drones

MDPI

Article

Open Collaborative Platform for Multi-Drones to Support Search and Rescue Operations

Yao-Hua Ho * and Yu-Jung Tsai

Department of Computer Science and Information Engineering, National Taiwan Normal University, Taipei 11677, Taiwan; 60147003s@ntnu.edu.tw
* Correspondence: yho@ntnu.edu.tw; Tel.: +886-2-7749-6693

Abstract: Climate-related natural disasters have affected the lives of thousands of people. Global warming creates warmer and drier conditions which increase the risk of wildfires. In large-scale disasters such as wildfires, search and rescue (SAR) operations become extremely challenging due to low visibility, difficulty to breath, and high temperature from fire and smoke. Unmanned aerial vehicles (UAVs), such as drones, have been used to support such operations. In our previous work, a Krypto module is proposed to "sniff" out wireless signals from mobile phones to locate any possible survivors. With the increased popularity of drones, it is possible to allow people to volunteer in SAR operations with their drones. In this paper, we propose an Open Collaborative Platform for multiple drones to assist SAR operations. The open platform manages different searching drones that carry the Krypto module to collaborate by sharing information and planning search paths/areas. With our Open Collaborative Platform, anyone can participate in SAR operations and contribute to finding possible survivors. The novelty of this work is the openness and collaboration of the platform that "crowdsourcing" the searching operation to a large group of people who share information and contribute to finding possible survivors in a large disaster such as wildfires. Our experimental study shows that the Open Collaborative Platform is effective in reducing both the number of drones required and the search time for finding survivors.

Keywords: drone; disaster; wildfires; localization; search and rescue (SAR) operation; unmanned aerial vehicle (UAV)

Citation: Ho, Y.-H.; Tsai, Y.-J. Open Collaborative Platform for Multi-Drones to Support Search and Rescue Operations. *Drones* **2022**, *6*, 132. https://doi.org/10.3390/drones6050132

Academic Editors: Alessandro Giuseppi and Francesco Liberati

Received: 19 April 2022
Accepted: 13 May 2022
Published: 20 May 2022

1. Introduction

Recently, the number of catastrophic natural disasters has increased over years due to climate change. Global warming creates warmer and drier conditions which increase the risk of wildfires. According to the Center for Research on Epidemiology of Disasters (CRED) [1], the U.S.A suffered 53% of total economic losses due to record-breaking storms and wildfires in 2020. Moreover, wildfires across the west coast of the U.S.A resulted in USD 10 billion in economic losses, more than 30 loss of lives, and more than 4 million acres of land burned. In the summer of 2021, wildfires resulted in eight lives lost, and hundreds evacuated in Turkey, Greece, Italy, and Spain.

According to World Health Organization (WHO) [2], wildfires lead to a deterioration of the air quality and the loss of property, such as crops, resources, animals, and people. In addition, the disasters disrupt transportation, communications, power and gas services, and water supply. One thing that most natural disasters have in common, besides the tremendous loss of life and property, is that they are immediately followed by an almost total loss of the ability to communicate with the outside world. Without any communication, it will not be easy for victims to reach for help and report their locations. Second, search and rescue (SAR) operations face great difficulties in finding victims in large disaster areas such as wildfires. Third, wildfire disaster is often low visibility, unfriendly, inaccessible, or even dangerous for rescue teams.

Unmanned aerial vehicles (UAVs) are a good option to provide support to remote sensing and monitoring operations, such as SAR operations, with low operating costs, fast deployment, and agile maneuverability. UAVs have been used to support SAR operations in searching for survivors as early as Hurricane Katrina in 2005 [3]. However, low visibility due to thick smoke or a high level of temperature makes vision-based or thermal-based UAV-supported SAR operations ineffective; and operations often continue overnight making such vision-based SAR operations difficult.

In our previous work [4], a Krypto module, named after the flying dog from Superman comic books, is developed to "sniff" out wireless signals from any mobile device. Moreover, a drone carrying the module is able to locate any possible survivors. The Krypto module is a low-cost module (i.e., USD 100 or less) with hardware components Raspberry PI, two Wi-Fi cards, GPS Dongle, and Portable Charger. The source code of the Krypto [5] is open source that can be downloaded and installed into Raspberry PI. The module can be easily extended by adding additional sensory devices (e.g., camera, thermos-sensor/camera, etc.,) or connecting to a drone with API for existing sensors information (e.g., via DJI SDK API [6,7]). Krypto module combined with aerial drones is able to effectively patrol and sense large disaster areas to assist SAR operation.

In recent years, the popularity of drones is increased to new heights according to Federal Aviation Administration (FAA). There are more than 310,000 commercial drones and 530,000 recreational drones have been registered with the agency in 2022 [8]. The drone market continues to grow at a rapid pace. Although not all can be participating in SAR operations, many of them have the hardware and software capabilities to assist in SAR operations. In combination with the Krypto module, anyone with drones can support SAR operations by contributing to their findings on any possible survivors.

In this paper, we proposed an **Open Collaborative Platform** for multiple drones to assist SAR operations. The open platform allows drones that carry Krypto modules to collaborate in search operations. The aim of the open platform is to manage different drones to collaborate with each other by sharing information and planning search paths/areas to reduce searching time on finding survivors. With the Open Collaborative Platform, anyone can participate in SAR operations and contribute to finding possible survivors. The novelty of this work is the openness and collaboration of the platform that "crowdsourcing" the searching operation to a large group of people who share information and contribute to finding possible survivors in a large disaster such as wildfires. To the best of our knowledge and based on our reviews of current research, we have not found a similar concept that allows anyone using a drone to collaborate with SAR operations.

The contributions of this paper are as follows: (1) We introduce the **Open Collaborative Platform** for multiple drones to assist SAR operations; (2) we propose three different searching paths (*Centralized*, *Distributed*, and *Hybrid* approaches) for drones to collaborate with each other; (3) we extend the capability of the Krypto module further with a *Dynamic Searching Path* that dynamically adjusted drone's flight path based on received Wi-Fi signal, return-to-home distance, and its current power level; (4) we demonstrate that the proposed platform can reduce the searching time for mission-critical tasks such as SAR operations.

The remainder of this paper is organized as follows. In Section 2, the related works are discussed and our previous work on Krypto is presented to make the paper more self-contained. The proposed Open Collaborative Platform for multiple drones is introduced in Section 3. Our experimental and simulation results are presented in Section 4. Finally, we drew our conclusions in Section 5.

2. Related Works

For locating victims in SAR operations, there are generally two approaches: Search-and-rescue (SAR) dog [9] and life detector/locator [10]. SAR dog is trained to locate the scent of any human in a specific search area. However, SAR dogs trained for wilderness or tracking simply cannot deal with the complexities of disaster sites. Disaster sites are often surrounded by debris, dust, fire, and smoke. All of which can reduce the performance of SAR dogs. A life detector is another way to support SAR operation. Life detector is based on signs of life such as tapping, scratching, banging, voices, and chest movements by breathing (using ultra-wideband signals).

Two of the well-known companies for life detectors, LEADER [11] in California, U.S.A and Geophysical Survey Systems [10] in New Hampshire, U.S.A, estimated that the average searching distance is around 30 m in free space. Therefore, in order to cover a large disaster area, life detectors will need to be carried and operated by search teams to the site if accessible. Thus, the searching area is limited and additional delay is introduced. While multiple detectors can be deployed, the cost (e.g., US $50,000 for an average one) and interference with other devices are some of the considerations for SAR operations.

According to Cisco's Annual Internet Report and Visual Networking Index (VNI) [12], global mobile devices will grow from 8.8 billion in 2018 to 13.1 billion by 2023. Over 70 percent of the global population will have mobile connectivity by 2023. Thus, if the search team can find a signal from a mobile phone, they most likely will find a person nearby. In this paper, Wi-Fi signals are used for localization. Notice that there are other signals (i.e., 3G/4G/5G or Bluetooth) from mobile phones that can be used for localization purposes.

There are many localization techniques that use Wi-Fi signals which are classified into three types—(a) *Infrastructure-based*; (b) *Fingerprinting-based*; and (c) *Model-based*. *Infrastructure-based* techniques (e.g., [13–17]) are relied on installing specific hardware (e.g., infrared beacons or ultrasound) for indoor localization. *Fingerprinting-based* techniques (e.g., [18–25]) are pre-surveyed RF signals such as to fingerprint the surrounding signatures at every location to build a fingerprint database. The location is determined by comparing the measured fingerprint (i.e., RF signals) with the database. *Model-based* techniques (e.g., [26–31]) are based on the RF propagation model (i.e., long-distance path loss model) to estimate locations according to the measured received signal strength (RSS) values.

In the Krypto module, a *Model-based* technique is used due to no pre-install hardware (i.e., *Infrastructure-based* approach) or pre-measurement (i.e., *Fingerprinting-based* approach) is required. In *Model-based* technique, there are two methods to estimate the distance between a transmitter (Tx) and a receiver (Rx)—(a) *Time of Flight* (TOF) (e.g., [32,33]) and (b) *Received Signal Strength Indicator* (RSSI) (e.g., [28,34,35]). In the Krypto module, passive listening of RSSIs from mobile phones is used for a disaster scenario. A common path loss model, *Log-Distance Path Loss Model*, is used for free space as follows:

$$p(d) = p(d_0) + 10n \cdot \log\left(\frac{d}{d_0}\right) + X \tag{1}$$

where $p(d)$ is the total path loss (i.e., received signal strength) which measures in dBm at the length of path d, n is the path loss exponent, d_0 is the reference distance, and X is a normal (or Gaussian) random variable with zero mean.

