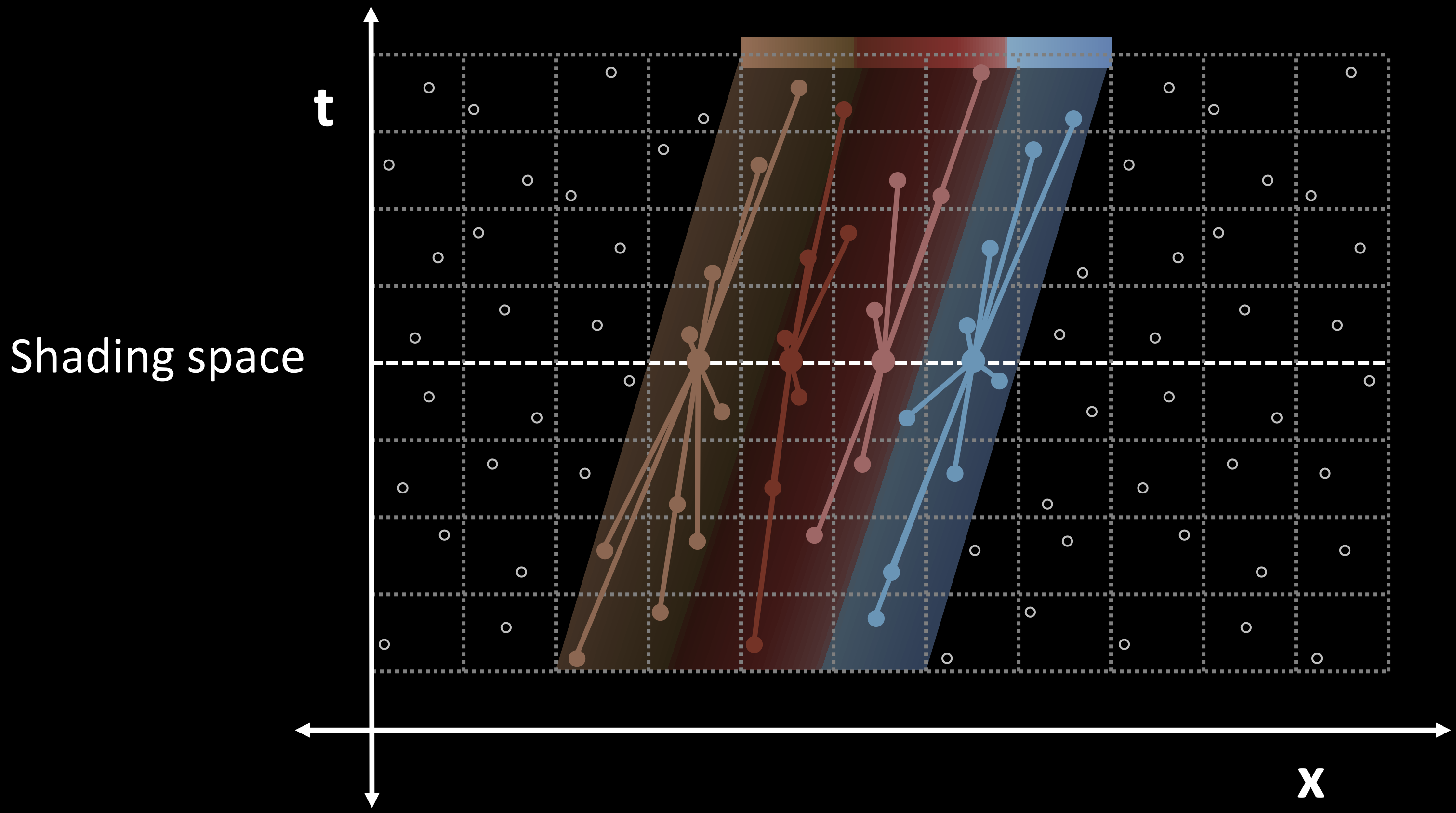
Shade each covered sample
(32 samples shaded)

Decoupled sampling



Shade once per pixel at fixed **t**
(4 samples shaded)

Decoupled sampling



Reuse shaded colors
at all samples

Decoupled sampling

- Define shading space
  – Fixed point on the aperture and in time
  – Shade once per “pixel”

- Re-project samples to shading space
  – Map to a “pixel” and reuse shaded color

- Implemented using a memoization cache
  – [Ragan-Kelley et al. 2011]

Decoupled sampling

- Q: Can we shade even less than once/pixel?
- A: Yes!

Overview

- Part I: Introduction to 5D
- Part II: Frequency analysis
- Part III: Results

Frequency analysis

- Objective: derive bounds of “useful” frequencies
  – Frequencies contributing to image
- Contains lots of approximations!
  – Aim is real-time good-enough quality
- Will not go into detailed equations

Integral of a pixel

- Surface contribution to pixel color is an integral over P:=[x,y,u,v,t]:
  - Color += ∫ L A S R V dP
    - L: radiance
    - A: aperture
    - S: shutter
    - R: pixel filter
    - V: visibility

Integral of a pixel

- Simplify: separate visibility
  - Color += $\sum (\int L\ A\ S\ R\ dP_i\ V(P_i))$

Approximation #1: visibility does not alter surface filtering

L: radiance
A: aperture
S: shutter
R: pixel filter
V: visibility

Integral of a pixel

- Approximate with decoupled sampling

$$L(x,y,u,v,t) \approx L_0\big(x_0(x,y,u,v,t),\ y_0(x,y,u,v,t)\big)$$

Approximation #2:
decoupled sampling at a
single aperture location and
a single instance in time

