

Real-time On-board Path Planning for UAS-based Wildfire Monitoring

S M Towhidul Islam, Xiaolin Hu, Georgia State University

Abstract— The use of Unmanned Aircraft System (UAS) in wildfire monitoring is increasing significantly due to its benefits in terms of flexibility, lower cost, and safety. To effectively monitor and collect data about a wildfire, UAS path planning is essential. Many of the applications adopt a centralized approach that uses a computer on the ground to provide path planning information for individual UASs. The applicability and robustness of such centralized approaches are limited due to the challenging wildfire environments. In addition, very few of the existing applications focus on the most actively spreading region of the fire which needs special monitoring attention. In this paper, we present an on-board path planning algorithm for UAS-based wildfire monitoring. The proposed algorithm uses real-time data collected by the UAS to direct the UAS to monitor the wildfire with special attention to the most active regions of the fire. Simulation results have been presented to show the effectiveness of the algorithm in different simulated wildfire scenarios.

I. INTRODUCTION

Thousands of wildfires occur every year which destroys millions of acres of land and properties around the world. The impact of this destruction is immense, and it is a great threat to human and wildlife, vegetation, and the ecosystem. To support wildfire management, collecting real-time data about the spreading fire is important. Different wildfire monitoring and data collection strategies and technologies have been used by researchers to support firefighting as efficiently as possible, including the use of satellite systems, manned aircraft systems, ground sensors etc. Each of these technologies has its own limitations in terms of application. These limitations include but are not limited to – lower effectiveness, lower safety, higher cost, and limited adaptiveness to the dynamically spreading wildfires. For example, helicopter missions are relatively unsafe and costly, the number of ground sensors required to sense a wildfire could be huge, and real-time deployment is difficult and time-consuming. Also, they are statically placed over different locations, it is very hard to make the sensor systems adaptive to the size and spread of the fire.

An effective, safer, and comparatively cost-efficient option for wildfire monitoring is Unmanned Aircraft Systems (UASs). Due to the recent developments of highly capable UASs, it is considered a very suitable option for collecting important information from a wildfire. Advanced UASs can fly over a wide range of altitudes at desired speed for hours. They can be equipped with different types of sensors and cameras to collect valuable information about the wildfire. Furthermore, intelligent and adaptive algorithms can be applied to the UAS so that it can monitor the fire with minimal external supervision and communication. Altogether, UASs

hold huge potential for collecting real-time wildfire data and supporting wildfire management.

Wildfire spread is a dynamic process influenced by a variety of factors. The three important factors that influence wildfire behavior are fuel, terrain, and weather. The fuel refers to the composite of variables that describe the vegetation the fire is spreading through. The terrain is described by slope and aspect, where slope is the inclination of a land surface and aspect is the direction the surface is facing. The weather has a dynamic influence on wildfire behavior and includes three components: wind speed, wind direction, and moisture content. Because of the dynamic weather, non-uniform terrain, and different fuel loadings, wildfire spreading is a highly heterogeneous process with non-uniform spreading speed and fireline intensity in both space and time. For a dynamically growing wildfire, the head of a fire typically has a faster-spreading speed and higher fireline intensity than the tail of the fire.

To collect the most useful information about a wildfire, it is necessary to assign different levels of monitoring attention to different parts of the fire. The more active regions of the fire are more important to be monitored and need to be visited more frequently; the less active regions have less changes in fire state and thus can be visited less frequently. Motivated by this need, previously we developed an importance-based multi-UAS path planning method that coordinates multiple UASs to monitor a wildfire with different levels of monitoring attention for different regions of the fire perimeter [1]. The developed method adopted a centralized coordination approach that assumes all the related wildfire and UAS data are available on a central computer (e.g., a ground station), and uses a search method to optimize the task allocations of the multiple UASs. The centralized path planning results are then communicated to each UAS for execution. While this centralized approach provides optimal results, it is not always practical for the wildfire monitoring task. Wildfires happen in extremely challenging environments, e.g., in mountain areas that have limited or unstable wireless communications. To achieve robust and autonomous path planning, it is necessary for the UASs to have real-time on-board path planning capabilities so that they can monitor a wildfire without depending on a centralized computing and decision-making system.

In this paper, we present a novel on-board path planning algorithm for UAS based wildfire monitoring. The proposed algorithm uses real-time data collected by the UAS and still supports importance-based monitoring by paying more attention to more active fire regions. Real-time on-board path planning in a decentralized way for wildfire monitoring that

faces several unique challenges compared to a centralized approach. First and foremost, the UAS lacks full knowledge about the wildfire. Wildfire spreading is a dynamic process that constantly changes the size and shape of the fire. For the problem of wildfire monitoring, the most important information is the boundary or perimeter of the fire and the rate of spread at different locations of the perimeter. To achieve effective path planning, a UAS needs to know the fire perimeter and rate of spread in real-time. Nevertheless, for large-scale wildfires, at any moment a UAS can only monitor a small portion of the fire perimeter that is below the UAS' flying trajectory. This means the UAS needs to have a mechanism to construct the full fire perimeter in real-time and estimate the rate of spread of the different segments of the perimeter. The constructed fire perimeter represents a "global view" of the fire to support the path planning. Second, due to real-time requirements and the limited computing resources of UAS, computation efficiency is important for real-time on-board path planning. This means the proposed algorithm should take computation cost into consideration and need to achieve a balance between accuracy and cost.

Based on the design principles described above, we developed the real-time onboard path planning algorithm to determine the UAS's best flying direction along the fire perimeter to effectively monitor a spreading wildfire. Results are presented for different simulated wildfire scenarios to prove its effectiveness. This paper focuses on the real-time on-board path planning using a single UAS. Based on encouraging results for a single UAS, we have plans to extend this algorithm to support multiple UASs in our future work. The remainder of the paper is organized as: Section II describes the related works. Section III describes the proposed on-board path planning algorithm in detail. Section IV shows the experiment results on different simulated fire scenarios. Finally, Section V concludes the paper.