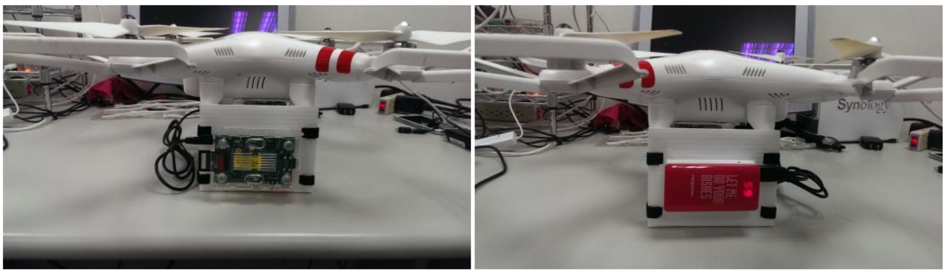
The hardware of the Krypto module consists of off-the-shelf components—Raspberry PI B+, two Wi-Fi cards, GPS Dongle, and Portable Charger (i.e., battery). The source code and executable file can be downloaded and installed on Raspberry PI for anyone who wants to participate in a search mission. Since the size and weight of the Krypto module are small and light, it can be easily mounted on most drones.

An example of the Krypto module is shown in Figure 1. An additional control wire (shown on the red, orange, and yellow ribbon cable in the left figure) can be connected to DJI's CAN port (circled in the middle figure) which allows the module to control the drone's flight path and download drone's status information (e.g., battery power, speed, location, etc.,). Note that DJI will prevent drones to fly over restricted areas (i.e., no-fly zones) such as cities or near the airport during normal use. However, the GEO No-Fly Zones can be unlocked by applying with DJI Flight Planner [36] for search and rescue operations.

(**a**) (**b**) (**c**)

Figure 1. Prototype of the Krypto module. (**a**) Krypto module consist of Raspberry PI B+ and two Wi-Fi cards. (**b**) CAN port on DJI M100 drone. (**c**) Control wire (shown on the red, orange, and yellow ribbon cable in the left figure) for connecting to DJI's CAN port.

Once, the Krypto module is connected, two Wi-Fi cards are used with one care a wireless access point (AP) and one to monitor the received signal strength (i.e., RSSI). Figure 2 shows the main components of the Krypto module on different drones. In Figure 2a, the Krypto is mounted on DJI's Phantom 2′s right-side landing gear and the portable charger on the left-side landing gear. Additional weight is added to balance both sides of the drone. In Figure 2b, the Krypto module is mounted on DJI's M100 with a ribbon cable connected to M100′s COM port to control the flight path and download the status of the drone. The total weight of Krypto is 264 g which included all Krypto's components, wiring, mounting brackets, and balance weights.



(**a**)

(**b**)

Figure 2. Krypto module on different drones. (**a**) DJI Phantom 2 with the Krypto module. (**b**) DJI M100 with the Krypto module.

3. Open Collaborative Platform

In this section, we describe the architecture of the **Open Collaborative Platform for multiple drones** to assist SAR operations in more detail.

3.1. Krypto Module

To make this paper self-contained, we first briefly describe the Krypto module's each sub-component and extended functions for the Open Collaborative Platform. Krypto has two main components: **victim localization** and **searching path planning**. To locate possible victims in a disaster area, Krypto needs to (1) detect signal strength (i.e., RSSIs) from the victim's mobile phone and (2) estimate its location according to the RSSIs. To do so, a *Virtual AP* is created by Krypto to receive any connection request from mobile phones. As our previous work [4] pointed out, several techniques are available for hacking a fake Wi-Fi connection and most mobile broadband providers also provide Wi-Fi hotspots for offloading for their users and services.

Once a mobile phone establishes a connection to Krypto, a *Sensing Program* on Krypto records the mobile phone's information, such as RSSI, timestamp, IP address, and MAC address. IP and MAC addresses are used to differentiate multiple signal sources from different mobile phones. Note that people can turn off their mobile phones' Wi-Fi. However, most people leave their Wi-Fi on all the time for advantages such as data offload and faster location estimation. We surveyed 100+ random people in our university to see if they keep their phones next to them (within their reach) all the time and turn off their Wi-Fi on their mobile phones. The results are that 99% of the people keep their phones within reach (i.e., less than 2 m) and 97% of the people never turn off their devices' Wi-Fi. In addition, other types of wireless infrastructure can be applied such as GSM/3G/4G by carrying a microcell wireless cell tower on Drone. Figure 3 shows the process of detecting process for RSSIs from mobile phones. To keep within the scope of this work, Wi-Fi signals from a mobile phone are used for locating possible victims.

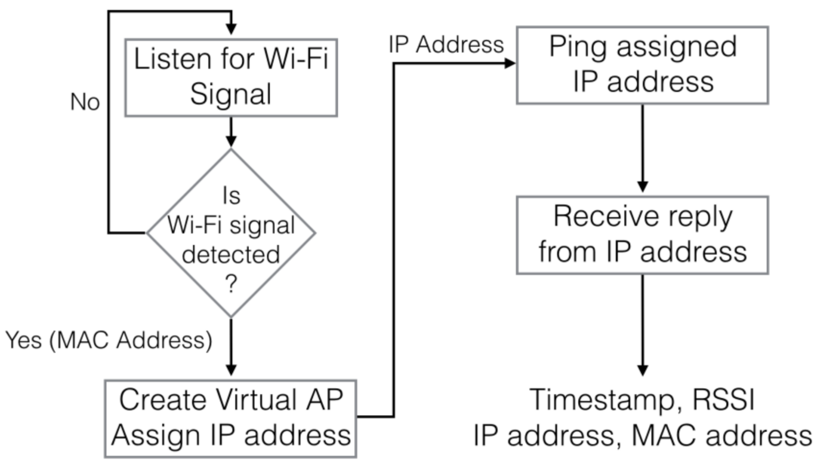


Figure 3. Signal sensing component of the Krypto module.

To estimate the location of a victim with his mobile phone, Krypto collects n number of sampling with received RSSIs for each signal source. There are 13 channels in 2.4 GHz. We can simultaneously detect good RSSIs for 6 to 8 channels per second in our experiments which can be improved if a 5 GHz network is used. Next, a *median filter* [37] is applied to the n samples to filter out noise. In our experiment, n is set to 15. This is also able to filter out other noise generated by the drone. Notice that there are several signal filtering techniques, e.g., *Kalman filters* or *Moving Average Low Pass filters*, that can be used. Next, the filtered RSSI is applied to (1) to estimate the distance between the drone and mobile phone. Once multiple estimated distance values are calculated, the location of the mobile phone can be estimated using common triangulation techniques in *Model-based* approaches as discussed in Section 2. To SAR operations, the main contribution of Krypto is not focusing on pinpointing the location of victims. It is only assisting the SAR team to identify possible victim locations in an estimated area. The SAR team needs to rely on a SAR dog or life detector to pinpoint the victim's exact location.

For the searching path planning, Krypto's searching path component is modified to allow multiple drones to collaborate using the proposed open platform. Instead of deciding on a searching path at an individual Krypto module, the Open Collaborative Platform assigns an initial searching path and updated the searching path if necessary (i.e., when other drones joined SAR operations). When the Krypto module sensed a signal source, it can dynamically adjust its path. We will discuss more on the Dynamic Searching Path in a later section. Krypto's main component is only on Raspberry Pi. The architecture of Krypto is shown in Figure 4.

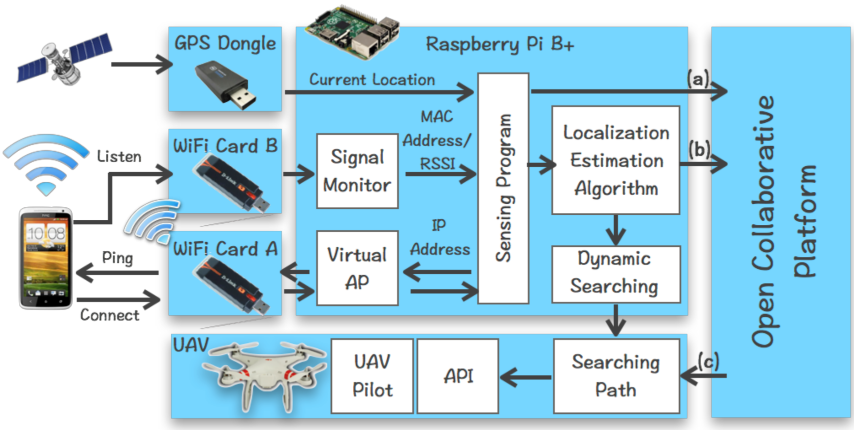


Figure 4. Architecture of the Krypto module. (a) Upload the location of drone to the platform. (b) Upload the estimated location of victims to the platform. (c) Download the searching path if necessary from the platform.

3.2. Open Collaborative Platform for Multiple Drones

Compared to [4], in which Krypto is a stand-alone module, the **Open Collaborative Platform** extends its capability for collaborating with multiple drones that carry the Krypto module. The platform is installed and operated in the control center of the SAR operation. For the experiments, the Open Collaborative Platform is installed and operated under Windows 10 O.S. on a computer with Intel i5-3570 K 3.4 GHz processor with 8G RAM. The platform centralizes path planning with the predefined searching area by the SAR team.

The platform assigns a searching path to any drone that joined the SAR operations through the registration process which is discussed in the next section. To allow drones to collaborate, the Krypto module is extended as follows. The information collected from the *Sensing Component* and estimated victim's location by *Initial Location Estimation Algorithm* are sent to the Open Collaborative Platform.

In previous work [4], we demonstrated that the Krypto can cover about 4.73 km^2 within 25 min on a single drone with a Space-Filling curve, e.g., Moore curve, Peano curve, etc. With the range of transmission range of 250 m (or 19,6350 m^2 sensing range), the *maximum flight distance* of Krypto is 10.8 (km) with 20% of power reserved for returning home and extra cost of energy due to crosswind (see Equation (5)). With multiple collaborated drones that carry the Krypto module, our platform can reduce the searching time for SAR teams in a large disaster area. This allows the SAR team to focus only on pinpointing the victim's location and saving lives.

