Efficient Computing for AI and Robotics

Vivienne Sze

Massachusetts Institute of Technology

In collaboration with Luca Carlone, Yu-Hsin Chen, Joel Emer, Sertac Karaman, Tushar Krishna, Thomas Heldt, Trevor Henderson, Hsin-Yu Lai, Peter Li, Fangchang Ma, James Noraky, Gladynel Saavedra Peña, Charlie Sodini, 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)

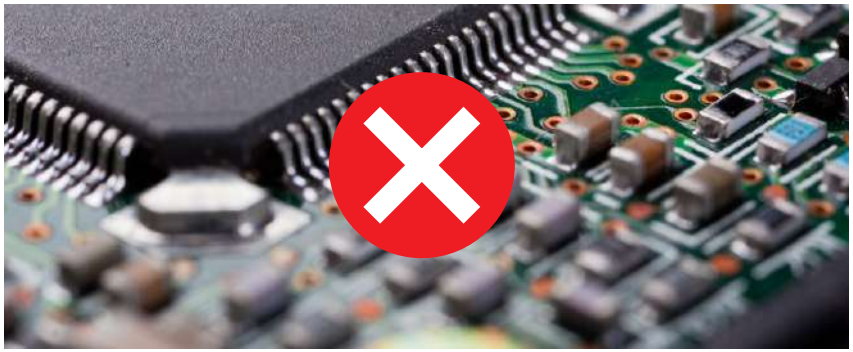
Cameras and radar generate ~6 gigabytes of data every 30 seconds.

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

Generates wasted heat and some prototypes need water-cooling!

Existing Processors Consume Too Much Power

< 1 Watt

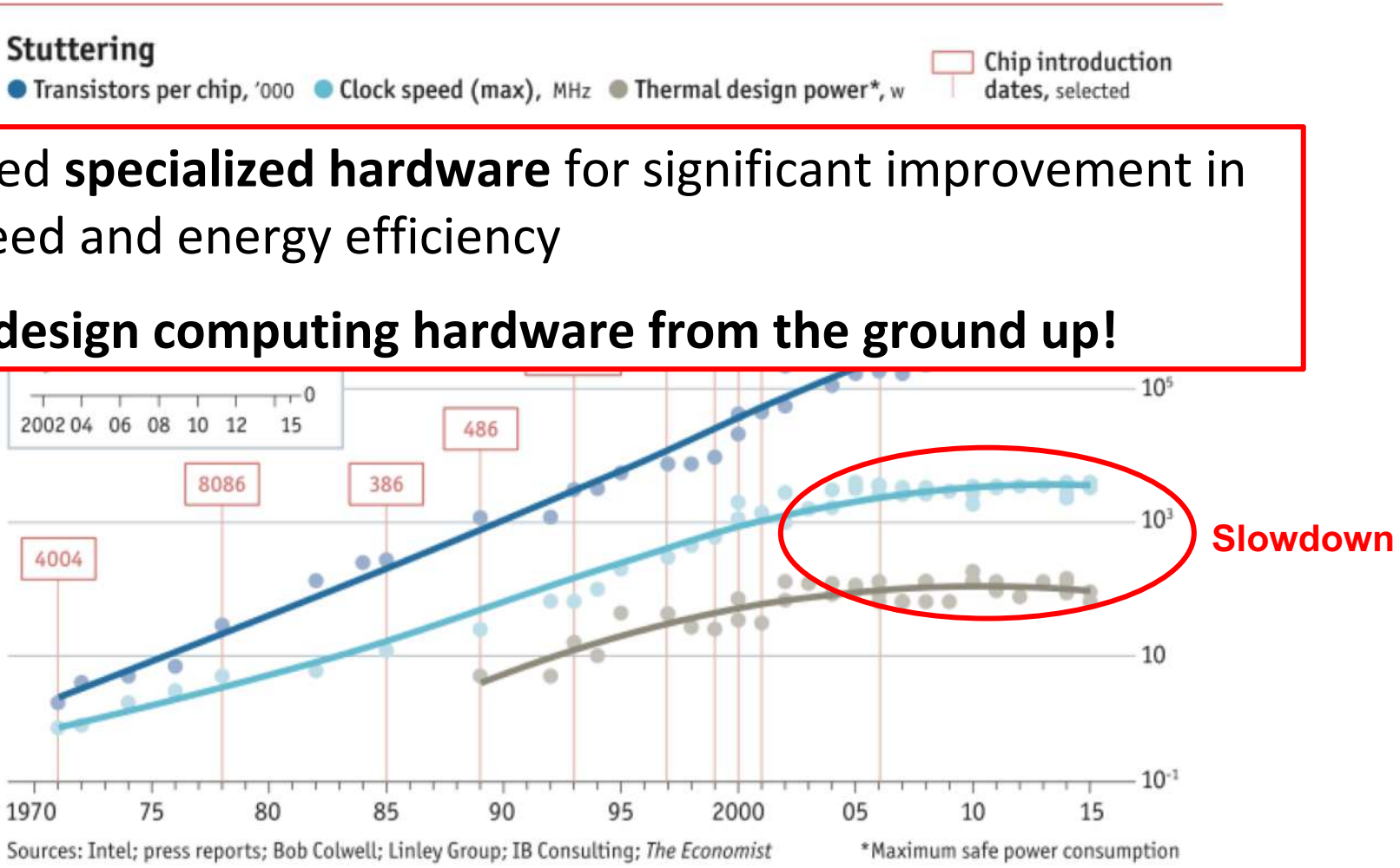


> 10 Watts

Transistors are NOT Getting More Efficient

Slow down of Moore's Law and Dennard Scaling

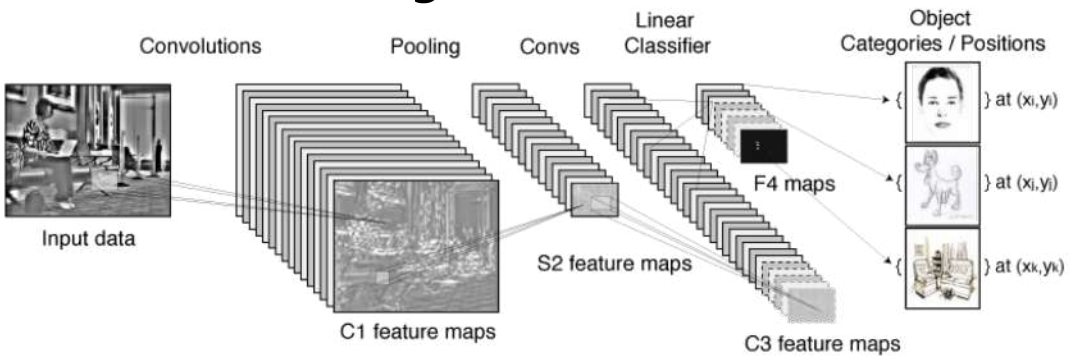
General purpose microprocessors not getting faster or more efficient



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

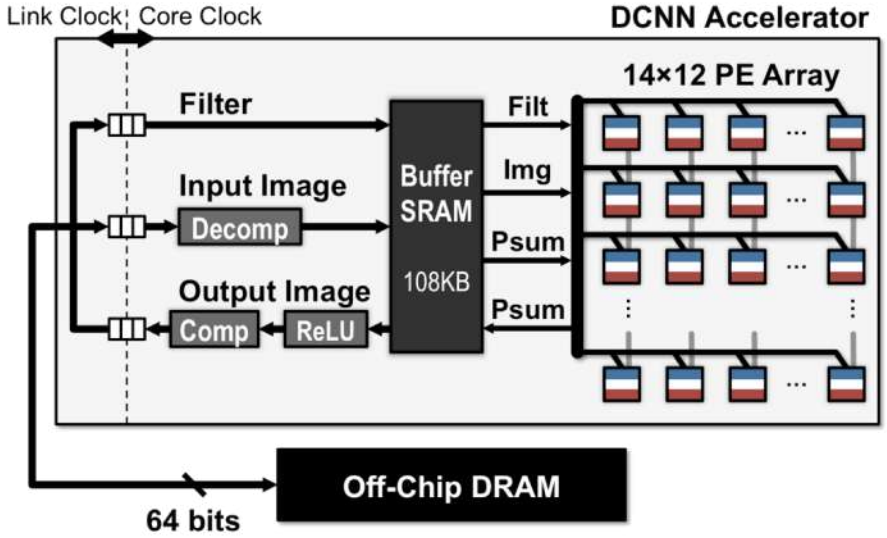
Energy-Efficient Computing with Cross-Layer Design

Algorithms

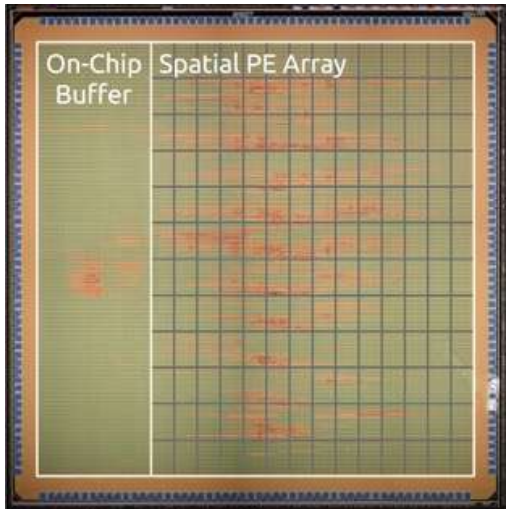


Systems

Architectures

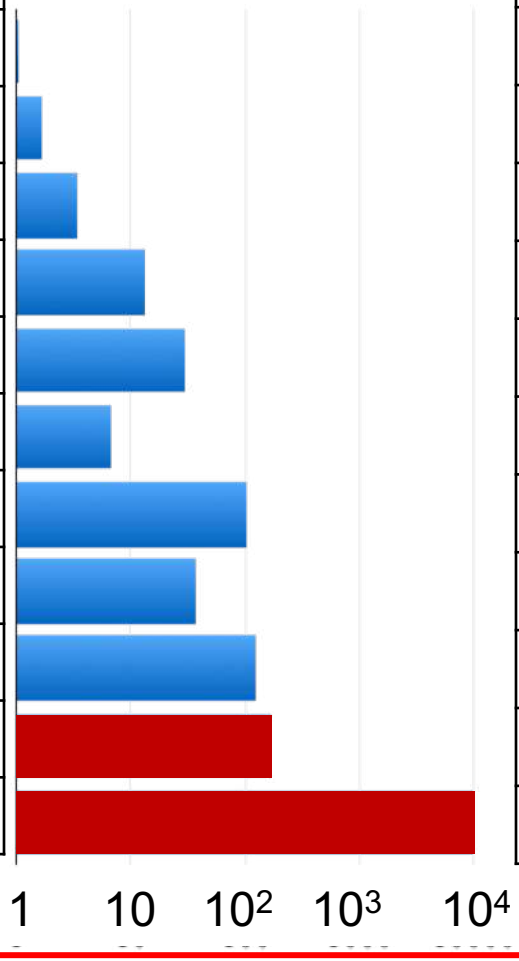


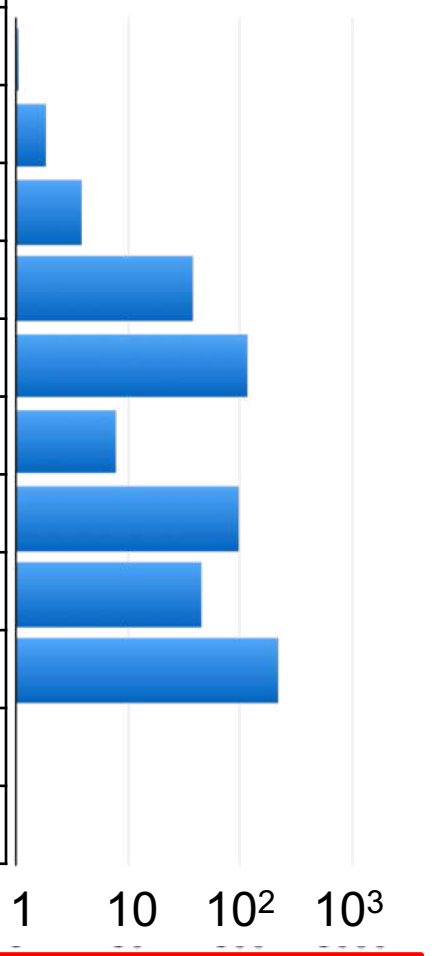
Circuits



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





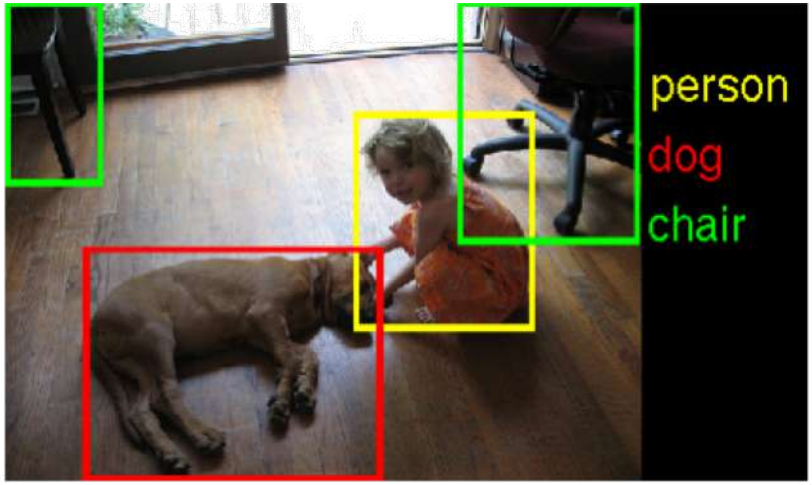
Memory access is **orders of magnitude** higher energy than compute

[Horowitz, ISSCC 2014]

Deep Neural Networks

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

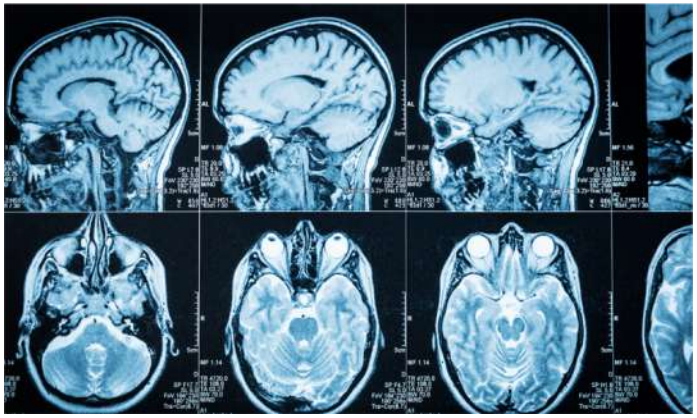
Computer Vision



Speech Recognition

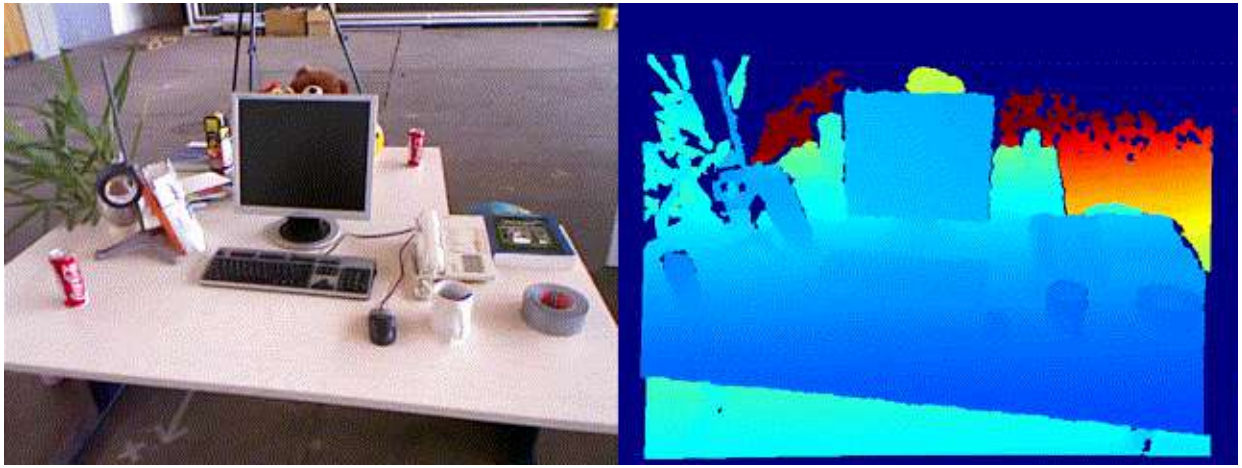
Game Play

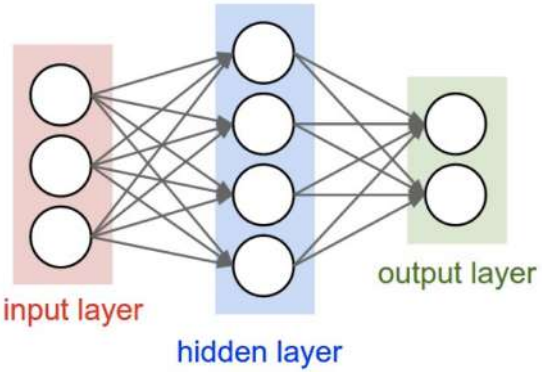
Medical



DNNs for Understanding the Environment

Depth Estimation





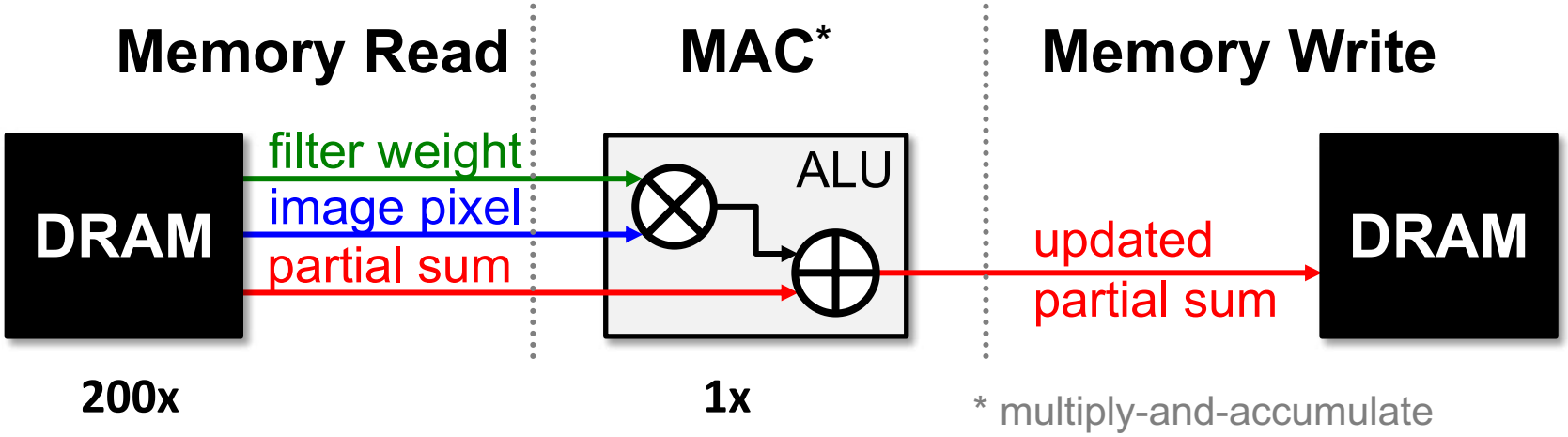
Semantic Segmentation

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

Properties We Can Leverage

- Operations exhibit **high parallelism**
  → **high throughput** possible
- Memory Access is the Bottleneck

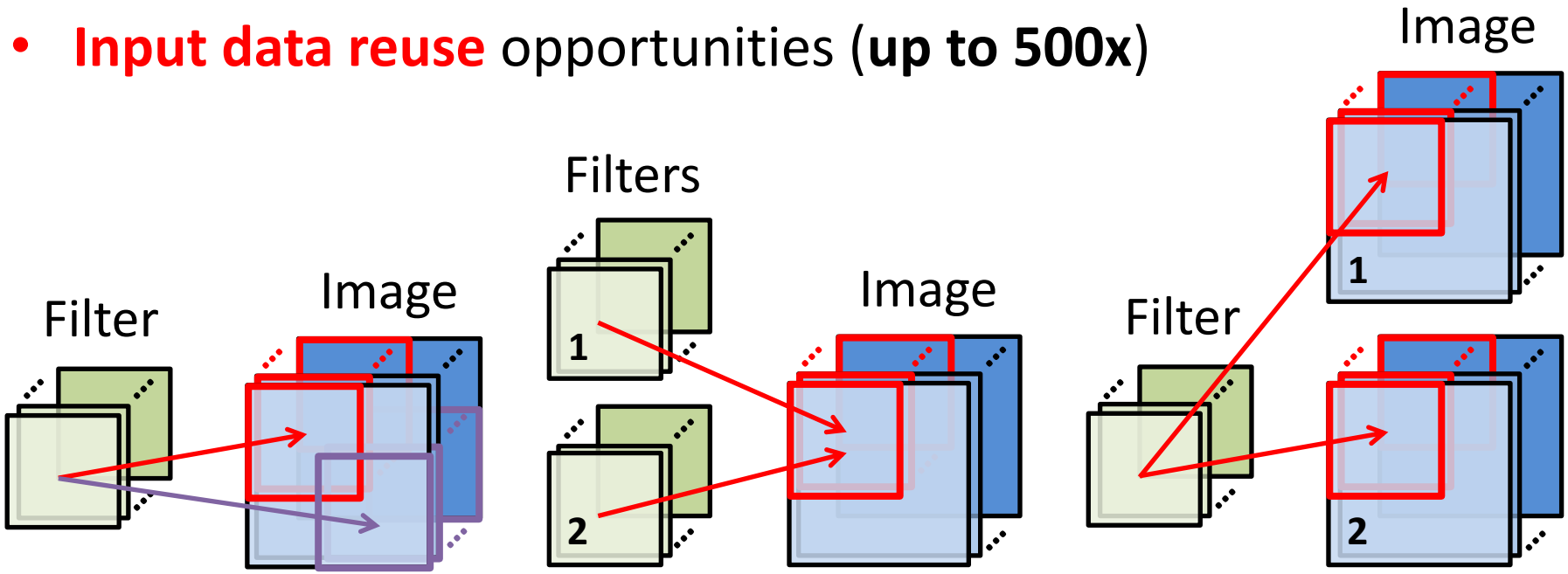


Worst Case: all memory R/W are **DRAM** accesses

- 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**)

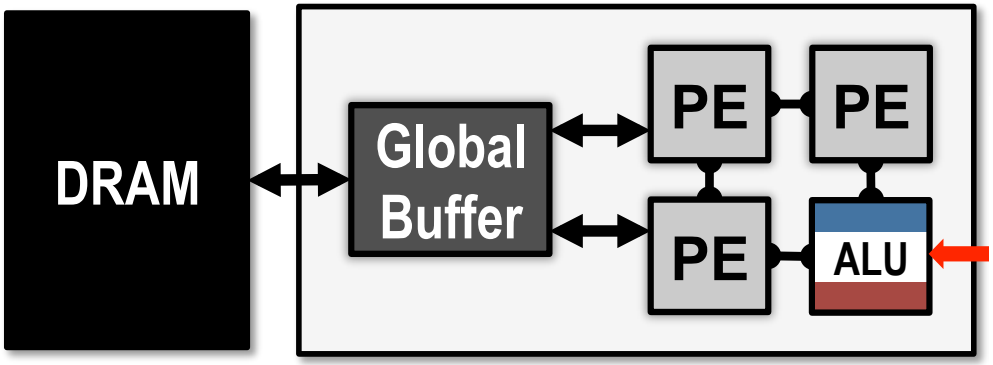


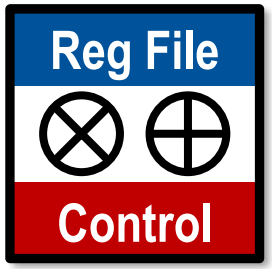
Convolutional Reuse (pixels, weights)

Image Reuse (pixels)

