SIGGRAPH2011 VANCOUVER

Filtering Approaches for Real-Time Anti-Aliasing

http://www.iryoku.com/aacourse/

SRAA: Subpixel Reconstruction Anti-Aliasing

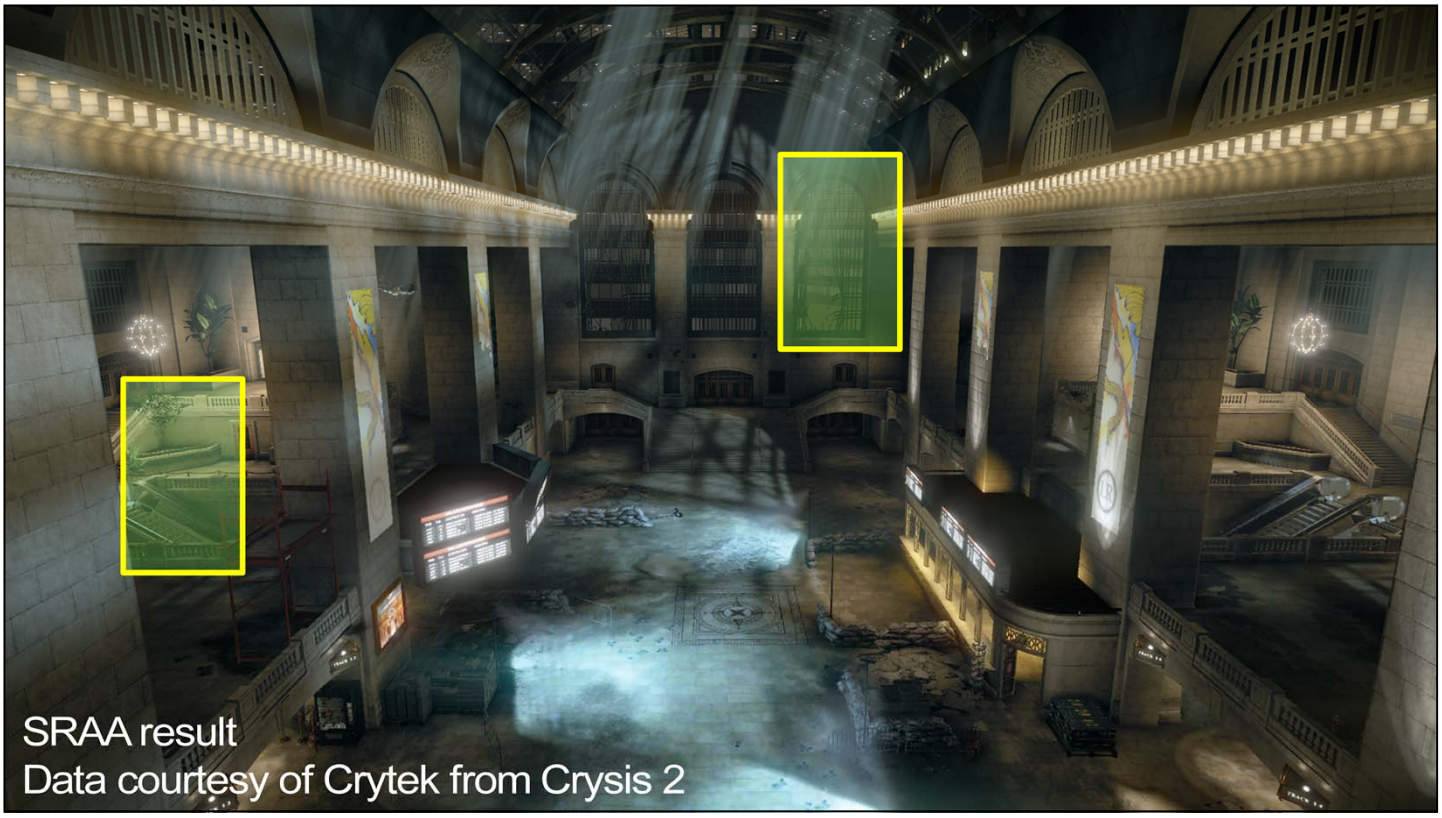
Morgan McGuire[1] Matthäus Chajdas[2] Sergey Puchin Miguel Sainz David Luebke

NVIDIA

SIGGRAPH2011

Also affiliated with: (1) Williams College (2) Technische Universität München

If you learn one thing in this course, it should not be about SRAA. It should be …



