NMSL@NTHU

Networking and Multimedia Systems Lab

Enhancing Situational Awareness with Adaptive Firefighting Drones: Leveraging Diverse Media Types and Classifiers

Tzu-Yi Fan (joyfan2@gmail.com)

Advisor: Cheng-Hsin Hsu

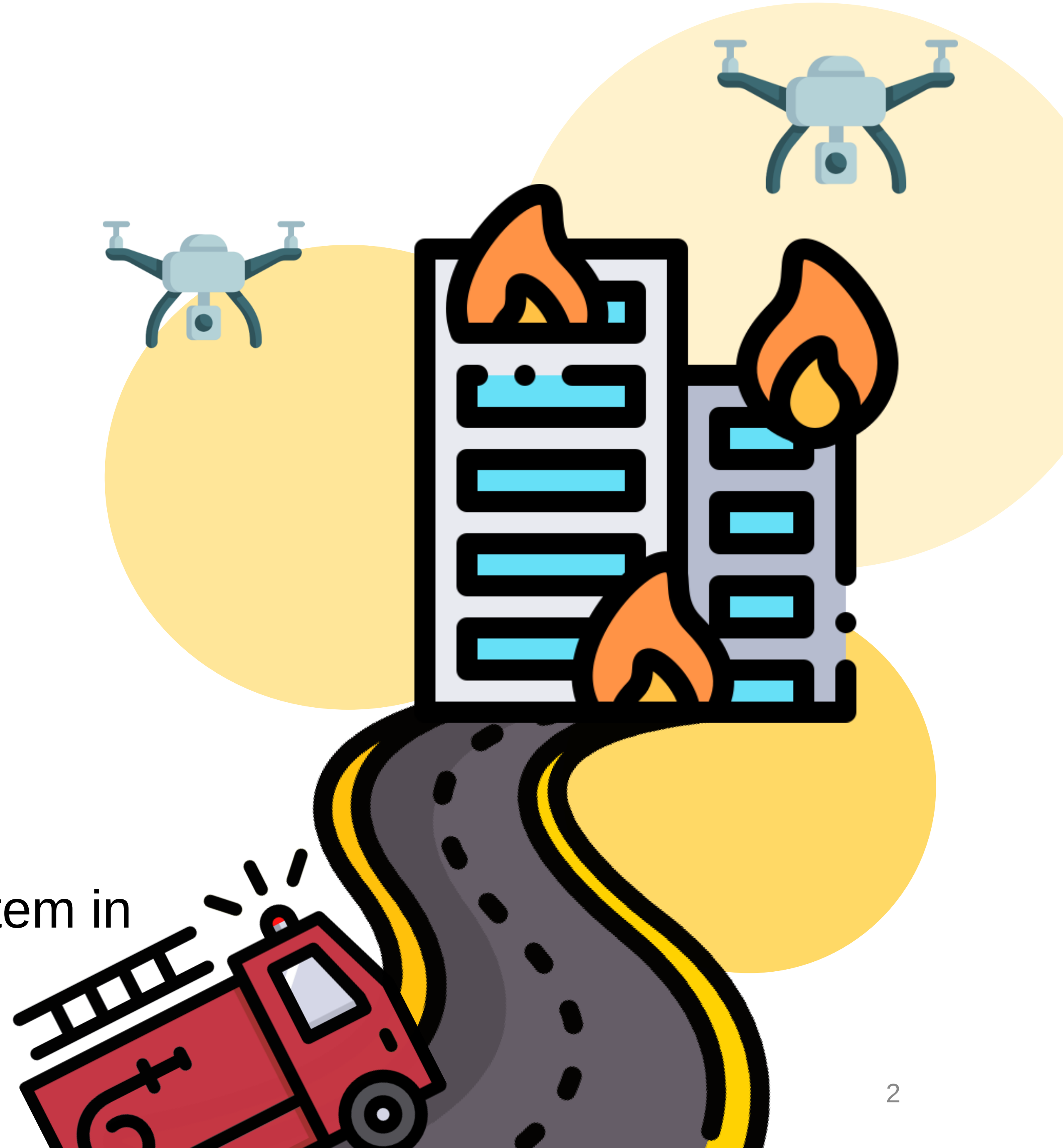
Networking and Multimedia Systems Lab, CS, National Tsing Hua University

img from: https://www.taiwannews.com.tw/en/news/3888278

http://www.fireemsleaderpro.org/2015/06/30/firefighters-smoke-disorienta

Outline

- Motivation
- Challenges & Goal
- Related Work
- System Overview
- Sensor Selection
- Design of Open Window Classifiers
- Measurement Selection Problem
- Measurement Selection Algorithm
- Implementation
- Evaluations
- Demo of a Complete Firefighting System in the Real World
- Conclusion & Future Work



Motivation

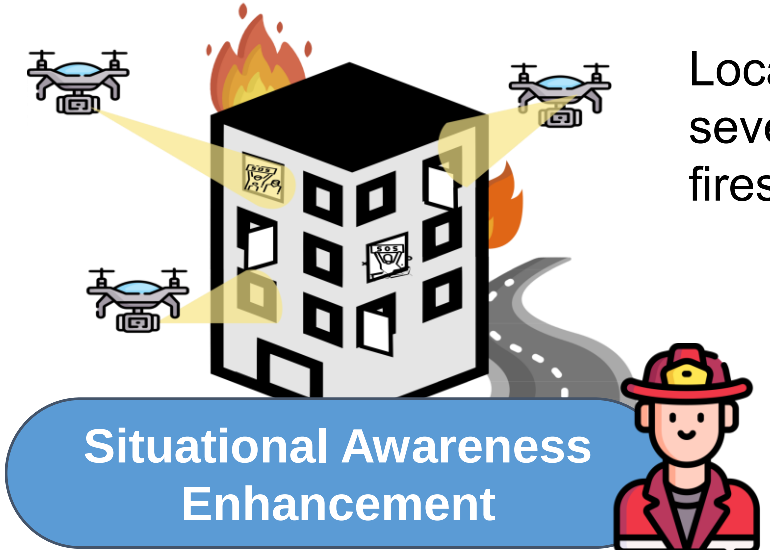
Importance of Situational Awareness

- High-rise buildings are too high to reach by fire ladders → Interior firefighting
- Unexpected situations may put firefighters in danger

- Situational awareness (detect critical situations in real time) is important

Locations and vital signs of humans

Locations and severity levels of fires



Regions affected by smoke

Window openness

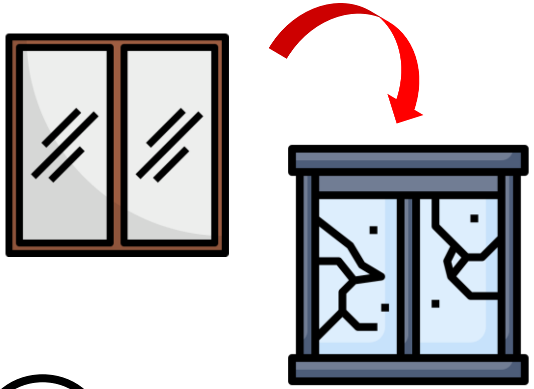
img from: https://www.highriseops.com/#gref

Challenges & Goal

Challenges in Situational Awareness

1 Height of high-rise buildings

2 Broken interior surveillance system



3 Changing situations

4 Multiple situations at once

Fast and correctly improve situational awareness

Drones,
which can fly around the high-rise buildings easily

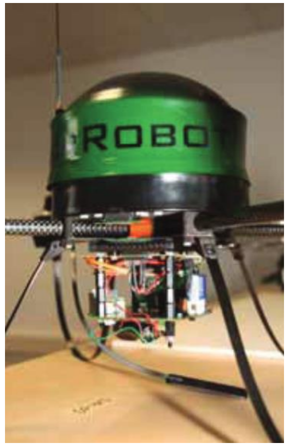
Multi-Modal Sensors,
which can detect provide various sensor data about the situations in high-rise fires

Heterogeneous Classifiers,
which can analyze the sensor data and provide the situation information

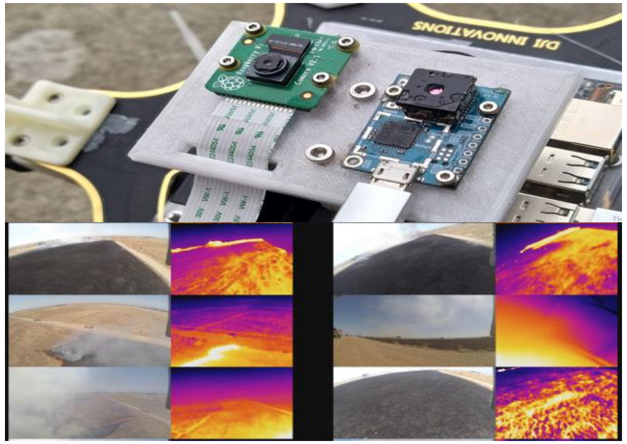
Related Work

Multi-Modal Sensors on Drones

[PE’12, SenSys’20]



A gas detector [PE’12]



A thermal, a RGB sensor, and sensor data. [SenSys’20]

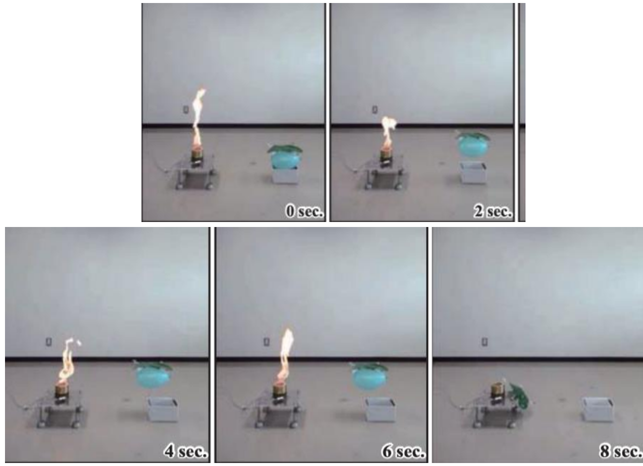
- Radiometer, gas and smoke detectors and a thermal [PE’12]
- Thermal and RGB sensors [SenSys’20]

Focus on fire detection only
Discuss the development of classifiers

Firefighting Drones

[IJIGSP’18, ROBIO’14]

Fire extinguishing system: quadrotor, gun, robot, and extinguishing balls. [IJIGSP’18]



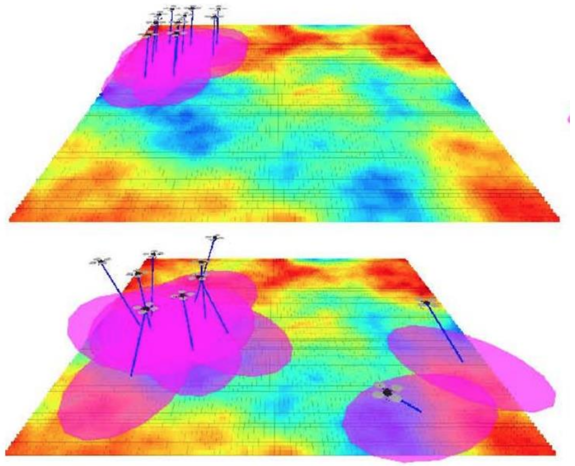
Snapshots of the flame extinguishment experiment. [ROBIO’18]

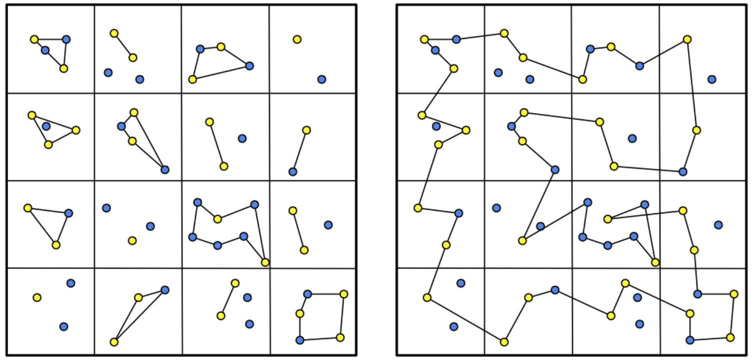
- Throw extinguishing balls [IJIGSP]
- Inject gas to extinguish fires [ROBIO’14]

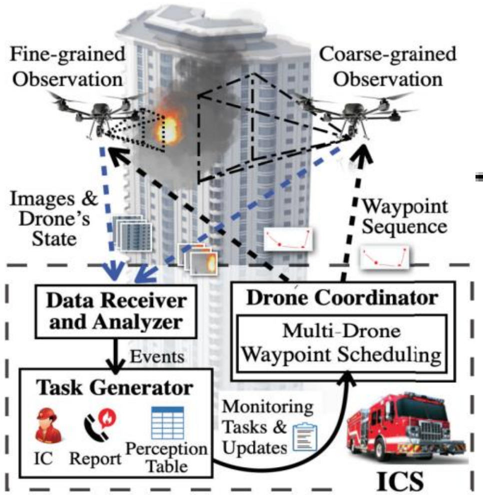
Focus on putting out fires

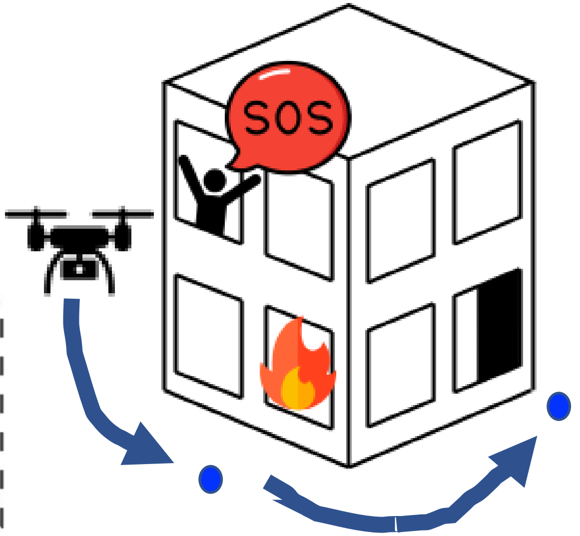
[SenSys’20]: T.Lewicki and K.Liu.2020.AerialSensingSystemforWildfireDetection:DemoAbstract.In Proc. of ACM SenSys. Yokohama, Japan, 595–596.
[PE’12]: W. Krull, R. Tobera, and et al. 2012. Early Forest Fire Detection and Verification Using Optical Smoke, Gas and Microwave Sensors. Procedia Engineering 45 (2012), 584–594.
[IJIGSP’18]: A. Alshbatat. 2018. Fire Extinguishing System for High-Rise Buildings and Rugged Mountainous Terrains Utilizing Quadrotor Unmanned Aerial Vehicle. MECS Press IJIGSP 11, 1 (Jan. 2018), 23.
[ROBIO’14]: S. Ogawa, S. Kudo, M. Koide, H. Torikai, and Y. Iwatani. 2014. Development and Control of an Aerial Extinguisher with an Inert Gas Capsule. In 2014 IEEE Intl. Conf. on ROBIO. 1320–1325.

Coarse-Grained Waypoint Scheduling for Drones









[Autom.Sci.Eng'15]:
- Designed a planner for drones to continuously monitor risky regions on a 2D map

[CDC'10]:
- Kept classification results valid in dynamic environments by a dynamic approach to patrol areas with drones

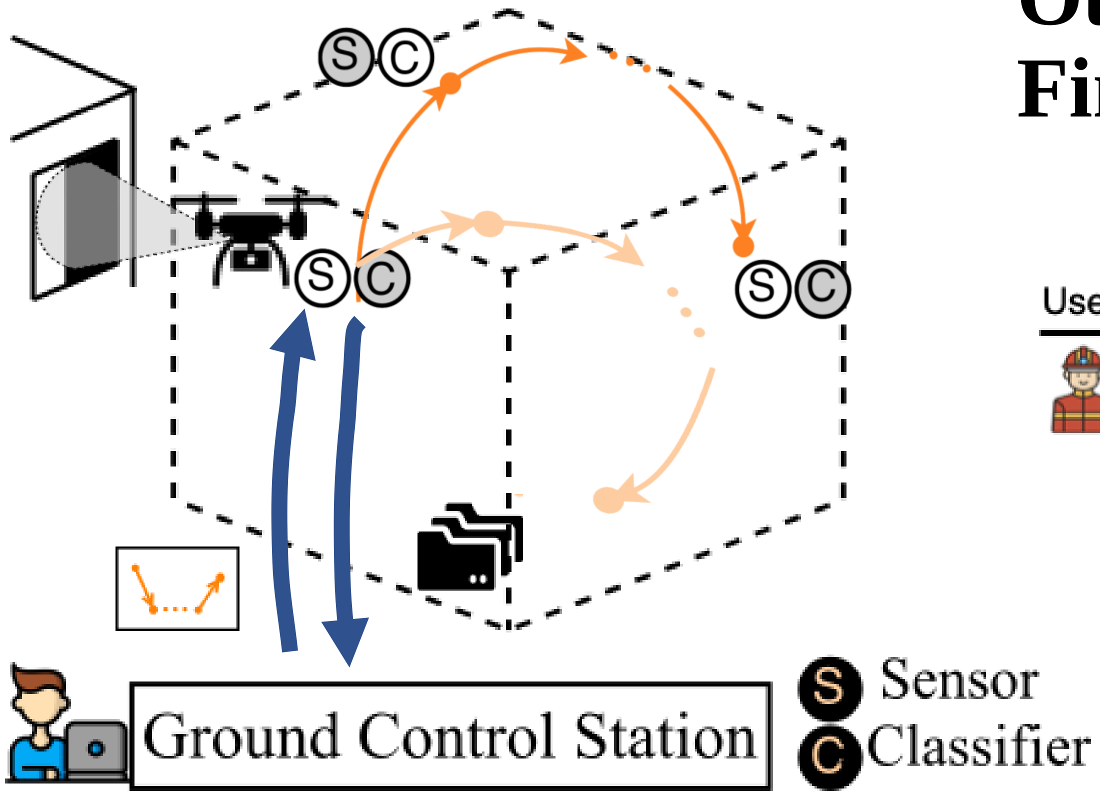
[SRDS'21]:
- Designed a drone-assisted high-rise fire monitoring system
- Computed waypoint schedules for drones to perform assigned tasks

