Efficient Computing for Autonomous Navigation using Algorithm-and-Hardware Co-design

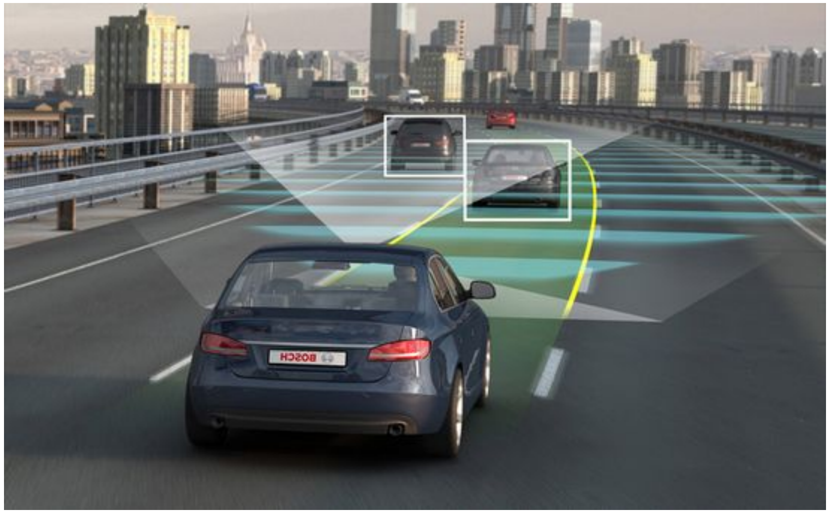
Zhengdong Zhang

Committee:
- Prof. Vivienne Sze (thesis advisor)
- Prof. Sertac Karaman
- Prof. Luca Carlone
- Prof. Phillip Isola

Feb 11, 2019

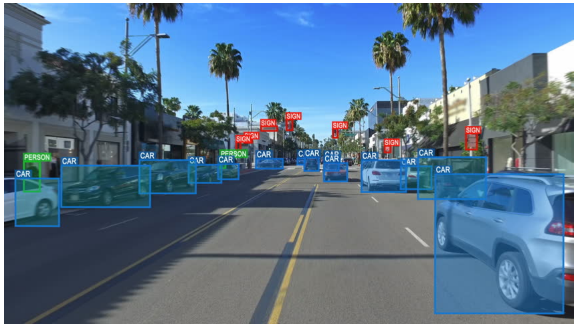
Autonomous Navigation

Disaster response



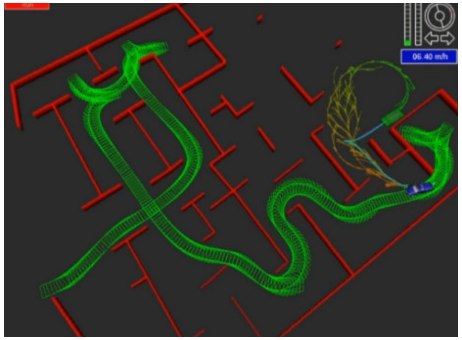
Self driving cars

Key Techniques

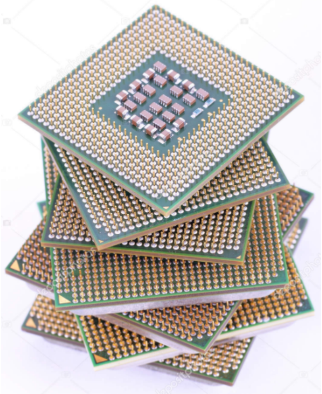


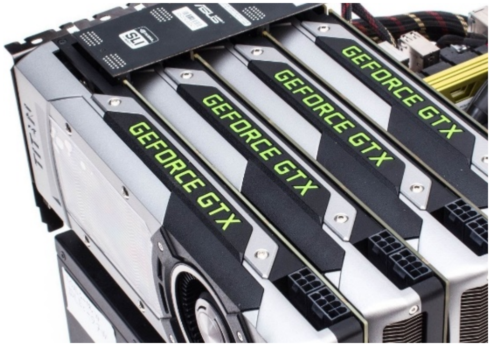
Visual Perception

Localization



Mapping





COMPUTATIONAL POWER

Goal

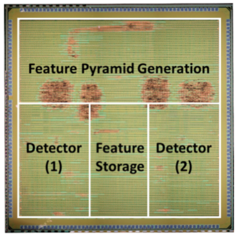
Run algorithms for autonomous navigation **LOCALLY**

Low delay

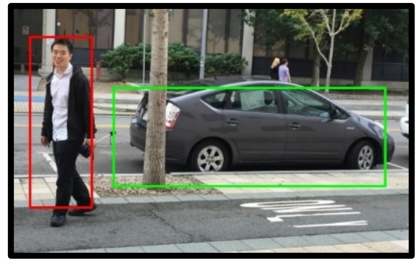
High speed

Energy efficiency

Thesis Contributions

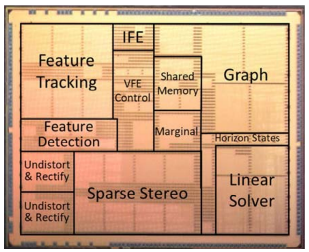


Object detection chip



Detection on full HD videos at 30 fps

In collaboration with Amr Suleiman



Visual Inertial Odometry (VIO) chip

Localization and mapping for miniature drones

In collaboration with Amr Suleiman

Information theoretic mapping algorithms and system on FPGA

Autonomous exploration of the environment

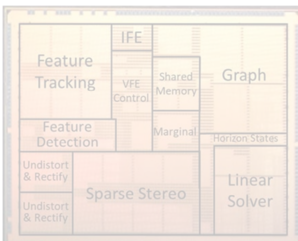
In collaboration with Trevor Henderson and Peter Li

Thesis Contributions

Object detection chip

Detection on full HD videos at 30 fps

In collaboration with Amr Suleiman



Visual Inertial odometry (VIO) chip

Localization and mapping for miniature drones

In collaboration with Amr Suleiman

Information theoretic mapping algorithms and system on FPGA

Autonomous exploration of the environment

In collaboration with Trevor Henderson and Peter Li

Navigation in Unknown Environment

Car for building Google Map

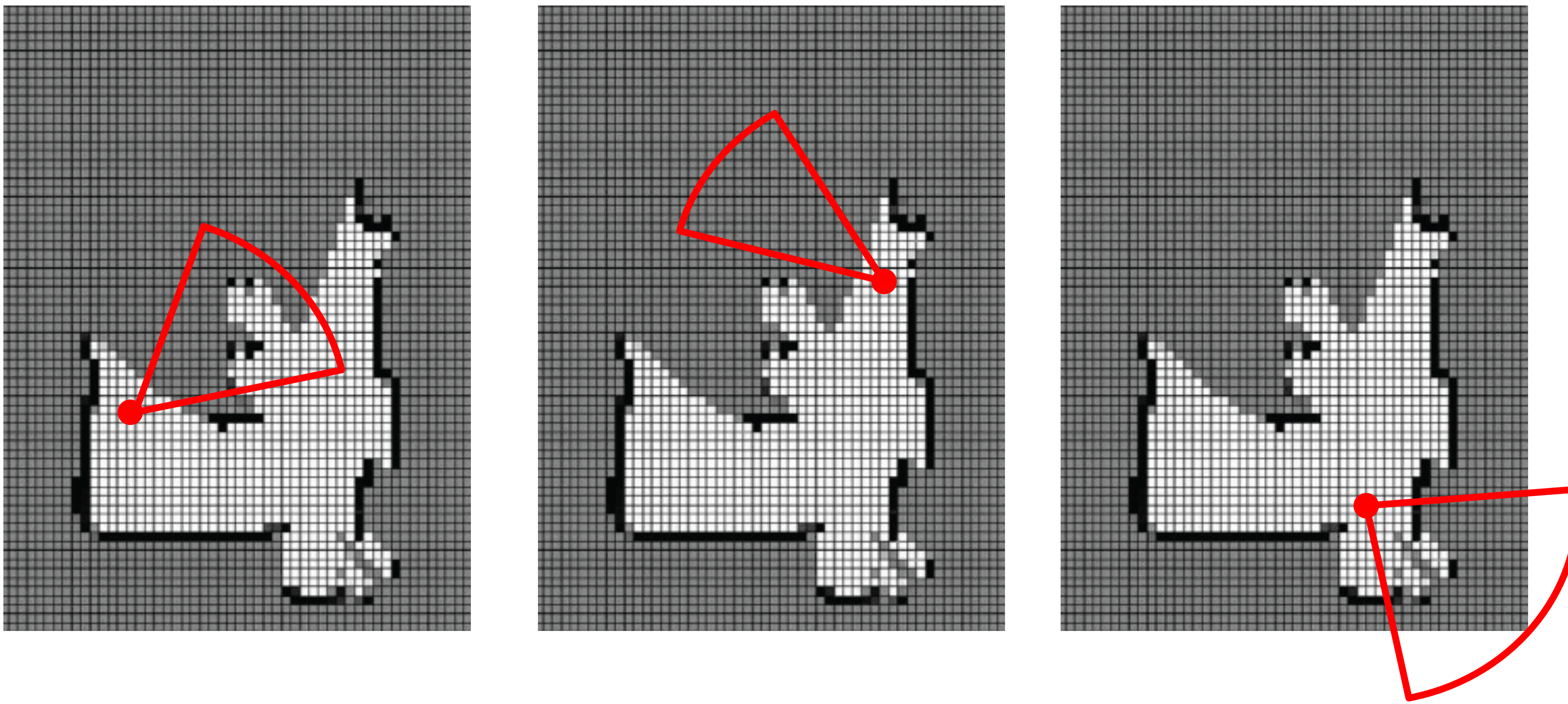
Drones

Build the map of an unknown environment with **depth sensor** scans

Time Sensitive Mapping Tasks

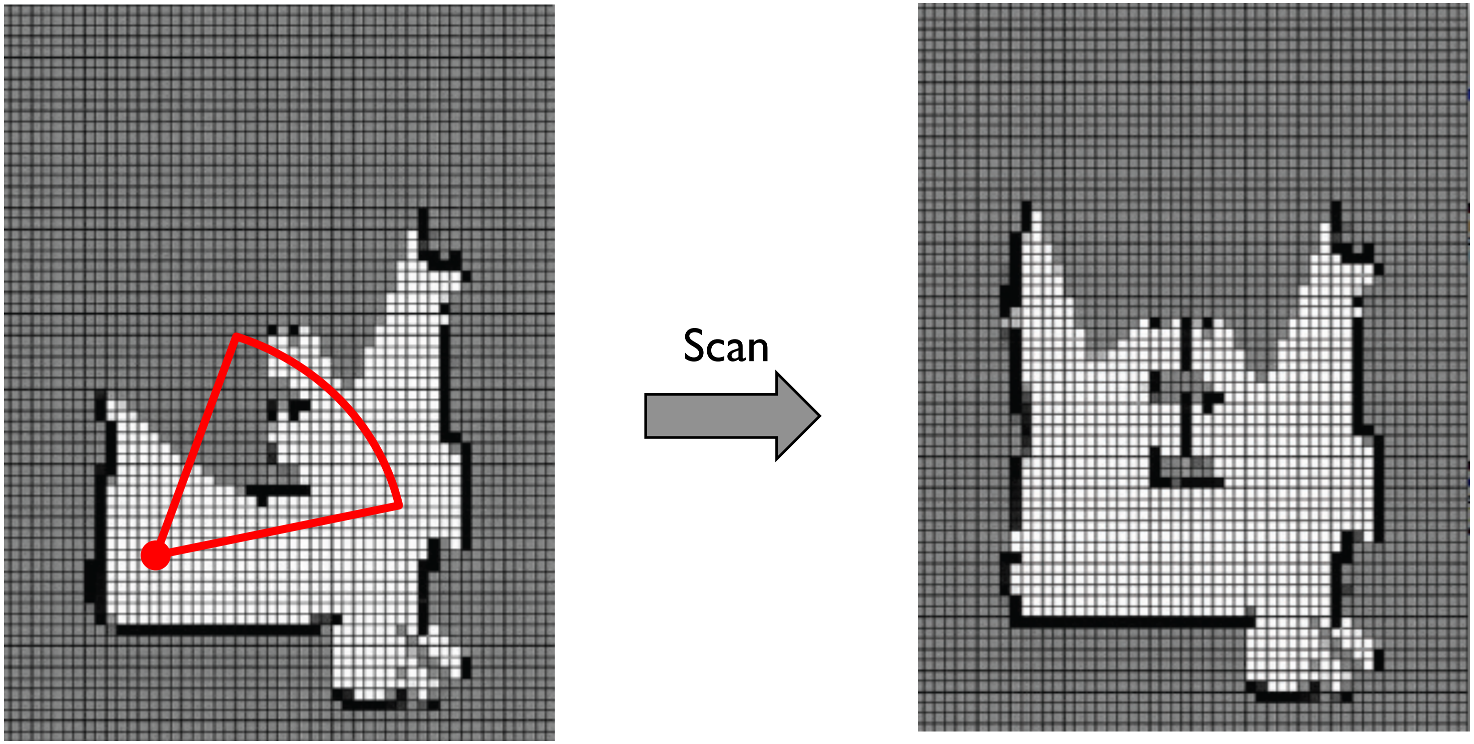
Need to explore and map the environment fast!

Where to Scan



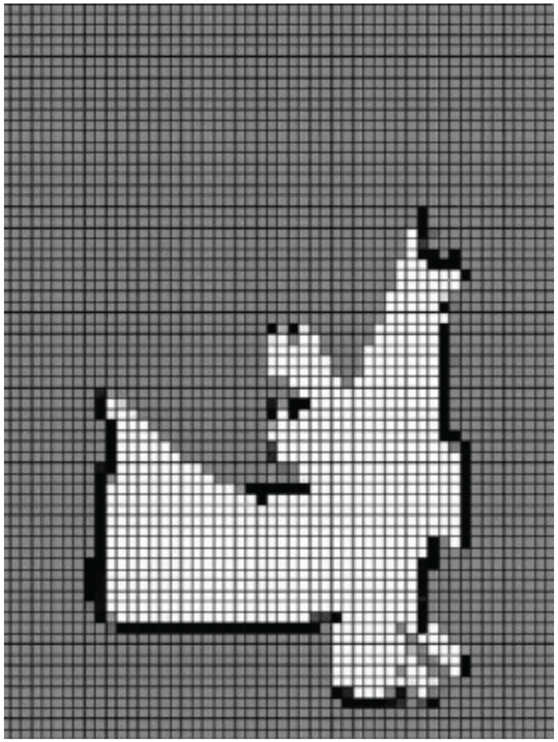
Different scan location sequences impact mapping time

Information Theoretic Mapping

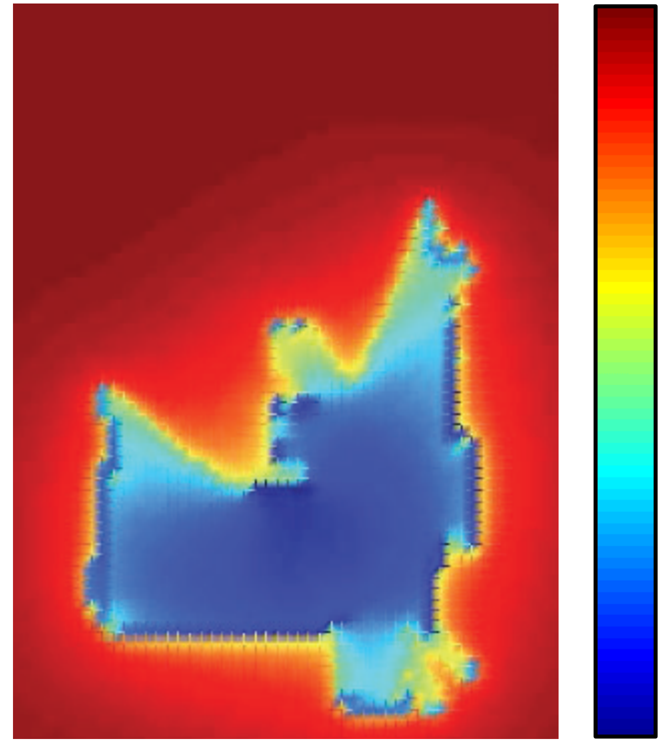


A **probabilistic framework** to model the mapping problem as the fastest **reduction of the uncertainty of the map**

