Efficient Computing for Autonomy and Navigation

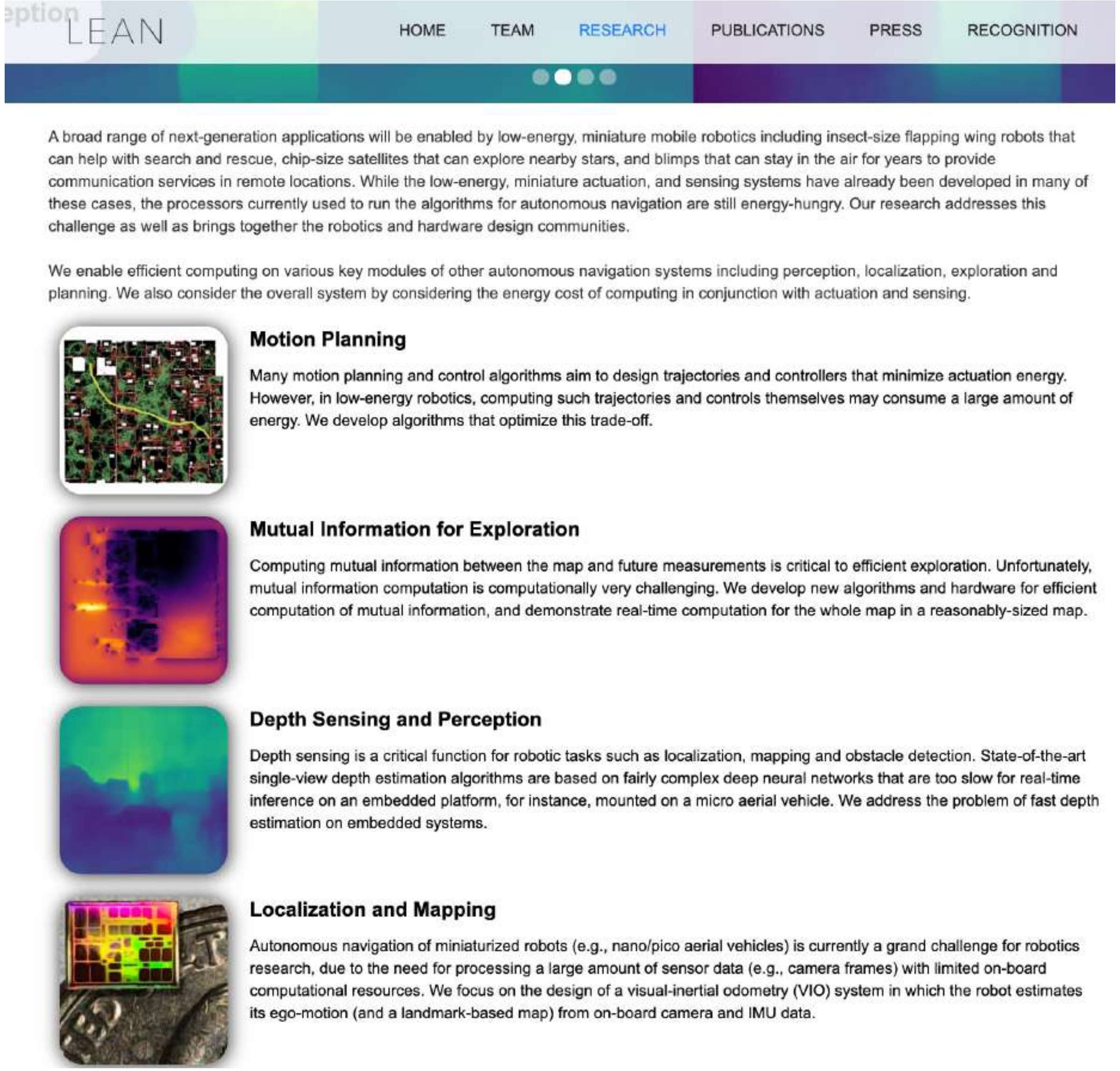
Vivienne Sze (@eems_mit)

Massachusetts Institute of Technology

In collaboration with Luca Carlone, Yu-Hsin Chen, Joel Emer, Keshav Gupta, Sertac Karaman, Tushar Krishna, Theia Henderson, Peter Li, Yi-Lun Liao, Fangchang Ma, James Noraky, Soumya Sudhakar, Amr Suleiman, Diana Wofk, Tien-Ju Yang, Zhengdong Zhang

Slides available at
http://sze.mit.edu/slides

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

Computing Challenge for Self-Driving Cars

JACK STEWART TRANSPORTATION 02.06.18 08:00 AM

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

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

NIKKI KAHN/THE WASHINGTON POST/GETTY IMAGES

WIRED

(Feb 2018)

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

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

Generates wasted heat and some prototypes need water-cooling!

Robots Consuming < 1 Watt for Actuation

Mini Autonomous Blimp (2017)

SOURCE: GEORGIA TECH

500 mW

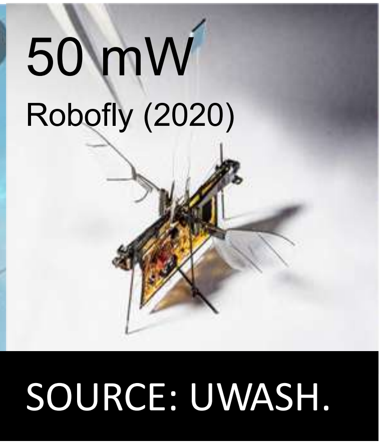
Seaglider (2003)

SOURCE: KONGSBERG

132 mW

Chipsat (2016)

SOURCE: CORNELL



50 mW

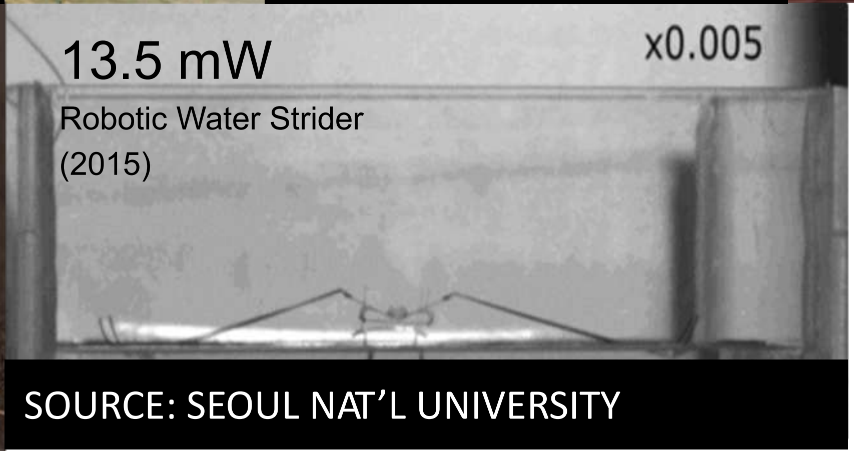
Robofly (2020)

SOURCE: UWASH.

31 mW

Robobee (2019)

SOURCE: HARVARD



13.5 mW

Robotic Water Strider (2015)

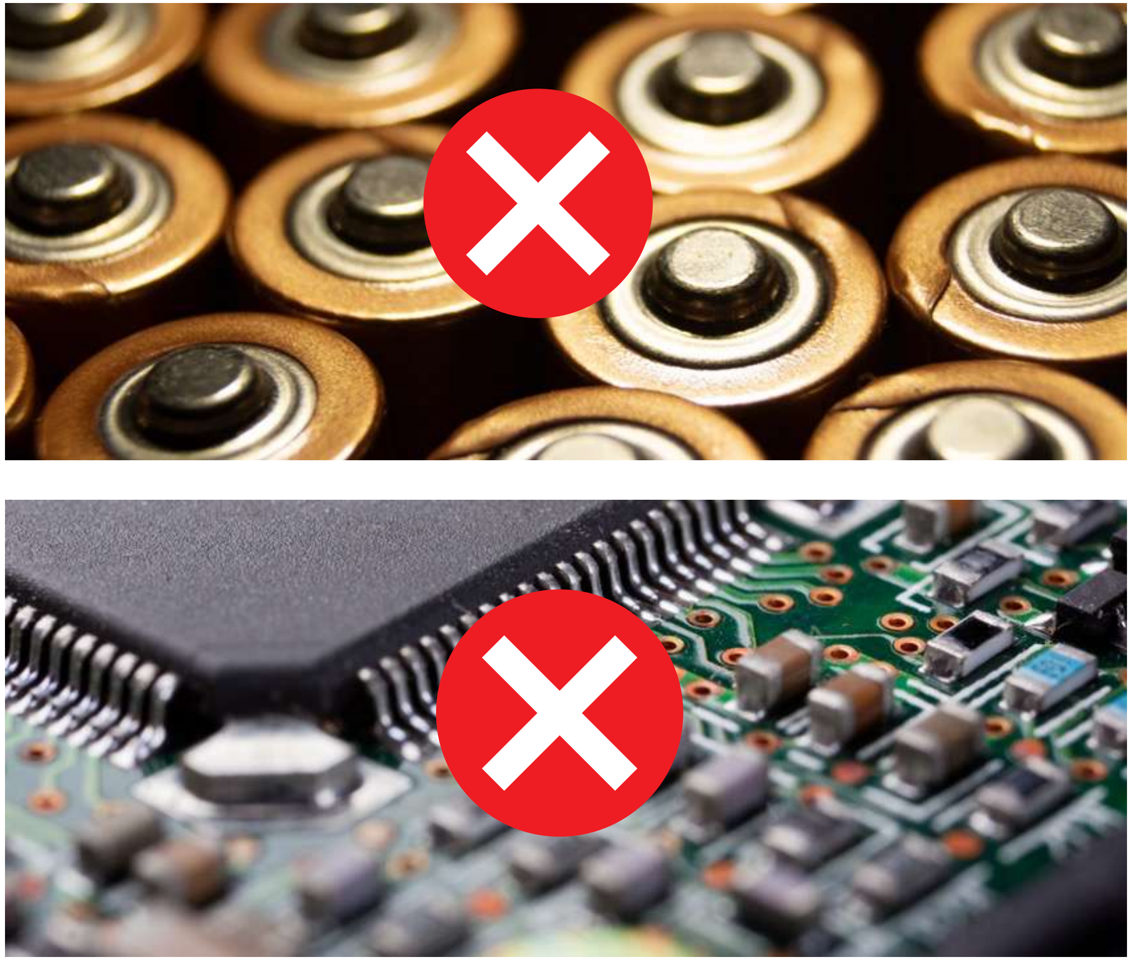
SOURCE: SEOUL NAT'L UNIVERSITY

Low Energy Robotics

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

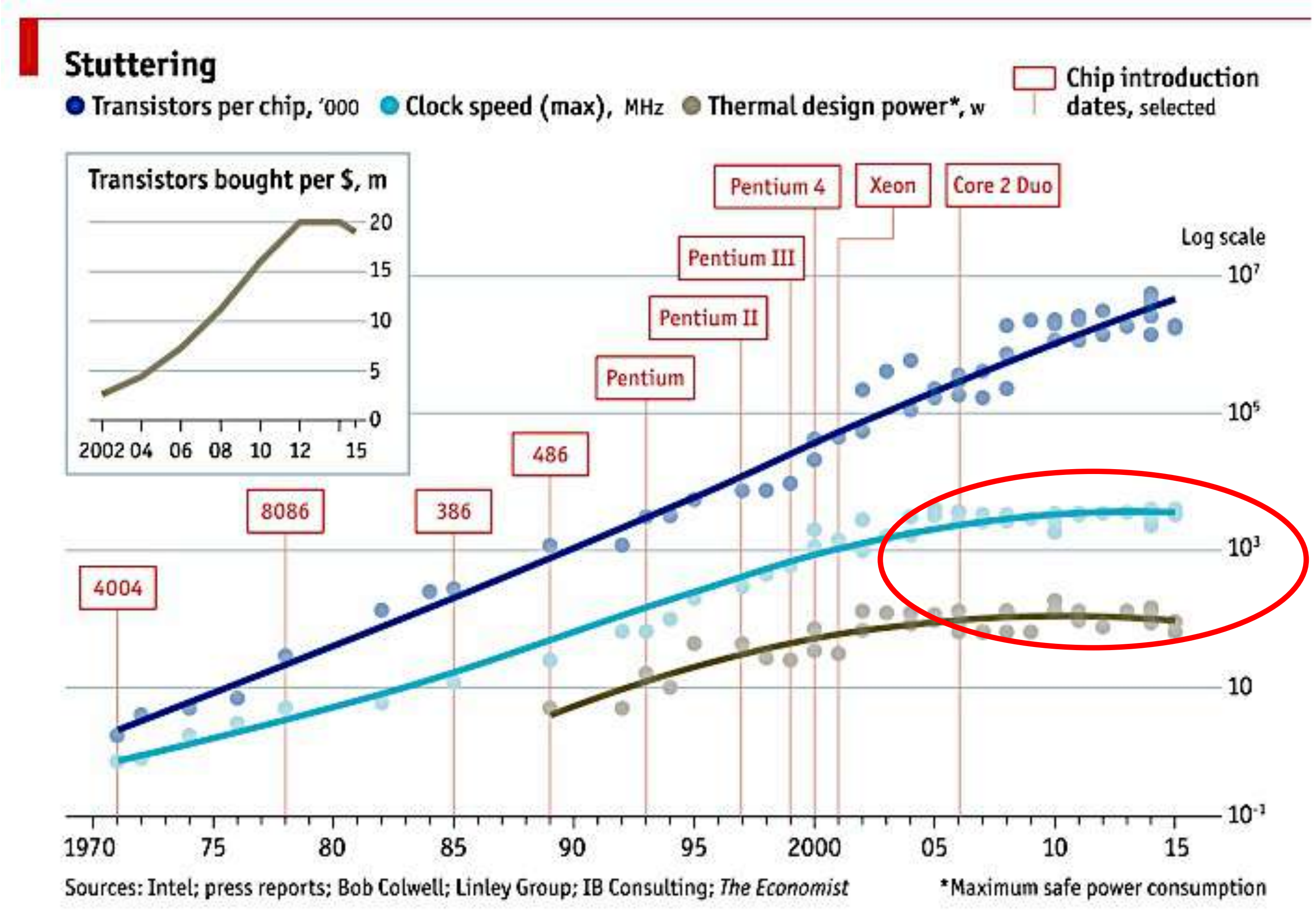
Existing Processors Consume Too Much Power

< 1 Watt



> 10 Watts

Transistors Are Not Getting More Efficient



Slowdown

Slowdown of Moore's Law and Dennard Scaling

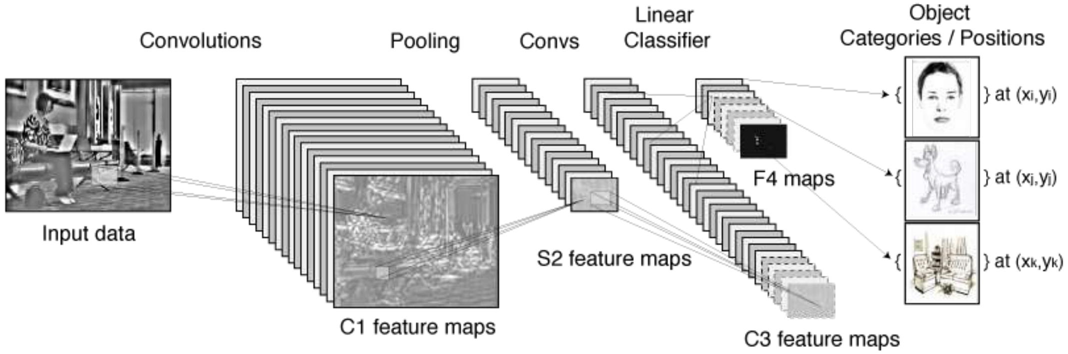
General purpose microprocessors are not getting faster or more efficient

Need **specialized hardware** for significant improvements in speed and energy efficiency

Redesign computer from the ground up!

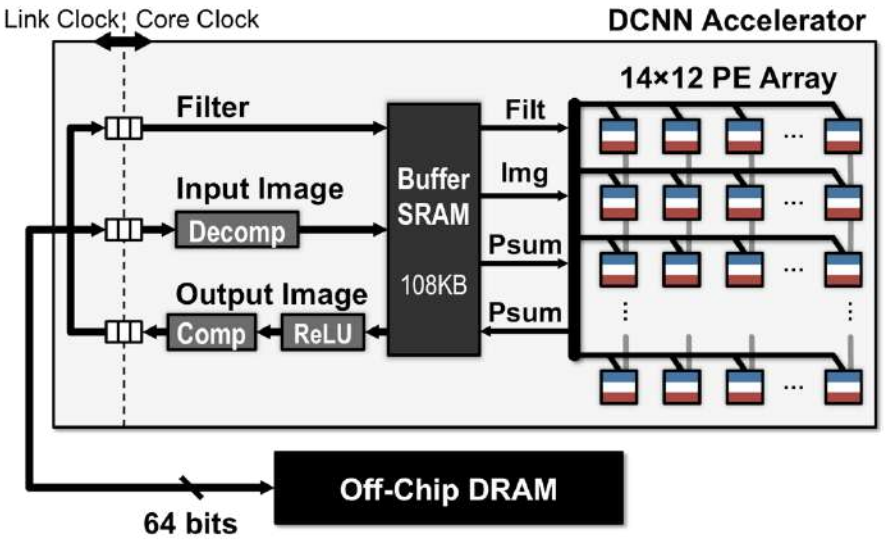
Efficient Computing with Cross-Layer Design

Algorithms

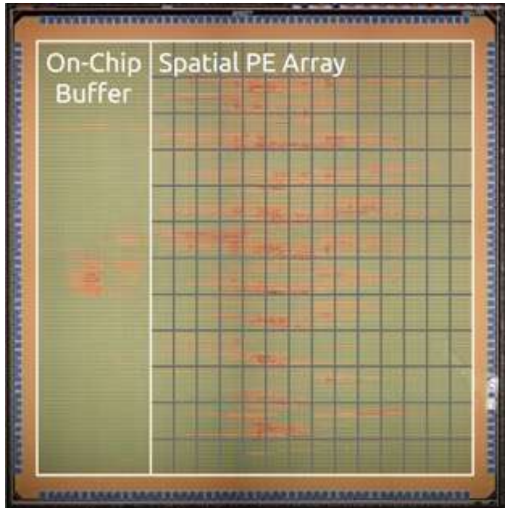


Systems

Architectures



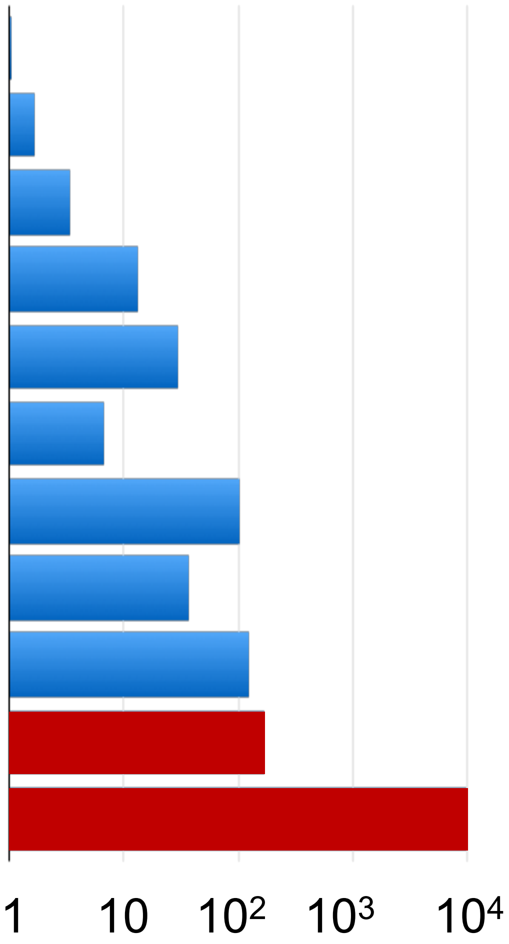
Circuits



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

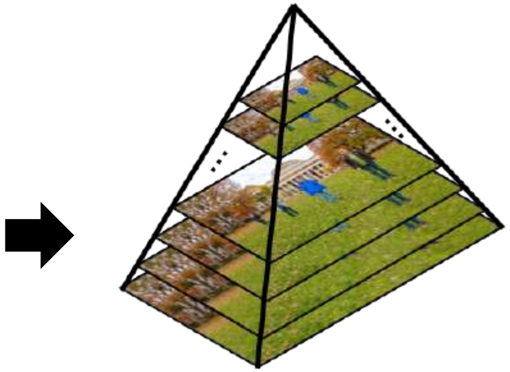
Semantic Understanding

- 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)

