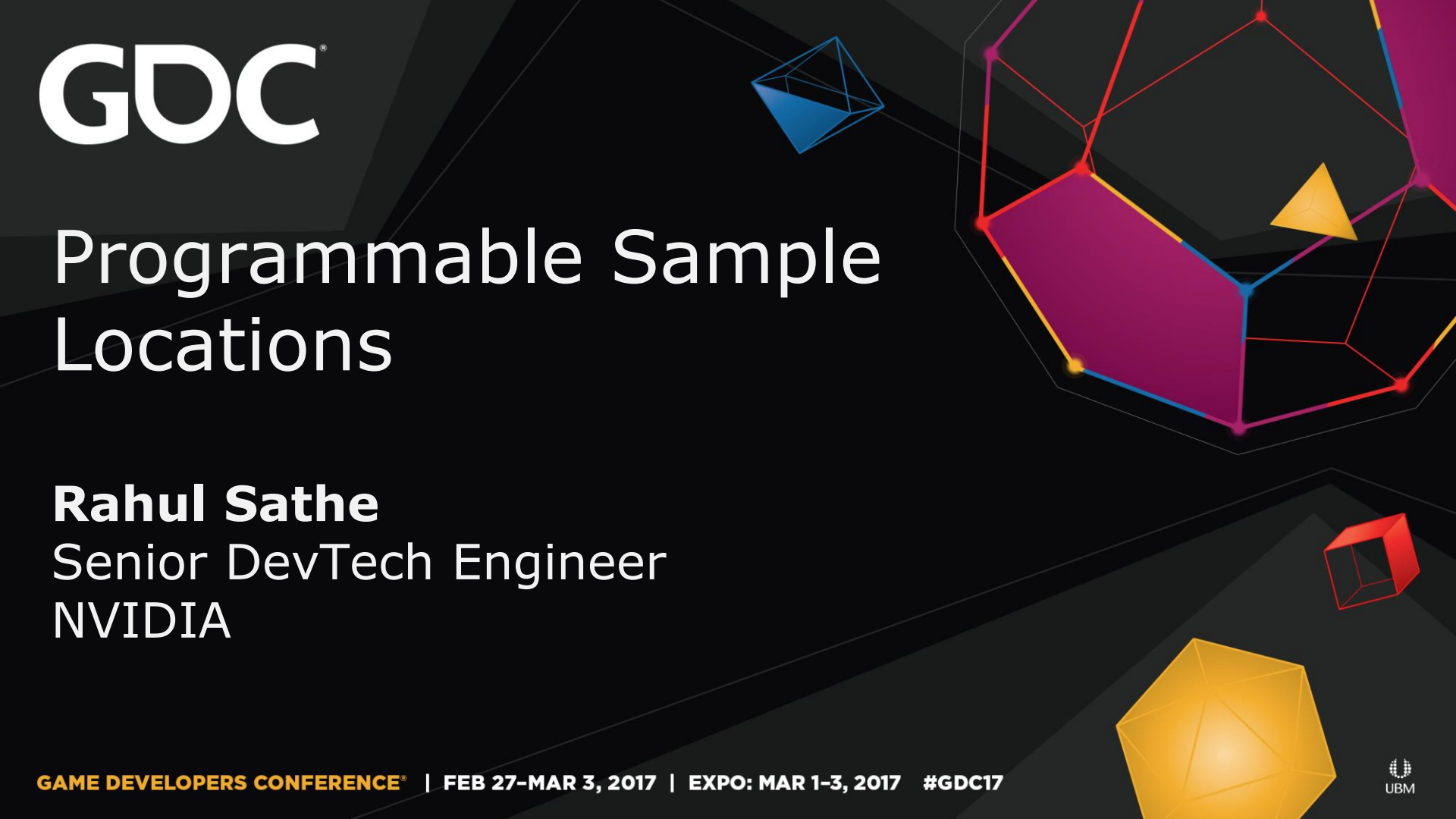
Conservative Rasterization and Raster Order Views

Rahul Sathe, Evgeny Makarov
Senior DevTech Engineer
NVIDIA



Programmable Sample Locations

Rahul Sathe
Senior DevTech Engineer
NVIDIA

Agenda

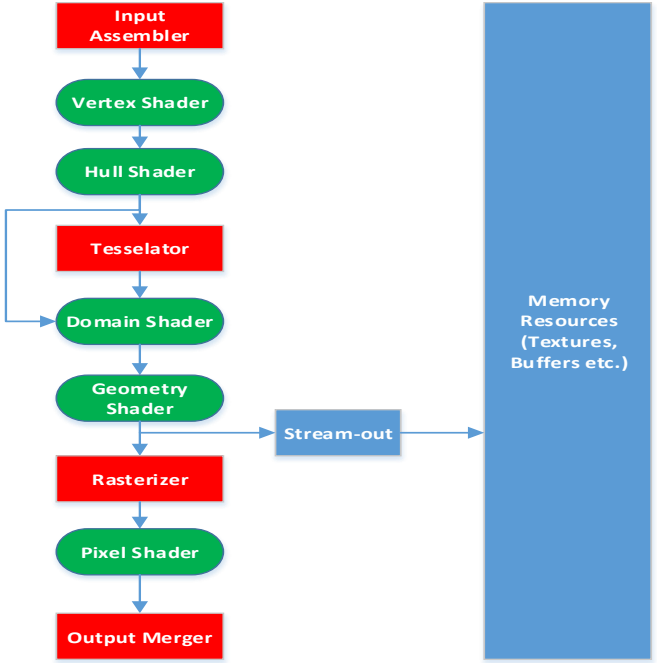
- Motivation: Programmable Sample Locations
- Rasterization Basics
- Conservative Rasterization
- Algorithm
  - Pull mode interpolation
- Clipper Issues and the work-arounds
- Temporal super sampling/TAA
- Future

Motivation

- Samples layout: Uniform Grid
  - Aliasing : Geometric, shader and texture
- Temporal super-sampling
  - Desired feature to tackle flickering
- Ray tracing requires richer sample patterns
  - Halton (2, 3), 0-2, Sobol Sequence etc.

