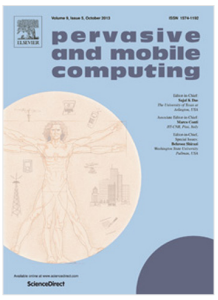
Journal Pre-proof



FastPlan: A three-step framework for accelerating drone-centric search operations in post-disaster relief

Sunho Lim, Ingyu Lee, Gyu Sang Choi, Jinseok Chae, Ellora Ashish, Eric Ward, Cong Pu

PII: S1574-1192(25)00006-9
DOI: https://doi.org/10.1016/j.pmcj.2025.102017
Reference: PMCJ 102017

To appear in: *Pervasive and Mobile Computing*

Received date : 6 June 2024
Revised date : 9 January 2025
Accepted date : 21 January 2025

Please cite this article as: S. Lim, I. Lee, G.S. Choi et al., *FastPlan:* A three-step framework for accelerating drone-centric search operations in post-disaster relief, *Pervasive and Mobile Computing* (2025), doi: https://doi.org/10.1016/j.pmcj.2025.102017.

FastPlan: A Three-Step Framework for Accelerating Drone-Centric Search Operations in Post-Disaster Relief

Sunho Lim[†*] Ingyu Lee[‡] Gyu Sang Choi[‡] Jinseok Chae[§*] Ellora Ashish[†◇] Eric Ward[†◇] Cong Pu[¶]
[†]T[2]WISTOR: TTU Wireless Mobile Networking Laboratory, Dept. of Computer Science, Texas Tech University, Lubbock, TX 79409, {sunho.lim, eashish, war71624}@ttu.edu
[‡]Dept. of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Korea, inleeatyu@yu.ac.kr, castchoi@ynu.ac.kr
[§]Dept. of Computer Science and Engineering, Incheon National University, Incheon 22012, Korea, jschae@inu.ac.kr
[¶]Dept. of Computer Science, Oklahoma State University, Stillwater, OK 74078, cong.pu@outlook.com

Abstract

Commercially well-known drones are increasingly popular and have been deployed in post-disaster relief to support traditional search and rescue operations. However, there are still barriers to conducting a drone-centric search operation, including but not limited to operational difficulty for non-professional drone pilots, inefficient pre-planned path, and more importantly initial setup delay. In this paper, we propose a small-scale prototype of a mobile framework, called *FastPlan*, to facilitate the development for expediting drone-centric search decisions and operations. The basic idea is to automate the required search operations with no or minimized user intervention. The framework consists of three major operations: map extraction, clustering, and path planning. We extract a set of target POIs and metadata from a public map (i.e., Google Maps) integrated with a customized local database to decide where and what to search. Then we deploy simple density-based clustering and search priority-based path planning strategies to efficiently group and cover the POIs. The framework can be extended flexibly by replacing individual operations with an alternative depending on the search priority, like a Lego block. We implement the framework as an Android-based mobile software (App) for a proof-of-the-concept and conduct extensive simulation experiments for performance evaluation. We analyze the different performance behavior and their implication and applicability. The results indicate that the proposed framework can be a viable approach to post-disaster relief.

Index Terms

Clustering, drone-centric search, path planning, public map, unmanned aerial vehicle (UAV)

I. Introduction

Low-altitude unmanned aerial vehicles (UAVs), commercially well-known drones (later short, drones), are increasingly popular because of their versatility, easy installation, and relatively low operating cost and have been widely deployed in diverse application domains [1], [2]. Drones equipped with sensors and devices have been playing an important role in post-disaster relief. Since the impacted areas caused by natural or man-made[1] disasters are admittedly wide, ranging from a few miles to several U.S. states, traditional rescue operations and equipment that heavily rely on a limited workforce and resources are expensive, inefficient, and time-consuming. Due to the collapse and unavailability of communication infrastructure, the

[*]Co-corresponding authors
[◇]Equal contribution
[1]In this paper, man-made disasters refer to significant and life-threatening accidents or events that could be caused by a human's mistake/error, ignorance, or even intention, such as industrial accidents, biological/chemical or oil spills, wildfires, cyberattacks, etc.

impacted areas often become a black spot, where 4G/5G and WiFi services are no longer available and survivors cannot use their smartphones for rescue through infrastructure networks. In light of these, a group of professional drone pilots flew multiple drones to locate isolated survivors, help search and rescue crews, and inspect power lines after Hurricane Idalia hit Florida in 2023 [3]. In [4], [5], drones can also locate survivors by detecting WiFi signals periodically transmitted from their smartphones without the help of infrastructure networks.

Although drones have been considered a promising alternative, we identify three major issues when drones are deployed in a time-sensitive search operation:

- *Operational Difficulty to Non-Professional Drone Pilots:* Professional drone pilots often require specialized training to complete drone-based operations successfully and safely. However, professional drone pilots may not be available when needed. Each rescue crew does not need to be a professional drone pilot but rescue crews should be able to control and utilize a drone.
- *Inefficient Pre-Planned Path for Drone Navigation:* Due to the limited manpower and resources, it is critical to efficiently search an impacted area within the golden hour[2] to maximize the number of survivors. Prior simple and pre-planned path approaches, such as an XY-path, may not work well because survivors are often located non-uniformly due to geographical and environmental reasons.
- *Initial Setup Delay before Conducting Drone-centric Operation:* Before initiating a drone-based operation, rescue crews may encounter a significant setup delay on how and where to launch the drone to complete the operation. Rescue crews may not be familiar with an impacted area geographically. This delay can significantly affect the golden hour and endanger potential survivors.

To address the issues, we propose a small-scale prototype of a mobile framework running on smartphones to promptly setup and conduct a drone-centric search operation. Unlike prior literature [4], [5], [7]–[9], we focus on a set of operations that rescue crews will face *before* launching a drone. The basic idea is to pre-program the required operations and conduct them automatically, minimizing the physical interactions between rescue crews and their drones. Our main contributions are summarized in three-fold:

- First, we implement a small-scale prototype of a mobile framework as an Android-based App, called *FastPlan*. The framework consists of three major operations: map extraction, clustering, and path planning. The framework automates search operations, reduces the setup delay of launching a drone for non-professional drone pilots (e.g., rescue crews), and improves their accessibility and usability.
- Second, we extract a set of POIs and metadata from a public map (i.e., Google Maps) to decide where and what to search by using Google Maps application programming interfaces (APIs). To seamlessly support the extraction, we build a customized local database that contains the key POIs and their geographic features and demographic information in a target area, the city of Lubbock, Texas.
- Third, the framework deploys simple density-based clustering and search priority-based path planning strategies for a proof-of-the-concept, incorporating the POI density and drone navigation distance factors. Depending on the pursued search priority, individual map extraction, clustering, or path planning operation can flexibly be replaced with an alternative in the framework, like a Lego block.

We conduct extensive simulation experiments for performance evaluation in a customized Matlab software environment. We measure and analyze the performance in terms of the path topology, percentage of covered POIs, and total navigation distance.

[2]If a severely injured or sick person does not receive care or medical treatment within the first 60 minutes, called the *golden hour*, the probability of survival decreases rapidly [6].

The simulation results indicate that the proposed framework is a viable approach to automate the pre-operations for expediting a drone-centric search operation.

The rest of the paper is organized as follows. The prior work is briefly reviewed and analyzed in Section II. We present the proposed framework with three major operations in Section III. The proposed framework is implemented and extensively simulated for performance evaluation in Section IV. Potential research issues and concluding remarks are presented in Sections V and VI, respectively.

II. Related Work

Drone-based Search and Rescue Operations: A drone carries a portable base station (BS), covers an infrastructure-less or -collapsed area, and conducts a search operation by connecting the first responders [7]. This drone-based mobile BS is deployed in an emergency or disaster spot quickly and creates an on-demand aerial network for communications. In [5], a drone equipped with a lightweight BS scans a target area and keeps track of victims' mobile phone signals to find their locations. The drone also can play a role as a mobile BS, estimate the locations of survivors, and predict their future locations. The drone adjusts its next trajectory to minimize the scan area based on the accumulated history of the predicted locations. A drone can also be equipped with a thermal camera and collect a set of thermal images to detect survivors in a post-disaster environment [10]. A survivor detection model is trained through deep convolutional neural networks. Both inference time and accuracy are considered because of the limited on-board computing power and time-sensitive detection operation. A multi-tier communication architecture [4] is proposed to relay an emergency message to a drone that is connected to the infrastructure. Mobile users equipped with multiple network interfaces are classified into a set of clusters and communicate with others to support search and rescue operations. The drone is positioned at a top tier and visits the cluster-head users and collects messages. Here, the aforementioned approaches [4], [5], [7] implicitly assume that survivors hold a mobile and wireless device and send an emergency message for rescue. When a drone is far away and disconnected from the infrastructure, however, a group of relay drones can form an aerial network [8]. The relay drones exchange control messages among themselves and maintain the connectivity between the drone and infrastructure. Group mobility of multiple drones [9] is investigated to cover non-uniformly distributed survivors in a wireless relay network. In [11], a disaster area is partitioned into a set of square-shaped layers from the center, where most survivors are located. Different numbers of drones are assigned to each layer depending on the number of survivors. It becomes a challenge to locate the center based on the density of survivors.