Information Theoretic Mapping



Occupancy grid map, M

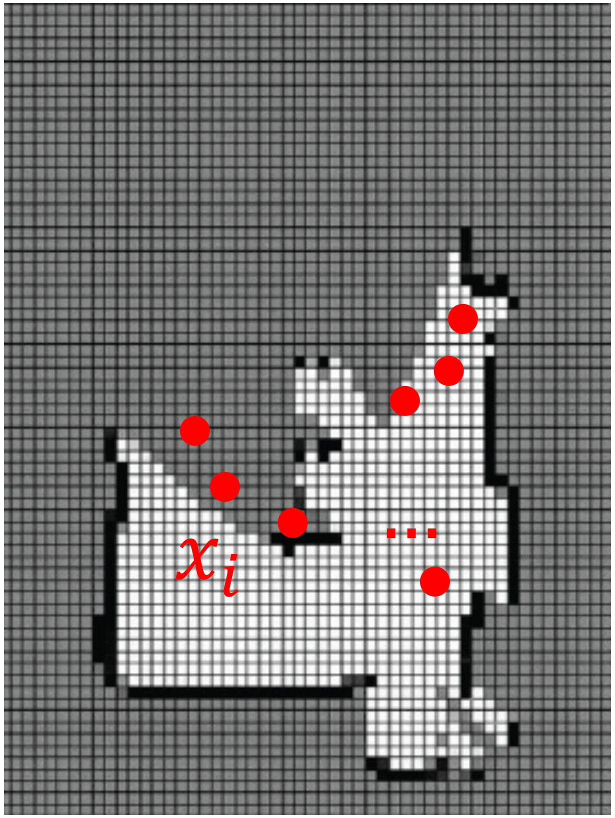


Mutual information map, $I(M;Z)$

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

Perspective updated map entropy — Current map entropy — Mutual information

Information Theoretic Mapping



Generate candidate scan locations $\mathcal{X}$

For $x_i \in \mathcal{X}$:
Evaluate $\boldsymbol{I}(\boldsymbol{M};\boldsymbol{Z}(\boldsymbol{x_i}))$

Find $x^* \leftarrow \text{argmax}\, \boldsymbol{I}(\boldsymbol{M};\boldsymbol{Z}(\boldsymbol{x_i}))/\boldsymbol{dist}(\boldsymbol{x_i})$

Vehicle moves to x^*

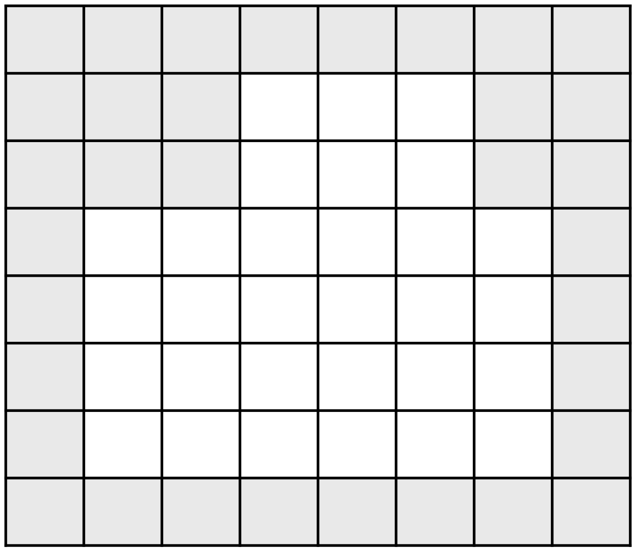
Vehicle scans at x^*

Update the occupancy map

Require evaluation of $\boldsymbol{I}(\boldsymbol{M};\boldsymbol{Z})$ at multiple locations
Bottleneck of the entire pipeline

Challenge 1: High Complexity

Map resolution λ_m: number of cells per meter



Map

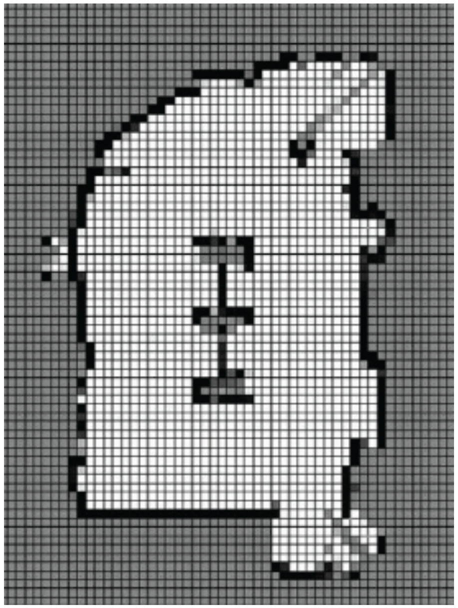
$$I(M_i; Z) = \boxed{\int_{z \geq 0} P(Z = z) f(\delta_i(z), r_i) dz}$$

Mutual Information

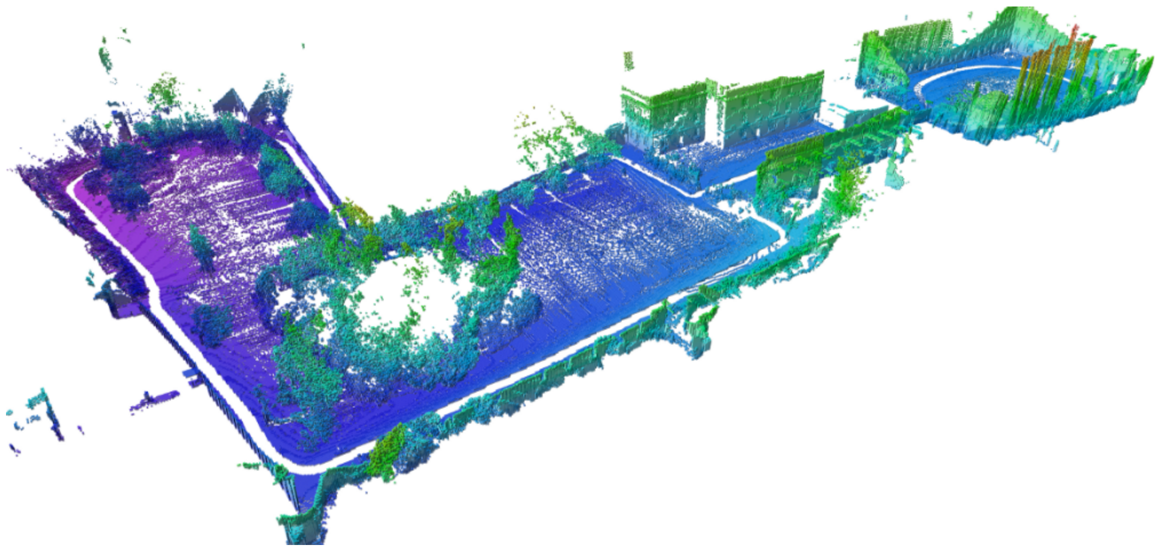
Expensive numerical integration at resolution λ_z

Time complexity of $O(\lambda_m^2 \lambda_z)$

Challenge 2: Amount of Input Data



2D map



3D map

A basic system needs to process huge amount of occupancy cells, each with complexity $O(\lambda_m^2 \lambda_z)$.

Solutions Presented in This Thesis

$$O(\lambda_m^2 \lambda_z) \rightarrow O(\lambda_m)$$

Better algorithm with lower complexity

Computation on the compressed 3D map

Specialized hardware for high-throughput, energy-efficient computation

Contributions

- **Better algorithm with lower complexity**
  - Exact FSMI algorithm
  - Approximation via noise Truncation
  - Even faster algorithm with uniform noise
  - Efficient implementation via look-up table and pre-computation
- **Computation on the compressed 3D map**
  - FSMI algorithm on the compressed OctoMap
  - Closed-form solution to the problem
  - Look-up table size reduction via noise approximation
  - Look-up table split and reconstruction
  - Algorithm for uniform case
- **Specialized hardware for high-throughput, energy-efficient computation**
  - Novel memory banking to minimize the collisions among multi cores
  - Packing multiple entries into one memory address
  - High throughput circuit to compute Shannon MI
  - Novel arbiter to resolve the conflicts of memory requests
  - Workload balance mechanism to reduce the overall latency

Contributions

- **Better algorithm with lower complexity**
  - Exact FSMI algorithm
  - Approximation via noise Truncation
  - Even faster algorithm with uniform noise
  - Efficient implementation via look-up table and pre-computation
- **Computation on the compressed 3D map**
  - FSMI algorithm on the compressed OctoMap
  - Closed-form solution to the problem
  - Look-up table size reduction via noise approximation
  - Look-up table split and reconstruction
  - Algorithm for uniform case
- **Specialized hardware for high-throughput, energy-efficient computation**
  - Novel memory banking to minimize the collisions among multi cores
  - Packing multiple entries into one memory address
  - High throughput circuit to compute Shannon MI
  - Novel arbiter to resolve the conflicts of memory requests
  - Workload balance mechanism to reduce the overall latency

Overview of Thesis Defense

$$I(M_i; Z) = \int_{z \geq 0} P(Z = z) f(\delta_i(z), r_i) dz$$

Preliminaries and notations

$$O(\lambda_m^2 \lambda_z) \rightarrow O(\lambda_m)$$

FSMI: Fast computation of Shannon Mutual Information

FSMI on compressed occupancy map for 3D mapping

Dedicated hardware for FSMI

Overview of Thesis Defense

$$I(M_i; Z) = \int_{z \geq 0} P(Z = z) f(\delta_i(z), r_i) dz$$

Preliminaries and notations

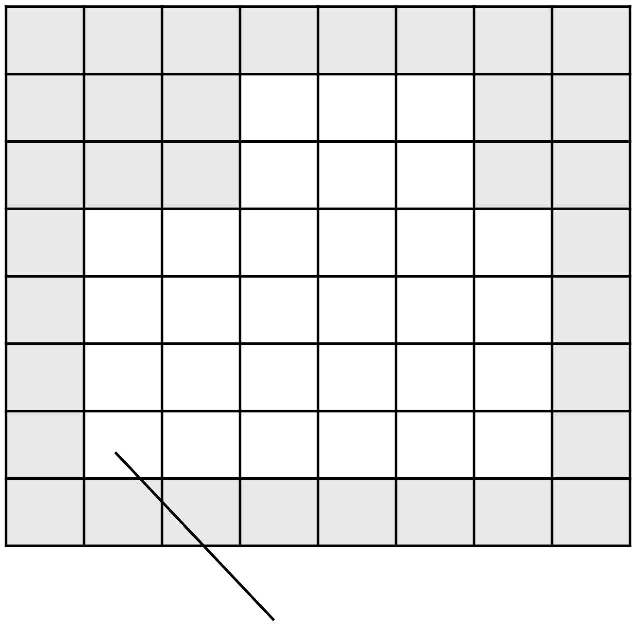
$$O(\lambda_m^2 \lambda_z) \rightarrow O(\lambda_m)$$

FSMI: Fast computation of Shannon Mutual Information

FSMI on compressed occupancy map for 3D mapping

Dedicated hardware for FSMI

Occupancy Grid Map



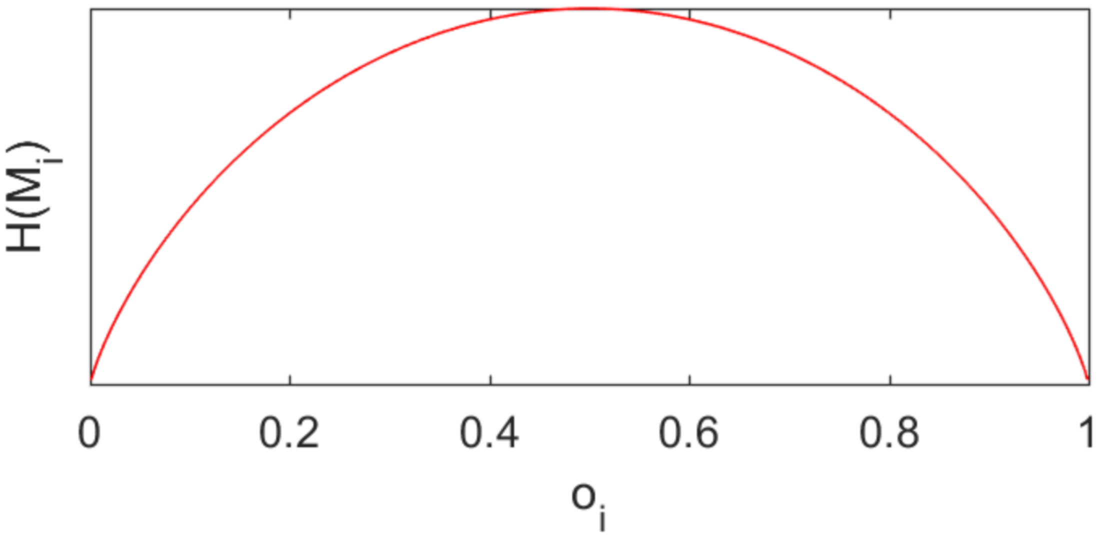
Occupancy cell, M_i

- $M_i \in \{0, 1\}$, 0 for empty, 1 for occupied
- Occupancy value $o_i = \Pr(M_i = 1)$
- Occupied $o_i = 1$, empty $o_i = 0$, unknown $o_i = 0.5$
- Odds ratio $r_i = \frac{\Pr(M_i=1)}{\Pr(M_i=0)} = \frac{o_i}{1-o_i}$, higher indicating the cell is more likely occupied

Map Entropy

Entropy of a cell:

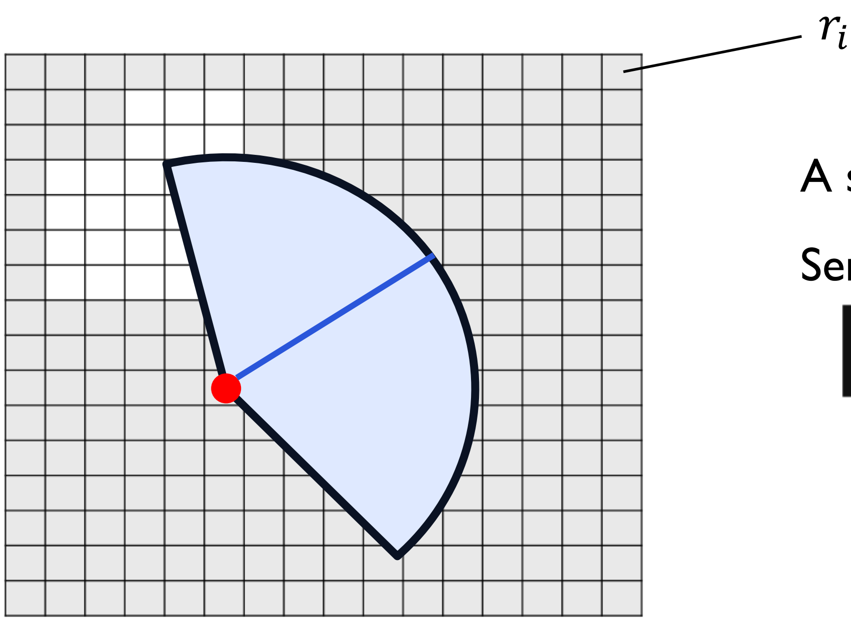
$$H(M_i) = -o_i \log o_i - (1 - o_i) \log(1 - o_i)$$

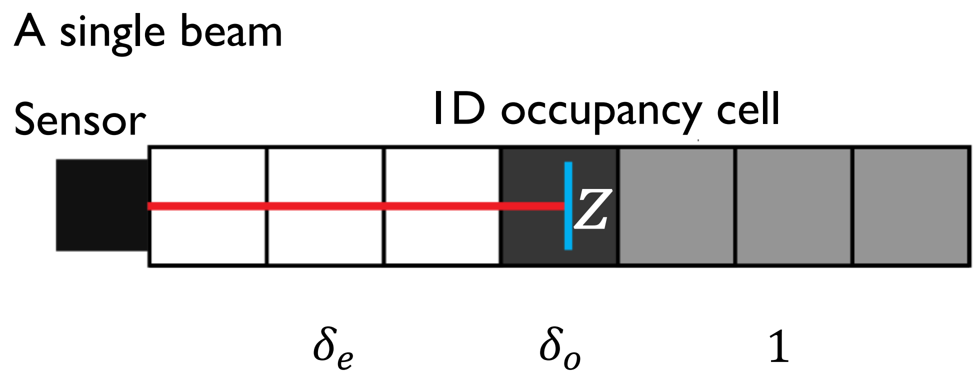


Goal: Reduce the **entropy of the map**

$$\sum_i H(M_i)$$

Update the Occupancy Map via the Bayesian Filter

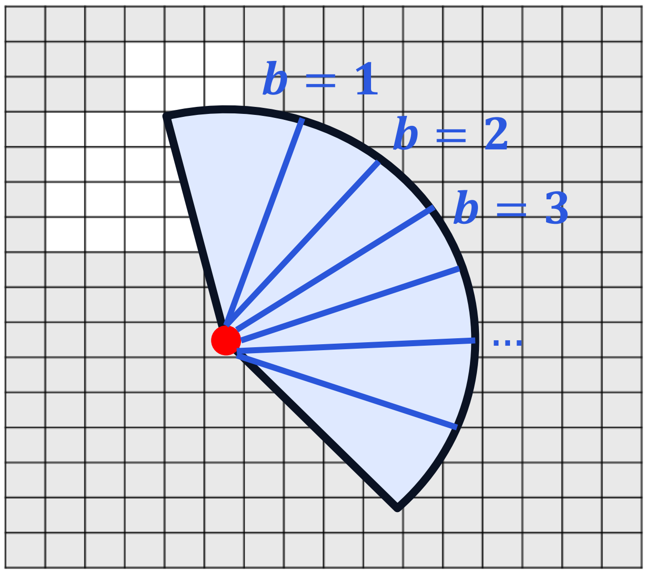




Inverse sensing model
$$\delta_i(z) = \begin{cases} \delta_e < 1 & \text{z indicates cell i empty} \\ \delta_o > 1 & \text{z indicates cell } i \text{ occupied} \\ 1 & \text{otherwise} \end{cases}$$

