

Article

An Aerial Robotic Missing-Person Search in Urban Settings—A Probabilistic Approach

Cameron Haigh * , Goldie Nejat  and Beno Benhabib *

Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S 3G8, Canada; nejat@mie.utoronto.ca

* Correspondence: cameron.haigh@mail.utoronto.ca (C.H.); benhabib@mie.utoronto.ca (B.B.)

Abstract: Autonomous robotic teams have been proposed for a variety of lost-person searches in wilderness and urban settings. In the latter scenarios, for missing persons, the application of such teams, however, is more challenging than it would be in the wilderness. This paper, specifically, examines the application of an autonomous team of unmanned aerial vehicles (UAVs) to perform a sparse, mobile-target search in an urban setting. A novel multi-UAV search-trajectory planning method, which relies on the prediction of the missing-person's motion, given a known map of the search environment, is the primary focus. The proposed method incorporates periodic updates of the estimates of where the lost/missing person may be, allowing for intelligent re-coverage of previously searched areas. Additional significant contributions of this work include a behavior-based motion-prediction method for missing persons and a novel non-parametric estimator for iso-probability-based (missing-person-location) curves. Simulated experiments are presented to illustrate the effectiveness of the proposed search-planning method, demonstrating higher rates of missing-person detection and in shorter times compared to other methods.

Keywords: urban search and rescue; multi-UAV robotic search; probabilistic search planning



Citation: Haigh, C.; Nejat, G.; Benhabib, B. An Aerial Robotic Missing-Person Search in Urban Settings—A Probabilistic Approach. *Robotics* **2024**, *13*, 73. <https://doi.org/10.3390/robotics13050073>

Academic Editors: Gerardo Flores, Hector M. Becerra, Juan-Pablo Ramirez-Paredes and Alexandre Brandão

Received: 10 April 2024
Revised: 30 April 2024
Accepted: 6 May 2024
Published: 9 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Autonomous vehicles (aerial and/or ground) have commonly been proposed for urban search and rescue (USAR) [1–9] and wilderness search and rescue (WiSAR) applications [10–18]. They have also been suggested for city searches [19–24], although, with limited scale or autonomy. Namely, most city-search approaches require human guidance or teleoperation of the robot searchers [19,20], while some also search only a small portion of an area relative to what the mobile missing person can actually cover [21–24].

The work presented herein considers the application of an autonomous team of aerial unmanned vehicles (UAVs) for a coordinated search of a missing person in an urban/city environment. The search is characterized by (i) an expanding and sparse search area, which cannot be fully explored with the provided resources during the search time, and (ii) the structure and topography of the urban environment, which guides the lost-person movement. Accordingly, the search method employed must predict the behavior of the lost person to best utilize the available limited resources.

Typical approaches to urban/city search for a missing person are reviewed below in Section 1.1, including autonomous robotic search techniques. Due to similarities between the USAR and WiSAR scenarios, several approaches for the latter are also discussed.

One need also note that in a crowded urban environment, the remote identification of a missing/lost person can be challenging. In order to address this problem, several works examining person reidentification from (remote) images are discussed in Section 1.2.

Lost/missing-person behavior is discussed in Section 1.3, including methods for estimating the missing person's location.

The novelty and contributions of our work are briefly outlined in Section 1.4, prior to detailed discussions in Sections 2–4. Comparison and robustness studies are subsequently presented in Section 5. Concluding remarks are discussed in Section 6.

1.1. Urban Search for a Missing/Lost Person

Typical approaches to finding a lost (or, missing) person in an urban/city environment involve using teams of SAR professionals and volunteers to coordinate and perform a search [25]. These employ the use of a variety of locomotion methods, such as motor vehicles, bicycles, and helicopters, as well as non-human resources such as canines. It has also recently been suggested that other aids such as UAVs could be useful in conducting searches for lost people [26].

UAV-based approaches rely on expert input to guide the UAVs' trajectories, as well as the use of an operator to identify the lost person [19,20]. Most focus on mapping the unknown environments [3,5,22] and are typically limited to small areas. They are more suited for disaster-rescue scenarios than to a missing-person search where the area containing the lost person is significantly larger, crowded, and would include private and public transportation vehicles.

While autonomous UAVs have seen limited use in urban searches, they have been widely proposed for mobile target searches in other environments [10–17,27]. These have included both homogeneous teams of UAVs [17,27,28], as well as heterogeneous teams with different UAVs and UGVs [10,12]. Central to these methods is the generation and use of probabilistic maps of the lost person's potential location, using predicted behavior to produce density maps [10,27–29] and other representations [11,12,15,17] of a lost person's location for guiding the search. Further details on the lost-person behavior used to motivate such probability maps are examined below in Section 1.3.

1.2. Lost-Person Identification

When searching for a lost person in an urban environment, the ability to detect and identify the missing/lost person is paramount to success. The identification of a person from images falls under the general class of person re-identification problems, for which solutions have been proposed using classical computer-vision approaches [30–33] and, more recently, deep-learning approaches [34]. These allow an unseen person to be identified from a text description [35,36]. Other works have examined re-identification in crowds [37] and from moving UAVs [38]. These methods have shown that person re-identification can be achieved in crowded areas and using moving UAVs in addition to or instead of a fixed-camera network.

1.3. Lost-Person Behavior in An Urban/City Environment

City environments are characterized by a high density of structures and well-defined pathways that a lost person can follow. Such environments have a profound impact on lost-person behavior. For example, [39] notes different statistics for how far and where a lost-person would travel in a city as compared to other environments. Furthermore, [25,39] both categorize lost-person behaviors into different movement strategies some of which, such as route following, are of particular interest to city environments where there can be large networks of interconnected pathways offering easy traversal for a lost-person.

Past studies have proposed approaches that simulate lost-person behavior based on historical rescue locations taken by previously rescued lost persons, for guiding the searchers. Several such basic geography-based methods were presented in [40]. These methods predict the potential position of a lost person using such techniques as distance travelled or watersheds. However, they only predict where the lost person might be and do not consider the trajectory they could have taken to get there. Other methods incorporate the different behaviors identified in [39] and grid-based maps of the environment to predict either a general density using diffusion [41,42] or simulate individual lost-person

trajectories [27,43,44]. Thus, it is commonly accepted that methods that can produce potential lost-person trajectories are desirable for planning searches.

1.4. Contributions

The work presented herein considers the application of an autonomous team of UAVs for a coordinated search of a missing person in an urban/city environment. The main novel contribution of the proposed method of searcher trajectory generation is in the intelligent re-coverage of previously searched areas. In scenarios such as an urban search for a missing person, which consider a sparse search with mobile targets that can move back into areas that searchers have previously covered, continual re-coverage of those areas is necessary for high search performance.

The proposed search method uses iso-probability curves, further discussed in Section 2.2, in a dynamic manner to iteratively generate search trajectories, allowing for the intelligent re-coverage of previously searched areas. When re-covering areas, the iso-probability curves are updated to reflect the likelihood that the target could be in those areas and search trajectories are planned accordingly. Furthermore, this re-computation of the iso-probability curves allows for new information to be easily incorporated into the search during its execution.

Our work also includes novelties in the prediction and estimation of the lost person's location. A novel parametric behavior-based method for lost-person trajectory prediction is presented. Furthermore, we propose a novel estimation method for iso-probability curves using kernel-based methods to produce an estimate from predicted mobile target trajectories. This estimation method produces smoother curves than prior histogram-based methods [45] and can be used whenever target (i.e., missing-person) position predictions are available.

2. Problem Statement: Robotic Urban Search for a Lost Person

This paper examines a lost- (or missing-) person search scenario, via autonomous UAVs, in an urban/city environment. The search proceeds over a fixed time horizon from t_{start} to t_{end} , where t_{start} is some time after the lost person has been reported missing. This proposed aerial search is to be conducted parallel to a ground search, which is not modeled in this paper, and the subsequent rescue of the lost person, after the identification, is left to the ground team. The search is assumed to be sparse, namely it is not possible to fully search the area that could be covered by the mobile target (in motion) and guarantee that the target (i.e., lost person) will be found with the resources provided. As such, the objective herein is to maximize the probability that the target will be found, with additional consideration to how quickly they are likely to be found.

2.1. Search Assumptions

Several assumptions about the behavior of the lost person, the search agents, and the planning of the search are made in this paper. Also, it is envisioned that the search method itself is decoupled from the modeling of the lost person; thus, it would still function if different assumptions about the lost person's behavior are made.

2.1.1. The Lost Person

One of the first steps in conducting a lost-person search is to create a (demographic) profile of that person [25], in order to plan an effective search. This paper assumes that such information is available in order to predict the motion of the lost person, in terms of route-traveling, direction-traveling, backtracking, and random-wandering behaviors.

2.1.2. The Searchers

Herein, it is assumed that the UAVs, employed as search agents, can follow arbitrary trajectories at their rated speeds. Additionally, it is assumed that the UAVs have a long

enough flight time to complete their assigned search trajectories, while moving at their rated speeds, and return for retrieval, without needing refueling or recharging.

All UAVs are assumed to be outfitted with cameras and other sensing equipment sufficient to allow for the lost person to be detected, provided an unobstructed view to the lost person exists. This is modeled using a binary-disk detection model which is obstructed by obstacles in the environment.

During the search, all UAVs are capable of accurate global positioning. They are also communicating with a central controller, which provides the UAVs with their search trajectories throughout the search.

2.1.3. Search Planning

A common framework for evaluating the search part of a search and rescue (SAR) operation [46] involves maximizing the probability of success (PoS) of the search. This is measured by combining two probability distributions: the probability of area (PoA) and the probability of detection (PoD). The former describes the likelihood of the search target to be at any given position. The latter, on the other hand, describes how likely the search target is to be found, assuming that it is at the location that is being searched. In this context, the probability of success, PoS, is obtained by combining the PoD at all points along the search trajectory with the PoA at those points, resulting in an overall probability of how likely the search is to be successful.

Optimal search trajectories are ones that maximize the PoS, resulting in the highest likelihood of a successful search, while observing searcher constraints, such as always moving at their rated speed. Using the above description of the PoS, such optimal trajectories will be ones that balance exploiting high-PoA areas with full coverage of the search area. One method of obtaining such paths is by leveraging iso-probability curves [12], which provide a means of planning search trajectories that balance the search effort in proportion to the likelihood of the search objective being in a certain location, while spreading the search effort out across the entire search area. Further information on iso-probability curves is briefly provided below in Section 2.2.

When conducting the search, it is assumed that there is a centralized base location from which the search is being coordinated. It is also assumed that, following standard practice, there is an SAR team operating out of this location and performing a ground search in conjunction with the proposed aerial search. Thus, when the location of the lost person is potentially identified by the aerial search team, a ground team can be dispatched to confirm and to ‘pick up’ the lost person.

2.2. Iso-Probability Curves

First introduced in [17], iso-probability curves provide a description of a lost person’s movement away from an initial location over time and they have been used for several WiSAR multi-robot search planning applications [11,12,17,18,45]. In a set of iso-probability curves, each curve is associated with a quantile; the q^{th} curve describes the radial position of the q^{th} furthest location the lost person may have gone in any given direction. The full set of curves corresponding to quantiles from 0 to 1 provides a complete description of where the lost person might be located. The curves are time varying and centered around the last known position (LKP) of the lost person, growing outward from the LKP as time progresses.