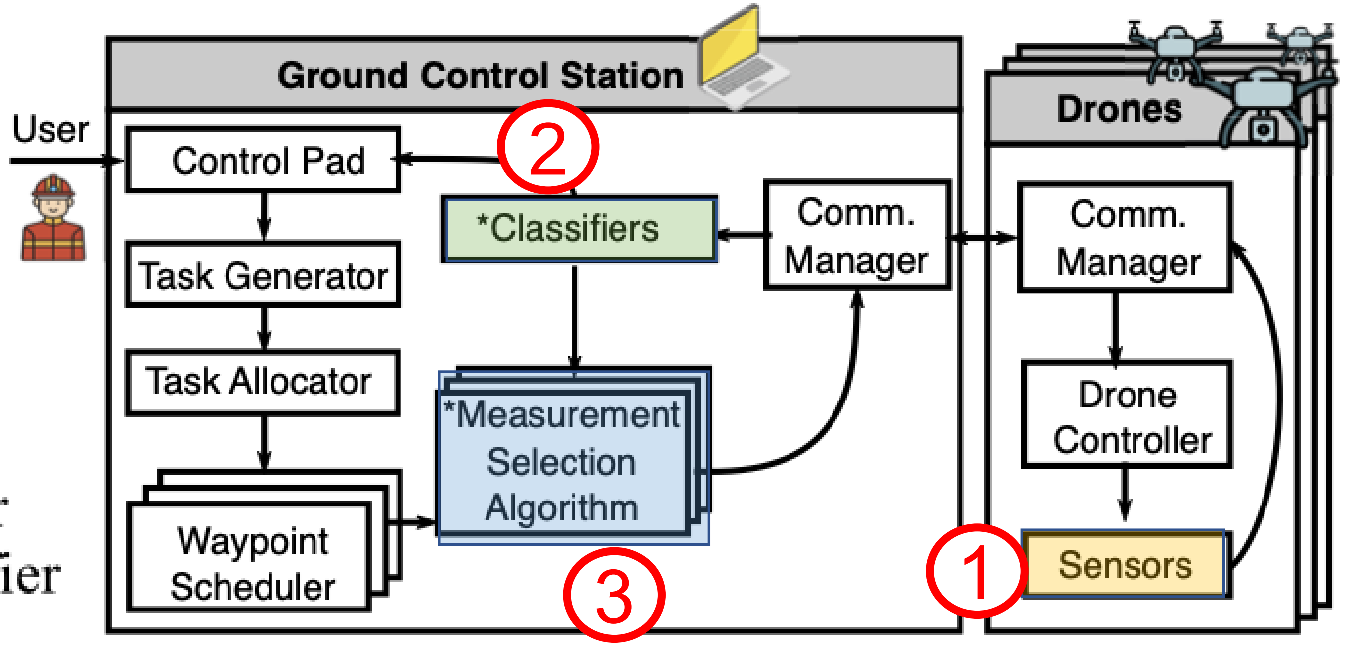
Not consider sensors, classifiers, and locations of measurement tasks

[Autom.Sci.Eng'15]A. Wallar, E. Plaku, and D. Sofge. 2015. Reactive Motion Planning for Unmanned Aerial SurveillanceofRisk-SensitiveAreas.IEEETrans.Autom.Sci.Eng.12,3(July2015),969–980.
[CDC'10] S.SmithandD.Rus.2010. Multi-Robot Monitoring in Dynamic Environments with Guaranteed Currency of Observations. In Proc. of IEEE CDC. Atlanta, GA, 514–521.
[SRDS'21] F. Liu, T. Fan, C. Grant, C. Hsu, and N. Venkatasubramanian. 2021. DragonFly: Drone-Assisted
High-Rise Monitoring for Fire Safety. In Proc. of IEEE SRDS. Virtual, 331–342.

System Overview

Our Work:
Fine-Grained Measurement Selection





Classifiers

- Analyze sensor data

Sensors

- Collect data of interested situations

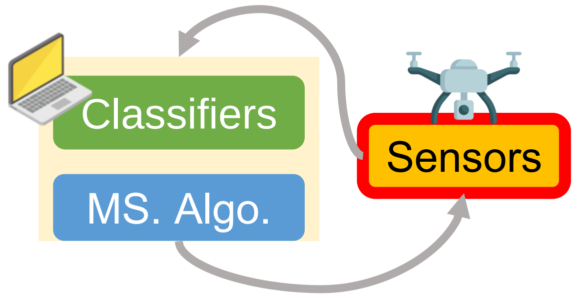
Measurement Selection Algorithm

- Instruct the detection location, and the usage of sensors and classifiers

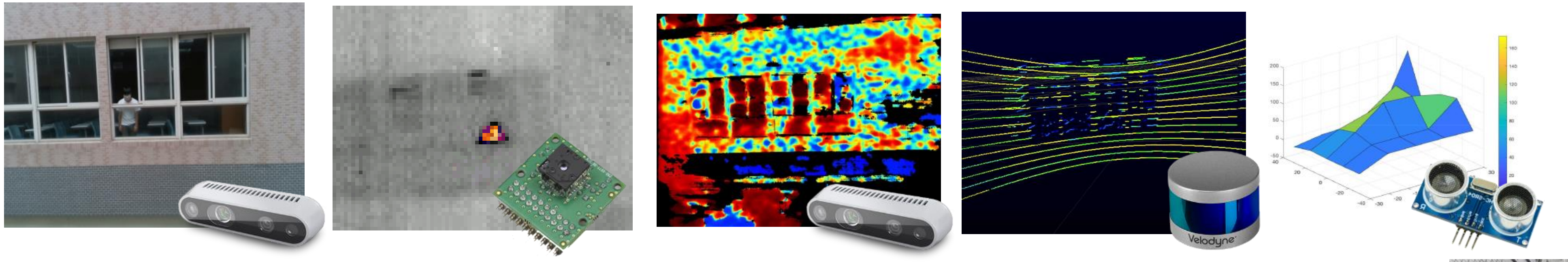
- Classifiers and measurement selection algorithm can be placed on drones or ground control station

Sensor Selection

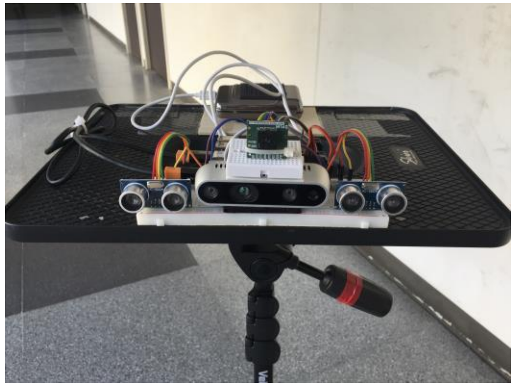
WinSet: The First Multi-Modal Window Dataset
[BuildSys’21]

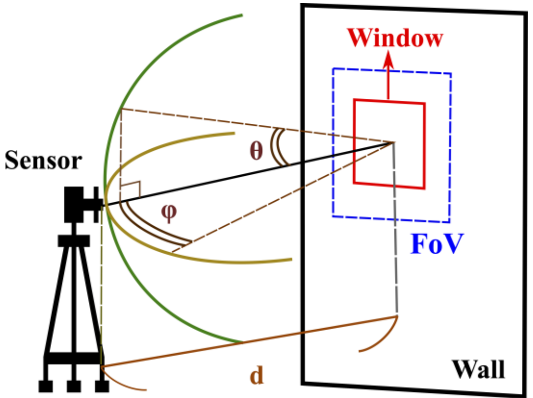


- Sensors: RGB, thermal, depth, LiDAR, and ultrasound



- Steps:
  - Set all sensors on a platform with a Rpi
  - Mount the platform on a tripod
  - Take window images for each sensor at different distances *d*, polar angles *θ*, and azimuthal angels φ at multiple window states

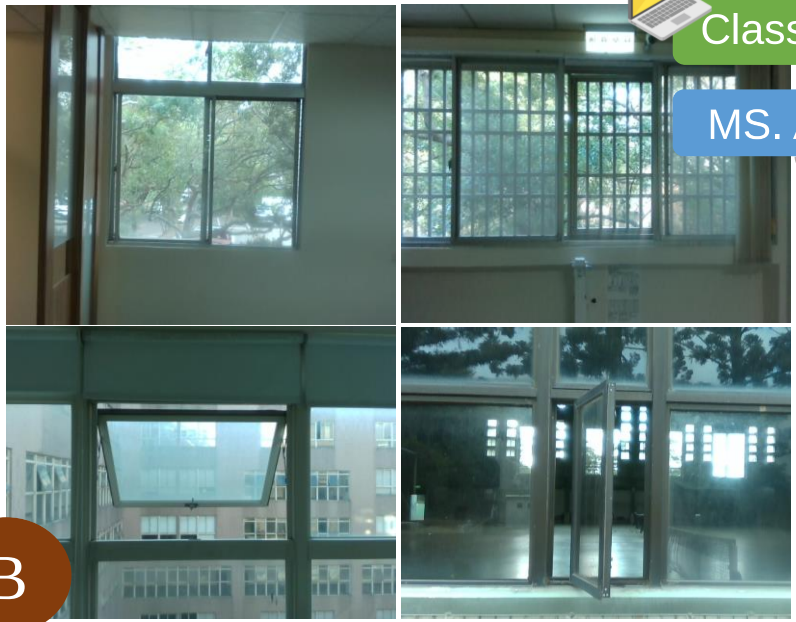




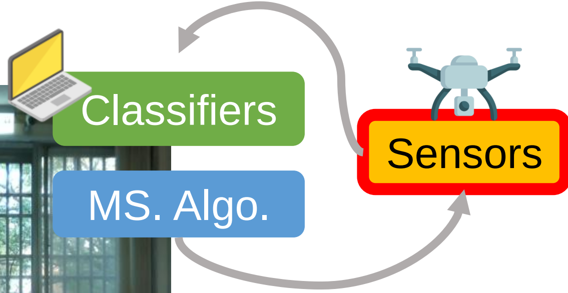
[BuildSys’21]T. Fan, T. Tsai, C. Hsu, F. Liu, and N. Venkatasubramanian. 2021. WinSet: The First Multi- Modal Window Dataset for Heterogeneous Window States. In Proc. of ACM BuildSys. 192– 195.

WinSet: Dataset A and B

A

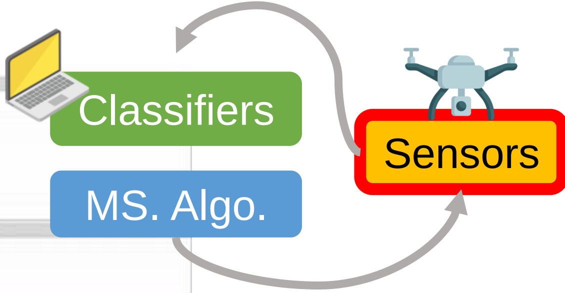


B

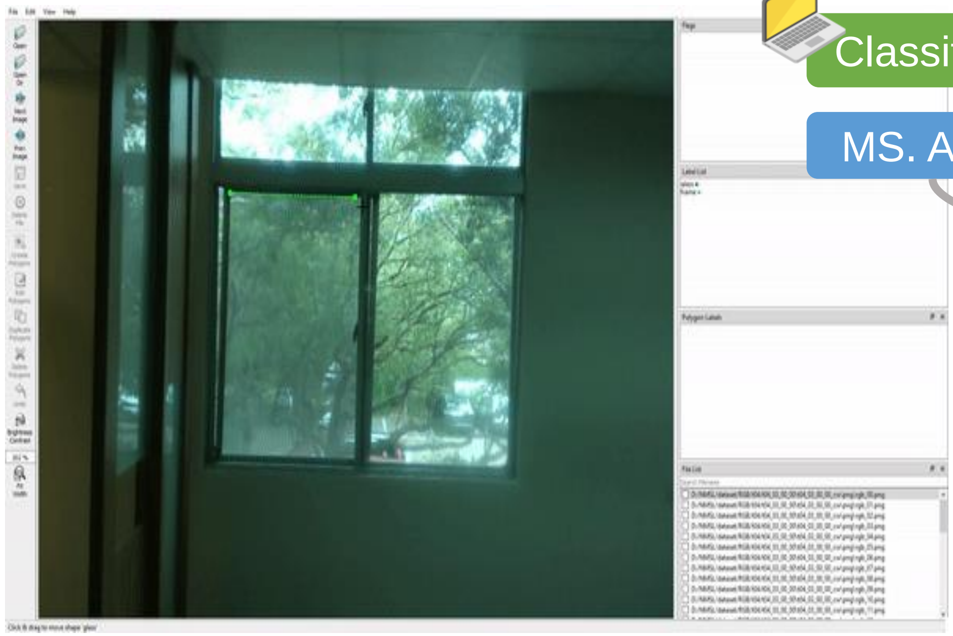


	A Different angles/distances/states	B Different window types
Sensor	RGB, Thermal, Depth, Lidar	RGB, Thermal, Depth, Ultrasound
Distance	3 m, 6 m, 12 m	1 m, 2 m, 3 m
Polar angle	0°, 30°	0°
Azimuthal angle	0°, 30°, 60°	0°
Window State	Openness, Human Behind, Lighting	Openness
Sample, Window Type	4, Sliding/Awing	12, [Sliding, Casement, Awing] x [Pure, Screen, Curtain, Barred]

Semantic Labeling

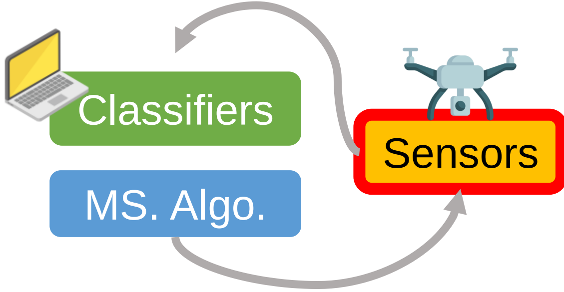


- Provide groundtruth for users
- Tool: Labelme
- File format: a .png with label number for each pixel

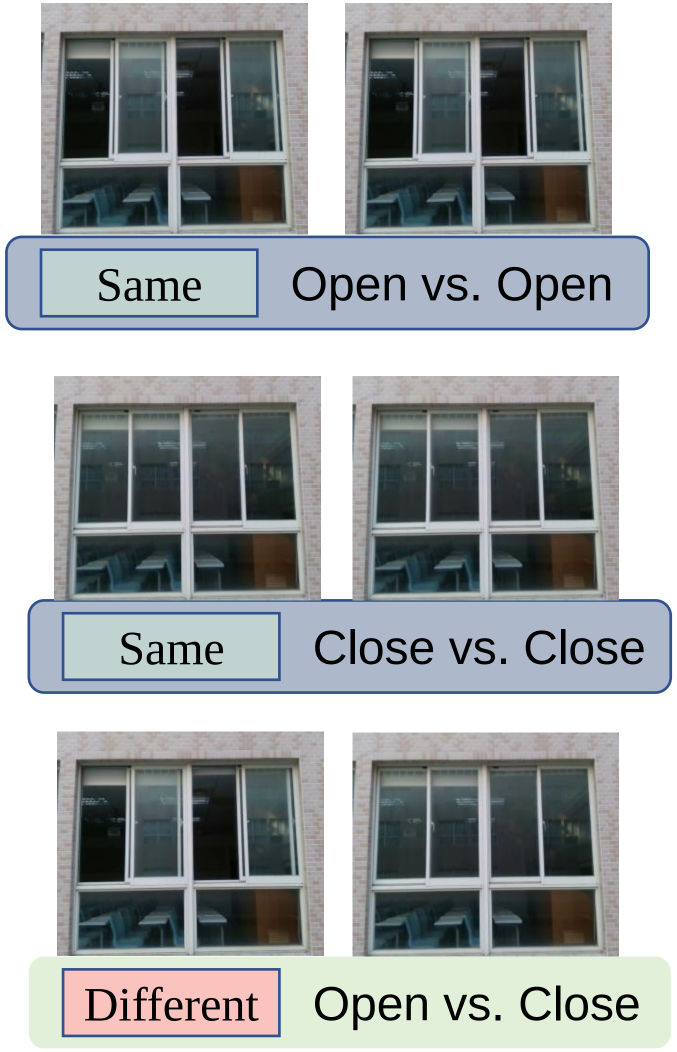


0	0	1	1	1	1	3	3
0	0	1	2	2	1	3	3
0	0	1	2	2	1	0	0
0	0	1	2	2	1	0	0
0	0	1	2	2	1	0	0
0	0	1	2	2	1	0	0
0	0	1	2	2	1	0	0
0	0	1	1	1	1	0	0

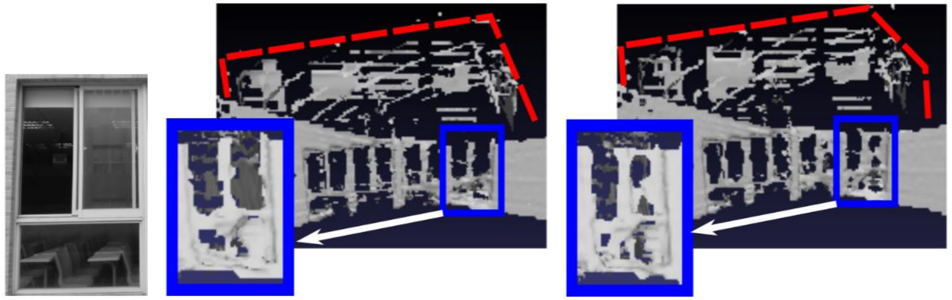
Distinguishability of Different Sensors



- Goal:
  Find the sensor with the best distinguishability among different window states
- Data:
  - Images from dataset A with openness window state
  - Experiment at different distances
- Idea:
  - Same window states: open vs. open, close vs. close → Low distinguishability
  - Different window states: open vs, close → High distinguishability
- Metrics:
  - Histogram Correlation (HC): value ↓, distinguishability ↑
  - Number of Gaussian till Homogeneity (NG): value ↑, distinguishability ↑