II. RELATED WORK

UAS-based wildfire monitoring has got more and more attention from researchers over the years as the technical capabilities of UASs increased. One of the major concerns among researchers is how to monitor a wildfire using UASs efficiently and autonomously. Wildfire monitoring can be regarded as a path planning problem where the area needed to be covered is the fire boundary and the mobile agents to be used are the UASs. A wide range of coverage path planning strategies with unmanned aerial vehicles on different shapes of areas of interest, such as rectangular, concave, and convex polygons has been presented in [6]. The authors described different flight patterns including geometric flight patterns, such as back-and-forth and spiral, and more complex grid-based solutions to cover different shapes. However, the problem of monitoring a wildfire is different in the extent that the shape of the fire is not a regular geometric shape such as a rectangle or a polygon, moreover, the shape also changes over time as the fire progresses.

Several previous works demonstrated the use of UASs in the context of wildfire monitoring. The wildfire monitoring problem has been defined as a fire perimeter coverage problem in [1]. The goal of this work is to have a balanced coverage of the fire perimeter using multiple UASs so that the UASs can collect the most useful information about the fire

and construct a fire shape as accurately as possible. The key concept that has been used here is to treat different parts of the fire have different levels of importance. UASs are assigned different regions of the perimeter to monitor those regions in a back-and-forth approach. Over time, adjacent UASs performs boundary negotiation between them to maintain a balanced responsibility. In this work, the necessary information for monitoring the fire is provided by a ground station.

A forest fire perception system has been presented in [4] that uses a fleet of aerial vehicles. That system integrates information from several aerial vehicles to estimate real-time fire spread. The proposed system has been tested on controlled fire experiments and it shows the potential of using UASs in wildfire monitoring. Here, a significant part of the computation is performed by two central systems - Central Perception System and Central Decision System. A deep learning-based forest fire monitoring system has been presented in [10]. That system uses images acquired from unmanned aerial vehicles through the connected optical sensor. A convolutional neural network is pre-trained with past forest fire images and the unmanned aerial vehicle sends query images to this network. Based on the past dataset a centralized system recognizes a wildfire. This work presented promising results; however, the presented system is dependent on a centralized system. Furthermore, to effectively monitor a wildfire, in addition to the recognition of a forest fire, more specific information about the spreading of wildfire is required.

A distributed leader-follower coalition formation model has been presented in [11], where a set of drones are grouped into multiple sub-groups to cover a designated field. When a fire incident is reported, a mission is initiated by the UASs. A set of follower UASs relies on a group leader UAS for the monitoring task. This work supports on-board decision making; however, it does not comprehensively support importance-based identification and monitoring of the most actively spreading region. Though a mechanism for putting more focus on the more active region has been mentioned, the approach seems not to be very robust. A more sophisticated approach is required for the identification and monitoring of the most active fire region.

Given the most recent advancements of using UASs in wildfire monitoring, there is a high need for an on-board path planning algorithm that focuses more closely on the most active region of a wildfire. Based on that need, the contributions of this paper include i) development of an importance-based on-board path planning approach for UAS based wildfire monitoring ii) design and implementation of the real-time on-board algorithm based on that importance-based approach iii) evaluation of the proposed algorithm's effectiveness on different simulated wildfire scenarios. To the best of our knowledge, none of the previous works considered importance based on-board path planning approach for wildfire monitoring.

III. THE ON-BOARD PATH PLANNING ALGORITHM

The overall goal of the real-time on-board path planning is to direct the UAS to collect the most useful information about a spreading wildfire based on on-board data collected

by the UAS itself (without relying on an external computing system). The algorithm is developed based on the following setting. First, fire monitoring focuses only on the burning perimeter of the fire. This is consistent with many of the other works in the literature because the location of the fire perimeter is the most important information for fire managers. Second, the UAS has the capability of boundary following to always fly on top of the fire perimeter. During the fire monitoring task, the UAS follows the fire perimeter by flying clockwise (CW) or counterclockwise (CCW) on top of the fire perimeter. With the above two assumptions, the path planning task essentially becomes deciding when the UAS needs to change its flying direction (CW or CCW) along the fire perimeter. In the simplest case, a UAS can keep circling the fire perimeter without changing its flying direction. This allows the UAS to cover the perimeter of the fire in a cyclic fashion. Nevertheless, it does not allow the UAS to pay more attention to the more active fire perimeter segments because all segments are treated equally.

A sample fire spreading scenario is presented in Fig. 1. The red region represents the overall fire progress, the white dot represents the ignition point (from where the fire started), the yellow line represents the UAS trajectory since the UAS has been deployed, and the white arrows represent fire spread at different directions of the fire. It is clear that fire spread is uneven across different regions of the fire and more monitoring attention is required on the fast-spreading region. Since fire spreads in a non-uniform way, it makes sense for the UAS to turn back and forth to monitor the fire segment that spreads very fast, while still covering the entire fire perimeter from time to time. The on-board path planning algorithm aims to decide when the UAS needs to change its flying direction along the fire perimeter.