As shown in [11], each q^{th} percentile iso-probability curve has the following polar coordinate description:

$$\left(F^{-1}(q|\theta, t), \theta \right) \quad \forall \theta \in [0, 2\pi], \quad (1)$$

where $F^{-1}(q|\theta, t)$ is the inverse cumulative distribution function for the probability density, $f(r|\theta, t)$, of a lost person being located at some distance, r , along a ray leaving the LKP in direction θ , at time t .

3. Proposed Methodology

In this section, the proposed methodology for the urban-search problem outlined above is described. An overview of the methodology is shown in Figure 1. First, the sub-problem of lost-person motion prediction, *Lost-Person Motion Model Simulation*, in Figure 1, is detailed in Section 3.1. Then, the sub-problem of search planning for the lost person, *Search Initialization and Trajectory Generation*, in Figure 1, is detailed in Section 3.2.

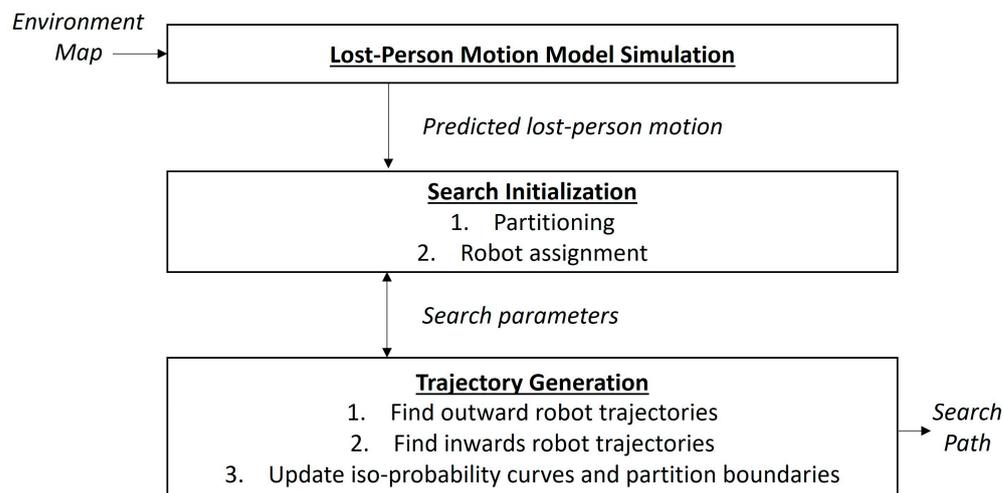


Figure 1. Overview of the proposed methodology.

3.1. Lost-Person Motion Modelling and Prediction in an Urban Environment

Several methods for lost-person behavior modeling have been reported in [27,41–44]. However, they mostly focus on predicting motion in wilderness environments. Several methods adapt the behaviors into grid-based methods for lost-person prediction [39]. However, in order to be applicable to urban environments, due to the high density of ‘obstacles’ as well as nearby roads going in different directions, an approach that is not limited to a simple grid representation of the environment and lost-person trajectory would be desirable. Herein, such a model is presented, first, using a description of the environment, which implements key features for lost-person motion modelling in an urban environment, in Section 3.1.1. The proposed lost-person motion model is detailed, next, in Section 3.1.2. It employs a Monte-Carlo simulation approach to estimate the possible trajectories that the lost person may have taken. The simulated trajectories are then used to construct an estimate of the iso-probability curves corresponding to the missing person’s location by means of a kernel-based iso-probability curve estimation method described in Section 3.1.3.

3.1.1. The Urban Environment

In order to effectively model and predict the motion of a lost person in an urban environment, the relevant features of that environment need to be incorporated into the model for the lost person. This, in turn, leads to a need for a representation of urban environments that can be used for the purpose of lost-person motion prediction. The lost-person behaviors considered, herein, are *route-traveling* and *direction-traveling*, which require both linear features in the environment for the lost person to follow, as well as the modeling of obstacles that can impede motion in a given direction. Figure 2 shows such an environment.

The environment is modeled as a collection of obstacles, primarily buildings, and linear features, such as roads, paths, and trails. It is assumed that the environment is sufficiently flat to not impact the lost person’s behavior; thus, changes in altitude are ignored, considering only a 2-dimensional map. Obstacles are represented as a sequence of points that are connected to form closed polygonal chains. Linear features are represented as a sequence of points that are connected to form open polygonal chains. In order to

model intersections, all linear features are split at locations where they cross each other, and additional breaks are included to provide more decision points when traveling along linear features.

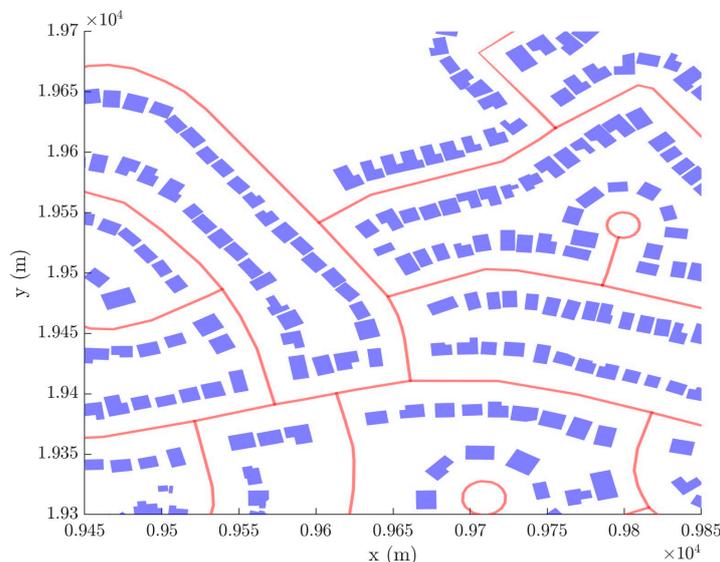


Figure 2. Sample map of an urban environment, with obstacles in *blue* and linear features in *red*.

3.1.2. Urban Lost-Person Behavior Model

In order to perform lost-person motion prediction in an urban environment, the wilderness lost-person model used in [12] was adopted herein and extended by incorporating common behaviors such as random-traveling, route-traveling, direction-traveling, and backtracking. The proposed motion-prediction method is based on the descriptions of these behaviors from [39], as well as the way that several of these behaviors have been modeled in grid-based motion-prediction methods [41,42], incorporating them into the pre-existing model. The generated lost-person trajectories are modeled as a sequence of trajectory segments, in the form of open polygon chains, each generated in an iterative manner by the proposed motion-prediction method. The method employs a parametric approach where, for each generated trajectory segment, the lost person follows one of two locomotion strategies and uses one of two decision-making strategies, with transitions between these strategies being controlled by the model parameters. An overview of the transitions between these strategies is shown in Figure 3. It omits the backtracking behavior since it modifies the other strategies. However, it is not directly used to determine lost-person motion.

The two locomotion approaches are the *direction* and *route* strategies, shown on the right in Figure 3. For the former, the lost person heads in a given direction with perturbations in heading, navigating around obstacles in the environment. For the latter, the lost person follows existing linear features in the environment such as roads and trails. The two decision-making strategies are *random* and *traveling*, shown on the left in Figure 3. When traveling, the lost person tries to keep a consistent heading, and, when moving randomly, the lost person moves in the environment in a less structured manner. Between each generated segment, the model has some probability of switching between the different locomotion and decision-making strategies as well as possibly backtracking and attempting to reverse direction for the next segment.

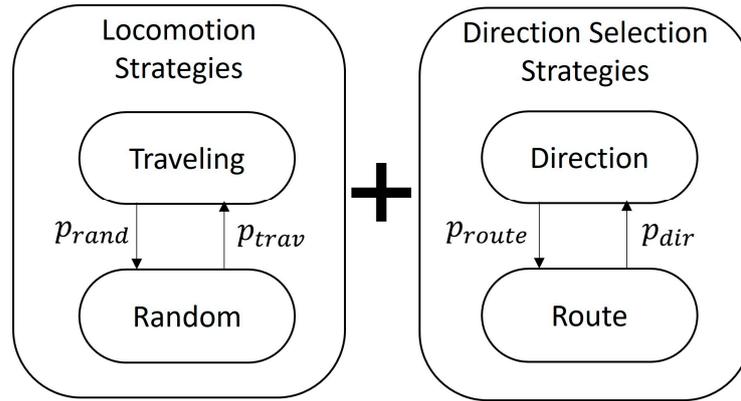


Figure 3. Overview of behaviors used to model lost-person motion in an urban environment.

The method for producing each trajectory segment depends on the currently employed strategies. When under the *direction* and *traveling* strategies, the segment takes the form of a line of varying length, d , that is uniformly sampled to be between the distances d_{min} and d_{max} :

$$d \sim U(d_{min}, d_{max}). \tag{2}$$

The heading of the line, θ , is determined by

$$\theta \sim N(\phi, \sigma^2), \tag{3}$$

where $N(\phi, \sigma^2)$ is a normal distribution from which θ is sampled with a mean of ϕ , the direction that the lost person was heading when this strategy was switched to, and the standard deviation of σ , a model parameter. If the line segment intersects with any obstacles in the environment; then, the lost person goes around the obstacle and continues along the segment. This is achieved by temporarily switching to the *route-locomotion* method and treating the edge of the obstacle as a linear feature to follow. While going around the obstacle, the distance traveled is counted towards the length of the line, decreasing the distance travelled along that line. If the length of the line is reached while avoiding an obstacle, then the step terminates. Since the line segment may intersect linear features in the environment, at each intersection point the model may decide, by means of a Bernoulli trial, to stop the current step and start the next step following that linear feature using the route-locomotion strategy:

$$e_{route} \sim \text{Bernoulli}(p_{route}), \tag{4}$$

where p_{route} is the probability that this occurs, and e_{route} is the event that the model changes behaviors.

Whenever a step is completed using the traveling strategy, there is some probability that the random strategy will be switched to. This is modelled by a Bernoulli trial:

$$e_{rand} \sim \text{Bernoulli}(p_{rand}), \tag{5}$$

where p_{rand} is the probability of this event, e_{rand} , occurring.

There is also a chance that the decision will be made to backtrack at the end of the step, which is modeled by another Bernoulli trial:

$$e_{back} \sim \text{Bernoulli}(p_{back}), \tag{6}$$

where p_{back} is the probability of doing so, and e_{back} is the event that the model changes behaviors.

If a decision to backtrack is made, the current and desired headings are reversed before the next step.

When using the *direction* and *random* strategies, the generated segment is constructed in a similar manner to when using the *direction* and *traveling* strategies, with two distinct changes. One difference is that the heading is determined using the heading of the previous segment instead of a fixed direction:

$$\theta \sim N(\theta, \sigma^2), \quad (7)$$

which leads to trajectories that slowly drift over time. The other difference is that once a step has been completed, instead of a chance to switch to the random strategy, there is a chance that the traveling strategy is used for the next step, modeled by a Bernoulli trial:

$$e_{trav} \sim \text{Bernoulli}(p_{trav}), \quad (8)$$

where p_{trav} is the probability of this event, e_{trav} , occurring.