Integral of a pixel

- Approximate the decoupled shading space

$$x_0 \approx x - t\,\mu_x - u\,\phi$$
$$y_0 \approx y - t\,\mu_y - v\,\phi$$

Approximation #3:
primitive moves at constant shading-space velocity

Approximation #4:
locally constant defocus radius

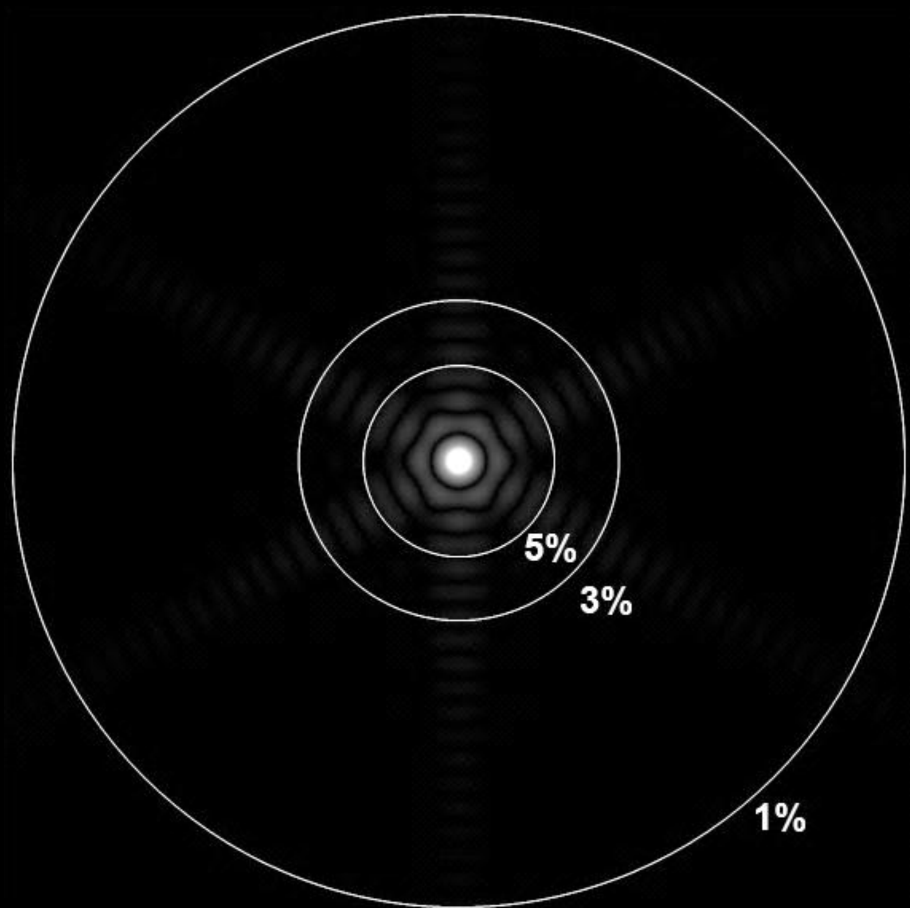
Integral of a pixel

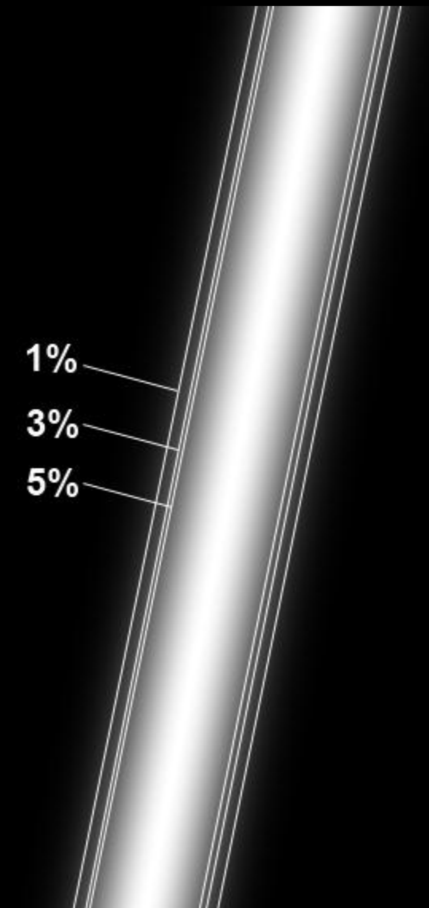
- What did we gain by doing that?
  - $O(x,y) = L_0 * A * S * R$

L: radiance
A: aperture
S: shutter
R: pixel filter

- Transform into frequency domain:
  - $\hat{O} = \hat{L_0}\,\hat{A}\,\hat{S}\,\hat{R}$

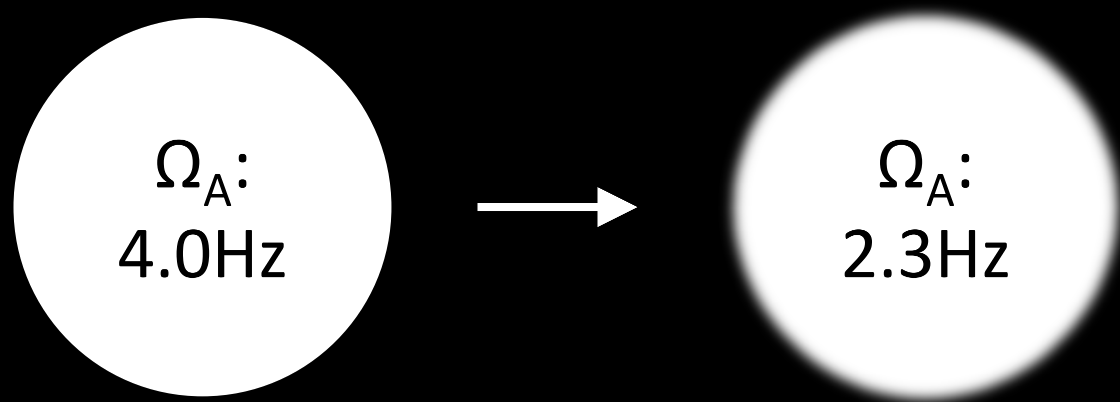
Example spectra





Observation

- Smoothing the aperture filter a little can narrow the frequency range a lot
  - Applies to the shutter as well



Band limited shading space

- Safe to low pass filter L_0 outside spectral support of A, S, R:
  - $\widehat{O} = \widehat{L_0}\,\widehat{A}\,\widehat{S}\,\widehat{R} = \widehat{L'}\,\widehat{A}\,\widehat{S}\,\widehat{R}$

