Efficient Computing For Low-Energy Robotics

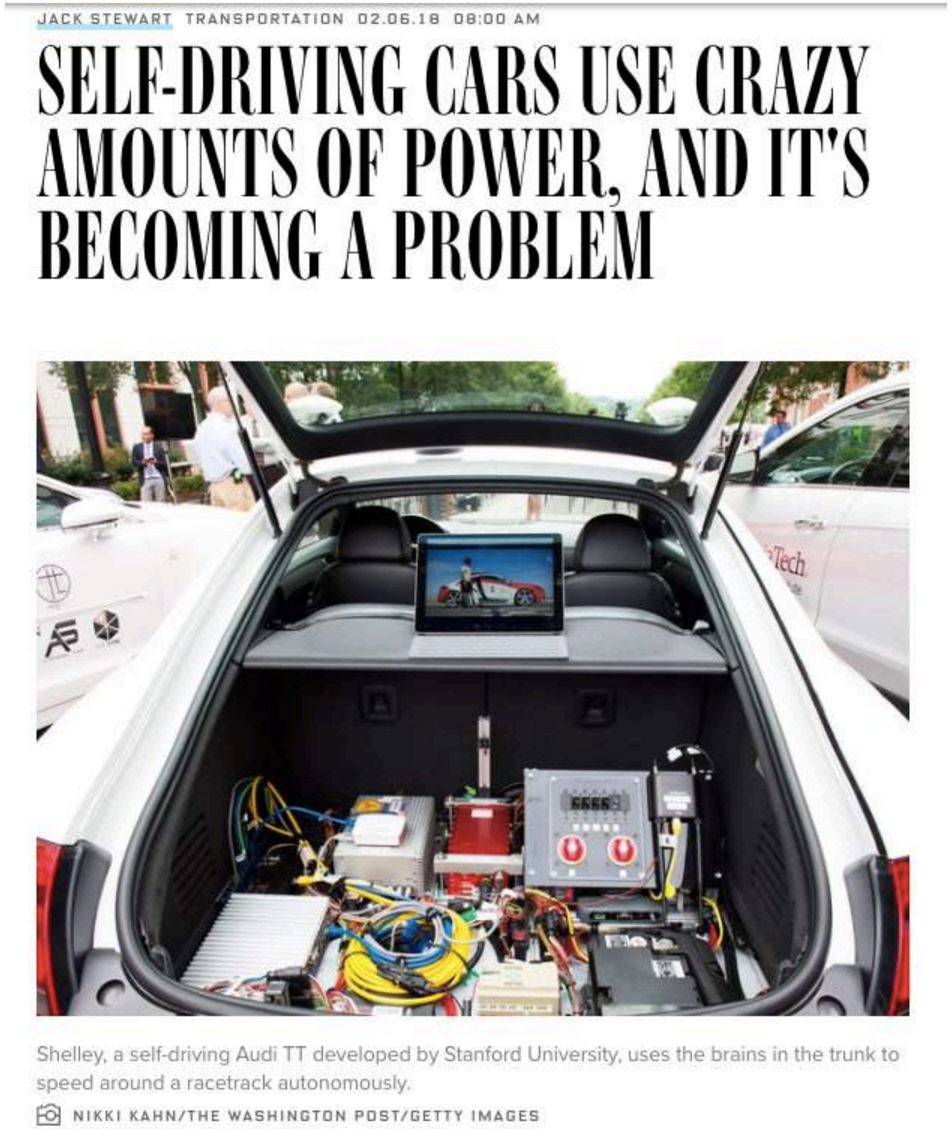
Vivienne Sze (@eems_mit)

Massachusetts Institute of Technology

In collaboration with Luca Carlone, Yu-Hsin Chen, Joel Emer, Sertac Karaman, Tushar Krishna, Thomas Heldt, Theia Henderson, Peter Li, Fangchang Ma, James Noraky, Soumya Sudhakar, Amr Suleiman, Diana Wofk, Nellie Wu, Tien-Ju Yang, Zhengdong Zhang

Slides available at
https://tinyurl.com/SzeMITDL2020

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

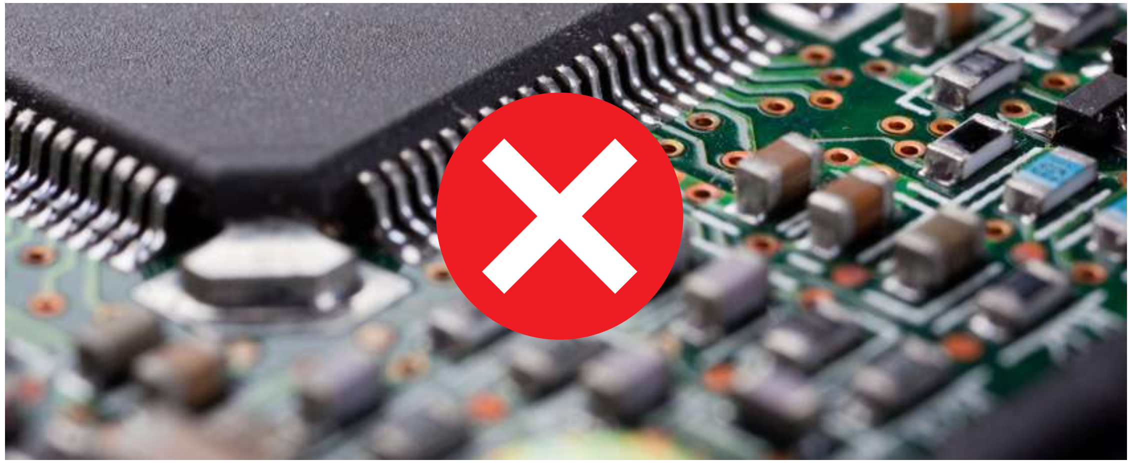
(Feb 2018)

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!

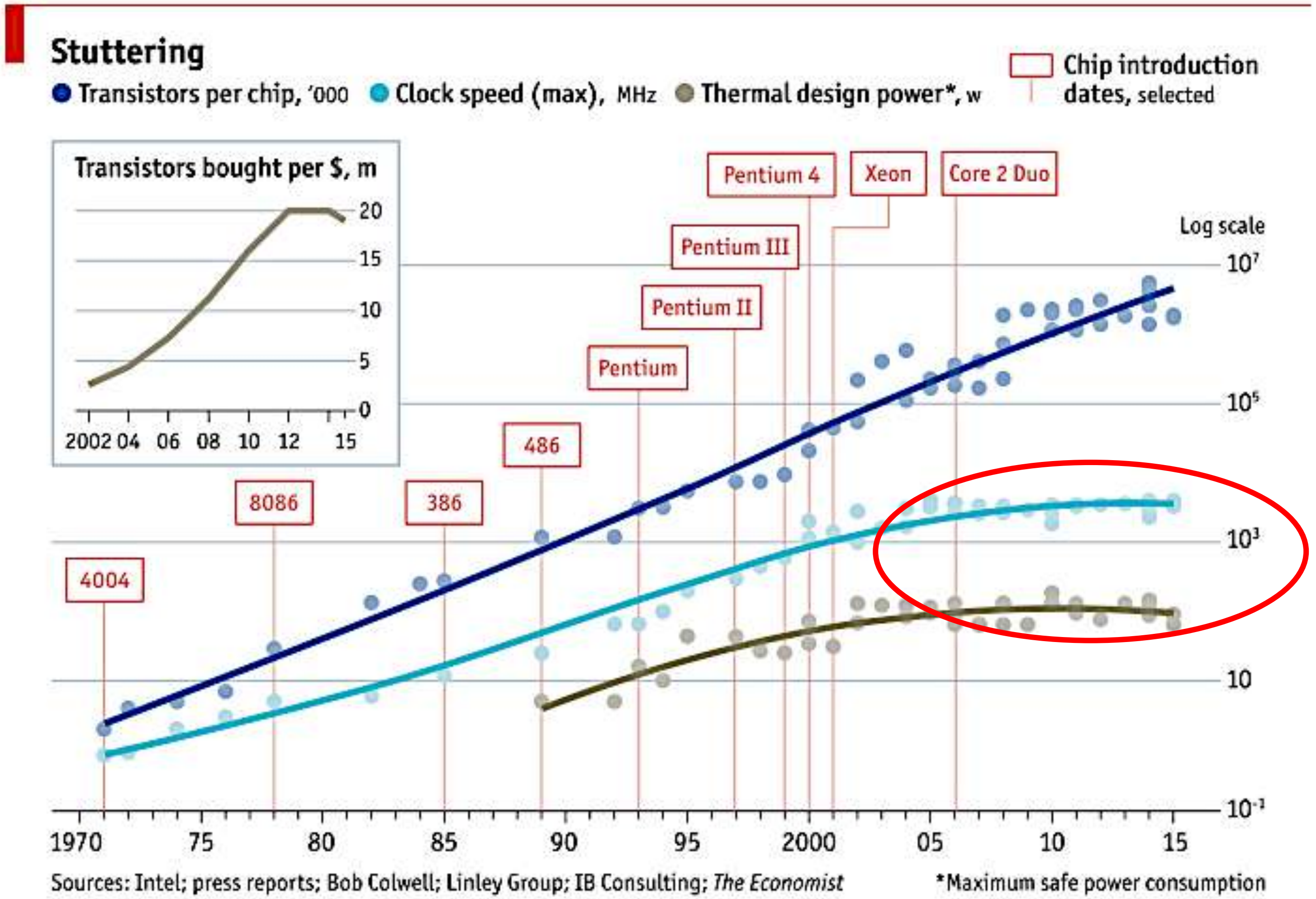
Existing Processors Consume Too Much Power



< 1 Watt

> 10 Watts

Transistors Are Not Getting More Efficient



Stuttering

Sources: Intel; press reports; Bob Colwell; Linley Group; IB Consulting; *The Economist*

*Maximum safe power consumption

Slowdown

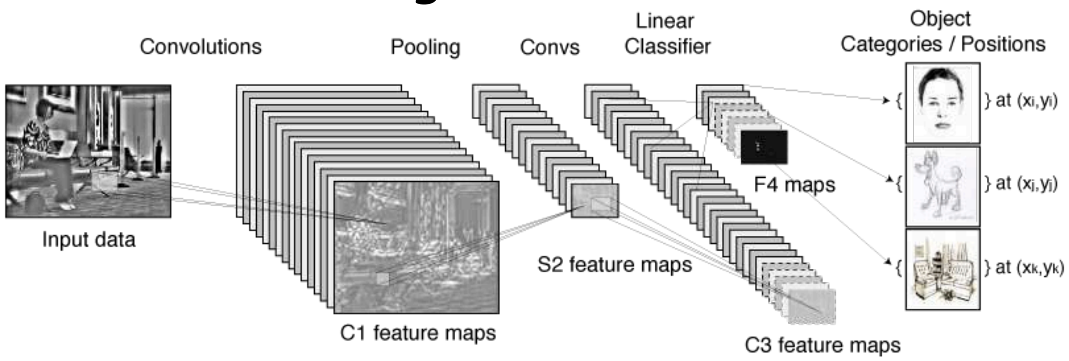
Slowdown of Moore's Law and Dennard Scaling

General purpose microprocessors are not getting faster or more efficient

Need **specialized / domain-specific hardware** for significant improvements in speed and energy efficiency

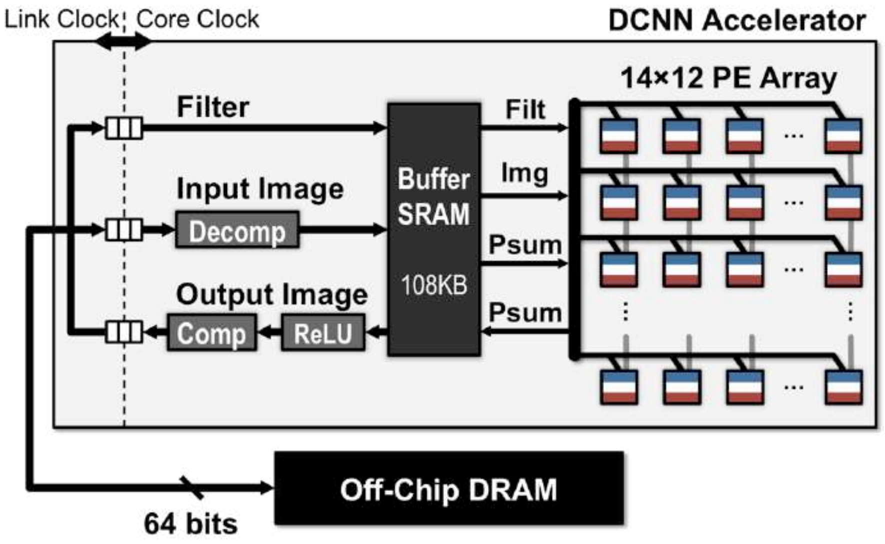
Efficient Computing with Cross-Layer Design

Algorithms

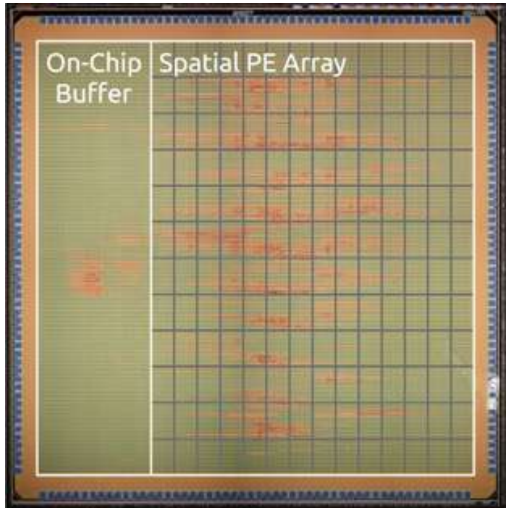


Systems

Architectures

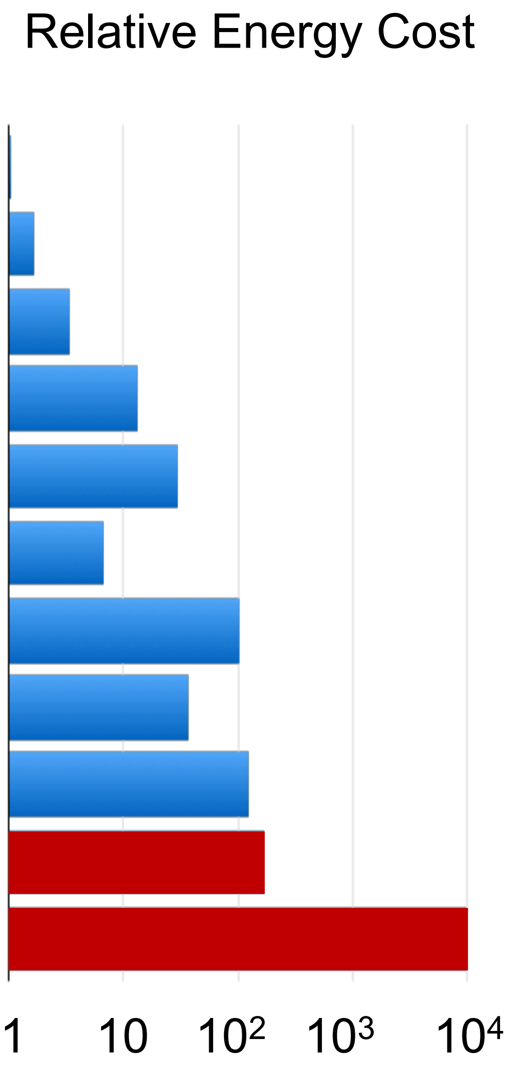


Circuits



Energy Dominated by Data Movement

Operation:	Energy (pJ)
8b Add	0.03
16b Add	0.05
32b Add	0.1
16b FP Add	0.4
32b FP Add	0.9
8b Multiply	0.2
32b Multiply	3.1
16b FP Multiply	1.1
32b FP Multiply	3.7
32b SRAM Read (8KB)	5
32b DRAM Read	640



Relative Energy Cost

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

[**Horowitz**, *ISSCC* 2014]

Autonomous Navigation Uses a Lot of Data

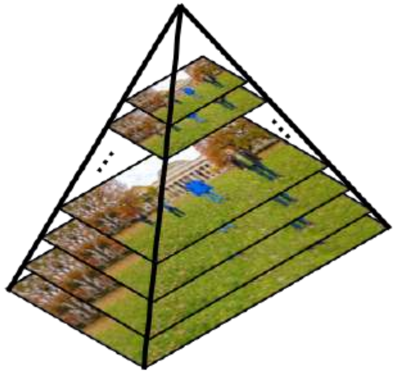
Semantic Understanding

- High frame rate
- Large resolutions
- Data expansion

Geometric Understanding

- Growing map size

2 million pixels

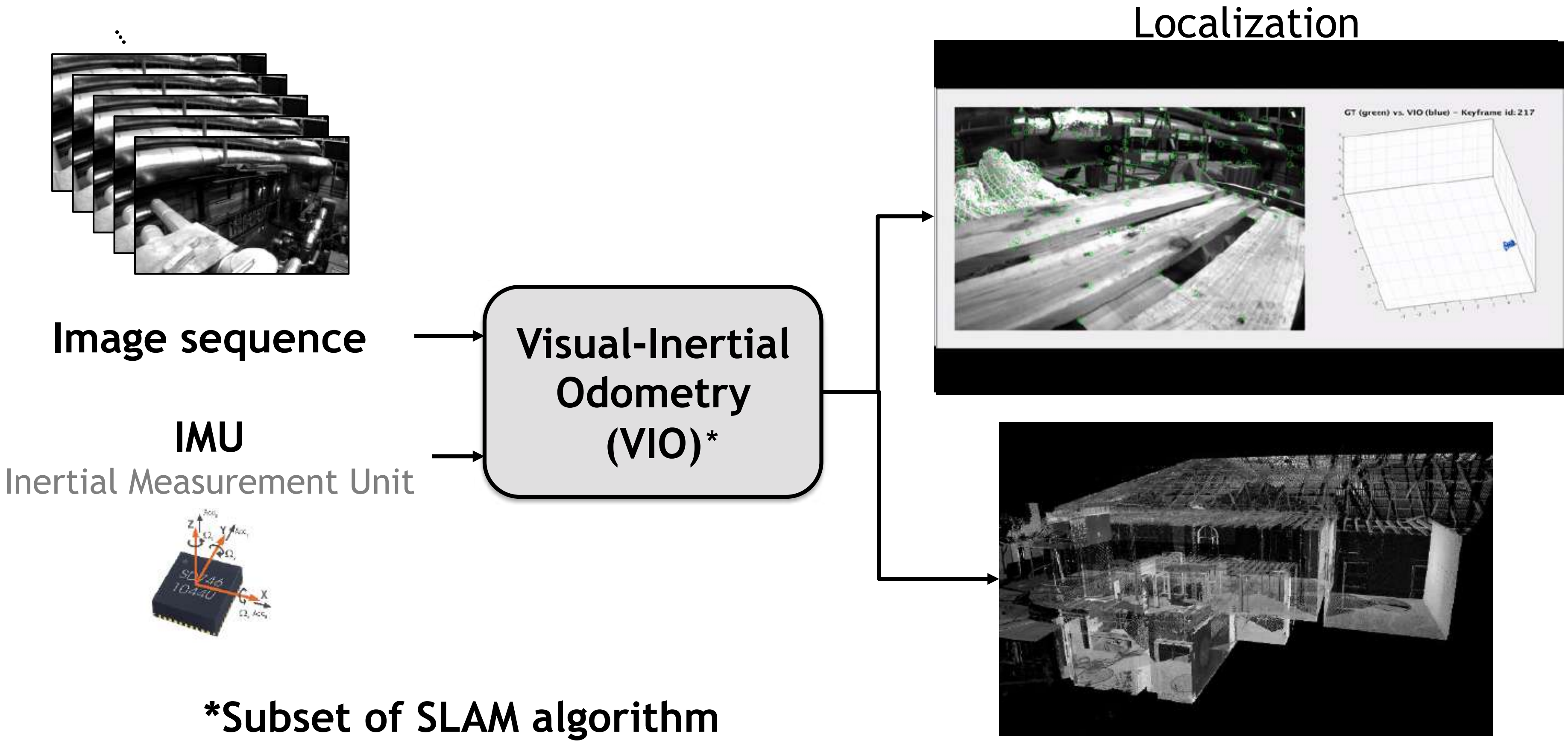


10x-100x more pixels

[**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)


Localization
Image sequence
IMU
Inertial Measurement Unit
Visual-Inertial Odometry (VIO)*
GT (green) vs. VIO (blue) – Keyframe id: 217
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

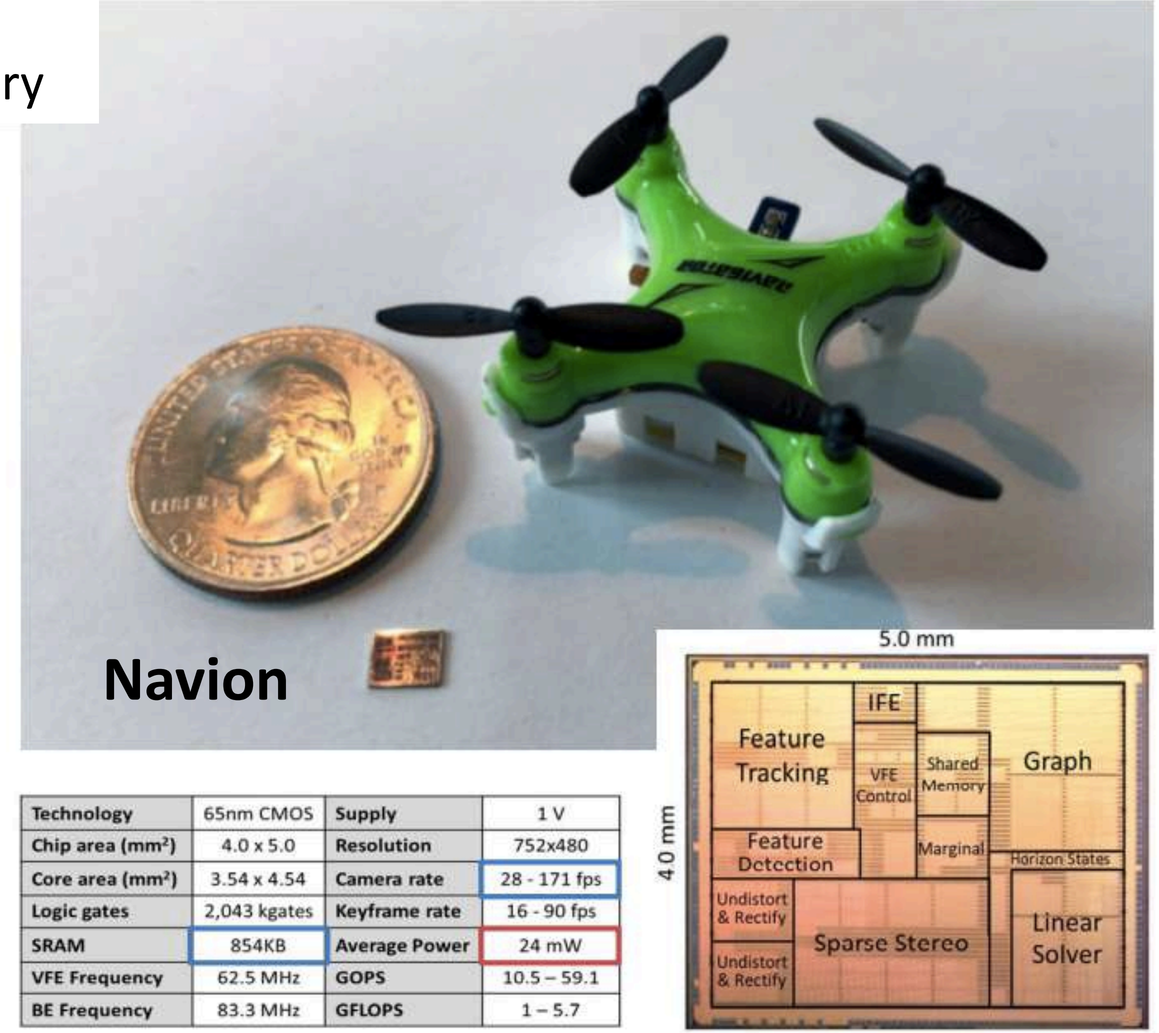
(Feature detection, tracking, and outlier elimination)