When using the *route* and *traveling* strategies, the lost person moves along linear features in the environment, trying to keep moving in the same direction when multiple linear features intersect. This produces trajectory segments that follow the polygonal chains of linear features in the environment. When starting a step with this strategy, all linear features within a user-defined distance parameter, d_{route} , are considered, and a random heading is chosen based on the previous heading, in the same manner as Equation (7), and the linear feature which best aligns with the new heading is selected. The trajectory segment starts by connecting the lost person's current location to the closest point on the linear feature and moving along that linear feature in the direction best aligning with the desired heading. The linear feature is followed until the end of the feature is reached, producing a sequence of points for each intermediate segment of the linear feature. When reaching the end of the trajectory segment, there is a chance that the next step will use the *direction* strategy, which is modeled by a Bernoulli trial:

$$e_{dir} \sim \text{Bernoulli}(p_{dir}), \quad (9)$$

where p_{dir} is the probability of that event, e_{dir} , occurring. Additionally, there is the same chance of switching to the random strategy and to backtrack as above, with the same backtracking effect on heading. Normally, the feature that had just been followed is not considered as an option for the next step; however, if the decision is made to backtrack, then it would be considered. If there are no available linear features to follow for the next step, then a switch to the *direction* strategy is made.

When using the *route* and *random* strategies, the next trajectory segment is constructed in a similar manner to the *route* and *traveling* strategies, with two differences. The first is that the desired heading is chosen without regard to the current heading:

$$\theta \sim U(-\pi, \pi). \quad (10)$$

The second is that, instead of having a chance to switch to the random strategy, there is a chance to switch to the *traveling* strategy after a step, using Equation (7).

3.1.3. Kernel-Based Iso-Probability Curve Estimate

Prior work involving the estimation of iso-probability curves has been categorized into two types [11,12,17,45]: (1) parametric estimation, where an underlying distribution is fitted and used to compute iso-probability curves, and (2) non-parametric estimation, which uses histogram-based estimation of the density function to compute the iso-probability curves. Since the distributions produced by the lost-person motion model may not be accurately modeled by a known distribution, a non-parametric estimation method is preferred herein.

Due to the binning of the lost-person information into a polar histogram, the existing methods introduce a tradeoff between a loss of information due to large bins and a roughness from small bins and the stochastic output of the lost-person model. When considering

density estimation, kernel-based methods are frequently preferred over histogram methods for their generally better estimation performance. Additionally, since these do not rely on an underlying grid and contain inherent smoothing through the selection of kernel bandwidth, they allow for smoother iso-probability curves to be produced without a large degradation of the estimation accuracy.

As such, a kernel-based estimation scheme for iso-probability curves is employed herein. It employs the following formulation for the cumulative distribution of a lost person being located at distance r in a given direction θ at time t :

$$F(r|\theta, t) = \frac{1}{nh_\theta} \frac{\sum_{i=1}^n \left(K_\theta \left(\frac{\theta - \theta_i(t)}{h_\theta} \right) \int_0^r K_r \left(\frac{\tilde{r} - r_i(t)}{h_r} \right) d\tilde{r} \right)}{\sum_{i=1}^n \left(\int_0^\infty K_r \left(\frac{\tilde{r} - r_i(t)}{h_r} \right) d\tilde{r} \right)}, \quad (11)$$

which uses a polar representation of the lost-person position trajectories, $(r_i(t), \theta_i(t))$. It takes a product kernel approach, with an individual Epanechnikov kernel applied on each variable in the polar trajectory, K_θ and K_r , with bandwidth parameters, h_θ and h_r , respectively. Similar to the bin size in a histogram, the bandwidths are chosen to provide smoothing and accuracy in the estimator. For the radial coordinate, since the distribution of interest is bounded by $[0, \infty]$, the reflection technique is applied to avoid the distribution artificially decreasing as the lower bound is approached. Furthermore, since iso-probability curves correspond to the cumulative probability in the radial direction, the integral of the kernel is used, and the overall result is normalized. For the angular coordinate, to account for the fact that angles wrap around every 2π , the difference between values is wrapped to the range $(-\pi, \pi)$ before applying the kernel.

As shown in Section 2.2, this CDF estimate can be inverted, $F(q|\theta, t)$, and used to describe the q^{th} percentile iso-probability curve using Equation (1).

3.2. Target Search

The proposed target search method is one where (searcher) robots follow a dynamic iso-probability curve guided search for the lost person (i.e., target). It is executed as a sequence of steps, shown in Figure 4 and Algorithm 1, which are repeated until the lost person is found or some other termination condition is reached.

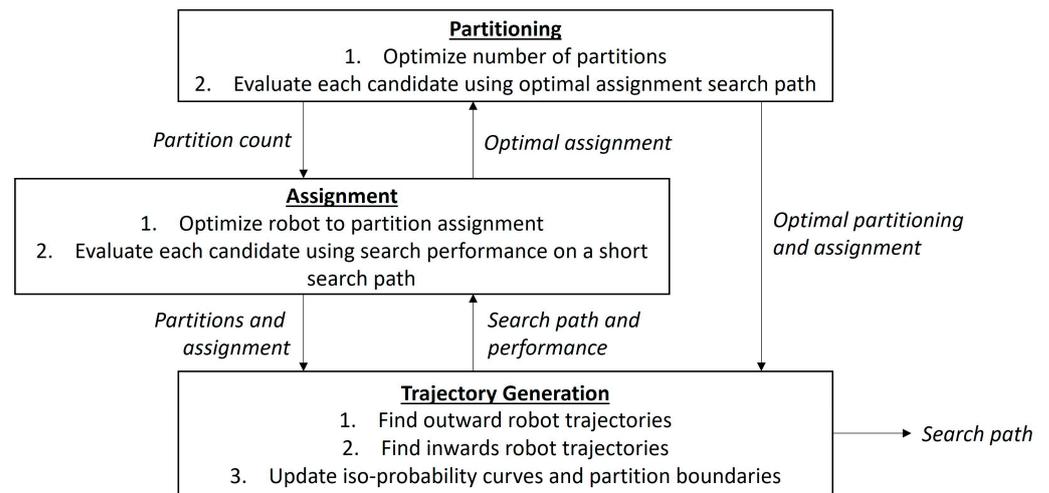


Figure 4. Outline of the proposed search-planning algorithm.

The search algorithm is divided into two sections, the pre-search optimization (Section 3.2.1), shown in Line 1 of Algorithm 1, and the iterative search process (Section 3.2.2), shown in Lines 3–21 of Algorithm 1, respectively. The pre-search optimization utilizes the iterative search process as part of the optimization, as shown in Figure 4, where the inner optimization for robot assignments utilizes a search to evaluate

candidate assignments to iso-probability curve partitions. The search process, trajectory generation in Figure 4, which constructs the search trajectories for all searchers, referred to as *robot_trajectories* in Algorithm 1, is divided into three distinct sections which are repeated until the search is completed. In the first section (Section 3.2.2, Outward Trajectory), Lines 5–10 of Algorithm 1, all robots search moving outward, bounded by the slowest robot. In the second section (Section 3.2.2, Inward Trajectory), Lines 12–17 of Algorithm 1, all robots search by moving back inward, bounded again by the slowest robot. Finally, in the third section (Section 3.2.2, Information Update), Lines 19–20 of Algorithm 1, the information used for planning the search is updated before the next iteration.

Algorithm 1. Proposed search-algorithm pseudocode.

```

1  partitions, assignments ← OptimizePartitionsAssignments(n_robot)
2
3  robot_trajectories ← EmptyTrajectory(n_robot)
4  while not CheckTermination(robot_trajectories)
5      slowest_trajectory, slowest_robot ← ShortestTrajectoryOutward()
6      append slowest_trajectory to robot_trajectories[slowest_robot]
7      for each robot that is not slowest_robot
8          trajectory ← SpiralOutward(robot)
9          append trajectory to robot_trajectories[robot]
10     end for
11
12     slowest_trajectory, slowest_robot ← FixedTrajectoryInward()
13     append slowest_trajectory to robot_trajectories[slowest_robot]
14     for each robot that is not slowest_robot
15         trajectory ← SpiralInward(robot)
16         append trajectory to robot_trajectories[robot]
17     end for
18
19     UpdateIsoProbabilityCurves()
20     partitions.bounds ← CompPartitionBounds()
21 end while

```

In order to coordinate the search paths of all robots involved, the iso-probability curves are divided into different partitions that are searched independently by non-overlapping subset of robots, the partitioning block in Figure 4. Our proposed search method divides the iso-probability curves into several contiguous and non-overlapping partitions, which span the 0% to 100% percentiles. This allows the search method to deploy robots to search the regions where they will provide the most benefit to the overall performance of the search.

In our search method, trajectory generation relies on an adapted approach for following iso-probability curves from [12], where modifications have been made to allow for trajectories that spiral inward in addition to the originally produced trajectories that spiral outward. The process for determining the path of Robot *i* in polar coordinates around the last-known position (LKP) of the lost person, when using this adapted method, henceforth referred to as the curve-following method, is described below.

The curve-following method relates the cumulative angular distance travelled by the robot to the iso-probability percentile that is being searched at a given time. The relationship is described as follows:

$$q_i(t|c, q_s, t_s) = q_s + c \int_{t_s}^t \theta_i(\tilde{t}) d\tilde{t}, \quad (12)$$

where q_s is the starting iso-probability curve quantile for the search trajectory, t_s is the starting time for the search trajectory, θ_i is the angular position of the i^{th} robot, and c is a user-provided curve progression rate that determines how much search effort is spent on

each iso-probability curve percentile. For $c > 0$, the resulting trajectories spiral outward, while for $c < 0$, they spiral inward.

When following a specific set of iso-probability curves, the above approach allows for the polar coordinates of the trajectory to be obtained by evaluating the curves at the specific percentile given by Equation (12) and combining it with the angular position:

$$\left(F^{-1}(q_i(t|c, q_s, t_s)) | \theta_i(t) \right) t \geq t_s. \quad (13)$$

If the searcher robot is always moving at its rated speed, v_r , then the angular position and the corresponding radial position can be determined using an iterative approach. This is an additional constraint that can be described by

$$\frac{d}{dt} F^{-1}(q_i(t|c, q_s, t_s)) + F^{-1}(q_i(t|c, q_s, t_s)) \frac{d}{dt} \theta_i(t) = v_r. \quad (14)$$

3.2.1. Search Initialization

The proposed search method relies on the selection of a number of parameters in addition to those provided by the constraints on the search problem. These correspond to the number and positions of the iso-probability curve partitions used for the search, as well as how the search agents (i.e., robots) should be allocated among those partitions for optimal search performance. Thus, the search initialization process involves solving an (initial) optimization problem to determine the parameters.

The process of determining the optimal number and placement of partitions as well as the allocation of robots to those partitions can be formulated as a mixed integer optimization problem. The proposed optimization method decouples the optimization of the partition boundaries leaving an integer programming problem to be solved for the partition count and robot assignments. When solving the integer program, candidate solutions are evaluated on a fixed time horizon search, and the optimal parameters can be chosen based on some metrics, (e.g., the median time to find a target, the overall probability of finding the target, etc.). The details involved in these optimization steps as well as sensible initializations for quick convergence are outlined below.