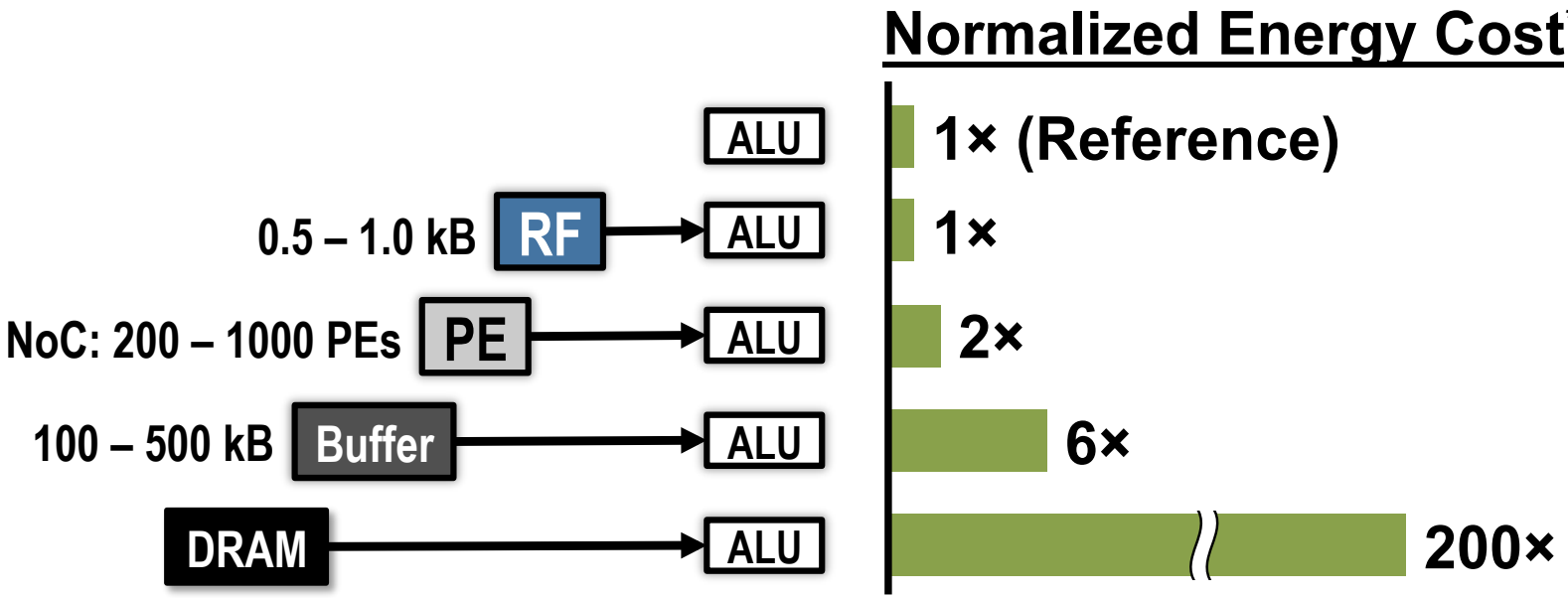
Filter Reuse (weights)

Exploit Data Reuse at Low-Cost Memories





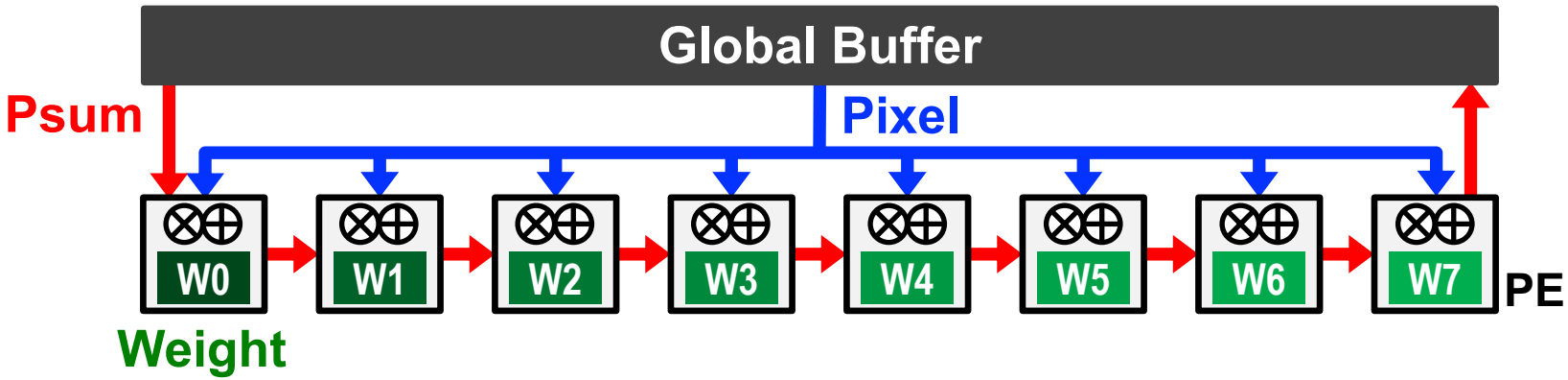
Specialized hardware with small (< 1kB) low cost memory near compute



* measured from a commercial 65nm process

Farther and **larger** memories consume more power

Weight Stationary (WS)

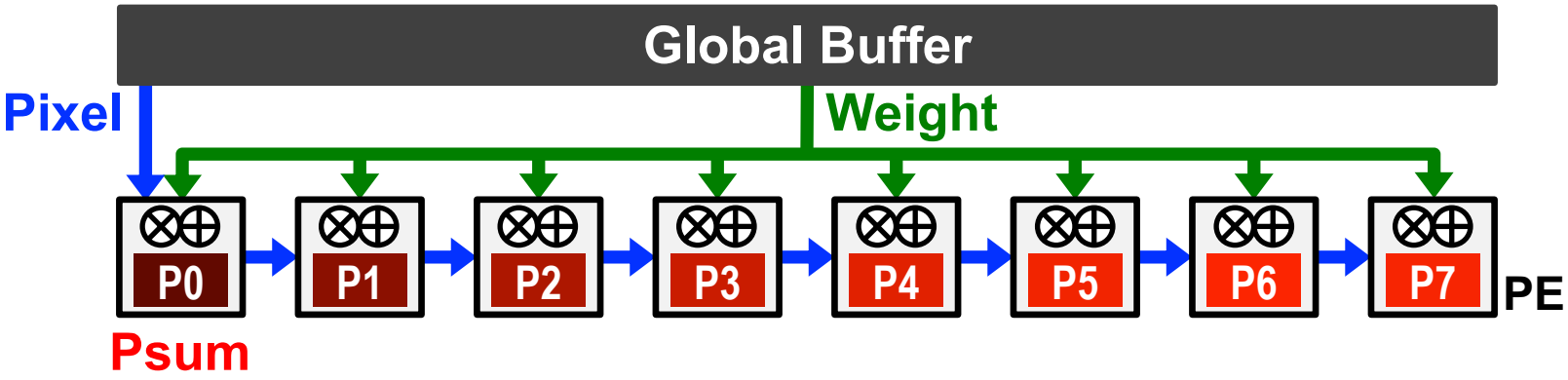


- **Minimize weight** read energy consumption
  - maximize convolutional and filter reuse of weights
- **Examples:**

[**Chakradhar**, *ISCA* 2010] [**nn-X (NeuFlow)**, *CVPRW* 2014]

[**Park**, *ISSCC* 2015] [**Origami**, *GLSVLSI* 2015]

Output Stationary (OS)



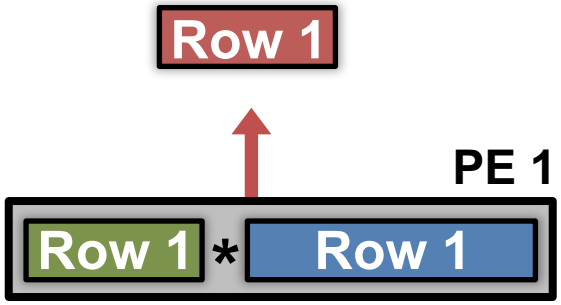
- **Minimize partial sum** R/W energy consumption
  - maximize local accumulation
- **Examples:**

  [**Gupta**, *ICML* 2015]

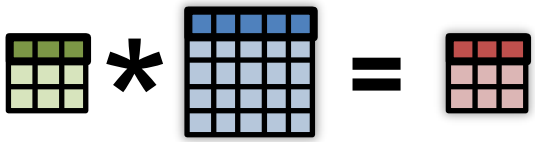
  [**ShiDianNao**, *ISCA* 2015]

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

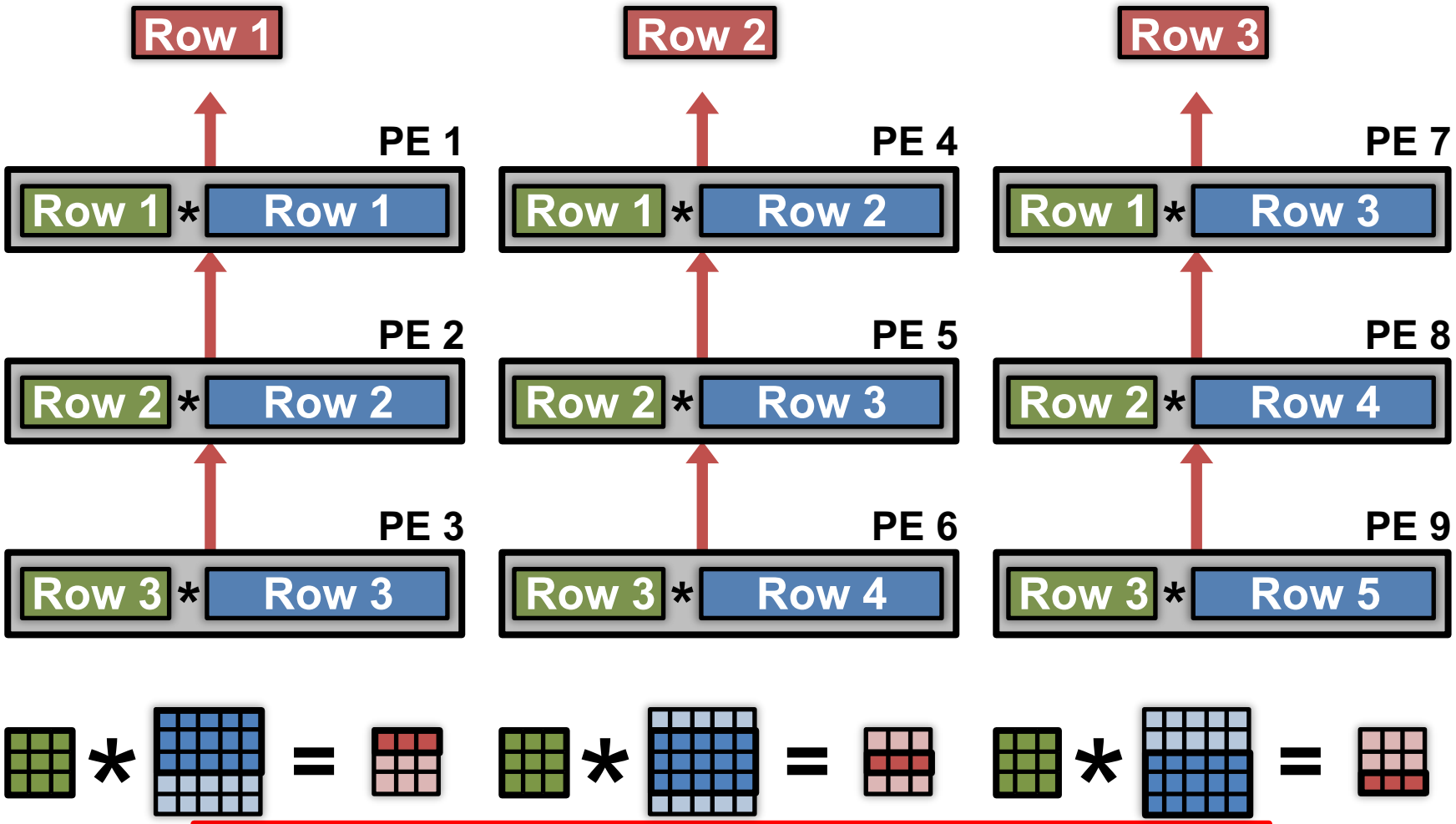
Row Stationary Dataflow



- 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



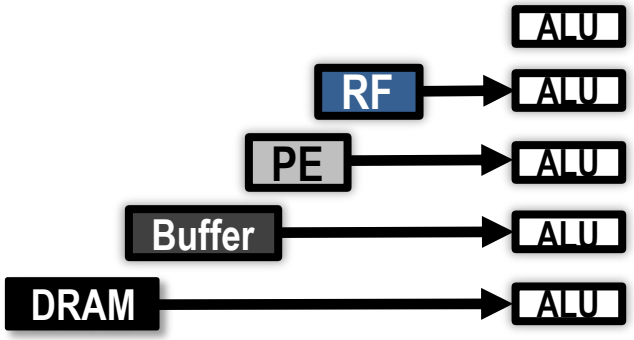
Optimize for **overall energy efficiency** instead for only a certain data type

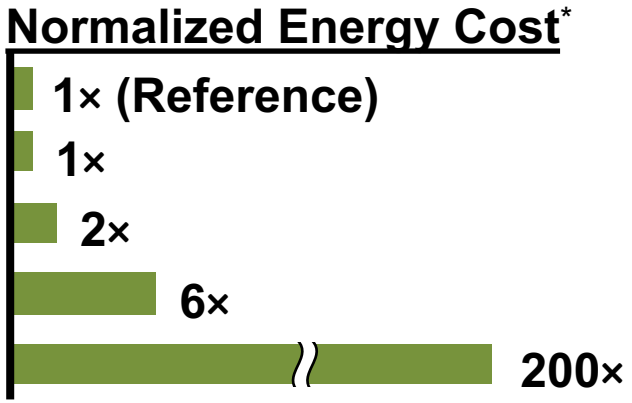
Evaluate Reuse in Different Dataflows

- **Weight Stationary**
  - Minimize movement of filter weights
- **Output Stationary**
  - Minimize movement of partial sums
- **No Local Reuse**
  - Don’t use any local PE storage. Maximize global buffer size.
- **Row Stationary**

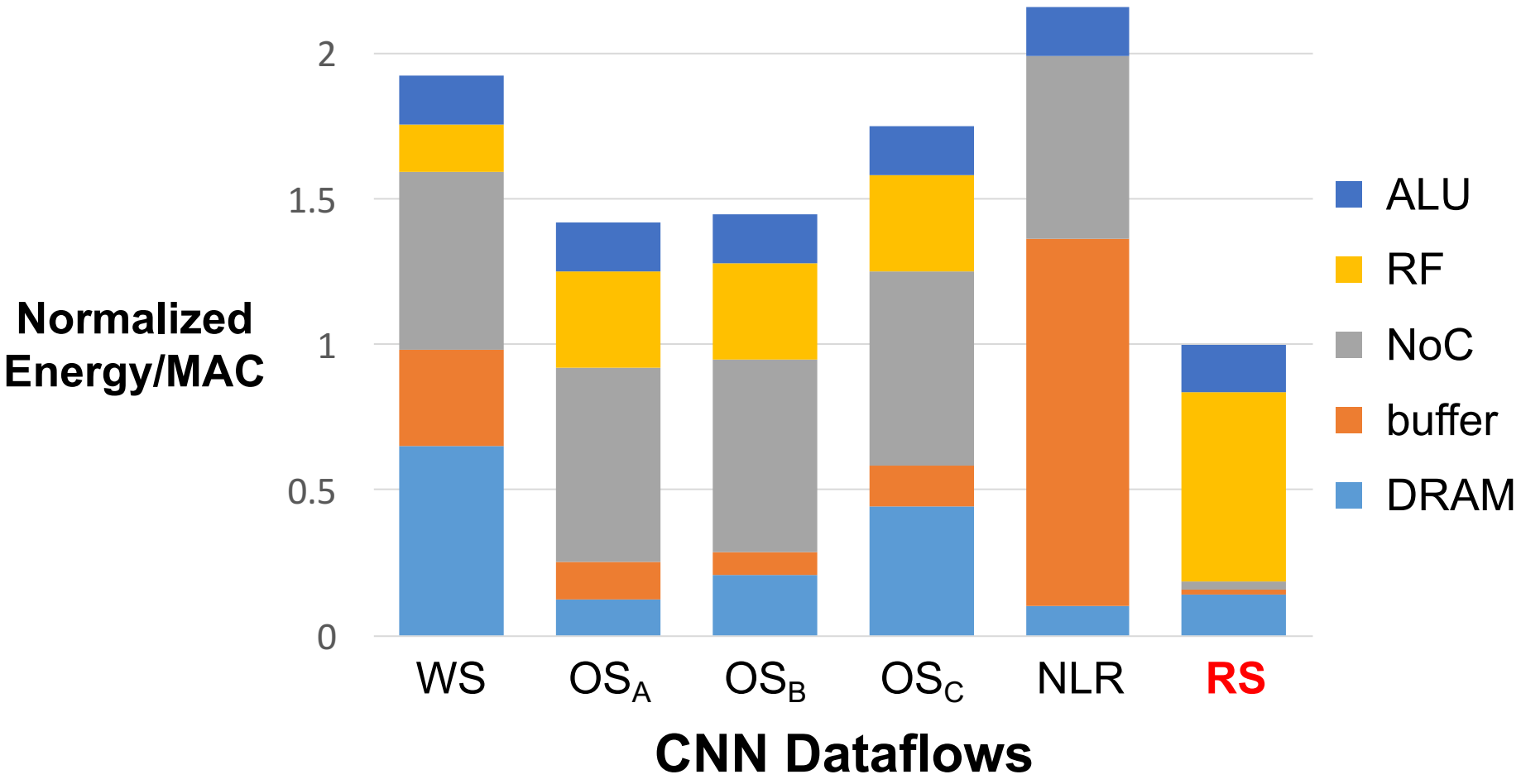
Evaluation Setup

- Same Total Area
- AlexNet
- 256 PEs
- Batch size = 16





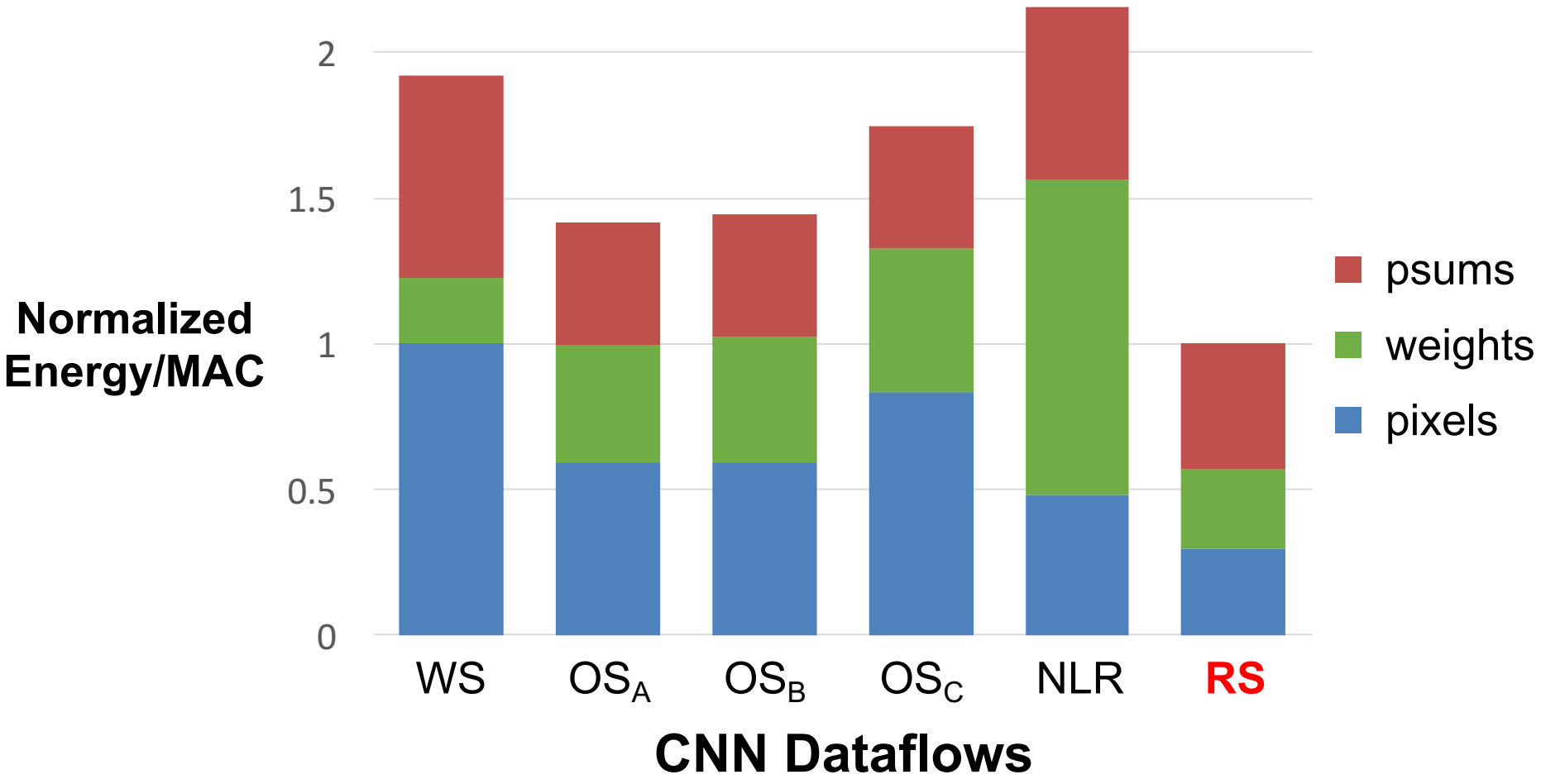
Dataflow Comparison: CONV Layers



RS uses **1.4× – 2.5× lower** energy than other dataflows

[Chen et al., ISCA 2016]

Dataflow Comparison: CONV Layers

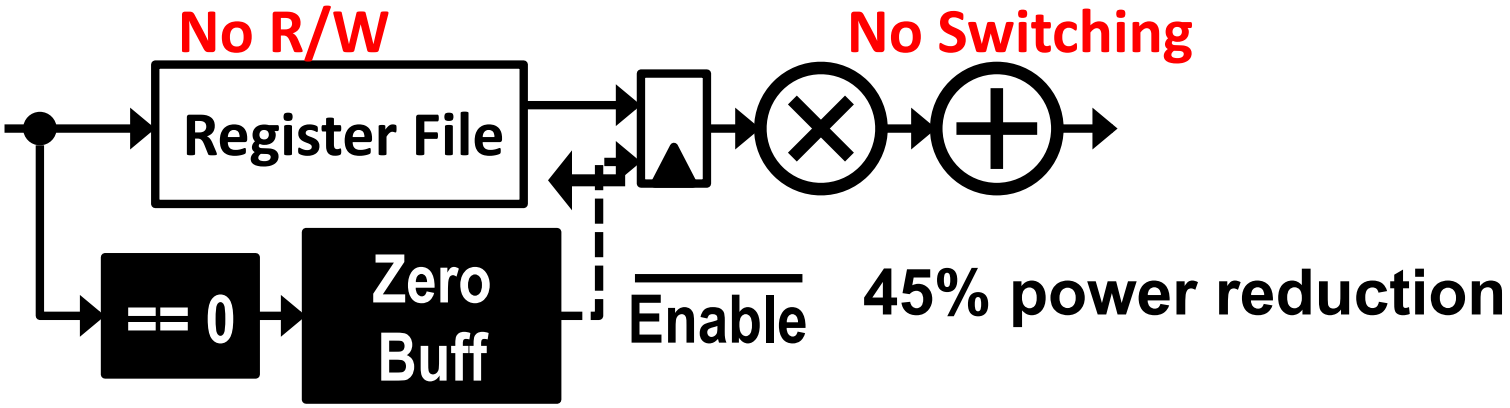


RS optimizes for the best **overall** energy efficiency

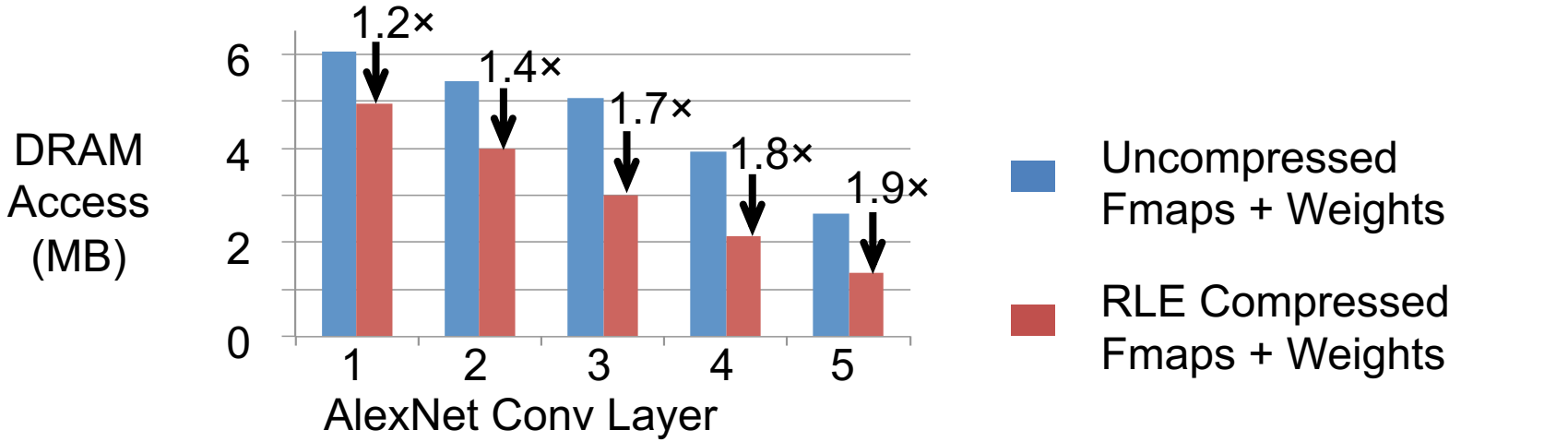
[Chen et al., ISCA 2016]

