UCDAVIS
UNIVERSITY OF CALIFORNIA

Real-Time Reyes

Analysis of a Programmable Rendering Pipeline

Anjul Patney, Stanley Tzeng and John D. Owens
University of California, Davis

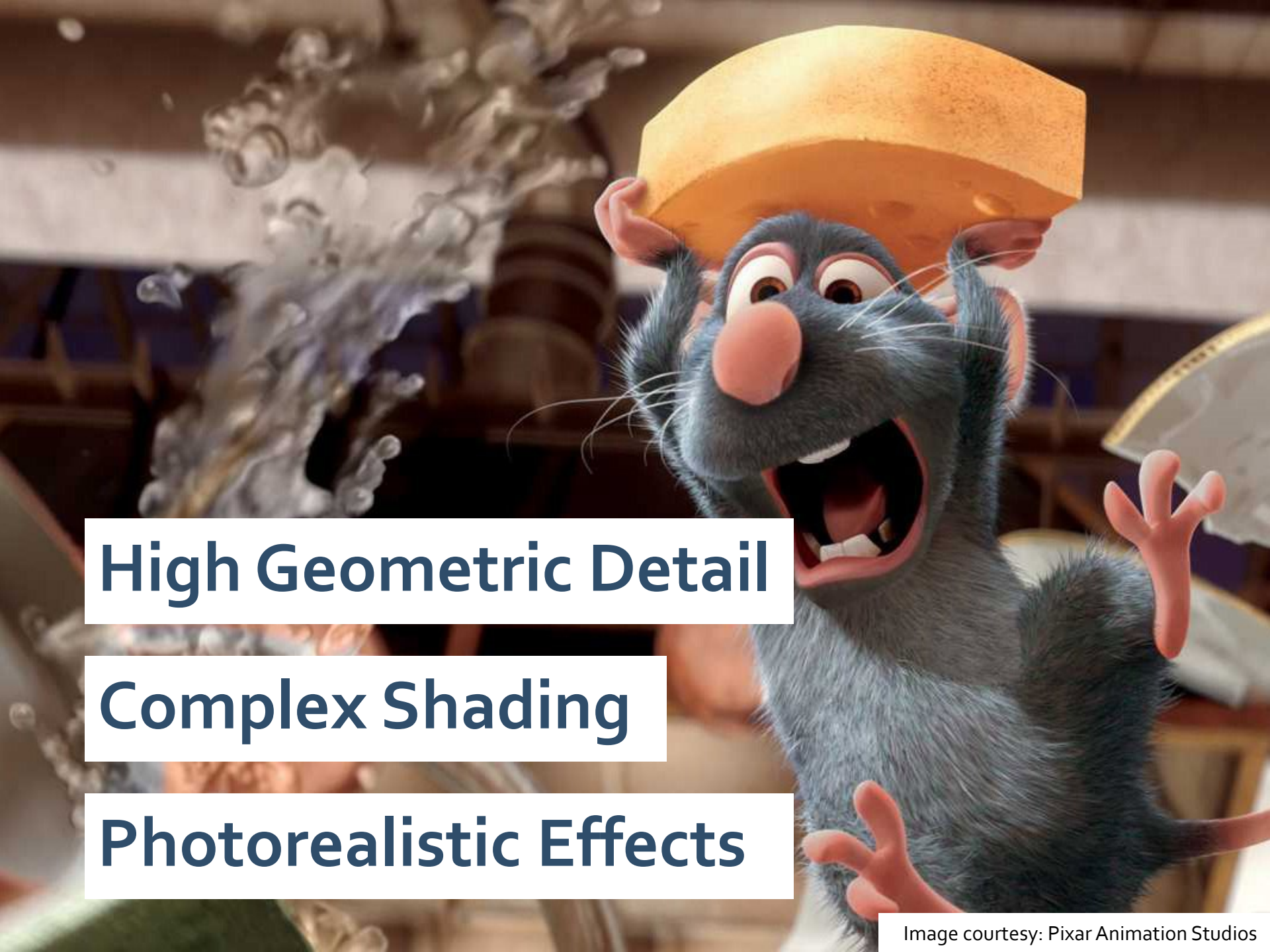
Overview

UCDAVIS
UNIVERSITY OF CALIFORNIA

- Introduction to Reyes
- Reyes IS the next big thing
- Reyes IS NOT the next big thing
- Conclusion

UCDAVIS
UNIVERSITY OF CALIFORNIA

Introduction to Reyes


High Geometric Detail
Complex Shading
Photorealistic Effects
Image courtesy: Pixar Animation Studios

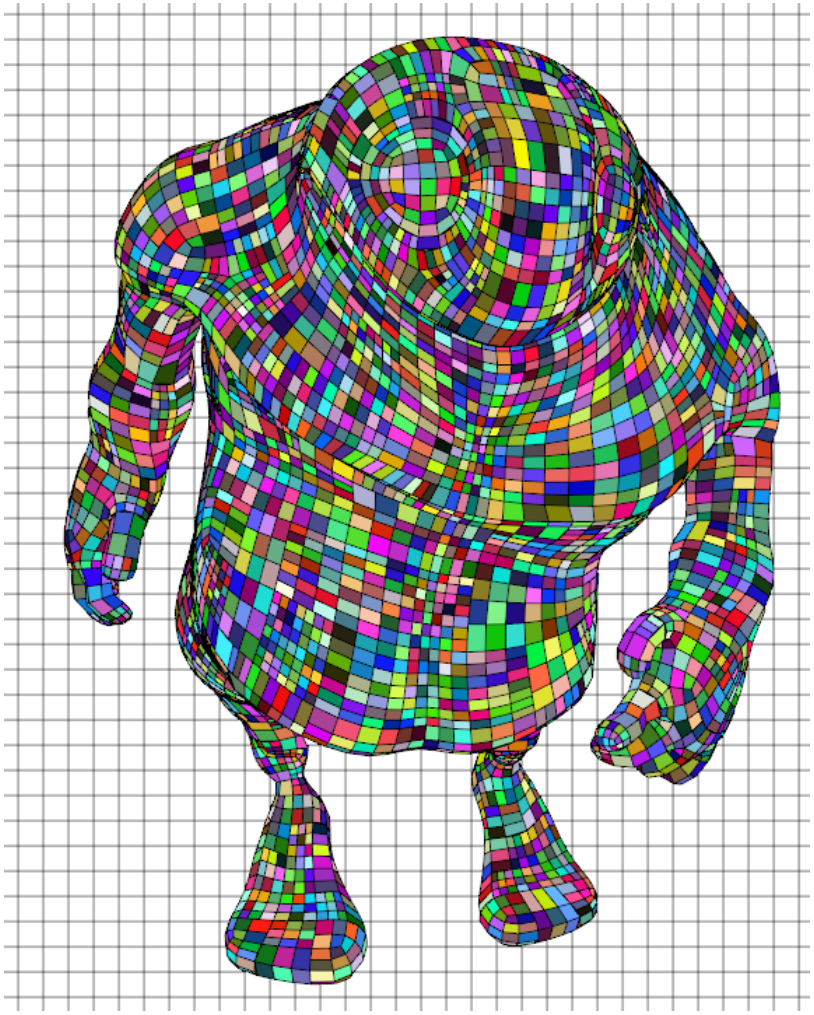
How to represent such high detail?