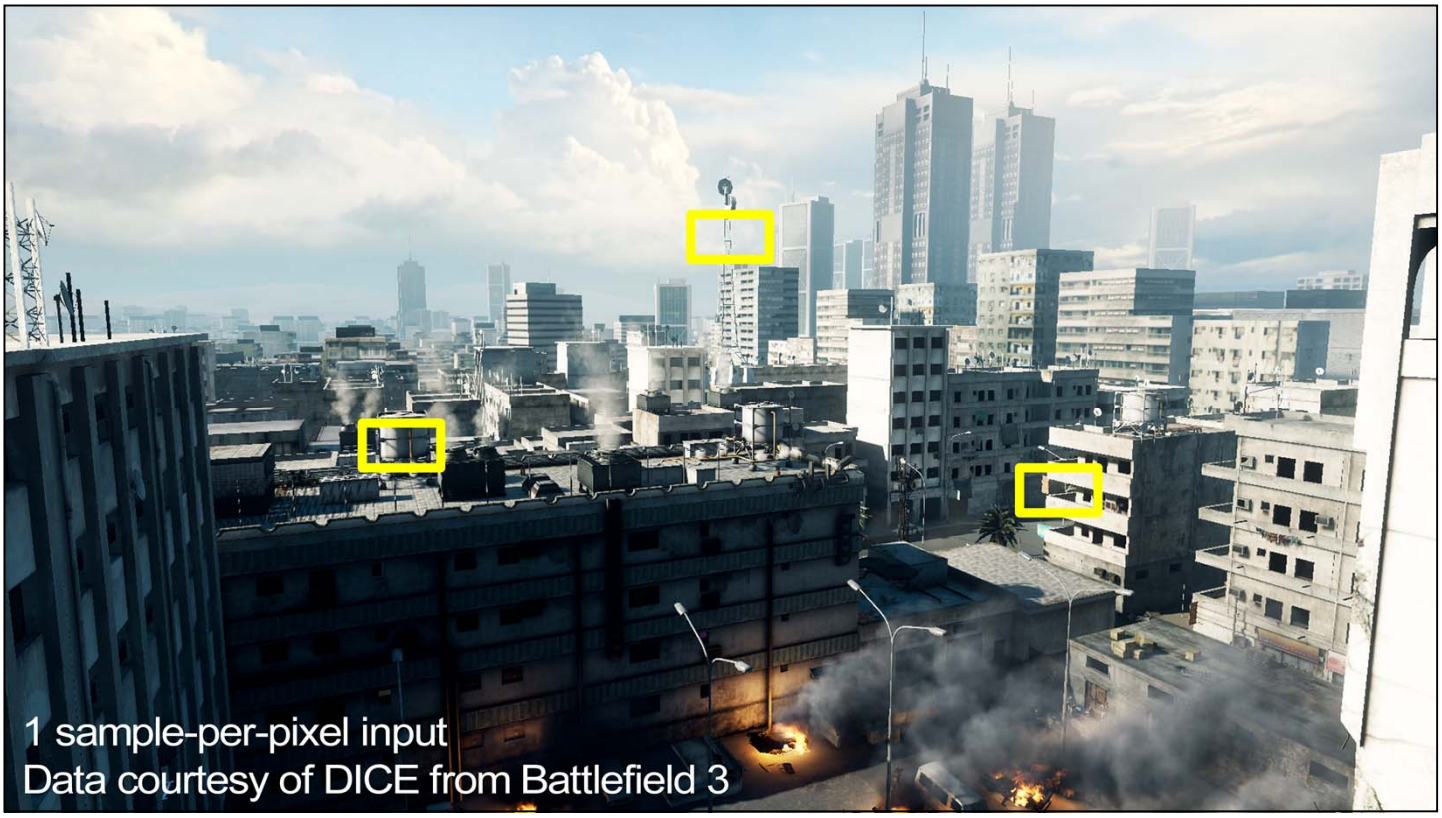
I'll zoom in on two areas with features on the order of one pixel thick.

You can see that SRAA produced antialiasing here comparable to MSAA…even though we didn't shade more than once per pixel, even when there were multiple objects in the pixel.

SRAA tracked 4x MSAA depth and then looked at nearby pixels to find shades for each of the four depth samples independently, allowing the shading and color to flow along thin objects for antialiasing. It doesn't work if there's an object that is so small that it only covers one sample ever, but it works well for individual thin objectslike wires and collections of single-sample objects like debris.