Partition Selection

Partition selection, partitioning in Figure 4, describes the process by which the proposed optimization method selects the number of partitions to use and the locations of the boundaries between them. This can be further divided into two nested problems. The first is determining the optimal number of partitions that should be selected for the search, while the second involves determining the placement of the boundaries between contiguous partitions for a given partition count. Herein, each problem is addressed independently, with the number of partitions being part of the larger optimization process outlined above, and partition boundaries being optimized independently from this optimization, considering only the number of partitions and not how robots are allocated between them.

Partition selection, for use in the search, involves two main steps: determining the number of partitions that should be used, as well as determining the locations of the boundaries between contiguous partitions.

For the second problem, a partitioning strategy, adopted from [11,12], is employed. This strategy balances the sizes of each partition by sizing them such that a single robot could search each partition, spending a similar amount of search effort on each intermediate percentile in the partition, within some fixed amount of time. Formally, given some number of partitions, n_{part} , the boundaries of the partitions, $\{q_{lb,i}, q_{ub,i}\}_{i=1}^{n_{part}}$, are optimized with the following objective and constraints:

$$\min \sum_{i=1}^{n_{part}} |c_i - \bar{c}|, \quad (15)$$

$$c_i = \underset{c}{\operatorname{argmin}} \left| q_i(t_{opt}|c, q_{lb,i}, t_{start}) - q_{ub,i} \right| \quad \forall i = 1, \dots, n_{part}, \quad \text{and} \quad (16)$$

$$q_{lb,0} = 0, \quad q_{ub,n_{part}} = 1. \quad (17)$$

As outlined above, the selection of the number of partitions is achieved via an optimization process that includes determining the optimal number robots assigned to the partitions. The selected number of partitions must be in the range $[1, n_{robot}]$, where n_{robot} is the number of robots available for the search. This ensures that there will always be a sufficient number of robots to have a minimum of one robot searching each partition. For large numbers of robots and partitions, this optimization can become computationally intensive. As such, a heuristic approach is proposed herein to provide a logical initial state for the optimization. This approach is adapted from [11], where the number of partitions was set as n_{robot}/n_c , and n_c was a value extracted from the environment of the search. The examined urban environments are much more complex than the environments examined in [11]; hence, the same information from the environment cannot be used. Thus, the proposed heuristic approach for determining an initial value for the optimization of the number of partitions used for the urban search is defined by n_{robot}/b , where b is a user-provided parameter. Based on our empirical studies, it was noted that a value of two for b works well for fast convergence.

Partition Assignment

Given a set number, n_{part} , of contiguous partitions, dividing a set of iso-probability curves, and number of robots, n_{robot} , a process which assigns at least one robot to each partition and assigns each robot to a partition is required. This process, assignment in Figure 4, is carried out as part of the larger integer programming problem outlined above. Thus, the partition assignment sub-problem can be described by

$$\underset{\mathbf{a}}{\operatorname{argmin}} s(\mathbf{a}), \quad (18)$$

$$\sum_{i=1}^{n_{part}} a_i = n_{robot}, \quad \text{and} \quad (19)$$

$$a_i \geq 1, \quad (20)$$

where \mathbf{a} is a vector of length n_{part} . Above, each element represents the number of searchers assigned to the corresponding partition, and $s(\mathbf{a})$ is the performance metric of interest evaluated using a short time horizon search with the provided partitions and robot assignments. For example, when the *median time to find the target* is used as a metric, it is a minimization, as shown above; however, if the *probability of finding the target* is used as the metric, then it is a maximization.

Similarly, to ensure fast convergence in the optimization process, a good initial value for the number of partitions is required. Based on our empirical studies, it was noted that, with the method for determining partition boundaries described previously, the configuration that places all excess robots in the lowest partition consistently yields the best results in terms of median time to find the lost person and probability of finding the lost person. As such the proposed heuristic for initializing the optimization process for partition assignment was set in our work as $a_1 = n_{robot} - n_{part} + 1$ and $a_i = 1, i \neq 1$.

3.2.2. Search Execution

The proposed search execution approach comprises three steps, shown in Figure 5, which are repeated until some termination condition is reached. These are a trajectory moving outward, *Outwards Trajectory* in Figure 5, taking each searcher from the lower bound of its partition to the higher bound; a trajectory moving inward, *Inwards Trajectory* in Figure 5, taking each searcher from the upper bound of its partition to the lower bound;

and an update step, where changes are made to the information guiding the search based on new information and the overall progress of the search, *Information Update* in Figure 5. Additional details on each step are provided in the following sub-sections.

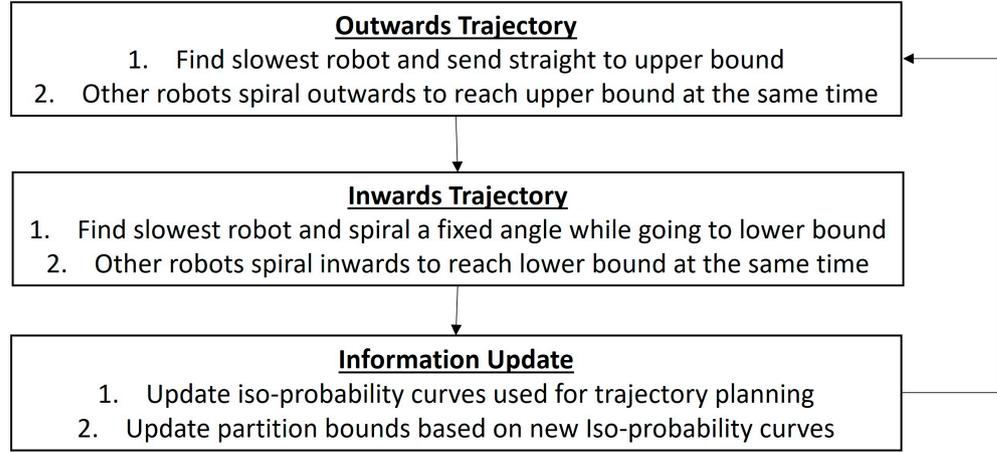


Figure 5. Overview of steps in the search-trajectory generation process.

Outward Trajectory

In the first part of the search execution, each robot must travel from the lower percentile bound of its partition toward the higher percentile bound, hereafter referred to as the robot's lower and upper bounds, respectively, which are calculated following the optimization procedure outlined in Section 3.2.1, Partition Selection. In the proposed search method, this outward trajectory is one such that all robots reach their upper bound at the same point in time, namely,

$$q_i(t_{out}) = q_{ub,i} \quad i = 1, 2, \dots, n_{robot}, \quad (21)$$

where t_{out} is the end time for all outward trajectories.

An outward trajectory is formulated such that the minimum amount of time is taken for the search to complete the outward portion while ensuring that the slowest robot can still reach its upper bound. Namely,

$$t_{out} = \max t_{out,i} \quad i = 1, 2, \dots, n_{robot}, \quad (22)$$

where $t_{out,i}$ is the time at which Robot i arrives at its upper bound, when following the shortest trajectory between its lower and upper bound.

Following the above approach, the outward trajectory for the slowest robot is considered as the shortest-time trajectory. For the remaining robots, their trajectories are slowed down by

$$c_i = \operatorname{argmin}_c \left| q_i(t_{out}|c, q_{lb,i}, t_{start}) - q_{ub,i} \right| \quad \forall i = 1, 2, \dots, n_{robot}; i \neq i_{slowest}, \quad (23)$$

where the trajectory for each robot is determined by the curve-following rate, c_i , which denotes how much search effort the robot expends on each intermediate iso-probability curve in its partition.

After all robots have completed moving along their outward trajectories, which corresponds to time t_{out} , they immediately continue to search moving on an inward trajectory.

Inward Trajectory

The inward-trajectory determination follows a similar formulation to the outward-trajectory generation, going from the upper to lower percentile bound of a robot's partition. However, the slowest robot, instead of going straight towards its destination, travels some cumulative angular distance ϕ while traveling toward its lower bound. As such, the curve

progression rate of the corresponding trajectory can be computed as $c = \frac{\phi}{q_{ub,i} - q_{lb,i}}$, which gives a trajectory that will traverse the desired angular distance when moving from the upper bound to lower bound of the robot's partition. Thus, the time that any given robot takes to travel this path is given by

$$t_{in,i} = \operatorname{argmin}_t \left| q \left(t \left| \frac{\phi}{q_{ub,i} - q_{lb,i}} \right. \right) - q_{lb,i} \right| \quad i = 1, 2, \dots, n_{robot}. \quad (24)$$

The time that bounds the length of this portion of the search is the time taken by the slowest robot to complete the path outlined above. Namely, the time at which this slowest robot completes its trajectory is given by

$$t_{in} = \max t_{in,i} \quad i = 1, 2, \dots, n_{robot}. \quad (25)$$

The slowest robot takes the trajectory that corresponds to the time t_{in} , while the remaining robots follow iso-probability curves inward, such that they reach the lower bound of their partition at the same time that the slowest robot completes its trajectory. This is achieved by the following equation, describing the computation of the curve-following rate for all remaining robots:

$$c_i = \operatorname{argmin}_c \left| q_{i(t_{in}|c, q_{ub,i}, t_{start})} - q_{lb,i} \right| \quad \forall i = 1, 2, \dots, n_{robot}; i \neq i_{slowest}. \quad (26)$$

The trajectories taken by these remaining robots are simply the iso-probability curve-following trajectories, starting at the upper bound of the robot's partition, which follow the curve-following rates computed for each robot.

Once the robots have completed their inward trajectories, they immediately start traveling back outward. However, these new trajectories are now determined using updated information which is outlined below.

Information Update

During search execution, new information may become available concerning the missing person. This could include data influencing their probable behavior, which could come from investigations and interviews, typically conducted at the beginning of a missing person search, tips from civilians, or unconfirmed sightings from external sources that influence the current belief of where the lost person may be located. Additionally, as the search is executed it would be beneficial to focus on regions that contain possible trajectories that have not previously been encountered during the search, which requires adjusting which information is used to plan the next iteration of the search. The information used for the search is updated between each iteration of the search. However, if significant new information, as determined by the user of the search method, is acquired during the search, that information could be used immediately and the current iteration ended early to leverage that information as soon as possible.

In the proposed search method, it is assumed that information is only updated based on the areas that have been searched during previous portions of the search. This is achieved by updating the iso-probability curve estimation by excluding any of the simulated trajectories which would have been found during prior iterations of the search. Namely, only possible positions that would not have been found are considered in the kernel-based estimation process.

When new information is incorporated into the search, it is likely that the locations of the different iso-probability curves change. Since the employed partitioning method is tied to the locations of those curves, they need to be recomputed after the iso-probability curves are updated. After the updating of the iso-probability curves, and the re-computation of the partition boundaries, it is likely that the current positions of the searches will no longer lie on the iso-probability curve percentiles that correspond to the lower boundaries of their

partitions. This scenario can be accounted for by changing the lower bound used for the outward search step, by replacing all $q_{lb,i}$ with $F(r_i(t_{start}) | \theta_i(t_{start}), t_{start})$.

4. Simulated Experiments—Example Results

In this section, two illustrative examples are provided, capturing the details of the full process of the proposed approach: Example 1 provides a comprehensive step-by-step description of the proposed search method, detailed in Section 3.2, for a simple scenario, followed by Example 2, which details the results of a more complex scenario.

4.1. Example 1

4.1.1. Environment

For this example, a representative urban environment was used. It is shown in Figure 6, with obstacles in blue and linear features in red. It constitutes an urban environment $20 \times 20 \text{ km}^2$ in size, which contains a high-density downtown area, a medium-density suburb areas, and some sparse areas further from the downtown core.