Image courtesy: Pixar Animation Studios

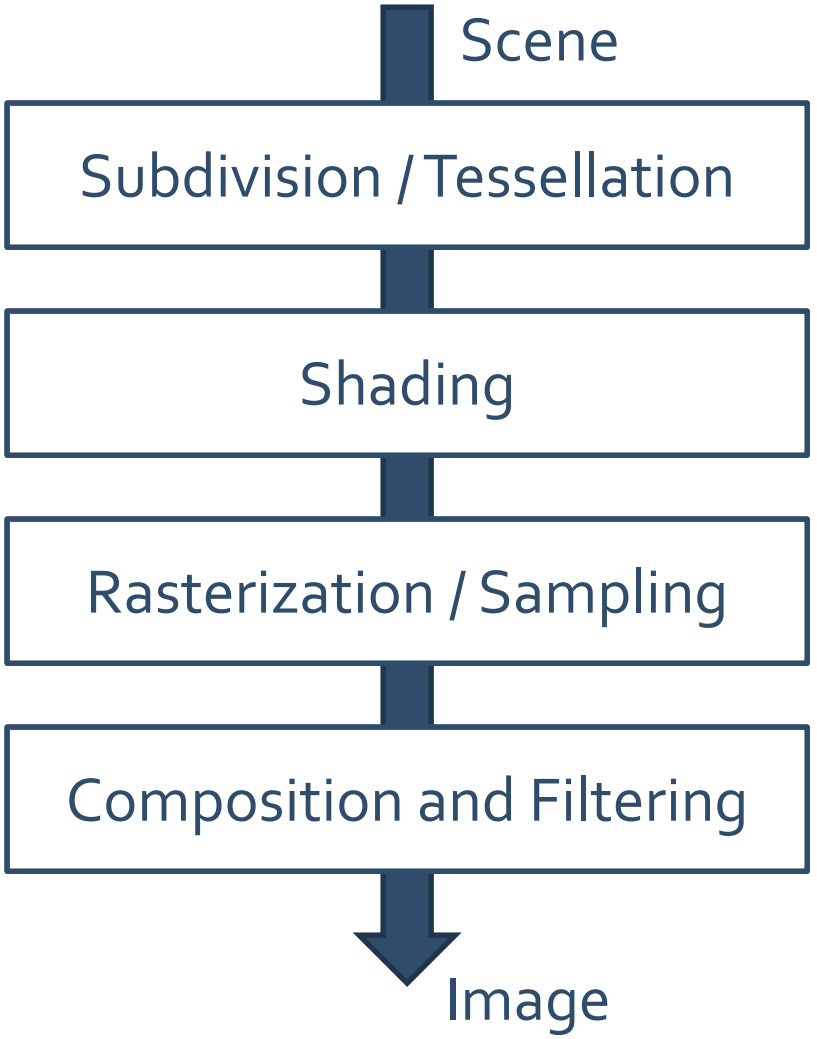
Micropolygons

UCDAVIS
UNIVERSITY OF CALIFORNIA

- 1x1 pixel (approx.) quads
  - Resolution-independent!
  - Defined in object space
- Allow detailed shading
  - Similar to fragments
- Fundamental units of Reyes Rendering