Map Extraction Techniques: In [12], pre-defined or analyzed map information is utilized to build a new path or enhance the capacity of path planning for a drone-based operation. However, the map information may not always be available in such an emergency spot or a post-disaster area. Time-varying environmental constraints could also reduce the accuracy of pre-processed map information. Map metadata can be updated during a drone flight to improve accuracy even with a simple camera rather than traditional expensive and complex visual sensors [13]. In [14], geospatial data of POIs are extracted by using web search engines. The types of POIs and the corresponding street names are input to Google Maps and OpenStreetMap, respectively. The search results are parsed and input to the Google search engine repetitively to further extract and validate the geospatial data. POI geo-location features are extracted on different map scales for location-based social networks, in which Google Maps is used as a base map [15]. POI data is further analyzed by incorporating social-media data that is collected from Twitter with geo-location in a metropolitan area. An online aggregation of geo-locations is proposed to accommodate continuously changing map data, associated with Google Places and Facebook [16]. Here, the POIs are confined to public places and extracted by using the Google Places APIs. In [17], historical maps are analyzed to automatically generate geo-metadata by extracting their text content. Then related location phrases are extracted from the text content and used to approximate geographic locations

in the maps. A large scale of map collections is analyzed to extract structured map metadata with little computer vision expertise [18]. Target maps are retrieved from web servers, divided into manageable sizes, and refined through deep neural network models. This approach is implemented by an open-source software library that consists of two major components, pre-processing/annotation and training/inference.

Clustering and Path Planning for Drones: Clustering techniques have been deployed to group and reduce the number of strategic locations in aerial networks. They are often partially integrated with path planning strategies to efficiently achieve the performance goal, such as covering a target area or minimizing the navigation distance. In [19], a set of cluster head drones is selected to facilitate intra- and inter-cluster routing in a large-scale drone swarm. The number of cluster head drones is optimized to minimize the communication delay based on the total number of drones. A graph model-based coverage path planning [20] is proposed for a single drone to quickly cover the area of interest by considering the cost of turns and the impact of obstacles. The drone has an on-board sensor with a static size of coverage area. In [21], multiple drones collaboratively conduct a search-and-reconnaissance operation and collect intelligence from a set of POIs. Each drone's trajectory is optimized to minimize the time to collect intelligence from all of the POIs. In [22], a set of drones is launched to cover the number of POIs that are spread out in a large-scale monitoring area. Each drone's optimal path seeks to maximize coverage rate in minimum time while considering the performance constraints of drones.

Path planning techniques can be further classified based on how environmental factors are collected and utilized. Global path planning uses geospatial data to build a path, in which heuristic algorithms [23] can be deployed. In [23], a drone equipped with a camera flies and covers a set of POIs, which is pre-processed from an existing map site. Any heuristic approach to the traveling salesman problem is applied to the POIs. Then the result of the solution is further adjusted by clustering the number of POIs to reduce the number of strategic locations and minimize energy consumption. Unlike global path planning approaches, where sudden obstacles or unexpected events are not considered, local path planning constantly senses and collects the factors from the surroundings and adjusts the current path [13]. Note that the shortest path may not always be the first priority in path planning. Depending on the performance goal, additional constraints and requirements can be considered in building a path. In addition, we discuss three-dimensional path planning in Subsec. V-B.

In summary, to the best of our knowledge, the proposed framework is the first attempt to expedite in launching and conducting of drone-centric operations with no or minimized user interventions for non-professional drone pilots, such as rescue crews. Unlike the aforementioned existing approaches in map extraction, clustering, and path planning operations, we integrate separate operations in a single framework and focus on the rescue crews' accessibility and usability. The framework is implemented as a small-scale prototype for the proof-of-concept by deploying simple operations, but it can easily be extended by replacing individual operations with a customized operation or existing approach, depending on the requirement of search operations.

III. The Proposed Framework

In this paper, we implement a small-scale prototype of a framework by developing an Android-based App, called *FastPlan*. The primary goal of the framework is to expedite a drone-based operation by reducing the setup delay of launching a drone. The basic idea is to automate required operations using smartphones with no or minimize the physical interactions between users and their drones. By reducing the reliance on skilled drone pilots, a non-professional drone pilot (e.g., a rescue crew) is able to conduct a drone-based operation quickly. In Fig. 1, we depict the proposed framework that consists of three major steps: POIs extraction, clustering, and path planning. We propose public map-based POIs extraction, density-based clustering, and search priority-based path planning operations in each step. It is noteworthy to mention that the framework can be extended by flexibly replacing individual operations with a customized algorithm or operation, like a Lego block. Thus, rescue crews

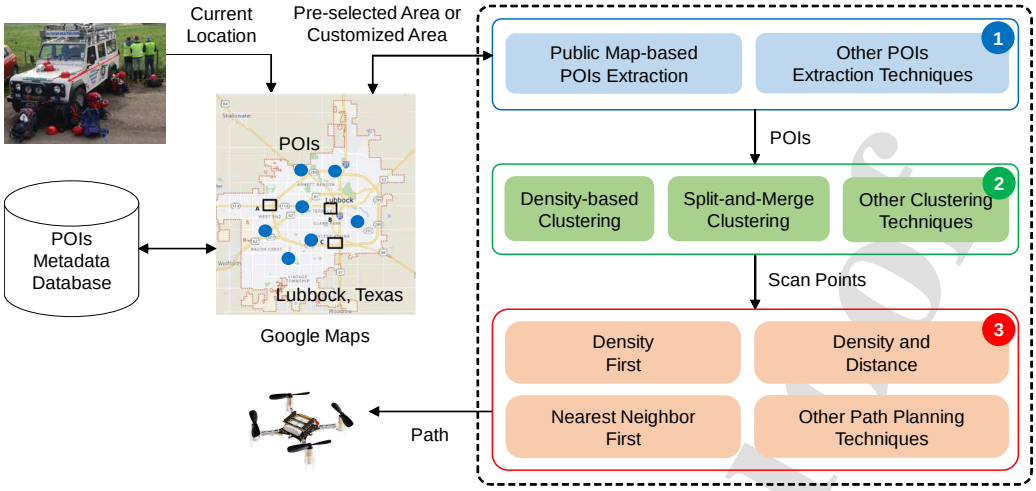


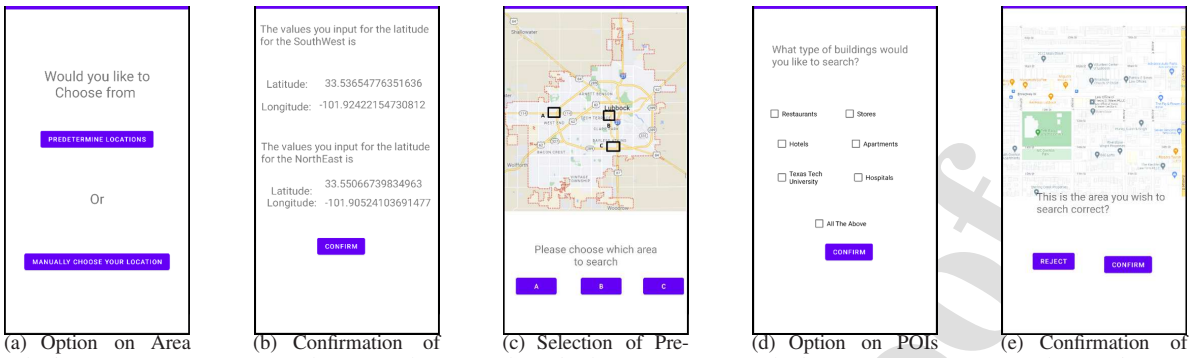
Fig. 1. An overview of the proposed framework, *FastPlan*, where three major steps and their corresponding input and output are numbered and depicted, respectively. Depending on the search requirements and environment, one of the operations can flexibly be selected in each step: (1) map extraction (see Subsec. III-A), (2) clustering (see Subsec. III-B), and (3) path planning (see Subsec. III-C). Here, a blue dot and a rectangle in Google Maps are a POI and a pre-selected area in the target area, Lubbock (TX), respectively.

can judiciously select one of the operations in each step based on the search requirements and environment. For example, they can select the three steps as public map-based POIs extraction, split-and-merge clustering, and density first path planning operations to conduct a drone-based operation. A set of variants can be available by simply combining the operations in each step. The detailed proposed operations in each step are followed.

A. Public Map-based POIs Extraction

Upon arrival in an impacted area, rescue crews may experience a non-negligible delay in identifying the POIs to search because they may not have enough geographical knowledge of the area. Since the area is often wide, rescue crews may experience another delay in deciding where to search first before conducting a drone-centric operation. To address the issues, we propose a public map-based POIs extraction using a customized and built-in database to quickly decide the size of the area to cover and identify a set of POIs to search.

