Efficient Computing for AI and Autonomy: From Hardware Accelerators to Algorithm Design

Vivienne Sze
Massachusetts Institute of Technology

In collaboration with Tanner Andrulis, Joel Emer, Keshav Gupta, Theia Henderson, Sertac Karaman, Peter Li, Fangchang Ma, Angshuman Parashar, Soumya Sudhakar, Po-An Tsai, Diana Wofk, Nellie Wu, Fisher Xue, Tien-Ju Yang, Zhengdong Zheng
Rapid Growth of Energy Consumption for Computing

Data centers accounted for 3% of US global electricity demand in 2022 and is expected to grow to 8% by 2030 [Goldman Sachs, April 2024]

Source: Nature (https://www.nature.com/articles/d41586-018-06610-y)
Compute Demands for Autonomous Vehicles

Autonomous vehicles (AVs) w/ 10 deep neural network (DNN) inferences at 60 Hz on 10 cameras:

One AV: 21.6 million inferences per hour driven

One million AVs (< 0.1% of vehicles worldwide): 21.6 trillion inferences per hour driven!

“[T]rillions of inference per day across Facebook’s data centers”

[Wu, MLSys 2021]
Existing Processors Consume Too Much Power

< 1 Watt

> 10 Watts
Efficient Computing with Cross-Layer Co-Design

Architectures

Circuits

Co-Design Across Hardware Stack

64 bits

Off-Chip DRAM

DCNN Accelerator

14×12 PE Array

Filter

Input Image Decomp

Buffer SRAM 108KB

Output Image Comp ReLU

Link Clock Core Clock

Filt Img Psum Psum ...

On-Chip Buffer Spatial PE Array

Vivienne Sze http://sze.mit.edu/
Efficient Computing with Cross-Layer Co-Design

Co-Design Algorithm and Hardware

Co-Design Across Hardware Stack

Algorithms

Architectures

Circuits

Vivienne Sze [http://sze.mit.edu/](http://sze.mit.edu/)
Efficient Computing with Cross-Layer Co-Design

Algorithms

Systems

Architectures

Circuits

Co-Design Across System

Co-Design Algorithm and Hardware

Co-Design Across Hardware Stack

Vivienne Sze http://sze.mit.edu/
Co-Design Across Hardware Stack

In collaboration with

Joel Emer  Tanner Andrusis  Nellie Wu  Fisher Xue  Angshu Parashar  Po-An Tsai

Vivienne Sze  http://sze.mit.edu/
Modeling for Design Space Exploration

Modeling Sparse Tensor Accelerators

Step 1: Dataflow Modeling
- Workload
- Mapping

Step 2: Sparse Modeling
- Sparseloop
- Sparse traffic stats
- Sparse Acceleration Features
- Architecture

Step 3: Micro-Architectural Modeling
- Global Buffer (GLB)
- PE0, PE1, PE2, PE3

Dense traffic stats

Energy
Cycles

Accelerergy
- Storage
- Compute logic
- Analog logic

Design Accelerators

Multiplication of Fractions
- Rank 1
- Rank 0
- Sparsity degrees

Sparisity Degree Spectrum
- 12 sparsity degrees

HighLight [Wu, MICRO 2023]

Tailors [Xue, MICRO 2023]

[Vivienne Sze, http://sze.mit.edu/]
## Energy Dominated by Data Movement

<table>
<thead>
<tr>
<th>Operation:</th>
<th>Energy (pJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8b Add</td>
<td>0.03</td>
</tr>
<tr>
<td>16b Add</td>
<td>0.05</td>
</tr>
<tr>
<td>32b Add</td>
<td>0.1</td>
</tr>
<tr>
<td>16b FP Add</td>
<td>0.4</td>
</tr>
<tr>
<td>32b FP Add</td>
<td>0.9</td>
</tr>
<tr>
<td>8b Multiply</td>
<td>0.2</td>
</tr>
<tr>
<td>32b Multiply</td>
<td>3.1</td>
</tr>
<tr>
<td>16b FP Multiply</td>
<td>1.1</td>
</tr>
<tr>
<td>32b FP Multiply</td>
<td>3.7</td>
</tr>
<tr>
<td>32b SRAM Read (8KB)</td>
<td>5</td>
</tr>
<tr>
<td>32b DRAM Read</td>
<td>640</td>
</tr>
</tbody>
</table>