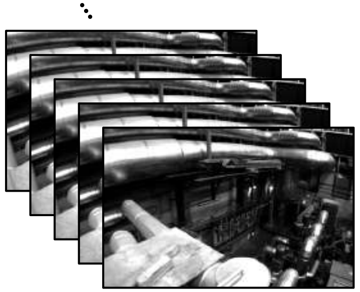
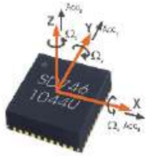


Image sequence

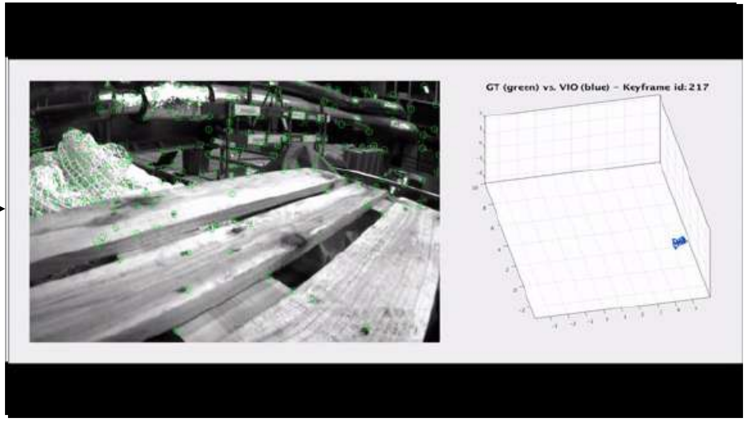
IMU

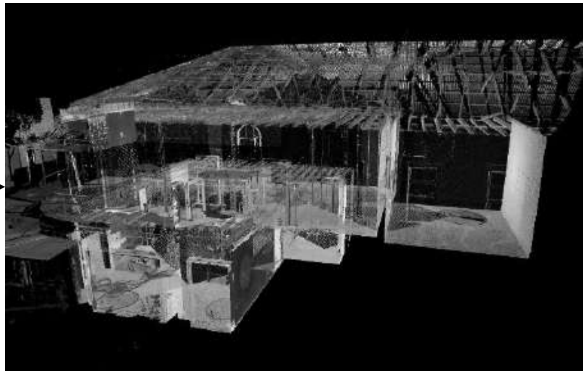
Inertial Measurement Unit



Visual-Inertial Odometry (VIO)*

Localization





Mapping

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

Localization at Under 25 mW

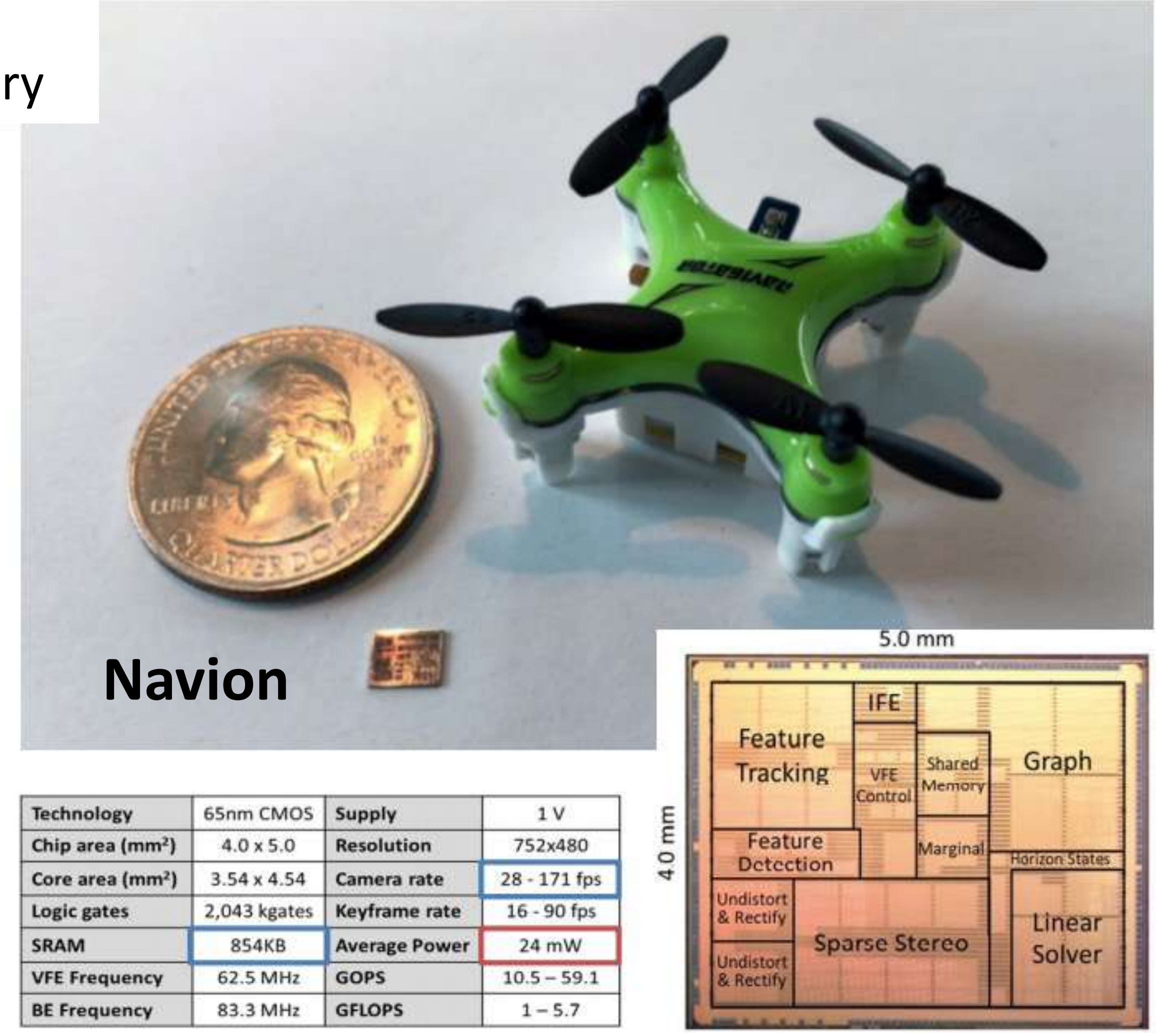
First chip that performs
complete Visual-Inertial Odometry

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

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

Back-End Optimization of Pose Graph

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



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

[**Zhang**, *RSS* 2017], [**Suleiman**, *VLSI-C* 2018]

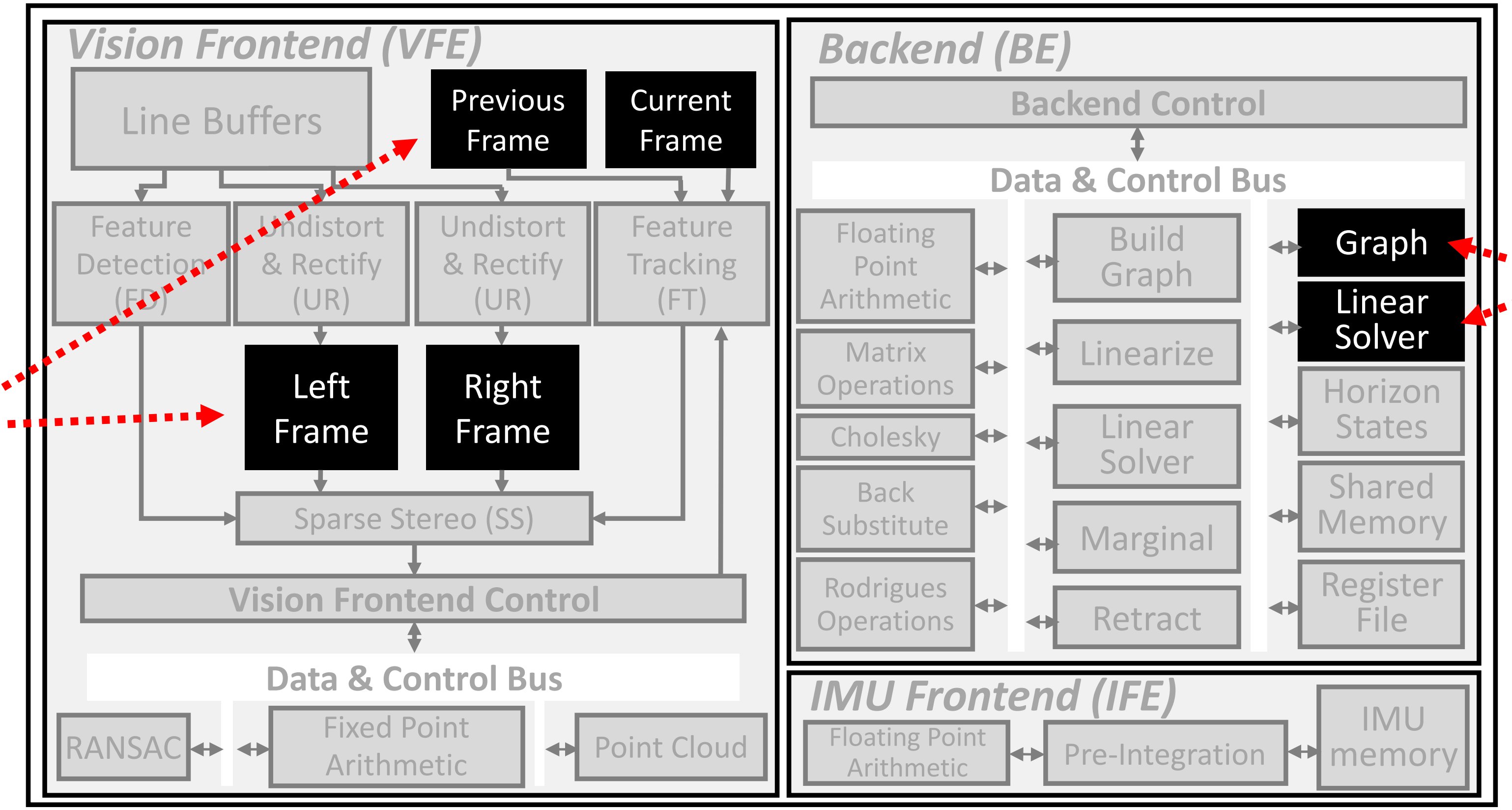
Joint work with Sertac Karaman

Key Methods to Reduce Data Size

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

http://navion.mit.edu

Apply Low Cost Frame Compression



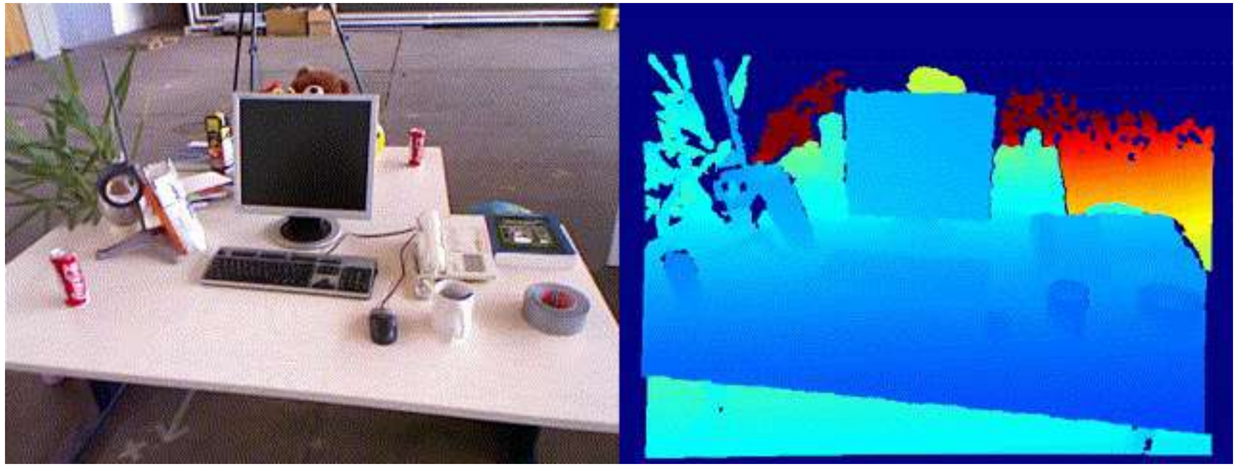
Exploit Sparsity in Graph and Linear Solver

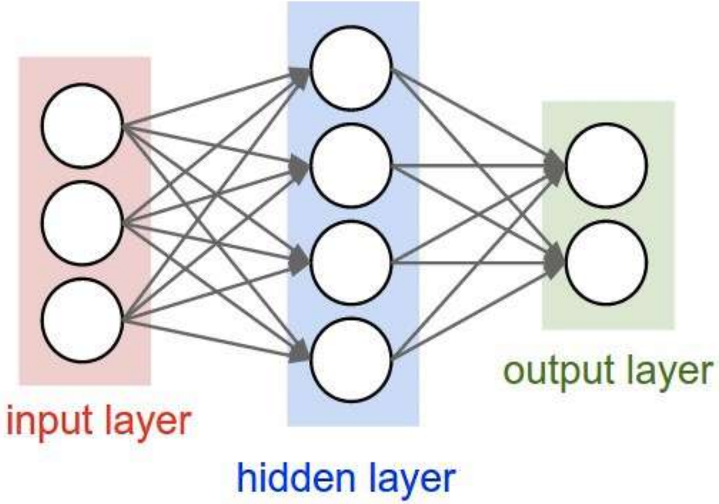
Use **compression** and **exploit sparsity** to reduce memory down to 854KB

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

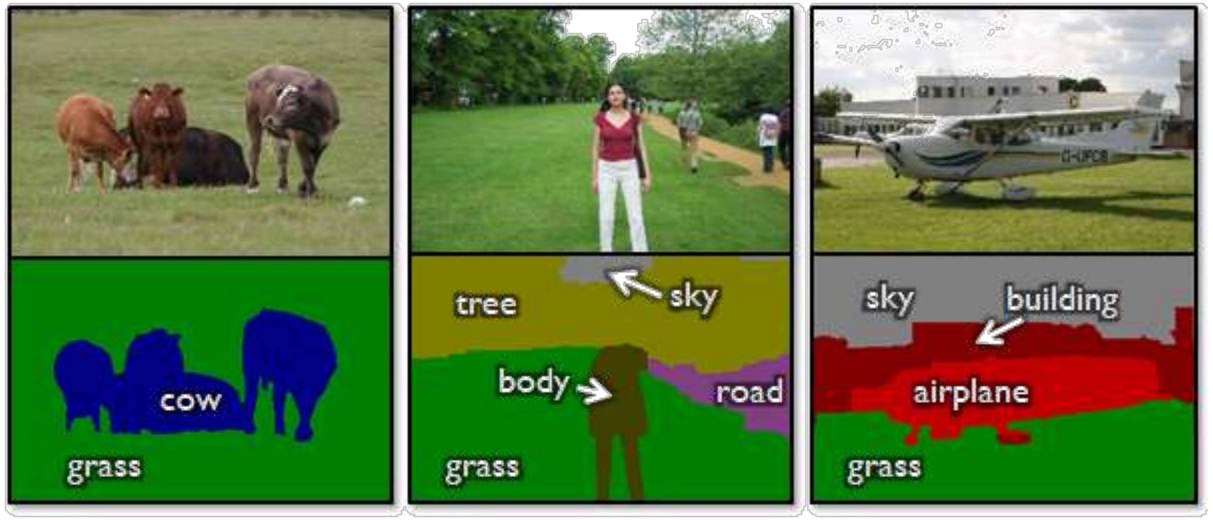
Understanding the Environment

Depth Estimation





Semantic Segmentation

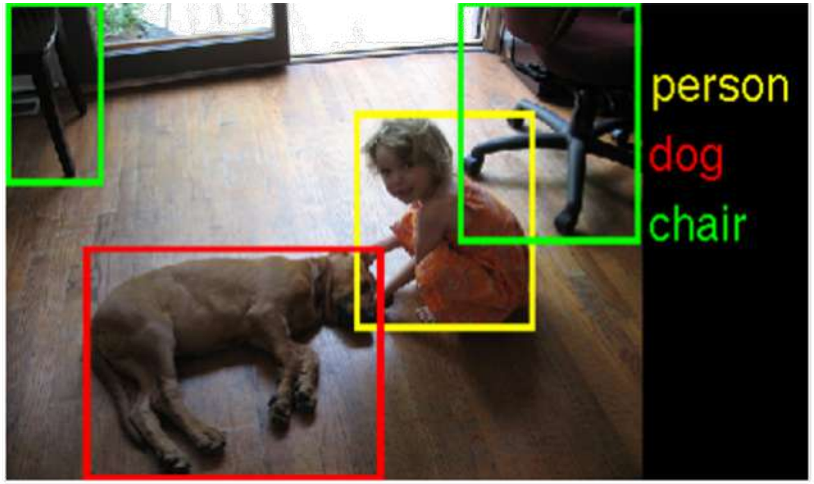


State-of-the-art approaches use **Deep Neural Networks,** which require **up to several hundred millions of operations and weights to compute!** *>100x more complex than video compression*