First, we deploy a *public map*, Google Maps, for its popularity and accessibility and extract map metadata using its application programming interfaces (APIs) [24]. Due to the restrictions on accessing privacy- and security-sensitive information, however, Google Maps provides only limited metadata to the public, such as longitude, latitude, or address. We envision that rescue crews will have a high-quality and customized map dedicated to extracting any embedded metadata for search and rescue operations, similar to a military map. Also, the Federal Emergency Management Agency (FEMA) will enable rescue crews to access key geographic features and demographic information stored in a national database for emergency management and post-disaster recovery operations. To realize this vision, we build a small-scale POIs database (later short, database) for a proof-of-the-concept and integrate it with Google Maps to seamlessly extract POIs and metadata. Here, we confine a POI as a place where people stay or work. The database encompasses the key geographic features and demographic information in the city of Lubbock, Texas, and contains a total of 2,394 POIs including 1,431 stores, 675 restaurants, 63 apartments, 21 hospitals, 83 hotels, and 121 Texas Tech University (TTU) buildings. Each POI record contains six metadata in the database, *[Name, Latitude, Longitude, Type, Address, and Capacity]*. Although both POI's name and address are not required, we include them to prevent duplicated records and ensure data integrity. Due to the lack of POI resources, we manually identify each POI and retrieve its official name, longitude, latitude, and address by referring to Google Maps to ensure metadata accuracy. To provide popular POIs, we classify POIs into six distinct types in Lubbock, *[Store, Restaurant, Apartment, Hospital, Hotel, TTU Campus*



(a) Option on Area Selection (b) Confirmation of Manual Area Selection (c) Selection of Predetermined Areas (d) Option on POIs Selection (e) Confirmation of Search Operation

Fig. 2. A set of steps in the FastPlan, in which rescue crews follow and decide a target searching area.

Building]. Each POI's capacity implies the maximum number of people that can be accommodated in the POI space. For the sake of simplicity, we refer to the maximum occupancy permitted in the POI based on the certificate of occupancy. Note that we do not deploy automatic map extraction techniques [14] to expedite building the database because of their implicit complexity and inaccuracy. Instead, we focus on how to efficiently utilize and extract the POIs from reliable databases or resources primarily built for emergent search and rescue operations in this paper.

Second, we develop a set of sequences for rescue crews to quickly setup a drone-centric search operation. Upon arrival in an impacted area, rescue crews run the FastPlan App and complete the following sequences: (i) Check the current location and its geographical area in Google Maps; (ii) Set the size of the search area to cover by zooming in and out search boundary lines; and (iii) Select single or multiple POI types to search. Then the FastPlan automatically sets the initial search area based on the rescue crews' current location, adjusts the size of the search area, and shows the target POIs located in the area to search. With the help of the FastPlan, simplified search area, and POI selections, rescue crews can quickly decide what to search and how widely to search the area. After completing the sequences, the set of selected POIs and their metadata becomes the input to the next operation, clustering. Since the infrastructure may not be available in an impacted area, the area map and its metadata can be pre-downloaded on rescue crews' smartphones for accessibility and usability. In Fig. 2, we show a set of steps for rescue crews to follow and decide on a target searching area in the FastPlan. Rescue crews are asked to type a searching area manually (Subfig. 2(b)) or select one of the predefined searching areas (Subfig. 2(c)) in the option on area selection (Subfig. 2(a)). In the manual area selection (Subfig. 2(b)), rescue crews can also zoom in and out of the map directly using two fingers to adjust the size of the search area. Then rescue crews select single or multiple POI types to search (Subfig. 2(d)). Then the target searching area with the preferred POIs is presented to rescue crews for confirmation (Subfig. 2(e)).

B. Density-based Clustering and Variant

The goal of our clustering operation is to assist a drone in quickly visiting all the extracted POIs. Each POI is a place where people are likely found. According to the metadata (e.g., population and type) in our database, people are non-uniformly distributed in the POIs depending on the POI type. More people are found in popular POIs. The locations of POIs are also non-uniformly distributed in the area. Popular POIs are closely located together in a small area.

First, we deploy a *density-based* clustering technique. We assume that a single drone equipped with a camera scans the area with a coverage radius of r, called *coverage area*, and visits all the POIs. The size of r is not changed during the flight. Multiple coverage areas are required to cover all the POIs. Then the problem becomes how to efficiently reduce the number of coverage areas, subject to the constraint that all the POIs are located within at least one coverage area. The basic approach is to group closely located POIs, cover them by assigning the coverage area, and remove the number of POIs to cover in an early

stage. We maintain each POI's number of neighbor POIs by counting the adjacent POIs that are located within the radius of the coverage area. Then the first coverage area is assigned to the POI that has the most neighbor POIs. The second coverage area is assigned to the POI that is not covered and has the second most neighbor POIs. The procedure repeats until all the POIs are covered. A snapshot of the clustering is shown in Subfig 4(a).

Second, we enhance the density-based clustering by differentiating densely and sparsely located POIs, called *split-and-merge* clustering. The basic idea is that densely located POIs are searched thoroughly by staying the drone longer over them. On the other hand, the drone should pass quickly or stay shorter in sparsely located POIs to expedite the search operation. Our approach is to cover the POIs by assigning coverage areas with different sizes of coverage radius. Densely located POIs are split and covered by the reduced size of the coverage radius, $\frac{r}{2}$. Sparsely located POIs are merged and covered by the extended size of the coverage radius, $2r$. Note that the drone equipped with a camera that focuses on the area with a smaller coverage radius captures clear views. The camera with a larger coverage radius encompasses more space but captures blurred views. Thus, if more number of smaller coverage areas are assigned in densely located POIs, the drone can send clear views to rescue crews to support search operations. In this paper, we deploy three different sizes of the coverage radii, $\frac{r}{2}$, r, or $2r$, for the sake of simplicity. Here, r is the default coverage radius.

For the split operation, we maintain each POI's number of neighbor POIs by counting the adjacent POIs that are located within the different sizes of the coverage radii. Suppose a POI i has the number of adjacent POIs located within $\frac{r}{2}$, r, or $2r$, denoted as $g_{i,\frac{r}{2}}$, $g_{i,r}$, $g_{i,2r}$, respectively. If $g_{i,r}$ is more than a threshold, τ_{max}, the size of the coverage area is reduced to $\frac{r}{2}$. If $g_{i,2r}$ is less than a threshold, τ_{min}, the size of the coverage area is extended to $2r$. Here, both τ_{max} and τ_{min} are approximated to the number of POIs that are densely and sparsely located in the area, respectively. For the merge operation, if a POI covered by the coverage area with r is covered by another coverage area with $2r$, the original smaller coverage area is removed. Also, if the distance between two isolated POIs that are covered by the coverage area with r each is less than $2r$, the coverage area with $2r$ is assigned in the middle of POI locations to cover them. After completing the clustering operation, the center locations of coverage areas become the input to the next operation, path planning. A snapshot of the clustering is shown in Subfig. 8(a).

Note that the clustering operation is initiated after the target area (i.e., pre-selected area or customized area) is determined (see Fig. 1). In the operation, the entire POIs located in the area (i.e., the city of Lubbock, TX) are not used as an input but a subset of POIs located only in the target area is used to produce a set of coverage areas. The cluster operation has three major sub-operations to count the number of neighbor POIs, cluster POIs, and further split and merge the coverage areas by changing three coverage radii. Then the time complexity becomes $\mathcal{O}(|n'|^2)$, where $|n'|$ is the total number of POIs located in the target area. Here, $|c| < |n'| \ll |n|$, where $|c|$ is the number of coverage areas and $|n|$ is the total number of POIs located in the entire area, respectively. $|n'|$ is a quite small number compared to $|n|$ even if the POI density is high, such as in a downtown area. The proposed clustering operation is summarized in Fig. 3.

C. Search Priority-based Path Planning

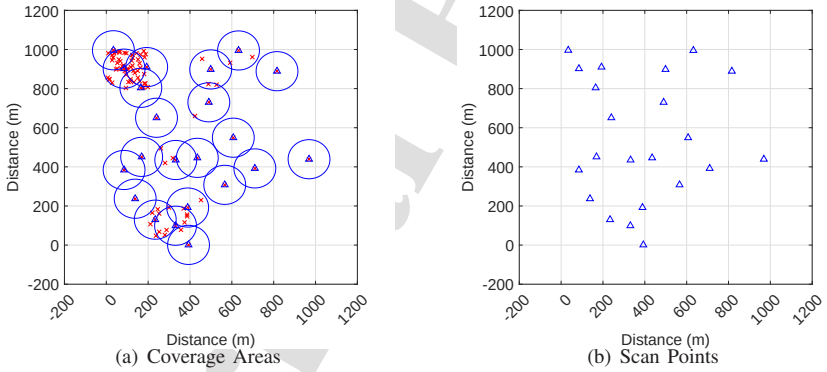
The goal of path planning is to guide a drone to search the area efficiently and quickly. In this paper, the drone launched from a base searches the area by flying toward the center locations of coverage areas, called *scan points*, to cover all the extracted POIs. We generate a set of coverage areas using the density-based clustering in Subfig. 4(a). For clarity, we only show the scan points to construct a path[3] in which the number of scan points is significantly lower than that of the POIs in Subfig. 4(b). A path is a sequence of the connected scan points. We do not blindly connect the scan points to build a path

[3]In this paper, we use the term *path* as a spatial construct consisting of a sequence of scan points to visit. We also use the term *segment* to refer to a subset of the path, specifically from an intermediate scan point to the next along the path. Then a path consists of multiple segments.