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

[Horowitz, ISSCC 2014]
Compute In Memory (CiM) Accelerators

Conventional

Input Memory → Weight Memory → Compute → Output Memory

Compute In Memory

Inputs → Weight Memory + Computing → Computed Results
Lots of Compute In Memory Research Across the Stack

Vivienne Sze
http://sze.mit.edu/

Lots of Compute In Memory Research Across the Stack

SESSION 34
Wednesday, February 21st, 1:30 PM
Compute-In-Memory

Co-Chairs:
Jaydeep Kulkarni, University of Texas at Austin
Masanori Yamaoka, Hitachi, Ltd.

10:15 AM
C3.1 122.7 TOPS/W Stencil-Based DNN Accelerator Based on Transition Density Data Representation, Clock-Less MAC Operation, Pseudo-Sparisty Exploitation in 40 nm, Animesh Gupta1,
Japesh Vohra2, Viveka Konandur Rajan3, Massimo Alioto4 National University of Singapore

A DNN whose activation magnitude is represented by digital transition rate is introduced for low energy, under the proposed Odyssey Digital Transition Modulation (DITM); MAC operations are simplified into transition counting, enabling 1) activation pseudo-sparisty for lower energy, 2) clock-less neuron operation via simple up-down asynchrous counters. >100 TOPS/W in 40 nm stddcell design is on ear with

Session 23: Neuromorphic Computing (NC) - Compute-in-Memory for Deep Learning
2:15 PM, Continental 6
Co-Chairs: Martin Frank, IBM and Duygu Kuzum, University of California San Diego

This session describes advances in compute-in-memory (CIM) technologies and 3D integration for deep learning. Such circuits hold promise for deep learning inference and training by enabling faster and more energy-efficient neural network operations than digital CMOS. The session will be opened with a report on an in-memory compute chip fabricated in 14 nm CMOS technology that employs a carbon-based linear underneath the phase-change material to improve temporal stability and inference accuracy. The second paper reports on low-temperature monolithic 3D integration of carbon nanotube FETs and resistive RAM (ReRAM) arrays to enable circuits in memristor for deep neural networks including a 16x16 layer operation and 50x50

ISSCC 2024
VLSI 2024
IEDM 2024
ISCA 2024

Many dedicated sessions on CiM at architecture, circuits, and devices conferences
Compute In Memory (CiM) Accelerators

- Inputs
- Weights
- Outputs

\[
\begin{array}{cccc}
A & B & C \\
D & E & F \\
G & H & I \\
\end{array}
\times
\begin{array}{ccc}
1 & 2 & 3 \\
\alpha & \beta & \gamma \\
\end{array}
= \begin{array}{c}
\alpha \beta \gamma \\
\end{array}
\]
The CiM Stack: Large Design Space
The CiM Stack: Large Design Space

\[ \text{Devices} \times \text{Stack} = \text{Result} \]
The CiM Stack: Large Design Space

\[
\text{Circuits} \times \text{Devices} = \text{Result}
\]
The CiM Stack: Large Design Space

Architecture

Circuits
DAC  ADC  MAC

Devices

\[ \text{Devices} \times \text{Circuits} = \text{Architecture} \]
The CiM Stack: Large Design Space

Workload $\times$ Architecture $=$ Circuits

$\times$ Devices

DAC  ADC  MAC

Vivienne Sze  http://sze.mit.edu/
The CiM Stack: Large Design Space

Workload \( \times \) Architecture = Devices

Mapping

Circuits

DAC
ADC
MAC

Vivienne Sze [http://sze.mit.edu/](http://sze.mit.edu/)
The CiM Stack: Large Design Space

Need for modeling tool to enable design space exploration → CiMLoop
CiMLoop: A Flexible, Accurate, and Fast CiM Modeling Tool

Inputs
- Workload
- Architecture

Circuits
- DAC
- ADC
- MAC

Devices

Data-Distribution-Dependent Component Model
Data Distribution Calculation
Data Distributions

Component Model Library

Timeloop + Accelery
Mapping

Full-Stack Model

Outputs
- System Energy,
- Area,
- Throughput

Code available at
https://emze.csail.mit.edu/cimloop

Vivienne Sze http://sze.mit.edu/

[Andrulis, ISPASS 2024] ISPASS Best Paper Award
CiMLoop: A Flexible, Accurate, and Fast CiM Modeling Tool

