ASSIMILATING UAV-BASED OBSERVATION DATA FOR WILDFIRE SPREAD SIMULATION

ABSTRACT

Wildfire spread simulation simulates the dynamic spread of wildfires by modeling fire spread behavior under various fuel, terrain, and weather conditions. To achieve real-time fire spread prediction using wildfire spread simulation models, it is essential to assimilate real-time observation data from active fires into the simulation models. Unmanned Aerial Vehicle (UAV) has been increasingly used to collect data from wildfires. This paper presents a data assimilation method that assimilates UAV-based observation data for wildfire spread simulation. We define the data assimilation problem that works with a discrete event wildfire spread simulation model and UAV-based observation data. A particle filter-based data assimilation algorithm is developed to carry out the data assimilation task. Experiment results show the effectiveness of the developed data assimilation method for assimilating UAV-based data for wildfire spread simulation.

Keywords: wildfire spread simulation, data assimilation, UAV, real-time observation data, particle filter.

1 INTRODUCTION

In recent years, wildfire has attracted people's attention as a serious nature hazard that causes severe impacts on environment and human society. According to National Interagency Fire Center (NIFC) [1], since 2000 an average of 72400 wildfires have burned approximately 7 million acres of the land each year. Wildfire spread simulation has been used to help researchers and practitioners to study wildfire behavior and analyze wildfire risk. It also holds the potential to simulate/predict real-time wildfire spread for supporting real-time decision making in wildfire management. To achieve real-time simulation/prediction of wildfire spread, it is essential to assimilate real-time observation data into the simulation models. The real-time data carries information about the real-time conditions of active fires. This allows a simulation to be aligned with the real fires to achieve more accurate simulation results.

Collecting real-time wildfire data is a challenging task. Traditional wildfire data collection methods include satellite systems, manned aircrafts, and ground fire sensors. Each of them has its own limitations. For example, satellite data typically has a low time frequency and a coarse spatial resolution; manned aircrafts have limitations in terms of mission duration, mission safety, and cost; and ground fire sensors are difficult to deploy on-demand and can be damaged by fires. In contrast, Unmanned Aerial Vehicle (UAV) technologies have advanced rapidly in recent years and show great potential for wildfire monitoring and data collection. The advantage of UAVs include (but are not limited to): flexible deployment that can be dynamically adjusted based on wildfire spread, data collection in high spatial and temporal resolution compared to satellite systems, safety and low cost compared to manned aircrafts.

The increasing availability of real-time data collected from UAV asks for effective methods to assimilate UAV-based observation data for wildfire spread simulation. Data assimilation is a statistical technique used for state estimation. It integrates observation data with a dynamic model and offers effective adjustment on the model to revise model error and generate more accurate simulation results. While a UAV brings many advantages in wildfire data collection, the UAV-based observation data have several unique features that makes assimilating them for wildfire spread simulation a challenging task. First, a UAV typically has limited field of views that cover relatively small areas on the ground. This results in partial observations of large wildfires because at any time a UAV can only observe a portion of the wildfire area. As a wildfire dynamically spreads, there is a need to estimate the whole fire front of the fire based on the partial observations from UAV data. Second, UAV data typically has a lot of noise due to the challenging operating environments and limited sensor capabilities. For example, the oscillation of a UAV's body caused by

Proc. of the 2024 Annual Simulation Conference (ANNSIM'24), May 20-23, 2024, American University, DC, USA
P.J. Giabbanelli, I. David, C. Ruiz-Martin, B. Oakes and R. Cárdenas, eds.

turbulence makes the thermal camera mounted on the UAV have unstable viewing angles, which leads to noisy sensor data.

This paper presents a data assimilation method to effectively assimilate UAV-based observation data for wildfire spread simulation. We define and formulate the data assimilation problem that works with a discrete event wildfire spread simulation model and the UAV-based observation data. The data assimilation provides state estimation of the dynamically evolving fire front of an active fire based on partial and noisy UAV data. The state space formulation of the data assimilation problem includes a state transition model that is based on the wildfire spread simulation using the DEVS-FIRE model [2, 3], and a measurement model that is based on fire front sensing using the UAV's thermal camera. A particle filter-based data assimilation algorithm is developed to carry out the data assimilation task. To work with UAV-based observation data that provides partial observations and has a lot of noise, special treatments are developed to add process noise to the fire front state in state transition model and to compute particles' weights using the UAV data. These special treatments exploit the spatial dependency and locality features of UAV-based observation data to effectively support data assimilation. Multiple experiments are designed to evaluate the effectiveness of the developed data assimilation method for assimilating UAV-based observation data for wildfire spread simulation.

The remainder of this paper is as followed: Section 2 describes the related work. Section 3 presents the data assimilation problem for assimilating UAV-based observation data into DEVS-FIRE-based wildfire spread simulation. Section 4 describes the particle filter algorithm for carrying out the data assimilation task. Section 5 describes the experiment results. Section 6 concludes this work and discusses potential future work.

