Learning to Explore Indoor Environments using Autonomous Micro Aerial Vehicles

Penn Engineering GRASP LABORATORY General Robotics, Automation, Sensing & Perception Lab

Yuezhan Tao, Eran Iceland, Beiming Li, Elchanan Zwecher, Uri Heinemann, Avraham Cohen, Amir Avni, Oren Gal, Ariel Barel, and Vijay Kumar

TECHNION Israel Institute of Technology

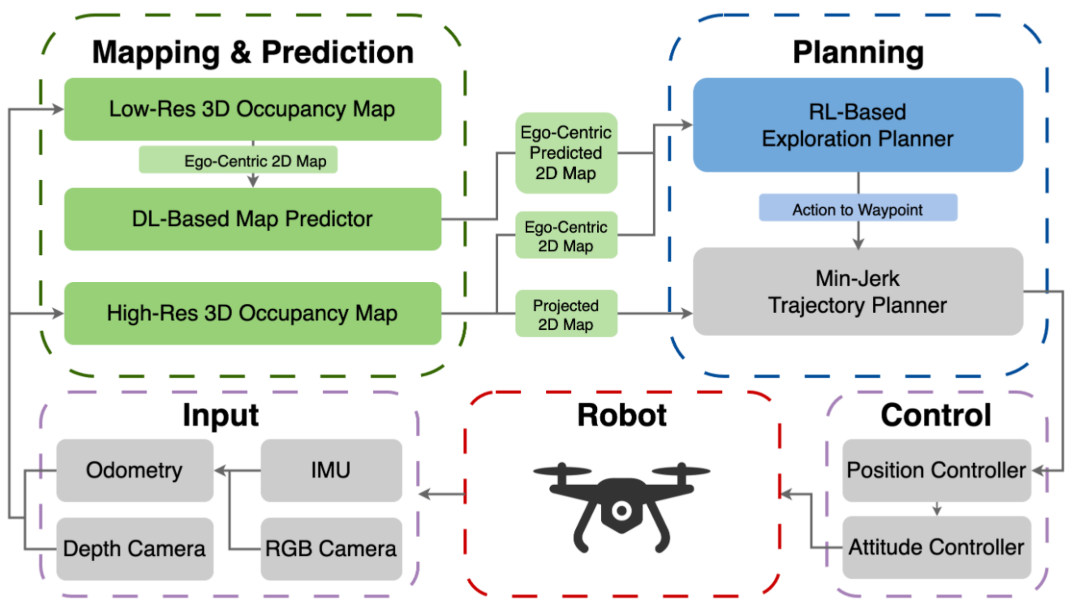
ICRA2024 YOKOHAMA | JAPAN

Enhanced Exploration of Unfamiliar Indoor Spaces with Autonomous Aerial Robots, Achieving over 50% Efficiency Improvement using Machine learning Tools

Motivation

- ❑ Autonomous exploration has many direct real-world applications
- ❑ Learn to explore instead of model-based approach
- ❑ Deploy on SWaP constrained MAVs

Proposed System



- ❑ A mapping & DL-based prediction module that construct occupancy maps and predict occupancy information for efficient exploration
- ❑ A DRL-based planning module that leverages the prediction and observations to select exploration actions that gather informative observations