Deep Neural Networks

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

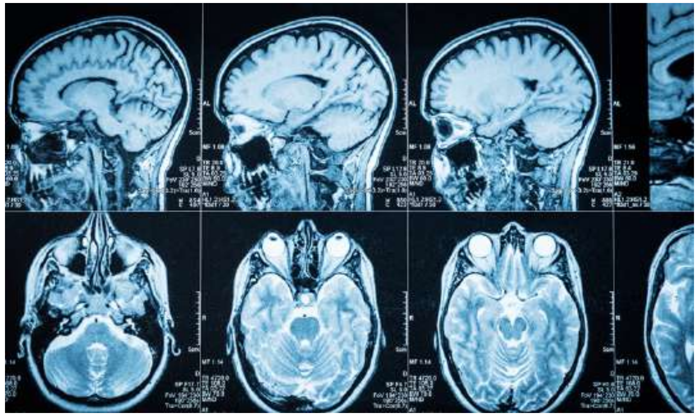
Computer Vision



Speech Recognition

Game Play

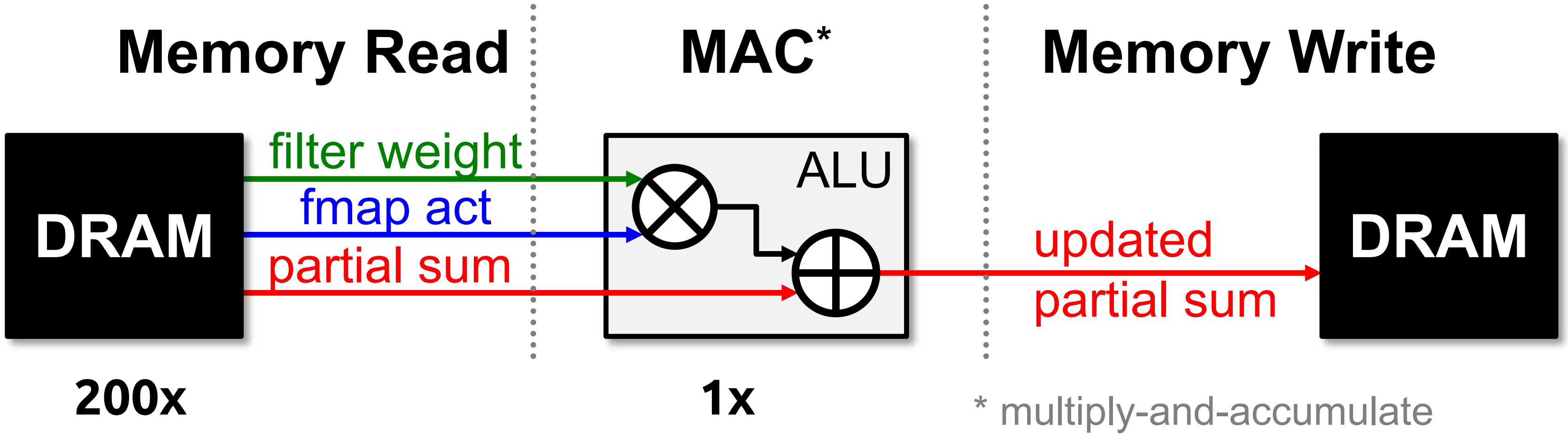
Medical



Properties We Can Leverage

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

- Memory Access is the Bottleneck



* multiply-and-accumulate

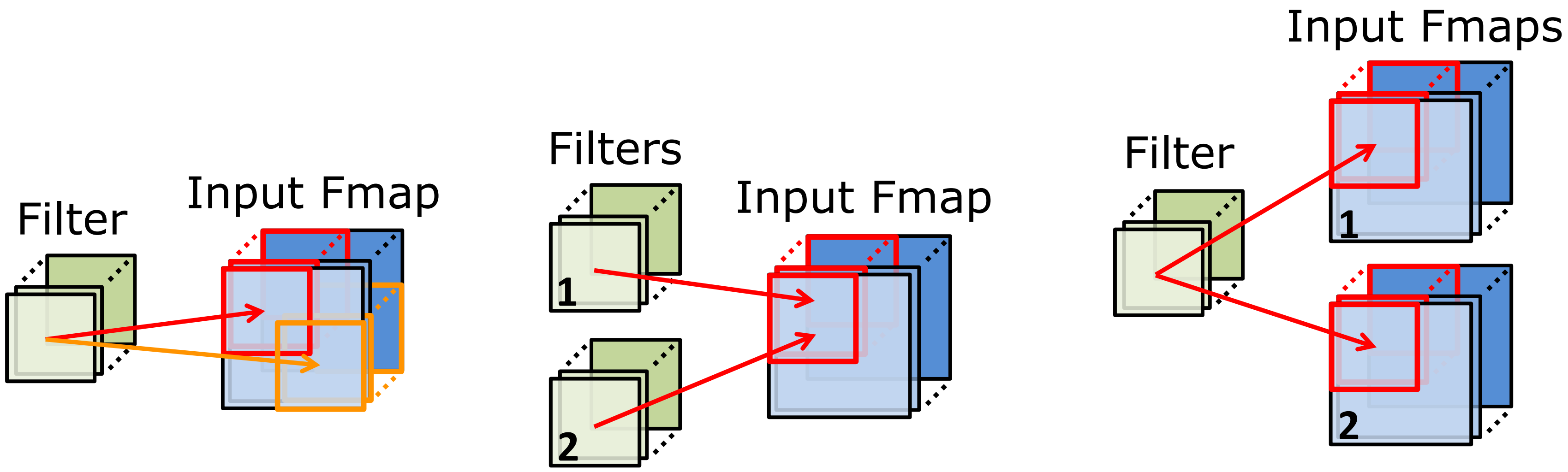
Worst Case: all memory R/W are **DRAM** accesses

- 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**)

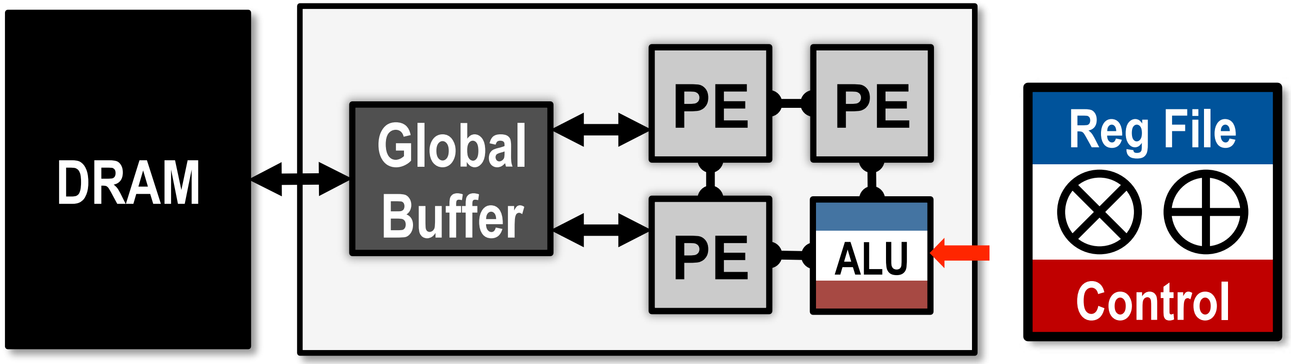


Convolutional Reuse
(Activations, Weights)
CONV layers only
(sliding window)

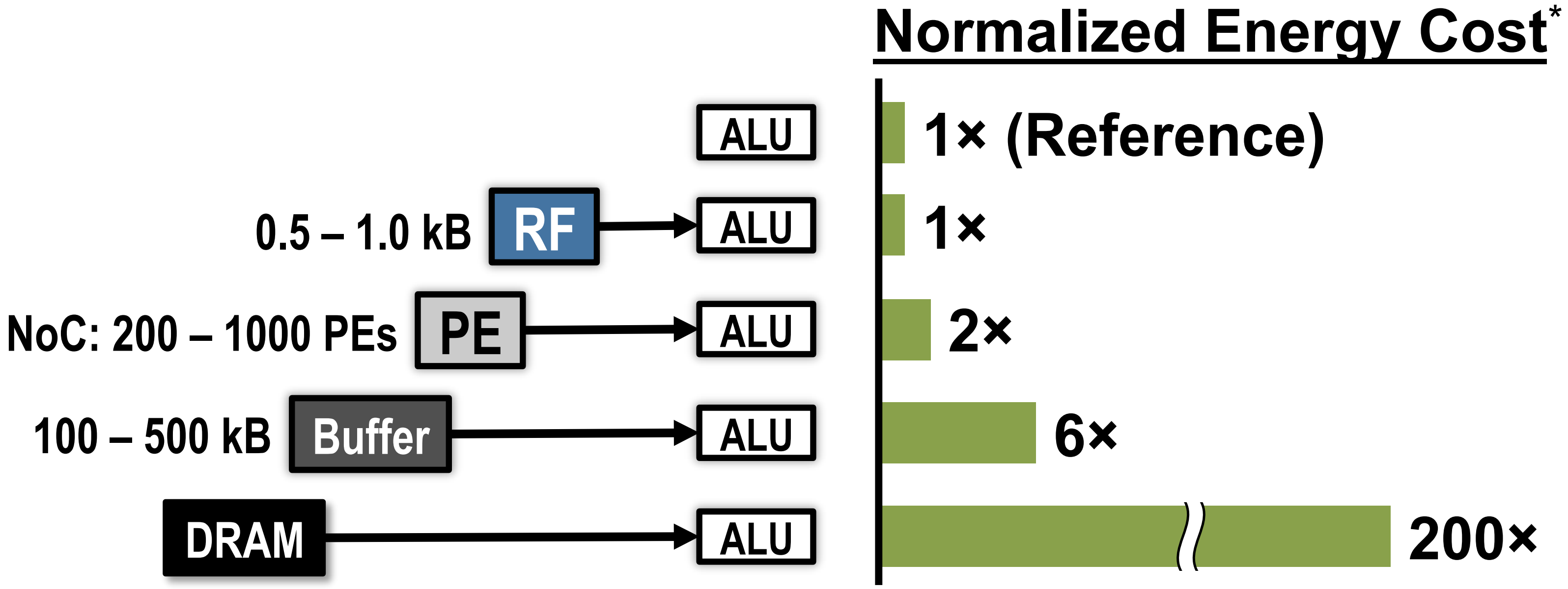
Fmap Reuse
(Activations)
CONV and FC layers

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

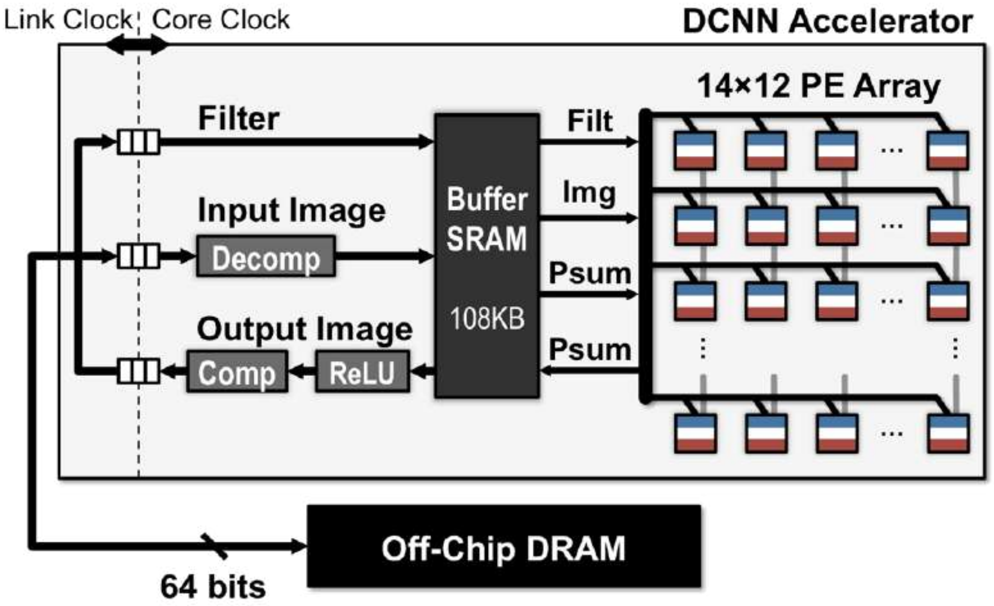


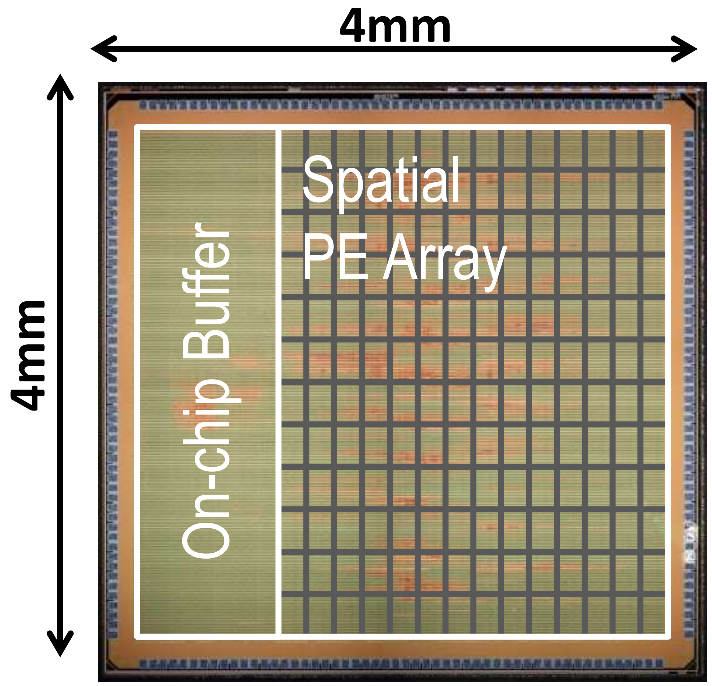
* measured from a commercial 65nm process

Farther and **larger** memories consume more power

Deep Neural Networks at Under 0.3W

Eyeriss: *Energy-Efficient Dataflow*
http://eyeriss.mit.edu





[**Chen**, *ISSCC* 2016], **Micro Top Picks**

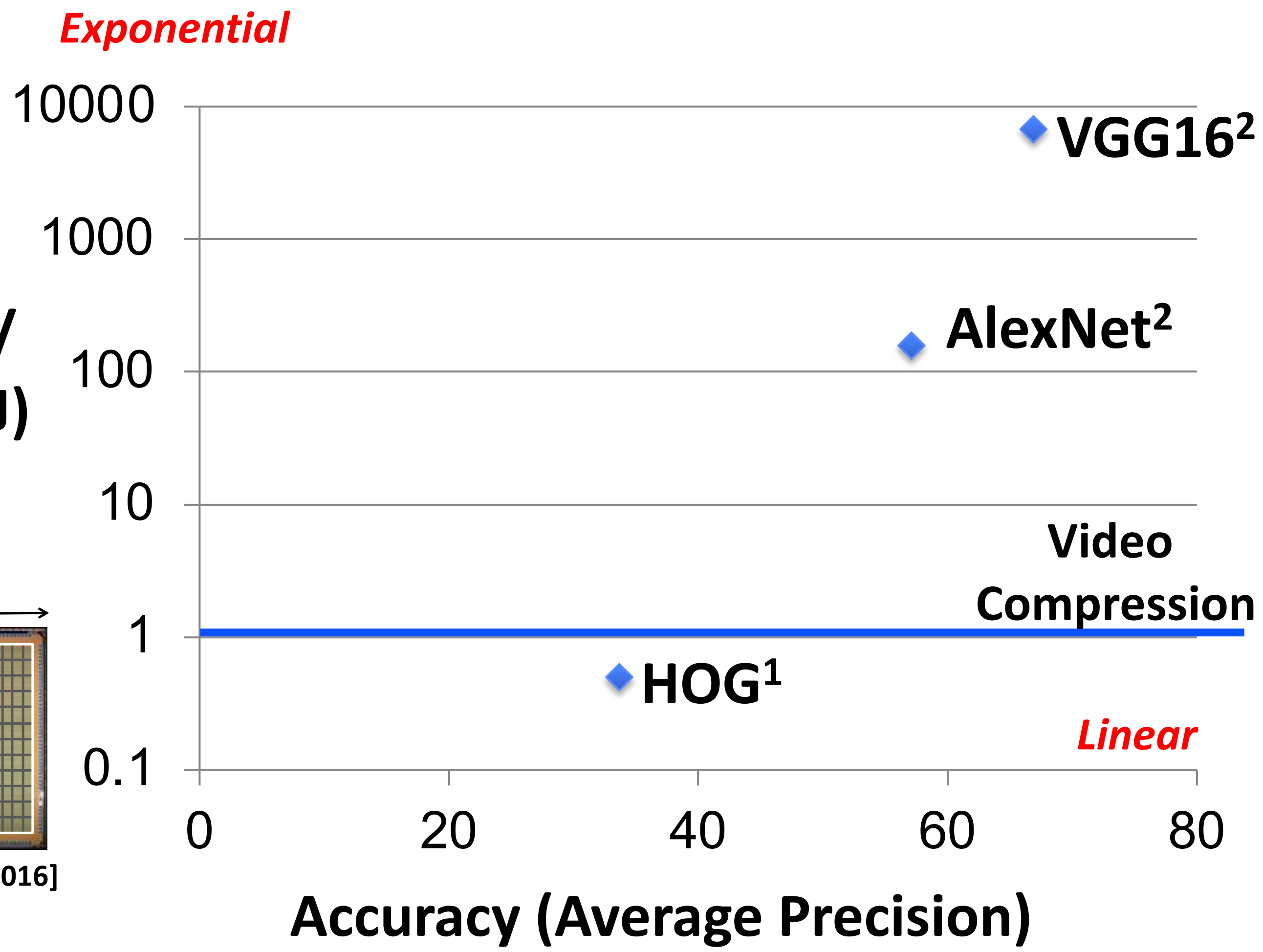
Exploits data reuse for **100x** reduction in memory accesses from global buffer and **1400x** reduction in memory accesses from off-chip DRAM

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

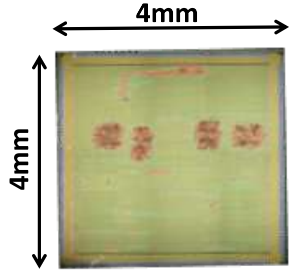
Results for AlexNet

Joint work with Joel Emer

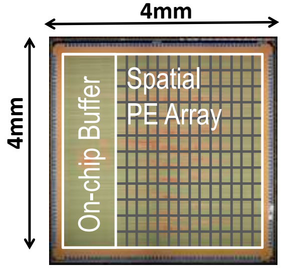
Features: Energy vs. Accuracy



Measured in 65nm*



❶ [Suleiman, VLSI 2016]



❷ [Chen, ISSCC 2016]