3.2.1. Register Drone to the Open Platform

We assume that people (e.g., volunteers or SAR teams) who want to join the SAR operation already have the Krypto module ready attached to their drones. As previously discussed, all of Krypto's hardware is off-the-shelf components, and its source code or executable file can be downloaded and installed into Raspberry PI. Initially, the Krypto module will register itself with its current location information to the Open Collaborative Platform's registration service through any common wireless connection (e.g., WLAN or Cellular Network). Additional information can be entered by the user, if necessary, e.g., drone's power level, drone model number, etc. In most drones on the market, APIs are usually provided to access information from a drone or give flight path instructions, e.g., DJI Onboard SDK [6], Mobile SDK [7], or Ground Station Control [38]. After the Krypto module is registered, the platform will assign a public key (i.e., applicationServerKey) and a private key that allow the push service to authenticate the user/application (i.e., Krypto module). Once the registration process is completed with the platform, an ID and an ***initial search path*** are assigned to the registered Krypto.

The Open Collaborative Platform is open to anyone/module who is registered with the platform and can be accessed by an HTTP request using methods like GET and POST to describe the actions, e.g., basic information from Krypto or searching path information from the platform. For those drones that did not provide any APIs for control and status information, users can register their modules using a mobile phone. The open platform will register the Krypto module's current location during the registration process. Once the user is registered, an email with the assigned initial search area (or initial search path) is sent to the drone operator. For ease of discussion, we assume APIs are accessible by the Krypto module. This allows dynamic update drone's searching path. Notice that the search area or searching path can be updated by email (or message) if necessary for those drones that did not provide APIs.

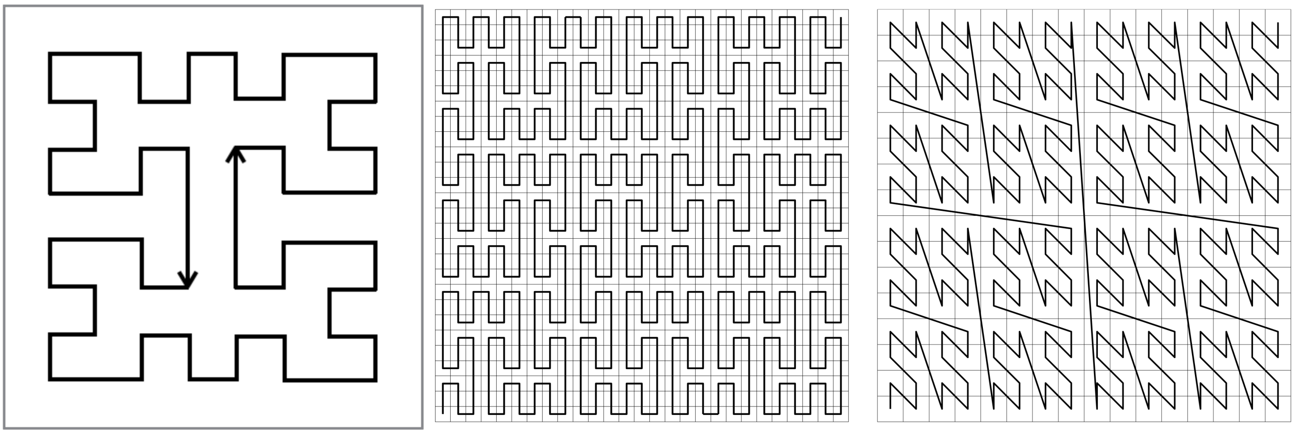
3.2.2. Communication between Krypto and Open Platform

Information is exchanged between the Krypto module and the Open Collaborative Platform using the common web protocol using GET and POST. While a drone carries the Krypto module and flies around the disaster area, a trajectory (i.e., set of location points) information is recorded and stored. When a Wi-Fi signal is detected, the sensed RSSIs are also logged. The following is the basic information sent from Krypto to the platform: *Krypto ID*, *trajectory*, *current power level*, and *timestamp*. If a Wi-Fi signal is detected, additional information will also be sent: *RSSI* logs, *IP,* and *MAC* address of the mobile phone, and *estimated location* of the mobile phone.

From the Open Platform, searching path information and update information are sent to a Krypto drone (or drone's pilot) if necessary. The information exchange between drones and the platform is encrypted with standard public/private encryption with public/private from the Platform's registration service on the payload as defined in the Message Encryption standard for Web Push. When a new drone joins the SAR operation, an update on the searching path also is sent to the Krypto module (or the drone operator) from the open platform using RESTful protocol's POST method through the common wireless connection. Different searching path planning approaches are discussed in the Section 3.3.

3.3. Collaborative Searching Path Planning

The size of a natural disaster area can be very large. The searching operations can be viewed as a *space-filling* problem with an infinite sequence of specified curves that fills the area without "holes." A Moore Curve is a variant of the Hilbert Curve that loops around the starting point and the endpoint, see Figure 5a. Other types of the space-filling curve (e.g., Peano Curve, Lebesgue Curve, Sierpinski Curve, or Sweeping Curve) [39] for the searching path can be used for different mission requirements or an irregular sharp searching area. Peano Curve is very similar to Moore Curve, see Figure 5b. Lebesgue Curve (Figure 5c) contains many sharp turns which reduce the performance of power consumption and flying speed. In addition, it is difficult for the drone to precisely follow the curve. Although Sierpinski Curve does not have sharp turns, it can be further modified to Four-circle or Four-leaf curve for an easier path to follow with a faster flying speed, see Figure 6. The sweeping Curve (Figure 7) contains fewer turns which decreases power consumption and increases flying speed with an easier path to follow.



(**a**) (**b**) (**c**)

Figure 5. Space-filling curves. (**a**) Moore curve. (**b**) Peano curve. (**c**) Lebesgue curve.

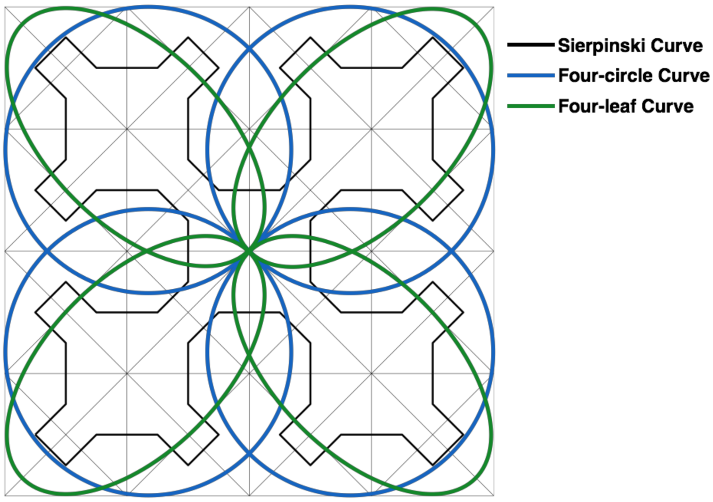
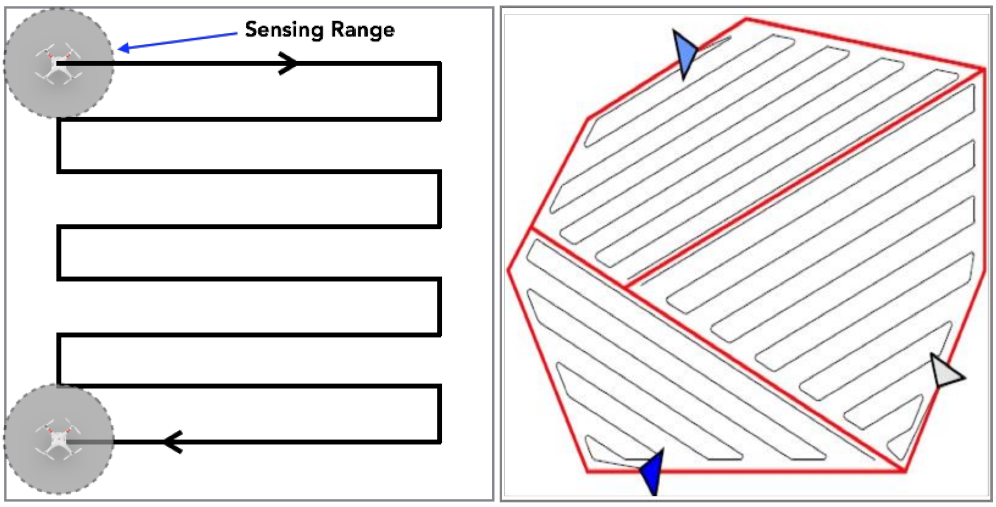


Figure 6. Sierpinski curve, four-circle curve, and four-leaf curve.



(**a**) (**b**)

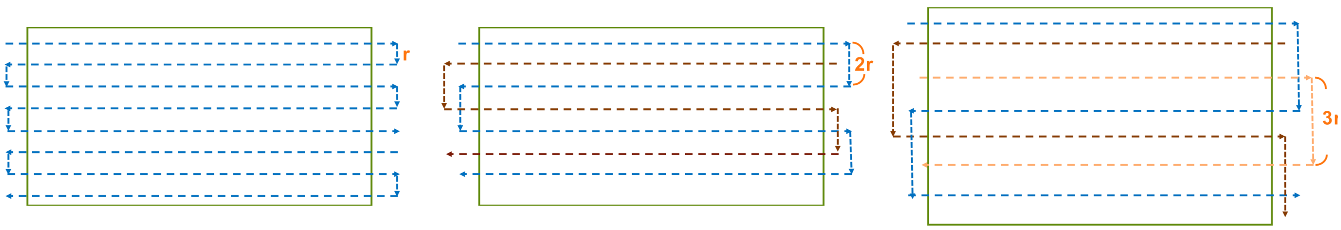
Figure 7. Searching path using sweeping curve. (**a**) Square decomposition. (**b**) Polygon decomposition.

