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

Tzu-Yi Fan[1], Fangqi Liu[2], Jia-Wei Fang[1], Nalini Venkatasubramanian[2], and Cheng-Hsin Hsu[1]
[1]Department of Computer Science, National Tsing Hua University, Hsin-Chu, Taiwan
[2]Department of Computer Science, University of California, Irvine, USA

ABSTRACT

High-rise fires are among the largest threats to safety in modern cities, and autonomous drones with multi-modal sensors can be employed to enhance situational awareness in such unfortunate disasters. In this paper, we study the fine-grained measurement selection problem for drones being dispatched to perform situation monitoring tasks in high-rise fires. Our problem considers multiple sensor/media types, classifier designs, and measurement locations, which were overlooked in prior waypoint scheduling studies. For concrete discussion, we adopt window openness as the target situation, while other situations can be readily supported by our solution as well. More specifically, we: (i) develop diverse window openness classifiers, (ii) mathematically formulate the fine-grained measurement selection problem and solve it using two algorithms, and (iii) create a photo-realistic simulator and an event-driven simulator to evaluate our algorithms. The evaluation results demonstrate that our proposed algorithms achieve higher classification accuracy (up to 50% improvement), deliver more feasible solutions (up to 100% improvement), and reduce energy consumption (up to 6.78 times reduction), compared to the current practices.

CCS CONCEPTS

• **Information systems → Multimedia information systems**.

KEYWORDS

Autonomous drones, multi-modal sensors, optimization

ACM Reference Format:
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 *13th ACM Multimedia Systems Conference (MMSys '22), June 14–17, 2022, Athlone, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3524273.3532910

1 INTRODUCTION

United Nations reported that 55% of people live in urban areas as of in 2018, which is projected to reach 68% by 2050 [29]. Such a high rate of urbanization makes housing the urban populations extremely challenging. Taller high-rise buildings have been constantly built to vertically accommodate more people for living, working, socializing, and entertaining. These high-rise buildings, however, are more *vulnerable* to hazards, such as fires, earthquakes, terrorism, and disease outbreaks. As the growth of multimedia technologies and hardware devices, more and more innovative multimedia systems are developed to improve the safety in these urban living environments.

In this paper, we consider well-recognized urban safety concerns: *high-rise fires*, which often result in severe casualties. For example, Grenfell Tower Fire in London led to 72 fatalities [17] in 2017. The challenges of firefighting in high-rise buildings are quite different from firefighting in traditional buildings for many reasons [3]. For instance, if fires break out at floors that are not reachable by fire ladders, firefighters must enter the high-rise buildings with oxygen masks/tanks, which put them in great danger. Such danger is mainly caused by the *situational unawareness* of firefighters when they are in high-rise buildings. The most critical situations in high-rise fires include: (i) the locations and severity levels of fires, (ii) regions affected by smoke, (iii) locations and vital signs of inhabitants, and (iv) window openness. These situations are not independent, e.g., sudden increases of fresh air due to a window loss could lead to rapid spreading of fires, known as wind-driven high-rise fires, which in turn result in unsurvivable environments [9].

One way to improve *situational awareness* in high-rise fires is to deploy sensors and surveillance systems throughout high-rise buildings. Doing so, however, is time-consuming, expensive, and error-prone, while sensors (such as magnetic switches) often work for homogeneous situations (such as window openness). Adding to that, sensors (like smoke detectors) are typically connected through proprietary networks, which may not interoperate with the communication networks used by firefighters. Therefore, it is very likely that firefighters are provided partial, if any, sensor readings, and have to search through fire scenes in a brute force manner, putting them in danger of fatal casualties.

In contrast, we aim to enhance the situational awareness in high-rise fires by leveraging fine-grained measurement selection from a suite of *heterogeneous classifiers* and *multi-modal sensors* mounted on *drones*, which fly around the exterior of buildings. As different sensors, classifiers, and measurement locations affect the detection results [12], it is challenging to find the best combinations of them. Fig. 1 reveals the considered usage scenario, in which a ground control station instructs one or multiple drones to monitor the situations in high-rise fires. Fig. 1(a) shows the coarse-grained waypoint scheduling problem addressed in prior studies [16, 25, 30], where the ground control station schedules waypoints for drones to perform different location-dependent monitoring tasks. To the

T. Fan was supported by a Novatak Fellowship. The research is also supported by the MOST of Taiwan (#110-2221-E-007-102), DARPA (#FA8750-16-2-0021), and UC (#LFR-20-653572).

MMSys '22, June 14–17, 2022, Athlone, Ireland

ACM ISBN 978-1-4503-9283-9/22/06…$15.00
https://doi.org/10.1145/3524273.3532910

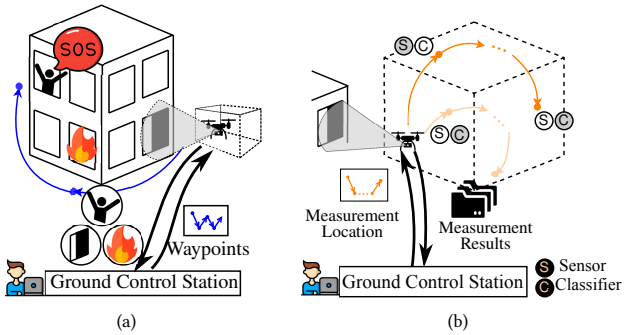


Figure 1: Adaptive firefighting drones: (a) coarse-grained waypoint scheduling and (b) fine-grained measurement selection.

best of our knowledge, these existing studies largely ignored the detailed properties of heterogeneous sensors, media types, and classifiers. Instead, they assumed that each monitoring task can be completed within a given time duration while achieving a known accuracy level. These assumptions, unfortunately, deviate from the reality. For example, each reading of an ultrasound sensor is a single distance value, and a window openness classifier may require multiple such readings at slightly different locations in front of a given window. Adding to that, high-rise fire scenes are diverse and dynamic, rendering the time and accuracy of different combinations of locations, sensors, and classifiers hard to predict.

