Efficient Computing for Robotics and AI

Vivienne Sze

Massachusetts Institute of Technology

In collaboration with Luca Carlone, Yu-Hsin Chen, Joel Emer,
***Sertac Karaman**, Tushar Krishna, Trevor Henderson, Peter Li,*
Fangchang Ma, James Noraky, Amr Suleiman, Diana Wofk,
Tien-Ju Yang, Zhengdong Zhang

Contact Info
email: sze@mit.edu
website: www.rle.mit.edu/eems

Processing at "Edge" instead of the "Cloud"

Communication

Privacy

Latency

Computing Challenge for Self-Driving Cars

JACK STEWART TRANSPORTATION 02.06.18 08:00 AM

SELF-DRIVING CARS USE CRAZY AMOUNTS OF POWER, AND IT'S BECOMING A PROBLEM

Shelley, a self-driving Audi TT developed by Stanford University, uses the brains in the trunk to speed around a racetrack autonomously.

NIKKI KAHN/THE WASHINGTON POST/GETTY IMAGES

WIRED

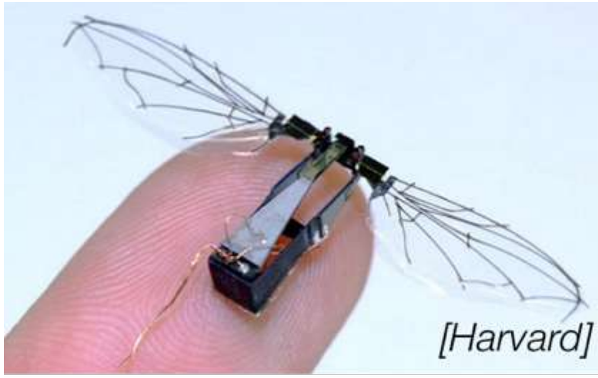
(Feb 2018)

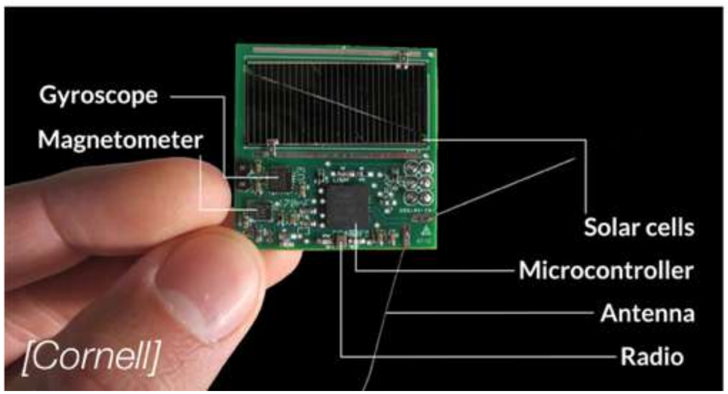
Cameras and radar generate ~6 gigabytes of data every 30 seconds.

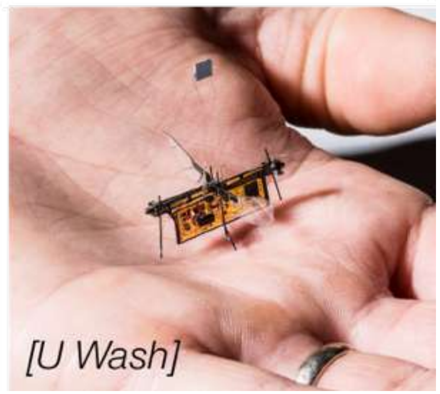
Self-driving car prototypes use approximately 2,500 Watts of computing power.

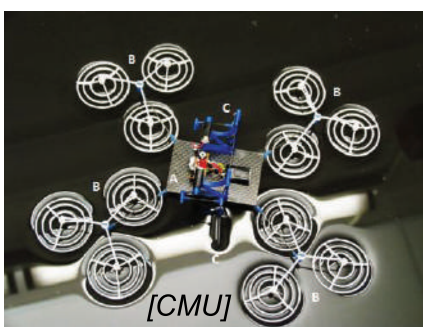
Generates wasted heat and some prototypes need water-cooling!

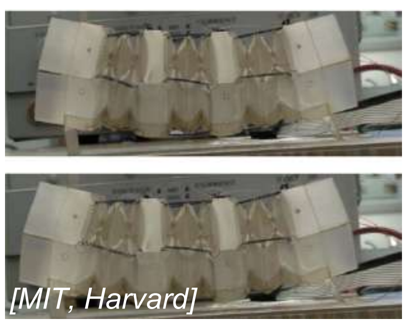
Robots Consuming < 1 Watt for Actuation

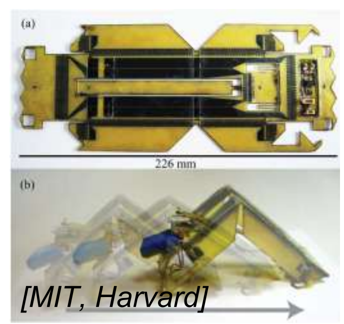








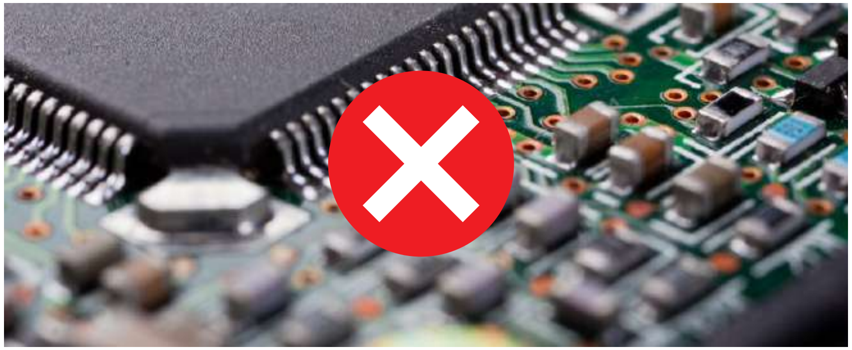




Low Energy Robotics

- Miniature aerial vehicles
- Lighter than air vehicles
- Micro unmanned gliders
- Miniature satellites

Existing Processors Consume Too Much Power



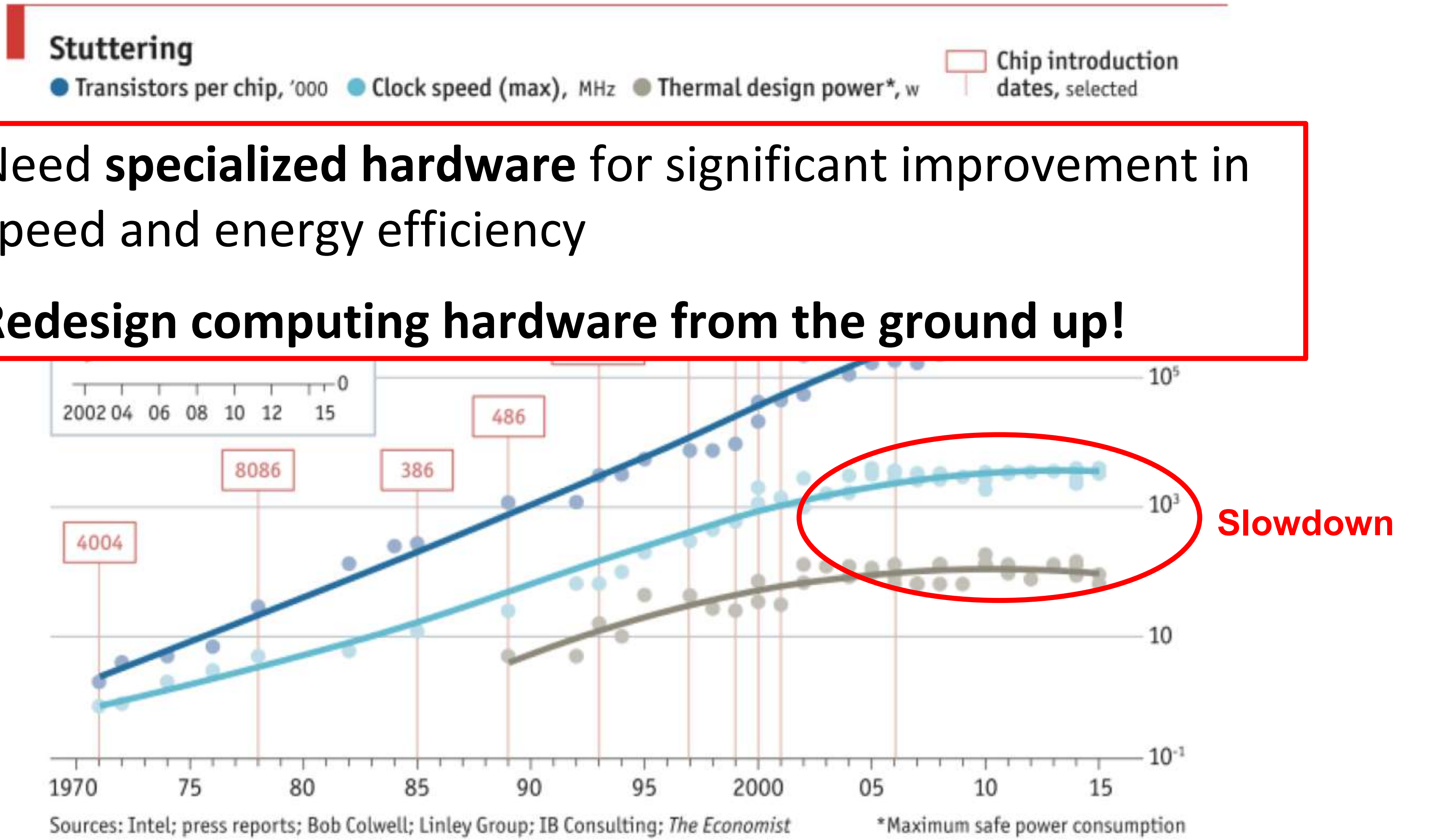
< 1 Watt

> 10 Watts

Transistors are NOT Getting More Efficient

Slow down of Moore's Law and Dennard Scaling

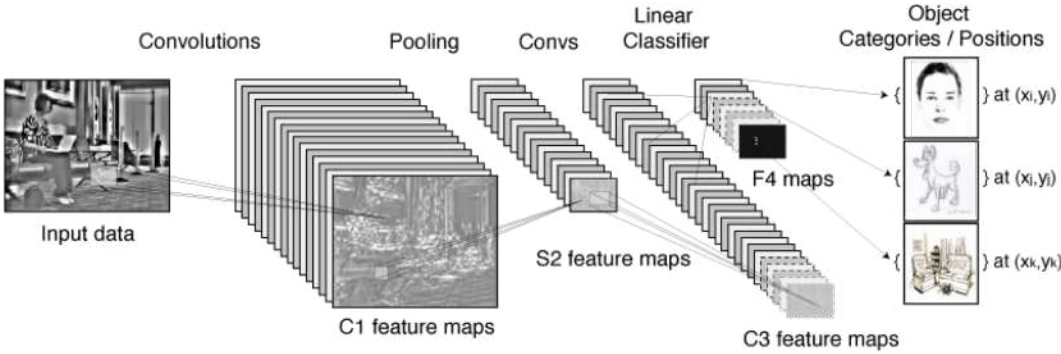
General purpose microprocessors not getting faster or more efficient



- Need **specialized hardware** for significant improvement in speed and energy efficiency
- **Redesign computing hardware from the ground up!**

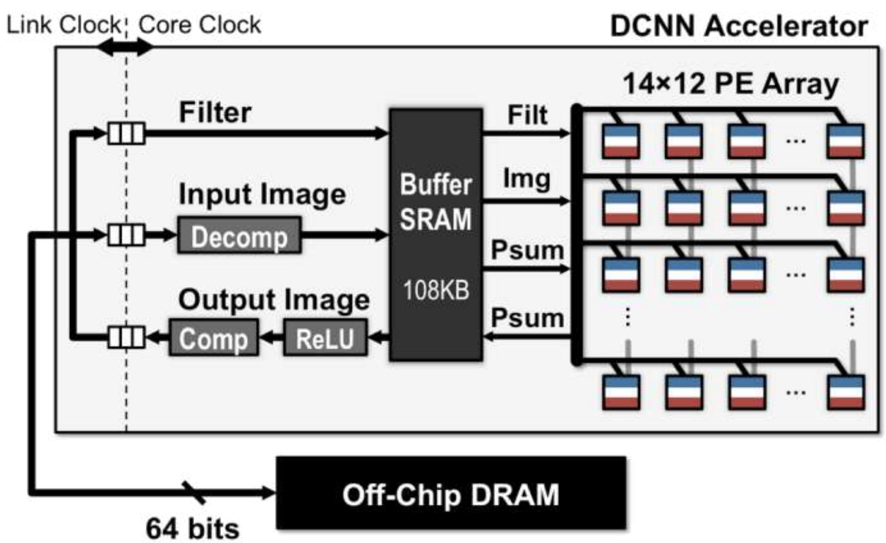
Energy-Efficient Computing with Cross-Layer Design

Algorithms

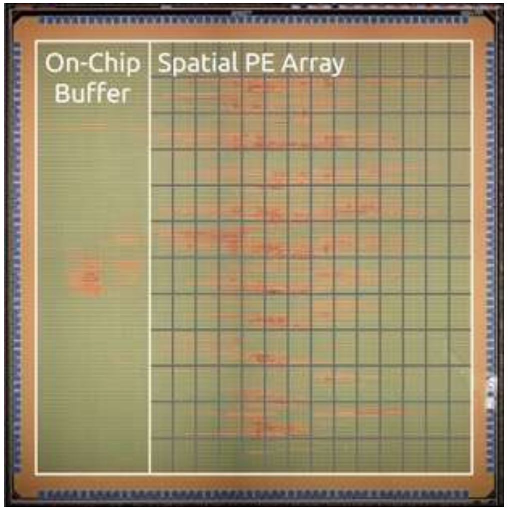


Systems

Architectures

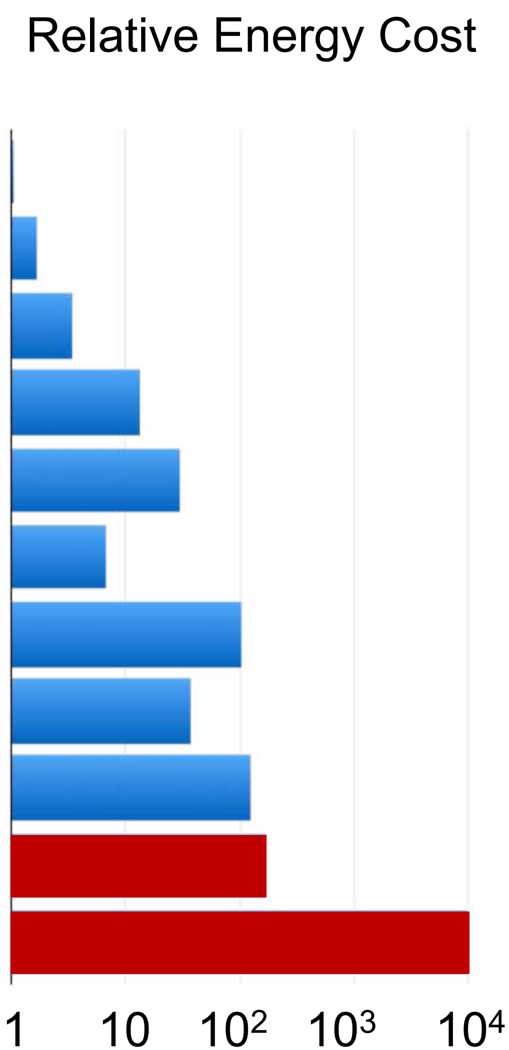


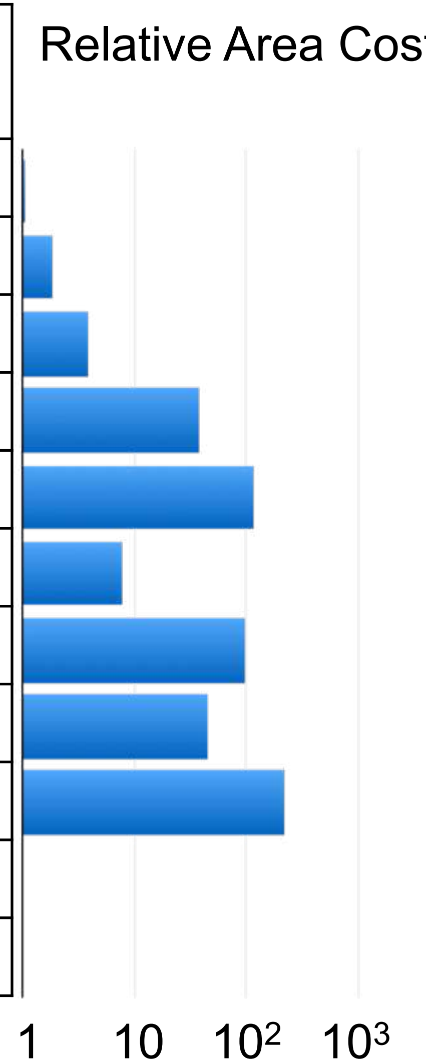
Circuits



Power Dominated by Data Movement

Operation:	Energy (pJ)	Area (μm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A





Memory access is **orders of magnitude** higher energy than compute

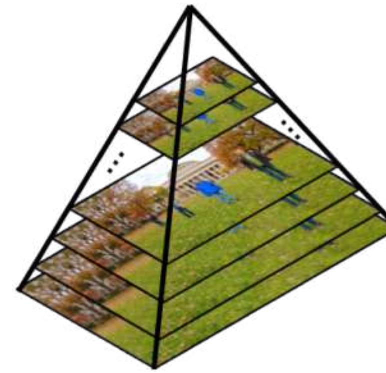
[Horowitz, ISSCC 2014]

Autonomous Navigation Uses a Lot of Data

- **Semantic Understanding**

- High frame rate
- Large resolutions
- Data expansion

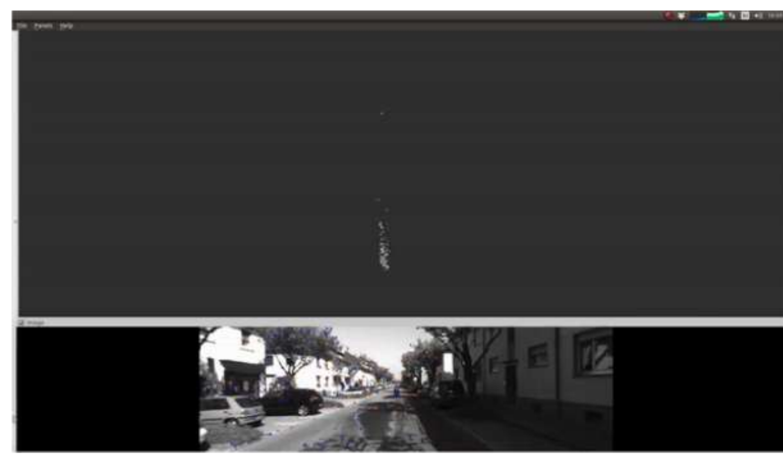
2 million pixels



10x-100x more pixels

- **Geometric Understanding**

- Growing map size



[Pire, RAS 2017]

Visual-Inertial Localization

Determines location/orientation of robot from images and IMU
(also used by headset in Augmented Reality and Virtual Reality)

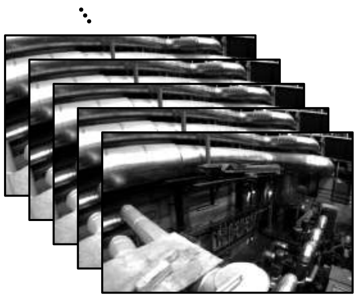
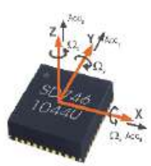


Image sequence

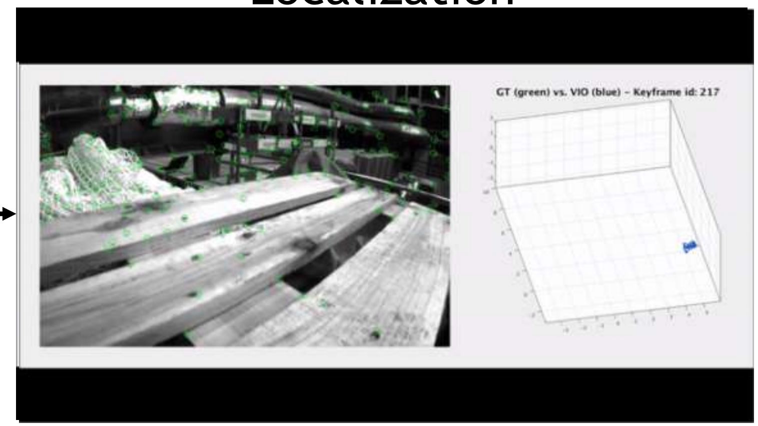
IMU

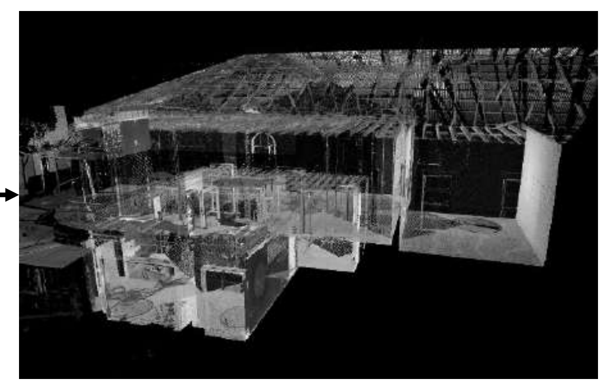
Inertial Measurement Unit



Visual-Inertial Odometry (VIO)*

Localization





Mapping

***Subset of SLAM algorithm**
(Simultaneous Localization And Mapping)

Localization at under 25 mW

First chip that performs ***complete*** Visual-Inertial Odometry

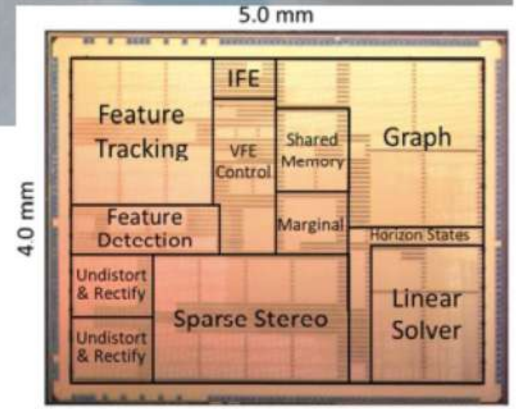
Front-End for camera
(Feature detection, tracking, and outlier elimination)