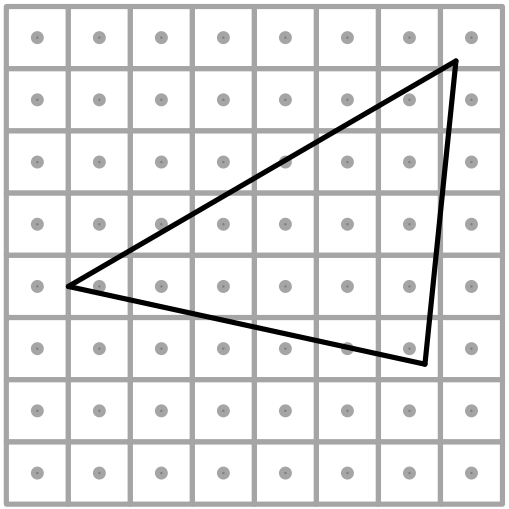
Rasterization Basics

- Rasterizer
  - Fixed Function
- Rasterization:
  - Is P(x,y) inside

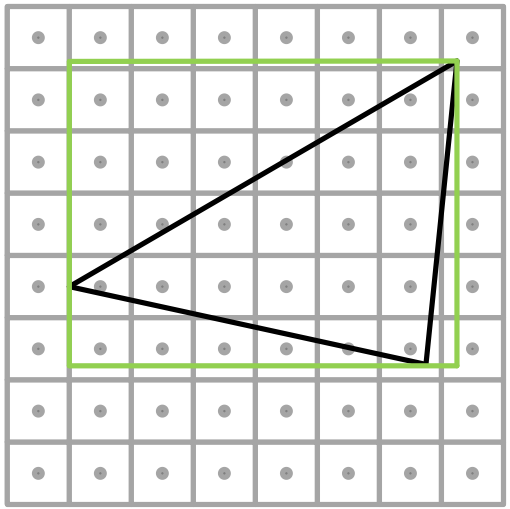


Rasterization Basics

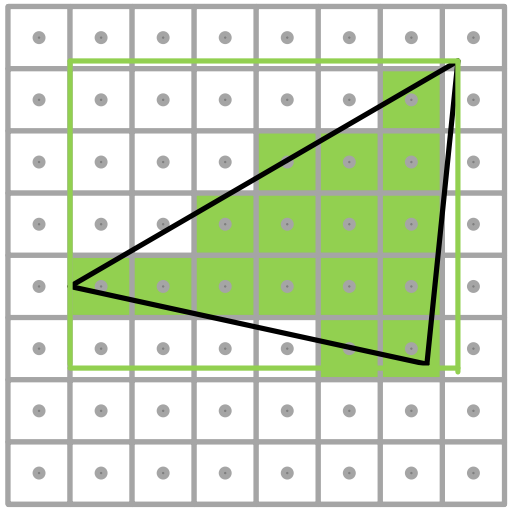
Rasterization Basics



Rasterization Basics

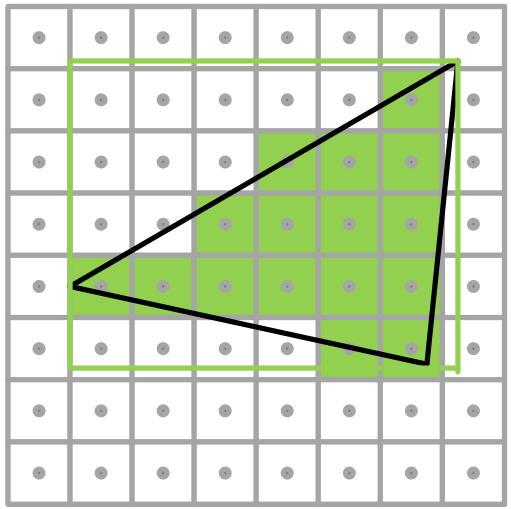


Rasterization Basics



- Edge Equations in Screen Space
  - Ax+By+C < 0 : Inside
  - Ax+By+C > 0 : Outside
  - Ax+By+C = 0 : On the edge
- Top-left rule when on the edge
- Hierarchical rasterization

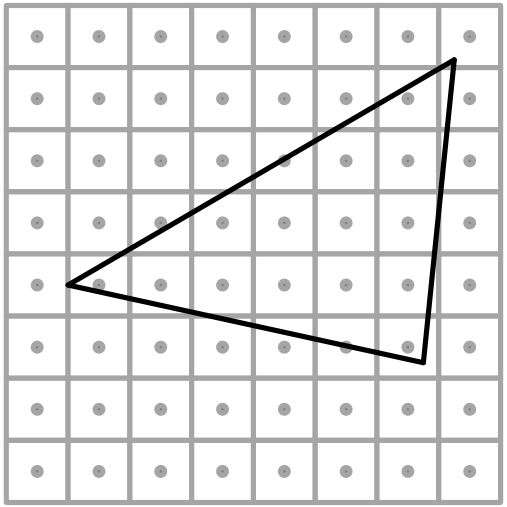
Rasterization Basics



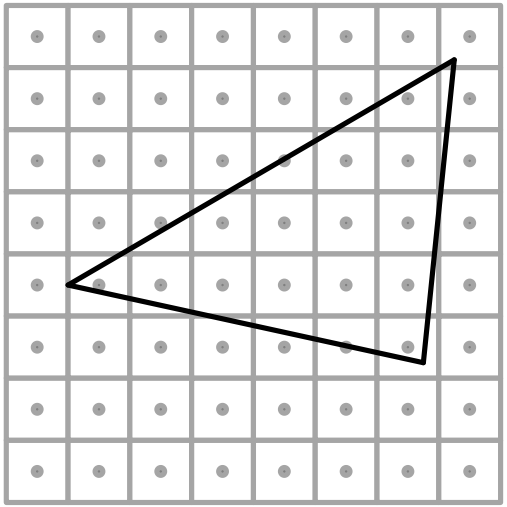
- Edge Equations in Screen Space
  - Ax+By+C < 0 : Inside
  - Ax+By+C > 0 : Outside
  - Ax+By+C = 0 : On the edge
- Top-left rule when on the edge
- Hierarchical rasterization