2 RELATED WORK

As a systematic method, wildfire simulation and modeling brings benefits in the wildfire study. Multiple wildfire simulation models have been developed, and have made great contribution in wildfire study. For example, FARSITE [4], BEHAVE-Plus [5], DEVS-FIRE [2,3] are all powerful tools for wildfire simulation and provide perspectives in static and dynamic fire spreading modeling. Rothermel's model [6] as an empirical fire behavior models, describes the fire midflame behavior and it is widely used in fire spreading rate and direction calculations in the wildfire simulation tools as the physical model.

Remote sensing techniques are effective tools in wildfire monitoring and data retrieving. Satellite sensing as a traditional technique has been widely used in wildfire detection and tracking in the world-wide areas [7, 8, 9]. UAV as a novel sensing technique has a great potential on wildfire monitoring. The state of the art on UAV based wildfire remote sensing has been studied [10]. UAVs have the advantage of being maneuverable, automatic, easy deploying and cost friendly which make UAV prominent in wildfire management [11]. The easy deployment and low cost of UAV comes with the benefit that fire data collection becomes convenient and fast. For example, UAV equipped with visible and infrared cameras are applied in real fire monitoring, using the captured data for image fusion [12]. UAV with thermal camera deployment is applied for both real-world wildfire monitoring [13, 14] and prescribed fire monitoring [15]. Infrared UAV images are used in LSTM model to predict forest fire spread rate [16]. The relevant works shows that UAV brings convenience in fire monitoring and the fire information can be efficiently retrieved. However, UAV sensors also have some weakness. A recent study demonstrates that UAV sensing camera has a limited scope of view with environment noise that only a portion of fire data could be retrieved. This way, the sensor data needs to be stitched and filtered to reconstruct the fire map [17].

Data assimilation has been popular in system state analysis with valid observation data. The Particle Filter (PF) [18], one of the popular data assimilation methods, outstands in non-Gaussian and dynamic system calibration. Much research has been done combining data assimilation technique and wildfire modeling benefits wildfire prediction. PF is applied with temperature sensing for fire shape simulation based on DEVS-FIRE [19]; A spatial partition-based particle filtering framework is also applied on wildfire simulation [20]. Focusing on the complexity of the dynamic system, Dynamic Data-Driven Simulation (DDDS) is proposed to calibrate online data and solves the issue of over-time characteristics changing in dynamic state prediction using PF [21, 22, 23]. There are other data assimilation method for example Kalman Filtering is developed when the system holds the properties of being linear and gaussian. For example, a reaction-diffusion model and a semi-empirical fire propagation model combining Ensemble Kalman Filter is proposed for wildfire forecasting [24]. Ensemble Kalman Filter on FARSITE for wildfire simulation is incorporated with data assimilation [25].

3 THE DATA ASSIMILATION PROBLEM

3.1 The state space formulation

Data assimilation typically uses a state space formulation that includes a state transition model and a measurement model. To carry out data assimilation using UAV-based observation data for wildfire spread simulation, we formulate the data assimilation problem using the state space formulation as shown below:

$$\textbf{State transition model}: fire_{t+1} = DEVS\_FIRE(fire_t) + \gamma_t, \qquad (1)$$

$$\textbf{Measurement model}: m_t \ = UAV\_Sensing(fire_t,\ P) + \zeta_t, \qquad (2)$$

The state transition model (1) describes how the system state (i.e., the fire state) evolves over time, where $fire_t$ and $fire_{t+1}$ are the fire states at data assimilation step t and t+1, respectively. The $DEVS\_FIRE()$ represents the DEVS-FIRE simulation model that captures how the fire state evolves over time, and γ_t is the process noise. The DEVS-FIRE model will be discussed in the next section. The process noise is necessary to capture the model error and the uncertainty that exist in fire spread. The measurement model (2) describes the mapping from the system state to the observation data. In this work, the observation data m_t includes the observed fire front locations. The fire state is mapped to the observation data m_t, based on how the UAV senses the fire state, denoted by $UAV\_Sensing()$, which takes the fire state $fire_t$ and UAV's position P as inputs. The ζ_t is the measurement noise that captures the noise involved in the UAV-based sensing. Details about the UAV-based sensing and the associated noise are described in Section 3.3.

3.2 The state transition model

The state transition model is defined by the DEVS-FIRE [2] wildfire spread simulation model plus process noise. DEVS-FIRE model is applied to formulate the state transition in fire spreading. DEVS-FIRE is a discrete event simulation model based on the Discrete Event System Specification (DEVS) formalism. The fire cell space in DEVS-FIRE is two-dimensional, in which each fire cell is the most basic element and is coupled with other fire cells to form the cell space. As an empirical physical model, DEVS-FIRE uses Rothermel model as the fire behavior model. Environment data such as GIS data, fuel data and weather data also becomes necessary system input to support the simulation.