Exploit Sparsity

Method 1. Skip memory access and computation

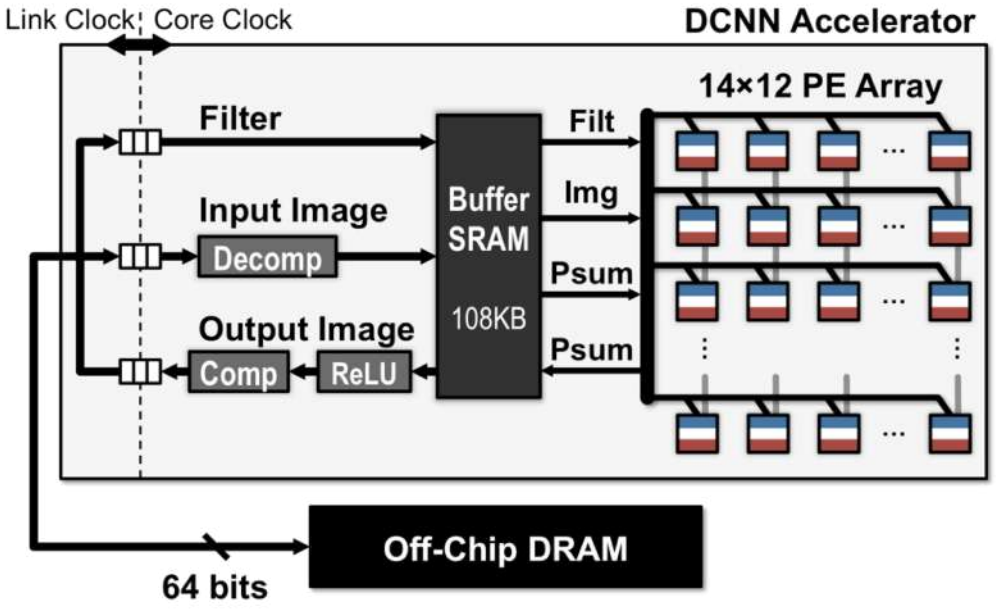


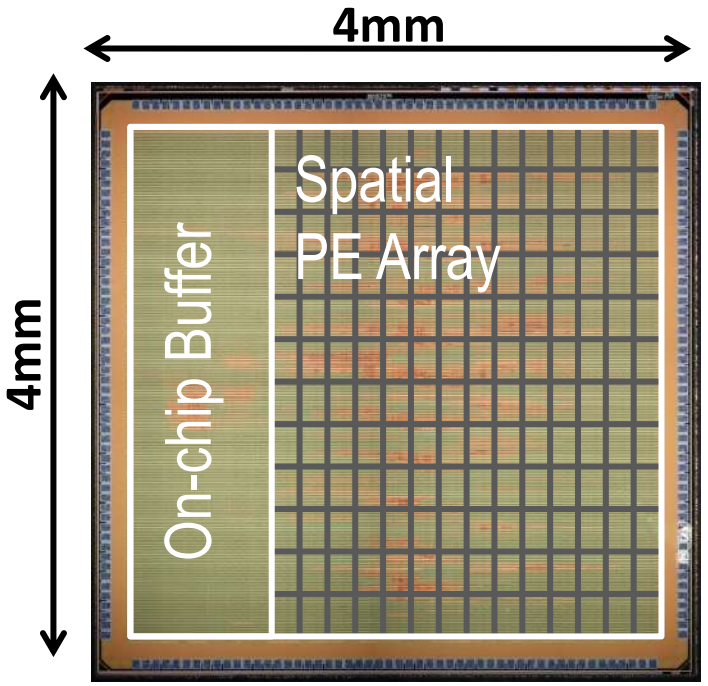
Method 2. Compress data to reduce storage and data movement



[Chen et al., ISSCC 2016]

Eyeriss: Deep Neural Network Accelerator





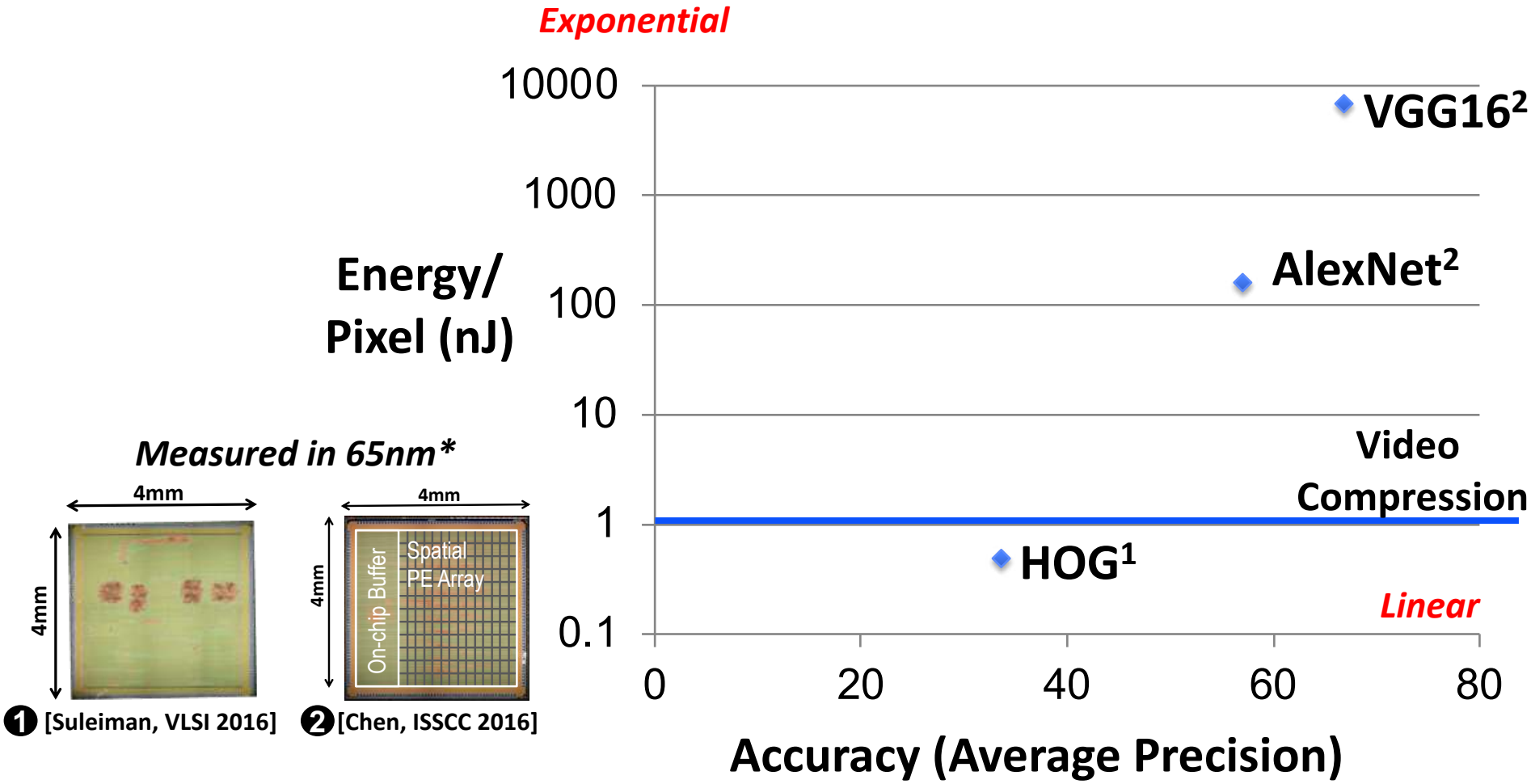
[Chen et al., ISSCC 2016, ISCA 2016]

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



** 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]

Energy-Efficient Processing of DNNs

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

Hardware Architectures for Deep Neural Networks

ISCA Tutorial

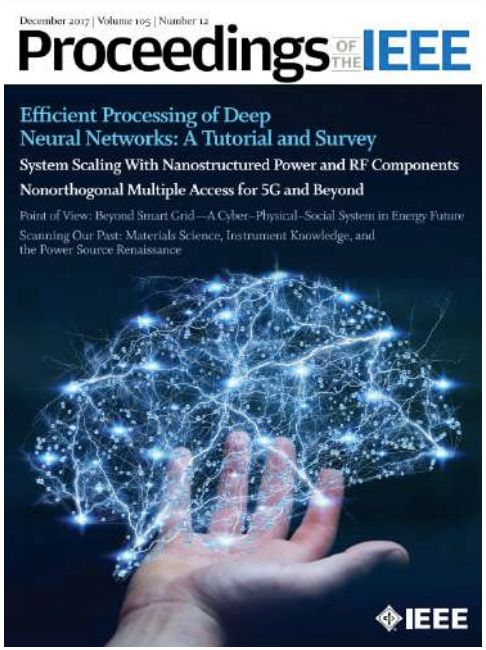
June 24, 2017

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

Massachusetts Institute of Technology

NVIDIA

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



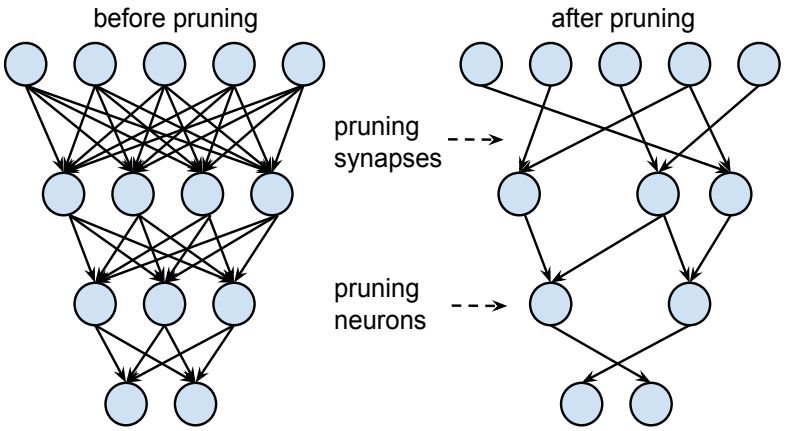
V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, "***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***," Proceedings of the IEEE, Dec. 2017

We identified various limitations to existing approaches

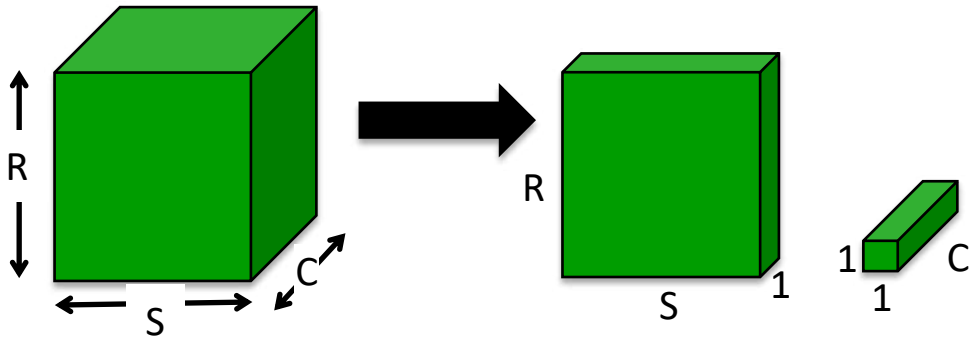
Design of Efficient DNN Algorithms

- Popular efficient DNN algorithm approaches

Network Pruning



Compact Network Architectures

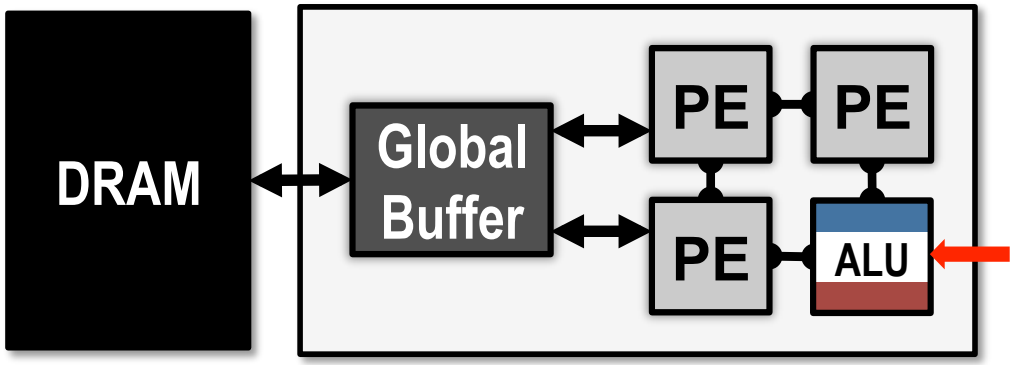


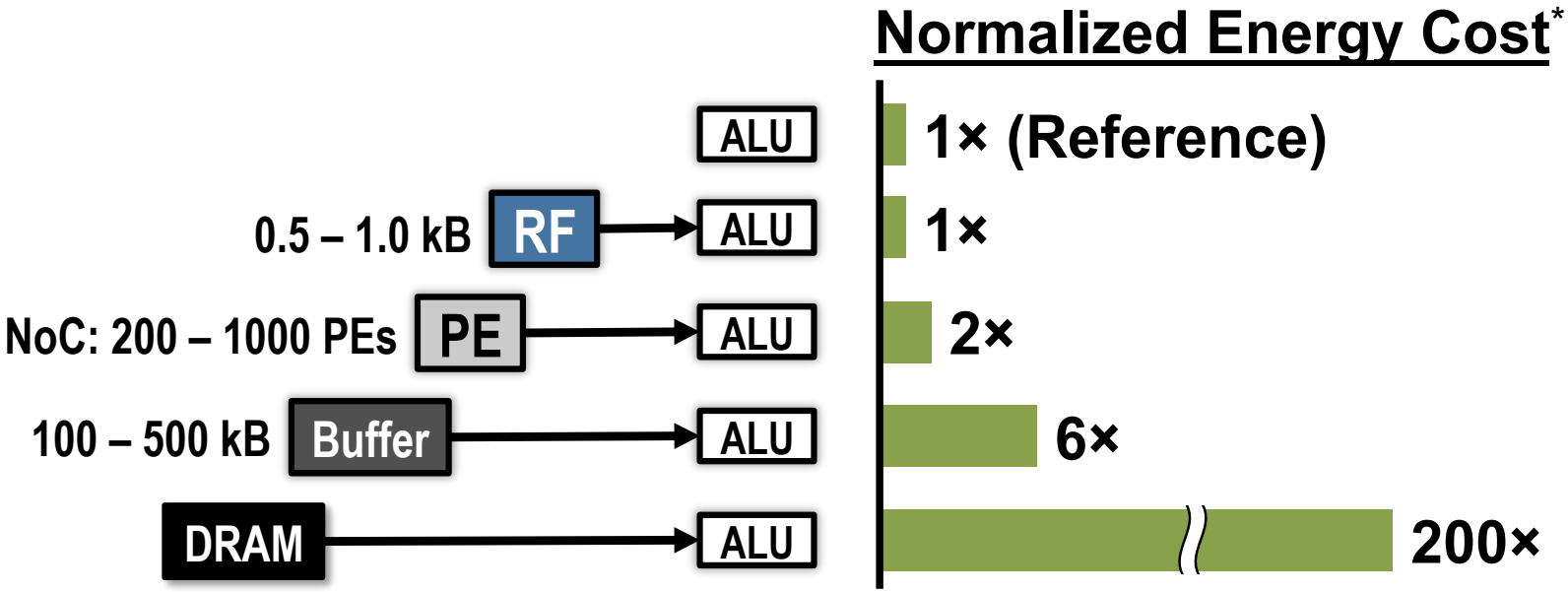
Examples: SqueezeNet, MobileNet

... also reduced precision

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

Data Movement is Expensive

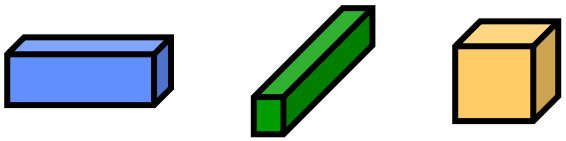




* measured from a commercial 65nm process

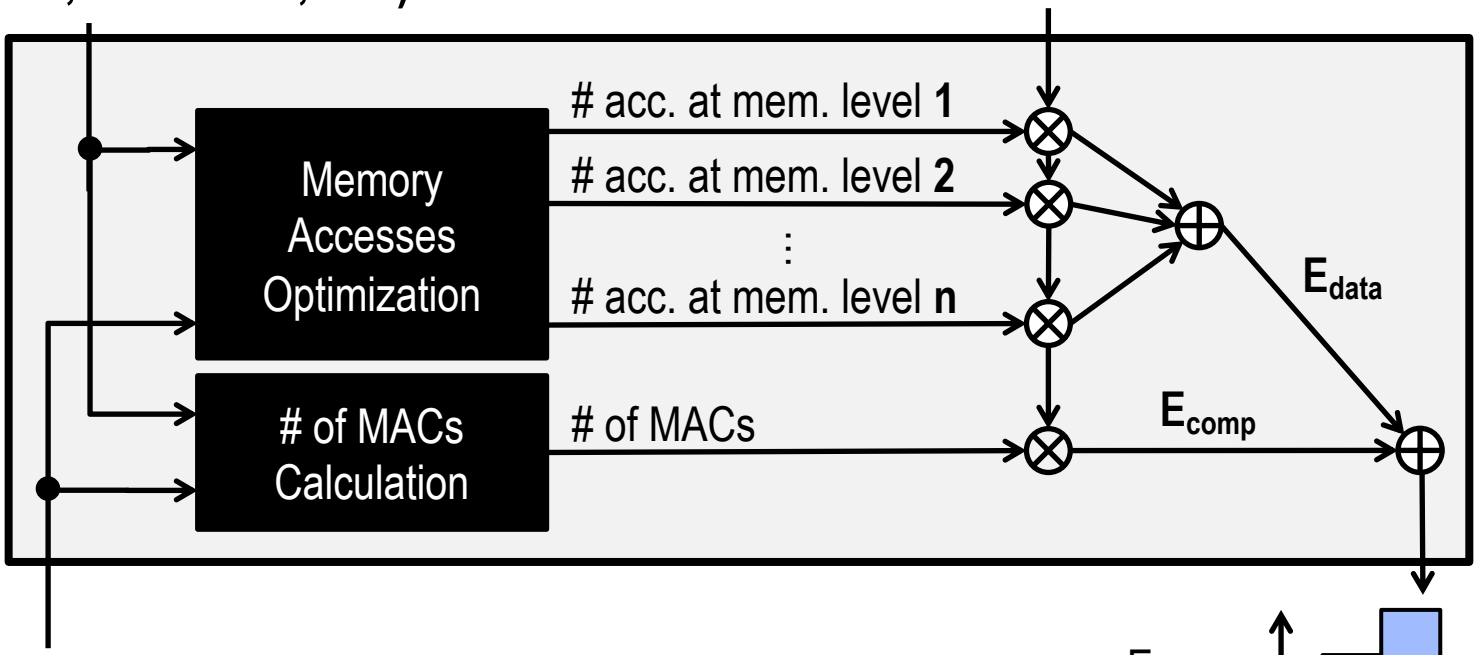
Energy of weight depends on **memory hierarchy** and **dataflow**

Energy-Evaluation Methodology



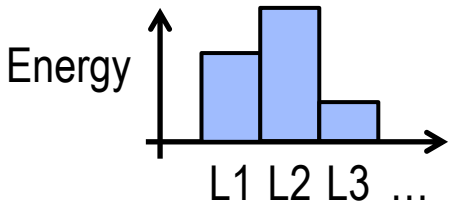
DNN Shape Configuration
(# of channels, # of filters, etc.)