• Flexibility to represent co-design space
  – **Challenge:** There are diverse choices at each level
  – **Solution:** Flexible user-defined specifications

[Andrulis, ISPASS 2024]
The Co-Design Space: Components

[Kim, JSSC 2021] [Sinangil, JSSC 2021] [Shiflett, ISCA 2021] [Wan, Nature 2022] [Jia, JSSC 2020] [Wang, VLSI 2022]
The Co-Design Space: Components

- Eight-way multiplexed NMC data path
- Pulsetrain DAC
- C-2C Multiplier
- 4b Flash ADC
- Asynchronous SAR ADC
- Weighting Capacitors
- Arrayed Waveguide Grating
- SL drivers
- Drivers and Registers
- 6T SRAM
- 8T SRAM
- 9T SRAM
- 4b Flash ADC
- Asynchronous SAR ADC
- R-2R DAC
- C-2C Multiplier
- M-ZM
- Star Coupler
- Photodiodes
- Analog Sample + Integrator
- Post Accumulator
- SRAM + Digital MAC
- Post Accumulator
- 9T SRAM
- 8T SRAM
- 6T SRAM

References:
- [Kim, JSSC 2021]
- [Sinangil, JSSC 2021]
- [Shiflett, ISCA 2021]
- [Wan, Nature 2022]
- [Jia, JSSC 2020]
- [Wang, VLSI 2022]

[Andrulis, ISPASS 2024]
The Co-Design Space: Components

Library of circuit and device models
+ Plug-in interface for users to create more models
CiMLoop: A Flexible, Accurate, and Fast CiM Modeling Tool

• Flexibility to represent co-design space
  – Challenge: There are diverse choices at each level
  – Solution: Flexible user-defined specifications

• Accurately model energy
  – Challenge: Workload values and representation affect component energy
  – Solution: Models capture these cross-stack interactions (error within 8%)
Data-value-dependence significantly impacts device and circuit energy. Prior works assume fixed energy → significant error.

Energy = 3fJ  Energy = 1fJ

$Voltage^2 \times Conductance \times Time$

$Conductance \; \mu \Omega = \frac{1}{Resistance \; \Omega}$
Accurately Modeling Energy: Data-Value-Dependence

What value are we processing?

How do we represent it?

Where do we map bits of this value?

Many encodings possible! Unsigned, differential, XNOR, 2’s comp...

Partition bits across components

What values are there? How do we represent them? Where do we map their bits?

\[ \text{Energy} = 3fJ \quad \text{Energy} = 1fJ \]

Capture data-value-dependence:

\[ \text{Energy} = \text{Voltage}^2 \times \text{Conductance} \times \text{Time} \]

\[ \text{Conductance} \frac{1}{\Omega} = \frac{1}{\text{Resistance} \Omega} \]
CiMLoop: A Flexible, Accurate, and Fast CiM Modeling Tool

• Flexibility to represent co-design space
  – Challenge: There are diverse choices at each level
  – Solution: Flexible user-defined specifications

• Accurately model energy
  – Challenge: Workload values and representation affect energy
  – Solution: Models capture these cross-stack interactions (error within 8%)

• Fast exploration of co-design space
  – Challenge: Accurate energy models may simulate many (>10^{12}) values
  – Solution: Statistical models that are 1000x faster than prior accurate models

[Andrulis, ISPASS 2024]
Fast and Accurate Statistical Energy Modeling

<table>
<thead>
<tr>
<th></th>
<th>Data-Value-Independent</th>
<th>Data-Value-Dependent</th>
<th>Data-Distribution-Dependent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Speed</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Model Accuracy</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Parashar, ISPASS 2019
Peng, TCAD 2021
Andrulis, ISPASS 2024

Average Energy Error (Relative to Data-Value-Dependent)

- Data-Value-Independent
- Data-Value-Dependent
- Data-Distribution-Dependent

Model Speed (Normalized)

- Greater than 1000x faster (use statistics)
- Same speed, 10x lower error (capture data dependency)

Vivienne Sze  http://sze.mit.edu/

[Andrulis, ISPASS 2024]
Many similarities in design of CiM and Photonic accelerators → Can model with CiMLoop!