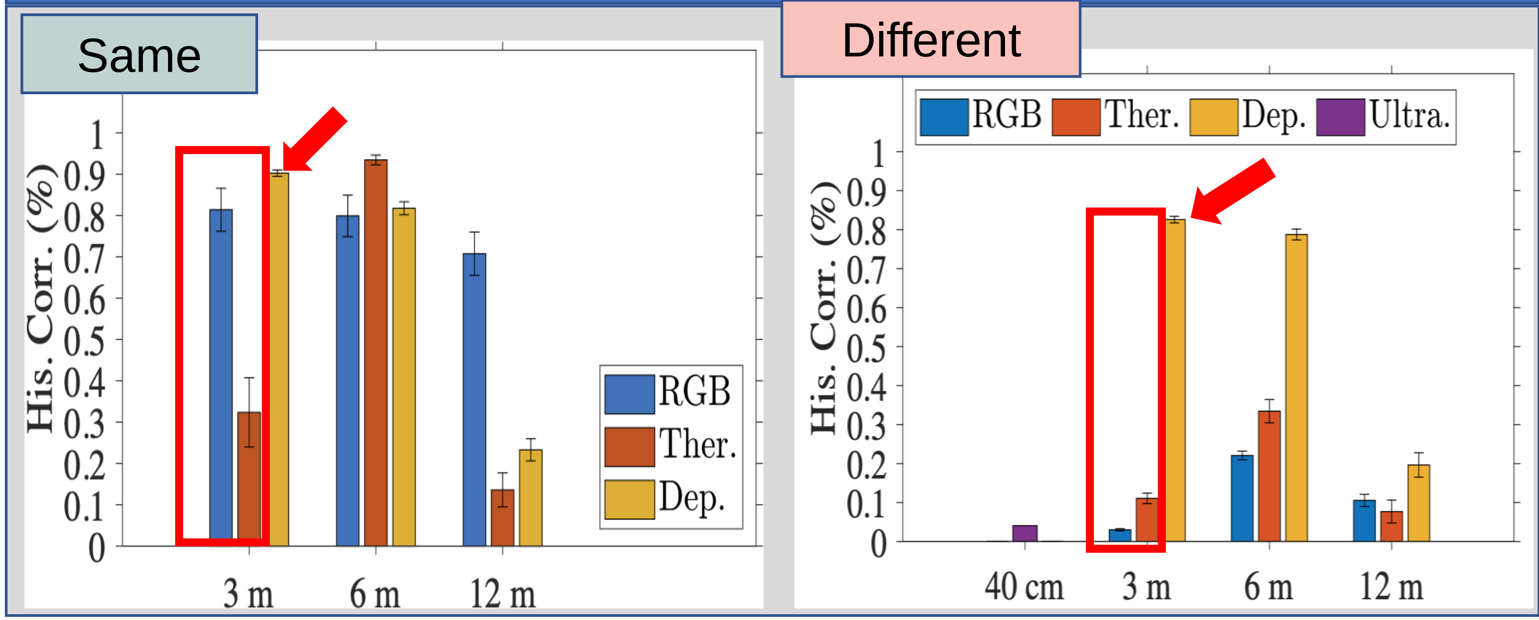
Distinguishability of Different Sensors



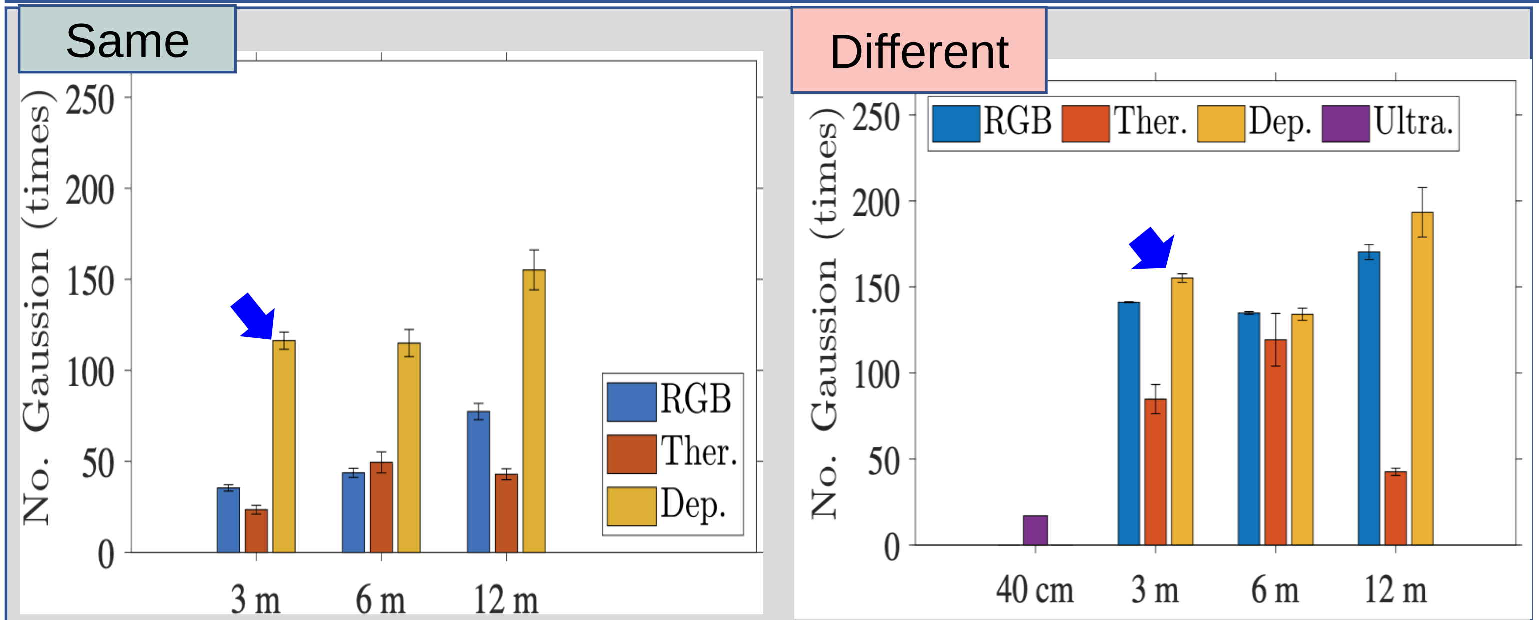
Close window   Open window

- LiDAR could detect classroom behind window in open/close windows
  → LiDAR with low distinguishability
- Depth does not work at all
- Both RGB and thermal work, but <u>RGB works better than thermal</u>

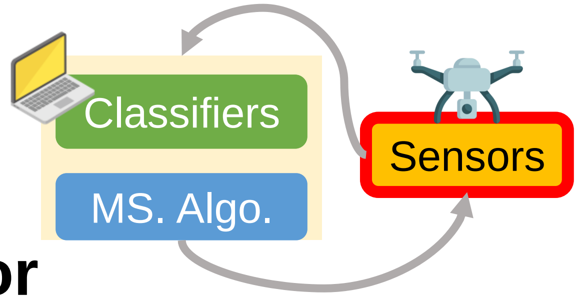
HC: value ↓, distinguishability ↑

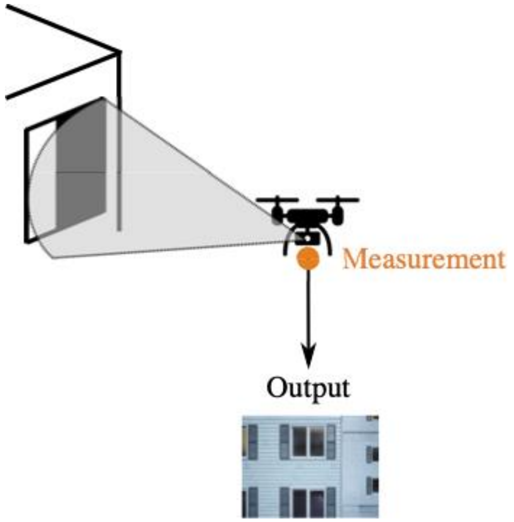


NG: value ↑, distinguishability ↑



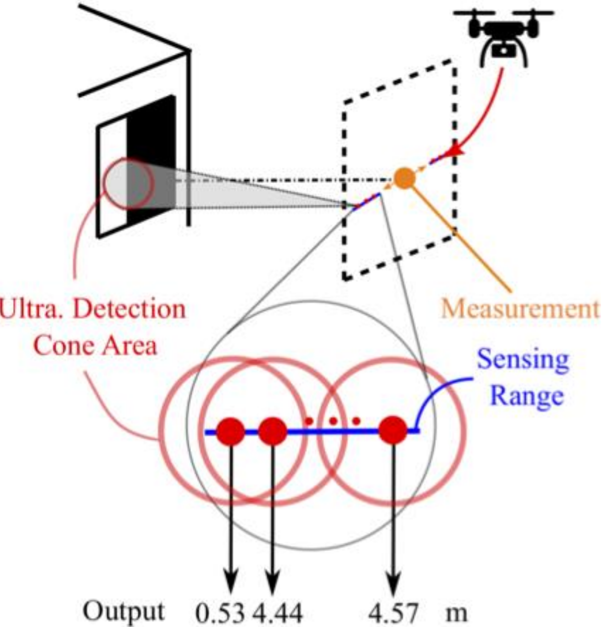
Class of Multi-Modal Sensors





1

One-shot sensor
- Get rich media data at one location
- Use data at one location as one measurement
- E.g., RGB, depth, and thermal cameras

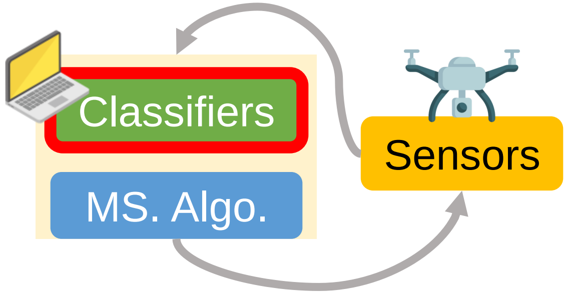


2

Accumulated sensor
- Get few media data from one location
- Combine data at multiple locations as one measurement
- E.g., ultrasound, humidity, temperature sensors

Design of
Open Window Classifiers

How Open Window Classifiers Work



Sensor Data

Result & Certainty

EX 1

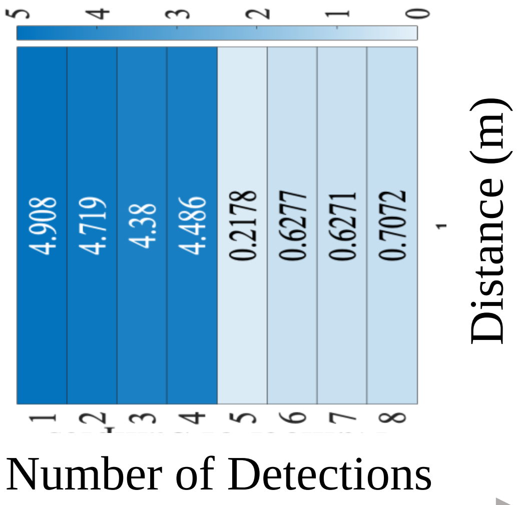
RGB

Open Window Classifier

Result: Close
Certainty: 83%

EX 2

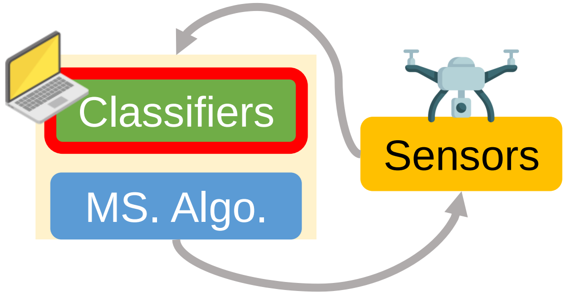
Ultrasound

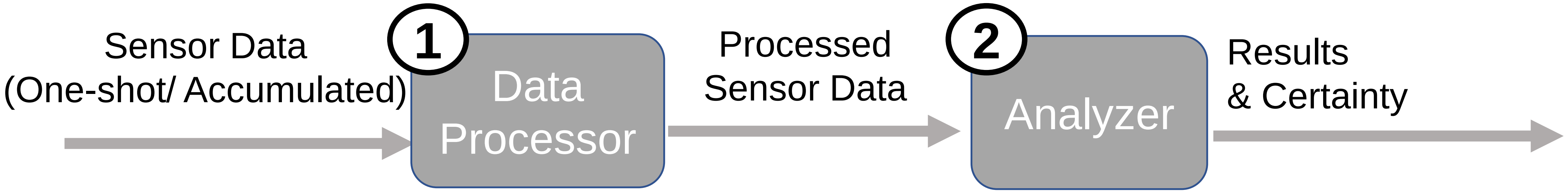


Open Window Classifier

Result: Open
Certainty: 99%

Open Window Classifiers Components





- Analyzers:
  - Histogram
  - Sum of Absolute Difference (SAD)
  - Oriented Fast and Rotated Brief (ORB)
  - Support Vector Machine (SVM)
  - Random Forest (RF)

Traditional image processing

Machine learning

Metrics for the Classifier Performance

Abbr.	Meaning
R	Result
Cer	Certainty
GT	Groundtruth

- Certainty: how confident the classifiers are of one classification

R: Open
Cer: 60%

R: Open
Cer: 83%

R: Close
Cer: 73%

- Accuracy: correctness of the classifiers from a large dataset

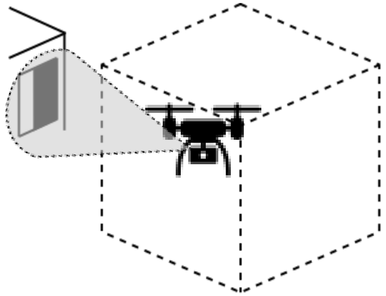
R: Open
GT: Close

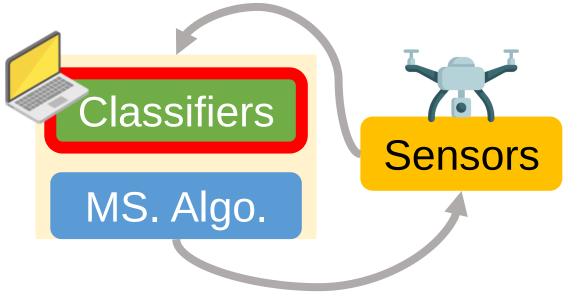
R: Open
GT: Open

R: Close
GT: Close

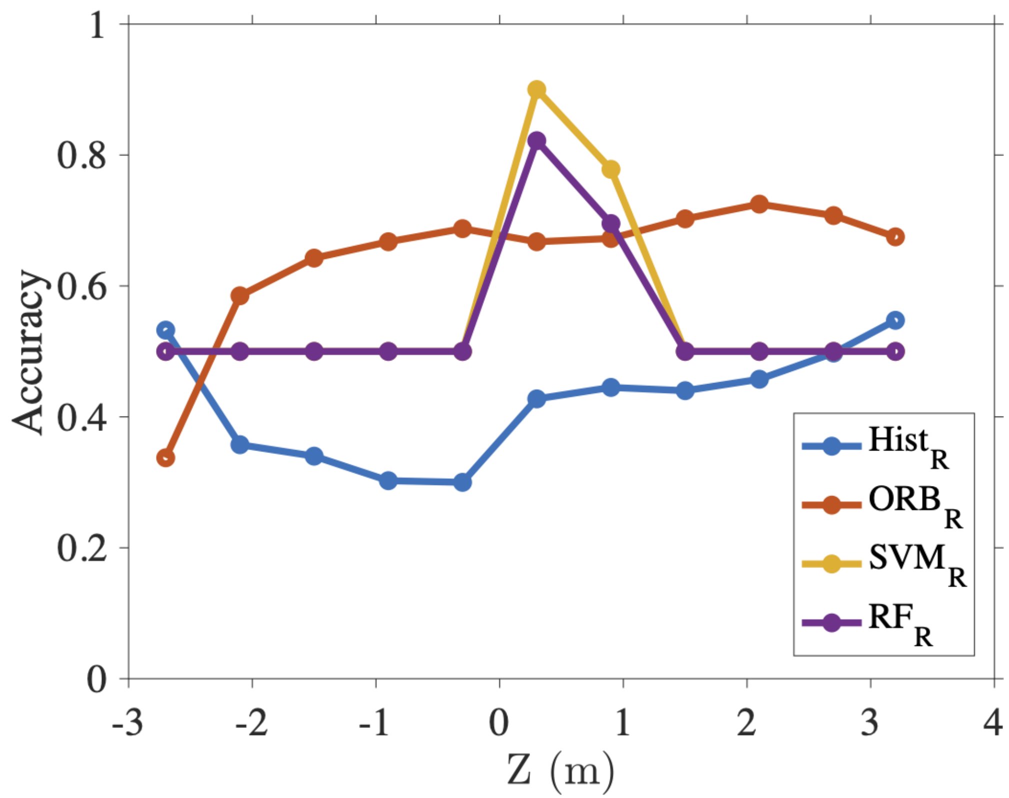
Acc: 66% (2/3)

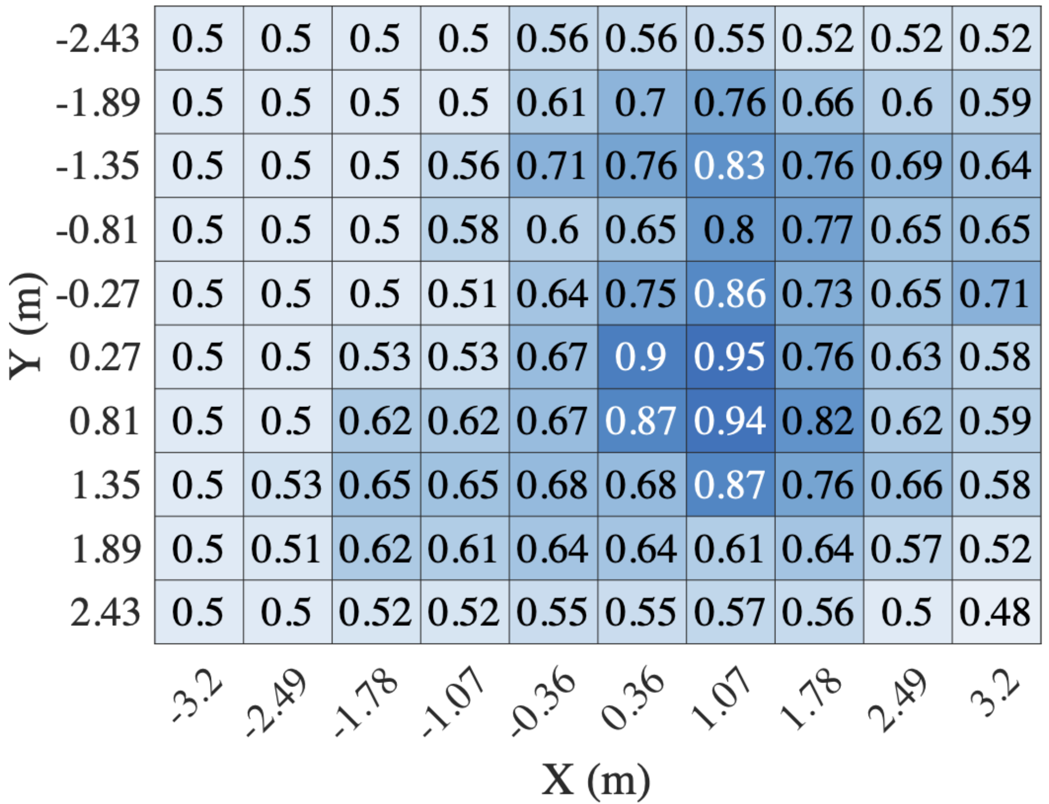
Site Survey





- Assume users need to do site survey before using our system
- Use sensors to collect data for several windows in the defined locations
- Evaluate the accuracy and avg certainty in the defined locations