L: radiance
A: aperture
S: shutter
R: pixel filter

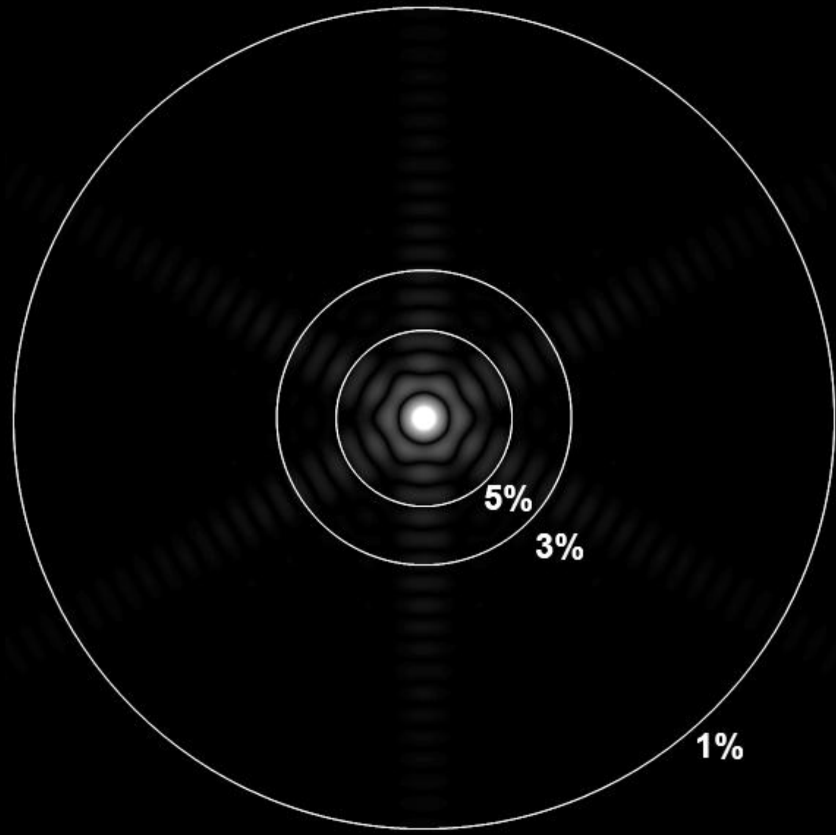
- How to actually band limit L’ is _not_ the focus of our work

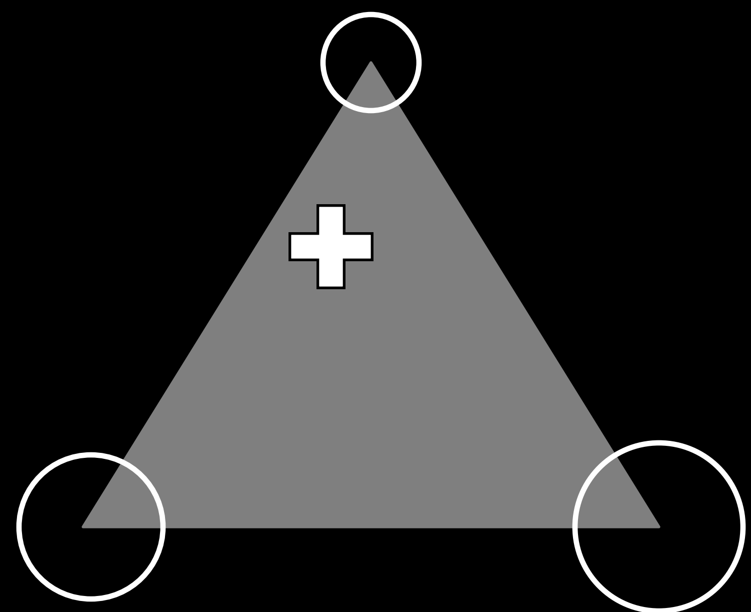
A primitive

- A primitive has:
  - Varying amount of defocus
  - Varying velocity
- How do we derive bounds for $\widehat{A}$, $\widehat{S}$?

A primitive

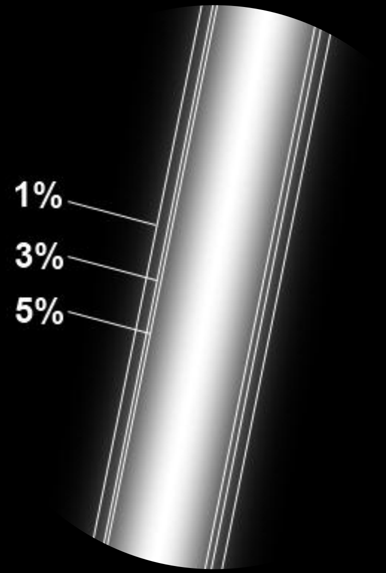
- Varying amount of defocus:
  – Use the smallest circle of confusion