Bayesian filter
$$r_i^{t+1} = r_i^t \, \delta_i(z)$$

Evaluation of MI



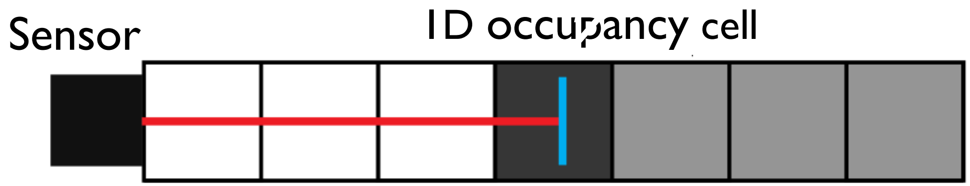
Assumption 1: Beams are independent

$$I(M; Z) = \sum_b I(M; Z_b)$$

$\boxed{\text{Only need to study the evaluation of MI on a single beam}}$

Evaluation of MI on 1D Beam

A single beam



i – index of the cell in the beam

Assumption 2 (Independence among cells on the same beam):

$$I(M;Z) = \sum_{i} I(M_i;Z)$$

Only need to study the evaluation of MI for a single cell

Evaluation of MI at a Single Cell

Theorem (MI at a single cell):

Let the probability for the j-th cell to be the first non-empty cell on the beam

$$P(e_j) = \boldsymbol{o_j} \prod_{i<j} (1 - \boldsymbol{o_i})$$

Let the probability for the perspective depth measurement being z be

$$P(z) = \sum_j P(e_j) \boldsymbol{P(z|e_j)}$$

Let the MI contribution for depth measurement being z to i-th cell

$$f(\delta_i(z), r_i) = \log\left(\frac{r_i + 1}{r_i + \delta_i^{-1}(z)}\right) - \frac{\log \delta_i(z)}{r_i \delta_i(z) + 1}$$

We have

$$I(M_i; Z) = \boxed{\int P(z) f(\delta_i(z), r_i) dz}$$

No known closed-form solution

Numerical Solution

$$I(M_i; Z) = \int P(z) f(\delta_i(z), r_i) dz$$

$$I(M_i; Z) \approx \sum_z P(z) f(\delta_i(z), r_i)\, \lambda_z^{-1}$$

$$I(M; Z) = \sum_i I(M_i; Z)$$

$n = \text{Beam Length} \times \lambda_m$ – number of cells on the beam

Time Complexity $\boldsymbol{O(n^2 \lambda_z)}$ **or** $\boldsymbol{O(\lambda_m^2 \lambda_z)}$

Content

$$I(M_i; Z) = \int_{z \geq 0} P(Z = z) f(\delta_i(z), r_i) dz$$

Preliminaries and notations

$$O(\lambda_m^2 \lambda_z) \rightarrow O(\lambda_m)$$

FSMI: Fast computation of Shannon Mutual Information

FSMI on compressed occupancy map for 3D mapping

Dedicated hardware for FSMI

Original Shannon MI Algorithm

For $i = 1$ To n:

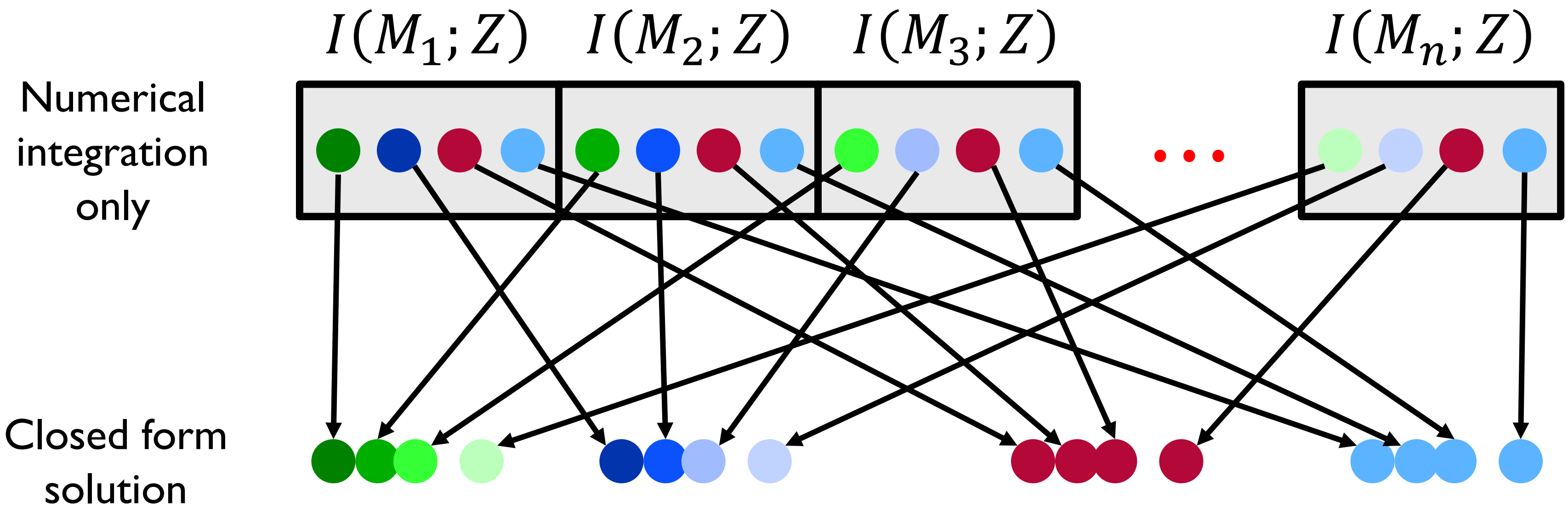
Evaluate $I(M_i; Z) \leftarrow \sum_z P(z) f(\delta_i(z), r_i)\, \lambda_z^{-1}$

Evaluate

$$I(M; Z) = \sum_i I(M_i; Z)$$

Evaluate $I(M_i; Z)$ **one-by-one and then sum up**

Fast computation of Shannon MI (FSMI)



FSMI: Evaluate the MI for all the cells in an entire beam altogether

Derivation of the FSMI Algorithm

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

$$P(z) = \sum_{j} P(e_j) P(z|e_j)$$

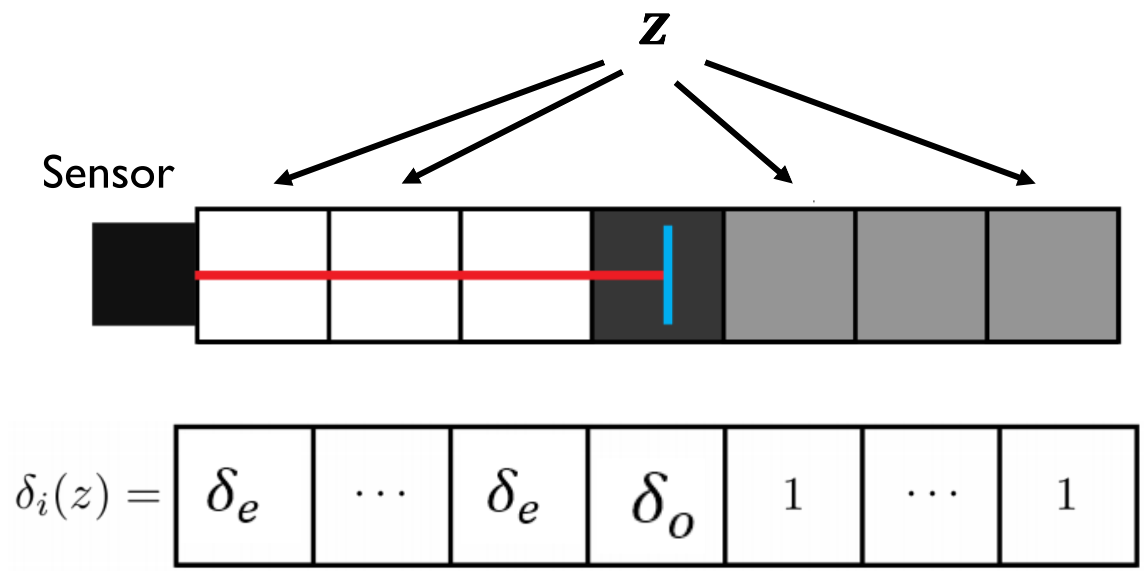
$$I(M;Z) = \sum_{i=1}^{n} \int_{z\geq 0} \sum_{j=1}^{n} P(e_j) P(z|e_j) f(\delta_i(z), r_i)\,dz$$

Switch order

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

Piecewise Constant MI Contribution

$$I(M;Z) = \sum_{j=1}^{n} P(e_j) \int_{z \geq 0} P(z|e_j) \left(\sum_i \boldsymbol{f(\delta_i(z), r_i)} \right) dz$$



$f(\delta_e, \mathrm{r}_1)$ $f(\delta_e, \mathrm{r}_{\mathrm{k}-1})$ $f(\delta_o, \mathrm{r}_{\mathrm{k}})$ 0 0

$$\sum_i f(\delta_i(z), r_i) = C_k$$ if z falls into the k-th cell

Derivation of the FSMI Algorithm

$$I(M;Z)=\sum_{j=1}^{n}P(e_j)\int_{z\geq 0}P(z|e_j)\left(\sum_i f(\delta_i(z),r_i)\right)dz$$

Break up the integral into summation of multiple integrals over the cell boundary

$$I(M;Z)=\sum_{j=1}^{n}P(e_j)\sum_k\int_{l_k}^{l_{k+1}}P(z|e_j)C_k dz$$

$$I(M;Z)=\sum_{j=1}^{n}P(e_j)\sum_k C_k\boxed{\int_{l_k}^{l_{k+1}}P(z|e_j)dz}$$

CDF function: $\Phi_j(l_{k+1})-\Phi_j(l_k)$

FSMI under Gaussian Noise Model

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

Most important equation of the defense!

where

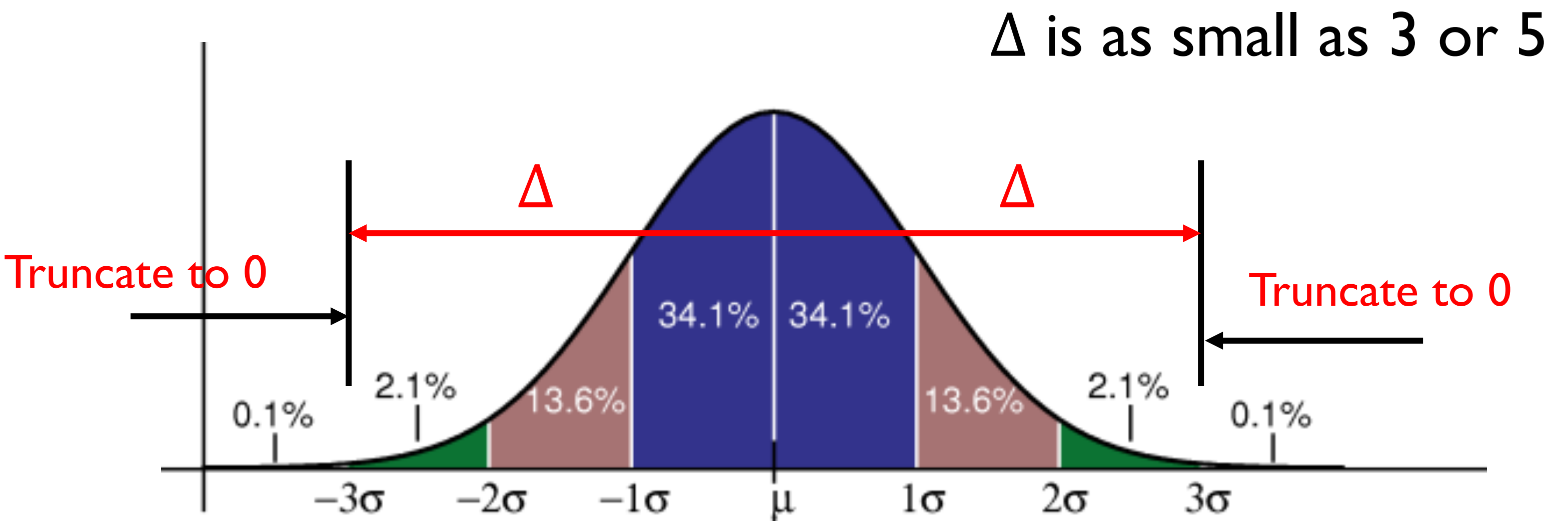
$$P(e_i) = o_i \prod_{j<i} (1 - o_j)\,, \mathrm{i} = 1, \dots, n \qquad O(n)$$

$$C_k = f(\delta_o, r_k) + \sum_{i<k} f(\delta_e, r_i)\,, \mathrm{i} = 1, \dots, n \qquad O(n)$$

$$G_{k,j} = \int_{l_k}^{l_{k+1}} P(z|e_j)\,dz = \Phi_j(l_{k+1}) - \Phi_j(l_k) \qquad O(1)$$

$I(M;Z)$ can be computed *exactly* in $\boldsymbol{O(n^2)}$

Approximation of Noise Model for Depth Sensor



Normal distribution

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

$\boxed{I(M;Z) \text{ can be computed } \textit{approximately} \text{ in } O(n)}$

Charrow et al., Information-theoretic mapping using cauchy-schwarz quadratic mutual information, ICRA 2015

Comparison against Alternative Metrics

Cauchy Schwarz Quadratic Mutual Information (CSQMI)

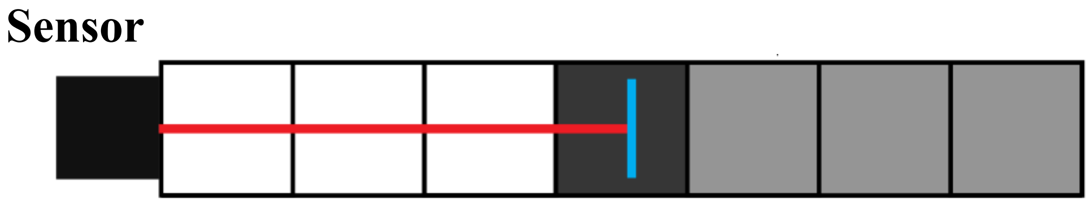
$$I_{CS}(M;Z) = \log\left(\sum_{l=0}^{n} w_l \mathcal{N}(0, 2\sigma^2)\right)$$
$$+ log\left(\prod_{i}^{n}(o_i^2 + (1-o_i)^2)\sum_{j=0}^{n}\sum_{l=0}^{n} P(e_j)P(e_l)\mathcal{N}(\mu_l - \mu_j, 2\sigma^2)\right)$$
$$-2\log\left(\sum_{j=0}^{n}\sum_{l=0}^{n} P(e_j) w_l \mathcal{N}(\mu_l - \mu_j, 2\sigma^2)\right)$$