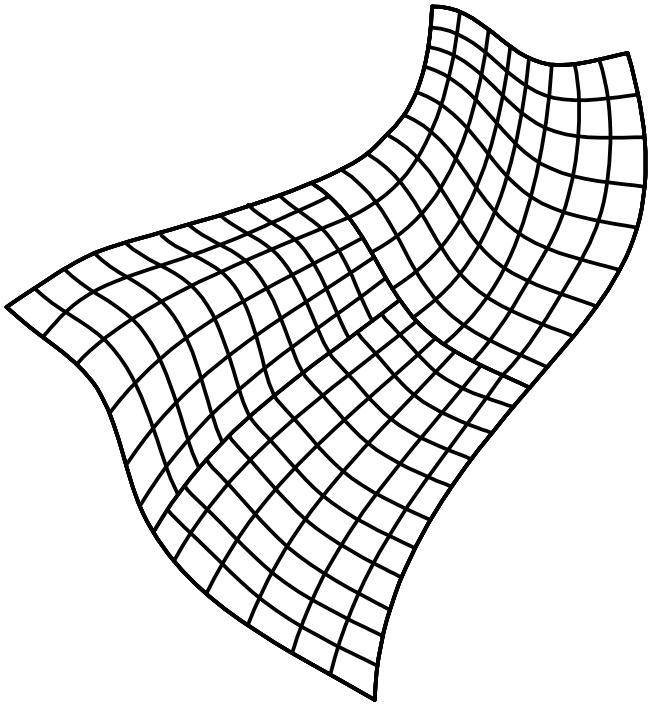
Pipeline Overview

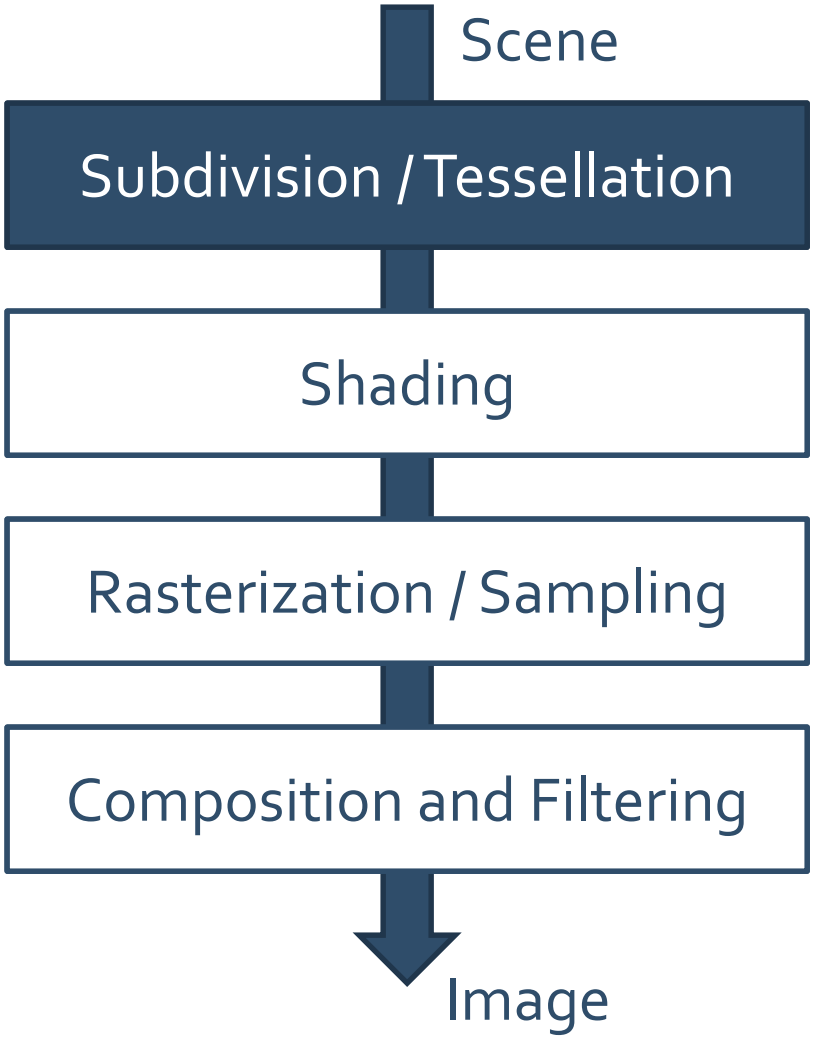


Pipeline Overview

UCDAVIS
UNIVERSITY OF CALIFORNIA

Obtain micropolygons from input surfaces

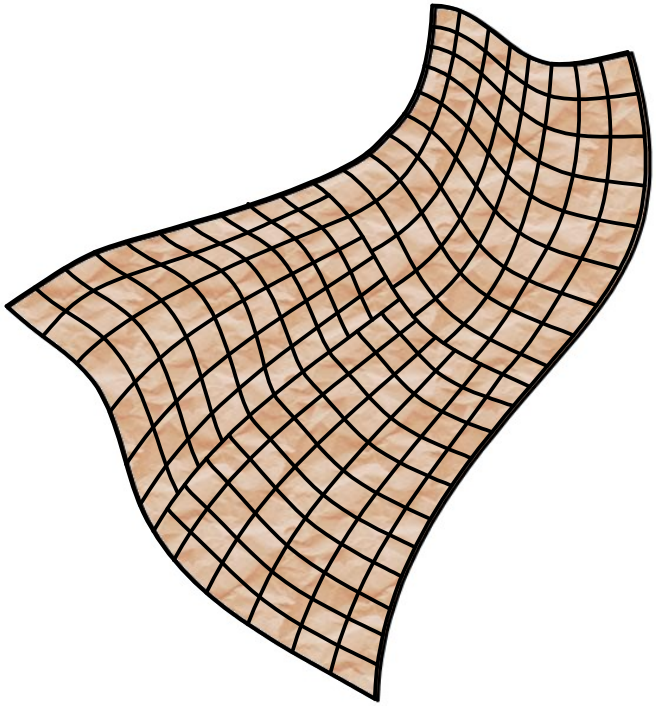


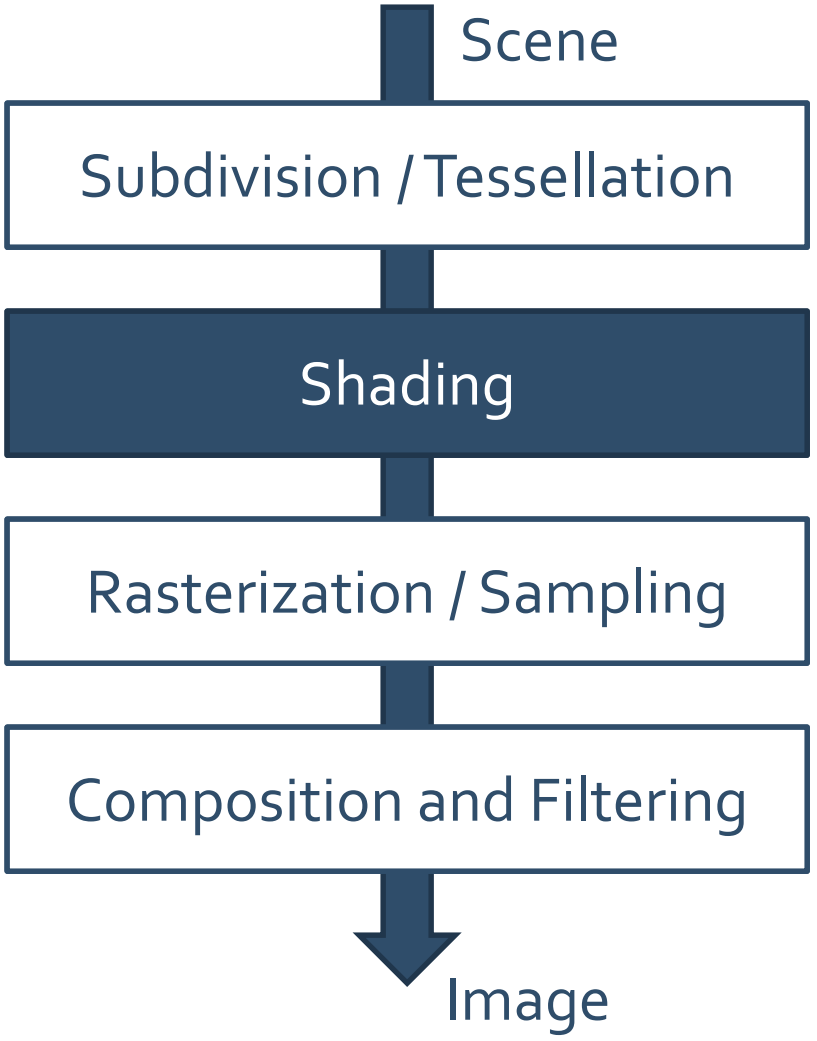


Pipeline Overview

UCDAVIS
UNIVERSITY OF CALIFORNIA

Shade micropolygons in world space

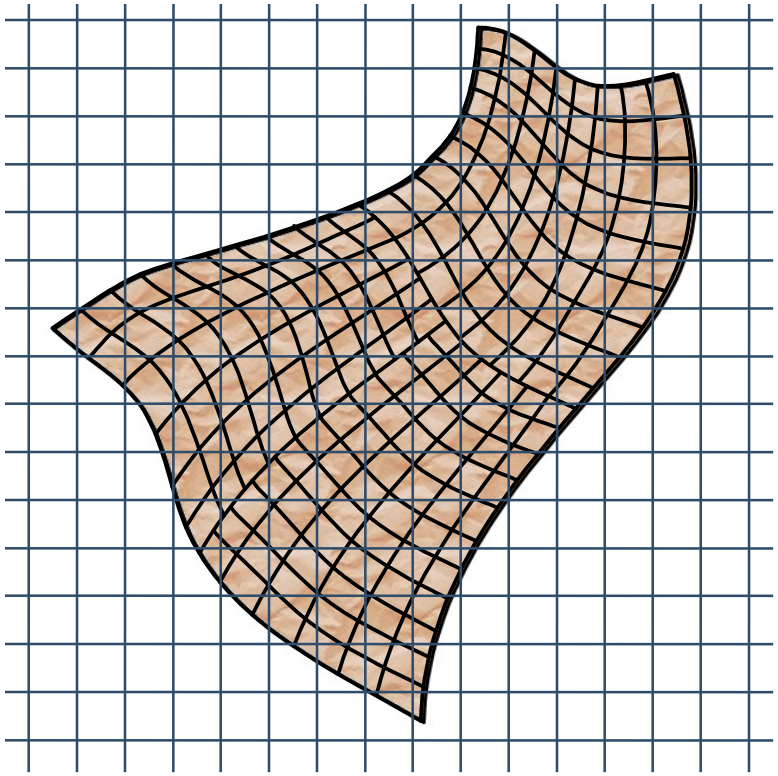


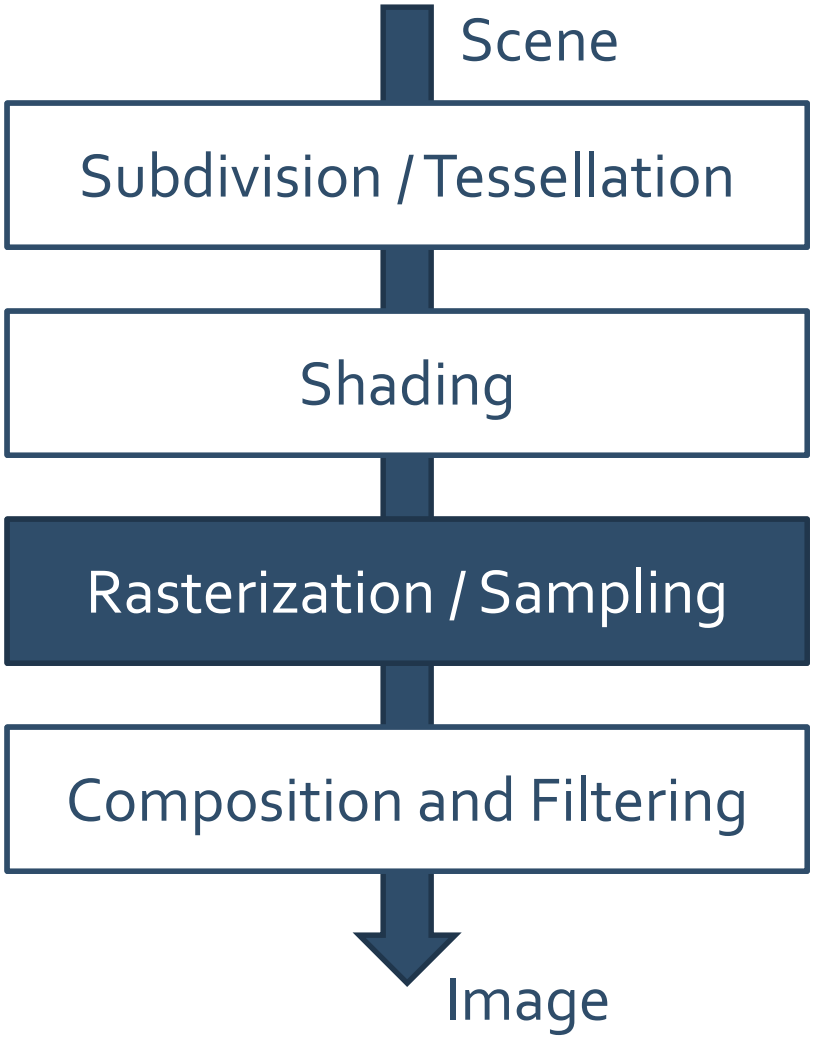


Pipeline Overview