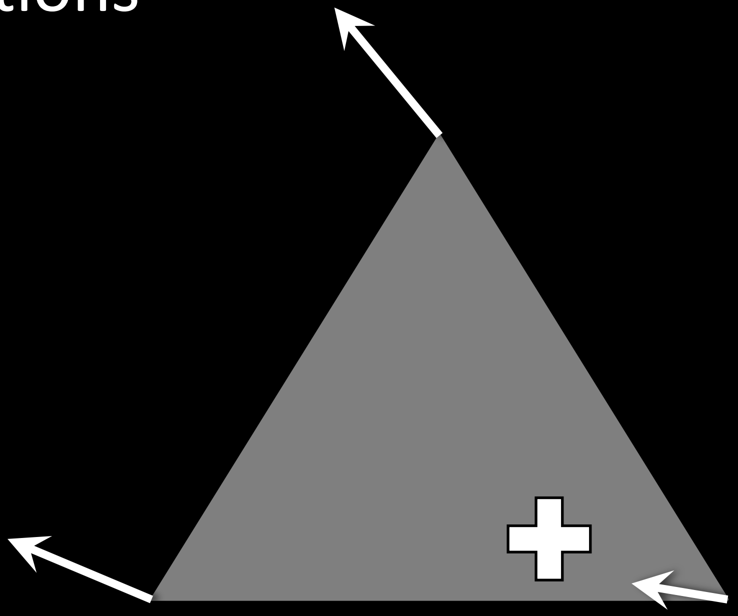




A primitive

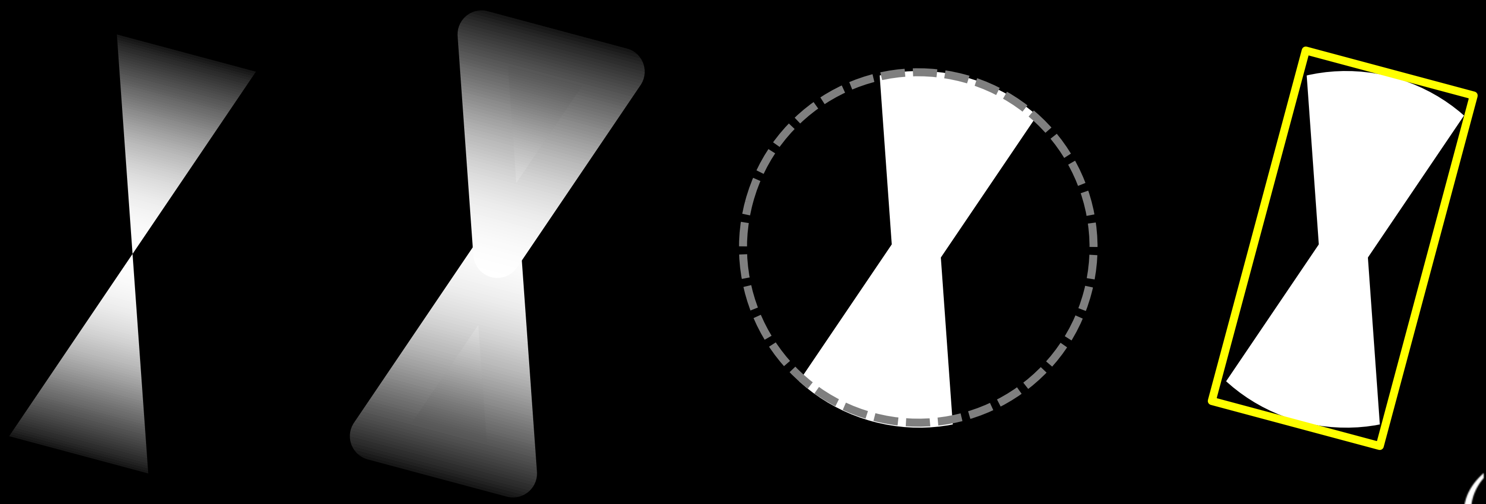
- Varying speed and direction:
  - Use the lowest speed
  - Enclose all motion directions





A primitive

- Putting it all together
  - Enclose all motion directions
  - Use the lowest speed
  - Use the smallest circle of confusion
  - Bound with motion-aligned bounding box

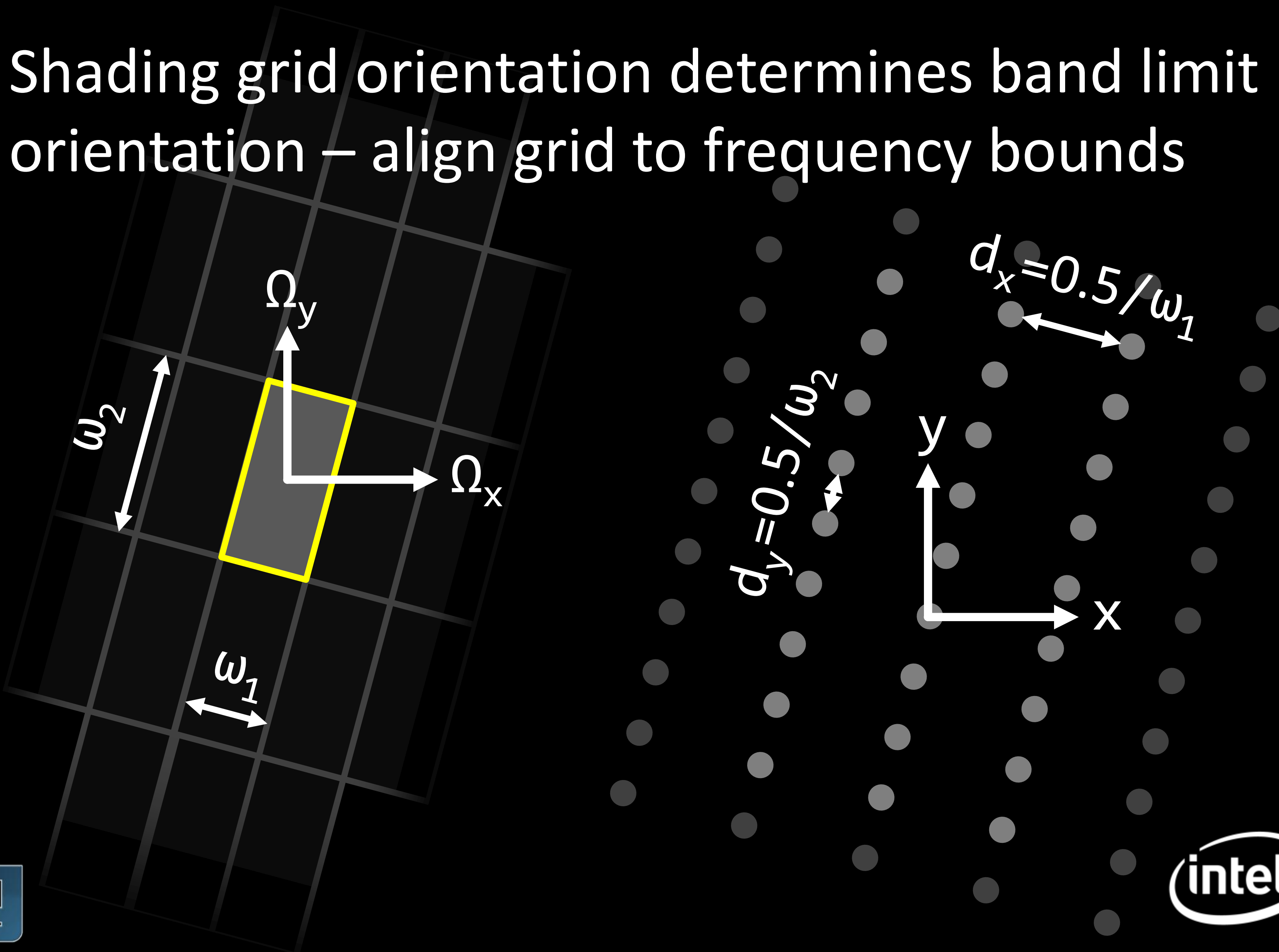


Sampling and reconstruction

- Shader knows how to filter itself
  - Sample spacing d → frequency limit $\omega = 0.5/d$
- We want a specific frequency limit
  - Reverse the logic, $d = 0.5/\omega$