Front-End for IMU
(pre-integration of accelerometer and gyroscope data)

Back-End Optimization of Pose Graph

Consumes **684× and 1582×** less energy than mobile and desktop CPUs, respectively

Technology	65nm CMOS	Supply	1 V
Chip area (mm²)	4.0 x 5.0	Resolution	752x480
Core area (mm²)	3.54 x 4.54	Camera rate	28 - 171 fps
Logic gates	2,043 kgates	Keyframe rate	16 - 90 fps
SRAM	854KB	Average Power	24 mW
VFE Frequency	62.5 MHz	GOPS	10.5 – 59.1
BE Frequency	83.3 MHz	GFLOPS	1 – 5.7



[Zhang, RSS 2017], [Suleiman, VLSI 2018]

[Joint work with Sertac Karaman (AeroAstro)]

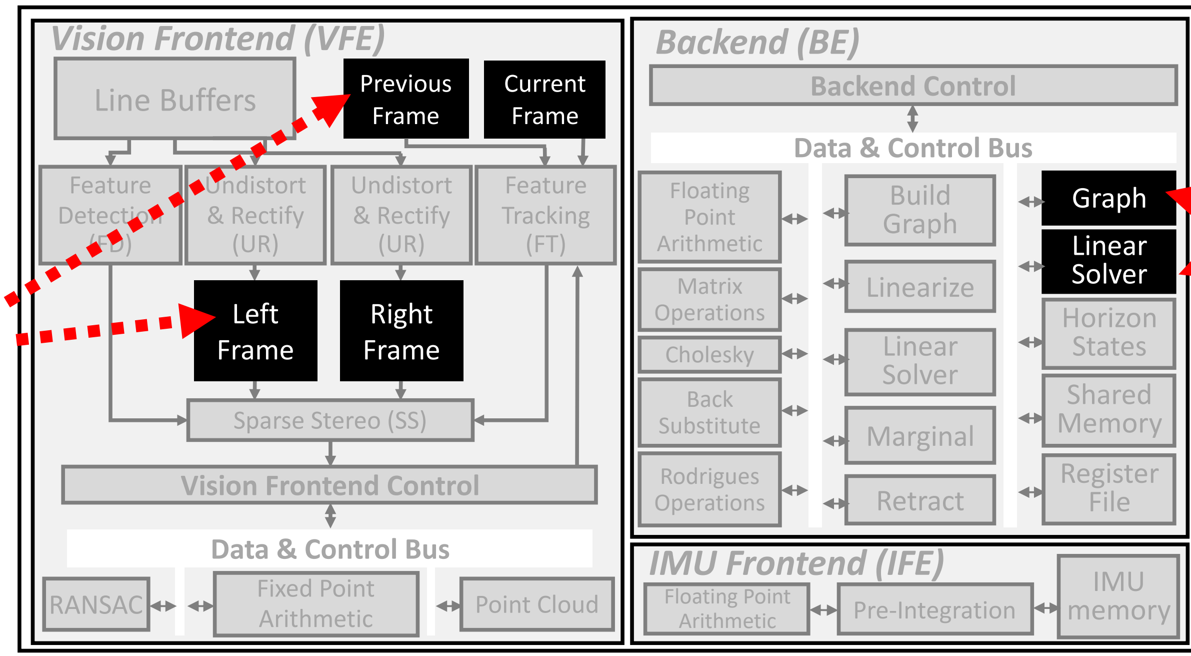
RESEARCH LABORATORY OF ELECTRONICS AT MIT

MTL microsystems technology laboratories massachusetts institute of technology

Key Methods to Reduce Data Size

Navion: *Fully integrated system – no off-chip processing or storage*

Apply Low Cost Frame Compression

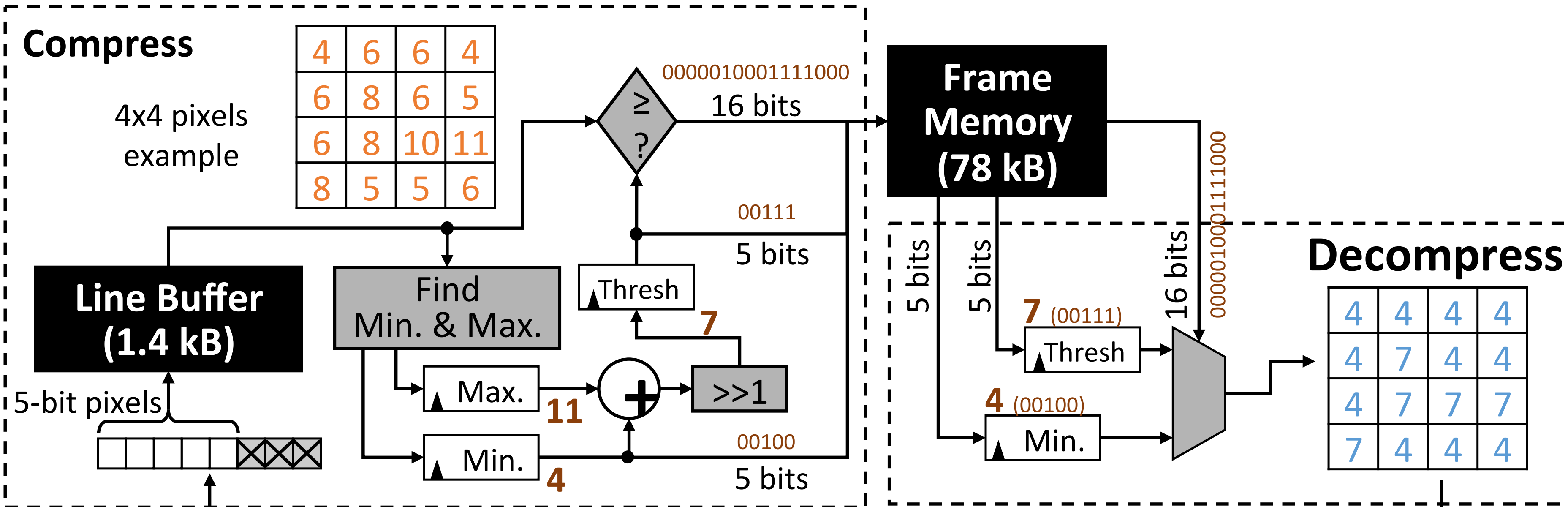


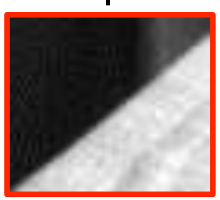
Exploit Sparsity in Graph and Linear Solver

Use **compression** and **exploit sparsity** to reduce memory down to 854kB

[Suleiman et al., VLSI 2018]

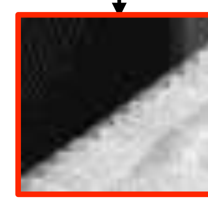
Frame Buffer Memory





8-bit/pixel

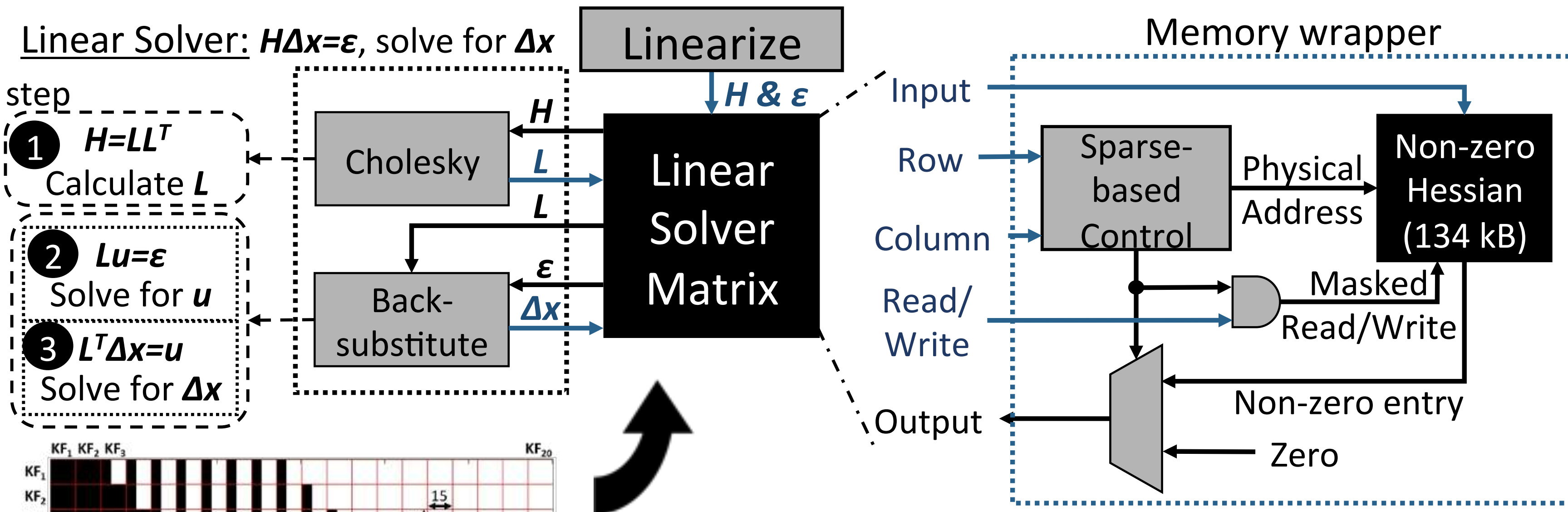
Original Image (352.5 kB) - - - - - → **4.4x reduction** **Compressed Image (79.4 kB)**

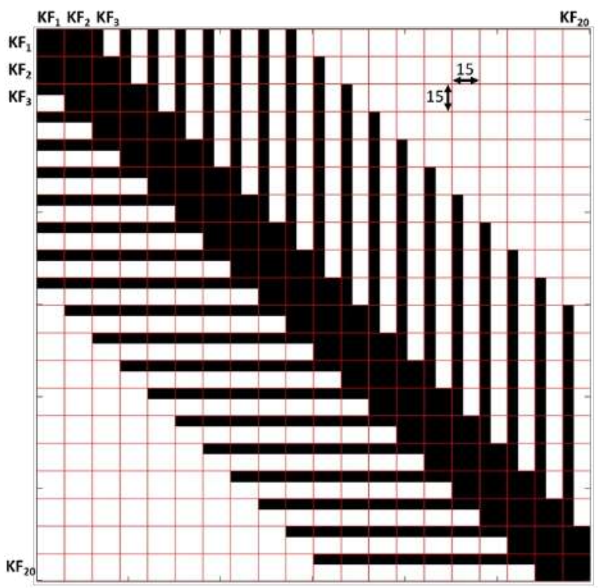


1.625-bit/pixel

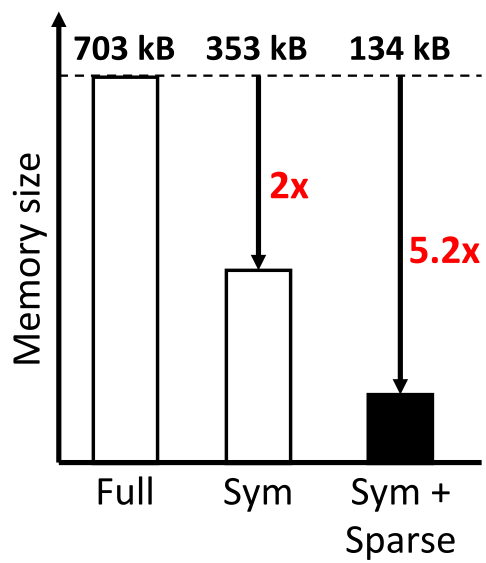
sparse and structured

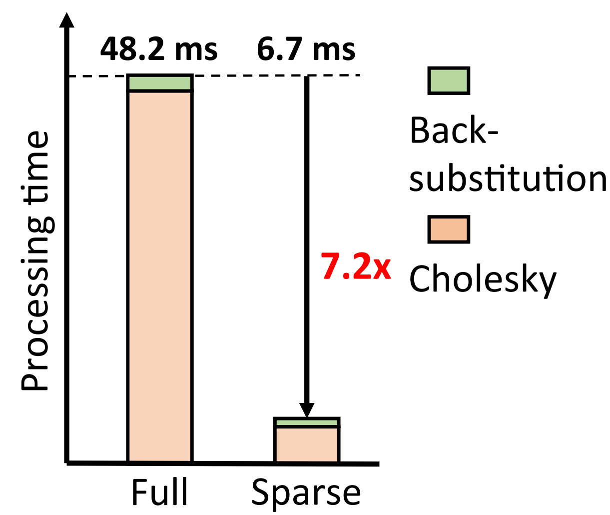
Linear Solver and Hessian Memory





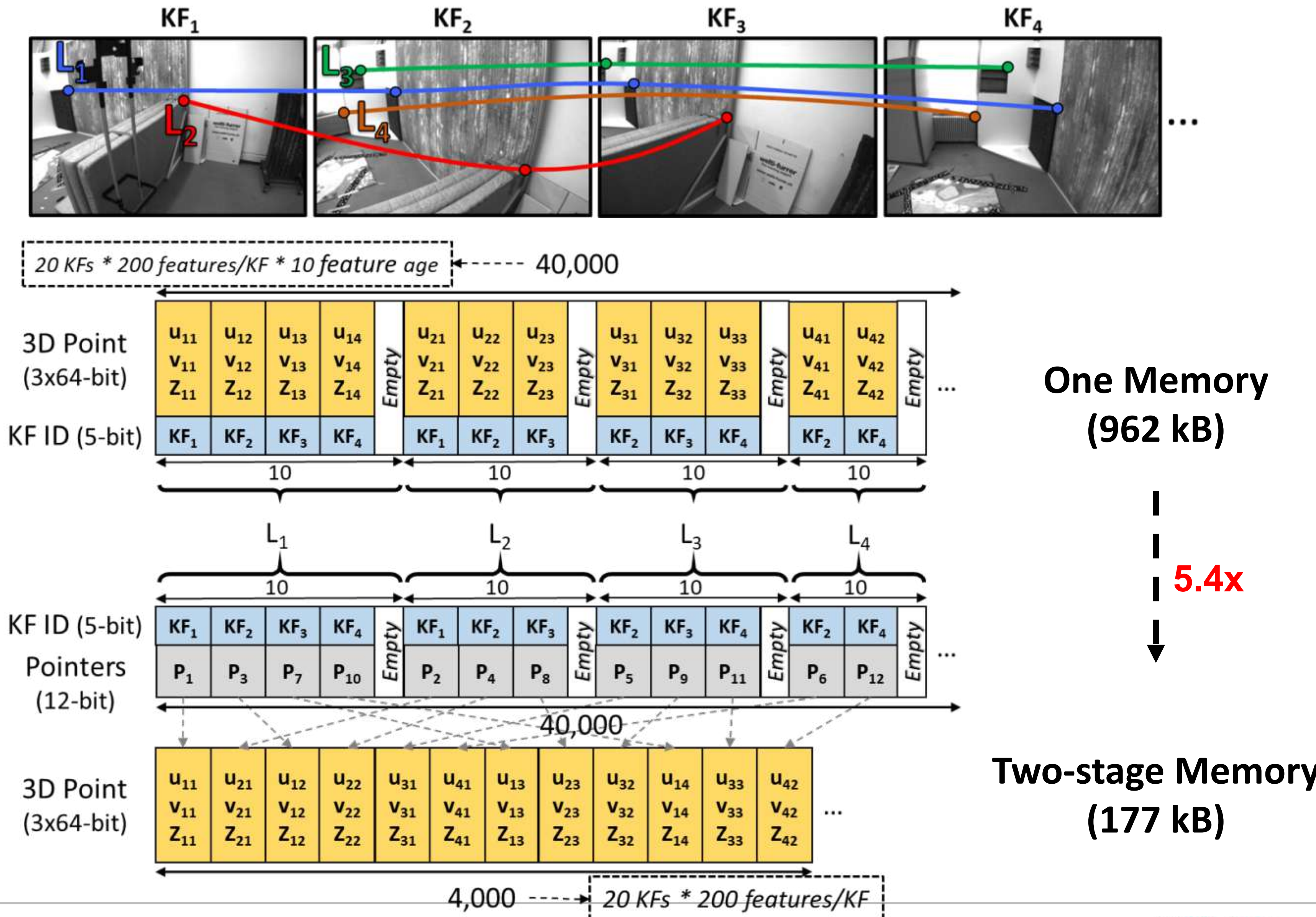
Sparsity pattern in both ***H*** & ***L***
(Non-zero: black)





sparse and unstructured

Factor Graph Memory

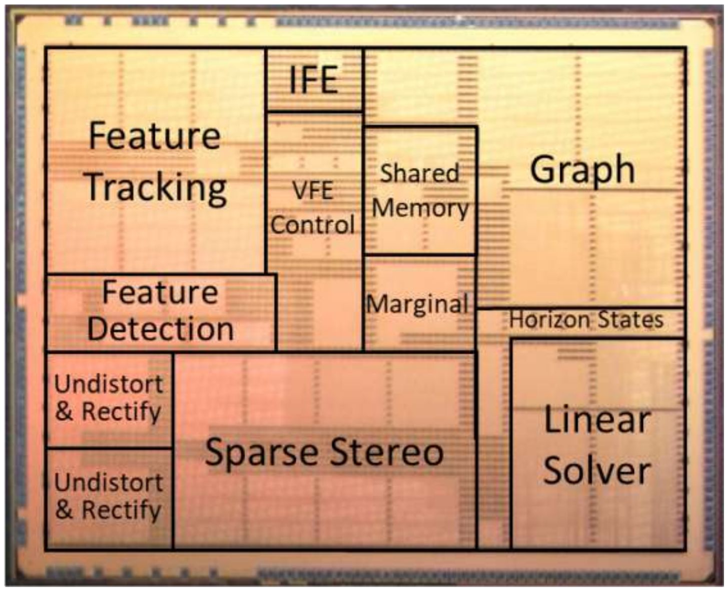


[Suleiman et al., VLSI 2018]

Navion Evaluation

5.0 mm

4.0 mm



65nm CMOS Test Chip

Over 250 configurable parameters *to adapt to different sensors and environments*

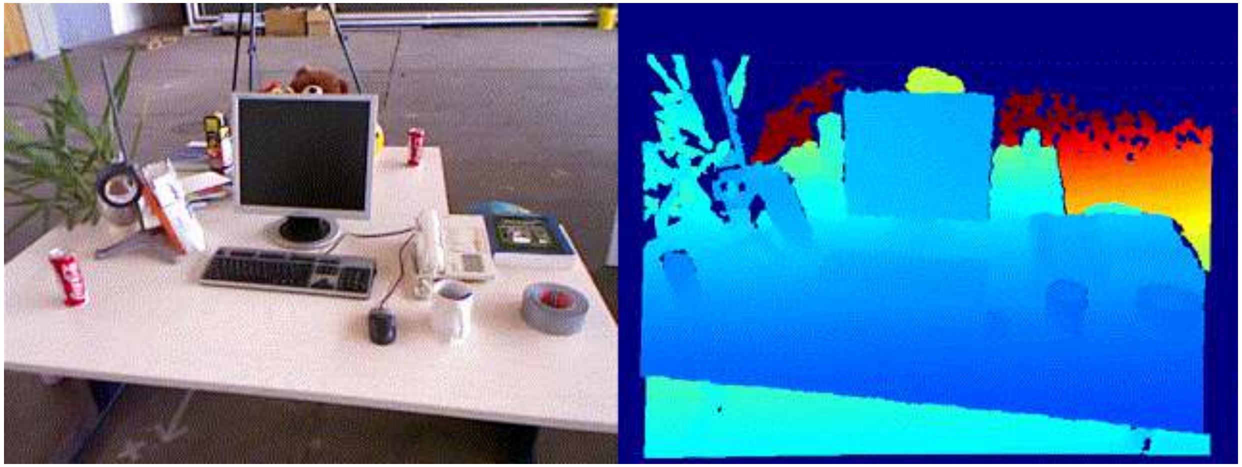
http://navion.mit.edu

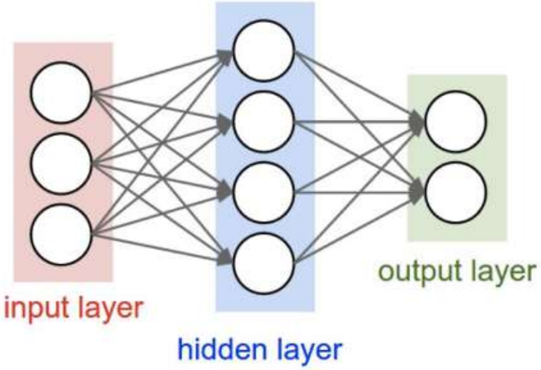
- **Peak Performance @ Maximum Configuration**
  - VFE: 28 – 171 fps (71 fps average)
  - BE: 16 – 90 fps (19 fps average)
  - Average Power Consumption: 24mW
  - Trajectory Error: 0.28%
- **Real-Time Performance @ Optimized Configuration**
  - VF: 20 fps
  - BE: 5 fps
  - Average Power Consumption: 2mW
  - Trajectory Error: 0.27%

Evaluated on EuRoC dataset

Understanding the Environment

Depth Estimation





Semantic Segmentation

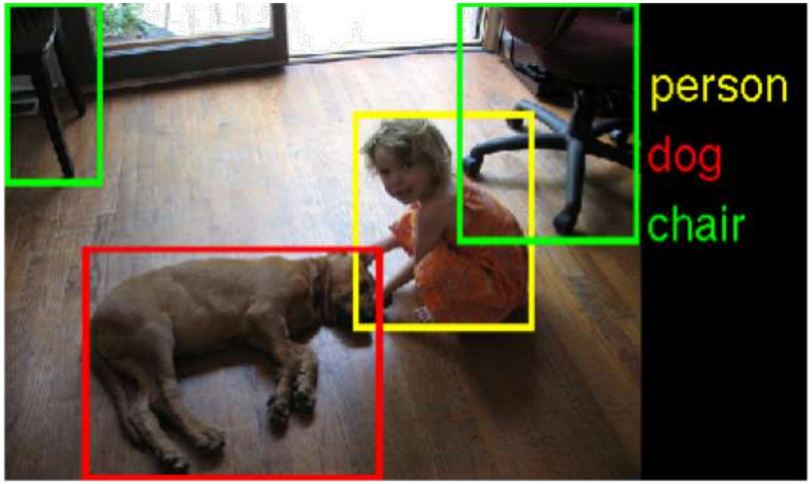
State-of-the-art approaches use **Deep Neural Networks,** which require **up to several hundred millions of operations and weights to compute!**