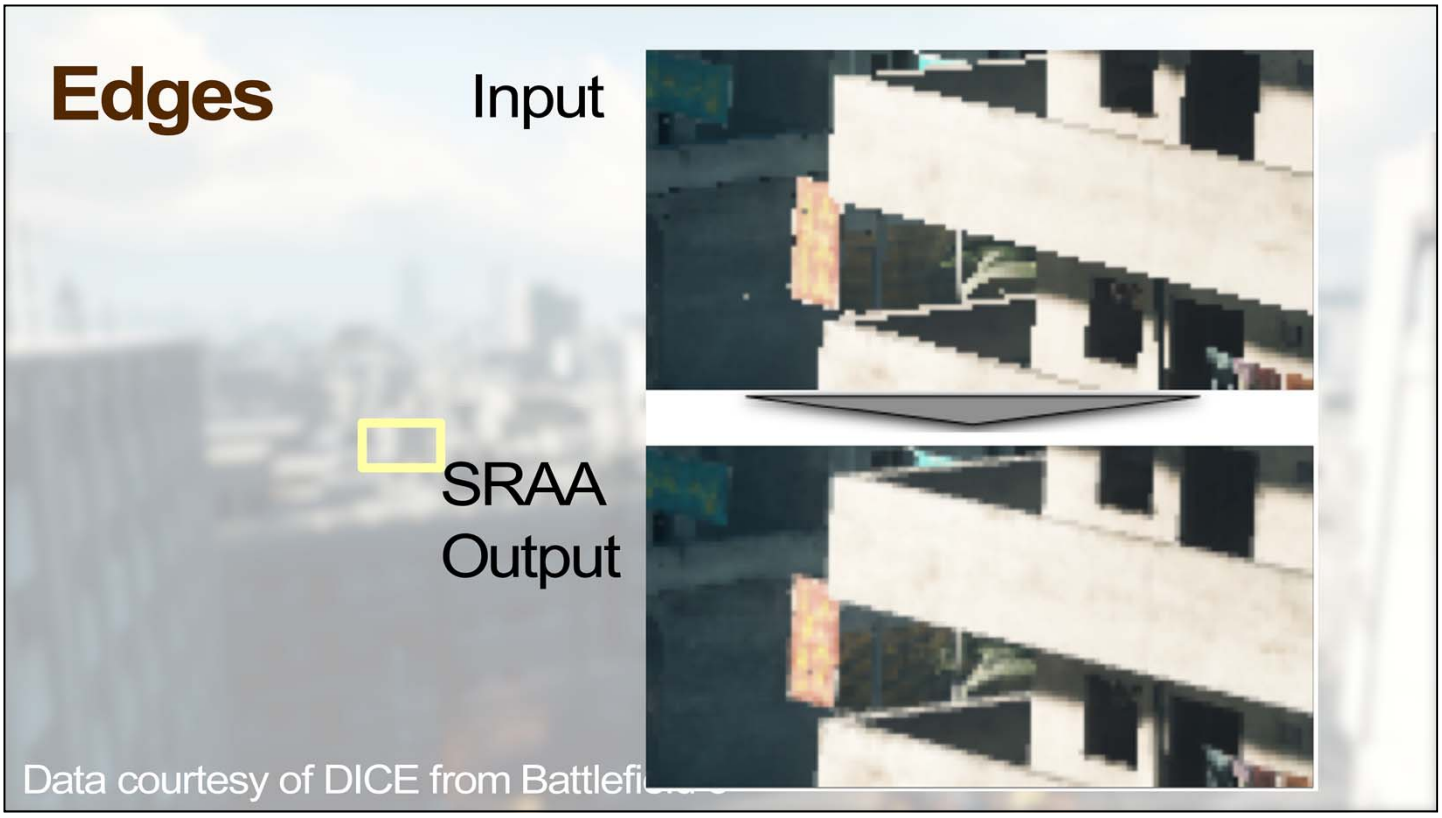
SRAA

- **SRAA is research** targeting cases where 1 spp morphological approaches fail
- Combined with another method for edges
- Performance and quality improvements since I3D 2011 paper (NVIDIA tech report coming soon)



Here's a scene by DICE rendered with one sample per pixel at 1080p. This is the input that you provide to a post-processing antialiasing algorithm.

I'm going to zoom in on three areas to show you how SRAA performs on this scene.