Notations:
- $n'(x_i, y_i)$: A location of POI i located in the target area.
- $c(x_j, y_j)$: A location of circle j with the coverage radius (i.e., $\frac{r}{2}$, r, or $2r$) covers one or more POIs, *coverage area*.

◇ Count the number of neighbor POIs,
for all POIs, $n'(x_i, y_i)$ **do**
  for all POIs, $n'(x_j, y_j)$ **do**
    $d \leftarrow \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}$;
    if $d \leq \frac{r}{2}$, $d \leq r$, or $d \leq 2r$ **then**
      $|g_{i,\frac{r}{2}}|$++, $|g_{i,r}|$++, or $|g_{i,2r}|$++, respectively;
sort all POIs $n'(x_i, y_i)$ in ascending order based on $|g_{i,r}|$;
◇ Cluster POIs with different coverage radii,
Create and assign a coverage area, $c(x_i, y_i)$, to the highest $|g_{i,r}|$ POI i;
for all POIs, $n'(x_i, y_i)$ **do**
  covered $\leftarrow$ false;
  for all coverage areas with r, $c(x_j, y_j)$ **do**
    $d \leftarrow \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}$;
    if $d \leq r$ **then** covered $\leftarrow$ true and **break**;
  if ! (covered) **then** /* Split and merge operations */
    if $|g_{i,r}| \geq \tau_{max}$ or $|g_{i,2r}| \leq \tau_{min}$ **then**
      Reduce r to $\frac{r}{2}$ or extend r to $2r$, respectively;
◇ Refine the merge operation,
for all coverage areas with r, $c(x_i, y_i)$ **do**
  for all coverage areas with $2r$, $c(x_j, y_j)$ **do**
    $d \leftarrow \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}$;
    if $d \leq 2r$ **then** Remove $c(x_i, y_i)$;
for (all coverage areas with r, $c(x_i, y_i)$) $\wedge$ ($|g_{i,r}| == 0$) **do**
  for (all coverage areas with $2r$, $c(x_j, y_j)$) $\wedge$ ($|g_{i,2r}| == 0$) **do**
    $d \leftarrow \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}$;
    if $d \leq 2r$ **then** Place $c(x_j, y_j)$ in the middle of locations;

Fig. 3. A pseudo code of the proposed split-and-merge clustering.



(a) Coverage Areas (b) Scan Points

Fig. 4. A snapshot of scan points is shown for the extracted POIs. Here, each scan point and POI is marked as a triangle and an x, respectively.

because each scan point covers a different number of POIs. The total distance of the path is the summation of all the segment distances.

We propose *search priority-based* path planning strategies in terms of two major factors, density and distance, and their combination. First, we consider the density as the number of POIs located in each coverage area to build a path. The basic idea is to treat each scan point differently and adjust the drone path accordingly. The drone should visit a densely populated POI area first, scan thoroughly, and stay longer. However, the drone should stay shorter in a sparsely populated POI area or even bypass it. A density of scan point s_i, ρ_i, can be calculated as $\rho_i = \frac{|g'_j|}{|n'|}$, where $|g'_j|$ is the number of POIs in coverage area j. The larger the number of covered POIs is, the higher the density is. The sequence follows the density of scan points in descending order. The drone will visit the scan point with the highest density first. We also consider the drone's navigation distance to build a path. To avoid any redundant or overlapping navigation (e.g., zigzag), it is essential to reduce the distance

TABLE I
A COMBINATION OF DENSITY AND DISTANCE FACTORS AND THEIR CORRESPONDING PRIORITY (⇑ = HIGH, ⇓ = LOW).

Density	Distance	Priority
⇑	⇑	2
⇑	⇓	1
⇓	⇓	3
⇓	⇑	4

when the drone moves from one scan point to another. Thus, we select the next scan point that is located closest to the current scan point. The procedure of finding the next scan point is repeated until the last scan point is added to the path.

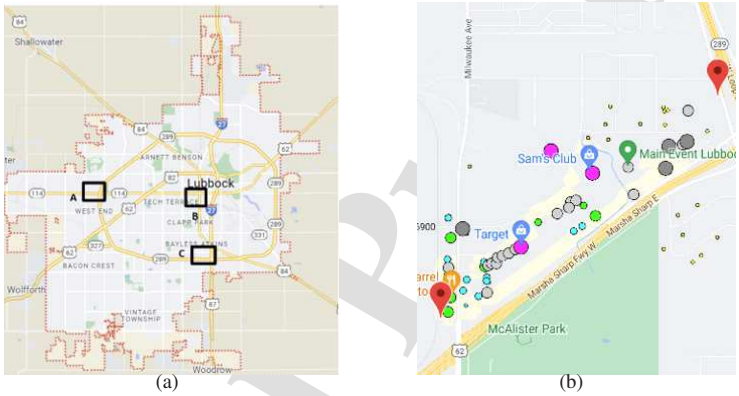
Second, the distance-based approach reduces the total navigation distances but it does not reflect the density of POIs. To balance both density and distance factors, we analyze the combination of factors and their corresponding priority in selecting the next scan point to visit from multiple scan point candidates in Table I. If one of the candidates has a high density of POIs but is a short distance apart from the current scan point, it will have the first priority to be selected as the next scan point. However, if a scan point has a low density and is located far away, it will have the least priority. If both factors are equal (e.g., all high or low), we put more weight on the density. If a scan point has a high density, it has a higher chance of being selected for the next visit. To combine the factors, we adjust the impact of each factor using a system parameter (δ) and calculate a weight (w) for the next candidates. The distance from the current scan point to the next is normalized by the default radius size of the coverage area. The scan point with the maximum weight is selected for the next visit. A weight of s_{i+1} located around s_i can be calculated as, $w_{i+1} = \delta \cdot \frac{|g'_{i+1}|}{|n'|} + (1-\delta) \cdot \frac{r}{t_i^d}$, where δ ranges from 0.2 to 0.8 and t_i^d is the segment distance between s_i and s_{i+1}.

Third, we propose three path planning strategies and investigate their implication and applicability. A random path is also presented for performance comparison. (i) *Density First (DF):* The sequence of a path is decided solely based on the density. The density is measured by the number of POIs located under the coverage area centered by the scan point. Thus, the drone visits the scan point first that covers the most POIs. Then the second most covered scan point is visited and so on. This path planning aims to cover the most POIs as quickly as possible, frequently required in search operations. (ii) *Nearest Neighbor First (NNF):* The navigation distance of a drone is considered to build a path. The drone selects the scan point that is the closest located to its current location as the next visit. Since the drone is powered by a battery, this path planning can reduce energy consumption that is proportional to the navigation distance. (iii) *Density and Distance (DD):* Both density and distance factors are balanced to select the next visit. We adjust the impact of each factor by changing the system parameter (δ), calculate the weights of candidates, and select the scan point with the highest weight. (iv) *Random (RAND):* The next visit scan point in the path is selected randomly. This path planning can be used as a performance lower bound. The proposed search priority-based path planning is summarized in Fig. 5.

Note that when a drone is launched from a base, it becomes an issue which scan point should be visited first. The first scan point can be selected randomly, or it can be the scan point that is the closest located to the base to reduce the navigation distance. In this paper, however, we select the scan point that has the highest density as the first visit. The rationale behind this approach is to quickly cover as many POIs as possible by visiting the scan points that cover the most POIs. More survivors can be found in higher populated POIs. Also, we do not traditionally pursue the shortest distance path in which both the navigation distance and flight time of the drone are directly proportional to battery energy consumption. Although the NNF has the merit of reducing the navigation distance, it may not work well in a time-sensitive search operation. Similarly, we do not minimize the navigation distance by considering the shortest return path to the base after visiting all scan points. Since the drone is equipped with a camera, it can stream the recorded impacted areas and survivors to the base in real time. Then

Notations:
ρ_i: A density of scan point s_i.
$|g'_j|$: The number of POIs in coverage area j.
$|n'|$: The total number of POIs located in the target area.
w_i: A weight based on the density and distance factors.
δ: A system parameter, ranging from 0.2 to 0.8.
t^d_{i-1}: A distance between s_{i-1} and s_i, where s_{i-1} is the current scan point.
◇ Fly and cover all scan points, $\mathbf{s} = \{s_1, s_2, \ldots, s_{|\mathbf{s}|}\}$,
repeat {
switch (Path Planning Strategy) **do**
case: DF /* Density First */
Select s_i with the highest density, $\rho_i = \frac{|g'_j|}{|n'|}$;
case: NNF /* Nearest Neighbor First */
Select s_i that is the closest located to the current location;
case: DD /* Density and Distance */
Select s_i with the maximum weight, $w_i = \delta \cdot \frac{|g'_i|}{|n'|} + (1-\delta) \cdot \frac{r}{t^d_{i-1}}$;
The drone flies toward the selected s_i;
$\mathbf{s} = \mathbf{s}$ - s_i;
} **until** $\mathbf{s} = \emptyset$; /* no more scan points left to cover */

Fig. 5. A pseudo code of the proposed search priority-based path planning.



(a) (b)

Fig. 6. Three sub-areas (L_A, L_B, and L_C) are marked by small rectangles in Subfig. 6(a). The POIs located in L_C are marked by the different sizes of circles in Subfig. 6(b).

