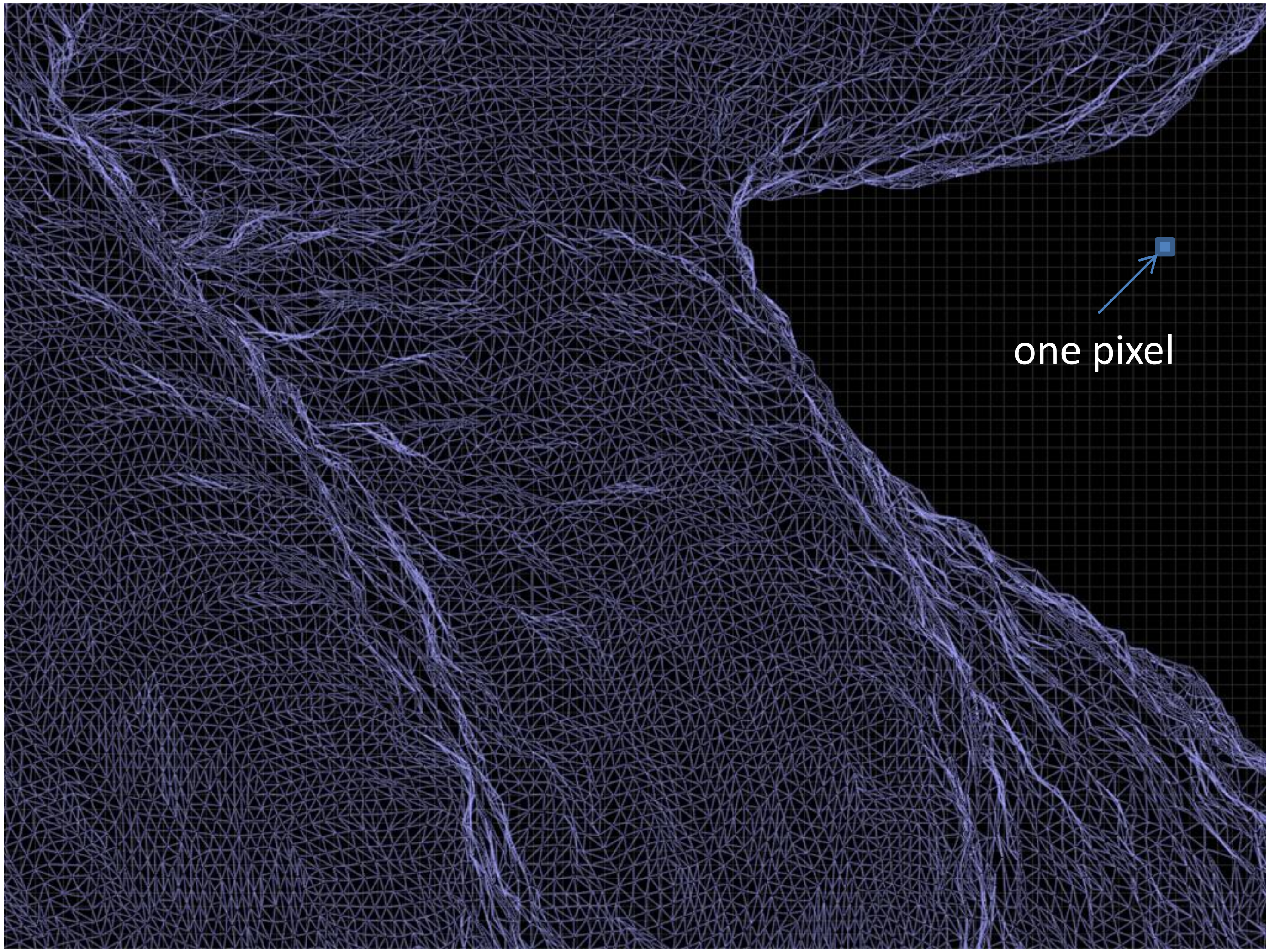
# Hardware Implementation of Micropolygon Rasterization with Motion and Defocus Blur

John Brunhaver, Kayvon Fatahalian, Pat Hanrahan

Stanford University

# Micropolygons



one pixel

# Micropolygons

# Real Time?

- Is micropolygon rasterization feasible?

- Is motion and defocus blur feasible?

# Why hardware?

- Hardware:
  - GTX 480 macro-triangle rasterization rate:
    - **3 billion tri/sec[1]**
- Software:
  - GTX 480 SM micro-triangle rasterization at 3 billion tri/sec
    - **~10 Cards[2]**
  - Larrabee micro-triangle rasterization at 3 billion tri/sec
    - **~10 Cards[2]**
- Motion and defocus blur?
  - Multiply by 7

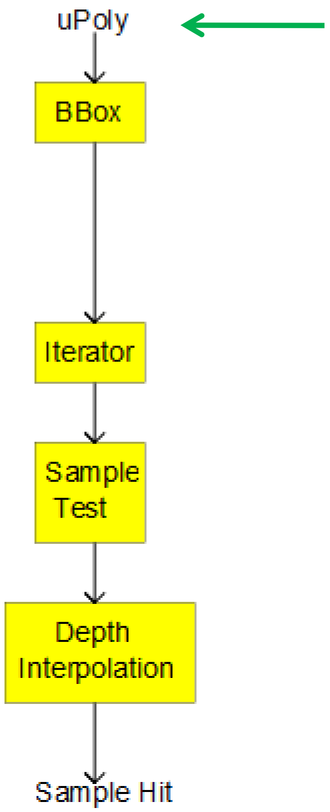
**[1] "GeForce GTX 480 And 470: From Fermi And GF100 To Actual Cards!" Tom's Hardware,** http://www.tomshardware.com/reviews/geforce-gtx-480,2585.html June 15th 2010

[2] FATAHALIAN K., LUONG E., BOULOS S., AKELEY K., MARK W. R., HANRAHAN P.: Data-parallel rasterization of micropolygons with defocus and motion blur. In *HPG '09: Proceedings of the Conference on High Performance Graphics 2009 (2009), ACM, pp. 59–68.*

# Contributions

- Parameterized fixed-function rasterization model
- Pareto optimal parameters for rasterization
  - Extended analysis to motion and defocus blur