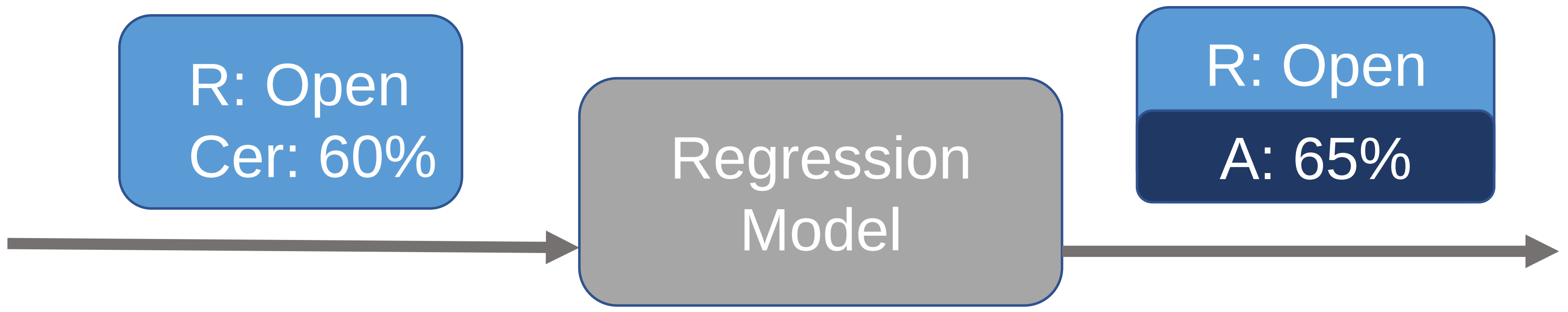




Certainty is Related to Accuracy [JBR'19]

Abbr.	Meaning
R	Result
Cer	Certainty
GT	Groundtruth

- Get accuracy and avg. certainty of our classifiers at different detection locations from site survey
- Make regression models for each classifier
- Use the regression model to map the certainty to an accuracy value



[JBR'19] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar. 2019. Customer Churn Prediction in Telecommunication Industry Using Data Certainty. Elsevier JBR 94 (Jan. 2019).

Human Classifiers (Ongoing)

Sensor Data

Result & Certainty

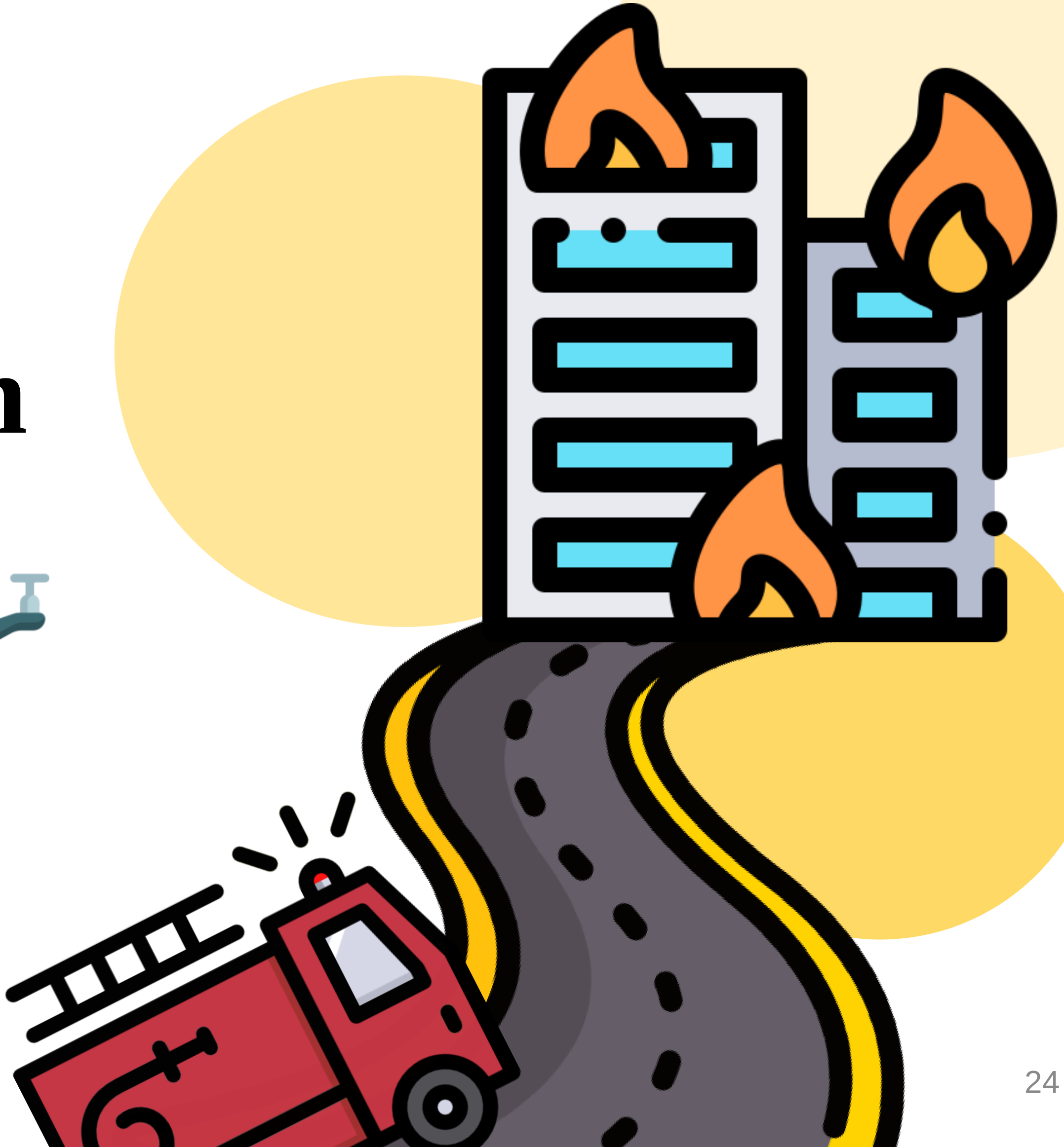
Human Classifier

Result: 1
Certainty: 83%

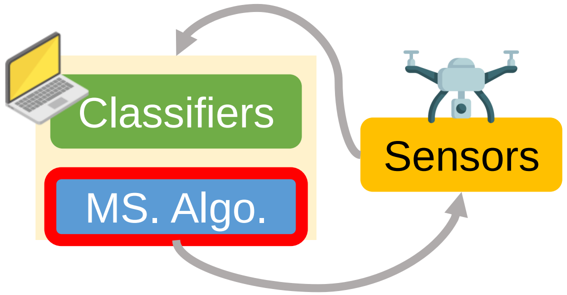
Human Classifier

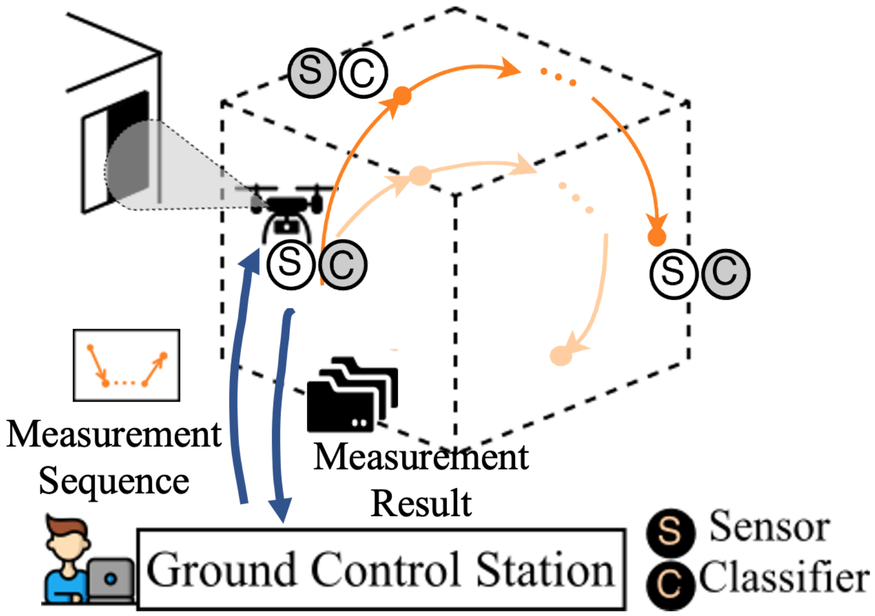
Result: 4
Certainty: 74%

Measurement Selection Problem



What is a Measurement Sequence?





- A series of measurements with specific locations, sensors, and classifiers

M → M → … → M

Location: (1, 0, 4)
Sensor: RGB
Classifier: ORB

L: (3, 0, 2)
S: Ultra.
C: SAD

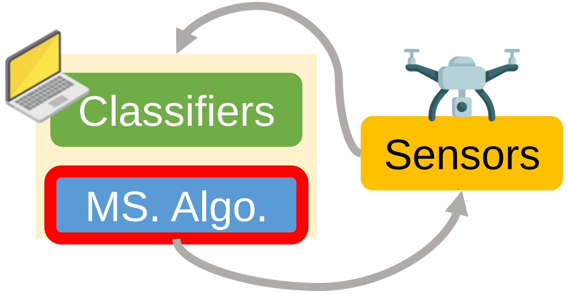
L: (2, 3, 5)
S: Ultra.
C: SVM

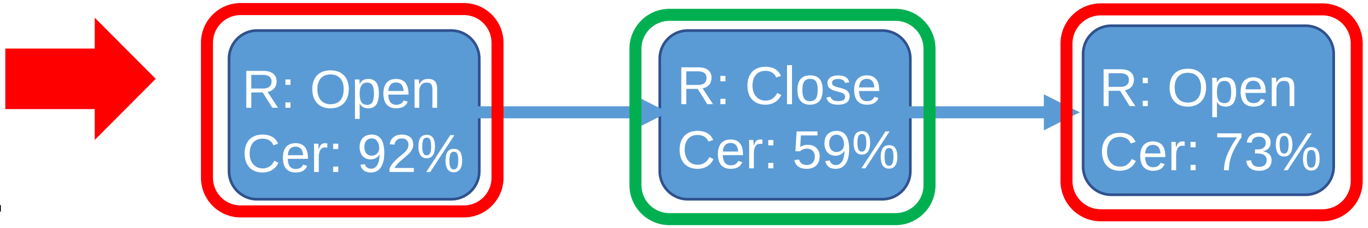
- A series of classification results from the measurement sequence

R: Open
Cer: 92%
→
R: Close
Cer: 59%
→ … →
R: Open
Cer: 73%

How to get the *final* result for a measurement sequence?

Get the Final Result from a Measurement Sequence





1 • Majority vote:

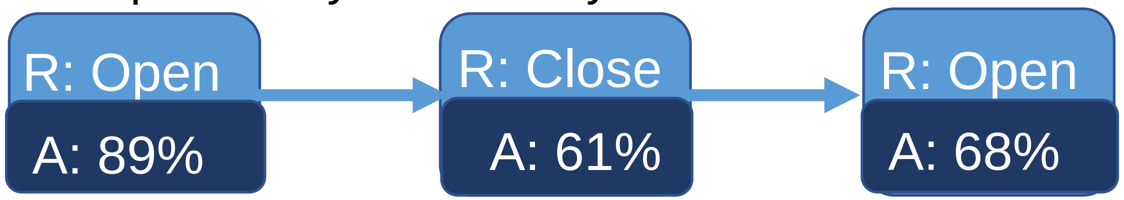
- Select the class with the most vote

  e.g., vote for Open: 2, vote for Close: 1 → Open

Consider results only

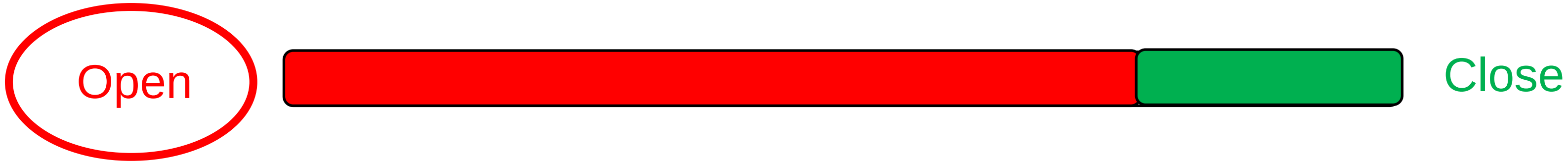
2 • Probability-based:

- Use regression models to map certainty to accuracy



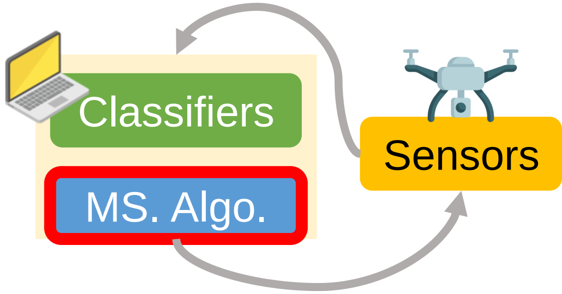
Consider both results and accuracy

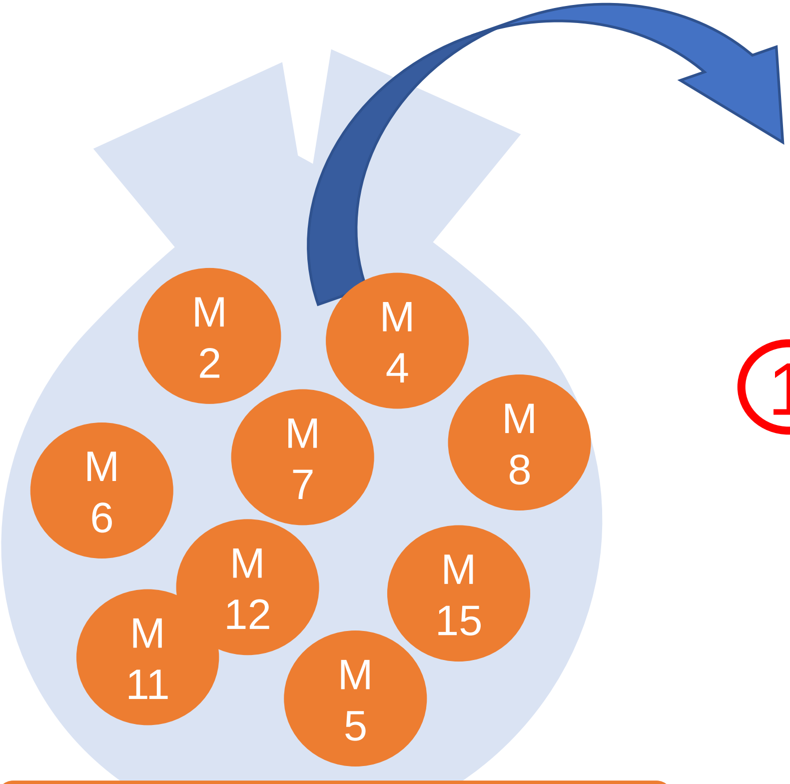
- Calculate the correct probability of open and close individually



- Select the class with the largest correct probability

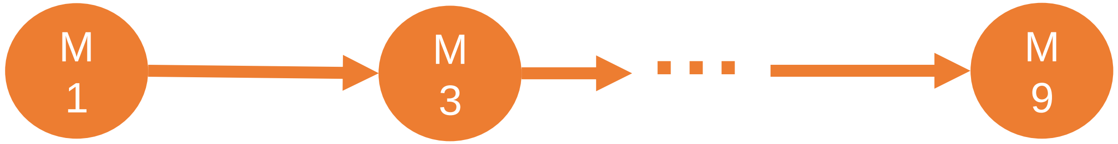
Measurement Selection Problem





Candidate Measurements

[Location] x [Sensor] x [Classifier]



- We have a set of candidate measurements

1 Select a measurement sequence from the candidate measurements

- We expect the measurement sequence can achieve:

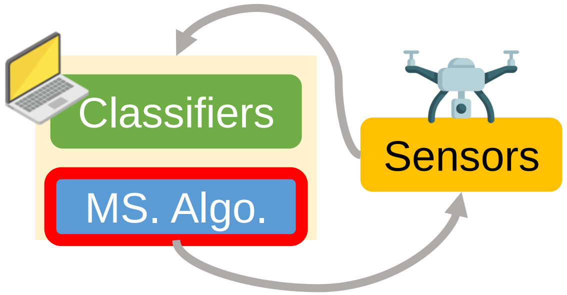
Highest Accuracy 

Shortest Measurement Time 

2 Dynamically adapt the MS. according to the classification results before the sequence is completed

How to quantify an accuracy and time consumption for a measurement sequence?