[Andrilis, ISPASS 2024]
CiMLoop Enabling Collaborations Across Stack at MIT

Computing in memory with programmable resistive devices [w/ Jesus del Alamo]

Computing with light [w/ Dirk Englund]

Modeling helps identify level of abstraction and facilitates communication between teams
Key Takeaways for Co-Design Across Hardware Stack

• Modeling used for design-space exploration in computer architecture community

• Extend modeling to support circuits and emerging devices to broaden design space and enable co-design across hardware stack

• Modeling can be used to enable collaboration across different research communities to increase impact of research
Co-Design Algorithm and Hardware

In collaboration with

Sertac Karaman  Keshav Gupta  Theia Henderson  Peter Li  Tien-Ju Yang  Zhengdong Zhang
Design Hardware-Aware DNN Algorithms

**Energy-Aware Pruning**
Remove weights based on energy consumption

- Normalized Energy (AlexNet)

- Original
- Magnitude Based Pruning
- Energy Aware Pruning

- 2.1x
- 3.7x

**NetAdapt: Platform-Aware DNN Adaptation**
Automatically adapt DNN to reach target latency/energy (hardware-in-the-loop)

**Rethink DNN design for CiM**
more weights, less layers

- Pretrained Network
- Adapted Network
- Measured
- Platform

- Empirical Measurements
  - Metric: Latency, Energy
  - Budget: Proposal A, Z

- Metric: Latency, Energy
  - Budget: Proposed A, Z

- Latency
  - 3.8
  - 10.5

- Energy
  - 41
  - 46

- Storage Element
  - R x S x C

Use hardware properties to drive the design of DNN workloads

[Vivienne Sze](http://sze.mit.edu/)

- [Yang, CVPR 2017]
- [Yang, ECCV 2018]
- [Yang, IEDM 2019]
“HD Maps”, deemed essential by some AV companies, is significant part of compute costs both in the data center and at the edge.

These maps are utilized extensively for localization, navigation, obstacle detection, ...

Need to build and store map while delivering real-time performance within energy constraints.
Robots Consuming < 1 Watt for Actuation

- **31 mW**
  - Robobee (2019)
  - Source: Harvard

- **13.5 mW**
  - Robotic Water Strider (2015)
  - Source: Seoul Nat’l University

- **500 mW**
  - Seaglider (2003)
  - Source: Georgia Tech

- **132 mW**
  - Chipsat (2016)
  - Source: Kongsberg

- **50 mW**
  - Robofly (2020)
  - Source: University of Washington

**Low Energy Robotics**
- Miniature aerial vehicles
- Lighter than air vehicles
- Micro unmanned gliders
- Miniature satellites

Vivienne Sze [http://sze.mit.edu/](http://sze.mit.edu/)
Building an Occupancy Map is a Key Task in Autonomy

- An occupancy map indicates the probability of an obstacle at a given location
- The map is initialized at 0.5 (unknown)
- To build map, robot scans surrounding with depth sensor (e.g., LiDAR)

Image Source: Velodyne Lidar
Building an Occupancy Map is a Key Task in Autonomy

- An occupancy map indicates the probability of an obstacle at a given location
- The map is initialized at 0.5 (unknown)
- To build map, robot scans surrounding with depth sensor (e.g., LiDAR)
  - Occupancy value approaches 0 (free space) or 1 (occupied)
- Map is probabilistic since there is sensor noise

Image Source: Velodyne Lidar
Building an Occupancy Map is a Key Task in Autonomy

Robot needs to **build** and **store** map
How do we select the next location to move to?

Determine where robot should go to get most new information about environment
Robot Exploration: Build Map of Unknown Environment

• Use Shannon **Mutual Information (MI)**
  – Indicates amount of **potential information gain** at a location
  – Provides **guarantees (provably next best location)** [Julian, *IJRR* 2014]
  – Accounts for depth sensor noise

Challenge: MI not feasible in real-time (several orders of magnitude too slow!)
FSMI: Fast Shannon Mutual Information

Compute Mutual Information obtained for each ray (depth sensor measurement)

Add up MI from each cell that the ray traverses

Original MI [Julian, IJRR 2014] compute MI from each cell one at a time
→ requires numerical integration at resolution $\lambda_z$. $O(n^2 \lambda_z)$

FSMI [Zhang, IJRR 2014] computes MI for all cells in entire ray altogether
→ removes numerical integration. $O(n^2)$

Analogy (Arithmetic Series): $1 + 2 + 3 + \cdots + n$

If compute all at once, then $\frac{n(n+1)}{2}$

FSMI is ~1000x faster than original MI
Experimental Results (Video Playback is 4x Real Time)

Exploration with a mini race car using motion capture for localization

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

MI surface

[Zhang, *IJRR* 2020]
Hardware Accelerator for FSMI

MI for each ray direction can be computed independently [Julian, *IJRR* 2014]. High throughput should be possible with multiple parallel processing elements (PEs).