# Micropolygon Rasterization



uPoly

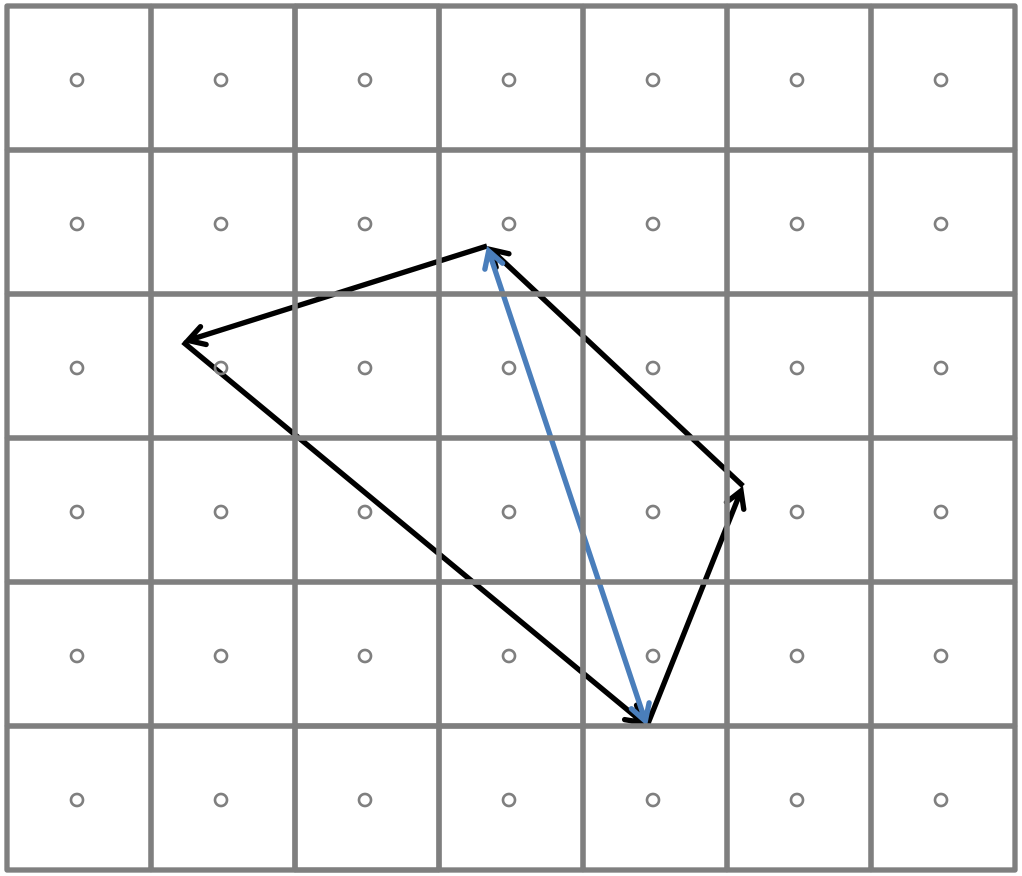
BBox

Iterator

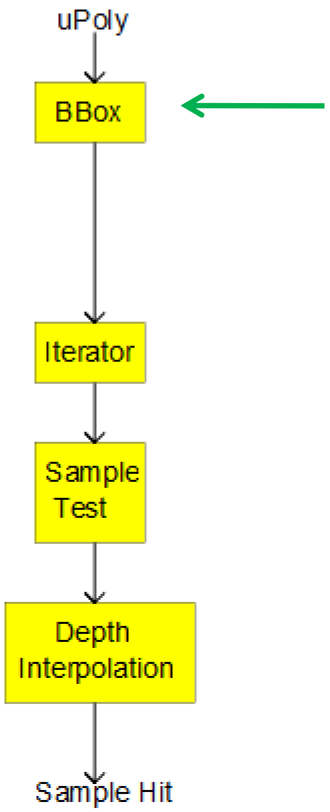
Sample Test

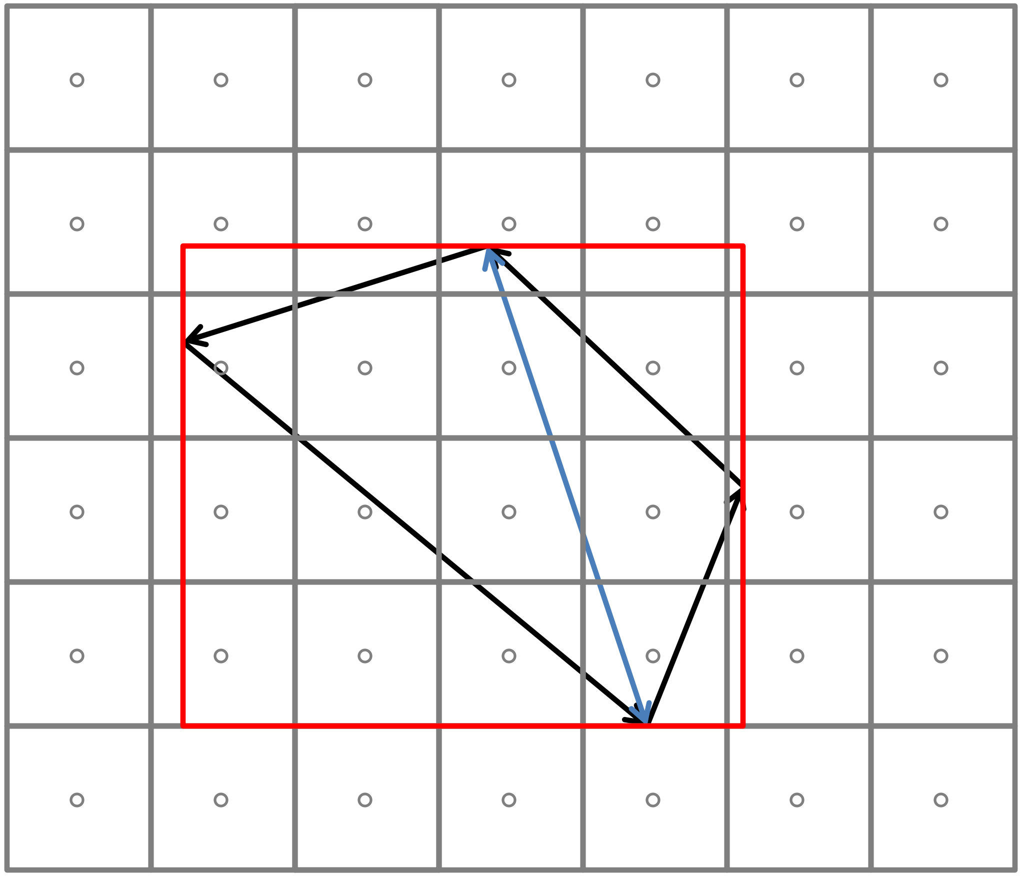
Depth Interpolation

Sample Hit

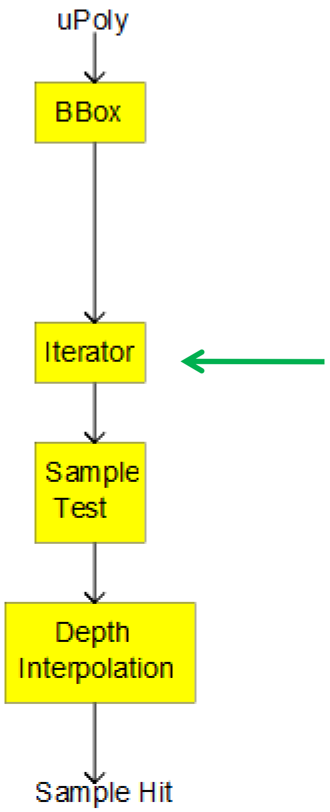


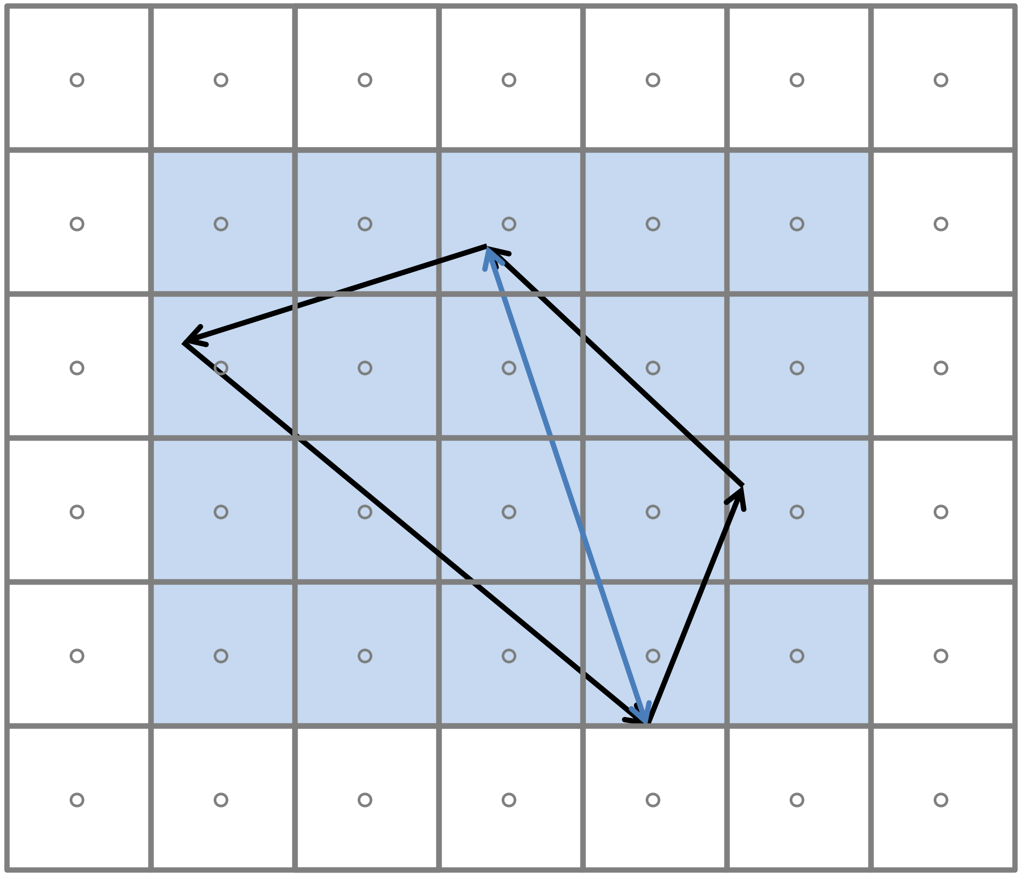
# Micropolygon Rasterization



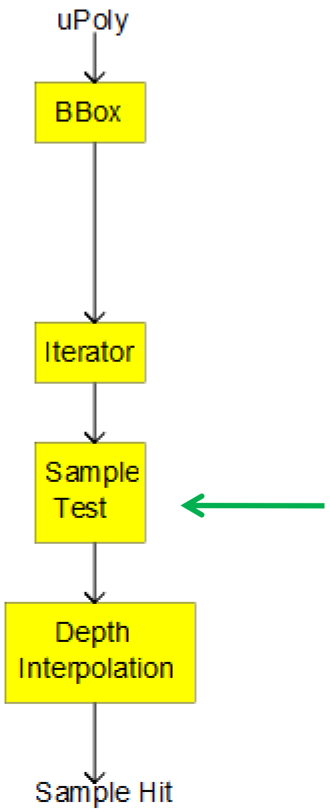


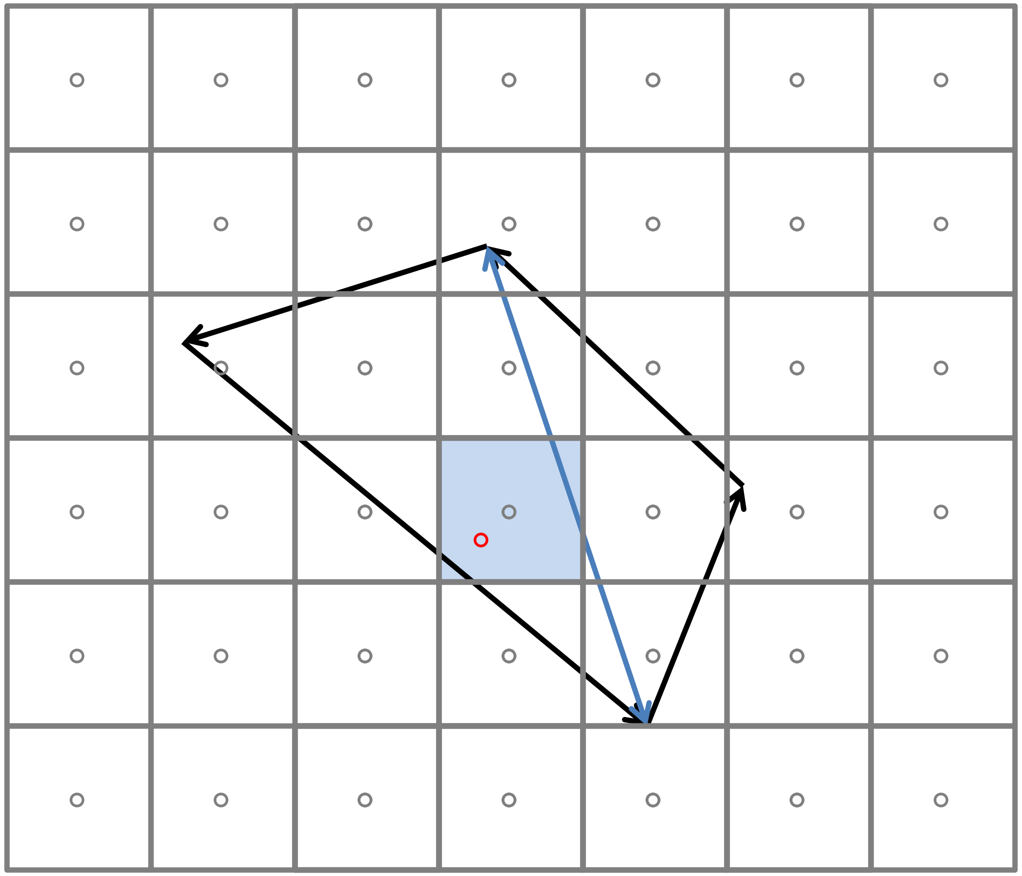