UCDAVIS
UNIVERSITY OF CALIFORNIA

Map micropolygons to pixel samples

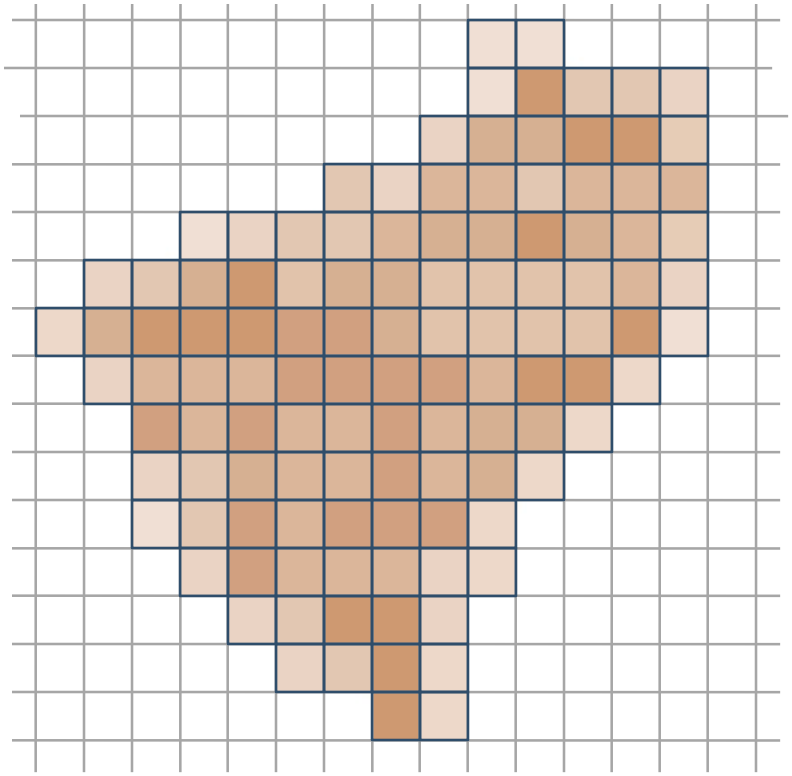


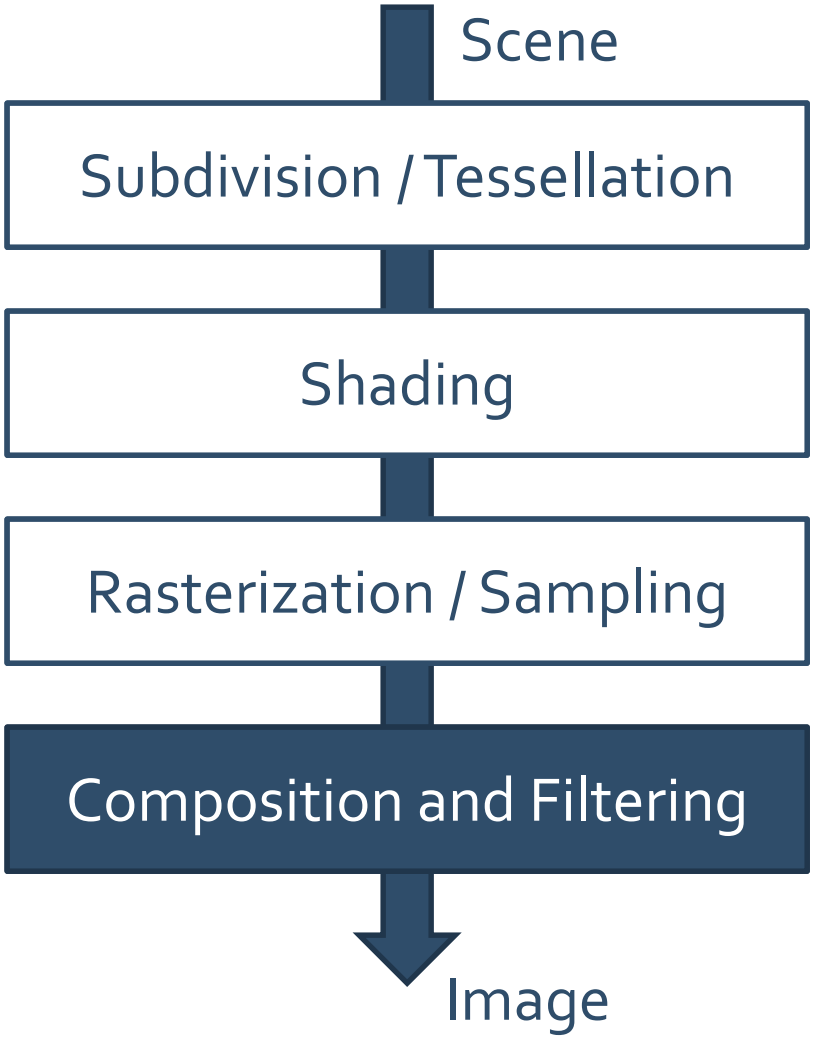


Pipeline Overview

UCDAVIS
UNIVERSITY OF CALIFORNIA

Reconstruct pixels from obtained samples





Why is Reyes better?

UCDAVIS
UNIVERSITY OF CALIFORNIA

- Smooth Surfaces
  - Pixel-level detail
  - Foliage, hair
- Stochastic Sampling
  - Anti-Aliasing
  - Motion Blur, Depth of Field
- Order-independent transparency