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

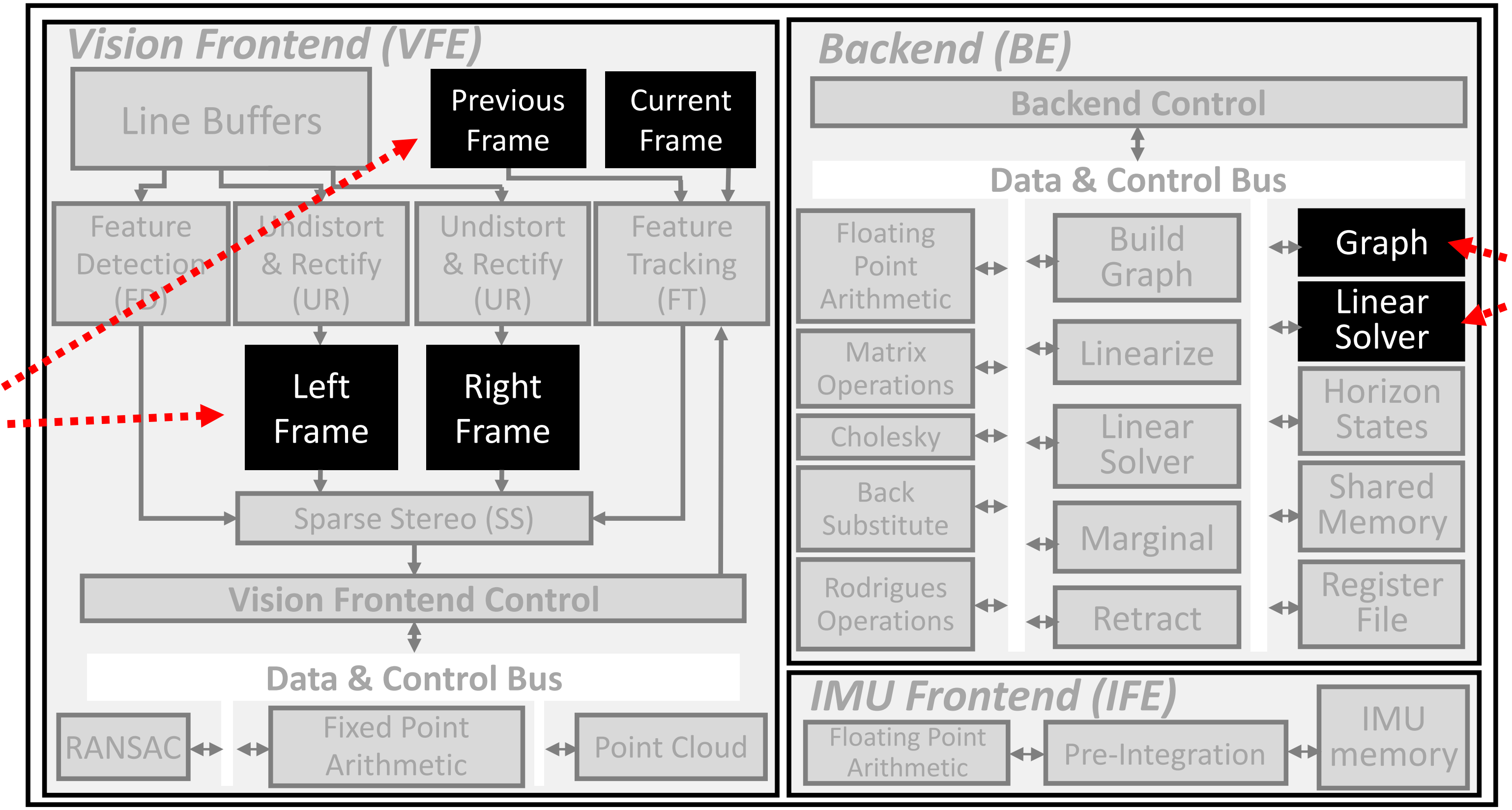
[Zhang et al., RSS 2017], [Suleiman et al., VLSI 2018]

[Joint work with Sertac Karaman (AeroAstro)]

Key Methods to Reduce Data Size

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

Apply Low Cost Frame Compression



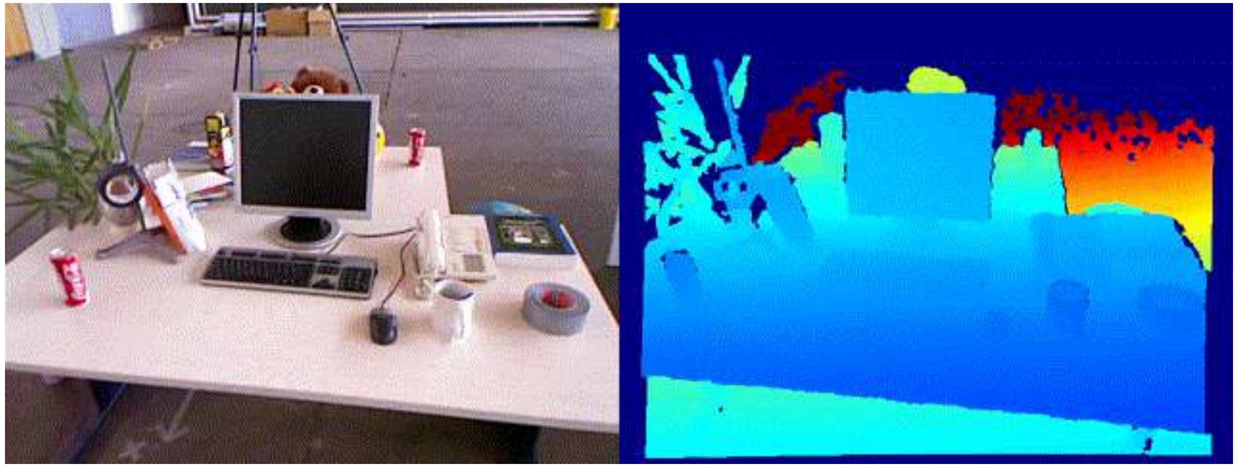
Exploit Sparsity in Graph and Linear Solver

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

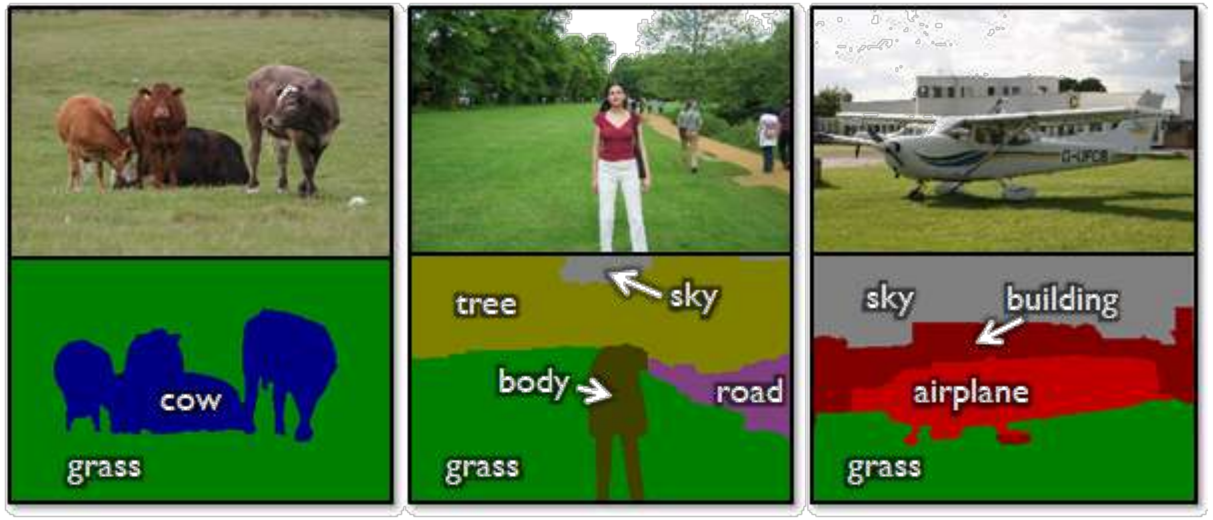
[**Suleiman**, *VLSI-C* 2018] **Best Student Paper Award**

Understanding the Environment

Depth Estimation

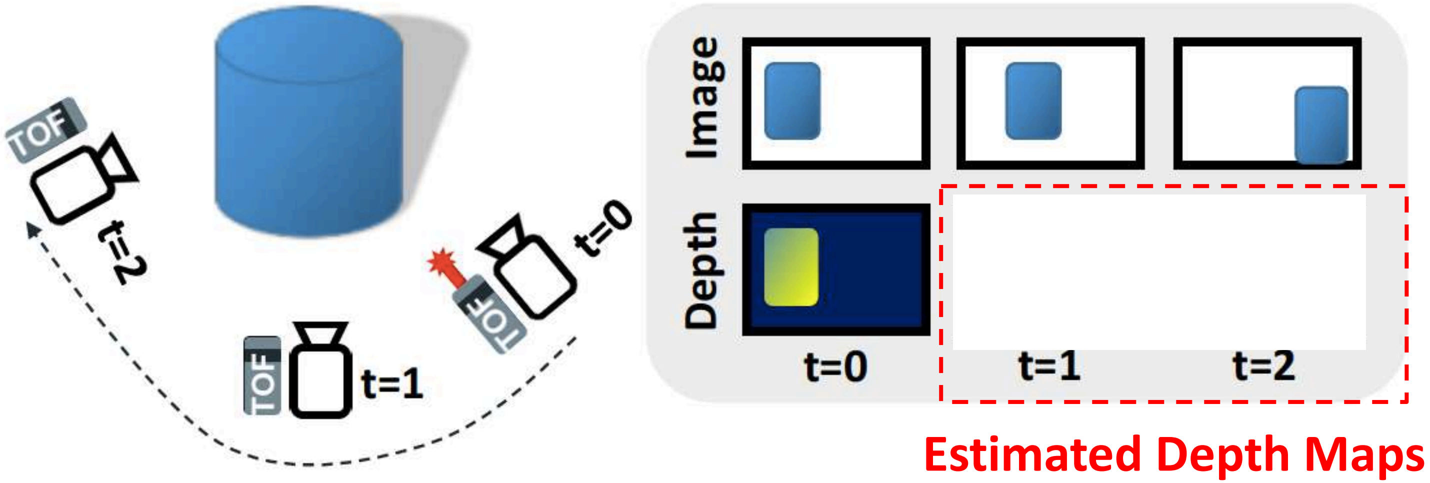


Semantic Segmentation



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**



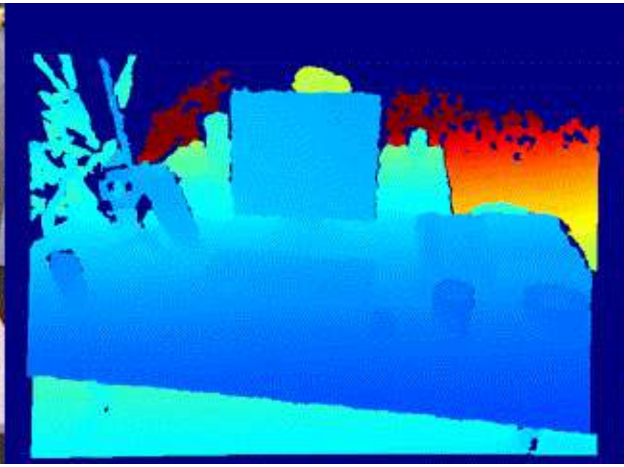
Estimated Depth Maps

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

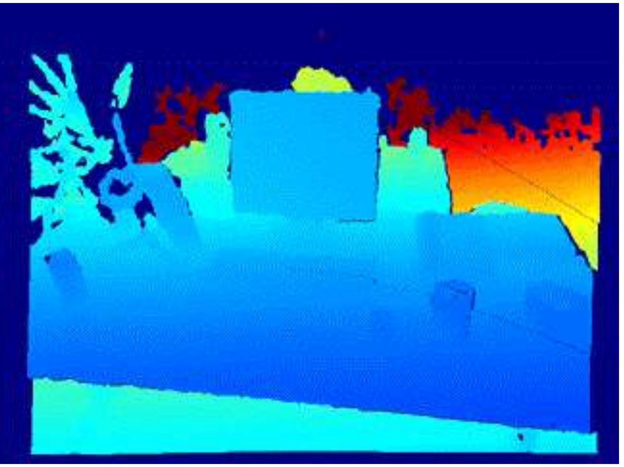
[**Noraky**, *ICIP* 2017]

Results of Low Power Depth ToF Imaging

RGB Image



Depth Map
Ground Truth



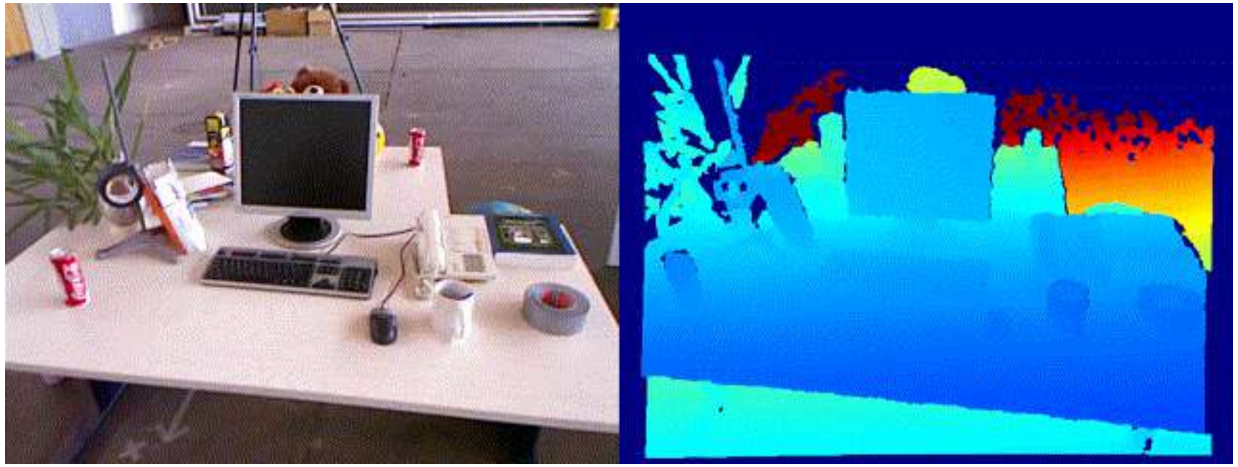
Depth Map
Estimated

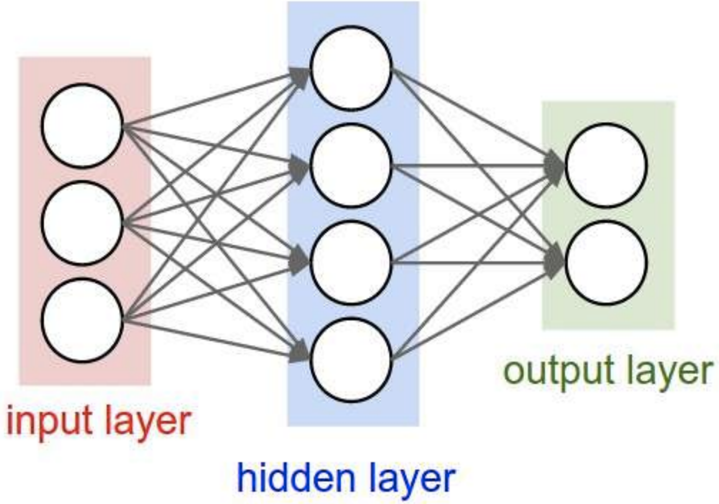
Mean Relative Error: 0.7%
Duty Cycle (on-time of laser): 11%

[**Noraky**, *ICIP* 2017]

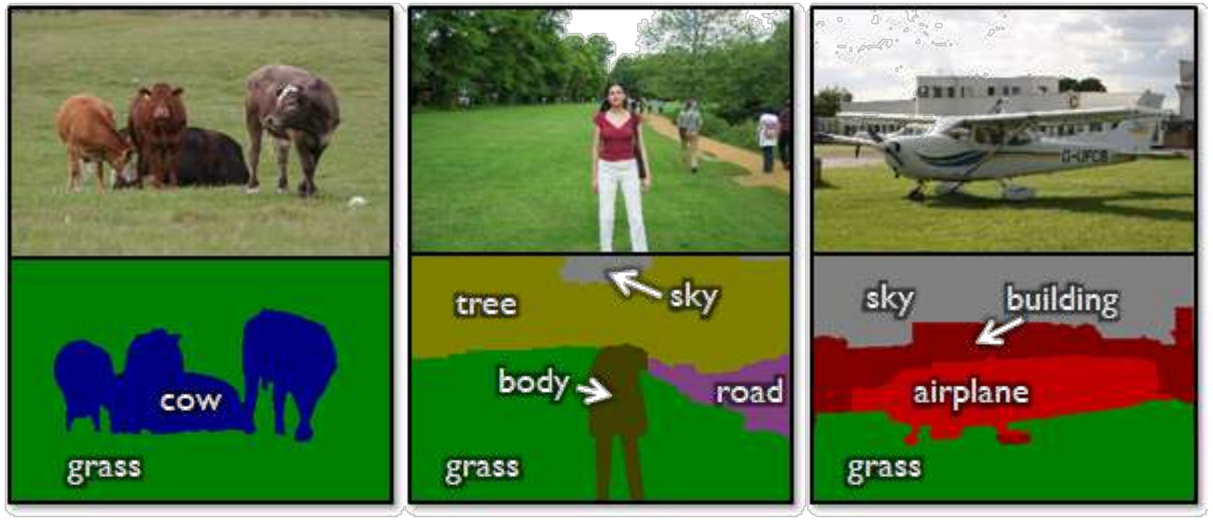
Understanding the Environment

Depth Estimation





Semantic Segmentation

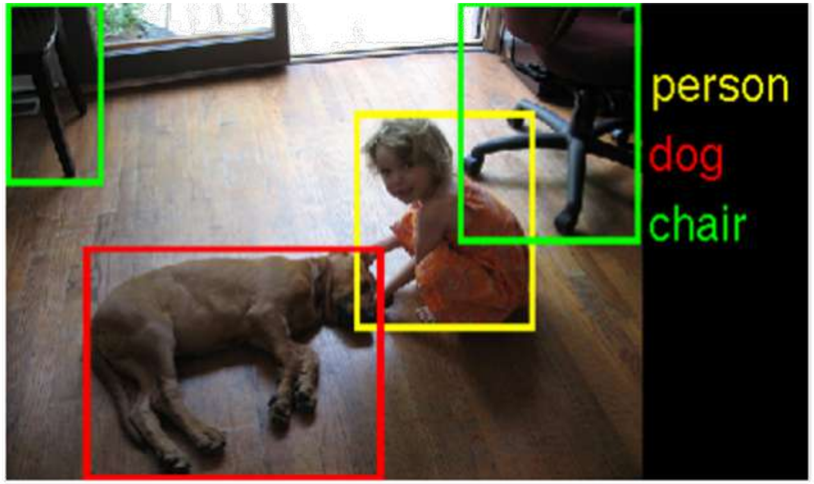


State-of-the-art approaches use **Deep Neural Networks,** which require **up to several hundred millions of operations and weights to compute!**
>100x more complex than video compression

Deep Neural Networks

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

Computer Vision

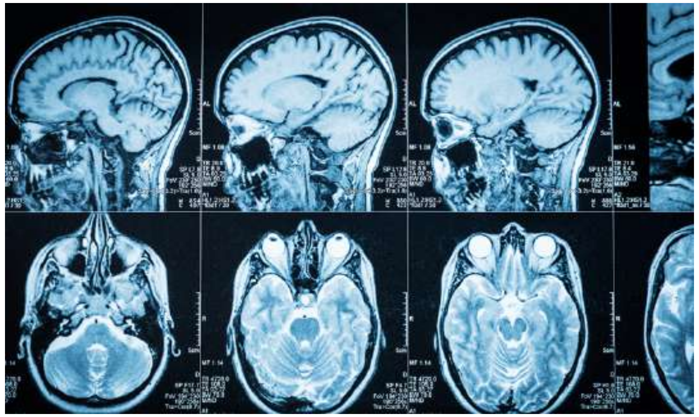


Speech Recognition

What can I help you with?

Game Play

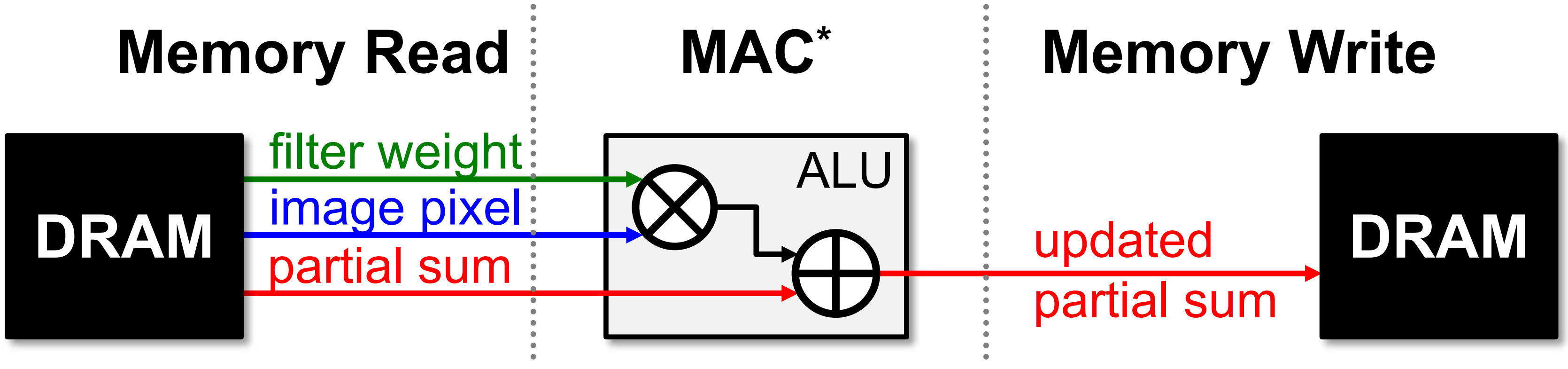
Medical



Properties We Can Leverage

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

- Memory Access is the Bottleneck



* multiply-and-accumulate

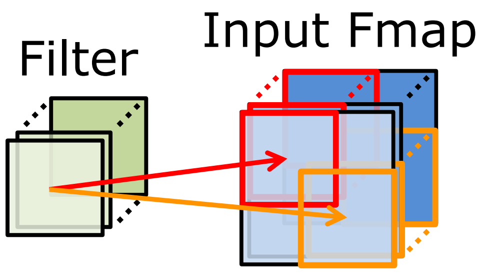
Worst Case: all memory R/W are **DRAM** accesses

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

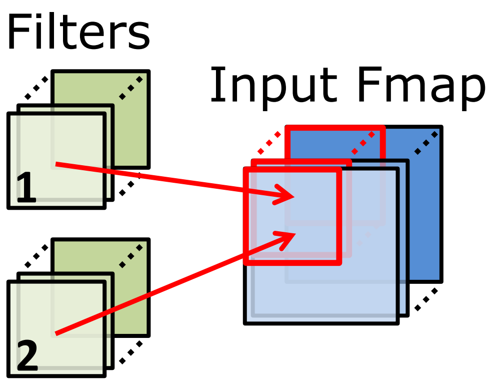
Properties We Can Leverage

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

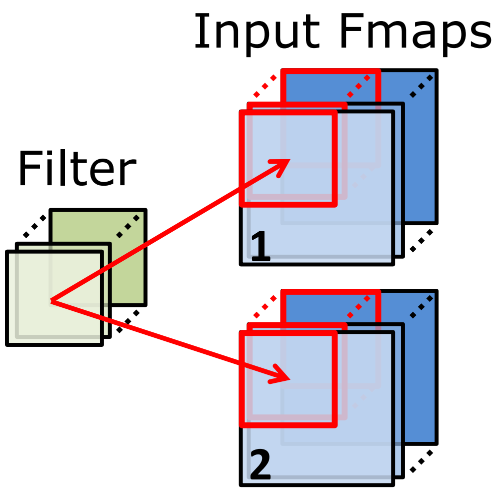
- **Input data reuse** opportunities (**up to 500x**)



Convolutional Reuse
(Activations, Weights)
CONV layers only
(sliding window)

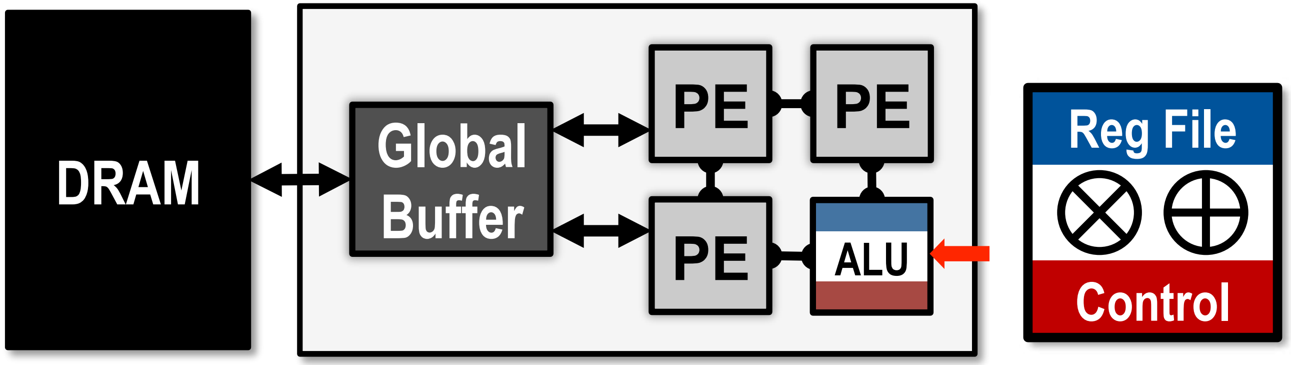


Fmap Reuse
(Activations)
CONV and FC layers



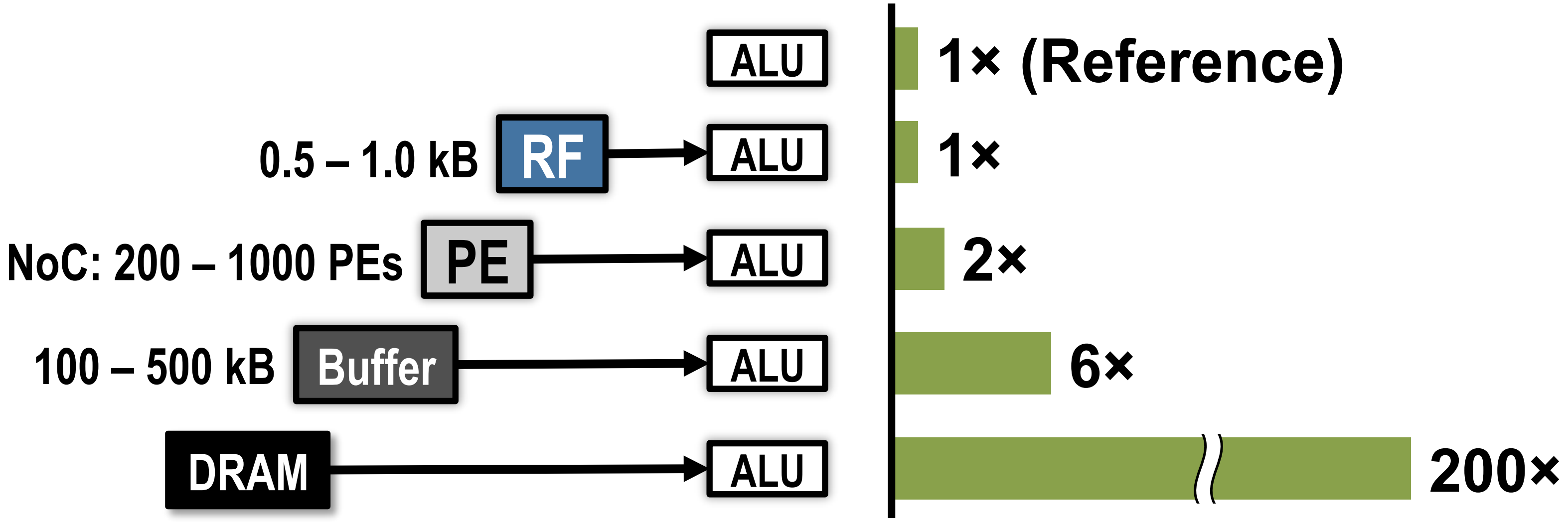
Filter Reuse
(Weights)
CONV and FC layers
(batch size > 1)