# Micropolygon Rasterization



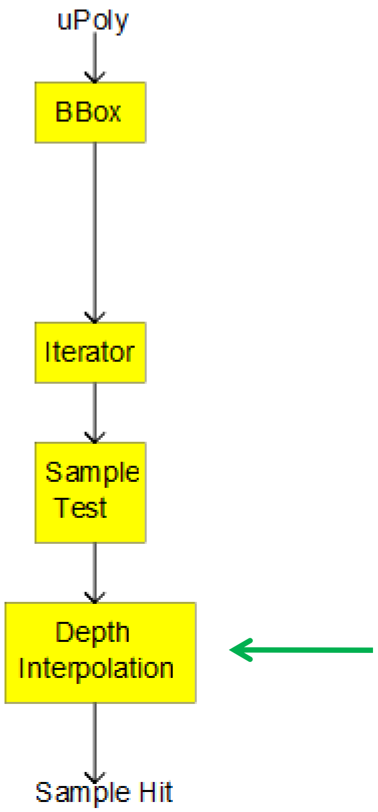


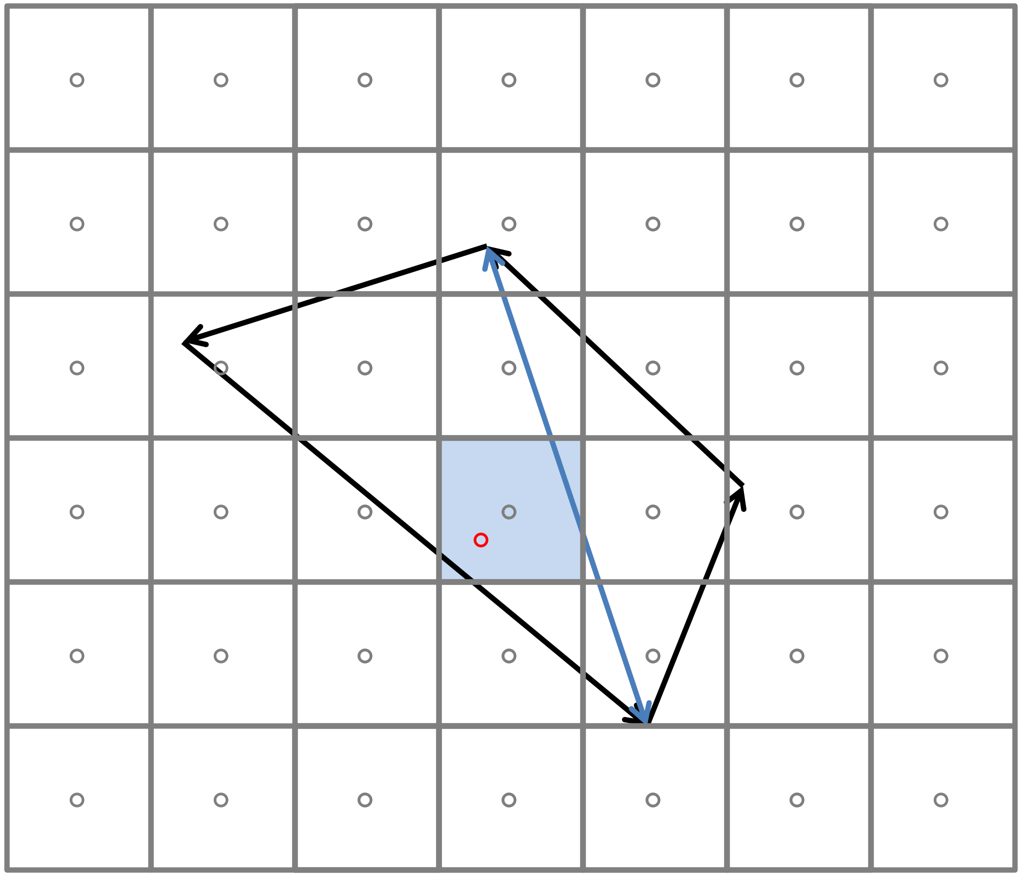
# Micropolygon Rasterization



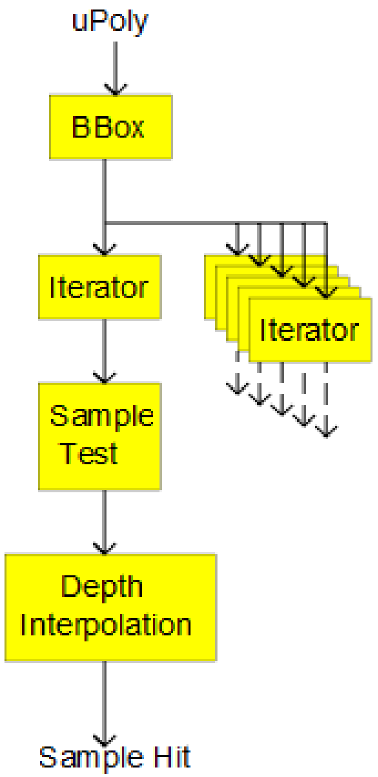


# Micropolygon Rasterization



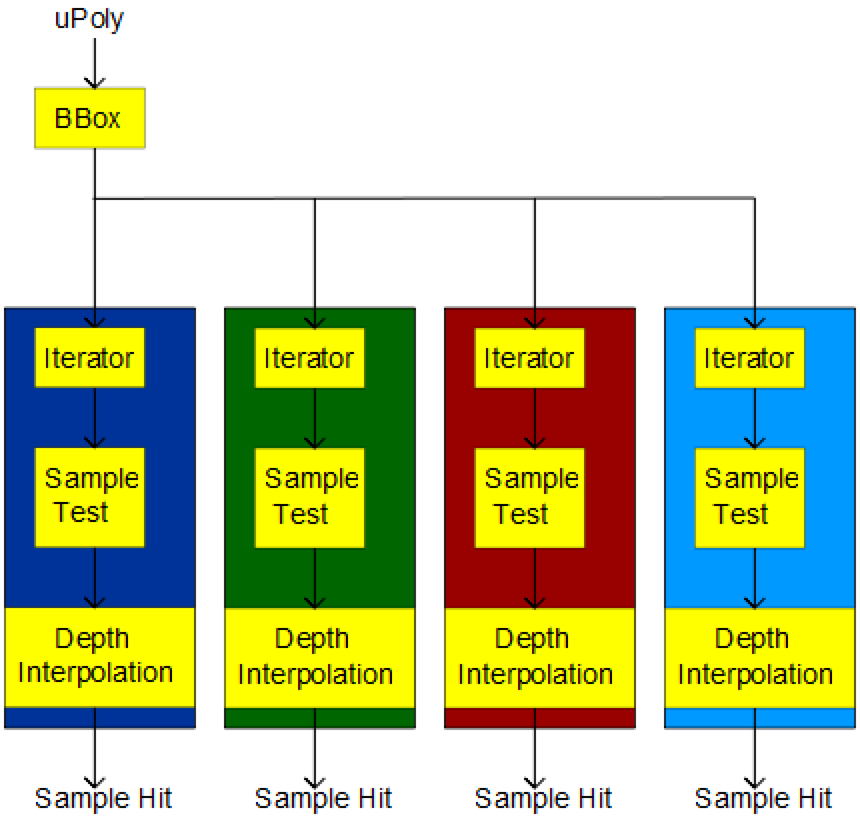


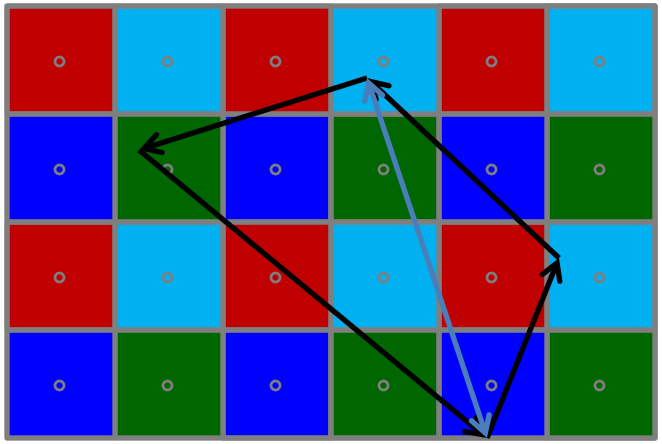
# Micro Architecture Parameters



- Parameters:
  - – Triangle Pairs
  - – Sample Stamp Size
  - – Sample Test Precision

# Sample Stamp





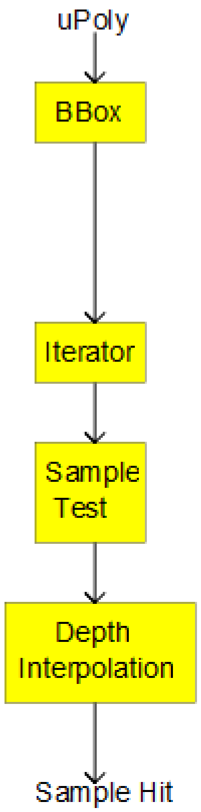
# Methodology

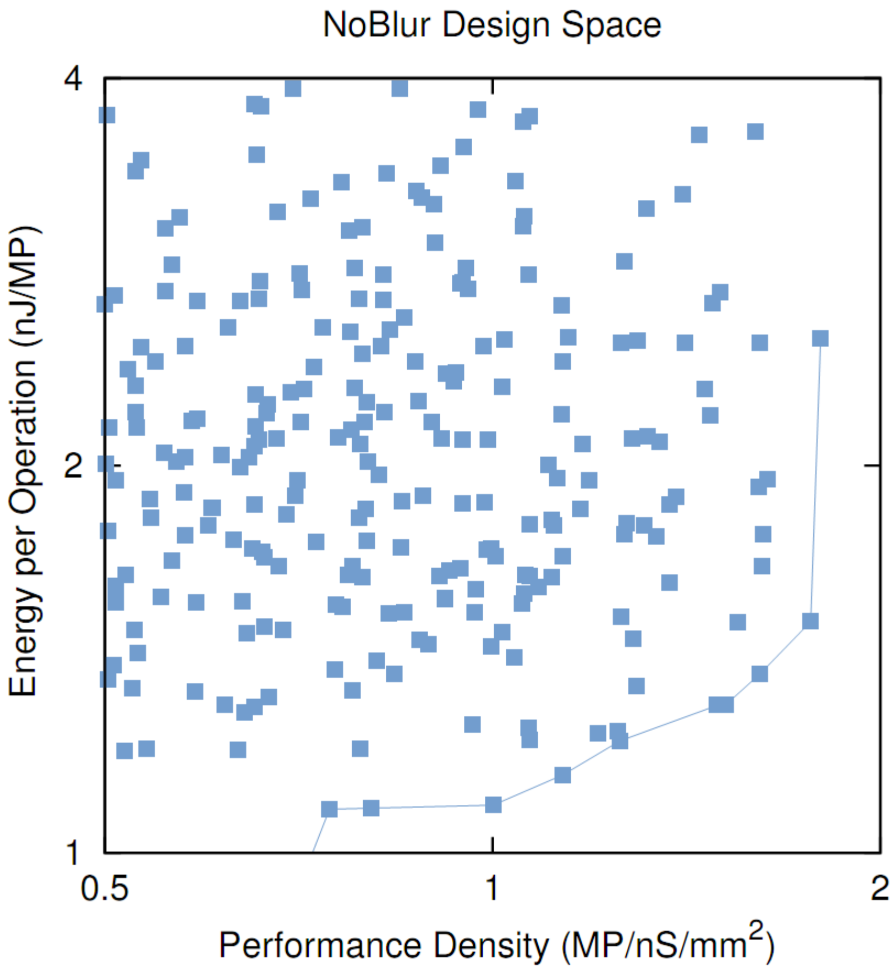
- Synthesized Verilog hardware model
  - TSMC 45nm Cell Libraries
  - Synopsis Design Compiler
  - Synopsis Design Ware Libraries
  - Calculated power and area
- Selected the Pareto optimal designs
- MSAA x16 and N=64

# Physical Design Parameters

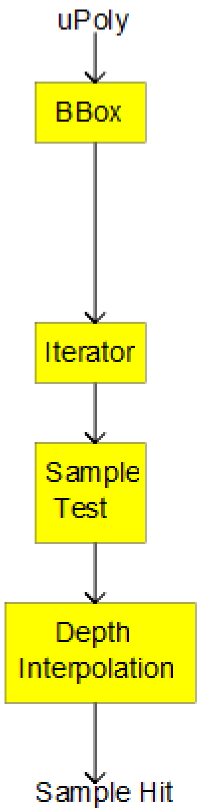
- Voltage ( .88-1.1 V)
- Threshold Voltage (high, nom, low)
- Frequency ( .25 – 5 GHz)
- Pipe stages (heuristically explored)

# Design Space





# Pareto Optimal


uPoly

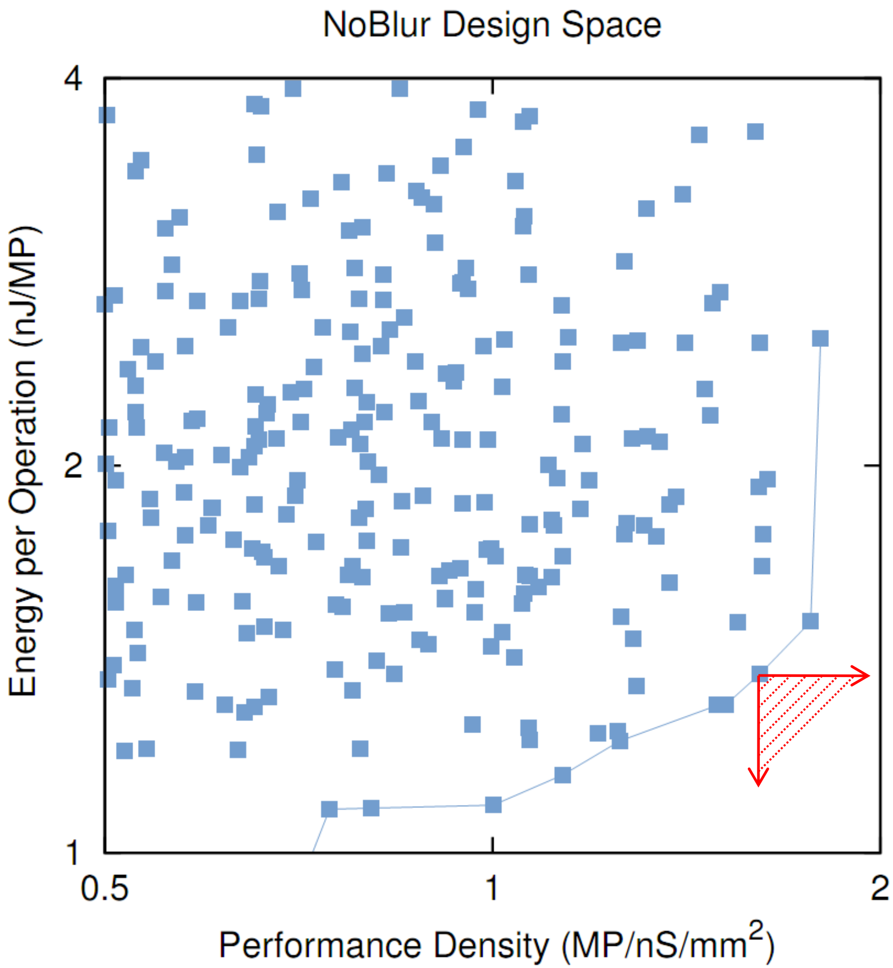
BBox

Iterator

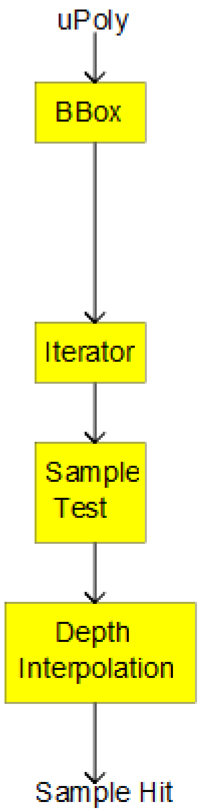
Sample Test

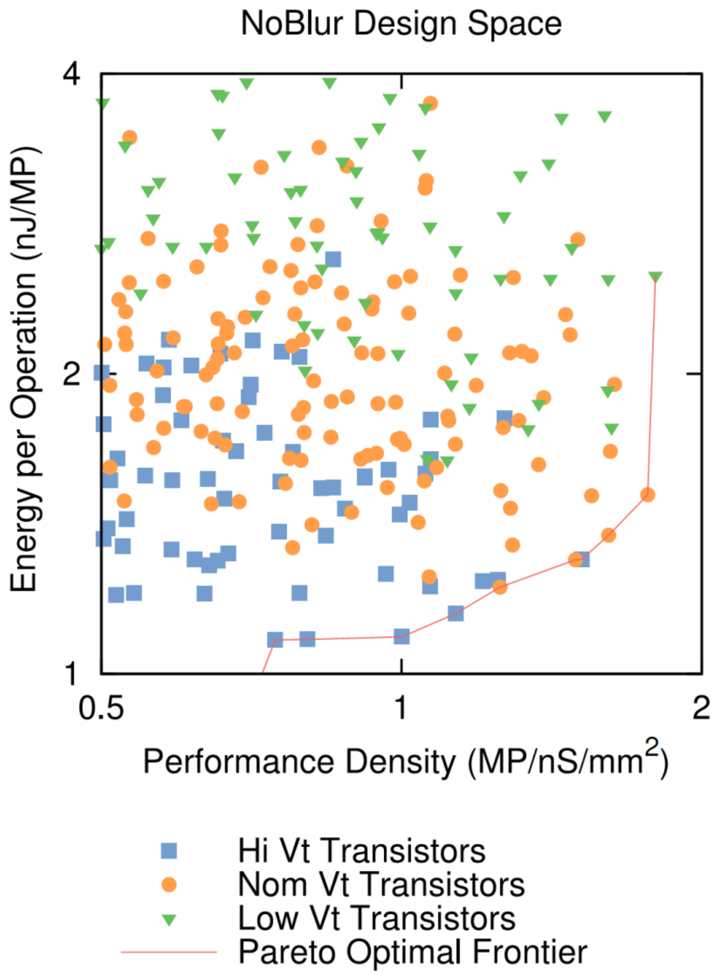
Depth Interpolation

Sample Hit

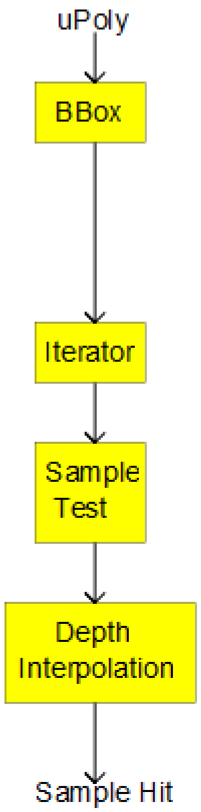


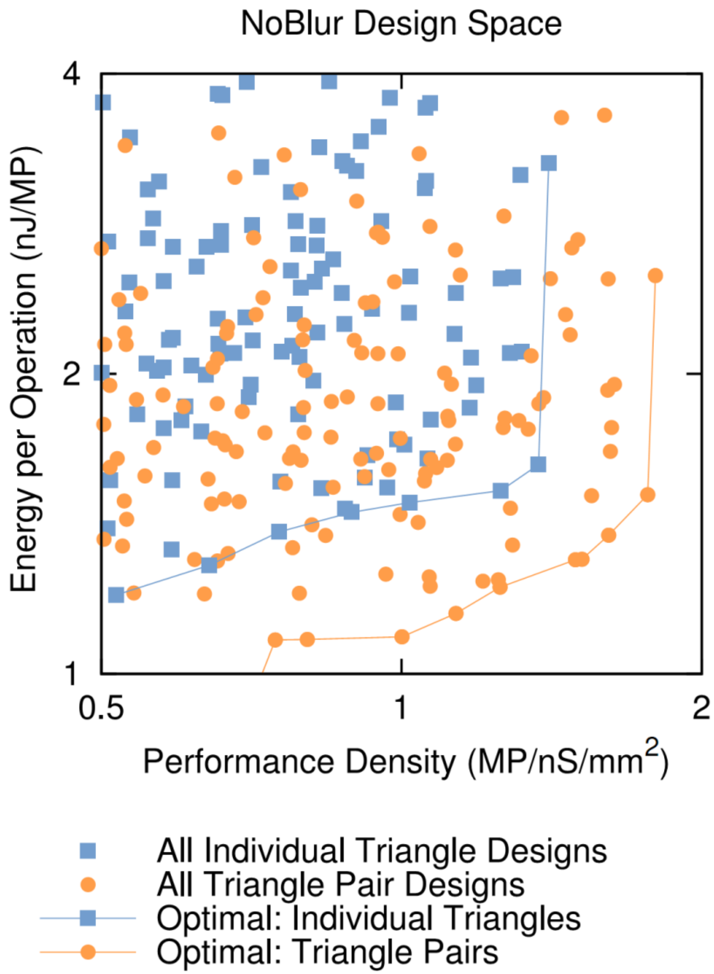
# Transistor Threshold Voltage



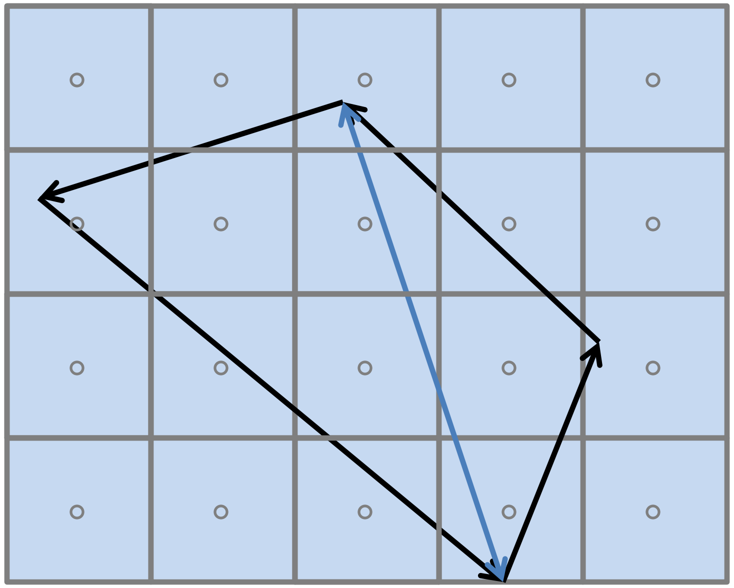


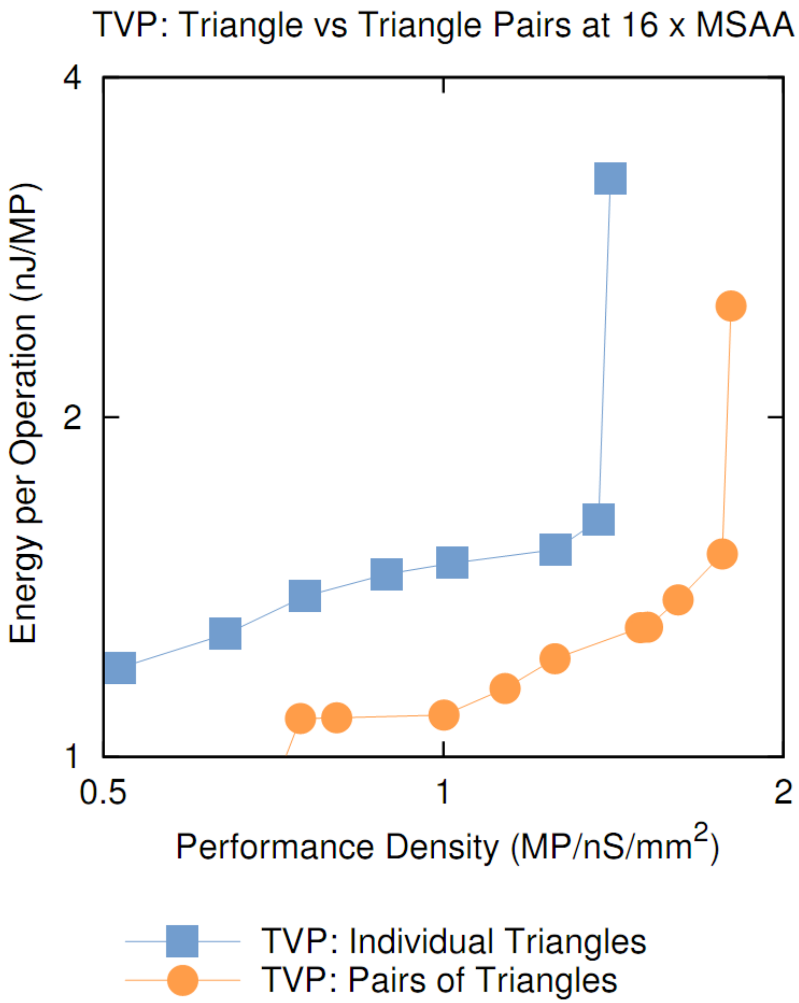
# Triangle Pairs vs Individual Triangles



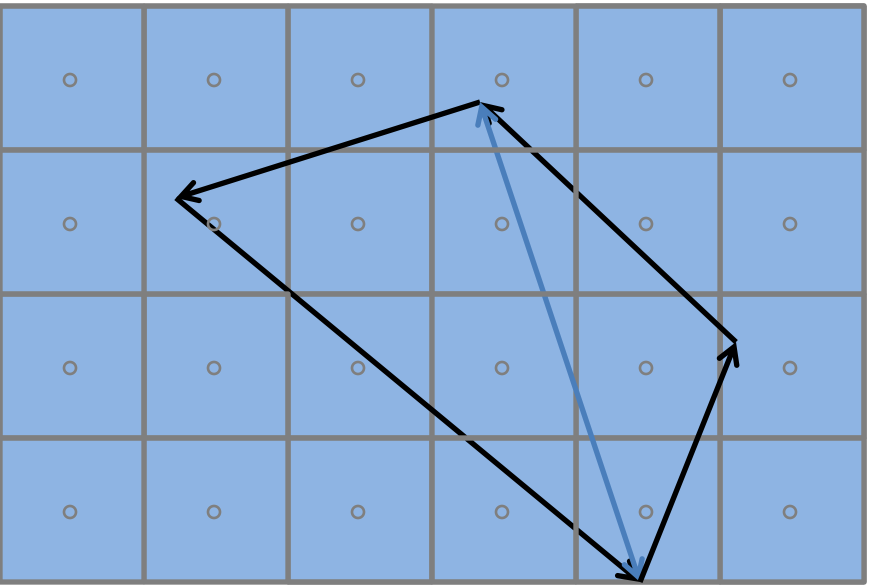


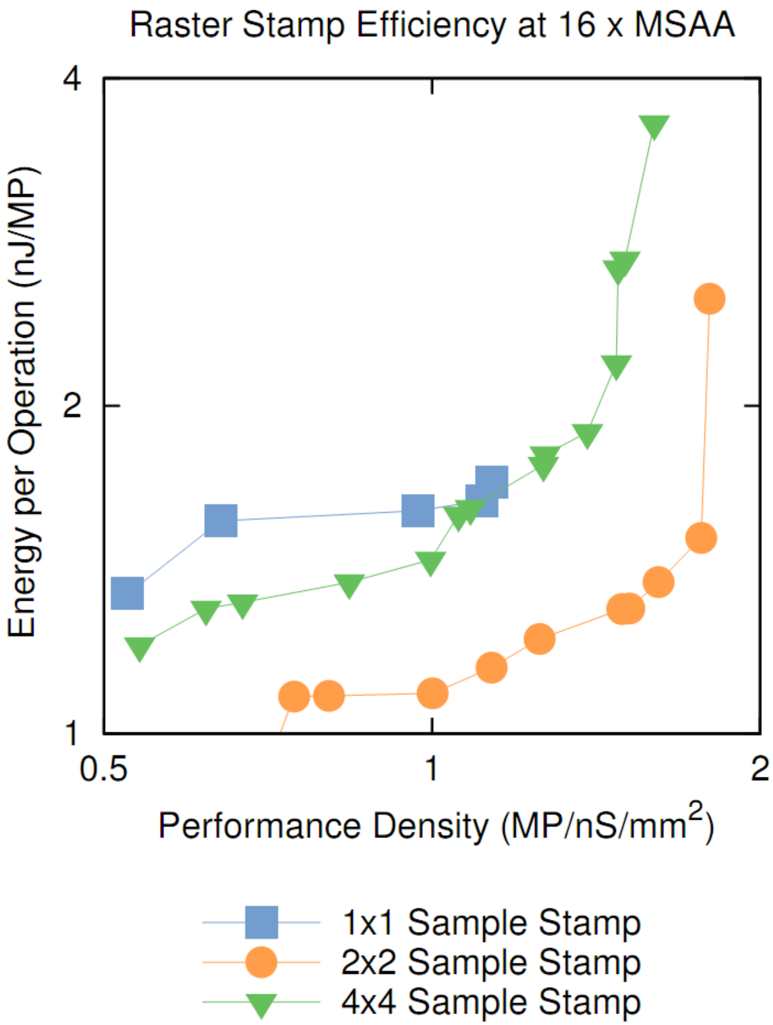
# Triangle Pairs is More Efficient



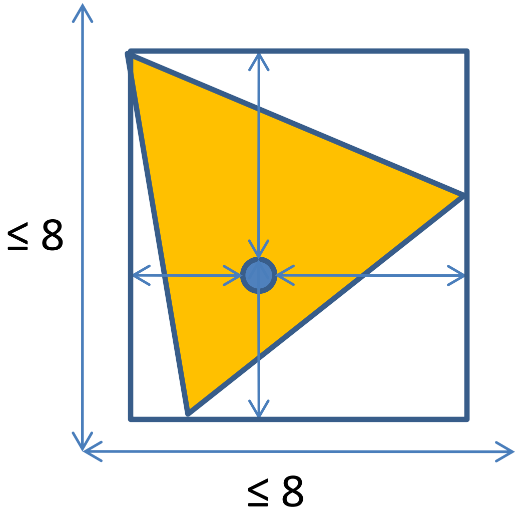

TVP: Triangle vs Triangle Pairs at 16 x MSAA
Energy per Operation (nJ/MP)
4
2
1
0.5
1
2
Performance Density (MP/nS/mm$^2$)
TVP: Individual Triangles
TVP: Pairs of Triangles

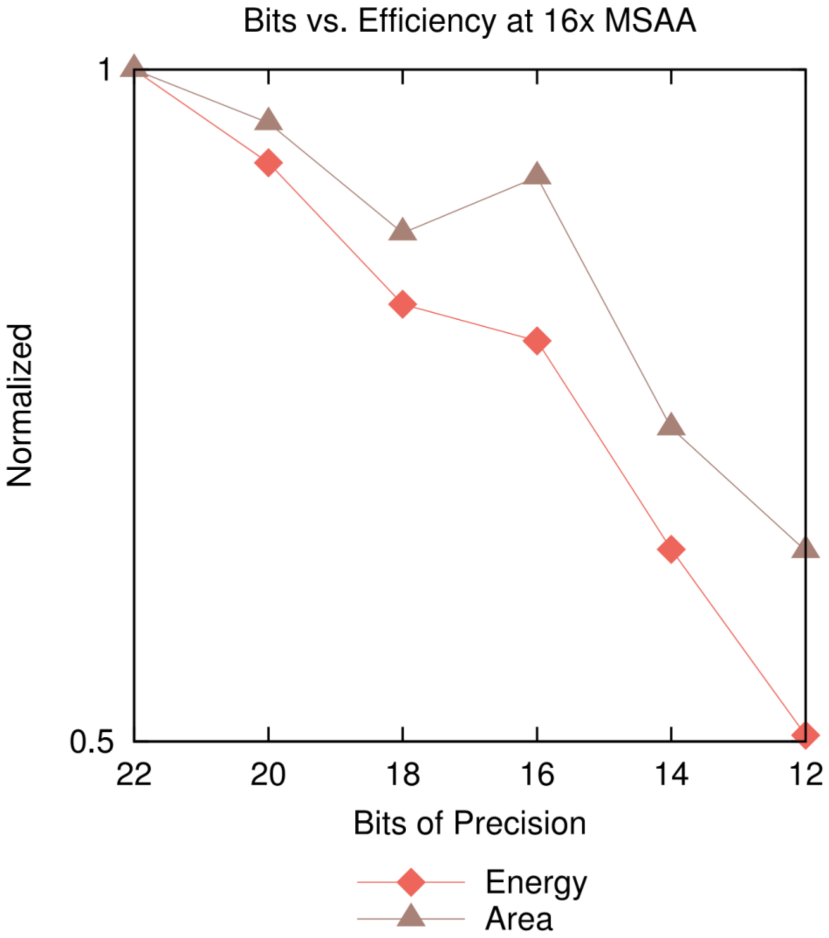
# 2x2 Sample Stamp is Most Efficient




Raster Stamp Efficiency at 16 x MSAA
Energy per Operation (nJ/MP)
4
2
1
0.5
1
2
Performance Density (MP/nS/mm$^2$)
1x1 Sample Stamp
2x2 Sample Stamp
4x4 Sample Stamp

# 4.8 Bit Precision for 2x Efficiency





# Efficient Rasterization Micro Architecture

- Operate on triangle pairs
- Sample stamp is 2x2
- 4.8 bits of precision in sample test

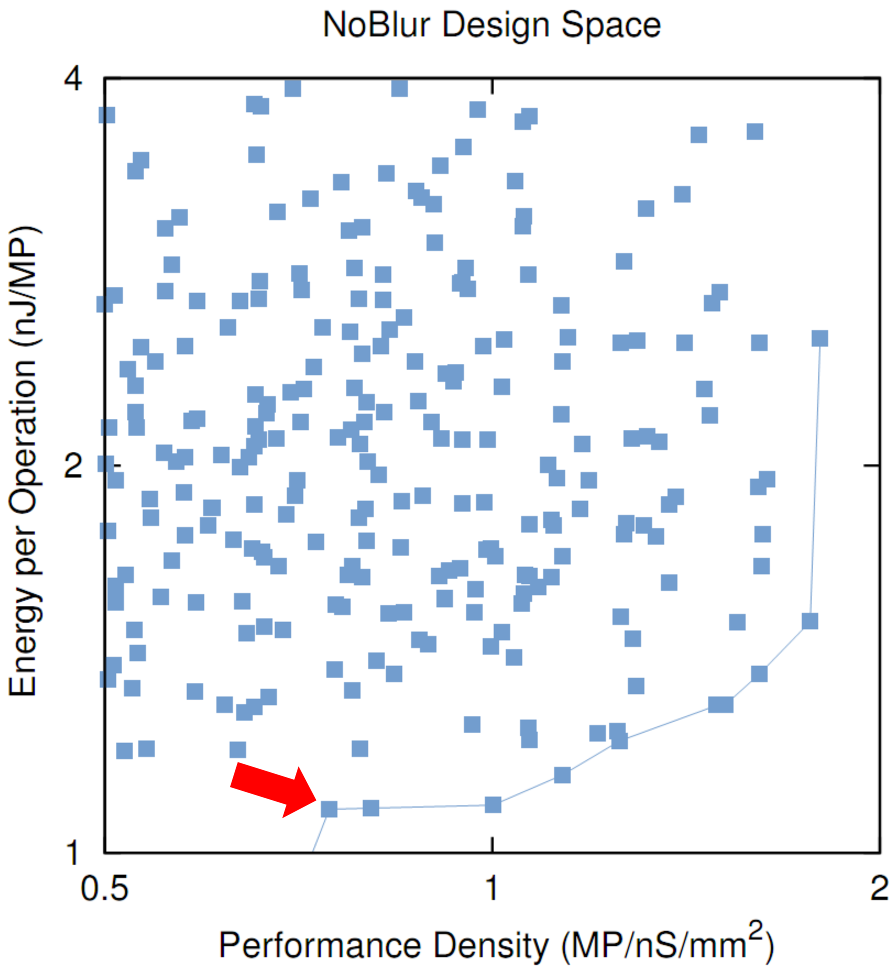
# Result

- 3 billion tri/sec
- MSAAx16
- Relative to GTX 480

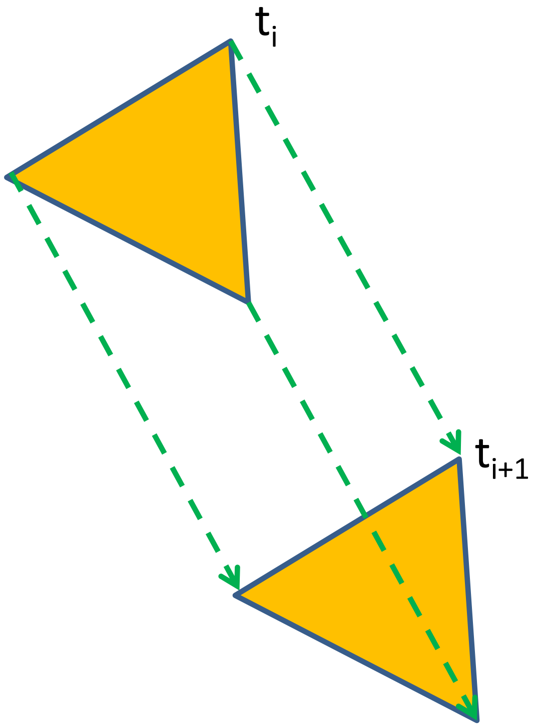
- 18 units

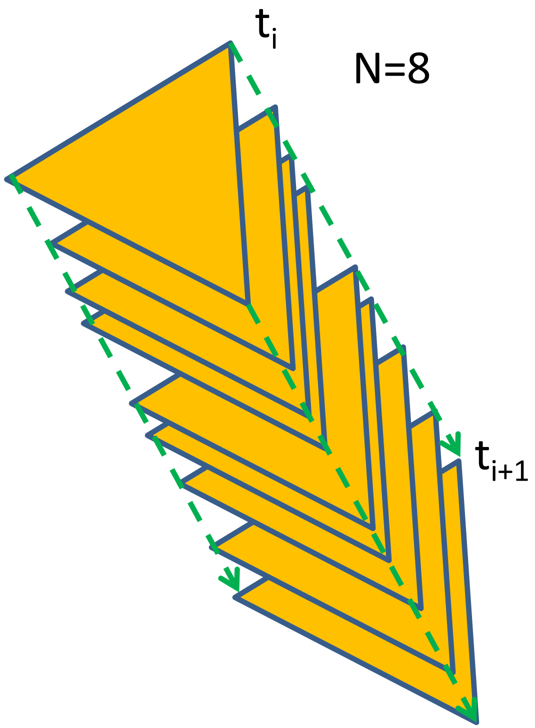
- 4.2mm$^2$
- 0.79% of die area

- 2.8W
- 0.78% of board power



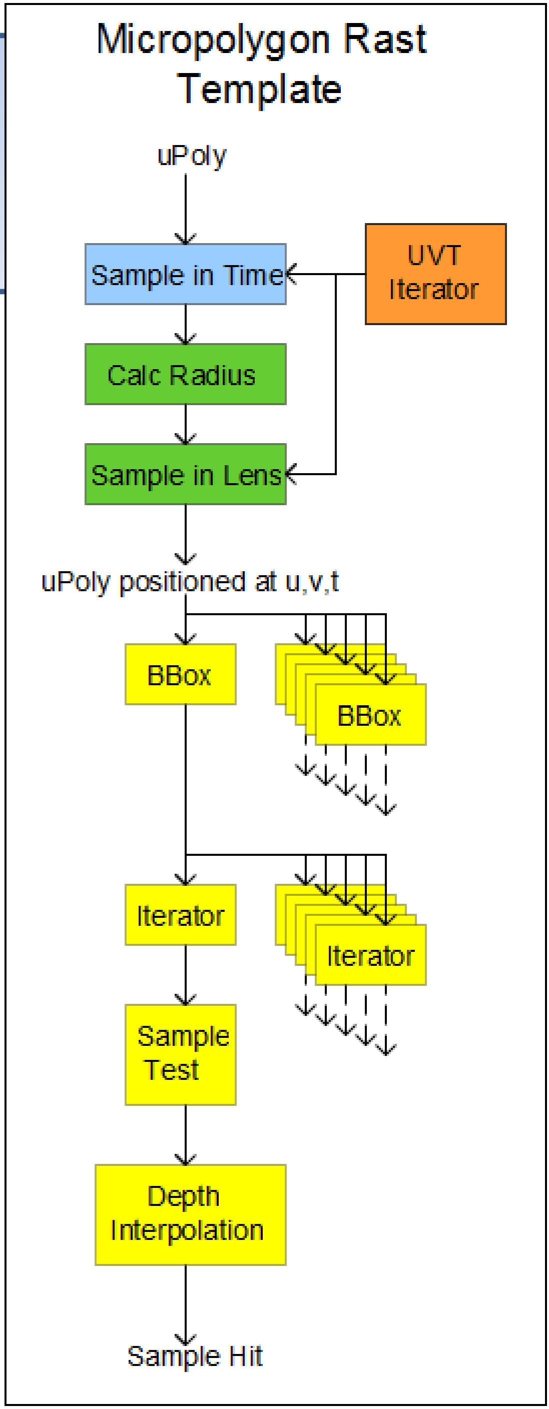
# Motion Blur with Interleave

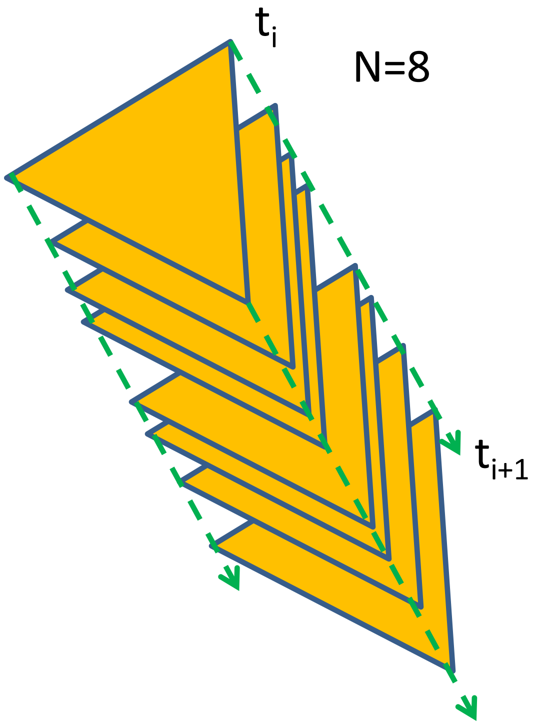


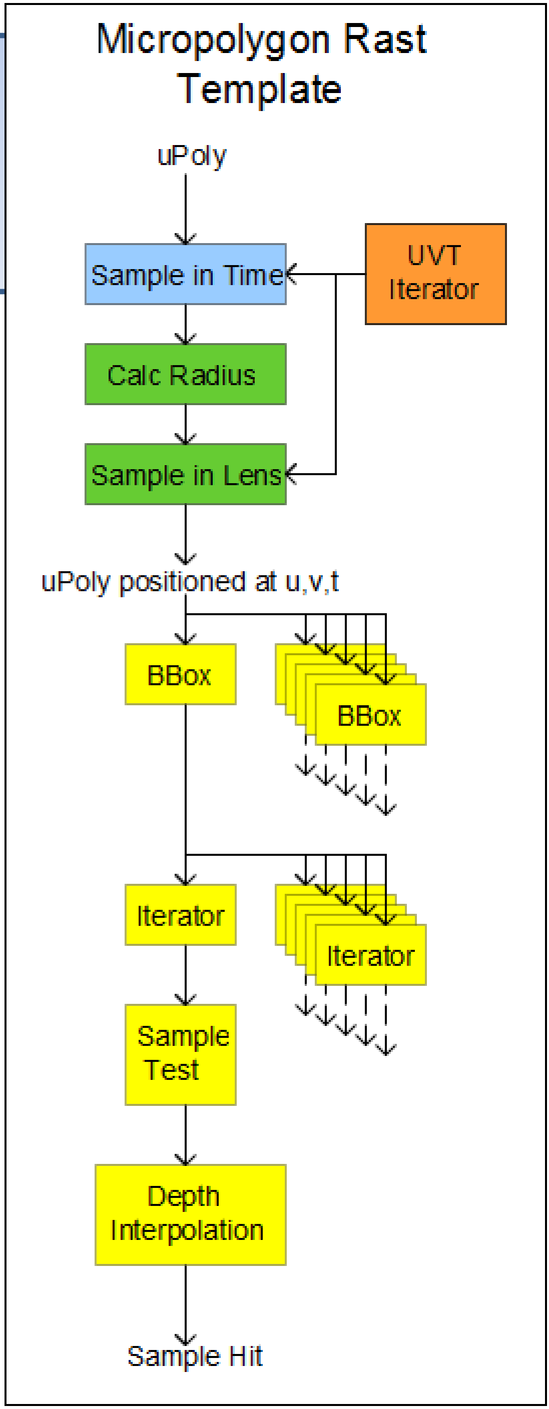


[FLB*09] Fatahalian K., Luong E., Boulos S., Akeley K., Mark W. R., Hanrahan P.: Data-parallel rasterization of micropolygons with defocus and motion blur. In *HPG '09: Proceedings of the Conference on High Performance Graphics 2009* (2009), ACM, pp. 59–68.