To fill in the gap, we zoom into the fine-grained planning within a bounding box of each monitoring task, as shown in Fig. 1(b). We consider multiple sensor/media types, classifier designs, and measurement locations to accomplish every monitoring task. More specifically, our job is to select the best sequence of *measurements*, which are the combinations of locations, sensors, and classifiers. Our problem has a general setup, e.g., some sensors/classifiers are *one-shot*, where a single measurement gives rich enough data for classifications; while others are *accumulated*, where all prior measurements are jointly analyzed for classifications. RGB cameras and ultrasound sensors are representative sensors that are one-shot and accumulated, respectively. Upon analyzed by machine-learning classifiers, semantically-meaningful measurement results are sent back to the ground control station, and presented to firefighting officials. While our system and algorithms can be applied to enhance the awareness of various high-rise fire related situations, we adopt window openness as a representative situation for concrete discussion.

Handling multi-modal sensor data and computing the best measurement sequence for each monitoring task are not easy tasks for several reasons. First, we need to generate multi-modal sensor dataset by ourselves as most of existing datasets only come with RGB images. Second, different sensors/classifiers achieve diverse accuracy levels while consuming different resource amounts. Hence, finding a good trade-off between accuracy and timeliness is critical to resource-constrained drones. Third, high-rise fire scenes are highly dynamic and thus the measurement sequence needs to be adaptive. For example, fires may cause thick smoke, which renders RGB camera less accurate and calls for other sensors/classifiers. Last, fusing results from different sensor/classifiers may lead to even better trade-off mentioned above. We address these challenges by: (i) collecting a dataset in a virtual environment and developing several classifiers to detect window openness (Sec. 4), (ii) mathematically formulating the measurement selection problem (Sec. 5), (iii) proposing efficient algorithms to compute the measurement sequence (Sec. 5), and (iv) evaluating the proposed algorithms in a photo-realistic drone simulator (Sec. 6).

2 RELATED WORK

Heterogeneous sensors on drones. Using heterogeneous sensors to classify situations is challenging, and the same challenge was faced by researchers building autonomous cars [8]. The research community has also investigated the possibility of mounting multi-modal sensors, such as thermal cameras [15], RGB cameras [4], and a combination of smoke, gas, and microwave sensors [14], on drones for fire detection. *We also propose to mount multi-modal sensors on drones.*

Firefighting drones. To ensure the safety of firefighters, drones have been used for firefighting. For example, Ando et al. [7] developed a fire extinguishing drone with a 2-meter long robot arm for stable water discharging. Ogawa et al. [19] proposed a fire extinguishing mechanism by injecting gases (CO_2, N_2, etc.). Alshbatat et al. [1] proposed to have drones throw fire extinguishing balls. *These projects are orthogonal to our work, as we aim to build classifiers to improve the situational awareness rather than directly putting out fires.*

Coarse-grained waypoint scheduling for drones. There are research papers on scheduling waypoints for multiple drones to monitor a large area for an extensive time duration. For example, Wallar et al. [30] designed a planner for drones to continuously monitor risky regions on a 2D map. To keep classification results valid in dynamic environments, Smith et al. [25] proposed a dynamic approach to patrol areas with drones, and Sea et al. [10] balanced the workloads of drones visiting multiple regions. Similarly, Liu et al. [16] proposed a drone-assisted high-rise fire monitoring system, which systematically computes waypoint schedules for individual drones to perform assigned tasks. *These waypoint scheduling studies overlooked the fine-grained decisions of measurement sequences in high-rise fire scenes, which are addressed in the current paper.*

3 DRONE-BASED HIGH-RISE FIREFIGHTING

As illustrated in Fig. 1, a drone-based high-rise firefighting system enhances the situational awareness by instructing drones to fly through waypoints for detecting situations. Our work focuses on the fine-grained planning at each waypoint. For completeness, we give a high-level component diagram in Fig. 2, and present an overview on both the coarse-grained waypoint scheduling and fine-grained measurement selection operations in the following.

Coarse-grained waypoint scheduling. Our system takes inputs form users, i.e., firefighting officials, for monitoring tasks based on fire reports, earlier classification results, and their domain knowledge. Sample tasks could be detecting the presence of fire, trapped humans, and opened windows. The *task generator* at the ground control station assigns tasks for drones by specifying the locations, visiting frequencies, and so on. Then the waypoint scheduler associates waypoints (3D coordinates) with individual drones, which then fulfill all tasks periodically. Here, each waypoint specifies a

single coordinate for performing a task, such as detecting window openness. The *precise*, or fine-grained, locations for activating sensors, such as RGB cameras and ultrasound sensors, are not determined as they may need to be dynamically adapted based on the earlier classification results.

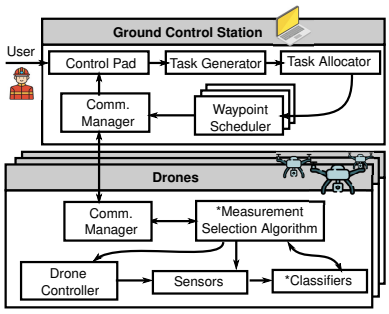


Figure 2: Drone-based high-rise firefighting system; asterisks indicate considered components.

(a)

(b)

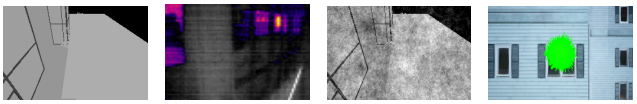
Figure 3: Our simulator: (a) high-rise buildings and (b) a flying drone.

Fine-grained measurement selection. The ground control station sends waypoint sequences to drones. Drones capture and analyze sensor data at each waypoint on the sequences. Whenever a drone reaches a waypoint, the *measurement selection algorithm* computes a measurement sequence within a given bounding box to optimize the classification results fused from previous measurements at that waypoint. The computation of the best measurement sequence considers accuracy level, resource consumption, available time, etc. A measurement leads to an one-time execution of a particular classifier on collected sensor data at a specific location. The three key components of our system are *sensors*, *classifiers*, and *measurement selection algorithms*, which are developed in the next two sections. Last, we mention that classifiers and measurement selection algorithms can be offloaded to the ground control station if: (i) drones have stringent resource constraints or (ii) algorithms have high complexity levels. The relevant data transfers are done via the *communication managers* at the ground control station and drones.