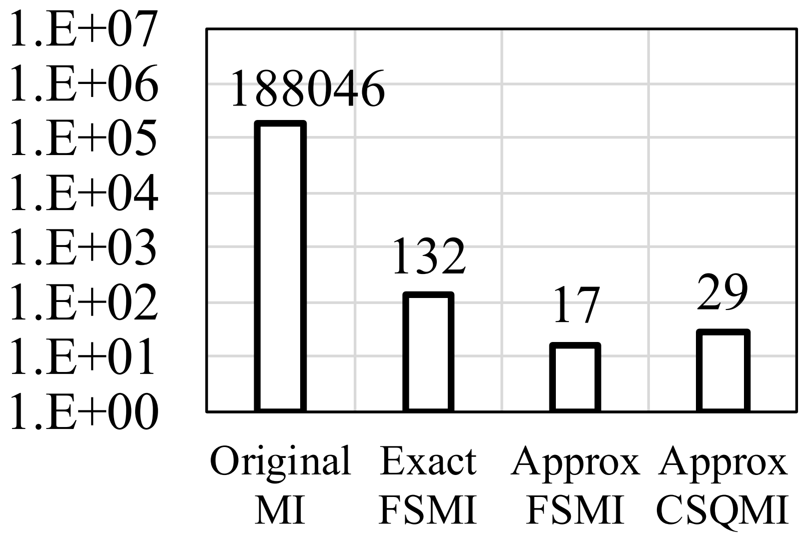
Two double for-loop

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

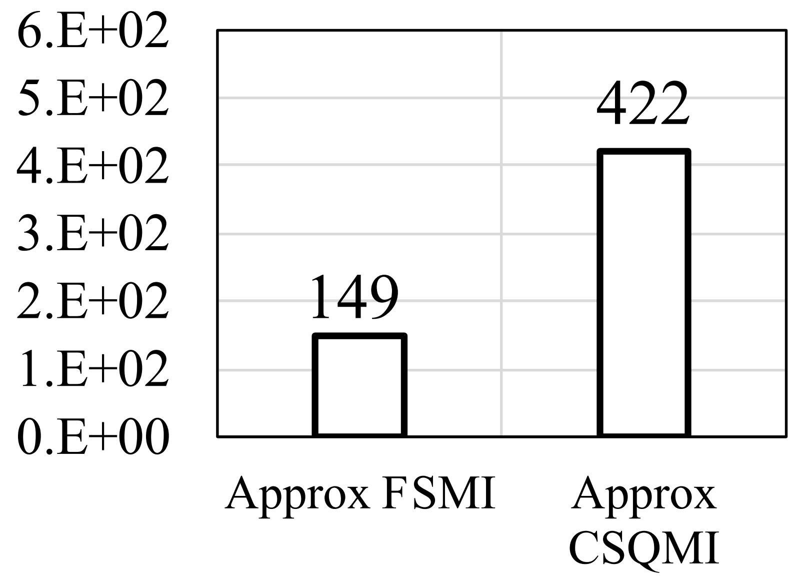
One double for-loop

Evaluation on 1D Synthetic Beam





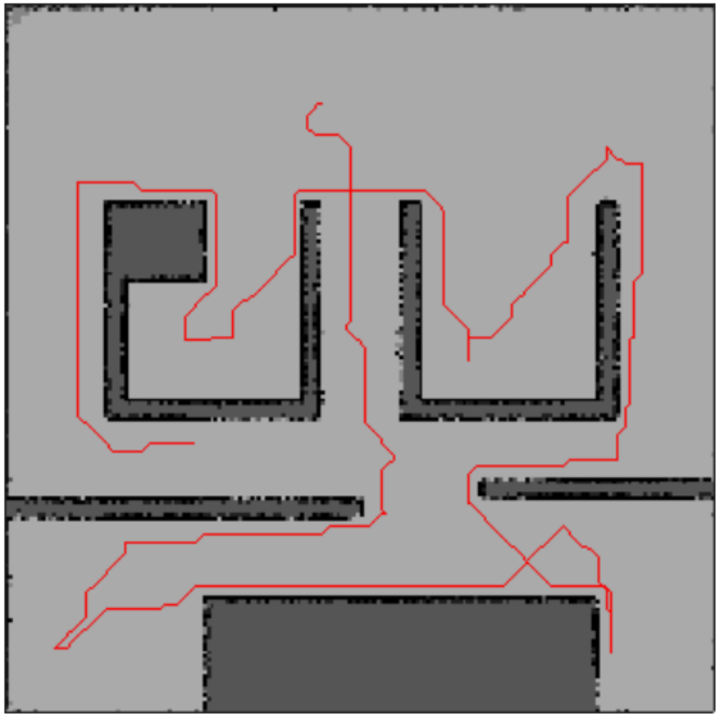
Average running time (us) on an **Intel CPU**



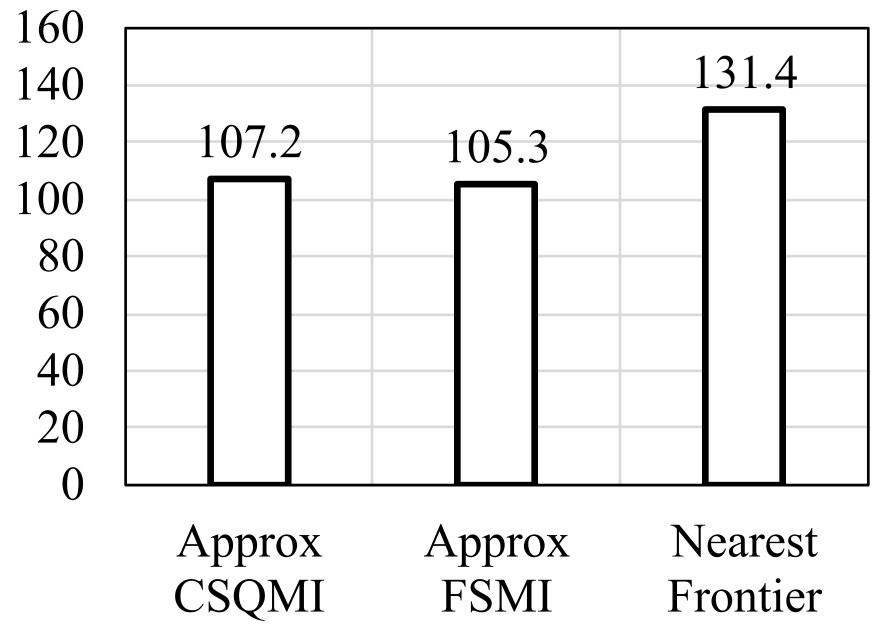
Average running time (us) on an **ARM CPU**

1.7× acceleration on an Intel CPU.
2.8× acceleration on an ARM CPU

Synthetic 2D experiment



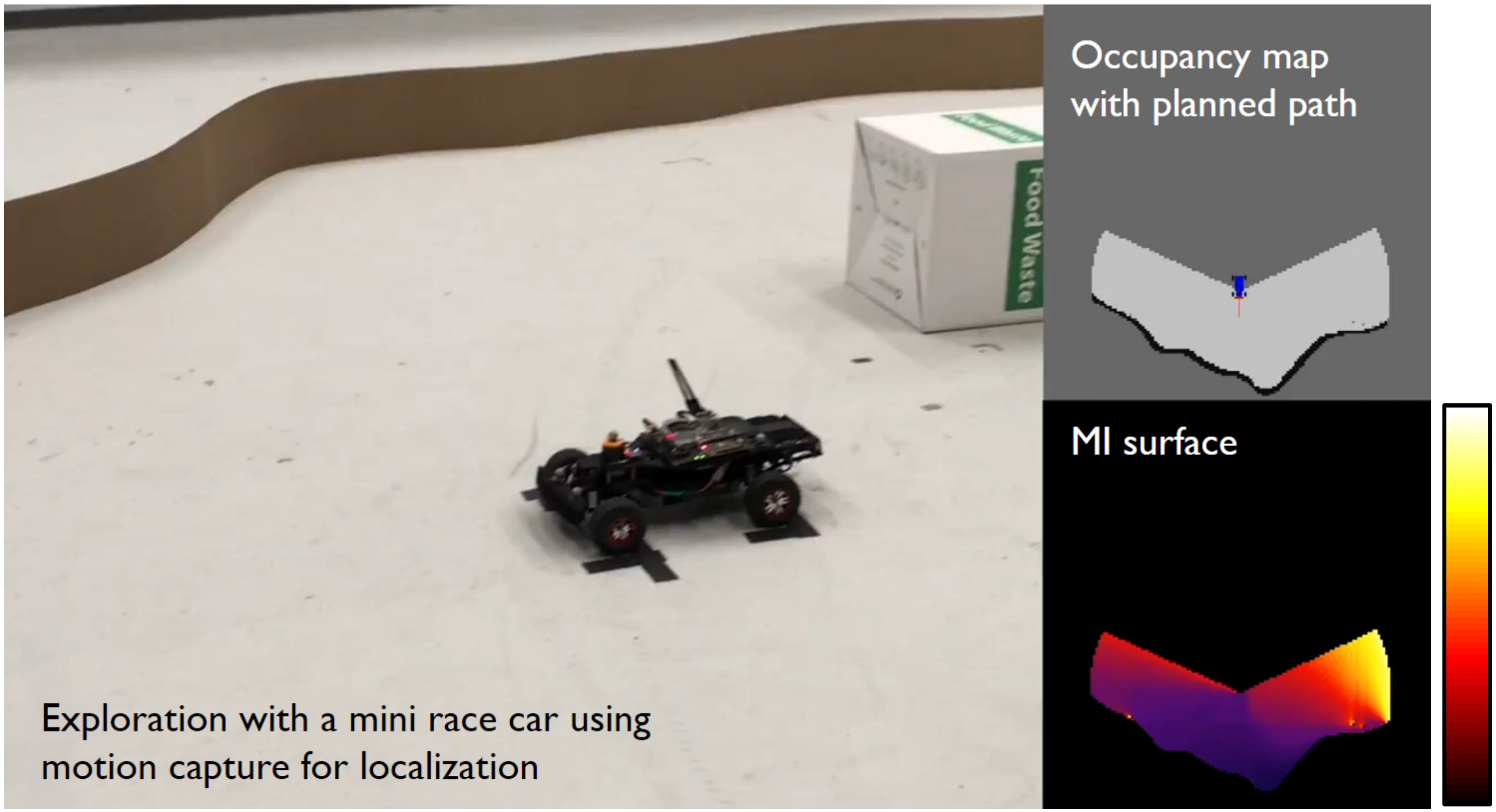
Map and Trajectory



Average Trajectory Length

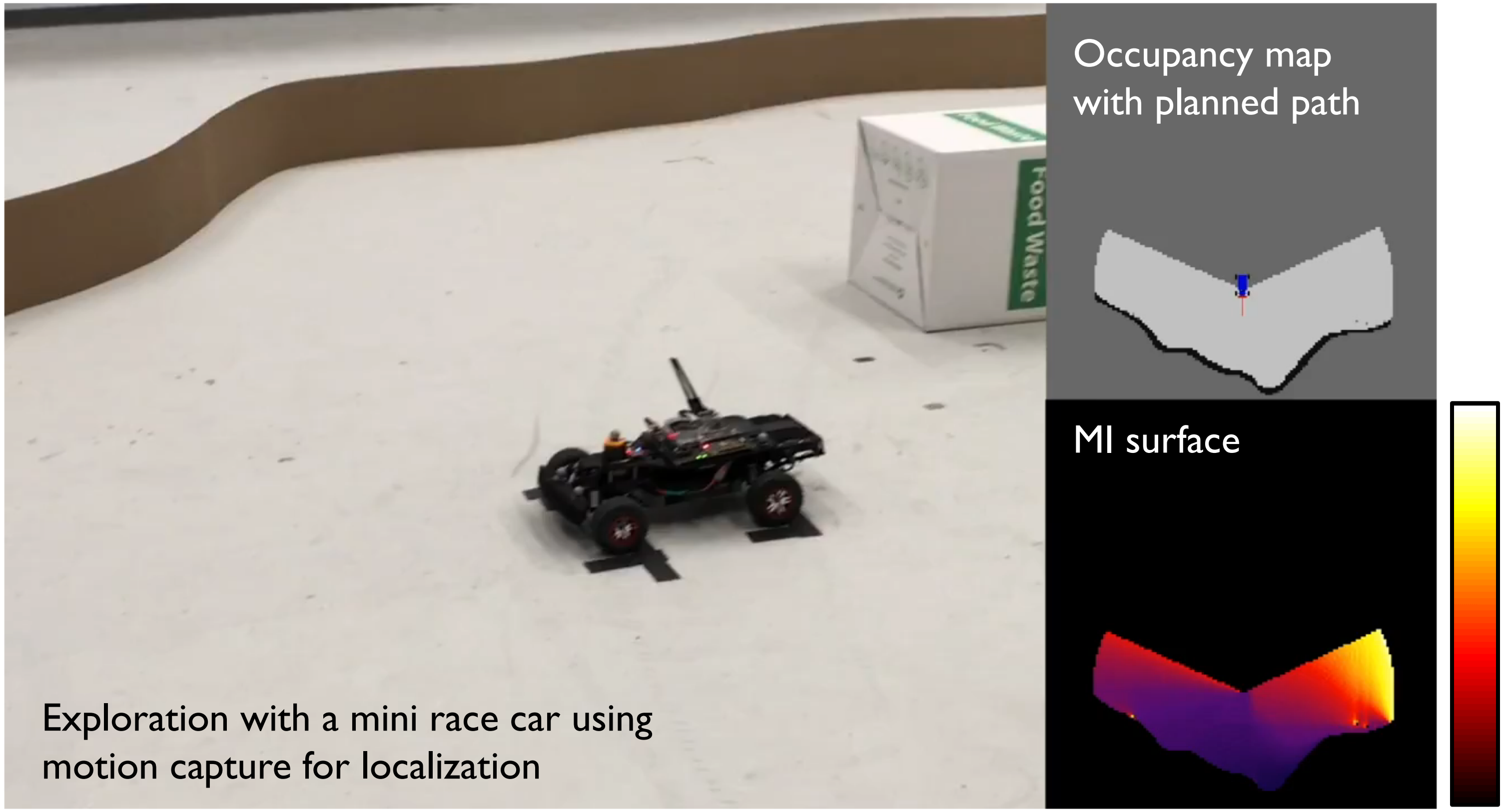
Acceleration with no penalty on trajectory length

Real Experiments (4×Realtime)



In collaboration with Trevor Henderson

Real Experiments (4×Realtime)



In collaboration with Trevor Henderson

Summary of Contributions

- **FSMI algorithm:** New formula that avoids numerical integration and computes the exact Shannon MI in $O(n^2)$
- **Approx FSMI algorithm:** Approximate the sensor noise model to compute Shannon MI in $O(n)$ with negligible accuracy loss
- **Tested in real system**: 2D mapping in the motion capture room with mini racecar

---

Compared with

Original Shannon MI from [1]	**(Alternative Metric) Approx CSQMI [2]**
$O(n^2\lambda_z)$	alternative information metric, $O(n)$
FSMI is more than $1000\times$ **faster (measured results)**	FSMI is around $1.7-2.8\times$ **faster (measured results)**

Published: Zhang, et. al, "FSMI: Fast computation of Shannon Mutual Information for Information Theoretic Mapping". ICRA, 2019

[1] Julian, et. al, "On mutual information based control of range sensing robots for mapping applications," IJRS, 2014
[2] Charrow et al., Information-theoretic mapping using cauchy-schwarz quadratic mutual information, ICRA 2015

Overview of Thesis Defense

$$I(M_i; Z) = \int_{z \geq 0} P(Z = z) f(\delta_i(z), r_i) dz$$

Preliminaries and notations

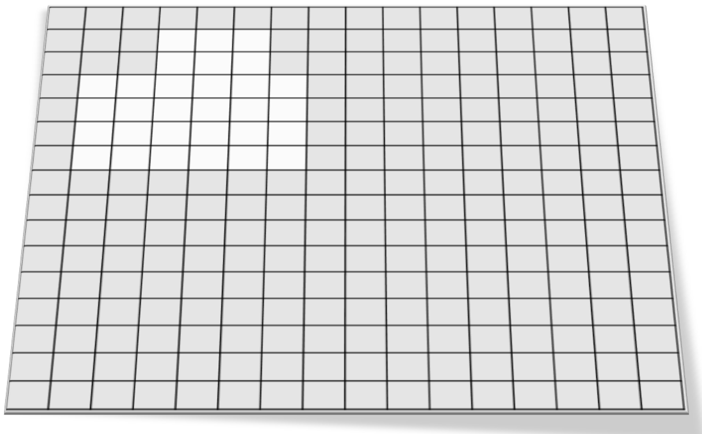
$$O(\lambda_m^2 \lambda_z) \rightarrow O(\lambda_m)$$

FSMI: Fast computation of Shannon Mutual Information

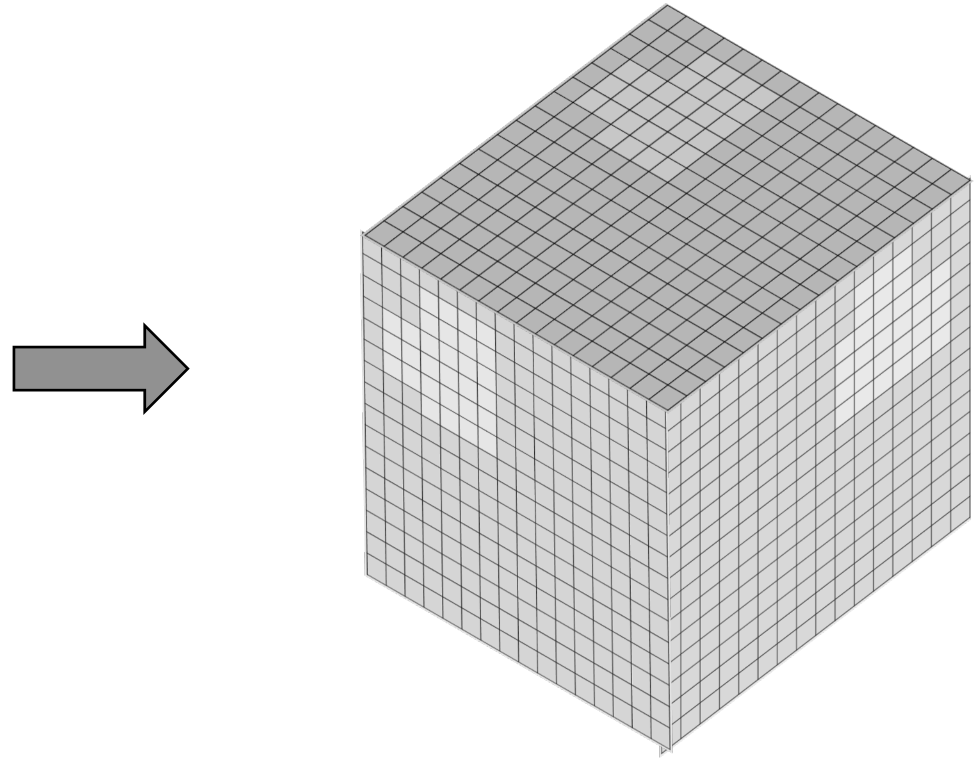
FSMI on compressed occupancy map for 3D mapping