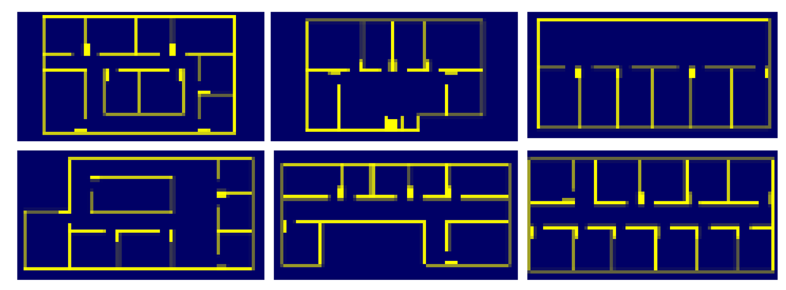
Hardware Platform

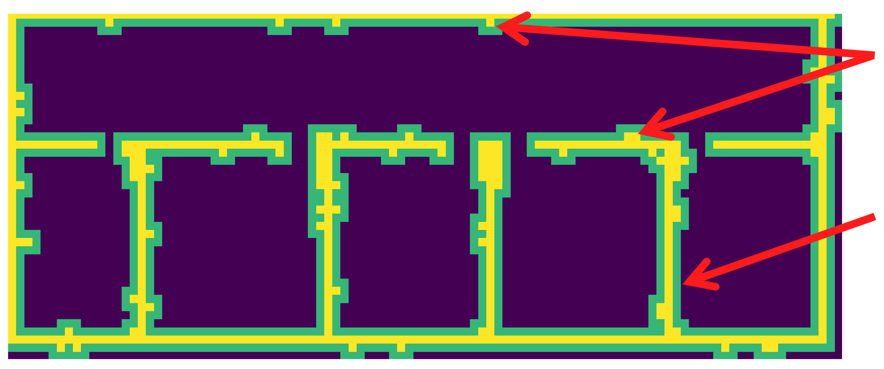
Pixhawk 4 Mini	Intel NUC 10
Intel RealSense D435i	ModalAI VOXL
Flight time ~6 mins	
Weight ~1.3kg	
3S 5200mAh Li-Po	

Methodology

Training Samples

- ❑ 50,000 synthetic examples
- ❑ Floor maps include doors and cabinets
- ❑ Furniture below 1.5M are ignored





Noise (yellow peaks)

Safety zone (green area next to walls)

Action Modeling

0 – stay put, 1 – turn right, 2 – turn left, 3 – move forward, 4 – turn right and move forward, 5 – turn left and move forward

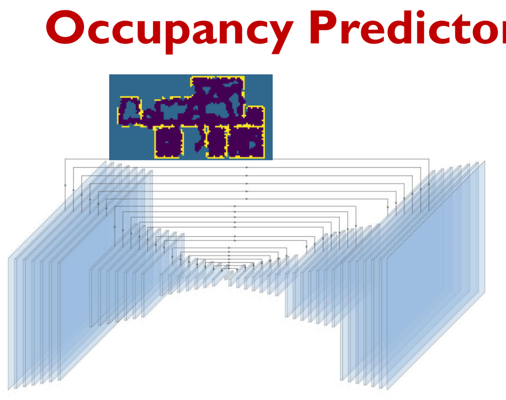
Observation Space

- ❑ Last action (to handle drone's dynamics)
- ❑ Building map in two resolutions

20 cm^2 per pixel

5 cm^2 per pixel

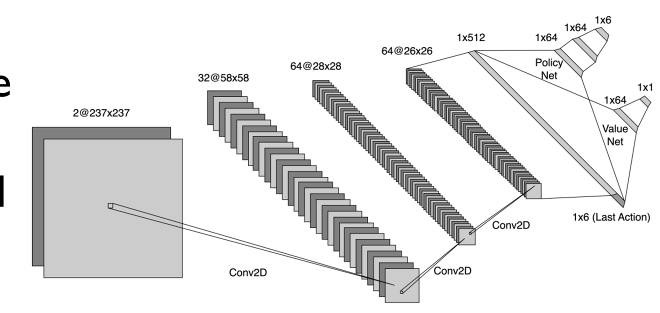
Occupancy Predictor



- ❑ Standard convolutional Encoder-Decoder:
  - 21 Encoding + 21 Decoding layers
  - Skip connections between each
- ❑ Dynamic thresholding during exploration
- ❑ $F_1 > 0.92$ with 98% coverage

RL Planner

- ❑ Standard CNN for feature extraction
- ❑ Additional fully connected layers (separate for actor and critic)



Reward

- ❑ Reward based on prediction
- ❑ $\mathcal{R}(t) = -1 + \begin{cases} -l & \text{if collision} \\ r(t) & \text{otherwise} \end{cases}$
- ❑ $r(t) = w \cdot [F_1(t+1) - F_1(t)]$
- ❑ $F_1 = TP/[TP + 0.5(N_M - T)]$
  - TP – Number of correctly predicted occupied cells
  - N_M – Number of interior cells
  - T – Total number of correctly predicted cells

Training

Train in two phases:

- ❑ Without prediction
- ❑ Gradually add predicted information

Results

Simulation Experiments

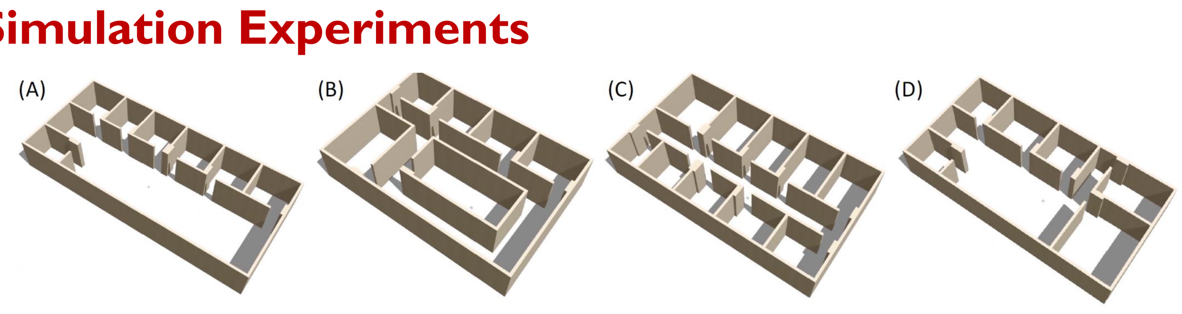
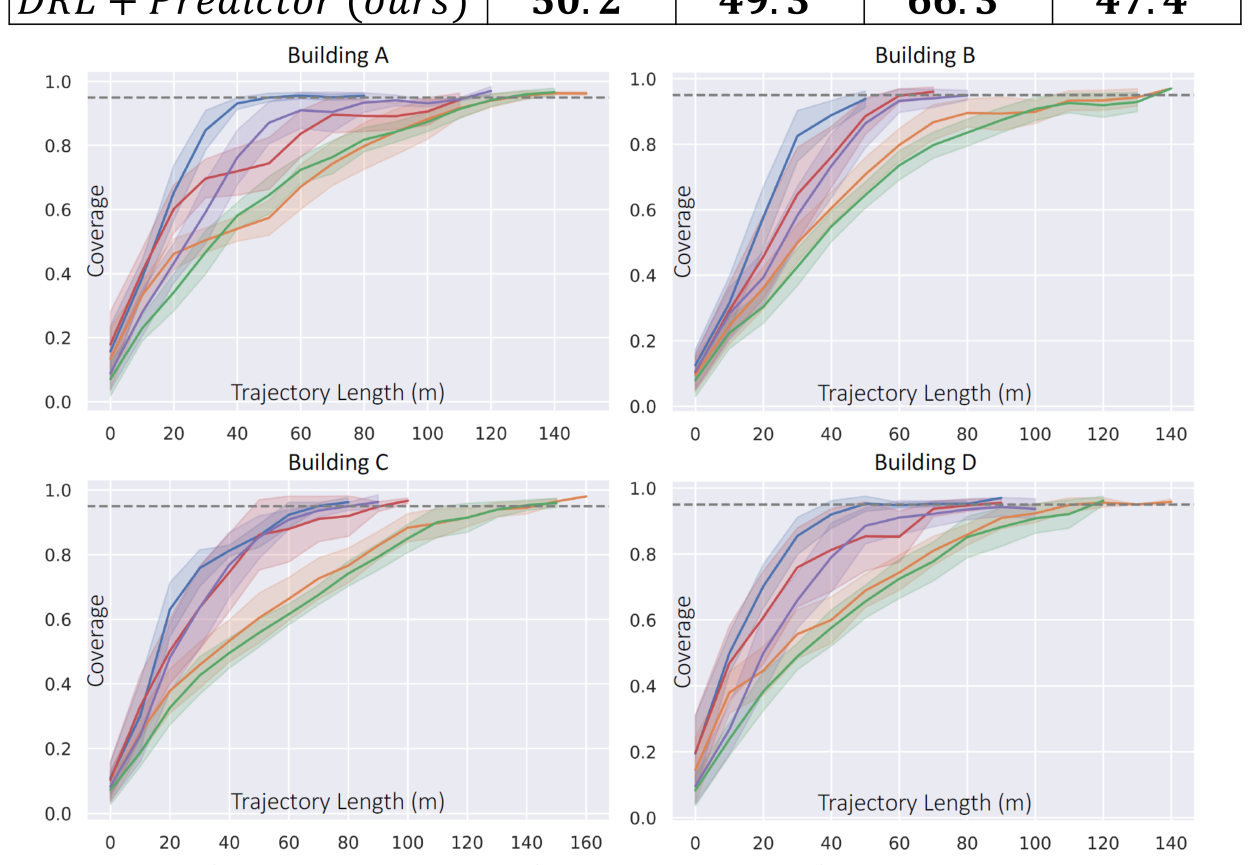
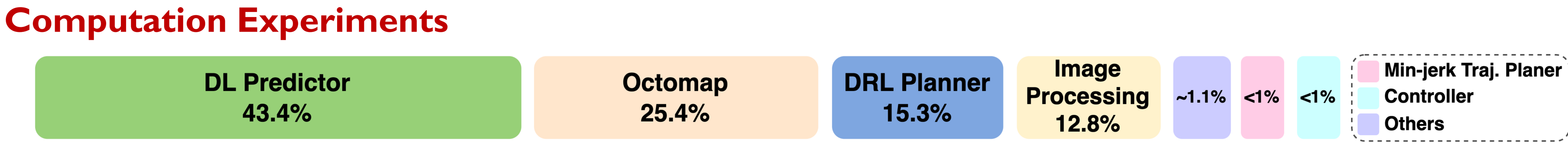