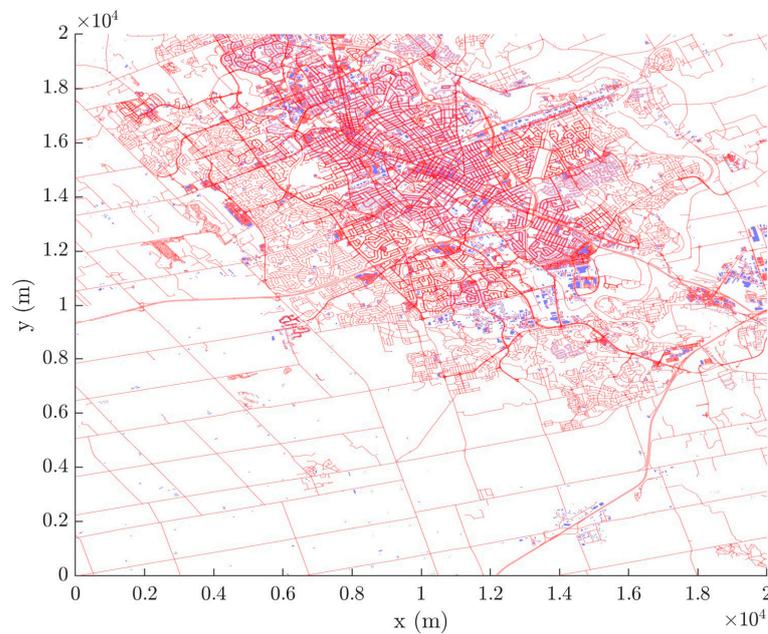


Figure 6. Urban environment used for Examples 1 and 2.

4.1.2. Lost-Person Motion Prediction

Parameters for the lost-person motion model were obtained via a simple gradient-based stochastic optimization technique to match the model output to data for people with dementia noted in [39]. Additional parameter sets were determined for different demographics to validate the lost-person motion model. A common random-numbers technique was used to reduce the variance when computing gradients. The lost person’s speed for each simulated trajectory was sampled from a normal distribution using the mean and standard deviation obtained from [47], as shown in Table 1.

Table 1. Parameters for the lost-person motion model used in Examples 1 and 2.

σ (rad)	p_{rand}	p_{trav}	p_{back}	p_{dir}
0.518	1	0.276	0	0.938
p_{route}	μ_v (m/s)	σ_v (m/s)	d_{route} (m)	
0.312	1.21	0.0815	10	

For the scenario considered, a total of 50,000 trajectories were simulated for the lost person. All target trajectories originated at the LKP of the lost person, at $(x, y) = (10, 181.8 \text{ m}, 10, 068.0 \text{ m})$, with a uniformly sampled initial heading. Each trajectory contained a potential path taken by the lost person for a 12 h period after they went missing. Figure 7 shows the first two hours of a representative set of 200 of these trajectories. In total, 20,000 of these trajectories were used to plan searcher trajectories, including the optimization of the search parameters, while the remaining 30,000 were used to evaluate the proposed search method.

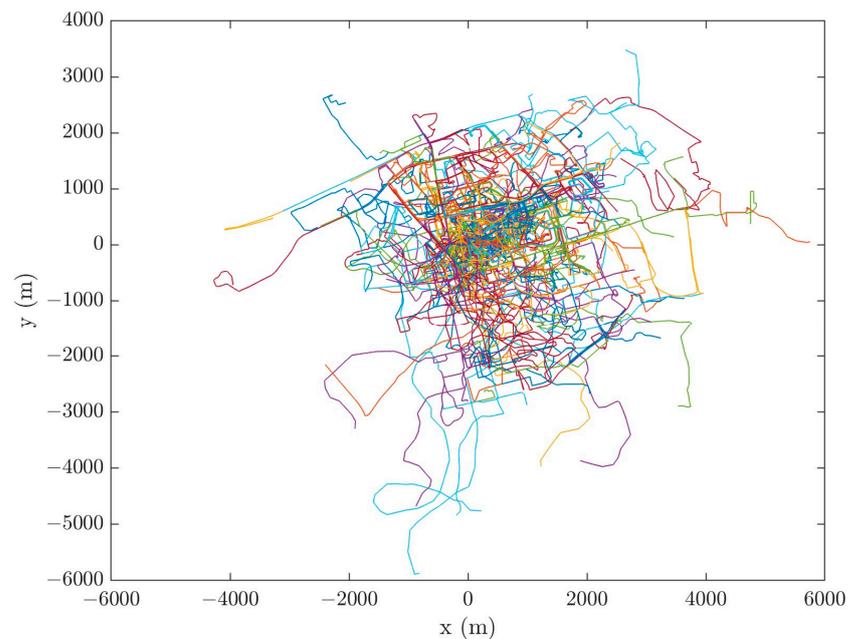


Figure 7. 200 sample lost-person trajectories over a 2 h period.

4.1.3. Search

The search for the missing person was carried out with five identical UAVs ($n_{robot} = 5$). The homogenous UAVs were deemed to be capable of following arbitrary paths at a speed of up to 30 m/s and detecting a person within a distance of 20 m provided that there is a line of sight not obstructed by obstacles in the environment. The UAVs were also assumed to be flying at a sufficient height to clear all obstacles. During the search, all times were measured from the time when the lost person was assumed to have gone missing and was located at the LKP ($t = 0 \text{ s}$). All UAVs were assumed to have arrived at the scene, at their respective initial deployment locations, 40 min ($t = 2400 \text{ s}$) after the lost person was reported to have gone missing and performed a search that lasted for two hours (from $t = 2400 \text{ s}$ to $t = 9600 \text{ s}$).

Search Initialization

In order to perform the search, the first step is to determine the partitioning of the iso-probability curves and the assignment of robots to those partitions. This is determined using the integer programming problem and partition boundary selection method outlined in Section 3.2.1. Solving this problem for five robots using iso-probability curves obtained using the 20,000 lost-person trajectories for search planning and the estimation method outlined in Section 3.1.3 yields the number of partitions and robot assignments used for the search. During the optimization process each candidate configuration was evaluated using a 30 min search and selecting for the minimum median time to find the lost person. This resulted in three partitions being selected with the robot assignments and boundaries shown in Table 2. Figure 8 shows a visualization of this starting configuration displaying

the iso-probability curves corresponding to the boundaries between partitions and the initial deployment locations of the robots (UAVs).

Table 2. Partitioning and robot assignment used in Example 1.

Partition #	1	2	3
Lower Bound (%)	0	54.8	81.2
Upper Bound (%)	54.8	81.2	100
Number of Robots	3	1	1

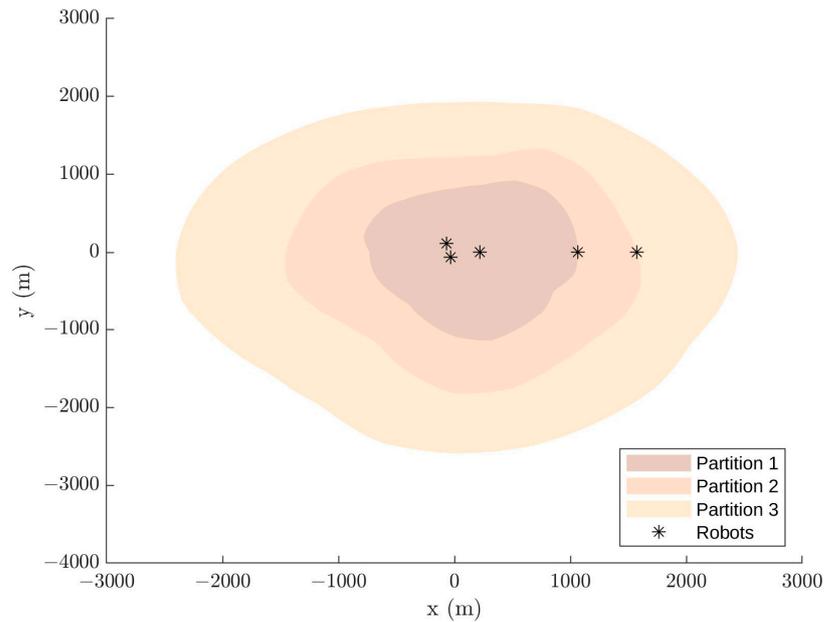


Figure 8. Partitions and initial robot deployment for Example 1.

Search Execution

The search trajectories were generated using the iterative strategy outlined in Section 3.2.2, comprising an outward, inward, and update step for each iteration. The outward trajectories for each partition during the first step of the search are shown in Figure 9. Robot 5 in Partition 3 was the slowest robot traveling in a straight line and limiting the time of the outward step. Figure 10 shows the inward trajectories for each partition during the first step of the search. Once more, Robot 5 in Partition 3 was the slowest robot traveling $\phi = 6.6$ radians around the LKP as described in Section 3.2.2, Inward Trajectory.

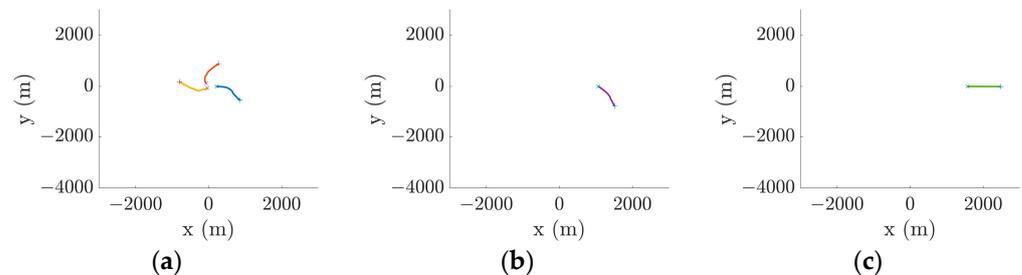


Figure 9. Robot search trajectories for the outward step during the first iteration of the search: (a) Robots 1 to 3 in Partition 1; (b) Robot 4 in Partition 2; and (c) Robot 5 in Partition 3. The markers x and + indicate the trajectory start and end, respectively.

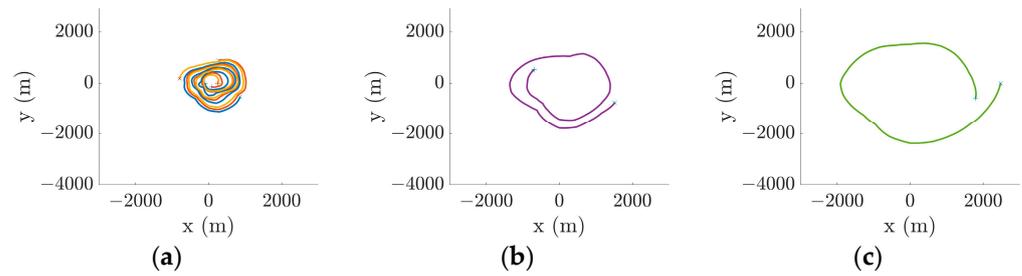


Figure 10. Robot search trajectories for the inward step during the first iteration of the search: (a) Robots 1 to 3 in Partition 1; (b) Robot 4 in Partition 2; and (c) Robot 5 in Partition 3. The markers x and + indicate the trajectory start and end, respectively.