There are in total three possible fire cell states based on the coordinate position: *{unburned, burning, burned}*. Figure 1 (a) shows an example of a screen shot from the DEVS-FIRE-based wildfire simulation. Green shows the unburned cells under a different fuel bed condition; red shows the burning cells; black shows the burned cells. Each fire cell is surrounded by other 8 cells in directions of *{north-west, north, north-east, east, south-east, south, south-west, west}* unless it hits the cell space boundary which might be less than 8. Once the simulation is triggered*, {unburned}*cells that receives an igniting signal change the fire states to *{burning};* burning cells calculate the spread direction and speed, and send igniting signals to their neighbor cells, and after a period of burning time they change the states to *{burned}*; burned cells are always passive for the rest of the simulation time and the states of them are no longer change.

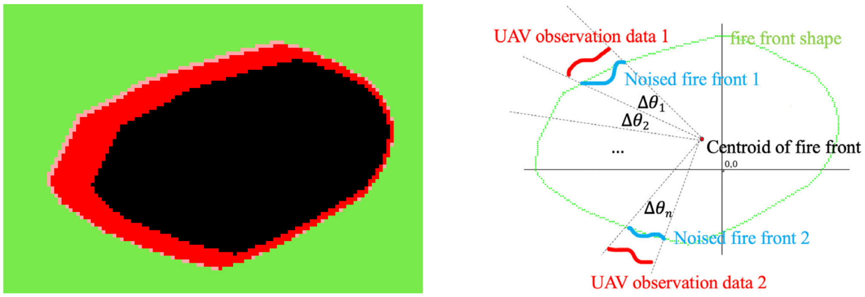

Figure 1: (a) DEVS-FIRE visualization; (b) Process noise illustration

The fire simulation follows the fire physical behavior without environment error. One important process in data assimilation is to process state noise treatment to have adjustment on the system states if environment errors appear. Thus, as shown in the system transition model (1), processing state noise γ_t on the fire states is an important step. In our work, state noise (or process noise) treatment is defined as additional segments of fire cells added to the original fire state and replace the original fire states. The state noise is a consistent segment, and it forms an enclosed shape with the original fire shape to form the new "noise fire shape" representing the processed fire states. The purpose of generating noise fire states is to make adjustment on the imprecise fire states caused by DEVS-FIRE model when the system contains error. Comparing to the fire states directly generated by DEVS-FIRE model, noise fire states holds potential to more accurately express the fire states.

The process noise is applied by referring to observation data. The detailed steps of adding state noise to the fire states are introduced in a prior work [19]. Considering the UAV observation data holds special properties, we apply similar steps of adding state noise but considering the local-spatial property of UAV observation data and applying gaussian-distributed noise. We extract the fire front, the outmost layer of the burning cells, out of the fire states, and divide the fire front into different segments for processing state noise. We apply local-spatial process noise. In this way, only the fire states in the segments with observation data detected will be adjusted by adding states noise to accommodate the error caused by the system. An appropriate utilization of UAV observation data is important to be established. In our work UAV observation data might be detected at any direction of the simulation cell space. We utilize its "angle-based division" on the fire front, which is defined as the angles that are evenly divided starting at the centroid of the fire front shape and forms n angle scopes. The direction converted to degrees are defined as: north – 0 degrees; east – 90 degrees; south – 180 degrees; west – 270 degrees. To process state noise, a special treatment we apply is that we only add the states noise on the fire front segments at the angle scopes have pieces of UAV observation data detected. For example in Figure 1 (b), $\Delta\theta_1 \dots \Delta\theta_n$ represent n different angle scopes, in which only $\Delta\theta_1$ and $\Delta\theta_n$ detect UAV observation data pieces (shown in red lines). Thus, we only process states noise for the fire front segments that fall in angle scope $\Delta\theta_1$ and $\Delta\theta_n$ to form the noise fire segments (shown in blue lines) and the original fire segments (shown in the green lines) are replaced by the noise fire segments. In this way, the state transitions are in reasonable control. The fire front segments with no state noise applied are fully driven by the DEVS-FIRE model until if any observation data is detected in the later steps. The segments processed with states noise becomes a potential adjustment on the fire front shape.

3.3 The measurement model

The measurement model describes how the UAV-based sensing works. To develop the measurement model, it is important to understand how the UAV-based observation data is obtained. In this work, we consider fire front sensing using UAV's thermal camera, and thus the observation data is the fire front location data. The thermal camera of a UAV takes thermal images that use different colors to represent different relative temperatures of objects. For the wildfire application, fire fronts are the locations where the fire is actively burning and thus have higher temperatures. These high temperature locations are reflected in the thermal images as brighter colors. Thus, given a thermal image, one can identify the bright pixels in the thermal images that correspond to the fire front locations.