3.3.1. Searching Path Planning

Several papers [40–42] have discussed the problem of area decomposition and optimizing sweep direction. Generally, a polygon decomposition is suggested and a sweeping pattern is used to cover decomposed area, see Figure 7b. Since the number of turns in the path can increase power consumption and reduce speed, the sweeping curve is also used for our initial searching path to fill the search area in this paper. The space between each row of the curve is the sensing range (r) of the Wi-Fi card on the Krypto, see Figure 7a.

To simplify our discussion, our assumptions are (1) the searching area is square; (2) the sensing range (i.e., Wi-Fi card) of the Krypto module is similar; and (3) each drone with a fully charged battery has a similar ability to fly similar distance. For drones, there are two requirements—(a) the assigned search path is less than the *maximum flight distance* due to its battery; and (b) remaining power is enough for the *return home* (i.e., initial starting point) *distance* to recharging/exchanging battery. Therefore, it is important to minimize the searching time while maximizing the searching area.

For multiple drones, their sweeping paths are staggered between each other, see Figure 8. Figure 8b shows two drones that initially joined the SAR operation. For each drone, the initial path is a sweeping curve where the distance between two consecutive rows is $2r$, where r is the sensing range. Similarly, if three drones initially join the SAR operation, the distance between each consecutive row for a drone is $3r$, as in Figure 8c. Since this approach needs the Open Collaborative Platform to assign a search path, we call this approach the *Centralized* approach. In this approach, when a new drone joins the SAR operation, new searching paths will be re-assigned to all drones.



(**a**) (**b**) (**c**)

Figure 8. Searching path with multiple drones. (**a**) Signal drone. (**b**) Two drones. (**c**) Three drones.

If the Krypto module is unable to connect to the open platform all the time, a *Distributed* approach is proposed. In this approach, the entire searching area is divided into m by m sub-regions. Initially, a drone randomly selects a sub-region and searches this region using a sweeping curve path. When the drone is finished searching the region, it randomly selects another sub-region. Figure 9 shows the entire searching area is divided into 49 sub-regions (e.g., 7 by 7). Two drones randomly select different sets of sub-regions (i.e., red and blue sub-regions) in the *Distributed* approach. Each drone will continue to select the next sub-region for searching if its battery power is allowed. We will discuss the return-to-home decision process in Session 3.3.3.

A drawback of the *Distributed* approach is overlapping search area by two drones selected in the same sub-region. To reduce the number of drones to select the same region, different techniques can be applied, e.g., hashing, geometric hashing, etc. Since the Krypto module has communication capability, an opportunistic type of routing algorithm (e.g., [43,44]) is applied to exchange information when two drones across each other. In this case, drones can exchange information such as which sub-regions have been visited by themselves or other drones. Therefore, when a drone selects the next sub-region to search, it will avoid those visited regions. If two drones are in the same sub-region, one of the drones will select another sub-region after the drones exchange the information. We let the drone with lower battery power stay in the current sub-region. The drone with higher battery power will select a new sub-region using geometric hashing. We call this approach the *Hybrid* approach. In both *Distributed* and *Hybrid* approaches, information such as visited sub-regions by drones is uploaded to the platform when a drone returns home or flies to a location where it can communicate with the platform. We compare the performance results of the *Centralized*, *Distributed*, and *Hybrid* approaches under different scenarios in Section 4.

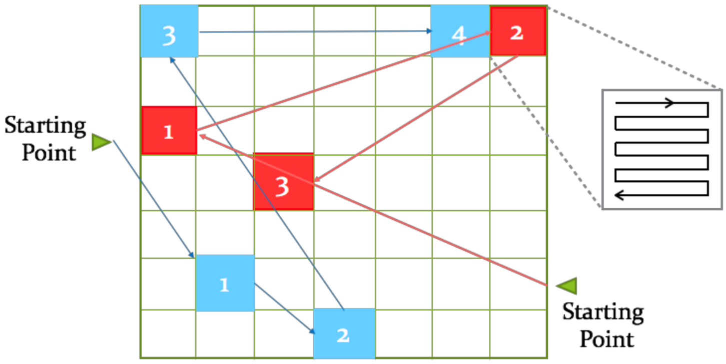


Figure 9. Distributed searching path with multiple drones.

3.3.2. Join or Rejoin SAR Operation

When a drone joins (or rejoins) an ongoing SAR operation after registering itself to the Open Collaborative Platform, an unsearched region is assigned by the platform as follows. For the *Centralized* approach, the platform recalculates the number of rows in the remaining unsearched area and divides rows equally among all the drones. In the case of being unable to divide the rows evenly, the drone with the lowest battery power level is assigned to fewer rows. The new searching path (i.e., rows) is then updated for all the drones.

For *Distributed* and *Hybrid* approaches, information of visited sub-regions (if it is available) is downloaded to its Krypto module when a new drone is registered to the platform. To begin searching, the drone's Krypto module randomly selects a sub-region and informs the open platform of its selected sub-region. When the drone reached the selected region, it begins to search the area using the Sweeping curve as discussed previously.

3.3.3. Return-to-Home

As in most SAR operations, searching drones need to return to (or re-collect by) SAR teams. Thus, we assume a drone needs to return to the starting point (e.g., initial takeoff point) or a specific location (e.g., a rendezvous point) for changing its battery, uploading data, or simply retrieving the device. Since the Krypto module records the trajectory information and other status information in real-time, a *Flight Distance* is calculated as how long and how far the drone has been flown. Next, the *Average Power Consumption* (*APC*) in *meters* per *Watts* (m/W) is calculated based on the recorded information as follows:

$$APC = Flight\ Distance\ /\ Power\ Consumed, \tag{2}$$

Since a drone needs to return to a specific location (e.g., home location), a *Return Distance* is calculated based on the (Euclidean) distance between the specific location and the drone's current location, see Figure 10. Thus, the *Remaining Distance* (in *meters*) is calculated as follows:

$$Remaining\ Dist. = Remaining\ Power \times APC \tag{3}$$

Based on our previous experiments in [4], the *maximum flight distance* consumed only 80% of power and is applied to ensure the drone is able to fly back in the case of downwind or crosswind. The maximum distance is calculated as follows:

$$Max.\ Distance = Max.\ Speed \times 80\%\ Battery\ Operating\ Time \tag{4}$$

According to (4), the *maximum flight distance* (*Maximum Distance*) for DJI Phantom 2 is as follows:

$$Max.\ Distance\ =\ 36\ (km/hr)\ \times\ 18\ (min)\ =\ 10.8\ (km) \tag{5}$$

Therefore, if the total distance between the *Flight Distance* and the *Return Home Distance* is greater than the *Maximum Distance* (4) and (5), Krypto will return to the specified location. Notice that the decision of returning home can be decided on Krypto or the Open Platform. In the case the mission is completed while additional power remains in the drone, a new mission can be requested by Krypto to the open platform, see Section 3.2.2.

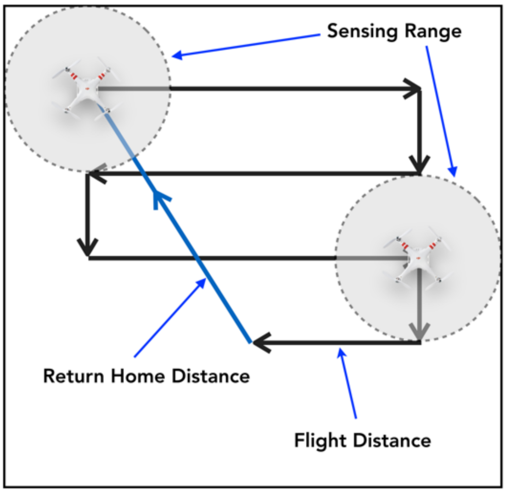


Figure 10. Return home distance.

For the *Centralized* approach, Krypto needs to determine if it is able to complete sensing the next row on its searching path. We defined the *Required Distance* for this approach is as follows:

$$\begin{aligned} Required\ Dist.\ =\ & Distance\ to\ next\ row\ +\ Length\ of\ the\ row \\ & +\ Return\ Home\ Distance\ from\ the\ end\ of\ the\ row \end{aligned} \tag{6}$$

For example, in the case of three drones joining SAR operation (see Figure 8c), the *Required Distance* is shown in Figure 11. Therefore, if the *Required Distance* is greater than the *Remaining Distance*, Krypto will decide to return home (or to the specified location).

$$\begin{aligned} Required\ Dist.\ =\ & Distance\ to\ next\ sub\_region \\ & +\ Distance\ to\ sensing\ the\ sub\_region \\ & +\ Return\ Home\ Distance\ from\ the\ sub\_region \end{aligned} \tag{7}$$

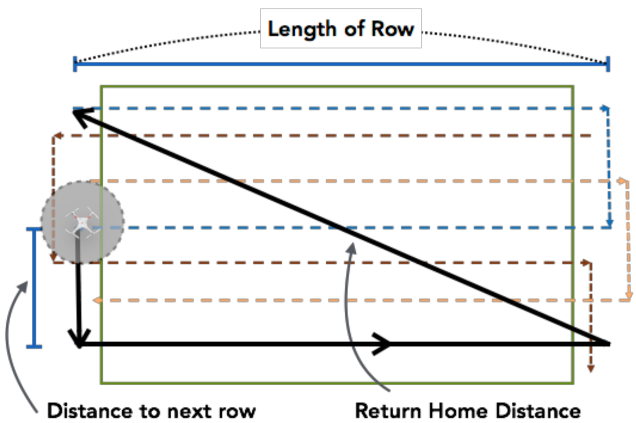


Figure 11. Required distance for centralized approach.