The shading grid

- Shading grid orientation determines band limit orientation – align grid to frequency bounds



The entire pipeline

- Several steps to final image:
  - Our algorithm: determine shading grid
  - Shader: compute band limited surface color
  - Visibility engine: sample shading space

Overview

- Part I: Introduction to 5D
- Part II: Frequency analysis
- Part III: Results

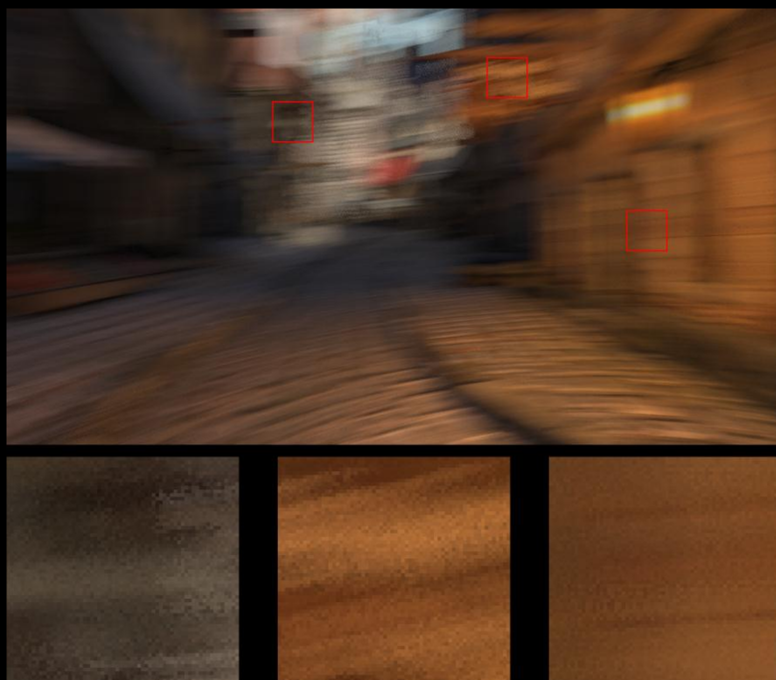
Results

- All shading rates with previous work:
  - Memoization cache capacity = 1k quads
- All shading rates with our algorithm:
  - Memoization cache capacity = 64 quads

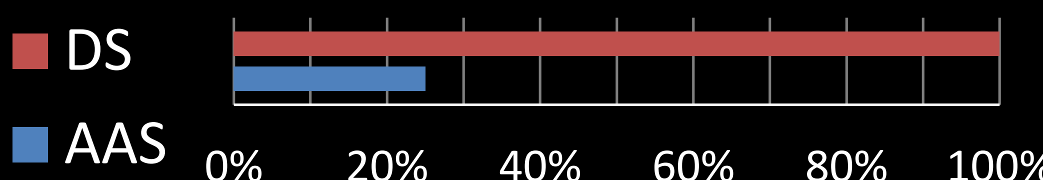
Results

- Citadel scene

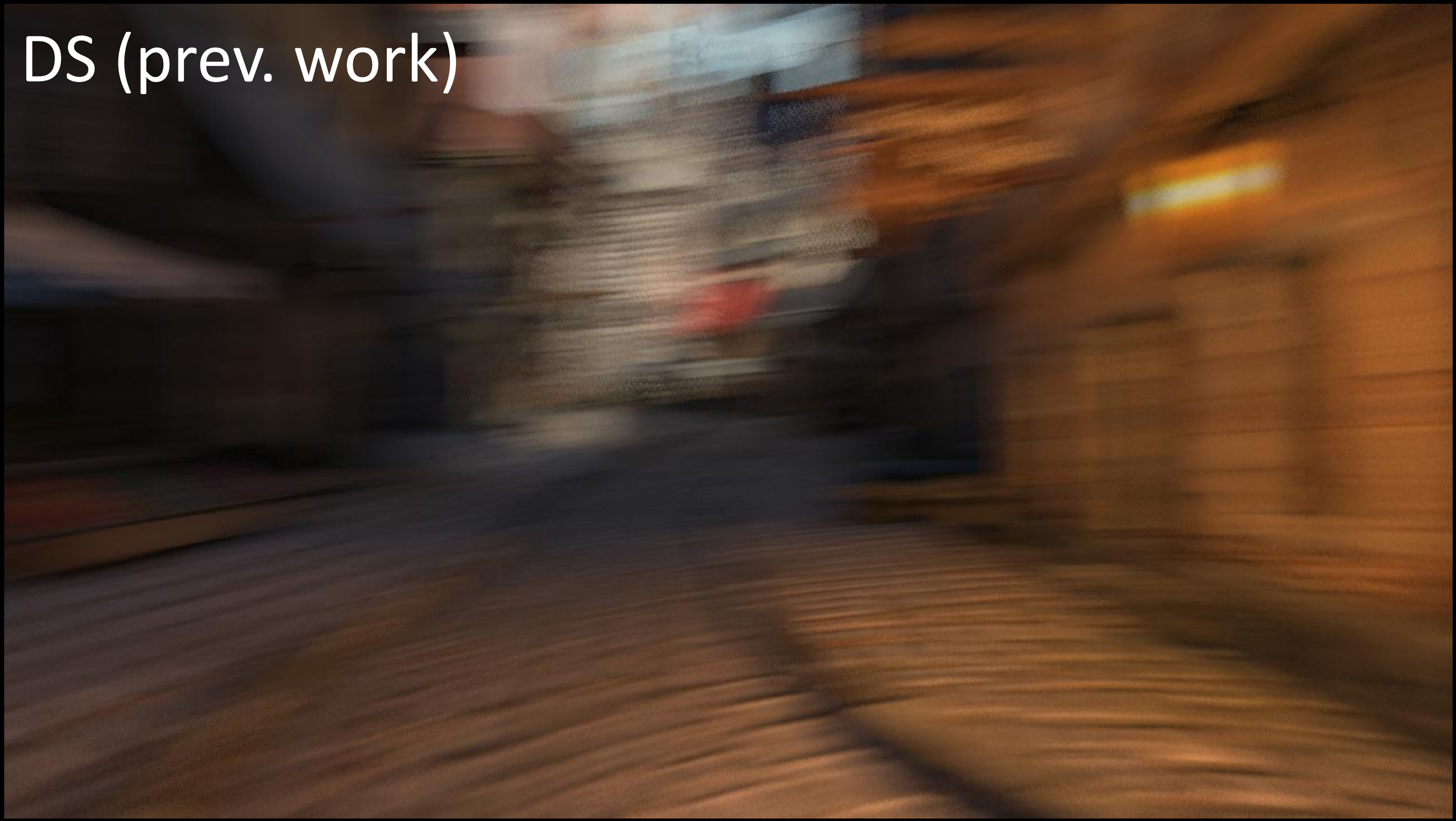
DS (prev. work)