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

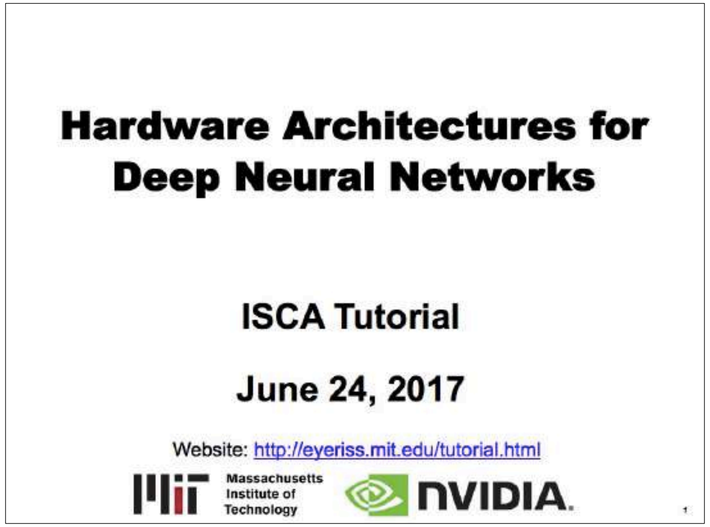
Measured in on VOC 2007 Dataset

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

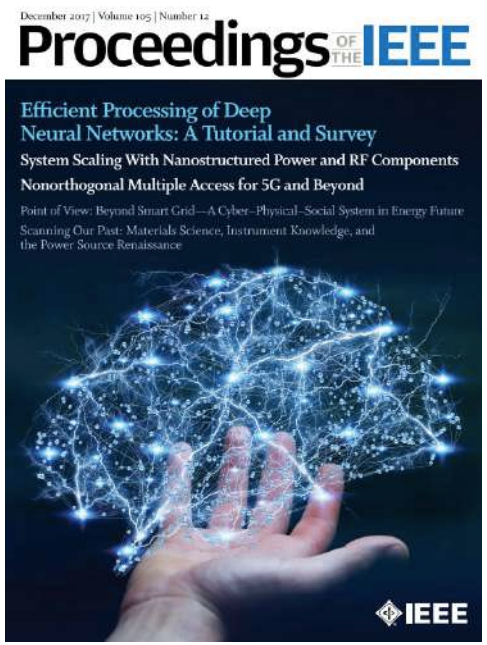
June 24, 2017

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

Massachusetts Institute of Technology

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

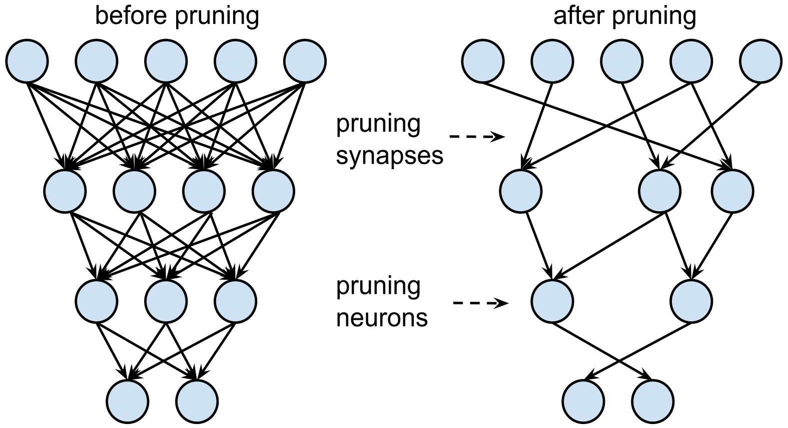
V. Sze, Y.-H. Chen,
T-J. Yang, J. Emer,
"***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***,"
Proceedings of the IEEE,
Dec. 2017

We identified various limitations to existing approaches

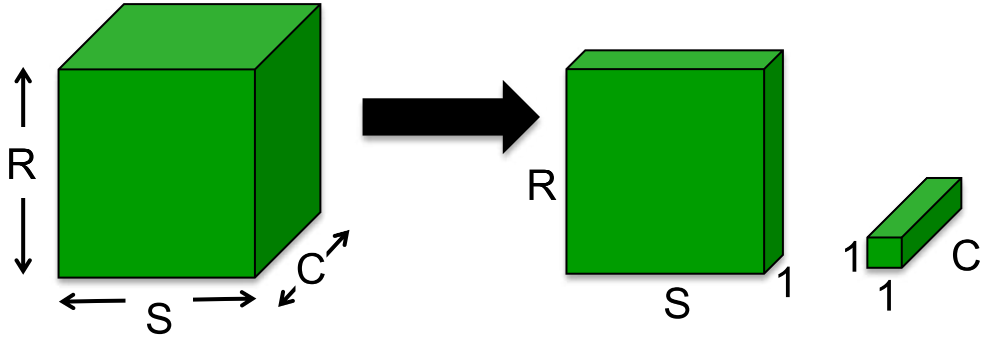
Design of Efficient DNN Algorithms

Popular efficient DNN algorithm approaches

Network Pruning



Efficient Network Architectures



Examples: SqueezeNet, MobileNet

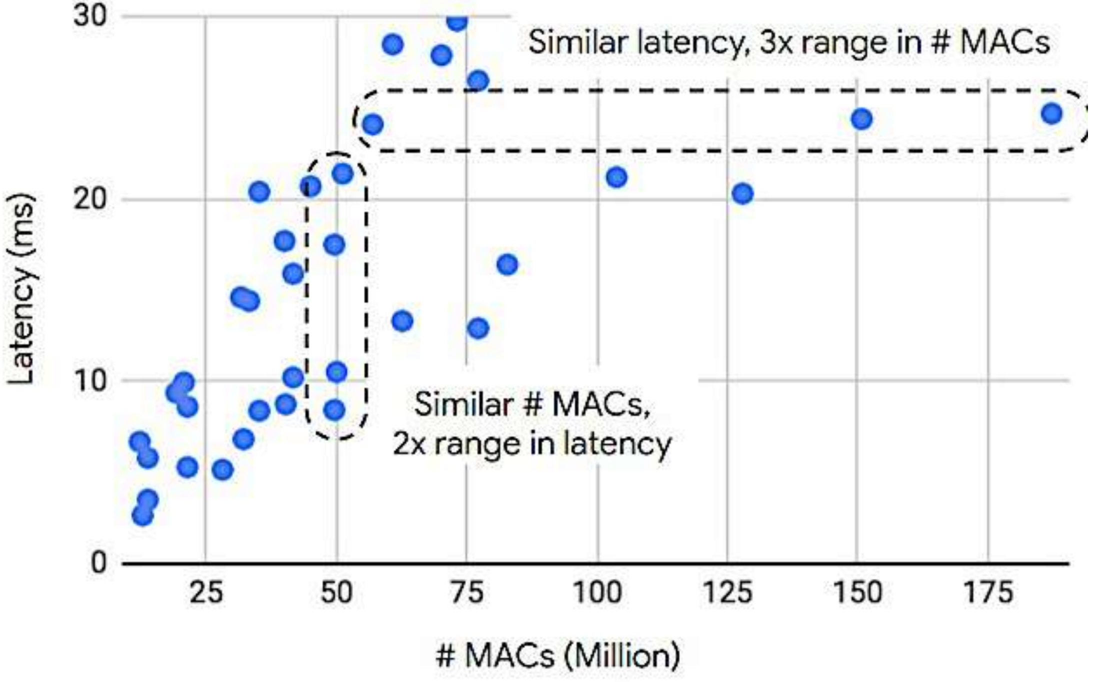
... also reduced precision

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

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

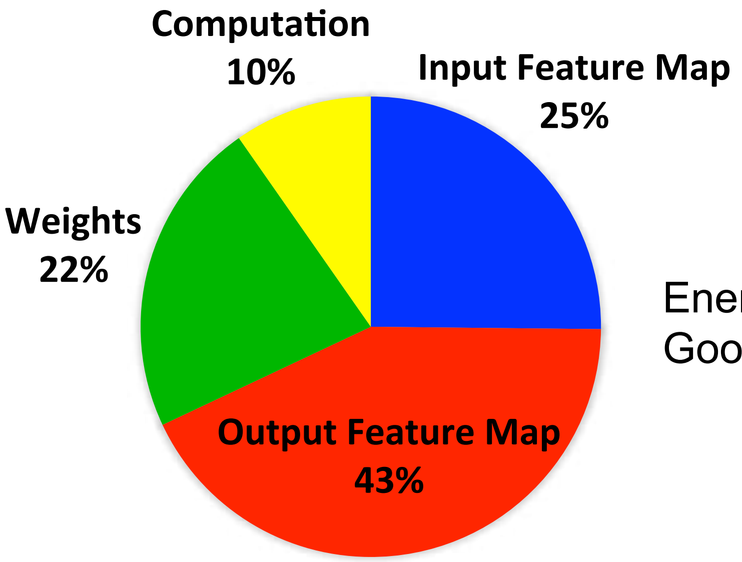
Number of MACs and Weights are Not Good Proxies

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]

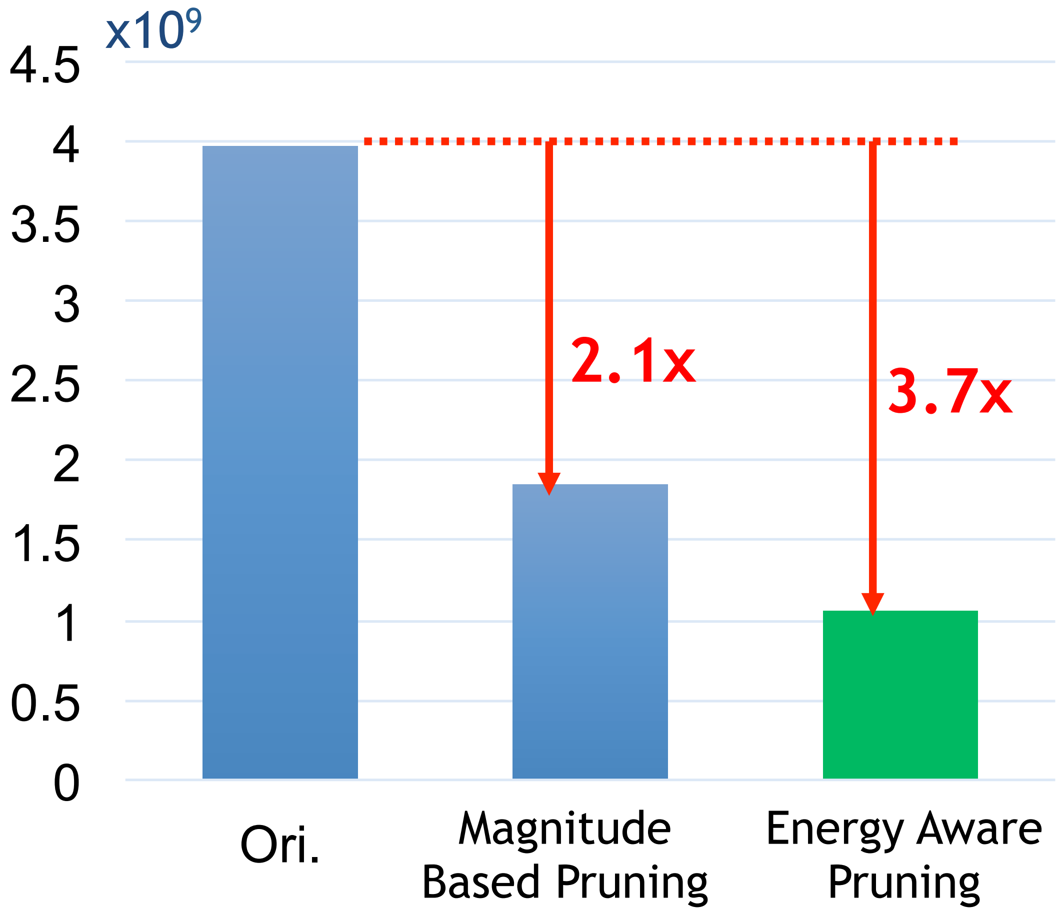
Energy-Aware Pruning

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]

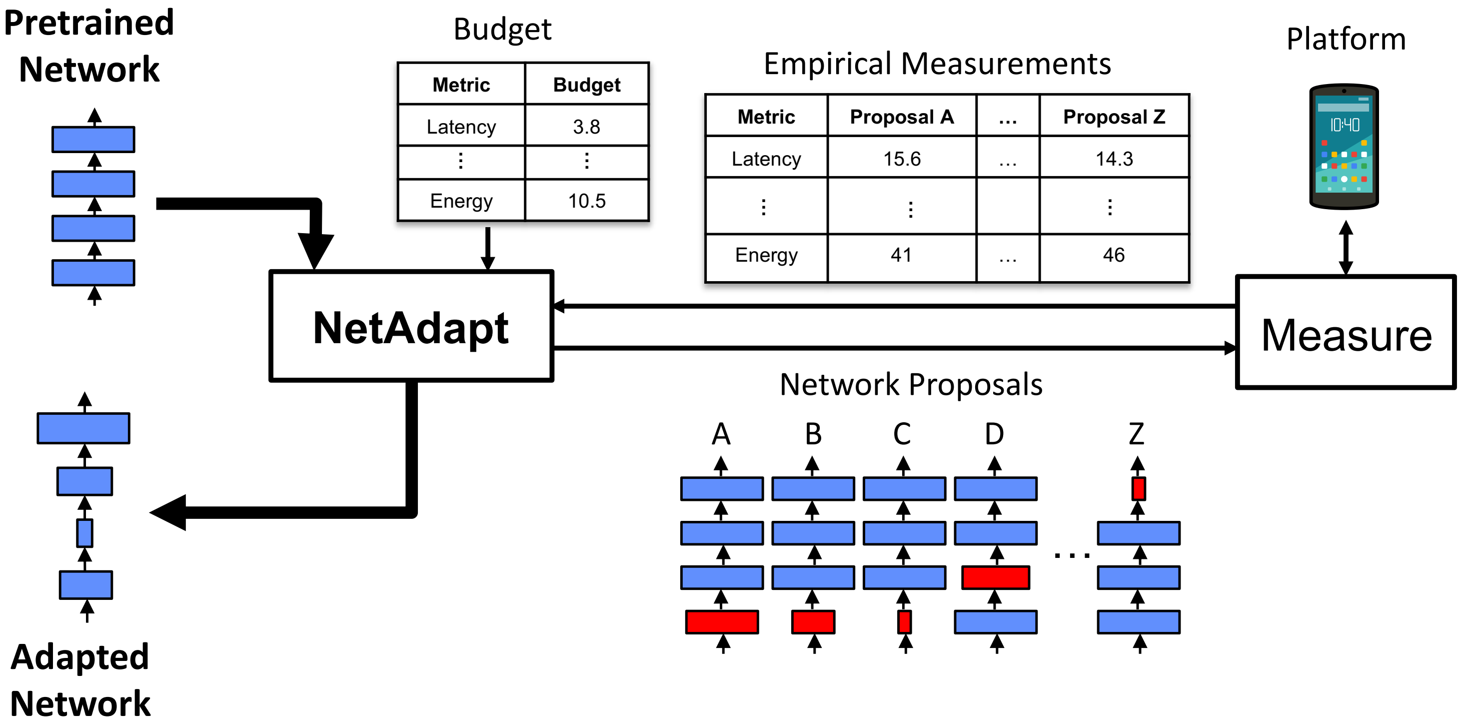
Normalized Energy (AlexNet)



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

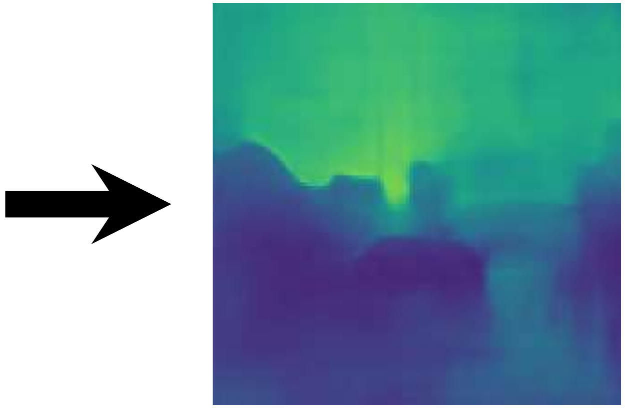
Joint work with Google's Mobile Vision Team

FastDepth: Fast Monocular Depth Estimation

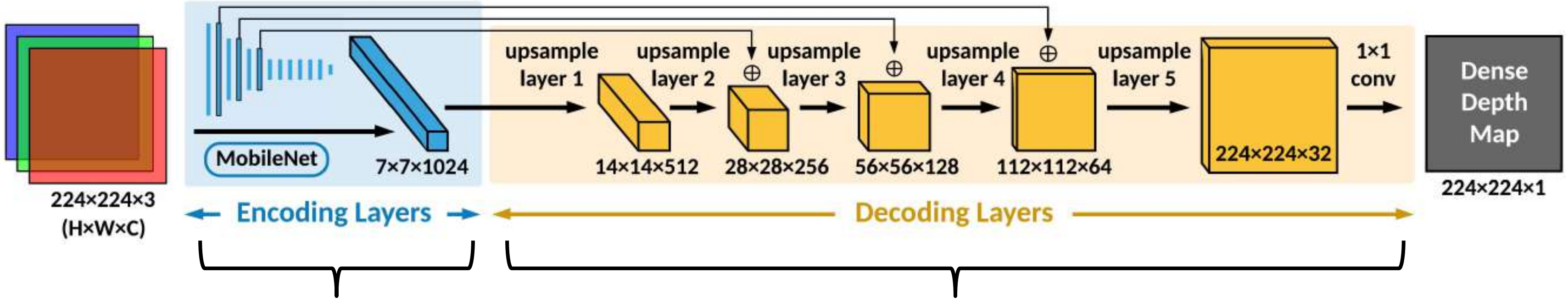
Depth estimation from a single RGB image desirable, due to the relatively low cost and size of monocular cameras.

RGB

Prediction



Auto Encoder DNN Architecture (Dense Output)

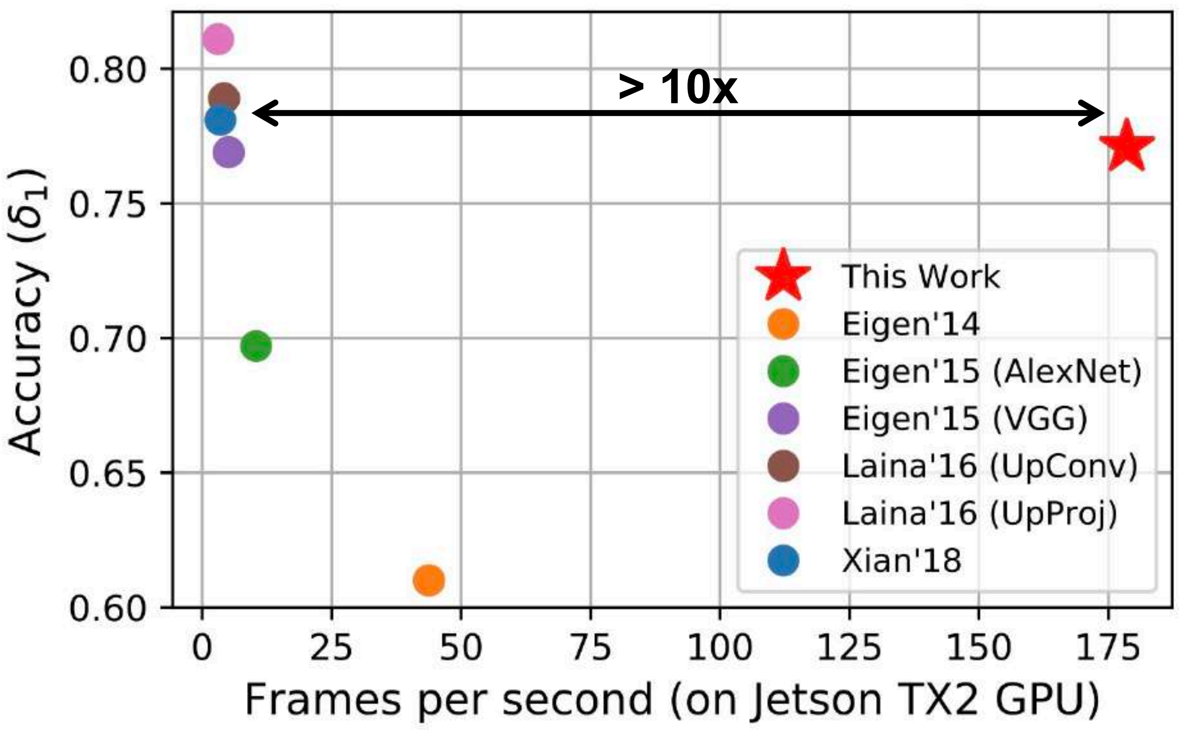


Reduction (similar to classification)

Expansion

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)