Dedicated hardware for FSMI

Extension from 2D to 3D



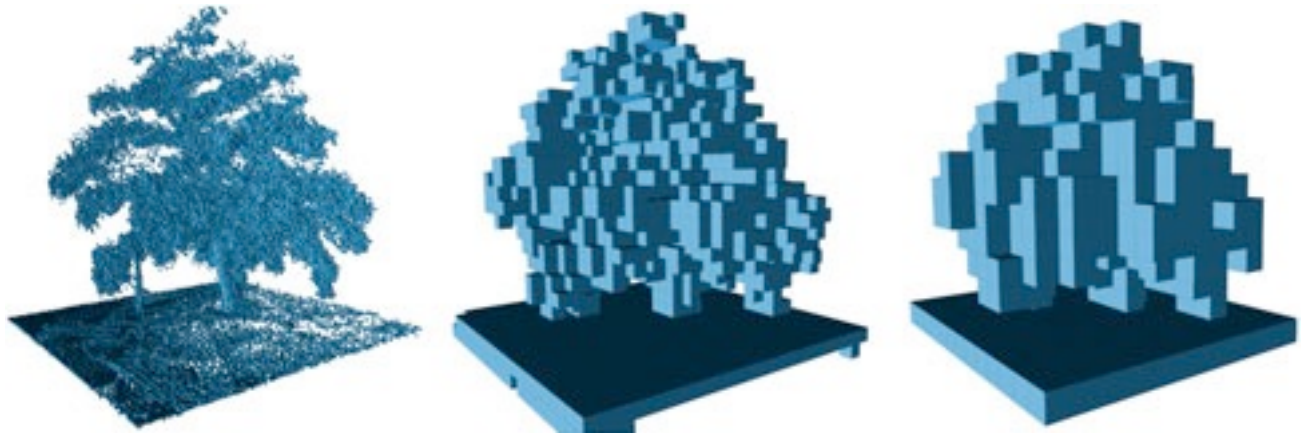
2D Occupancy Map



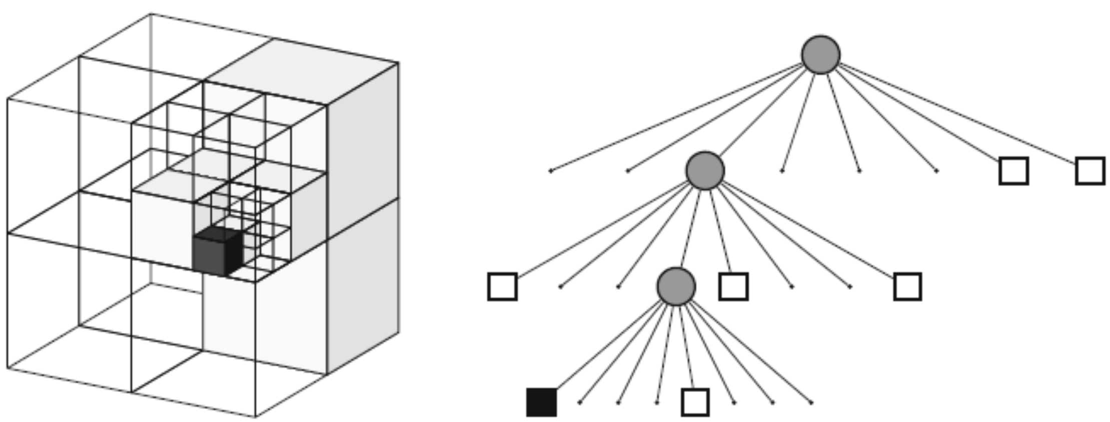
3D Voxel Map

The size of the **memory** grows more than **10×**

OctoMap for Compression



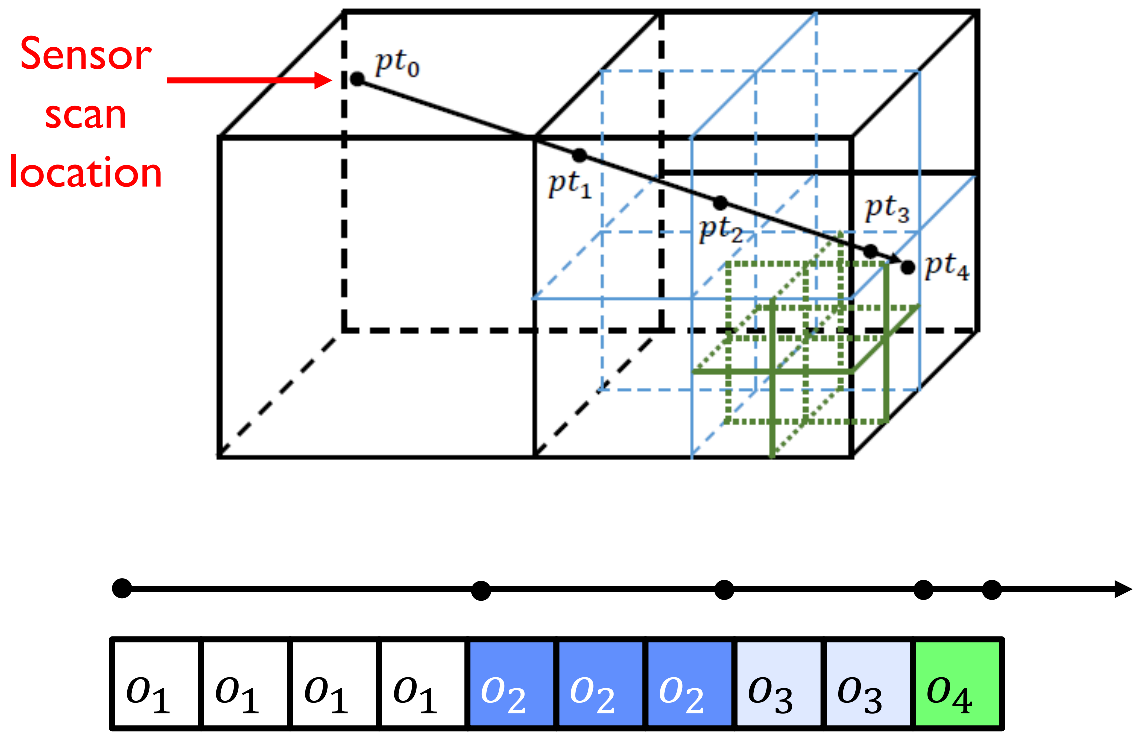
Representation at different scales



Adaptive 3D representation supporting multiple scales

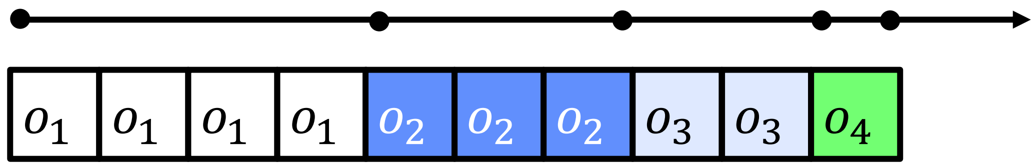
Hornung, et al., OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees, Autonomous Robots, 2013

Ray-tracing on the OctoMap



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

FSMI on Compressed Input



Uncompressed input format

$$\underbrace{o_1, o_1, \dots, o_1}_{L_1}, \underbrace{o_2, \dots, o_2}_{+ \; L_2}, \underbrace{o_3, \dots, o_3}_{+ \; L_3}, \dots, \underbrace{o_{n_r}, \dots, o_{n_r}}_{+ \; L_{n_r}} \; = \; n$$

Compressed format (Run Length Encoding)

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

Time complexity of Approx FSMI

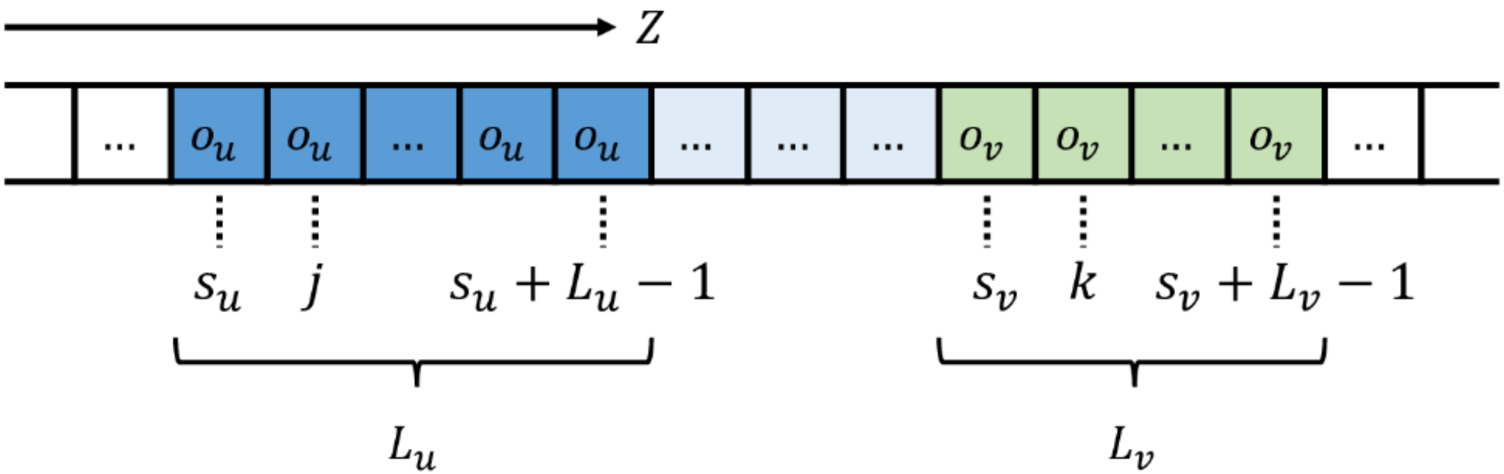
$$\boldsymbol{O(n)}$$

Goal: achieve the complexity of

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

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

FSMI on Compressed Input

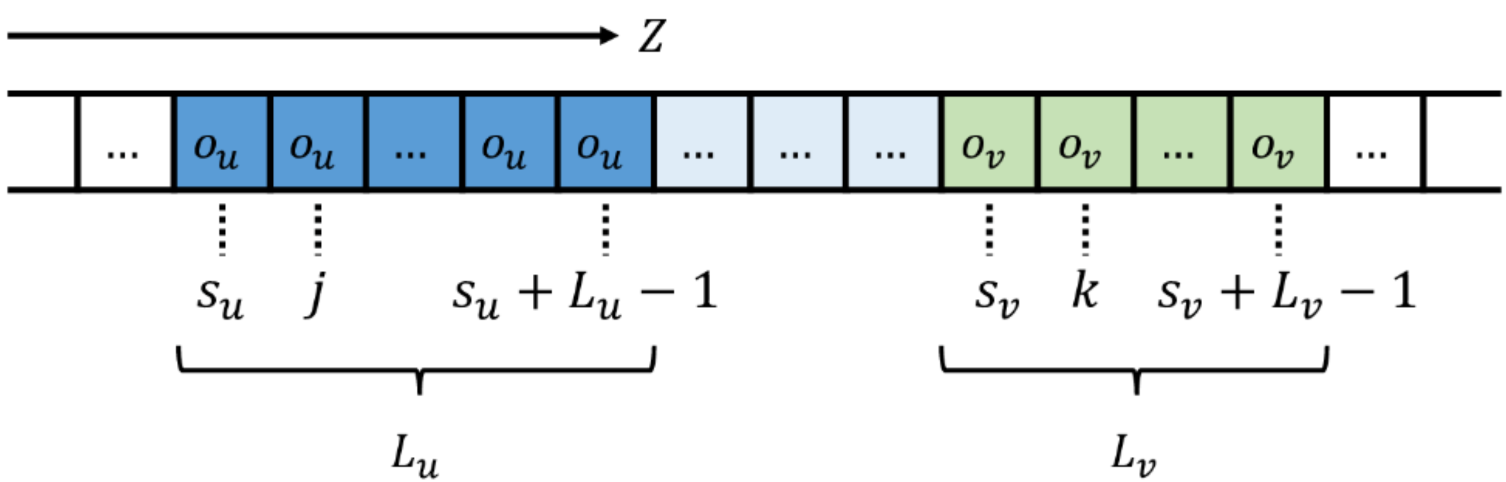


FSMI: $$I(M;Z) = \sum_{j=1}^{n}\sum_{k=1}^{n} P(e_j) C_k G_{k,j}$$

$$I(M;Z) = \sum_{u=1}^{n_r}\sum_{v=1}^{n_r}\left(\sum_{j=1}^{L_u}\sum_{k=1}^{L_v} P(e_{s_u+j}) C_{s_v+k} G_{s_u+j, s_v+k}\right)$$

All the computation between two groups of cells of length L_u and L_v.

FSMI on Compressed Input



$$I(M;Z) = \sum_{u=1}^{n_r} \sum_{v=1}^{n_r} \left(\sum_{j=1}^{L_u} \sum_{k=1}^{L_v} P\left(e_{s_u+j}\right) C_{s_v+k} G_{s_u+j,s_v+k} \right)$$

$$P_E(u)(\mathbf{1} - \boldsymbol{o_u})^j o_u$$

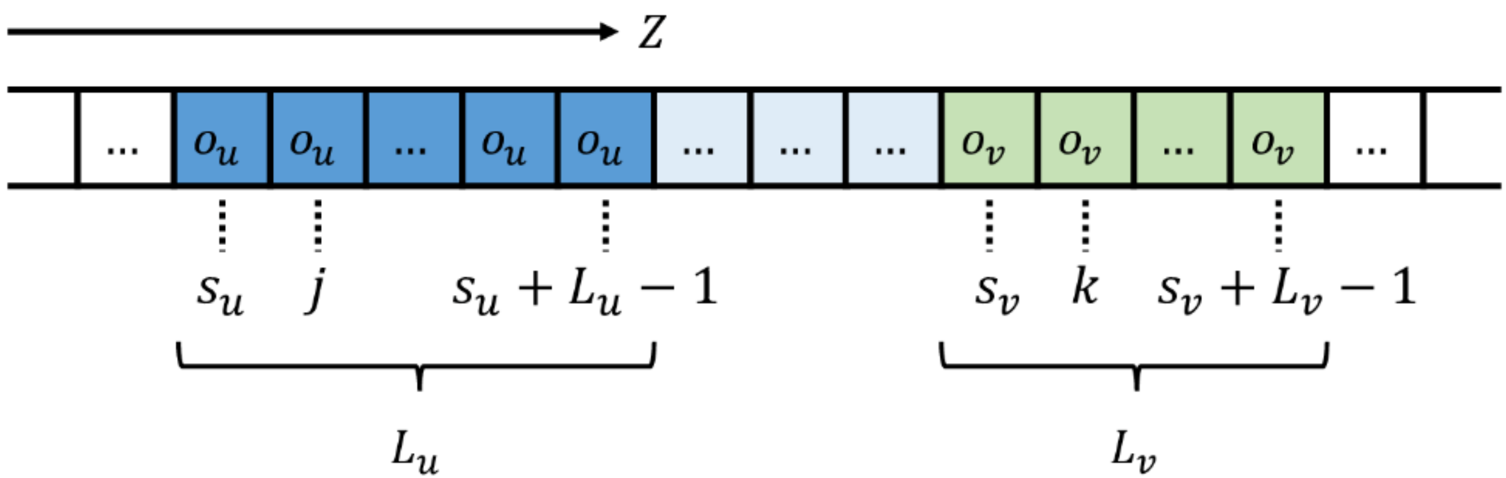
$$D_E(\mathrm{v}) + k \cdot f\left(\delta_{emp}, r_v\right) + f\left(\delta_{occ}, r_v\right)$$

$$\frac{1}{Z} \exp\left(-\frac{(j - k + t)^2}{2\sigma^2} \right)$$

$t = s_u - s_v$,

relative distance of the two blocks

FSMI on Compressed Input



$$\sum_{j=1}^{L_u} \sum_{k=1}^{L_v} k \cdot x^j \exp\left(-\frac{(j-k+t)^2}{2\sigma^2}\right)$$

If the summation is evaluable in $\boldsymbol{O(1)}$, the exact FSMI can be evaluated in $\boldsymbol{O(n_r^2)}$, and the approx. FSMI can be evaluated in $\boldsymbol{O(n_r)}$

Analytical Solution

$$\sum_{j=1}^{L_u}\sum_{k=1}^{L_v} k \cdot x^j \exp\left(-\frac{(j-k+t)^2}{2\sigma^2}\right)$$

Approximation

$$\int\int k \cdot x^j \exp\left(-\frac{(j-k+t)^2}{2\sigma^2}\right) dj\, dk$$

There is a closed-form solution!