In order to illustrate the information update carried out between iterations, Figure 10 provides a visualization of the changes in the iso-probability curves. Figure 11a shows the iso-probability curves used for planning the first iteration at $t = 2857$ s, when the first iteration is completed. Figure 11b shows the simulated lost-person locations that were used to build those iso-probability curves, highlighting those that would have been found during the first iteration of the search. Figure 11c shows the updated iso-probability curves at $t = 2857$ s, which is used for the second iteration of search planning and considers only the lost-person locations that would not have been found during the first iteration. The majority of the locations that would have been found are located inside the initial 50% iso-probability curve, in part due to more UAVs searching the first partition, which leads to those lower percentile curves being larger in the updated iso-probability curves.

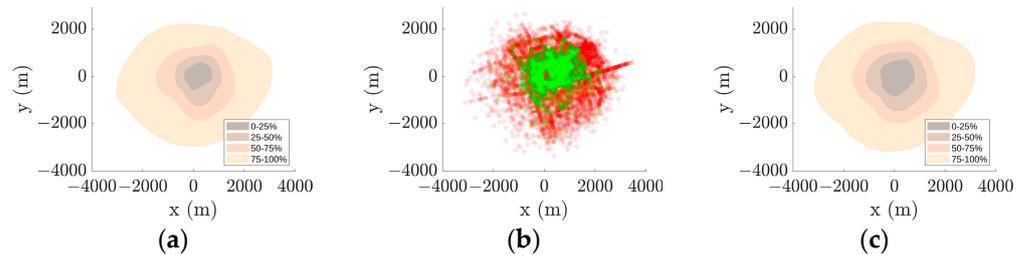


Figure 11. Iso-probability curves and predicted lost-person locations at $t = 2857$ s: (a) Iso-probability curves used for the first iteration of search; (b) Predicted lost-person locations (replications that would have been found during the first iteration are green; all others are red); (c) Iso-probability curves used for the second iteration of search.

In order to evaluate the performance of the proposed search method, it was utilized to search for the 30,000 target trajectories that were not used for searcher trajectory planning. This was performed over the same 2 h search window ($t = 2400$ s to $t = 9600$ s) for which the search was planned. The evaluation determined that 76% (of the 30,000 simulated targets) were found using the proposed search method with a median find-time of 888 s after the start of the search (at $t = 2400$ s). Due to the search being sparse, one cannot expect a 100% find rate.

4.2. Example 2

In this example, a search is performed with a larger number of, though slower, identical UAVs/robots: $n_{robot} = 15$, maximum (rated) speed of up to 10 m/s, and detection distance of 20 m. The environment and the lost person are the same as in Example 1.

Table 3 shows the partitions and robot assignments at the start of the search ($t = 2400$ s). Figure 12 shows the initial robot deployments and iso-probability curve partitions.

Table 3. Partitioning and robot assignment used in Example 2.

Partition #	1	2	3	4	5	6	7
Lower Bound (%)	0	34	53	65	76	85	93
Upper Bound (%)	34	53	65	76	85	93	100
Number of Robots	9	1	1	1	1	1	1

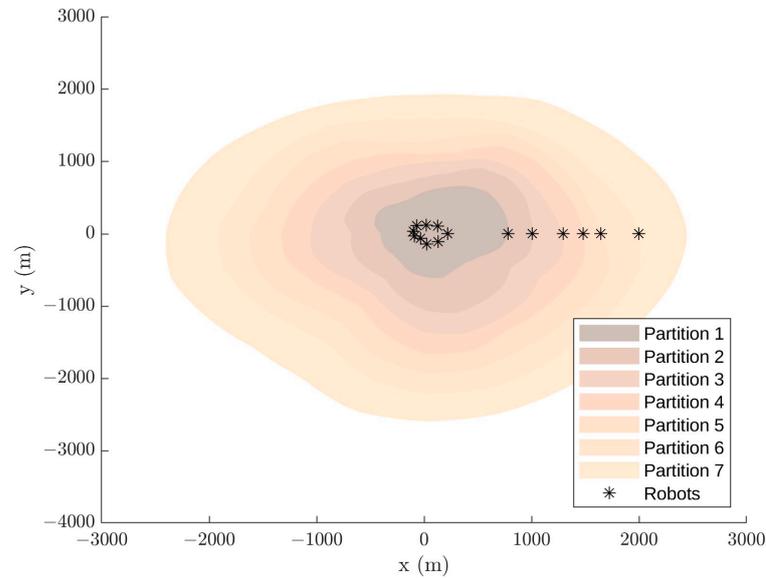


Figure 12. Partitions and initial robot deployment.

The first iteration of the search is visualized in Figure 13: Figure 13a shows the outward moving search trajectories from $t = 2400$ s to $t = 2464$ s; and Figure 13b shows the inward moving search trajectories from $t = 2464$ s to $t = 4120$ s.

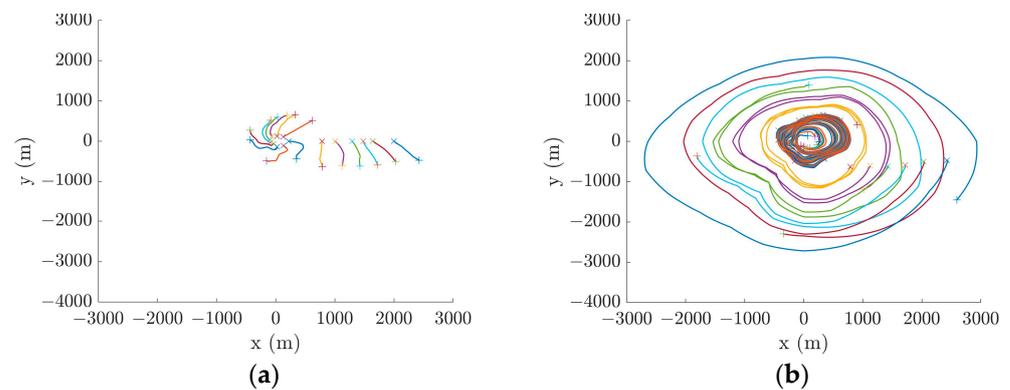


Figure 13. Robot search trajectories during the first iteration of the search: (a) During the outward step; (b) During the inward step. The markers x and + indicate the trajectory start and end, respectively.

Using the 30,000 target trajectories that were not used for search planning, the performance of the full two-hour search ($t = 2400$ s to $t = 9600$ s) was evaluated. The evaluation determined that 78% (of the 30,000 simulated targets) were found using the proposed search method with a median find-time of 871 s after the start of the search (at $t = 2400$ s).

5. Comparison and Robustness Studies

In this section, further results of simulated experiments are presented for (1) a robustness study to determine the effect of estimation errors in assumed lost-person behavior modelling on the effectiveness of a planned search (Section 5.1), and (2) comparing the proposed search method to alternative search methods (Section 5.2), respectively.

5.1. Performance for Different Search Team and Robot Characteristics

The performance of the proposed search method is influenced by the number and characteristics of the searcher UAVs. This section examines the performance of the method, in the same search scenario as in Section 4.1, with different numbers of robots moving at different speeds. Simulations were performed for UAV numbers of 5, 10, and 15, for speeds of 10, 20, and 30 m/s. The results are summarized in Tables 4 and 5.

Table 4. *Targets Found (%)* after two hours of searching.

# of Robots	Robot Speed (m/s)		
	10	20	30
5	51	70	76
10	69	83	90
15	78	90	93

Table 5. *Median Find Time (s)* after two hours of searching.

# of Robots	Robot Speed (m/s)		
	10	20	30
5	1488	1323	888
10	1183	684	515
15	871	603	458

As one can note, the results improve both for larger numbers of robots and for faster robots. However, one can state that deficiencies in searcher speed can be accounted for by providing a larger number of searchers and vice versa, allowing for the search method to be implemented with a larger collection of lower-cost and slower robots or to use a small number of higher-speed robots for comparable results.

5.2. Robustness to Lost-Person Model Inaccuracies

The proposed search method relies on information generated by a lost-person simulation that requires fitting several parameters based on the predicted behavior of the lost person. During the search, one of the first steps is information gathering, so if the search is to be deployed as soon as possible, then it may be deployed with inaccurate assumptions about the lost person. Further investigation can serve to mitigate these inaccuracies; however, it comes at the cost of a later start to the search and will likely still have some inaccuracies in the lost-person motion model. As such, one cannot expect that the simulated lost-person trajectories would correspond exactly to the behavior exhibited by the lost person in question. Thus, a robustness study of the proposed search method was conducted to determine the impact of this difference in *assumed* (A) and *real* lost-person behavior on search outcomes.

In the simulated experiments, the search path was planned using a predicted lost-person behavior, labeled as (A), whose parameters were outlined in Section 4.1.2, whereas the actual search was evaluated/performed on modified/different *real* lost-person behaviors. The modified behaviors were generated by changing the parameters of the assumed (A) model, resulting in different behavior and lost-person distributions. Six different sets of parameters were generated: (A---), (A--), (A-), (A+), (A++), and (A+++), respectively, and the parameters for these models are given in Appendix A. A *minus* sign indicates a slower target, whereas a *plus* sign indicates a faster target, respectively. The slower targets have smaller iso-probability curves than the set for (A), whereas the faster targets have larger iso-probability curves, respectively. Figure 14 shows the iso-probability curves for all targets.

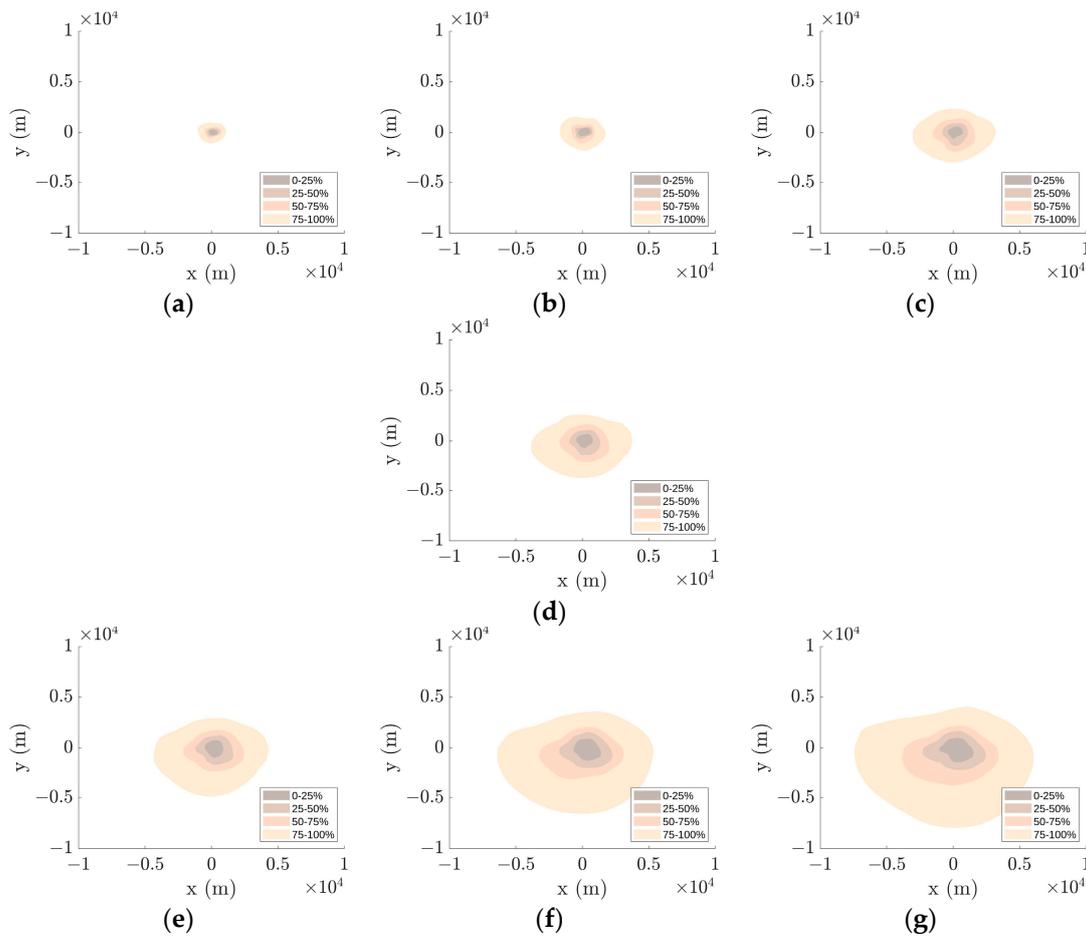


Figure 14. Iso-probability curves at $t = 4000$ s. Curves shown for lost-person models (a) A---, (b) A--, (c) A-, (d) A, (e) A+, (f) A++, and (g) A+++.