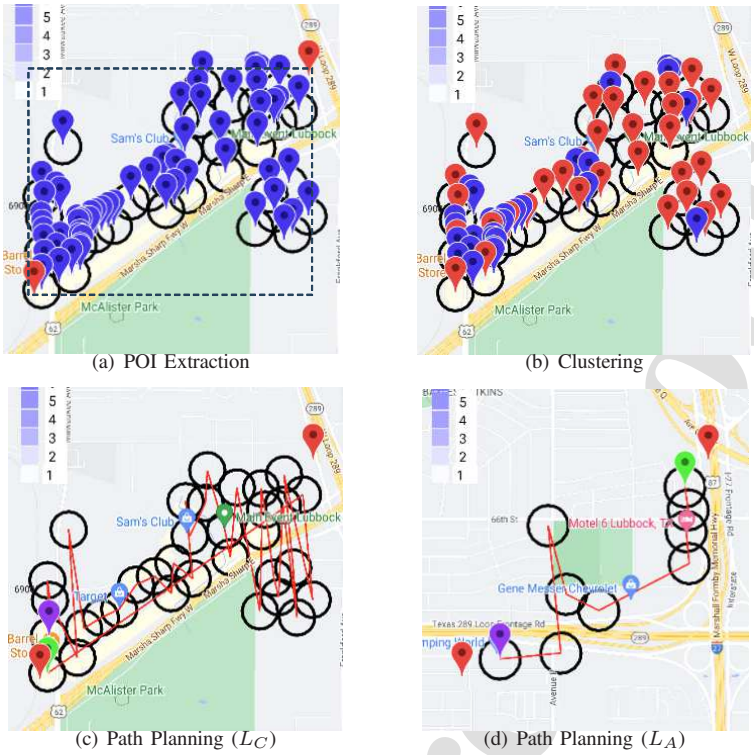
rescue crews are able to update and determine the rescue plan and decision quickly. Thus, minimizing the total navigation distance may not be the first priority in our drone-centric search operation. In addition, the drone follows a pre-determined path based on density and/or distance factors. Due to the unavailability of professional drone pilots and initial setup delay, we do not prefer path planning that may require any manipulation of drone position control or can adjust the path by dynamic factors during flight.

IV. Performance Evaluation

In this section, we implement a small-scale prototype of the framework to show the potential feasibility of the proposed approach, and then evaluate its performance with extensive simulation experiments.

A. Implementation Results

1) Implementation Testbed: We extract a set of POIs to search and its embedded map data by using the Google Maps APIs in Android Studio. To seamlessly support the POIs extraction, we build a small-scale database that contains the POIs and their metadata located in the city of Lubbock, Texas. Three sub-areas in Lubbock are deployed for experiments based on the number of POIs, POIs distribution, and local population, L_A, L_B, and L_C, as shown in Subfig. 6(a). L_A is located outside of the city and has the lowest population and few POIs. However, L_C is one of the busiest commercial areas with the highest population,



(a) POI Extraction (b) Clustering

(c) Path Planning (L_C) (d) Path Planning (L_A)

Fig. 7. Snapshots of the extracted POIs and their corresponding density-based clustering and DF path in sub-areas L_C and L_A.

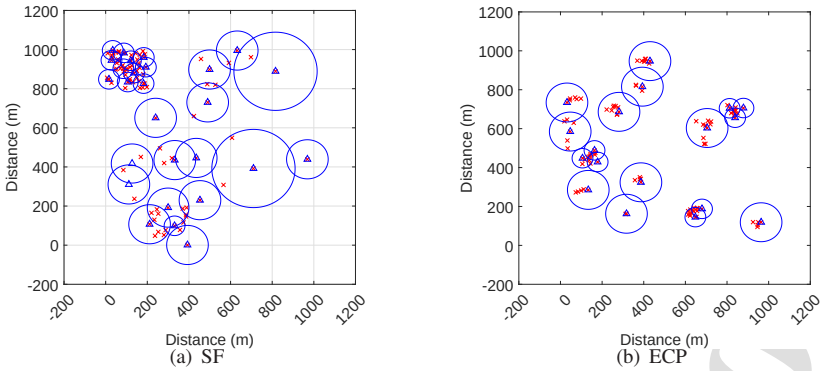
where major shopping malls, hotels, and restaurants are densely located. L_B is a non-commercial area with a mid-population and has a few POIs. In Subfig. 6(b), we show the number of POIs marked by different sizes of circles in L_C. A bigger circle implies a popular POI, where more people are found.

2) Performance Evaluation: We run the FastPlan that automatically activates three operations, map extraction, clustering, and path planning, in L_C. After the map extraction is conducted, 59 POIs are identified within a search scope (e.g., a dashed rectangle) in Subfig. 7(a). Here, the POIs are marked by blue balloons and the size of the search scope is indicated by two red balloons located in the left-most and bottom and the right-most and top, respectively. The locations of the extracted POIs become the input to the next clustering operation. A simple density-based clustering is deployed in the FastPlan to group the POIs, and 32 scan points are identified and marked by red balloons as shown in Subfig. 7(b). In the path planning, all the scan points are connected to build a path based on the DF in Subfig. 7(c). After the path is transferred to a drone, rescue crews are able to conduct a drone-based search operation. The drone is launched from the base located in the left-most and bottom and visits the first scan point with the highest density. The green and purple balloons indicate the first and last drone visits, respectively. For the same, we also run the FastPlan in another sub-area, L_A, and show a path in Subfig. 7(d). In the path, it is observed that the drone visits the scan point first which is far away from the base. This is because the scan point that has the highest POI density is selected as the first visit according to the search priority.

For the proof-of-the-concept, we deploy simple clustering and path planning operations in the FastPlan. As aforementioned, each operation can be replaceable with a customized operation flexibly depending on the search priority and requirement.

B. Simulation Results

1) Simulation Testbed: We deploy Matlab software in a customized simulation environment running on a Windows 10 machine equipped with a 1.5 GHz Intel i7 processor and 16 GB RAM. We generate 100 POIs distributed non-uniformly in a 1000 $\times$ 1000 (m^2) rectangle network. The POIs are non-uniformly distributed and follow either the scale-free (SF) [25] or



(a) SF (b) ECP

Fig. 8. The extracted POIs are covered by the different sizes of coverage radii. Here, the distribution of POIs follows either the SF or ECP.

enhanced cluster process (ECP) [26] in Fig. 8. In the SF, a network is logically divided into a set of rectangular subareas, $\mathbf{a} = \{a_1, a_2, \ldots, a_{|\mathbf{a}|}\}$. Initially, there is no POI assigned to any subarea, and thus, the population of each subarea is set to zero. The probability of each subarea being assigned for a POI is equal. When a new POI is assigned to one of the subareas, the probability changes depending on the current number of POIs assigned in each subarea. Suppose a POI is assigned to one of the subareas, a_j $(1 \leq j \leq |\mathbf{a}|)$, in the network. The popularity of each subarea (φ_j) can be calculated by $\varphi_j \leftarrow \frac{(k_j+1)^\epsilon}{\sum_j (k_j+1)^\epsilon}$, where ϵ is the clustering exponent and k_j is the population of subarea a_j. Most likely the subarea with higher popularity is chosen as a destination repeatedly. Some subareas have more POIs than others and become hub areas. The position of the POI, $n(x_i, y_i)$, is randomly located within the chosen subarea. In this paper, we divide the network into a set of subareas, 200 $\times$ 200 (m^2), in Subfig. 8(a). Since the distribution is sensitive to ϵ, we can set it to 1.4, 1.6, or 2.0. When ϵ is 2.0, a single hub area is created, where the most POIs are witnessed.

In the ECP, the distribution of POIs is sensitive to the intensity of parent points (λ), which become clusters similar to hub areas. A number of parent points are determined based on the Poisson random process with λ. The parent points are independently and uniformly located in the network. Each parent point has several child points that are independently and normally distributed in a virtual disc centered around the parent within θ radius. We set θ to 0.02 to see densely and closely located child points around the parent point. If θ is close to 1, most child points are scattered around the parent point within the disc. The number of child points is also calculated by the Poisson process with α, the mean number of children for each parent point, which is set to 0.9. Then we enhance the original cluster process to reflect the preferential attachments. We repeatedly conduct the Poisson process to generate child points in multiple rounds. In each round, the number of current child points of each parent is used to determine the number of child points in the next round. Consequently, the parent points with more child points receive more child points in the next round. In the ECP, the distribution is sensitive to λ, which can be set to 20, 30, or 50 in this paper. The distribution becomes uniform as λ increases. Compared to the SF, where only a few big hubs are observed, the ECP shows a larger number of smaller-sized clusters in Subfig. 8(b). The ECP also has no or few isolated POIs that are frequently witnessed in the SF.

In this paper, we partially follow the parameter values used in [25], [26]. ϵ and λ are set to 1.6 and 30 to clearly see the hubs and clusters, respectively. Unless otherwise specified, we deploy the proposed split-and-merge clustering in which both τ_{max} and τ_{min} are set to 10 and 2, respectively. We set the default size of the coverage radius (r) to 100 (m). The drone follows scan points to cover all POIs. If a POI is located under multiple coverage areas, the drone visits only one scan point that covers the POI. The location of the base is set to the left-most and bottom of the network, where the drone is launched. To balance the impact of density and distance factors, δ varies ranging from 0.2 to 0.8 with the step 0.2 to build a path.