4 CLASSIFIERS FOR WINDOW OPENNESS

4.1 Realistic Dataset Generation

Window states play important roles to building fires, while openness is the most critical window-related situations in high-rise fires. Unfortunately, existing window datasets either focus on a single sensor modality, like thermal cameras [24], or provide no window state labels [13]. We recently created the first multi-modal window dataset [12] with multiple (partially labeled) window states, such as openness, human behind, and light on/off. The resulting datasets are, however, too small for developing window openness classifiers, especially the machine-learning ones. In this work, we decide to build a detailed simulator, implement virtual sensors, and collect a new window dataset with rich state labels. We have chosen the photo-realistic AirSim [23] as the starting point, which captures aerodynamic simulations, offers virtual RGB cameras (and other sensors), and has been integrated with a packet-level network simulator [27]. Leveraging on public 3D models [21], we have built several high-rise buildings in different styles and heights to collect a realistic dataset. Fig. 3 illustrates sample views of our high-rise community.



(a) (b) (c) (d)

Figure 4: Sample thermal images from: (a) an AirSim camera, (b) a FLIR camera, and (c) our enhanced AirSim camera; (d) detection area of our AirSim ultrasound sensor.

We consider the vision and non-vision virtual sensors in AirSim, including RGB/thermal/depth cameras, LiDAR, and distance sensors. Depth cameras and LiDAR, however, do not work well for classifying window states [12]. In AirSim, a RGB camera can be configured following real cameras' specifications, such as Field-of-View (FoV). If not otherwise specified, we apply the specifications of Intel RealSense D435. A thermal camera in AirSim only considers infrared emitted from the objects for the sake of simplicity. Fig. 4(a) shows such a sample thermal image in which each object has a uniform color (temperature). This is quite different from images from a real thermal camera, shown in Fig. 4(b). To address this issue, we have augmented the virtual thermal camera to introduce Gaussian and uniform noises to emulate the Johnson, flicker, and fixed-pattern noises [6]. Fig. 4(c) shows an improved thermal image captured by the augmented thermal camera, which is more realistic. The distance sensor in AirSim emits a ray to measure the distance of an object in front of it. This over-simplified design deviates from real distance sensors, such as an ultrasound sensor, which analyzes the echoed sound waves bounced back from a *cone area* rather than a *single point*. Therefore, we have built a virtual ultrasound sensor in AirSim by introducing multiple rays covering a circular detection area. To make the detection area more realistic, we add a Gaussian noise to its radius, as illustrated in Fig. 4(d). The detection area is a function of the distance and detection angle. If not otherwise specified, we set the detection angle to be 15°, following the specifications of HC-SR04. Our ultrasound sensor returns the smallest distance among all considered rays.

Among these virtual sensors, we adopt the RGB camera as a representative one-shot sensor and the ultrasound sensor as a representative accumulative sensor in the rest of this paper for brevity. Other virtual sensors can be classified into one-shot or accumulative ones, and readily incorporated into the proposed solution.

4.2 Classifier Designs

Our classifiers are composed of two components: the *data processor* and *analyzer*. The data processors gather and clean the sensor data to meet the input requirements of the analyzers. Then, the analyzers determine the window states. The data processors of single-shot classifiers take one measurement of rich-media data, such as images. They crop the subject window out from the collected data[1], and apply image filters (such as color transform) required by the analyzers. The data processors of accumulative classifiers, on the

[1]We assume the window location/dimension are given by floor-plans (dictated for public buildings [20]) or vision-based localization approaches [18].

other hand, combine all prior measurements at the current waypoint into rich-media data (images) using interpolation/extrapolation. By doing so, the same set of analyzers can also be leveraged.

Some classifiers may require sensor data from a *site survey*, which could be done at different granularities. For example, ultrasound sensors only collect distances, and thus may live with surveying a single window of the whole building. In contrast, RGB cameras are sensitive to the difference among windows (e.g., at different floors), and thus require surveying individual windows. Higher-value buildings, such as city halls, schools, and hospitals are wroth for finer-grained site surveys for higher classification accuracy. In addition to sensor modalities, some analyzers may also impose additional site survey requirements. We propose the following sample analyzers to determine window openness:

- **Histogram.** This analyzer checks the pixel histogram and declares that a window is close iff a single local minimum is located. It requires no site survey nor training for new high-rise buildings.
- **Sum of Absolute Differences (SAD).** We conduct site survey to get two sample images (open versus close) from each window. At the run-time, this analyzer computes SAD to determine whether the subject window is closer to the open or close one.
- **Oriented Fast and Rotated Brief (ORB).** We adopt ORB [22] as the local feature detector, in order to quantify the similarity between a subject window image with two sample images (open versus close) collected when surveying each window.
- **Support Vector Machine (SVM).** This analyzer searches for the best hyperplane to separate the labeled window images, where the training data (window images) are collected during site survey. We adopt shapes, textures, and color distributions as inputs to train the SVM model. Compared to other analyzers, SVM incurs higher site survey and model training overhead.
- **Random Forest (RF).** It is an ensemble method, which adopts the same features as SVM, and also incurs higher overhead.

We have implemented and tested four sample classifiers for each of the representative sensor modalities: the RGB camera and ultrasound sensor. We avoid the combination of the RGB camera and SAD analyzer, because SAD is very sensitive to minor spatial displacement. We also skip the combination of the ultrasound sensor and ORB analyzer, because images interpolated/extrapolated from ultrasound measurements are smooth, and thus featureless.

4.3 Classifier Certainty and Accuracy

The performance of classifiers in our system can be quantified in two different metrics: *certainty* and *accuracy*. Here, certainty λ captures how confident when a classifier determines the window state. Take ORB analyzer as an example, its certainty could be the fraction of matched features. Certainty, by nature, is short-term: each measurement gives a certainty. In contrast, accuracy a describes the long-term performance of a classifier: what the fraction of correctly classified states is. The long-term accuracy has been shown to be a function of short-term certainty [2, 28]. We therefore model accuracy as a function of certainty, i.e., $a = R(\lambda)$, which can be built in various ways. In this paper, we adopt a regression model $R(\lambda)$ for each location, sensor, and classifier, using a large dataset collected from our simulator. With these models, we get to map classifier certainty to its accuracy at run-time. Last, we notice that the definition of accuracy can be generalized to the results from multiple classifiers in a measurement sequence, which is detailed in the next section.