For all sets of parameters simulations, 30,000 lost-person trajectories were run for search planning using the parameters for the original target model (A). Thereafter, the searches for 20,000 trajectories were executed, with the same number and specifications of robots used in Section 4.1, using the *real* target models (A---), (A--), (A-), (A+), (A++), and (A+++), respectively. The results of these evaluations are shown in Table 6 and illustrate the robustness of the method.

Table 6. Robustness results after two hours of searching.

Evaluation Set	Effective Radius (m)	Targets Found (%)	Median Find Time (s)
A---	1027	86	444
A--	1666	89	421
A-	2789	82	779
A	3362	76	888
A+	3941	71	937
A++	5264	59	970
A+++	6051	55	994

For ease of comparison, the first column in the table shows the effective radius of each set as the mean distance to the 100% iso-probability curve at $t = 4000$ s. As can be noted in Columns 2 and 3, for faster targets, one can observe a decline in search performance, which is expected as the searched distribution gets further away from the planned distribution; and, for slower targets, one can observe an increase in search performance, which is

expected as the proposed method favors a large number of searchers for the lower iso-probability partitions, causing it to be searched near exhaustively.

Overall, the results clearly indicate that the proposed search method is quite robust to deviations in lost-person behavior modeling for targets that are faster in reality than what is assumed, naturally, up to certain limits (i.e., $< A+$, or even $< A++$). For targets that are slower in reality than what is assumed (i.e., $A-$, $A--$, $A---$), the method is totally robust.

5.3. Comparison to Alternative Methods

In order to evaluate the effectiveness of the proposed search method, comparative simulations were conducted for two alternative search methods: a coverage-based search method [48], typical for searches when target movement is not incorporated into the search, and an exhaustive search approach [49], which ensures the 100% coverage of an area while accounting for a target's potential movement in any direction during the search.

All methods were simulated using the search environment, lost-person behavior simulations, robot numbers, and specifications from the example in Section 4.1. The searches were simulated starting at $t = 2400$ s for the same 2 h search window ($t = 2400$ s to $t = 9600$ s). Additional examples are also included for comparison, which use different lost-person behavior distributions but are otherwise planned and evaluated in the same manner, with the same 20,000 trajectories to 30,000 trajectories split in their respective distributions for planning and evaluation.

The coverage-based search method uses the spiral coverage method presented in [48] adapted to our search scenario by evenly dividing up the required spiral path into equal length segments for each robot. The spacing in the coverage spiral is optimized such that at the end of the 2 h search window the radius matches the maximum distance that a lost person could have traveled based on the simulated trajectories for planning. All searchers start the search at the beginning of their respective spiral trajectory segments and travel along them during the search. Figure 15a shows the overall coverage spiral and the trajectory segments for each searcher.

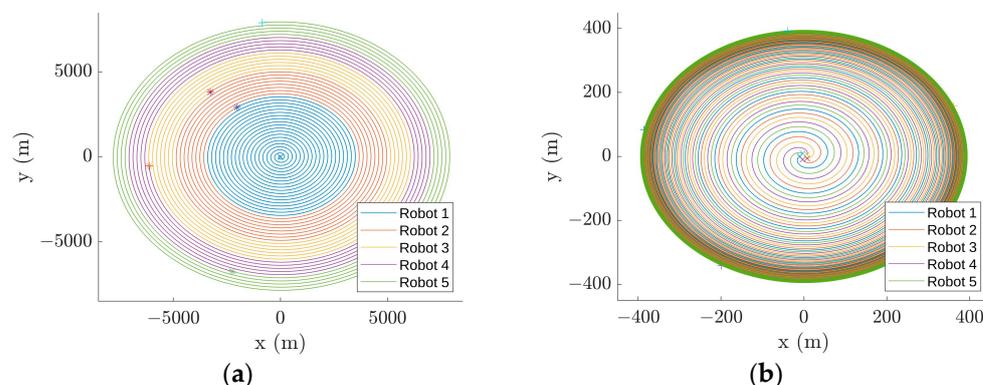


Figure 15. Robot search trajectories for alternative methods: (a) the coverage-based method and (b) the exhaustive search method. The markers x and $+$ indicate the trajectory start and end, respectively.

The exhaustive search method uses the approach presented in [49] which ensures full coverage of an area by a team of cooperative UAVs. The area that can be covered by the method is determined by the number and capabilities of the provided UAVs, which results in the outwards spiraling search trajectories for each searcher as shown in Figure 15b. As can be seen, the method starts with all searchers spaced around the same position, in this case the LKP, and spirals outwards with progressively tighter spirals towards some maximum radius.

The search trajectories for the alternative methods were evaluated in the same manner as for the proposed search method, yielding the (averaged) results shown in Table 7 for three example scenarios. While the first alternative method covers the entire area that could contain the lost person, it does not account for their potential movement, resulting

in the lowest probability of finding the lost person and the longest median time to do so. The second approach performs better with a higher probability of finding the target and lower median find-time than the exhaustive method. This improvement is due to the full coverage near the LKP, which has the highest likelihood of containing the lost person and ensuring that regardless of how the lost person moves within this area they will be found. The proposed method combines both the overall coverage of the area containing the lost person and the re-coverage of areas to emulate full coverage over a larger area. This results in the highest probability of finding the lost person and the lowest median time to do so out of all of the examined methods.

Table 7. Results after two hours of searching averaged over three example scenarios.

<i>Method</i>	<i>Targets Found (%)</i>	<i>Median Find Time (s)</i>	<i>Find Time IQR (s)</i>
Proposed (ours)	84	509	1317
Exhaustive	53	854	1967
Uniform-coverage	27	2230	2817

6. Conclusions

In this work, a novel multi-UAV cooperative search-planning method is presented. This method utilizes a parametric-based simulation, which considers the behavior of a given demographic and the layout of a city, to predict lost-person trajectories. From these trajectories, a novel kernel-based estimator is used to produce smooth iso-probability curve estimates that are suitable for robot-trajectory planning. Search trajectories are planned using an iterative planner that updates the estimates of the lost-person's potential location based on past search performance, and intelligently re-searches areas where the lost person may have re-entered a searched area.

Simulated experiments were presented to verify the effectiveness of the proposed method. Comparative simulations showed the proposed method outperforming several alternative methods in the task of finding a missing person in an urban environment, demonstrating a higher probability of finding the target while having a lower median time to do so. Additionally, robustness experiments showed that the proposed method performs well even with reasonable amounts of error in the predicted lost-person behavior.

Future work will consider the incorporation and cooperation with an autonomous vehicular ground team or the use of heterogeneous UAV search teams.

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

Funding: This research was funded by the Natural Sciences and Engineering Research Council of Canada.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

This Appendix provides the parameters used for the lost-person motion models in Section 5.1. They are presented in Tables A1–A6.

Table A1. Parameters for the lost-person motion model A---.

σ (rad)	p_{rand}	p_{trav}	p_{back}	p_{dir}
0.932	1	0.0551	0	0.938
p_{route}	μ_v (m/s)	σ_v (m/s)	d_{route} (m)	
0.312	0.242	0.0815	10	

Table A2. Parameters for the lost-person motion model A--.

σ (rad)	p_{rand}	p_{trav}	p_{back}	p_{dir}
0.829	1	0.110	0	0.938
p_{route}	μ_v (m/s)	σ_v (m/s)	d_{route} (m)	
0.312	0.484	0.0815	10	

Table A3. Parameters for the lost-person motion model A-.

σ (rad)	p_{rand}	p_{trav}	p_{back}	p_{dir}
0.621	1	0.221	0	0.938
p_{route}	μ_v (m/s)	σ_v (m/s)	d_{route} (m)	
0.312	0.968	0.0815	10	

Table A4. Parameters for the lost-person motion model A+.

σ (rad)	p_{rand}	p_{trav}	p_{back}	p_{dir}
0.414	0.800	0.331	0	0.938
p_{route}	μ_v (m/s)	σ_v (m/s)	d_{route} (m)	
0.312	1.45	0.0815	10	

Table A5. Parameters for the lost-person motion model A++.

σ (rad)	p_{rand}	p_{trav}	p_{back}	p_{dir}
0.207	0.400	0.441	0	0.938
p_{route}	μ_v (m/s)	σ_v (m/s)	d_{route} (m)	
0.312	1.94	0.0815	10	

Table A6. Parameters for the lost-person motion model A+++.

σ (rad)	p_{rand}	p_{trav}	p_{back}	p_{dir}
0.104	0.200	0.496	0	0.938
p_{route}	μ_v (m/s)	σ_v (m/s)	d_{route} (m)	
0.312	2.18	0.0815	10	

References

1. Manuel, M.P.; Faied, M.; Krishnan, M.; Paulik, M. Robot Platooning Strategy for Search and Rescue Operations. *Intell. Serv. Robot.* **2022**, *15*, 57–68. [[CrossRef](#)]
2. Blitch, J.G. Artificial Intelligence Technologies for Robot Assisted Urban Search and Rescue. *Expert Syst. Appl.* **1996**, *11*, 109–124. [[CrossRef](#)]
3. García, R.M.; de la Iglesia, D.H.; de Paz, J.F.; Leithardt, V.R.Q.; Villarrubia, G. Urban Search and Rescue with Anti-Pheromone Robot Swarm Architecture. In Proceedings of the 2021 Telecoms Conference (Conf^{TELE}), Leiria, Portugal, 11–12 February 2021; pp. 1–6.