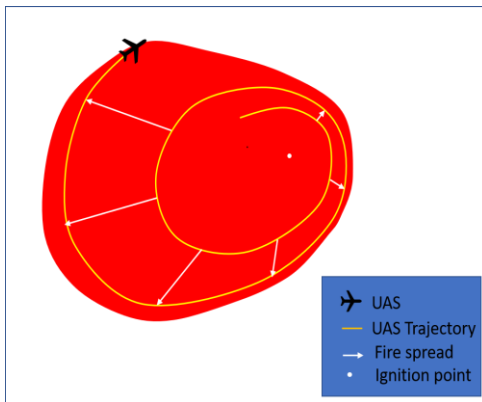


Figure 1: Sample fire spread scenario

This paper focuses on the high-level path planning and ignores the low-level UAS control. We represent the fire area as a 2D space and discretize it into a cell space, whose size can be set according to the granularity of path planning. Accordingly, in the path planning algorithm, we represent the fire perimeter as a discretized boundary comprising individual cells corresponding to the locations of the fire perimeter. Based on this discretized approach, we assume the UAS always flies from the center of a cell to the center of a neighboring cell that is on the fire perimeter along its flying direction (CW or CCW). After reaching the destination cell,

it finds a new neighboring cell on the fire perimeter as the next destination to fly to.

Below we explain how the cell importance is calculated as well as the design principles of the importance based on-board path planning algorithm.

A. Cell Importance

The basic idea of the importance-based path planning is to treat different segments of a fire perimeter to have different importance levels that represent different levels of monitoring attention. Each cell that makes up the fire perimeter is assigned a dynamic value that represents the importance of visiting that cell for data collection. This importance can be thought of as the value of the data to be collected from a cell, which is related to the information uncertainty of that cell. If a specific segment of the fire perimeter spreads very fast and/or has not been visited for a long time, then there is more information uncertainty for that segment of the fire perimeter, and thus the segment has more importance.

Based on the above idea, the algorithm utilizes several key measurements that are necessary for quantifying the importance of the different parts of the fire perimeter. For the on-board path planning, the UAS needs to calculate those measurements in real-time based on the information collected along its flying trajectory. As the UAS flies, it saves the visiting time for each cell that it has visited. The cells that have been visited by the UAS are referred to as the trajectory cells in this paper. For each trajectory cell, the UAS assigns an angle value to it to represent the direction of the cell from the ignition point of the fire. Due to limited computing resources, we only consider a discretized set of angle values. The granularity of the discretization of the angles is a tradeoff between computation precision and on-board computation efficiency. In this work, we consider 360 discrete angles to keep track of the visited cells as it provides adequate precision and efficiency. Fig. 2 illustrates four trajectory cells A, B, C, and D and their corresponding angles θ_i and θ_j . Note that Cell A and B have the same angle θ_j , and cell C and D have the same angle θ_i . The cell O represents the cell where the UAS is first deployed. In this example, the UAS flies clockwise around the fire without changing direction. The real fire shape is not shown in the figure.

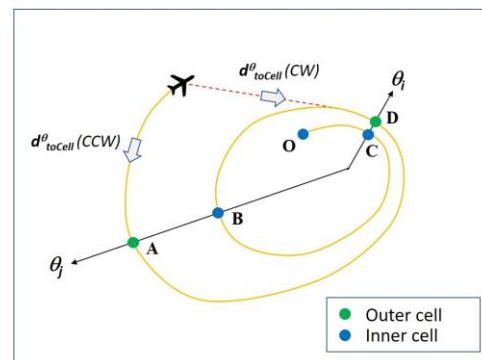


Figure 2: Angle based trajectory cell tracking

The importance of a cell is directly related to how fast the fire spreads in the direction of the cell and how much time has elapsed since the cell was last visited. Based on this idea, we consider three measurements for computing the importance of

a cell at an angle θ . 1) the rate of spread (ROS^θ); 2) the time elapsed since the cell was last visited ($t_{sinceVisit}^\theta$), and 3) the time it will take for the UAS to reach the cell from its current location (t_{toCell}^θ).

First, the ROS^θ is the measure of how fast the fire is spreading along the direction of angle θ . To calculate ROS^θ , we use the most recent trajectory cell along the angle θ and the previous trajectory cell along the same angle. The former is referred to as the *outer cell*, and the latter is referred to as the *inner cell*. Specifically, in our implementation for each angle θ , the UAS keeps track of the distance of the outer cell and inner cell from the ignition point (d_{outer}^θ and d_{inner}^θ) and the visit time of those cells (t_{outer}^θ and t_{inner}^θ). When the UAS reaches a new cell at the same angle θ , the previous outer cell at that angle becomes the inner cell. The outer cells and inner cells at two sample angles θ_i and θ_j are displayed in Fig. 2. With the information of the outer cell and inner cell, for each angle θ , the ROS^θ is calculated using (1).

Secondly, for each angle θ , the time since last visit $t_{sinceVisit}^\theta$ is measured by the difference between the current time ($t_{current}$) and the visit time of the outer cell at the direction θ and calculated as per (2).

$$ROS^\theta = \frac{d_{outer}^\theta - d_{inner}^\theta}{t_{outer}^\theta - t_{inner}^\theta} \quad (1)$$

$$t_{sinceVisit}^\theta = t_{current} - t_{outer}^\theta \quad (2)$$

$$t_{toCell}^\theta = \frac{d_{toCell}^\theta}{UAS_SPEED} \quad (3)$$

$$importance^\theta = ROS^\theta * (t_{sinceVisit}^\theta + t_{toCell}^\theta) \quad (4)$$

The previous two measurements allow us to compute how far the fire has spread in the direction of a cell since the cell was last visited. This spreading distance is directly related to the importance of the cell at the current moment. While we can develop the path planning algorithm based on the current importance of cells, it makes more sense to consider the “potential importance” of cells, which is the importance when the UAS actually reach the direction of the cell from its current location following the fire perimeter. This is because it takes time for the UAS to reach a cell and during this period the fire will spread further in the direction of the cell. The longer it will take for the UAS to reach a cell, the larger the “potential importance”. Based on this idea, when computing the importance of a cell we consider the third measurement t_{toCell}^θ , which is the time needed for the UAS to travel from its current location to the cell at the angle θ . Since the UAS always flies CW or CCW along the fire perimeter, the t_{toCell}^θ would be the “perimeter distance” to the cell, denoted as d_{toCell}^θ , divided by the flying speed of the UAS, as shown in (3).