For example, if we have two drones in the SAR operation, the *Required Distance* is shown in Figure 12. Similarly, if the *Required Distance* is greater than the *Remaining Distance*, Krypto will decide to return home.

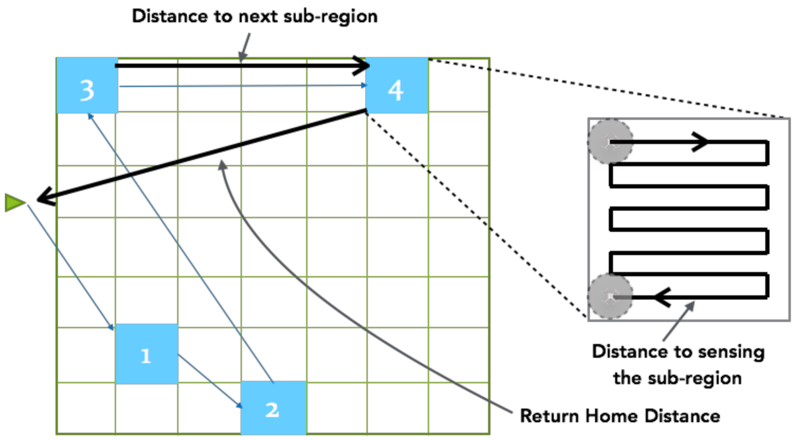


Figure 12. Required distance for distributed and hybrid approach.

3.3.4. Dynamic Searching Path

With an initial searching path assigned by the open platform, a Krypto drone follows the searching path until it sensed a Wi-Fi signal. To improve the accuracy of locating victims, Krypto will dynamically adjust its searching path. If it sensed a signal source while flying on the initial searching path, Krypto records the initial RSSI value (*Initial_R*) and continues to its path according to the initial searching path for d distance until the sensed signal becomes weaker than $Initial\_R + \alpha$.

Next, Krypto will fly a "Figure-Eight" path with a diameter of each circle equal to d and perpendicular to the initial searching path, see Figure 13. The sweeping curve is used as our initial search path. The dynamically adjusted path will produce more quadrant planes of the sensing point for the signal source. As a result, better accuracy in the location estimation. The performance result of the dynamic searching path is shown in Section 4.1.1.

To decide which half of the "Figure 8" path (e.g., top or bottom) to fly first, Krypto selects the part of the area that has not been previously scanned. Once it is finished the half of the "Figure 8" path, Krypto can decide to (*a*) finish scanning the remaining "Figure 8" path if most of the sensing points are on more than two quadrant planes or (*b*) continue the initial searching path (e.g., dotted arrow line in Figure 13) if Krypto has enough sensing points to estimate the location of the signal source.

To guarantee for a drone is able to return home, we modify the Adaptive Return-to-Home Sensing (ARS) algorithm [45] to address additional power consumption due to the dynamic searching path. The additional distance (i.e., a circumference of the circle) from the dynamic searching path is added to the *Required Distance* when deciding if it needs to return home.

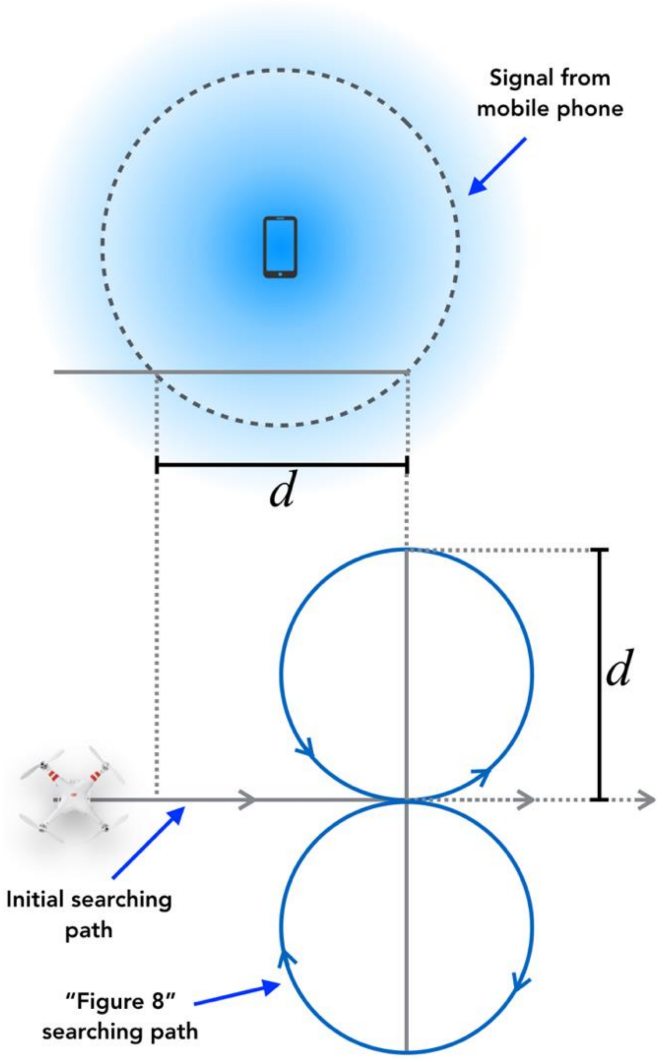


Figure 13. Dynamic searching path.

4. Performance Evaluation

We conducted field experiments and simulation studies to evaluate the proposed platform. The evaluations and results of the proposed platform are discussed in this section.

4.1. Field Experiments

First, we conducted a field experiment with our prototype to verify the effective sensing range. The Open Collaborative Platform is installed and operated under Windows 10 O.S. on a computer with Intel i5-3570 K 3.4 GHz processor with 8G RAM. Figure 14 showed the experiment location which is the track field at National Taiwan Normal University (NTNU). Krypto module mounted on DJI's M100 model with Onboard APIs that allow the Krypto module to control the flight path. The red dot was the location of the mobile phone. The average transmission range of a mobile phone is 70~110 m. RSSIs were collected on Path 1 and Path 2. Figure 14 showed an example of the experiment runs.

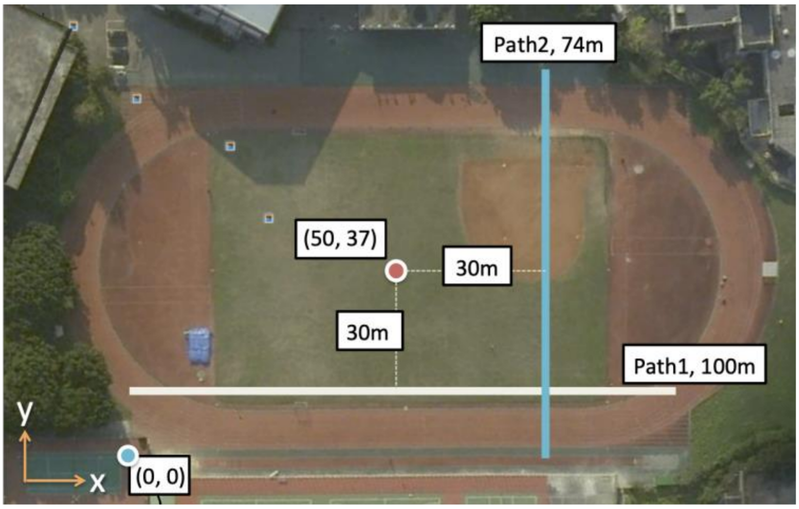


Figure 14. Experiment on NTNU's track field.

4.1.1. Verify Settings for Location Estimation

An example of RSSIs collected on Path 1 and Path 2 is shown in Figure 15. Although RSSIs increased as Krypto moved closer to the mobile phone, the RSSIs fluctuated as a result of noise and other factors. From the experiments, $n = 15$ obtained good results in terms of noise reduction with the *median filter*. After applying the noise filter and RSSI-based localization technique that mapped signal strength into distances using the *Path Loss Model* (1), the distance between the signal source (i.e., a mobile phone) and Krypto was 10 m in order to get the result of location error that was under 2 m of the radius with cumulative destruction function (CDF) of 95%. Therefore, we set the "effective" sensing range r for drones to 10 m.

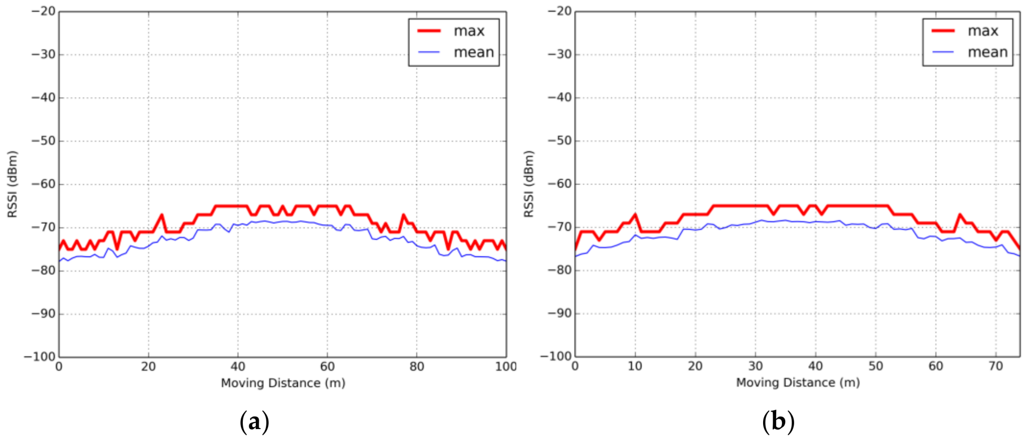


Figure 15. RSSIs collected from (**a**) Path 1 and (**b**) Path 2.