Quantify an Accuracy for a Measurement Sequence



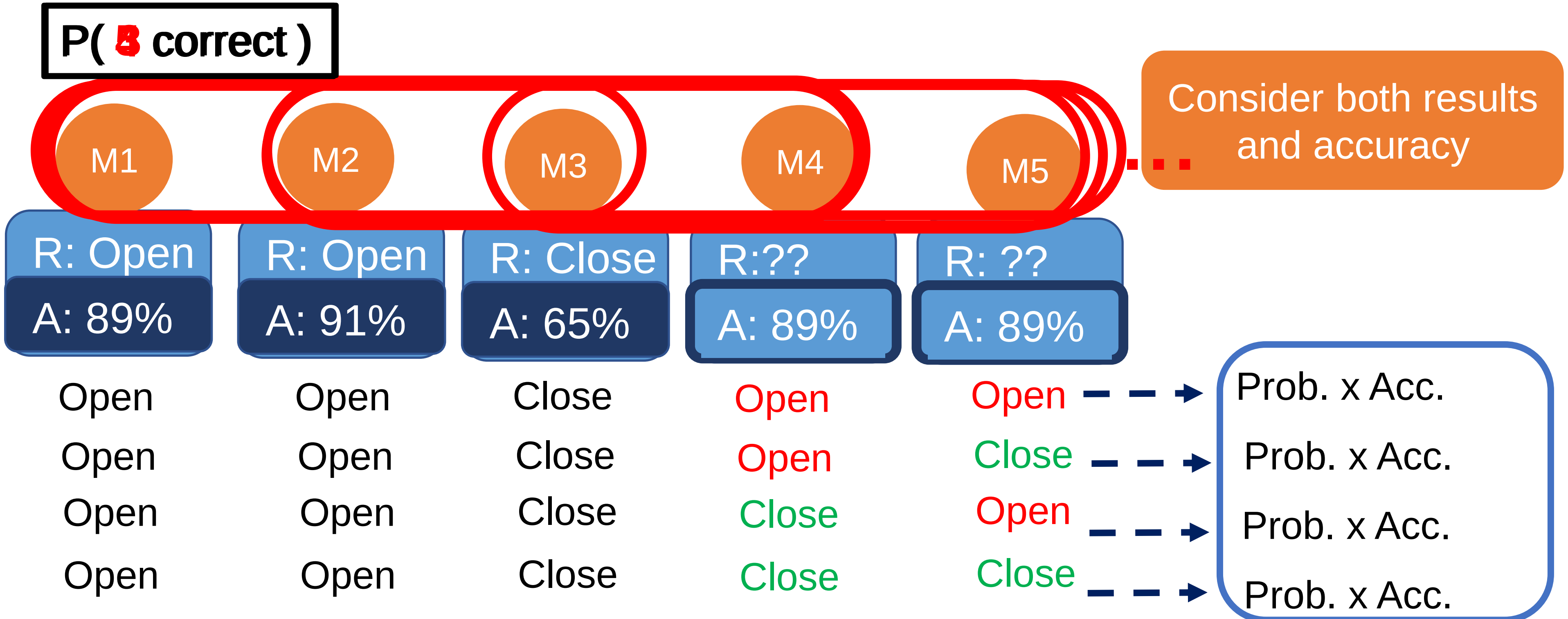
1 Majority vote:

- Compute the prob. that the results from half of the measurements are correct
- E.g. A_M = P(3 correct) + P(4 correct) + P (5 correct)

2 Probability-based:

- Create all possible MS.
- Calculate Prob. x Acc for each possible MS.
- Accumulate Prob. x Acc. of each possible MS. to get an accuracy expectation

Consider accuracy only



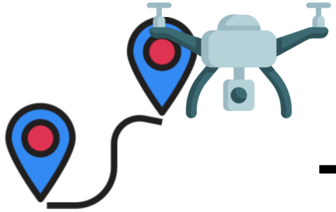
Time Consumption for the Measurement Sequence

- Total time of a measurement:

Measurement Sequence Time =

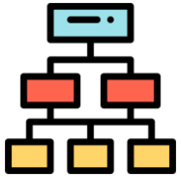
M.S. Algo. Running Time +

Data Transferring Time +



Moving Time +

Sensing Time +



Classification Time

Formulation

- Objective function: maximize the utility score

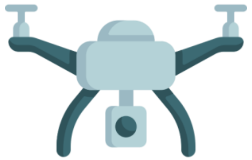
$$\max_{\mathbf{L}} U(\underline{A(\mathbf{L})}, \underline{T(\mathbf{L})}) = \sqrt{(1 - e^{-\alpha A(\mathbf{L})}) \times e^{-\beta \frac{T(\mathbf{L})}{\hat{T}}}}$$

- Higher utility score caused by:

High Accuracy

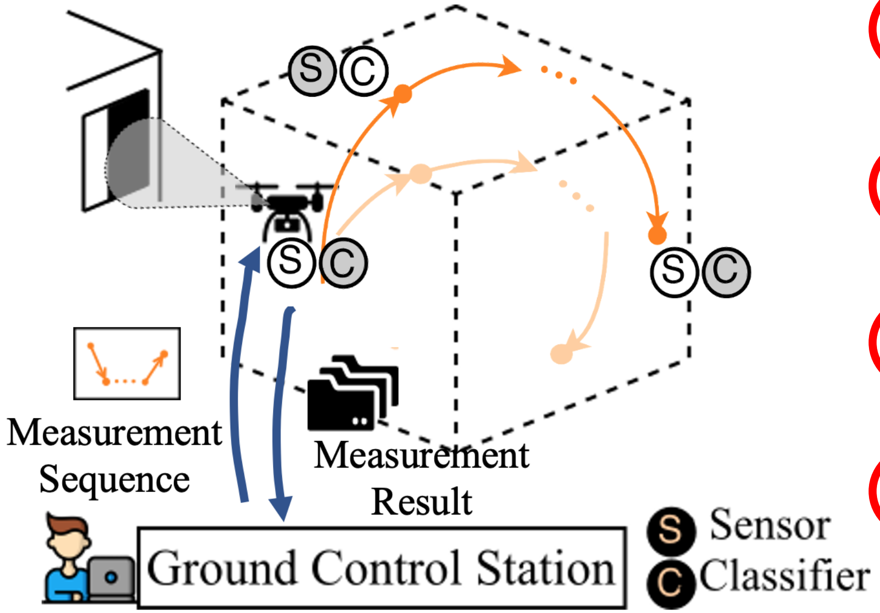
Low Time Consumption

- Constraints:
  - Accuracy > target accuracy
  - Time cost < time limit

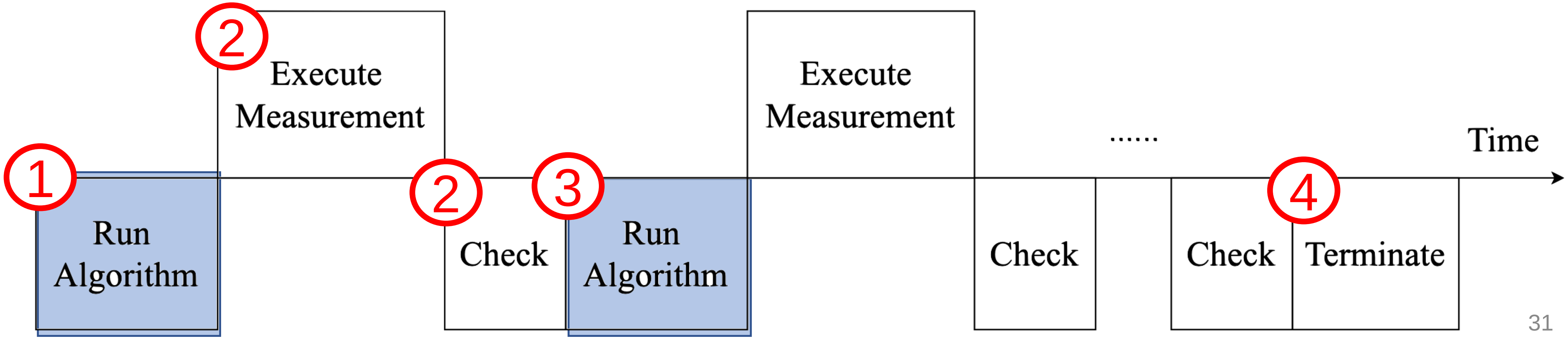


Measurement Selection Algorithm

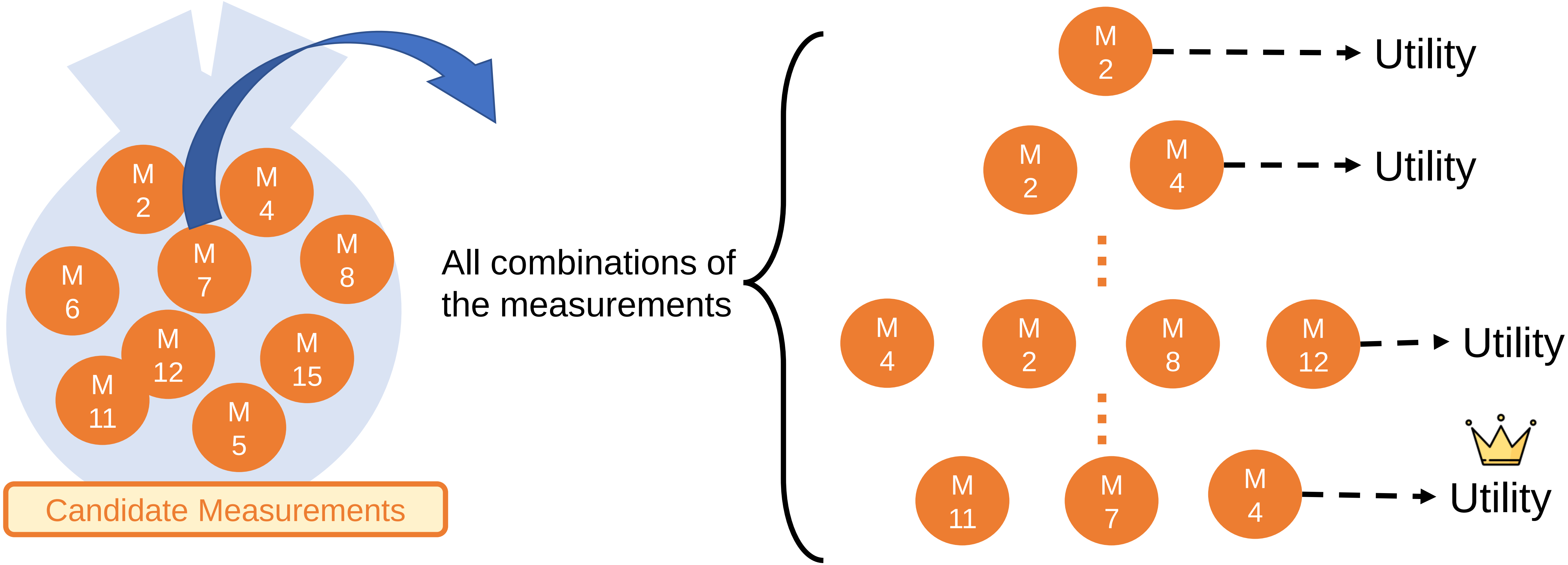
When to Run Measurement Selection Algorithm



1. Run the algo at the beginning to have an initial measurement sequence
2. After detecting one measurement, check if we meet our target
3. If not → Run algo again to create a new best measurement sequence
4. Terminate one window detection after running out of time or meet the target accuracy



Benchmark: Exhaustive Search *OPT*

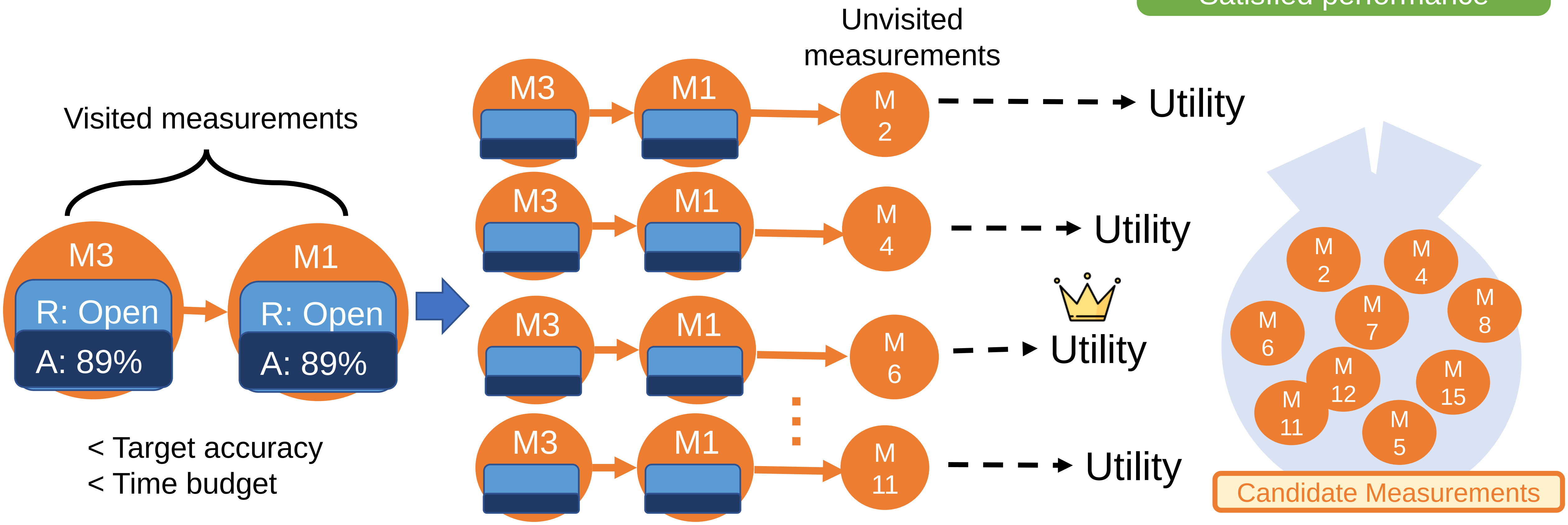


1. Create every combinations of the measurements
2. Calculate the utility value for each measurement
3. Choose the one with the largest utility value as the final measurement sequence

Long running time

Heuristic Algorithm

Real-time response
Satisfied performance



① Consider every unvisited measurement and calculate the utility after add them

② Select the one that can increase the utility the most

Implementation

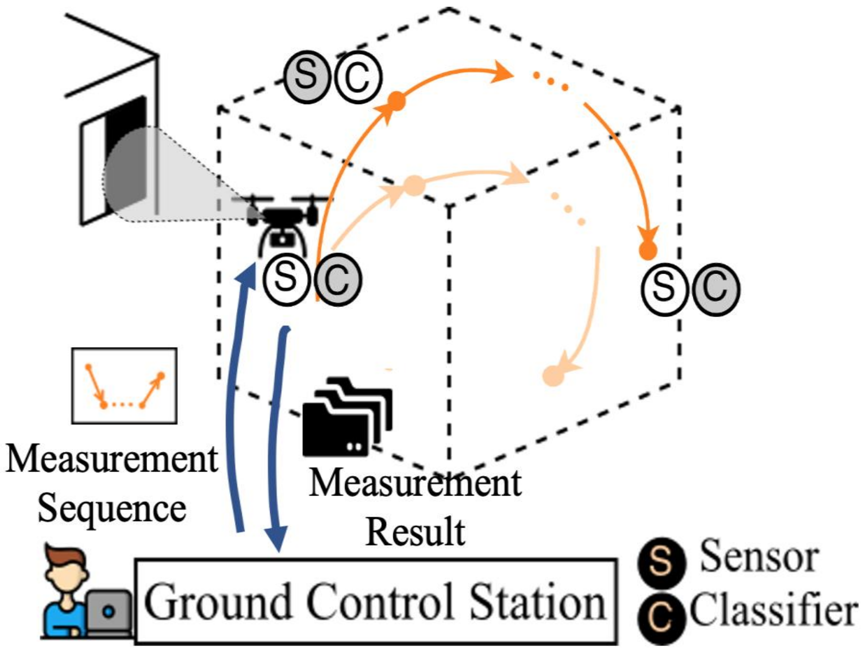
Photo-Realistic Simulator

- Build a city model in Unreal Engine
- Install sliding windows that can open and close to the buildings
- Simulate drones with multi-modal sensors (RGB and ultrasound)by AirSim [MobiCom’21]
- Select a 10-th floor building for evaluation
  - Collect data of 20 windows as site survey
- Site survey:
  - Train the classifier model
  - Generate acc. of candidate measurements

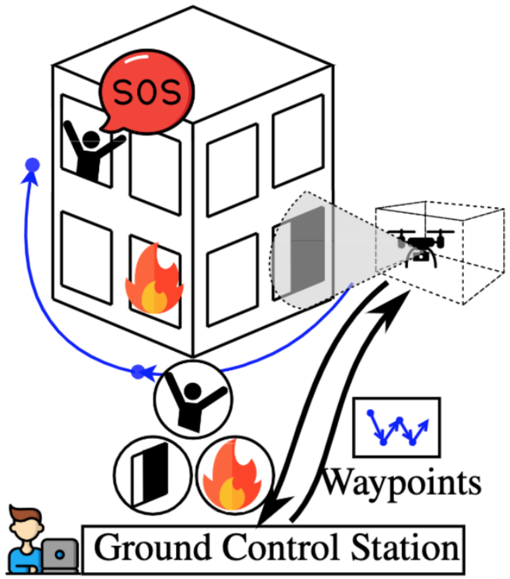
[MobiCom’21]: S. Tang, C. Hsu, Z. Tian, and X. Su. 2021. An Aerodynamic, Computer Vision, and Network Simulator for Networked Drone Applications. In Proc. of ACM Annual Intl. Conf. on MobiCom.