Table 1: Averaged mapping path length (m) in Gazebo

Method	(A)	(B)	(C)	(D)
Frontier	125.0	108.8	132.7	110.0
Fuel	123.6	122.0	128.6	106.0
Frontier + Predictor	83.0	60.4	69.2	63.0
FUEL + Predictor	77.3	64.7	74.7	67.8
DRL + Predictor (ours)	**50.2**	**49.3**	**66.3**	**47.4**

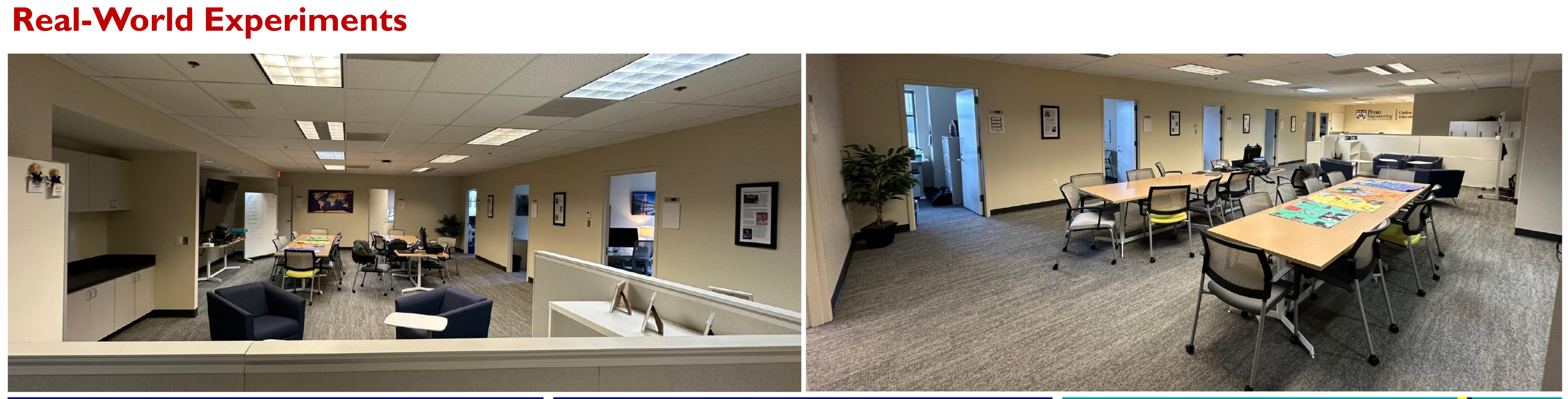


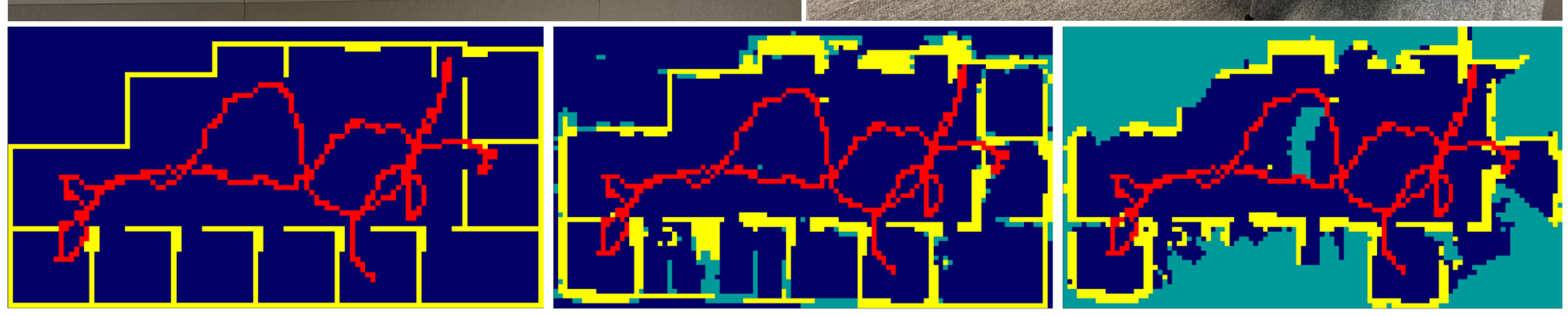
Computation Experiments



- ❑ **CPU Utilization 32.7% in total**
- ❑ **4 TOP Coral TPU: Predictor 150ms; Planner 8ms**

Real-World Experiments





The proposed method achieves a 50-60% shorter overall path length compared to the classic and the state-of-the-art methods

Scan QR for more details

arXiv:2309.06986