To calculate t_{toCell}^θ , we need to know d_{toCell}^θ . Several things are worthy mention for computing the d_{toCell}^θ . First, depending

on if the UAS flies CW or CCW, the distance to a specific cell would be different. In the example of Fig. 2, if the UAS keeps moving CW, the d_{toCell}^θ to cell A would be the length covering all the perimeter cells between the UAS and cell A along the CW direction. If the UAS turns back and moves CCW, the d_{toCell}^θ to cell A would be the length covering the perimeter cells between the UAS and cell A along the CCW direction. Second, as the fire is always spreading, the UAS does not have full knowledge about the fire perimeter. This means to compute d_{toCell}^θ the UAS needs to construct the fire perimeter based on the information it has collected. The fire perimeter construction needs to take into account the fact that the trajectory cells following the UAS might not be smoothly aligned into a closed loop at the current location of the UAS. This is because, as the fire is continuously spreading, there is some unvisited fire region ahead of the UAS. To counter this monitoring gap, it is important to construct an accurate fire shape based on the UAS trajectory. We use a fire shape reconstruction procedure to estimate the fire perimeter of that unvisited region as shown by the dashed red line in Fig. 2 (see later for more details). The reconstructed fire perimeter represents the best knowledge about the real fire shape based on the UAS trajectory. Thus, we use this reconstructed perimeter as the real fire perimeter to calculate d_{toCell}^θ . Third, because the fire is always spreading, by the time when the UAS reaches the direction of a target cell, the fire has already spread further. In other words, there is an error between the d_{toCell}^θ that we calculate based on the “current perimeter” and the actual travel distance for reaching the direction of the target cell (in fact, d_{toCell}^θ would be smaller than the actual travel distance because the fire grows larger). In our current implementation, we ignore this error and use the currently estimated fire perimeter to compute the d_{toCell}^θ .

Based on the three measurements described above, we calculate the importance of a cell using (4). As can be seen, the importance of a cell is essentially defined by the new fire spread distance in the direction of the cell between when the cell was last visited and when the cell is projected to be revisited by the UAS.

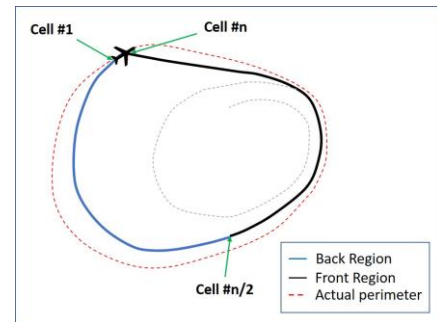


Figure 3: Importance based path planning

B. Importance based path planning

Based on the importance of each cell, the next step is to utilize cell importance values to carry out path planning. In our design, we say that the UAS should keep moving forward unless it is more beneficial to turn back, i.e., to change its flying direction. It makes sense for the UAS to turn back when the fire is spreading significantly faster on the back of the UAS. In such situations, the cell importance values of the

back region are also higher. Therefore, a mechanism is required to identify if the total cell importance at the back region is higher compared to the remaining region of the perimeter. For that purpose, we divide the entire perimeter into two regions – one region represents the perimeter section at the back of the UAS denoted as the back region and the other region represents the remaining perimeter section denoted as the front region. In this work, we considered each of those two regions to be equal. Therefore, 50% of the perimeter cells reside within the back region and the rest 50% cells reside within the front region. Among those two defined regions, the UAS dynamically chooses the one with more importance. For that purpose, we calculate the total importance of the back region ($IMPORTANCE_{BACK}$) and compare it to the total importance in the front region ($IMPORTANCE_{FRONT}$). Assuming there are total n cells in the current perimeter, $IMPORTANCE_{BACK}$ and $IMPORTANCE_{FRONT}$ are calculated according to (5) and (6) respectively, where θ_i is the angle of the i th perimeter cell. If $IMPORTANCE_{BACK}$ is larger, it means that the back region is more active than the front region, therefore, the UAS changes its flying direction along the perimeter to cover the more active region. Every time the UAS reaches a new cell, this procedure for checking the best flying direction is invoked. Thus, the real-time UAS path planning is performed dynamically by guiding the UAS towards the best flying direction along the perimeter.

$$IMPORTANCE_{BACK} = \sum_{i=1}^{n/2} importance^{\theta_i} \quad (5)$$

$$IMPORTANCE_{FRONT} = \sum_{i=\frac{n}{2}+1}^n importance^{\theta_i} \quad (6)$$

The overall concept of dividing the fire perimeter into the back and front region is illustrated in Fig. 3. Based on the fire spreading scenario presented in Fig. 1, the dashed red line represents the perimeter of the actual fire shape. The blue line represents the UAS back region, the black line represents the UAS front region. It is worth noting that the back region and front region are based on the fire perimeter constructed by the UAS using the on-board data in real-time. This constructed fire perimeter is different from the actual fire perimeter (the red line in Fig. 3), which is unknown to the UAS. The rest of the UAS trajectory marked by the grey dashed line is not a part of the reconstructed fire perimeter. Therefore, in this sample scenario, $IMPORTANCE_{BACK}$ is the sum of the importance of the cells falling under the blue line (cell number 1 to $n/2$), and $IMPORTANCE_{FRONT}$ is the sum of the importance of the cells falling under the black line (cell number $n/2+1$ until n). By comparing these two values, the best flying direction is determined.

C. The Real-Time On-board Algorithm

Based on the above descriptions, this section describes the design and implementation details of the real-time on-board path planning algorithm. The algorithm can be decomposed into three main components: i) Initialization of the on-board fire map ii) Fire shape reconstruction iii) Determining the optimal flying direction. The overall steps involved in the algorithm are shown in listing 1.