Analytical solution

$$\frac{e^{\frac{1}{2}\sigma^2 \mathrm{Log}[x]^2}\sqrt{\frac{\pi}{2}}\,\sigma\left(-2\,x^{\frac{1}{2}+L2}\,\mathrm{Erf}\left[\frac{L2+\sigma^2\,\mathrm{Log}[x]}{\sqrt{2}\,\sigma}\right]-2\,e^{-\frac{1}{2}\sigma^2\mathrm{Log}[x]^2}\sqrt{\frac{2}{\pi}}\sqrt{x}\,\sigma\,\mathrm{Log}[x]+2\,e^{-\frac{L2^2+\sigma^4\mathrm{Log}[x]^2}{2\sigma^2}}\sqrt{\frac{2}{\pi}}\sqrt{x}\,\sigma\,\mathrm{Log}[x]-x^{\frac{1}{2}+L2}\,\mathrm{Erf}\left[\frac{L2+\sigma^2\,\mathrm{Log}[x]}{\sqrt{2}\,\sigma}\right]\mathrm{Log}[x]+2\,L2\,x^{\frac{1}{2}+L2}\,\mathrm{Erf}\left[\frac{L2+\sigma^2\,\mathrm{Log}[x]}{\sqrt{2}\,\sigma}\right]\mathrm{Log}[x]\right.}{2\,x\,\mathrm{Log}[x]^2}$$

$$\frac{\left.+e^{-\frac{1}{2}\sigma^2\mathrm{Log}[x]^2}\sqrt{x}\,\mathrm{Erf}\left[\frac{L2}{\sqrt{2}\,\sigma}\right](2+\mathrm{Log}[x])+\sqrt{x}\,\mathrm{Erf}\left[\frac{\sigma\,\mathrm{Log}[x]}{\sqrt{2}}\right](2+\mathrm{Log}[x])\right)}{2\,x\,\mathrm{Log}[x]^2}-$$

$$\frac{e^{\frac{1}{2}\sigma^2\mathrm{Log}[x]^2}\sqrt{\frac{\pi}{2}}\,x^{\frac{1}{2}(-1-\sigma^2\mathrm{Log}[x])}\,\sigma\left(-2\,x^{\frac{1}{2}\sigma^2\mathrm{Log}[x]}\,\mathrm{Erf}\left[\frac{L1-\sigma^2\,\mathrm{Log}[x]}{\sqrt{2}\,\sigma}\right]-2\,x^{L2+\frac{1}{2}\sigma^2\mathrm{Log}[x]}\,\mathrm{Erf}\left[\frac{-L1+L2+\sigma^2\,\mathrm{Log}[x]}{\sqrt{2}\,\sigma}\right]-2\,e^{-\frac{L1^2}{2\sigma^2}}\sqrt{\frac{2}{\pi}}\,x^{L1}\,\sigma\,\mathrm{Log}[x]+2\,e^{-\frac{(L1-L2)^2}{2\sigma^2}}\sqrt{\frac{2}{\pi}}\,x^{L1}\,\sigma\,\mathrm{Log}[x]-\right.}{2\,\mathrm{Log}[x]^2}$$

$$\frac{\left.x^{\frac{1}{2}\sigma^2\mathrm{Log}[x]}\,\mathrm{Erf}\left[\frac{L1-\sigma^2\,\mathrm{Log}[x]}{\sqrt{2}\,\sigma}\right]\mathrm{Log}[x]-x^{L2+\frac{1}{2}\sigma^2\mathrm{Log}[x]}\,\mathrm{Erf}\left[\frac{-L1+L2+\sigma^2\,\mathrm{Log}[x]}{\sqrt{2}\,\sigma}\right]\mathrm{Log}[x]+2\,L2\,x^{L2+\frac{1}{2}\sigma^2\mathrm{Log}[x]}\,\mathrm{Erf}\left[\frac{-L1+L2+\sigma^2\,\mathrm{Log}[x]}{\sqrt{2}\,\sigma}\right]\mathrm{Log}[x]-x^{L1}\,\mathrm{Erf}\left[\frac{L1}{\sqrt{2}\,\sigma}\right](-2+(-1+2\,L1)\,\mathrm{Log}[x])+x^{L1}\,\mathrm{Erf}\left[\frac{L1-L2}{\sqrt{2}\,\sigma}\right](-2+(-1+2\,L1)\,\mathrm{Log}[x])\right)}{2\,\mathrm{Log}[x]^2}$$

8 different erf() evaluations, 10 evaluations of non-trivial terms, about 60 multiplications to combine them

Tabulation based Solution

$$A[L_u, L_v, x, t] = \sum_{j=1}^{L_u} \sum_{k=1}^{L_v} k \cdot x^j \, exp\left(-\frac{(j-k+t)^2}{2\sigma^2}\right)$$

Size of the table: $n^3|X|$, where $|\boldsymbol{X}|$ is the number of possible quantization levels of occupancy value o_i

When $n = 256, |X| = 100,$ the size of the table is **6.25GB**

Reducing the Table Size via Decomposition

$$A[L_u, L_v, x, t] = \sum_{j=1}^{L_u} \sum_{k=1}^{L_v} k \cdot x^j \, exp\left(-\frac{(j-k+t)^2}{2\sigma^2}\right)$$

Remove a dimension for t from the tabulation

$$\alpha[L_u, L_v, x] = \sum_{j=1}^{L_u} \sum_{k=1}^{L_v} k \cdot x^j \, exp\left(-\frac{(j-k)^2}{2\sigma^2}\right)$$

Size of the table is reduced from $n^3|X|$ to $n^2|X|$

Reconstruct an entry in $A[]$ from entries in $\alpha[]$:

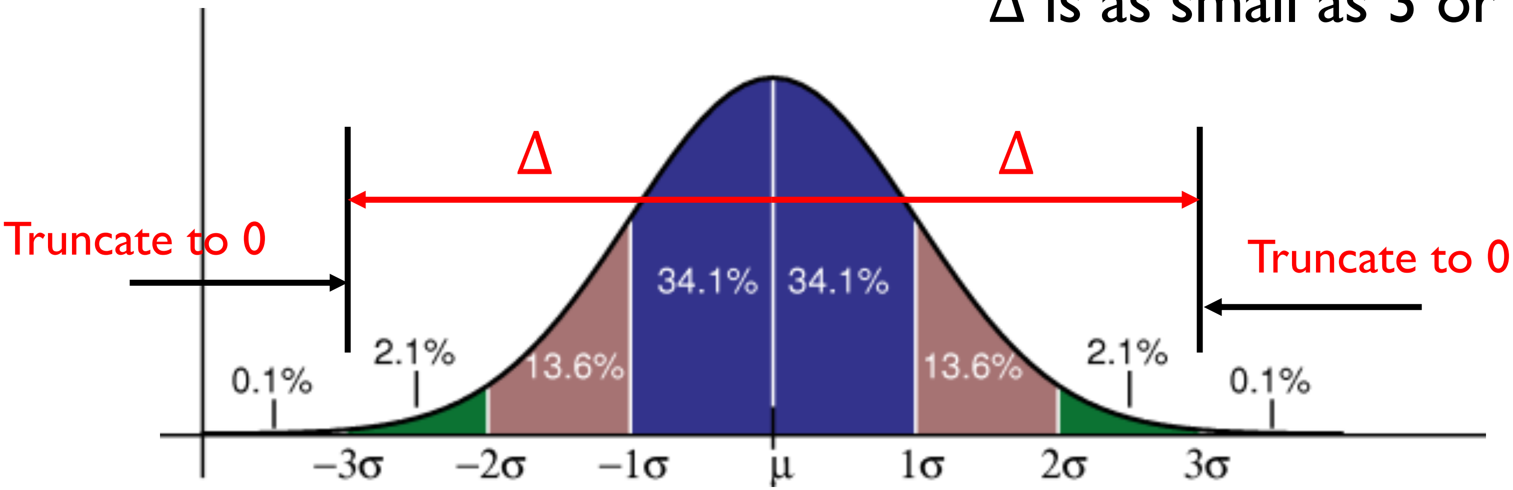
When $t > 1$, $A[L_u, L_v, x, t] = x^{-t}(\alpha[x, L_u + t, L_v] - \alpha[x, t, L_v])$

Compute the entries of a big table with multiple accesses to a smaller table

Gaussian Truncation

Gaussian Truncation: $\exp\left(-\frac{(j-k)^2}{2\sigma^2}\right) \approx 0$, when $|j-k| > \Delta$

Δ is as small as 3 or 5

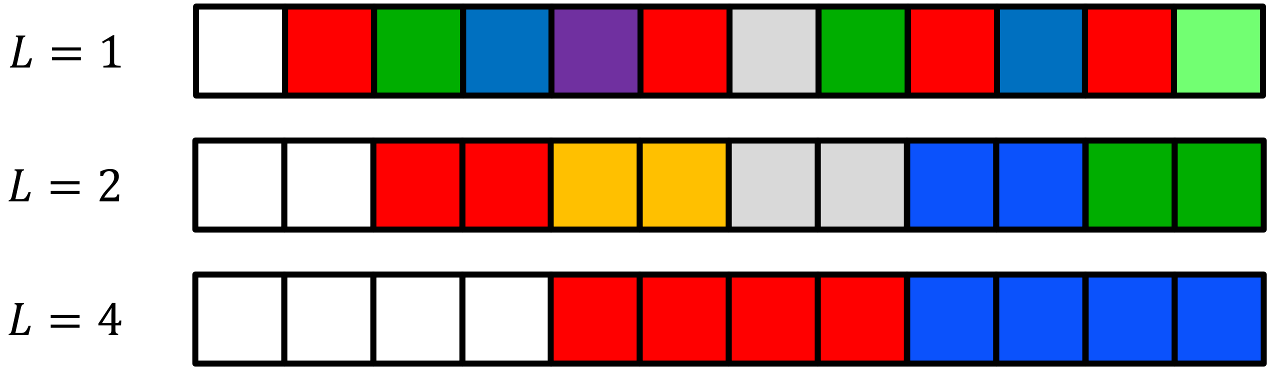


Normal distribution

This bounds L_u, L_v, t by Δ instead of n.
This reduces the table size $(2\Delta)^2|X|$

When $n = 256, \Delta = 5, |X| = 100$, the size of the table is reduced from **6.5GB** to **40KB**

Acceleration on 1D Synthetic Data



Average Completion Time (us) on a Beam of 256 cells

Baseline (Approx FSMI): 56us

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

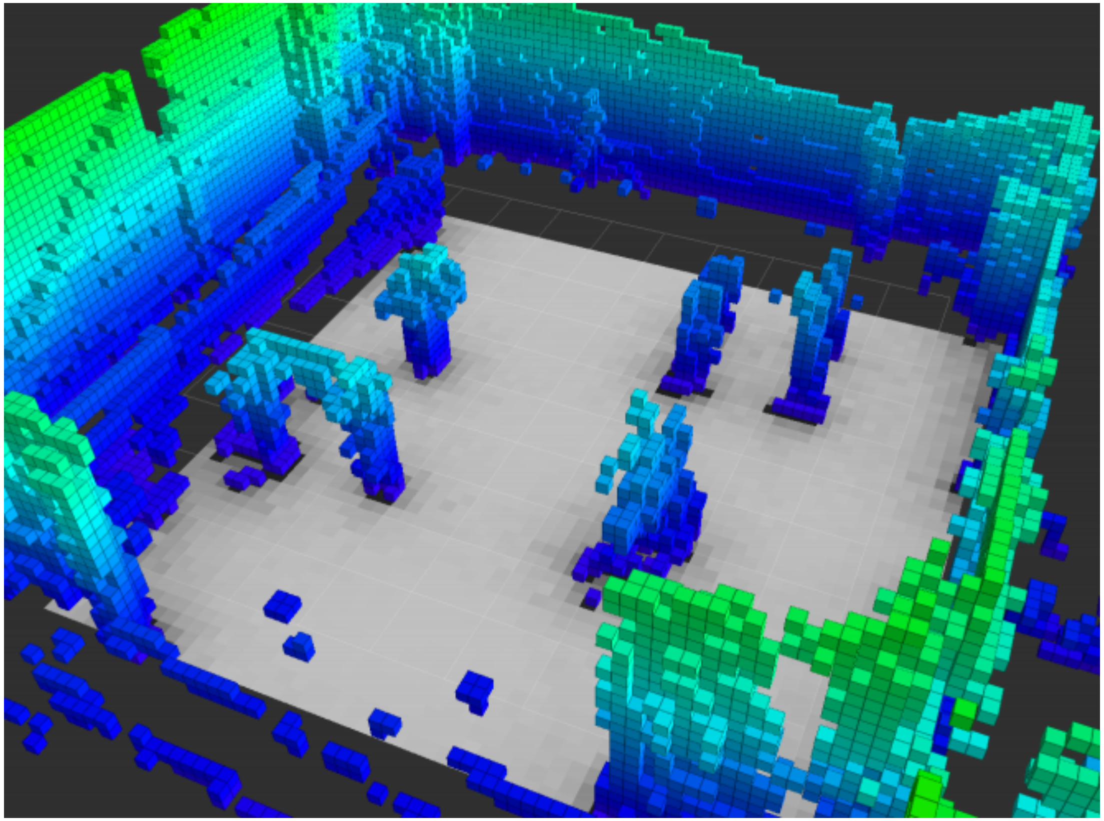
Measured on an Intel Xeon CPU

Experiments

Motion capture room of size 10m x 10m x 5m, at 0.05m, with 200x200x100 = 4M cells

In collaboration with Trevor Henderson

Experiments



We record an average compression ratio of around $18\times$, with an acceleration ratio of $8\times$

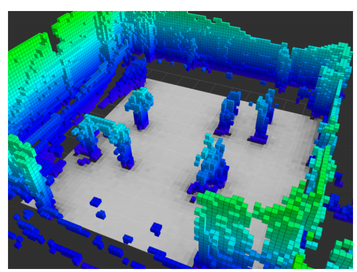
Summary of Contributions

FSMI-RLE Algorithm that computes directly on a compressed format by run-length encoding, yielding $8\times$ acceleration for 3D mapping with OctoMap

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

Gaussian truncation, and table decomposition to reduce size of the look-up table by **162, 500×**, in practice this reduces a table of size **6.5 GB to 40 KB**

Tested on real platforms and integrated into a real system of racecar in motion capture room of 10m x 10m x 5m



In preparation for submission

Overview of Thesis Defense

$$I(M_i; Z) = \int_{z \geq 0} P(Z = z) f(\delta_i(z), r_i) dz$$

Preliminaries and notations

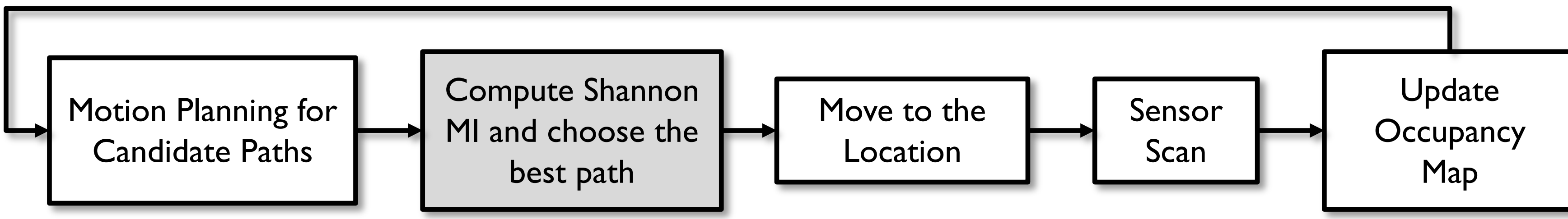
$$O(\lambda_m^2 \lambda_z) \rightarrow O(\lambda_m)$$