5 MEASUREMENT SELECTION PROBLEM

5.1 Notations

A measurement sequence consists of a series of measurement $m_i = (p_i, s_i, c_i)$, specifying the location p_i, sensor s_i, and classifier c_i. While the selections of sensors and classifiers are discrete by nature, the selections of locations can be either continuous or discrete. We opt to discretize the locations for two reasons. First, selections of continuous locations lead to higher site survey and certainty/accuracy modeling overhead. Second, some sensor modalities only work at very few locations. For instance, an ultrasound sensor emits sound wave within a cone, which should completely fall on the window glass (without hitting nearby wall). Considering the properties of sensors and classifiers, we define a set of *measurement candidates* $\mathbf{M} = \{m_1, m_2, \ldots, m_M\}$. Given m_i, we can compute its expected accuracy $a_i = \Theta(m_i)$, sensing time $t_i^s = \Omega(s_i)$, and classification time $t_i^c = \Phi(s_i, c_i)$, where $\Theta(\cdot)$, $\Omega(\cdot)$, and $\Phi(\cdot)$ are empirically derived functions. We can estimate the data transfer time t_i^t by the bandwidth and data size.

The measurement selection problem computes a measurement sequence $\mathbf{L} = (m_{l(1)}, m_{l(2)}, \ldots, m_{l(L)})$, where $l(i) \in [1, M]$, so that the accuracy of the final result *fused* from the outputs of all L measurements can reach a user specified target $\hat{A}$ within a time limit $\hat{T}$. Since high-rise fire scenes are dynamic, our system *recomputes* a measurement sequence after completing *every* measurement. Doing so allows us to: (i) check if the achieved fused accuracy exceeds $\hat{A}$, and exits this waypoint earlier, and (ii) revise the remaining measurement sequence to adapt to the fire-scene dynamics. To facilitate the *recurring* re-computations, for each execution of our measurement selection algorithms, we use $\mathbf{L}'$ to denote the completed measurement sequence and $\mathbf{M}'$ to denote the set of unselected measurement candidates.

5.2 Fusing the Measurement Results

Suppose we get measurement results from L binary classifiers in $\mathbf{L}$, whose accuracy levels are $a_{l(1)}, a_{l(2)}, \ldots, a_{l(L)}$. Without loss of generality, we use '1' and '0' to represent the output of the classifiers. We define all measurement results from $\mathbf{L}$ as $\mathbf{R}(\mathbf{L}) = \{\mathbf{L}_1, \mathbf{L}_0\}$, where $\mathbf{L}_1 \subseteq \mathbf{L}$ indicates the measurements with result '1', and $\mathbf{L}_0 = \mathbf{L} \setminus \mathbf{L}_1$ represents the measurements with result '0'. We use $\hat{R}(\mathbf{R}(\mathbf{L}))$ to represent the fused result from all measurement results, and propose two policies to derive it.

Majority vote. This policy adopts the majority results, i.e., $\hat{R}(\mathbf{R}(\mathbf{L})) = \arg\max_{i=\{0,1\}} |\mathbf{L}_i|$. This policy is simple as it does not take the individual measurement accuracy into consideration. However, L must be odd to avoid tie breaking.

Probability-based. We compute the final classification result using all measurement accuracy. More specifically, we define

$P(1|\mathbf{R}(\mathbf{L}))$ and $P(0|\mathbf{R}(\mathbf{L}))$ to be the probabilities of results '1' and '0' being true, given $\mathbf{L}$. We then go with the one with higher probability, i.e., $\hat{R}(\mathbf{R}(\mathbf{L})) = \arg\max_{i\in\{0,1\}} P(i|\mathbf{R}(\mathbf{L}))$. We concretize these two probabilities in the following theorem; its proof is omitted.

Theorem 1. *The probability that '1' being true equals to:*

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

and the probability of '0' being true equals to $P(0|\mathbf{R}(\mathbf{L})) = 1 - P(1|\mathbf{R}(\mathbf{L}))$.

5.3 Formulation

Our problem computes a complete measurement sequence $\mathbf{L}$ before the time limit $\hat{T}$ so that the *expected accuracy* $A(\mathbf{L})$ can reach $\hat{A}$. Here, $\mathbf{L}$ is composed of the prior measurement sequence $\mathbf{L}'$ and the following measurement sequence $\hat{\mathbf{L}}$. It is not hard to see that a Traveling Salesman Problem (TSP) can be reduced to our problem in polynomial time by setting a uniform measurement accuracy > 0.5 and $\hat{T} = \infty$. Hence, our problem is NP-hard.

The precise definition of $A(\mathbf{L})$ depends on the adopted fusing policy. With the majority vote policy, we define $A_M(\mathbf{L})$ by summing up the probability of all measurement results, where more than half of them are correct. We assume that the accuracy from each classifier is independent, and write:

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

In this equation, a_i represents: (i) the expected accuracy from long-term average models $\Theta(m_i)$ if the measurement belongs to $\hat{\mathbf{L}}$, and (ii) the mapped accuracy from short-term certainty models $R(\lambda_i)$ if the measurement belongs to $\mathbf{L}'$.

Suppose the drone gets results $\mathbf{R}(\mathbf{L}') = \{\mathbf{L}'_1, \mathbf{L}'_0\}$ from its prior measurements. We define $A_P(\mathbf{L})$ as expected accuracy gotten by $\mathbf{L} = (\mathbf{L}', \hat{\mathbf{L}})$ following the probability-based policy, and set $P(1|R(\emptyset)) = 0.5$ if $\mathbf{L}' = \emptyset$. The following theorem derives the fused accuracy; its proof is omitted.