Image courtesy: www.renderpixie.com

UCDAVIS
UNIVERSITY OF CALIFORNIA

Reyes IS the next big thing

Real-Time Reyes

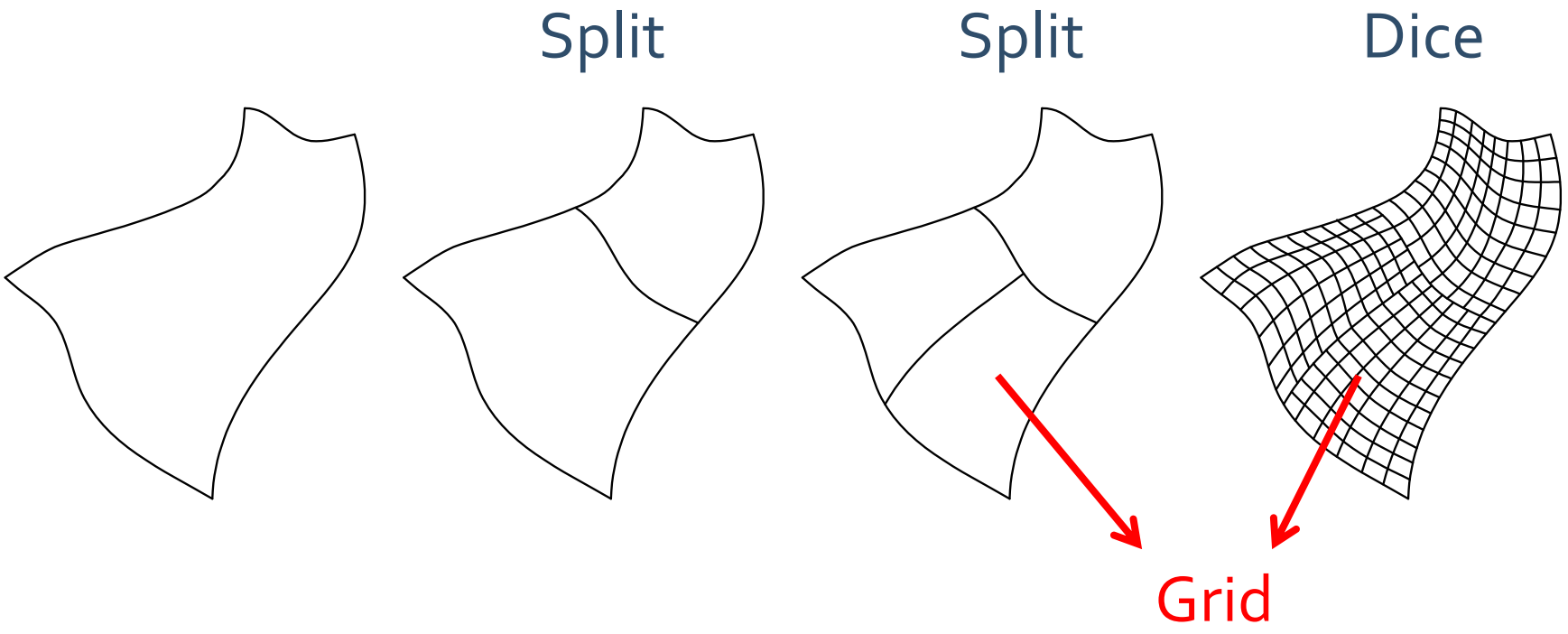
UCDAVIS
UNIVERSITY OF CALIFORNIA

- Reyes offers abundant parallelism
- GPUs offer a flexible interface
  - CUDA
  - OpenCL
  - DirectCompute
- What does the union look like?

Parallel Subdivision

UCDAVIS
UNIVERSITY OF CALIFORNIA

Reyes-Style Subdivision



Parallel Subdivision: Dice

UCDAVIS
UNIVERSITY OF CALIFORNIA

- Uniformly sample a parametric domain
- Easily parallelized
  - Map thread ID to (u,v)
  - Evaluate surface at (u,v)
- Direct3D 11 tessellation is similar to dicing

```
For each thread,

u = (tid.x / (blockSize.x));
v = (tid.y / (blockSize.y));

P = evaluateSurface (u, v);

Store P;
```

Parallel Subdivision: Split

UCDAVIS
UNIVERSITY OF CALIFORNIA

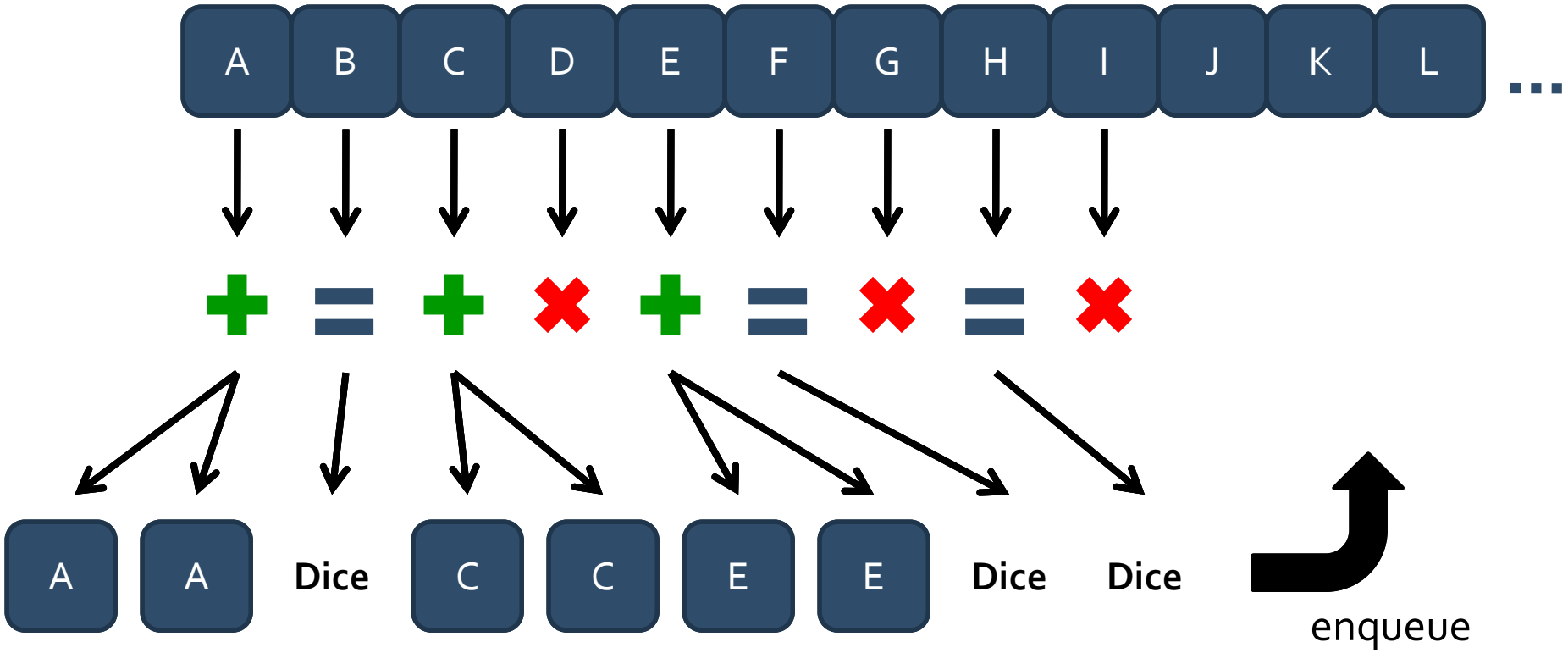
- Recursively subdivide a surface
  - Not easily parallelized
- Work-queue-based approach
  - [Patney and Owens 2008]
  - [Eisenacher et al. 2009]
- Fixed-function Split
  - [Fisher et al. 2009]

```
For each thread,

S = dequeue(splitQueue);

if(isSplit(S)){
   dir = splitdir(S);
    Snew[ ] = splitSurface(S,dir);
   enqueue(splitQueue, Snew[ ]);
} else {
   enqueue(diceQueue, S);
}
```