LISTING 1: OVERVIEW OF THE PROPOSED ALGORITHM

1. Initialize the on-board fire map when UAS is deployed
2. In each step (after UAS reaches a new cell), DO:
 - i. Reconstruct fire shape based on the on-board fire map
 - ii. Determine the importance of the back and front region
 - iii. IF $IMPORTANCE_{BACK} > IMPORTANCE_{FRONT}$
 - Change flying direction
 - ELSE
 - Keep flying in the current flying direction
 - iv. Select the next cell along flying direction and fly over there

Initialization of the on-board fire map: The UAS is deployed at the boundary of the fire after a certain time since the fire was started. The UAS needs the initial fire spread information such as which cells are burning at the time of deployment. It is assumed that the UAS has been provided with that initial spread information. Based on that initial information, the UAS generates an on-board fire map and keeps track of the necessary measurements for each cell as it moves along the fire perimeter. The structure of the on-board fire map is shown in Fig. 4. The map is based on 360 discrete angles around the ignition point and it keeps track of the outer cells and inner cells along each angle. It is worthwhile to note that when there are multiple cells in between two discrete angles, the on-board map also keeps track of those intermediate cells. Keeping track of the intermediate cells is necessary for the precise calculation of the back and front region importance.

Angle	Outer Cell	Inner Cell	Intermediate Cells
0	cell ₀	cell ₀	cell_list ₀
1	cell ₁	cell ₁	cell_list ₁
...
359	cell ₃₅₉	cell ₃₅₉	cell_list ₃₅₉

Figure 4: On-board fire map

When the UAS is initially deployed, it has no knowledge about the inner cells of the initial fire perimeter. In our implementation, the ignition point is considered as the initial inner cell. Then, as the UAS moves along the perimeter, the outer cells, inner cells, and intermediate cells are updated on the map accordingly. The on-board fire map initialization procedure is shown in listing 2.

LISTING 2: PROCEDURE FOR ON-BOARD FIRE MAP INITIALIZATION

INPUT: Initial fire spread information

OUTPUT: Initialized on-board fire map

FOR each perimeter cell in the provided fire map **DO**:

- i. Determine the angle θ of the cell from the ignition point
- ii. Add the cell as the outer cell at the angle θ in on-board fire map
- iii. Add ignition point as the inner cell at the angle θ

END FOR

Fire shape reconstruction: Every time the UAS arrives at a new cell, an on-board fire perimeter reconstruction procedure is invoked to generate a more accurate fire perimeter. The on-board fire perimeter reconstruction procedure utilizes the previous trajectory cells to form a full loop of the fire perimeter. To deal with the issue that the

trajectory cells following the UAS might not be smoothly aligned into a closed loop at the current location of the UAS (as illustrated in Fig. 5(a)), a scan method is used to find the best cell among the previous trajectory cells that can generate a smooth closed loop by connecting with the UAS' current cell. The scan method works as - starting from the angle next to UAS, it keeps scanning as long as the angle from UAS's current cell to the trajectory cell keeps increasing. This scan method is based on the assumption that fire spread tends to generate convex shapes.

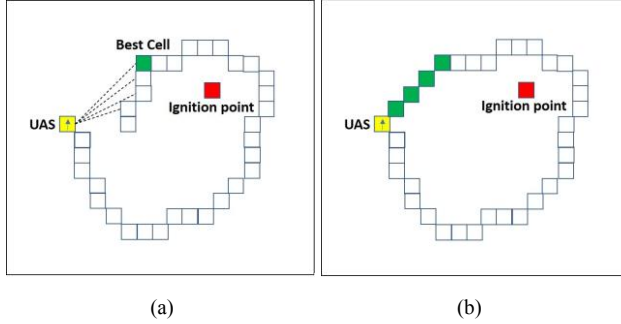


Figure 5: On-board fire perimeter reconstruction a) before reconstruction b) after reconstruction

Fig. 5 illustrates how the on-board fire perimeter reconstruction works. In Fig. 5(a), the white cells are previous trajectory cells; the yellow cell is the UAS's location, and the best cell found by the scan method is represented by the green cell. This is the best cell because the angle (shown by dashed lines) from UAS's current position keeps increasing until this cell. Beyond this cell, the angle decreases if scanned further along the trajectory cells. Finally, the trajectory cells between the UAS occupied cell and the best cell are replaced by the cells that fall under the connecting straight line, as highlighted by the green cells in Fig. 5(b). This way a smooth fire shape is generated using the on-board information collected by the UAS.

Determining the optimal flying direction:

LISTING 3: PROCEDURE FOR FINDING OPTIMAL FLYING DIRECTION

INPUT: On-board fire map, UAS's current flying direction

OUTPUT: Optimal flying direction (CW/CCW) along the fire perimeter

Calculate the importance of the perimeter as:

FOR each cell "*current_cell*" in the on-board fire perimeter **DO**:

- i. Calculate ROS^θ of the *current_cell*
- ii. Calculate $t_{sinceVistr}^\theta$ of the *current_cell*
- iii. Calculate t_{toCell}^θ of the *current_cell*
- iv. Determine $importance^\theta$ of the *current_cell*

END FOR

Calculate $IMPORTANCE_{BACK}$ and $IMPORTANCE_{FRONT}$

Change the flying direction if $IMPORTANCE_{BACK} > IMPORTANCE_{FRONT}$

After the fire perimeter reconstruction, the final step is to calculate $IMPORTANCE_{BACK}$ and $IMPORTANCE_{FRONT}$ and decide the optimal flying direction for the UAS. The decision-making procedure is shown in listing 3. This procedure needs the on-board fire map and UAS's current flying direction as inputs. From the current position, the UAS first extracts a list of connected cells that represents the perimeter of the

reconstructed fire shape. Next, it calculates the importance of each cell on the reconstructed perimeter according to (4). Finally, the total importance of the back and the front region is calculated from those individual cell importance values. The UAS changes its current flying direction if $IMPORTANCE_{BACK}$ is larger than $IMPORTANCE_{FRONT}$. It is worth noting that this algorithm is independent of the cell size, the decision regarding UAS's optimal flying direction doesn't depend upon it.