Exploit Data Reuse at Low-Cost Memories



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

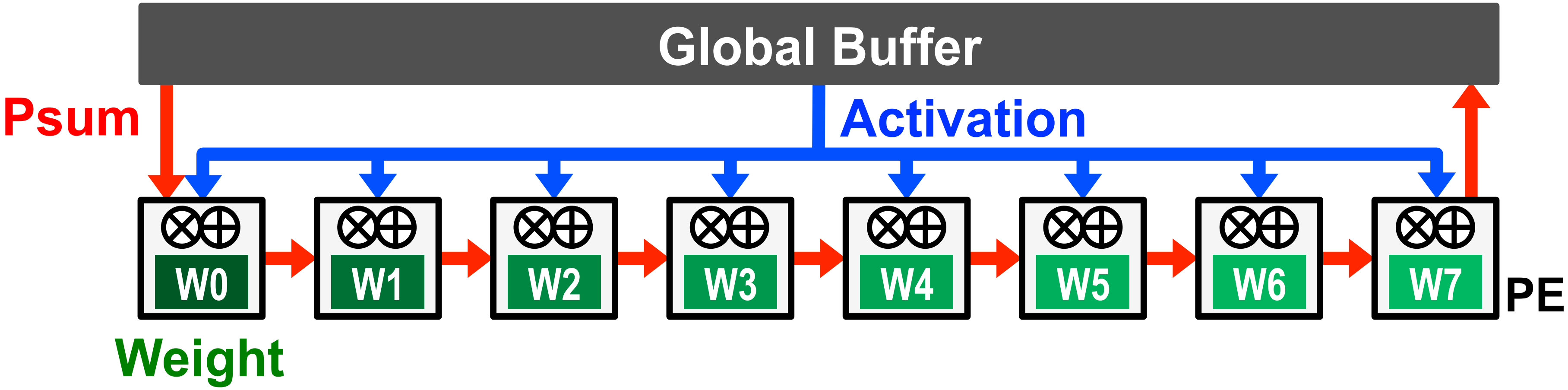
Normalized Energy Cost*



* measured from a commercial 65nm process

Farther and **larger** memories consume more power

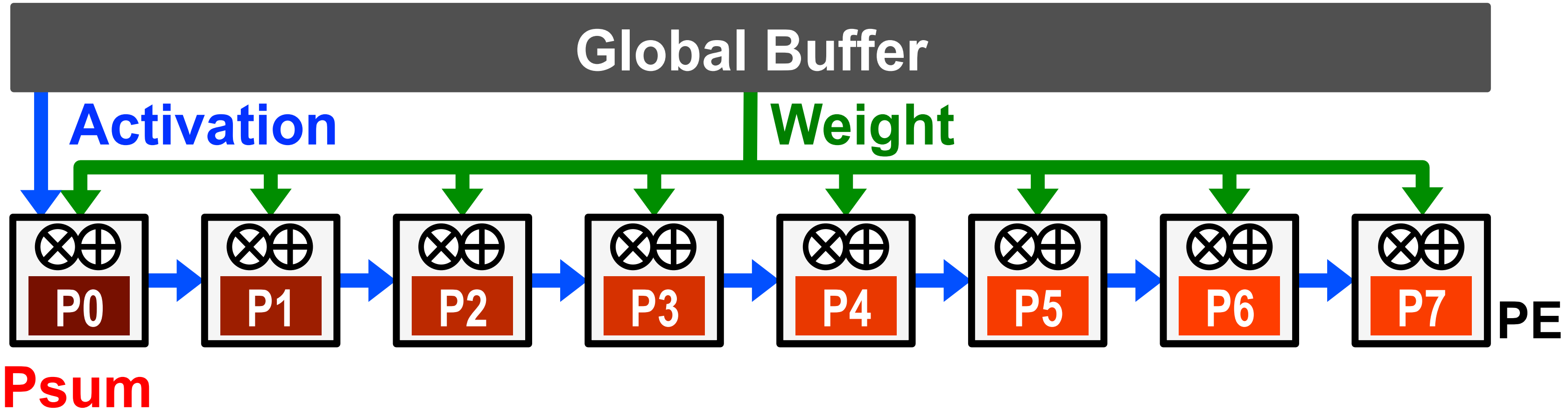
Weight Stationary (WS)



- **Minimize** **weight** read energy consumption
  - maximize convolutional and filter reuse of weights
- **Broadcast** **activations** and **accumulate** **partial sums** **spatially** across the PE array
- Examples: **TPU [Jouppi**, *ISCA* 2017], **NVDLA**

[**Chen**, *ISCA* 2016]

Output Stationary (OS)



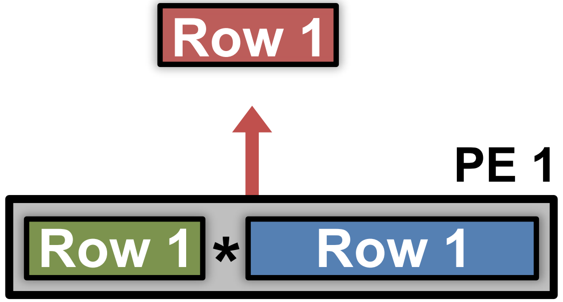
- **Minimize** **partial sum** R/W energy consumption
  - maximize local accumulation

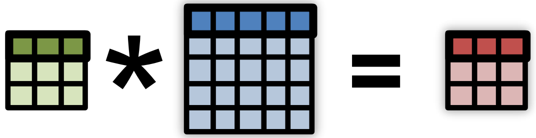
- **Broadcast/Multicast** **filter weights** and **reuse** **activations** **spatially** across the PE array

- Examples: [**Moons**, *VLSI* 2016], [**Thinker**, *VLSI* 2017]

[**Chen**, *ISCA* 2016]

Row Stationary Dataflow

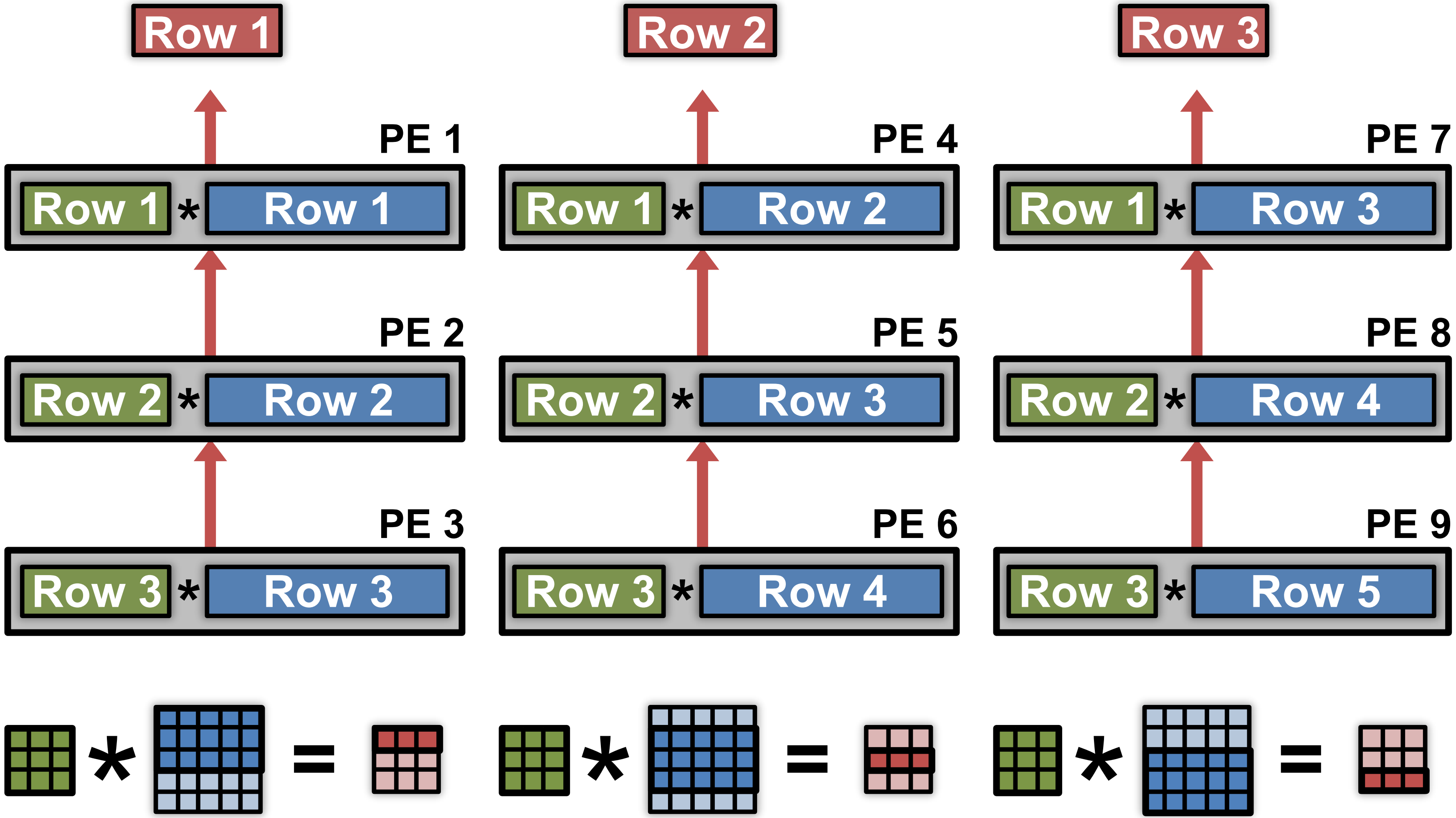




- Maximize row **convolutional reuse** in RF
  - Keep a **filter** row and **fmap** sliding window in RF
- Maximize row **psum accumulation** in RF

[**Chen**, *ISCA* 2016] **Select for Micro Top Picks**

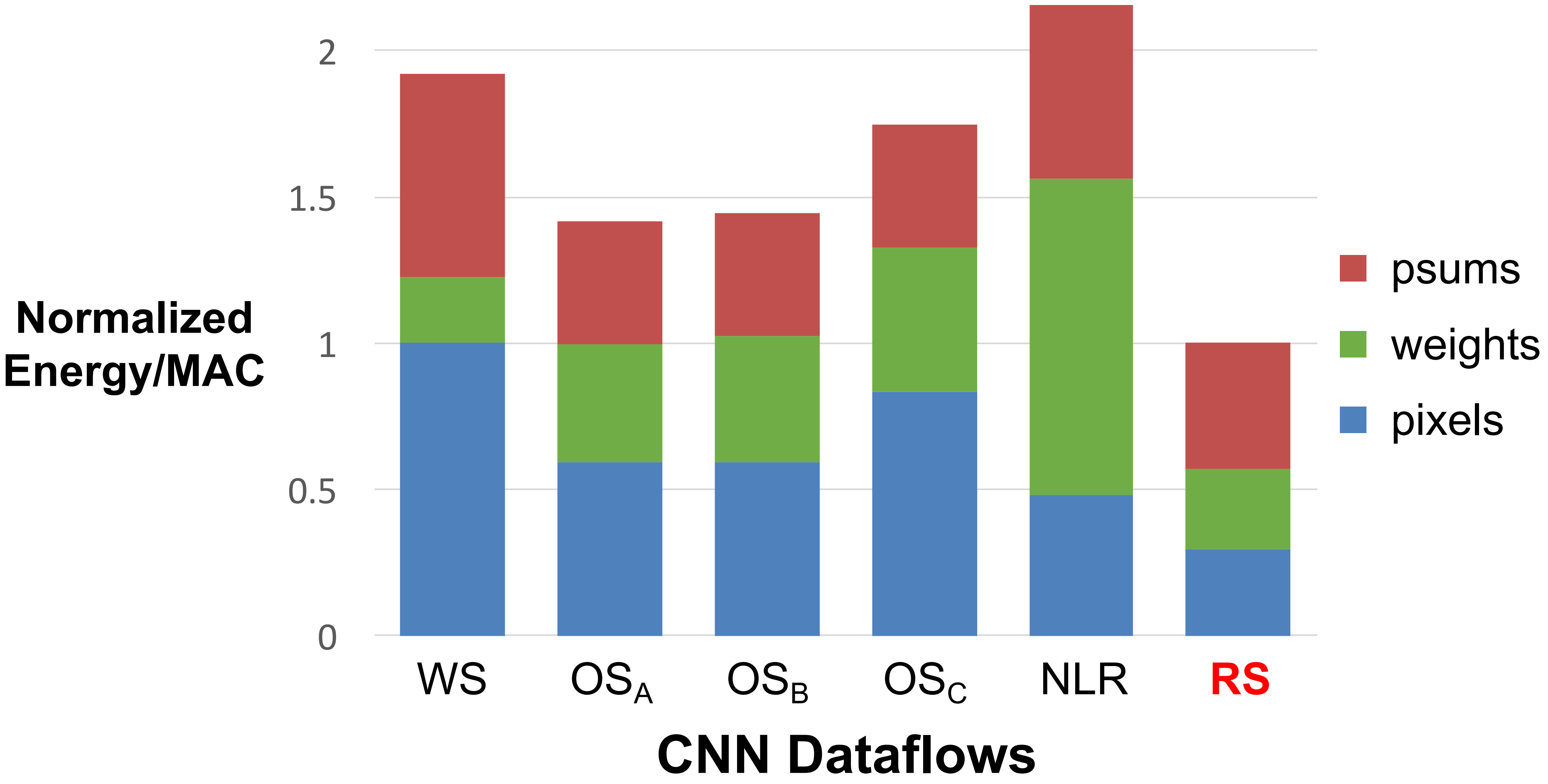
Row Stationary Dataflow



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

[**Chen**, *ISCA* 2016] **Select for Micro Top Picks**

Dataflow Comparison: CONV Layers

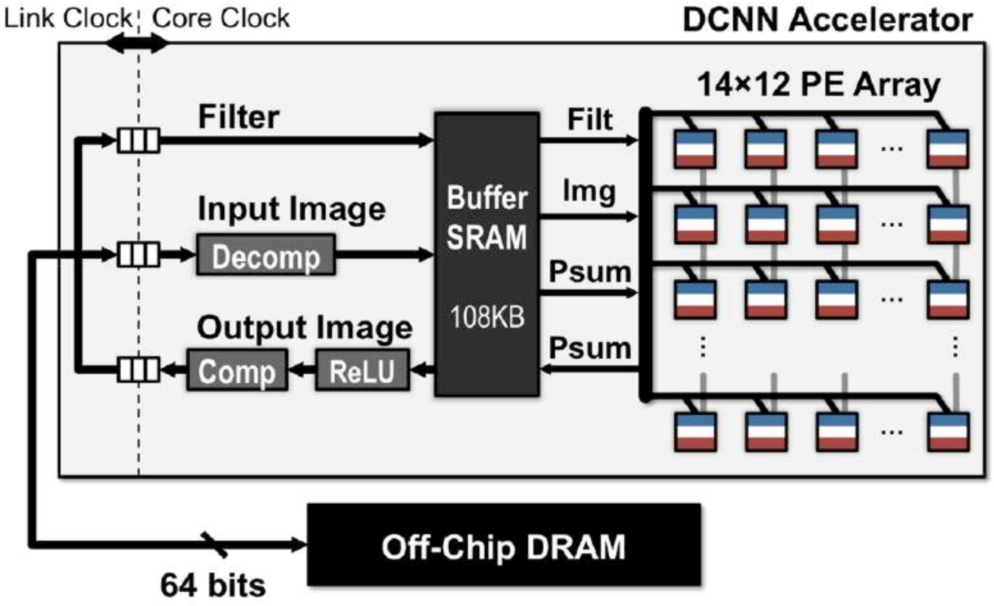


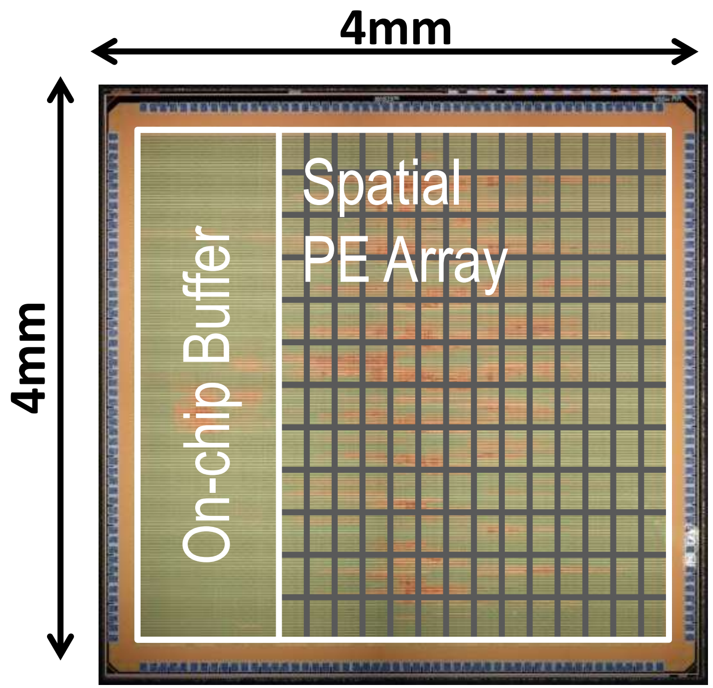
RS optimizes for the best **overall** energy efficiency

[**Chen**, *ISCA* 2016]

Deep Neural Networks at Under 0.3W

Eyeriss





[**Chen**, *ISSCC* 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)

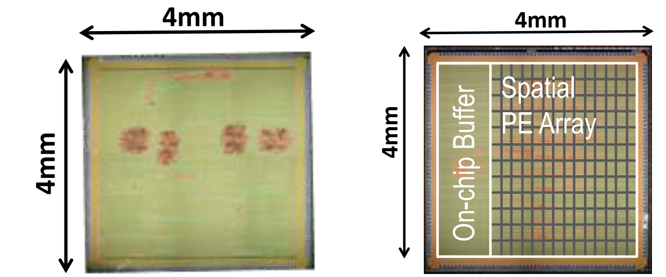
Results for AlexNet

Eyeriss Project Website: http://eyeriss.mit.edu

[Joint work with Joel Emer]

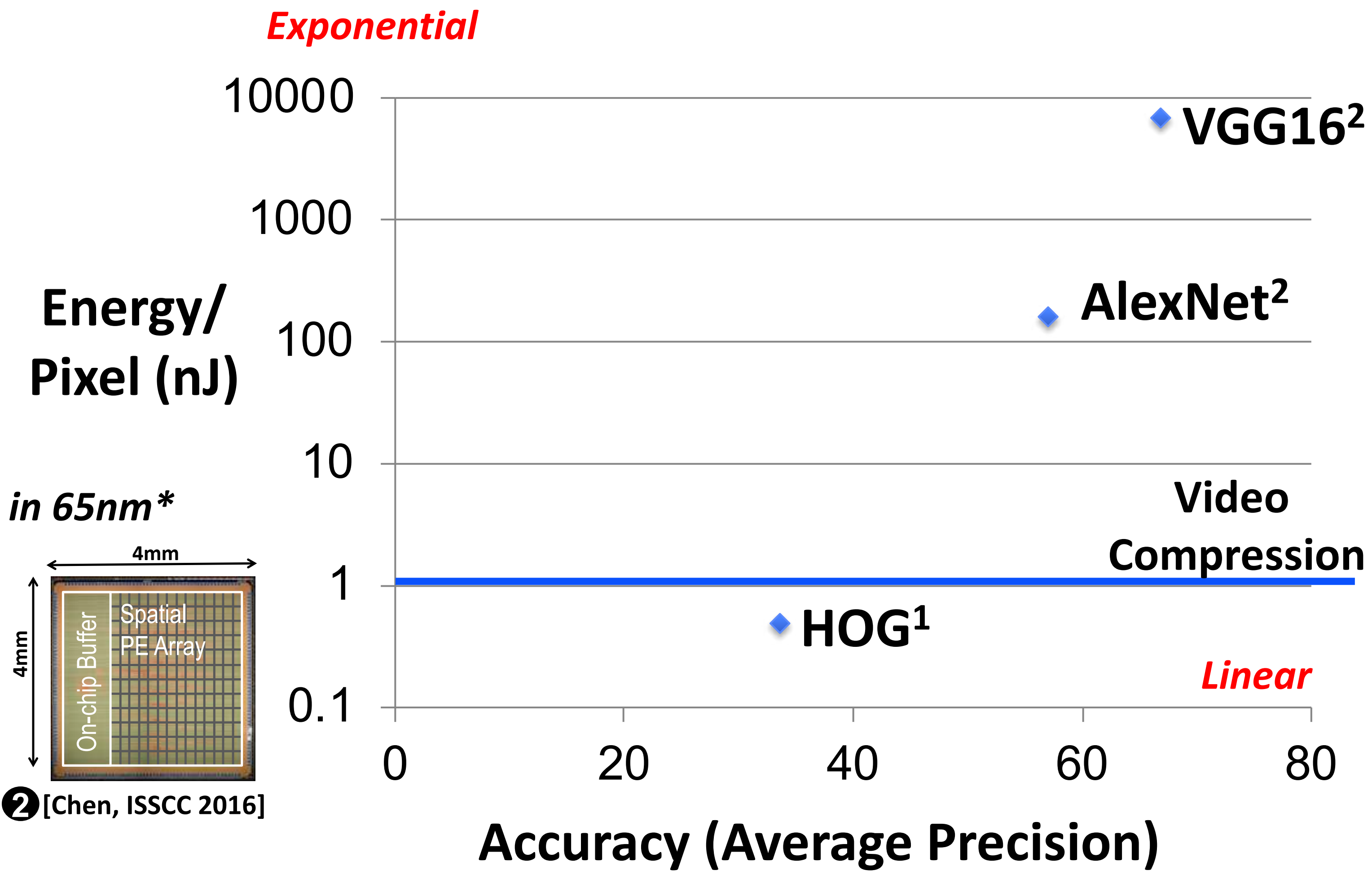
Features: Energy vs. Accuracy

*Measured in 65nm**



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

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



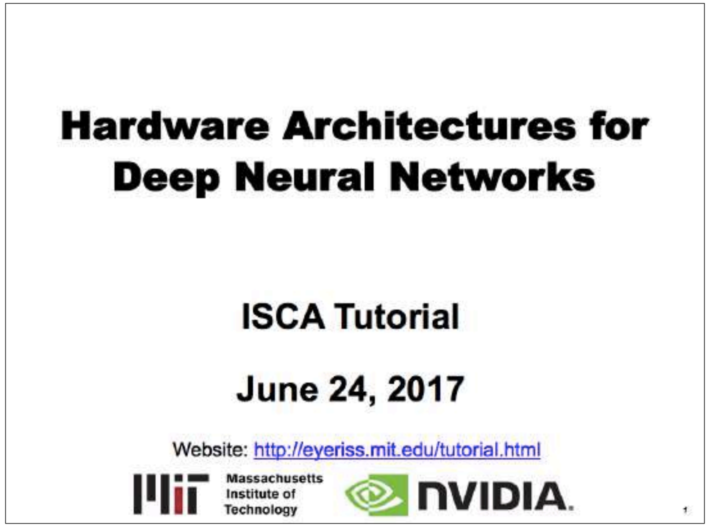
Measured in on VOC 2007 Dataset

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

[**Suleiman**, *ISCAS* 2017]

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

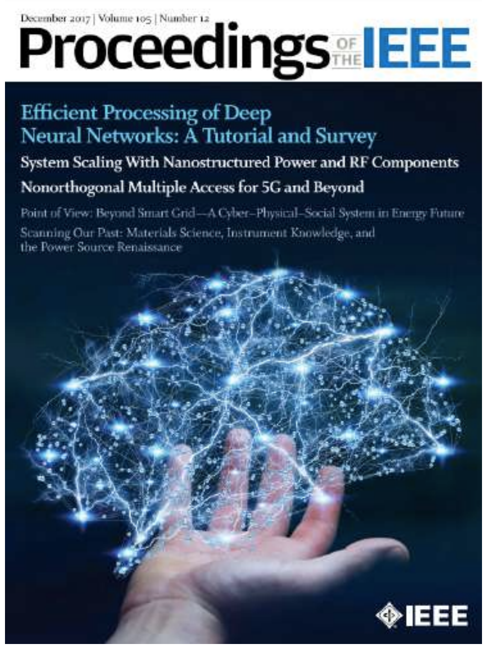
Website: http://eyeriss.mit.edu/tutorial.html

Massachusetts Institute of Technology

NVIDIA.