>100x more complex than video compression

Deep Neural Networks

Deep Neural Networks (DNNs) have become a ***cornerstone of AI***

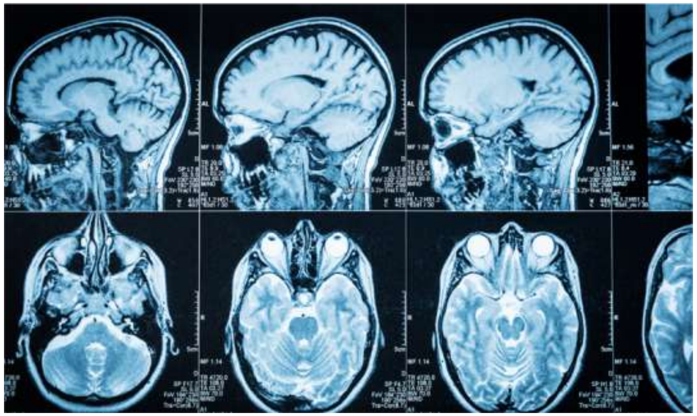
Computer Vision



Speech Recognition

Game Play

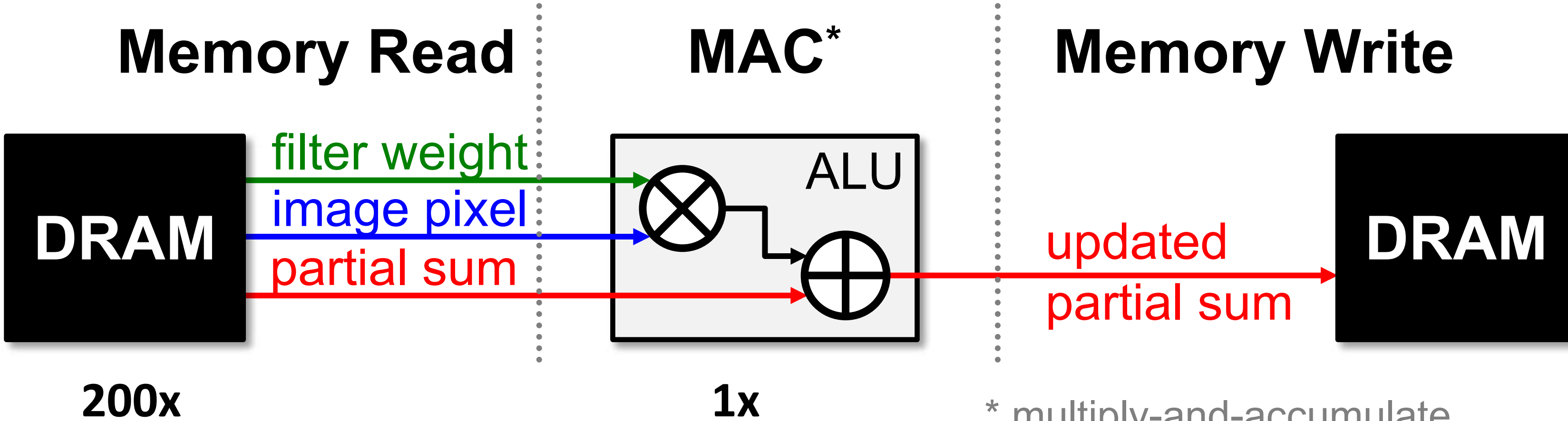
Medical



Properties We Can Leverage

- Operations exhibit **high parallelism**
  → **high throughput** possible

- Memory Access is the Bottleneck



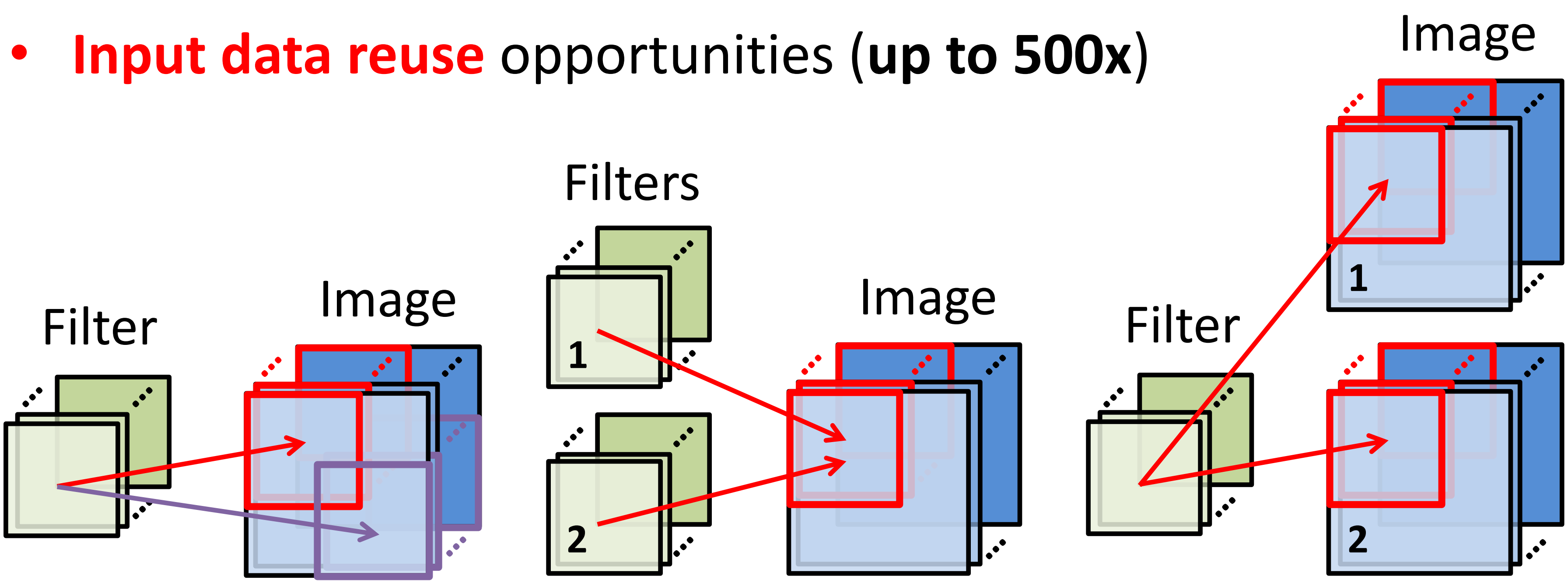
* multiply-and-accumulate

<u>Worst Case</u>: all memory R/W are **DRAM** accesses

- Example: AlexNet has **724M** MACs
  → **2896M** DRAM accesses required

Properties We Can Leverage

- Operations exhibit **high parallelism**
  → **high throughput** possible
- **Input data reuse** opportunities (**up to 500x**)

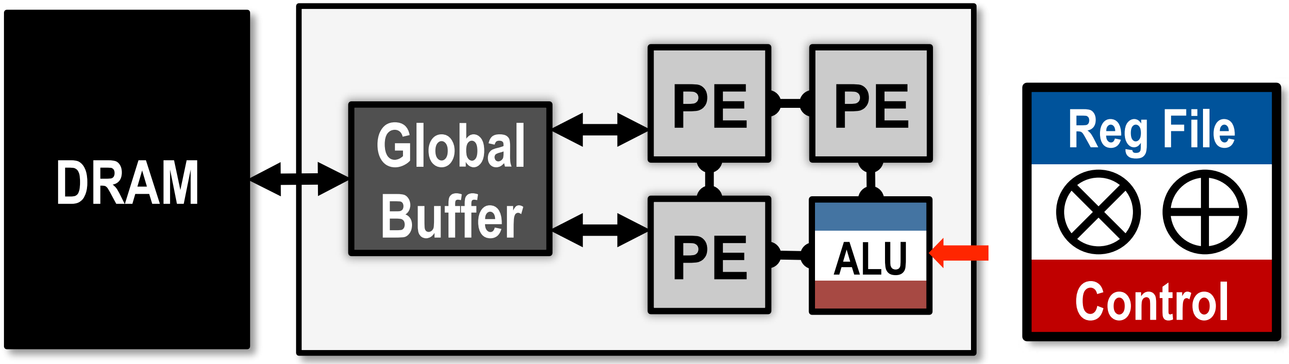


Convolutional Reuse (pixels, weights)

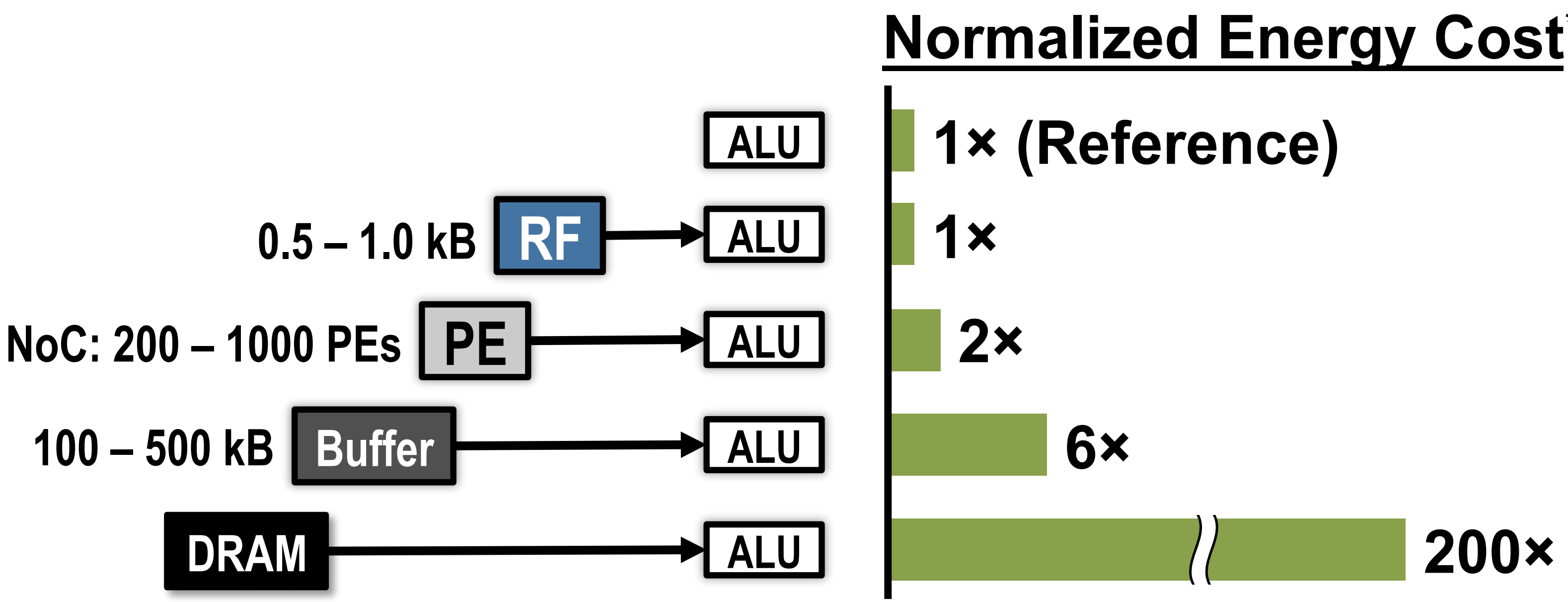
Image Reuse (pixels)

Filter Reuse (weights)

Exploit Data Reuse at Low-Cost Memories



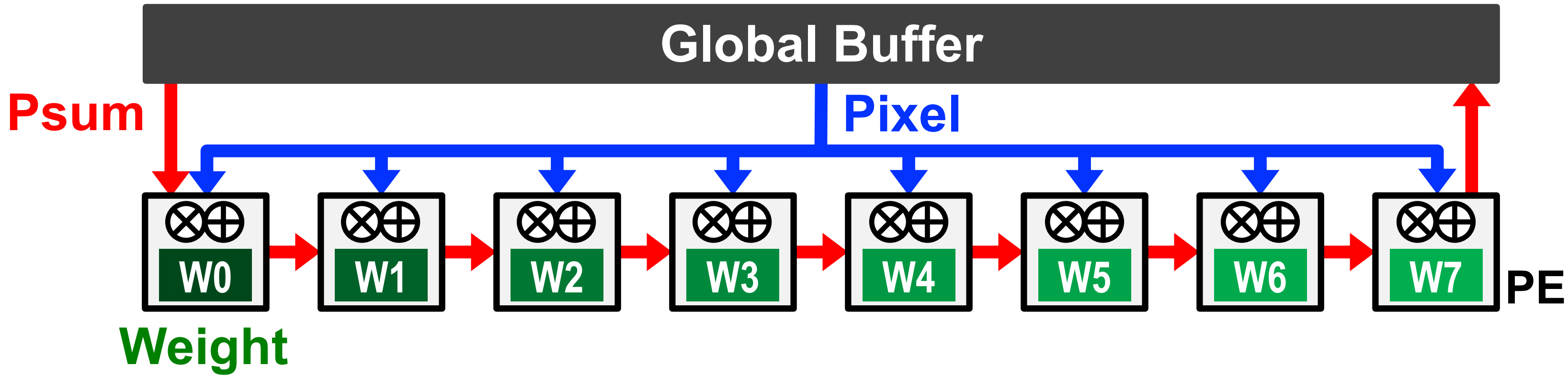
Specialized hardware with small (< 1kB) low cost memory near compute



* measured from a commercial 65nm process

Farther and **larger** memories consume more power

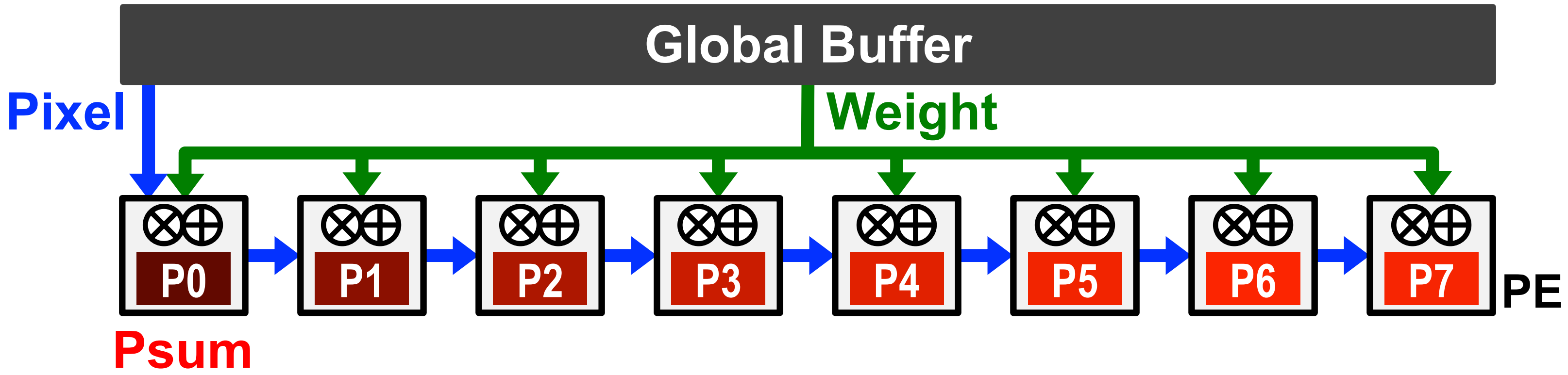
Weight Stationary (WS)



- **Minimize** **weight** read energy consumption
  - maximize convolutional and filter reuse of weights
- **Examples:**

[**Chakradhar**, *ISCA* 2010] [**nn-X (NeuFlow)**, *CVPRW* 2014]
[**Park**, *ISSCC* 2015] [**Origami**, *GLSVLSI* 2015]

Output Stationary (OS)



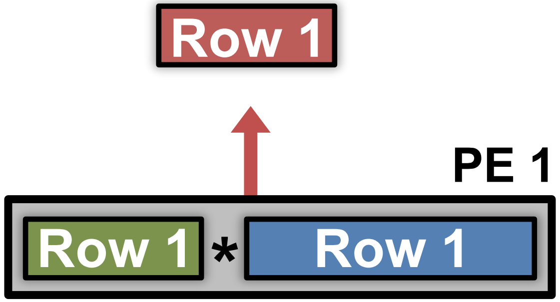
- **Minimize** **partial sum** R/W energy consumption
  - maximize local accumulation

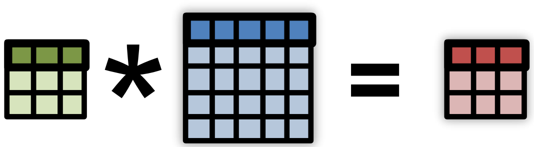
- **Examples:**

  [**Gupta**, *ICML* 2015] [**ShiDianNao**, *ISCA* 2015]

  [**Peemen**, *ICCD* 2013]

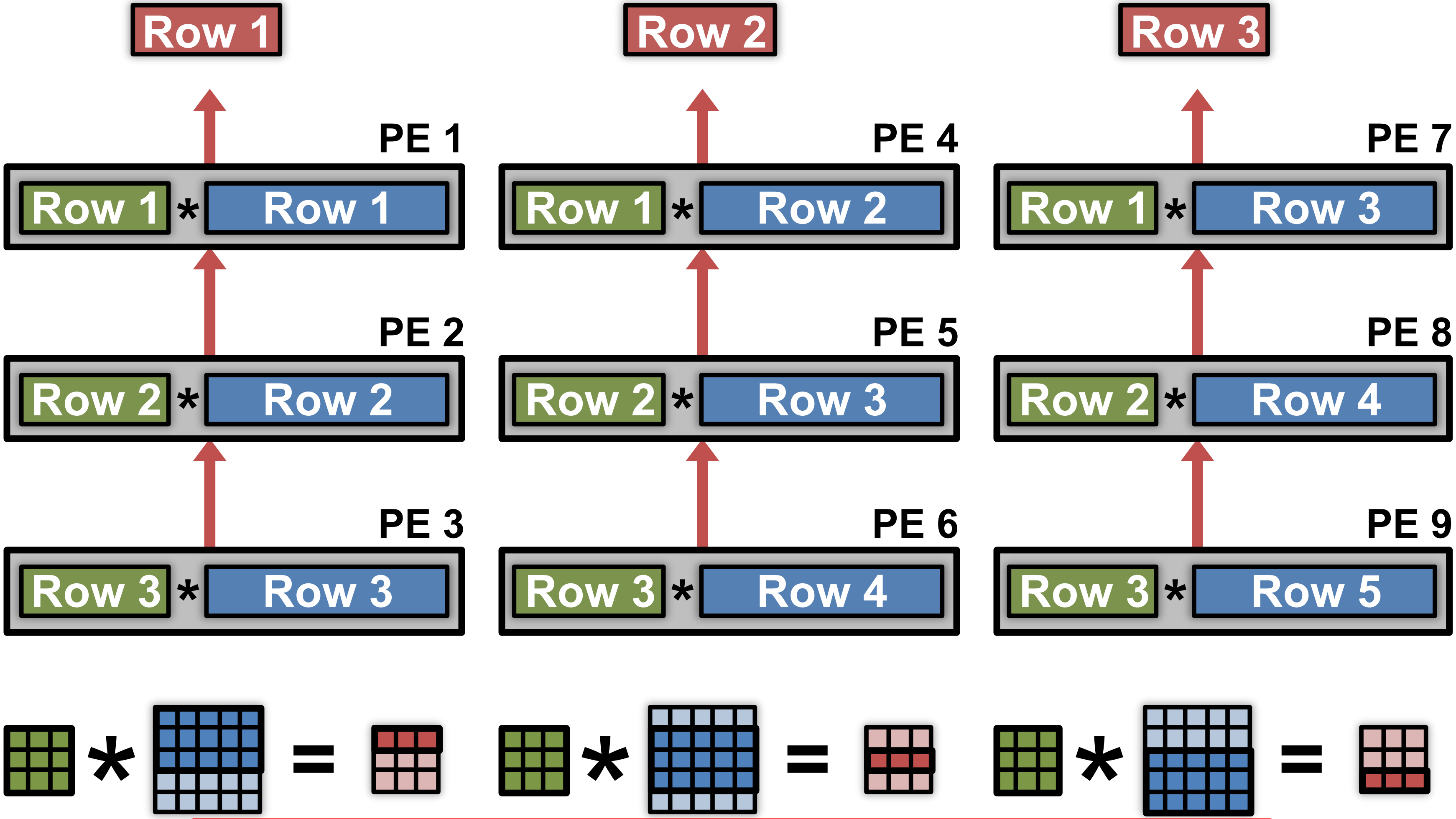
Row Stationary Dataflow





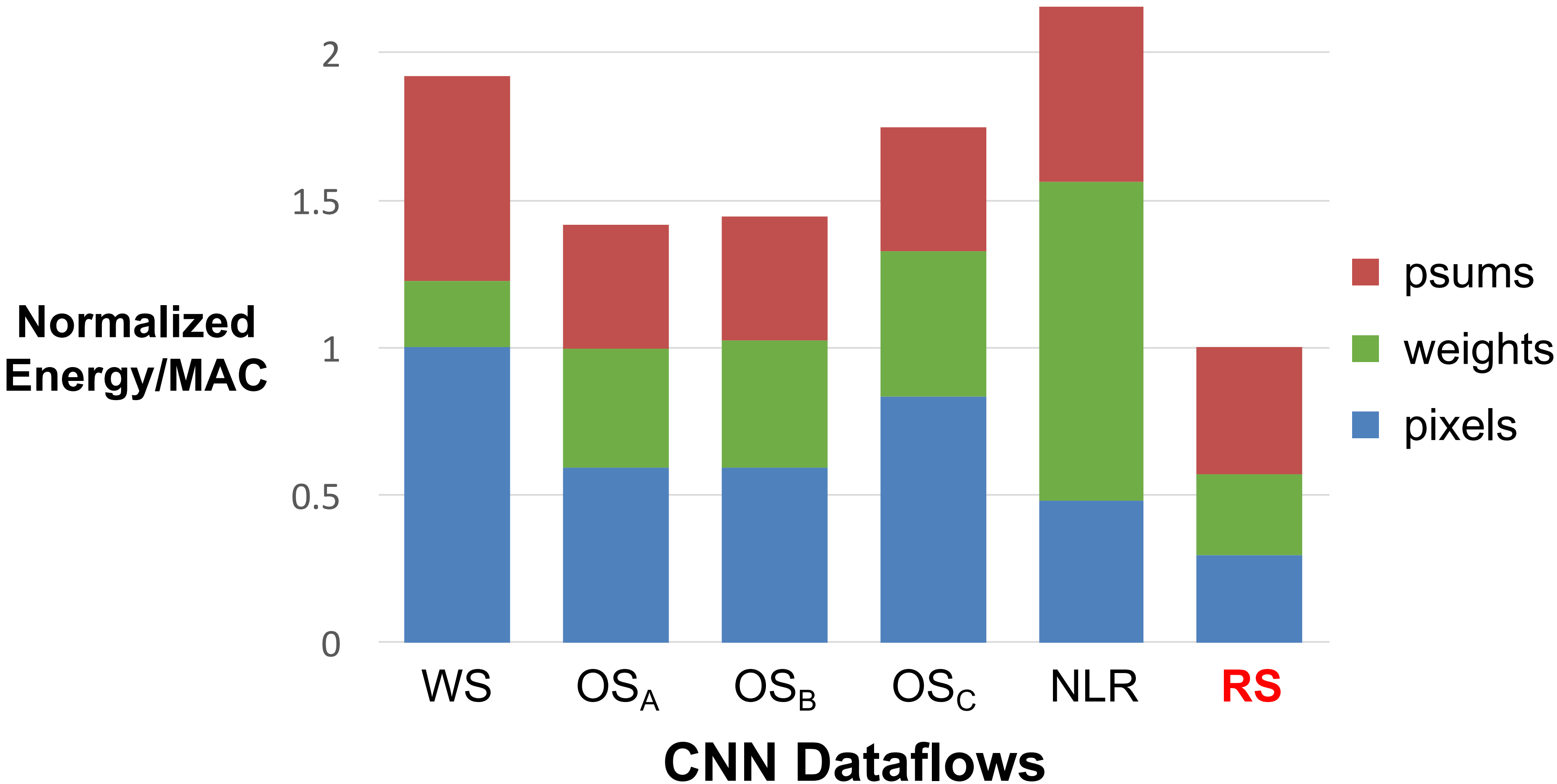
- Maximize row **convolutional reuse** in RF
  - Keep a **filter** row and **fmap** sliding window in RF
- Maximize row **psum accumulation** in RF