Theorem 2. *We can compute* $A_P(\mathbf{L})$ *by:*

$$A_P(\mathbf{L}) = \sum_{\substack{\hat{\mathbf{L}}_1\subseteq\hat{\mathbf{L}},\\ \hat{\mathbf{L}}_0=\hat{\mathbf{L}}\setminus\hat{\mathbf{L}}_1}} \max\Big(P(1|\mathbf{R}(\mathbf{L}')) \prod_{m_i\in\hat{\mathbf{L}}_1} a_i \prod_{m_j\in\hat{\mathbf{L}}_0}(1-a_j), (1-P(1|\mathbf{R}(\mathbf{L}'))) \prod_{m_i\in\hat{\mathbf{L}}_1}(1-a_i) \prod_{m_j\in\hat{\mathbf{L}}_0} a_j\Big). \quad (2)$$

Next, we use $T(\mathbf{L})$ to indicate the total time of finishing all measurements in $\mathbf{L}$, which equals to:

$$T(\mathbf{L}) = \sum_{i=1}^{L} (t^s_{l(i)} + t^c_{l(i)} + \frac{|p_{l(i)} - p_{l(i-1)}|}{V} + t^a_{l(i)} + t^t_{l(i)}), \quad (3)$$

where $t^s_{l(i)}$, $t^c_{l(i)}$, and $t^t_{l(i)}$ are the sensing, classification, and data transfer times of $m_{l(i)}$, respectively. The third term in the summation is the moving time. Here, V represents the flying speed and $p_{l(0)}$ is the drone's initial location at the current waypoint. The fourth term $t^a_{l(i)}$ is the measurement selection algorithm's running time.

With the above derived functions, we write our measurement selection problem as:

$$\max_{\mathbf{L}} \; U(A(\mathbf{L}), T(\mathbf{L})) = \sqrt{(1-e^{-\alpha A(\mathbf{L})}) \times e^{-\beta \frac{T(\mathbf{L})}{\hat{T}}}} \quad (4a)$$

$$\text{s.t.} \quad A(\mathbf{L}) \geq \hat{A}; \quad T(\mathbf{L}) \leq \hat{T}. \quad (4b)$$

Here, the objective function $U(A(\mathbf{L}), T(\mathbf{L}))$ in Eq. (4a) aims to find a good trade-off between the accuracy and time, controlled by weights α and β, which are system parameters. The utility value increases when: (i) the accuracy increases or (ii) the time decreases. We normalize $T(\mathbf{L})$ by $\hat{T}$ to keep it between 0 to 1. Notice that, our proposed algorithms work with utility functions other than the one in Eq. (4a), as no mathematical properties are assumed. For example, a weighted sum of $A(\mathbf{L})$ and $T(\mathbf{L})$ could be adopted as the utility function. The constraints in Eq. (4b) guarantee that $A(\mathbf{L})$ is higher than the target accuracy $\hat{A}$, and $T(\mathbf{L})$ is less than the time limit $\hat{T}$. We emphasize that our formulation does not specify the length of $\mathbf{L}$, i.e., L is also an output of our problem. Last, we note that by adopting either $A_M(\mathbf{L})$ or $A_P(\mathbf{L})$, the same formulation works for either majority vote or probability-based fusion.

5.4 Proposed Algorithms

The measurement selection problem may be solved optimally using exhaustive search or dynamic programming. We refer to the optimal algorithm as OPT, which serves as a benchmark. For real-time measurement selections, we propose an efficient heuristic algorithm that greedily adds the measurement that leads to the highest utility function value increase. We repeat the process until: (i) the time runs out, (ii) the target accuracy is achieved, or (iii) the utility function value can not be further improved. We call this algorithm as *HEU*, which comes with two variants: HEU_M and HEU_P.

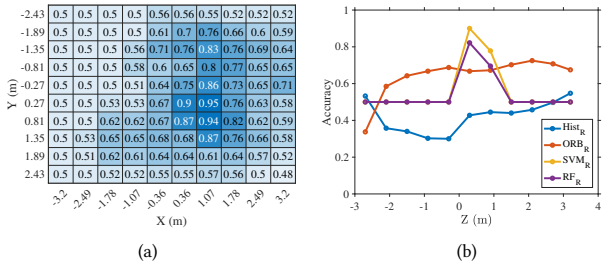
6 PERFORMANCE EVALUATIONS

6.1 Implementations

We have constructed a high-rise community in AirSim (see Fig. 3). Our AirSim environment adopts lightmap to simulate the reflection of window glasses. We chose a 10-floor high-rise building to conduct our evaluations. This building consists of 56 horizontal sliding windows. We separate them into two groups: (i) 20 windows for site surveying (2 per floor) and (ii) 36 windows for evaluations (half of them are open)[2]. We define a bounding box with a width $x = 7.1$ m, a height $y = 5.4$ m, and a depth $z = 6.0$ m for each window. Outside of this bounding box, the RGB camera or ultrasound sensor cannot cover a good portion of the target window. For the RGB camera, we equally cut the bounding box into 10^3 candidate locations. For the ultrasound sensor, we place all candidate locations along the centered horizontal axis of each (horizontal sliding) window. We vary the number of locations per half-window among $\{1, 2, 3, 4\}$ and the distance to windows among $\{0.5, 1, 1.5, 2\}$ m. This leads to 16 location sets clustering around the center of the two halves of the window, where any two adjacent ones are apart by about 4 cm. Using AirSim, we develop a set of scripts to perform the site survey to construct look-up tables for expected accuracy $\Theta(\cdot)$ for each candidate location we set in the bounding box using different

[2]Real site surveys and measurements can be done if cost is not a concern.

sensors/classifiers, as well as sensing time $\Omega(\cdot)$ and classification time $\Phi(\cdot)$. We also model the certainty to accuracy mapping $R(\cdot)$ as linear functions. These tables/models are used as inputs by our algorithms and simulator.



(a) (b)

Figure 5: Sample accuracy: (a) SVM an RGB ($Z = 0.03$ m) and (b) classifiers on RGB at various Z's (centered).

Our proposed classifiers achieve good accuracy after being trained with the 20 site-surveyed windows. Due to space limit, we show sample accuracy results in Fig. 5. Fig. 5(a) reveals that the SVM classifier works well with an RGB camera, especially at the center of individual windows. Fig. 5(b) reports the performance of all four classifiers with an RGB camera at different depth (while X and Y are centered). As we train our classifiers at the central point of the bounding box, the accuracy gets better when close a location to center. An exception is the histogram classifier, which requires no training (site surveying). Because none of the classifiers/sensors perform well at all locations, our measurement selection algorithm that fuses multiple measurements is crucial to achieve high accuracy.