1

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



December 2017 | Volume 105 | Number 12

Proceedings OF THE IEEE

Efficient Processing of Deep Neural Networks: A Tutorial and Survey

System Scaling With Nanostructured Power and RF Components

Nonorthogonal Multiple Access for 5G and Beyond

Point of View: Beyond Smart Grid—A Cyber–Physical–Social System in Energy Future

Scanning Our Past: Materials Science, Instrument Knowledge, and the Power Source Renaissance

IEEE

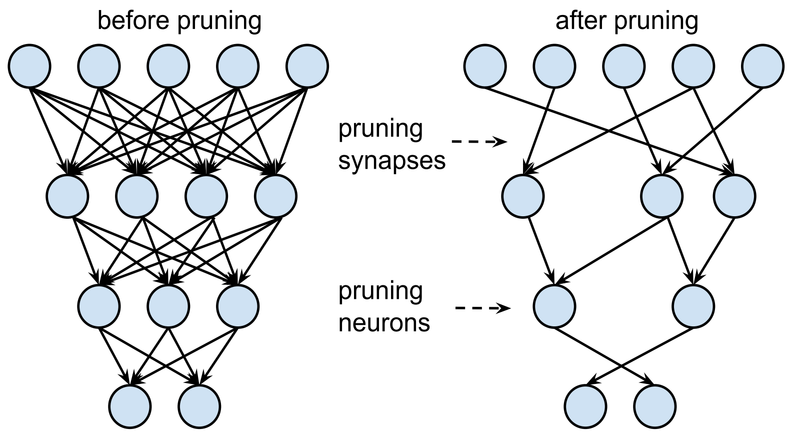
V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, ***"Efficient Processing of Deep Neural Networks: A Tutorial and Survey,"*** Proceedings of the IEEE, Dec. 2017

We identified various limitations to existing approaches

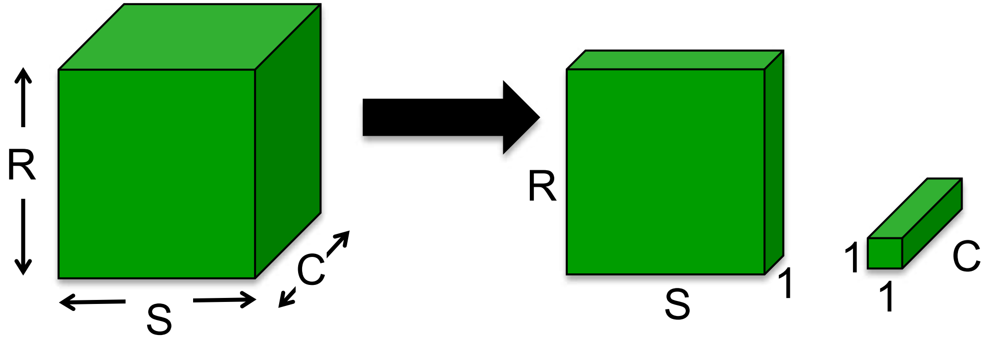
Design of Efficient DNN Algorithms

Popular efficient DNN algorithm approaches

Network Pruning



Efficient Network Architectures



Examples: SqueezeNet, MobileNet

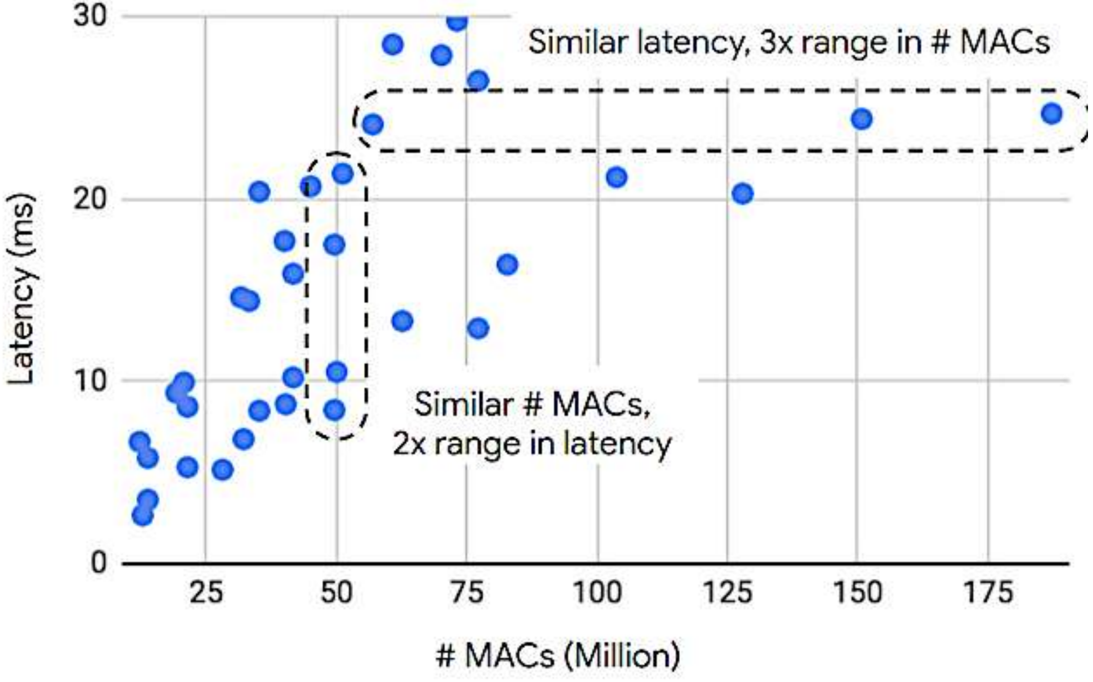
... also reduced precision

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

[**Chen*, Yang***, *SysML* 2018]

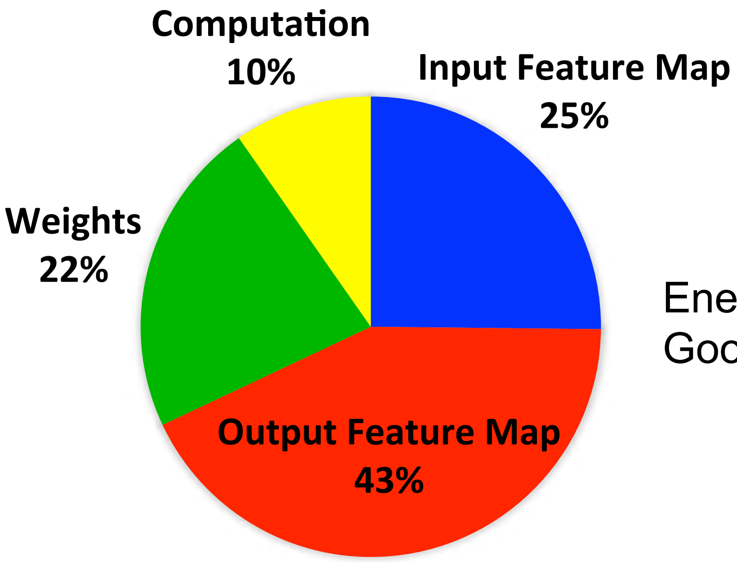
Number of MACs and Weights are Not Good Proxies

# of operations (MACs) does not approximate latency well



Source: Google (https://ai.googleblog.com/2018/04/introducing-cvpr-2018-on-device-visual.html)

# of weights ***alone*** is not a good metric for energy (**All data types** should be considered)



Energy breakdown of GoogLeNet

https://energyestimation.mit.edu/

[**Yang**, *CVPR* 2017]

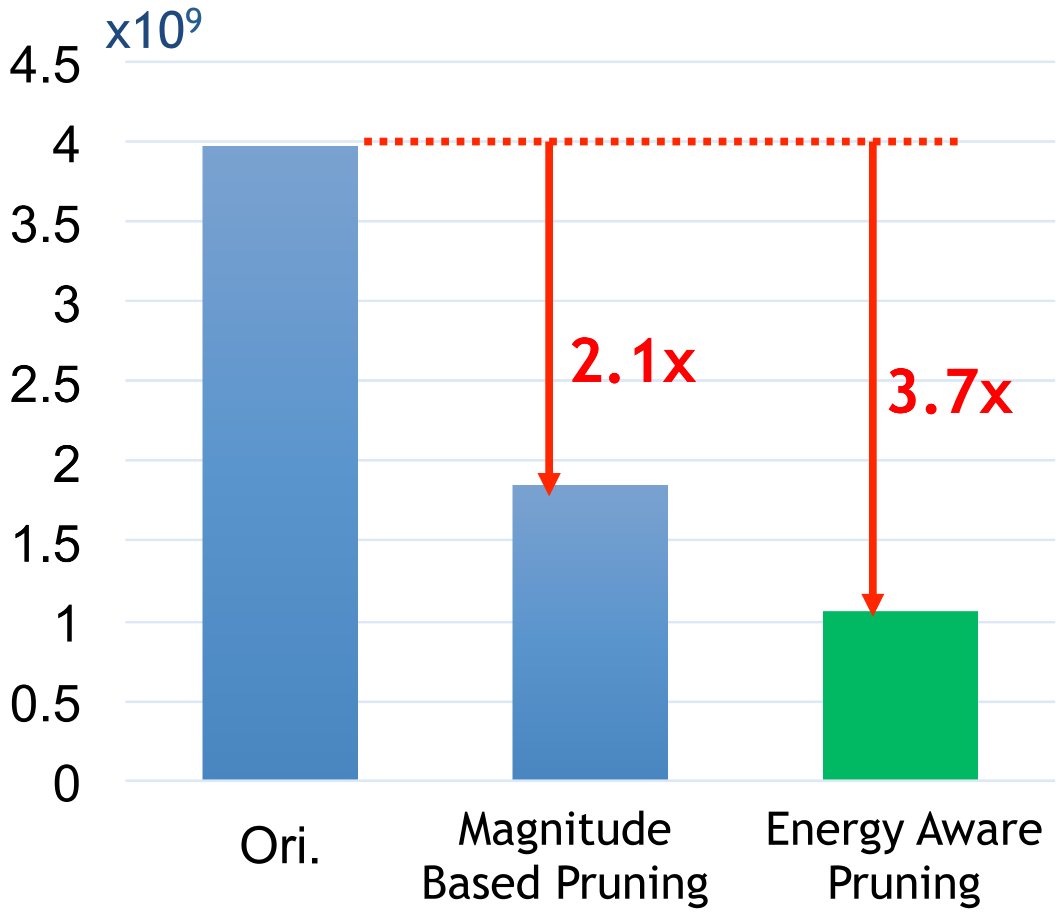
Energy-Aware Pruning

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 the most energy first
- **Energy-aware pruning** reduces AlexNet energy by **3.7x** w/ similar accuracy
- Outperforms magnitude-based pruning by **1.7x**

[**Yang**, *CVPR* 2017]

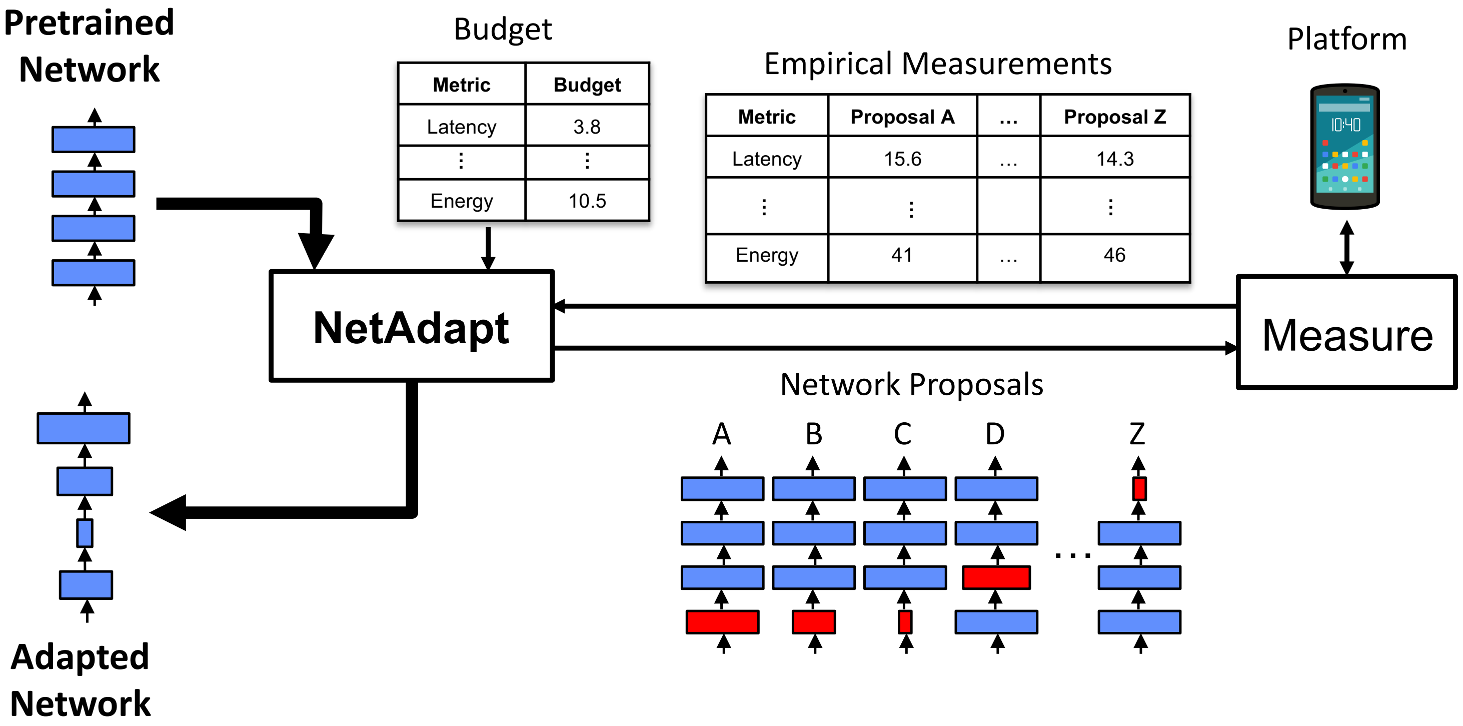
Normalized Energy (AlexNet)



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

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)
- **Few hyperparameters** to reduce tuning effort
- **>1.7x speed up** on MobileNet w/ similar accuracy



Budget

Metric	Budget
Latency	3.8
⋮	⋮
Energy	10.5

Empirical Measurements

Metric	Proposal A	…	Proposal Z
Latency	15.6	…	14.3
⋮	⋮		⋮
Energy	41	…	46

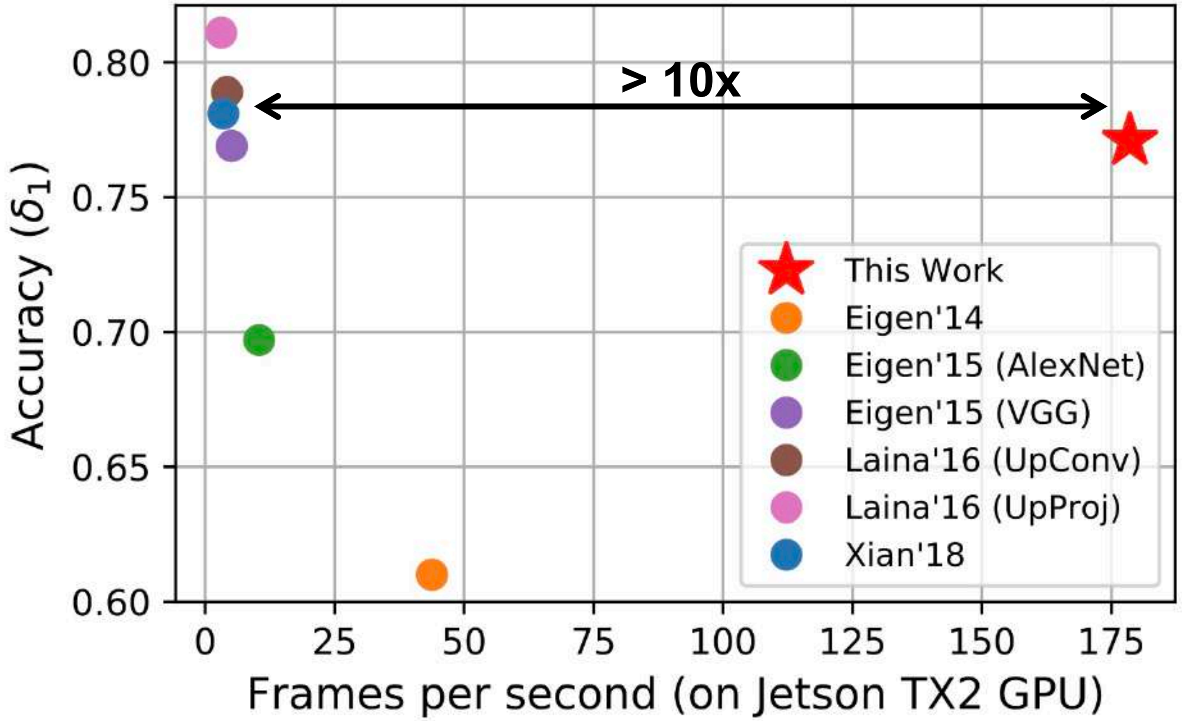
[**Yang**, *ECCV* 2018]

Code available at
http://netadapt.mit.edu

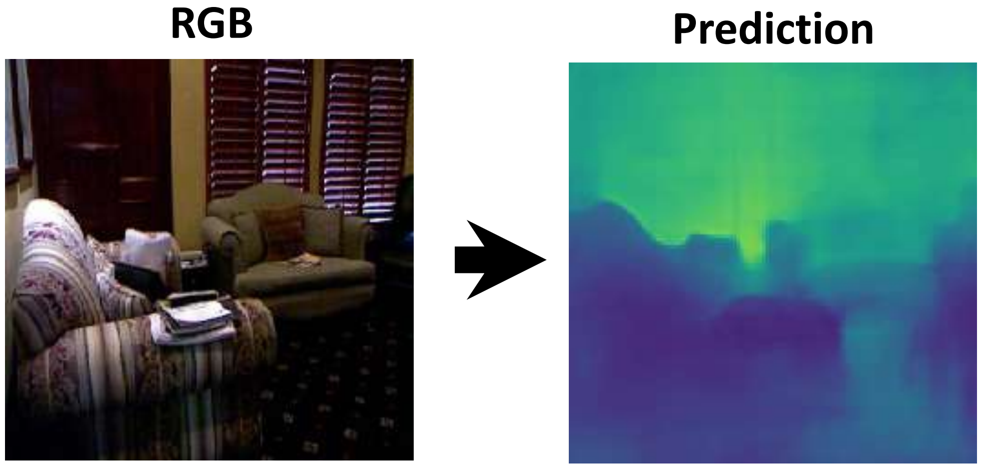
[In collaboration with Google's Mobile Vision Team]

FastDepth: Fast Monocular Depth Estimation

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



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



~40fps on an iPhone

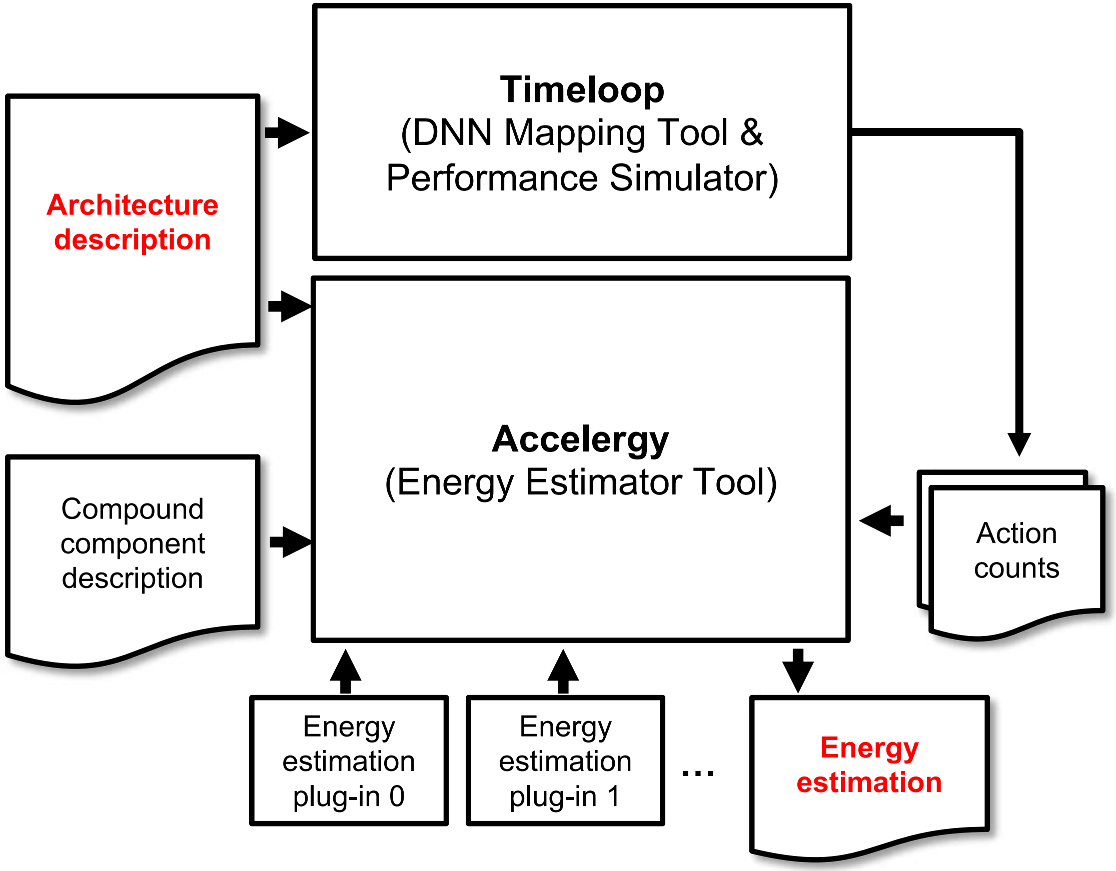
Models available at http://fastdepth.mit.edu

[**Wofk*, Ma***, *ICRA* 2019]

[Joint work with Sertac Karaman]

DNN Accelerator Evaluation Tools

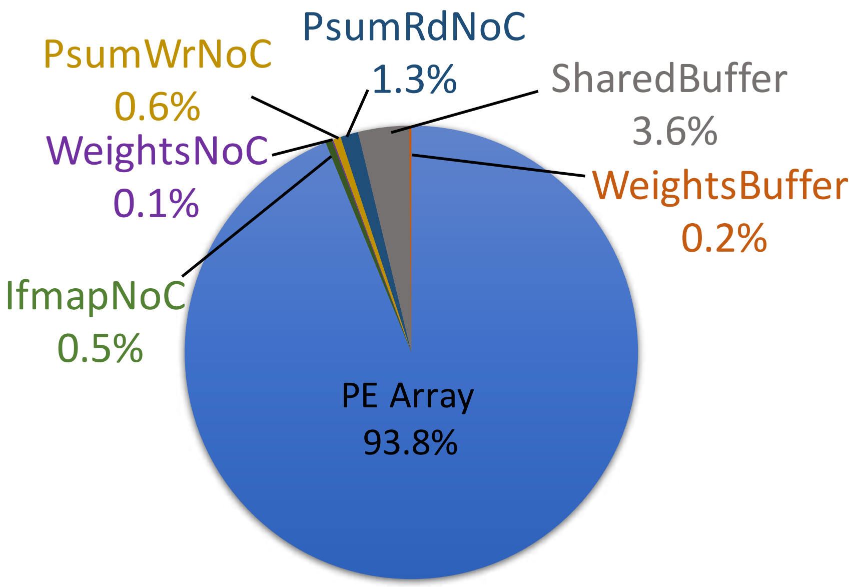
- Require systematic way to
  - Evaluate and compare DNN accelerators
  - Rapidly explore design space
- Accelergy [Wu, *ICCAD* 2019]
  - Early stage estimation tool at the architecture level
    - Estimate energy based on architecture level components (e.g., # of PEs, memory size, on-chip network)
  - Evaluate architecture level impact of emerging devices
    - Plug-ins for different technologies
- Timeloop [Parashar, *ISPASS* 2019]
  - DNN mapping tool
  - Performance Simulator → Action counts


Architecture description
Timeloop
(DNN Mapping Tool & Performance Simulator)
Accelergy
(Energy Estimator Tool)
Compound component description
Action counts
Energy estimation plug-in 0
Energy estimation plug-in 1
…
Energy estimation

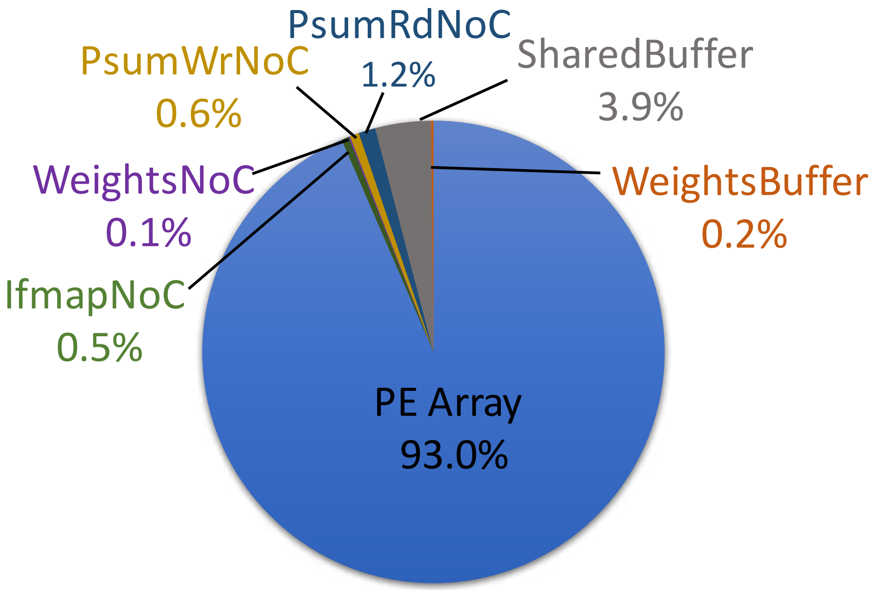
Open-source code available at:
http://accelergy.mit.edu