Compute MI of each ray in parallel with multiple PEs

FSMI shifts bottleneck from compute to memory
Challenge is Data Delivery to All Processing Elements

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 Banking for Ray Access Pattern

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

Specialized banking results in throughput within 6% of theoretical limit (unlimited bandwidth)

[Li, RSS 2019]
FCMI: Fast Continuous Mutual Information

[Henderson, ICRA 2020]
FCMI: Fast Continuous Mutual Information

[Henderson, ICRA 2020]
FCMI: Fast Continuous Mutual Information

[Henderson, ICRA 2020]
FCMI: Fast Continuous Mutual Information

[Henderson, ICRA 2020]
FCMI: Fast Continuous Mutual Information

[Henderson, ICRA 2020]
FCMI: Fast Continuous Mutual Information

[Henderson, ICRA 2020]
FCMI: Fast Continuous Mutual Information

[Henderson, ICRA 2020]
FCMI: Fast Continuous Mutual Information

[Henderson, ICRA 2020]
Reformulate algorithm to allow MI at different locations to \textit{recursively reuse computation}.

Replace \textbf{multiple rays} in the same direction with a \textbf{single ray} that crosses entire map.
Reformulate algorithm to allow MI at different locations to **recursively reuse computation**

Replace **multiple rays** in the same direction with a **single ray** that crosses entire map
Reformulate algorithm to allow MI at different locations to recursively reuse computation

Replace multiple rays in the same direction with a single ray that crosses entire map

Exploit this recursive structure across entire map
**FCMI: Fast Continuous Mutual Information**

Compute MI at all locations in map

Only need **one ray** to intersect each cell in each direction

$n = \text{cells per ray}$
$L = \text{number of rays}$
$H^2 = \text{size of map}$

FSMI: $O(nLH^2)$  $→$  FCMI: $O(LH^2)$

*Two orders of magnitude speed up over FSMI!*
Hardware Accelerator for FCMI

• Time-interleave independent rays to handle recursive dependencies without stalls

• Diagonal banking to support parallel processing of rays

Vivienne Sze [http://sze.mit.edu/]

[Gupta, IROS 2021]
Several Orders of Magnitude Speed up Via Co-Design

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

- **Optimize memory subsystem (banking)** for multi-beam parallel processing
- **Reformulate** using a continuous occupancy map framework and exploit recursive structure
- **Evaluate MI for all cells in entire ray altogether** removes numerical integration
- **Optimize memory subsystem**, time-interleave PEs and approximate computing

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

[Julian, IJRR 2014] [Zhang, ICRA 2019] [Li, RSS 2019] [Henderson, ICRA 2020] [Gupta, IROS 2021]
Reduce Storage Cost of Map

Existing Works

- Low Compactness
- Multi-Pass Input Processing
- Large Memory Overhead During Computation

Gaussian-Based Representation

- Compact Gaussian Representation
- Single Pass Input Processing
- Reduced Memory Overhead Using Gaussian Operations
Compact 3D Representation Using Gaussians

2D Depth Images (thousands)

Point Cloud

Reduce by 10x

Voxel-based 3D Map (red cubes)

GMMap give 16x reduction in size at comparable accuracy, but Gaussian construction is usually not memory efficient!

3D Gaussians in GMMap (red ellipsoids)

Reduce by 16x
SPGF: Single-Pass Depth Image Compression

Compute Gaussians directly from 2D depth image to exploit inherent structure to enable a single-pass row-by-row based approach.

Reduces memory overhead by 50x compared to prior multi-pass approaches that construct from point cloud.