FSMI: Fast computation of Shannon Mutual Information

FSMI on compressed occupancy map for 3D mapping

Dedicated hardware for FSMI

System Overview



Map and candidate scan locations →

← Shannon MI at the input locations

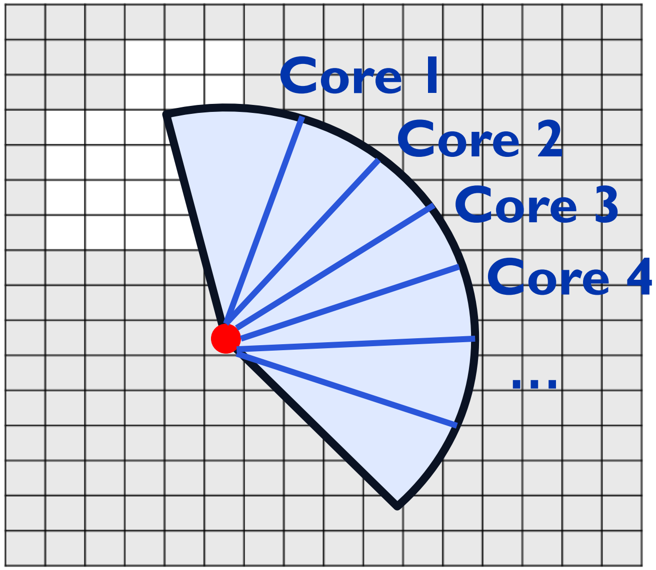
NVIDIA Jetson TX2

FPGA

In collaboration with Peter Li

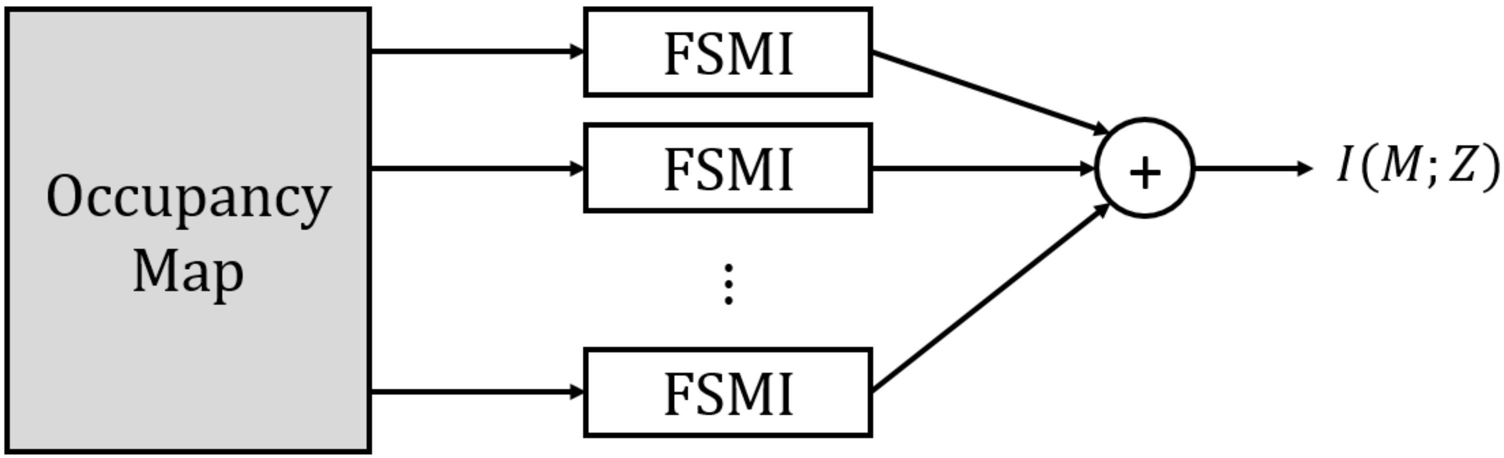
FSMI Algorithm is Parallelizable

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



FSMI can run in parallel among different beams of a scan

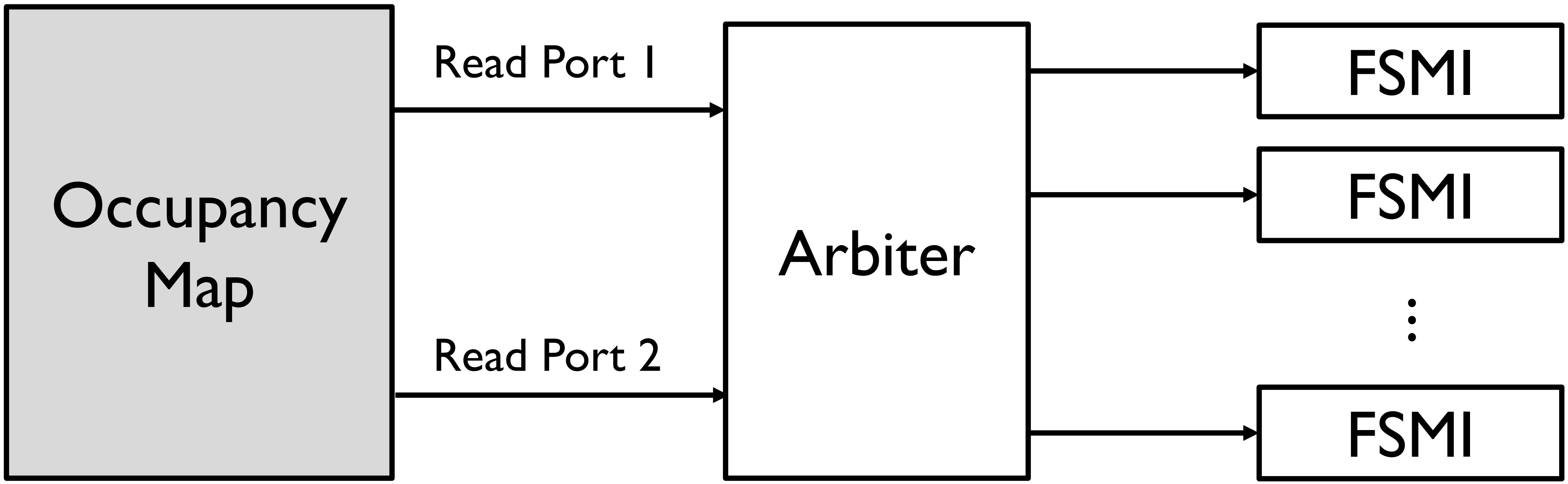
High-level Architecture



10× faster than an Intel Xeon Core

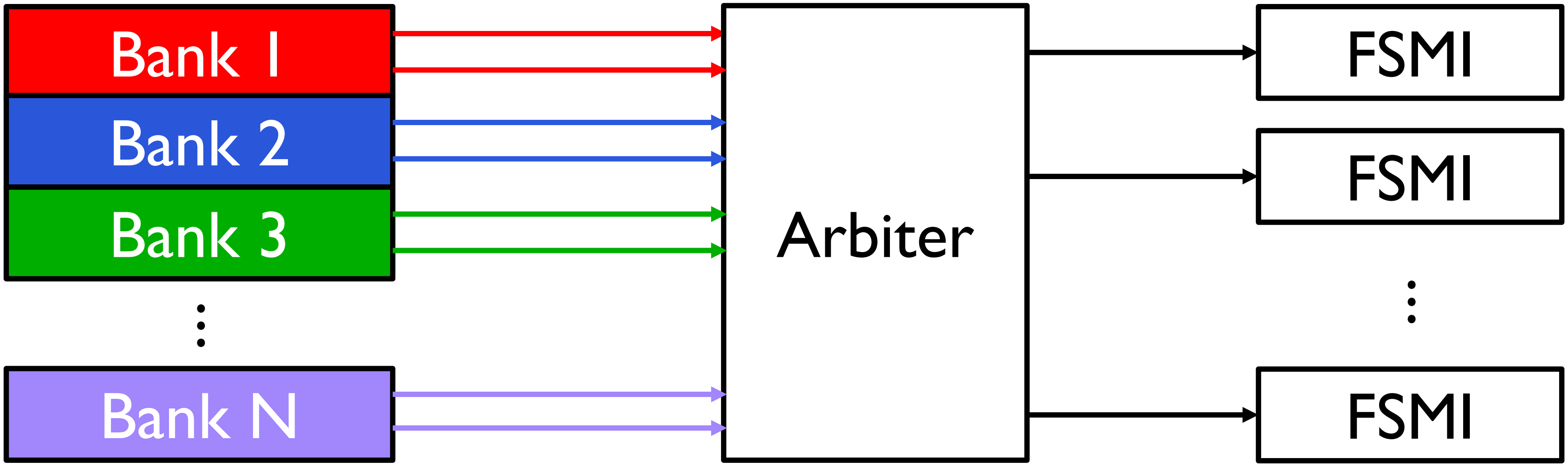
Challenge: provide enough **memory bandwidth** to keep all the FSMI cores busy

Memory Bandwidth on FPGA



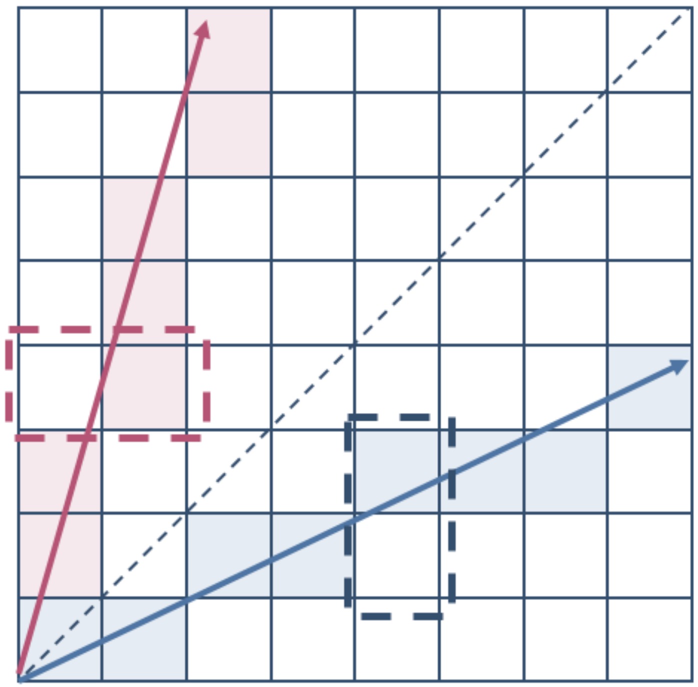
On FPGA, an SRAM only has at most **two read ports**

Solution: Banking

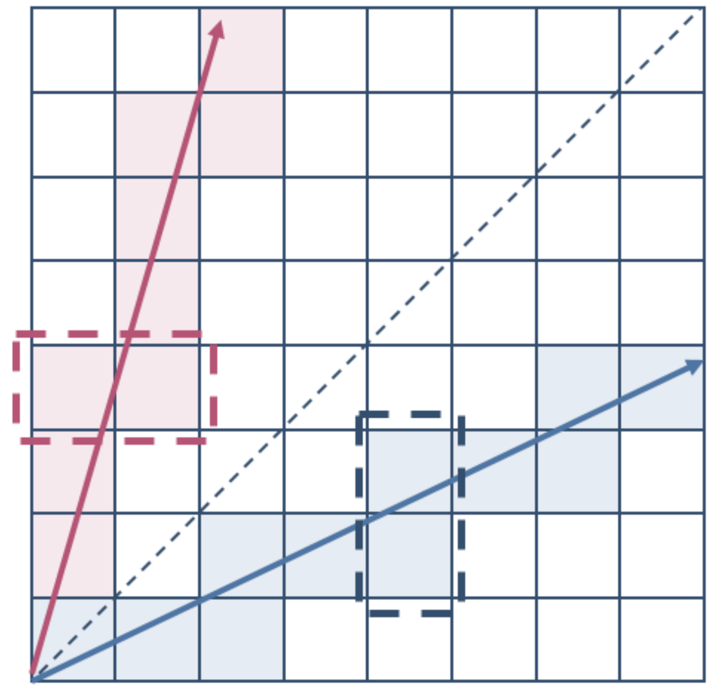


Challenge: How to break up the memory to reduce the chance for multiple FSMI cores to visit the same bank?

Special Memory Access Pattern



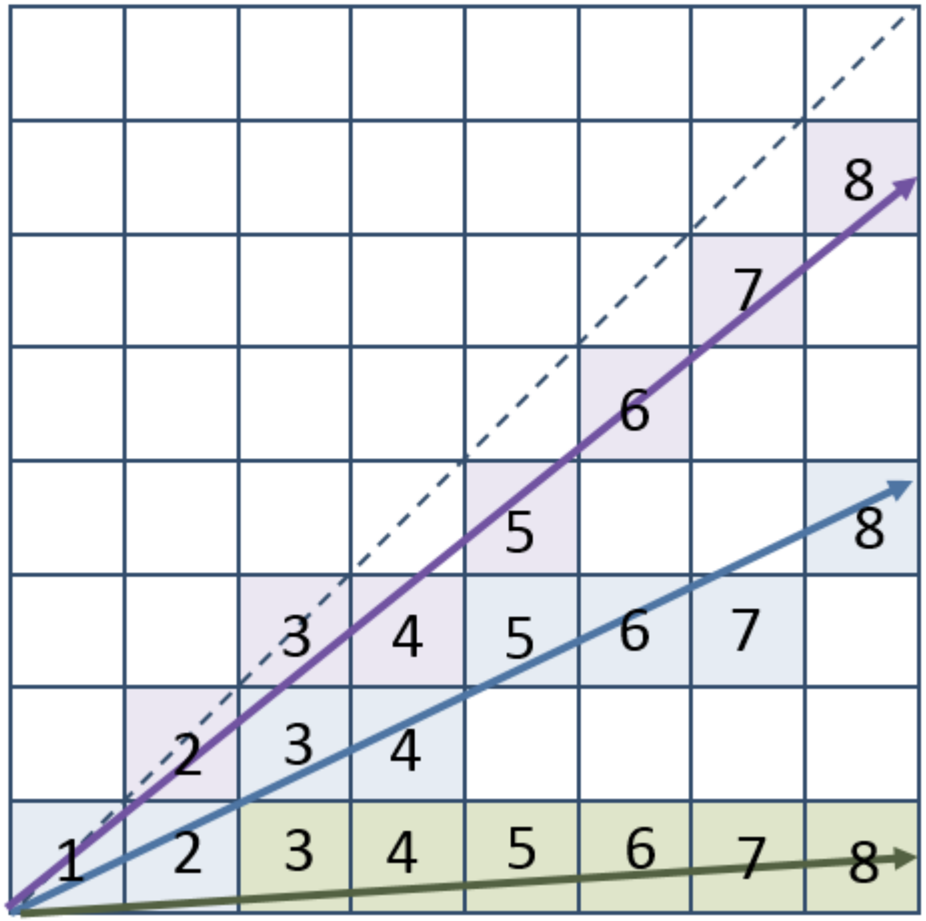
Bresenham's Ray-tracing Algorithm



Naïve Ray-tracing Algorithm

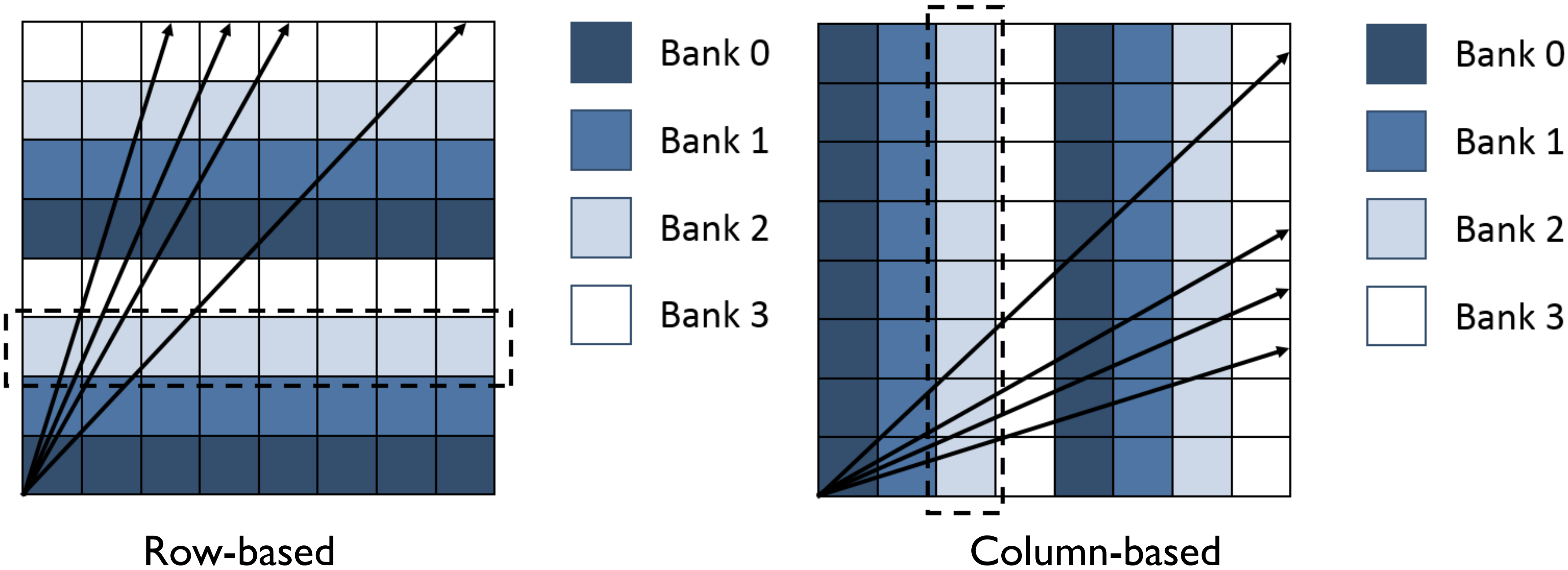
Along the major axis, only **one cell per step**

Special Memory Access Pattern



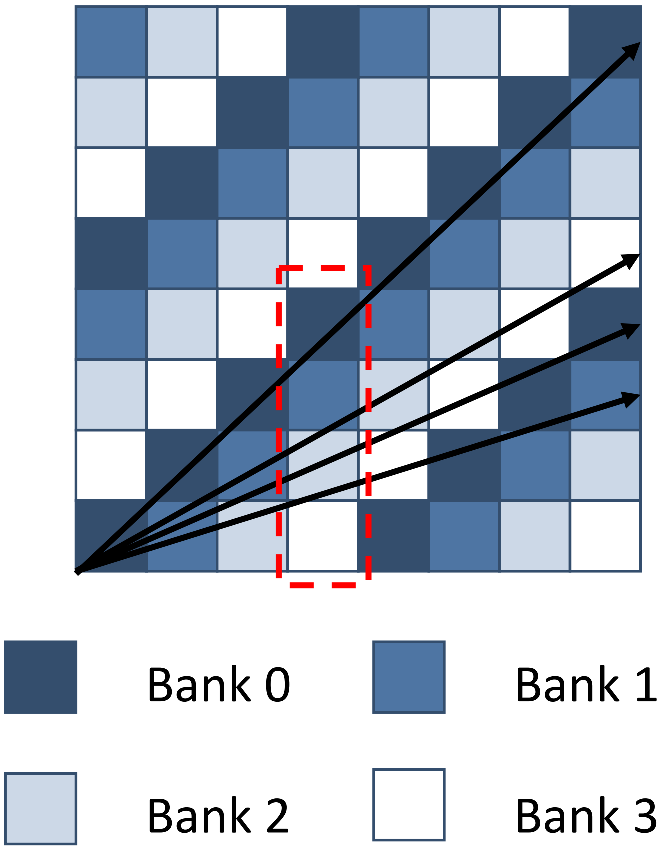
Cells with the same numbers are accessed at the same time. Ideally they should be stored in different banks. Otherwise there would be **read conflicts and some cores will stall**.