We have built an event-driven simulator using C++. For each target window, we first invoke the measurement selection algorithm to get $\mathbf{L}$. We then fly the drone to the location of the next measurement on $\mathbf{L}$. Next, our simulator calls AirSim to gather the virtual sensor (e.g., a photo-realistic RGB camera) data and then executes the classifiers. Upon getting the (single) measurement result, we check if the time runs out. If not, we invoke the measurement selection algorithm again for an updated $\mathbf{L}$. Note that, the measurement selection algorithms and classifiers can be hosted on the drone or at the ground control station depending on the capabilities of hardware devices. If not otherwise specified, we place these algorithms on the ground control station. Our simulator saves detailed log files, which are analyzed offline to quantify the performance of different measurements selection algorithms under the same scenarios.

We have implemented our proposed HEU_M and HEU_P algorithms. We have also implemented the benchmark OPT algorithm by enumerating all possible measurements. We are not aware of any prior work considering the fine-grained measurement selection algorithm. Hence, we also implemented the following three algorithms to mimic the current practices: (i) CHC_R (Central Histogram Classifier) performs a measurement with the RGB camera and histogram classifier at the center of the bounding box [31], (ii) CHC_U selects 4 locations at 1 m from the window for the ultrasound sensor [11], and (iii) RAN (Random) selects random measurements until the time $\hat{T}$ is up or the target accuracy $\hat{A}$ is met.

6.2 Setup

We run the same evaluation setup with our and other measurement selection algorithms to compare their performance. We set the drone speed $V = 3.5\ m/s$ and network bandwidth to be 24 Mbps. We set the drone rotor power consumption to be 3800 W [5], WiFi energy consumption to be 0.13 mJ/bit [26]. An RGB camera takes 0.36 W (and 1 s) to take an image, and an ultrasound sensor takes 0.01 W (and 0.49 s) to collect a distance value. We vary the following parameters. *Target accuracy* $\hat{A} \in \{0.7, 0.8, 0.9\}$, where 0.8 is the default parameter. *Time limit* $\hat{T} \in \{5, 10, 30, 60\}$ *seconds* for each window, where 30 is the default parameter. To avoid unnecessary measurements, we remove candidates with expected accuracy $<$ 0.5. We define three *candidate sampling policies* to filter out some measurement candidates for better accuracy and lower overhead. The policies are: (i) E_8 (Equally-distance), which keeps $8 \times 8 \times 8$ centered locations, (ii) E_4, which skips every other location of E_8 along all three axes, and (iii) E_2, which does the same to E_4. We refer to no sampling as E_{10}, which is the default parameter. Note that candidate sampling is not applied on the ultrasound sensor, as it has much fewer candidates.

We measure the following performance metrics:

- *Overall accuracy* $O_M(\mathbf{L})$ and $O_P(\mathbf{L})$ represent the accuracy compared to the ground truth using the majority vote and probability-based fusion, respectively.
- *Expected accuracy* $A_M(\mathbf{L})$ and $A_P(\mathbf{L})$ represent the expected accuracy computed by Eqs. (1) and (2).
- *Utility function* $U_M(\cdot)$ and $U_P(\cdot)$ represent the utility function values computed by Eq. (4a) using the majority vote and probability-based fusing policies.
- *Total measurement time* $T(\mathbf{L})$ computed by Eq. (3). We also report the *running time* of the algorithm.
- *Number of measurements* is the length of $\mathbf{L}$.
- *Feasible ratio of measurements* F_M and F_P are the fraction of computed measurement selection that satisfy the accuracy $\hat{A}$ and time $\hat{T}$ constraints in Eq. (4b); the subscript indicates the fusion policy.
- *Energy* consumption of the drone when moving, sensing, and transferring data.

6.3 Results

Our proposed algorithms outperform the baselines. We first plot the accuracy and utility function values over time from a sample window in Fig. 6. We make several observations from this figure. First, all considered algorithms complete all measurements within $\hat{T}$. Second, our proposed HEU_M and HEU_P significantly outperform the baselines in terms of expected accuracy (Figs. 6(a) and 6(b)), which could be as high as 40%. Third, our HEU_P finishes much earlier with a single measurement while achieving very high accuracy. The good performance of HEU_M and HEU_P could be attributed to the higher utility values in Figs. 6(c) and 6(d), compared to other algorithms. Note that the utility values drop along the time axis, because the penalty from longer measurement time. This is however not a concern as the achieved accuracy increases over time. Last, as similar observations are made with metrics defined with majority vote and probability-based fusion, we only report the majority vote variant in the following figures for brevity.

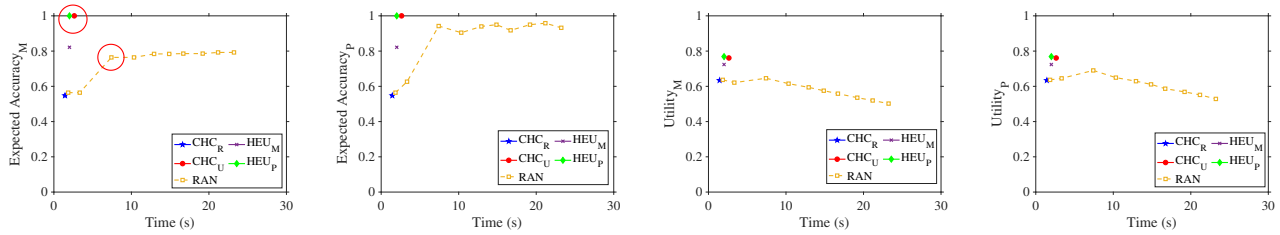


Figure 6: Results from a sample window: (a)–(b) expected accuracy and (c)–(d) utility function. We circle the ultrasound measurements in (a).