Row Stationary Dataflow



Optimize for **overall energy efficiency** instead for only a certain data type

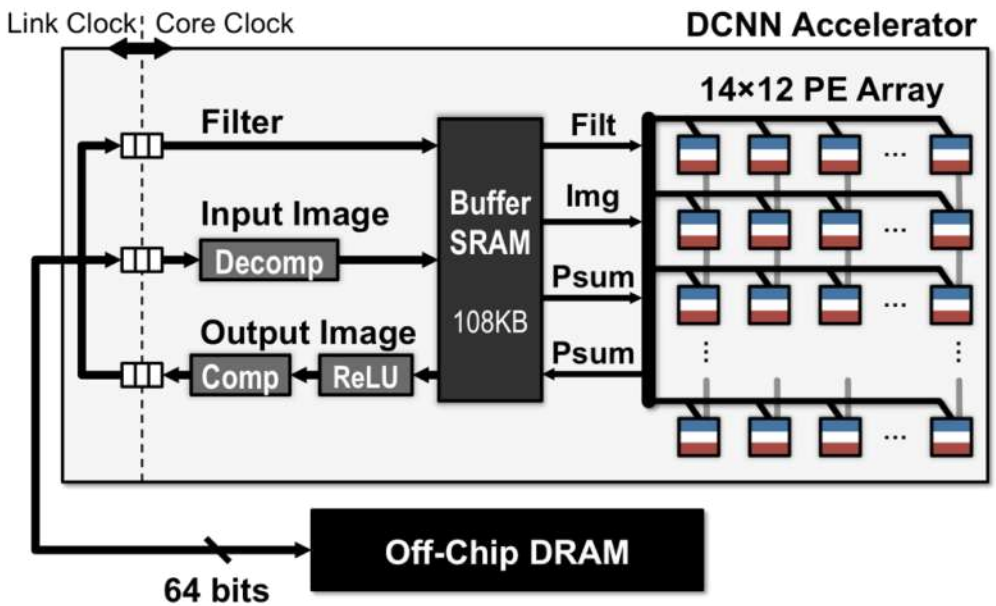
Dataflow Comparison: CONV Layers

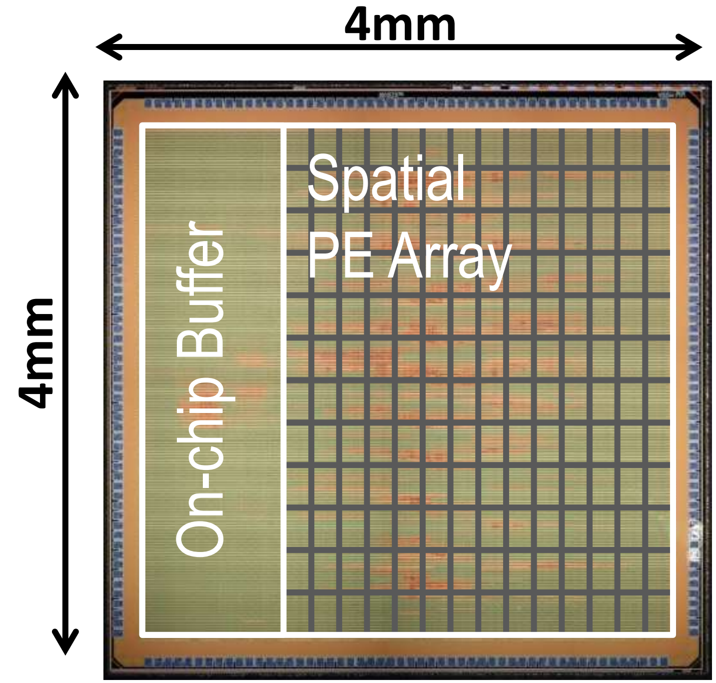


RS optimizes for the best **overall** energy efficiency

Deep Neural Networks at Under 0.3W

Eyeriss





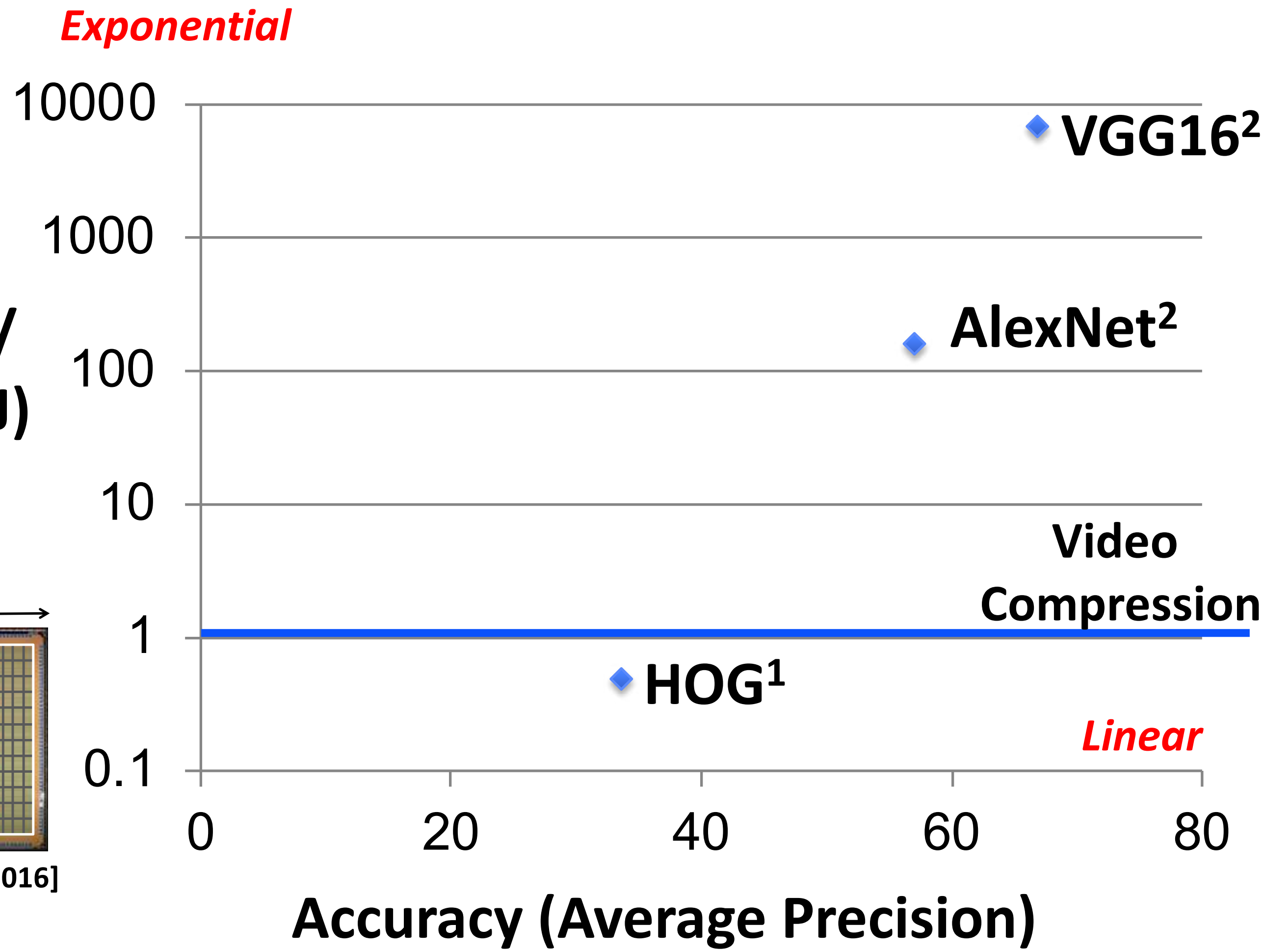
[Chen et al., ISSCC 2016, ISCA 2016]

Exploits data reuse for **100x** reduction in memory accesses from global buffer and **1400x** reduction in memory accesses from off-chip DRAM

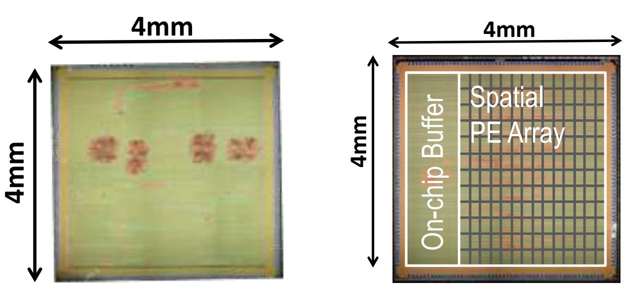
Overall **>10x energy reduction** compared to a mobile GPU (Nvidia TK1)

Results for AlexNet

Features: Energy vs. Accuracy



*Measured in 65nm**



❶ [Suleiman, VLSI 2016] ❷ [Chen, ISSCC 2016]

** Only feature extraction. Does not include data, classification energy, augmentation and ensemble, etc.*

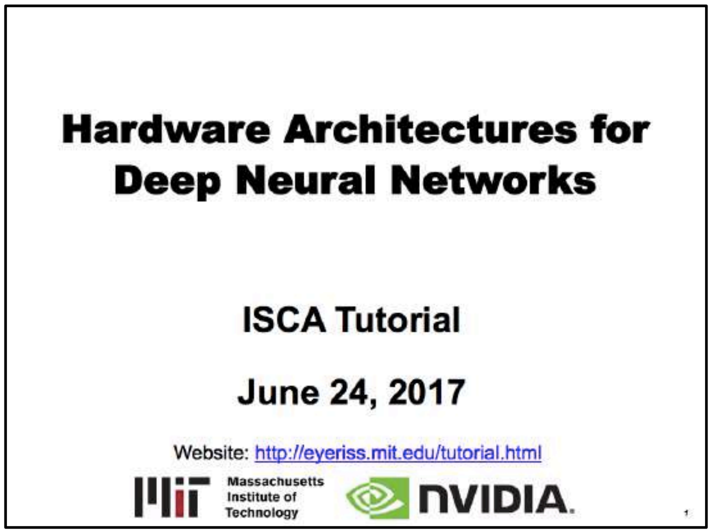
Measured in on VOC 2007 Dataset

1. DPM v5 [Girshick, 2012]
2. Fast R-CNN [Girshick, CVPR 2015]

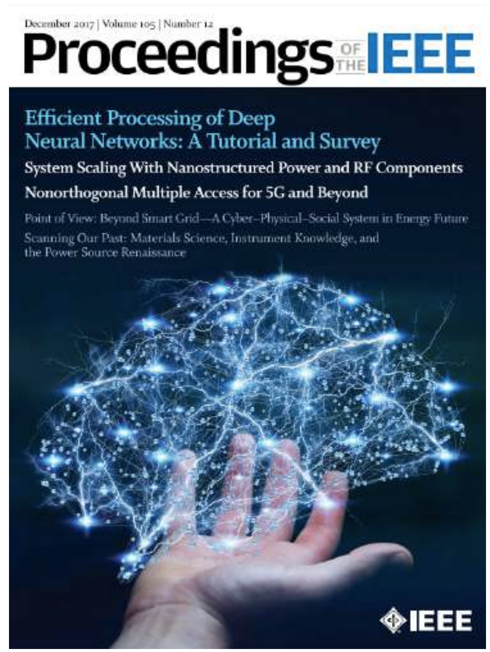
[Suleiman et al., ISCAS 2017]

Energy-Efficient Processing of DNNs

A significant amount of algorithm and hardware research on energy-efficient processing of DNNs



http://eyeriss.mit.edu/tutorial.html



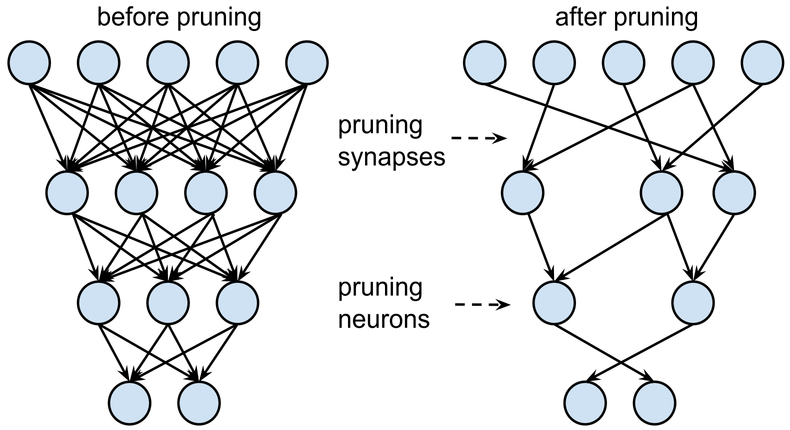
V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, "***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***," Proceedings of the IEEE, Dec. 2017

We identified various limitations to existing approaches

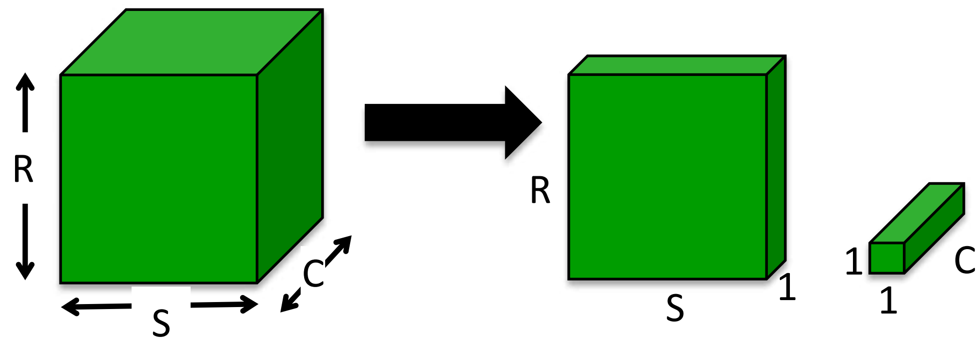
Design of Efficient DNN Algorithms

- Popular efficient DNN algorithm approaches

Network Pruning



Compact Network Architectures

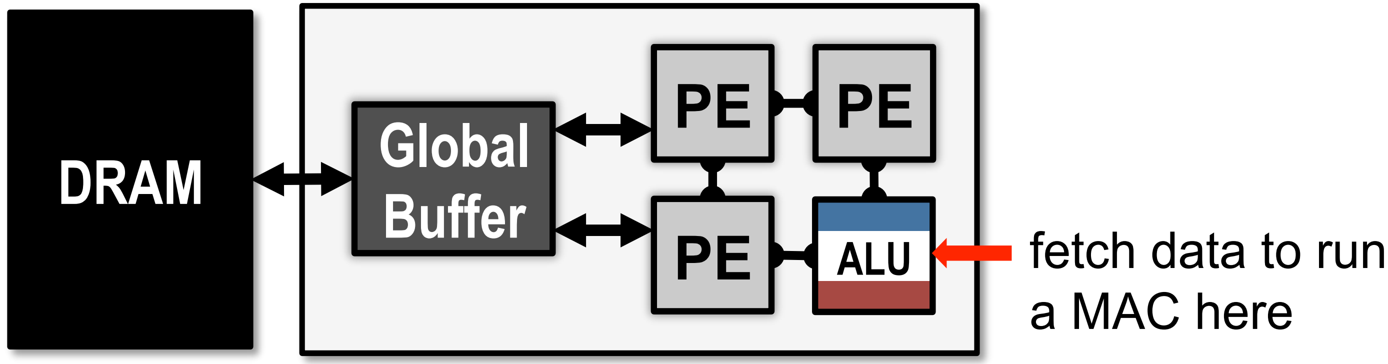


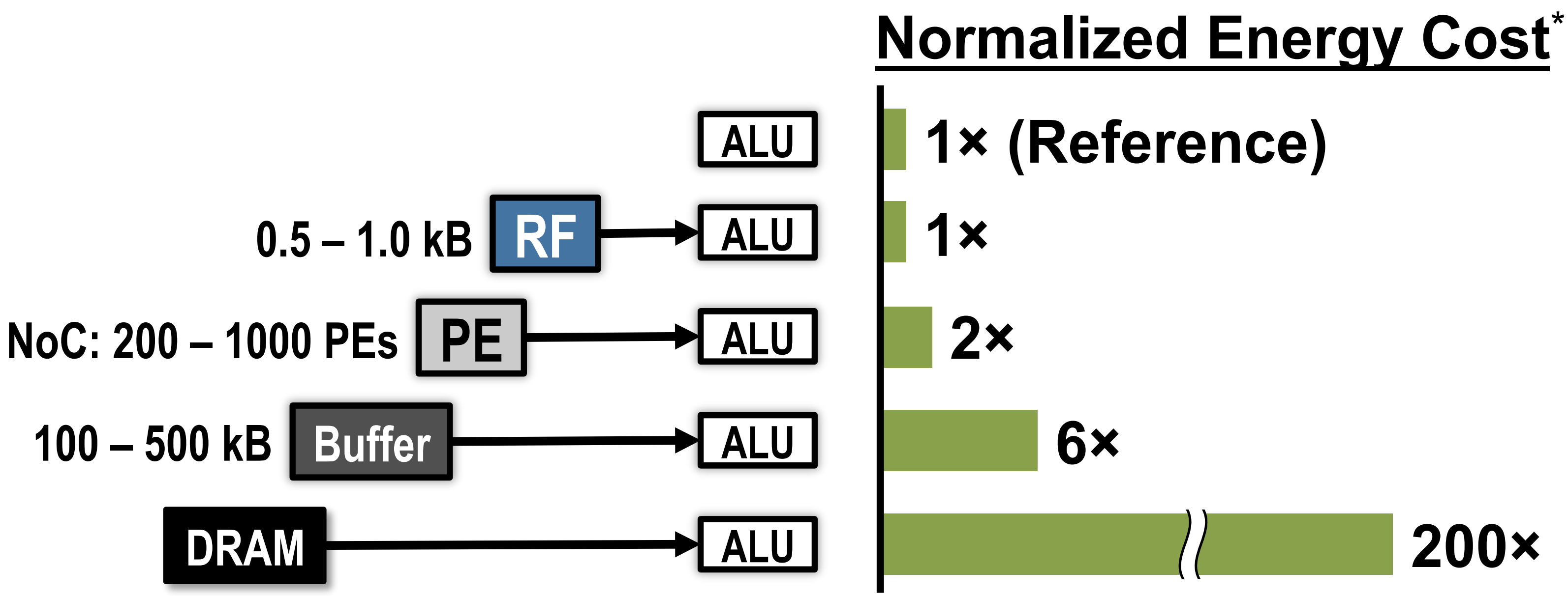
Examples: SqueezeNet, MobileNet

... also reduced precision

- Focus on reducing **number of MACs and weights**
- **Does it translate to energy savings?**