D. Extension for Multiple-UAS

To make the algorithm applicable for multiple UASs, it is important to make sure that the UASs do not concentrate on the same part of the perimeter. To do so, some sort of inter-UAS communication may be required. One potential idea is for the multiple UAS to share their on-board maps periodically. Merging the on-board maps from multiple UASs will result in a global on-board fire map that can be used by multiple UASs. Using that global map, the UASs can then decide how they may divide the monitoring task. We leave this part as our future work.

IV. EXPERIMENT RESULTS

A. Wildfire Spread Simulation

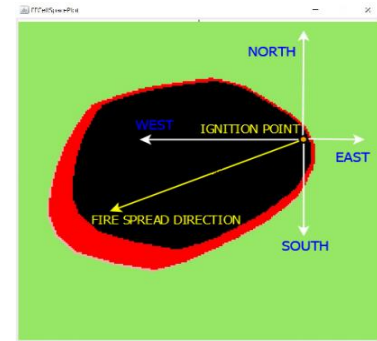


Figure 6: The Fire Spread Simulation Model

To simulate different wildfire scenarios for our experiments, we have used DEVS-FIRE [2, 3], a discrete event simulation model for surface fire spread simulation developed based on the Discrete Event System Specification (DEVS) formalism [5]. It uses a cellular space to represent a wildland area, where each cell has its own terrain and fuel (vegetation) data corresponding to the sub-regions in the area. All cells are coupled to a weather model to receive weather data (wind speed and wind direction) over time. Once a cell is ignited, it uses Rothaermel's model [7] to compute the fire spread rate and direction within the cell. Fire spreading is modeled as a propagation process as burning cells ignite their unburned neighboring cells. The DEVS-FIRE model has been validated by comparing with other wildfire simulation tools [8] and by simulating historical wildfires [9]. A sample fire spread scenario using DEVS-FIRE has been shown in Fig. 6. The green regions represent the part of the land that is not impacted by the fire yet. The red region represents the currently burning area and the black region represents the area that has already burnt out. To describe the fire spread and path planning results, we specify the directions of the cell space as shown in Fig. 6. In this scenario, the fire is mostly spreading towards the west and south-west direction since it started from the ignition point.

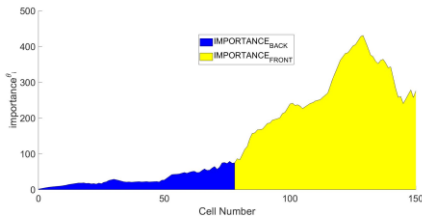
We are going to present simulation results for several fire scenarios resulting from different wind speed and wind direction configurations. For the experiments, the wildland has been modeled as a 200x200 cell space using DEVs-FIRE. A specific cell is ignited in the beginning and starting from there, the fire spreads as per the fire spread model and weather configuration. The UAS is deployed on the fire perimeter 1 hour after the fire has started.

B. Demonstration of on-board path planning

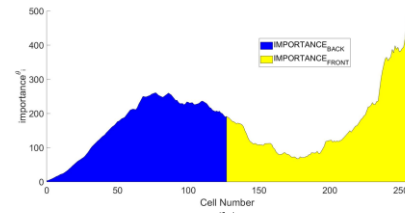
First, we are going to demonstrate the path planning result in detail through one specific fire spread scenario. In this scenario, as shown in Fig. 7, the fire starts from the northeast side of the cell space and spreads towards the southwest direction over time. During the first 3 hours of the simulation, the fire was not spreading very fast, as a result, the UAS did not make too many revisits during that period. However, for the remaining 3 hours of simulation, the fire was spreading significantly faster. During that time, the UAS made frequent back and forth revisits to cover the most spreading part of the fire towards the southwest direction.



Figure 7: UAS trajectory based on the on-board algorithm



(a)



(b)

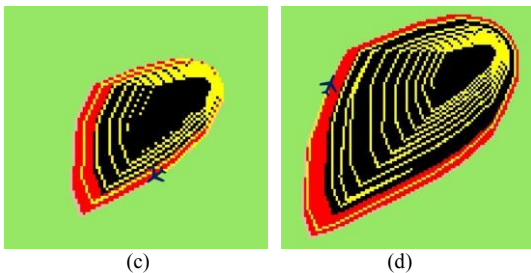


Figure 8: Illustration of back and front region importance

Fig. 8(c) presents a snapshot from the simulation where the UAS is moving towards the most active fire region; the spread in the back of the UAS is very slow. The corresponding importance values of the reconstructed fire perimeter cells have been plotted in Fig. 8(a). From the plot, we can see that the total importance in the back region is significantly smaller than in the front region. Therefore, it is better for the UAS to keep flying in the same direction in this scenario. In contrast, Fig. 8(d) presents a snapshot where the fire is spreading much faster behind the UAS. The importance of the reconstructed fire perimeter cells has been plotted in Fig. 8(b). In this case, the total importance in the back region has just got larger than the total importance in the front region. Hence, the UAS needs to change its flying direction to cover the most active region of the fire.

For the fire monitoring scenario presented in Fig. 7, $IMPORTANCE_{BACK}$ and $IMPORTANCE_{FRONT}$ have been recorded whenever the UAS moved to a new cell. How the total importance in the back and front region changed over time has been shown in Fig. 9. The importance in the back region keeps increasing until it becomes equal to the importance of the front region (points where the two curves met each other in Fig. 9). At those points, the UAS changes its flying direction and continues to fly in that direction until $IMPORTANCE_{BACK}$ becomes larger than $IMPORTANCE_{FRONT}$ and so on.