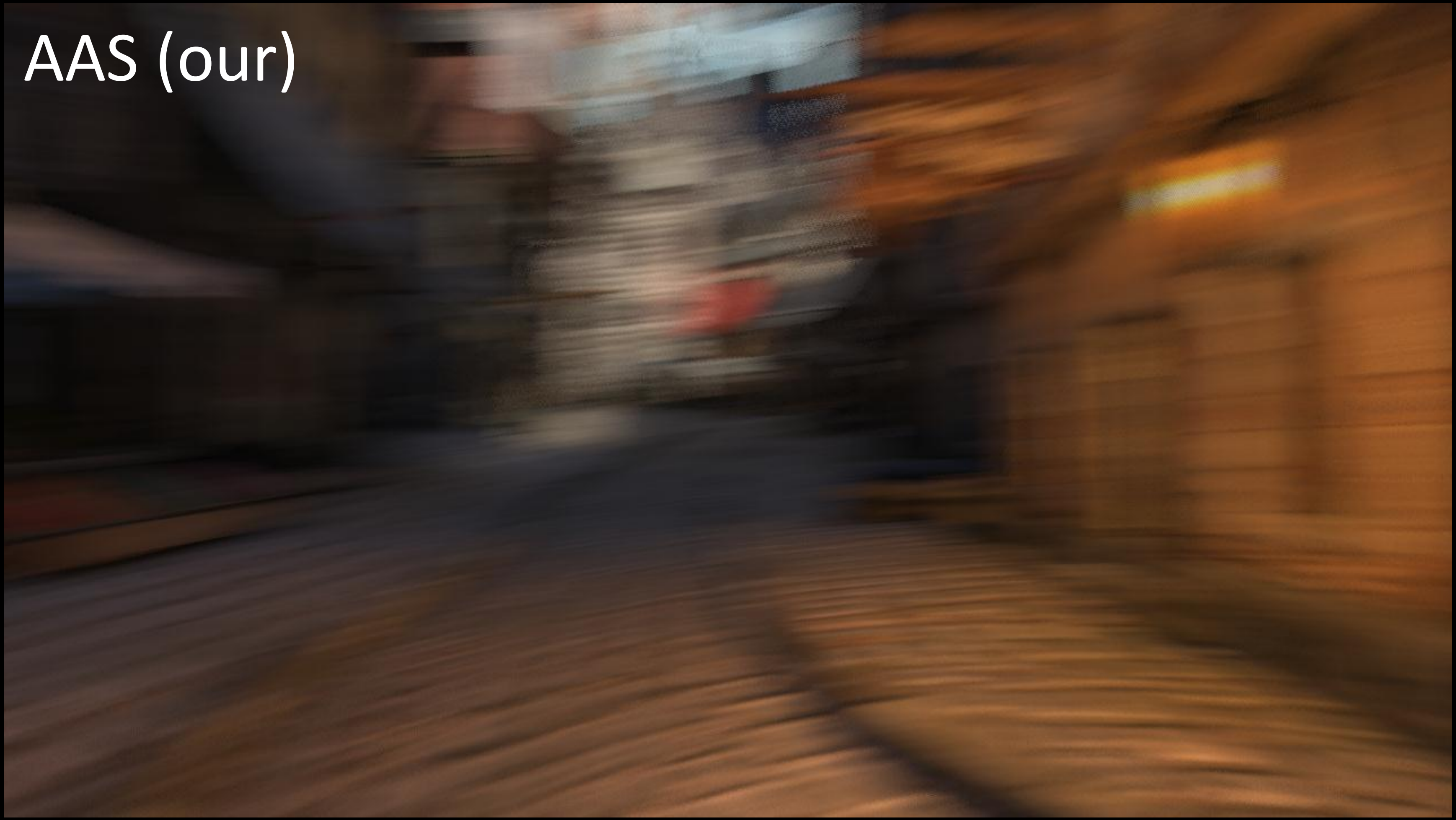
AAS (our)