Parallel Subdivision: Split

UCDAVIS
UNIVERSITY OF CALIFORNIA

Work-queue-based Split

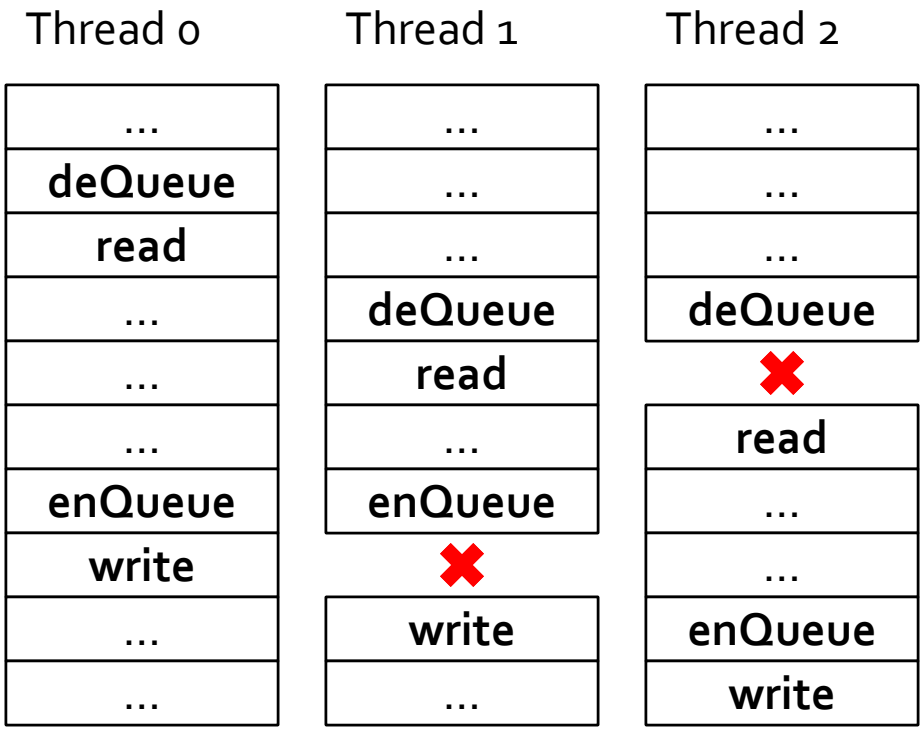


Parallel Subdivision: Split

UCDAVIS
UNIVERSITY OF CALIFORNIA

Parallel Queuing Techniques

- Atomic operations
- Implicit synchronization
- Slow on current generation GPUs

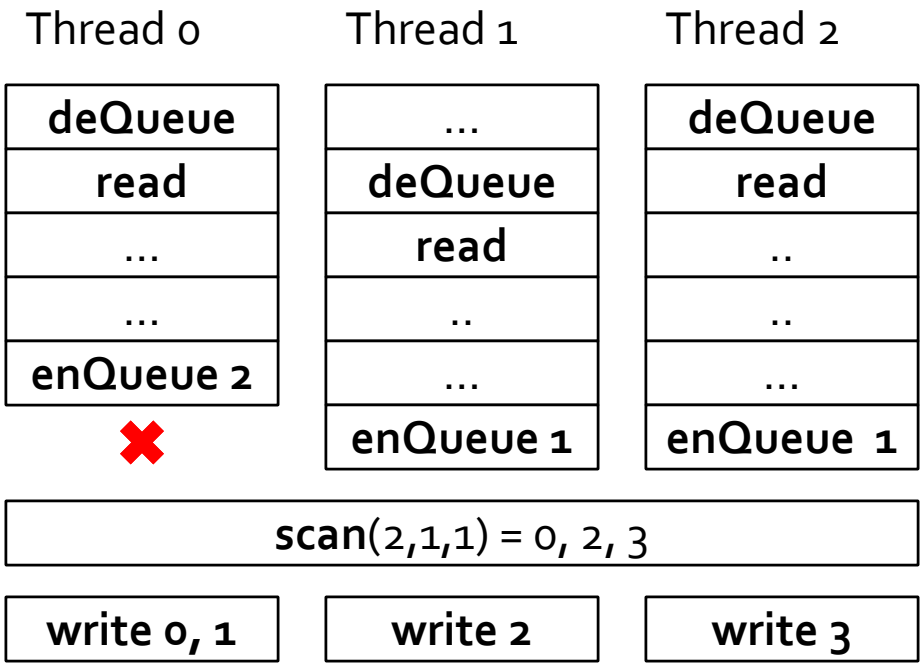


Parallel Subdivision: Split

UC DAVIS
UNIVERSITY OF CALIFORNIA

Parallel Queuing Techniques

- Scan-based updates
- Explicit synchronization
- Currently faster, but more involved



Parallel Subdivision

UCDAVIS
UNIVERSITY OF CALIFORNIA

Parametric Surfaces

- Simple representation
- Restricted flexibility and modeling ease
- 256M upolys/sec in CUDA
  [Patney et al. 2008]

Parallel Subdivision

UCDAVIS
UNIVERSITY OF CALIFORNIA

Subdivision Surfaces

- Complex, recursive definition
- Easier to model and animate
- 3M faces/sec in CUDA
  [Patney et al. 2009]

Parallel Subdivision

UCDAVIS
UNIVERSITY OF CALIFORNIA

Approximate Subdivision Surfaces

- Allow treating subdivision surfaces as parametric
- Modeled as subdivision surfaces
- 256M upolys/sec in CUDA!

Parallel Shading

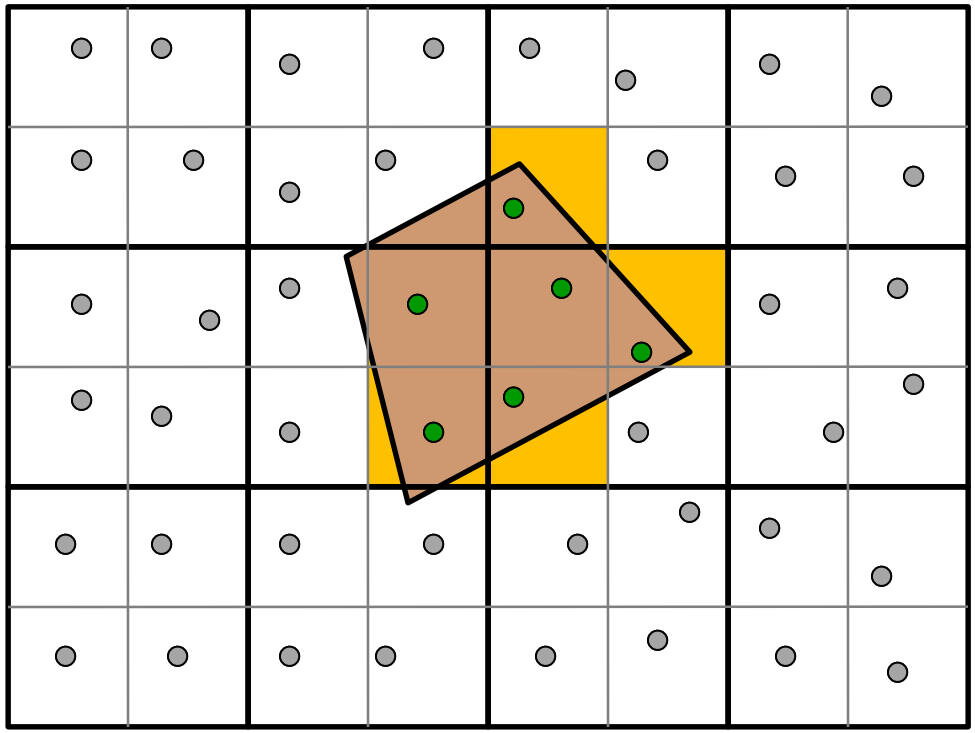
UCDAVIS
UNIVERSITY OF CALIFORNIA

- Execute shader at grid vertices
  - Data-Parallel
  - Primitive-level locality
- Vectorized shading at grid level
  - Highly SIMD-friendly
  - Easy access to derivatives
- Texturing similar to a polygon pipeline
  - Access is more coherent