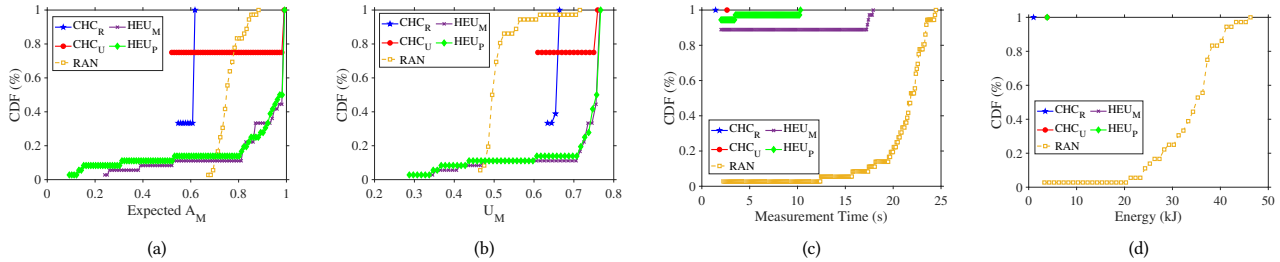


Figure 7: CDFs from 36 windows: (a) expected accuracy, (b) utility, (c) measurement time, and (d) energy consumption.

Table 1: Overall Results with Default Parameters

Algorithm	CHC_R	CHC_U	RAN	HEU_M	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)

Next, we plot the final results across all windows as Cumulative Distribution Functions (CDFs) in Fig. 7. Fig. 7(a) reveals that baselines result in over 60% of windows suffering from expected accuracy lower than 52% (CHC_U), 62% (CHC_R), and 78% (RAN); while our HEU_M and HEU_P result in 90+% expected accuracy on 60+% windows. Our good performance can be explained by the higher utility values reported in Fig. 7(b). Fig. 7(c) presents the total measurement time. Our HEU_M and HEU_P are both faster than RAN. *This means that our algorithms achieve higher accuracy within a shorter time.* Shorter measurement times also lead to lower energy consumption as reported in Fig. 7(d). Our algorithms consume one-seventh energy compared to RAN.

Table 1 reports the overall results compared to the ground truth. It is clear that our HEU_M and HEU_P outperform the baselines in terms of overall accuracy (by at least 13.89%) and feasible ratio (by at least 27.77%). Such a good performance is achieved without high overhead, on average only 1.67 and 1.22 measurements are needed for each window. *In summary, our proposed HEU_M and HEU_P achieve much higher overall and expected accuracy (up to 100% and 50% improvement), realize much higher feasible ratio (up to 100% improvement), and consume much less energy (only one-seventh), compared to the baselines.* Note that because O_M and O_P (and F_M and F_P) lead to similar observations, we report sample results in O_P (and F_P) in the rest of this paper.

Implications of diverse parameters. We first plot the performance of different algorithms under diverse target accuracy in Fig. 8, where errorbars represent the 95% confidence levels across windows. Fig. 8(a) shows that our HEU_M and HEU_P deliver good overall accuracy, which meets the increasing target accuracy. This can be observed by the feasible ratios of 80% (HEU_M) and 100% (HEU_P) at $\hat{A} = 0.90$ in Fig. 8(b), which are at least 25% higher than that of RAN. Fig. 8(c) reveals that the measurement times of HEU_M and HEU_P do not increase even when $\hat{A} = 0.90$, showing that our algorithms scale well with the target accuracy. This can be observed by the constant energy consumption in Fig. 8(d). Our algorithms also perform well under different time limits between 5 and 60 s. The figures are omitted due to page limit. Fig. 9 gives the implications of different sampling policies, where smaller E values mean fewer measurement candidates and thus lower complexity. Due to space limit, we only report overall accuracy and energy consumption. We observe that our proposed HEU_M and HEU_P still achieve almost perfect overall accuracy (Fig. 9(a)) when E is reduced. In addition, their energy consumption levels remain almost constant when E is increased. Last, we compare our proposed algorithms against the benchmark OPT using a sample window and highly-sampled measurement candidates (with 7 candidates in total) under default settings. We cannot consider larger problem size because of the exponential complexity of OPT. Compared to the expected accuracy of OPT (0.68 and 0.83 in A_M and A_P), our HEU_M and HEU_P achieve (almost) optimal accuracy, i.e., 0.69/0.83 and 0.63/0.74, respectively.

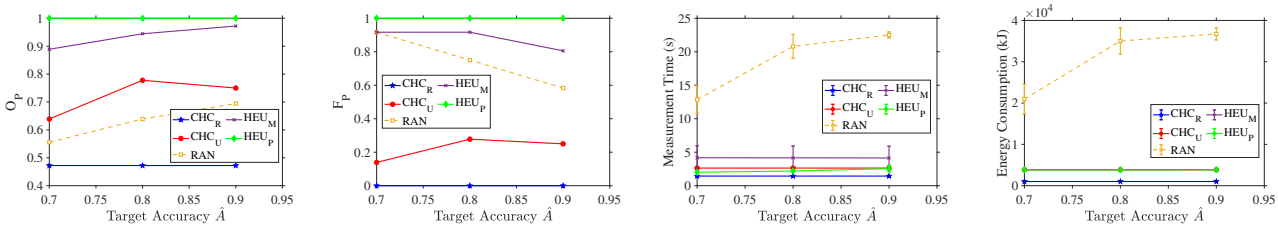


Figure 8: Implications of diverse target accuracy: (a) overall accuracy, (b) feasible ratio, (c) measurement time, and (d) energy consumption.

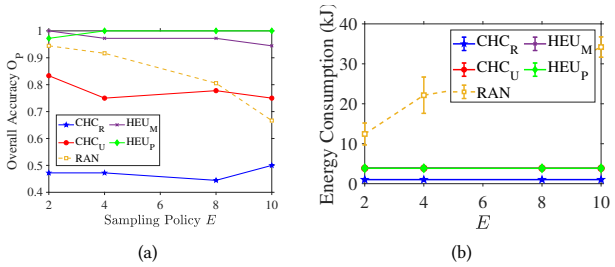


Figure 9: Implications of diverse sampling policies: (a) overall accuracy and (b) energy consumption.

In summary, our proposed algorithms achieve good performance under diverse target accuracy and sampling policies.