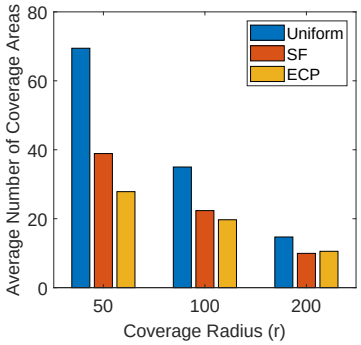


Fig. 9. The average number of coverage areas against different sizes of the coverage area is shown by using density-based clustering. A set of POIs follows a uniform, SF, or ECP distribution. Here, ϵ and λ are set to 1.6 and 30, respectively.

2) Performance Evaluation and Comparison: We measure and compare the performance in terms of the path topology, percentage of covered POIs, and total navigation distances by changing key simulation parameters. The proposed path planning scenarios are also compared by changing the first visit in the path. (i) *Closest + Nearest Neighbor First (CLNF)*: The drone selects the closest located scan point from the base as the first visit. Then the drone repeatedly selects the closest scan point for the next visit to reduce the navigation distance. (ii) *Closest + Random (CLRD):* Like the CLNF, the first visit scan point is the closest located scan point from the base. The next visit scan point in the path is chosen randomly. Here, the CLRD can be used as a performance lower bound for comparison. The rationale behind this first visit comparison is that not all scan points are the same. Depending on the search priority, the first visit scan point may affect the performance significantly. In this paper, our search priority is to quickly find as many survivors as possible, e.g., the golden hour. Thus, the first visit scan point based on the POI density is the same for all the DF, NNF, DD, and even RAND. However, like the shortest path, both CLNF and CLRD visit the closest scan point first from the base.

First, we compare the average number of coverage areas to cover all POIs by changing the size of r in Fig. 9. For comparison purposes, we generate 100 POIs that follow a uniform, SF, or ECP distribution. As the size of r increases, the number of coverage areas decreases for entire distributions. In particular, the size of r affects the uniform distribution most because POIs are uniformly spread out in the network. The number of covered POIs is proportional to the size of r. For non-uniform distributions, SF and ECP, a few coverage areas are enough because most POIs are located under hubs or clusters. Thus, a single coverage area covers more number of POIs. The SF shows a higher number of coverage areas than that of the ECP. This is because the SF often shows few hubs as well as many isolated POIs that require separate coverage areas. For the ECP, however, it shows many smaller-sized clusters compared to hubs, and few isolated POIs are observed. When r is 200 (m), the SF shows a smaller number of coverage areas than that of the ECP. This is because POIs are located relatively closer together in the SF and thus, a wider coverage area can cover more isolated POIs. The ECP shows widely spread-out POIs and their clusters in the network.

Second, we generate a set of POIs that follows the SF and deploy the proposed path planning scenarios to cover them in Fig. 10. The path performance is shown in terms of the percentage of covered POIs in Fig. 11. Two hubs are observed in the network as shown in Subfig. 8(a), where POIs are densely populated in the left-most and upper and the left-most and bottom subareas of the network. The scan point with the highest density is the first visit from the base. In this paper, the first scan point to visit is the same for all the path planning scenarios. In Subfig. 10(a), the DF shows a series of scan points in descending order based on the density of POIs. The first 10 scan points that show higher density than others are visited in sequence. Since the drone visits the area with high density first, the percentage grows rapidly as shown in Fig. 11. However, the DF does

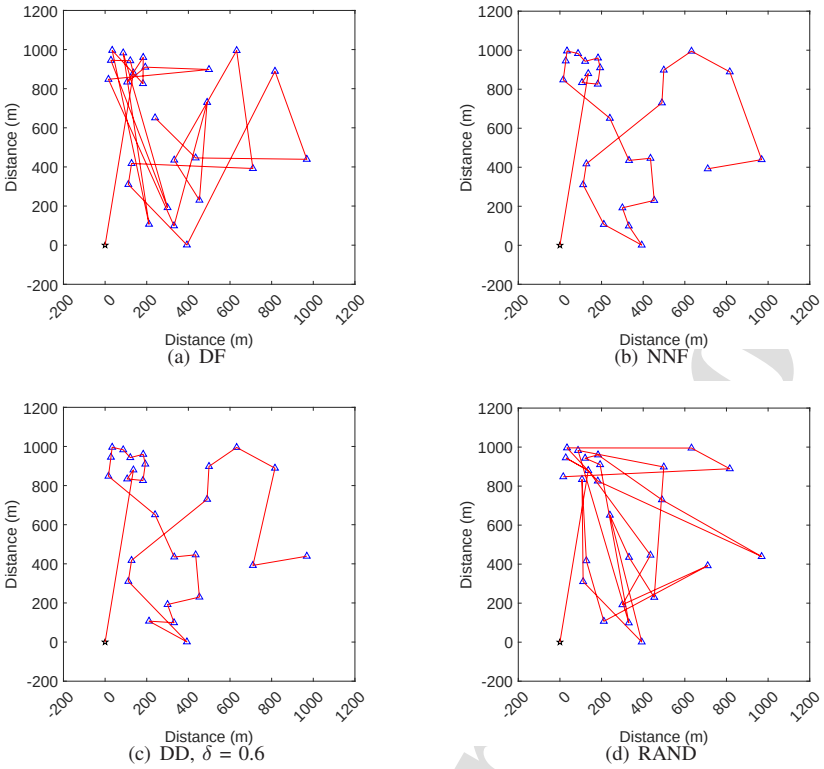


Fig. 10. A set of paths is shown based on the proposed path scenarios. All POIs follow the SF with $\epsilon = 1.6$. Here, a base is marked as a star at the left-most and bottom of the network.

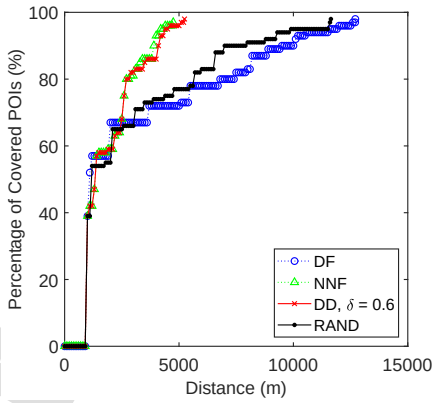


Fig. 11. The comparison of the percentage of covered POIs is shown based on the SF with $\epsilon = 1.6$.

not reduce the navigation distance because the next visit hub is far away. Nevertheless, the DF would be an alternative for a time-sensitive operation that requires covering the desired POIs or high-density POIs quickly. Unlike the prior DF approach, the NNF shows a simple and easy-to-following path without a mixed visit pattern in Subfig. 10(b). This is because the drone selects the scan point for the next visit that is the closest location to its current location. Thus, the NNF shows the shortest navigation distance for the entire path planning scenarios in Fig. 11. In Subfig. 10(c), we balance the impact of density and distance factors, $\delta = 0.6$. Since the distance affects selecting the next visit scan point, the path is similar to that of the NNF but different from that of the DF. More importantly, the DD reduces the navigation distance and covers the POIs quickly in Fig. 11. In Subfig. 10(d), the RAND shows the mixed visits because any scan point can be chosen randomly for the next visit in the network. As expected, the RAND has a longer navigation distance than that of the NNF and DD because of the mixed

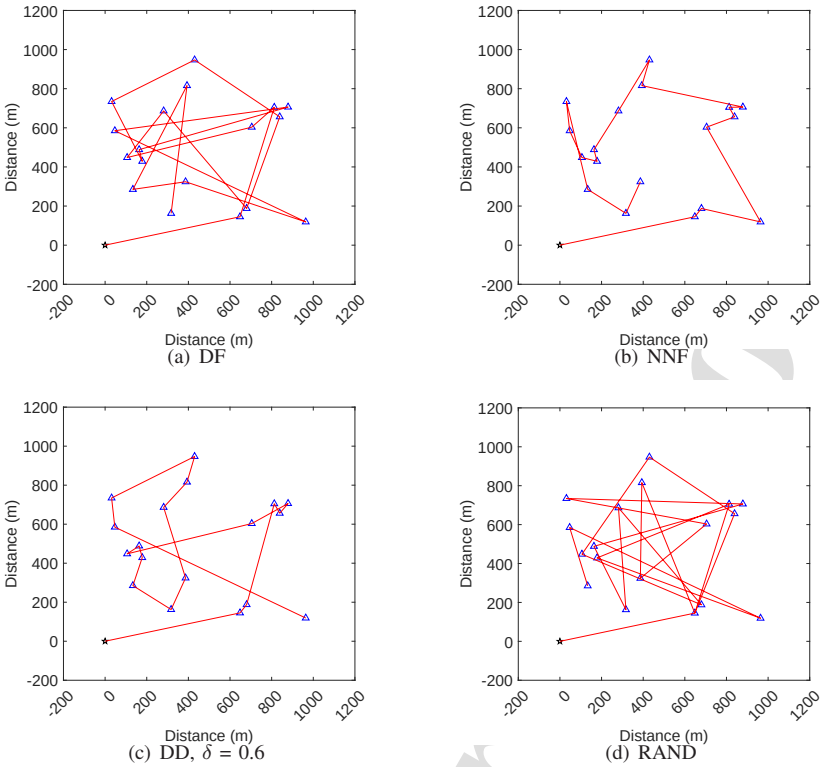


Fig. 12. A set of paths is shown based on the proposed path scenarios. All target points follow the ECP with $\lambda = 30$. Here, a base is marked as a star at the leftmost and bottom of the network.