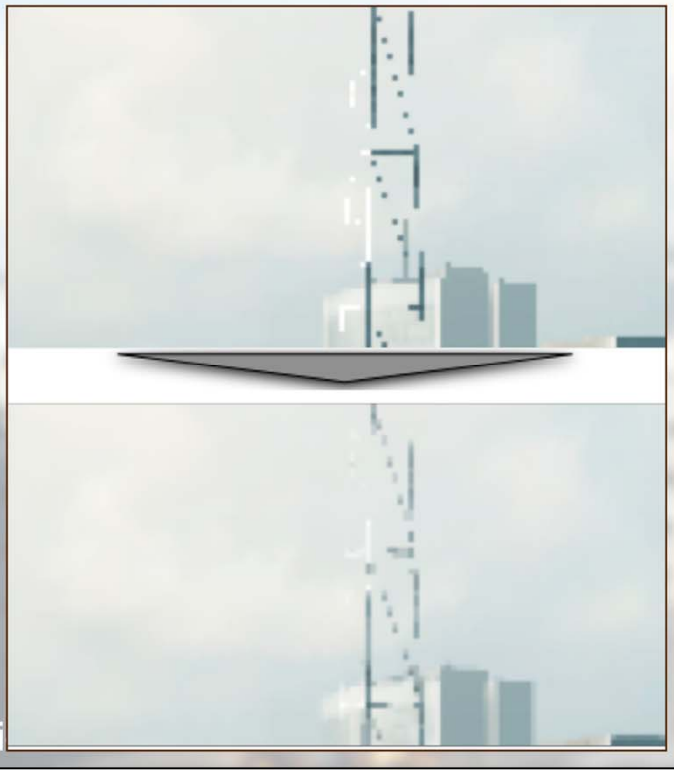
This balcony is dominated by high-contrast edges between large polygons.

SRAA produces a result comparable to MSAA on this input without any adjustment, but we've seen that morphological methods can surpass MSAA quality. This is the case where a subpixel should be combined with something like FXAA for superior results.

Now let's look at the cases where morphological methods fail, which are thin objects

