A Lazy Object-Space Shading Architecture With Decoupled Sampling

Christopher A. Burns
Kayvon Fatahalian[†]
William R. Mark

Intel Corporation

[†] Stanford University

I. Motivation

The Long-Term Objective

- High geometric detail
- Motion blur and depth-of-field
- Efficient support for these in a real-time system

*Ray traced by Gilles Tran

*Photo by "Austinii" @ http://austinii.deviantart.com

Two Possible Approaches...

	GPU	Reyes
Motion Blur, DOF		X
Detailed Geometry		X
Adaptive Tessellation	X	
Fine Occlusion Culling	X	

Two Possible Approaches...

1. Start with GPU pipeline, add missing features

	GPU	Reyes
Motion Blur, DOF	[Ragan-Kelley '10]	X
Detailed Geometry	[Fatahalian '10]	X
Adaptive Tessellation	X	
Fine Occlusion Culling	X	

Two Possible Approaches...

1. Start with GPU pipeline, add missing features
2. Start with Reyes pipeline, improve overall efficiency

	GPU	Reyes
Motion Blur, DOF	[Ragan-Kelley '10]	X
Detailed Geometry	[Fatahalian '10]	X
Adaptive Tessellation	X	
Fine Occlusion Culling	X	

Instead, Modify the Reyes Pipeline

We evolve Reyes in two ways:

1. Decouple shading from triangle vertices (Decoupling)
2. Shade partial grids after rasterization (Lazy Shading)

Instead, Modify the Reyes Pipeline

We evolve Reyes in two ways:

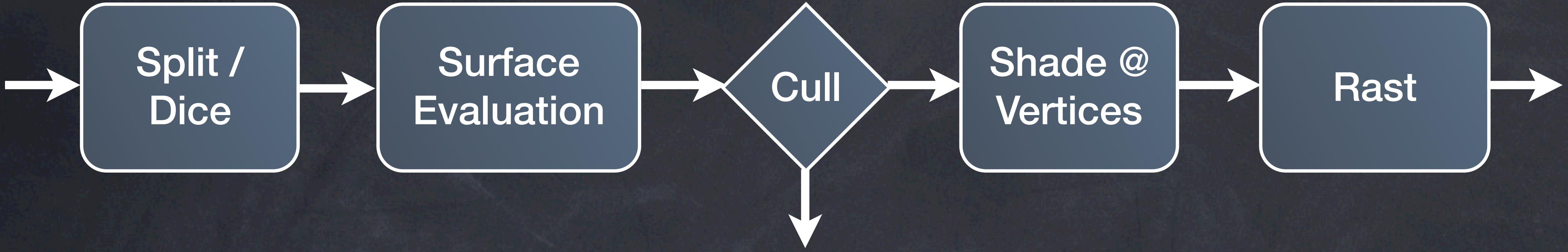
1. Decouple shading from triangle vertices (Decoupling)
2. Shade partial grids after rasterization (Lazy Shading)

We achieve two things:

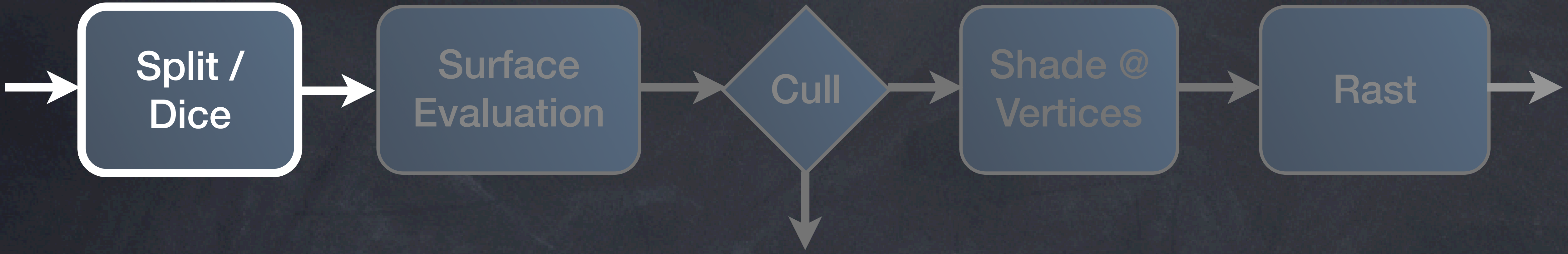
1. Eliminate the need for half-pixel polygons
2. Eliminate redundant shading via fine-grained occlusion culling