Accelergy Estimation Validation

- Validation on Eyeriss [**Chen**, *ISSCC* 2016]
  - Achieves 95% accuracy compared to post-layout simulations
  - Can accurately captures energy breakdown at different granularities



Ground Truth Energy Breakdown



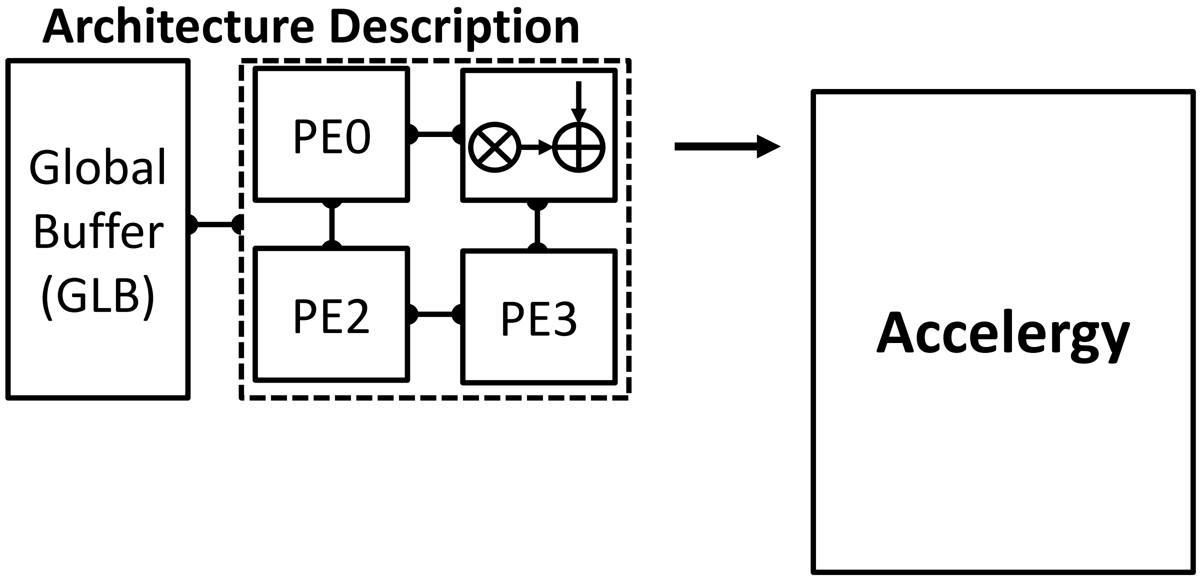
Accelergy Energy Breakdown

Open-source code available at: http://accelergy.mit.edu

[**Wu**, *ICCAD* 2019]

Accelergy Infrastructure

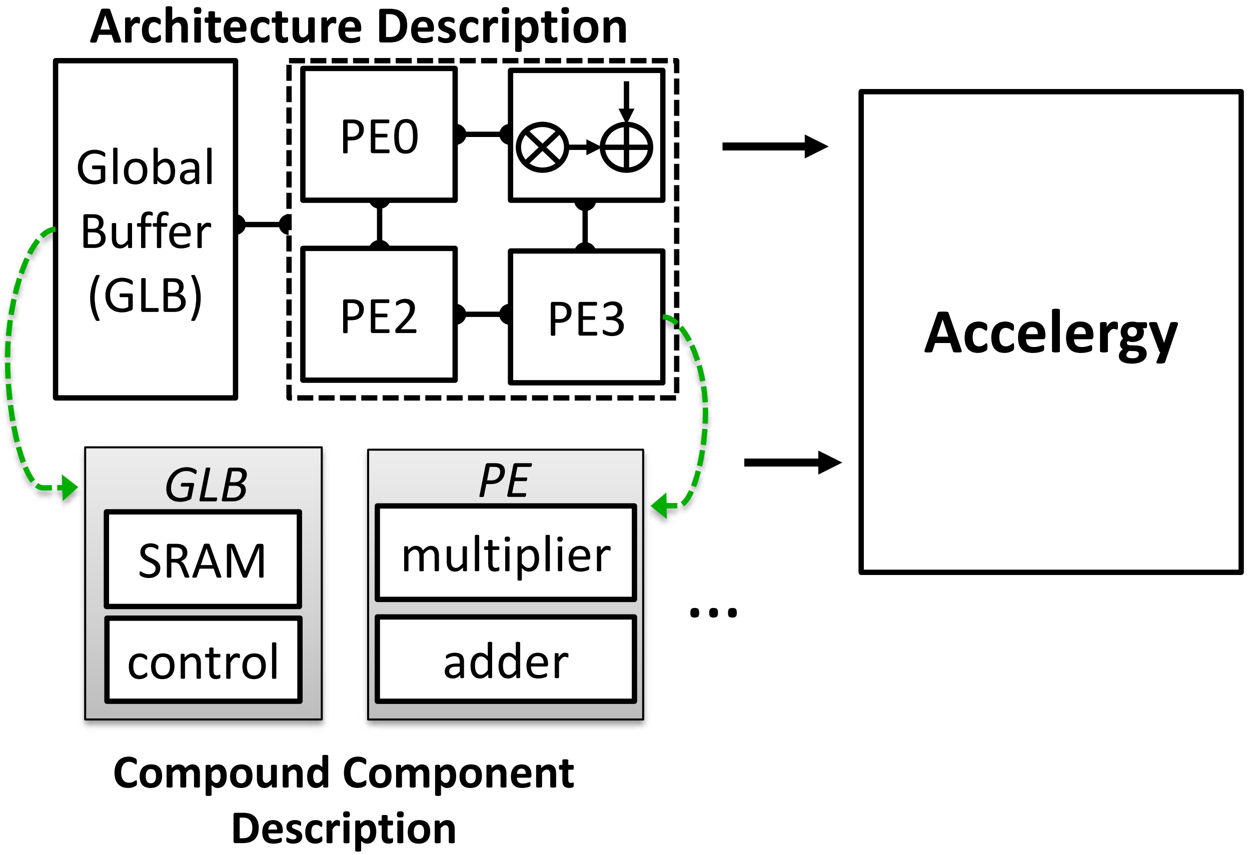
Open-source code available at:
http://accelergy.mit.edu



[**Wu**, *ICCAD* 2019]

Accelergy Infrastructure

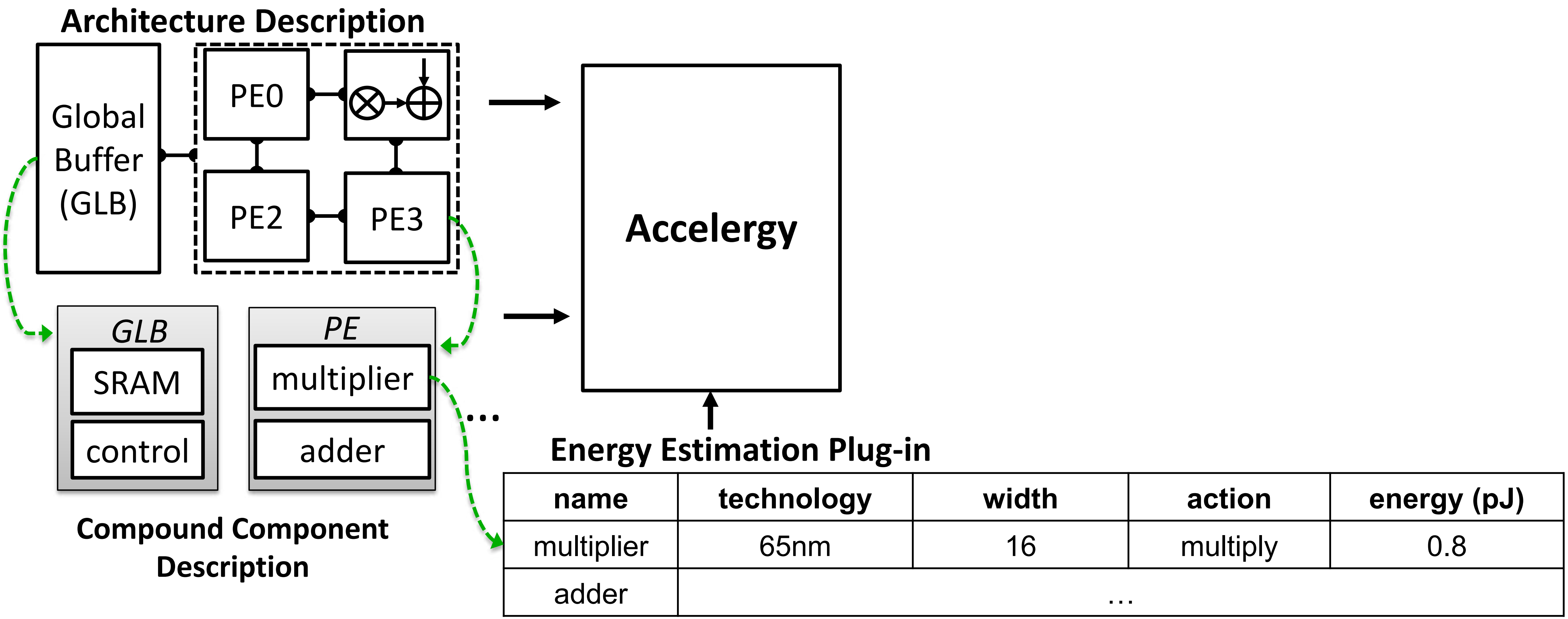
Open-source code available at:
http://accelergy.mit.edu



[**Wu**, *ICCAD* 2019]

Accelergy Infrastructure

Open-source code available at:
http://accelergy.mit.edu

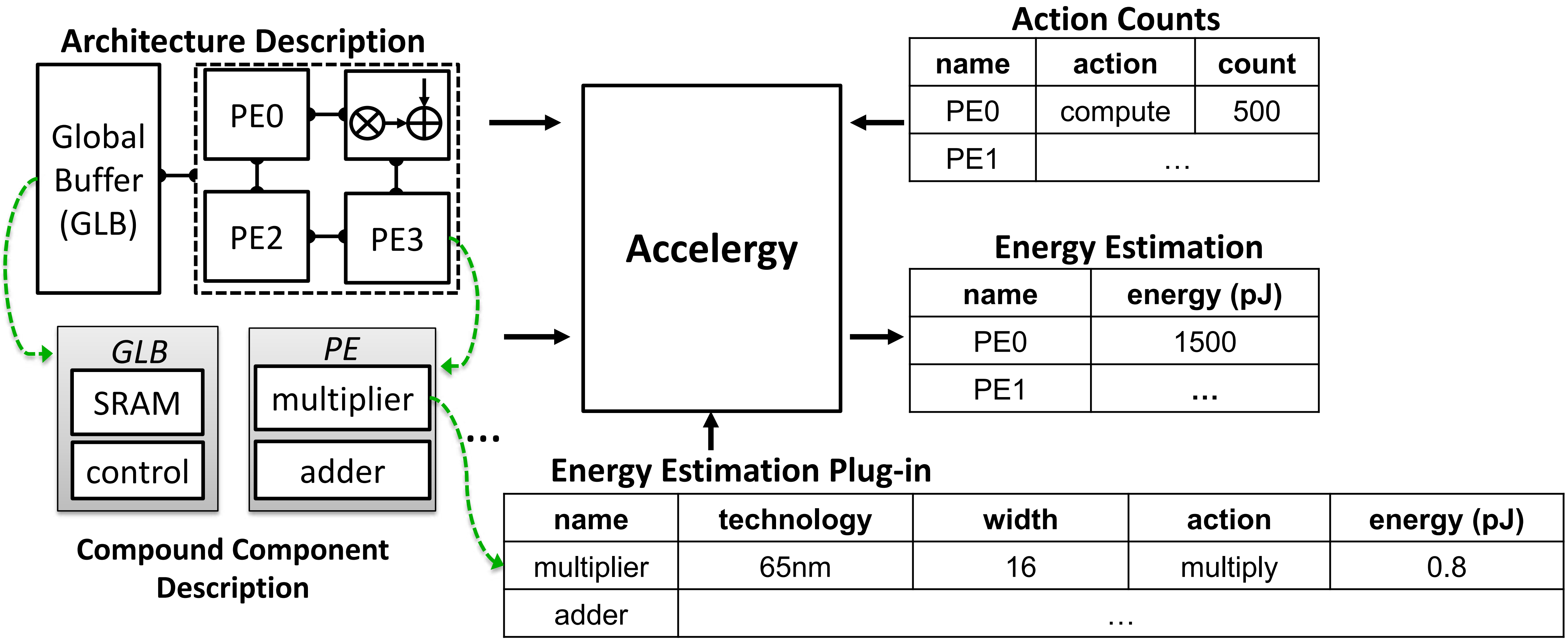


name	technology	width	action	energy (pJ)
multiplier	65nm	16	multiply	0.8
adder	…			

[**Wu**, *ICCAD* 2019]

Accelergy Infrastructure

Open-source code available at:
http://accelergy.mit.edu



Action Counts

name	action	count
PE0	compute	500
PE1	…	

Energy Estimation

name	energy (pJ)
PE0	1500
PE1	…

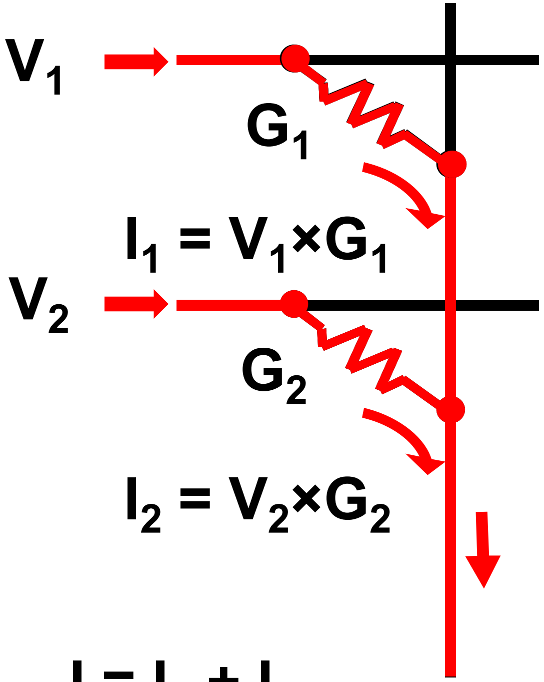
Energy Estimation Plug-in

name	technology	width	action	energy (pJ)
multiplier	65nm	16	multiply	0.8
adder	…			

[**Wu**, *ICCAD* 2019]

In-Memory Computing (IMC)

Activation is input voltage (V_i)
Weight is resistor conductance (G_i)



Psum is output current

$$I = I_1 + I_2$$
$$= V_1 \times G_1 + V_2 \times G_2$$

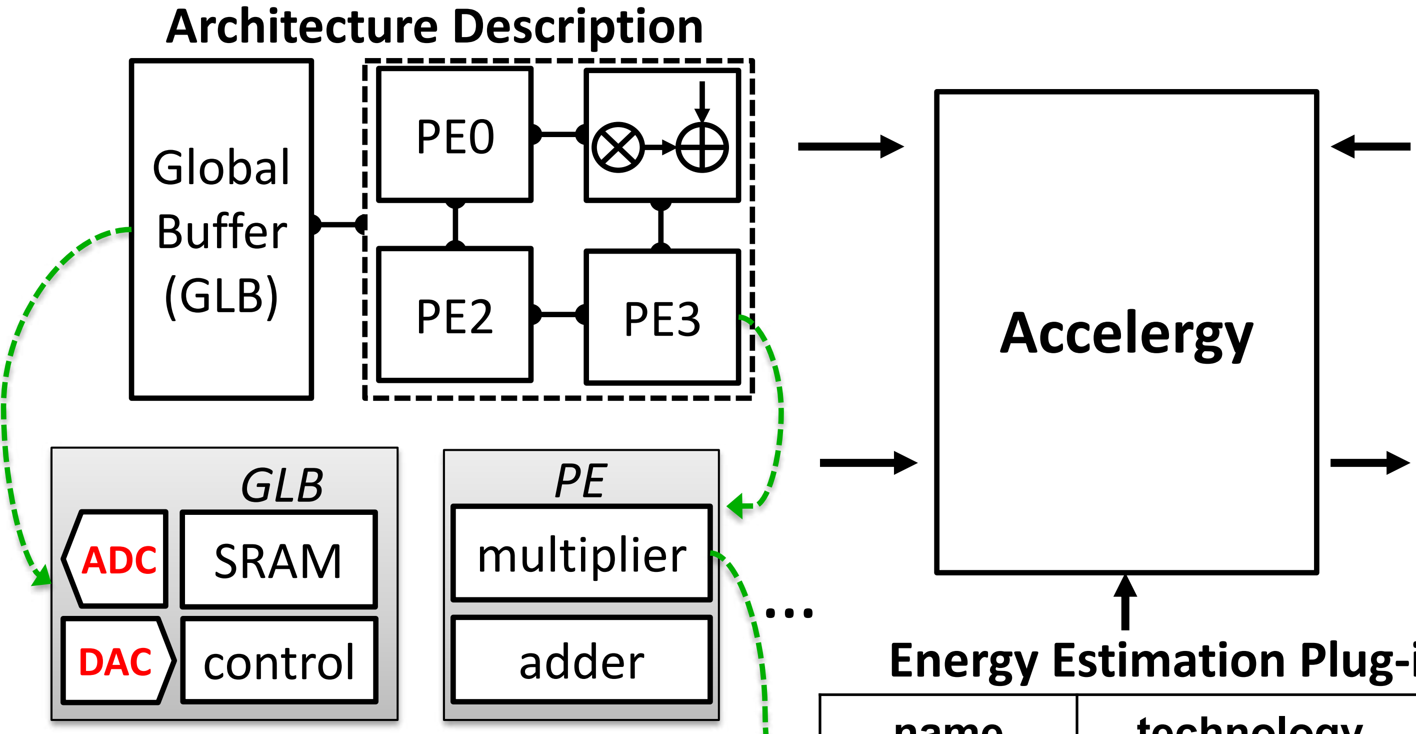
Image Source: [**Shafiee**, *ISCA* 2016]

- Reduce data movement by **moving compute into memory**
- Compute MAC with memory storage element
- **Analog Compute**
  - – Activations, weights and/or partial sums are encoded with analog voltage, current, or resistance
  - – Increased sensitivity to circuit non-idealities
  - – A/D and D/A circuits to interface with digital domain
- Leverage **emerging memory device technology**

eNVM:[**Yu**, *PIEEE* 2018], SRAM:[**Verma**, *SSCS* 2019]

Accelergy for IMC

Open-source code available at:
http://accelergy.mit.edu



Compound Component Description

Action Counts

name	action	count
PE0	compute	500
PE1	…	

Energy Estimation

name	energy (pJ)
PE0	~~1500~~ E_{total}
PE1	…

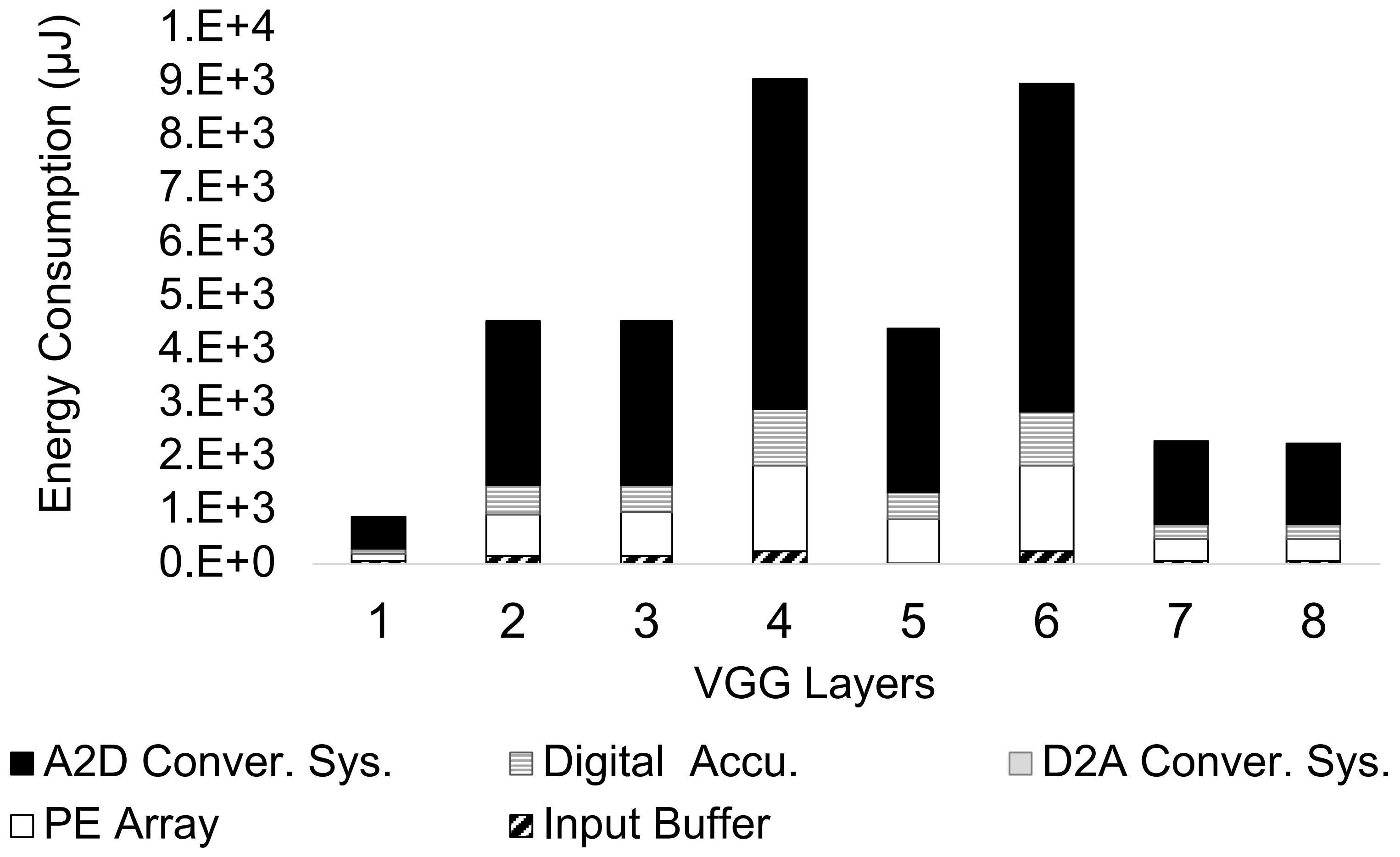
Energy Estimation Plug-in

name	technology	width	action	energy (pJ)
multiplier	~~65nm~~ memristor	16	multiply	~~0.8~~ E_{mult}
adder	…			
ADC				
DAC				

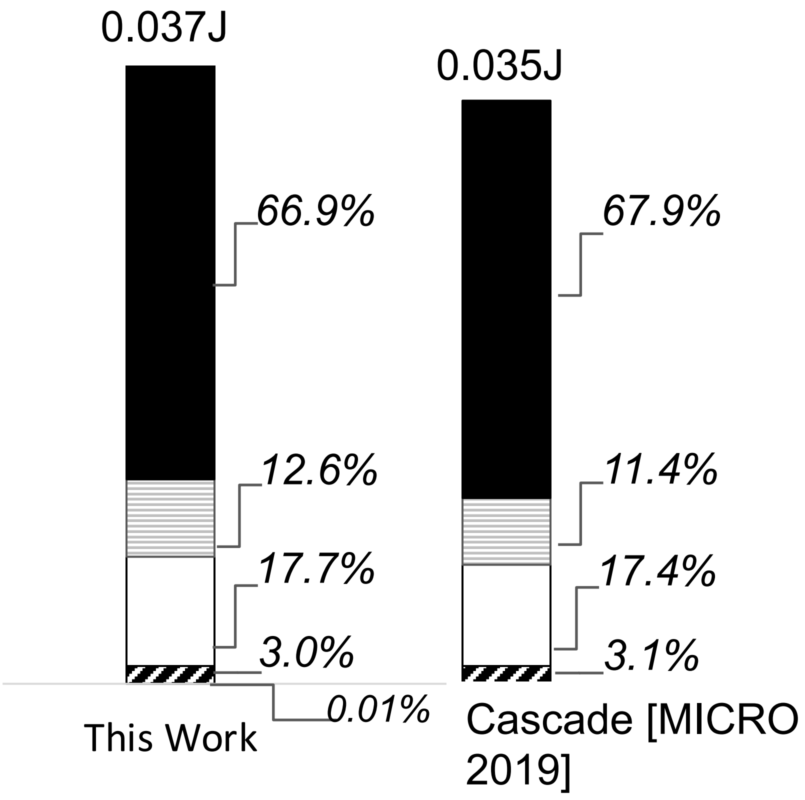
Accelergy for IMC

Open-source code available at:
http://accelergy.mit.edu

Energy breakdown across layers



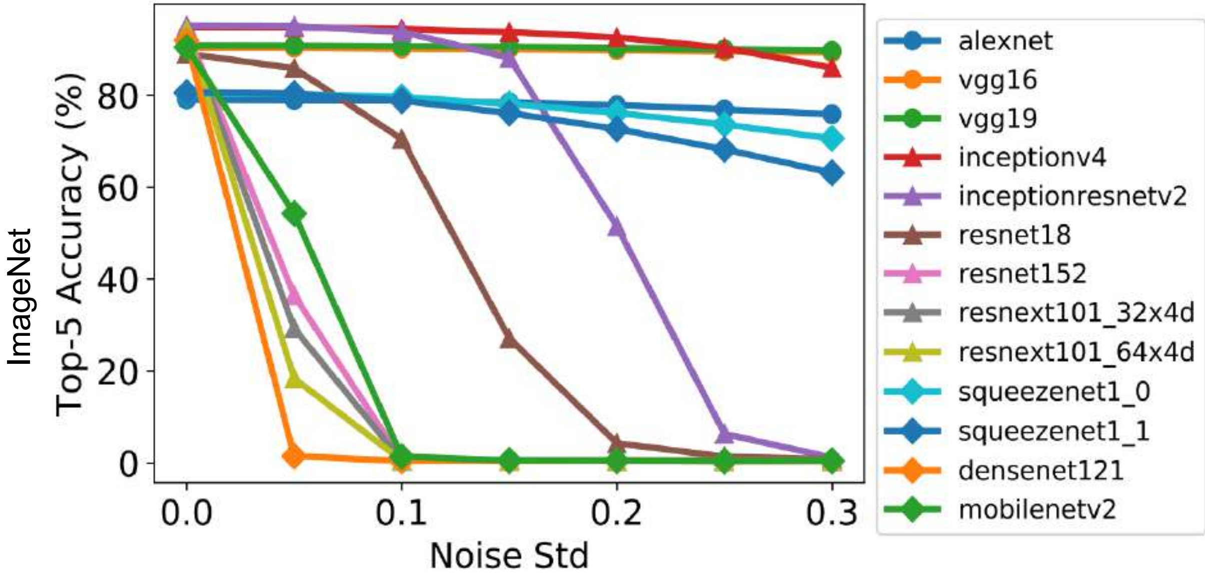
Achieves ~95% accuracy

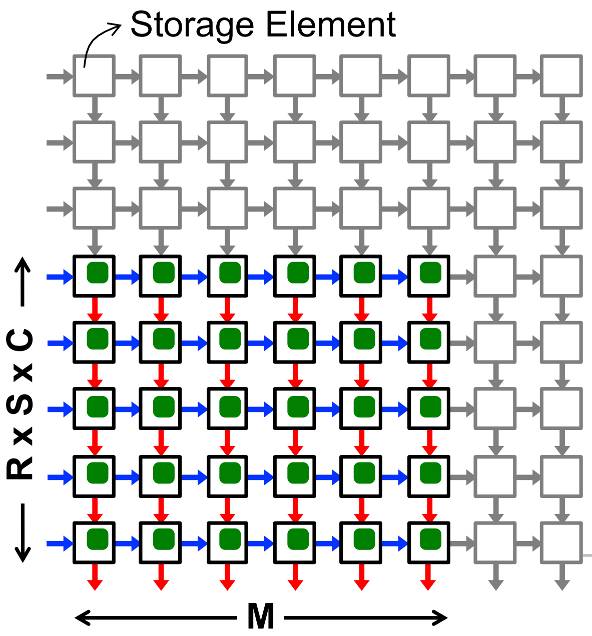


[**Wu**, *ISPASS* 2020]