Data Movement is Expensive

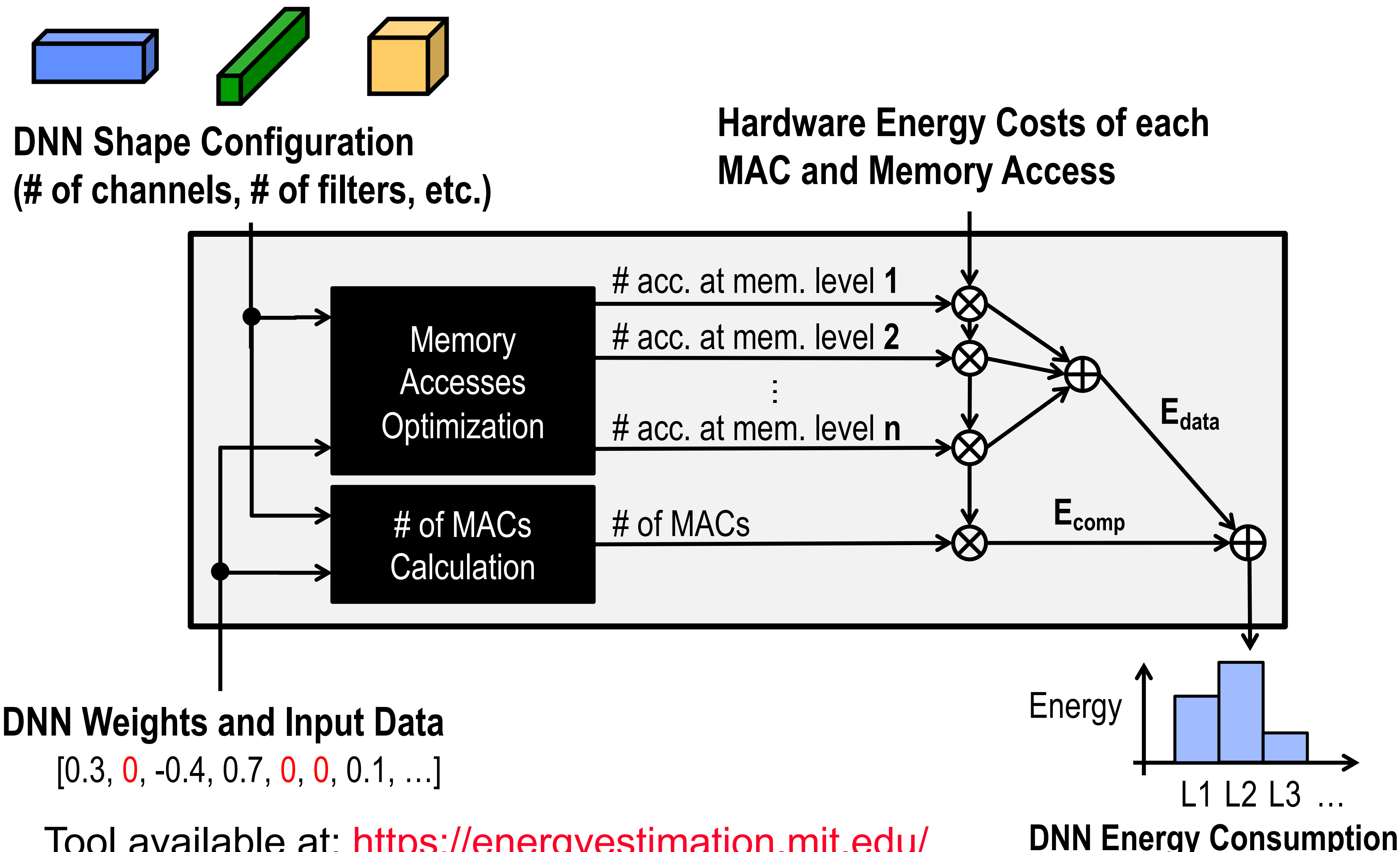




* measured from a commercial 65nm process

Energy of weight depends on **memory hierarchy** and **dataflow**

Energy-Evaluation Methodology

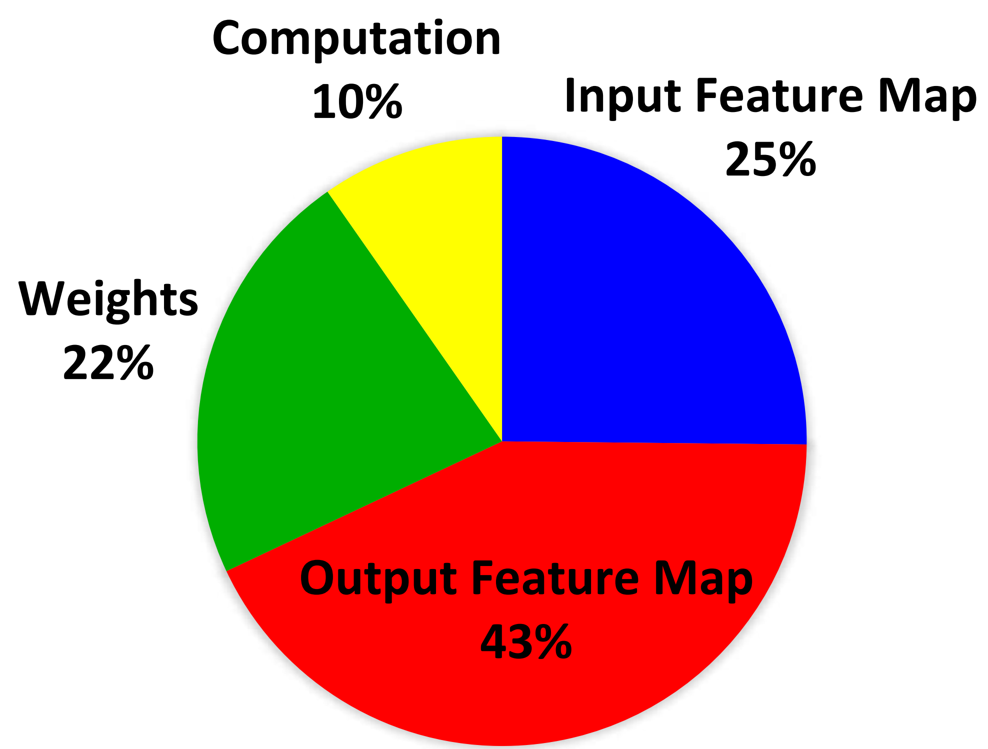


Tool available at: https://energyestimation.mit.edu/

Key Observations

- Number of weights ***alone*** is not a good metric for energy
- **All data types** should be considered

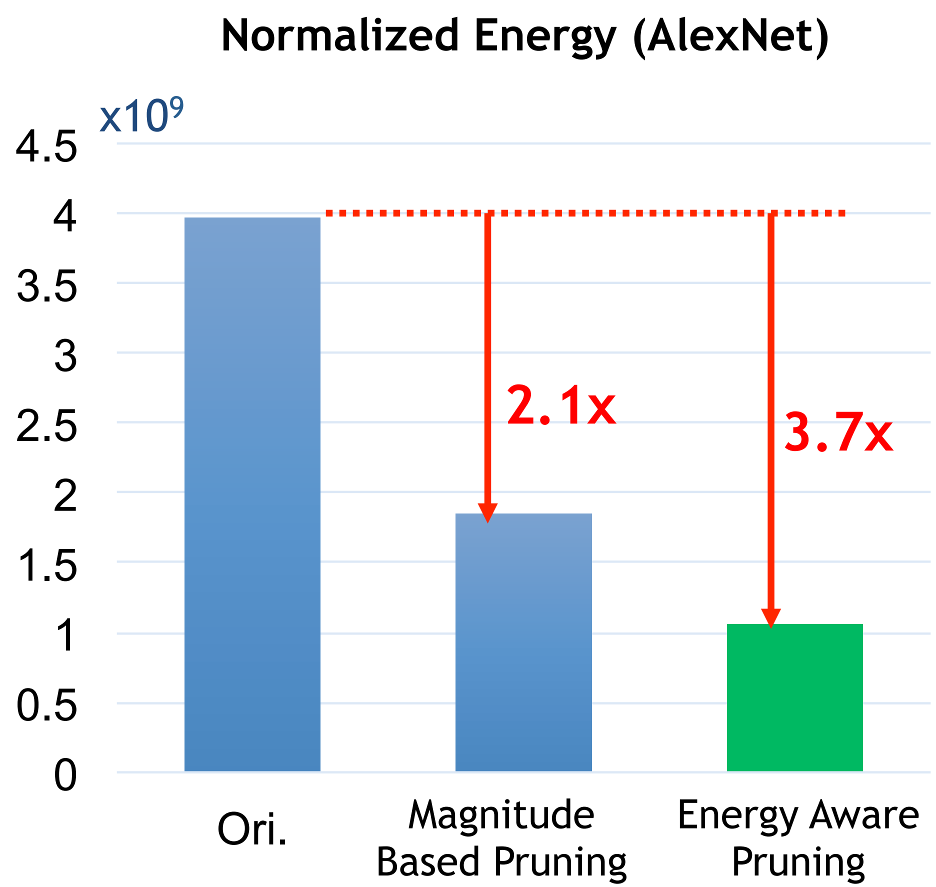
Energy Consumption of GoogLeNet



Energy-Aware Pruning

Directly target energy and incorporate it into the optimization of DNNs to provide greater energy savings

- Sort layers based on energy and prune layers that consume most energy first
- EAP reduces AlexNet energy by **3.7x** and outperforms the previous work that uses magnitude-based pruning by **1.7x**

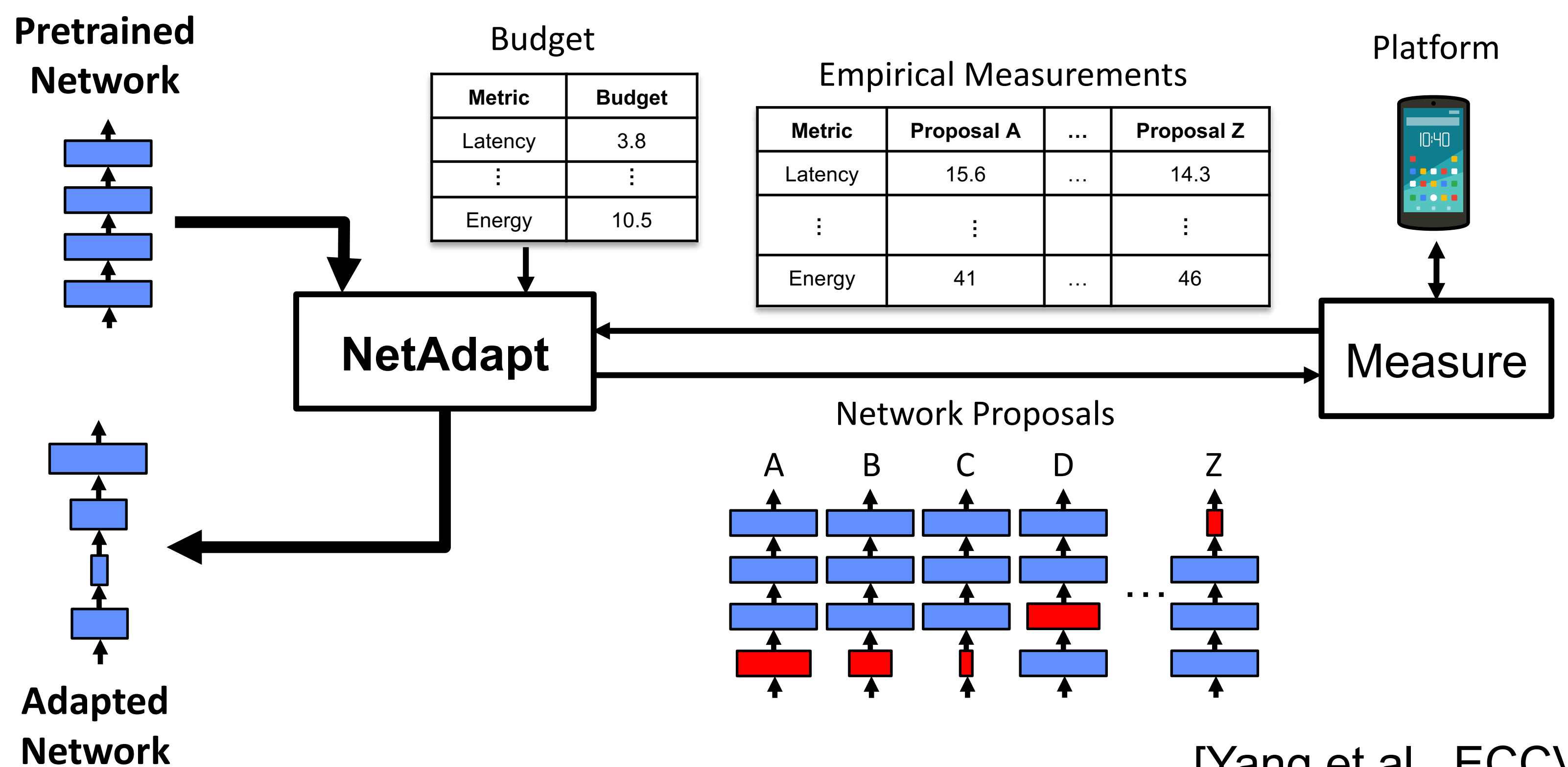


Pruned models available at
http://eyeriss.mit.edu/energy.html

[Yang et al., CVPR 2017]

NetAdapt: Platform-Aware DNN Adaptation

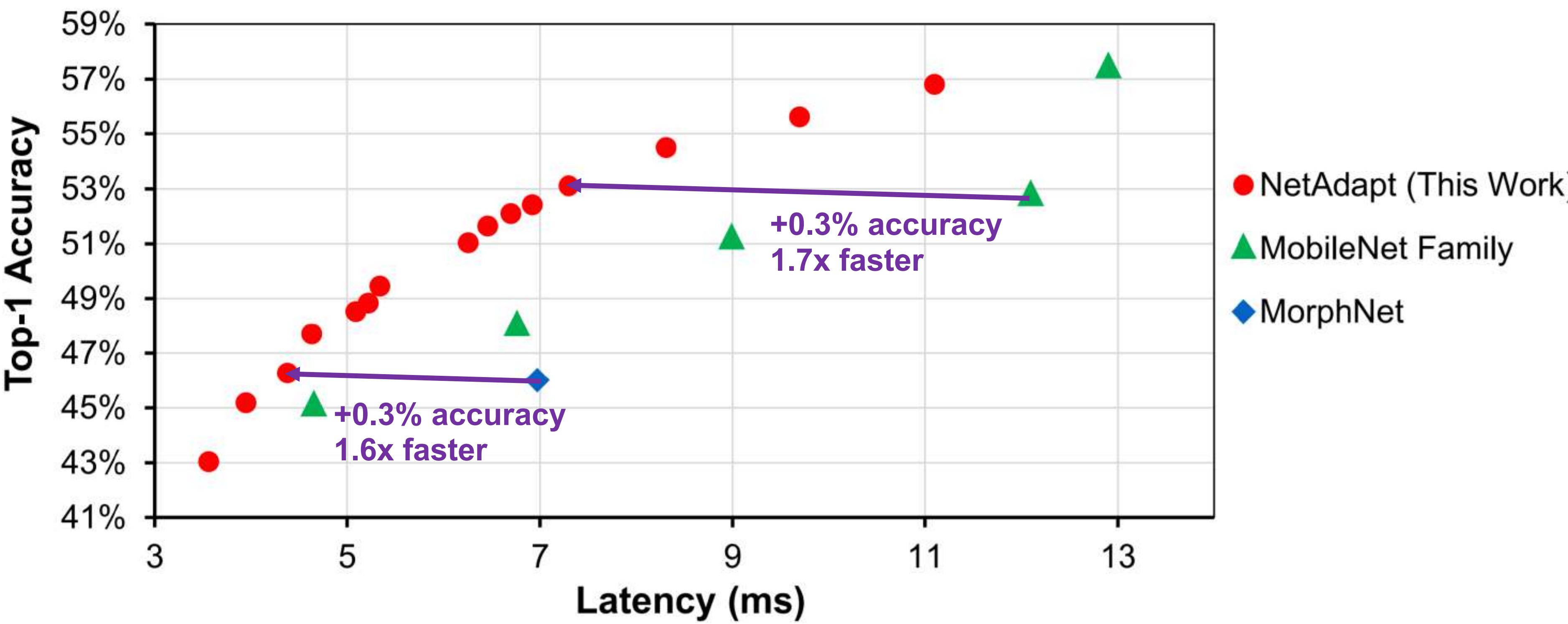
- **Automatically adapt DNN** to a mobile platform to reach a target latency or energy budget
- Use **empirical measurements** to guide optimization (avoid modeling of tool chain or platform architecture)



[Yang et al., ECCV 2018]

Improved Latency vs. Accuracy Tradeoff

- NetAdapt boosts **the real inference speed** of MobileNet by up to 1.7x with higher accuracy



*Tested on the ImageNet dataset and a Google Pixel 1 CPU

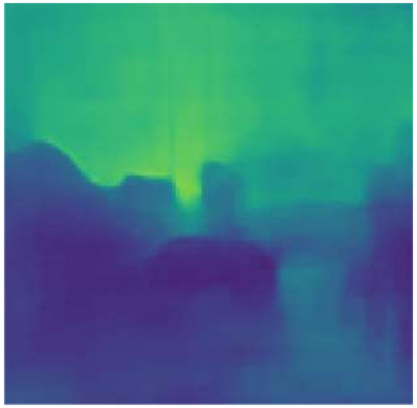
Reference:
MobileNet: Howard et al, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", arXiv 2017
MorphNet: Gordon et al., "Morphnet: Fast & simple resource-constrained structure learning of deep networks", CVPR 2018

FastDepth: Fast Monocular Depth Estimation

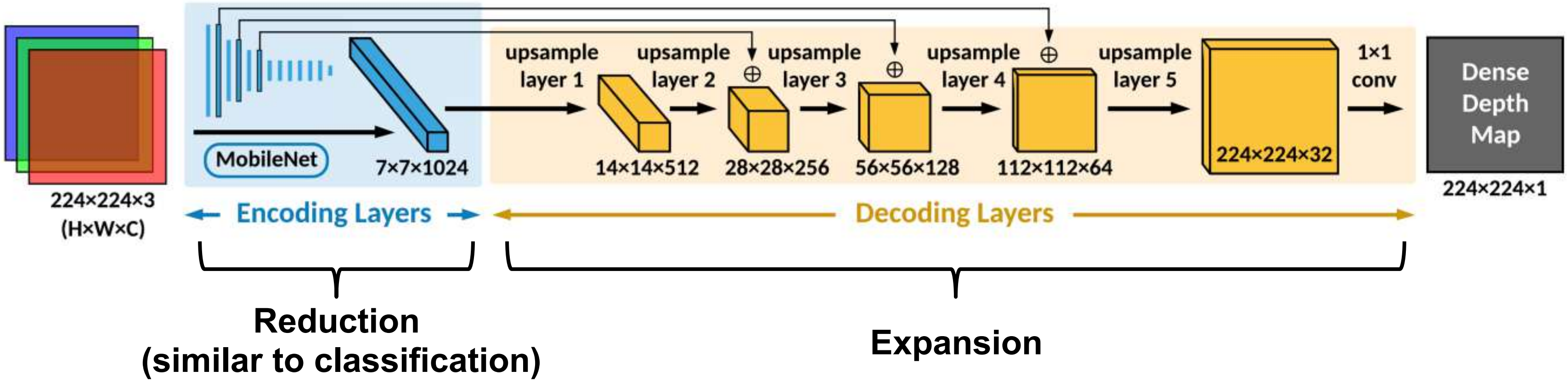
Depth estimation from a single RGB image desirable, due to the relatively low cost and size of monocular cameras.

RGB

Prediction



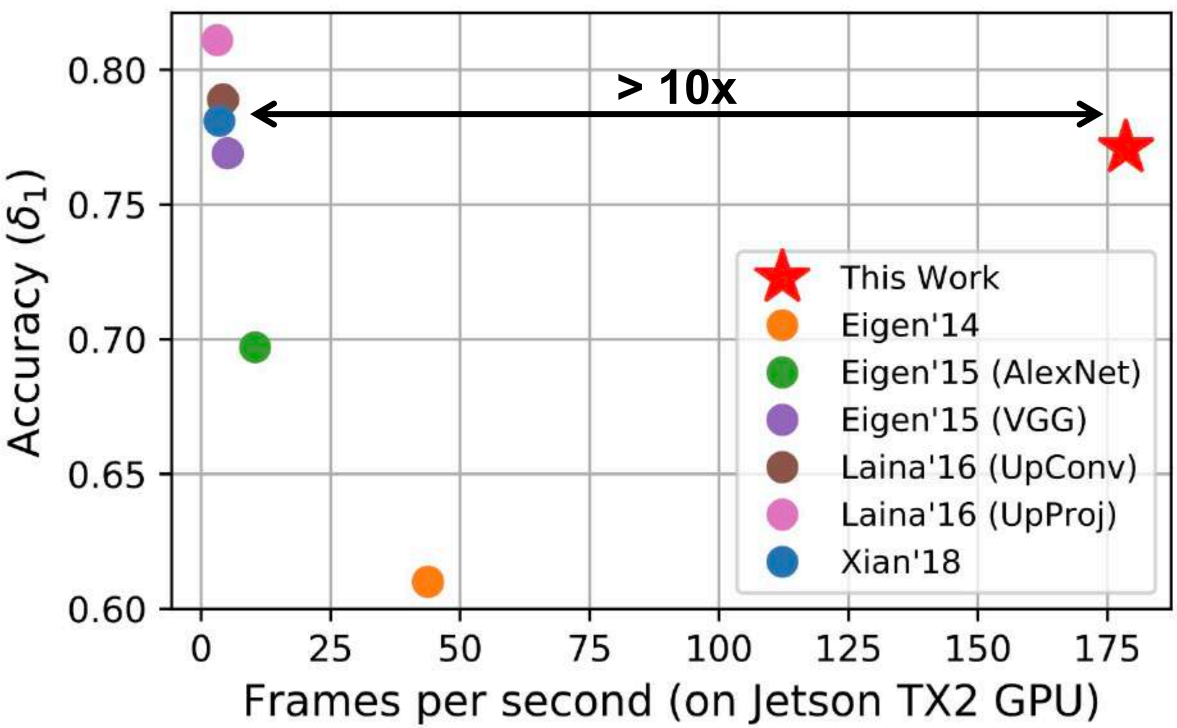
Auto Encoder DNN Architecture (Dense Output)



[Joint work with Sertac Karaman]

FastDepth: Fast Monocular Depth Estimation

Apply *NetAdapt*, *compact network design*, and *depth wise decomposition* to decoder layer to enable depth estimation at **high frame rates on an embedded platform** while still maintaining accuracy

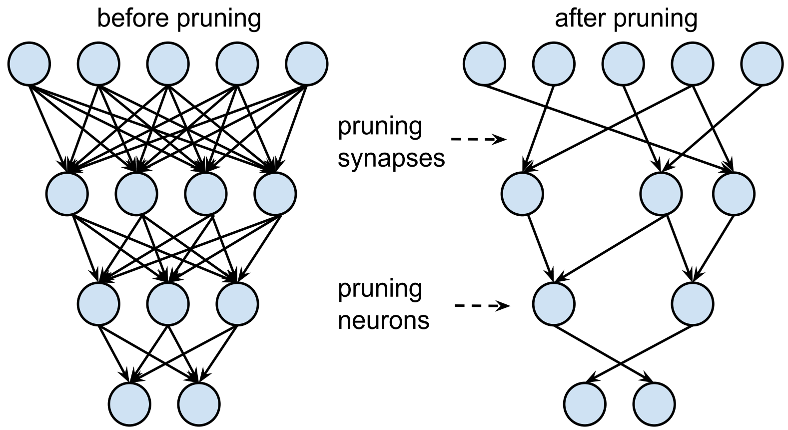


Configuration: Batch size of one (32-bit float)