II. Reyes Overview

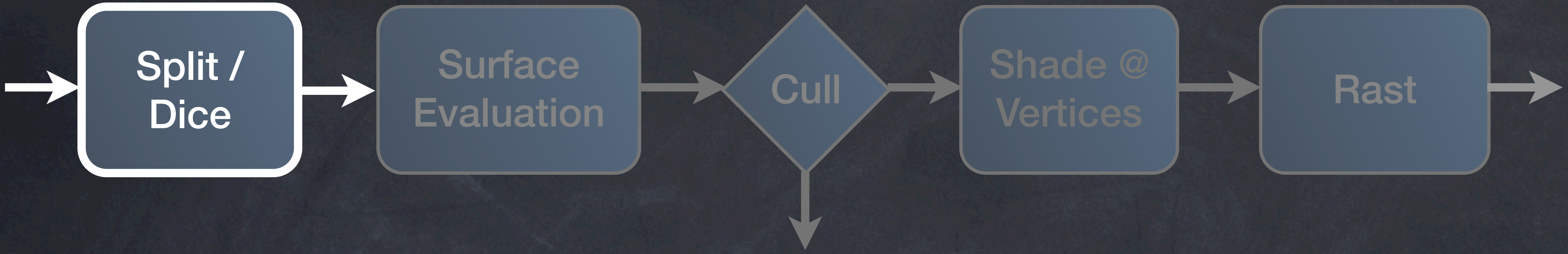
Reyes Pipeline Overview



Reyes Pipeline Overview



Reyes Pipeline Overview



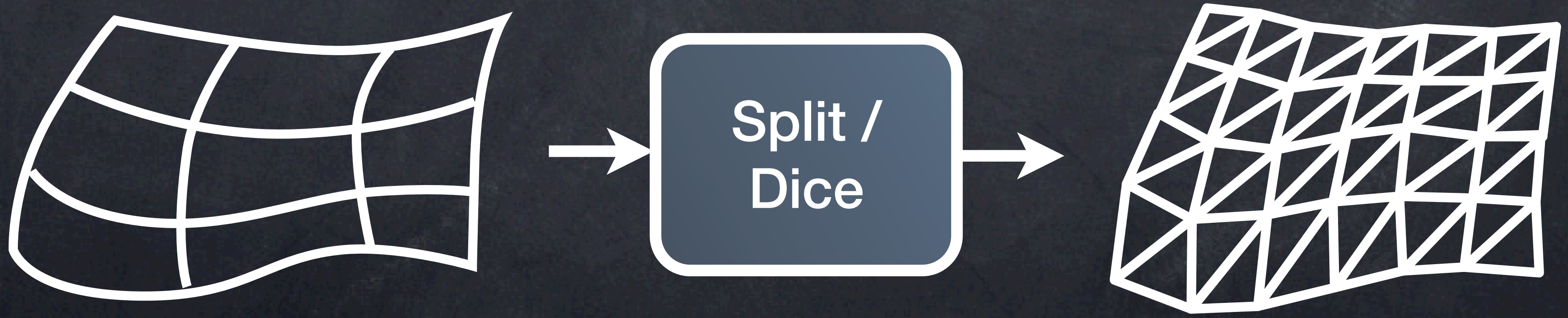
Split / Dice

Surface Evaluation

Cull

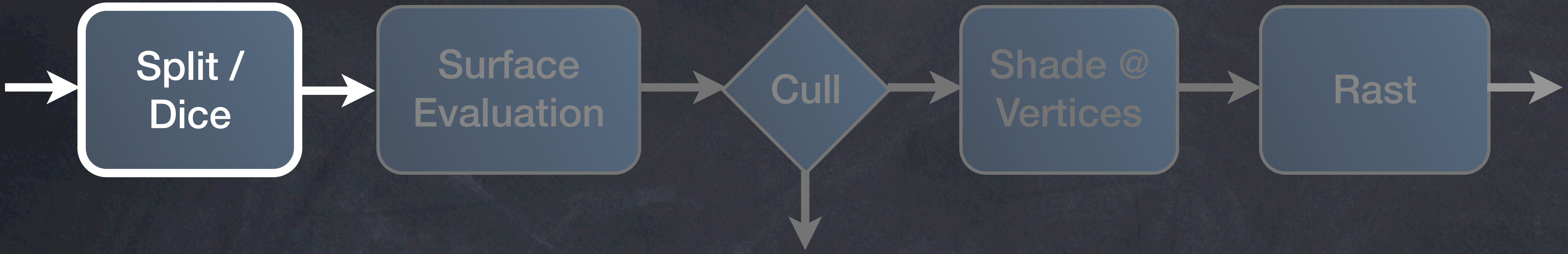
Shade @ Vertices

Rast

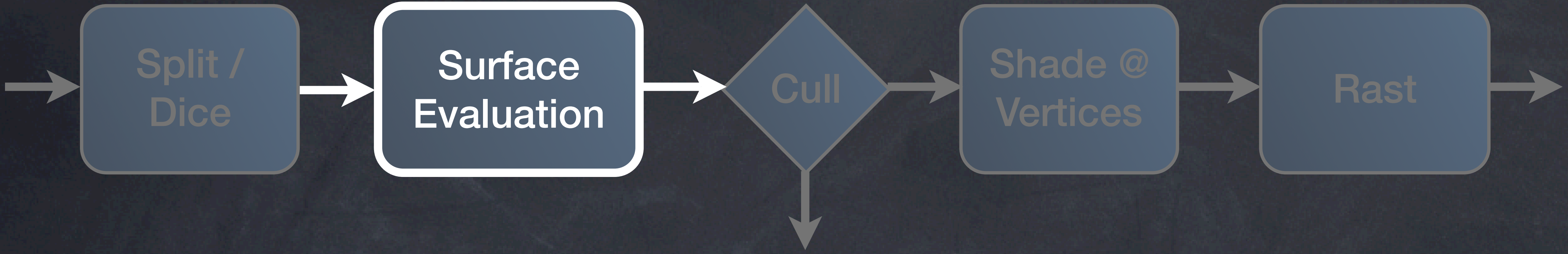


Split / Dice

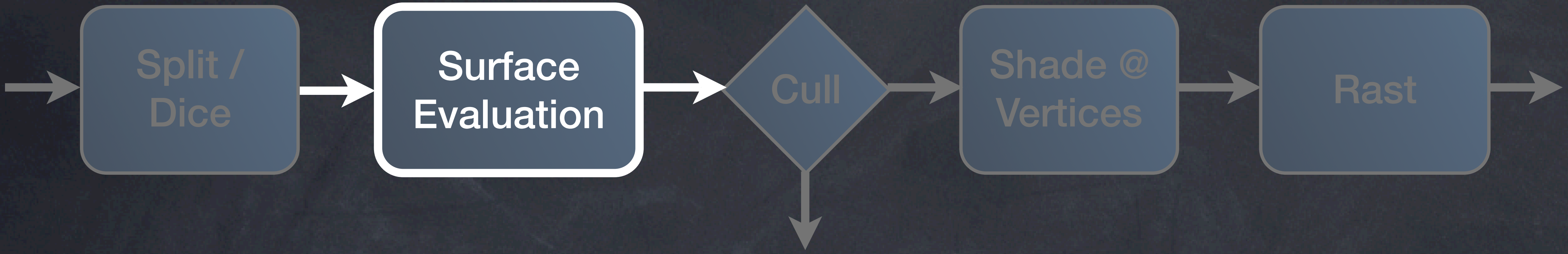
Reyes Pipeline Overview



Reyes Pipeline Overview



Reyes Pipeline Overview



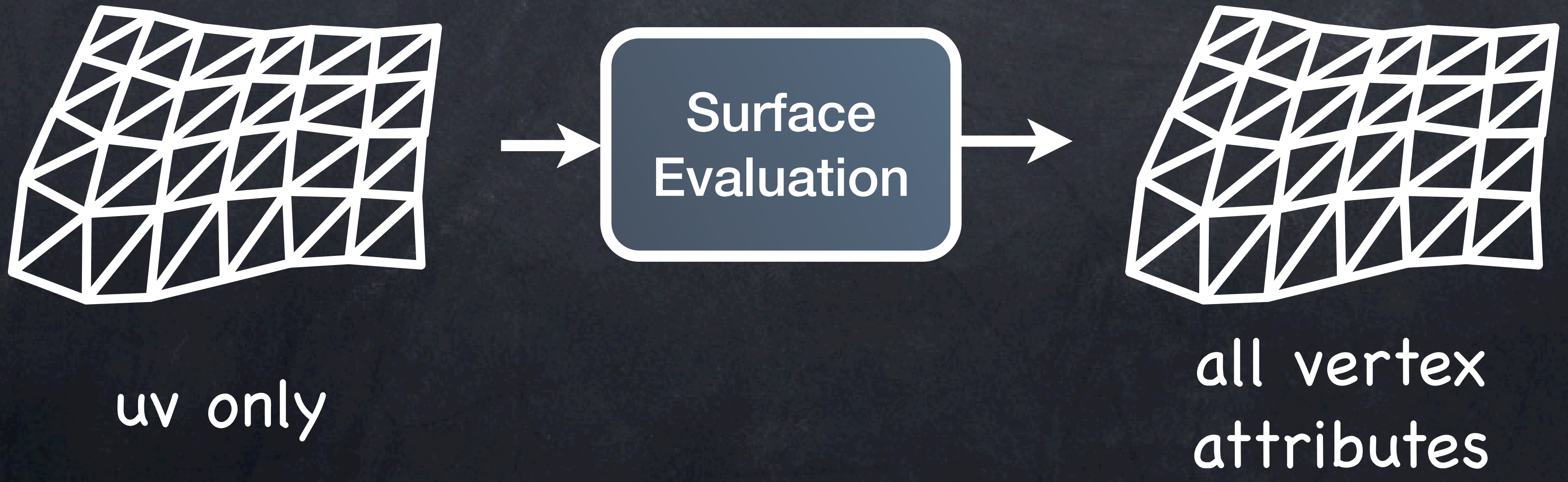
Split / Dice

Surface Evaluation

Cull

Shade @ Vertices