7 CONCLUSION

We studied the problem of developing autonomous firefighting drones to improve situational awareness in high-rise fires. Different from prior arts that focus on the coarse-grained waypoint scheduling problem, we solved the fine-grained measurement selection problem at each waypoint. Our problem is to adaptively select the best combinations of the sensor modality, classifier, and locations to achieve a target accuracy within a given time limit. We formulated the problem into an optimization problem and proposed two algorithms, HEU_M and HEU_P to solve it. We also implemented an event-driven simulator leveraging on the photo-realistic AirSim to evaluate our algorithms. Our extensive evaluation results revealed the merits of our proposed algorithms: they achieve high expected/overall accuracy, high feasible ratios, while consuming shorter measurement time and lower energy consumption, compared to baseline algorithms. Furthermore, our algorithms work well under diverse target accuracy, time limits, and sampling policies.

Our work can be extended in several dimensions. For example, the various site survey strategies and more complex network conditions can be further compared for pros/cons. We could also test the performance of our classifiers in diverse environments, such as different window types and lighting conditions. Moreover, more comprehensive measurement selection algorithms can be developed for performance guarantees at the cost of longer running time. Last, we plan to integrate the considered fine-grained measurement selection with the coarse-grained waypoint schedule algorithms for a coherent, complete system of autonomous firefighting drones.

REFERENCES

[1] 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.

[2] 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).

[3] Central Programme Office, National Fire Chiefs Council, UK. 2020. *Hazard - Fires in Tall Buildings.* Retrieved Nov. 1, 2021 from https://tinyurl.com/yycf755x

[4] H. Dang-Ngoc and H. Nguyen-Trung. 2019. Aerial Forest Fire Surveillance - Evaluation of Forest Fire Detection Model Using Aerial Videos. In *Intl. Conf. on ATC.* 142–148.

[5] DJI. [n.d.]. DJI Agras MG-1P Sprayer Drone. https://tinyurl.com/2p8pvmee.

[6] E. Bondi et al. 2018. Airsim-W: A Simulation Environment for Wildlife Conservation with UAVs. In *Proc. of ACM SIGCAS COMPASS.* 1–12.

[7] H. Ando et al. 2018. Aerial Hose Type Robot by Water Jet for Fire Fighting. *IEEE RAL* 3, 2 (Apr. 2018), 1128–1135.

[8] J. Hu et al. 2020. A Survey on Multi-Sensor Fusion Based Obstacle Detection for Intelligent Ground Vehicles in Off-Road Environments. *Front. Inf. Technol. Electron. Eng.* (2020).

[9] Madrzykowski et al. 2009. *Fire Fighting Tactics Under Wind Driven Conditions: Laboratory Experiments.* Fire Protection Research Foundation.

[10] Sea et al. 2018. Frequency-Based Multi-agent Patrolling Model and Its Area Partitioning Solution Method for Balanced Workload. In *Proc. of Springer Intl. Conf. on CPAIOR.*

[11] Z. Huang et al. 2018. Glass Detection and Recognition Based on the Fusion of Ultrasonic Sensor and RGB-D Sensor for the Visually Impaired. In *Proc. of SPIE TABS IV.*

[12] 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.

[13] M. Jarzabek, D. Lin, and H. Maas. 2020. Supervised Detection of Facade Openings in 3D Point Clouds with Thermal Attributes. *MDPI Remote Sensing* 12, 3 (February 2020), 543.

[14] Wolfgang Krull, Robert Tobera, and et al. 2012. Early Forest Fire Detection and Verification Using Optical Smoke, Gas and Microwave Sensors. *Procedia Engineering* 45 (2012), 584–594.

[15] T. Lewicki and K. Liu. 2020. Aerial Sensing System for Wildfire Detection: Demo Abstract. In *Proc. of ACM SenSys.* Yokohama, Japan, 595–596.

[16] 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.

[17] M Moore Bick. 2019. Grenfell Tower Inquiry: Phase 1 Report Overview - Report of the Public Inquiry into the Fire at Grenfell Tower on 14 June 2017. https://tinyurl.com/58zed8dc.

[18] Marcel Neuhausen, Christian Koch, and Markus Konig. 2016. Image-Based Window Detection: An Overview.

[19] 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.

[20] OSHA et al. 2015. Fire Service Features of Buildings and Fire Protection Systems. *OSHA US Department of Labor* (2015).

[21] QuadArt. 2017. Modular House. https://tinyurl.com/3hdta33c.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. 2011. ORB: An Efficient Alternative to SIFT or SURF. In *IEEE Intl. Conf. on computer vision.* 2564–2571.

[23] S. Shah, D. Dey, C. Lovett, and A. Kapoor. 2018. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *FSR.* Springer, 621–635.

[24] B. Sirmacek, L. Hoegner, and U. Stilla. 2011. Detection of Windows and Doors from Thermal Images by Grouping Geometrical Features. In *Proc. of IEEE JURSE.* Munich, Germany, 133–136.

[25] S. Smith and D. Rus. 2010. Multi-Robot Monitoring in Dynamic Environments with Guaranteed Currency of Observations. In *Proc. of IEEE CDC.* Atlanta, GA, 514–521.

[26] L. Sun, R. Sheshadri, W. Zheng, and D. Koutsonikolas. 2014. Modeling WiFi Active Power/Energy Consumption in Smartphones. In *IEEE Intl. Conf. on Distrib. Comput. Sys.* 41–51.

[27] 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.*

[28] C. Thiel. 2004. *Multiple Classifier Fusion Incorporating Certainty Factors.* Ph.D. Dissertation. Verlag nicht ermittelbar.

[29] UN DESA. 2018. *68% of the World Population Projected to Live in Urban Areas by 2050, Says UN.* Retrieved November 5, 2021 from https://tinyurl.com/8u8sxewn

[30] A. Wallar, E. Plaku, and D. Sofge. 2015. Reactive Motion Planning for Unmanned Aerial Surveillance of Risk-Sensitive Areas. *IEEE Trans. Autom. Sci. Eng.* 12, 3 (July 2015), 969–980.

[31] H. Zheng, F. Li, H. Cai, and K. Zhang. 2019. Non-Intrusive Measurement Method for the Window Opening Behavior. *Elsevier Energy Build.* 197 (August 2019), 171–176.