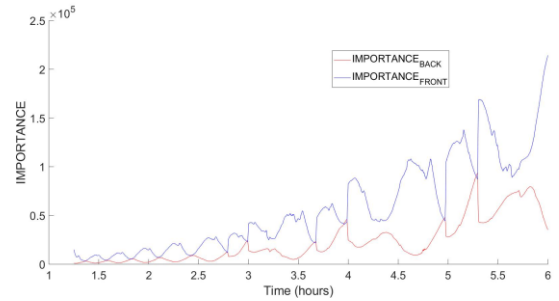


Figure 9: Change of importance in the back and front region

C. Path planning for different fire spread behaviors

We consider two different fire spreading behaviors and evaluate how the proposed algorithm performs on those special fire spread behaviors. First, we consider a scenario where fire spreads very fast towards a specific direction as shown in Fig. 10(a). Here, the fire started from the northeast corner of the cell space and spread towards the southwest direction much faster. As a specific region of this fire is spreading very fast, the UAS revisits that fire front very frequently to put more monitoring attention. Second, we consider a scenario where the fire is spreading very slow as shown in Fig. 10(b). There is no significantly fast-spreading region in this scenario, therefore $IMPORTANCE_{BACK}$ never grown larger than $IMPORTANCE_{FRONT}$. Consequently, the UAS always moved forward in a cyclic fashion. Thus, when the fire spread is slow, the proposed algorithm works like a basic circling algorithm. In contrast, if there is a region that is spreading significantly faster, the UAS changes its flying direction over time to cover the most active region of the fire. Thus, the proposed algorithm is adaptive to different fire spread behaviors.

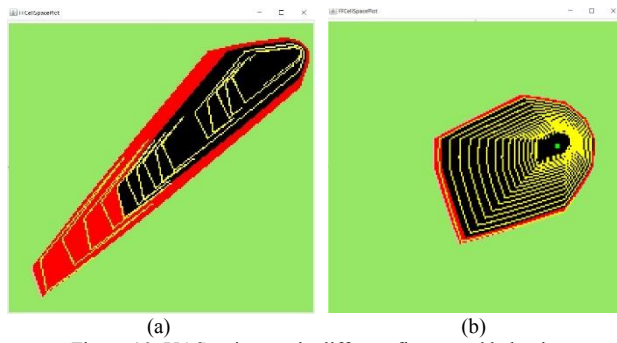


Figure 10: UAS trajectory in different fire spread behaviors

D. Comparison with the circling-based approach

One of the baseline approaches for monitoring a wildfire is just circling the perimeter for the entire time. This method is straightforward and doesn't consider the uneven importance of the fire perimeter. This approach makes sense when the fire is spreading slowly and evenly at all angles around the fire center. However, if a section of the fire is spreading much faster, this approach is unable to provide higher monitoring attention to that region. Fig. 11(a) and Fig. 11(b) show the UAS trajectories based on the circling-based approach corresponding to the fire spread scenarios used in Fig. 7 and Fig. 10(a) respectively. For both scenarios, we can notice larger monitoring gaps on the faster-spreading southwest side of the fire compared to the on-board algorithm.

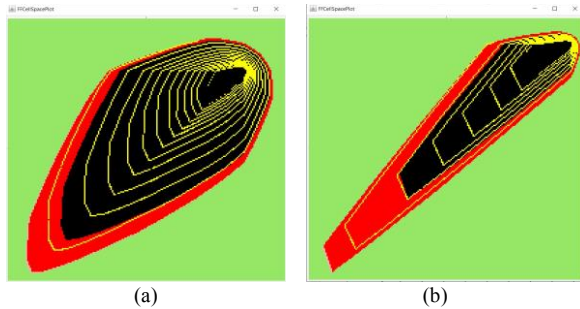


Figure 11: UAS trajectory based on the circling-based approach

While the UAS is visiting a section of the perimeter, the fire is possibly spreading in other sections with different spreading speeds. Therefore, the UAS lacks knowledge about those unvisited parts until it visits those sections again. This knowledge gap results in inaccuracies between the actual fire shape and fire shape constructed from UAS data. To quantify this inaccuracy for both the circling-based method and the proposed algorithm, we measure the difference between the original fire shape and the constructed fire shapes at each of the 360 discrete angles. We name this difference as "distance error" in this paper. Fig. 12 illustrates the concept of distance error for a sample fire monitoring scenario. The original fire shape has been highlighted by the red region and the fire shape constructed from the UAS trajectory has been marked by the blue region. To construct fire shape from UAS trajectory, first, we used the convex hull algorithm [12] to construct the convex hull (the smallest convex set containing a set of points) of the on-board perimeter cells. Then, we have filled in that convex hull to get constructed fire shape. Between these two shapes, different amount of distance error is present in different parts of the fire as shown by the yellow arrows in Fig. 12.

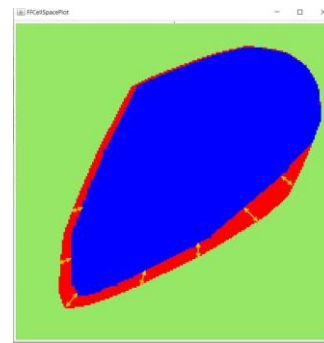


Figure 12: Distance error between the original fire shape and constructed fire shape

One of the major goals of UAS based wildfire monitoring is to construct a fire shape from the collected data in real-time as accurately as possible. The maximum distance error for a more accurate fire shape will be smaller compared to a less accurate fire shape. To quantitatively compare the proposed algorithm with the circling-based method, we have recorded this maximum distance error during the simulation whenever the UAS moved to a new cell. Fig. 13(a) compares the maximum distance error of these two approaches for the fire spread scenario shown in Fig. 11(a). From the comparison, we can see that the maximum distance error is similar for the first 3 hours of the simulation as the fire was spreading slowly during that time. However, when the fire started to spread faster after that period, the maximum distance error of the proposed algorithm is smaller than the circling-based method on average.