Rast

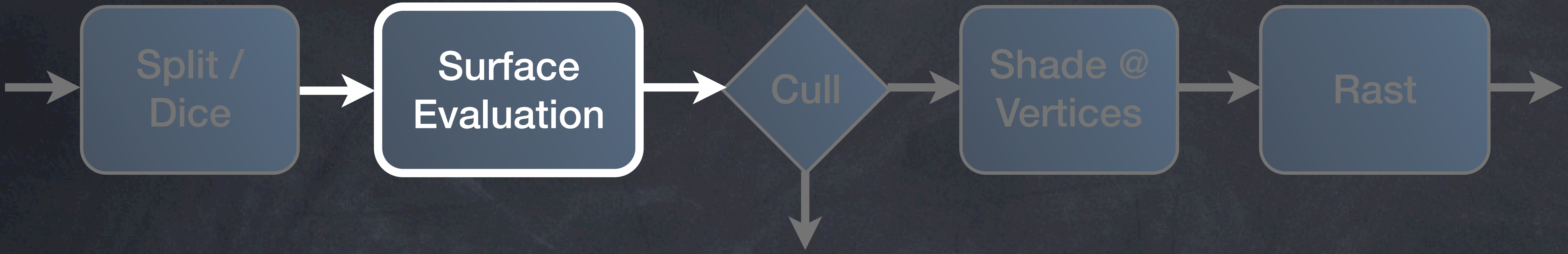


Surface Evaluation

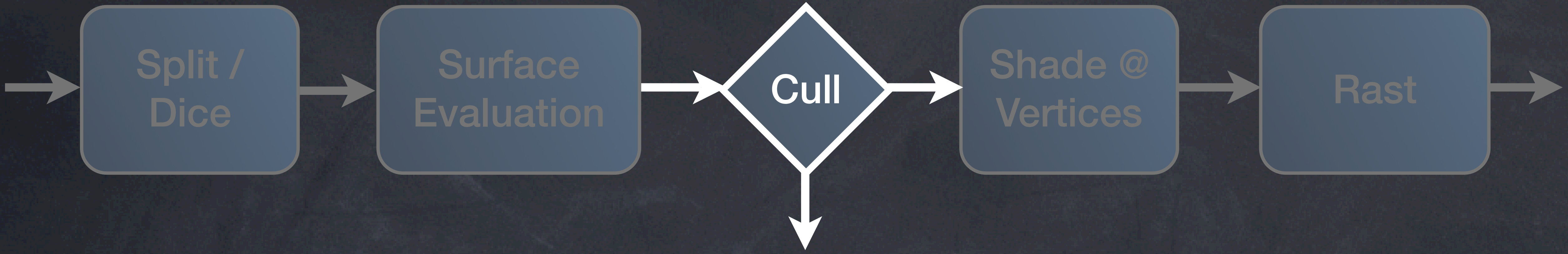
uv only

all vertex attributes

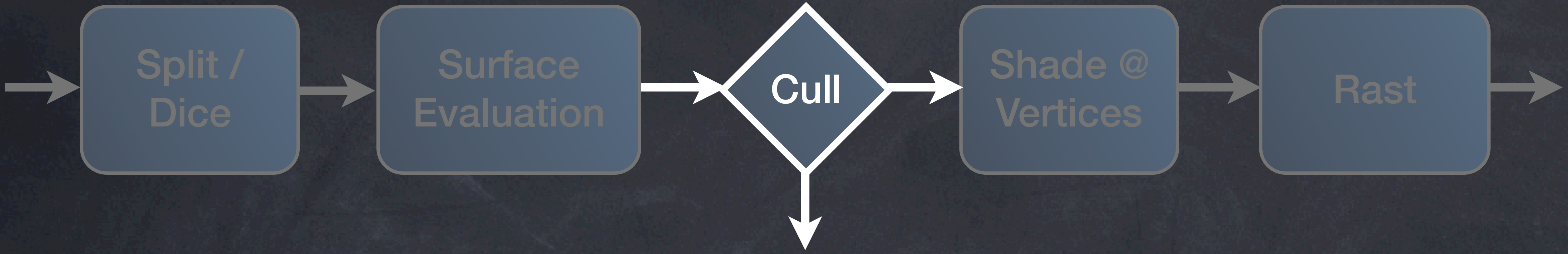
Reyes Pipeline Overview

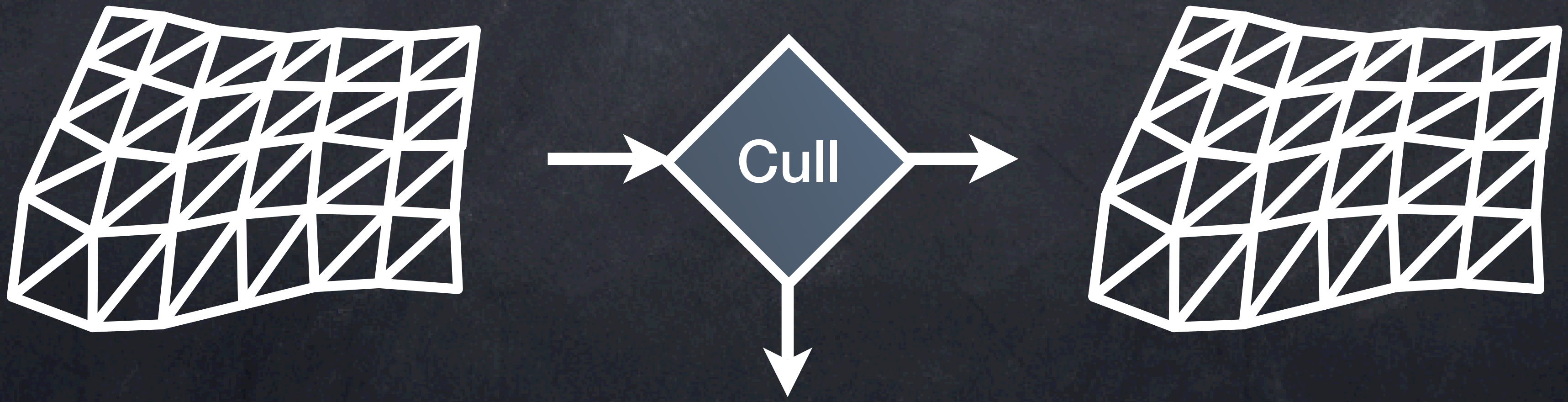


Reyes Pipeline Overview

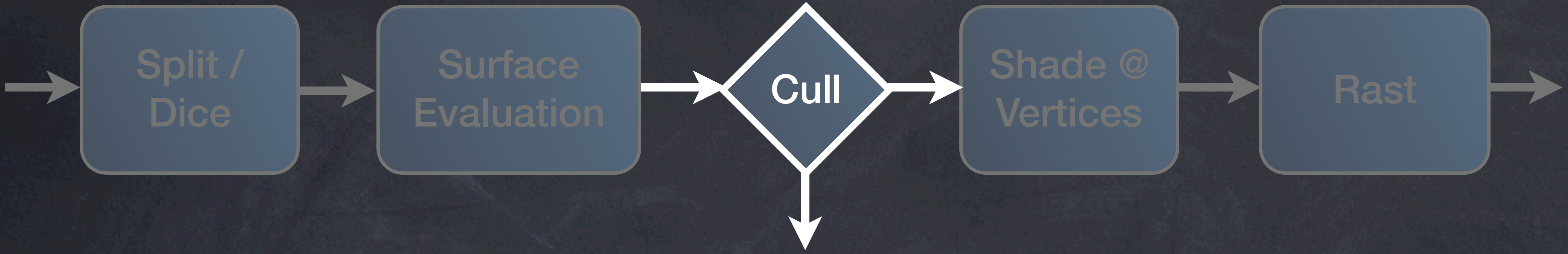


Reyes Pipeline Overview

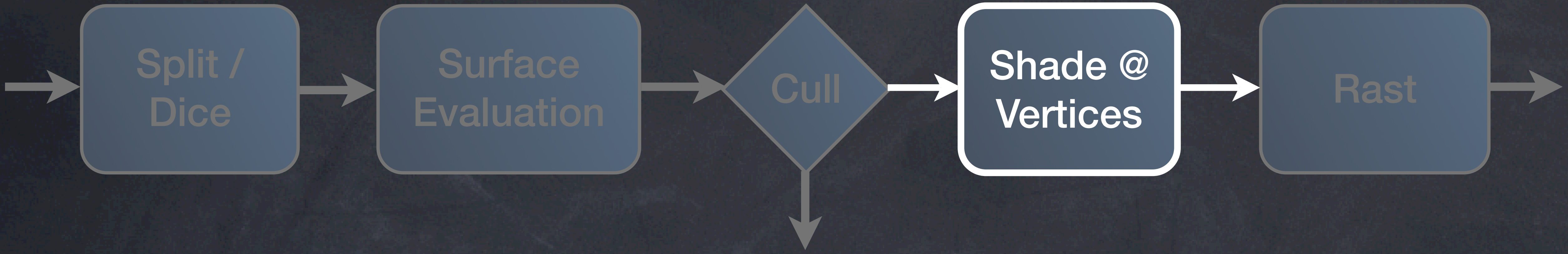




Reyes Pipeline Overview



Reyes Pipeline Overview



Split / Dice

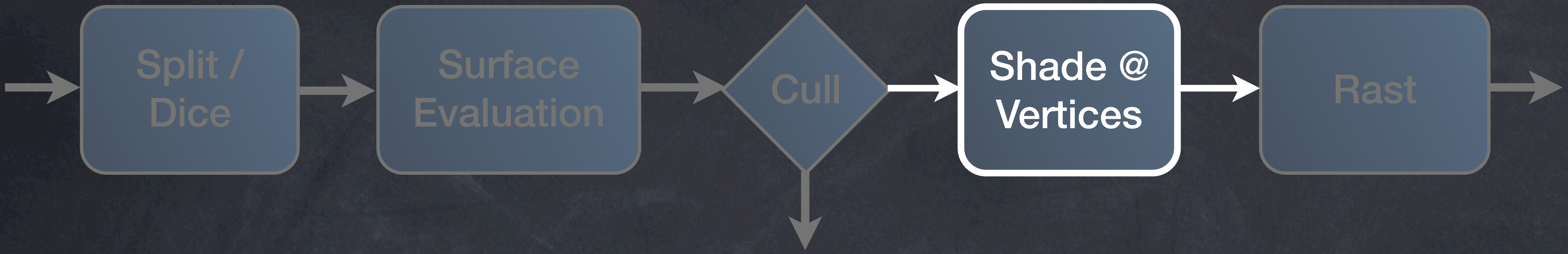
Surface Evaluation

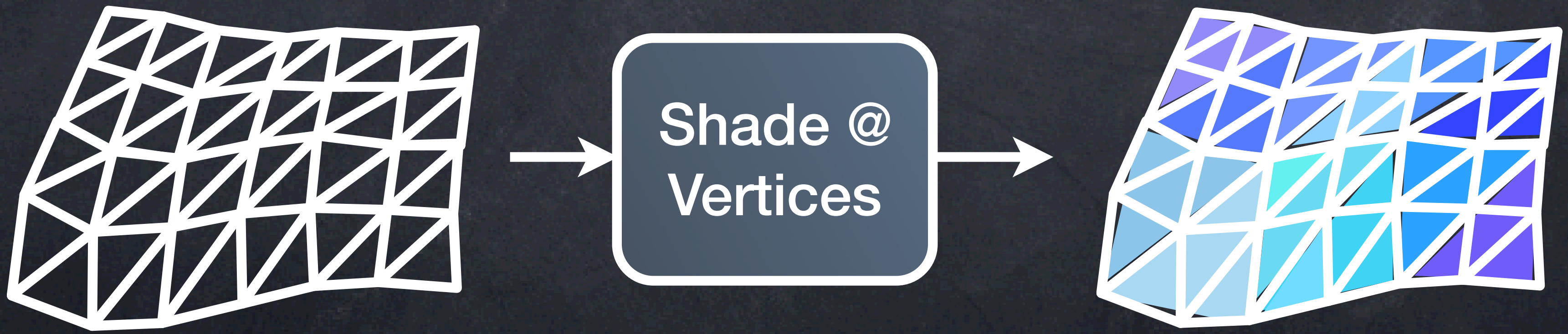
Cull