Results



Results

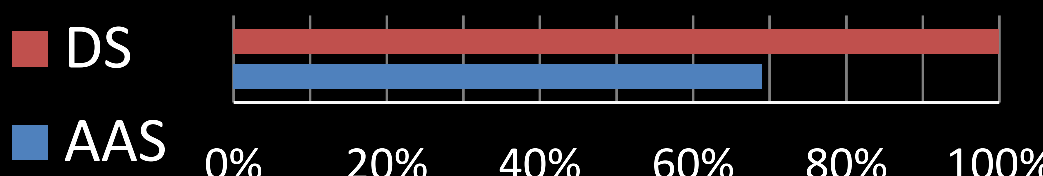


Results

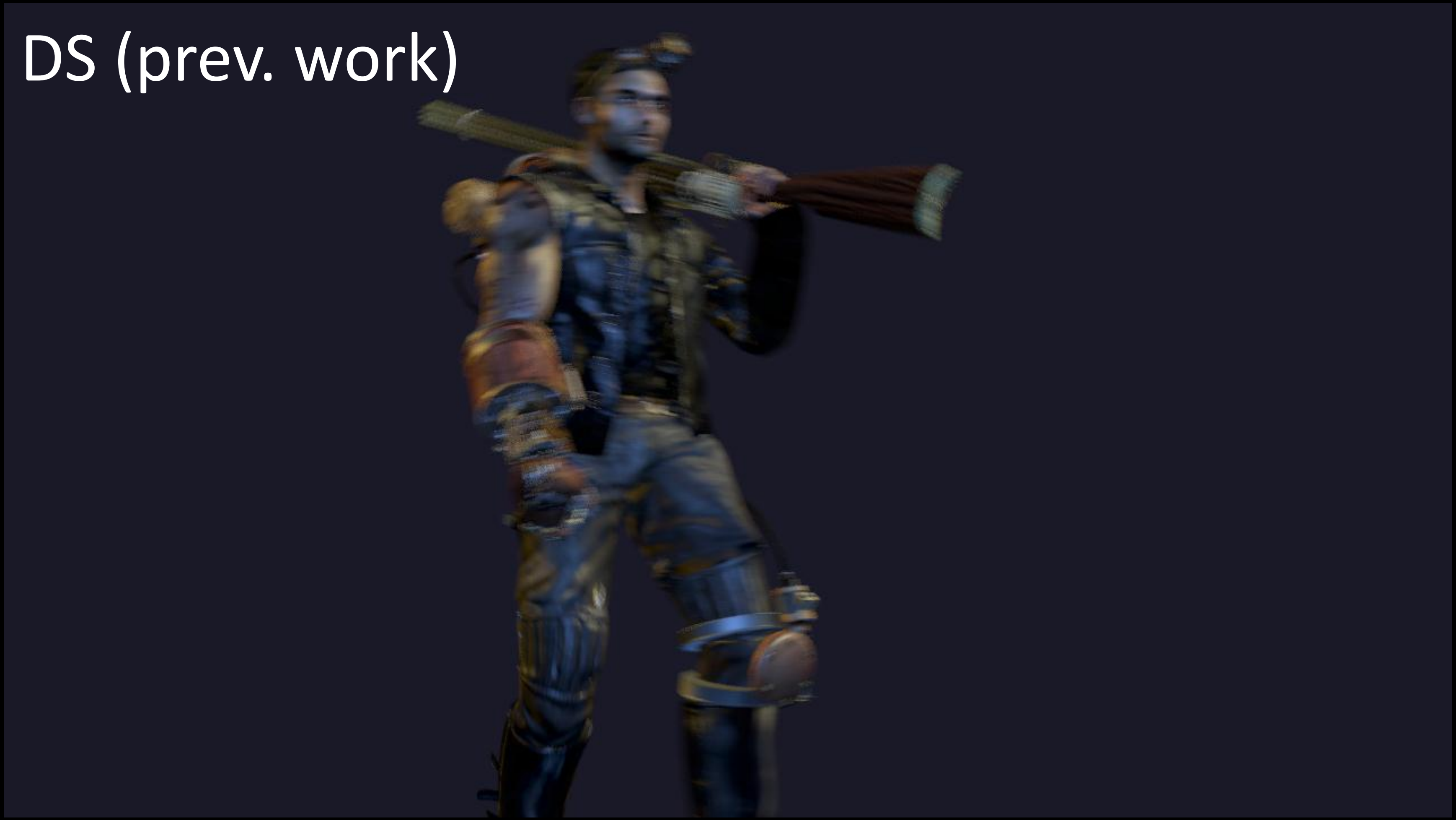
- SubD11 scene

DS (prev. work)

AAS (our)