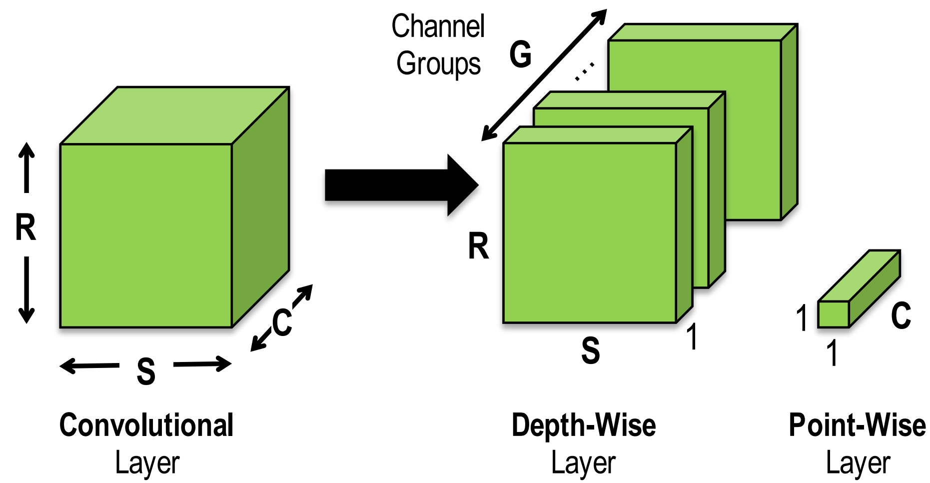
Models available at
http://fastdepth.mit.edu

Many Efficient DNN Design Approaches

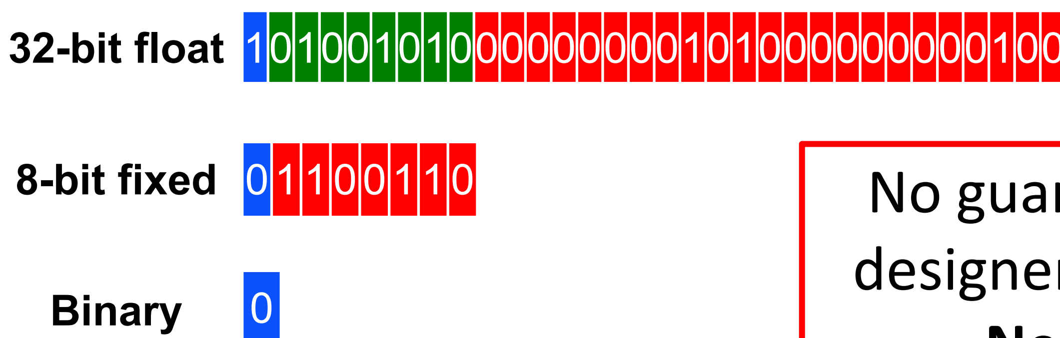
Network Pruning



Compact Network Architectures



Reduce Precision



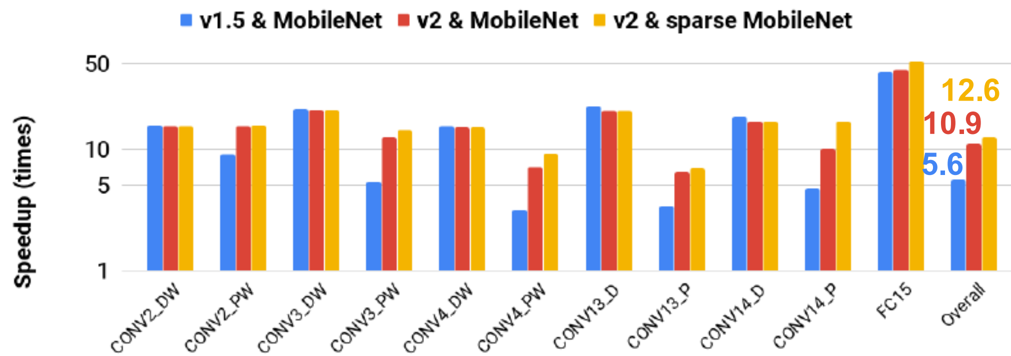
No guarantee that DNN algorithm designer will use a given approach.
Need flexible hardware!

[Chen et al., SysML 2018]

Eyeriss v2: Balancing Flexibility and Efficiency

Efficiently supports

- Wide range of filter shapes
  - Large **and** Compact
- Different Layers
  - **CONV, FC, depth wise, etc.**
- Wide range of sparsity
  - Dense **and** Sparse
- **Scalable architecture**



Speed up over Eyeriss v1 scales with number of PEs

# of PEs	256	1024	16384
AlexNet	17.9x	71.5x	1086.7x
GoogLeNet	10.4x	37.8x	448.8x
MobileNet	15.7x	57.9x	873.0x

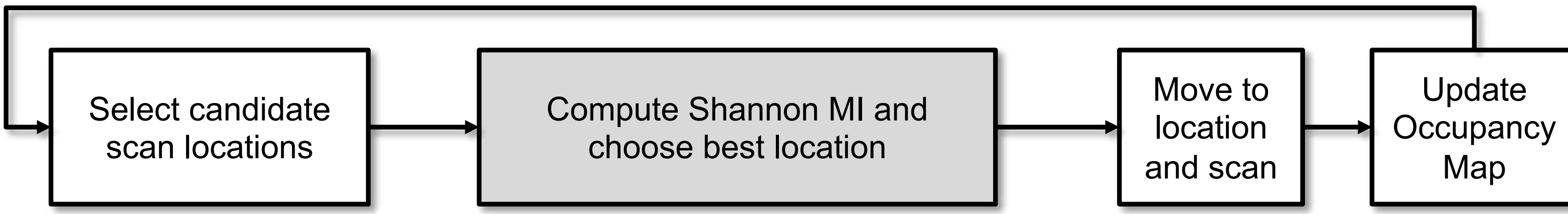
Over an order of magnitude faster and more energy efficient than Eyeriss v1

[Chen et al., JETCAS 2019]

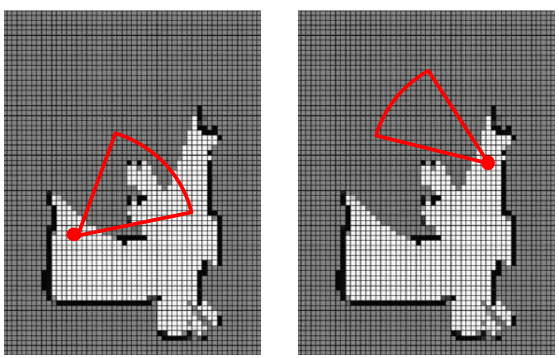
[Joint work with Joel Emer]

Where to Go Next: Planning and Mapping

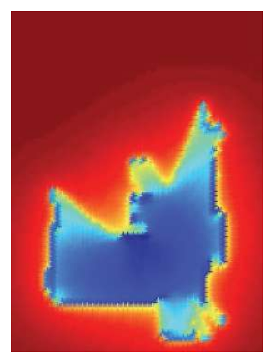
Robot Exploration: *Decide where to go by computing Shannon Mutual Information*



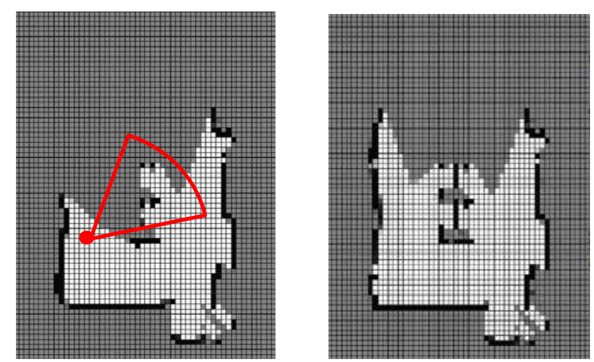
Where to scan?



Mutual Information

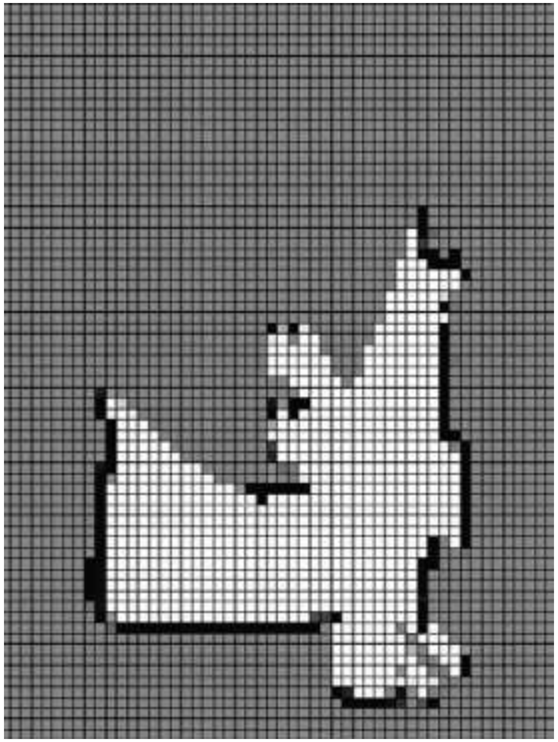


Updated Map

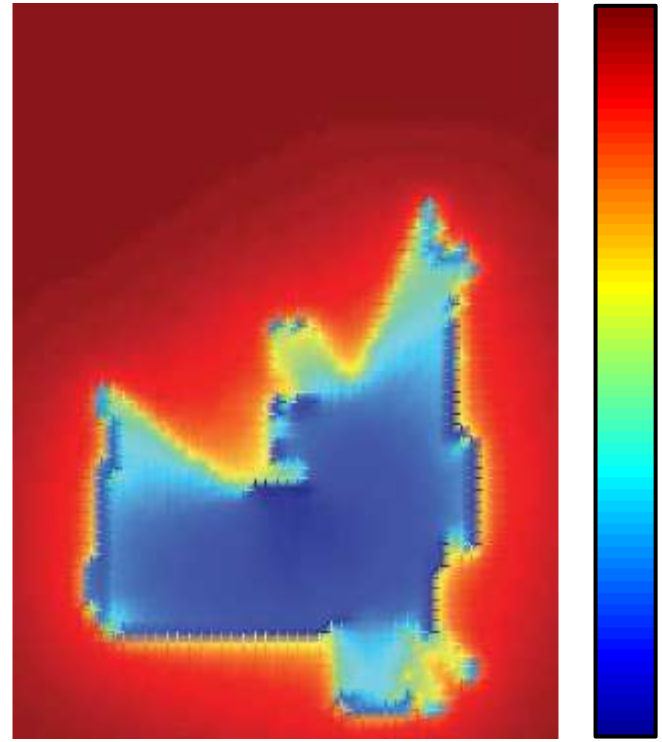


[Joint work with Sertac Karaman]

Information Theoretic Mapping



Occupancy grid map, M



Mutual information map, $I(M;Z)$

$$H(M|Z) = H(M) - I(M;Z)$$

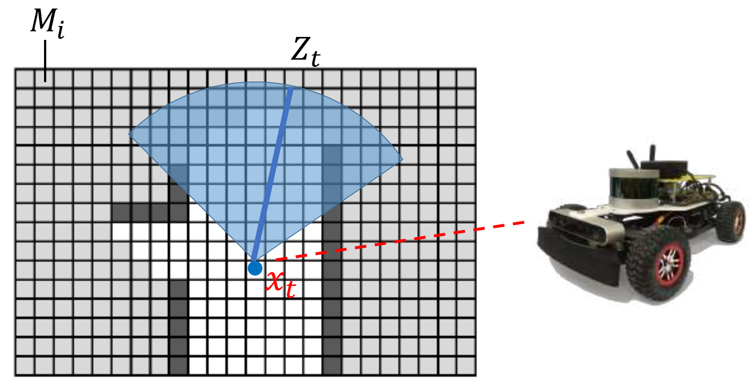
Perspective updated map entropy = Current map entropy − Mutual information

FSMI: Fast Shannon Mutual Information

Shannon Mutual Information
(between beam Z and map M)
[Julian et al., IJRR 2014]

$$I(M;Z)=\sum_{i=1}^{n}\boxed{\int_{z\geq 0}P(z)f(\delta_i(z),r_i)dz}$$

No closed form solution. Requires expensive **numerical integration at resolution** λ_z. $\boldsymbol{O(n^2\lambda_z)}$



FSMI: Fast Shannon Mutual Information

$$I(M;Z)=\sum_{j=1}^{n}\sum_{k=1}^{n}P(e_j)C_kG_{k,j}$$

Evaluate MI for all cells in entire beam altogether **removes numerical integration**. $\boldsymbol{O(n^2)}$

Approximate FSMI

$$I(M;Z)=\sum_{j=1}^{n}\sum_{k=j-\Delta}^{j+\Delta}P(e_j)C_kG_{k,j}$$

Approximate noise model of depth sensor with **truncated Gaussian***. $\boldsymbol{O(n)}$

*Charrow et al., ICRA 2015

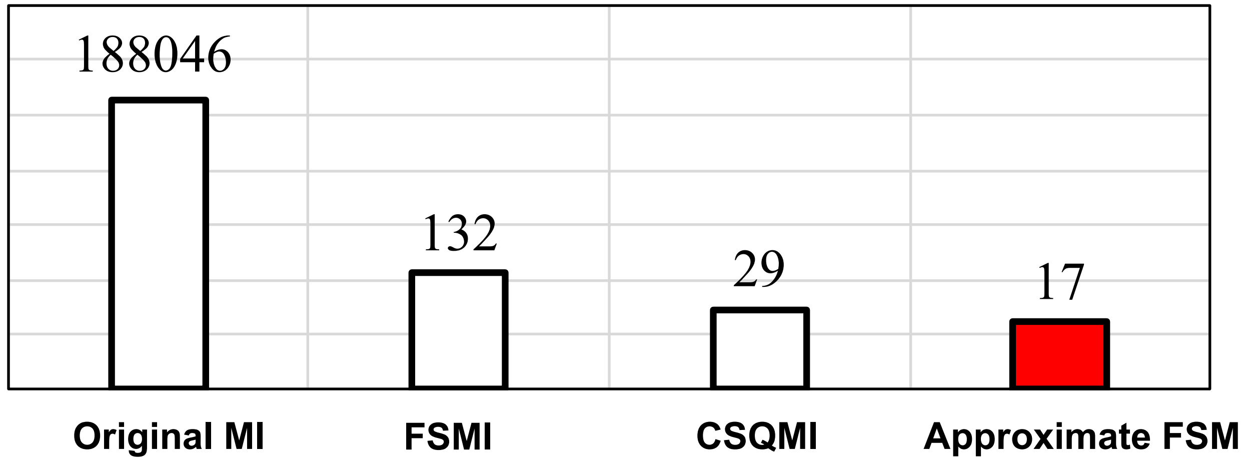
[Z. Zhang et al., ICRA 2019]

FSMI: Fast Shannon Mutual Information

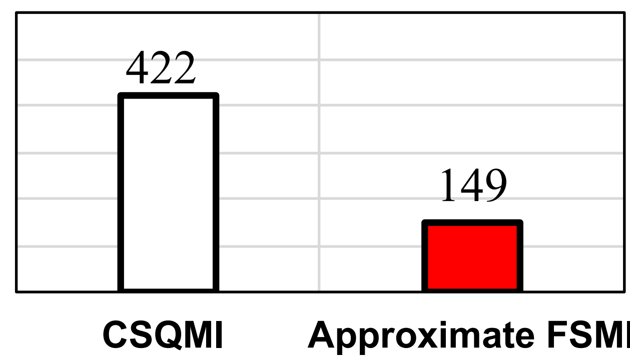
Original MI[1]	FSMI	CSQMI[2]	Approximate FSMI
$O(n^2\lambda_z)$	$O(n^2)$	$O(n)$	$O(n)$

[1] Julian et al., IJRR 2014; [2] Charrow et al., ICRA 2015

Measured run time per beam (μsec) on an **Intel Xeon core (desktop)**

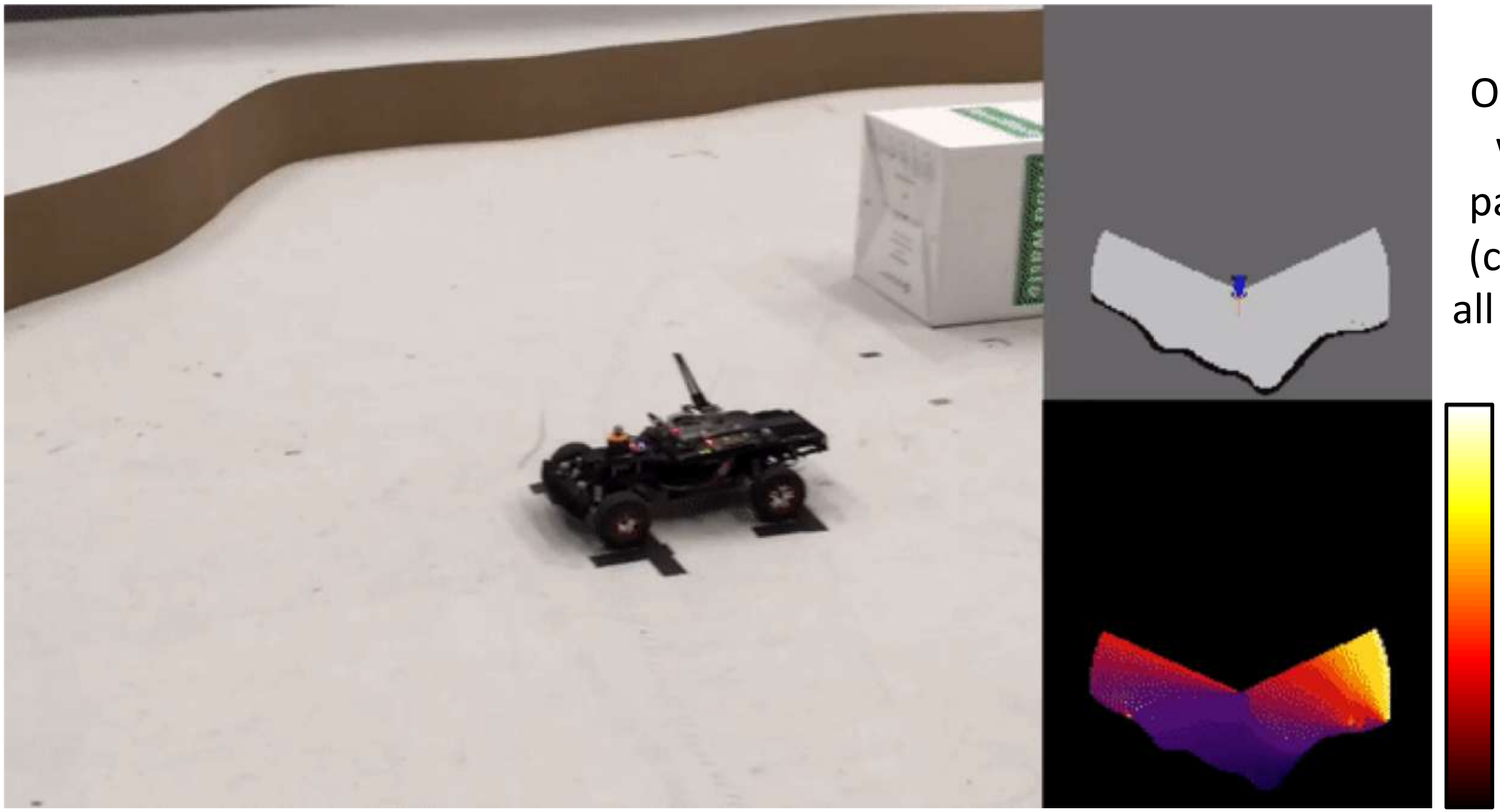


Measured run time per beam (μsec) on an **ARM Cortex-A57 core (embedded)**



Approximate FSMI is **over 1000x faster** than original MI and **1.7 – 2.8x faster** than CSQMI

Experimental Results (4x Real Time)