~40fps on an iPhone

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

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

Joint work with Sertac Karaman

NetAdapt v2: Reduce Adaption Time

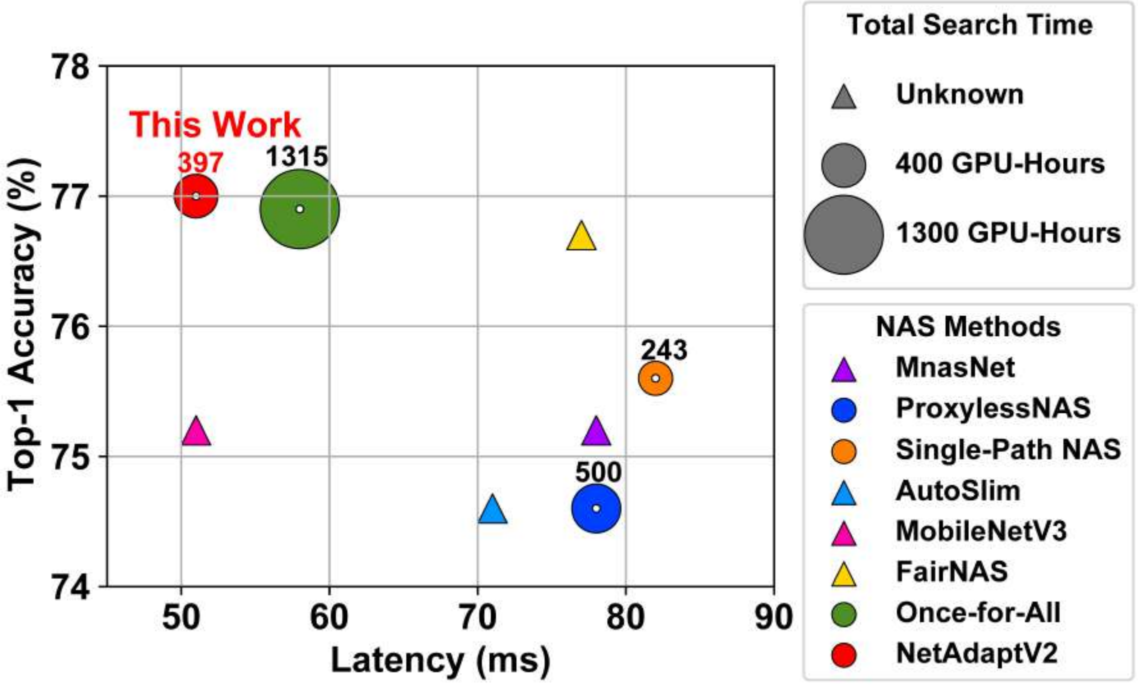
Reduce time to find efficient DNN that adapts to hardware by up to 5.8x

Typical Steps in Neural Architecture Search (NAS):

1) Train super-network (search space of DNNs)
2) Sample and evaluate different DNNs
3) Fine tune the final DNN

Contributions

- ***Ordered dropout:*** train multiple DNNs in *single* forward pass (reduce step 1)
- ***Channel-level bypass:*** merge layer depth and channel width into a *single* search dimension (reduce step 2)
- ***Multi-layer coordinate descent optimizer:*** consider joint effect of multiple layers (reduce step 2 & support non-differentiable metrics, e.g., latency)

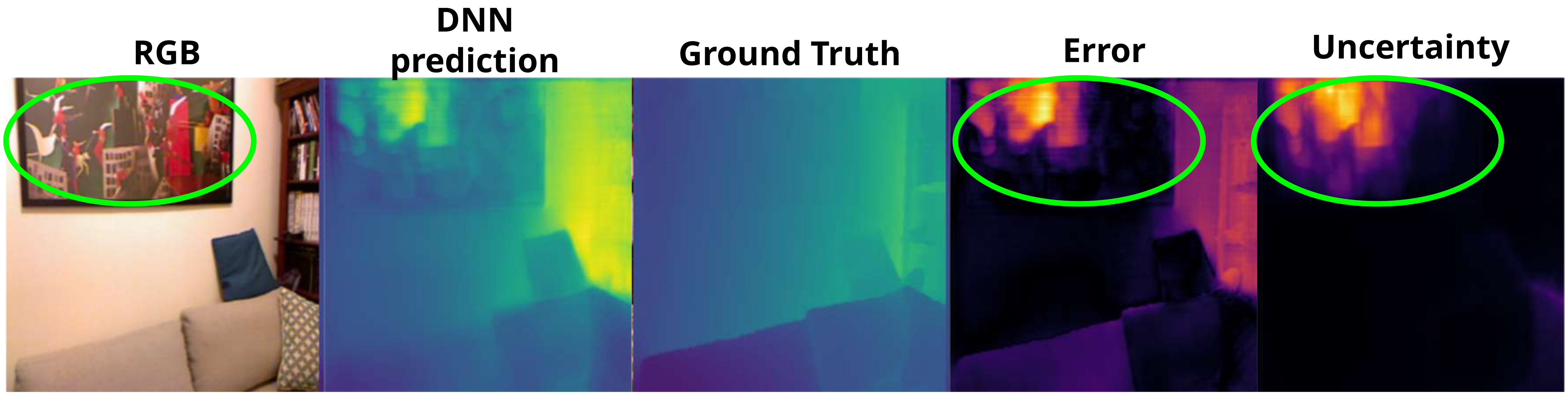


More info at http://netadapt.mit.edu

[**Yang**, *CVPR* 2021]

Measuring Uncertainty in DNN Monocular Depth Estimation

Need to estimate uncertainty (sensor noise model) for robot decision making

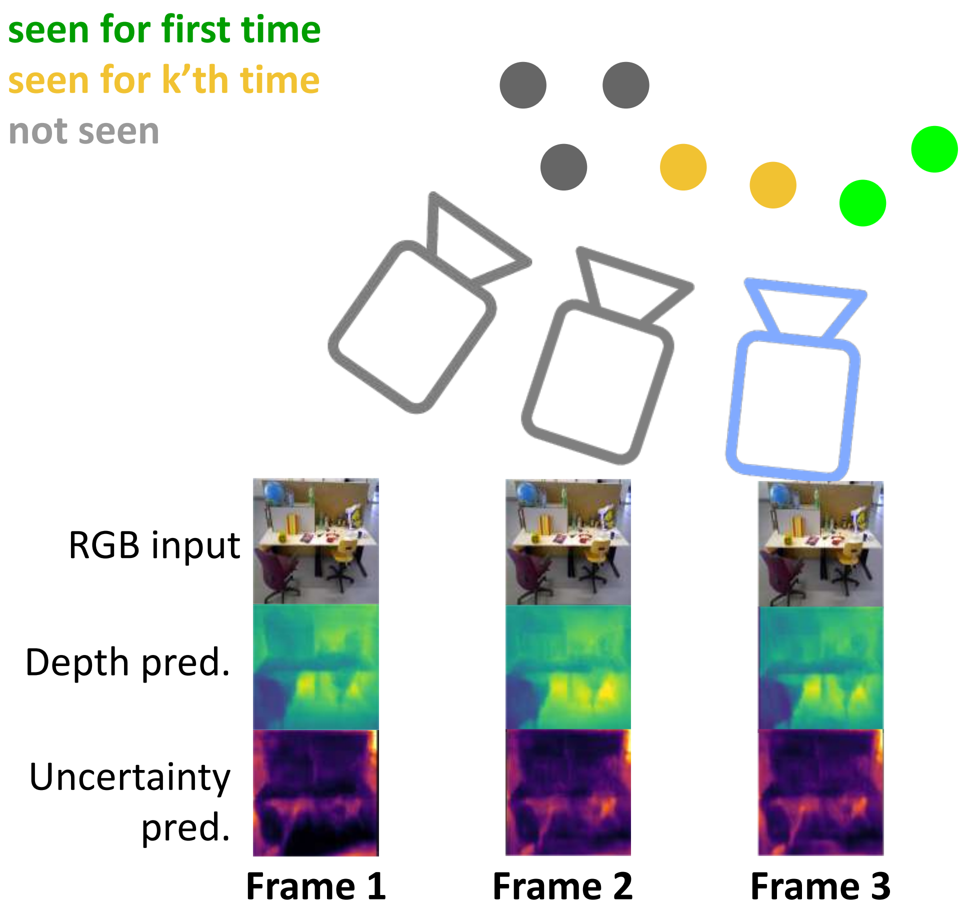


Popular approaches involve running ***multiple*** DNNs on the same input

[**Sudhakar**, *ICRA* 2022]

Joint work with Sertac Karaman

Uncertainty from Motion (UfM)



UfM needs to run only ***one*** DNN per input

It exploits **temporal redundancy** in video inputs by **merging outputs that belong to the same point in 3D space across multiple views** to estimate uncertainty

[**Sudhakar**, *ICRA* 2022]

Joint work with Sertac Karaman

Mapping with Gaussian Mixture Models

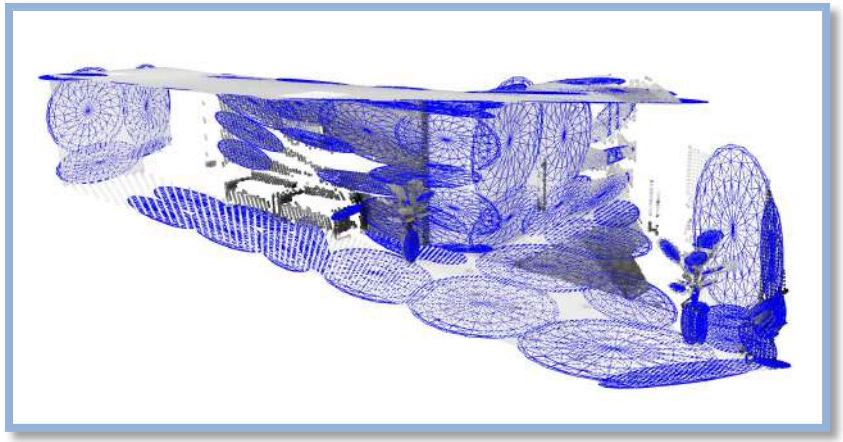
Convert depth images to Gaussian Mixture Models (GMMs) to construct a compact 3D map of an environment.

2D Depth Image

307,200 pixels (3.5MB)

Convert →

Gaussian Mixture Models (blue)



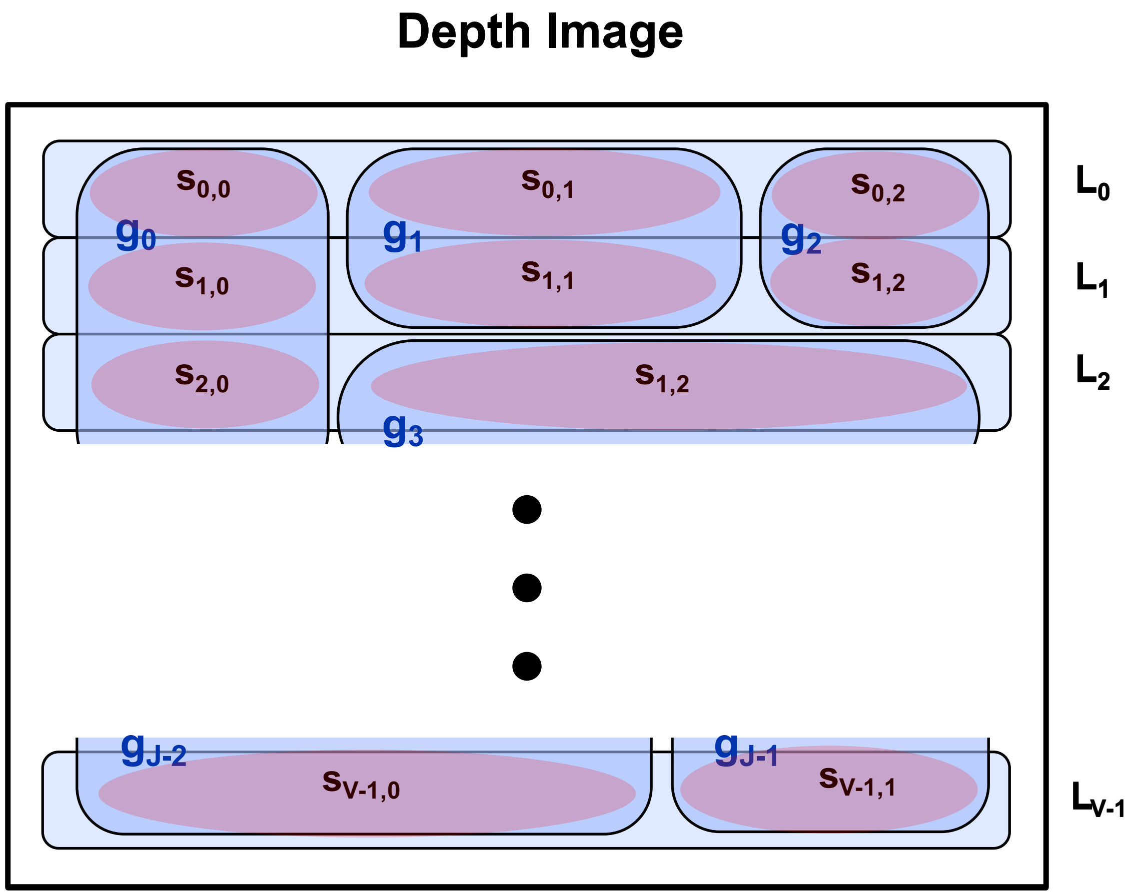
Around 1000 parameters (12-18 kB)

While existing approaches focus on reducing map size, they do not account for the memory cost ***during*** the conversion process

[**Li**, *ICRA* 2022]

Joint work with Sertac Karaman

Single Pass Gaussian Fitting (SPGF)


Depth Image

SPGF Approach: Scanline Segmentation + Segment Fusion

- **Single pass** reduces storage of inputs and temporary variables
- **Row-by-row based approach** allows for accurate and efficient inference of surface geometries in a single pass

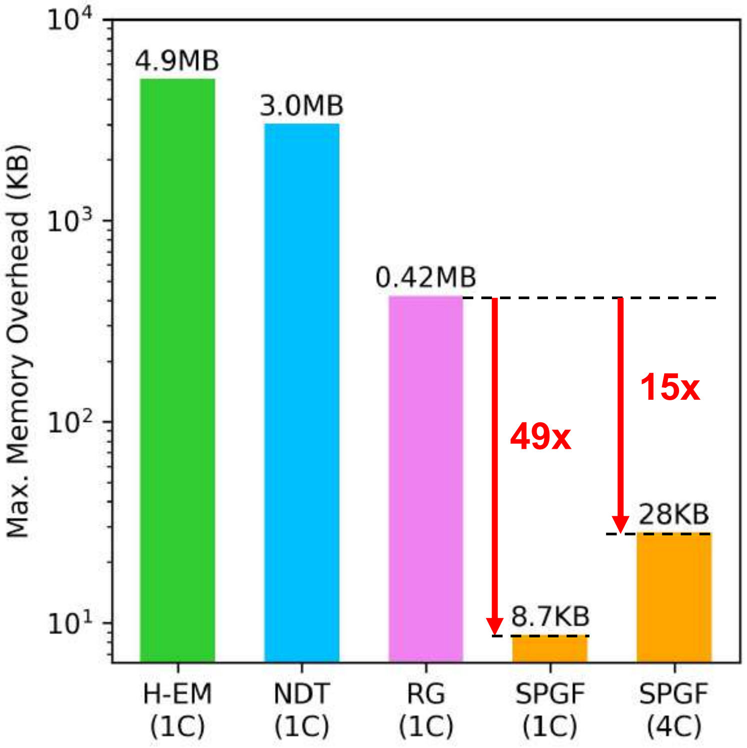
[**Li**, *ICRA* 2022]

Joint work with Sertac Karaman

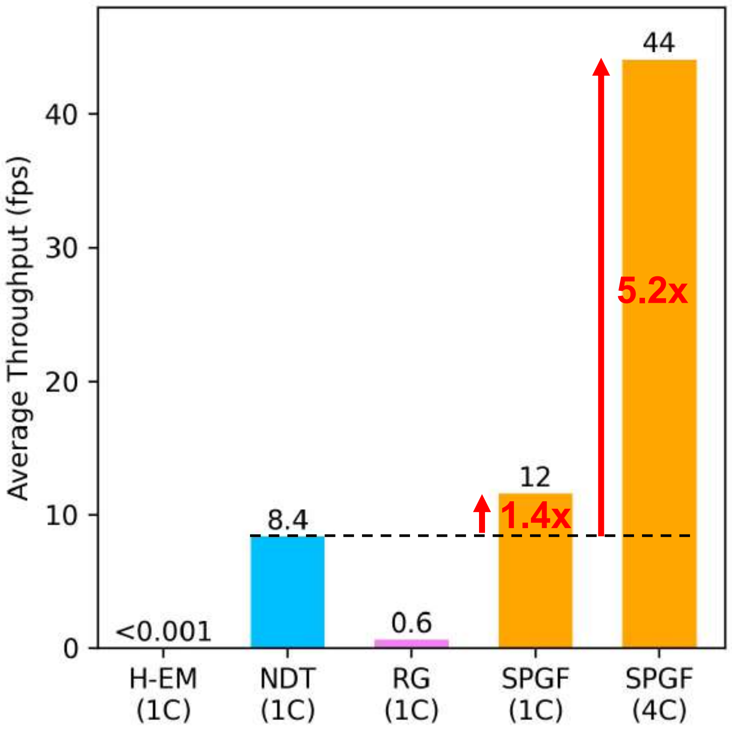
SPGF Results on TUM RGB-D Room

Comparison of SPGF with other approaches at similar accuracy and compactness

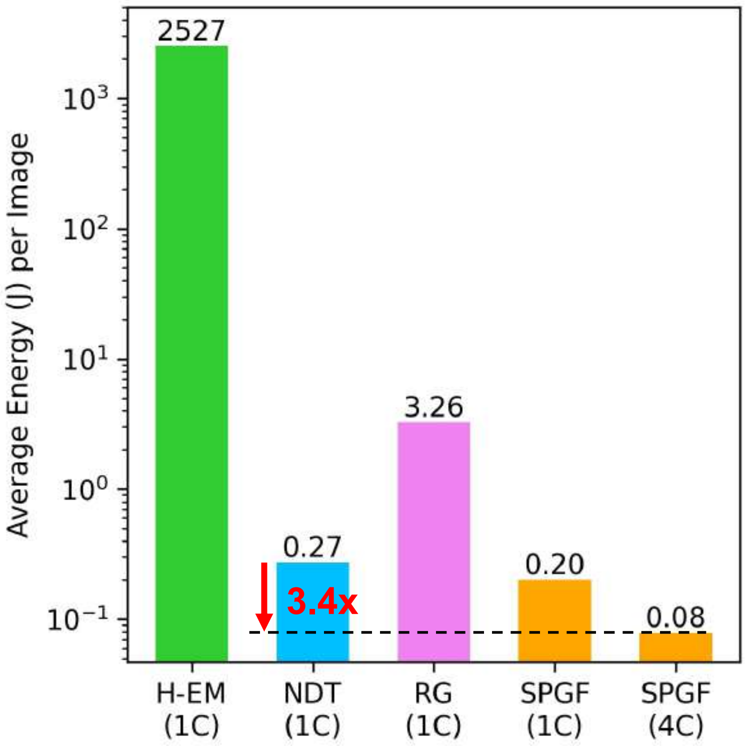
Memory Overhead



Throughput



Energy Consumption



Hierarchical EM (**H-EM**):[**Eckart**, *CVPR* 2016], Normal Distance Transform (**NDT**):[**Saarinen**, *IJRR* 2013], Region Growing (**RG**):[**Dhawale**, *RSS* 2020]