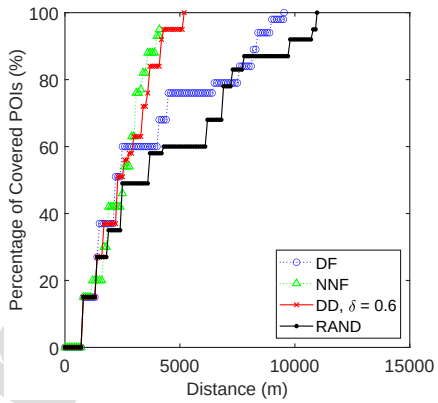


Fig. 13. The comparison of the percentage of covered POIs is shown based on the ECP with $\lambda = 30$.

visits of scan points. The percentage also grows slowly because each scan point in the path is randomly selected in Fig. 11.

Third, we also generate a set of POIs that follows the ECP and show a set of path topologies in Fig. 12. The path performance is also shown in Fig. 13. Compared to the SF, the ECP generates more clusters but the number of POIs located within each cluster is smaller in Subfig. 8(b). The number of coverage areas is also less than that of the SF. Unlike the SF, a single or an isolated POI that is covered by the coverage area is not observed. Thus, the extended coverage area (e.g., $2r$) is not deployed. In Subfig. 12(a), the DF shows a zigzag pattern in the path because the number of small clusters is scattered over the network. The path shows a similar pattern to random visits. The percentage quickly increases at initial but the total navigation distance increases in Fig. 13. The NNF shows a simple path and achieves the shortest navigation path in Subfig. 12(b). This is because, unlike the SF, all the POIs are located together in a small number of clusters. The percentage also increases quickly in Fig.

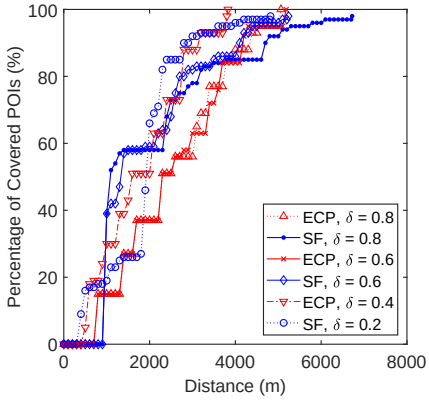


Fig. 14. The percentage changes of covered POIs of the DD are shown with different δ ranging from 0.2 to 0.8. Here, the SF and ECP are set to $\epsilon = 1.6$ and $\lambda = 30$, respectively.

13. In Subfigs. 12(c), we balance the density and distance factors (e.g., $\delta = 0.6$), show a simpler path than that of the DF, and achieve as short a navigation distance as the NNF. In Subfig. 12(d), the RAND shows a mixed visit pattern after the drone arrives at the scan point with the highest density located in the middle and bottom of the network. Then the number of scan points to visit is selected in any random cluster. Thus, the RAND shows the longest navigation distance in Fig. 13.

Fourth, we observe the path performance of the DD with different δ ranging from 0.2 to 0.8 in Fig. 14. In the SF (see Subfig. 8(a)), the percentage increases quickly within the first few visits because most POIs are located in one or two hubs in the network, where multiple hubs are not closely located. However, there are long tails observed for the entire δ. The longest tail is shown when δ is 0.8. This is because a non-negligible number of scan points covers a few POIs. To completely cover the remaining 5 to 10% of POIs, it would still take time for the drone to visit the scan points that are scattered over the network, leading to an increase in the navigation distance. The size of tails reduces with lower δ, 0. 6 and 0.2, implying shorter navigation distance. When δ is 0.2, the percentage reduces significantly compared to that of when δ is 0.8 because the drone visits closely located scan points that do not cover many POIs. In the ECP (see Subfig. 8(b)), the percentage increases linearly because most scan points equally likely cover a similar number of POIs. Unlike the SF, the navigation distance increases when δ becomes higher, 0.6 and 0.8. Since multiple small clusters are scattered in the network, the drone may fly in a zig-zag pattern to visit them, resulting in a longer navigation distance. The higher percentage and shorter navigation distance are achieved when δ is the lowest, 0.2. Unlike the SF, the ECP does not have such a long tail that can increase the navigation distance.

Finally, we measure the impact of the first scan point on the percentage of covered POIs by selecting the closest located scan point from the base in Fig. 15. To clearly see the impact, we deploy a simple density-based clustering (see Subfig 4(a)). Originally we selected the scan point that has the highest POI density as the first point to visit. We compare the performance of the DD with the CLNF and CLRD. Since both CLNF and CLRD do not consider the density of POIs, the percentage grows slowly compared to the DD under the SF in Subfig. 15(a). The drone that follows the CLNF or CLRD needs to fly more distances than that of the DD to cover the same amount of scan points. The CLNF also shows a rapid increase in the percentage because scan points with high density could be visited opportunistically. The SF that creates few and highly dense hubs can affect the percentage significantly. Thus, the CLNF can even show a longer distance than the CLRD to cover about 50% of scan points. In Subfig. 15(b), however, both DD and CLNF show a competitive percentage. The CLNF covers more POIs than that of the DD before the distance of 3000 (m). This is because the ECP creates smaller-sized clusters compared to the hubs. Thus, the density factor may not strongly affect the percentage. The performance of the CLNF depends on the given non-uniform distribution but the DD shows better and steady performance. In reality, we do not know how densely survivors

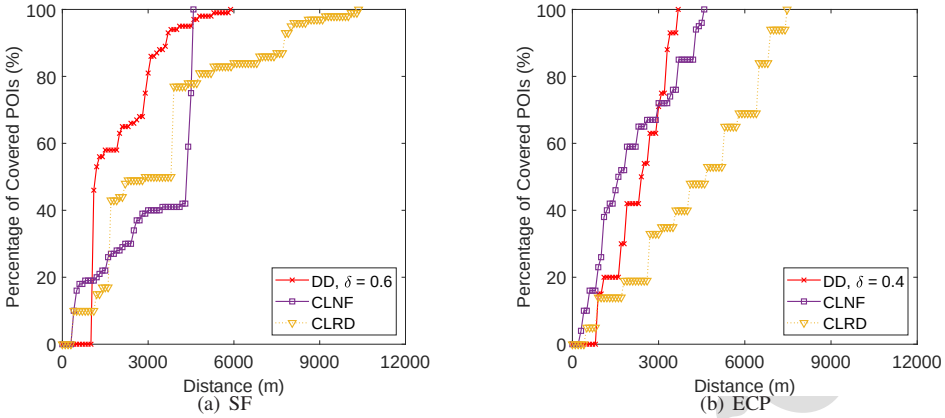


Fig. 15. The impact of the first visit scan point on the percentage of covered POIs is shown. Here, the SF and ECP are set to $\epsilon = 1.6$ and $\lambda = 30$, respectively.

are distributed in an emergency site. It would be judicious to consider both distance and density to quickly cover POIs, i.e., survivors.

V. Discussion

We identify two major issues to see the full potential of the proposed framework and suggest future research directions.

A. Covering Mobile POIs and Infrastructure Changes

In this paper, we implicitly assume static POIs that are stationary or do not move in a short period. We relax this assumption and consider the mobility of POIs, mobile POIs [27]. The locations of mobile POIs are time-varying, such as wildfire, vehicles, soldiers, drones, and so on. As the POIs move, the initial coordinates of coverage areas (i.e., scan points) and their corresponding path will become obsolete shortly. After a drone is launched with the original path, there is no further update on the path. To efficiently cover mobile POIs, the drone can collaborate with a location-based service (LBS) server to update their whereabouts. Whenever the drone arrives at the next scan point, it communicates with the server, updates the coordinates of coverage areas, and revises the original path. In the presence of high mobility, however, these operations would be a burden because the drone is limited by inherent resource constraints in terms of communication, computing, and battery power. Depending on the mobility of mobile POIs, the size of the coverage area can be changed flexibly incorporating mobility management. The coverage area can be extended (or shrunk) when the mobility is high (or low). The coordinates of flexible-sized coverage areas do not have to be the same as the locations of scan points as far as more POIs can be covered.

In addition, we consider dynamic changes in road topologies and accessibilities primarily caused by natural disasters. For example, the recent category 4 hurricane Helene incurred tremendous damage to properties and losses of human life and left more than 200 deaths in the United States (U.S.) Southeast in 2024. Due to the following floodings and landslides, infrastructures including roads, bridges, or waterways were blocked, collapsed, or even buried [28]. The impacted areas are often changed from what they were originally shown on a public map. Thus, rescue crews face difficulties in accessing the areas, resulting in a significant rescue delay. However, the proposed framework can seamlessly conduct a drone-centric search operation because rescue crews can use a pre-downloaded public map or utilize a customized rescue map that is not affected by the changes in the infrastructure.