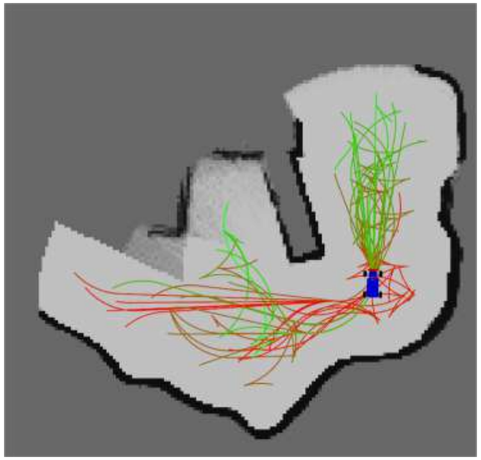


Exploration with a mini race car using motion capture for localization

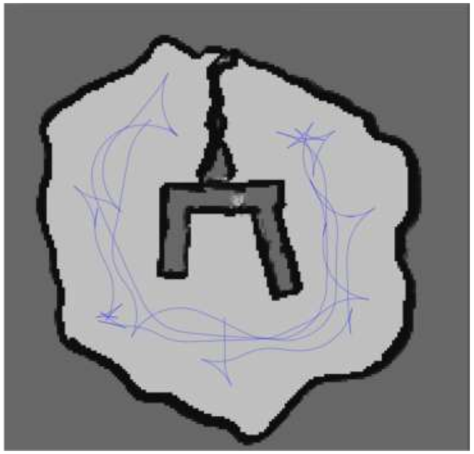
Quality of Result

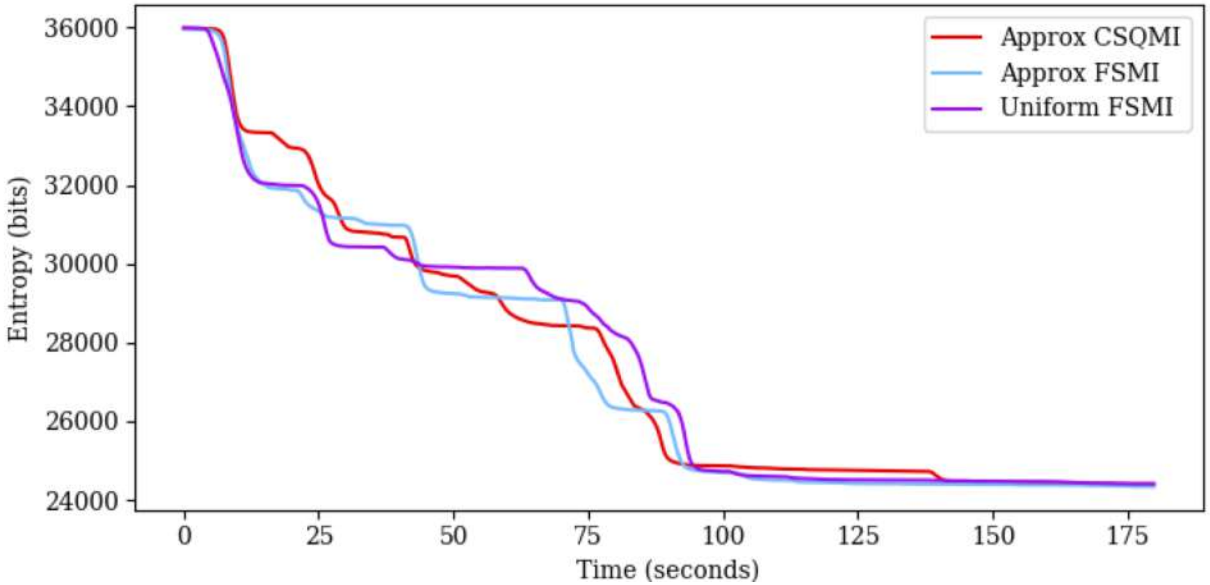
Experiment Environment

Paths with high MI per meter in green



Complete map and trajectory





Compute time per beam

CSQMI = 422.7 µsec

Approximate FSMI = 111.4 µsec

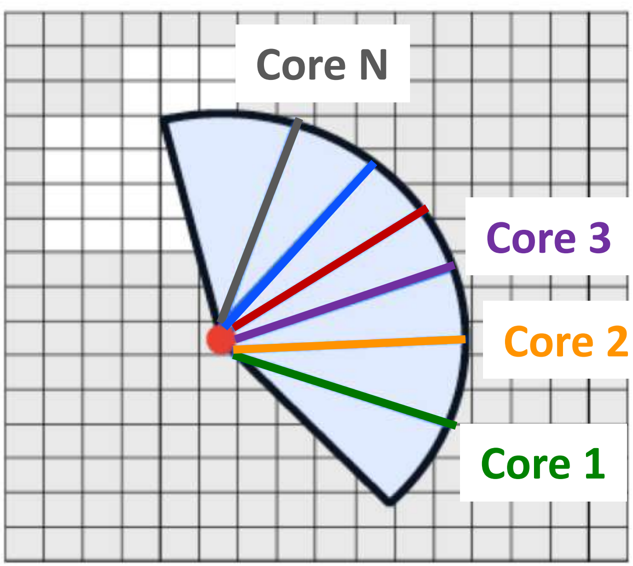
Approximate FSMI reduces entropy of map at same rate as CSQMI while computing Shannon Mutual Information

Building Hardware to Compute MI

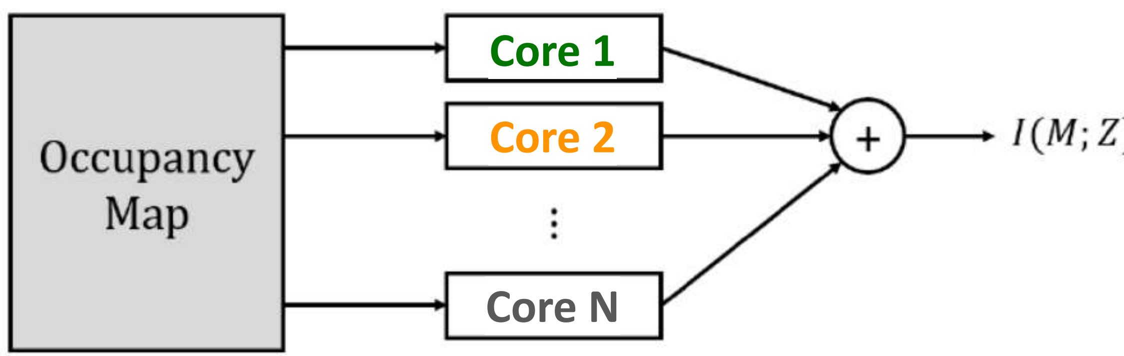
Motivation: Compute MI faster for faster exploration!

Approximate FSMI $$I(M;Z) = \sum_{j=1}^{n} \sum_{k=j-\Delta}^{j+\Delta} P(e_j) C_k G_{k,j}$$

Algorithm is ***embarrassingly*** parallel!
High throughput ***should*** be possible with multiple cores.

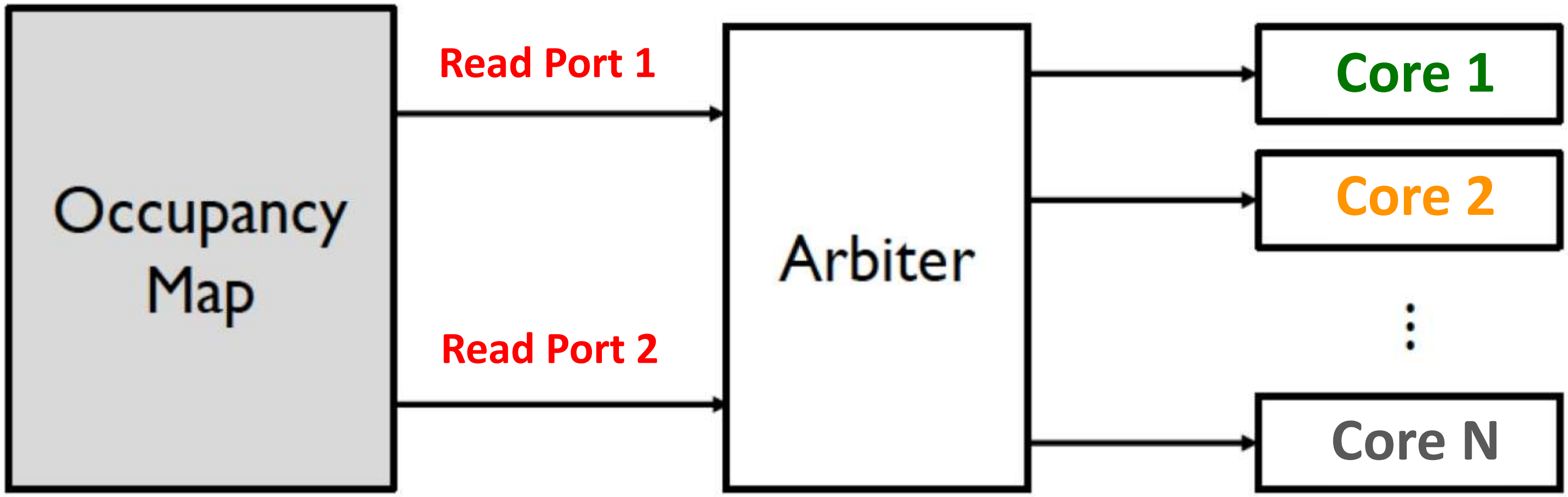


Process beams in parallel with multiple cores



Challenge is Data Delivery to All Cores

Power consumption of memory scales with number of ports.
Low power SRAM limited to two-ports!

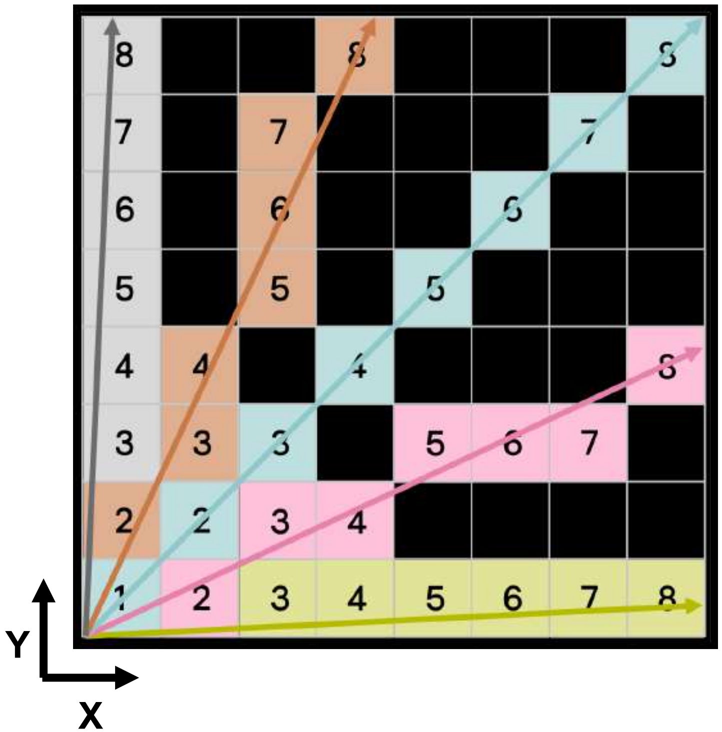


Data delivery, specifically memory bandwidth, limits the throughput (not compute)

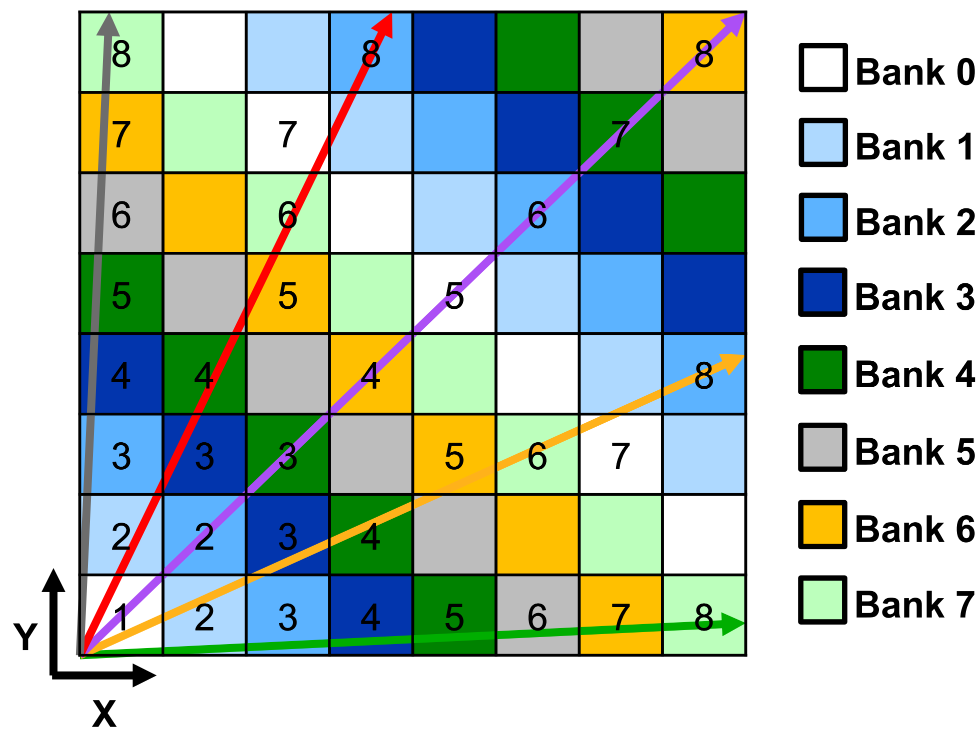
Specialized Memory Architecture

Break up map into **separate memory banks** and novel storage pattern to minimize read conflicts when processing different beams in parallel.

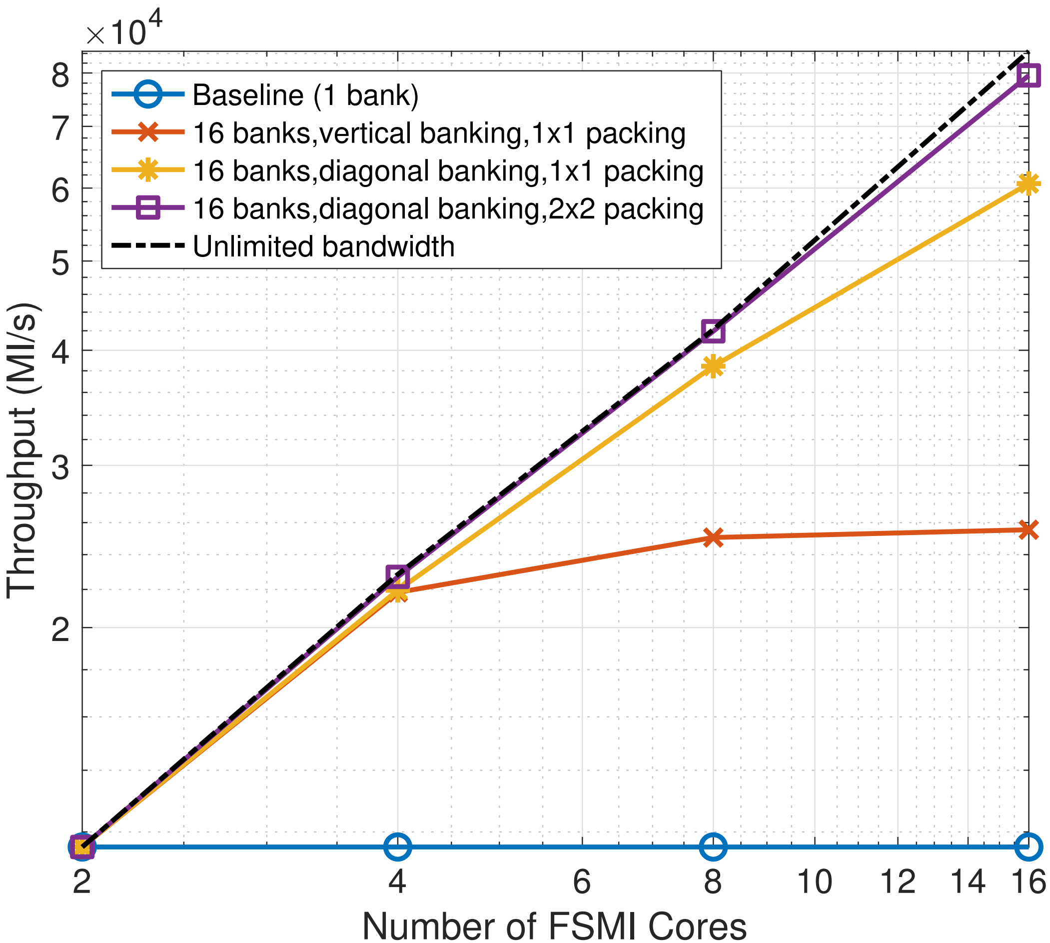
Memory Access Pattern



Diagonal Banking Pattern



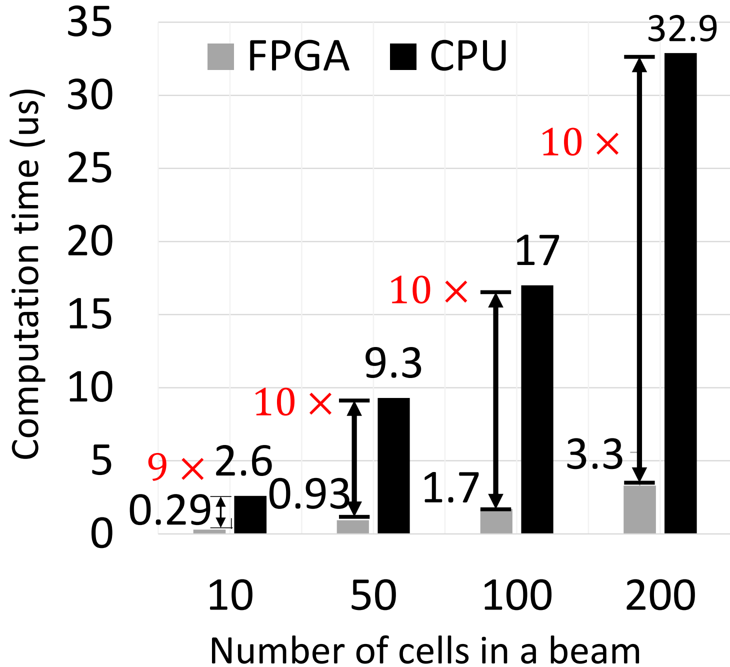
Experimental Results



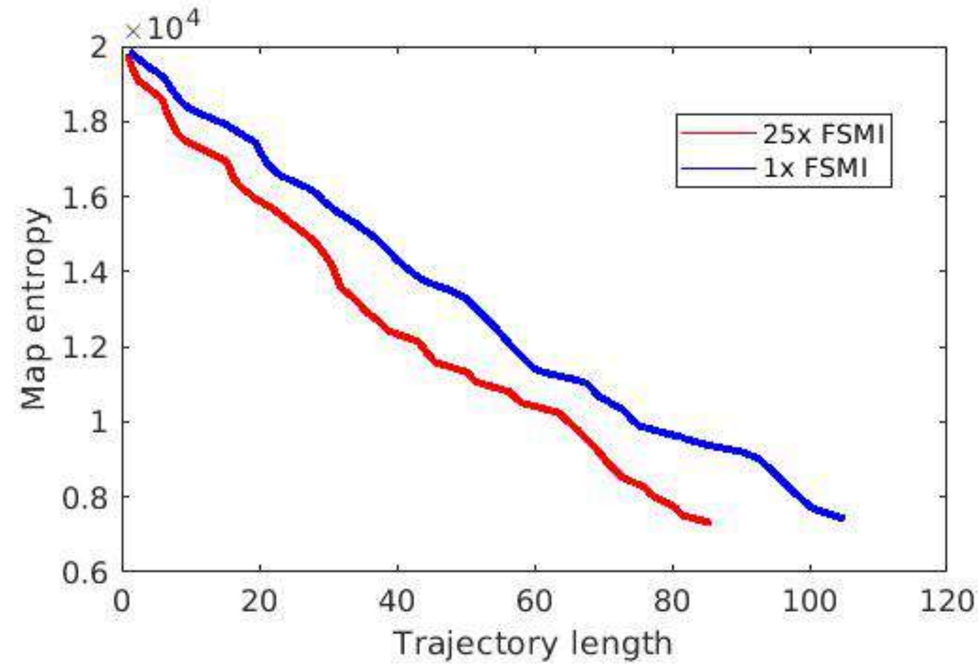
Specialized banking, efficient memory arbiter and packing multiple values at each address results in throughput **within 94% of theoretical limit** (unlimited bandwidth)

Experimental Results

Each FSMI core on FPGA faster than CPU core by 10x



Compute FSMI on more locations reduce trajectory length

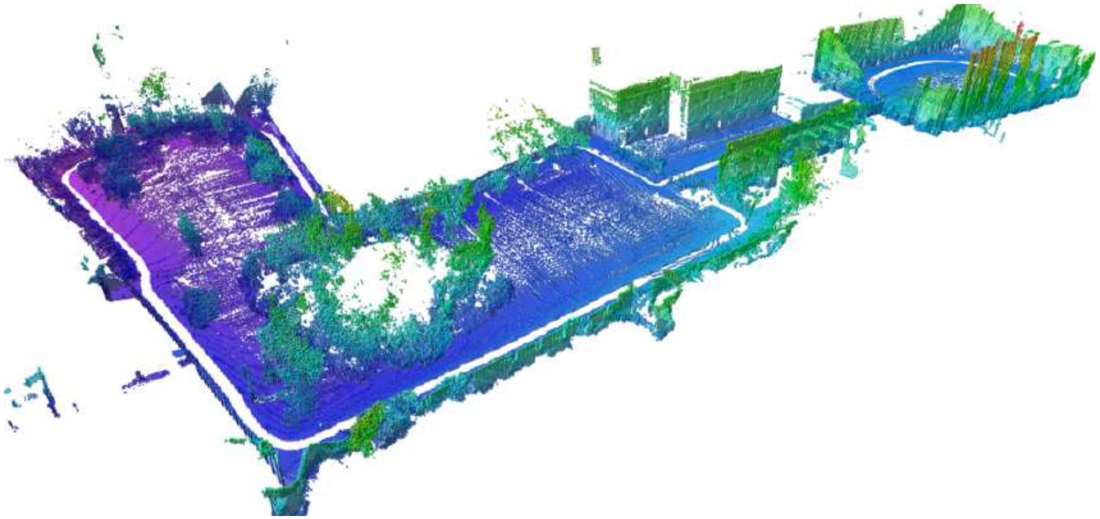


Compute the mutual information for an **entire map** of 20m x 20m at 0.1m resolution **in under a second while consuming under 2W on an FPGA***

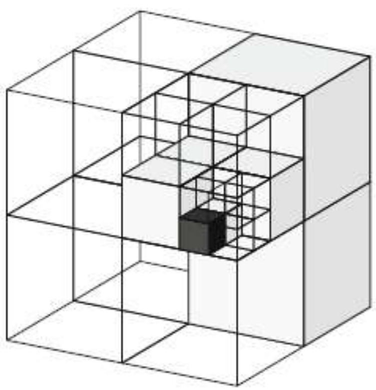
**estimate another order of magnitude reduction with ASIC*