SPGF only uses **KBs** of memory overhead and achieves **real-time** on a low-power ARM Cortex-57 CPU

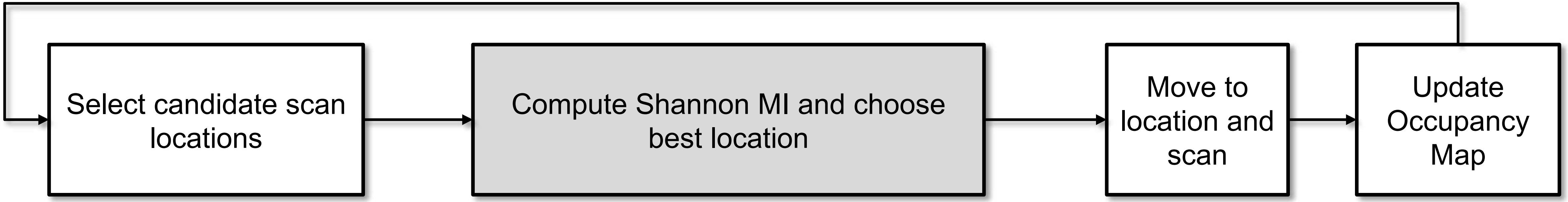
Note: All algorithms were similarly optimized in C++

Where to Go Next: Planning and Mapping

Robot Exploration

Mutual-Information-Based Exploration

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



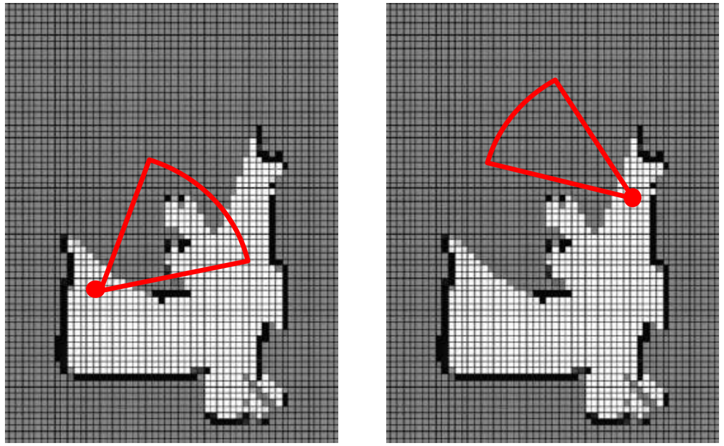
Select candidate scan locations

Compute Shannon MI and choose best location

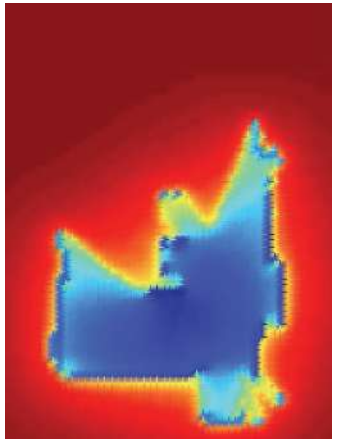
Move to location and scan

Update Occupancy Map

Where to scan?



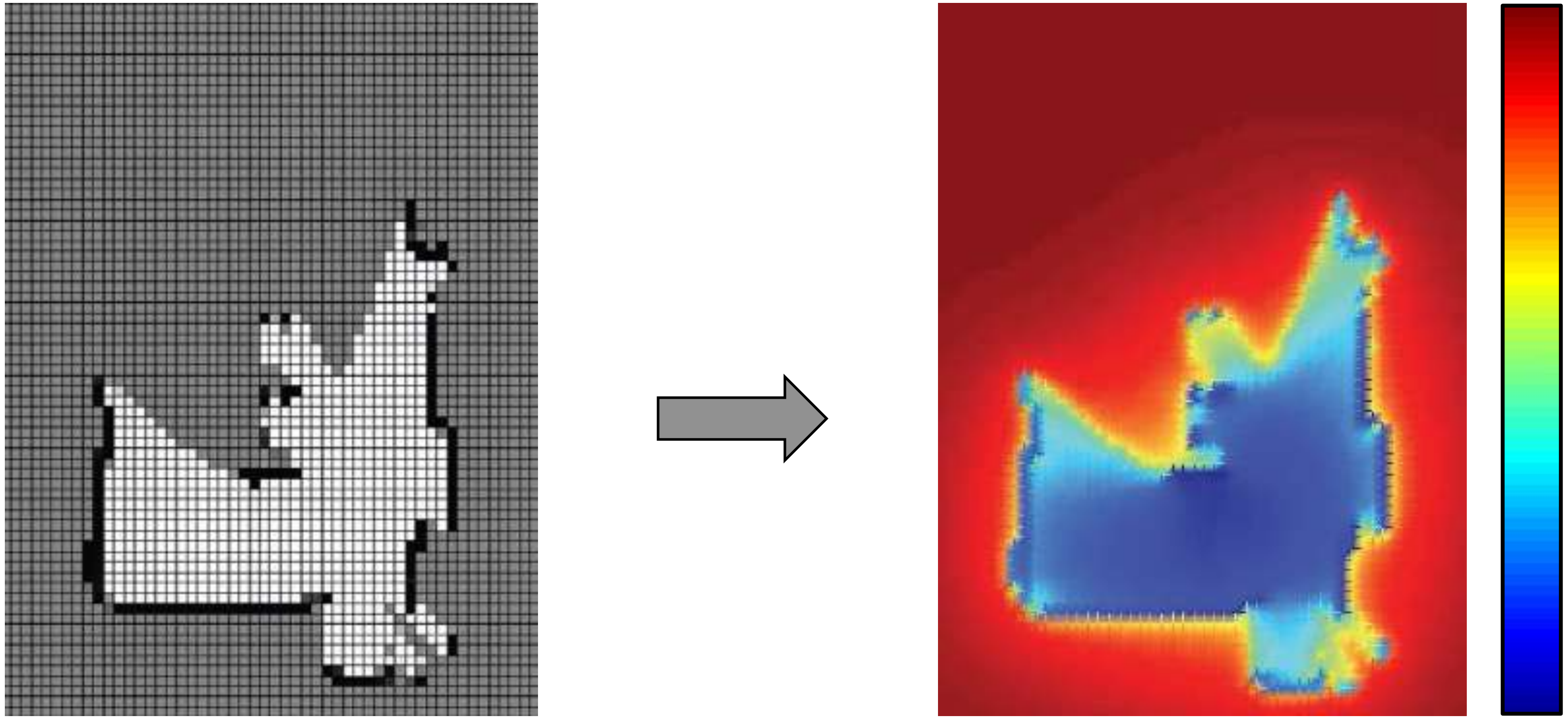
Mutual Information



Updated Map

Joint work with Sertac Karaman

Information Theoretic Mapping



Occupancy grid map, M

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

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

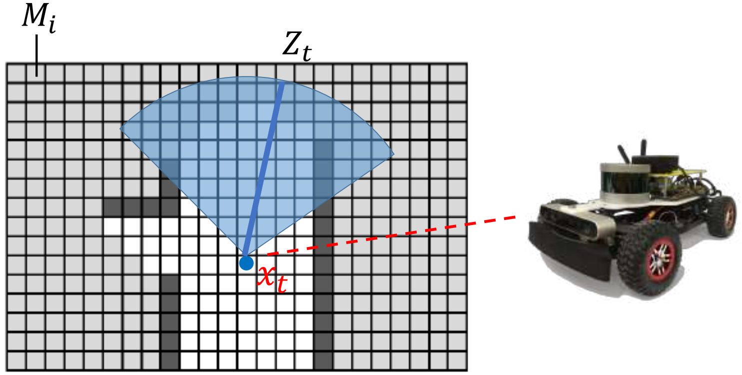
Perspective updated map entropy	Current map entropy	Mutual information

FSMI: Fast Shannon Mutual Information

Shannon Mutual Information
(between ray Z and map M)
[**Julian**, *IJRR* 2014]

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

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



FSMI: Fast Shannon Mutual Information

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

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

Approximate FSMI

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

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

*Charrow et al., ICRA 2015

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

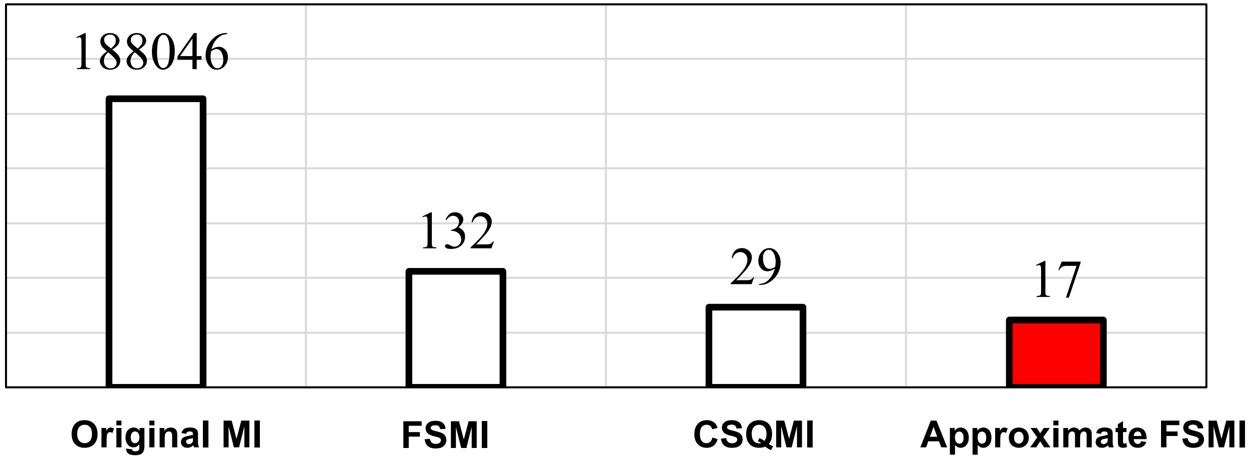
Joint work with Sertac Karaman

FSMI: Fast Shannon Mutual Information

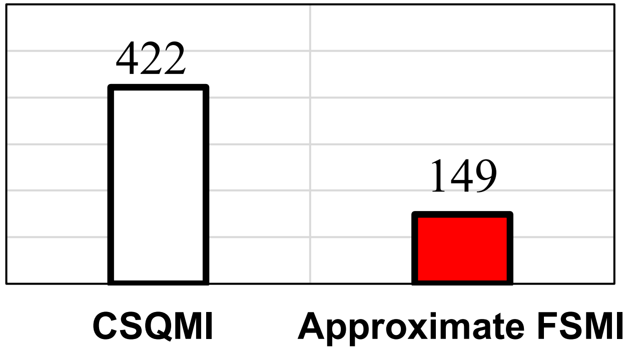
Original MI[1]	FSMI	CSQMI[2]	Approximate FSMI
$O(n^2\lambda_z)$	$O(n^2)$	$O(n)$	$O(n)$

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

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



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



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

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

Joint work with Sertac Karaman

Experimental Results (4x Real Time)

Exploration with a mini race car using motion capture for localization

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

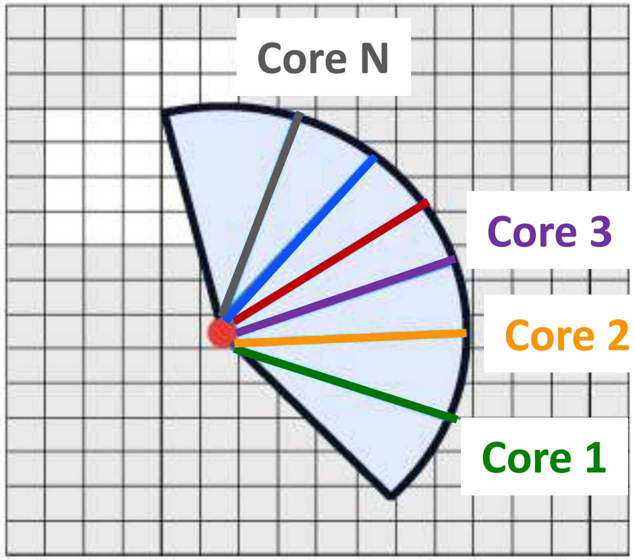
Joint work with Sertac Karaman

Building Hardware to Compute FSMI

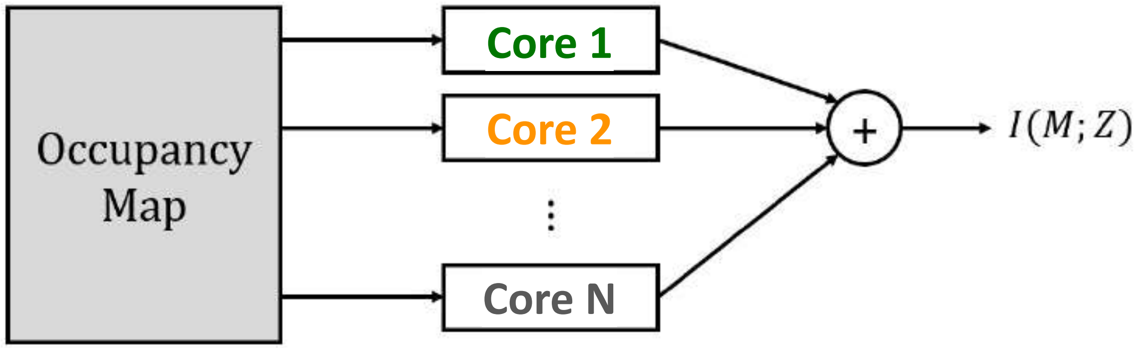
Motivation: Compute MI faster for faster exploration!

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

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



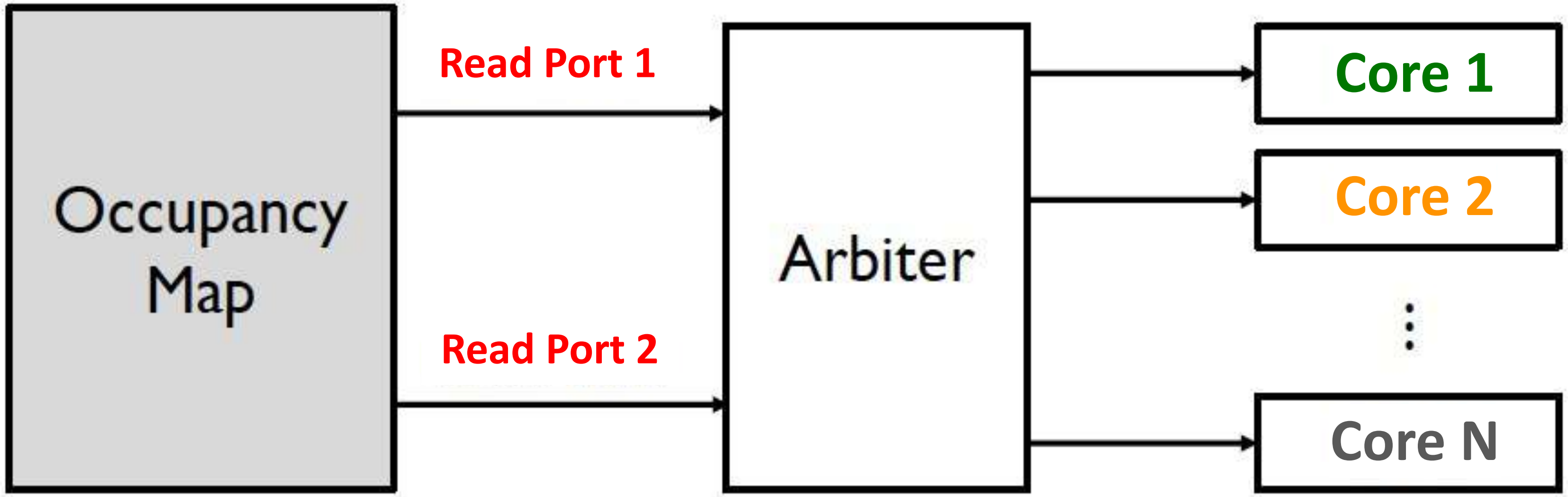
Process beams in parallel with multiple cores



Challenge is Data Delivery to All Cores

Power consumption of memory scales with number of ports.

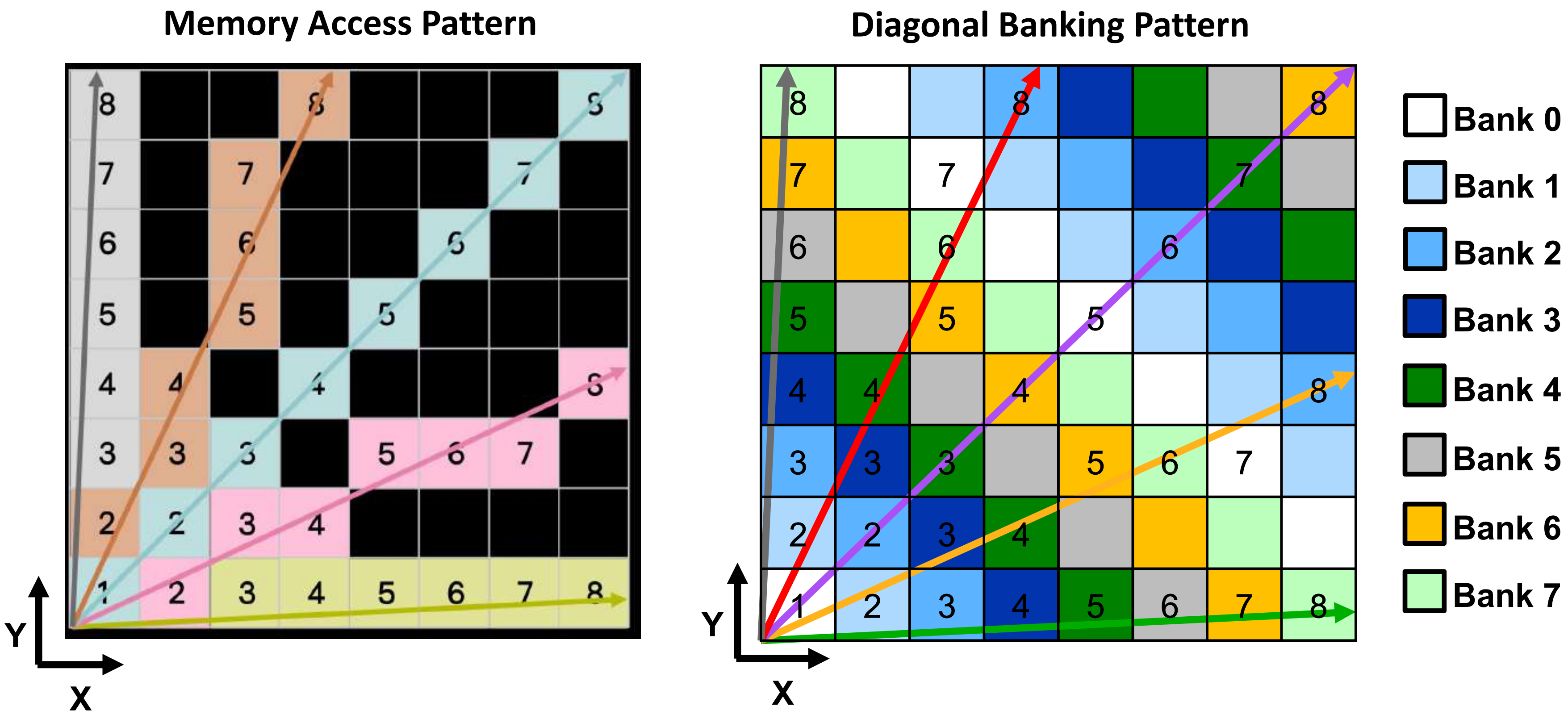
Low power SRAM limited to two-ports!