4.1.2. Effect on Accuracy of Location Estimation with Dynamic Searching Path

In our previous work, we discussed that the estimated location will have less error if sensor points were on a greater number of planes. In this experiment, we showed (1) the effect of the number of planes on the accuracy of location estimation and (2) the results after applying the *Dynamic Searching Path*. First, the RSSIs collected from Path 1 and Path 2. The location estimation points were calculated from collected RSSI data. The estimated location points were classified according to the number of planes where RSSI data are used with respect to the mobile phone, see Figure 16.

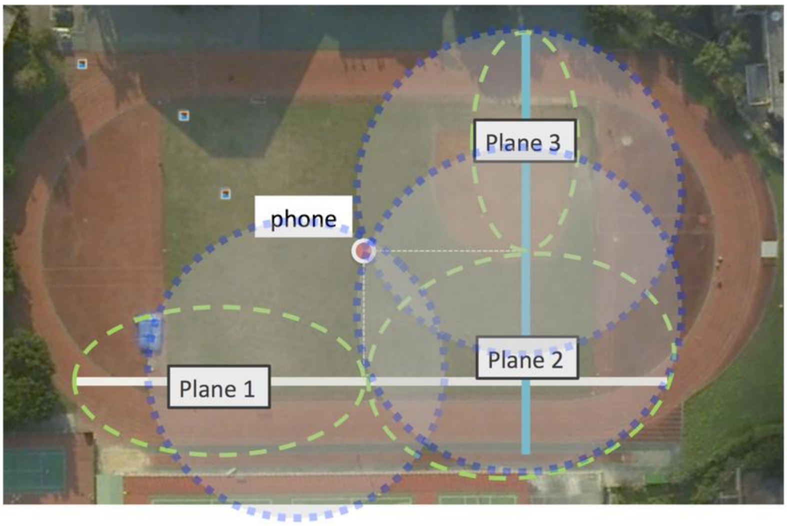


Figure 16. Sensor points at different planes.

Figure 17 shows that location estimation using sensing points on three planes results in better accuracy compared to using sensor points on only one or two planes. The location error was less than 10 *m* using sensor points on three planes. The location error for using sensor points on only one or two planes was about 40~50 m. Therefore, it is important for Krypto to dynamically adjust its searching path if it sensed a signal source to produce more sensing points on more different planes as proposed in the *Dynamic Searching Path* in Section 3.3.4.

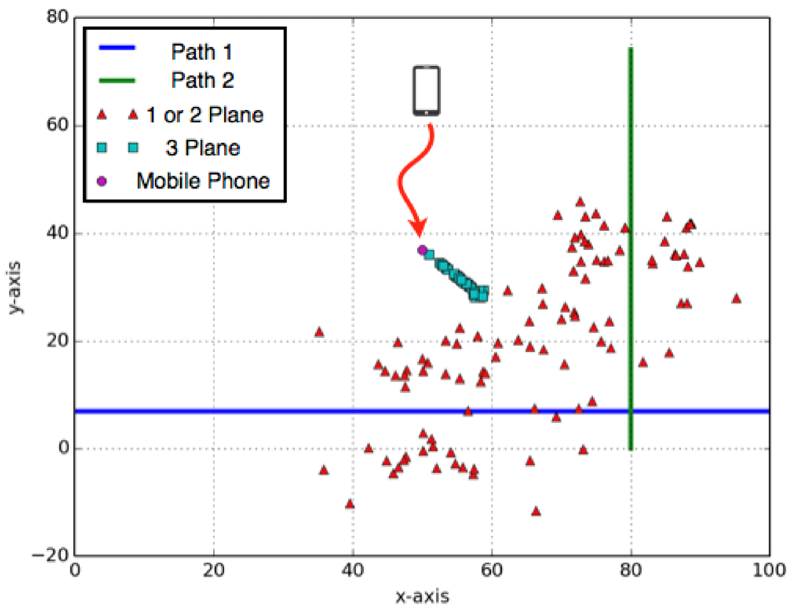


Figure 17. Effect on number of planes to location estimation.

Figure 18 shows that more estimated location points were using three or four plans which improved location estimation accuracy when applying the *Dynamic Searching Path*. We can simply solve the *geometric centroid* to those estimated location points of three or four plans for the estimated location of the signal source (e.g., a mobile phone) to reduce the error to less than 5 m.

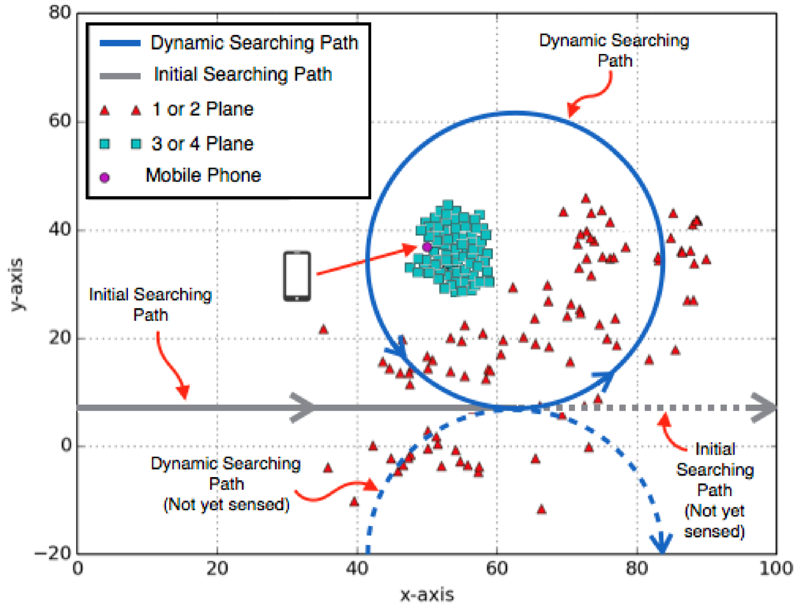


Figure 18. Number of planes with dynamic searching path.

4.2. Simulation Study

To study the effect of the number of drones and different searching paths (i.e., Centralized, Distributed, and Hybrid) under a large area, simulation was used.

4.2.1. Settings

The simulation was implemented in C++ under Windows 10 O.S on a computer with Intel i5-3570 K 3.4 GHz processor with 8G RAM. The technical specification of the drone is followed by DJI Phantom and the results from [4]. The drone had a 15.2 V with a 4480 mAh battery. Instead of using *Max. Speed* discussed in Equations (4) and (5), the drone flying speed was set to a more reasonable speed of 18 km/h with 20 min of flight time. The spatial granularity of the moving distance of the drone was one second. The effective sensing range for drones was set to 10 m (i.e., r = 10) according to our field experiments. The sensing frequency was 20 samples (i.e., 20 RSSIs) per second (0.2 Hz). A *median filter* with n = 15 was applied to the collected samples according to our field experiments.

The field configuration was an area of 1000 m by 1000 m. The number of drones varied in different settings. Initial starting points of drones were randomly placed at the edge of the field. The initial starting time (e.g., joining the SAR operation) of drones was also random. Multiple simulation runs (100 runs per setup on average) were conducted for each scenario and collected data were averaged over those runs with a 95% confidence interval.

4.2.2. Implemented Schemes

To verify different searching path designs of the proposed platform, four schemes were implemented and compared—***Random*** approach, ***Centralized*** approach, ***Distributed*** approach, and ***Hybrid*** approach.

1. In the ***Random*** approach, all drones moved in a random waypoint mobility model in which a random destination was selected. The drone moved to the selected destination before another random destination was selected.
2. In the ***Centralized*** approach, all drones followed an assigned searching path (i.e., a sweeping curve that staggered with each other) from the Open Collaborative Platform.
3. In the ***Distributed*** approach, each drone randomly selected a sub-region and searched the selected sub-region using a sweeping curve. Each sub-region was 200 m by 200 m. When a drone finished searching the selected sub-region, it randomly selected another sub-region and searched the subsequently selected sub-region.

4. In the ***Hybrid*** approach, all drones can exchange information about searched sub-regions by themselves or with other drones. Two drones can exchange information when they were within 50 m of each other. With the exchanged information, drones avoided selecting those already searched sub-regions.

4.2.3. Results on Number of Drones vs. Coverage Area

In this experiment, we varied the number of drones from 5 to 50. Each drone started randomly at a random location around the edge of the field. Figure 19 showed the *Centralized* approach required a smaller number of drones to search the entire area compared with the other two techniques (i.e., *Distributed* and *Hybrid*). The *Centralized* approach covered almost the entire area with only 15 drones. The *Hybrid* approach achieved 95% of coverage with 20 drones. Both the *Distributed* approach and the *Random* approach required more than 30 drones to achieve 95% of coverage. In addition, the time required for the *Centralized* and *Hybrid* approaches to complete the search of the entire area was far less than the *Distributed* and *Random* approaches, see Figure 22.

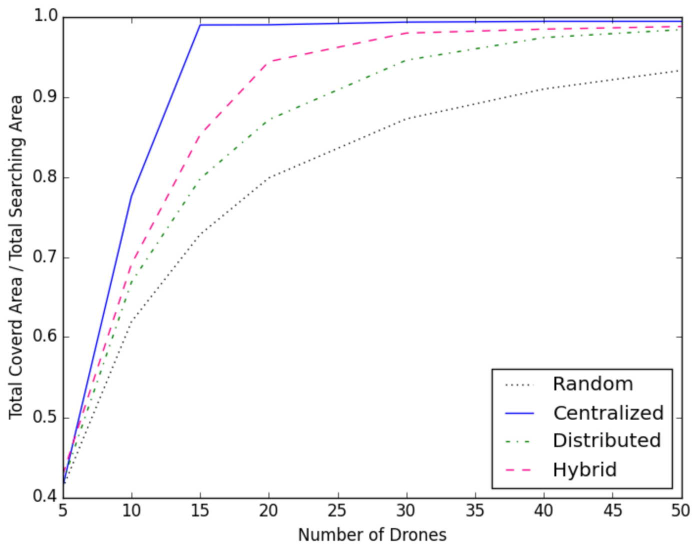


Figure 19. Number of drones vs. coverage area.