Shade @ Vertices

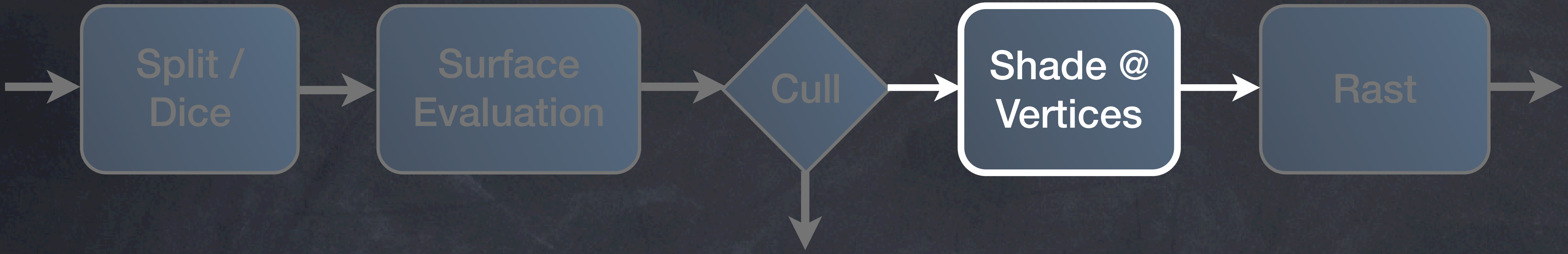
Rast

Reyes Pipeline Overview

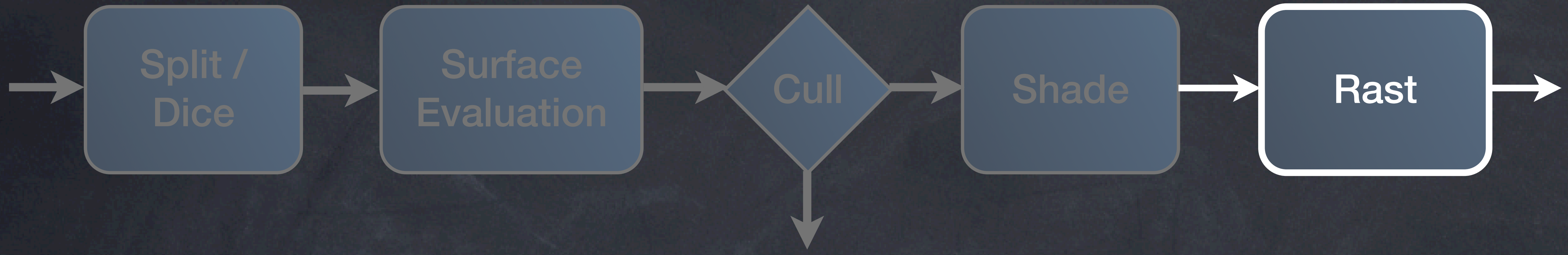


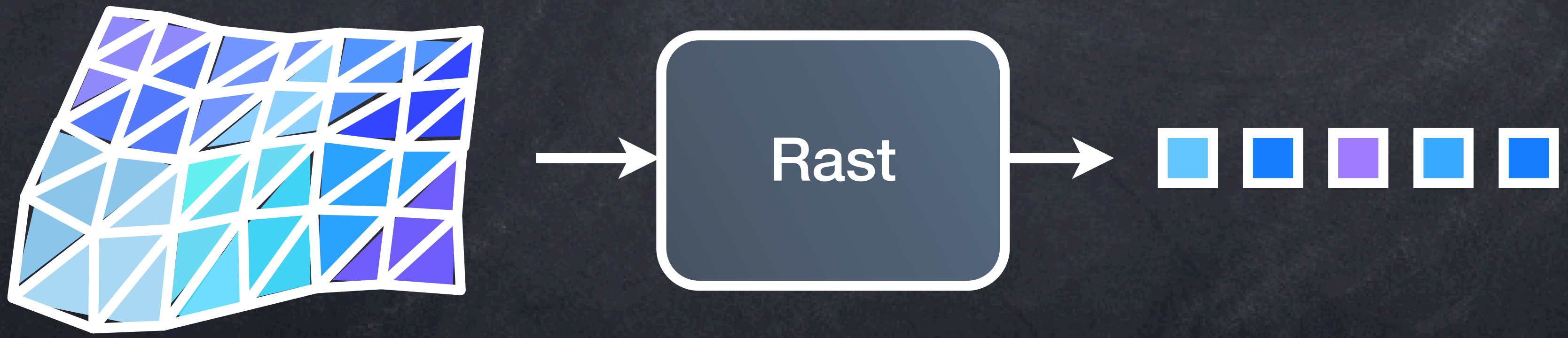


Reyes Pipeline Overview



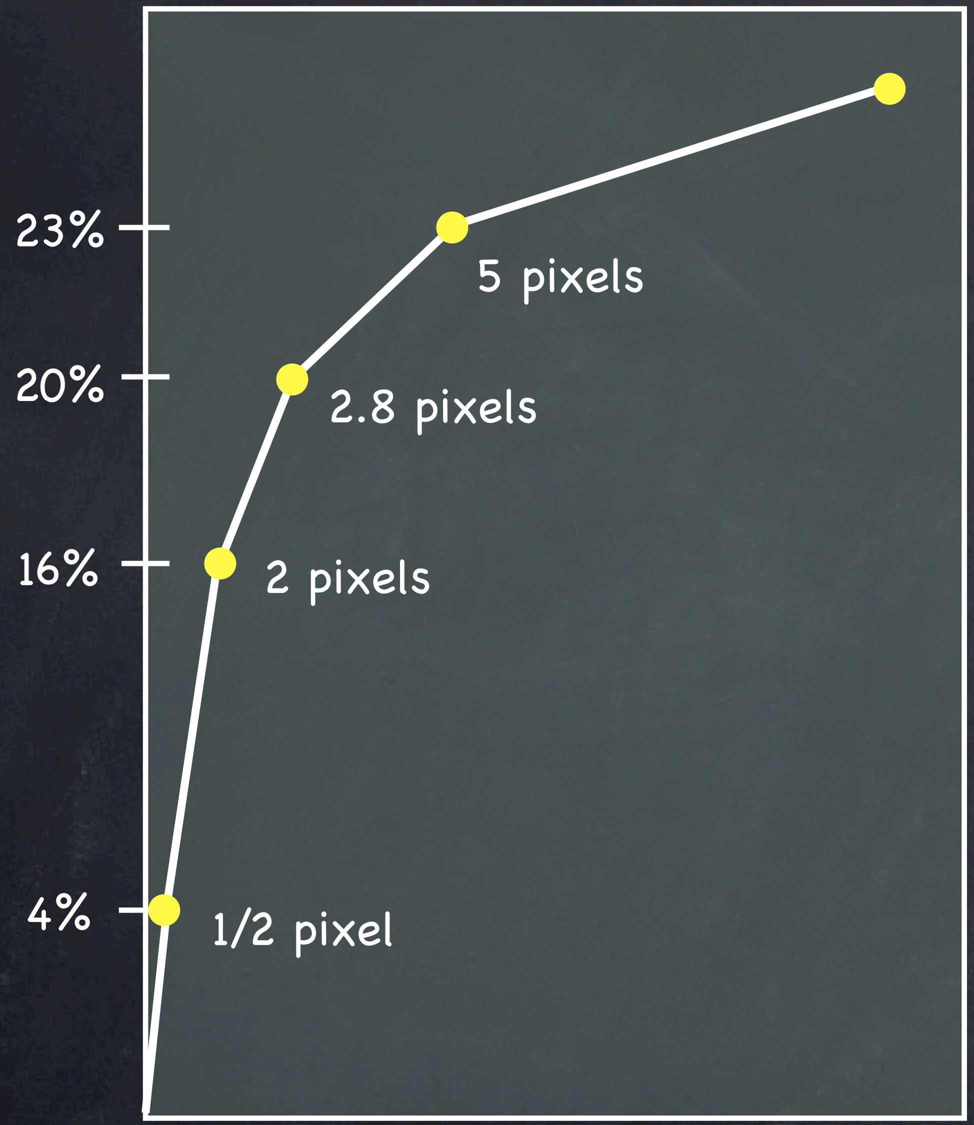
Reyes Pipeline Overview



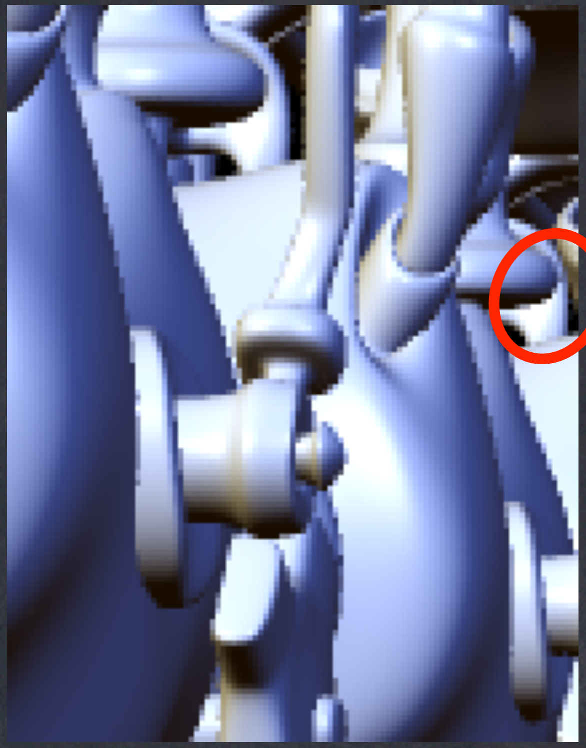


III. Decoupled Shading

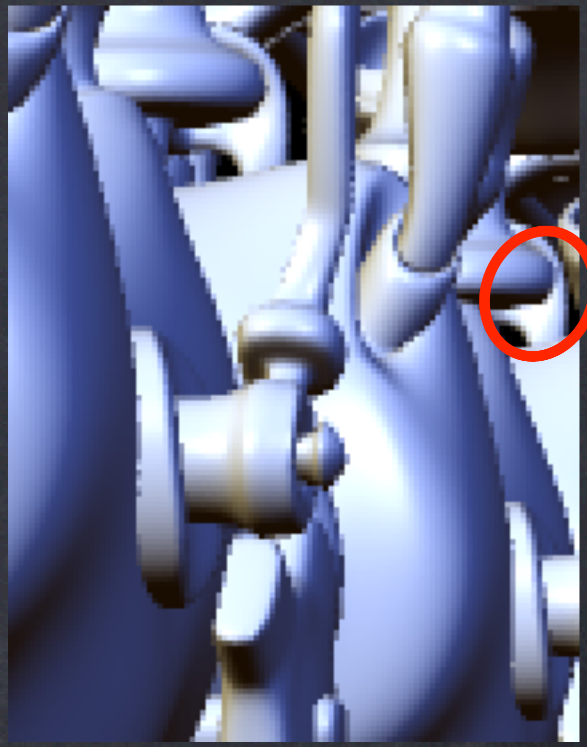
Micropolygons are Expensive



Sample Test Efficiency
v. Triangle Area



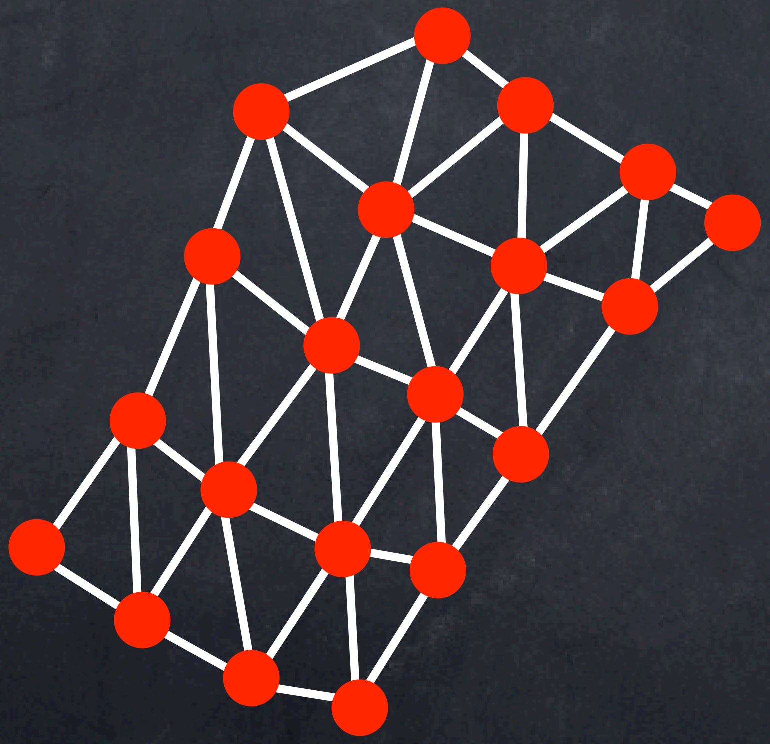
0.5 pixels/tri



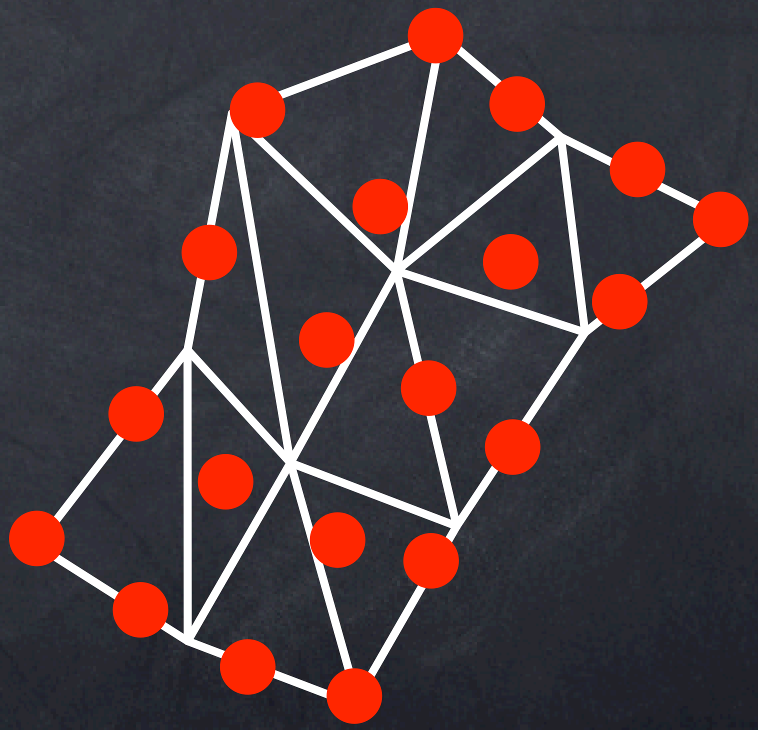
7.7 pixels/tri