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

Specialized Memory Architecture

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

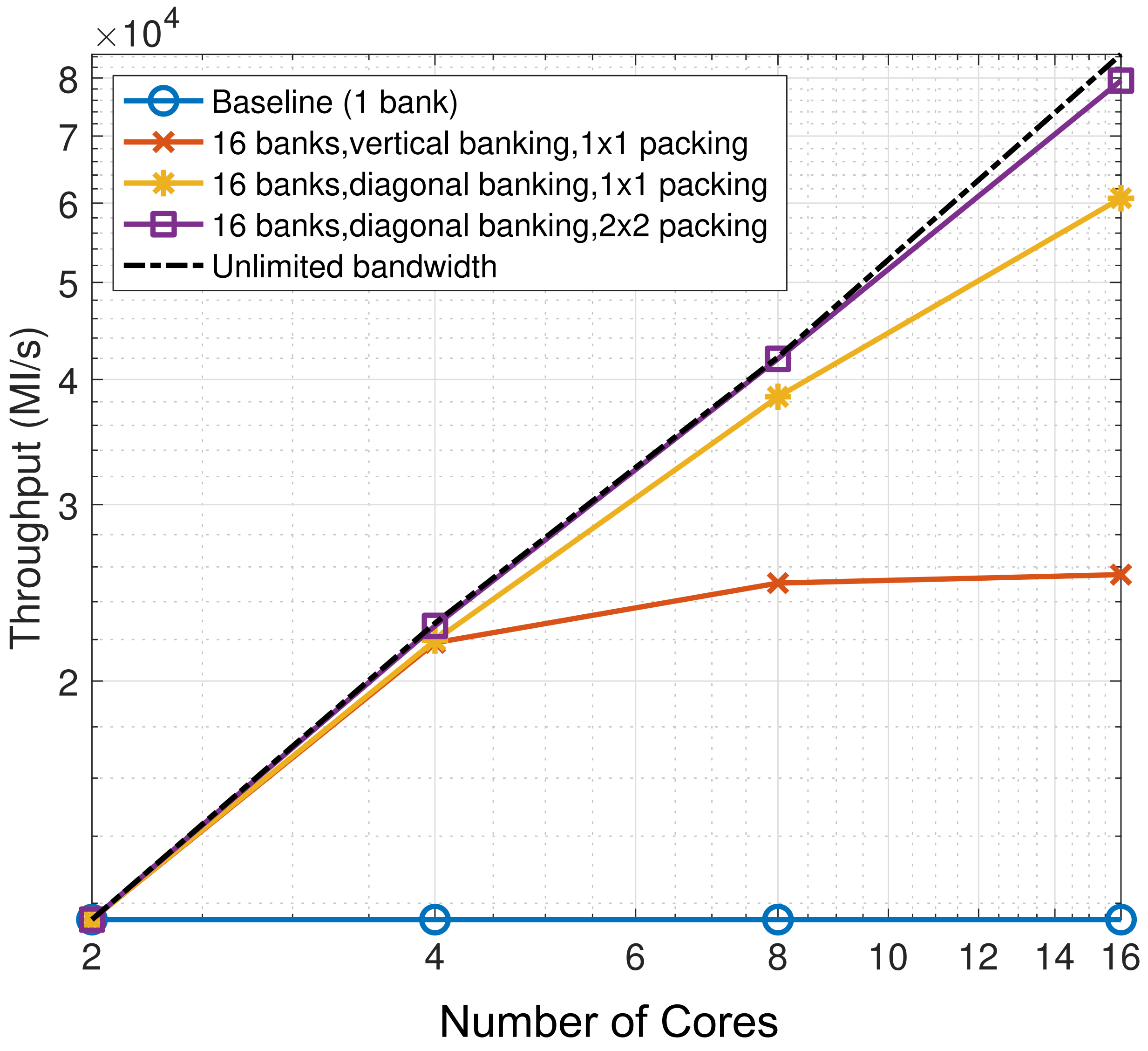


Memory Access Pattern **Diagonal Banking Pattern**

[Li, *RSS* 2019]

Joint work with Sertac Karaman

Experimental Results



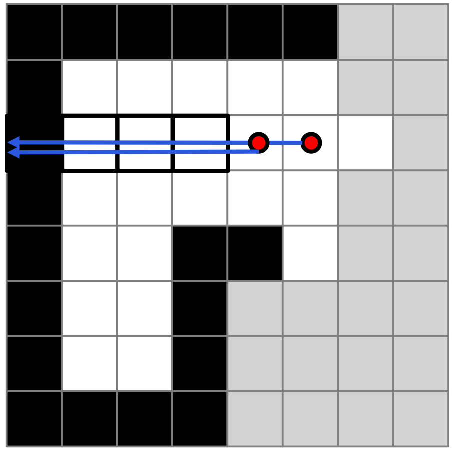
Specialized banking, efficient memory arbiter and packing multiple values at each address results in throughput **within 94% of theoretical limit** (unlimited bandwidth)

Compute MI for an **entire map** of 20m x 20m at 0.1m resolution **in under a second** while consuming **under 2W** on a ZC706 FPGA (100x faster than CPU at 10x lower power)

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

FCMI: Fast Continuous Mutual Information

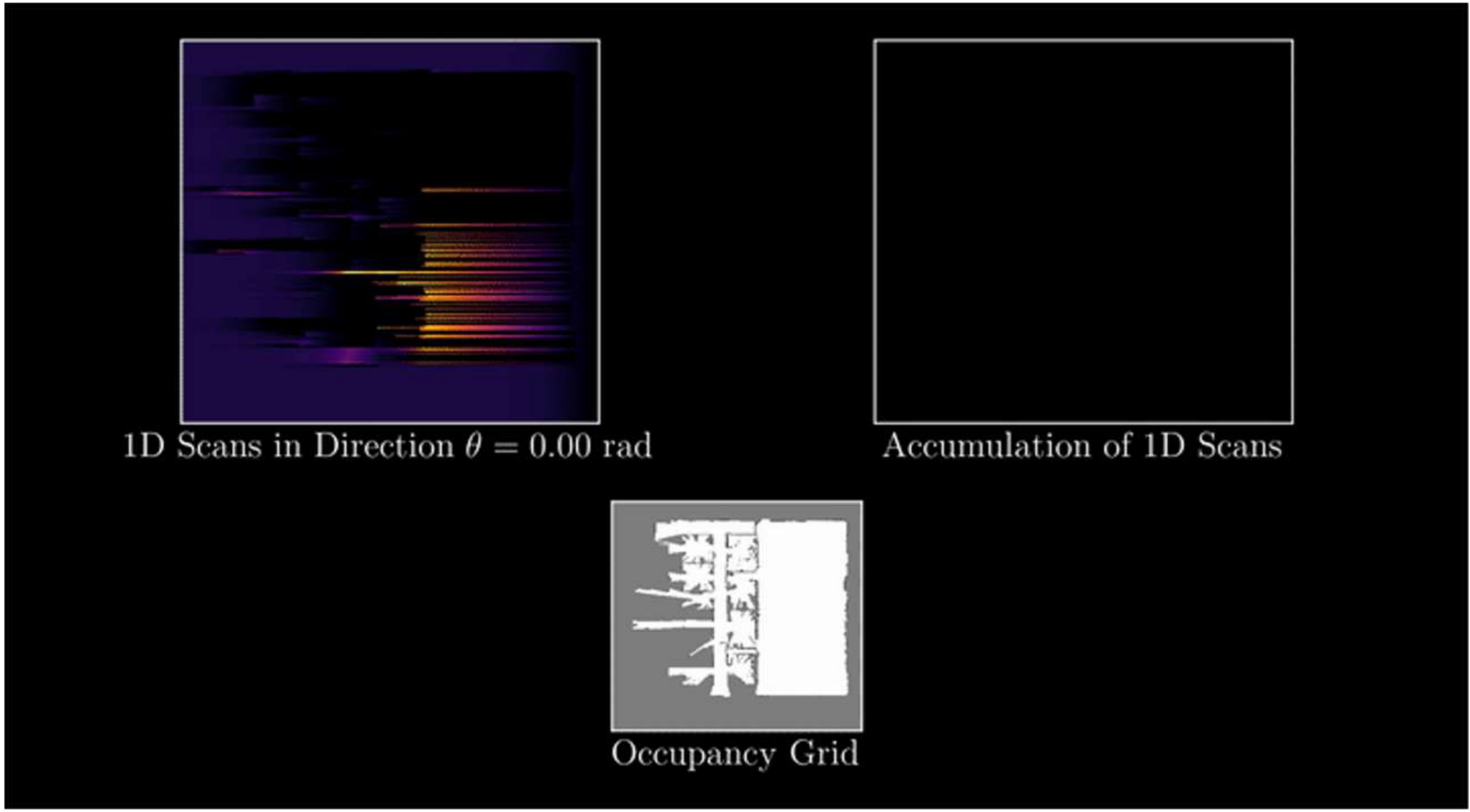
Reformulate with a ***continuous*** occupancy map framework and exploit recursive structure when computing MI across ***entire*** map



n = cells per ray
L = number of rays
H^2 = size of map

FSMI: $O(nLH^2)$ → FCMI: $O(LH^2)$

Two orders of magnitude speed up over FSMI!

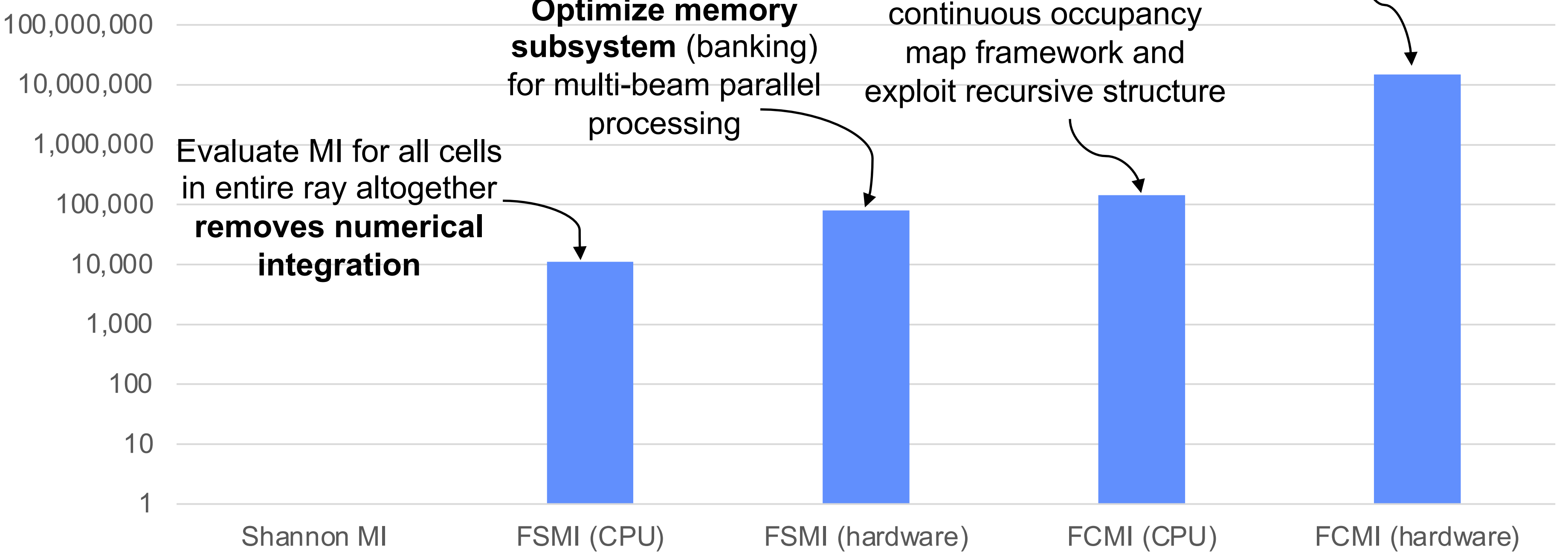


[**Henderson**, *ICRA* 2020]

Joint work with Sertac Karaman

Several Orders of Magnitude Speed up Via Co-Design

For a 200x200 Map
(Note: Speed up increases for larger maps)



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

Optimize memory subsystem (banking) for multi-beam parallel processing

Reformulate using a continuous occupancy map framework and exploit recursive structure

Optimize memory subsystem, **time-interleave cores and approximate computing**

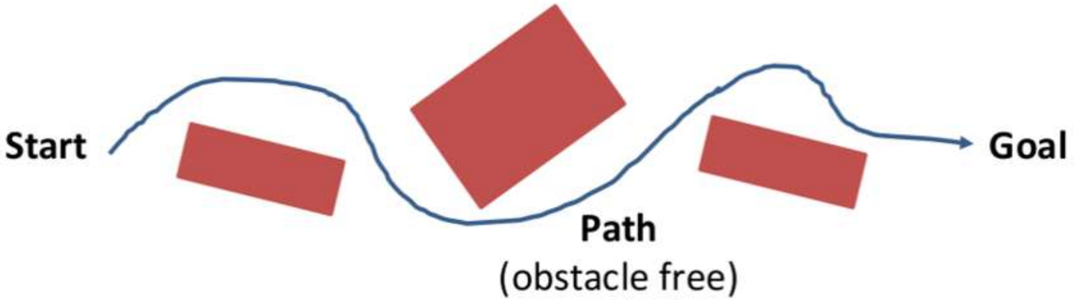
[**Julian**, *IJRR* 2014] [**Zhang**, *ICRA* 2019] [**Li**, *RSS* 2019] [**Henderson**, *ICRA* 2020] [**Gupta**, *IROS* 2021]

Compute mutual information for the **entire map** in real time for the first time!

Joint work with Sertac Karaman

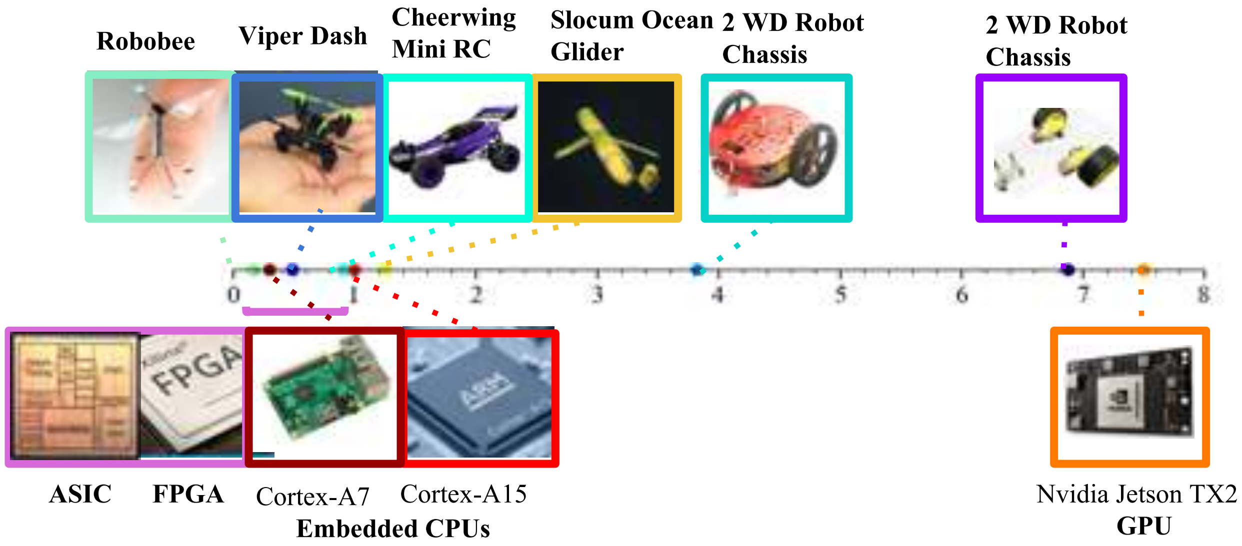
Balancing Actuation and Computing Energy

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



Low-Energy 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])

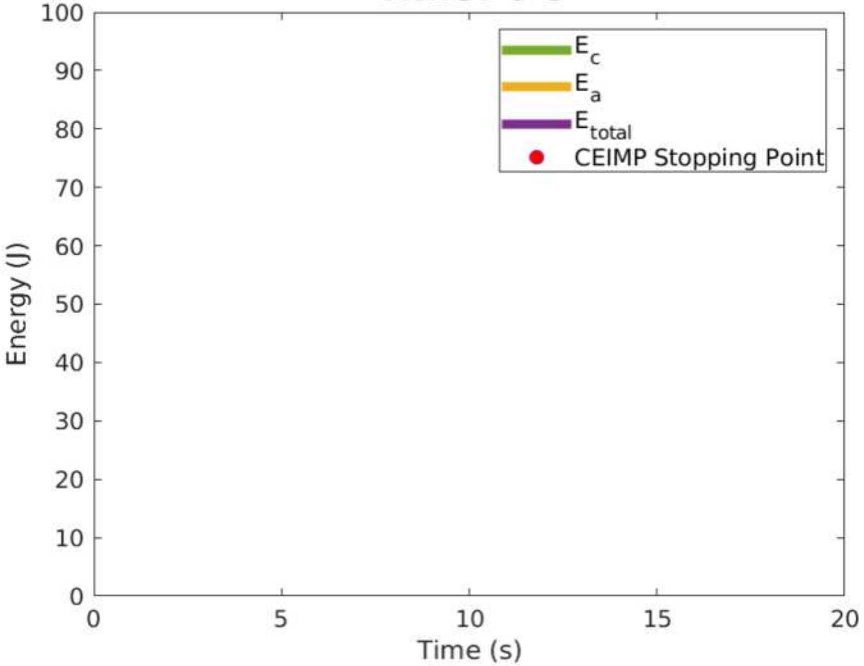
[**Sudhakar**, *ICRA* 2020]

Joint work with Sertac Karaman

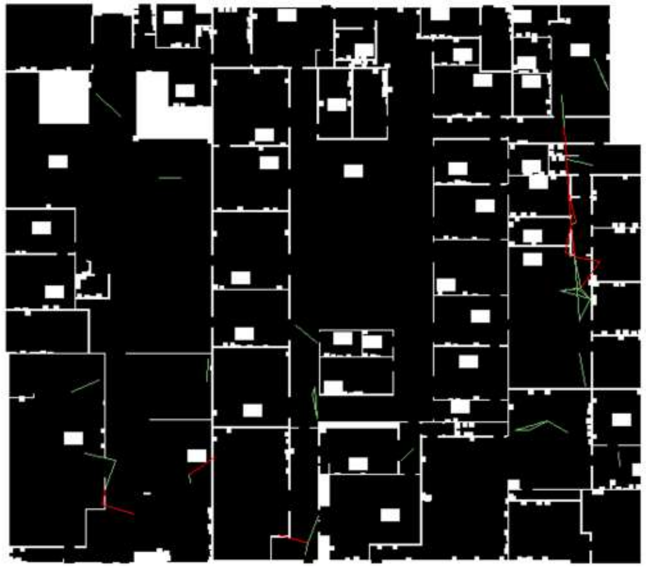
Balancing Actuation and Computing Energy

Baseline
(compute 20,000 samples)

Time: 0 s



CEIMP



Compute Energy Included Motion Planning (CEIMP)