4. Chen, X.; Zhang, H.; Lu, H.; Xiao, J.; Qiu, Q.; Li, Y. Robust SLAM System Based on Monocular Vision and LiDAR for Robotic Urban Search and Rescue. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 41–47. [[CrossRef](#)]
5. Chatziparaschis, D.; Lagoudakis, M.G.; Partsinevelos, P. Aerial and Ground Robot Collaboration for Autonomous Mapping in Search and Rescue Missions. *Drones* **2020**, *4*, 79. [[CrossRef](#)]
6. Hong, A.; Igharoro, O.; Liu, Y.; Niroui, F.; Nejat, G.; Benhabib, B. Investigating Human-Robot Teams for Learning-Based Semi-Autonomous Control in Urban Search and Rescue Environments. *J. Intell. Robot. Syst. Theory Appl.* **2019**, *94*, 669–686. [[CrossRef](#)]
7. Chen, J.; Li, S.; Liu, D.; Li, X. AiRobSim: Simulating a Multisensor Aerial Robot for Urban Search and Rescue Operation and Training. *Sensors* **2020**, *20*, 5223. [[CrossRef](#)] [[PubMed](#)]
8. Arnold, R.D.; Yamaguchi, H.; Tanaka, T. Search and Rescue with Autonomous Flying Robots through Behavior-Based Cooperative Intelligence. *J. Int. Humanit. Action* **2018**, *3*, 18. [[CrossRef](#)]
9. Baxter, J.L.; Burke, E.K.; Garibaldi, J.M.; Norman, M. Multi-Robot Search and Rescue: A Potential Field Based Approach. *Stud. Comput. Intell.* **2007**, *76*, 9–16.
10. Furukawa, T.; Bourgault, F.; Lavis, B.; Durrant-Whyte, H.F. Recursive Bayesian Search-and-Tracking Using Coordinated UAVs for Lost Targets. *Proc. IEEE Int. Conf. Robot. Autom.* **2006**, *2006*, 2521–2526. [[CrossRef](#)]
11. Ku, S.Y.; Nejat, G.; Benhabib, B. Wilderness Search for Lost Persons Using a Multimodal Aerial-Terrestrial Robot Team. *Robotics* **2022**, *11*, 64. [[CrossRef](#)]
12. Kashino, Z.; Nejat, G.; Benhabib, B. Aerial Wilderness Search and Rescue with Ground Support. *J. Intell. Robot. Syst.* **2020**, *99*, 147–163.
13. Rodríguez, M.; Al-Kaff, A.; Madridano, Á.; Martín, D.; de la Escalera, A. Wilderness Search and Rescue with Heterogeneous Multi-Robot Systems. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 110–116.
14. Talha, M.; Hussein, A.; Hossny, M. Autonomous UAV Navigation in Wilderness Search-and-Rescue Operations Using Deep Reinforcement Learning. In Proceedings of the AI 2022: Advances in Artificial Intelligence; Aziz, H., Corrêa, D., French, T., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 733–746.
15. Peake, A.; McCalmon, J.; Zhang, Y.; Raiford, B.; Alqahtani, S. Wilderness Search and Rescue Missions Using Deep Reinforcement Learning. In Proceedings of the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Abu Dhabi, United Arab Emirates, 4–6 November 2020; pp. 102–107.
16. Schedl, D.C.; Kurmi, I.; Bimber, O. An Autonomous Drone for Search and Rescue in Forests Using Airborne Optical Sectioning. *Sci. Robot.* **2021**, *6*, eabg1188. [[CrossRef](#)] [[PubMed](#)]
17. Macwan, A.; Nejat, G.; Benhabib, B. Optimal Deployment of Robotic Teams for Autonomous Wilderness Search and Rescue. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 4544–4549.
18. Vilela, J.; Kashino, Z.; Ly, R.; Nejat, G.; Benhabib, B. A Dynamic Approach to Sensor Network Deployment for Mobile-Target Detection in Unstructured, Expanding Search Areas. *IEEE Sens. J.* **2016**, *16*, 4405–4417. [[CrossRef](#)]
19. Hanna, D.; Ferworn, A.; Lukaczyn, M.; Abhari, A.; Lum, J. Using Unmanned Aerial Vehicles (UAVs) in Locating Wandering Patients with Dementia. In Proceedings of the 2018 IEEE/ION Position, Location and Navigation Symposium, PLANS 2018, Monterey, CA, USA, 5 June 2018; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2018; pp. 809–815.
20. Hanna, D.; Ferworn, A. A UAV-Based Algorithm to Assist Ground SAR Teams in Finding Lost Persons Living with Dementia. *2020 IEEEION Position Locat. Navig. Symp. PLANS* **2020**, *2020*, 27–35. [[CrossRef](#)]
21. Nagrare, S.R.; Chopra, O.; Jana, S.; Ghose, D. Decentralized Path Planning Approach for Crowd Surveillance Using Drones. *2021 Int. Conf. Unmanned Aircr. Syst. ICUAS* **2021**, *2021*, 1020–1028. [[CrossRef](#)]
22. Reardon, C.; Fink, J. Air-Ground Robot Team Surveillance of Complex 3D Environments. In Proceedings of the 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Lausanne, Switzerland, 23–27 October 2016; pp. 320–327.
23. Semsch, E.; Jakob, M.; Pavlicek, D.; Pechoucek, M. Autonomous UAV Surveillance in Complex Urban Environments. In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Venice, Italy, 15–18 September 2009; Volume 2, pp. 82–85.
24. Zhang, M.; Wang, H.; Wu, J. Multi-UAVs Target Tracking in Urban Environment Based on Distributed Model Predictive Control and Levy Flight-Salp Swarm Algorithm. In Proceedings of the 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), Xiamen, China, 10–12 August 2018; pp. 1–6.
25. Young, C.S.; Wehring, J. *Urban Search: Managing Missing Person Searches in the Urban Environment*; dbS Productions LLC: Charlottesville, VA, USA, 2007.
26. Young, C.S. The Search Intelligence Process. *J. Search Rescue* **2020**, *4*, 136–164. [[CrossRef](#)]
27. Heintzman, L.; Hashimoto, A.; Abaid, N.; Williams, R.K. Anticipatory Planning and Dynamic Lost Person Models for Human-Robot Search and Rescue. *Proc. IEEE Int. Conf. Robot. Autom.* **2021**, *2021*, 8252–8258. [[CrossRef](#)]

28. Alanezi, M.A.; Bouchekara, H.R.E.H.; Apalara, T.A.-A.; Shahriar, M.S.; Sha'aban, Y.A.; Javaid, M.S.; Khodja, M.A. Dynamic Target Search Using Multi-UAVs Based on Motion-Encoded Genetic Algorithm with Multiple Parents. *IEEE Access* **2022**, *10*, 77922–77939. [[CrossRef](#)]
29. Waharte, S.; Symington, A.; Trigoni, N. Probabilistic Search with Agile UAVs. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, Alaska, 3–8 May 2010; pp. 2840–2845.
30. Farenzena, M.; Bazzani, L.; Perina, A.; Murino, V.; Cristani, M. Person Re-Identification by Symmetry-Driven Accumulation of Local Features. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2360–2367. [[CrossRef](#)]
31. Prosser, B.; Zheng, W.S.; Gong, S.; Xiang, T. Person Re-Identification by Support Vector Ranking. In Proceedings of the British Machine Vision Conference, BMVC 2010, Wales, UK, 30 August—2 September 2010. [[CrossRef](#)]
32. Ma, B.; Su, Y.; Jurie, F. Covariance Descriptor Based on Bio-Inspired Features for Person Re-Identification and Face Verification. *Image Vis. Comput.* **2014**, *32*, 379–390. [[CrossRef](#)]
33. Ma, B.; Su, Y.; Jurie, F. Local Descriptors Encoded by Fisher Vectors for Person Re-Identification. *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinf.* **2012**, *7583*, 413–422.
34. Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; Hoi, S.C.H. Deep Learning for Person Re-Identification: A Survey and Outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 2872–2893. [[CrossRef](#)]
35. Ye, M.; Liang, C.; Wang, Z.; Leng, Q.; Chen, J.; Liu, J. Specific Person Retrieval via Incomplete Text Description. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, 22 June 2015; pp. 547–550.
36. Han, X.; He, S.; Zhang, L.; Xiang, T. Text-Based Person Search with Limited Data. In Proceedings of the 32nd British Machine Vision Conference, Online, 22–25 November 2021.
37. Behera, N.K.S.; Sa, P.K.; Muhammad, K.; Bakshi, S. Large-Scale Person Re-Identification for Crowd Monitoring in Emergency. *IEEE Trans. Autom. Sci. Eng.* **2023**, 1–9. [[CrossRef](#)]
38. Nguyen, H.; Nguyen, K.; Sridharan, S.; Fookes, C. Aerial-Ground Person Re-ID. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Brisbane, Australia, 10–14 July 2023; pp. 2585–2590. [[CrossRef](#)]
39. Koester, R.J. *Lost Person Behavior: A Search and Rescue Guide on Where to Look for Land, Air and Water*; dbis Productions LLC: Charlottesville, VA, USA, 2008.
40. Koester, R. Determining Probabilistic Spatial Patterns of Lost Persons and Their Detection Characteristics in Land Search & Rescue. Ph.D. Thesis, University of Portsmouth, Portsmouth, UK, 2018.
41. Hashimoto, A.; Abaid, N. An Agent-Based Model of Lost Person Dynamics for Enabling Wilderness Search and Rescue. In Proceedings of the ASME 2019 Dynamic Systems and Control (DSC) Conference, Park City, Utah, 8–11 October 2019; Volume 2. [[CrossRef](#)]
42. Hashimoto, A.; Heintzman, L.; Koester, R.; Abaid, N. An Agent-Based Model Reveals Lost Person Behavior Based on Data from Wilderness Search and Rescue. *Sci. Rep.* **2022**, *12*, 5873. [[CrossRef](#)]
43. Mohibullah, W.; Julier, S.J. Developing an Agent Model of a Missing Person in the Wilderness. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Washington, DC, USA, 13–16 October 2013; pp. 4462–4469. [[CrossRef](#)]
44. Dacey, K.; Whitsed, R.; Gonzalez, P. Using an Agent-Based Model to Identify High Probability Search Areas for Search and Rescue. *Aust. J. Emerg. Manag.* **2022**, *37*, 88–102. [[CrossRef](#)]
45. Kashino, Z.; Kim, J.Y.; Nejat, G.; Benhabib, B. Spatiotemporal Adaptive Optimization of a Static-Sensor Network via a Non-Parametric Estimation of Target Location Likelihood. *IEEE Sens. J.* **2017**, *17*, 1479–1492. [[CrossRef](#)]
46. Syrotuck, W.G. *An Introduction to Land Search Probabilities and Calculations*; Arner Publications: Rome, NY, USA, 1975.
47. Schimpl, M.; Moore, C.; Lederer, C.; Neuhaus, A.; Sambrook, J.; Danesh, J.; Ouwehand, W.; Daumer, M. Association between Walking Speed and Age in Healthy, Free-Living Individuals Using Mobile Accelerometry—A Cross-Sectional Study. *PLoS ONE* **2011**, *6*, e23299. [[CrossRef](#)]
48. Zhao, Q.; Zhang, L.; Han, Y.; Fan, C. Polishing Path Generation for Physical Uniform Coverage of the Aspheric Surface Based on the Archimedes Spiral in Bonnet Polishing. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2019**, *233*, 095440541983865. [[CrossRef](#)]
49. Brown, D.; Sun, L. Exhaustive Mobile Target Search and Non-Intrusive Reconnaissance Using Cooperative Unmanned Aerial Vehicles. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1425–1431.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.