Hardware Energy Costs of each MAC and Memory Access



DNN Weights and Input Data

[0.3, 0, -0.4, 0.7, 0, 0, 0.1, …]



DNN Energy Consumption

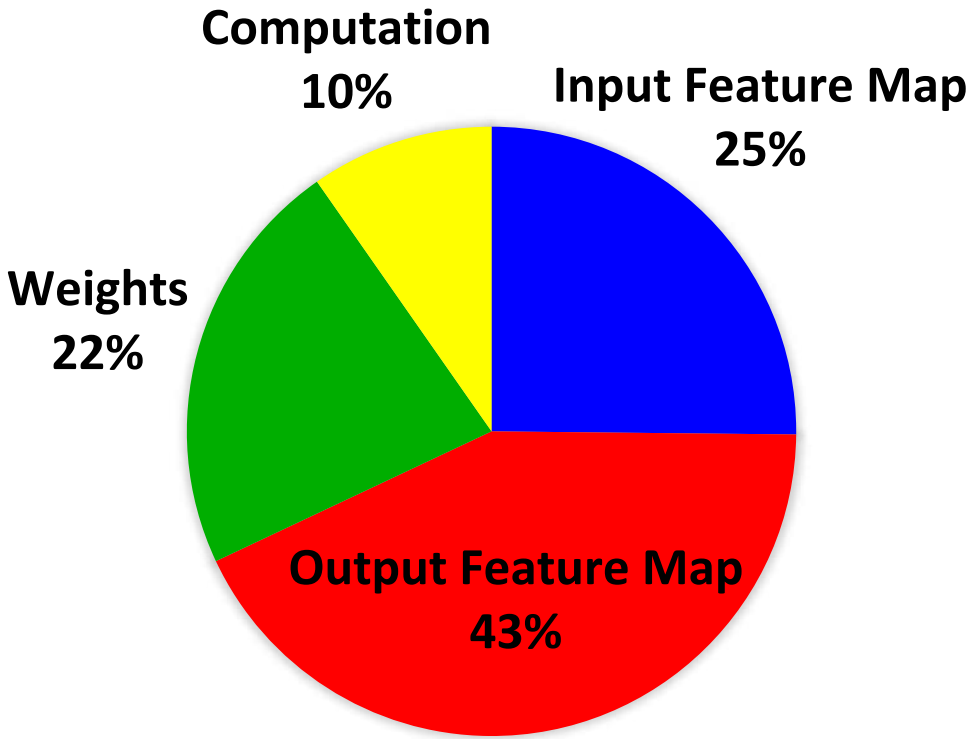
Tool available at: https://energyestimation.mit.edu/

[Yang et al., CVPR 2017]

Key Observations

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

Energy Consumption of GoogLeNet

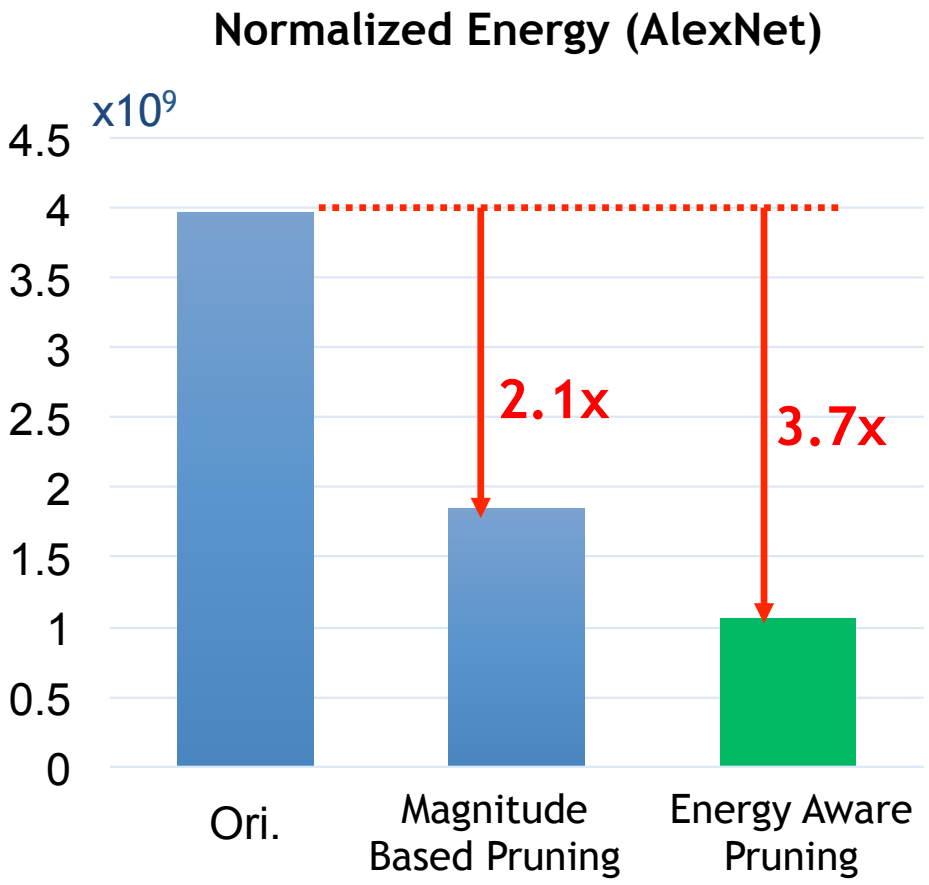


[Yang et al., CVPR 2017]

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**

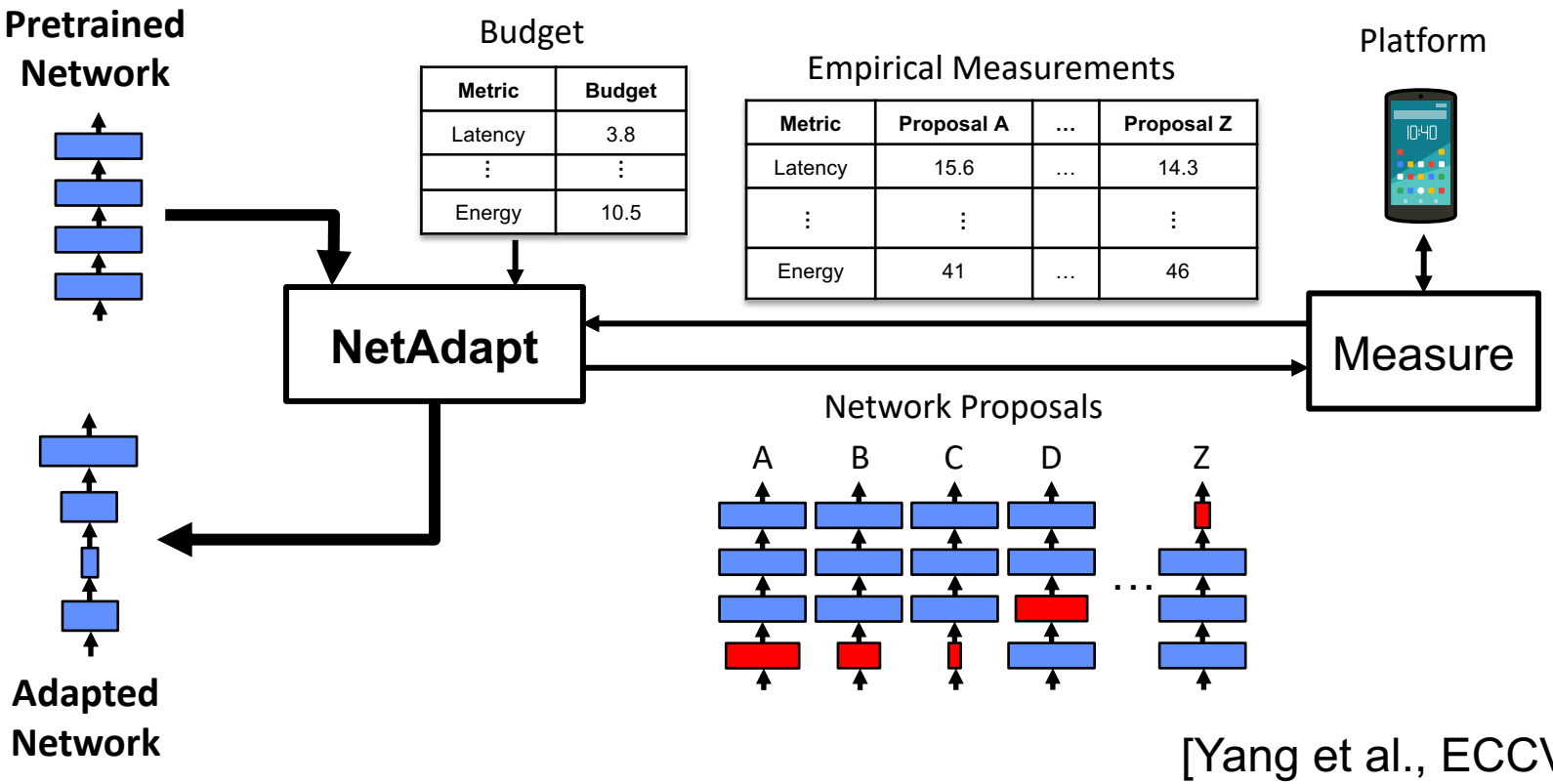


Pruned models available at
http://eyeriss.mit.edu/energy.html

[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]

Problem Formulation

$$\max_{Net} Accuracy(Net) \ \ subject\ to\ Resource_j(Net) \leq Budget_j, j = 1, \cdots, m$$

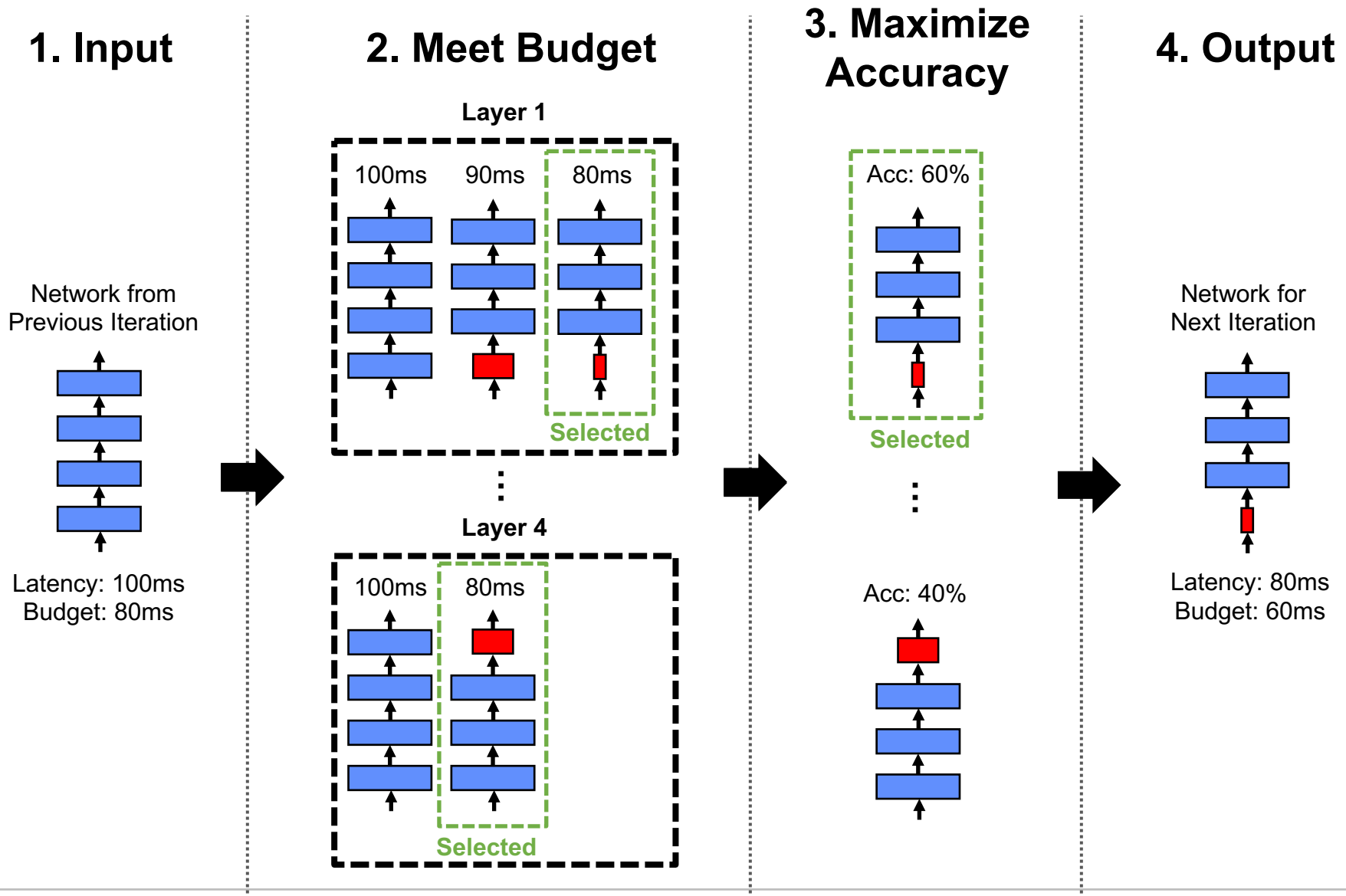
Break into a set of simpler problems and solve iteratively

$$\max_{Net_i} Acc(Net_i) \ \ subject\ to\ Res_j(Net_i) \leq Res_j(Net_{i-1}) - \Delta R_{i,j}, j = 1, \cdots, m$$

**Acc*: accuracy function, *Res*: resource evaluation function,
***ΔR*: resource reduction**, *Bud*: given budget
Budget incrementally tightens $\boldsymbol{Res_j(Net_{i-1}) - \Delta R_{i,j}}$

- **Advantages**
  - Supports multiple resource budgets at the same time
  - Guarantees that the budgets will be satisfied because the resource consumption decreases monotonically
  - Generates a family of networks (from each iteration) with different resource versus accuracy trade-offs
  - Intuitive and can easily set one additional hyperparameter ($\boldsymbol{\Delta R_{i,j}}$)

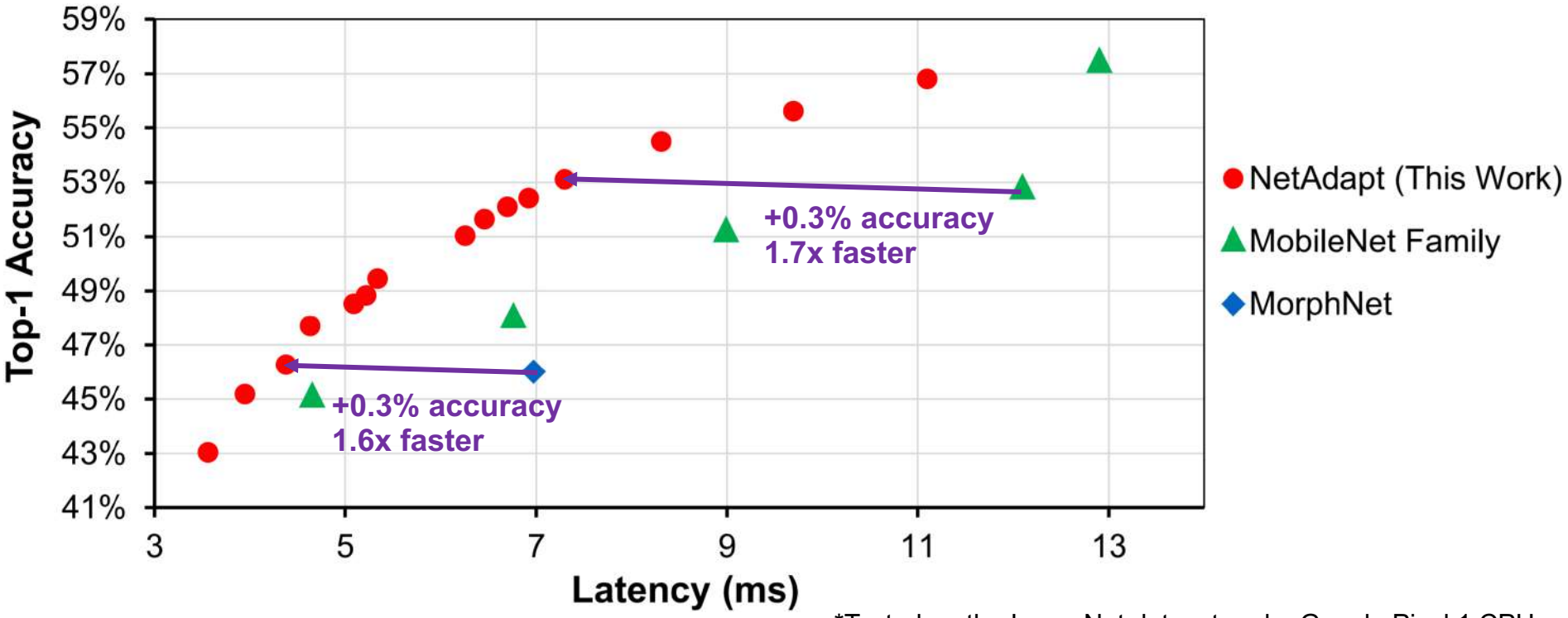
Simplified Example of One Iteration



[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

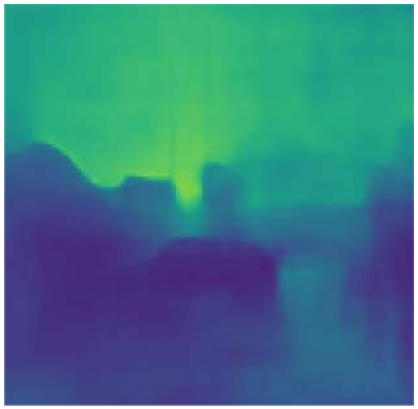
Reference:
MobileNet: Howard et al, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", arXiv 2017
MorphNet: Gordon et al., "Morphnet: Fast & simple resource-constrained structure learning of deep networks", CVPR 2018

FastDepth: Fast Monocular Depth Estimation

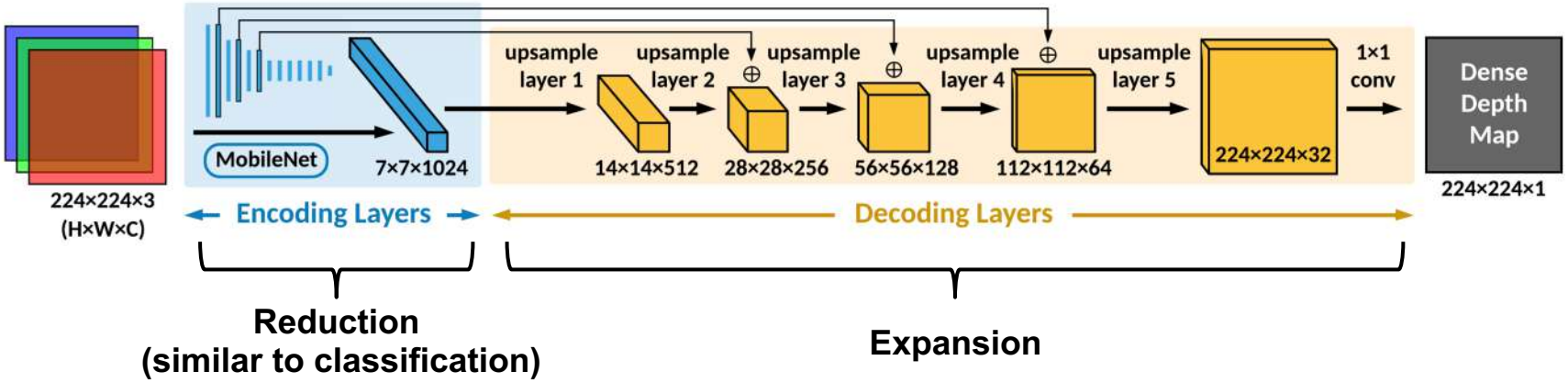
Depth estimation from a single RGB image desirable, due to the relatively low cost and size of monocular cameras.

RGB

Prediction

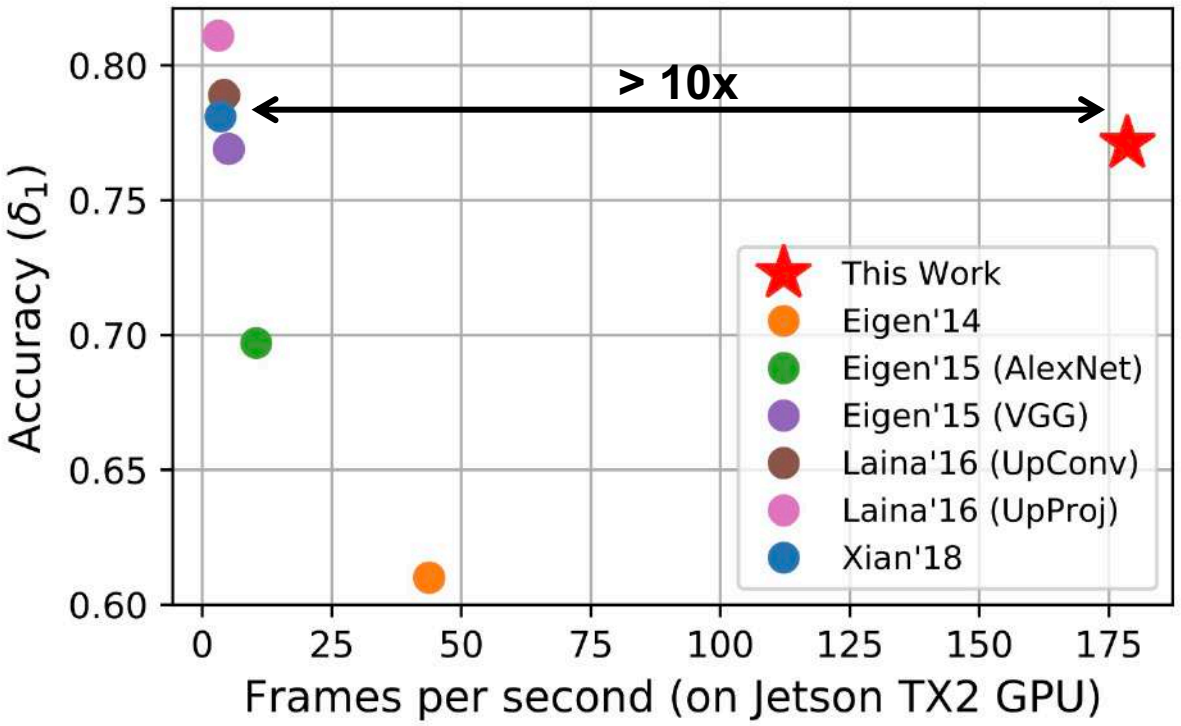


Auto Encoder DNN Architecture (Dense Output)



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

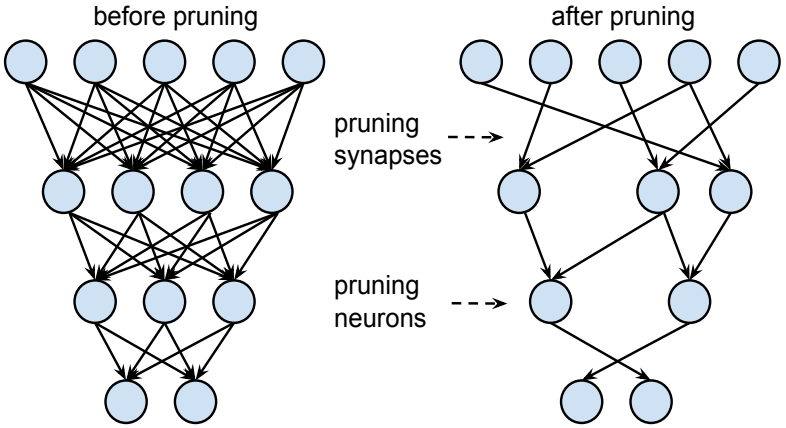


Configuration: Batch size of one (32-bit float)

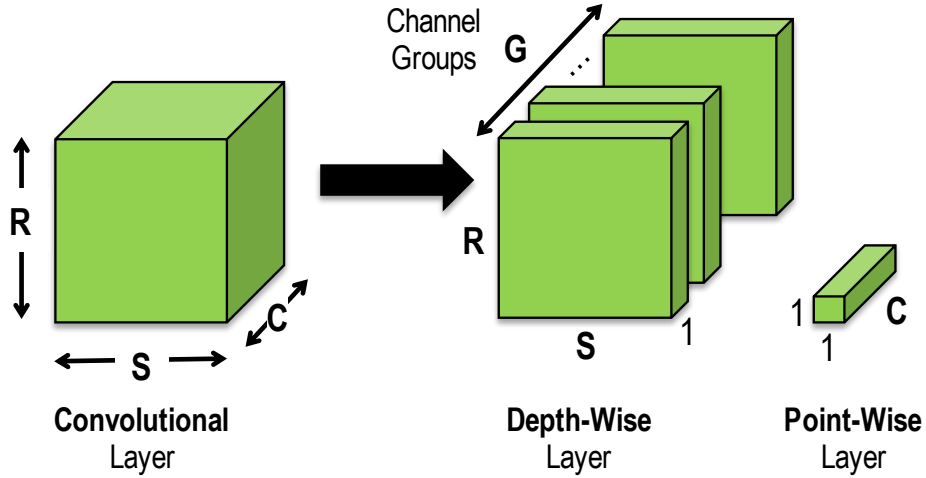
Models available at
http://fastdepth.mit.edu

Many Efficient DNN Design Approaches

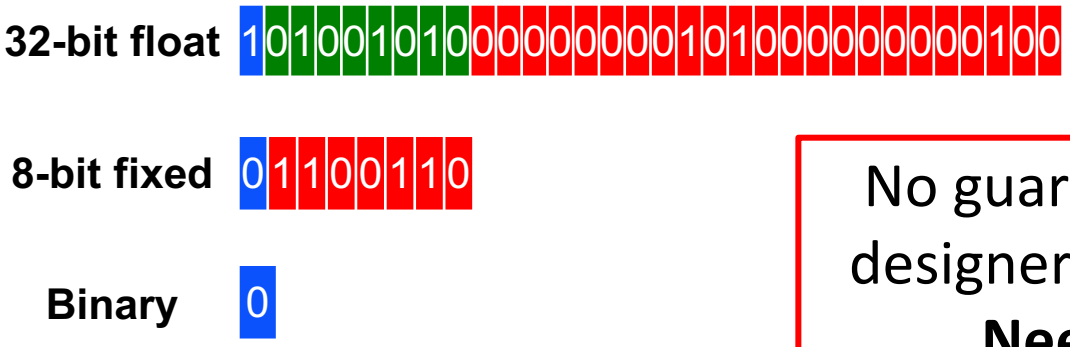
Network Pruning



Compact Network Architectures



Reduce Precision

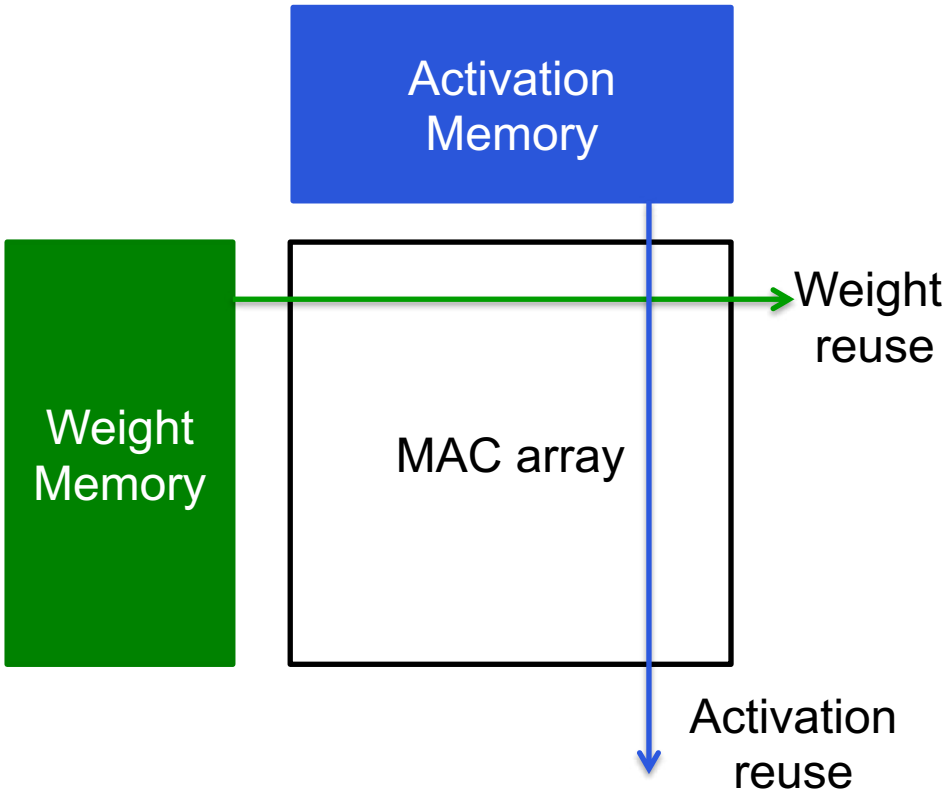


No guarantee that DNN algorithm designer will use a given approach.
Need flexible hardware!