Decouple Shading From Vertices

- Reyes gives one dial: shading rate
- We give two: shading rate and tessellation rate
- Micropolygons unnecessary for good shading



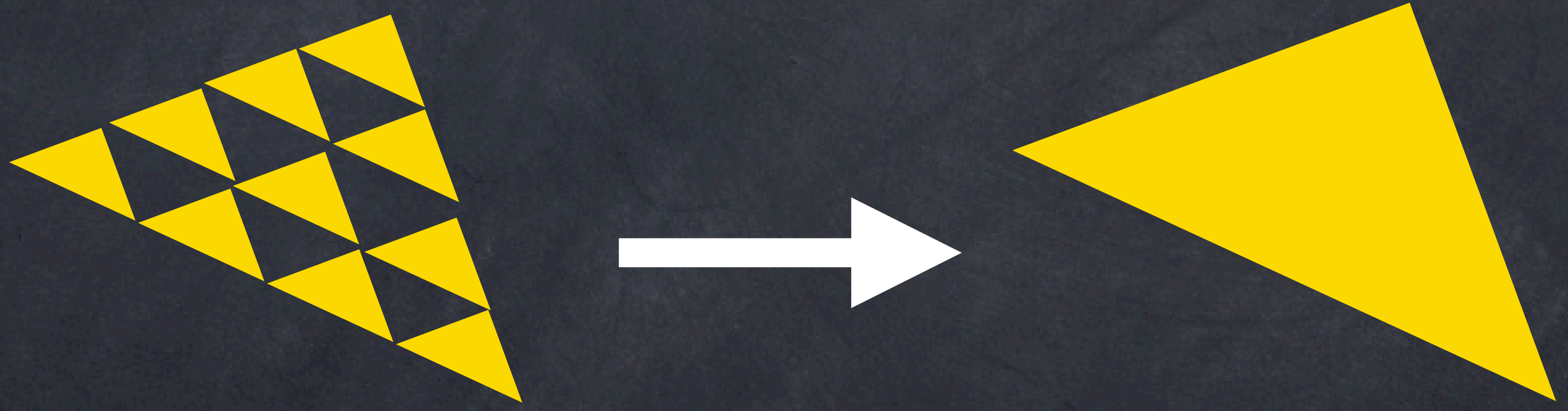
Reyes Vertex Shading



Decoupled w/ Shading Grid

Decoupled Shading Results

Larger Triangles = Fewer Triangles

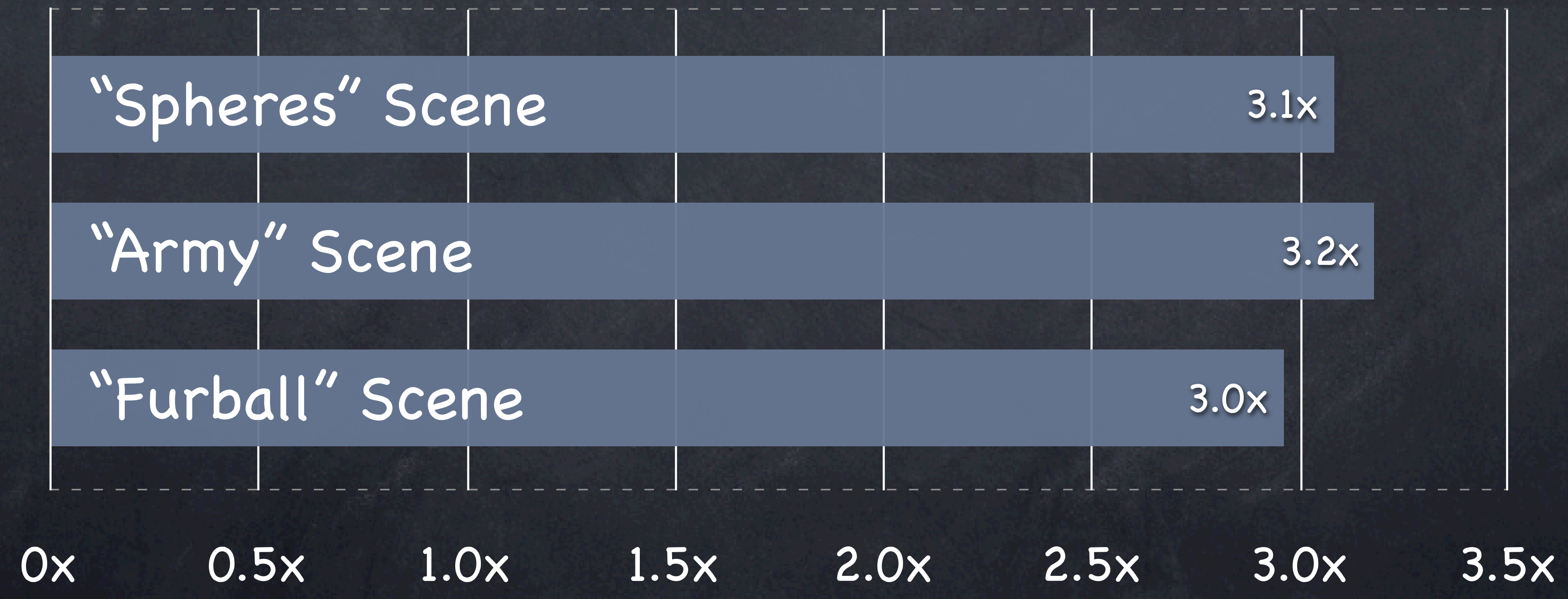


0.5 pixels
per triangle

7 pixels
per triangle

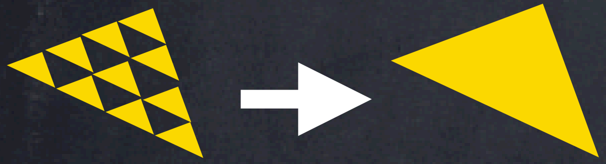
Decoupled Shading Results

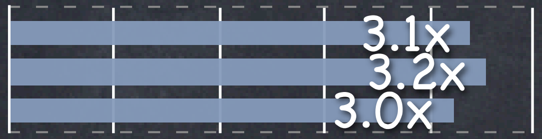
Larger Triangles = Greater Sample Test Efficiency