[Li, ICRA 2022]
GMMap: Fuse Multiple Depth Images to form 3D Map

Fuse multiple depth images directly using Gaussians rather than point cloud reduces memory overhead by 8x

[Li, T-RO 2024]
Key Takeaways for Co-Design Algorithms and Hardware

• There is a limit to improvement that can be achieved from hardware optimization alone → algorithm design critical to close the gap

• Hardware can influence algorithm design → exploit hardware capabilities (e.g., parallelism, reuse) and avoid hardware bottlenecks (e.g., memory)

• Algorithm design can open up opportunities for architects by shifting the bottleneck
Co-Design Across System

In collaboration with

Sertac Karaman  Fangchang Ma  James Noraky  Soumya Sudhakar  Diana Wofk  Tien-Ju Yang

Vivienne Sze  http://sze.mit.edu/
Low Power 3D Time of Flight Imaging

- Pulsed Time of Flight: Measure distance using round trip time of laser light for each image pixel
  - Illumination + Imager Power: 2.5 – 20 W for range from 1 - 8 m

- Use computer vision techniques and passive images to estimate changes in depth without turning on laser
  - CMOS Imaging Sensor Power: < 350 mW

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

[Noraky, TCSVT 2019]
Results of Low Power Depth ToF Imaging

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

[Noraky, TCSVT 2019]
FastDepth: Fast Monocular Depth Estimation

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

~40fps on an iPhone

Speed up from algorithm design
(NetAdapt, compact network design, and depth wise decomposition)

Configuration: Batch size of one (32-bit float)
Need to estimate uncertainty (sensor noise model) for robot decision making.

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

[Sudhakar, ICRA 2022]
Exploit **temporal redundancy** in video inputs by merging outputs that belong to the same point in 3D space across multiple views to estimate uncertainty

Prior work run multiple inferences per frame

UfM runs one inference per frame

[Sudhakar, ICRA 2022]
Key Takeaways for Co-Design Across System

• Expand design space to include choice of energy allocation across compute, sensing and actuation (for robotics)

• Use efficient computing to reduce energy consumption of other components in system → *magnify savings beyond compute*
• Use modeling to bridge architects with other research communities

• Use algorithm design to open new opportunities for computer architects

• Use co-design to extend the impact of computer architecture research

Collaborate, Co-Design and get out of comfort zone!
Acknowledgements

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

Emze Group
Co-Led by Joel Emer and Vivienne Sze

Overview
We explore the modeling and design of efficient and flexible hardware accelerators.

PhD Students
- Tanner Andrulis
- Michael Gilbert
- Fisher Zi Yu Xue
- Yu-Hsin Chan (Alumni)
- Yirong Nelle Wu (Alumni)

Publications
* Indicates authors contributed equally to the work

Architecture Modeling for evaluation and design space exploration
- T. Andrulis, J. S. Emer, V. Sze, "CMUloop: A Flexible, Accurate, and Fast Compute-In-Memory Modeling Tool," *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, May 2024. [paper LINK] [project website LINK] [code github] [Best Paper Award]

Low-Energy Autonomy and Navigation (LEAN) Group

A broad range of next-generation applications will be enabled by low-energy, miniature mobile robotics including in-ship-sized flapping wing robots that can help with search and rescue, ship-size satellites that can explore nearby stars, and blimps that can stay in the air for years to provide communication services to remote locations. While the low-energy, miniature actuators, 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 by bringing together the robot 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 conjuction 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 the 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 miniature robots (e.g., nanosatellites and 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.

http://emze.mit.edu

http://lean.mit.edu

Vivienne Sze http://sze.mit.edu/
References: Co-Design Across Hardware Stack


• Z. Y. Xue, Y. N. Wu, J. S. Emer, V. Sze, “Tailors: Accelerating Sparse Tensor Algebra by Overbooking Buffer Occupancy,” MICRO 2023

• T. Andrulis, J. S. Emer, V. Sze, “CiMLoop: A Flexible, Accurate, and Fast Compute-In-Memory Modeling Tool,” ISPASS 2024

References: Co-Design Algorithms and Hardware


• P. Z. X. Li, S. Karaman, V. Sze, “GMMap: Memory-Efficient Continuous Occupancy Map Using Gaussian Mixture Model,” IEEE Transactions on Robotics (T-RO), 2024
References: Co-Design Across System


• S. Sudhakar, V. Sze, S. Karaman, “Uncertainty from Motion for DNN Monocular Depth Estimation,” *ICRA 2022.*