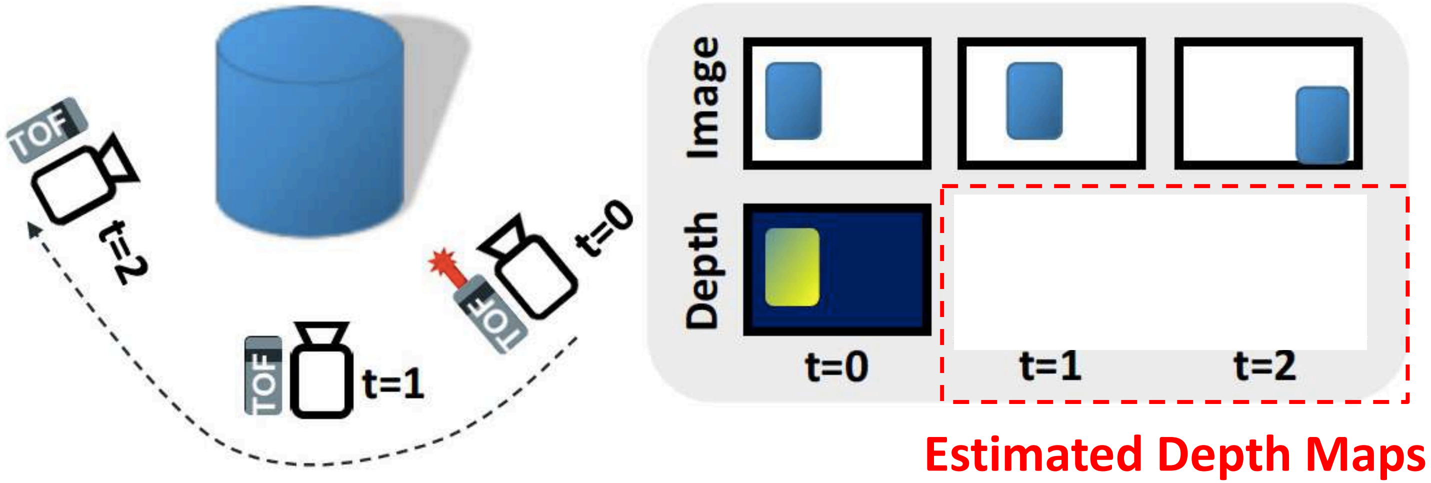
A framework to balance the energy spent on **computing** a path and the energy spent on **moving** along that path **(Don't think too hard!)**

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

Joint work with Sertac Karaman

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**



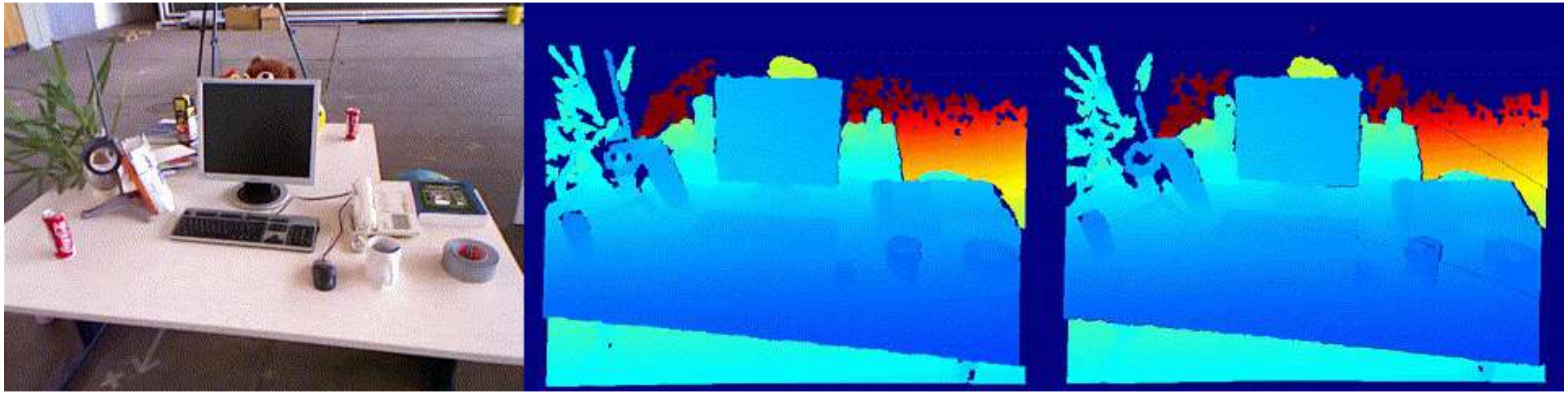
Estimated Depth Maps

Real-time Performance on Embedded Processor

VGA @ 30 fps on Cortex-A7 CPU (< 0.5W active power)

[**Noraky**, *TCSVT* 2019]

Results of Low Power Depth ToF Imaging



RGB Image

Depth Map
Ground Truth

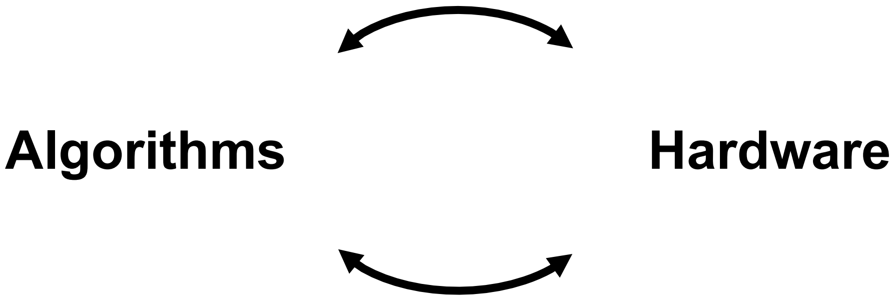
Depth Map
Estimated

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

[**Noraky**, *TCSVT* 2019]

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 creates new opportunities for the co-design of algorithms and hardware → **Innovation opportunities for the future of robotics!**



Algorithms

Hardware

Acknowledgements

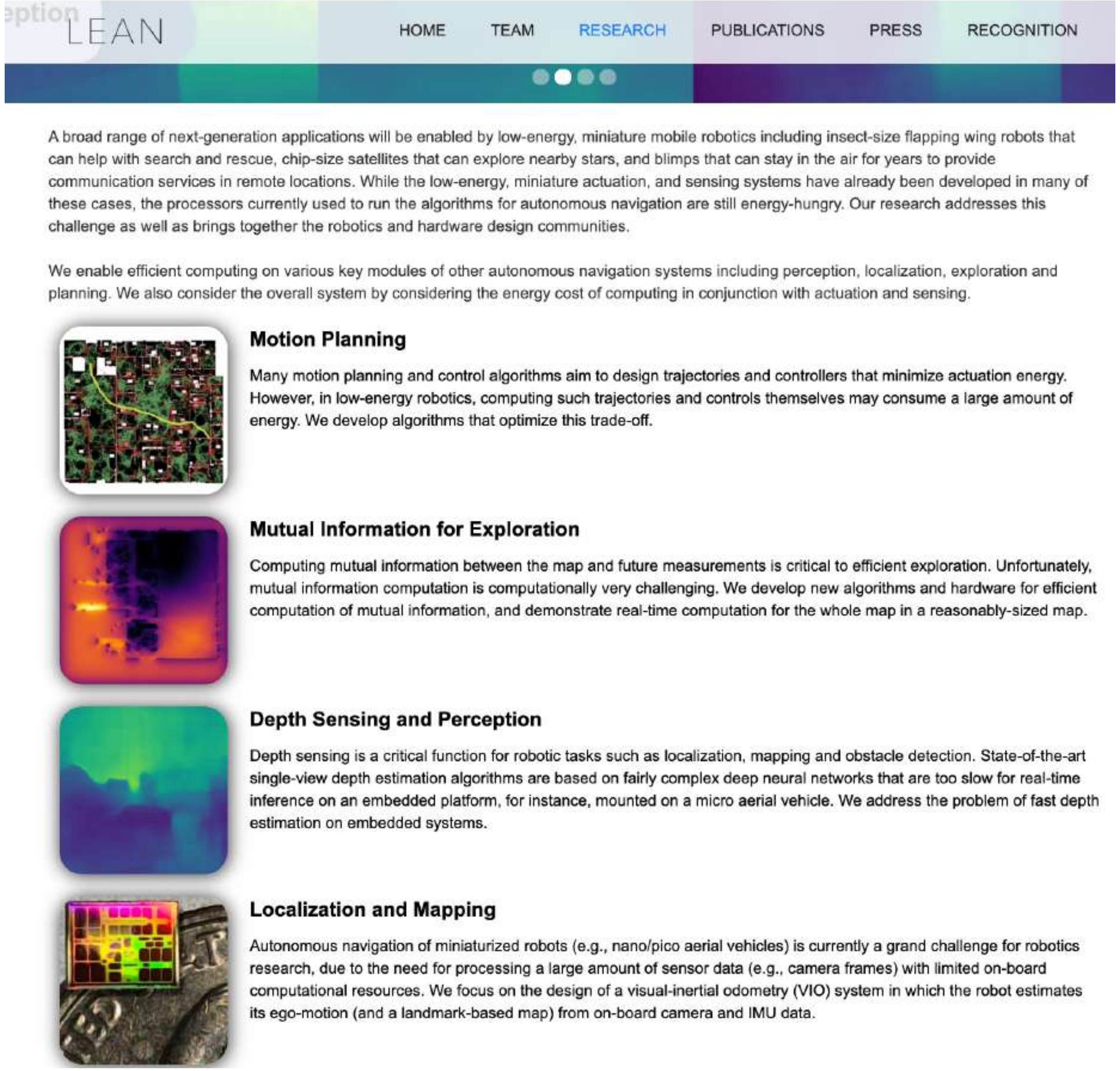
Joel Emer

Sertac Karaman

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

AFOSR, 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

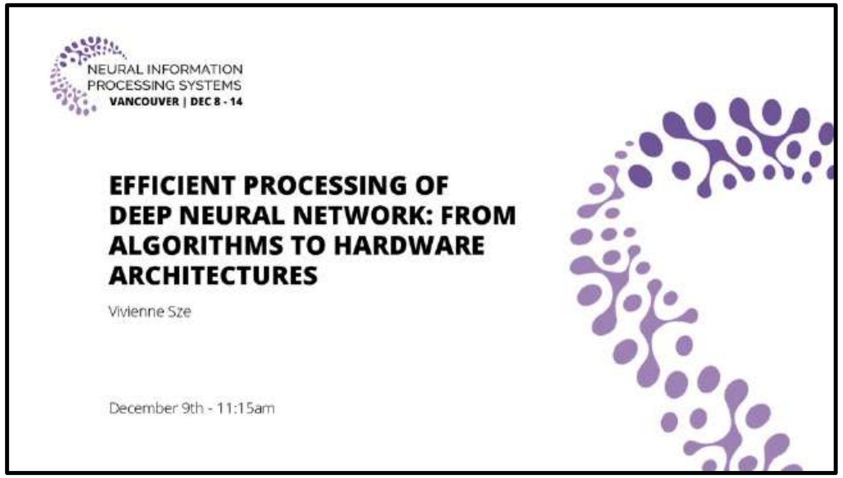
Depth sensing is a critical function for robotic tasks such as localization, mapping and obstacle detection. State-of-the-art single-view depth estimation algorithms are based on fairly complex deep neural networks that are too slow for real-time inference on an embedded platform, for instance, mounted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

Resources on Efficient Processing of DNNs

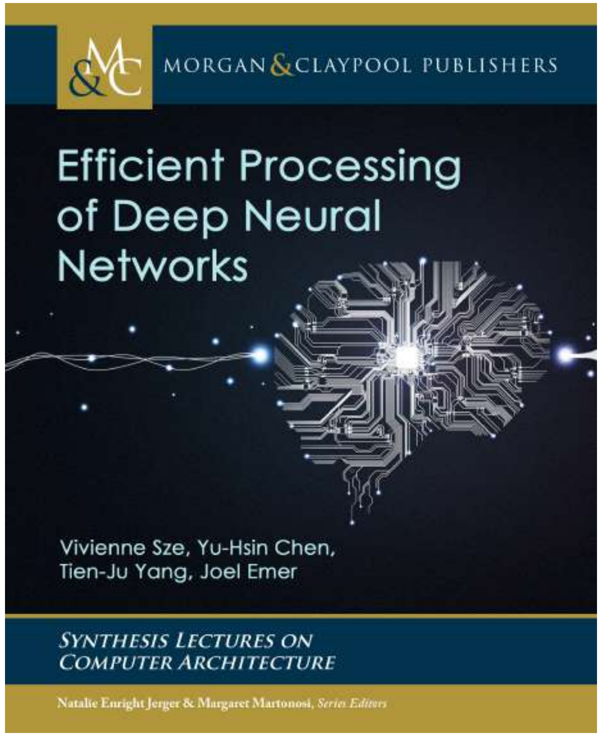


NEURAL INFORMATION PROCESSING SYSTEMS
VANCOUVER | DEC 8 - 14

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

Vivienne Sze

December 9th - 11:15am



MORGAN & CLAYPOOL PUBLISHERS

Efficient Processing of Deep Neural Networks

Vivienne Sze, Yu-Hsin Chen,
Tien-Ju Yang, Joel Emer

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE

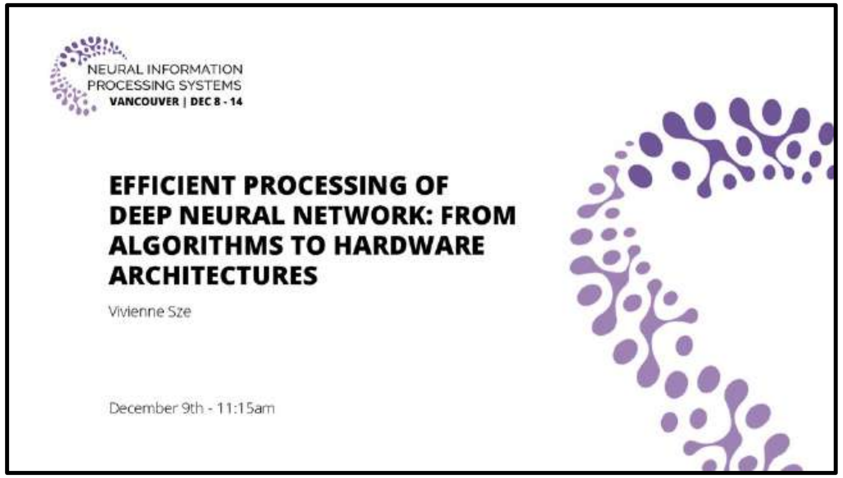
Natalie Enright Jerger & Margaret Martonosi, *Series Editors*

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

Additional Resources

Talks and Tutorial Available Online

http://sze.mit.edu/slides



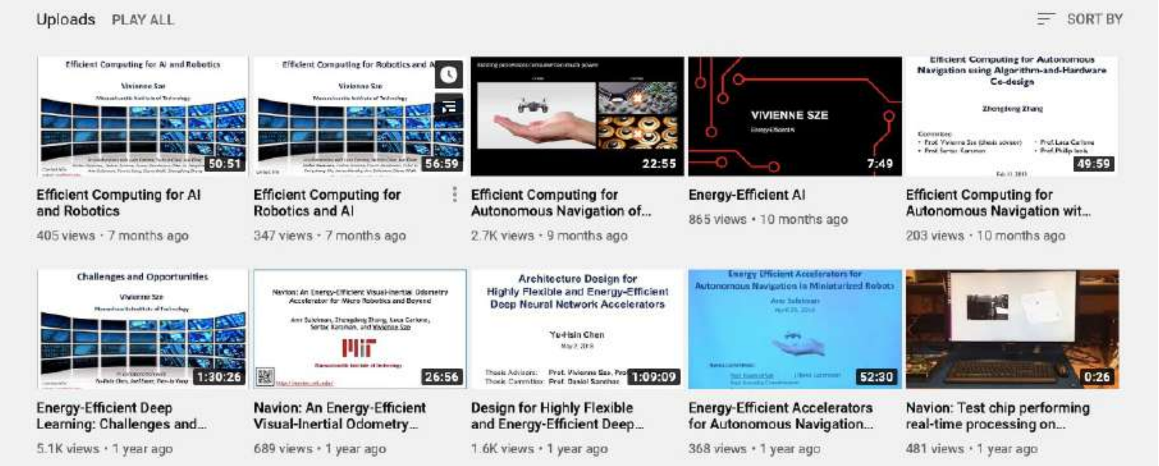
NEURAL INFORMATION PROCESSING SYSTEMS
VANCOUVER | DEC 8 - 14

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

Vivienne Sze

December 9th - 11:15am

YouTube Channel
EEMS Group – PI: Vivienne Sze



50:51	56:59	22:55	7:49	49:59
Efficient Computing for AI and Robotics	Efficient Computing for Robotics and AI	Efficient Computing for Autonomous Navigation of...	Energy-Efficient AI	Efficient Computing for Autonomous Navigation wit...
405 views · 7 months ago	347 views · 7 months ago	2.7K views · 9 months ago	865 views · 10 months ago	203 views · 10 months ago
1:30:26	26:56	1:09:09	52:30	0:26
Energy-Efficient Deep Learning: Challenges and...	Navion: An Energy-Efficient Visual-Inertial Odometry...	Design for Highly Flexible and Energy-Efficient Deep...	Energy-Efficient Accelerators for Autonomous Navigation...	Navion: Test chip performing real-time processing on...
5.1K views · 1 year ago	689 views · 1 year ago	1.6K views · 1 year ago	368 views · 1 year ago	481 views · 1 year ago

References

- **Energy-Efficient Visual Inertial Localization**
  - **Project website:** http://navion.mit.edu
  - 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 Fully Integrated Energy-Efficient Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones," IEEE Symposium on VLSI Circuits (VLSI-Circuits), June 2018
  - 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
- **Efficient Map Compression**
  - **Project website:** https://lean.mit.edu/highlights/localization-mapping
  - P. Z. X. Li, S. Karaman, V. Sze, "Memory-Efficient Gaussian Fitting for Depth Images in Real Time," IEEE International Conference on Robotics and Automation (ICRA), May 2022.

References

- **Efficient Processing 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, “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.
  - 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
  - T.-J. Yang, Y.-L. Liao, V. Sze, “NetAdaptV2: Efficient neural architecture search with fast super-network training and architecture optimization,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2021.
- **Monocular Depth Estimation using Deep Neural Networks**
  - **Project website:** https://lean.mit.edu/highlights/depth-sensing
  - 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/
  - S. Sudhakar, V. Sze, S. Karaman, “Uncertainty from Motion for DNN Monocular Depth Estimation,” IEEE International Conference on Robotics and Automation (ICRA), May 2022.

References

- **Fast Shannon Mutual Information for Robot Exploration**
  - **Project website:** https://lean.mit.edu/highlights/mutual-information
  - 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,” International Journal of Robotics Research (IJRR), August 2020
  - 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
  - K. Gupta, P. Z. X. Li, S. Karaman, V. Sze, “Efficient Computation of Map-scale Continuous Mutual Information on Chip in Real Time,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2021.

References

- **Balancing Actuation and Computation**
  - **Project website:** https://lean.mit.edu/highlights/motion-planning
  - S. Sudhakar, S. Karaman, V. Sze, “Balancing Actuation and Computing Energy in Motion Planning,” IEEE International Conference on Robotics and Automation (ICRA), May 2020
- **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.