TABLE II
SUMMARY OF SYMBOLS

Symbol	Meaning
r	A coverage radius
$g_{i,r}$	A set of neighbor POI i located within r
τ_{max}, τ_{min}	A maximum or minimum threshold on a number of neighbor POIs
d	The Euclidean distance between two POIs
ρ_j	A density of scan point s_i
$\|g_j\|$	The number of POIs in coverage area j
$\|n\|$, $\|n'\|$	The total number of POIs located in the entire and target areas, respectively
w	A weight based on the distance and density factors
δ	A system parameter, ranging from 0.2 to 0.8
t_i^d	A distance between s_i and s_{i+1}
L_A, L_B, L_C	A set of sub-areas in the city of Lubbock, Texas
a_j, k_j	A subarea j and its population
ϵ	A cluster exponent
φ_j	A popularity of subarea j
λ, θ	An intensity of parent points and a radius

B. Three-Dimensional Path Planning

The proposed path planning scenarios are implicitly assumed to work in a 2D network. We extend our scenarios by considering three-dimensional (3D) path planning to reflect real-life scenarios and achieve flexible mobility. Rapidly-exploring random tree (RRT) and its numerous variants have been widely deployed to optimize high-dimensional path planning problems [29]. The basic idea is to build a path tree and extend it by including a new random waypoint that does not cause a collision. A large-scale point cloud map and a modified RRT are integrated to assist 3D path planning. A signal to interference & noise ratio (SINR) map [30] is constructed over 3D locations based on a channel gain map and loading factors in a cellular-enabled drone network, where drones communicate with ground base stations during flight. By leveraging the SINR map, the shortest path is pursued to reduce the navigation distance. Different heights of buildings and obstacles that can be extracted from a realistic map are used to determine the path from the base to target POIs. The proposed framework can be updated by deploying 3D-based clustering and path planning techniques.

VI. Concluding Remarks

In this paper, we presented a small-scale prototype of a framework to reduce a setup delay and expedite a drone-centric search operation by automating the required operations. We extracted POIs and metadata from Google Maps by creating and leveraging a customized local database. To efficiently group the POIs, density-based clustering and its variant algorithms are proposed. We also presented three path planning scenarios considering POI density and navigation distance factors and analyzed their performance implication and applicability. We implemented the framework as an Android-based mobile software and conducted extensive simulation experiments for performance evaluation and analysis. We measured and compared the performance in terms of the path topology, percentage of covered POIs, and total navigation distance. The framework can be extended flexibly by replacing an individual operation with a customized operation depending on the drone mission. We envision that the proposed prototype of the framework can play an important role in accelerating drone-centric operations in post-disaster relief.

Appendix

In Table II, we summarize the key symbols used in this paper. The symbols are listed in the order of appearance on the paper.

ACKNOWLEDGMENT

This research was supported in part by the Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (NRF–2021H1D3A2A01080645) and the International Cooperative Research Grant in 2022 from Incheon National University (Incheon, South Korea).

REFERENCES

[1] F. B. Sorbelli, F. Corò, S. K. Das, L. Palazzetti, and C. M. Pinotti, "On the Scheduling of Conflictual Deliveries in a Last-mile Delivery Scenario with Truck-carried Drones," *Pervasive and Mobile Computing*, vol. 87, 2022.
[2] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned Aerial Vehicles (UAVs): Practical Aspects, Applications, Open Challenges, Security Issues, and Future Trends," *Intelligent Service Robotics*, vol. 16, 2023.
[3] *Drones are now being used not only to respond to natural disasters but better predict them*, https://abc11.com/post/2024-atlantic-hurricane-season-drones-are-now-being/15005554/, Jun 2024.
[4] F. Mezghani, P. Kortoci, N. Mitton, and M. D. Francesco, "A Multi-tier Communication Scheme for Drone-assisted Disaster Recovery Scenarios," in *Proc. IEEE PIMRC*, 2019.
[5] A. Albanese, V. Sciancalepore, and X. Costa-Perez, "SARDO: An Automated Search-and-Rescue Drone-based Solution for Victims Localization," *IEEE Trans. on Mobile Computing*, 2021.
[6] J. Campbell, *International Trauma Life Support for Emergency Care Providers (8th Global Edition)*. Pearson, 2018.
[7] K. Namuduri, "Flying Cell Towers to the Rescue," *IEEE Spectrum*, vol. 54, no. 9, 2017.
[8] T. Yang, C. H. Foh, F. Heliot, C. Y. Leow, and P. Chatzimisios, "Self-Organization Drone-Based Unmanned Aerial Vehicles (UAV) Networks," in *Proc. IEEE ICC*, 2019.
[9] H. Asano, H. Okada, C. Naila, and M. Katayama, "A UAV Flight Method for Non-Uniform User Distributions in Aerial Wireless Relay Networks," in *Proc. IEEE CCNC*, 2022.
[10] J. Dong, K. Ota, and M. Dong, "UAV-Based Real-Time Survivor Detection System in Post-Disaster Search and Rescue Operations," *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 2, no. 4, 2021.
[11] E. T. Alotaibi, S. S. Alqefari, and A. A. Koubaa, "LSAR: Multi-UAV Collaboration for Search and Rescue Missions," *IEEE Access*, vol. 7, 2019.
[12] A. H. K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework based on Octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
[13] Y. Lu, Z. Xue, G. Xia, and L. Zhang, "A Survey on Vision-based UAV Navigation," *Geo-spatial Information Science*, vol. 21, no. 1, pp. 21–32, 2018.
[14] Y. Zhang, M. Gao, X. Zhang, P. Yang, Q. Ma, C. Wang, H. He, and X. Hu, "An Automatic Approach to Extracting Geographic Information from Internet," *IEEE Access*, vol. 6, 2018.
[15] H. Xie, D. Li, Y. Wang, and Y. Kawai, "A Graph Neural Network-based Map Tiles Extraction Method Considering POIs Priority Visualization on Web Map Zoom Dimension," *IEEE Access*, vol. 10, 2022.
[16] M. Toccu, G. Psaila, and D. Altomare, "On-line Aggregation of POIs from Google and Facebook," in *Proc. ACM Symposium on Applied Computing*, 2019.
[17] Z. Li, Y. Y. Chiang, S. Tavakkol, B. Shbita, J. H. Uhl, S. Leyk, and C. A. Knoblock, "An Automatic Approach for Generating Rich, Linked Geo-metadata from Historical Map Images," in *Proc. ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining*, 2020.
[18] K. Hosseini, D. C. S. Wilson, K. Beelen, and K. McDonough, "MapReader: A Computer Vision Pipeline for the Semantic Exploration of Maps at Scale," in *Proc. ACM SIGSPATIAL Int'l Workshop on Geospatial Humanities*, 2022.
[19] X. Zhu, C. Bian, Y. Chen, and S. Chen, "A Low Latency Clustering Method for Large-Scale Drone Swarms," *IEEE Access*, vol. 7, 2019.
[20] Y. Huang, J. Xu, M. Shi, and L. Liu, "Time-Efficient Coverage Path Planning for Energy-Constrained UAV," *Wireless Communications and Mobile Computing*, 2022.
[21] D. Kim, L. Xue, D. Li, Y. Zhu, W. Wang, and A. O. Tokuta, "On Theoretical Trajectory Planning of Multiple Drones To Minimize Latency in Search-and-Reconnaissance Operations," *IEEE Trans. on Mobile Computing*, vol. 16, no. 11, 2017.
[22] J. Li, Y. Xiong, J. She, and M. Wu, "A Path Planning Method for Sweep Coverage With Multiple UAVs," *IEEE Internet of Things Journal*, vol. 7, no. 9, 2020.
[23] H. Gedawy, A. Al-Ali, A. Mohamed, A. Erbady, and M. Guizani, "UAVs Smart heuristics for Target Coverage and Path Planning Through Strategic Locations," in *Proc. IEEE IWCMC*, 2021.
[24] *Maps SDK for Android*, https://developers.google.com/maps/documentation/android-sdk.
[25] S. Lim, C. Yu, and C. R. Das, "Realistic Mobility Model for Wireless Networks of Scale-Free Node Connectivity," *International Journal of Mobile Communications (IJMC)*, vol. 8, no. 3, pp. 351–369, 2010.
[26] D. P. Kroese and Z. I. Botev, "Spatial Process Generation," 2013, [Online]. Available: arXiv: 1308.0399.
[27] S. Chinthi-Reddy, S. Lim, G. Choi, J. Chae, and C. Pu, "DarkSky: Privacy-Preserving Target Tracking Strategies Using A Flying Drone," *Vehicular Communications*, vol. 35, 2022.
[28] *Before-and-after images show Helene wiped parts of North Carolina off the map*, https://www.cnn.com/2024/10/03/weather/hurricane-helene-north-carolina-before-after-images-climate/index.html, Oct 2024.
[29] Z. Zheng, T. R. Bewley, and F. Kuester, "Point Cloud-Based Target-Oriented 3D Path Planning for UAVs," in *Proc. Int'l Conf. on Unmanned Aircraft Systems*, 2020.
[30] S. Zhang and R. Zhang, "Radio Map-Based 3D Path Planning for Cellular-Connected UAV," *IEEE Trans. on Wireless Communications*, vol. 20, no. 3, 2021.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: