GPU TECHNOLOGY CONFERENCE

POWER EFFICIENT VISUAL COMPUTING ON MOBILE PLATFORMS

BRANT ZHAO, NVIDIA

MAX LV, NVIDIA

- Performance
- Energy Efficiency

Power Efficient GPU Programming
- Case Studies & Findings

Case study #1: Image Pyramid Blending

Image Pyramid Blending

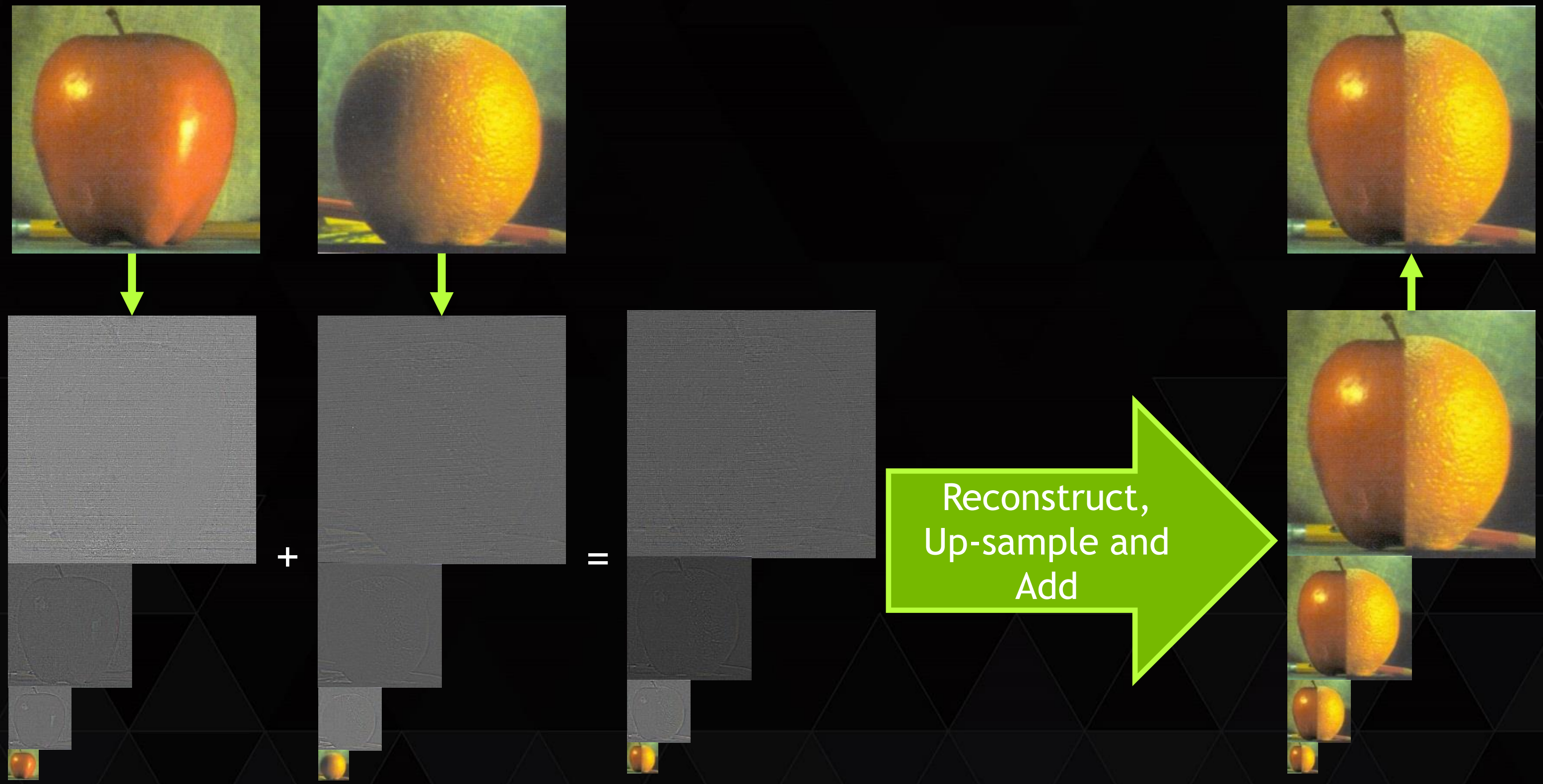


Image Pyramid Blending

- A naïve CUDA implementation

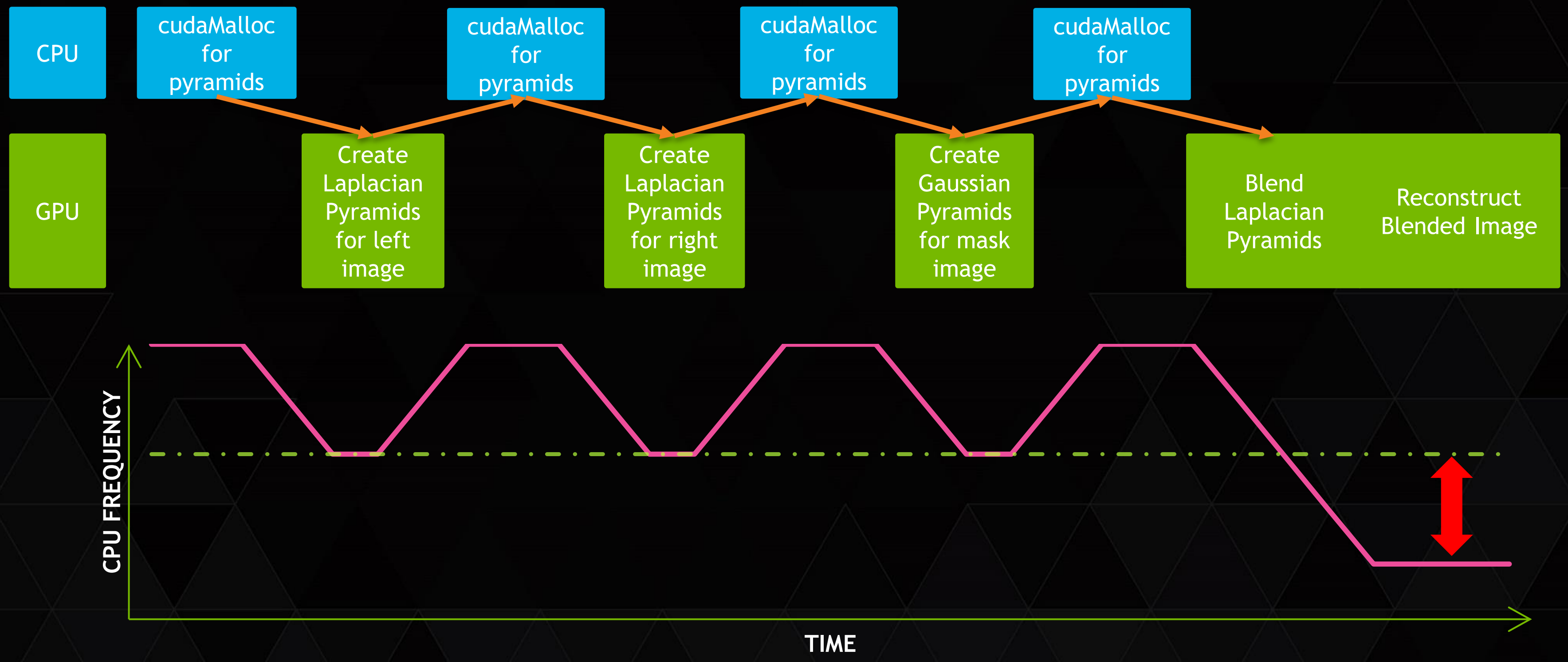
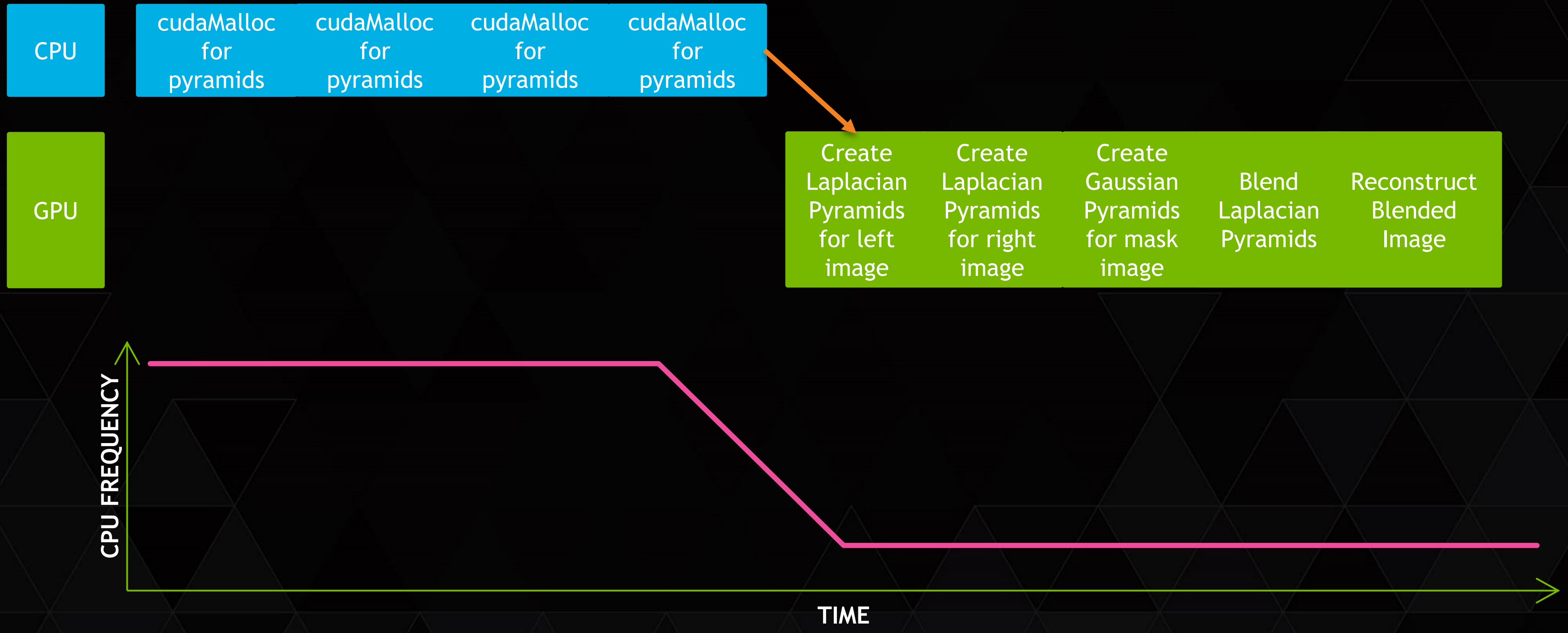


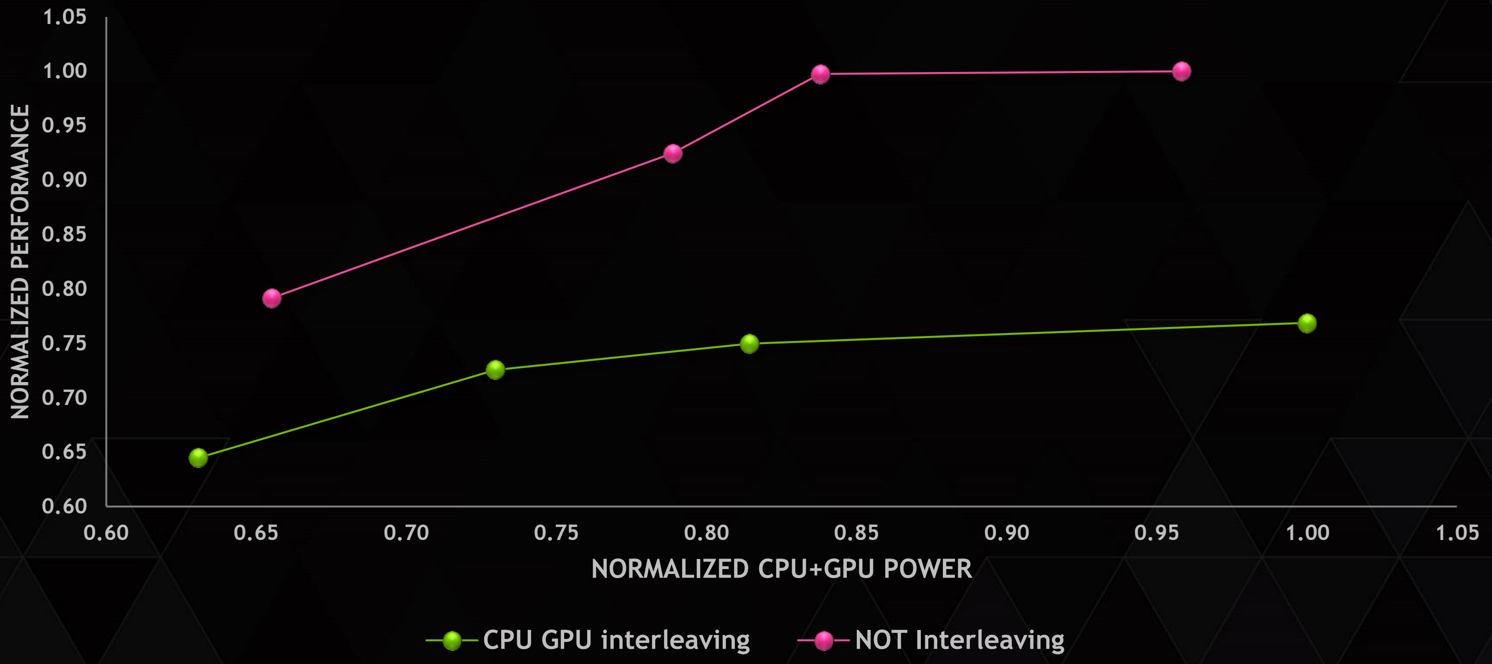
Image Pyramid Blending

- Power optimized: Avoid CPU<->GPU interleaving



GPU TECHNOLOGY CONFERENCE

Image Pyramid Blending
- Perf/Watt comparison



Case study #2: 2D Convolution

2D Convolution

0.25	0	0
0	0	0
0	0	0.75

+

1	2	1	2	
0	0	2	3	
2	0	1	10	

=

	1			

2D Convolution

0.25	0	0
0	0	0
0	0	0.75

+

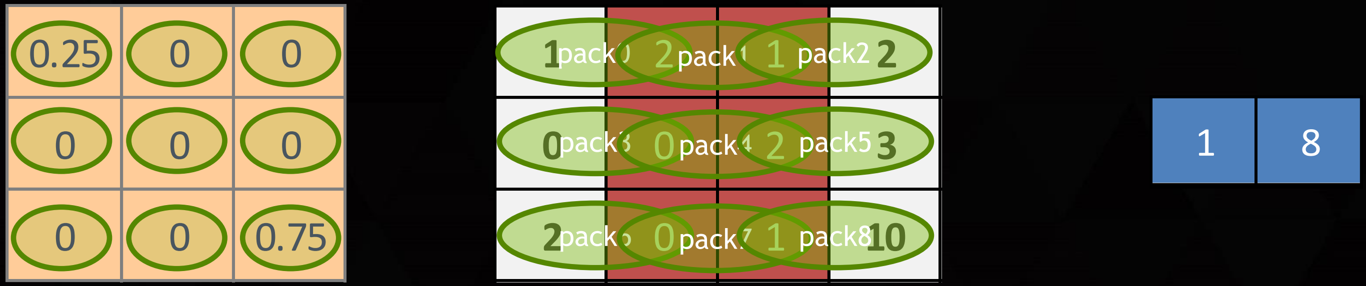
1	2	1	2	
0	0	2	3	
2	0	1	10	

=

	1	8		

2D Convolution

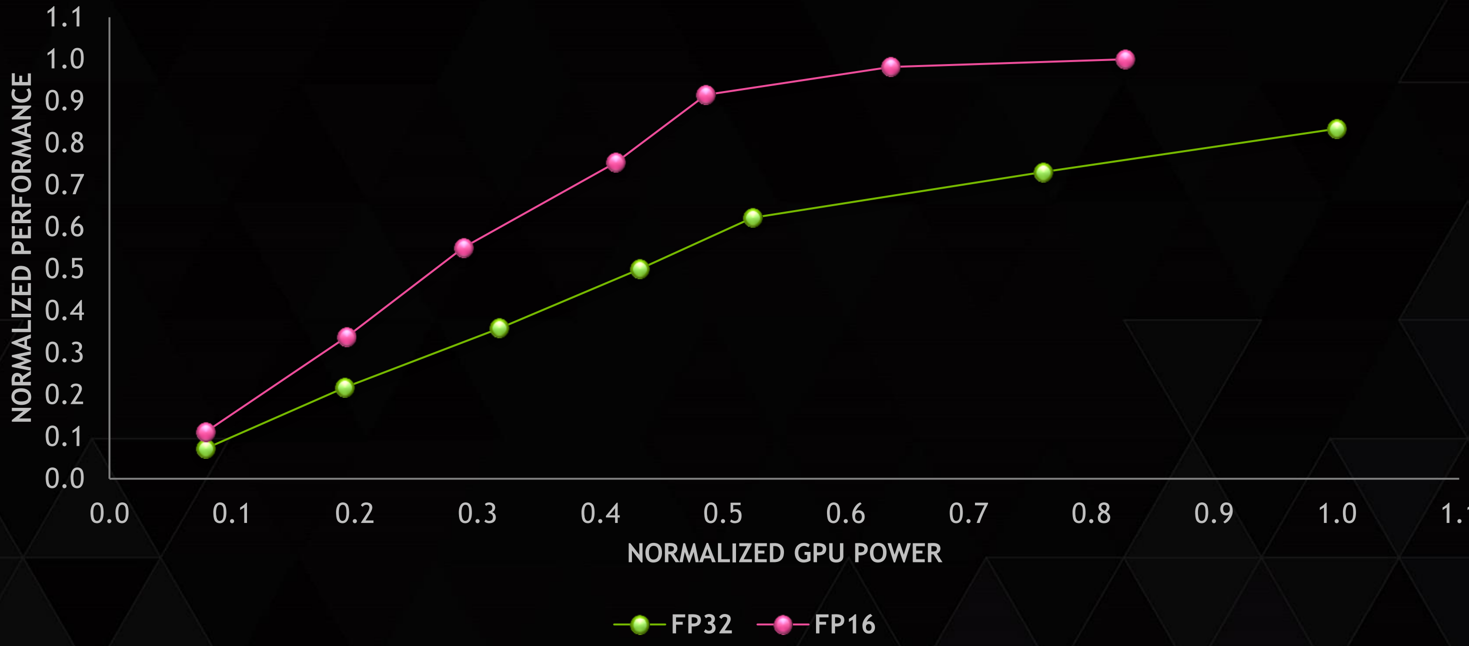
- 3x3 2D convolution with FP16



- Basic operations for 2 output pixels
  - 9 packed FP16 MAD

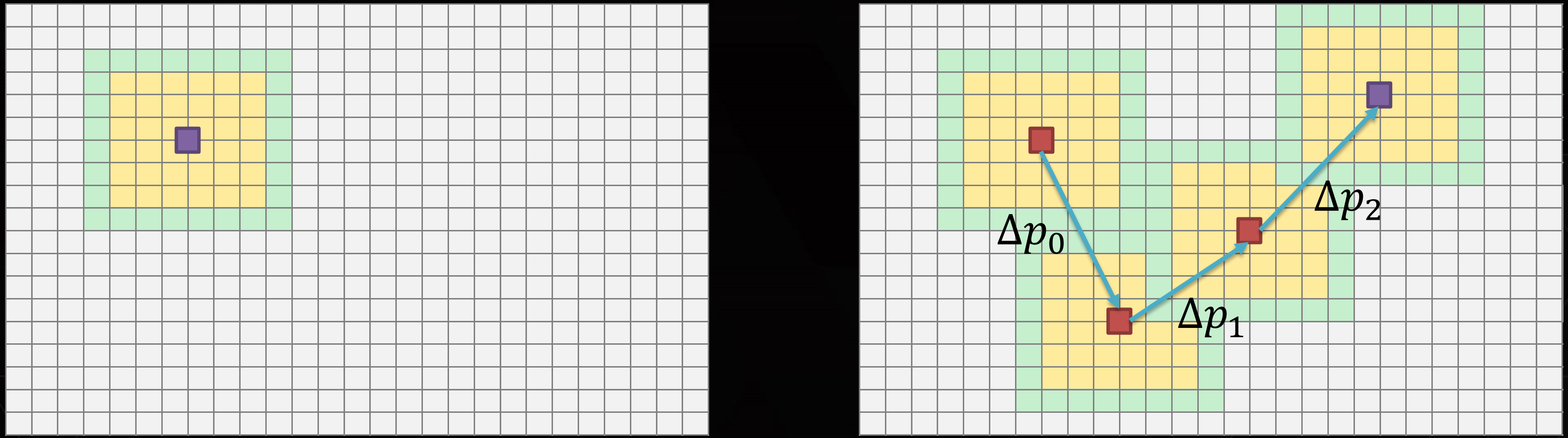
2D Convolution

- Perf/Watt comparison



Case study #3: Sparse Lucas-Kanade Optical Flow (SparseLK)

SparseLK



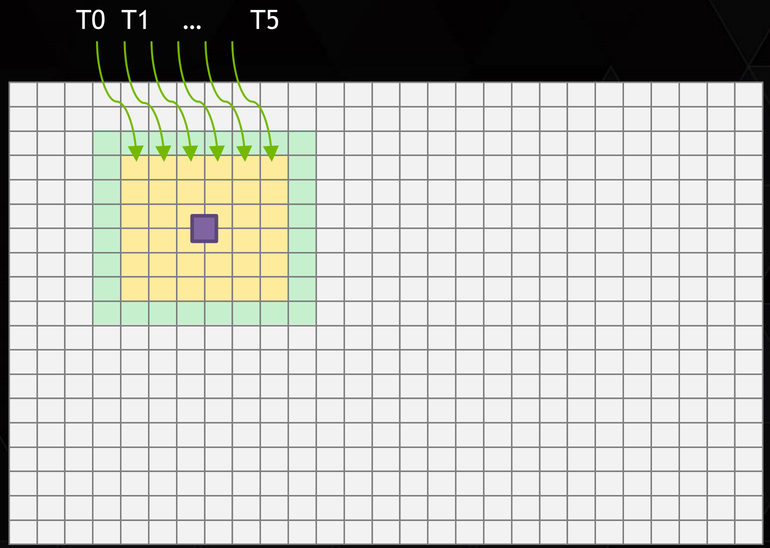
First Frame I — Second Frame I_{next}

$$\Delta p = \left[\sum_{x\in\Omega}\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\right]^{-1} \sum_{x\in\Omega}\left(I(x) - I_{next}(x+\Delta p_{prev})\right)\begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

SparseLK
- Solution#1

$$\Delta p = \left[\sum_{x\in\Omega}\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\right]^{-1} \sum_{x\in\Omega}\left(I(x) - I_{next}(x + \Delta p_{prev})\right)\begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

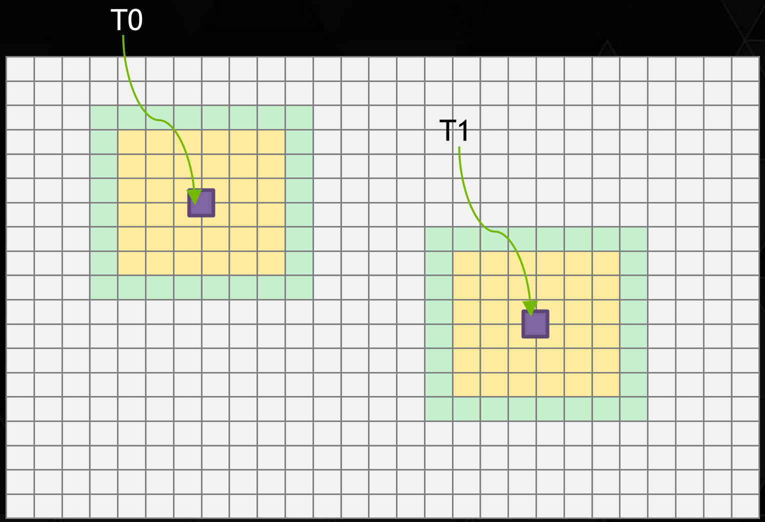
- Multiple threads for a feature point
- Share data via shared memory or shuffle
- Reduction needed to get final results
- High thread level parallelism(TLP) but more instructions needed



SparseLK
- Solution#2

$$\Delta p = \left[\sum_{x\in\Omega}\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\right]^{-1} \sum_{x\in\Omega}\left(I(x) - I_{next}\left(x + \Delta p_{prev}\right)\right)\begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

- Each thread handles a feature point
- No need to shuffle data
- No need to do reduction
- Need more registers to hold data
- High instruction level parallelism(ILP) but low occupancy



SparseLK

- Instruction# and Perf/Watt

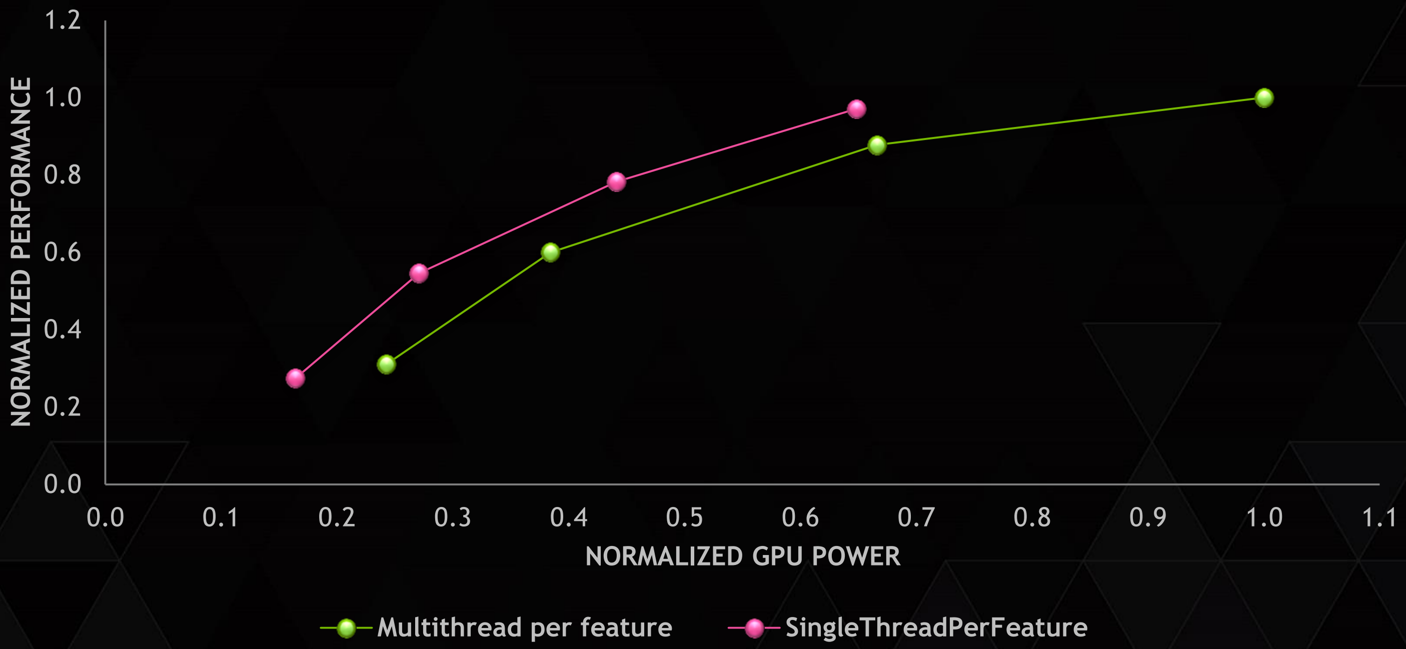
$$\frac{Perf}{Watt} = \frac{\frac{Workload}{Sec}}{\frac{Energy}{Sec}} = \frac{Workload}{Energy}$$

$$\begin{aligned} Energy = \; & EnergyPerInst_{shuffle} * Intruction_{shuffle} \\ & + EnergyPerInst_{reduction} * Instruction_{reduction} \\ & + EnergyPerInst_{other} * Instruction_{other} \\ & + Power_{wasted} * Time \end{aligned}$$

GPU TECHNOLOGY CONFERENCE

SparseLK

- Perf/Watt comparison



Summary

- Analyze the whole pipeline at the system level
- Use energy efficient features on the target platform
- Balance between TLP and ILP

THANK YOU

brantz@nvidia.com

JOIN THE CONVERSATION

#GTC15