Subpixel Features

Input



SRAA Output

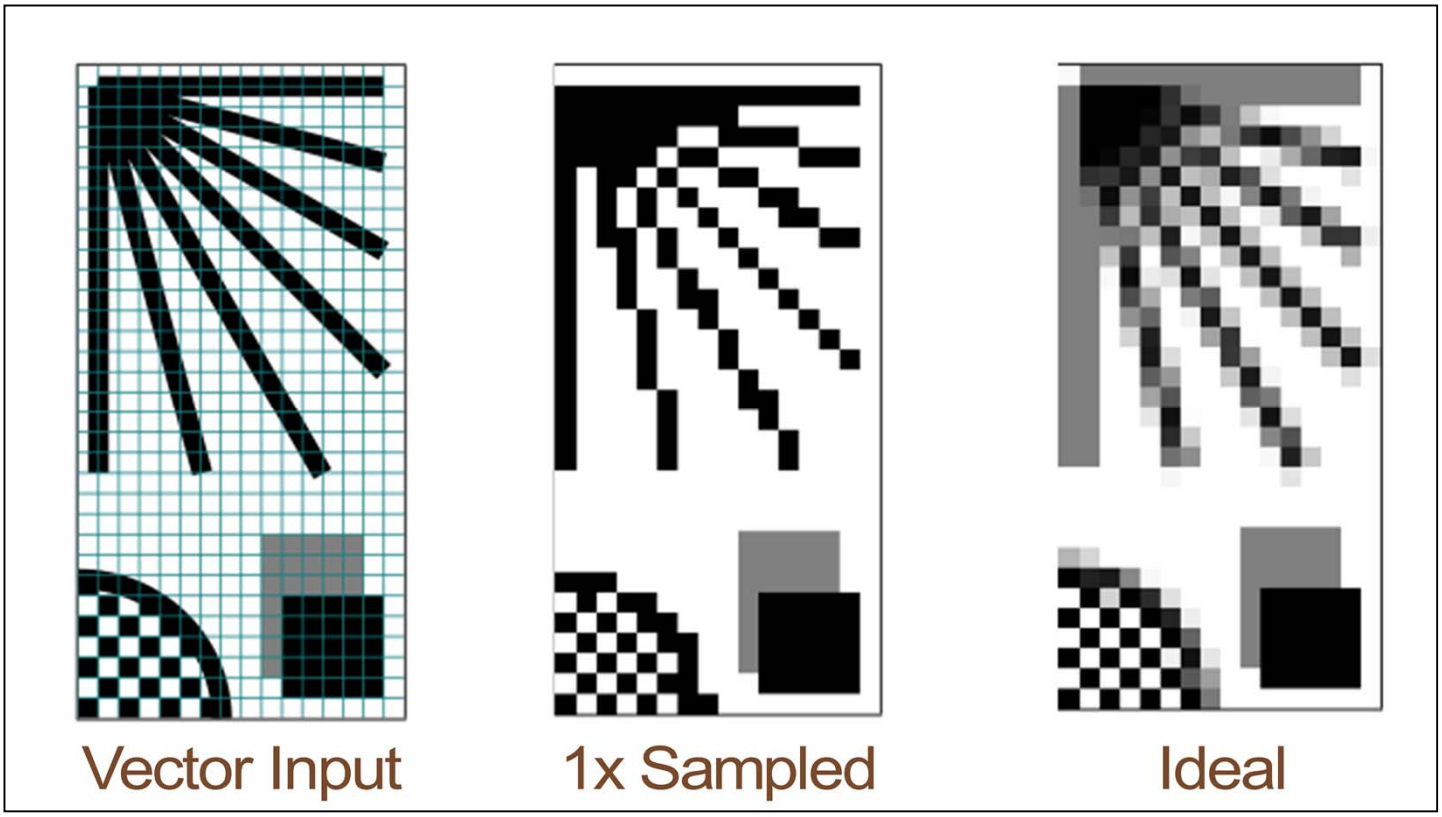
Data courtesy of DICE from Battlefi

Subpixel Features

Input

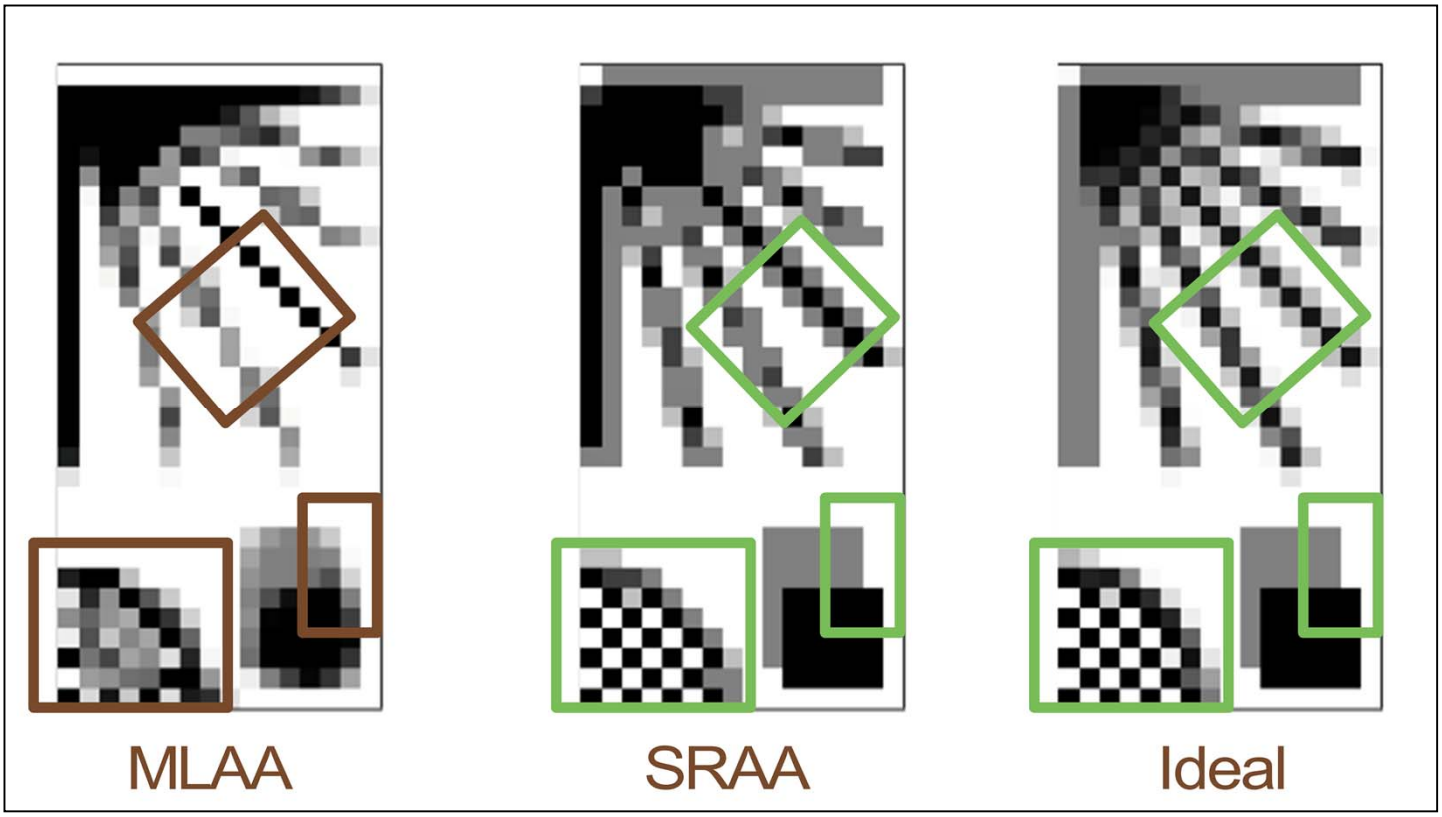
SRAA Output

Data courtesy of DICE from Battlefield 3



Here’s a more controlled experiment to show what is going on.

We constructed this vector scene and rasterized it at 1 sampler per pixel and thousands of samples per pixel



Preserve high frequencies

Preserve sharp edges that aren’t jagged from aliasing

Preserve line thickness

SRAA Attributes

- Reconstructs subpixel precision from 1x color, e.g., for:
  - Polygon edges
  - Power lines
  - Fences
  - Trees
  - Distant characters
  - CAD wireframe
  - Grass
  - Antennae
- About 1 ms on GeForce 560 at 1080p
- Requires MSAA depth buffer
- Accepts optional MSAA normal/uv/ID buffers
- Alternative: temporal or subpixel-morphological (SMAA)

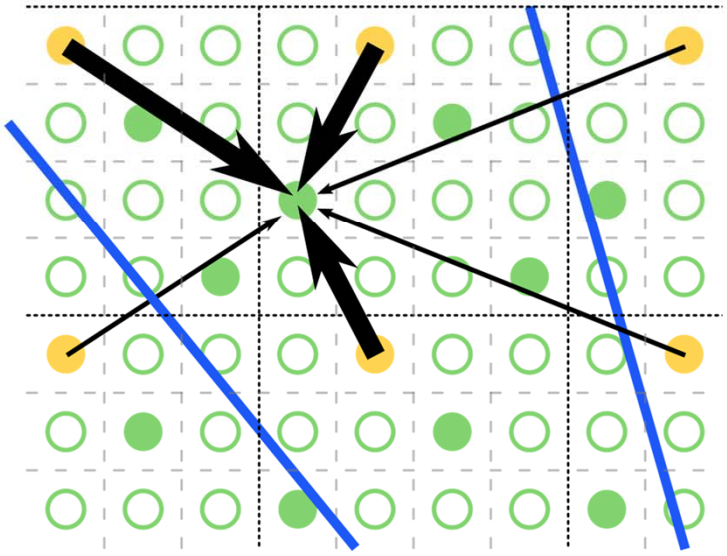
Temporal AA was used effectively on Crysis 2 and Halo Reach.

The Key Idea

Shaded sample

Edge

Geometric sample

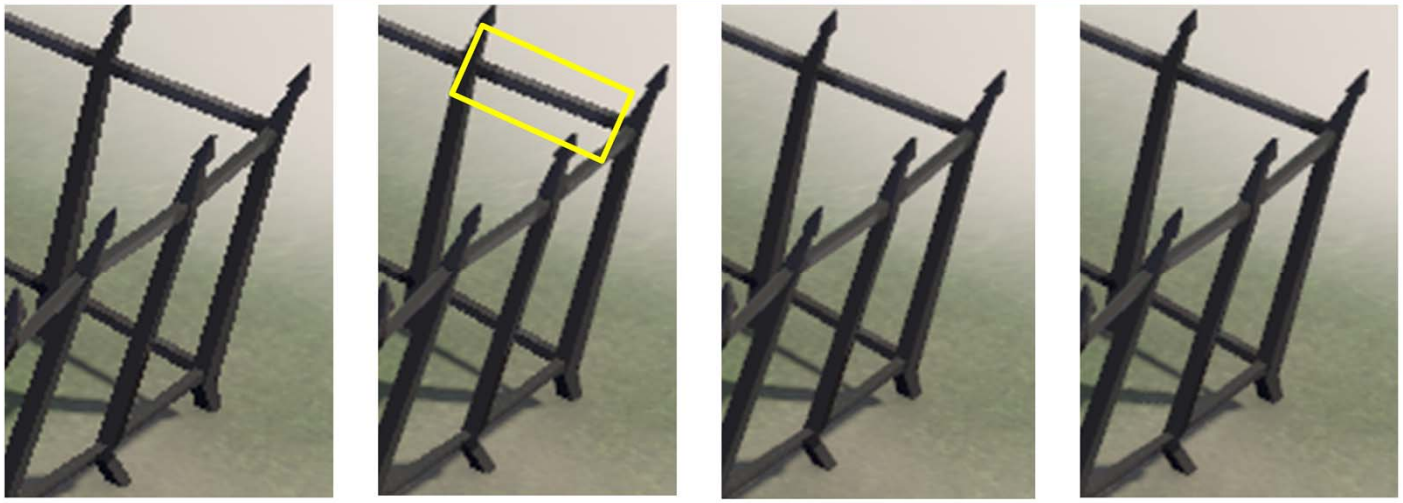


Implementation Notes

- **Combine with other methods**
  - Run FXAA on 1x color before SRAA (not shown today)
  - Keep alpha-blending for billboards
  - Primitive antialiasing on lines or select polygons
  - Compatible with SSAA, MSAA and CSAA
- **Super-sampled depth is "free"**
  - Depth-only 4x MSAA renders 3x *faster* than even 1x flat shading
  - Depth buffer read/writes are compressed on many architectures
  - Still have to pay memory cost
- **"discard" or stencil-mask pixels without edges**

Space and quality

Space and quality



Input (no AA)	Depth16F	Depth16F + RGB8 Normal	R8 Primitive ID
	16.6 MB	49.8 MB	8.2 MB

All run in about 0.55 ms at 1920x1080 on GeForce 560 Ti

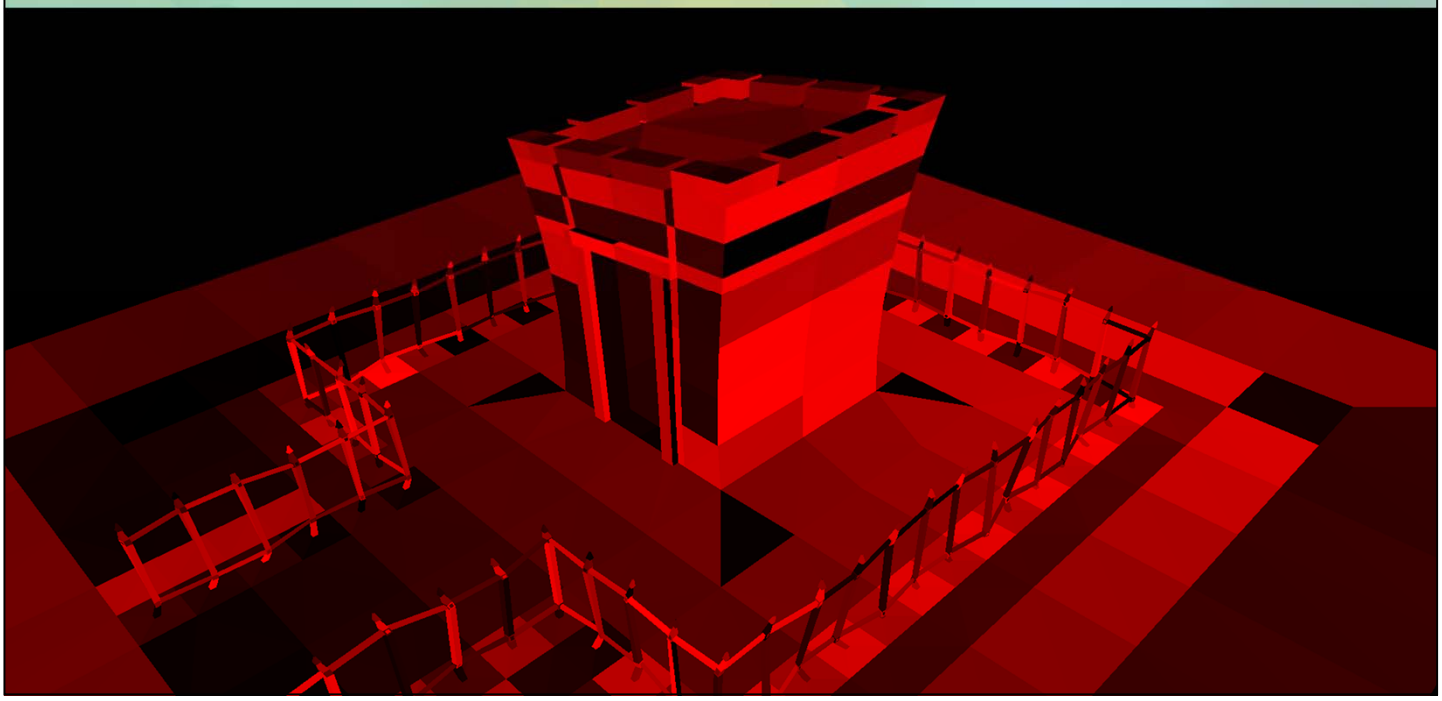
Depth alone is efficient to render, but at 1080p 4x costs an additional 16.6 MB over the original framebuffer even at 16 bits per sample.