Conservative Rasterization

Conservative Rasterization

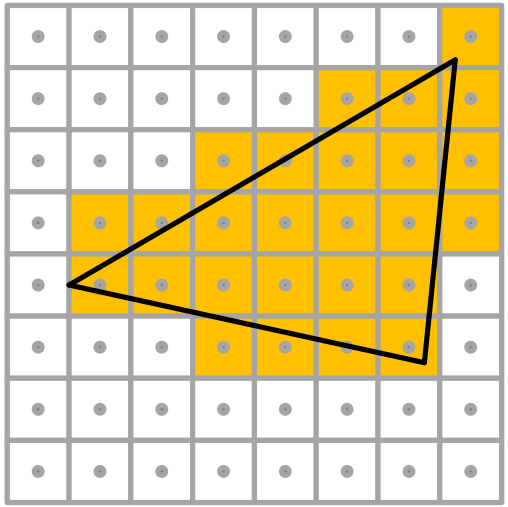


Conservative Rasterization



- Rasterizes pixels if their extents overlap the primitive
- Feature Level 12_1
- APIs : D3D12 and D3D11.3
- Tier 3 : SV_InnerCoverage

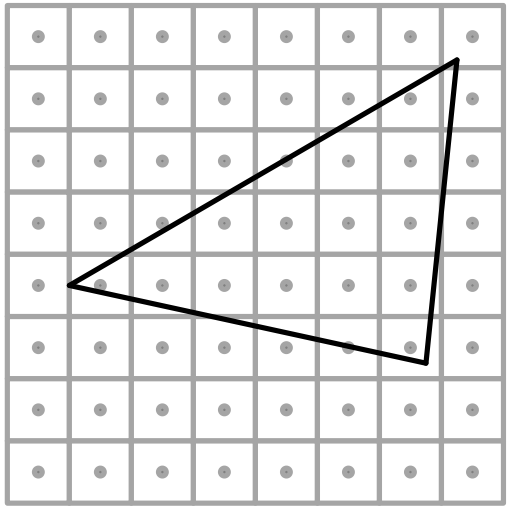
Conservative Rasterization



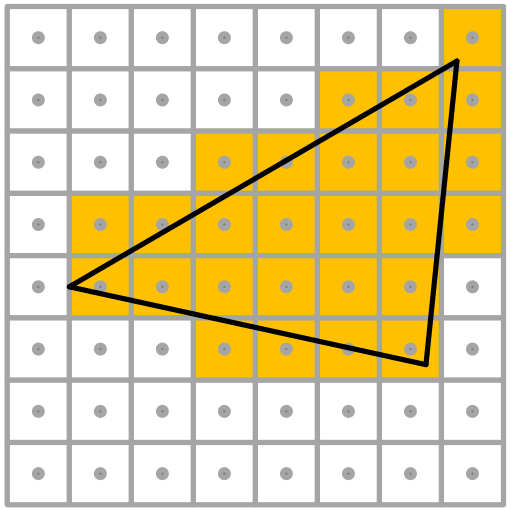
- Rasterizes pixels if their extents overlap the primitive
- Feature Level 12_1
- APIs : D3D12 and D3D11.3
- Tier 3 : SV_InnerCoverage

Algorithm

Algorithm

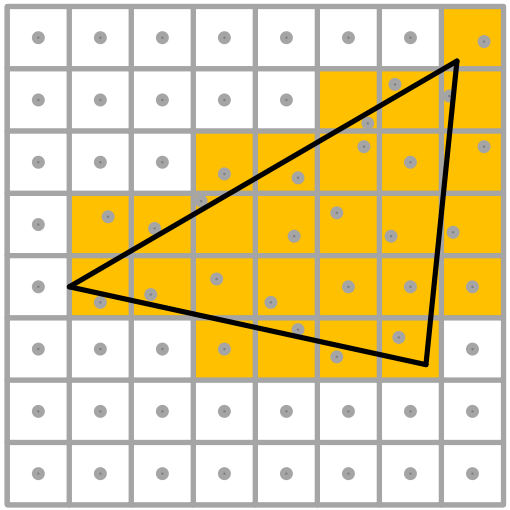


Algorithm



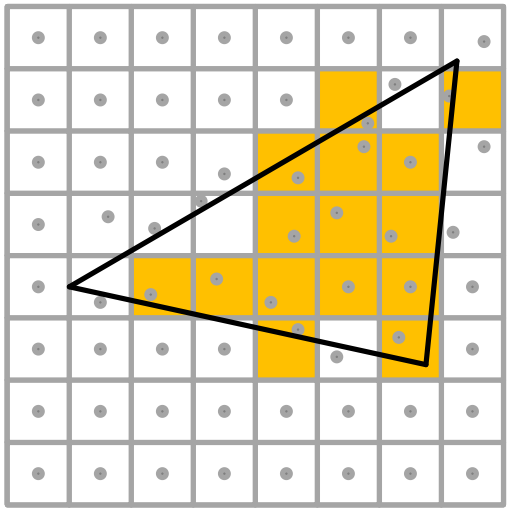
- GS
  - edge equations
- Conservative Rasterizer
- PS
  - Random offsets
  - If outside discard
  - If inside interpolate
  - Output the depth

Algorithm



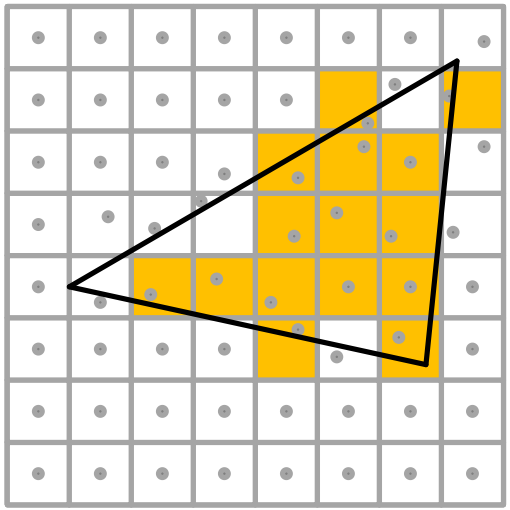
- GS
  - edge equations
- Conservative Rasterizer
- PS
  - Random offsets
  - If outside discard
  - If inside interpolate
  - Output the depth

Algorithm



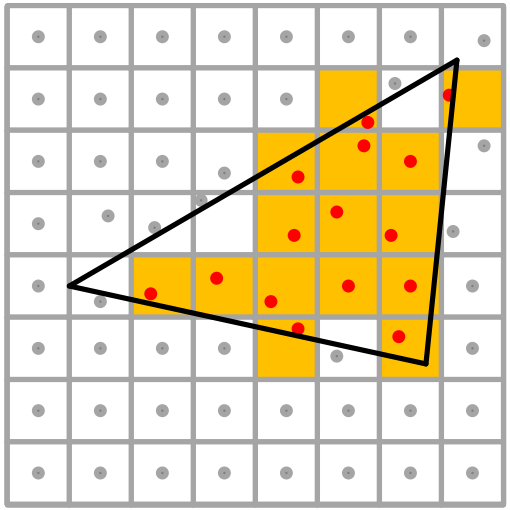
- GS
  - edge equations
- Conservative Rasterizer
- PS
  - Random offsets
  - If outside discard
  - If inside interpolate
  - Output the depth

Algorithm



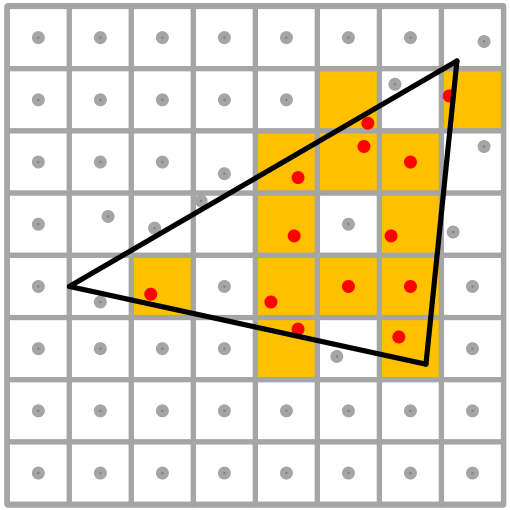
- GS
  - edge equations
- Conservative Rasterizer
- PS
  - Random offsets
  - If outside discard
  - If inside interpolate
  - Output the depth

Algorithm



- GS
  - edge equations
- Conservative Rasterizer
- PS
  - Random offsets
  - If outside discard
  - If inside interpolate
  - Output the depth

Algorithm



- GS
  - edge equations
- Conservative Rasterizer
- PS
  - Random offsets
  - If outside discard
  - If inside interpolate
  - Output the depth

Pull mode interpolation

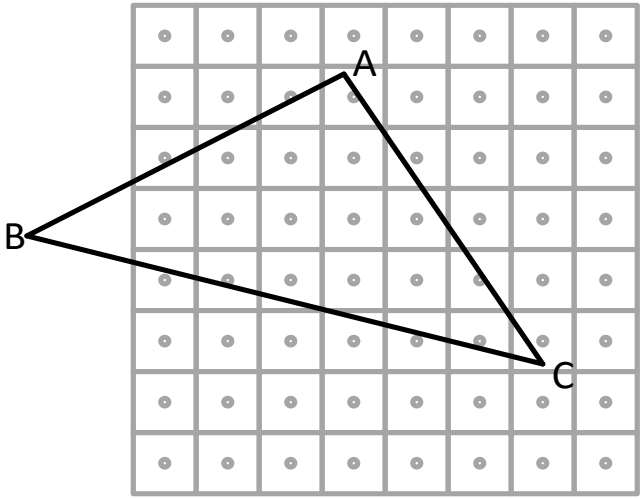
- Interpolation is done in the shader
  - EvaluateAttributeAtCentroid
  - EvaluateAttributeAtSample
  - EvaluateAttributeSnapped
    - 16x16 possible discrete offsets

Other Interesting Details

- SV_Depth output forces the late Z/stencil
- Consistent offsets from a given viewpoint
- SampleCount > 1
  - PS should generate per sample output depth
  - Pass SV_SampleIndex as input to the PS

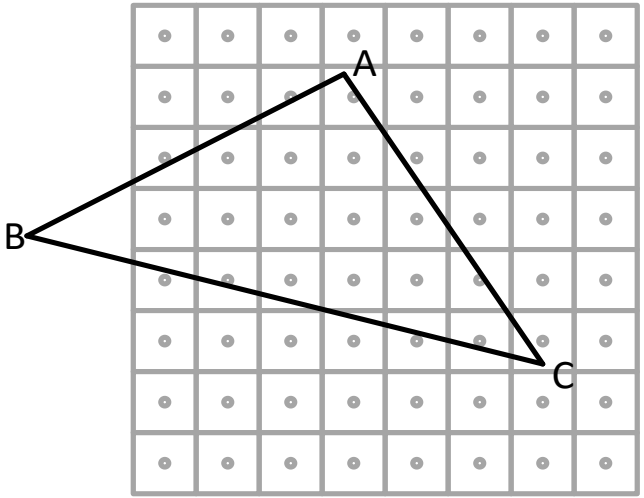
Clipper

Clipper



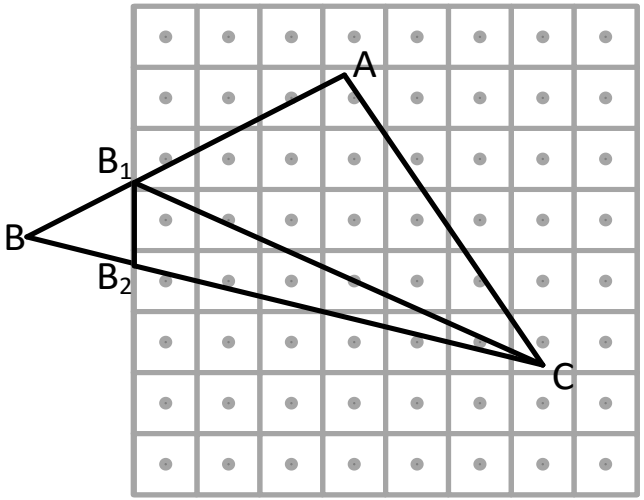
- Clips large triangles
- 1.f – (1.f – t) ≠ t
- Must use fixed point math
- But the GS sent the original triangle’s edge equations
- Pixels along shared edges get shaded multiple times

Clipper



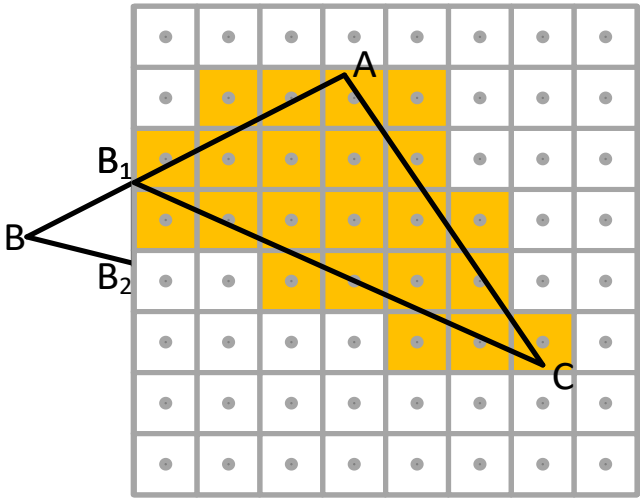
- Clips large triangles
- 1.f – (1.f – t) ≠ t
- Must use fixed point math
- But the GS sent the original triangle’s edge equations
- Pixels along shared edges get shaded multiple times

Clipper



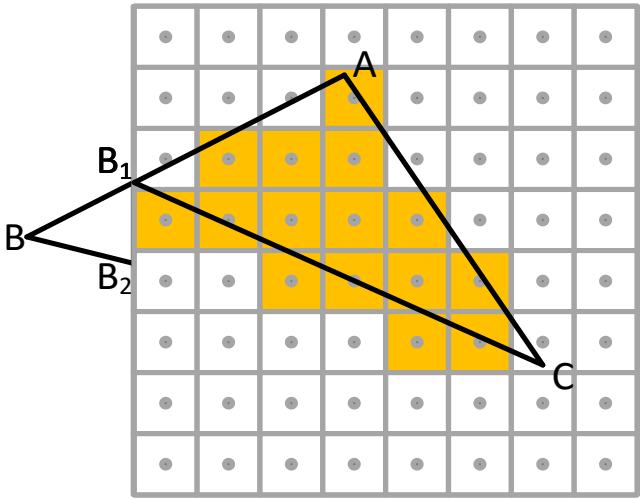
- Clips large triangles
- 1.f – (1.f – t) ≠ t
- Must use fixed point math
- But the GS sent the original triangle's edge equations
- Pixels along shared edges get shaded multiple times

Clipper



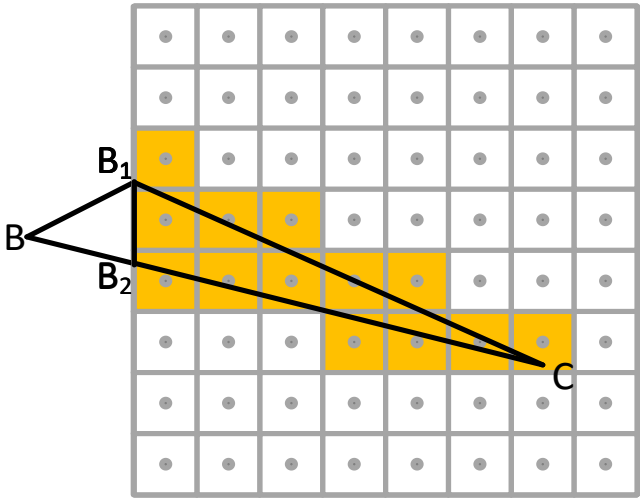
- Clips large triangles
- 1.f – (1.f – t) ≠ t
- Must use fixed point math
- But the GS sent the original triangle’s edge equations
- Pixels along shared edges get shaded multiple times

Clipper



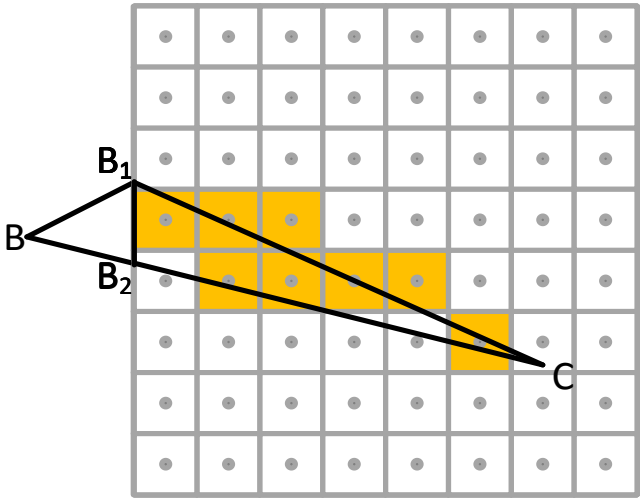
- Clips large triangles
- 1.f – (1.f – t) ≠ t
- Must use fixed point math
- But the GS sent the original triangle’s edge equations
- Pixels along shared edges get shaded multiple times

Clipper



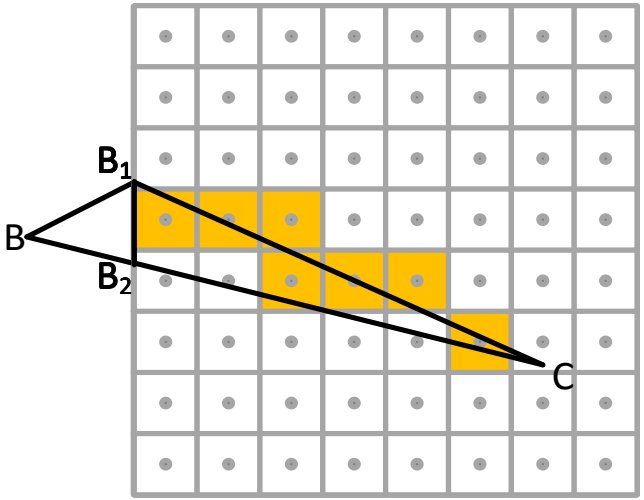
- Clips large triangles
- 1.f – (1.f – t) ≠ t
- Must use fixed point math
- But the GS sent the original triangle's edge equations
- Pixels along shared edges get shaded multiple times

Clipper



- Clips large triangles
- 1.f – (1.f – t) ≠ t
- Must use fixed point math
- But the GS sent the original triangle's edge equations
- Pixels along shared edges get shaded multiple times

Clipper



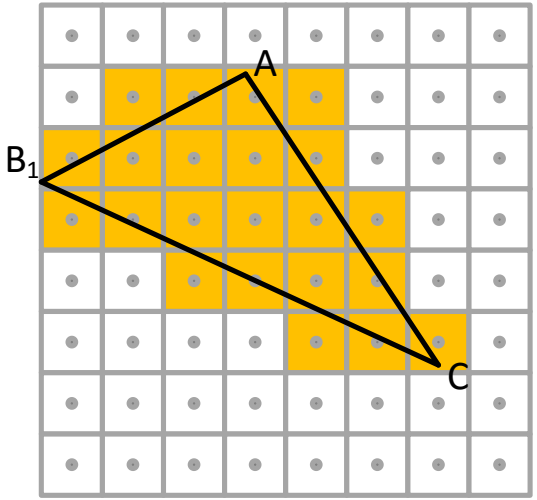
- Clips large triangles
- 1.f – (1.f – t) ≠ t
- Must use fixed point math
- But the GS sent the original triangle's edge equations
- Pixels along shared edges get shaded multiple times

Raster Order Views

Raster Order Views

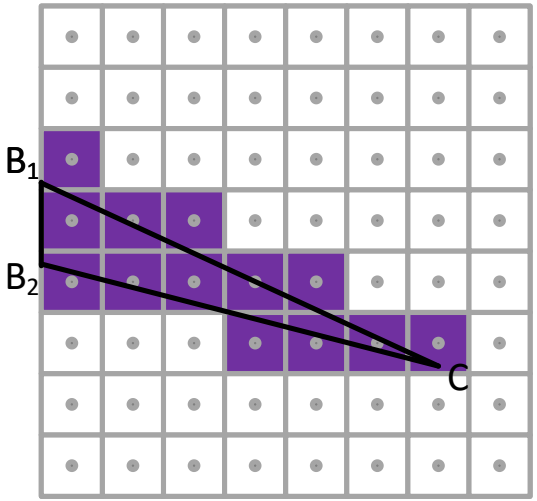
- SM5.1 with D3D11.3
- Similar to UAVs, but
- Impose API ordering

Raster Order Views



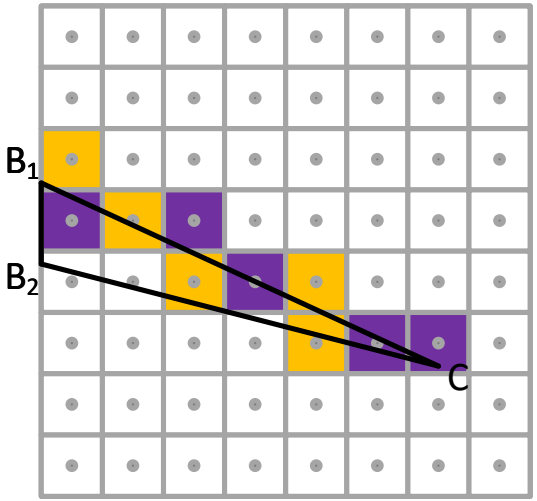
- SM5.1 with D3D11.3
- Similar to UAVs, but
- Impose API ordering

Raster Order Views



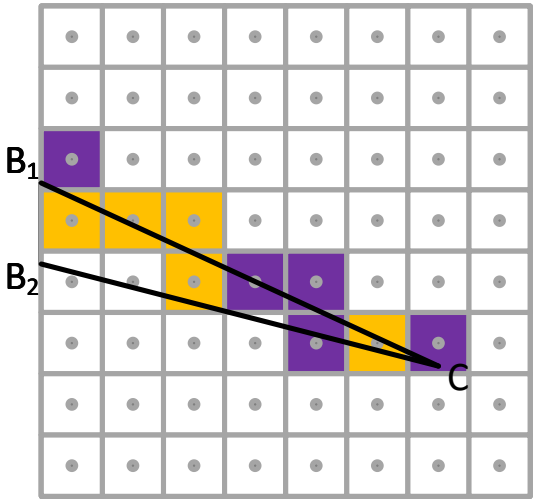
- SM5.1 with D3D11.3
- Similar to UAVs, but
- Impose API ordering

Raster Order Views



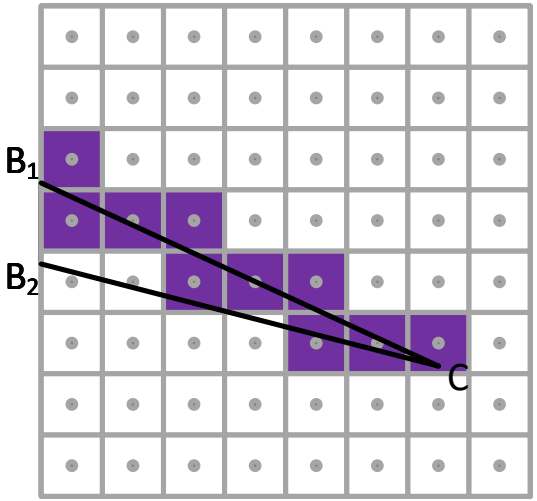
- SM5.1 with D3D11.3
- Similar to UAVs, but
- Impose API ordering

Raster Order Views



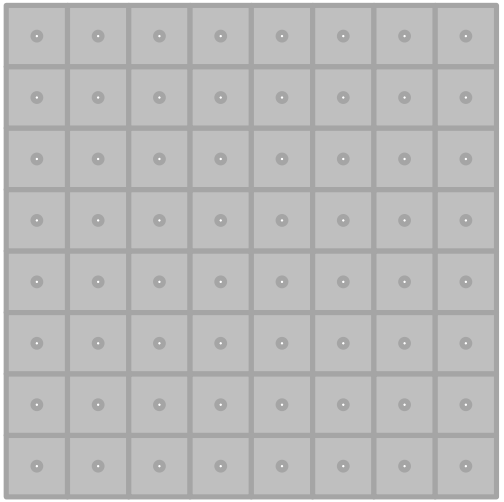
- SM5.1 with D3D11.3
- Similar to UAVs, but
- Impose API ordering

Raster Order Views

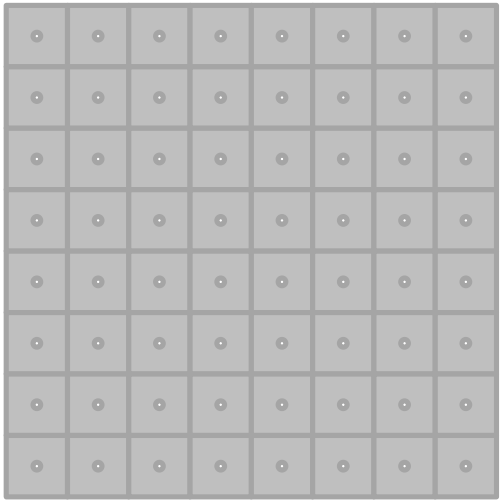


- SM5.1 with D3D11.3
- Similar to UAVs, but
- Impose API ordering

Handling clipped triangles



Handling clipped triangles

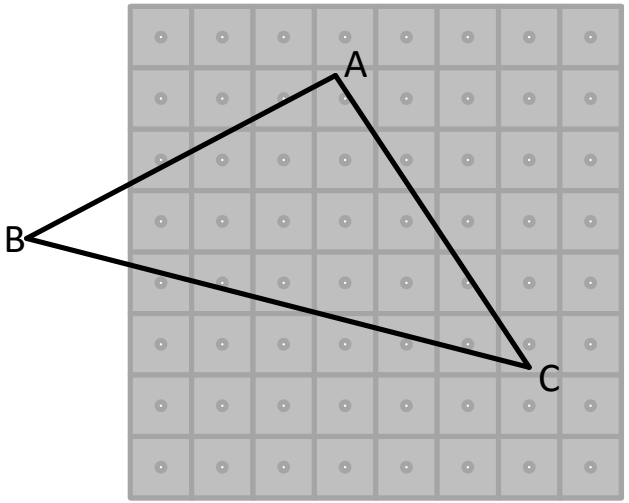


- Initialize ROV with -1
- GS assigns primId
- PS

```
if (Input.primId != ROV[xy]) {
  ROV[xy] = Input.primId;
  Shade();
} else
  discard;
```

- SV_Innercoverage

Handling clipped triangles

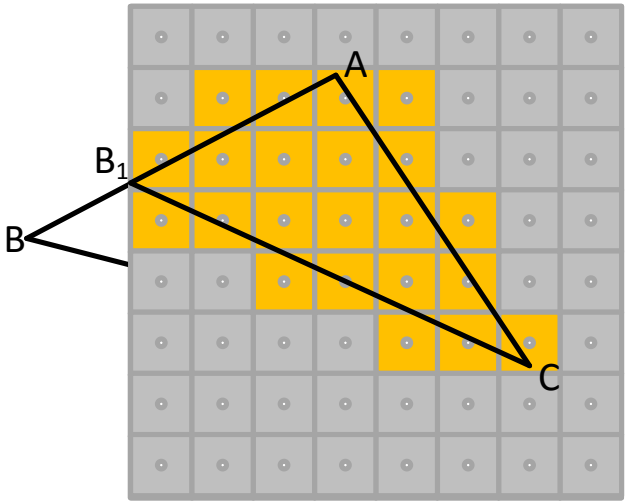


- Initialize ROV with -1
- GS assigns primId
- PS

```
if (Input.primId != ROV[xy]) {
  ROV[xy] = Input.primId;
  Shade();
} else
  discard;
```