[Chen et al., SysML 2018]

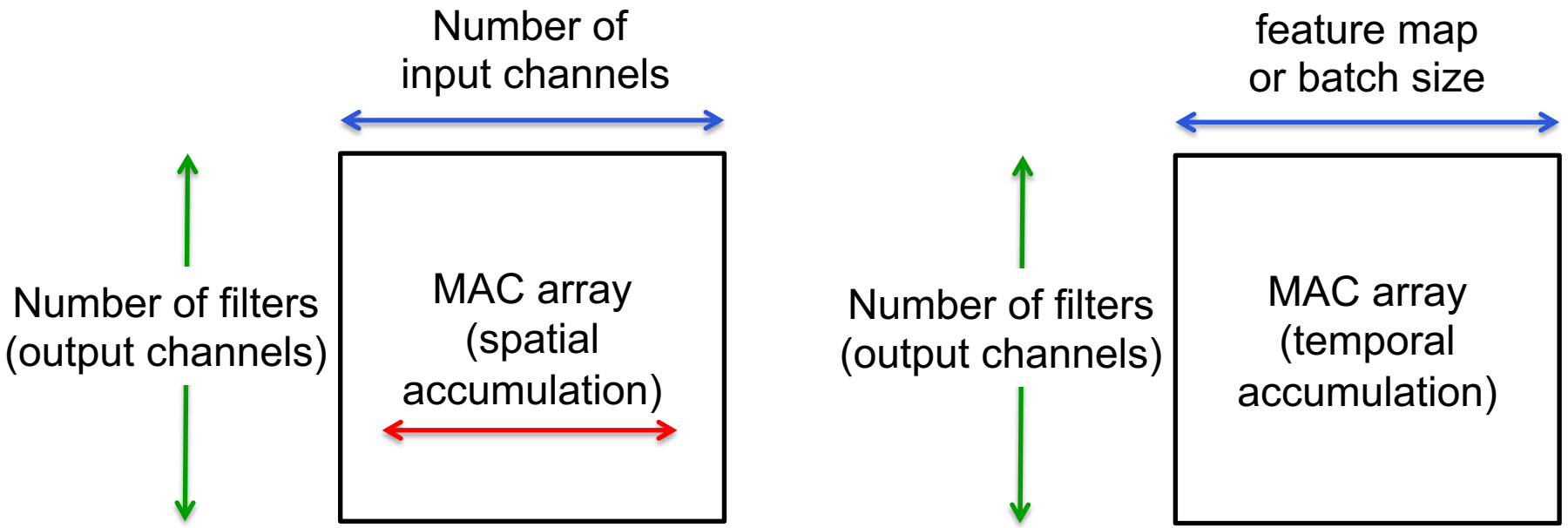
Existing DNN Architectures

- Specialized DNN hardware often rely on certain properties of DNN in order to achieve high energy-efficiency
- **Example:** Reduce memory access by amortizing across MAC array



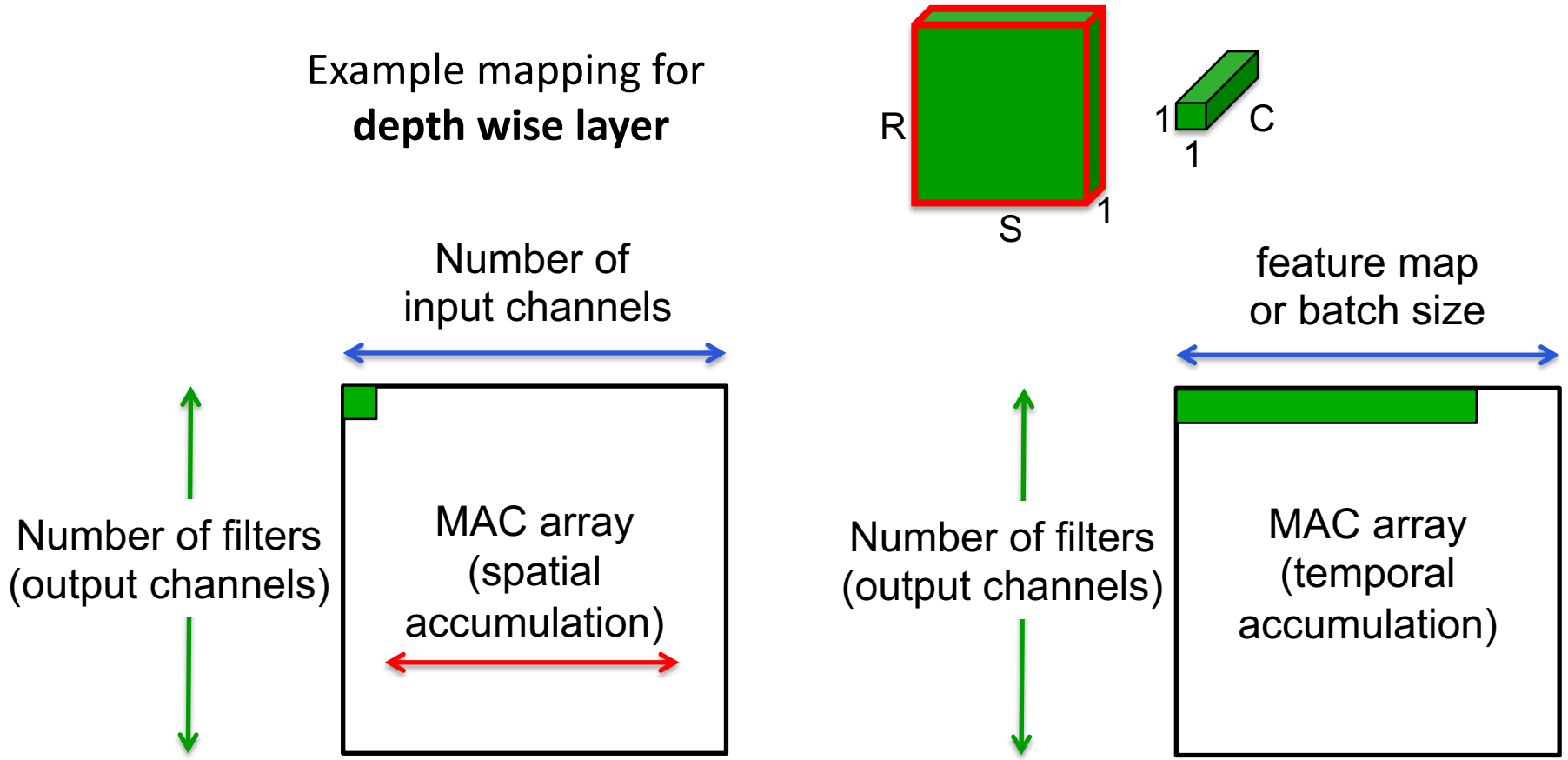
Limitation of Existing DNN Architectures

- **Example:** Reuse and array utilization depends on # of channels, feature map/batch size
  - Not efficient across all network architectures (e.g., compact DNNs)



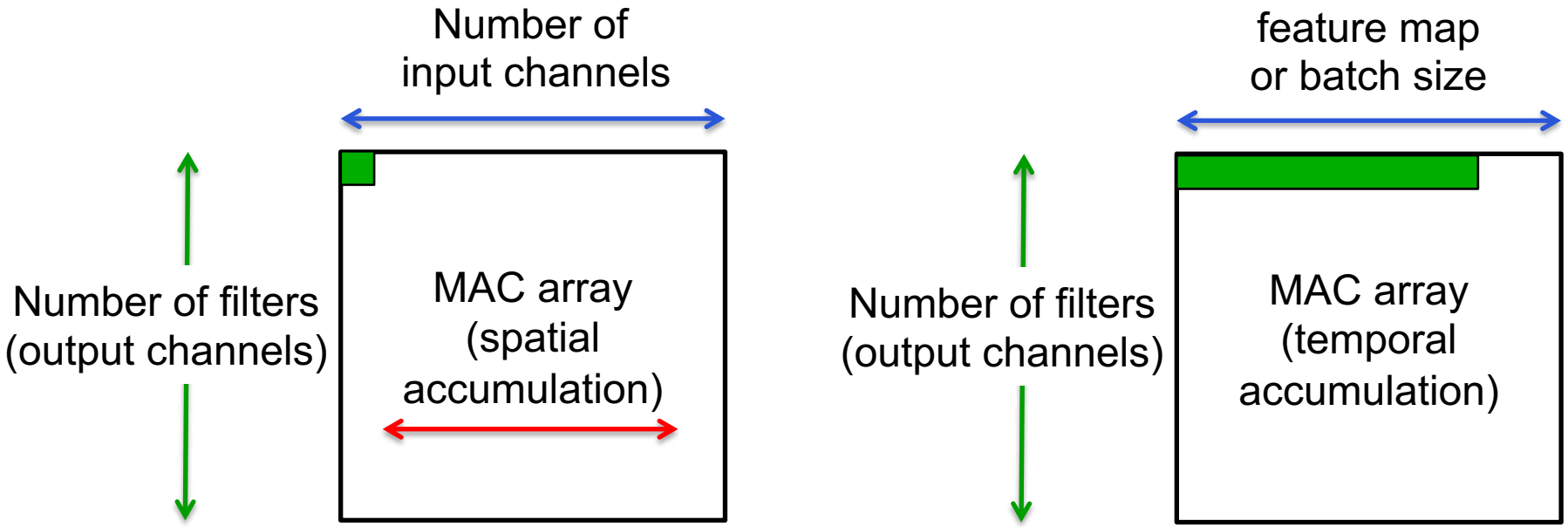
Limitation of Existing DNN Architectures

- **Example:** Reuse and array utilization depends on # of channels, feature map/batch size
  - Not efficient across all network architectures (e.g., compact DNNs)



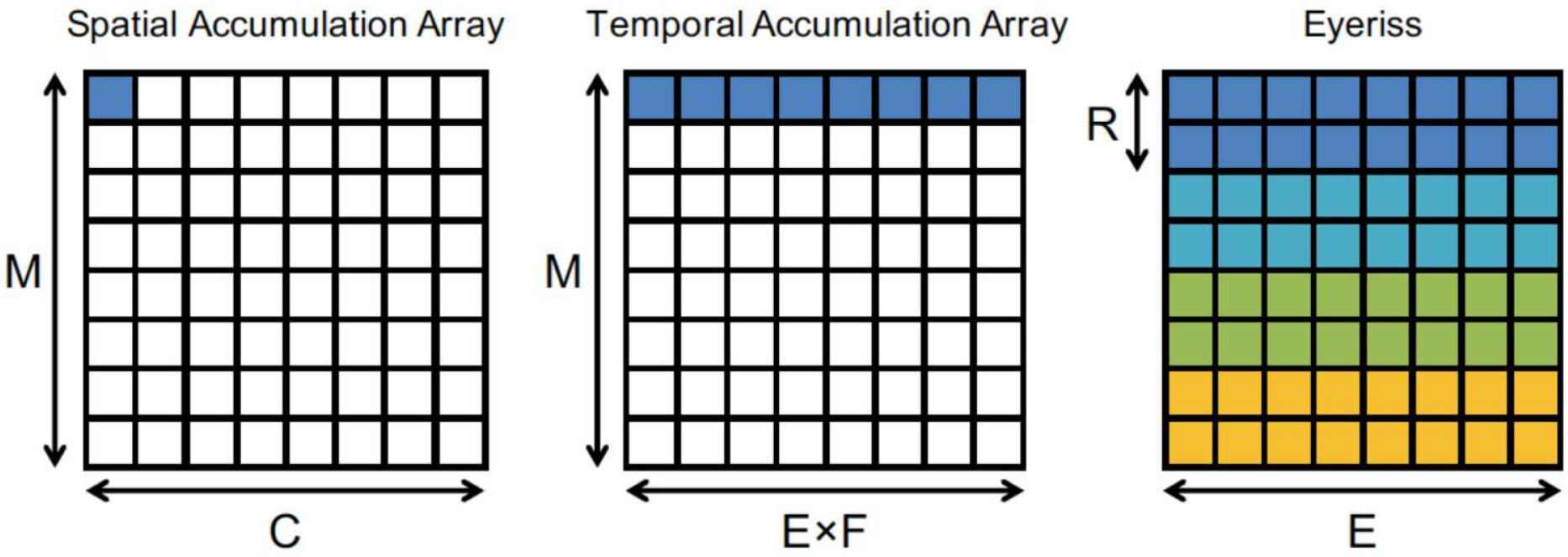
Limitation of Existing DNN Architectures

- **Example:** Reuse and array utilization depends on # of channels, feature map/batch size
  - Not efficient across all network architectures (e.g., compact DNNs)
  - Less efficient as array scales up in size
  - Can be challenging to exploit sparsity



Need Flexible Dataflow

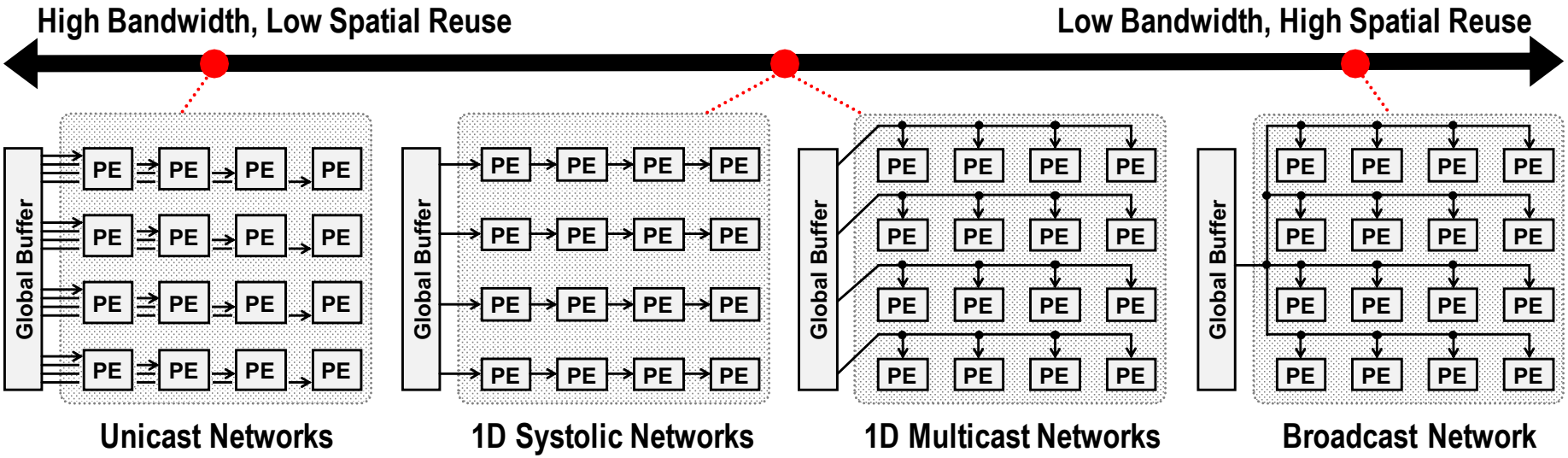
- Use flexible dataflow (**Row Stationary**) to exploit reuse in any dimension of DNN to increase energy efficiency and array utilization



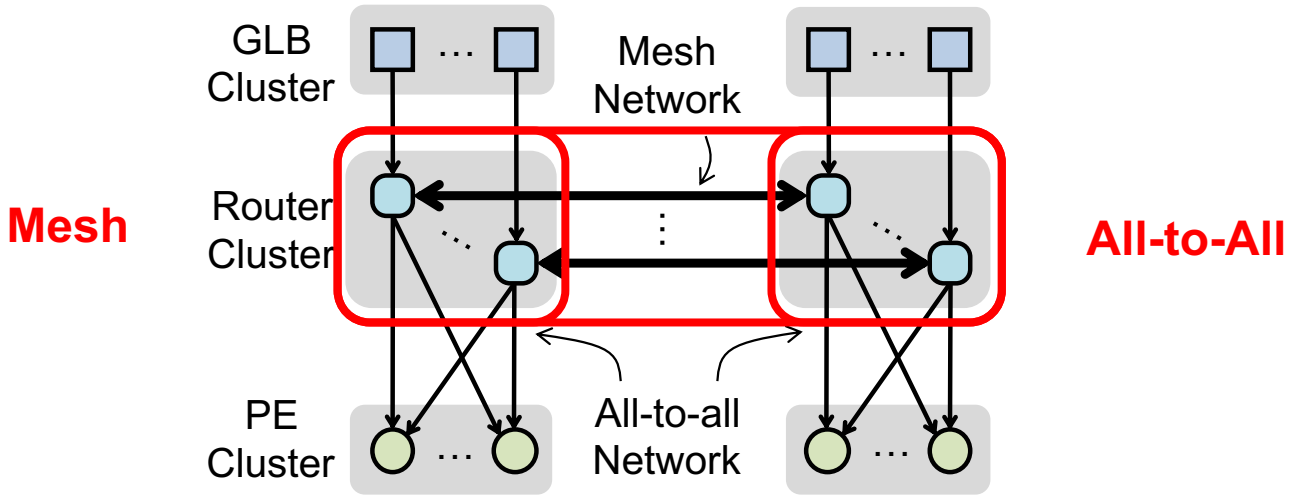
Example: Depth-wise layer

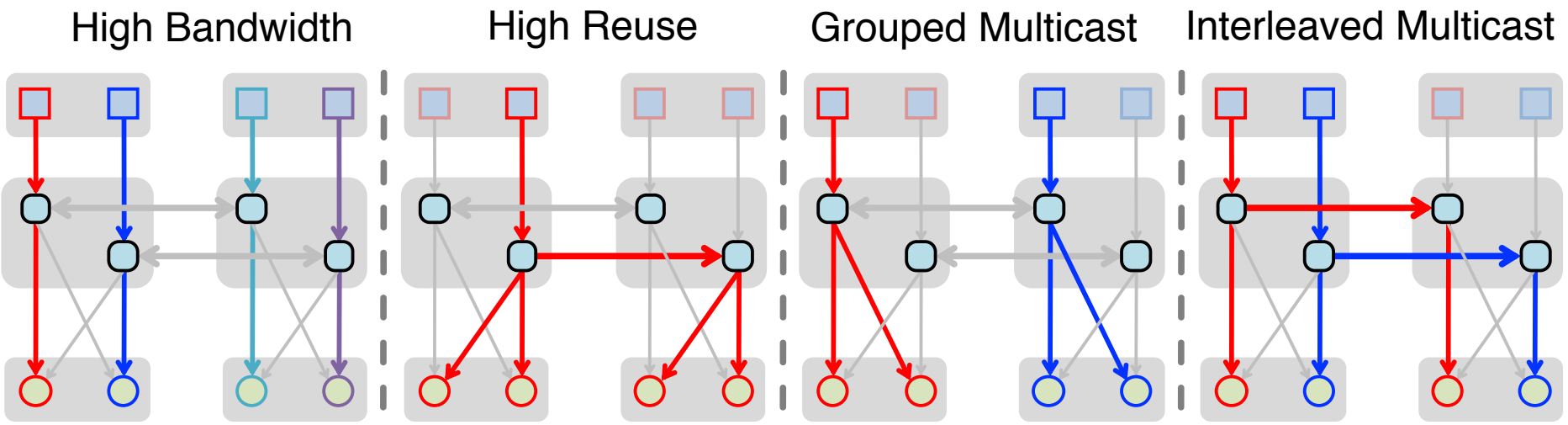
Need Flexible NoC for Varying Reuse

- When reuse available, need **multicast** to exploit spatial data reuse for energy efficiency and high array utilization
- When reuse not available, need **unicast** for high BW for weights for FC and weights & activations for high PE utilization
- An **all-to-all** satisfies above but too expensive and not scalable



Hierarchical Mesh



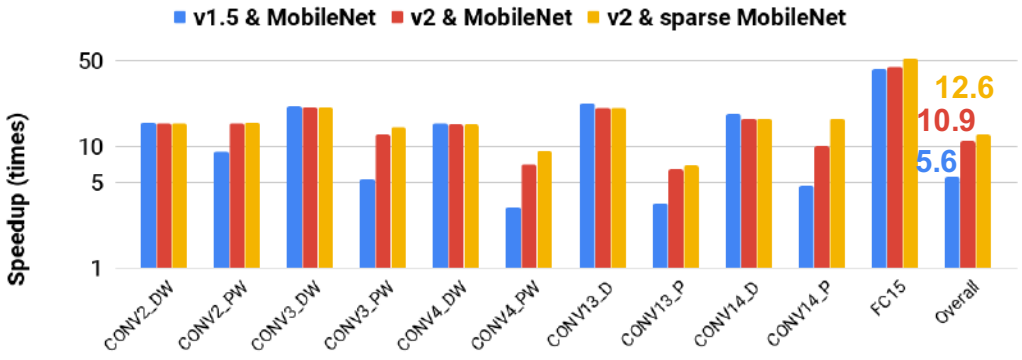


[Chen et al., JETCAS 2019]

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

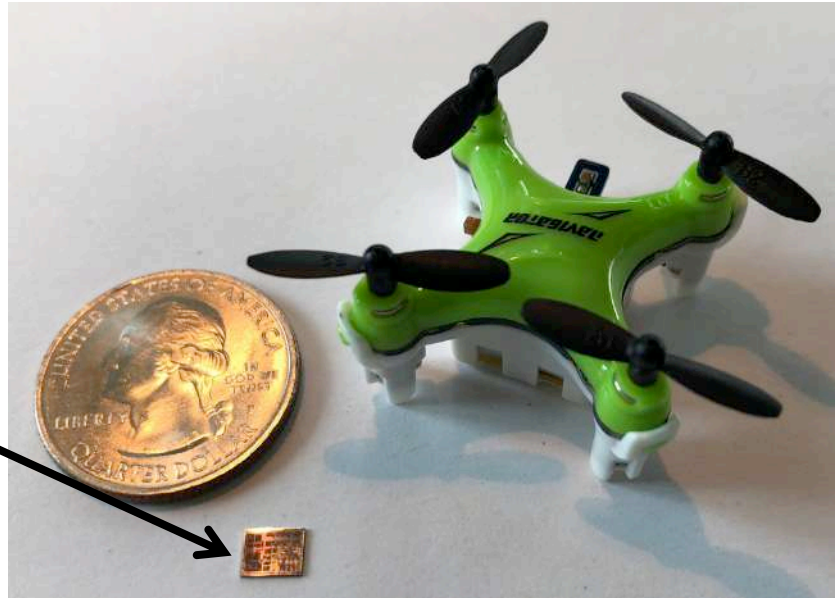
# of PEs	256	1024	16384
AlexNet	17.9x	71.5x	1086.7x
GoogLeNet	10.4x	37.8x	448.8x
MobileNet	15.7x	57.9x	873.0x

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

[Chen et al., JETCAS 2019]

Energy-Efficient Autonomous Navigation

Navion Chip
Localization and Mapping at 2mW
(full integration on-chip)



Enable energy-efficient navigation for **Search and Rescue**

http://navion.mit.edu

[Zhang et al., RSS 2017],
[Suleiman et al., VLSI 2018]

In collaboration with
Sertac Karaman (AeroAstro)