It also fails to discriminate all cases.

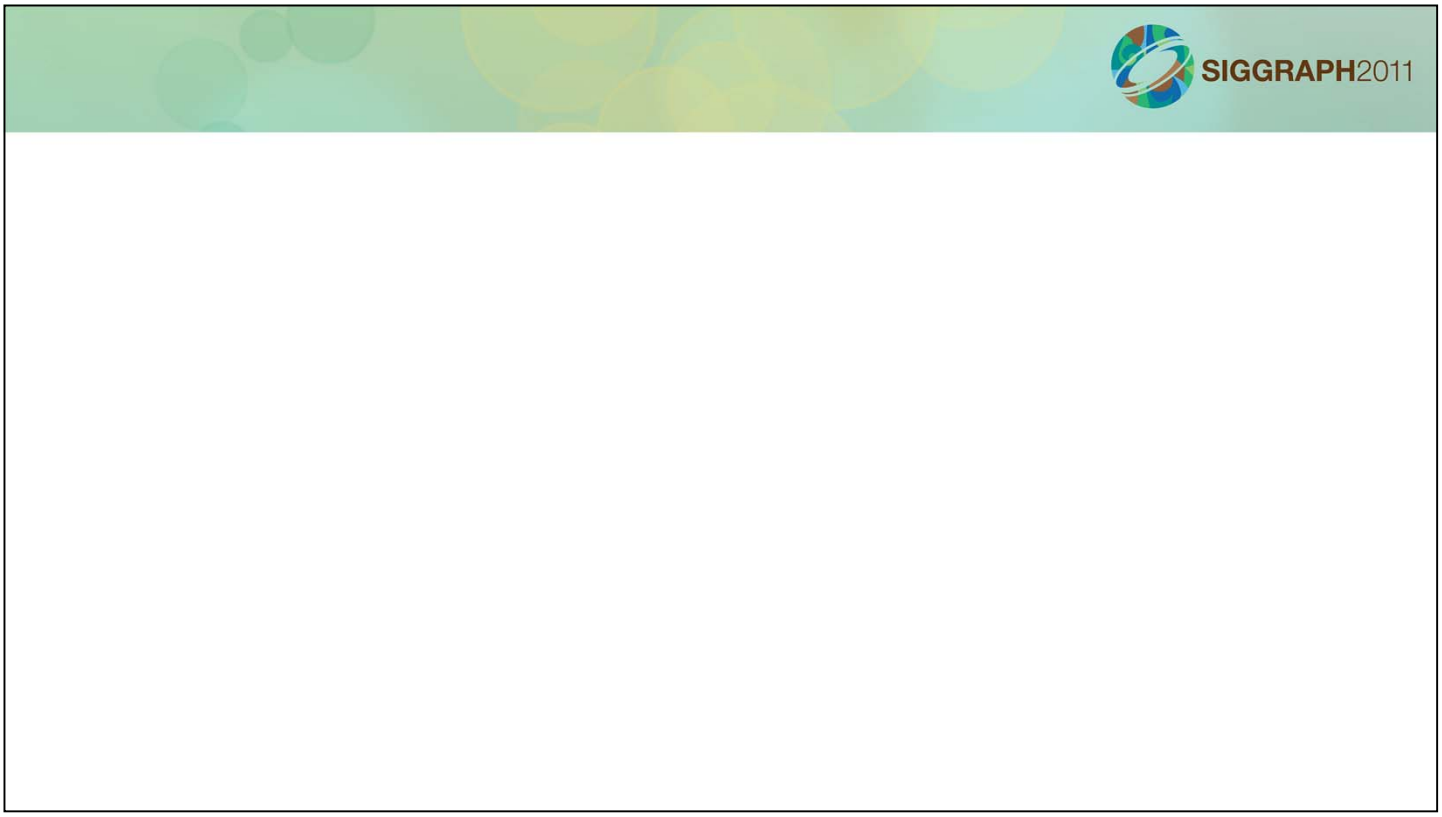
Additional information, such as normals, produces better cross-bilateral weights but requires significant space.

We found that as simple a heuristic as the tessellation primitive ID is sufficient for antialiasing purposes in many scenes and can be compressed into eight bits.

R8 Primitive ID Buffer



Here’s what the primitive ID buffer looks like.


SIGGRAPH2011

Conclusion

- **I currently use FXAA** for large-triangle edge antialiasing
- **A small amount of MSAA** data can improve subpixel-triangle antialiasing
  - FXAA + SRAA, SMAA
- **Primitive IDs are an effective** and compact heuristic discriminator for AA