Event-Driven Simulator

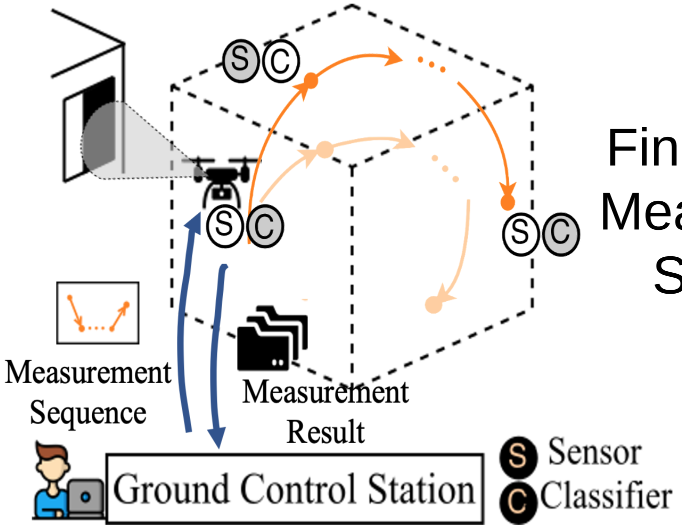
- Made by C++
- Simulate the whole process of our system
- Connect to the photo-realistic simulator for real time detection



Compared Algorithms



Coarse-Grained Waypoint Scheduling



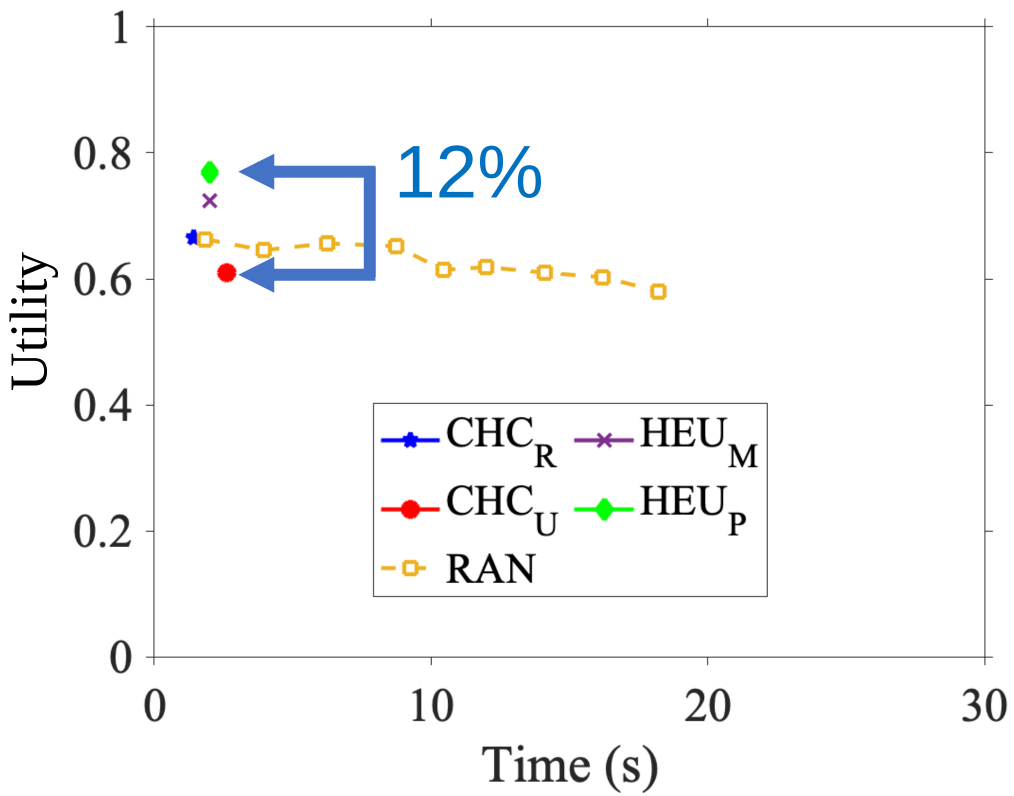
+ Fine-Grained Measurement Selection

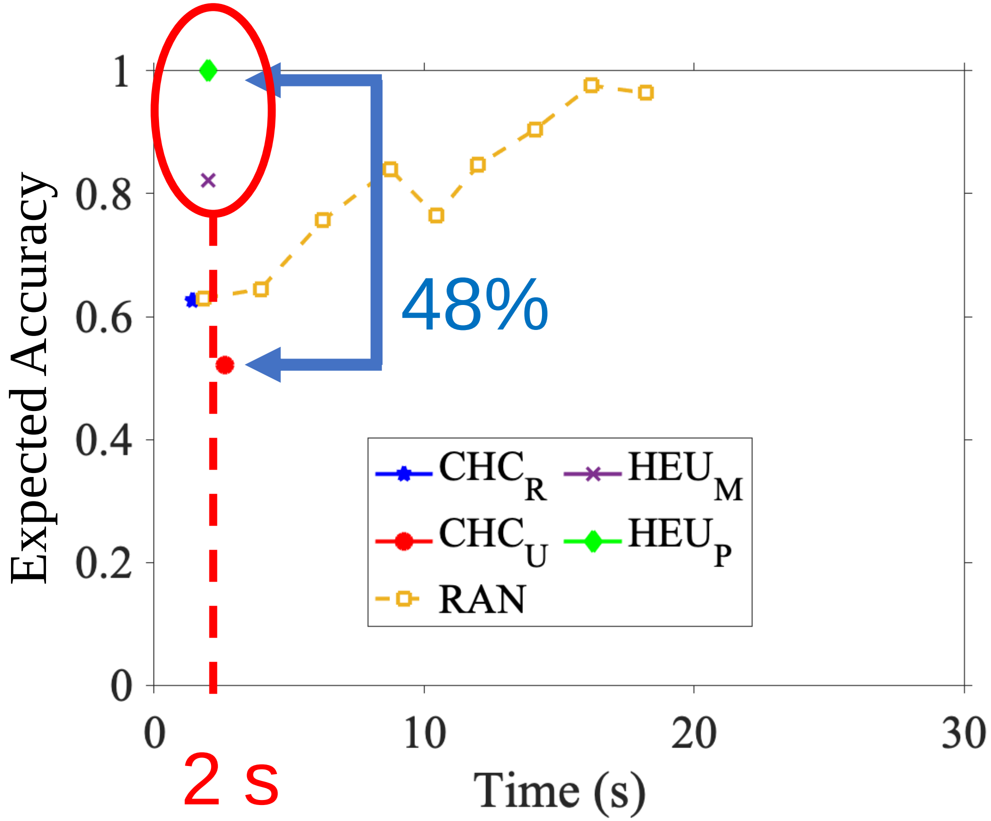
Algo.	Location	Sensor	Classifier
1 CHC_R	Middle of the bounding box	RGB	Histogram
2 CHC_U	Middle of the bounding box	Ultrasound	Histogram
3 RAN	Randomly select from the candidate measurements		
4 OPT	Use exhaustive search to find the best measurement seq.		
5 HEU_M	Use heuristic algo. with majority vote policy to generate a measurement seq.		
6 HEU_P	Use heuristic algo. with probability-based policy to generate a measurement seq.		

Evaluations

Results of One Sample Window

Target acc: 0.8
Time limit: 30 s





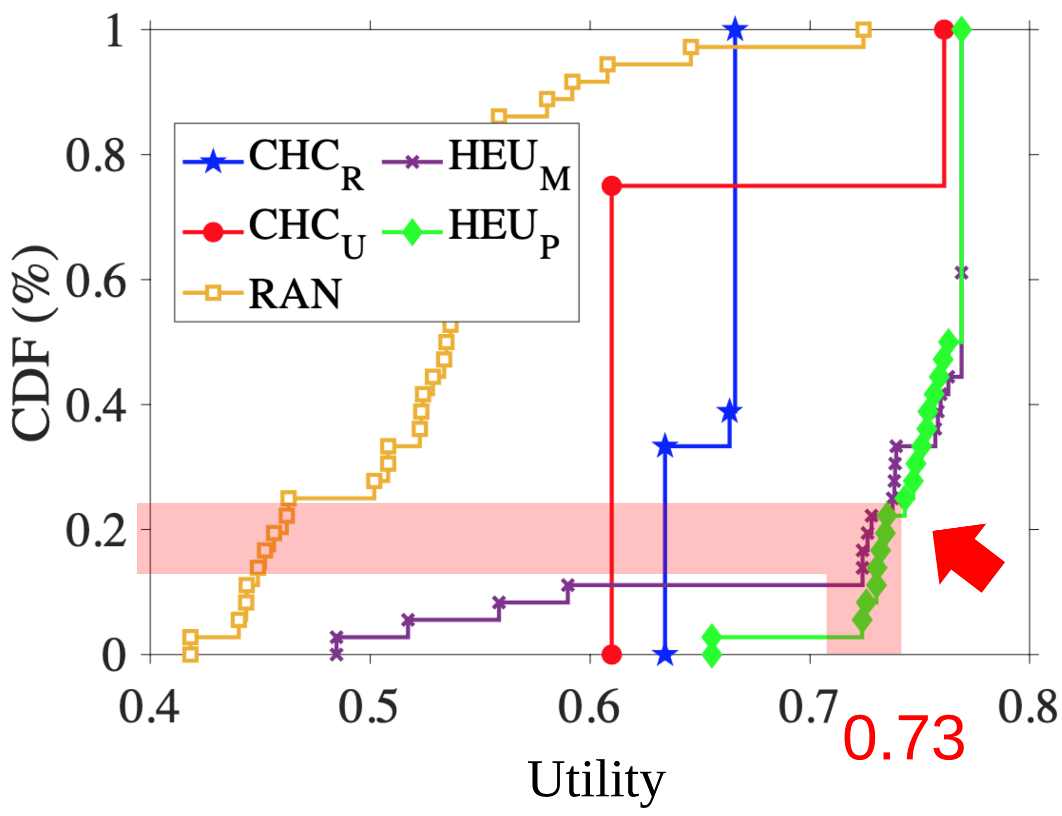
- HEU_M/HEU_P achieve the target accuracy (> 0.8) and the time limit (< 30 s)

→ HEU_M/HEU_P work as we expect

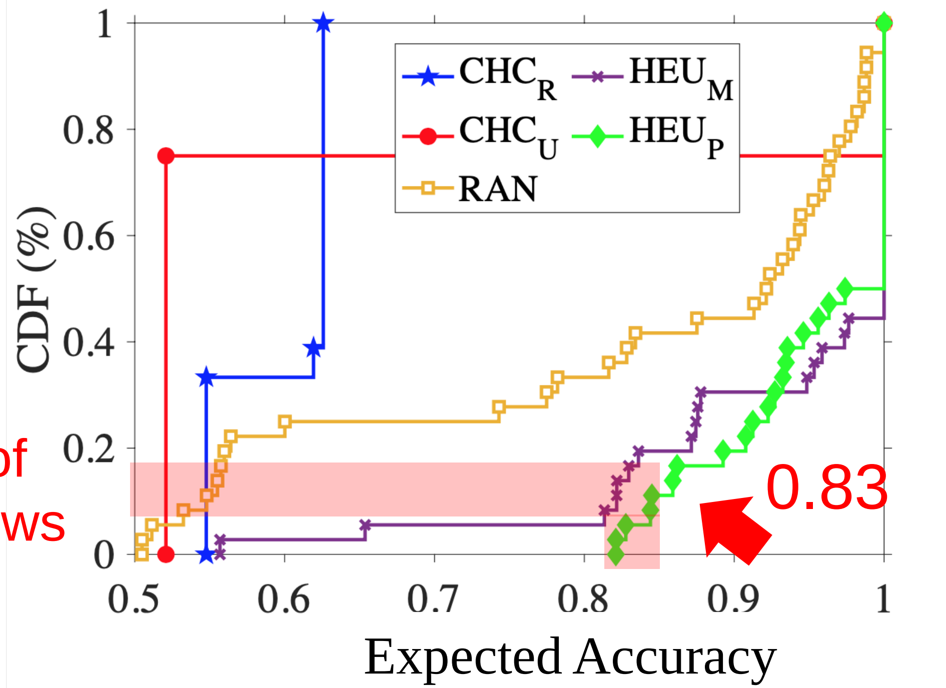
- HEU_M/HEU_P outperform baselines in utility and expected accuracy

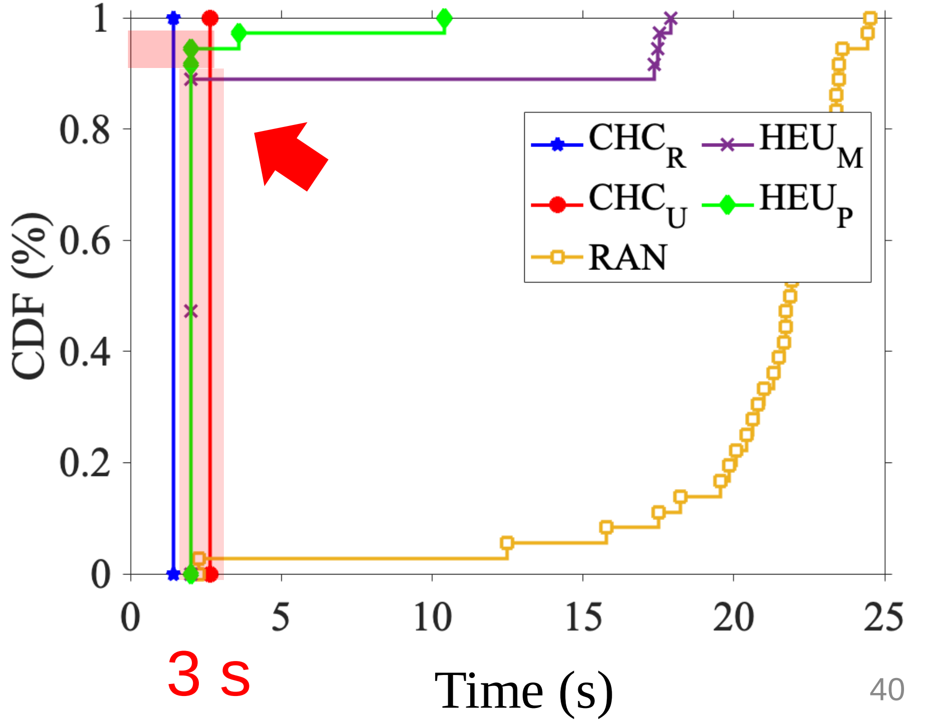
Accumulated Results from All Windows

80% of
windows



80% of
windows



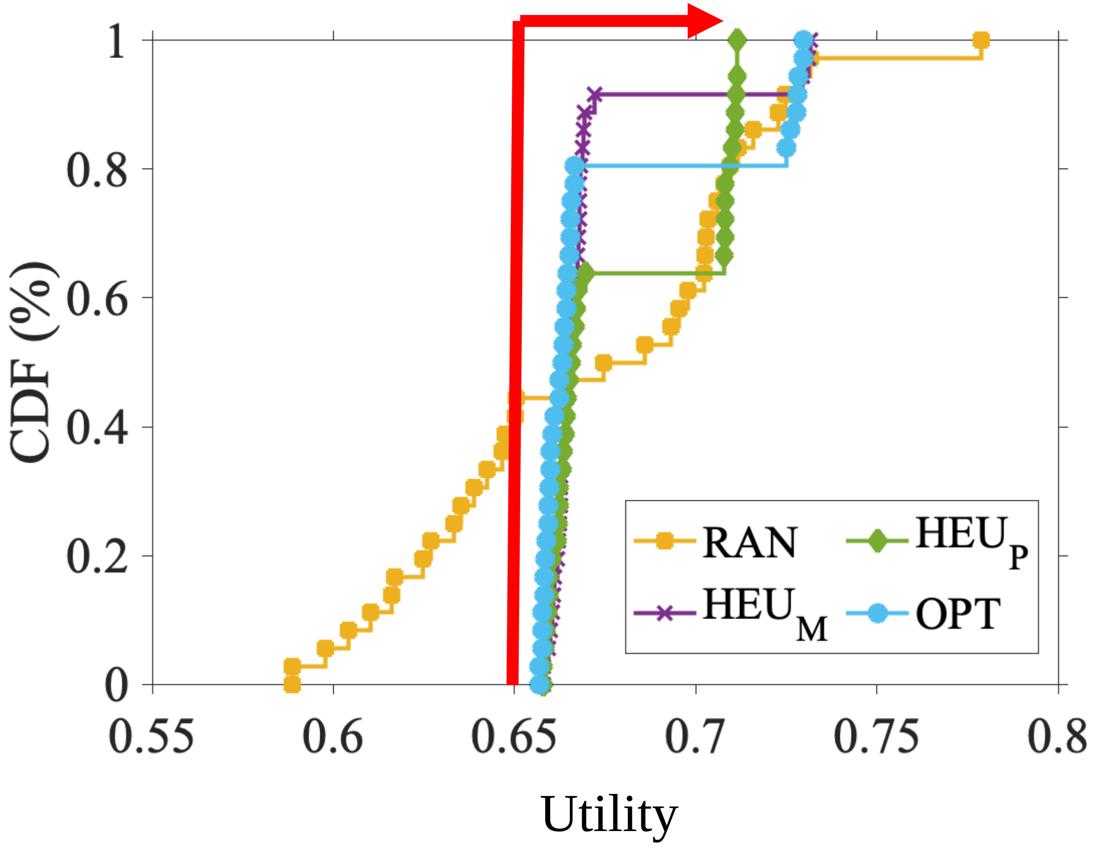


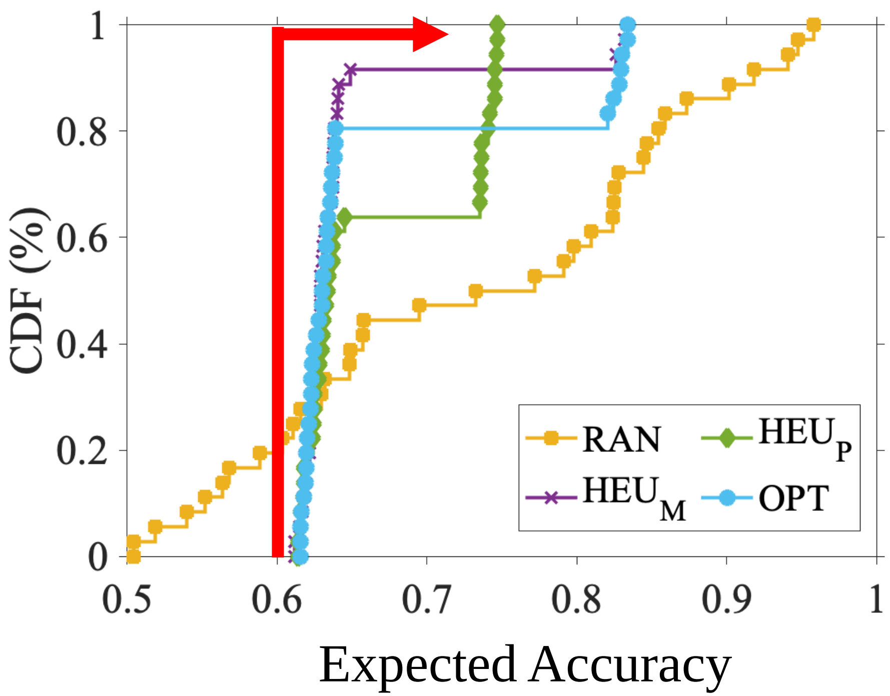
- HEU$_M$/HEU$_P$ achieve utility larger than 0.73 and expected accuracy larger than 0.83 in 80% of windows