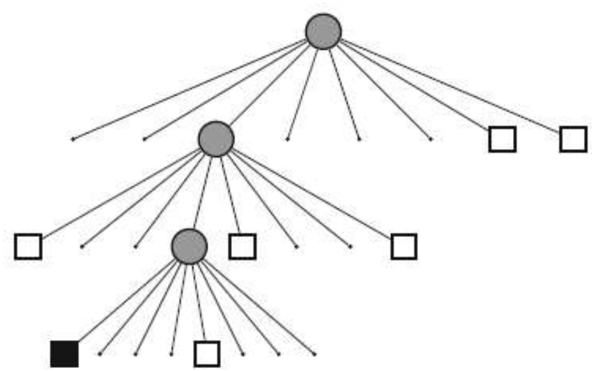
Extend FSMI to 3D Environments

Computing MI on a **3D map** requires significant amounts of storage and compute



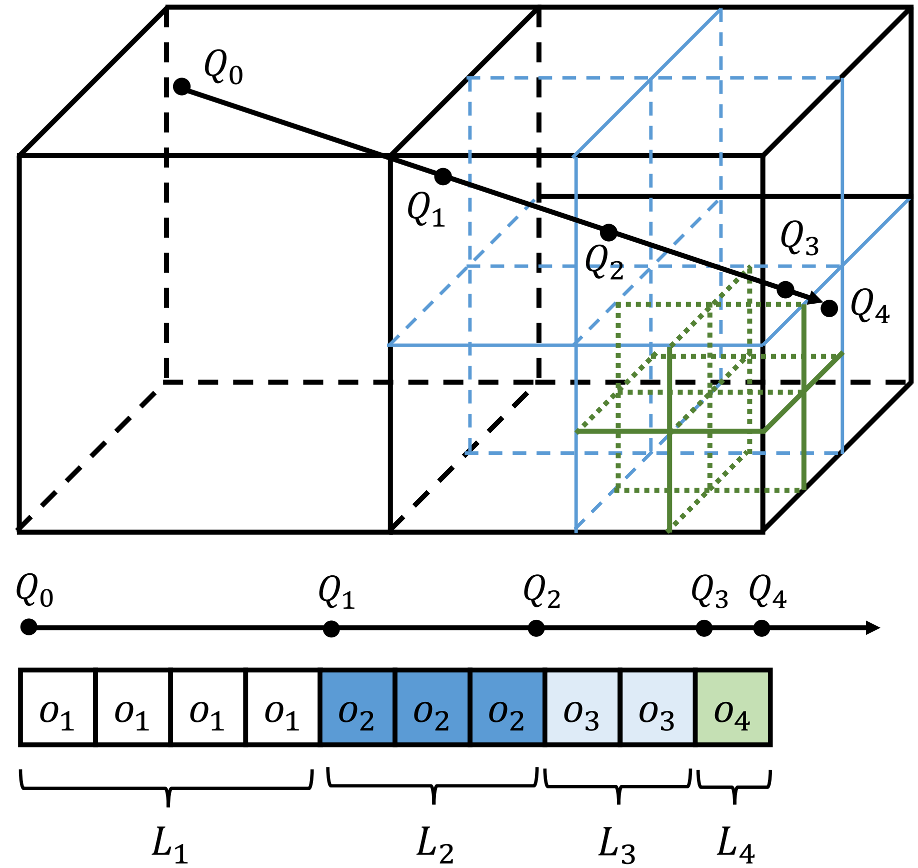
Compress map with OctoMap
[Hornung, et al., Autonomous Robots, 2013]





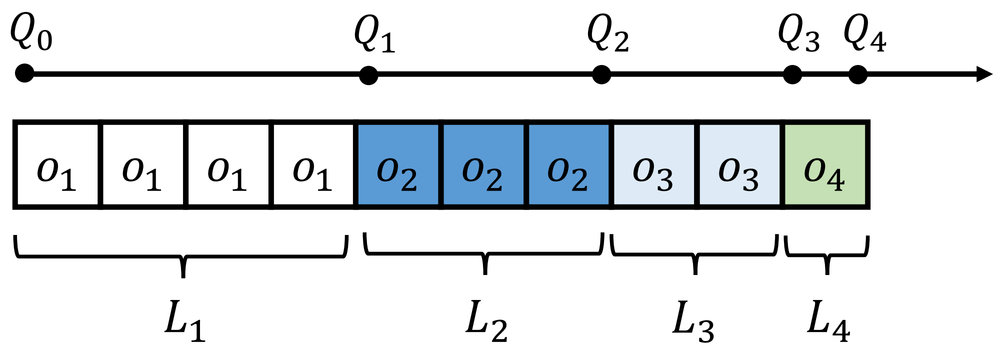
Compute FSMI on Compressed 3D

Ray-tracing in OctoMap



The 1D occupancy vector consists of multiple segments of **repeated occupancy values**

FSMI on Compressed Input



Uncompressed input format

$$\underbrace{o_1, o_1, \ldots, o_1}_{L_1}, \underbrace{o_2, \ldots, o_2}_{+\ L_2}, \underbrace{o_3, \ldots, o_3}_{+\ L_3}, \ldots, \underbrace{o_{n_r}, \ldots, o_{n_r}}_{+\ L_{n_r}}$$

$$L_1 + L_2 + L_3 + L_{n_r} = n$$

Compressed format (Run Length Encoding)

$$\underbrace{(o_1, L_1), (o_2, L_2), (o_3, L_3), \ldots, (o_{n_r}, L_{n_r})}_{n_r}$$

Time complexity of Approx FSMI

$$\boldsymbol{O(n)}$$

Goal: achieve the complexity of

$$\boldsymbol{O(n_r)}$$

$\boldsymbol{n_r \ll n}$, significant reduction if the constants are comparable

Complexity of 2D and 3D FSMI

	FSMI	Approximate FSMI
2D	$O(n^2)$	$O(n\Delta)$
3D (compress with RLE)	$O({n_r}^2)$	$O(n_r\Delta)$

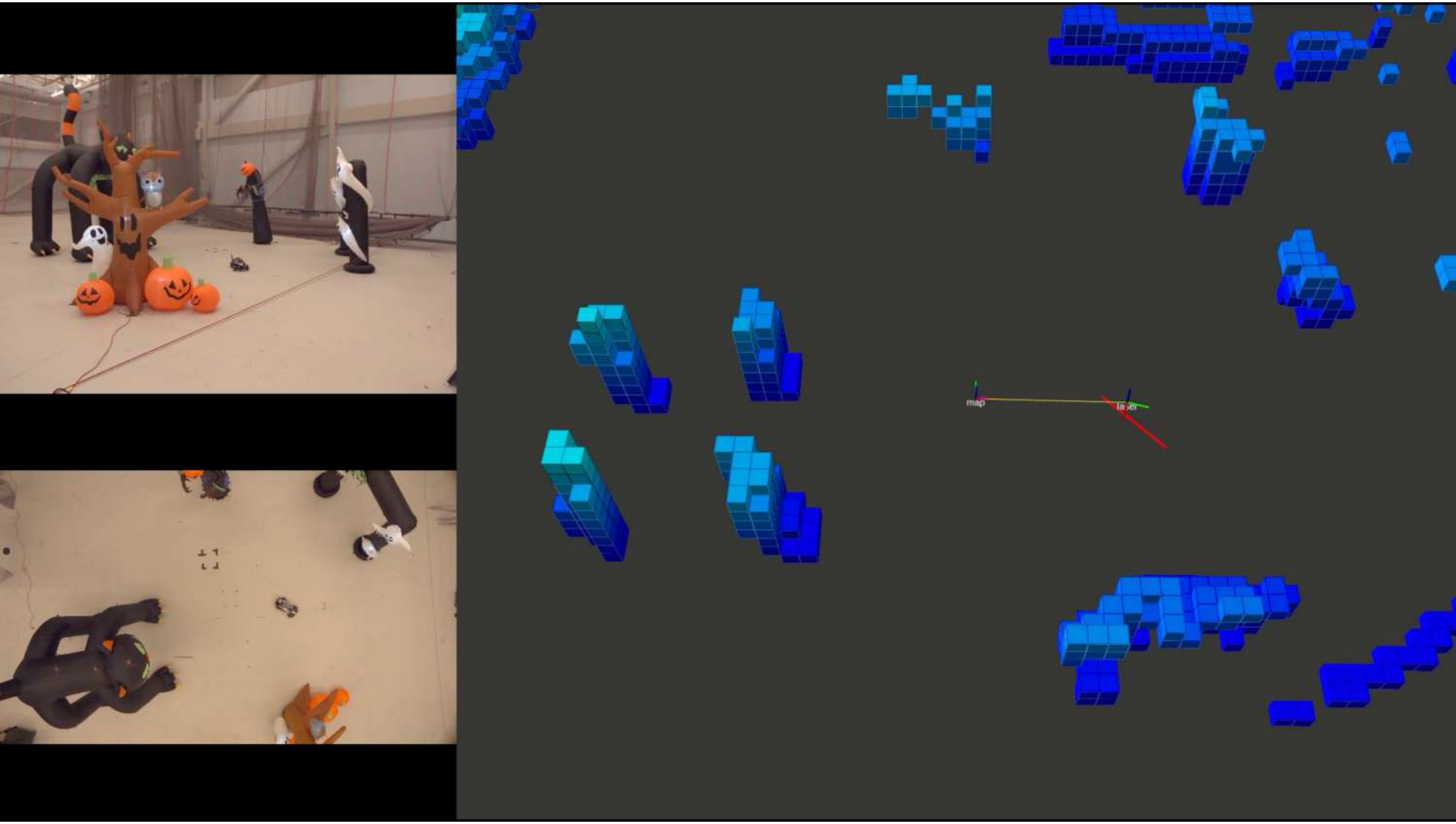
Measured speed up for a beam of 256 cells on an Intel Xeon CPU core for different degrees of compression (L)

	$L=1$	$L=2$	$L=4$	$L=8$	$L=16$	$L=32$	$L=64$	$L=128$
Approx FSMI-RLE	240.9	79.4	31.5	12.3	7.6	4.9	3.4	2.3
Acceleration	$0.2\times$	$0.7\times$	$1.8\times$	$4.6\times$	$7.4\times$	$11.2\times$	$16.5\times$	$24.4\times$

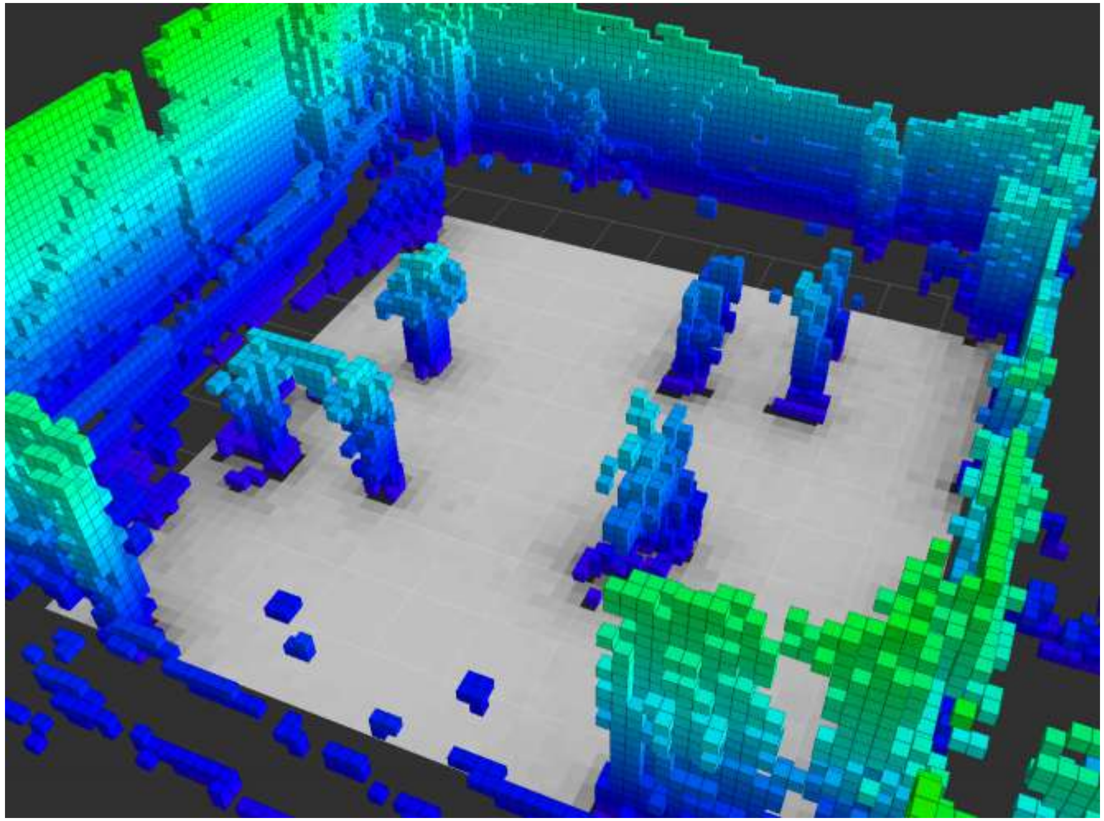
Baseline (Approx FSMI): 56µsec

Z. Zhang et al., FSMI: Fast computation of Shannon Mutual Information for information-theoretic mapping, *arXiv 2019*
http://arxiv.org/abs/1905.02238

Experiments of 3D FSMI (4x Real Time)



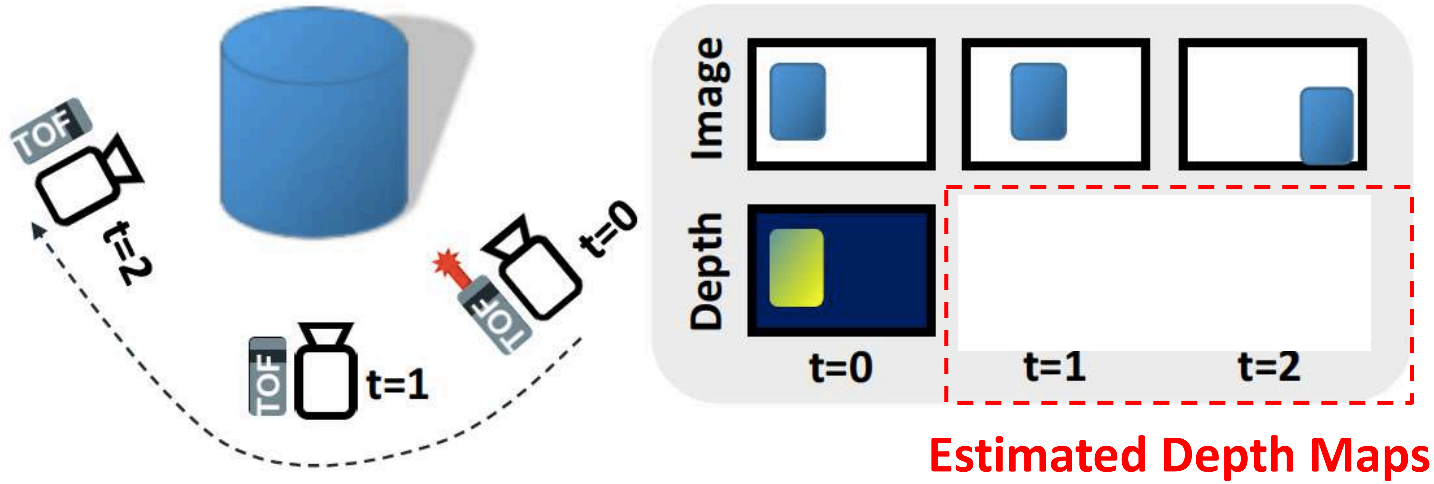
Experiments of 3D FSMI



We achieve an average compression ratio of around 18×, with an acceleration ratio of 8×

Low Power 3D Time of Flight Imaging

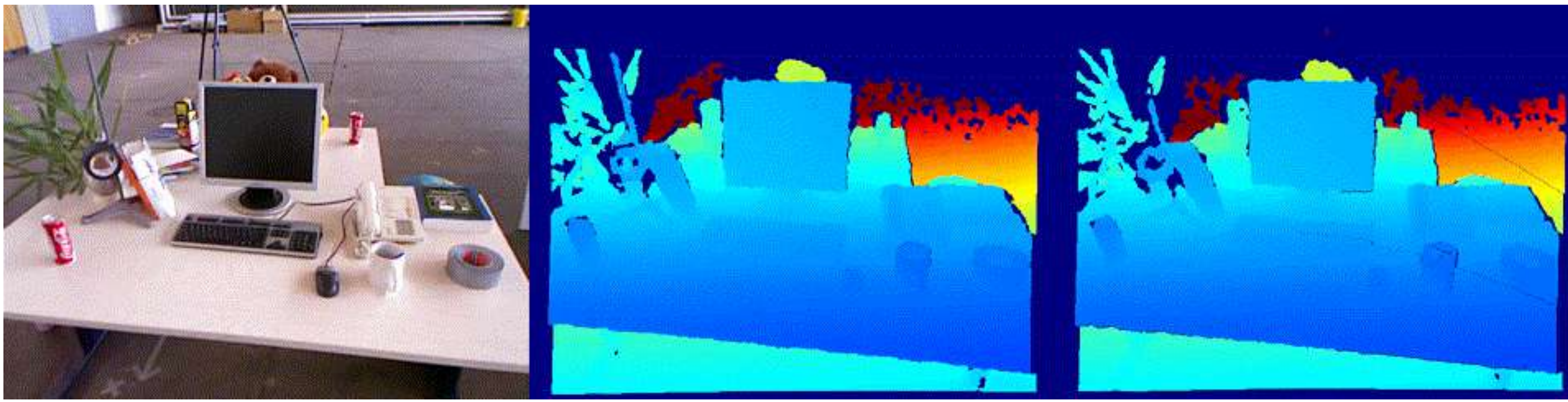
- Pulsed Time of Flight: Measure distance using round trip time of laser light for each image pixel
  - **Illumination + Imager Power: 2.5 – 20 W for range from 1 - 8 m**
- Use computer vision techniques and passive images to estimate changes in depth without turning on laser
  - **CMOS Imaging Sensor Power: < 350 mW**



Real-time Performance on Embedded Processor
VGA @ 30 fps on Cortex-A7 (< 0.5W active power)

[Noraky et al., ICIP 2017]

Results of Low Power Depth ToF Imaging

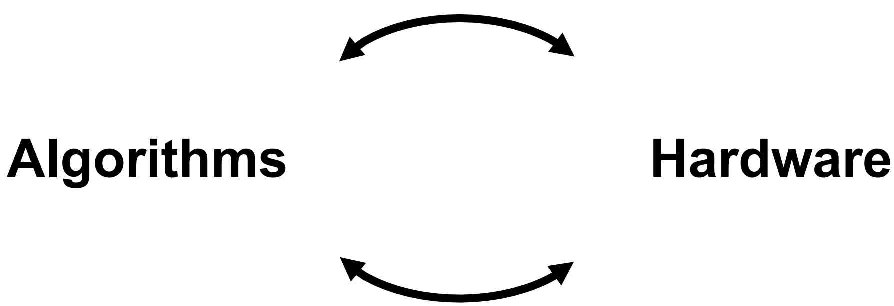


RGB Image	Depth Map **Ground Truth**	Depth Map **Estimated**

Mean Relative Error: 0.7%
Duty Cycle (on-time of laser): 11%

Summary

- Efficient computing is critical for advancing the progress of autonomous robots, particularly at the smaller scales. → **Critical step to making autonomy ubiquitous!**
- In order to meet computing demands in terms of power and speed, need to redesign computing hardware from the ground up → **Focus on data movement!**
- Specialized hardware opens up new opportunities for the co-design of algorithms and hardware → **Innovation opportunities for the future of robotics!**



Acknowledgements

Joel Emer

Sertac Karaman

Research conducted in the **MIT Energy-Efficient Multimedia Systems Group** would not be possible without the support of the following organizations:

For updates on our research Follow @eems_mit

References

- **Energy-Efficient Hardware for Deep Neural Networks**
  - **Project website:** *http://eyeriss.mit.edu*
  - *Y.-H. Chen, T. Krishna, J. Emer, V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE Journal of Solid State Circuits (JSSC), ISSCC Special Issue, Vol. 52, No. 1, pp. 127-138, January 2017.*
  - *Y.-H. Chen, J. Emer, V. Sze, "Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks," International Symposium on Computer Architecture (ISCA), pp. 367-379, June 2016.*
  - *Y.-H. Chen, T.-J. Yang, J. Emer, V. Sze, "Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices," IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), June 2019.*
  - *Eyexam: https://arxiv.org/abs/1807.07928*
- **Limitations of Existing Efficient DNN Approaches**
  - *Y.-H. Chen*, T.-J. Yang*, J. Emer, V. Sze, "Understanding the Limitations of Existing Energy-Efficient Design Approaches for Deep Neural Networks," SysML Conference, February 2018.*
  - *V. Sze, Y.-H. Chen, T.-J. Yang, J. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295-2329, December 2017.*
  - *Hardware Architecture for Deep Neural Networks: http://eyeriss.mit.edu/tutorial.html*

References

- **Co-Design of Algorithms and Hardware for Deep Neural Networks**
  - *T.-J. Yang, Y.-H. Chen, V. Sze, "Designing Energy-Efficient Convolutional Neural Networks using Energy-Aware Pruning," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.*
  - *Energy estimation tool:* http://eyeriss.mit.edu/energy.html
  - *T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, V. Sze, H. Adam, "NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications," European Conference on Computer Vision (ECCV), 2018.*
  - *D. Wofk*, F. Ma*, T.-J. Yang, S. Karaman, V. Sze, "FastDepth: Fast Monocular Depth Estimation on Embedded Systems," IEEE International Conference on Robotics and Automation (ICRA), May 2019. http://fastdepth.mit.edu/*

- **Energy-Efficient Visual Inertial Localization**
  - **Project website:** *http://navion.mit.edu*
  - *A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, V. Sze, "Navion: A Fully Integrated Energy-Efficient Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones," IEEE Symposium on VLSI Circuits (VLSI-Circuits), June 2018.*
  - *Z. Zhang*, A. Suleiman*, L. Carlone, V. Sze, S. Karaman, "Visual-Inertial Odometry on Chip: An Algorithm-and-Hardware Co-design Approach," Robotics: Science and Systems (RSS), July 2017.*
  - *A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, V. Sze, "Navion: A 2mW Fully Integrated Real-Time Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones," IEEE Journal of Solid State Circuits (JSSC), VLSI Symposia Special Issue, Vol. 54, No. 4, pp. 1106-1119, April 2019.*

References

- **Fast Shannon Mutual Information for Robot Exploration**
  - *Z. Zhang, T. Henderson, V. Sze, S. Karaman, "FSMI: Fast computation of Shannon Mutual Information for information-theoretic mapping," IEEE International Conference on Robotics and Automation (ICRA), May 2019.*
  - *P. Li*, Z. Zhang*, S. Karaman, V. Sze, "High-throughput Computation of Shannon Mutual Information on Chip," Robotics: Science and Systems (RSS), June 2019*
  - *Z. Zhang, T. Henderson, S. Karaman, V. Sze, "FSMI: Fast computation of Shannon Mutual Information for information-theoretic mapping," extended preprint on arXiv, May 2019* http://arxiv.org/abs/1905.02238
- **Low Power Time of Flight Imaging**
  - *J. Noraky, V. Sze, "Low Power Depth Estimation of Rigid Objects for Time-of-Flight Imaging," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 2019.*
  - *J. Noraky, C. Mathy, A. Cheng, V. Sze, "Low Power Adaptive Time-Of-Flight Imaging For Multiple Rigid Objects," IEEE International Conference on Image Processing (ICIP), September 2019.*
  - *J. Noraky, V. Sze, "Depth Estimation of Non-Rigid Objects For Time-Of-Flight Imaging," IEEE International Conference on Image Processing (ICIP), October 2018.*
  - *J. Noraky, V. Sze, "Low Power Depth Estimation for Time-of-Flight Imaging," IEEE International Conference on Image Processing (ICIP), September 2017.*