Decoupled Shading Analysis

10x Fewer Triangles + 3x Improved Efficiency ≈ 4x Reduced Rast Costs

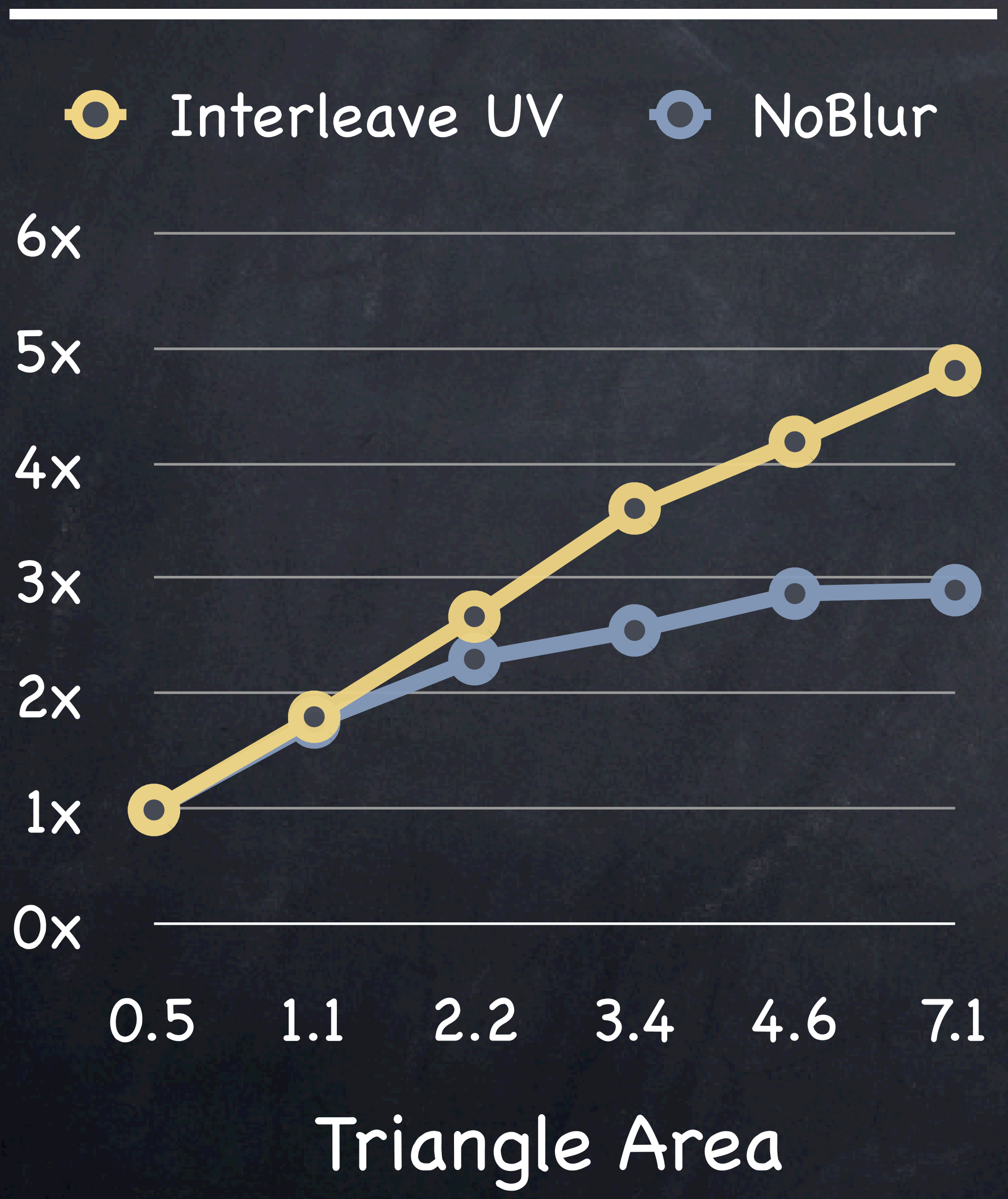




Derived from timing scalar implementation of UVT-interleave at 16 samples per pixel

Decoupled Shading Analysis

Rasterization Speedup vs. Triangle Size



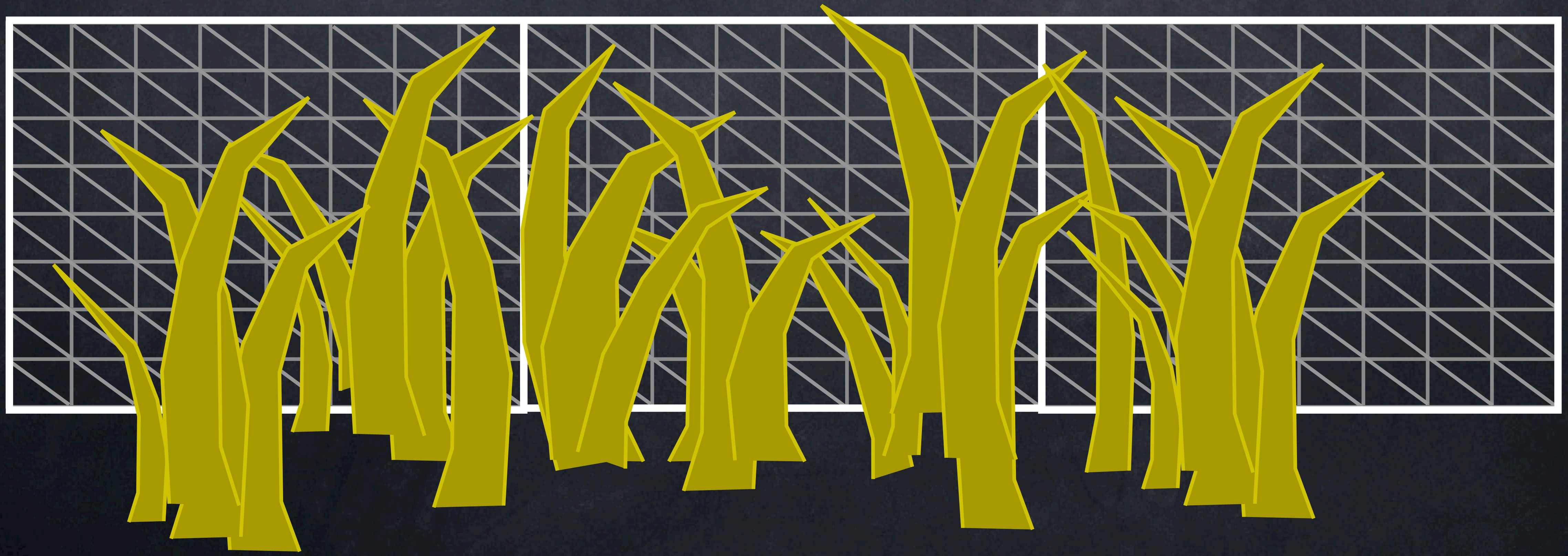
Marginal cost of adding blur support to rasterization decreases as triangle size increases