4.2.4. Results on Number of Drone vs. Overlapping Area

Figure 20 showed the ratio of the overlapping area within the searched area. The overlapping area for the *Centralized* approach was only about 22% of the total covered area. This meant the *Centralized* approach achieved high coverage without excessive overlapping when sensing the area. Compared to the *Distributed* approach, the *Hybrid* approach had slightly better performance in terms of the overlapping area. However, given the *Hybrid* approach had a more covered area (Figure 19), the actual size of the overlapping area was less than the *Distributed* approach. After an increase to 22 drones, the overlapping area for the *Random* approach became less than the *Distributed* approach and the *Hybrid* approach. The reason was that both the *Distributed* approach and the *Hybrid* approach select one sub-region at a time. Thus, once the number of drones was closely equal to the number of sub-regions in the area (i.e., 22 drones with 25 sub-regions in simulation), the probability of selecting already sensed sub-regions was higher and the overlapping area was the entire sub-region if the same sub-regions is selected.

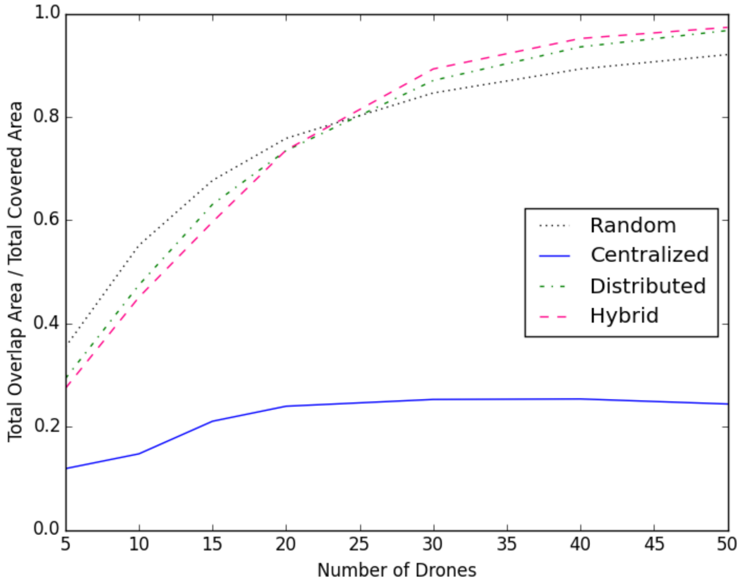


Figure 20. Number of drones vs. overlapping area.

4.2.5. Effectiveness of Each Approach vs. Number of Drones

Figure 21 shows the ratio of the total covered area to the total area covered by the flight paths of all the drones. This represented the effectiveness (or utilization) of adding more drones for each approach. As more drones were added to searching the area of 1000 m by 1000 m, more overlapped sensing areas will cover by more drones. In the *Centralized* approach, 15 drones already covered 98% of the area. Thus, it was less effective (i.e., searching unsearched areas) to add more drones. However, more drones could reduce search time, as shown in Figure 22, which was a very important concern in SAR operations.

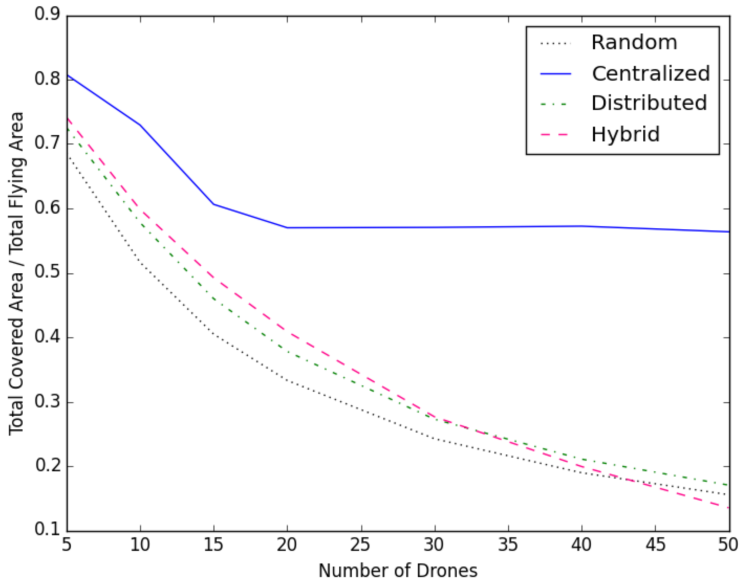


Figure 21. Effectiveness of each approach vs. number of drone.

4.2.6. Results on Average Searching Time

In this experiment, we varied the number of drones from 5 to 15. Figure 22 shows the results of the simulation running for 20 min. According to the results, the *Centralized* approach required less time (i.e., 1100 s or 18 min) to search the entire area. In fact, searching time could reduce to almost half when we doubled the number of drones. The *Hybrid* approach and the *Distributed* approach achieved only 80% of the search area after 20 min. The *Random* approach only searched less than 70% of the entire area after 20 min. Although, as we increased the number of drones, the performance of the *Distributed* and the *Random* approaches did not increase as much as the *Centralized* and *Hybrid* approaches. Thus, the results showed that an organized searching operation and sharing of the information provided by the Open Collaborative Platform could indeed reduce searching time, increase searching area, and reduce the number of drones needed in SAR operations.

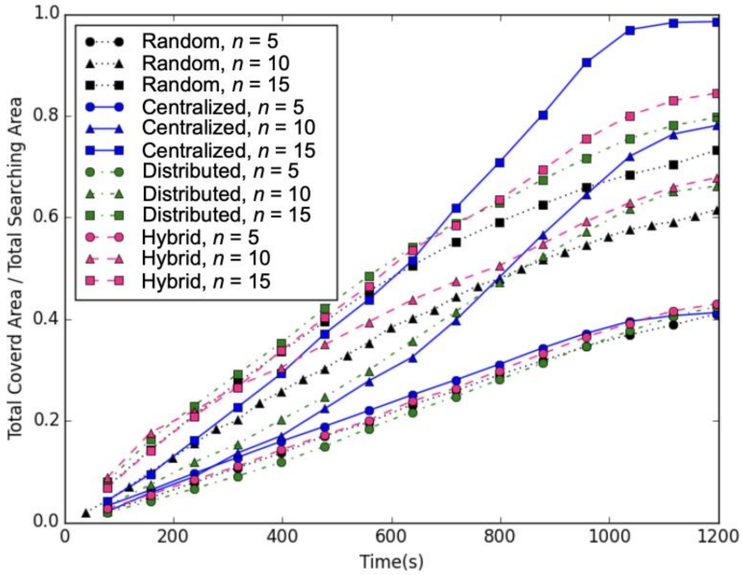


Figure 22. Average search time vs. covered area.

5. Conclusions and Future Work

In this paper, we proposed an **Open Collaborative Platform** for multiple drones to assist SAR operations. The open platform allows drones that carry the Krypto module to collaborate with each other by sharing information and planning search paths/areas in search operations. The module is a low-cost module and easy to set up with downloadable software. We extended the Krypto module for the open platform to manage and share information with others. The extended module is able to control and download the status of the drone with provided APIs, and communicate with the Open Collaborative Platform and other drones that carry the module.

In addition, we proposed three strategies, *Centralized*, *Distributed*, and *Hybrid*, for searching path planning to demonstrate the collaborative SAR operation in the open platform. To further improve the location estimation, *Dynamic Searching Path* on Krypto module is proposed. The proposed **Open Collaborative Platform** allows anyone with a drone with an attached Krypto module to participate, contribute, and share information in SAR operations. Our experimental and simulation results showed that the Open Collaborative Platform not only improves the location estimation of victims but also reduces the time and resources needed in SAR operations.

In the future, the platform can be extended to combine SAR teams on the ground with different techniques (i.e., life detector, SAR dog, thermal image, etc.,) to locate victims. In fact, a simplified Krypto module can carry by members of a searching team or even searching dogs to share the information. In addition, Krypto's main component is Raspberry Pi which can be extended with additional sensors such as air quality sensors (CO, CO_2, or particulate matter) and temperature sensors to detect any danger that the SAR team might encounter in the operation. To increase the efficiency and scalability of the searching operation, the signal from a cellular phone (4G or 5 G) can be used to narrow down the location of the victims. For the searching path of drones, the dynamic flight path can be extended to allow a drone to fly automatically toward the possible location of victims based on the detected RSSIs. Moreover, the altitude of the flight path should be dynamically adjusted when a signal is detected to improve the location estimation of victims. With the advance in sensor and wireless technologies, Open Collaborative Platform can adapt to future technology.