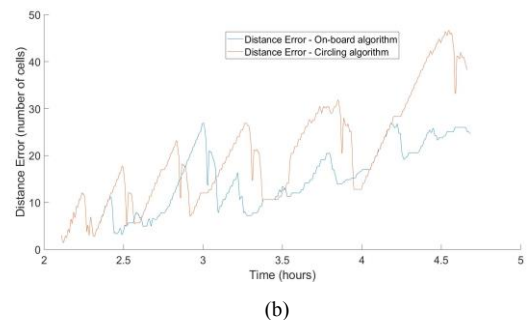
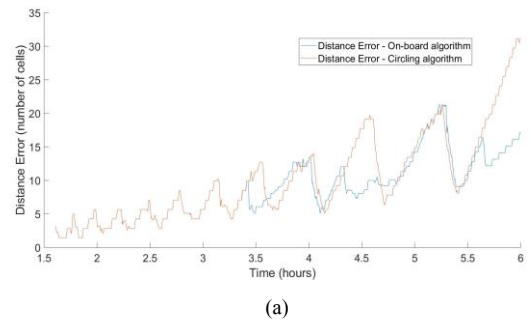


Figure 13: Comparison of the on-board algorithm and circling based approach

Fig. 13(b) compares the maximum distance error for the fire spread scenario shown in Fig. 11(b). In this scenario, the fire was spreading very fast towards the southwest direction from the very beginning. From the comparison, we can see that the circling-based approach has much larger maximum distance errors for this fast-spreading scenario. This is

because, when the UAS was visiting the slower spreading northeast side of the fire to make full cycles, the fire on the southwest side was spreading very fast, which resulted in higher inaccuracy for the circling-based method. In contrast, the on-board algorithm was making frequent revisits to the faster-spreading southwest side, which resulted in smaller maximum distance errors.

V. CONCLUSION AND FUTURE WORK

In this work, we presented a new approach for real-time on-board path planning intended to be used for UAS based wildfire monitoring. Real-time and on-board path planning can be a very useful mechanism in challenging scenarios like lack of a ground system, unknown environment, and lack of information about the land and weather, and many more. Simulation results for multiple fire scenarios have been presented to show the effectiveness and adaptiveness of the proposed algorithm. In the future, we want to extend the algorithm for multiple UASs and evaluate the algorithm under more diverse and challenging fire spreading scenarios. We believe that the algorithm presented in this paper will act as a platform to develop more interesting on-board path planning algorithms in the future.

ACKNOWLEDGMENT

This work is supported in part by the US Department of Agriculture (USDA) National Institute of Food and Agriculture (NIFA) under grant number 2019-67021-29011.

REFERENCES

- [1] X. Hu, J. Bent and J. Sun, "Wildfire Monitoring with Uneven Importance Using Multiple Unmanned Aircraft Systems," 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 2019, pp. 1270-1279, doi: 10.1109/ICUAS.2019.8797793.
- [2] L. Ntamo, X. Hu, and Y. Sun, "DEVS-FIRE: Towards an integrated simulation environment for surface wildfire spread and containment," *SIMULATION*, 84(4), 137-155, 2008.
- [3] X. Hu, Y. Sun, and L. Ntamo, "DEVS-FIRE: Design and Application of Formal Discrete Event Wildfire Spread and Suppression Models," *SIMULATION*, 88(3), 259-279, 2012.
- [4] L. Merino, F. Caballero, J. Martínez-de-Dios, I. Maza, and A. Ollero, "An Unmanned Aircraft System for Automatic Forest Fire Monitoring and Measurement," *Journal of Intelligent & Robotic Systems*, 65(1), 533-548, 2012.
- [5] B.P. Zeigler, H. Praehofer, and T.G. Kim, *Theory of modeling and simulation*, 2nd ed. New York, NY, USA: Academic Press, 2000.
- [6] Cabreira, T.M.; Brisolara, L.B.; Ferreira Jr., P.R. Survey on Coverage Path Planning with Unmanned Aerial Vehicles. *Drones* 2019
- [7] R. C. Rothermel, "A mathematical model for predicting fire spread in wildland fuels." Research Paper INT-115. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 40 p, 1972.
- [8] F. Gu, X. Hu, and L. Ntamo, "Towards validation of DEVS-FIRE wildfire simulation model," *Proceedings of the 2008 Spring Simulation Multiconference, SpringSim 2008*, 2008.
- [9] N. Dahl, H. Xue, X. Hu, and M. Xue, "Coupled Fire-Atmosphere Modeling of Wildfire Spread Using DEVS-FIRE and ARPS," *Natural Hazards*, 77(2), 1013-1035, 2015.
- [10] S. Kim, W. Lee, Y. Park, H. Lee and Y. Lee, "Forest fire monitoring system based on aerial image," 2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), Vienna, 2016, pp. 1-6, doi: 10.1109/ICT-DM.2016.7857214.
- [11] F. Afghah, A. Razi, J. Chakareski and J. Ashdown, "Wildfire Monitoring in Remote Areas using Autonomous Unmanned Aerial Vehicles," *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, 2019, pp. 835-840, doi: 10.1109/INFOCOMW.2019.8845309.
- [12] F.P. Preparata, and M.I. Shamos, *Convex hulls: Basic algorithms*. In *Computational Geometry: An Introduction* (ed. F.P. Preparata & M.I. Shamos), pp. 95148, New York, NY, USA: Springer-Verlag, 1991