Visual-Inertial Localization

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

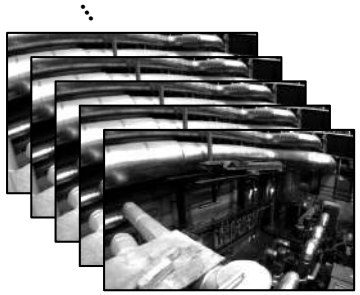
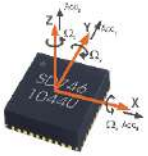


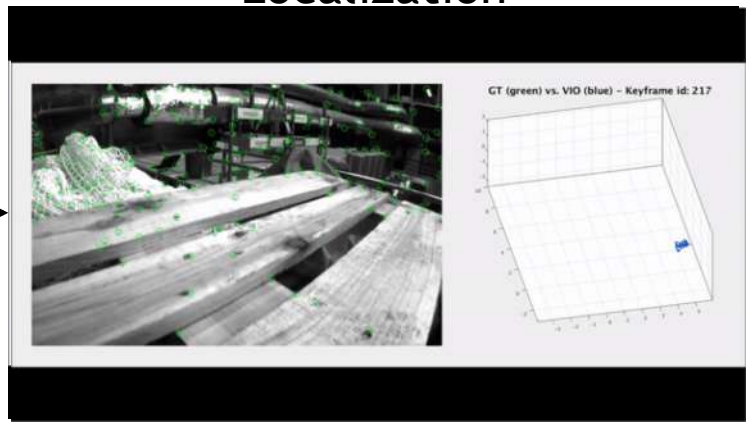
Image sequence

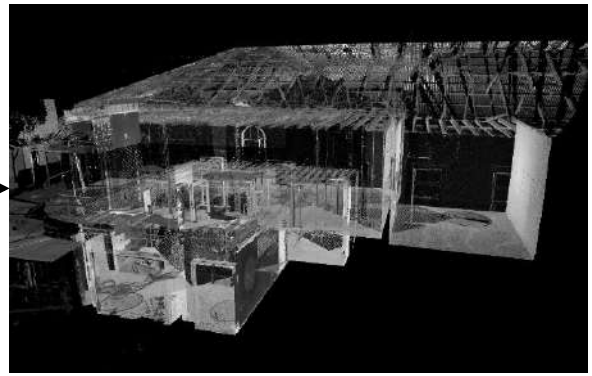
IMU
Inertial Measurement Unit



Visual-Inertial Odometry (VIO)*

Localization

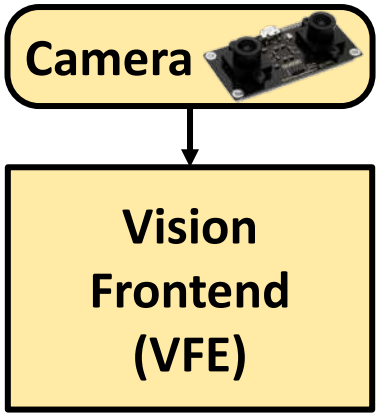




Mapping

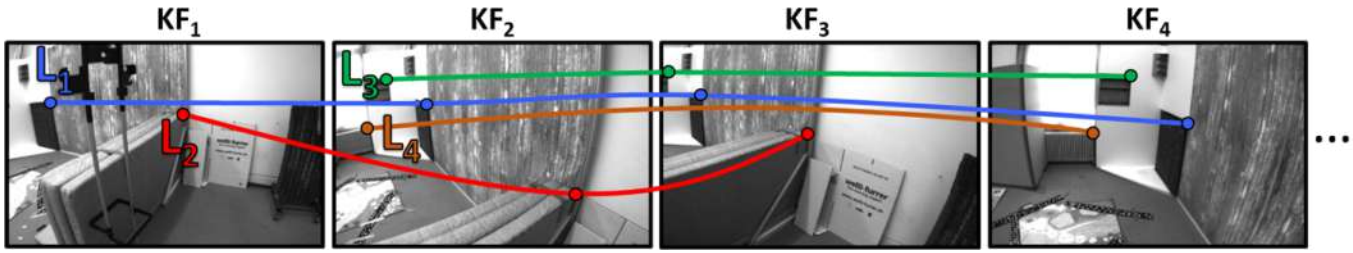
***Subset of SLAM algorithm**
(Simultaneous Localization And Mapping)

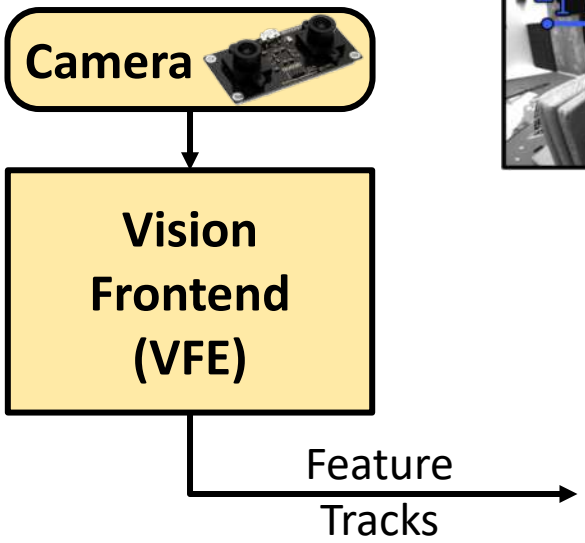
Frontend: Processing Sensors Data

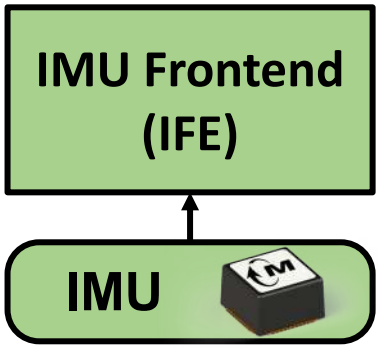


[Zhang et al., RSS 2017]

Frontend: Processing Sensors Data

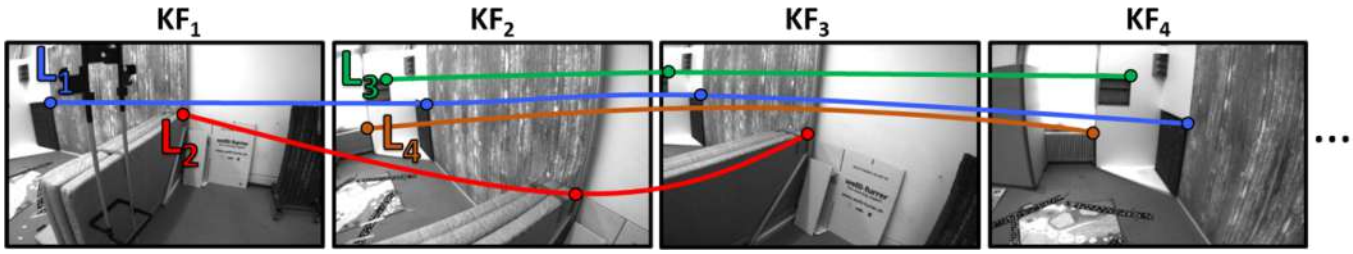


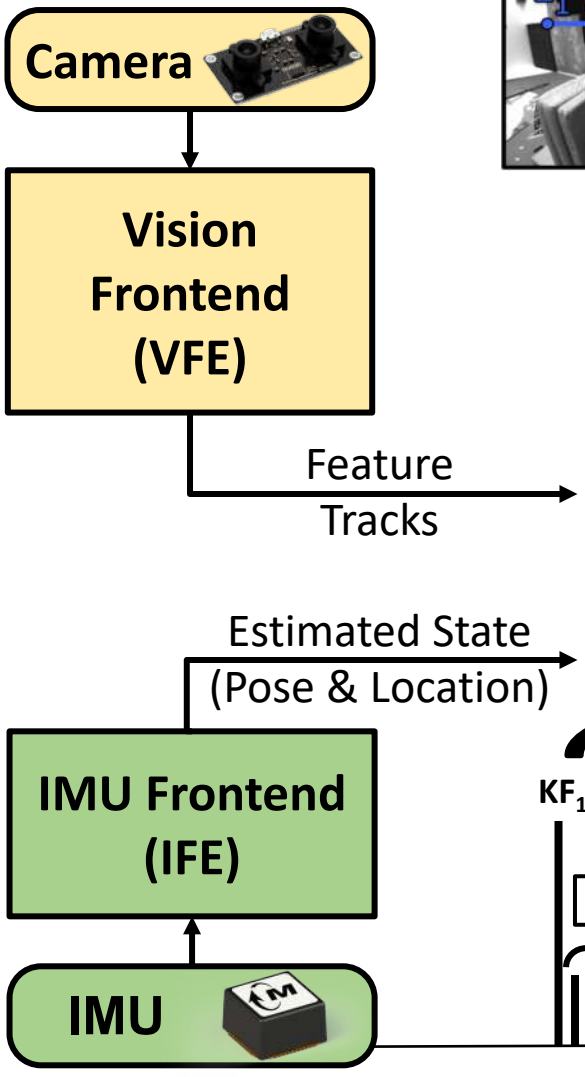


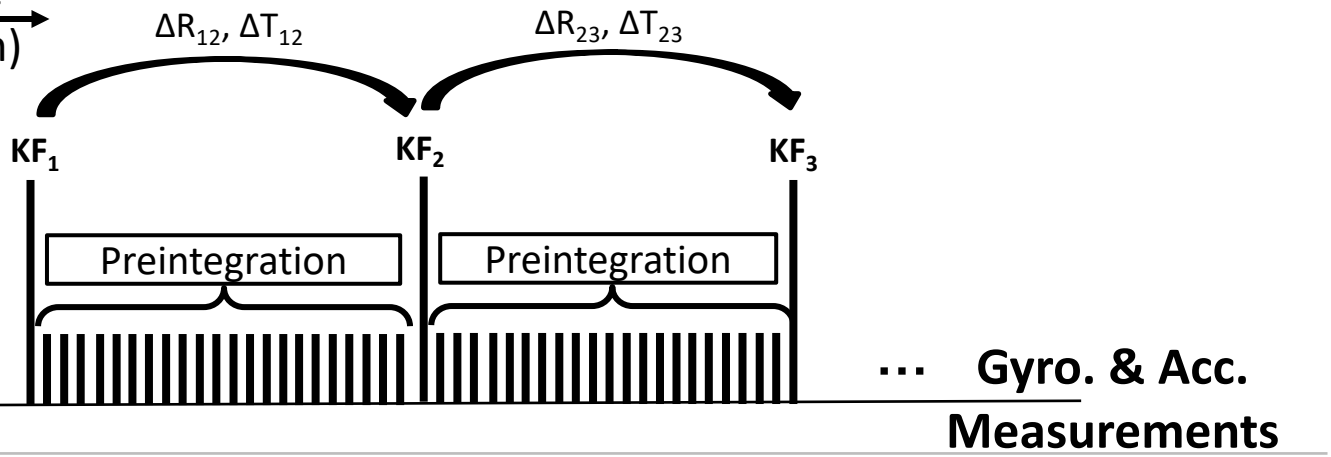


[Zhang et al., RSS 2017]

Frontend: Processing Sensors Data

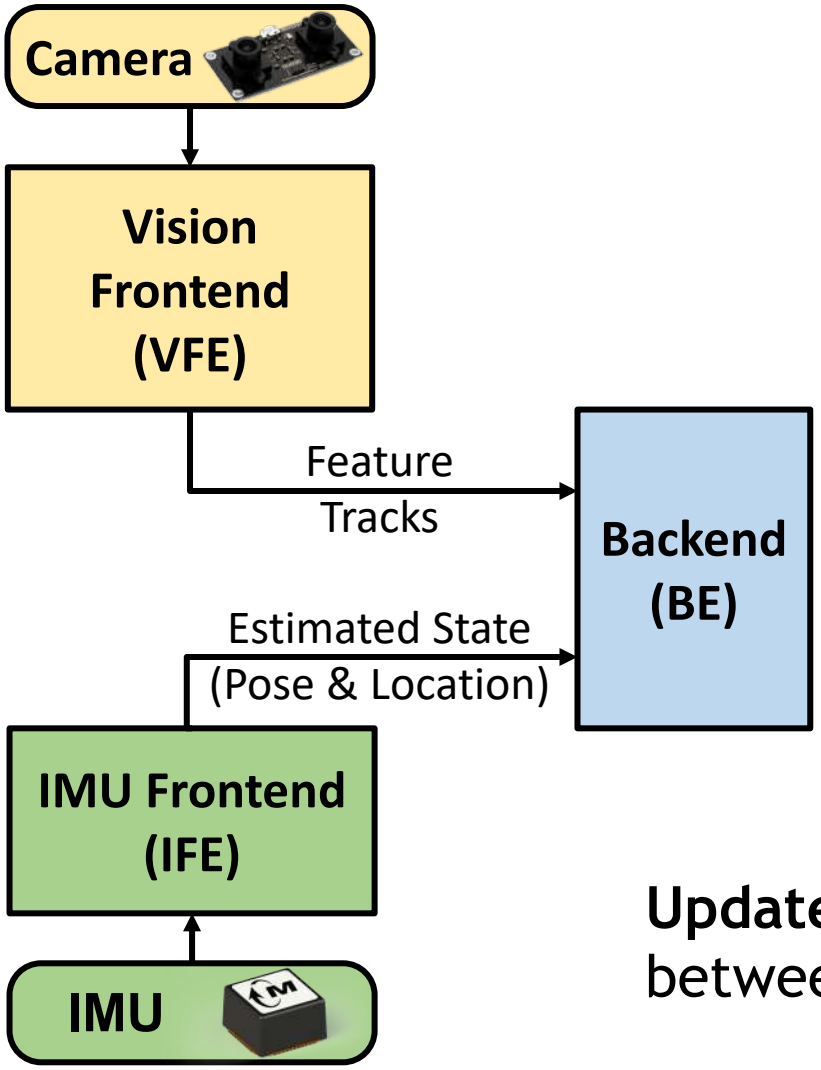


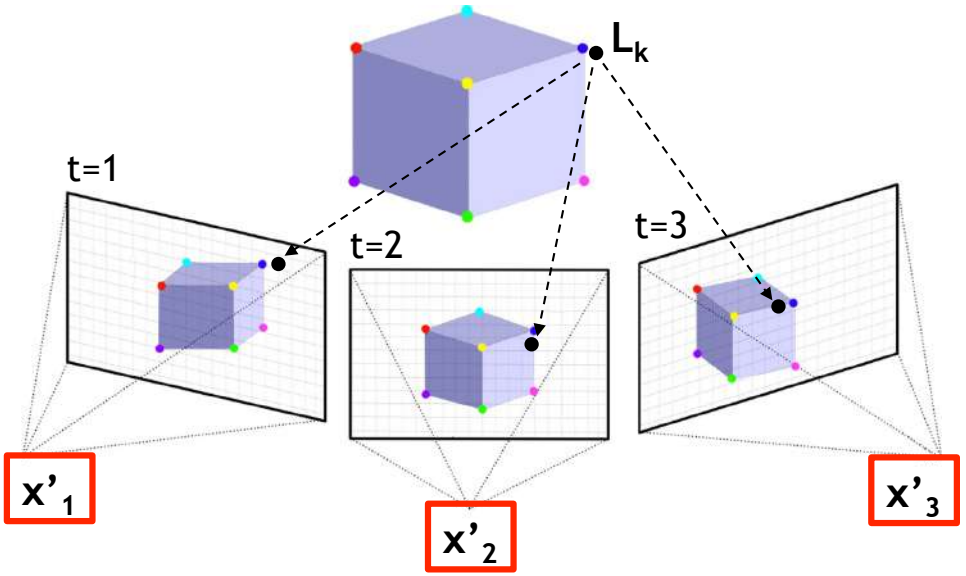




[Zhang et al., RSS 2017]

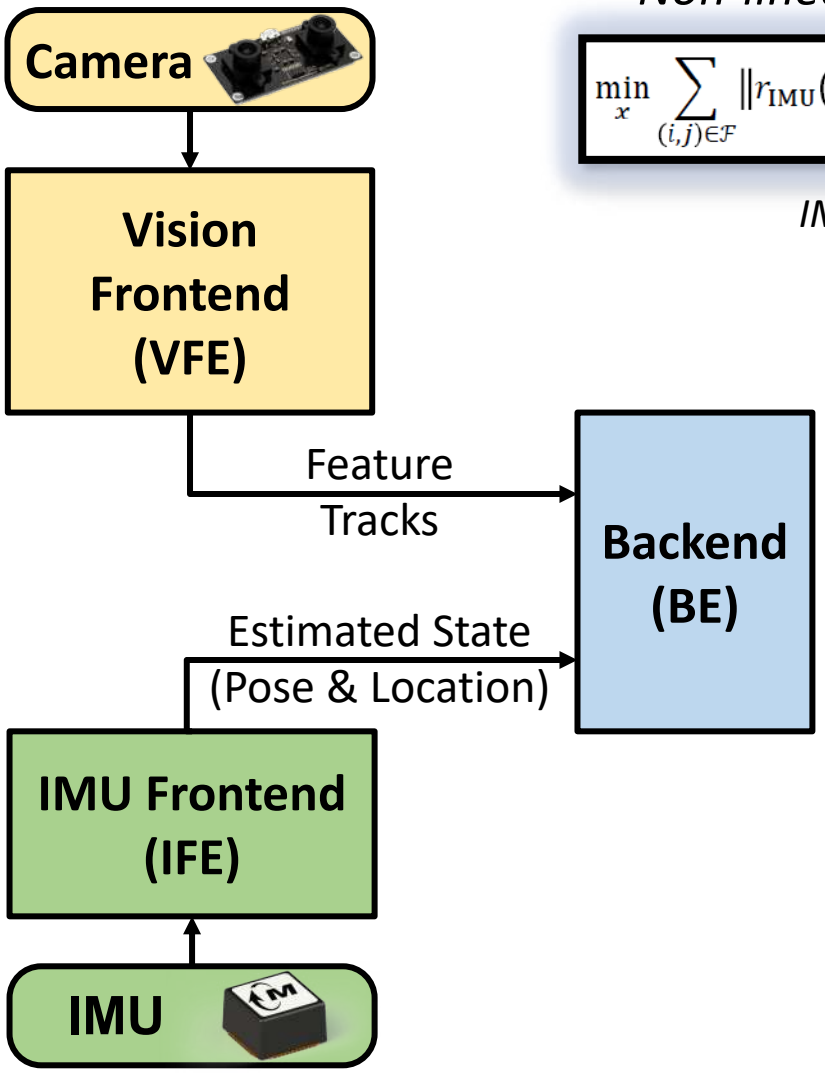
Backend: Reduce Inconsistency





Update states (x_i) to minimize inconsistencies between measurements across time

Backend: Factor Graph to Infer State of Drone

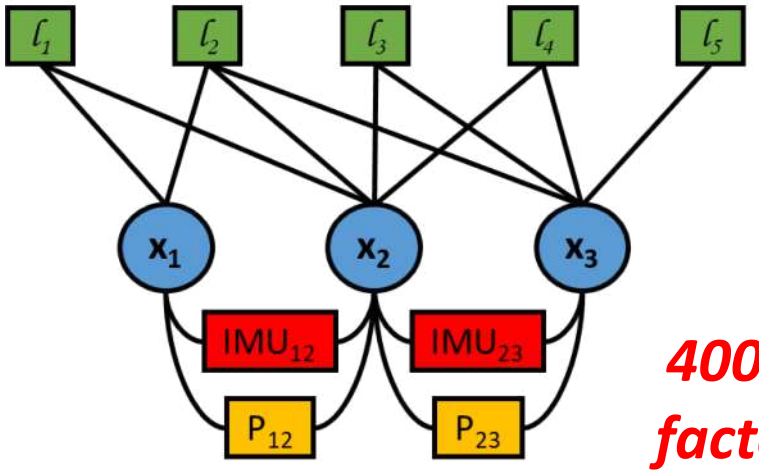


Non-linear least squares factor graph optimization

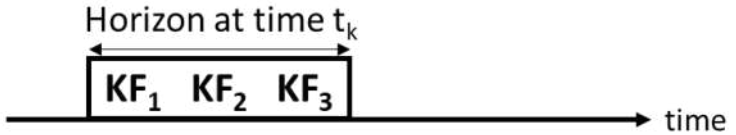
$$\min_{x} \sum_{(i,j)\in\mathcal{F}} \left\| r_{\text{IMU}}\left(x, \Delta\tilde{R}_{ij}, \Delta\tilde{p}_{ij}, \Delta\tilde{v}_{ij}\right) \right\|^2 + \sum_{k\in\mathcal{L}} \sum_{i\in\mathcal{F}_k} \left\| r_{\text{CAM}}\left(x, l_k, u_{ik}^l, u_{ik}^r\right) \right\|^2 + \left\| r_{\text{PRIOR}}(x) \right\|^2$$

IMU Factors *Vision Factors* *Other Factors*

Factor Graph



4000+ factors



[Zhang et al., RSS 2017]

Backend: Factor Graph to Infer State of Drone

Non-linear least squares factor graph optimization

$$\min_{x} \sum_{(i,j)\in\mathcal{F}} \left\| r_{\text{IMU}}(x, \Delta\tilde{R}_{ij}, \Delta\tilde{p}_{ij}, \Delta\tilde{v}_{ij}) \right\|^2 + \sum_{k\in\mathcal{L}} \sum_{i\in\mathcal{F}_k} \left\| r_{\text{CAM}}(x, l_k, u_{ik}^l, u_{ik}^r) \right\|^2 + \left\| r_{\text{PRIOR}}(x) \right\|^2$$

IMU Factors *Vision Factors* *Other Factors*

Vision Frontend (VFE)

Feature Tracks

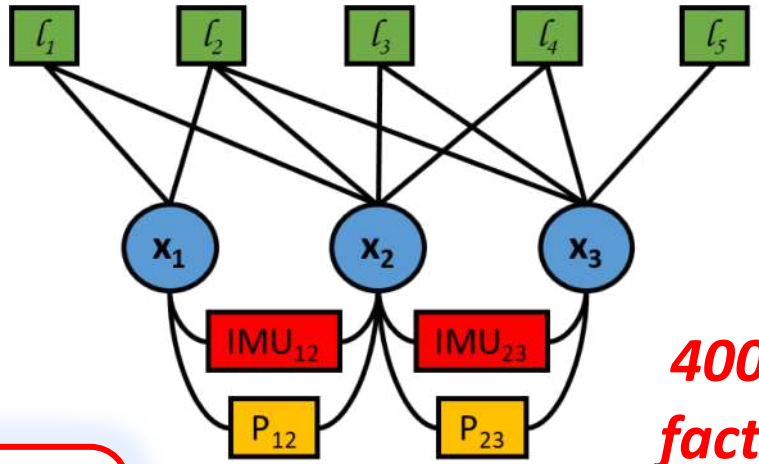
Backend (BE)

Estimated State (Pose & Location)

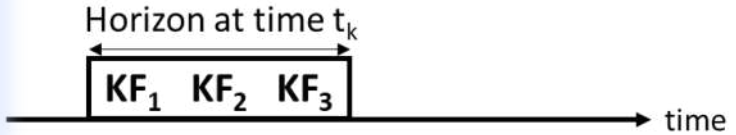
IMU Frontend (IFE)