Parallel Sampling

UCDAVIS
UNIVERSITY OF CALIFORNIA

Intersect micropolygons with jittered pixel samples

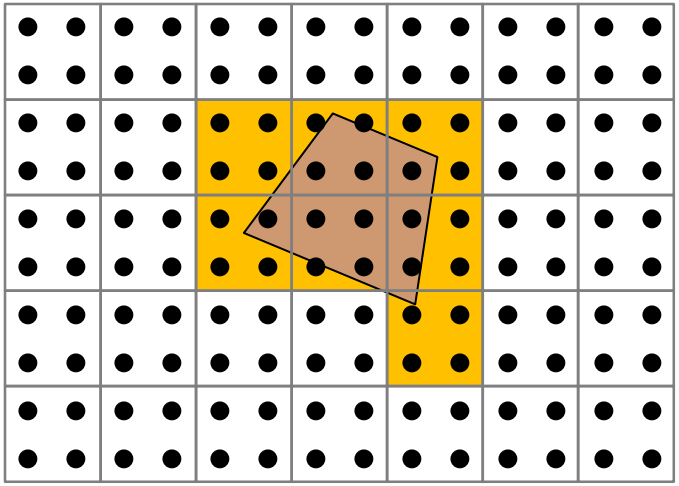


Parallel Sampling

UCDAVIS
UNIVERSITY OF CALIFORNIA

Example Approach 1 [Zhou et al. 2009]

- For each micropolygon,
  - Bound the number of samples



- Allocate space for micropolygons

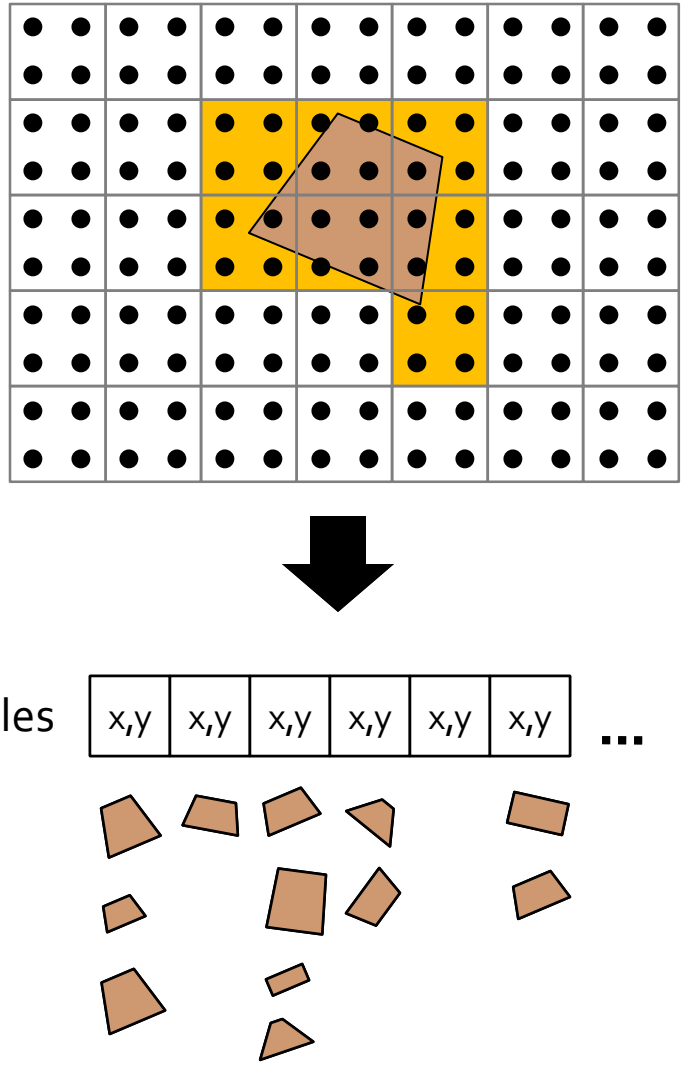
- For each micropolygon,
  - Test with possible samples
  - Store samples

Parallel Sampling

UCDAVIS
UNIVERSITY OF CALIFORNIA

Example Approach 2

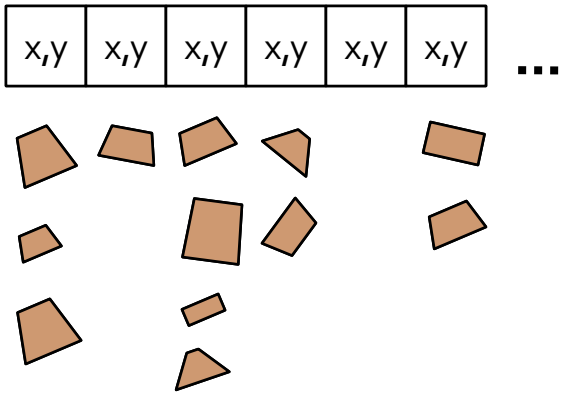
- For each micropolygon,
  - Estimate coarse bound
  - Append to a global queue
- Sort global queue
- For each sample,
  - Test with possible micropolygons
  - Also blend samples



Parallel Composite and Filter

UCDAVIS
UNIVERSITY OF CALIFORNIA

- Sort all samples by depth
- For each subpixel,
  - Blend samples front-to-back
- For each pixel
  - Blend colors and opacity of samples



UCDAVIS
UNIVERSITY OF CALIFORNIA

Reyes IS NOT the next big thing

Delayed Visibility

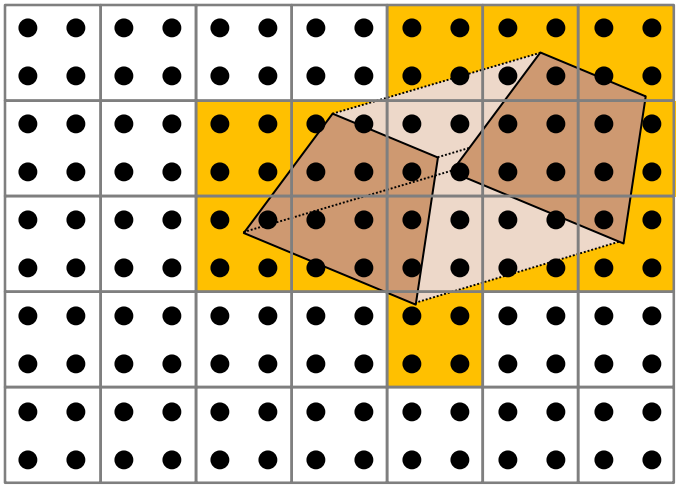
UCDAVIS
UNIVERSITY OF CALIFORNIA

- Tessellation and Shading are data-parallel
  - But a lot of micropolygons will get rejected
  - Overtessellation, overshading
  - Cost proportional to (depth complexity) x (resolution)
- Require efficient occlusion culling
- Application-level culling is important