Author Contributions: Conceptualization, Y.-H.H. and Y.-J.T.; methodology, Y.-H.H. and Y.-J.T.; software, Y.-J.T.; validation, Y.-H.H. and Y.-J.T.; formal analysis, Y.-H.H. and Y.-J.T.; investigation, Y.-H.H. and Y.-J.T.; resources, Y.-H.H. and Y.-J.T.; data curation, Y.-H.H. and Y.-J.T.; writing—original draft preparation, Y.-H.H.; writing—review and editing, Y.-H.H.; visualization, Y.-H.H.; supervision, Y.-H.H.; project administration, Y.-H.H.; funding acquisition, Y.-H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Taiwan Ministry of Science and Technology with grant number MOST 110-2221-E-003-001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Centre for Research on the Epidemiology of Disasters, "Disaster Year in Review 2020". Available online: https://cred.be/sites/default/files/CredCrunch65.pdf (accessed on 12 April 2022).
2. World Health Organization (WHO). Wildfires. Available online: https://www.who.int/health-topics/wildfires#tab=tab_1 (accessed on 12 April 2022).
3. National Science Foundation (NSF). Hurricane Katrina Small, Unmanned Aircraft Search for Survivors in Katrina Wreckage. 2005. Available online: http://www.nsf.gov/news/news_summ.jsp?cntn_id=104453 (accessed on 12 April 2022).
4. Ho, Y.-H.; Chen, Y.-R.; Chen, L.-J. Krypto: Assisting Search and Rescue Operations using Wi-Fi Signal with UAV. In *ACM International Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*; DroNet: Florence, Italy, 2015.
5. Ho, Y.-H.; Chen, Y.-R. Krypto Source Code. Available online: https://github.com/NTNUNSL/WIFILocatedDrone (accessed on 12 April 2022).
6. DJI Onboard SDK. Available online: https://developer.dji.com/onboard-sdk/documentation/ (accessed on 15 May 2021).
7. DJI Mobile SDK. Available online: https://developer.dji.com/mobile-sdk/documentation/ (accessed on 15 May 2021).
8. Federal Aviation Administration "Drones by the Numbers". Available online: https://www.faa.gov/uas/resources/by_the_numbers/ (accessed on 12 April 2022).
9. National Association for Search and Rescue. Available online: http://nasar.org (accessed on 12 April 2022).
10. Life Locator Geophysical Survey Systems. Available online: https://www.geophysical.com/products/lifelocator-trx (accessed on 19 May 2022).
11. LEADER Search and Rescue Equipment. Available online: http://www.leadernorthamerica.com (accessed on 19 May 2022).
12. Cisco Annual Internet Report (2018–2023) White Paper. Available online: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed on 12 April 2022).
13. Ni, L.M.; Liu, Y.; Lau, Y.C.; Patil, A.P. LANDMARC: Indoor location sensing using active RFID. *Wirel. Netw.* **2004**, *10*, 701–710. [CrossRef]
14. Niculescu, D.; Nath, B. VOR base stations for indoor 802.11 positioning. In Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, Philadelphia, PA, USA, 26 September 2004; pp. 58–69.
15. Priyantha, N.B.; Chakraborty, A.; Balakrishnan, H. The cricket location-support system. In Proceedings of the ACM MobiCom, Boston, MA, USA, 1 August 2000; pp. 32–43.
16. Want, R.; Hopper, A.; Falcão, V.; Gibbons, J. The active badge location system. *ACM Trans. Inf. Syst. (TOIS)* **1992**, *10*, 91–102. [CrossRef]
17. Youssef, M.; Youssef, A.; Rieger, C.; Shankar, U.; Agrawala, A. Pinpoint: An asynchronous time-based location determination system. In Proceedings of the ACM MobiSys, Uppsala, Sweden, 19 June 2006; pp. 165–176.
18. Azizyan, M.; Constandache, I.; Roy Choudhury, R. Surroundsense: Mobile phone localization via ambience fingerprinting. In Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, Uppsala, Sweden, 20 September 2009; pp. 261–272.
19. Bahl, P.; Padmanabhan, V.N. RADAR: An in- building RF-based user location and tracking system. In Proceedings of the IEEE INFOCOM, Tel Aviv, Israel, 26–30 March 2000; Volume 2, pp. 775–784.
20. Bhasker, E.S.; Brown, S.W.; Griswold, W.G. Employing User Feedback for Fast, Accurate, Low- Maintenance Geolocationing. In Proceedings of the IEEE PerCom, Los Alamitos, CA, USA, 17 March 2004; pp. 111–120.

21. Cheng, Y.C.; Chawathe, Y.; LaMarca, A.; Krumm, J. Accuracy characterization for metropolitan-scale Wi-Fi localization. In Proceedings of the ACM MobiSys, Seattle, WA, USA, 6 June 2005; pp. 233–245.
22. Griswold, W.G.; Shanahan, P.; Brown, S.W.; Boyer, R.; Ratto, M.; Shapiro, R.B.; Truong, T.M. ActiveCampus: Experiments in community-oriented ubiquitous computing. *Computer* **2004**, *37*, 73–81. [CrossRef]
23. Park, J.; Charrow, B.; Curtis, D.; Battat, J.; Minkov, E.; Hicks, J.; Teller, S.; Ledlie, J. Growing an organic indoor location system. In Proceedings of the ACM MobiSys, San Francisco, CA, USA, 15 June 2010; pp. 271–284.
24. Varshavsky, A.; de Lara, E.; Hightower, J.; LaMarca, A.; Otsason, V. GSM indoor localization. *Pervasive Mob. Comput.* **2007**, *3*, 698–720. [CrossRef]
25. Youssef, M.; Agrawala, A. The Horus WLAN location determination system. *Wirel. Netw.* **2008**, *14*, 357–374. [CrossRef]
26. Chintalapudi, K.; Padmanabha Iyer, A.; Padmanabhan, V.N. Indoor localization without the pain. In Proceedings of the ACM MobiCom, Chicago, IL, USA, 20–24 September 2010; pp. 173–184.
27. Ji, Y.; Biaz, S.; Pandey, S.; Agrawal, P. ARIADNE: A dynamic indoor signal map construction and localization system. In Proceedings of the ACM MobiSys, Uppsala, Sweden, 19–22 June 2006; pp. 151–164.
28. Jiang, J.; Han, G.; Xu, H.; Shu, L.; Guizani, M. LMAT: Localization with a Mobile Anchor Node Based on Trilateration in Wireless Sensor Networks. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), Houston, TX, USA, 5–9 December 2011.
29. Lim, H.; Kung, L.C.; Hou, J.C.; Luo, H. Zero- configuration indoor localization over IEEE 802.11 wireless infrastructure. *Wirel. Netw.* **2010**, *16*, 405–420. [CrossRef]
30. Madigan, D.; Einahrawy, E.; Martin, R.P.; Ju, W.H.; Krishnan, P.; Krishnakumar, A.S. Bayesian indoor positioning systems. In Proceedings of the IEEE INFOCOM, Miami, FL, USA, 13–17 March 2005; Volume 2, pp. 1217–1227.
31. Sahu, P.K.; Wu, E.H.-K.; Sahoo, J. DuRT: Dual RSSI trend based localization for wireless sensor networks. *IEEE Sens. J.* **2013**, *13*, 3115–3123. [CrossRef]
32. Hara, S.; Anzai, D.; Yabu, T.; Kyesan, L.; Derham, T.; Zemek, R. A Perturbation Analysis on the Performance of TOA and TDOA Localization in Mixed LOS/NLOS Environments. *IEEE Transl. Commun.* **2013**, *6*, 679–680. [CrossRef]
33. Kotwal, S.B.; Verma, S.; Sharma, A. Region based collaborative Angle of Arrival localization for wireless sensor networks with maximum range information. In Proceedings of the Computational Intelligence and Communication Networks (CICN), Bhopal, India, 26–28 November 2010.
34. Weiss, A.J.; Picard, J.S. Maximum likelihood localization of wireless networks using biased range measurements. In Proceedings of the IEEE International Symposium on Communications and Information Technologies (ISCIT), Sydney, NSW, Australia, 17–19 October 2007.
35. Will, H.; Hillebrandt, T.; Yuan, Y.; Yubin, Z.; Kyas, M. The Membership Degree Min-Max Localization Algorithm. In Proceedings of the IEEE Ubiquitous Positioning, Indoor Navigation and Location Based Service (UPINLBS), Helsinki, Finland, 3–4 October 2012.
36. DJI Flight Planning Unlock. Available online: http://www.dji.com/zh-tw/flysafe/custom-unlock (accessed on 15 May 2021).
37. Arce, G.R. *Nonlinear Signal Processing: A Statistical Approach*; Wiley: Hoboken, NJ, USA, 2005.
38. DJI Ground Station. Available online: http://www.dji.com/product/pc-ground-station/feature (accessed on 15 May 2021).
39. Buchin, K. Organizing Point Set. Ph.D. Thesis, Fachbereich Mathematik und Informatik der Freien Universitat, Berlin, Germany, 2007. Available online: http://www.mi.fu-berlin.de/inf/groups/ag-ti/theses/phd_finished/buchin_kevin/index.html (accessed on 12 April 2022).
40. Araujo, J.F.; Sujit, P.B.; Sousa, J.B. Multiple UAV area decomposition and coverage. In Proceedings of the Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Singapore, 16–19 April 2013.
41. Maza, I.; Ollero, A. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In *Distributed Autonomous Robotic Systems 6*; Springer: Tokyo, Japan, 2007; pp. 221–230.
42. Mehdizade, M. A Decomposition Strategy for Optimal Coverage of an Area of Interest Using Heterogeneous Team of UAVs. Master's Thesis, Concordia University, Montreal, QC, Canada, 2012; pp. 77–100.
43. Chen, L.-J.; Huang, J.-H.; Chen, Y.-Y.; Tung, L.-P. Using Known Vectors to Improve Data Dissemination in Opportunistic Networks. *Int. J. Ad Hoc Ubiquitous Comput. Indersci.* **2014**, *17*, 59–69.
44. Ho, Y.-H.; Ho, A.-H.; Hua, K.A. Routing Protocols for Inter-Vehicular Networks: A Comparative Study in High-Mobility and Large Obstacles Environments. *Comput. Commun. J.* **2008**, *31*, 2767–2780. [CrossRef]
45. Huang, Y.-T.; Ho, Y.-H.; Chu, H.-H.; Chen, L.-J. Adaptive Drone Sensing with Always Return-To-Home Guaranteed. In Proceedings of the 1st International Workshop on Experiences with the Design and Implementation of Smart Objects, Paris, France, 7 September 2015.