Implication: Micropolygons make blur more expensive

IV. Shading Post-Rasterization

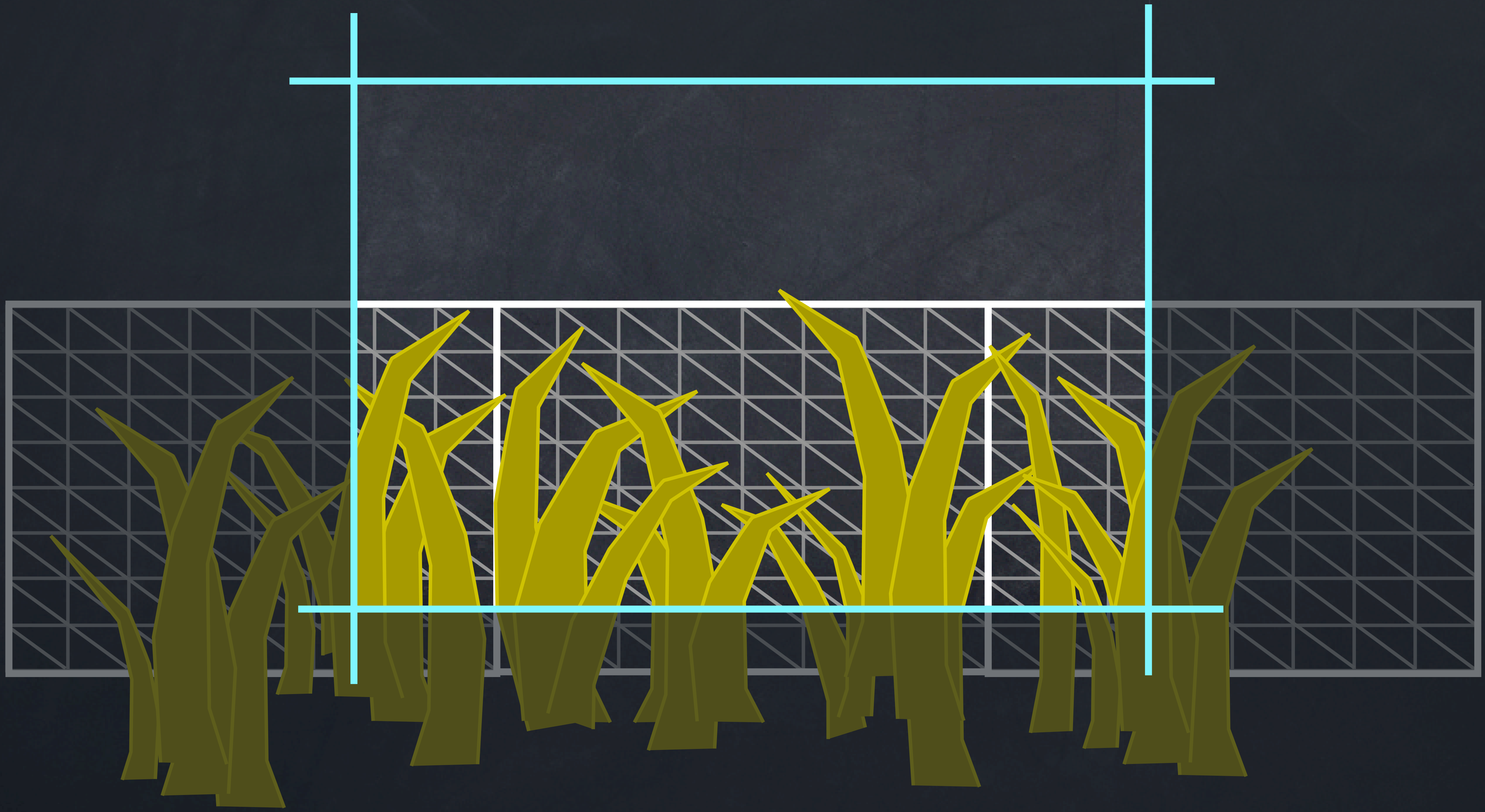
Reyes Shades Things Unseen

If any part of a grid is visible, the entire grid is shaded

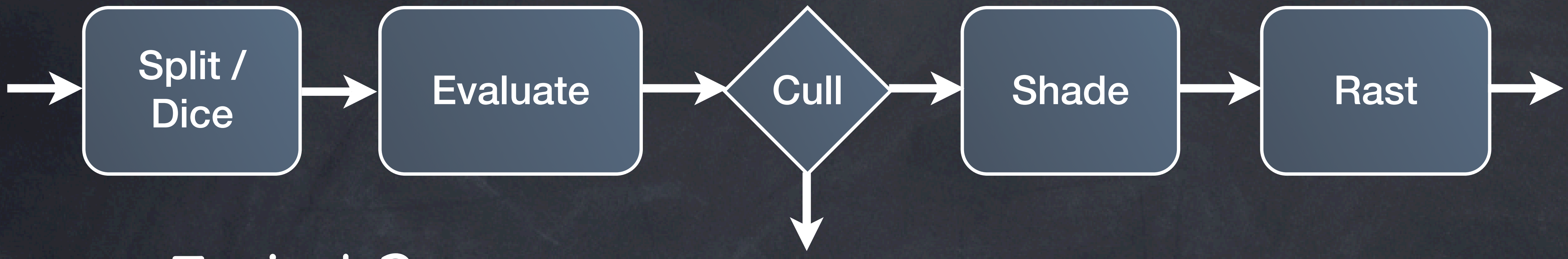


Reyes Shades Things Unseen

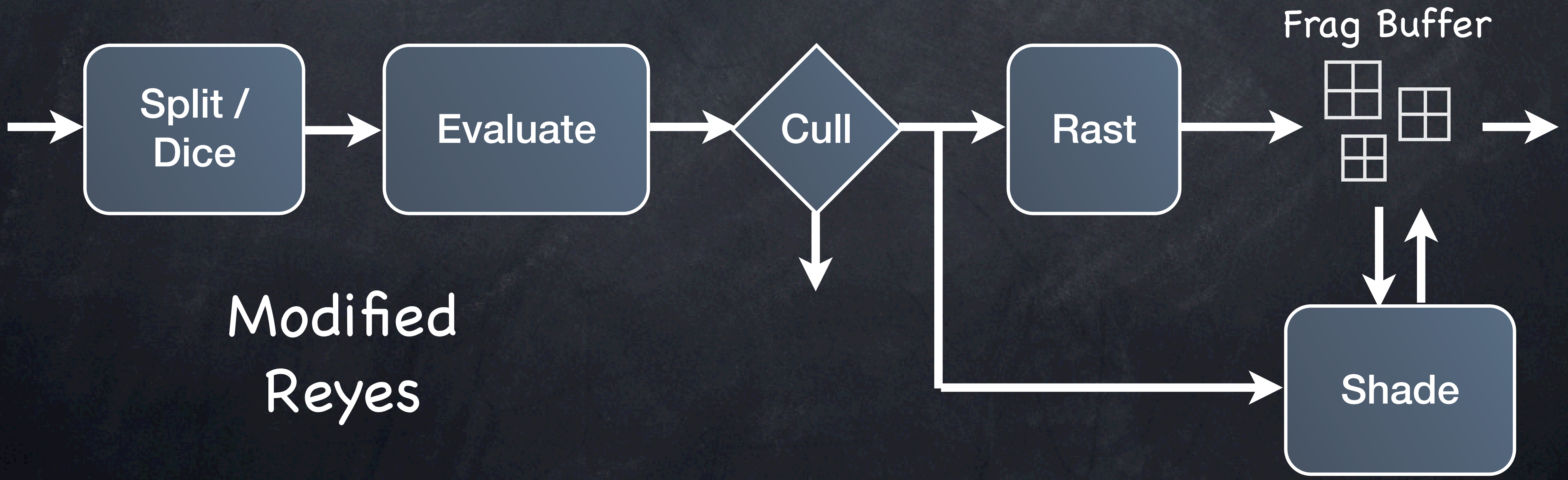
If any part of a grid is visible, the entire grid is shaded