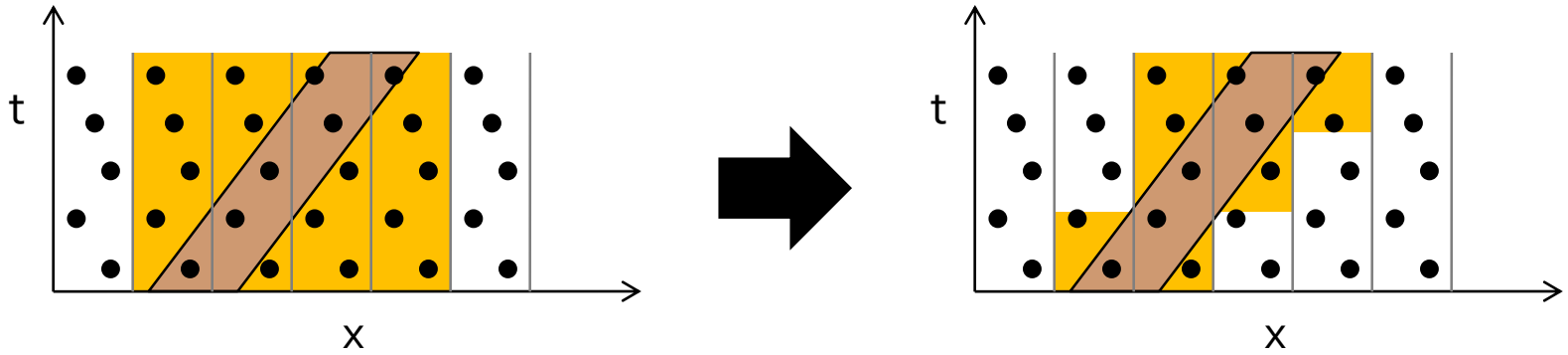
Inefficient Sampling

UCDAVIS
UNIVERSITY OF CALIFORNIA

Motion-Blur, Depth-of-Field



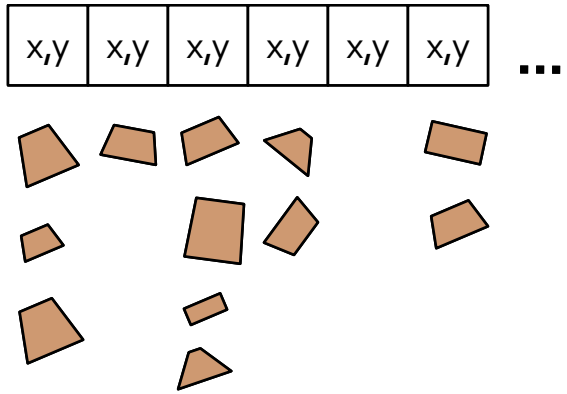
- Bounds become loose
  - Low efficiency (by up to 10 times!)
- **Alternative**
  - Distribution across time [Fatahalian et al. 2009]



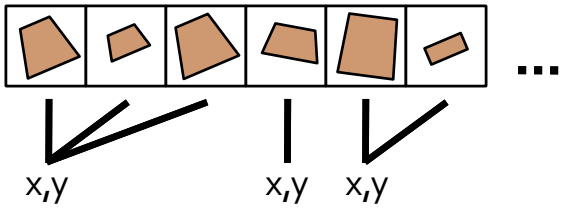
Inefficient Compositing

UCDAVIS
UNIVERSITY OF CALIFORNIA

- Varying number of samples per subpixel



- **Alternative**
  - Parallelize compositing across individual samples
  - "Segmented reduction"



Mixing Pipelines

UCDAVIS
UNIVERSITY OF CALIFORNIA

- Sort-middle rendering
  - Compute sub-patches and assign to tiles
  - Render each tile separately
  - [Loop et al. 2009]
- Shading after visibility
  - Generate fragments from grids
- Allowing both macro and micro polygons?

Architectural Enablers

UCDAVIS
UNIVERSITY OF CALIFORNIA

- Caching
  - Potential benefit to shading performance
- Fast atomic operations
  - Convenient queuing
  - Scatter accumulation
- Fixed-function additions
  - Micropolygon rasterizer
  - Splitter

Summary

UCDAVIS
UNIVERSITY OF CALIFORNIA

Real-time Reyes-style rendering

- Tessellation, Shading
  - ✓ Well-expressed in parallel
  - ✗ A lot of work goes to waste
- Sampling, Composition
  - ✗ Can be inefficient for motion-blur, depth-of-field
  - ✗ Work inefficiency due to irregular workload

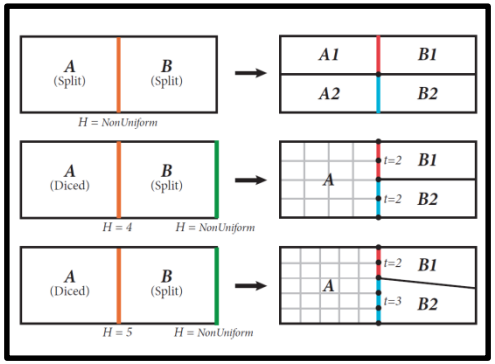
Conclusion

UCDAVIS
UNIVERSITY OF CALIFORNIA

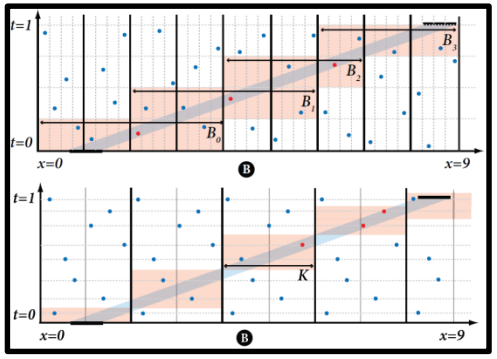
- Increasing detail in real-time graphics
  - Scene complexity will continue to increase
- Reyes-style rendering will soon be feasible
  - Faster graphics hardware
  - Research in Parallel rendering algorithms
- A lot of scope for hybrid rendering
- Ongoing architectural evolution

Further Reading

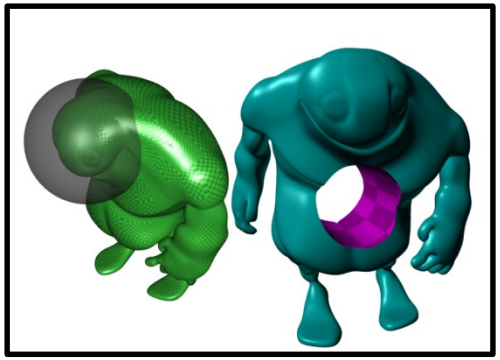
Real-time Reyes-Style Surface Subdivision
Patney et al. 2008



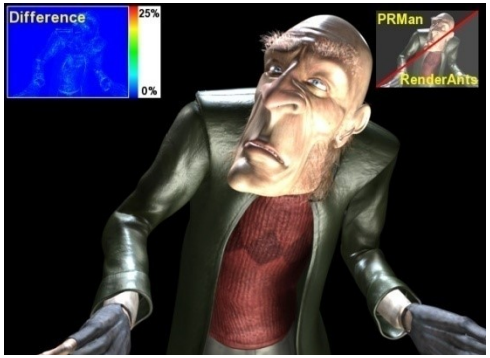
DiagSplit
Fisher et al. 2009



Data-Parallel Rasterization of Micropolygons
Fatahalian et al. 2009



Real-Time Sort-Middle Rendering
Loop et al. 2009



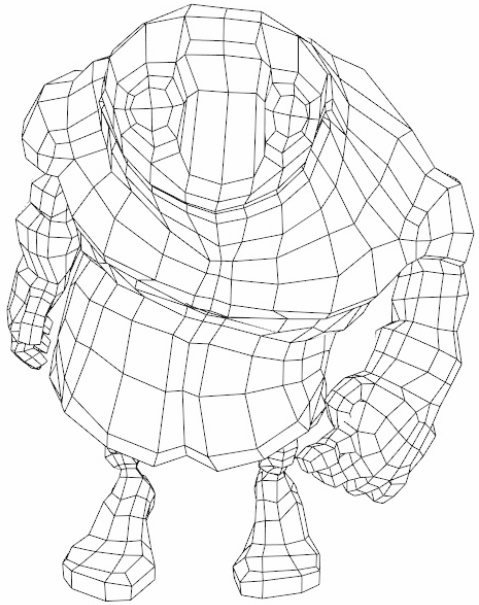
RenderAnts
Zhou et al. 2009

Acknowledgments

- John D. Owens
- Stanley Tzeng
- NVIDIA Research Fellowship
- SciDAC Institute for Ultrascale Visualization

Thanks!

UCDAVIS
UNIVERSITY OF CALIFORNIA



UCDAVIS
UNIVERSITY OF CALIFORNIA

Thanks!

apatney@ucdavis.edu