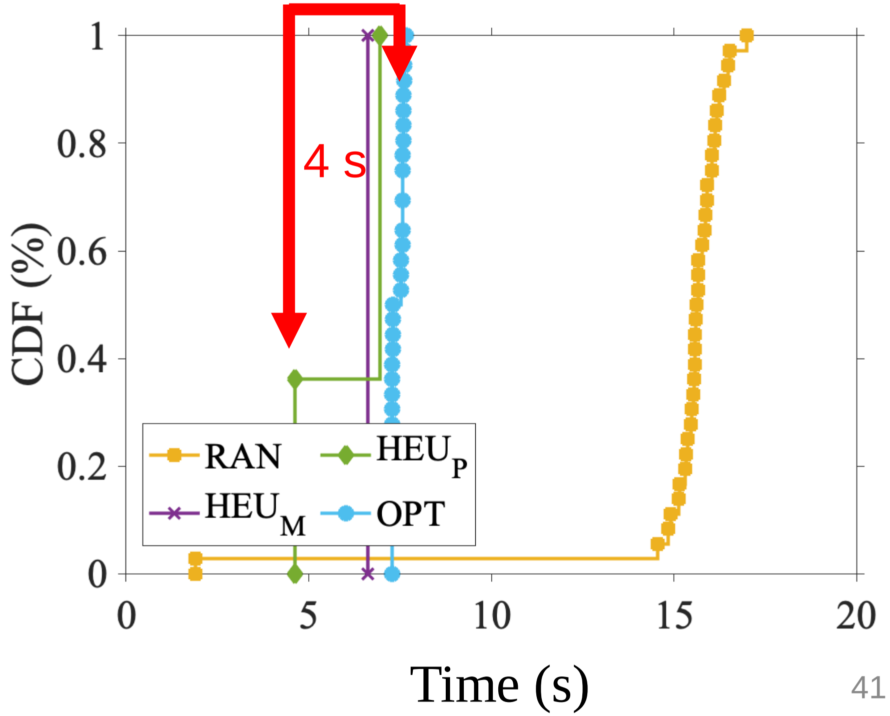
→ HEU$_M$/HEU$_P$ outperform baselines in most windows

- HEU$_M$/HEU$_P$ always cost less than 3 s → Real time response

Compare Results of All Windows to OPT







- Select the measurement seq. from smaller problem size
- HEU$_M$/HEU$_P$ /OPT achieve utility larger than 0.65 and expected accuracy larger than 0.6 for all windows

→ The results of HEU$_M$/HEU$_P$ are close to OPT for all windows

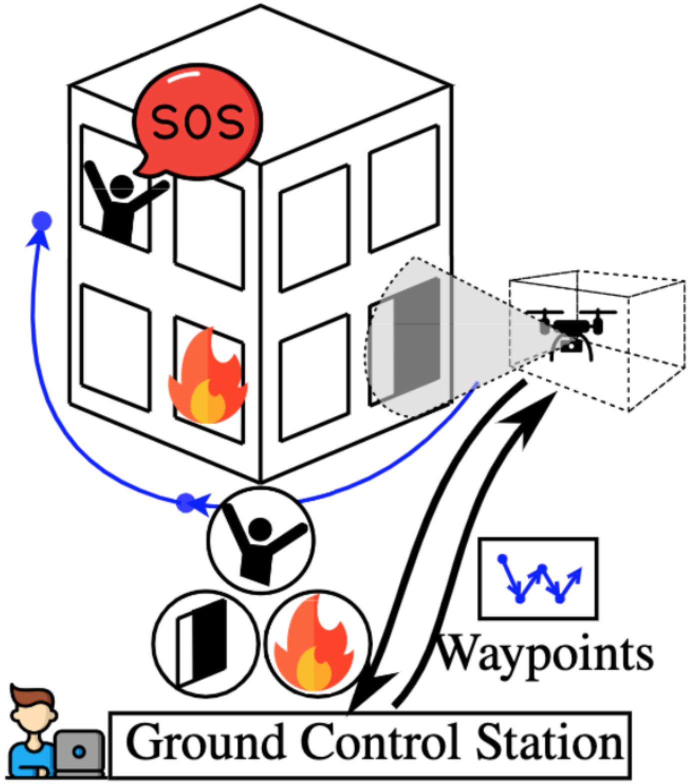
- HEU$_M$/HEU$_P$ cost 4 s less than OPT at most

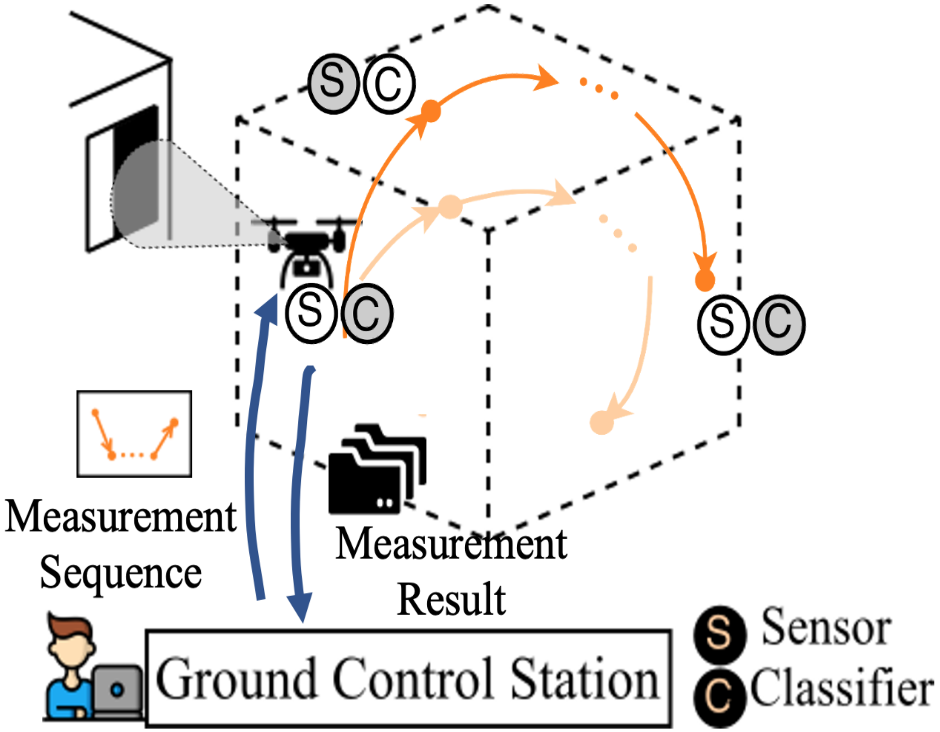
→ HEU$_M$/HEU$_P$ meets the real-time response requirements

Demo of a Complete Firefighting System in the Real World (Ongoing)

System Architecture

- Combine coarse-grained waypoint scheduling with the fine-grained measurement selection
  → Improve situational awareness



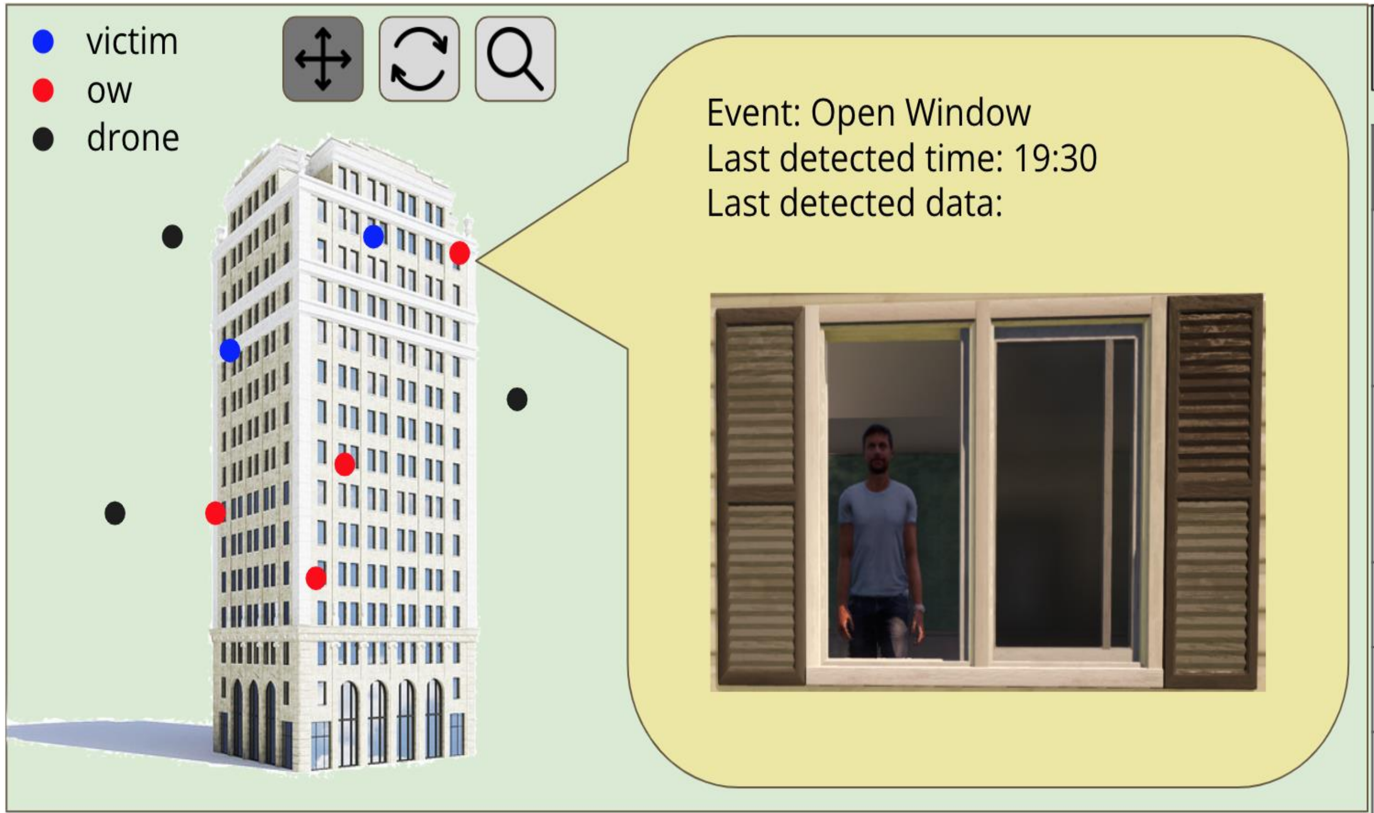


Demo

- Build a high-rise building model using a bookshelf
- Use Tello drone to fly through the waypoints we set
- Use our laptop to simulate the ground control station to analyze the sensor data and generate measurement sequence

Dashboard to Show the Collected Data

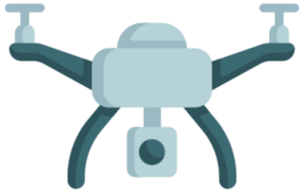
- Design the user interface for the firefighters
- A 3D building demonstrate where the victims and the open windows are
- A table that could add task for drones to monitor

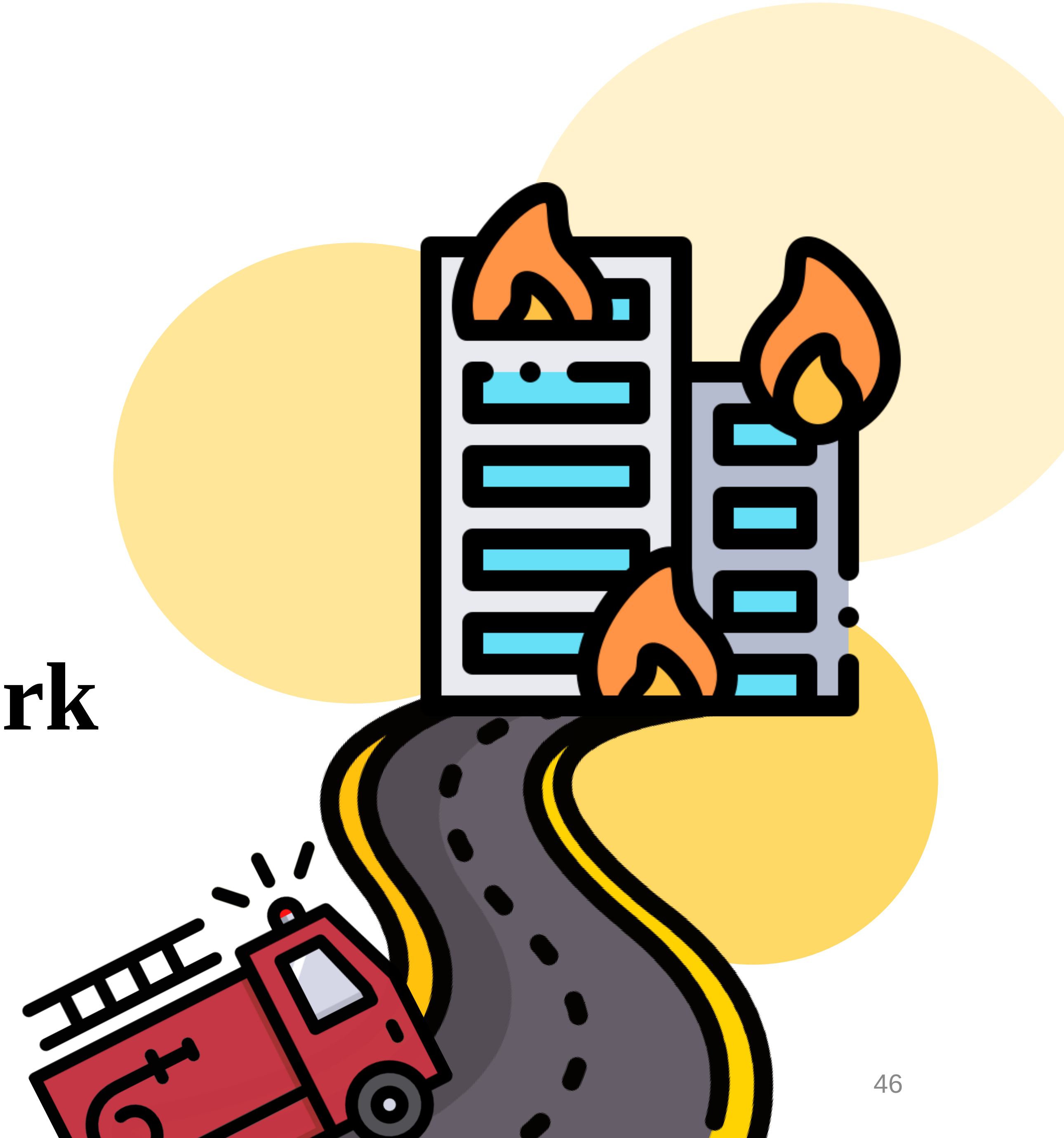


Time: 19:20 | Event: | Location: | Enter

Time	Event	Area	Sig.	Fre.
18:50	Fire	Win. on F6 Win. on F7 Win. on F8	3	0
18:59	Human	Win. in R605 Win. in R601 Win. in R602	3	0
19:10	Human	Win. in R604	3	2
19:18	Smoke	Win. on F10	2	0.2
19:19	Fire	Win. on R501	3	2

Conclusion & Future Work





Conclusion

1 Create the first multi-modal window dataset

2 Develop diverse window openness classifiers

3 Solve the fine-grained measurement selection problem

4 Create photo-realistic/event-driven simulators to evaluate our algorithms

- Our measurement selection algo:
  5 Reach the target acc. and the time limit in 86% of windows
  6 Achieve accuracy larger than 88% after comparing to the groundtruth

Algorithm	$\mathrm{CHC_R}$	$\mathrm{CHC_U}$	RAN	$\mathrm{HEU_M}$	$\mathrm{HEU_P}$
O_M (%)	50	75	72.22	88.89	88.89
O_P (%)	50	75	66.67	94	100
F_M (%)	0	25	19.44	88.89	86.11
F_P (%)	0	25	66.67	94.44	100
Mean (std.) L	1 (0)	1 (0)	9.36 (1.79)	1.22 (0.64)	1.67 (0.85)

Future Work

We will consider:

1 Various types/amounts of site survey

2 Complicated network conditions

3 Realistic situations

Thank you for listening !

TZU-YI FAN (Email: joyfan2@gmail.com)

Thanks for the help of Prof. Hsu, Prof. Venkatasubramanian,
Fangqi Liu, Jia-Wei Fang, Tun-Chi Tsai, Yan-Mei Tang and Yi-Ting Wang, and all lab mates.