Results



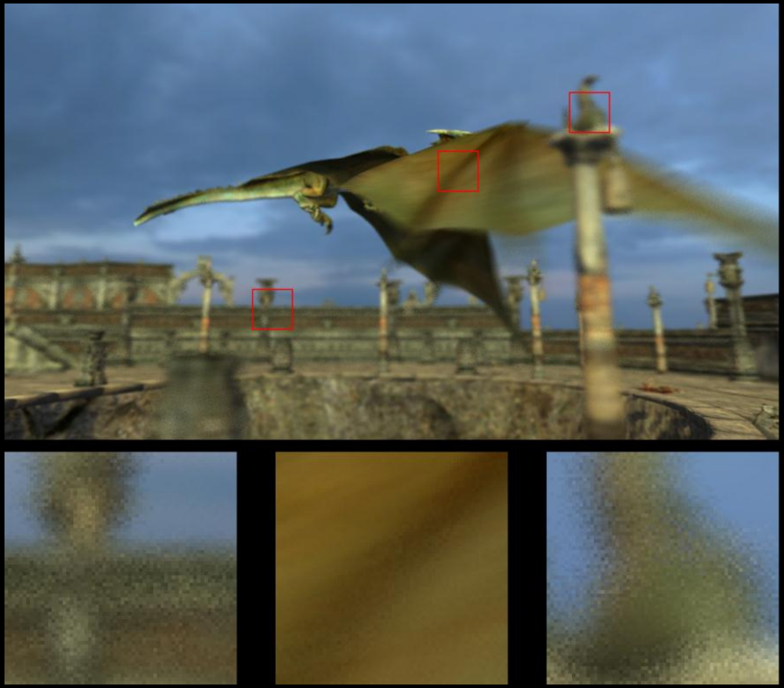
Results

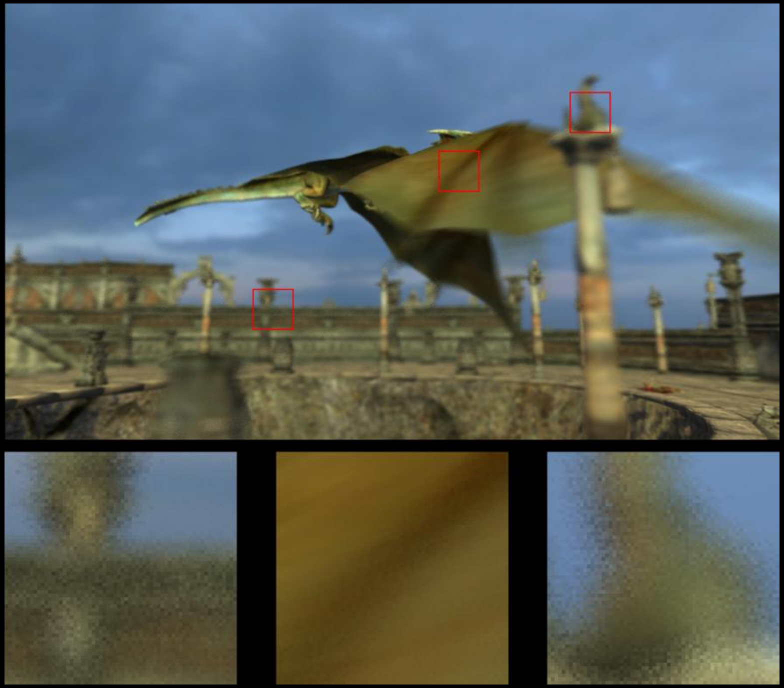
Results

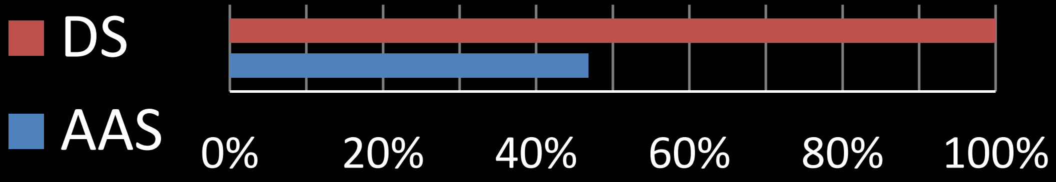
- Arena scene

DS (prev. work)

AAS (our)



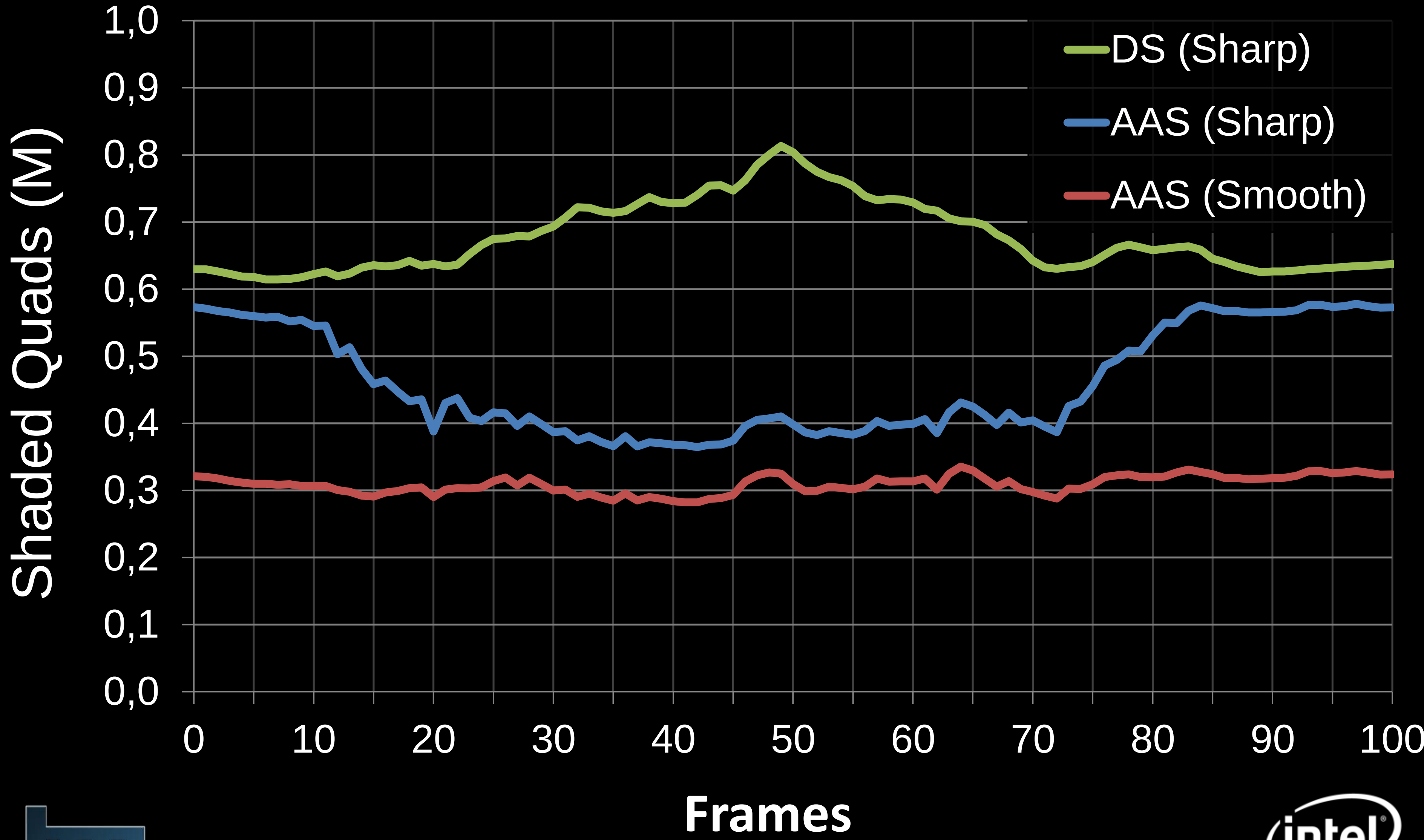




Results

Results

Results



DS (Sharp)
AAS (Sharp)
AAS (Smooth)
Shaded Quads (M)
1,0
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0,0
0
10
20
30
40
50
60
70
80
90
100
Frames

Results

- Cost:
  - Approximately 100 ops per triangle
  - 500k triangles @ 60Hz = 3 GFLOPS (0.1% of high-end GPU)

Conclusions

- We have developed a low-cost technique for determining a blur-aware shading grid for decoupled sampling.

- Benefits:
  - Reduced amount of shading
  - Smaller decoupling cache size
  - Less noise due to low-pass filter
  - No major changes to the decoupled sampling pipeline

Questions?

- Thanks to:
  - Aaron Coday, Charles Lingle, Tom Piazza, for supporting this research
  - Intel Advanced Rendering Technology team, for valuable feedback