Updated States (x_i) & Sparse 3D Map

Factor Graph

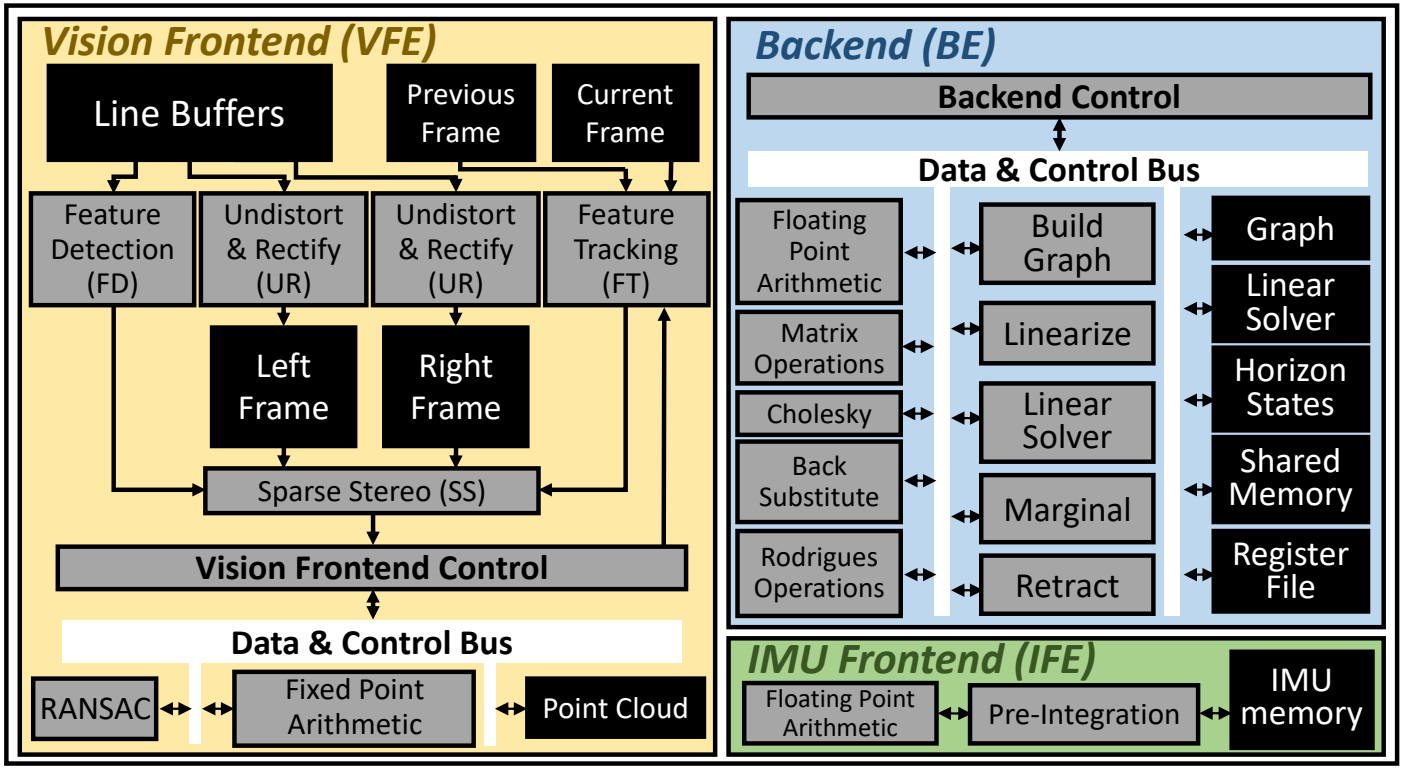


4000+ factors



[Zhang et al., RSS 2017]

Navion Chip Architecture



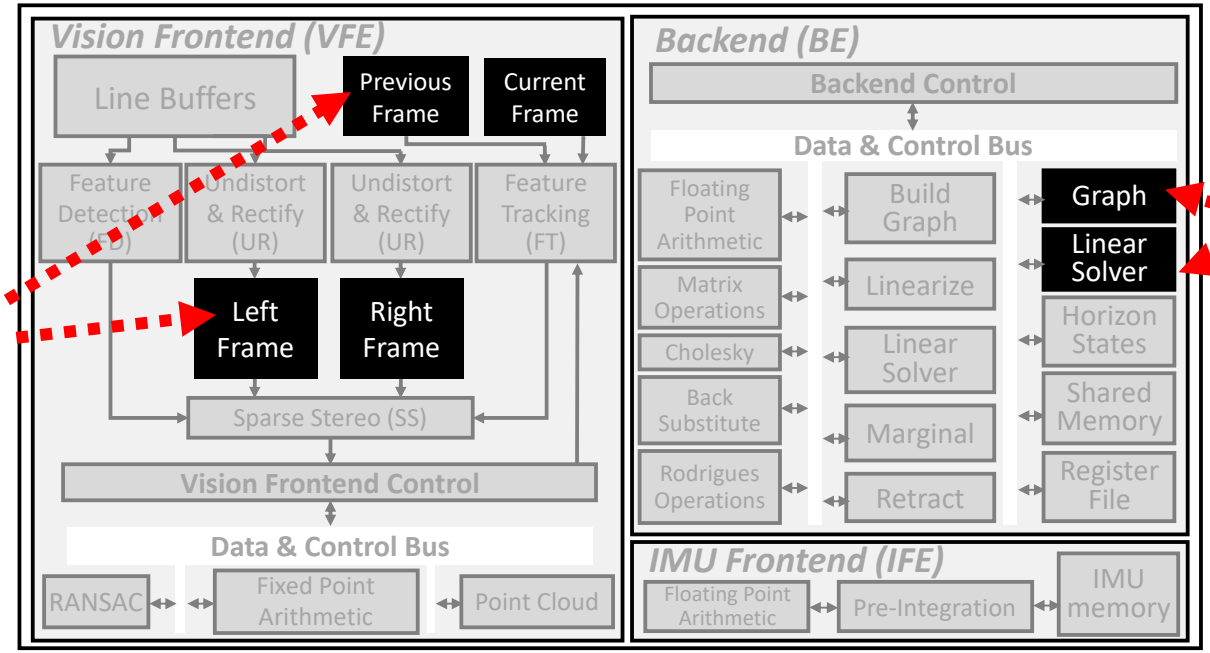
Navion is a <u>fully integrated</u> system:
No off-chip storage or processing

[Suleiman et al., VLSI 2018]

Key Methods to Reduce Data Size

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

Apply Low Cost Frame Compression

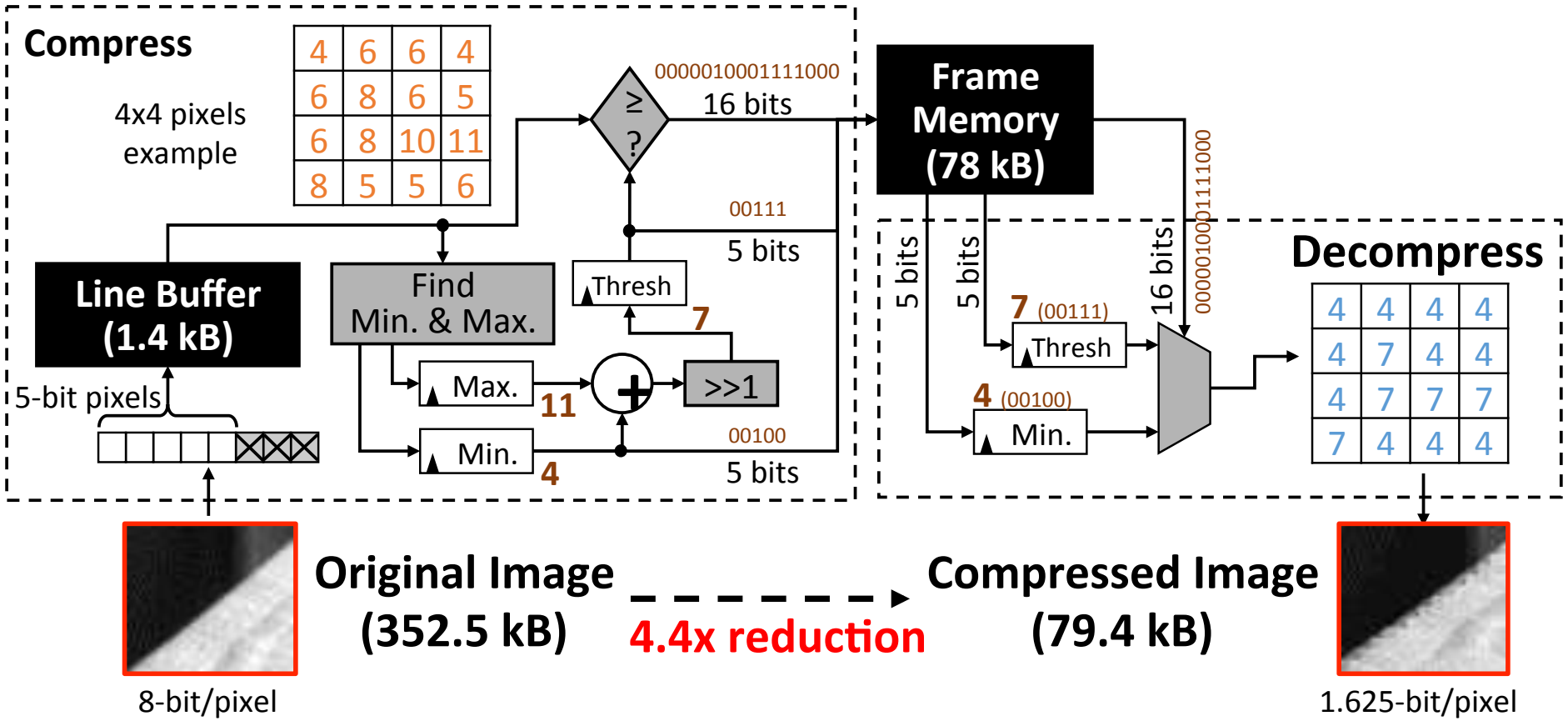


Exploit Sparsity in Graph and Linear Solver

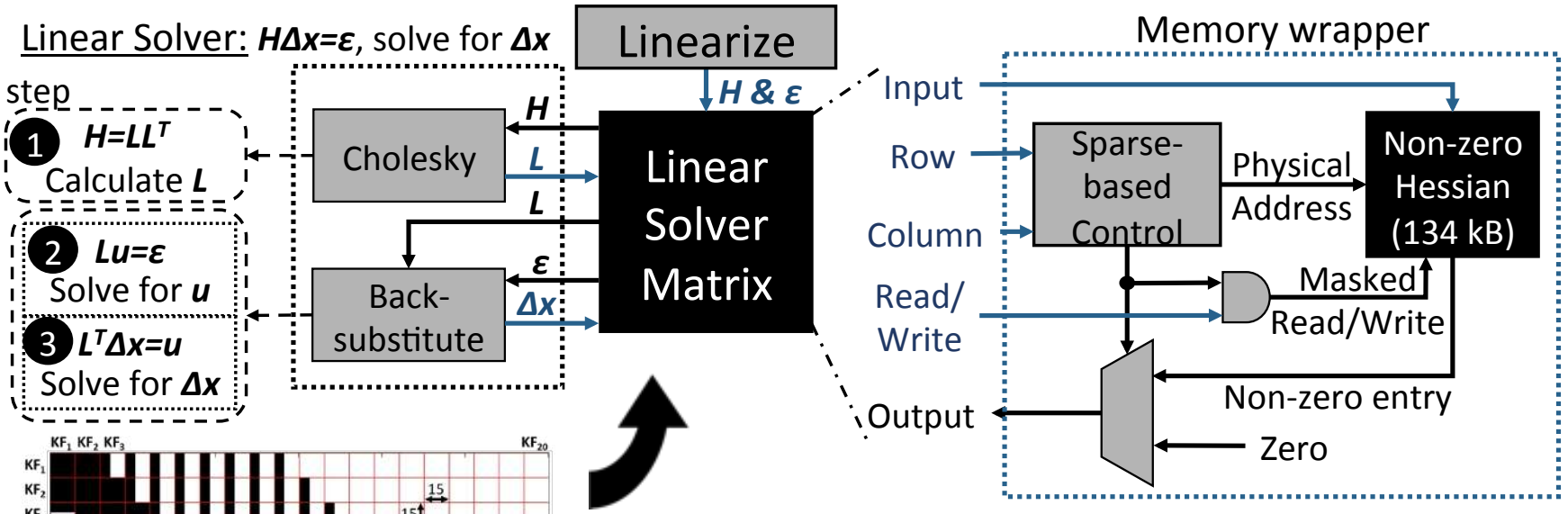
Use **compression** and **exploit sparsity** to reduce memory down to 854kB

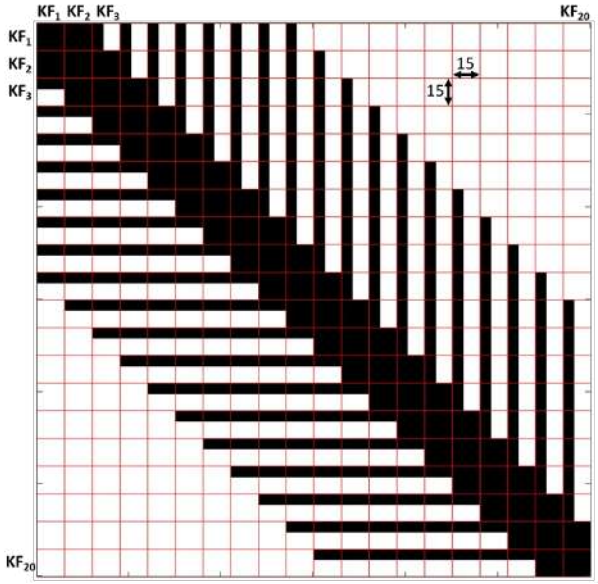
[Suleiman et al., VLSI 2018]

Frame Buffer Memory

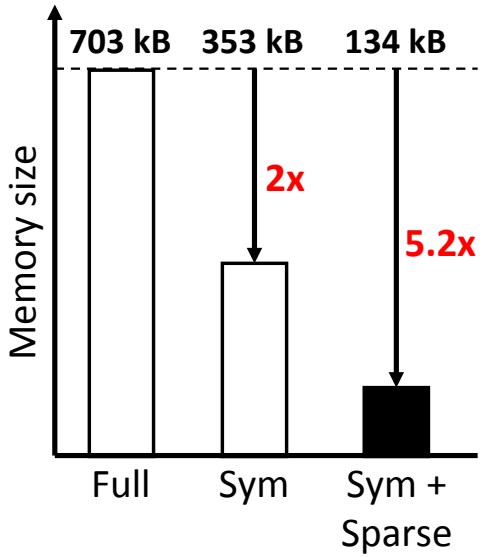


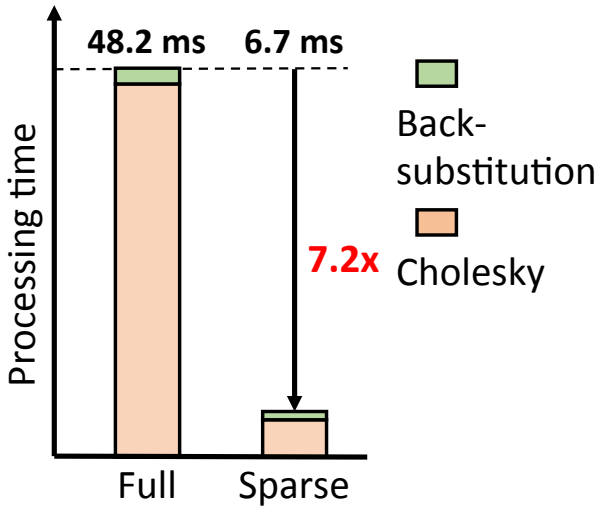
Linear Solver and Hessian Memory





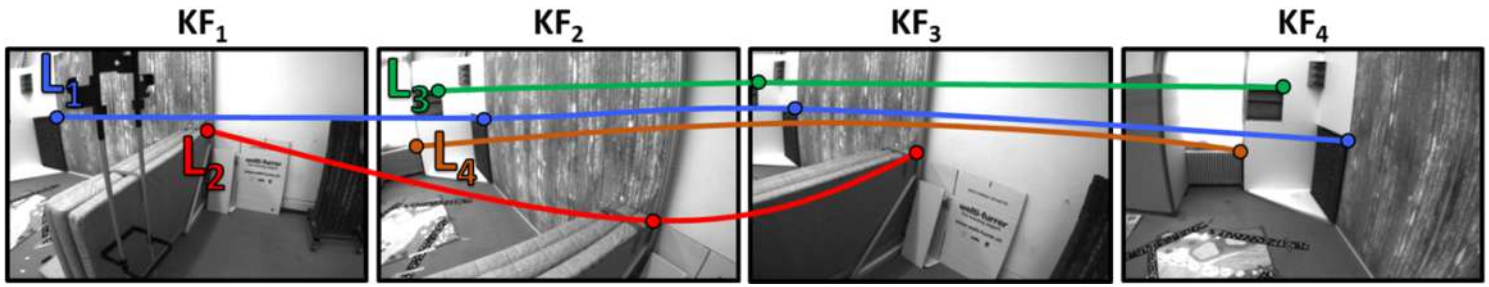
Sparsity pattern in both ***H*** & ***L***
(Non-zero: black)

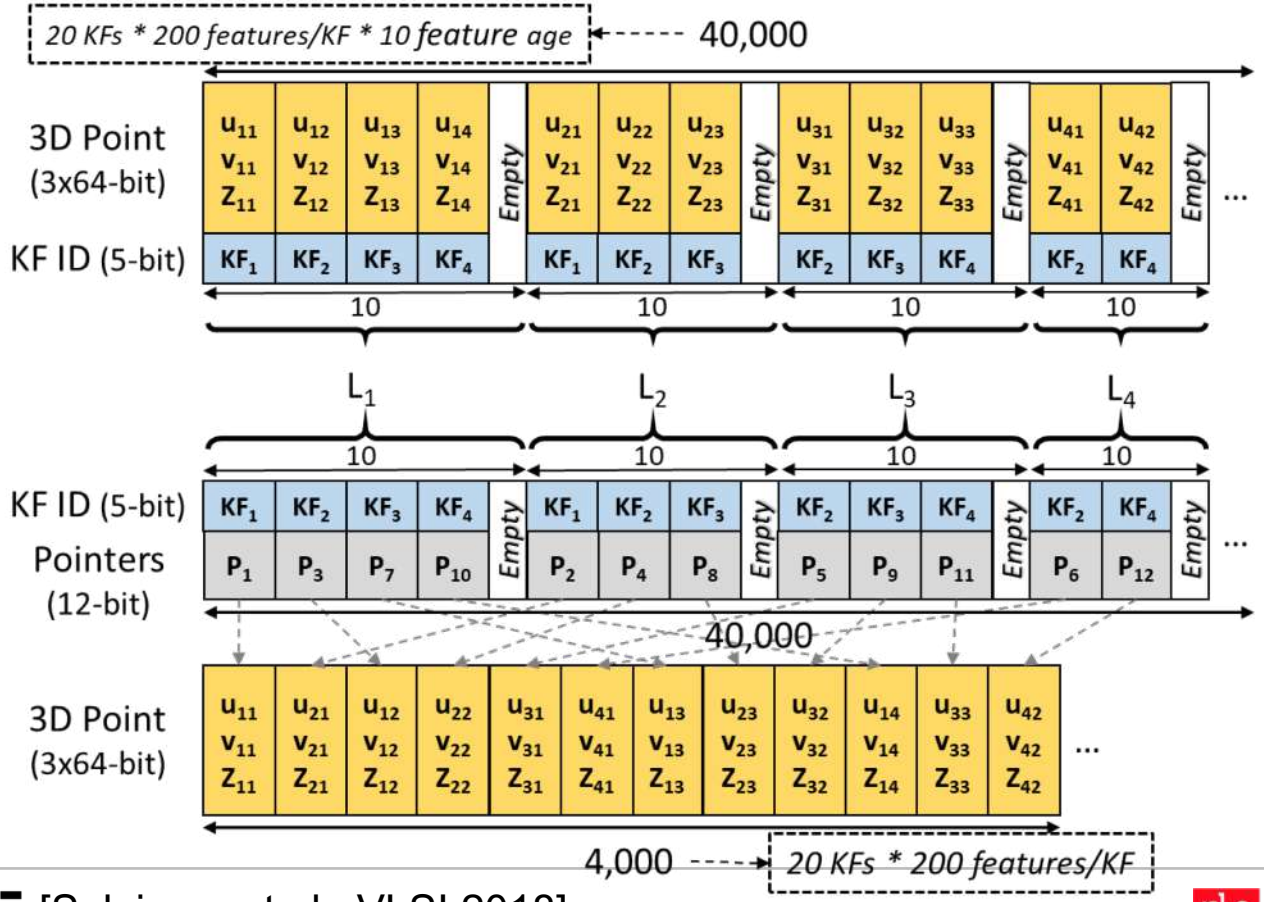




[Suleiman et al., VLSI 2018]

Factor Graph Memory





One Memory (962 kB)

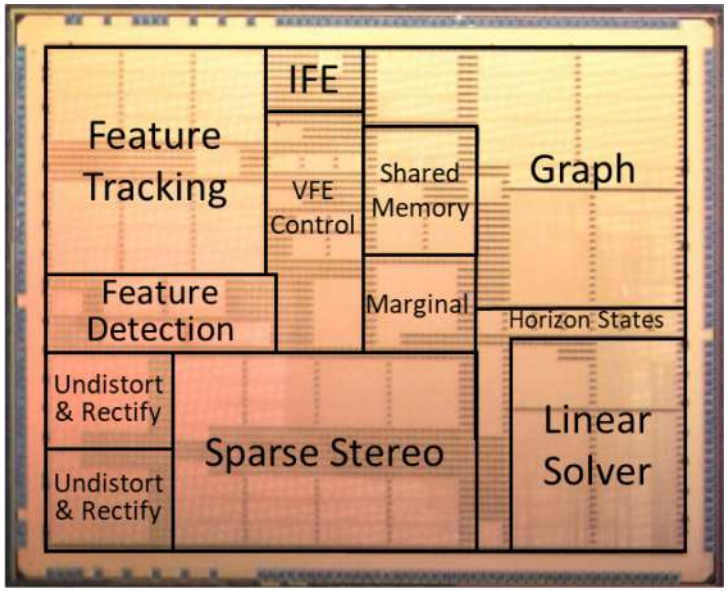
5.4x

Two-stage Memory (177 kB)

[Suleiman et al., VLSI 2018]

Navion Evaluation

5.0 mm



4.0 mm

65nm CMOS Test Chip

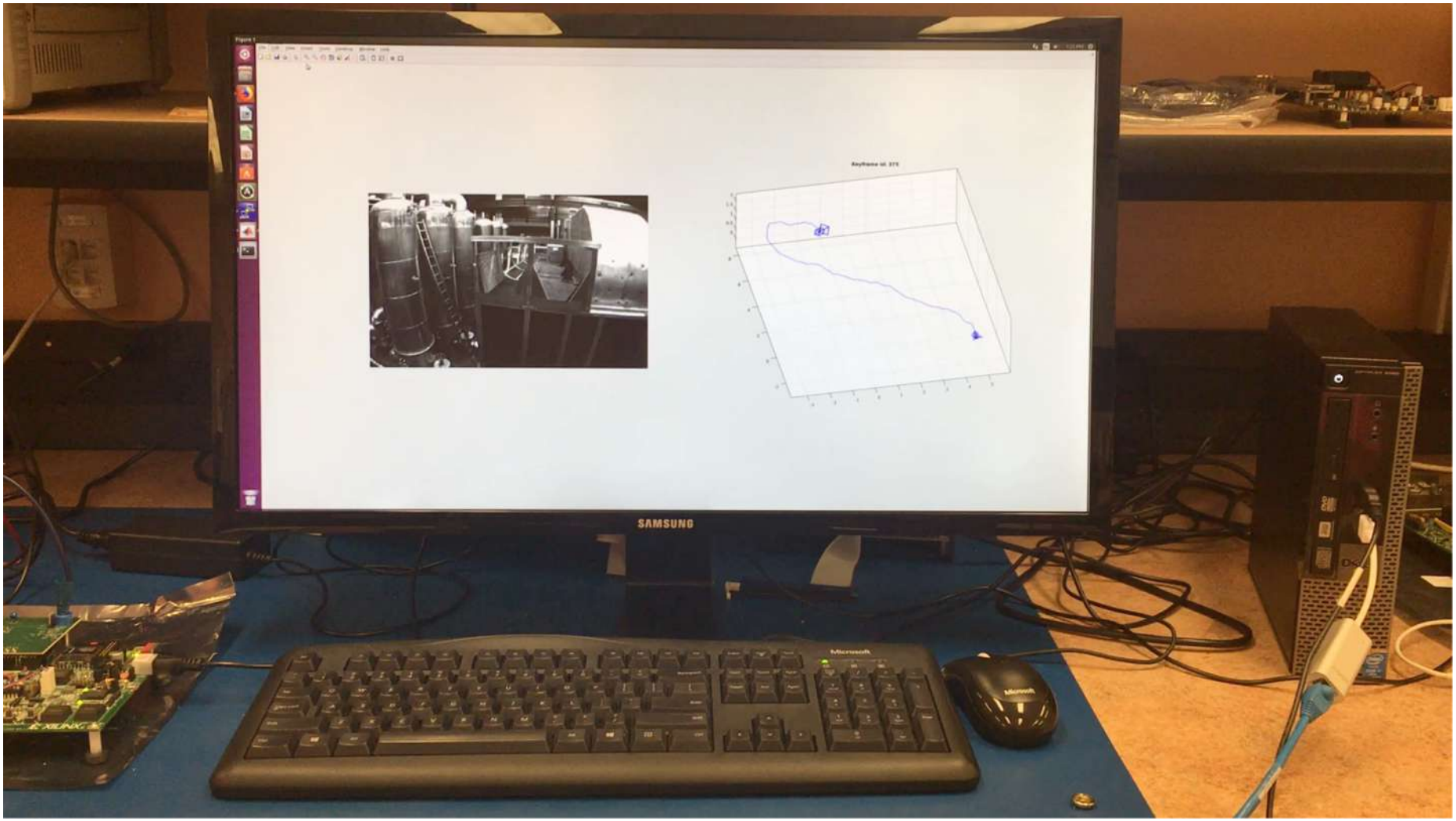
Over 250 configurable parameters *to adapt to different sensors and environments*