Designing DNNs for IMC

- Designing DNNs for IMC may differ from DNNs for digital processors
- Highest accuracy DNN on digital processor may be different on IMC
  - Accuracy drops based on robustness to non-idealities
- Reducing number of weights is less desirable
  - Since IMC is weight stationary, may be better to reduce number of activations
  - IMC tend to have larger arrays → fewer weights may lead to low utilization on IMC

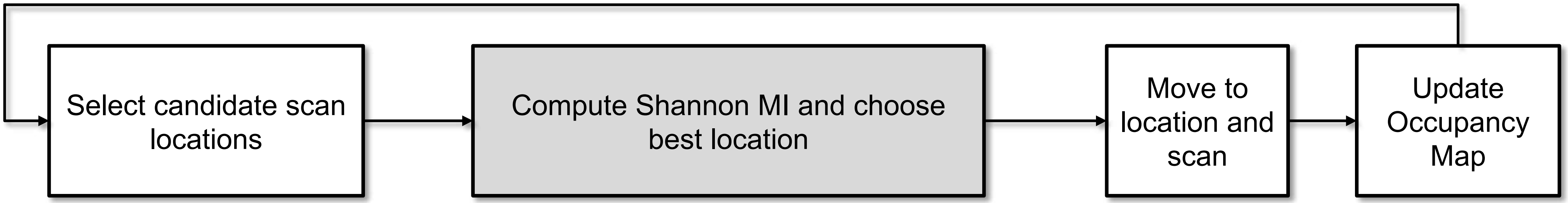




[**Yang**, *IEDM* 2019]

Where to Go Next: Planning and Mapping

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



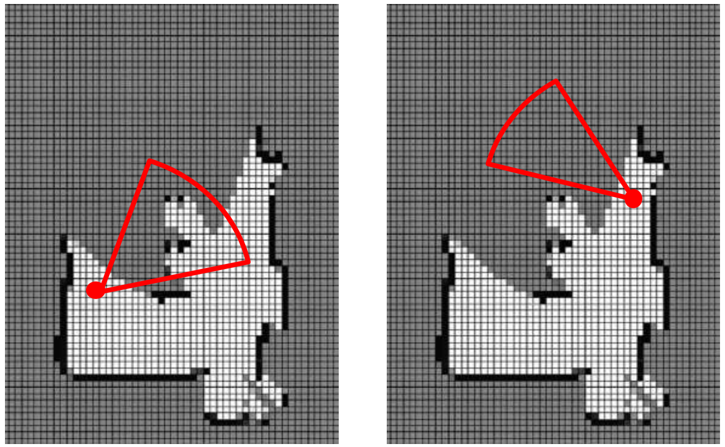
Select candidate scan locations

Compute Shannon MI and choose best location

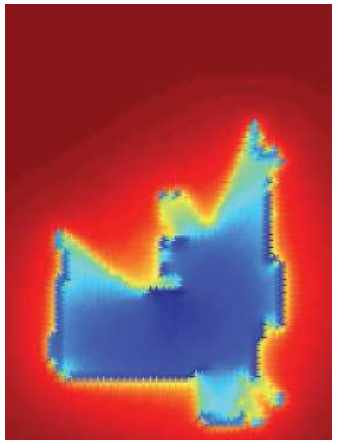
Move to location and scan

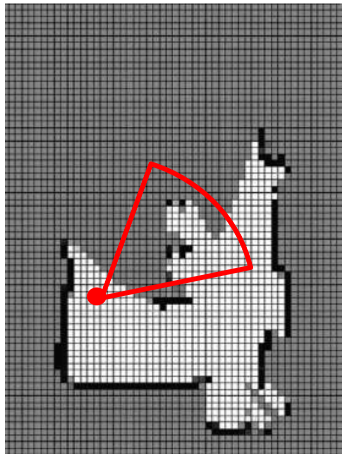
Update Occupancy Map

Where to scan?

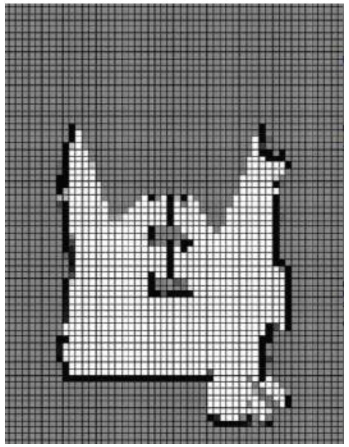


Mutual Information



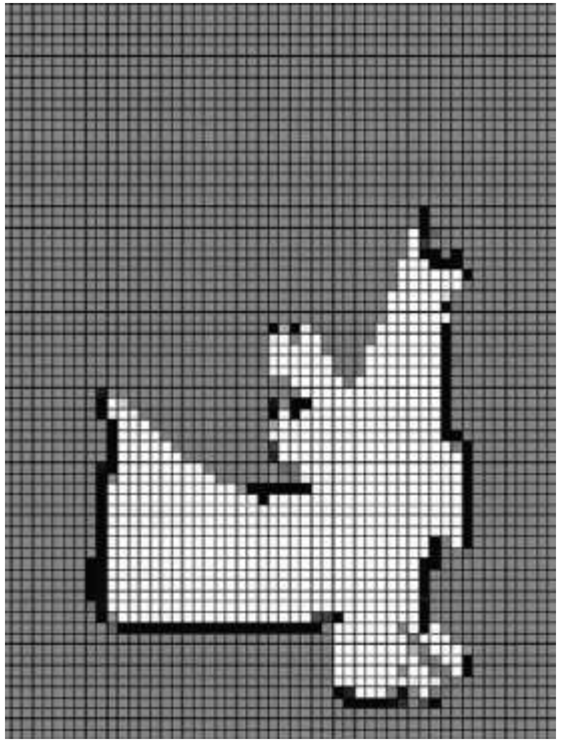


Updated Map

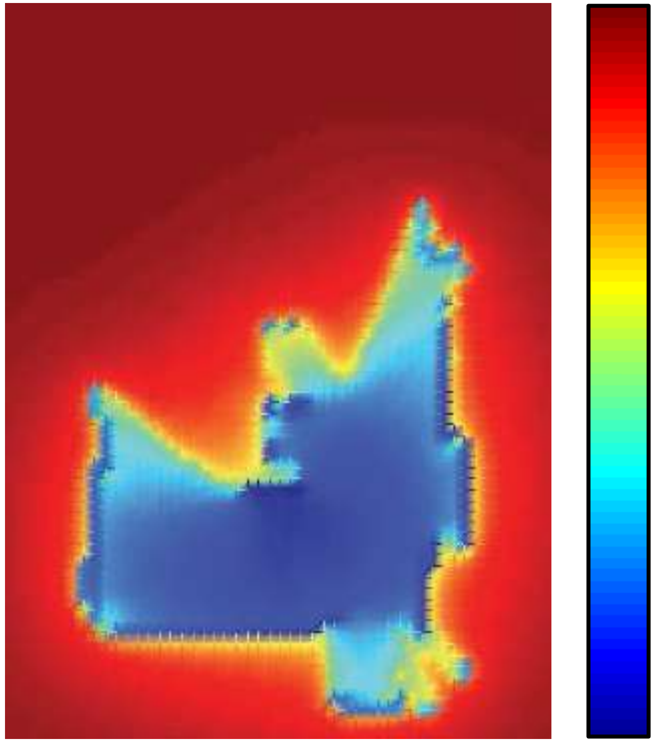


[Joint work with Sertac Karaman]

Information Theoretic Mapping



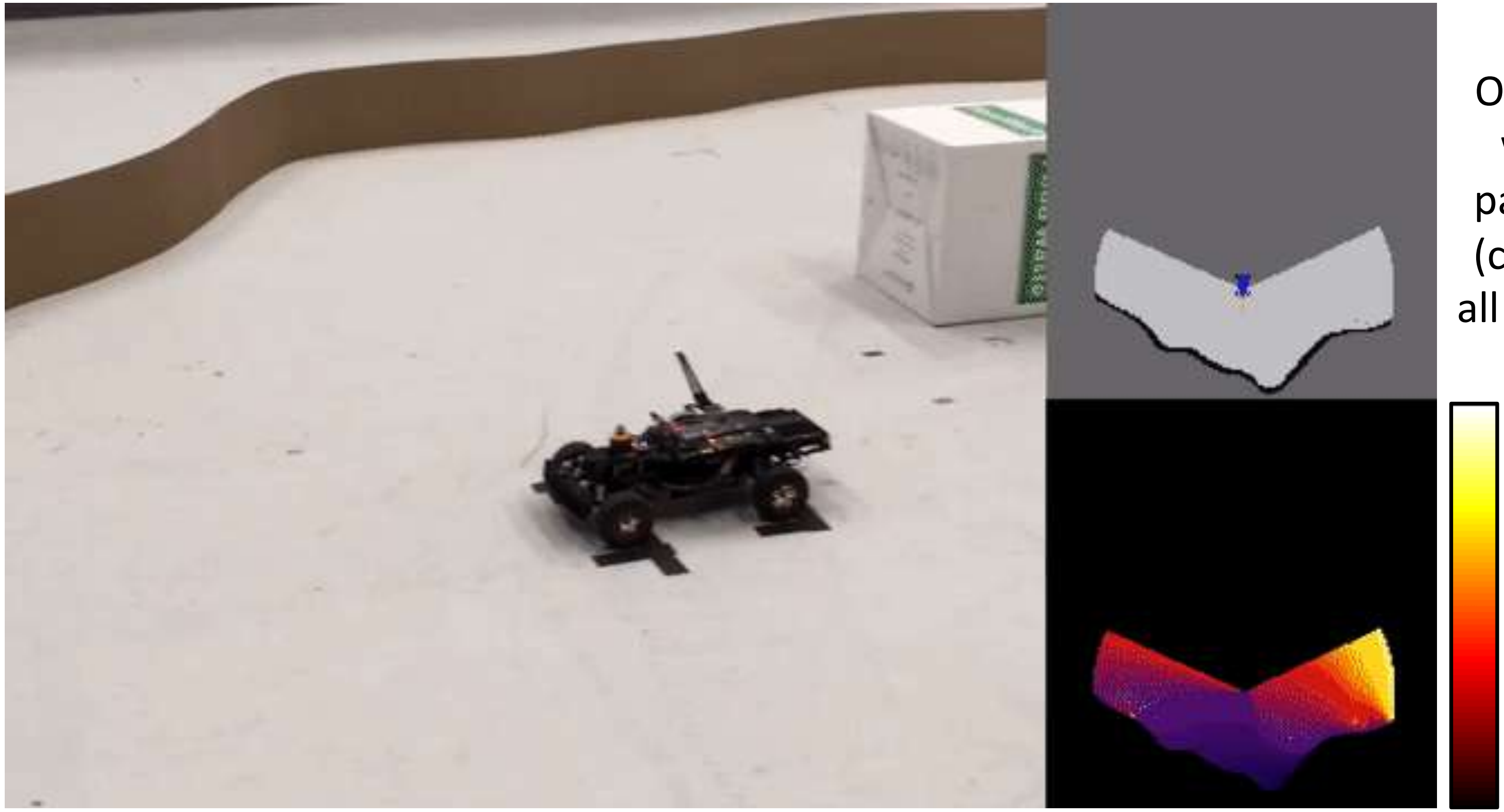
Occupancy grid map, M



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

$$\underbrace{H(M|Z)}_{\text{Perspective updated map entropy}} = \underbrace{H(M)}_{\text{Current map entropy}} - \underbrace{I(M;Z)}_{\text{Mutual information}}$$

Experimental Results (4x Real Time)



Occupancy map with planned path using RRT* (compute MI on all possible paths)

MI surface

Exploration with a mini race car using motion capture for localization

[**Zhang**, *ICRA* 2019]

Building Hardware to Compute MI

Motivation: Compute MI faster for faster exploration!

Approximate FSMI

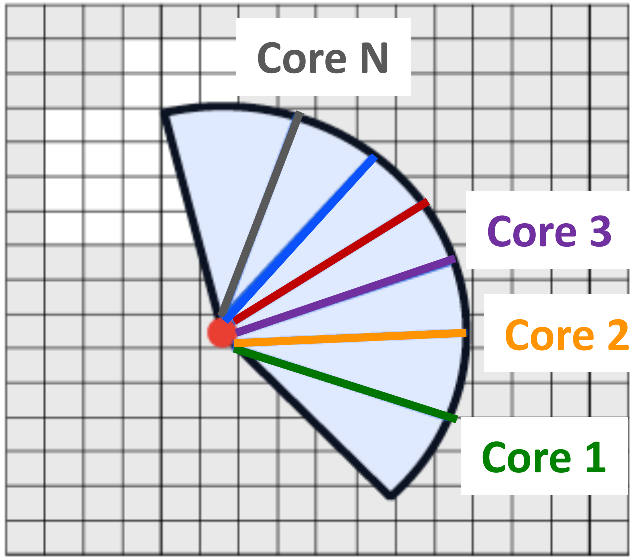
[**Zhang**, *ICRA* 2019]

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

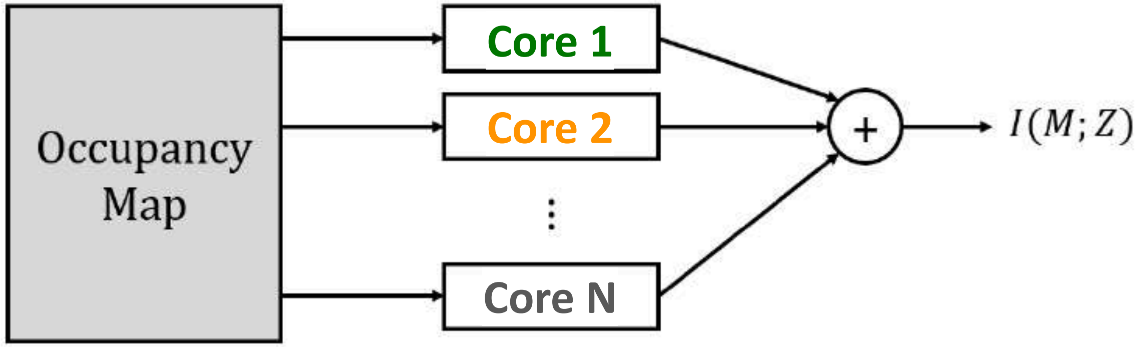
Evaluate MI for all cells in entire beam altogether **removes numerical integration**

Algorithm is ***embarrassingly*** parallel!

High throughput ***should*** be possible with multiple cores.

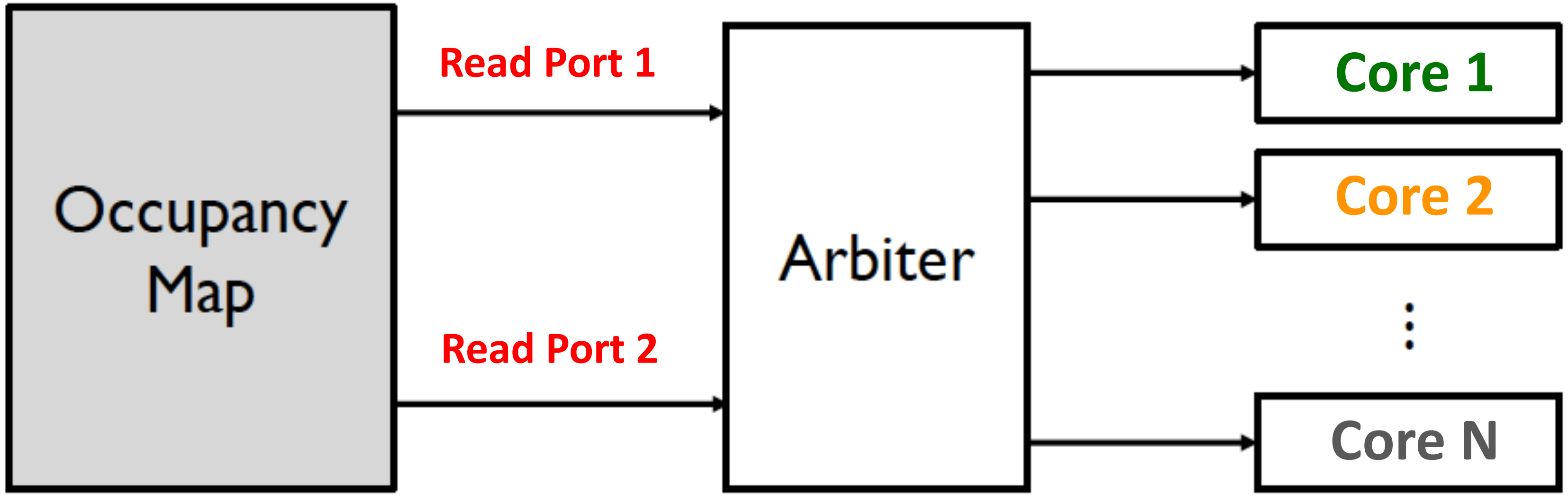


Process beams in parallel with multiple cores



Challenge is Data Delivery to All Cores

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

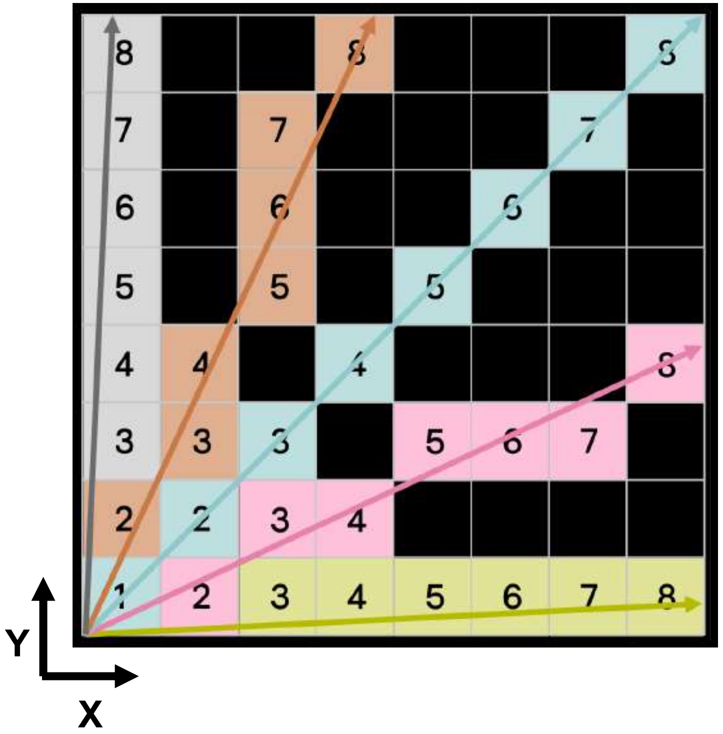


Data delivery, specifically memory bandwidth, limits the throughput (not compute)

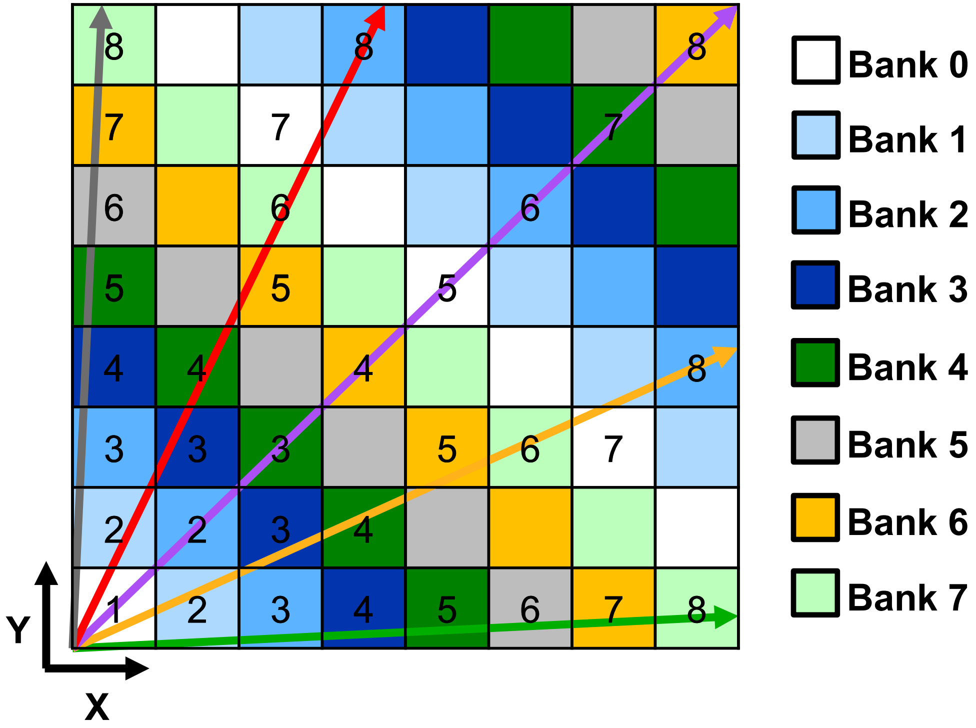
Specialized Memory Architecture

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

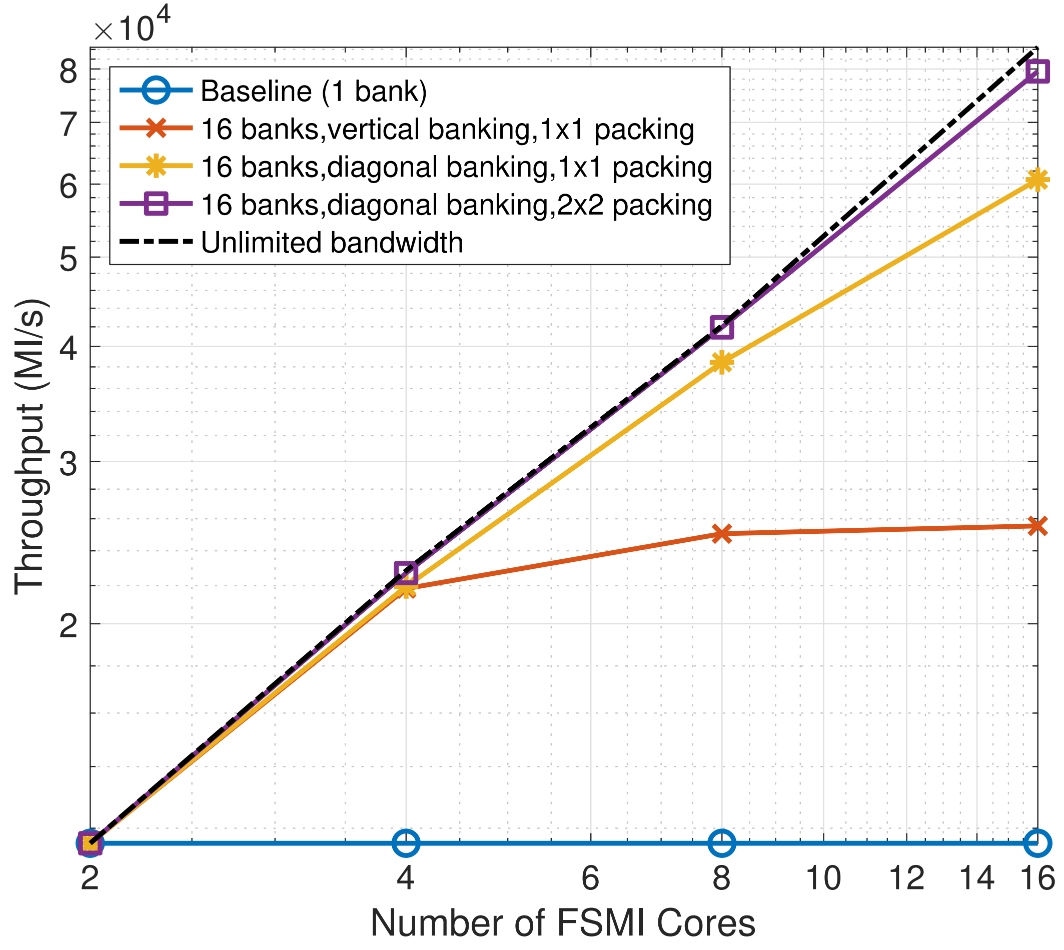


Compute the mutual information for an **entire map** of 20m x 20m at 0.1m resolution **in under a second** → a 100x speed up versus CPU for 1/10th of the power