# Microarchitecture



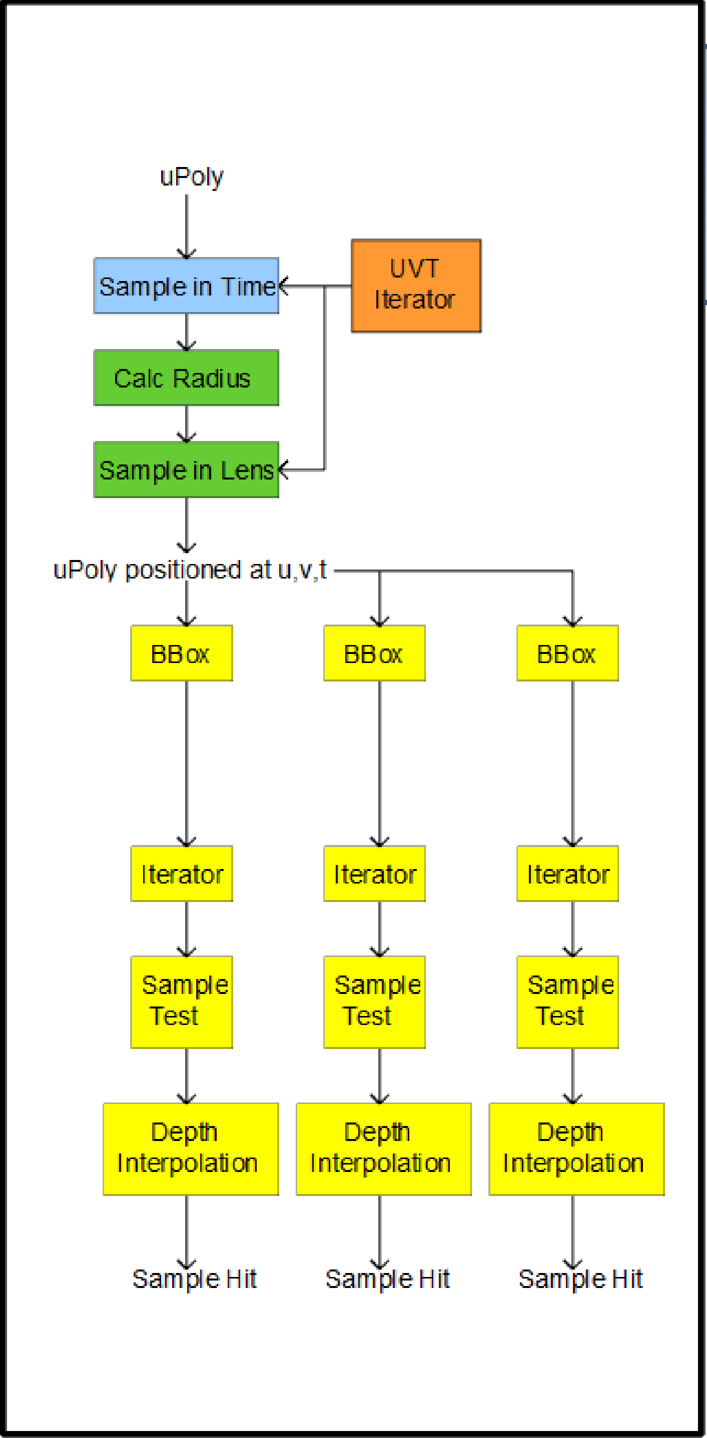


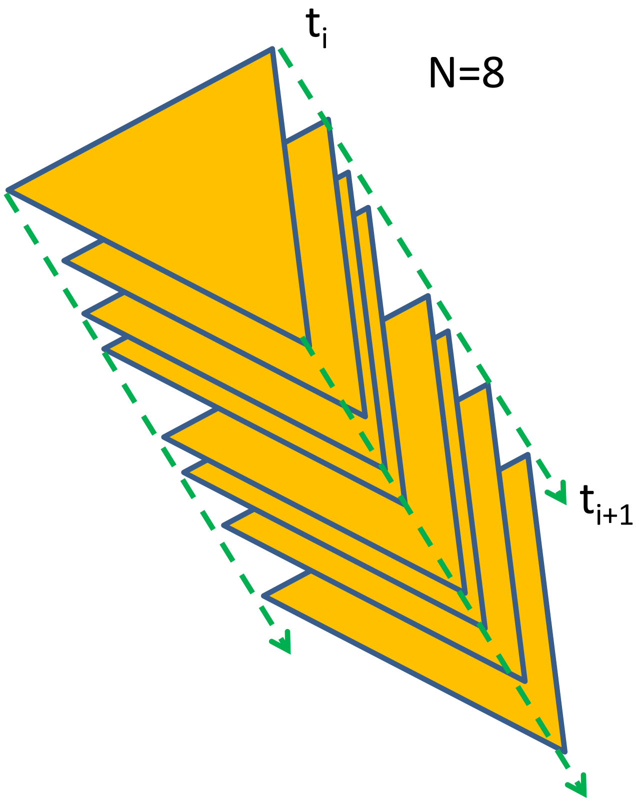


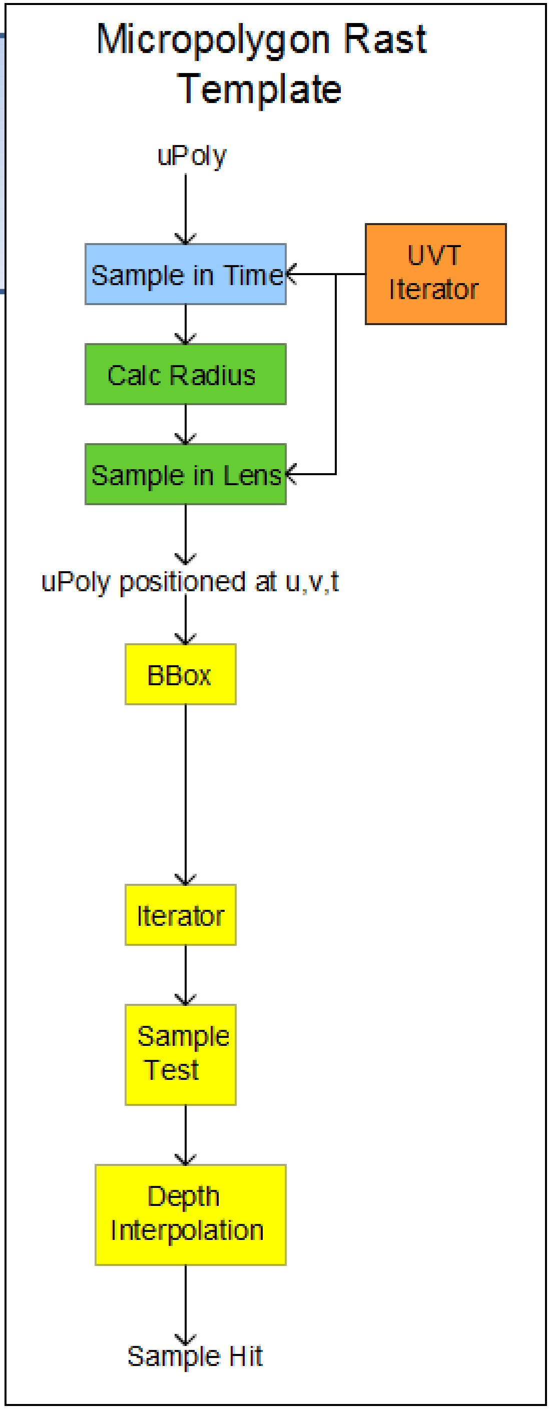
# Microarchitecture Parameters

- Triangle pairs
- uvt setup rate
- Parallelism over uvt
- Sample stamp size
- Precision

# uvt Parallelism of 3

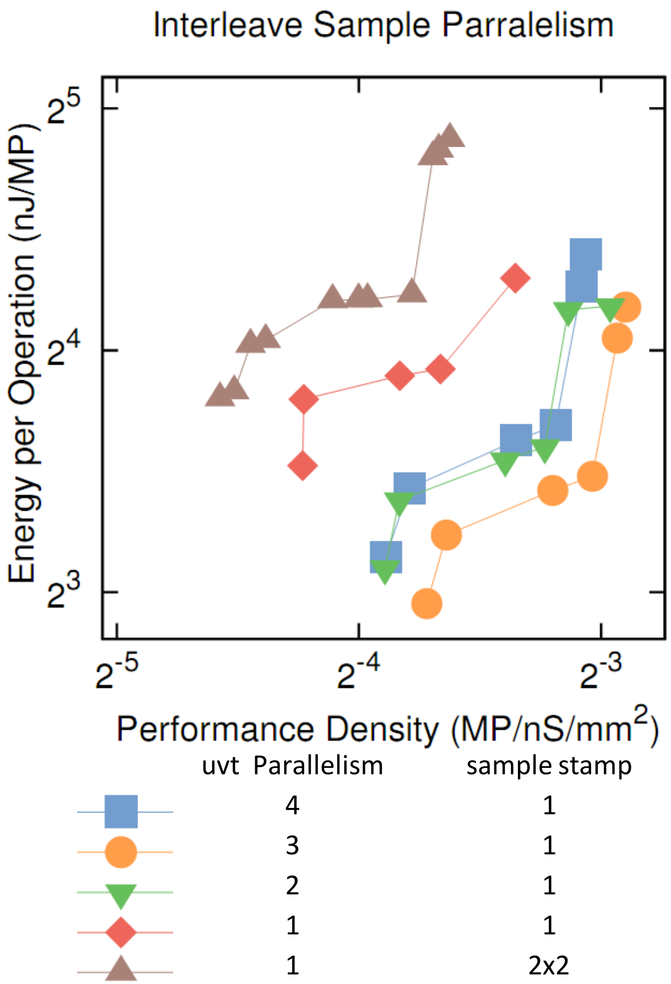






# 1x1 Sample Stamp uvt Parallelism of 3

N=64
MSAA x16



| | uvt Parallelism | sample stamp |
|---|---|---|
| ■ | 4 | 1 |
| ● | 3 | 1 |
| ▼ | 2 | 1 |
| ◆ | 1 | 1 |
| ▲ | 1 | 2x2 |

# Efficient Micro Architecture for Blur

- Operate on triangle pairs
- Setup Rate is 1 uvt tuple a cycle
- Evaluate 3 uvt tuples simultaneously
- Sample Stamp 1x1
- 4.8 bit precision in sample test

# Blur Will Cost 9-10x More

| | No Blur | Blur (Interleave UVT) | Ratio |
|---|---|---|---|
| Units | 18 | **72** | |
| Area | 4.2 $mm^2$ | **37 $mm^2$** | |
| % of GTX 480 Die Area | 0.79% | **7.0%** | 9 |
| Power | 2.8 W | **28 W** | |
| % of  GTX 480 Board Power | 0.78% | **7.8%** | 10 |

# Micropolygon Rasterization in Real-Time

- Is micropolygon rasterization feasible?
  - YES!
- Is motion and defocus blur feasible?
  - Maybe

# Limitations

- Functional unit only
  - Interconnect could be relatively expensive
  - Ordered parallel rasterization is difficult

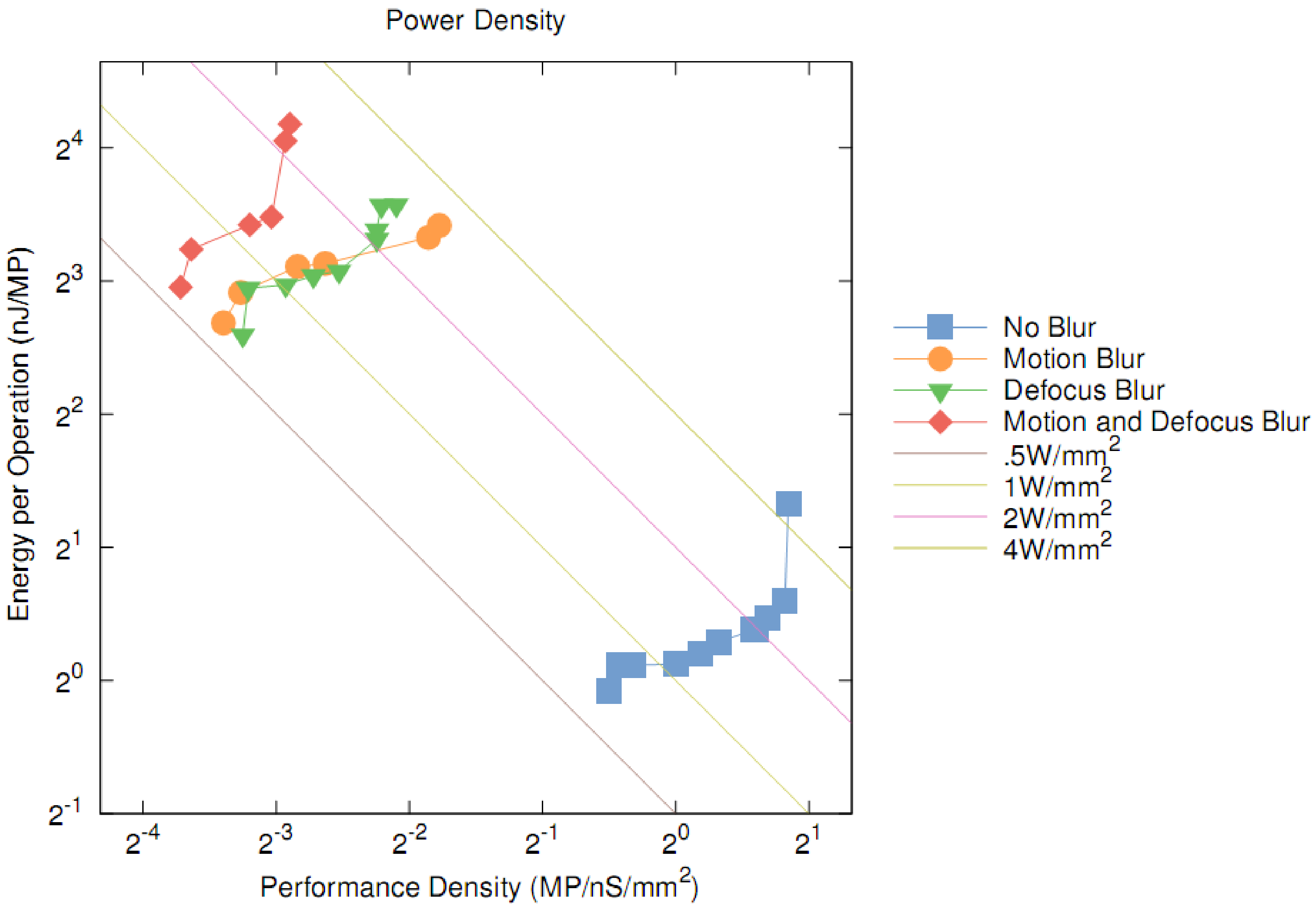
- Only micropolygons
  - Might need to efficiently handle all cases

# Conclusion

- Hardware implementations will be required
- Better algorithms are needed for blur
- Push better algorithms into hardware

# Backup

# Power Density



# Marginal Cost

- Energy and Delay Sensitivity with respect to a parameter ( Marginal Cost ).

$$Sens\left(V_{dd}\right) = -\left.\frac{\partial E / \partial V_{dd}}{\partial D / \partial V_{dd}}\right|_{V_{dd}=V_{dd}{}^{*}}$$

- At the optimal point all sensitivities should be the same. (The slope at the point on the Pareto optimal curve)

Ack: M. Horowitz

# Inefficiencies in a CPU

- Compare to an ASIC/SFU/FFU/Accelerator
- In the case of micropolygon rasterization
  - Instruction Fetch
  - Decode
  - Register Access
  - Precision