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

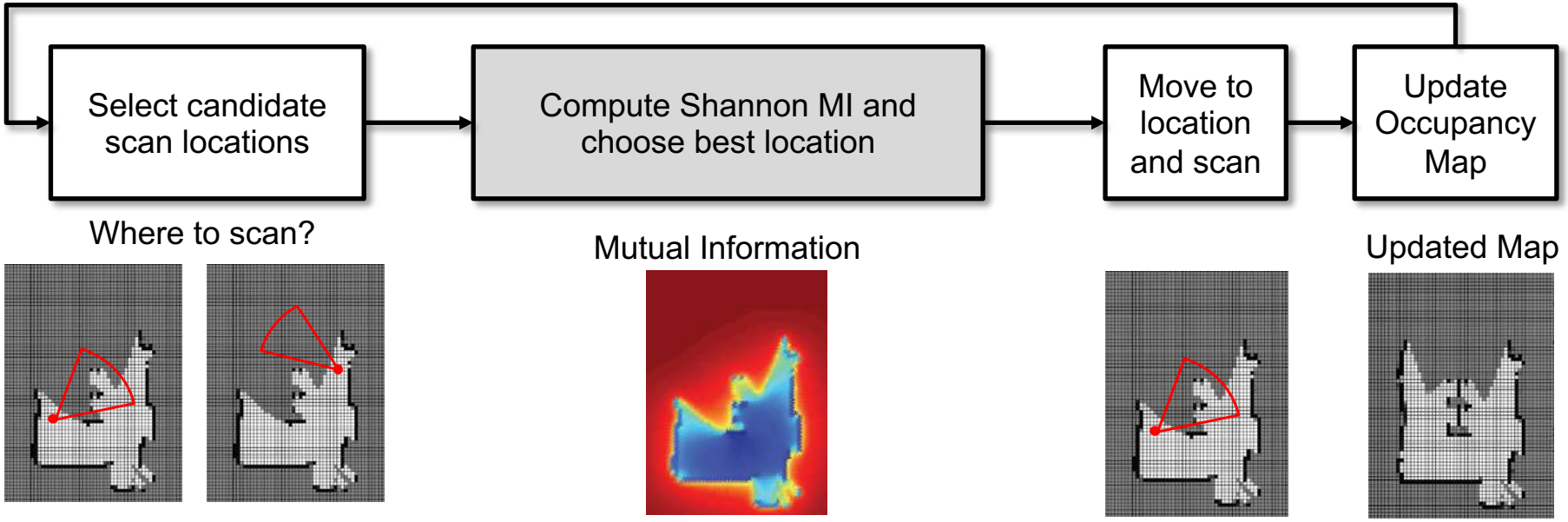
Navion System Demo



https://youtu.be/X5VZkPo_704

Where to Go Next: Planning and Mapping

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



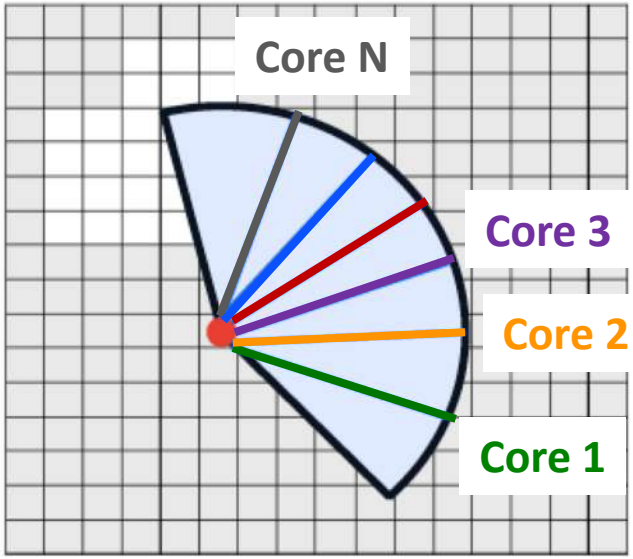
Exploration with a mini race car using motion capture for localization

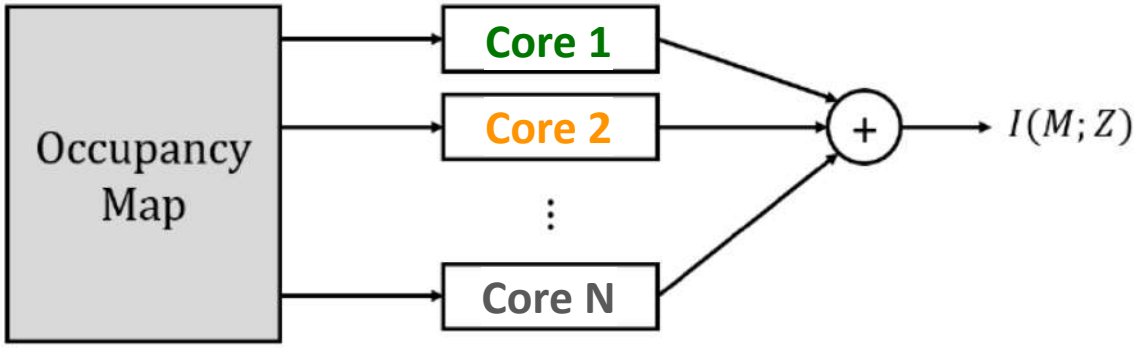
Occupancy map with planned path

MI surface

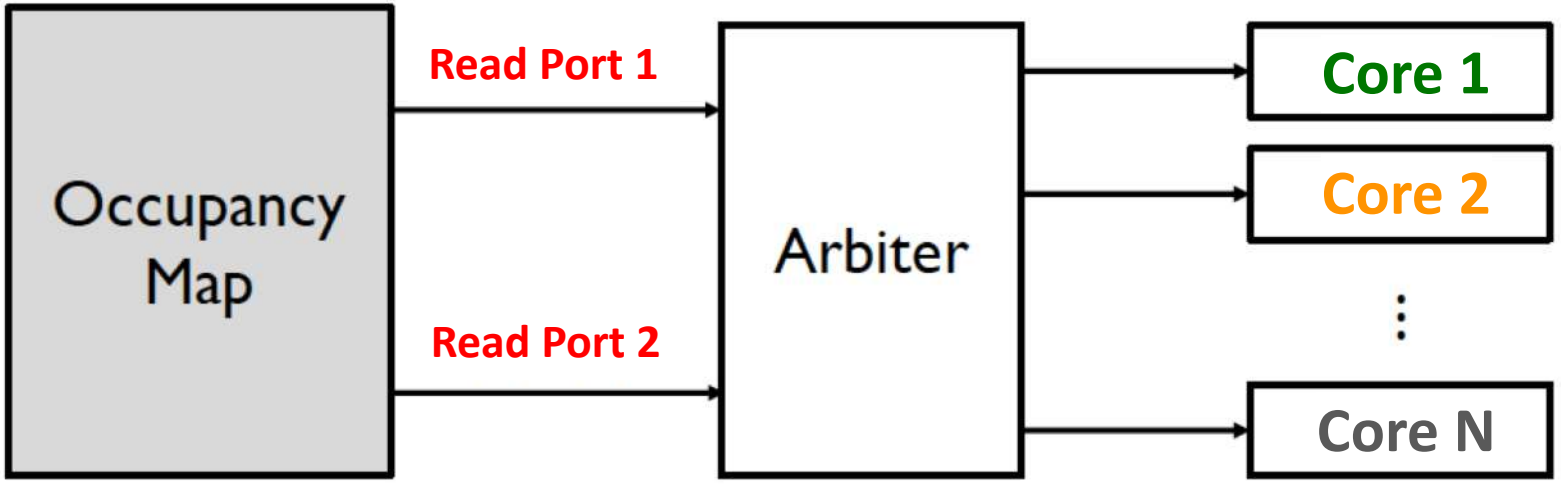
Challenge is Data Delivery to All Cores

Process multiple beams in parallel





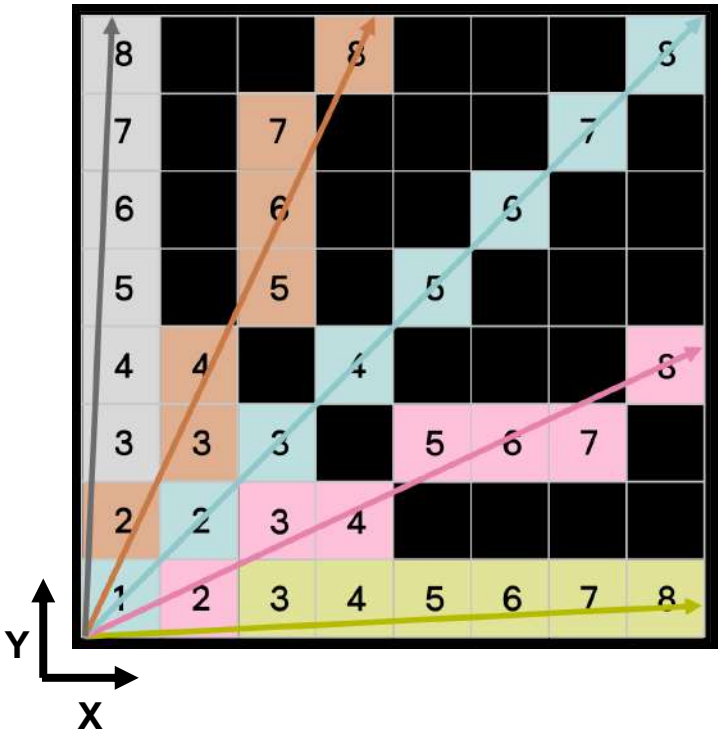
Data delivery from memory is limited



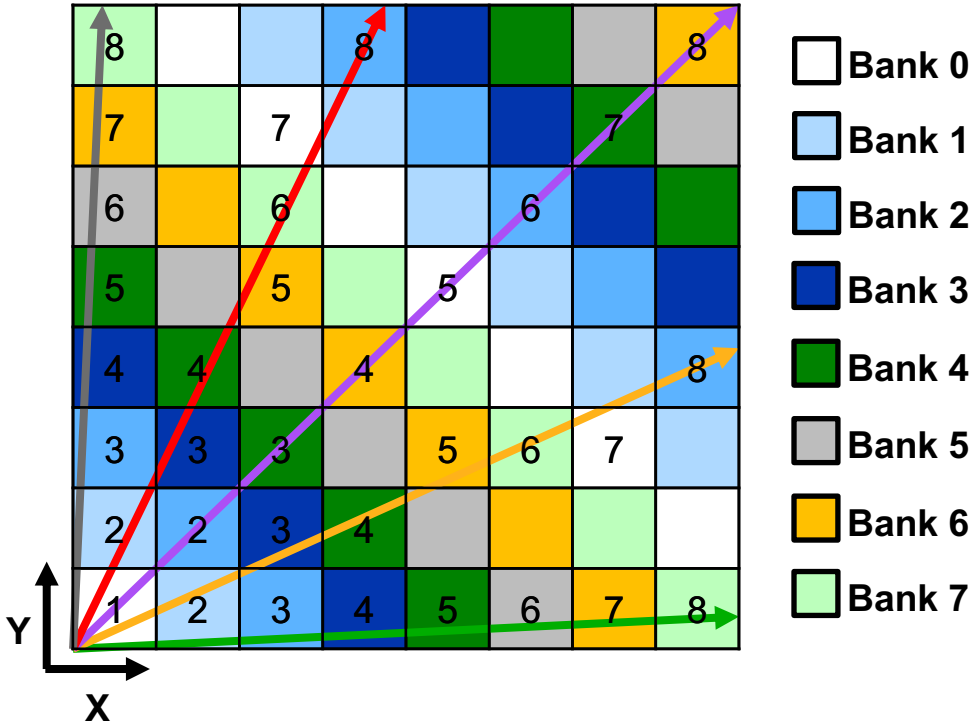
Specialized Memory Architecture

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



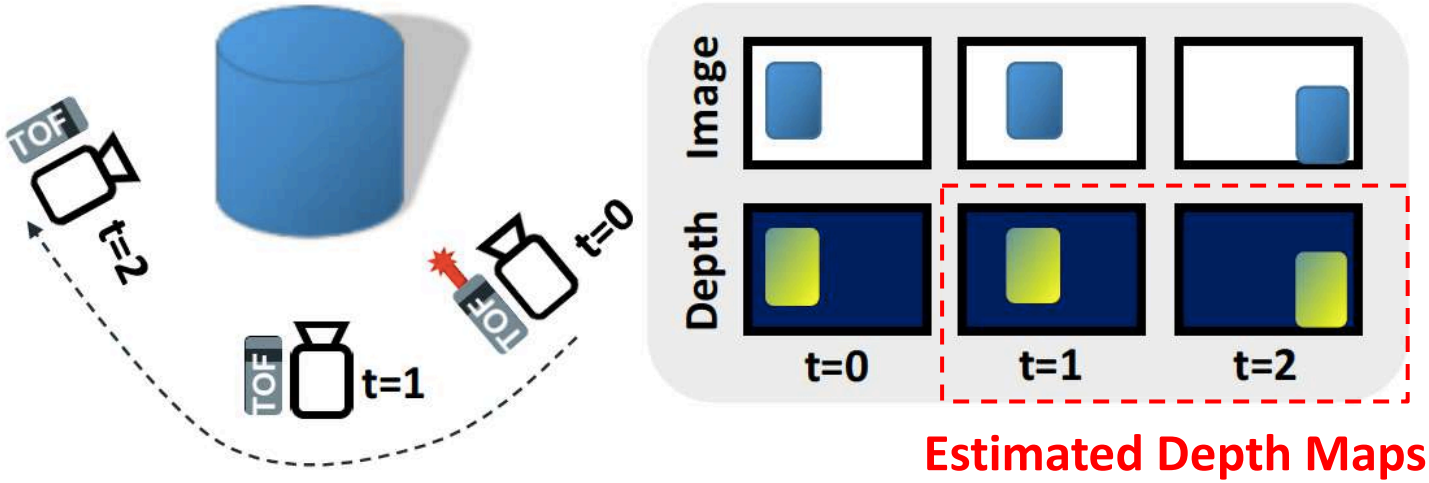
Compute the mutual information for an **entire map** of 20m x 20m at 0.1m resolution **in under a second** → a 100x speed up versus CPU for 1/10th of the power.

[Joint work with Sertac Karaman (AeroAstro)]

[Li et al., RSS 2019]

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**

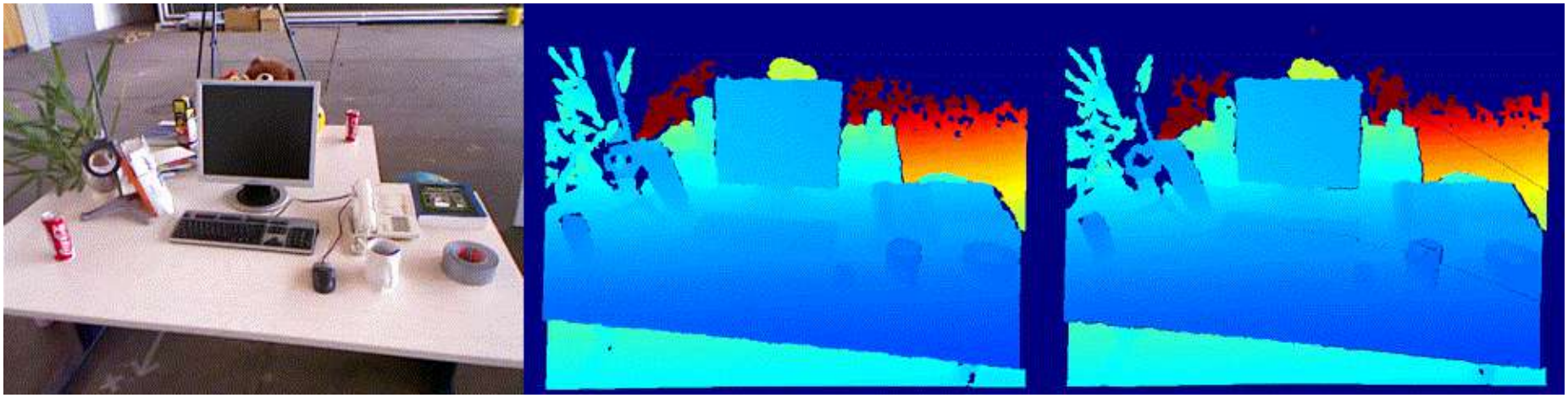


Estimated Depth Maps

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%

[Noraky et al., ICIP 2017]

Monitoring Neurodegenerative Disorders

Dementia affects 50 million people worldwide today (75 million in 10 years) [World Alzheimer's Report]

Mini-Mental State Examination (MMSE)

Q1. What is the year? Season? Date?

Q2. Where are you now? State? Floor?

Q3. Could you count backward from 100 by sevens? (93, 86, …)

Clo[illegible]est

Age and Ageing, 1998.

In collaboration with Thomas Heldt (IMES)

- Neuropsychological assessments are **time consuming** and **require a trained specialist**
- Repeat **medical assessments** are **sparse**, mostly **qualitative**, and suffer from **high retest variability**

Use Eye Movements for *Quantitative* Evaluation

Eye movements can be used to quantitatively evaluate severity, progression or regression of neurodegenerative diseases

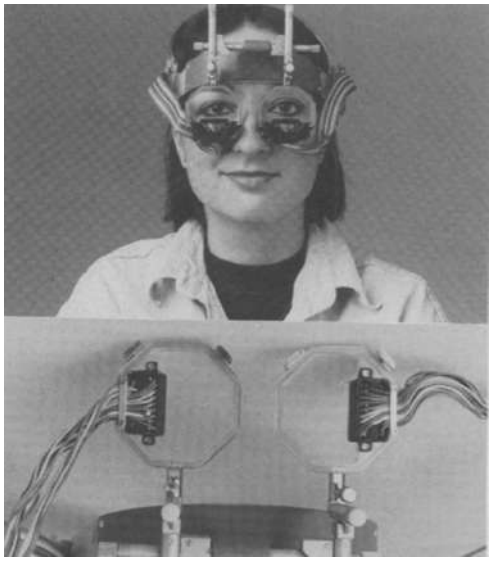
High-speed camera

Phantom v25-11

Substantial head support

SR EYELINK 1000 PLUS

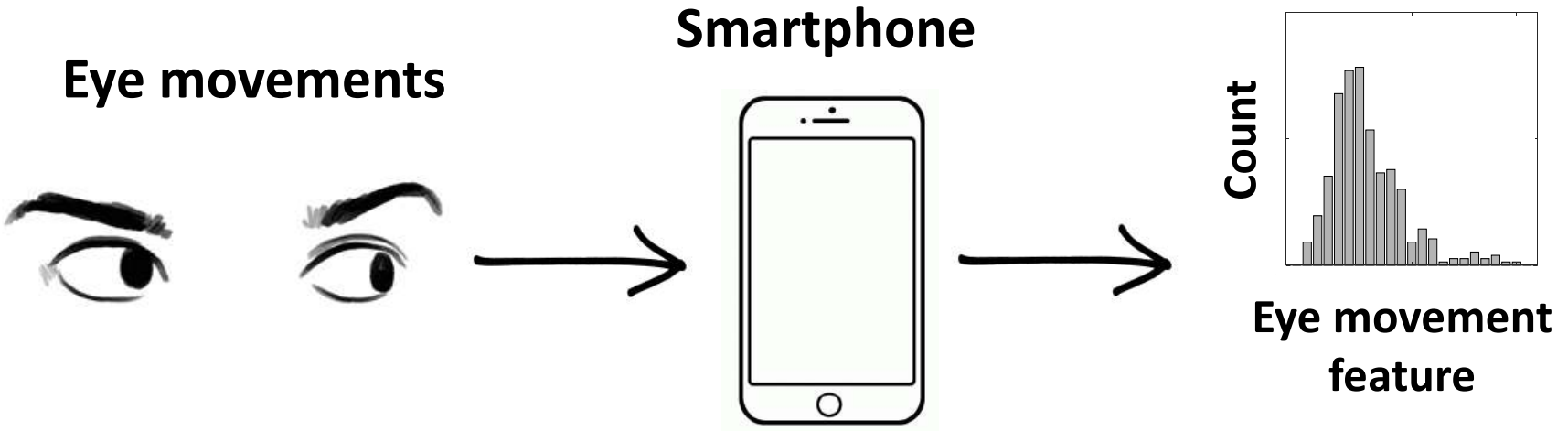
IR illumination



Reulen et al., *Med. & Biol. Eng. & Comp,* 1988.

Clinical measurements of saccade latency are done in constrained environments that rely on specialized, costly equipment.

Measure Eye Movements Using Phone

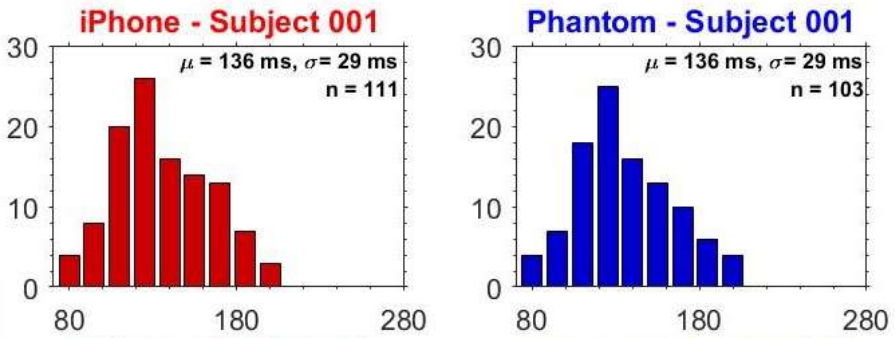


Develop algorithm to measure eye movement using a **consumer-grade camera** rather than high-cost research-grade camera.

Enable low-cost in-home longitudinal measurements.

iPhone 6 (< $1k)

Phantom ($100k)



Summary of Key Insights

- **Data movement dominates energy consumption**
  - Use dataflow that maximizes data reuse for ***all data types***
- **Design considerations for co-design of algorithm and hardware**
  - Incorporate ***direct metrics*** into algorithm design for improved efficiency
  - Diverse workloads requires a ***flexible dataflow and NoC*** to exploit data ***reuse in any dimension*** and increase core utilization for speed and scalability
- **Diverse compact representations to reduce data storage**
  - ***Adapt compression*** based on key properties (dense or sparse; structured or unstructured) to maximize compression efficiency and minimize overhead
- **Limited memory BW affects speed of highly parallel algorithms**
  - ***Balance banking and arbitration cost*** to minimize energy and maximize core utilization

Acknowledgements

Joel Emer

Sertac Karaman

Thomas Heldt

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

AFOSR AIR FORCE OFFICE OF SCIENTIFIC RESEARCH · NSF · ANALOG DEVICES · BROADCOM · Google · intel · IBM · DARPA · SRC · 3M · NVIDIA · QUALCOMM · SAMSUNG · Texas Instruments · tsmc

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*
- **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, 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.*
- **Monitoring Neurodegenerative Disorders Using a Phone**
  - *H.-Y. Lai, G. Saavedra Peña, C. Sodini, T. Heldt, V. Sze, "Enabling Saccade Latency Measurements with Consumer-Grade Cameras," IEEE International Conference on Image Processing (ICIP), October 2018.*
  - *G. Saavedra Peña, H.-Y. Lai, V. Sze, T. Heldt, "Determination of saccade latency distributions using video recordings from consumer-grade devices," IEEE International Engineering in Medicine and Biology Conference (EMBC), 2018.*
  - *H.-Y. Lai, G. Saavedra Peña, C. Sodini, V. Sze, T. Heldt, "Measuring Saccade Latency Using Smartphone Cameras," to appear in IEEE Journal of Biomedical and Health Informatics (JBHI)*