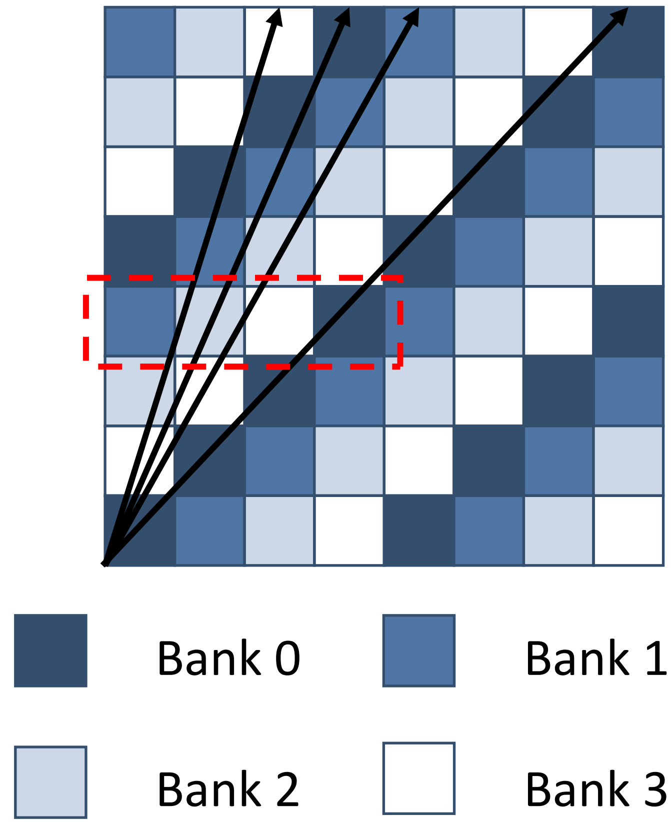
Simple Banking does NOT Work



Different beams always collide

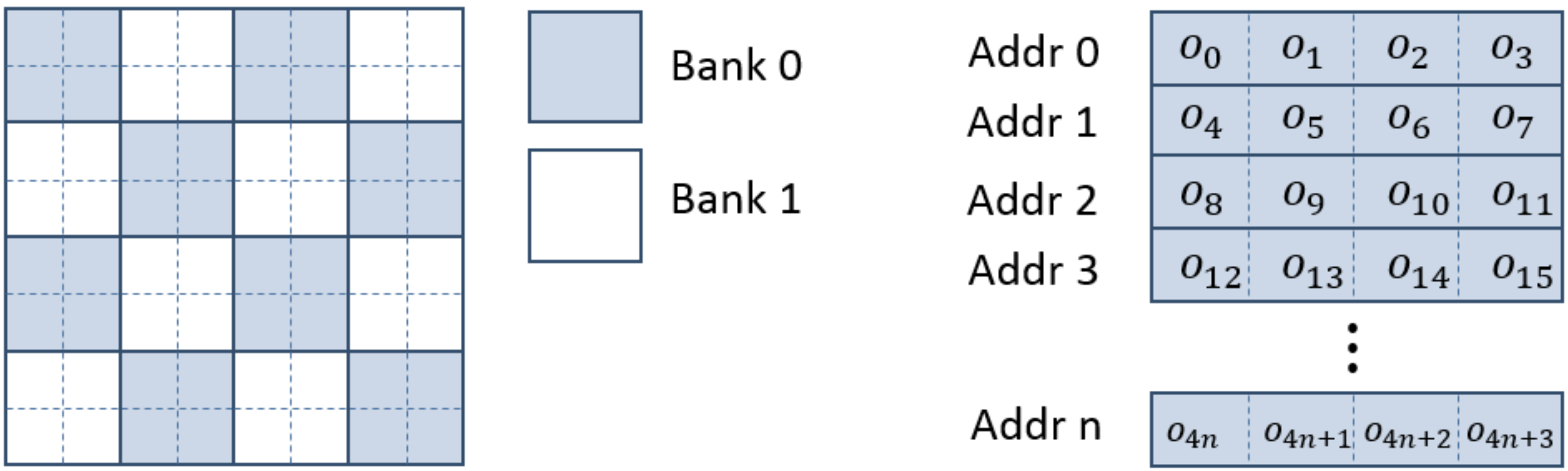
Proposed Banking Pattern





If there are N banks, no collision up to the N-th column/row.

Further Improvement on Bandwidth

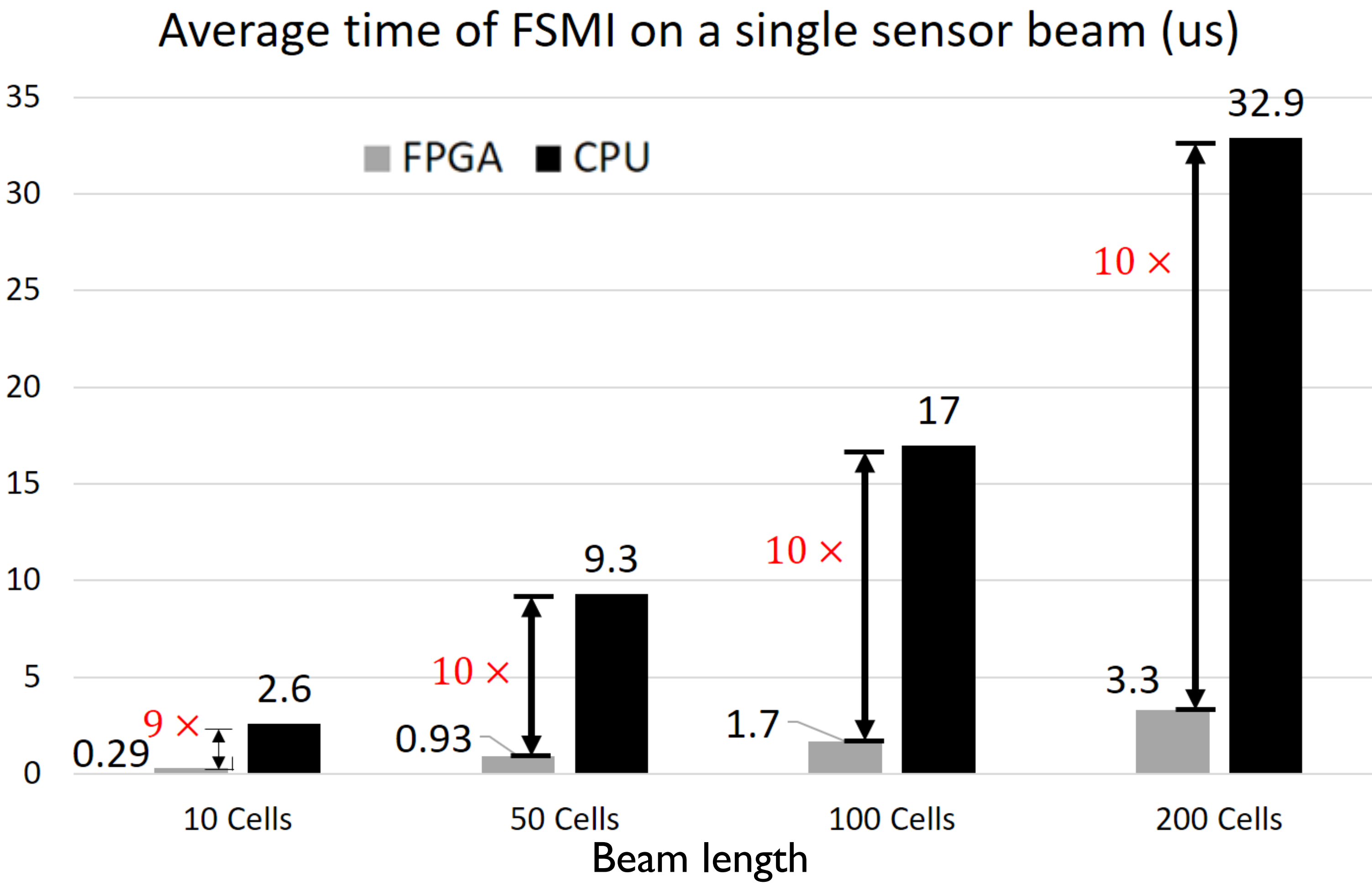


Increase the bandwidth by packing more values in one address.

FPGA Implementation

- **Xilinx Zynq-7000(XC7Z045) FPGA**
- **16 FSMI cores**
- **512x512 Occupancy Map**
- **Baseline:** Intel Xeon E5-4627 CPU

Experimental Results



With 16 cores, the system is over **100x faster than an Intel Xeon core**.
Can compute MI for a complete 200×200 map at 2Hz.

Experimental Results

	Baseline	Vertical banking & No packing	Diagonal banking & No packing	Diagonal banking & Pack 2x2	Unlimited Bandwidth
Latency	$86.53us$	$39.15us$	$16.48us$	$12.58us$	$11.84us$
Speed up	$1\times$	$2.21\times$	$5.25\times$	$6.88\times$	$7.31\times$

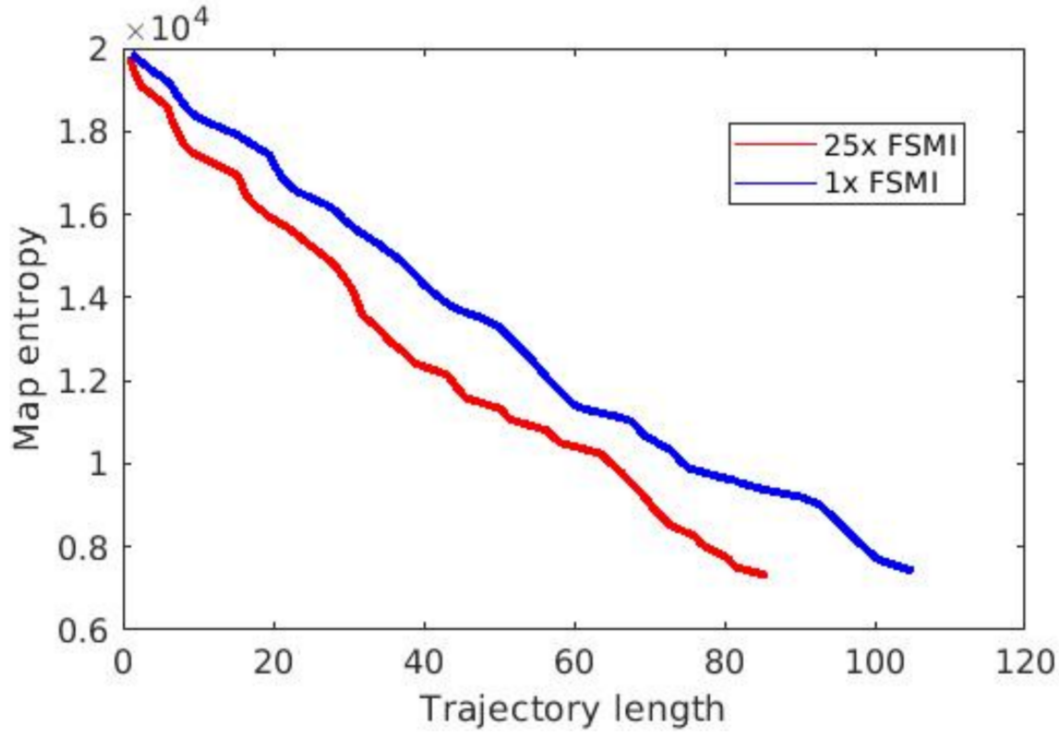
The final design is only 6.25% slower than the ideal case with unlimited bandwidth.

FPGA Profile

Module	LUT (Logic)	LUT (Reg)	BRAM	LUT (RAM)	DSP	Dynamic Power
Angle Assigner	131	437	0	0	0	0.001W
Bresenham (16x)	10602	3017	0	1408	32	0.042W
Arbiter	25226	2055	0	662	0	0.107W
Occupancy Grid Map	0	0	64	0	0	
FSMI Core (16x)	163141	36856	40	49	288	1.827W
Total	**199101**	**42365**	**104**	**2119**	**320**	**1.977W**
Utilization of FPGA	**91.1%**	**9.7%**	**19.1%**	**3.0%**	**35.6%**	**NA**

Less than 10% of the CPU power, despite more than 100x faster.

Impact of Acceleration



Being able to evaluate 25x more FSMI leads to 19% shorter exploration path in a synthetic 2D exploration task.

Summary Of Contributions

- Optimization on memory design to provide **6.8× more memory bandwidth** for **16 FSMI cores** to run in parallel
- **Diagonal stripe banking pattern** for the special memory access pattern introduced by the Bresenham's algorithm
- **Packing multiple occupancy values** into one cell to increase the memory bandwidth
- More than **100× faster** than an Intel CPU core while consuming **less than 10% of power**
- 200x200 MI map at 2 Hz
- In submission

Summary of Contributions Presented in the Defense

- FSMI Algorithm that computes Shannon Mutual Information in $\boldsymbol{O(n)}$, three orders of magnitude faster
- FSMI-RLE Algorithm that computes FSMI on **Compressed Input** from OctoMap, around 8x faster
- **Novel architecture** on FPGA to run FSMI 100x faster than CPU while consuming 10% power.

$$\boldsymbol{O}(\lambda_m^2\lambda_z) \rightarrow \boldsymbol{O}(\lambda_m)$$

Comparison with Previous Works

- **Three orders of magnitude** faster than the original algorithm to Compute Shannon MI between perspective range measurements and map [1]
- **2-3 times faster** than less well-understood, alternative metric CSQMI [2]
- **First work** to study the computation of mutual information on OctoMap for 3D mapping
- **First work** to build dedicated accelerator for the computation of mutual information

[1] Julian, et. al, "On mutual informationbased control of range sensing robots for mapping applications," IJRS, 2014
[2] Charrow et al., Information-theoretic mapping using cauchy-schwarz quadratic mutual information, ICRA 2015

Conclusion of the Thesis

- **Algorithm-hardware co-design** can improve the overall energy efficiency and throughput of the system more than what could be achieved from optimization each individually
- Compressing the data and directly performing **computation on the compressed data structure** enables significant acceleration
- Even if the algorithm is parallelizable, it is critical to design a memory architecture that can **provide enough memory bandwidth** such that the cores can be fully utilized to deliver higher throughput

Acknowledgement

- Mom
- Vivienne Sze
- Sertac Karaman, Luca Carlone, Phillip Isola
- Amr Suleiman, James Noraky, Yu-Hsin Chen, Mehul Tikekar, Tien-Ju Yang, Hsin-Yu Lai, Gladynel Pena, Yannan Wu, Peter Li, Soumya Sudhakar
- Edward Adelson, Wenzhen Yuan, Bei Xiao, Lavanya Sharan, Derya Akkaynak, Rui Li, Shaoxiong Wang, Yu Zhang, Tianfan Xue, Chiyuan Zhang, Xuhong Zhang, Daniel Li, Ling Ren, Peng Wang, Jiajun Wu, Xue Feng, Sizhuo Zhang, Tianren Liu, Chengtao Li, Guolong Su, Xiao Fang, Yu Wang, Duanhui Li, Xi Yang, Yuan Zhang, Lixin Shi, Xiaoxue Wang, Dongying Shen, Bai Liu, Fangchang Ma, Trevor Henderson, and so many other colleagues and friends
- …

Questions?

$$\boldsymbol{O}\left(\lambda_m^2\lambda_z\right) \rightarrow \boldsymbol{O}(\lambda_m)$$