- SV_Innercoverage

Handling clipped triangles

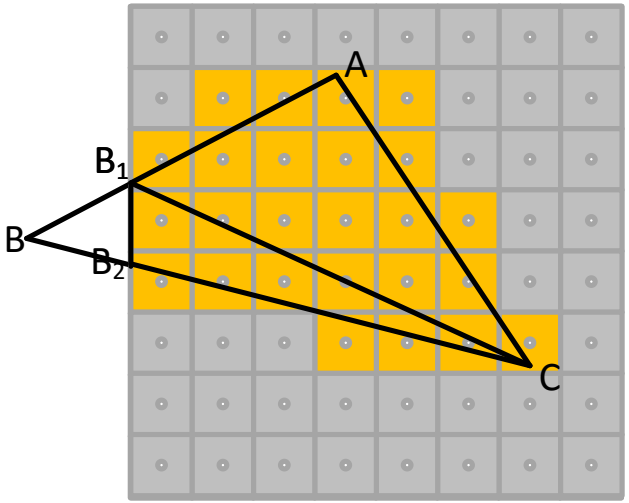


- Initialize ROV with -1
- GS assigns primId
- PS

```
if (Input.primId != ROV[xy]) {
  ROV[xy] = Input.primId;
  Shade();
} else
  discard;
```

- SV_Innercoverage

Handling clipped triangles



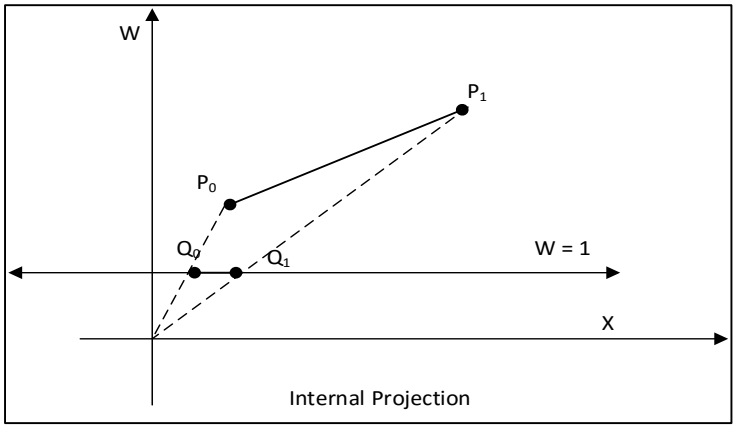
- Initialize ROV with -1
- GS assigns primId
- PS

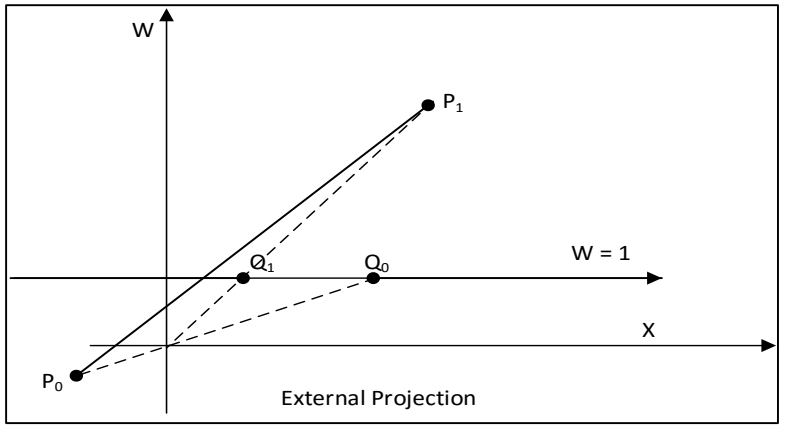
```
if (Input.primId != ROV[xy]) {
  ROV[xy] = Input.primId;
  Shade();
} else
  discard;
```

- SV_Innercoverage

Clipping when w < 0

- Produces external projections on w=1
- Cannot use edge equations ☹





http://www.gamasutra.com/view/news/168577/Indepth_Software_rasterizer_and_triangle_clipping.php

Clipping when w < 0

Clipping when w < 0

- Clip against the front plane in the GS
  - Might produce inconsistent vertices
  - But they are on the same edge ➔ same coefficients
- When both the vertices are behind the eye
  - Mark as invalid edge
  - Skip in-out tests in the PS

Temporal Super-sampling

- Plays well with the Temporal AA
- Filter Weights must be calculated per pixel
- Rest of the algorithm stays same
- Tends to have less flickering

Future

- Avoid Geometry Shader and late Z/stencil
- Shade @ pixel rate when SampleCount > 1
- Foveated rendering

References

- Akeley, K. (1993). Reality Engine Graphics. *Siggraph* (pp. 109-116). ACM.
- Brian Karis. (2014). High Quality Temporal Supersampling (Advances in Real-time Rendering in Games: Course). *Siggraph.*
- Microsoft. (2015). *Conservative Rasterization*
- Microsoft. (2015). *Direct3D Feature Levels*
- Microsoft. *Rasterization Rules (Windows)*
- *Raster Order Views*. (2015, July)
- Pharr, & G. Humphreys, *Physically Based Rendering from Theory to Implementation* (pp. 279-296). Morgan Kaufmann.
- Yeung, S. (2012, April 16). *In-depth: Software rasterizer and triangle clipping*

Acknowledgement

- Gareth Thomas (AMD)
- Andrei Tatarinov (NVDIA)

Q/A