- Publications:
  - T. Fan, T. Tsai, C. Hsu and F. Liu, and N. Venkatasubramanian. 2021. WinSet: The First Multi-Modal Window Dataset for Heterogeneous Window States. In Proc. of The 8th ACM International Conference on Systems for Energy- Efficient Buildings, Cities, and Transportation (BuildSys’21), November 17–18, 2021, Coimbra, Portugal.
  - T. Fan, F. Liu, J. Fang, N. Venkatasubramanian, and C. Hsu. 2022. Enhancing Situational Awareness with Adaptive Firefighting Drones: Leveraging Diverse Media Types and Classifiers. In Proc. of the 13th ACM Multimedia Systems Conference (MMSys ’22), June 14–17, 2022, Athlone, Ireland.
  - F. Liu, T. Fan, C. Grant, C. Hsu, and N. Venkatasubramanian. 2021. DragonFly: Drone-Assisted High-Rise Monitoring for Fire Safety. In Proc. of IEEE SRDS. Virtual, 331–342.
  - Target to create a journal paper for our SRDS and MMSys paper

Accuracy Fusion for Binary Classifiers after Measuring

- E.g., open window classifiers, classes: open (1) vs close (0)
- All measurement results from L: R(L) = {L_0, L_1}
- Majority vote: Select the class with the most vote
  e.g., 1 1 1 0 0 → 1

$$\hat{R}(\mathbf{R}(\mathbf{L})) = \arg\max_{i=\{0,1\}} |\mathbf{L}_i|$$

- Probability-based: Select the class with the largest correct probability under the prediction from the measurement sequence
  e.g., 1, 0.7 | 0, 0.8 | 0, 0.6

$$\hat{R}(\mathbf{R}(\mathbf{L})) = \arg\max_{i\in\{0,1\}} P(i|\mathbf{R}(\mathbf{L}))$$

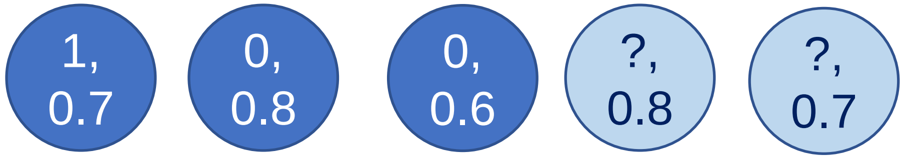
$$P(1|\mathbf{R}(\mathbf{L})) = \frac{\prod_{m_i\in\mathbf{L}_1} a_i \prod_{m_j\in\mathbf{L}_0}(1-a_j)}{\prod_{m_i\in\mathbf{L}_1} a_i \prod_{m_j\in\mathbf{L}_0}(1-a_j) + \prod_{m_i\in\mathbf{L}_1}(1-a_i)\prod_{m_j\in\mathbf{L}_0} a_j}$$

Accuracy Fusion for Binary Classifiers before Measuring

- Majority vote:
  Compute the probability that the results from half of the selected measurements are correct

$$A_M(\mathbf{L}) = \sum_{k=\lceil L/2 \rceil}^{L} \sum_{\substack{\mathbf{L}_1 \subseteq \mathbf{L}, |\mathbf{L}_1| = k, \\ \mathbf{L}_0 = \mathbf{L} \setminus \mathbf{L}_1}} \Big(\prod_{m_i \in \mathbf{L}_1} a_i \prod_{m_j \in \mathbf{L}_0} (1 - a_j) \Big)$$

- Probability-based:
  Calculate the probability after the new measurements guessing 0 and 1
  Select the result with larger probability



$$A_P(\mathbf{L}) = \sum_{\substack{\hat{\mathbf{L}}_1 \subseteq \hat{\mathbf{L}}, \\ \hat{\mathbf{L}}_0 = \hat{\mathbf{L}} \setminus \hat{\mathbf{L}}_1}} \max \Big(P(1|\mathbf{R}(\mathbf{L}')) \prod_{m_i \in \hat{\mathbf{L}}_1} a_i \prod_{m_j \in \hat{\mathbf{L}}_0} (1 - a_j), \\ (1 - P(1|\mathbf{R}(\mathbf{L}'))) \prod_{m_i \in \hat{\mathbf{L}}_1} (1 - a_i) \prod_{m_j \in \hat{\mathbf{L}}_0} a_j \Big)$$

Time Calculation for the Measurement Sequence

Algo running time

$$T(\mathbf{L}) = \sum_{i=1}^{L} (t^s_{l(i)} + t^c_{l(i)} + \frac{|p_{l(i)} - p_{l(i-1)}|}{V} + t^a_{l(i)} + t^t_{l(i)}$$

Sensing Classification Moving Data transferring

Formulation

$$\max_{\mathbf{L}} U(A(\mathbf{L}), T(\mathbf{L})) = \sqrt{\left(1 - e^{-\alpha A(\mathbf{L})}\right) \times e^{-\beta \frac{T(\mathbf{L})}{\hat{T}}}}$$

$$\text{s.t.} \quad A(\mathbf{L}) \geq \hat{A}; \quad T(\mathbf{L}) \leq \hat{T}$$

Accuracy

Consuming Time

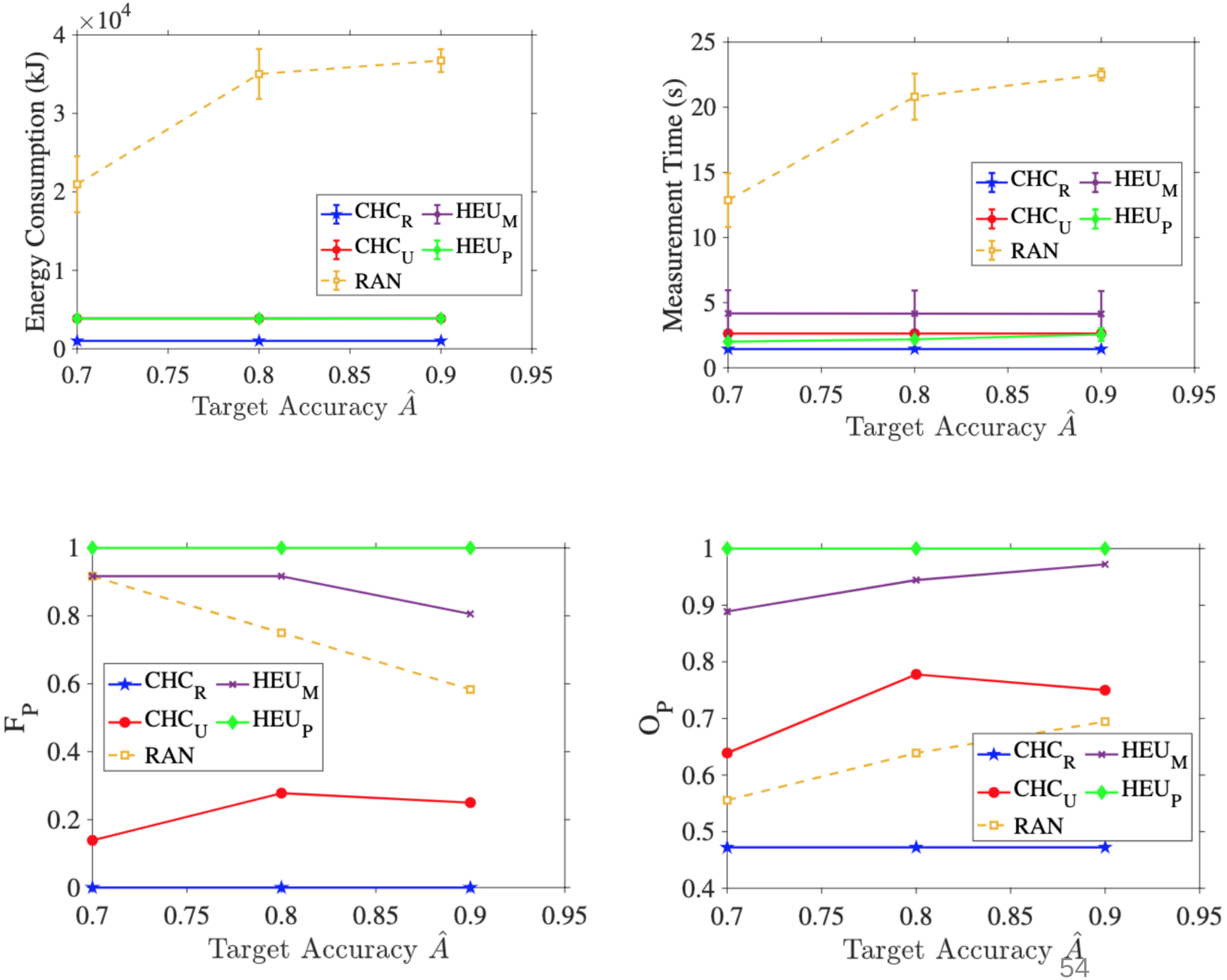
Evaluations: Setup

- Parameters:
  - Target accuracy = {0.7, <u>0.8</u>, 0.9}
  - Time limit for each window = {5, 10, <u>30</u>, 60} seconds
  - Candidate sampling policies = {$\underline{E_{10}}$, E_8, E_8, E_8}
  - Default settings: under-lined values

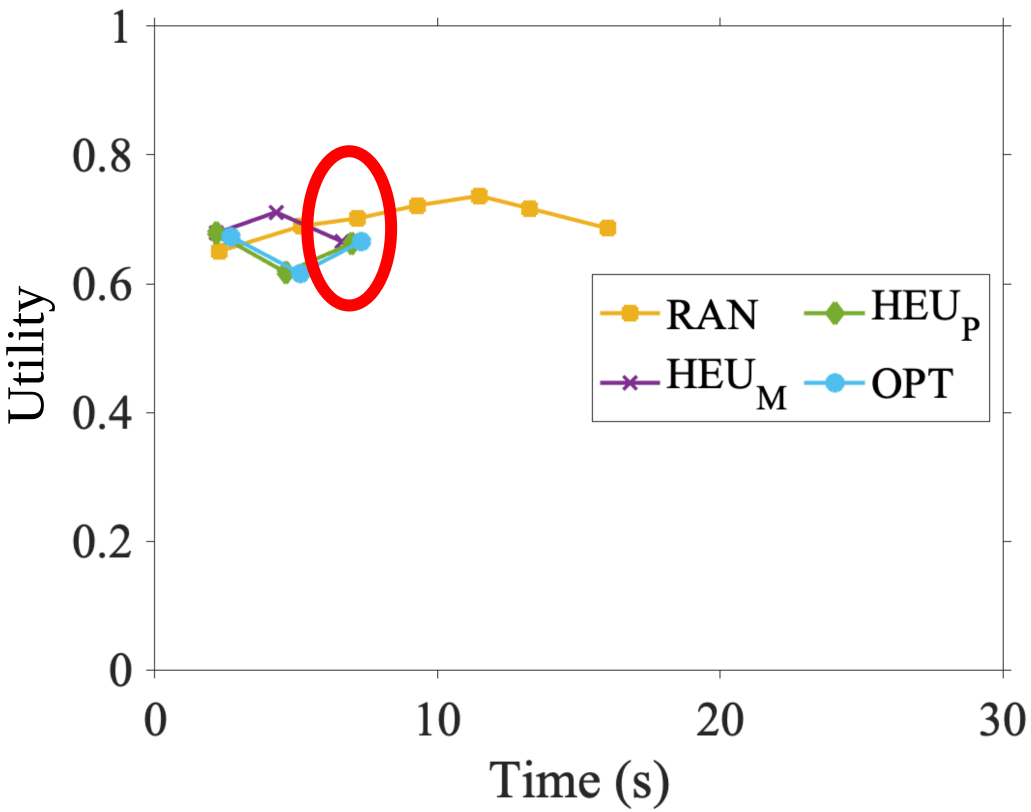
- Metrics:
  - Overall accuracy $O_M(L)/ O_P(L)$
  - Expected accuracy $A_M(L)/ A_P(L)$
  - Utility function U_M / U_P
  - Total measurement time $T(L)$
  - Number of measurement |L|
  - Feasible ratio of measurements F_M / F_P: the fraction of L satisfying the target accuracy and the time limit
  - Energy consumption

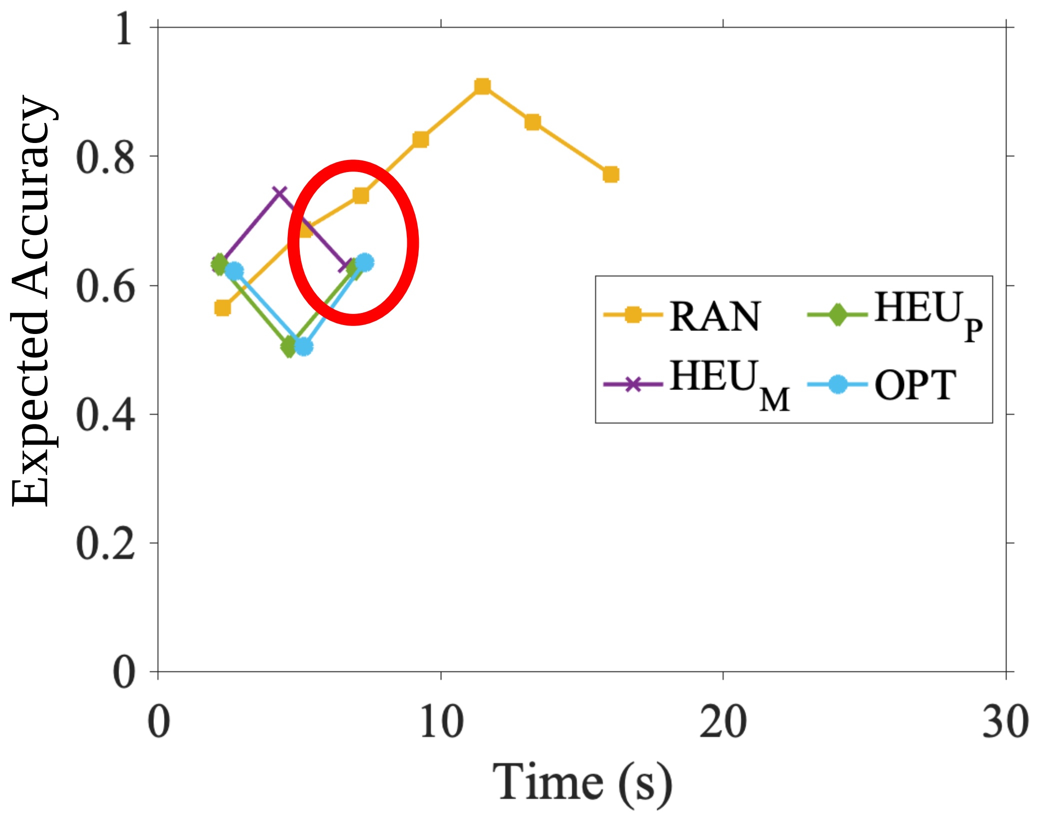
Performance at Different Target Accuracy

- HEU$_M$/HEU$_P$ can keep the good performance under different target accuracy



Compare Result of One Sample Window to OPT

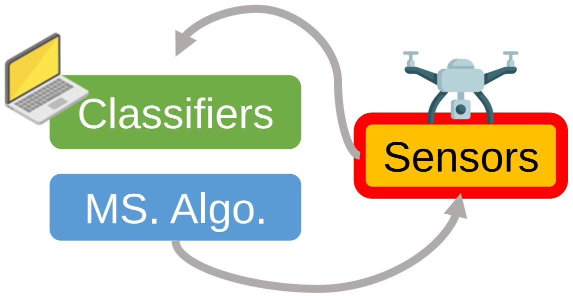




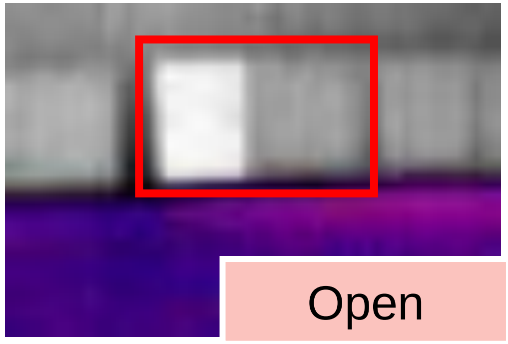
- Select the measurement seq. from smaller problem size
- HEU$_M$/HEU$_P$ /OPT achieve 0.66 utility
- HEU$_M$/HEU$_P$ /OPT achieve 0.65 expected accuracy

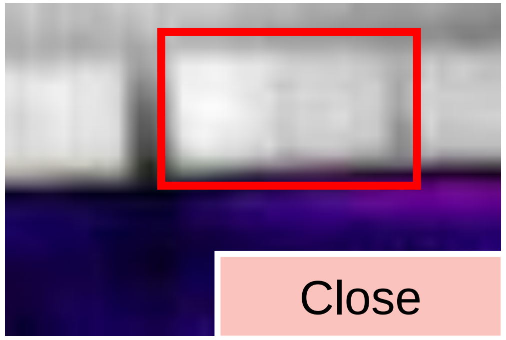
→ The results of HEU$_M$/HEU$_P$ are close to OPT for one sample window

Open Window Classifiers in WinSet



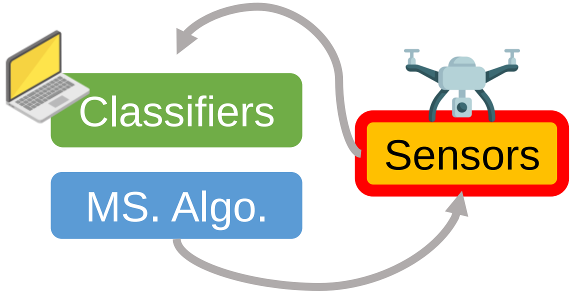
- Thermal Window Classification (TWC)
  - Open window: diverse temperature → Different color
  - Close window: constant temperature → Same color



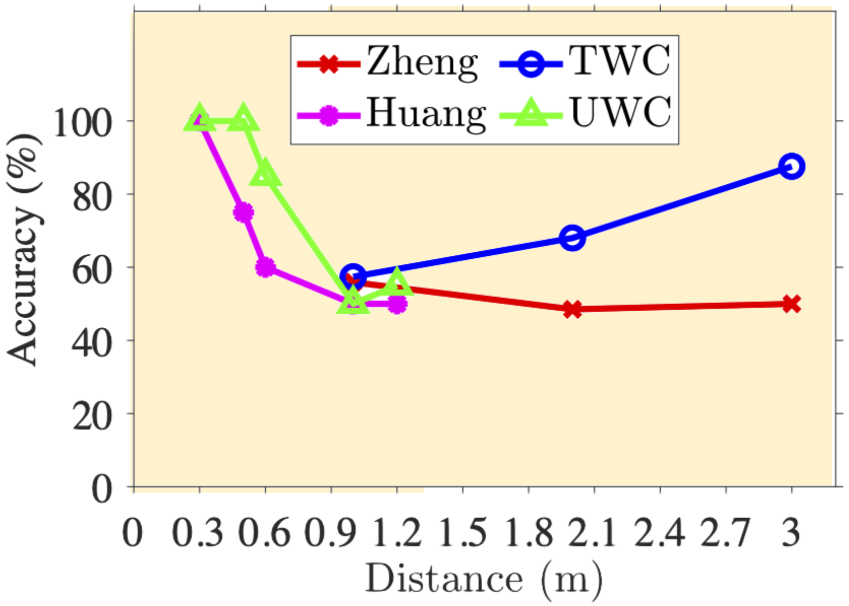


- Ultrasound Window Classification (UWC)
  - a Value returned by the ultrasound sensor - b Actual distance to a window (derived from the GPS coordinates or building blue-print)
- Baselines:
  - Zheng [Energy Build.'19]: same method as TWC but with RGB images
  - Huang [SPIE Target and Background Signatures'18]: same method as UWC but change b to the value from the depth sensor

Evaluations of the Open Window Classifiers in WinSet



- Ultrasound-based classifiers (Huang and UWC):
  - Accuracy & F1-Score: UWC > Huang (> 0.3 m and < 1.2 m)
- RGB/thermal-based classifiers (Zheng and TWC):
  - Accuracy: TWC > Zheng (> 1 m and < 3 m)
  - If the distance is getting larger, TWC works much better than Zheng



UWC ← 60 cm → TWC