[Joint work with Sertac Karaman]

[**Li**, *RSS* 2019]

Experimental Results

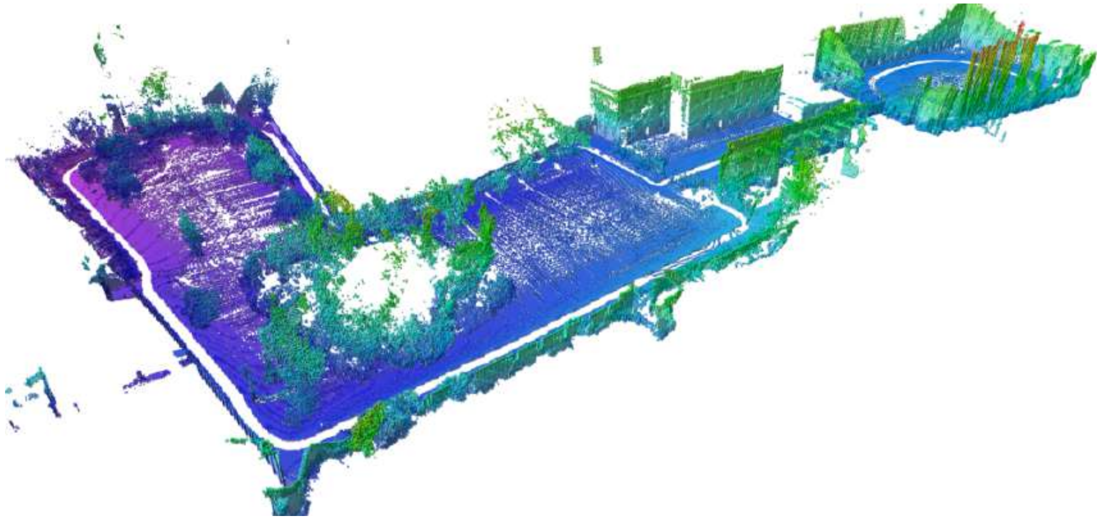


Specialized banking, efficient memory arbiter and packing multiple values at each address results in throughput **achieves 94% of theoretical limit** (unlimited bandwidth)

[**Li**, *RSS* 2019]

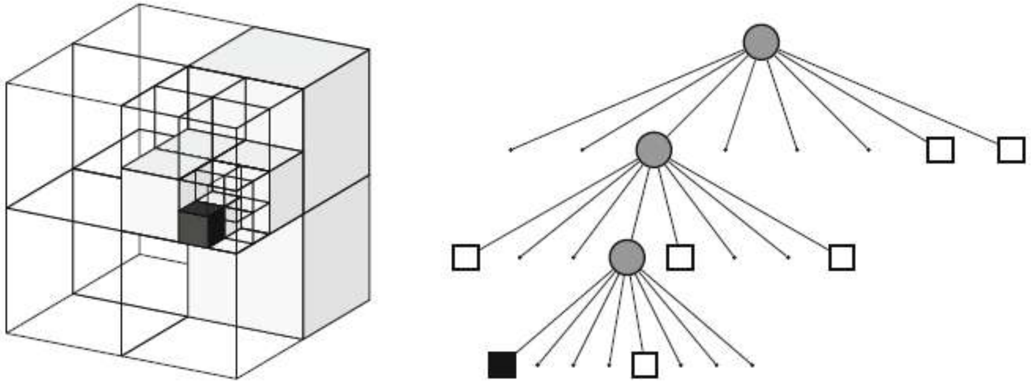
Extend FSMI to 3D Environments

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

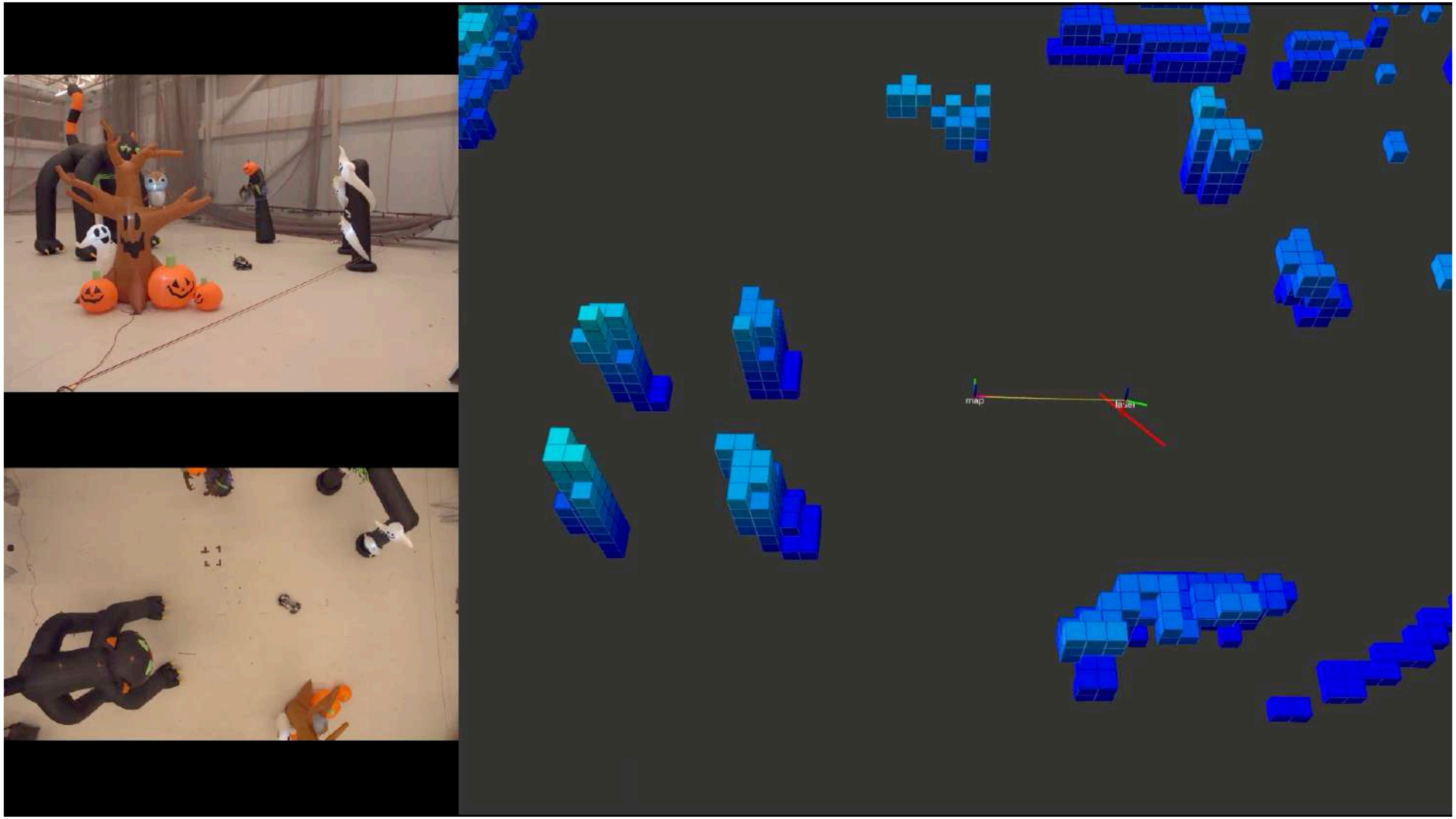


Compress map with OctoMap

[Hornung, et al., Autonomous Robots, 2013]

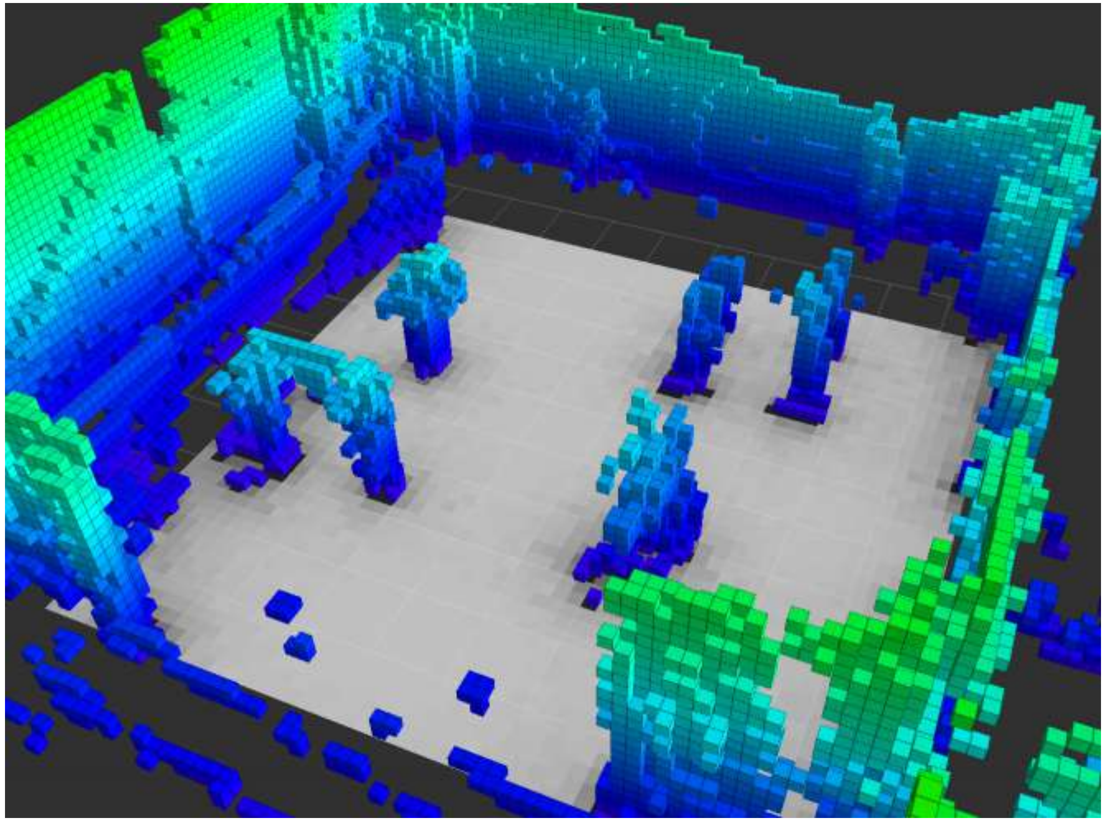


Experiments of 3D FSMI (4x Real Time)



[**Zhang**, *IJRR* 2020]

Experiments of 3D FSMI



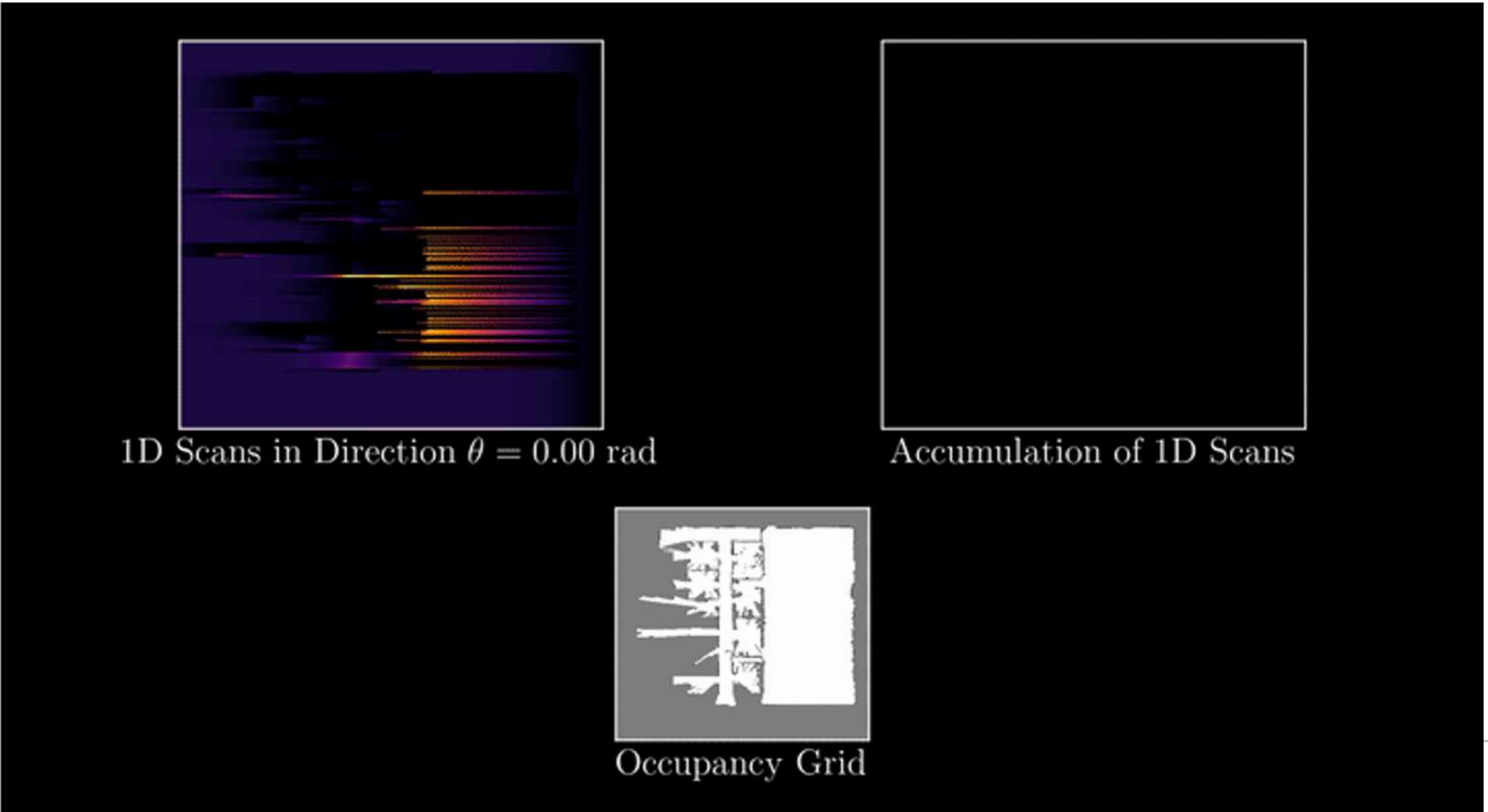
We achieve an average compression ratio of around 18×, with an acceleration ratio of 8×

[**Zhang**, *IJRR* 2020]

FCMI: Fast Continuous Mutual Information

Reformulate with a ***continuous*** occupancy map framework and exploit recursive structure when computing MI across ***entire*** map

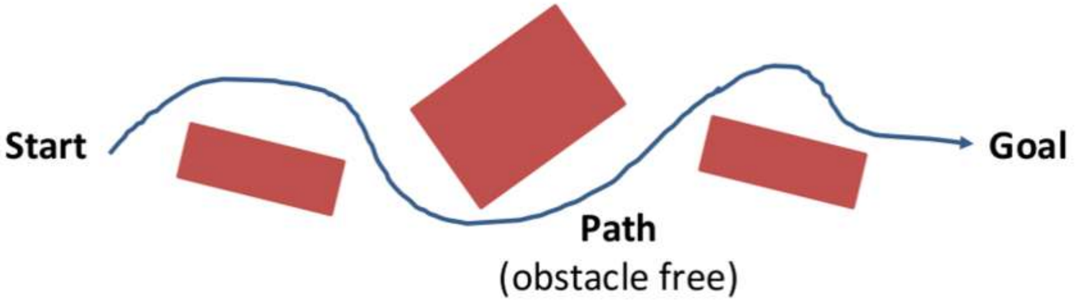
→ ***two orders of magnitude speed up over FSMI!***



[**Henderson**, *ICRA* 2020]

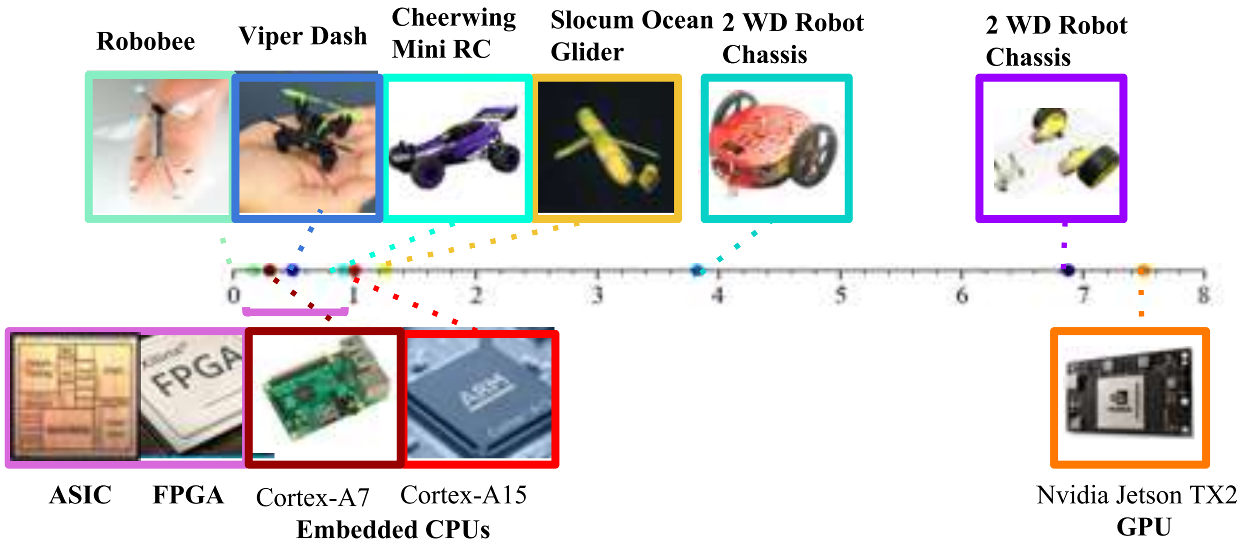
Balancing Actuation and Computing Energy

Motion Planning
Find a feasible (obstacle-free) path
[typically optimize for shortest path]



Low-power Robotics
Actuation and computing energy are similar order of magnitude

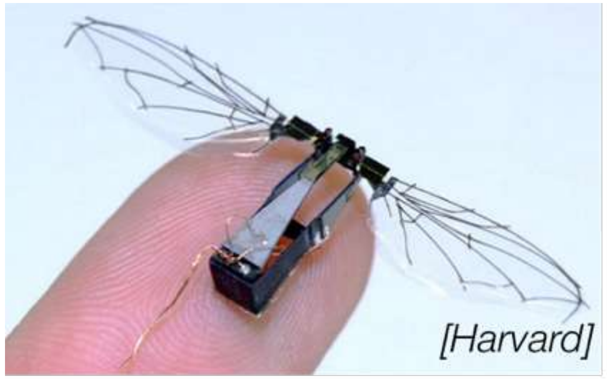
Energy to move 1 more meter (P_a/v [W/(m/s)])



Energy to compute 1 more second (P_c [W])

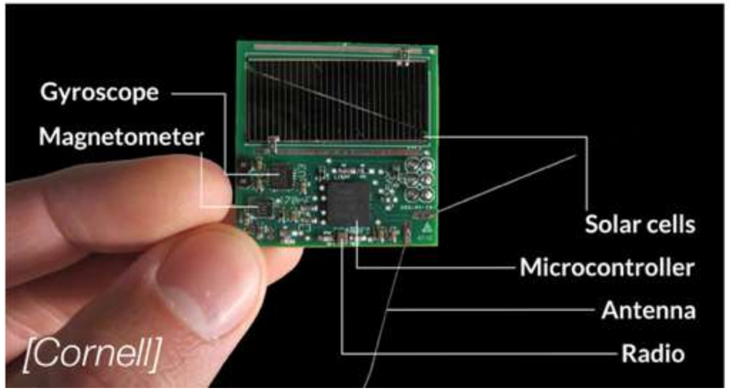
[**Sudhakar**, *ICRA* 2020]

Robots Consuming < 1 Watt for Actuation

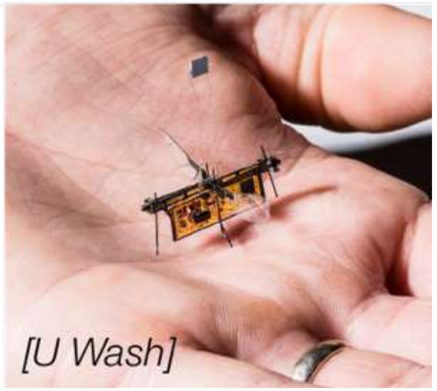


[Harvard]

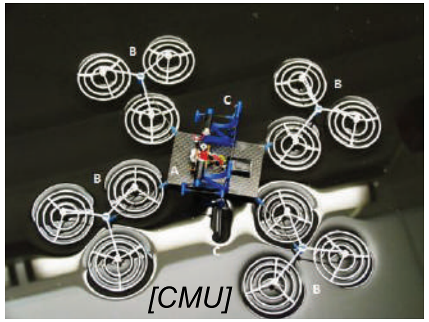
[GaTech]



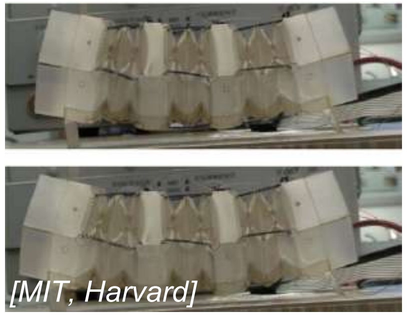
[Cornell]



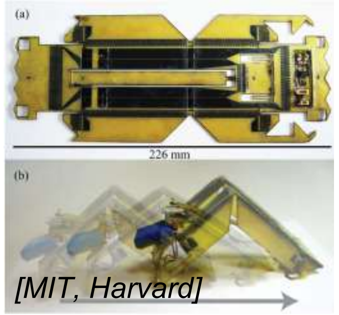
[U Wash]



[CMU]



[MIT, Harvard]



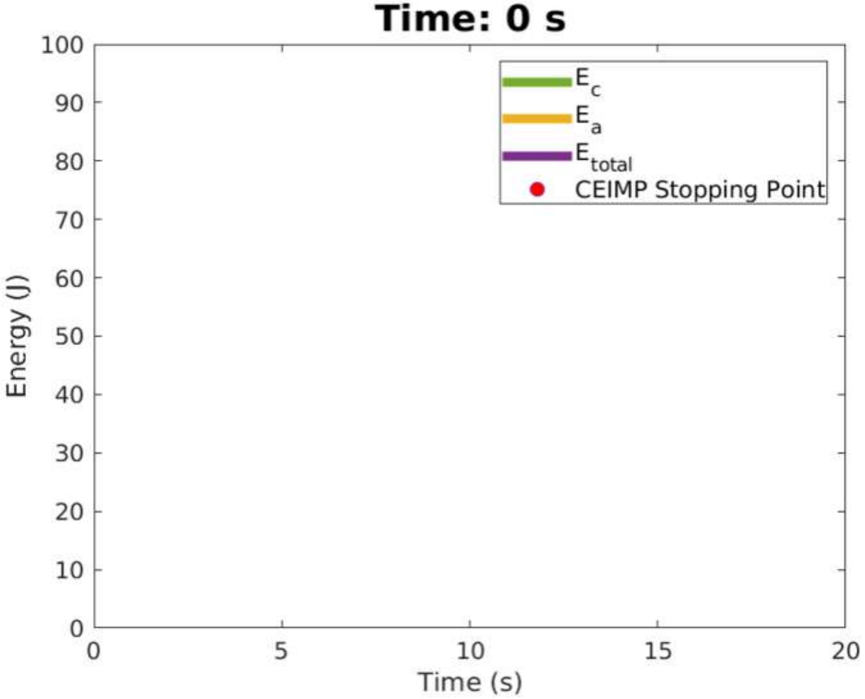
[MIT, Harvard]

Low Energy Robotics

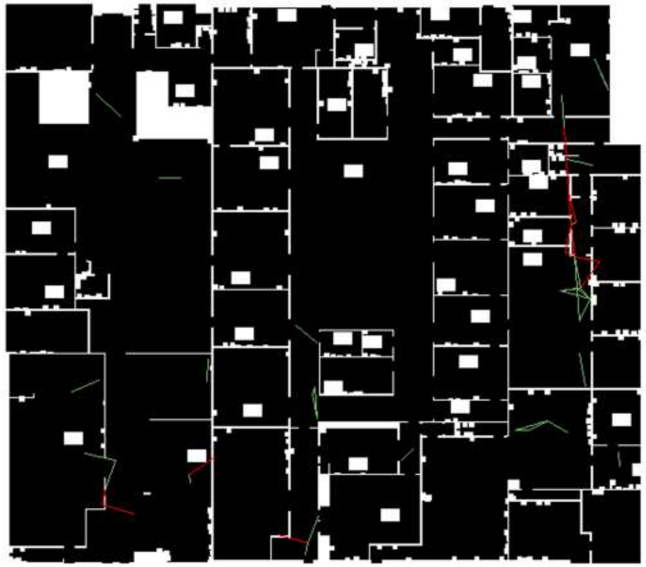
- Miniature aerial vehicles
- Lighter than air vehicles
- Micro unmanned gliders
- Miniature satellites

Balancing Actuation and Computing Energy

Baseline
(compute 20,000 samples)



CEIMP

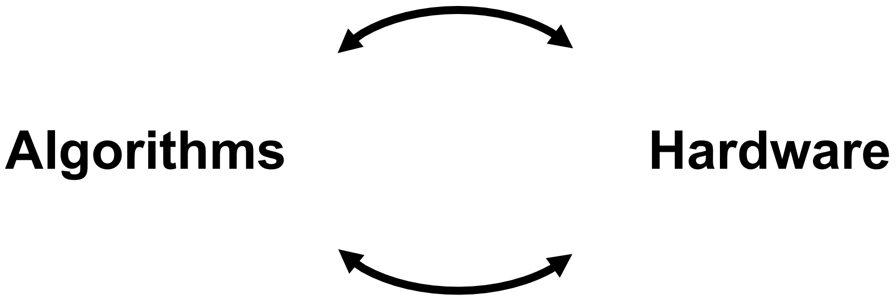


Compute Energy Included Motion Planning (CEIMP)
A framework to balance the energy spent on **computing** a path and the energy spent on **moving** along that path **(Don't think too hard!)**

[**Sudhakar**, *ICRA* 2020]

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

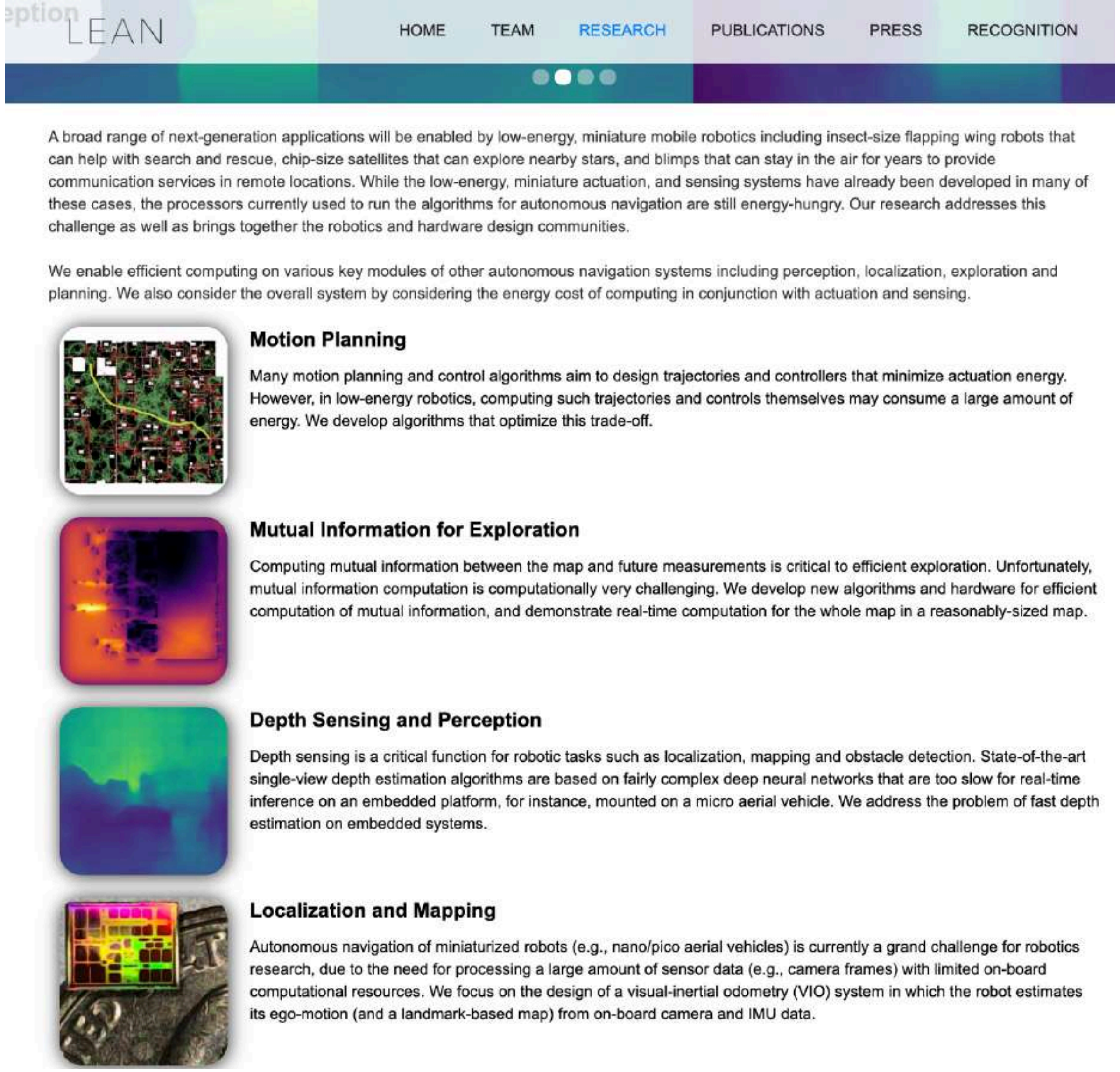
Sertac Karaman

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

AFOSR AIR FORCE OFFICE OF SCIENTIFIC RESEARCH NSF ANALOG DEVICES BROADCOM Google intel IBM

DARPA SRC 3M NVIDIA QUALCOMM SAMSUNG TEXAS INSTRUMENTS tsmc

Low-Energy Autonomy and Navigation (LEAN) Group



A broad range of next-generation applications will be enabled by low-energy, miniature mobile robotics including insect-size flapping wing robots that can help with search and rescue, chip-size satellites that can explore nearby stars, and blimps that can stay in the air for years to provide communication services in remote locations. While the low-energy, miniature actuation, and sensing systems have already been developed in many of these cases, the processors currently used to run the algorithms for autonomous navigation are still energy-hungry. Our research addresses this challenge as well as brings together the robotics and hardware design communities.

We enable efficient computing on various key modules of other autonomous navigation systems including perception, localization, exploration and planning. We also consider the overall system by considering the energy cost of computing in conjunction with actuation and sensing.

Motion Planning

Many motion planning and control algorithms aim to design trajectories and controllers that minimize actuation energy. However, in low-energy robotics, computing such trajectories and controls themselves may consume a large amount of energy. We develop algorithms that optimize this trade-off.

Mutual Information for Exploration

Computing mutual information between the map and future measurements is critical to efficient exploration. Unfortunately, mutual information computation is computationally very challenging. We develop new algorithms and hardware for efficient computation of mutual information, and demonstrate real-time computation for the whole map in a reasonably-sized map.

Depth Sensing and Perception

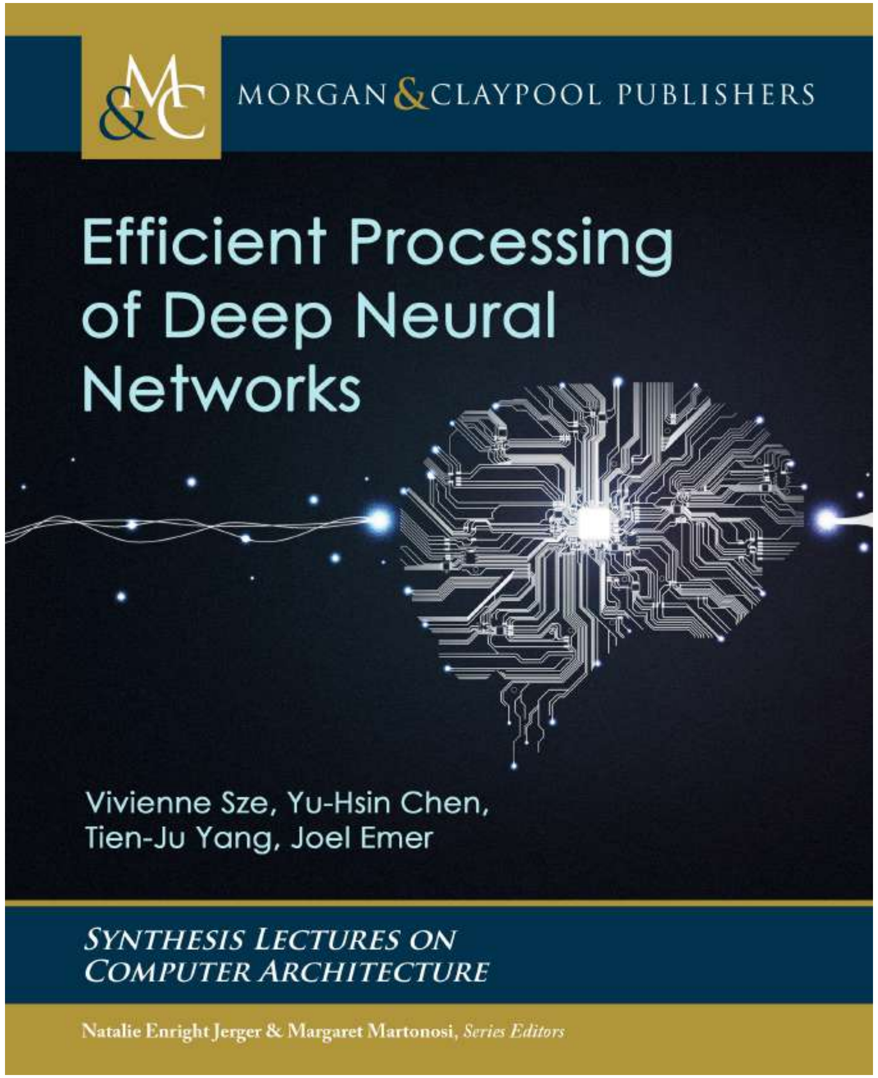
Depth sensing is a critical function for robotic tasks such as localization, mapping and obstacle detection. State-of-the-art single-view depth estimation algorithms are based on fairly complex deep neural networks that are too slow for real-time inference on an embedded platform, for instance, mounted on a micro aerial vehicle. We address the problem of fast depth estimation on embedded systems.

Localization and Mapping

Autonomous navigation of miniaturized robots (e.g., nano/pico aerial vehicles) is currently a grand challenge for robotics research, due to the need for processing a large amount of sensor data (e.g., camera frames) with limited on-board computational resources. We focus on the design of a visual-inertial odometry (VIO) system in which the robot estimates its ego-motion (and a landmark-based map) from on-board camera and IMU data.

Group Website: http://lean.mit.edu

Book on Efficient Processing of DNNs


M&C
MORGAN & CLAYPOOL PUBLISHERS
Efficient Processing of Deep Neural Networks
Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, Joel Emer
SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE
Natalie Enright Jerger & Margaret Martonosi, Series Editors

Part I Understanding Deep Neural Networks

Introduction

Overview of Deep Neural Networks

Part II Design of Hardware for Processing DNNs

Key Metrics and Design Objectives

Kernel Computation

Designing DNN Accelerators

Operation Mapping on Specialized Hardware

Part III Co-Design of DNN Hardware and Algorithms

Reducing Precision

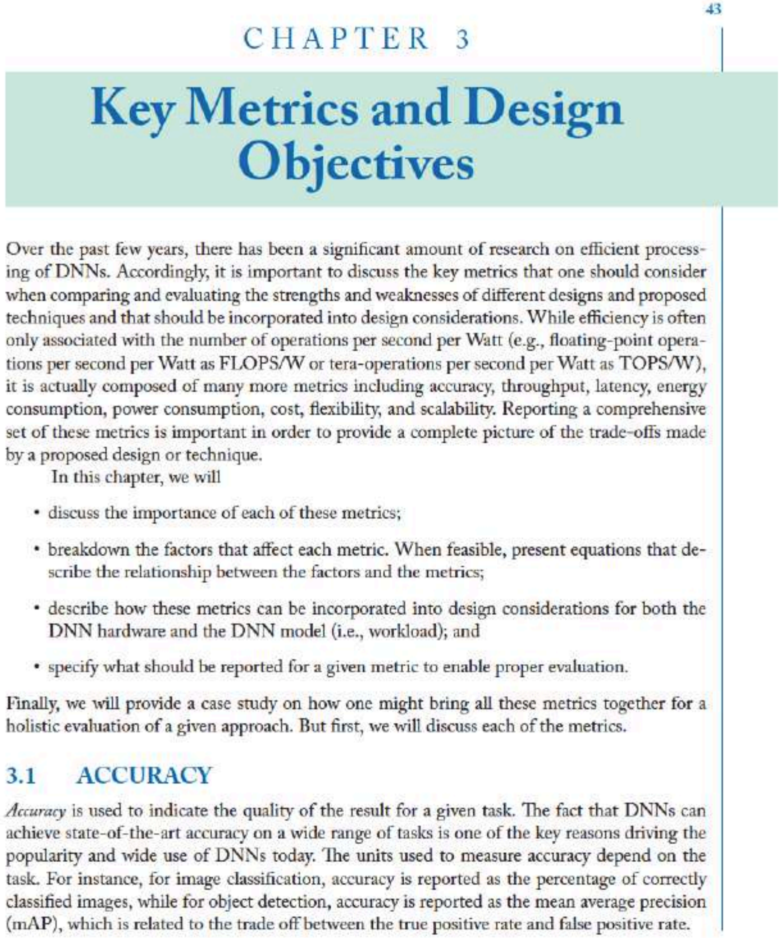
Exploiting Sparsity

Designing Efficient DNN Models

Advanced Technologies

https://tinyurl.com/EfficientDNNBook

Excerpts of Book



CHAPTER 3

Key Metrics and Design Objectives

Over the past few years, there has been a significant amount of research on efficient processing of DNNs. Accordingly, it is important to discuss the key metrics that one should consider when comparing and evaluating the strengths and weaknesses of different designs and proposed techniques and that should be incorporated into design considerations. While efficiency is often only associated with the number of operations per second per Watt (e.g., floating-point operations per second per Watt as FLOPS/W or tera-operations per second per Watt as TOPS/W), it is actually composed of many more metrics including accuracy, throughput, latency, energy consumption, power consumption, cost, flexibility, and scalability. Reporting a comprehensive set of these metrics is important in order to provide a complete picture of the trade-offs made by a proposed design or technique.

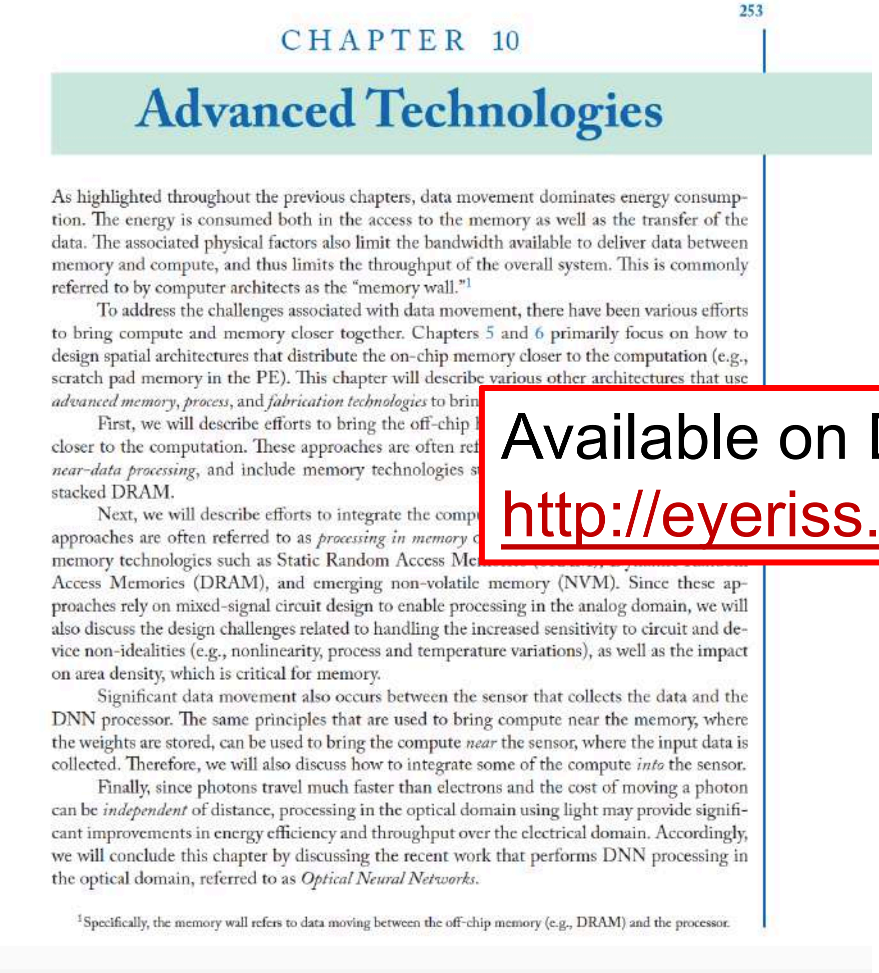
In this chapter, we will

- discuss the importance of each of these metrics;
- breakdown the factors that affect each metric. When feasible, present equations that describe the relationship between the factors and the metrics;
- describe how these metrics can be incorporated into design considerations for both the DNN hardware and the DNN model (i.e., workload); and
- specify what should be reported for a given metric to enable proper evaluation.

Finally, we will provide a case study on how one might bring all these metrics together for a holistic evaluation of a given approach. But first, we will discuss each of the metrics.

3.1 ACCURACY

Accuracy is used to indicate the quality of the result for a given task. The fact that DNNs can achieve state-of-the-art accuracy on a wide range of tasks is one of the key reasons driving the popularity and wide use of DNNs today. The units used to measure accuracy depend on the task. For instance, for image classification, accuracy is reported as the percentage of correctly classified images, while for object detection, accuracy is reported as the mean average precision (mAP), which is related to the trade off between the true positive rate and false positive rate.



CHAPTER 10

Advanced Technologies

As highlighted throughout the previous chapters, data movement dominates energy consumption. The energy is consumed both in the access to the memory as well as the transfer of the data. The associated physical factors also limit the bandwidth available to deliver data between memory and compute, and thus limits the throughput of the overall system. This is commonly referred to by computer architects as the "memory wall."[1]

To address the challenges associated with data movement, there have been various efforts to bring compute and memory closer together. Chapters 5 and 6 primarily focus on how to design spatial architectures that distribute the on-chip memory closer to the computation (e.g., scratch pad memory in the PE). This chapter will describe various other architectures that use *advanced memory*, *process*, and *fabrication technologies* to brin

First, we will describe efforts to bring the off-chip closer to the computation. These approaches are often ref *near-data processing*, and include memory technologies s stacked DRAM.

Next, we will describe efforts to integrate the comp approaches are often referred to as *processing in memory* c memory technologies such as Static Random Access Me Access Memories (DRAM), and emerging non-volatile memory (NVM). Since these approaches rely on mixed-signal circuit design to enable processing in the analog domain, we will also discuss the design challenges related to handling the increased sensitivity to circuit and device non-idealities (e.g., nonlinearity, process and temperature variations), as well as the impact on area density, which is critical for memory.

Significant data movement also occurs between the sensor that collects the data and the DNN processor. The same principles that are used to bring compute near the memory, where the weights are stored, can be used to bring the compute *near* the sensor, where the input data is collected. Therefore, we will also discuss how to integrate some of the compute *into* the sensor.

Finally, since photons travel much faster than electrons and the cost of moving a photon can be *independent* of distance, processing in the optical domain using light may provide significant improvements in energy efficiency and throughput over the electrical domain. Accordingly, we will conclude this chapter by discussing the recent work that performs DNN processing in the optical domain, referred to as *Optical Neural Networks*.

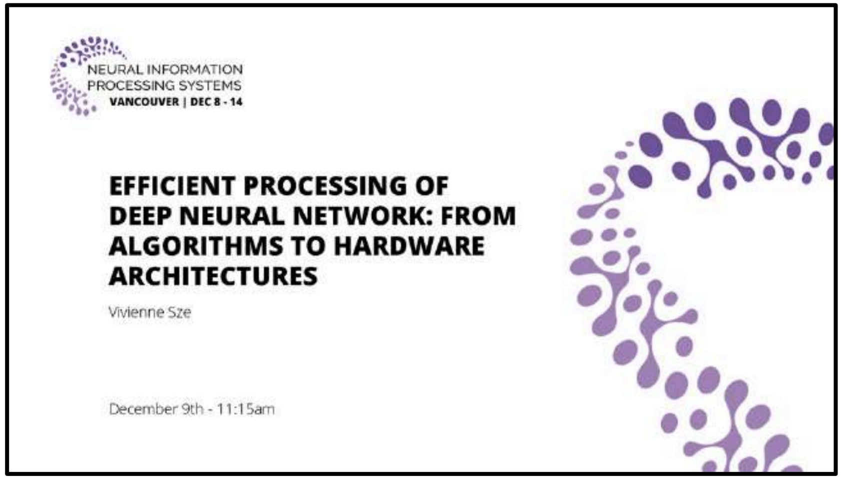
[1] Specifically, the memory wall refers to data moving between the off-chip memory (e.g., DRAM) and the processor.

Available on DNN tutorial website
http://eyeriss.mit.edu/tutorial.html

Additional Resources

Talks and Tutorial Available Online

https://www.rle.mit.edu/eems/publications/tutorials/



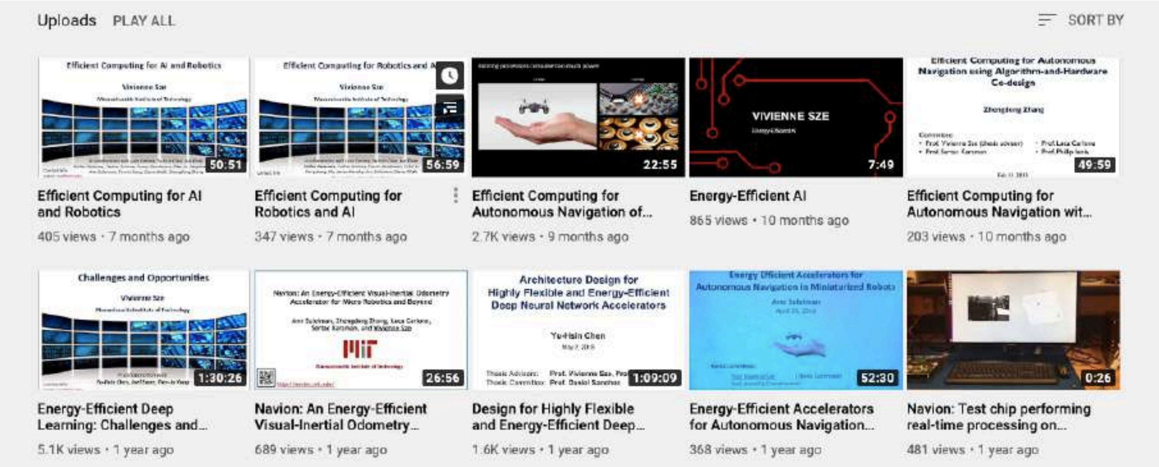
NEURAL INFORMATION PROCESSING SYSTEMS
VANCOUVER | DEC 8 - 14

EFFICIENT PROCESSING OF DEEP NEURAL NETWORK: FROM ALGORITHMS TO HARDWARE ARCHITECTURES

Vivienne Sze

December 9th - 11:15am

YouTube Channel
EEMS Group – PI: Vivienne Sze



Uploads PLAY ALL SORT BY

50:51 Efficient Computing for AI and Robotics
405 views · 7 months ago

56:59 Efficient Computing for Robotics and AI
347 views · 7 months ago

22:55 Efficient Computing for Autonomous Navigation of...
2.7K views · 9 months ago

7:49 Energy-Efficient AI
865 views · 10 months ago

49:59 Efficient Computing for Autonomous Navigation wit...
203 views · 10 months ago

1:30:26 Energy-Efficient Deep Learning: Challenges and...
5.1K views · 1 year ago

26:56 Navion: An Energy-Efficient Visual-Inertial Odometry...
689 views · 1 year ago

1:09:09 Design for Highly Flexible and Energy-Efficient Deep...
1.6K views · 1 year ago

52:30 Energy-Efficient Accelerators for Autonomous Navigation...
368 views · 1 year ago

0:26 Navion: Test chip performing real-time processing on...
481 views · 1 year ago

References

- **Efficient Processing for Deep Neural Networks**
  - **Project website:** http://eyeriss.mit.edu
  - 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), Vol. 9, No. 2, pp. 292-308, June 2019.
  - 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, "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.
  - Y. N. Wu, J. S. Emer, V. Sze, "Accelergy: An Architecture-Level Energy Estimation Methodology for Accelerator Designs," International Conference on Computer Aided Design (ICCAD), November 2019. http://accelergy.mit.edu/
  - Y. N. Wu, V. Sze, J. S. Emer, "An Architecture-Level Energy and Area Estimator for Processing-In-Memory Accelerator Designs," to appear in IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), April 2020.
  - A. Suleiman*, Y.-H. Chen*, J. Emer, V. Sze, "Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision," IEEE International Symposium of Circuits and Systems (ISCAS), Invited Paper, May 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. http://netadapt.mit.edu
  - 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/
  - T.-J. Yang, V. Sze, “Design Considerations for Efficient Deep Neural Networks on Processing-in-Memory Accelerators,” IEEE International Electron Devices Meeting (IEDM), Invited Paper, December 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 Map Estimation of Dynamic Scenes Using Prior Depth Information,” arXiv, February 2020. https://arxiv.org/abs/2002.00297
  - 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.

References

- **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**
  - **Project website:** http://lean.mit.edu
  - 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," to appear in International Journal of Robotics Research (IJRR). http://arxiv.org/abs/1905.02238
  - T. Henderson, V. Sze, S. Karaman, "An Efficient and Continuous Approach to Information-Theoretic Exploration," IEEE International Conference on Robotics and Automation (ICRA), May 2020.
- **Balancing Actuation and Computation**
  - **Project website:** http://lean.mit.edu
  - S. Sudhakar, S. Karaman, V. Sze, "Balancing Actuation and Computing Energy in Motion Planning," IEEE International Conference on Robotics and Automation (ICRA), May 2020