Shading Lazily in Reyes



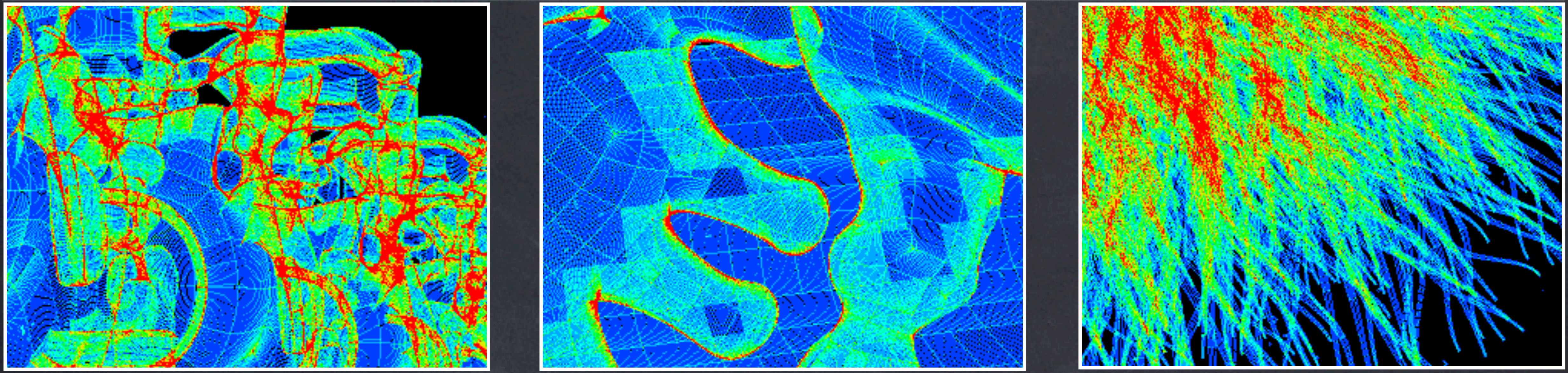
Typical Reyes



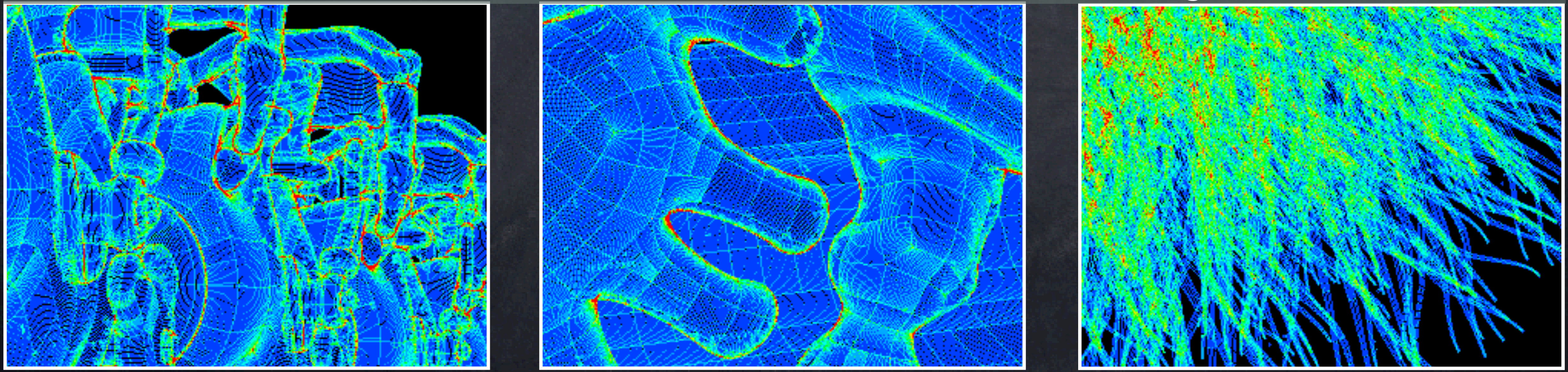
Modified Reyes

Shader Execution is Reduced

Traditional Reyes



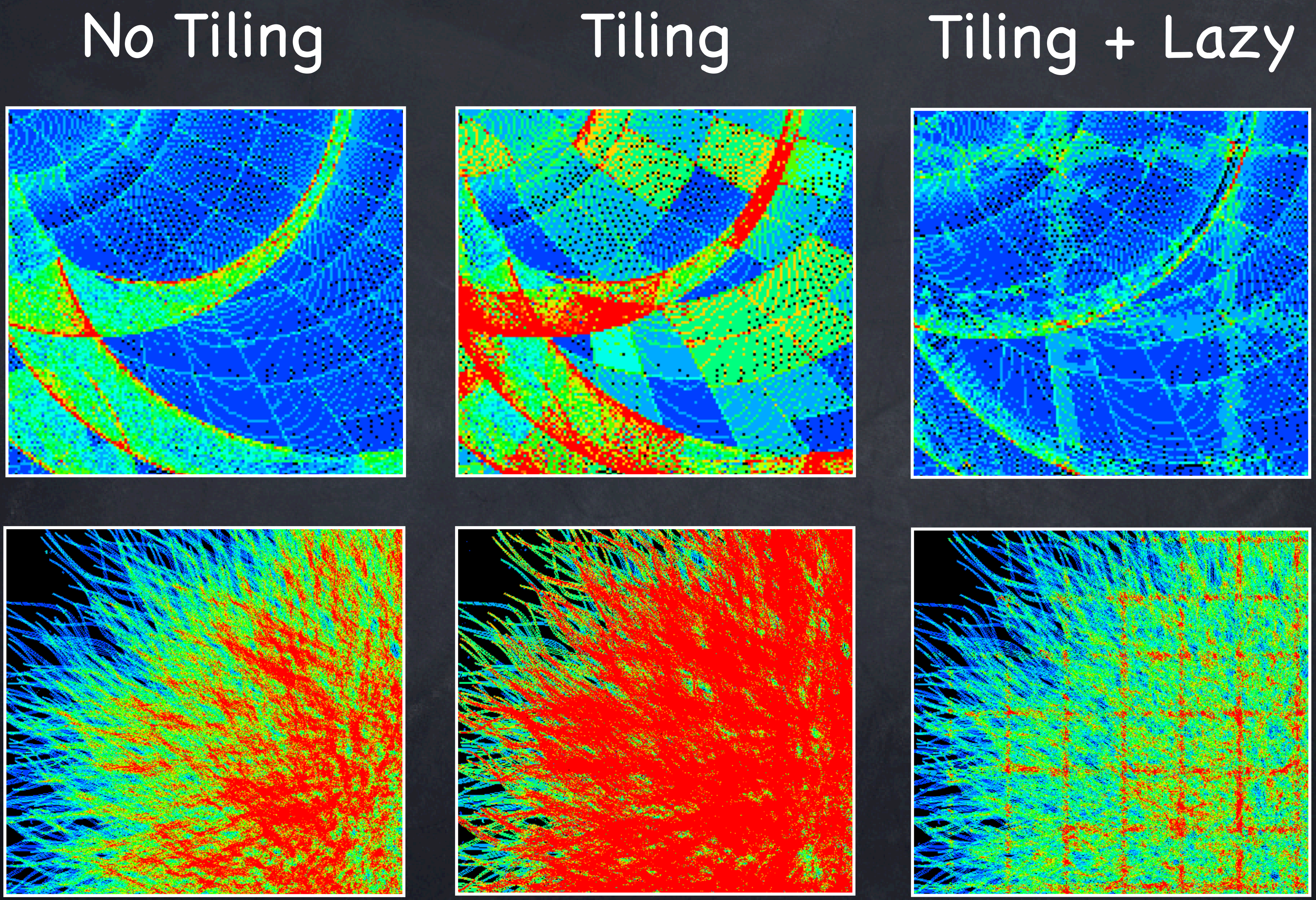
Reyes w/ Post-Rasterization Shading



Shader Executions Per Pixel

1	2	3	4	5	6	>6

Tiling Worsens Overshading

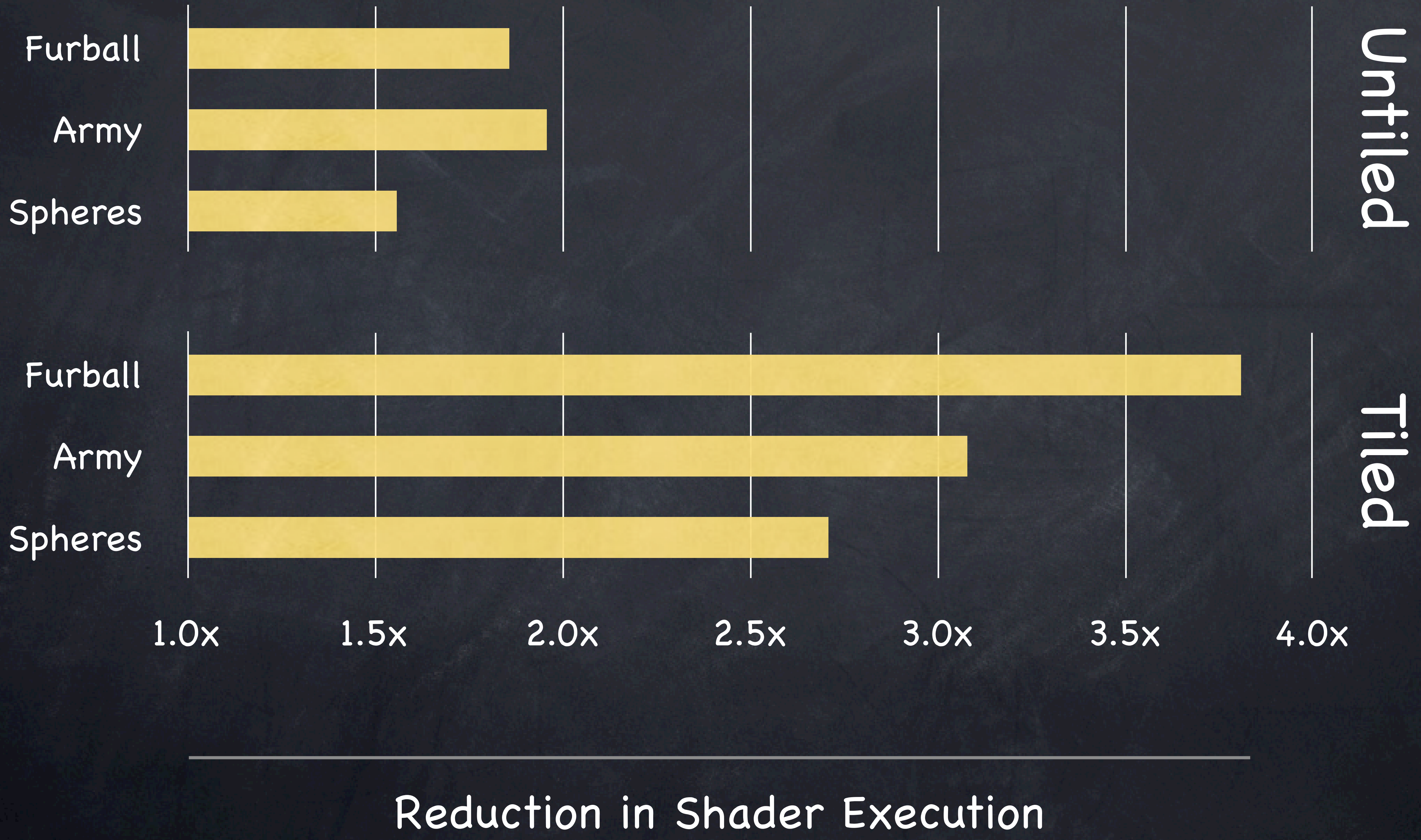


No Tiling

Tiling

Tiling + Lazy

We Shade Less Post-Rasterization



Conclusion - Summary

1. Object-space shading can be done w/out micropolygons
   - Significantly reduces rasterization costs
   - Reduced marginal cost of stochastic rasterization

2. Redundant shading can be significantly reduced
   - Especially in a tiled renderer

Final Thoughts

- Micropoylgons mostly unnecessary
  - Displaced geometry approx. by shading at non-silhouettes
  - Need fancy tessellator to optimize adaptive tessellation
- GPUs and Reyes may converge
  - We want best of both worlds
  - Much recent work in this direction [Fatahalian10, Ragan-Kelley10]