An important feature of the UAV-based sensing is that the thermal camera has limited field of view, and thus can cover only a relatively small region of a large wildfire. This means when the UAV flies around the fire area, the thermal images from the UAV at different time instants cover different parts of the fire area. Our work uses a position-based approach to compute the location of the fire area covered by the UAV's thermal images. Figure 3(a) illustrates this approach, where the light blue area represents the area covered by a thermal image taken by the UAV's thermal camera. This approach assumes the fire area is a flat area on the ground. It also assumes the UAV's position $(x_{uav}, y_{uav}, z_{uav})$ and the orientation angle of the thermal camera at any time instant is known, where z_{uav} is the flying height of the UAV. The position $(x_{uav}, y_{uav}, z_{uav})$ can be obtained from the UAV's GPS sensor; the orientation angle of the thermal camera can be derived from the Roll, Pitch, and Yaw angles of the UAV as well as the mount angle of the thermal camera on the UAV's body. For simplicity, Figure 3(a) only illustrates the computation on the x-z plane, where θ is the corresponding angle of the thermal camera's orientation on the x-z plane. Let x_{fov} be the x coordinate of the center of the UAV's field of view on the ground. Then $x_{fov} = x_{uav} + z_{uav} * tan\theta$. Similarly, the y_{fov} can be computed. The computed (x_{fov}, y_{fov}) represents the center of the field of view on the ground that is covered by the thermal image. We note that in order to obtain high quality observation data it is better to make the orientation angle θ small, i.e., making the thermal camera face downward.

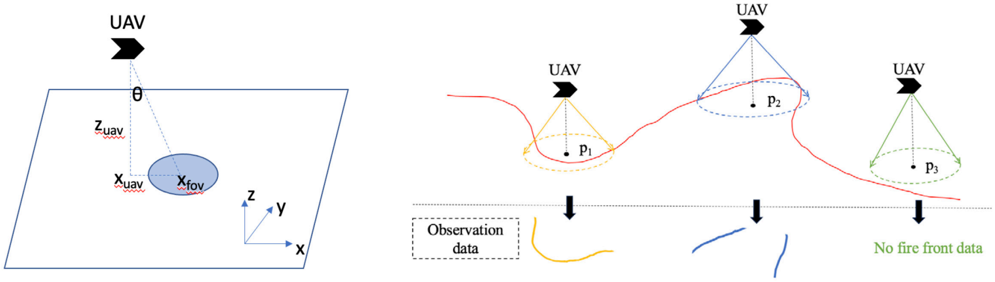

Figure 3: (a) UAV field of view's location; (b) UAV's observation data at different time

After the location of the area covered by a thermal image is known, using the relative positions of the identified fire front pixels on the thermal image, one can then extract the specific locations the fire fronts captured by the thermal image. When a UAV flies around the fire area, its thermal camera takes thermal images in a certain frequency (e.g., every 1 second). Based on these thermal images, using the approach described above, the corresponding observation data are then computed.

The above description assumes perfect knowledge about the UAV's position and the facing angle of the thermal camera. In reality, GPS sensor has noise and the sensors measuring UAV's Roll, Pitch, and Yaw angles are not perfect either. This is especially true in a wildfire environment as there exist constant turbulence and updraft that brings challenges to UAV's flight. This imperfect knowledge brings inaccuracy to the computed fire front locations. For example, when x_{uav} is inaccurate, the computed x_{fov} would be inaccurate, which cause the extracted fire front locations to be shifted from the actual locations of the fire front. Similarly, the orientation angle errors of the thermal camera cause the extracted fire front locations to have a rotation error from the actual locations of the fire front. These errors need to be modeled by using the measurement noise.

Based on the above descriptions, we develop the measurement model (a) as a mapping function from the fire state to fire front locations that includes three steps. Step 1 is to compute the location of the thermal camera's field of view based on the UAV's position and the orientation angle of its thermal camera. In this work, we assume the orientation angle is facing down all the time, and uses the measurement noise (described in Step 3) to account for inaccuracy of the orientation angle. Step 2 is to extract the fire front segments within the field of view based on the fire state. Step 3 is to add measurement noise ζ_t. Specifically, we model ζ_t as a Gaussian noise with two terms: 1) a Gaussian-based shift noise that shifts the extracted fire front locations with a Gaussian-based distance; and 2) a Gaussian-based rotation noise that rotates the fire front locations (around the center of the field of view) with a Gaussian-based angle.

Figure 3(b) illustrates the UAV's observation data at three different time instances as the UAV flies in the fire area. In the figure, the red line represents the fire front; the dotted circles represent the UAV's field of view on the ground, and the fire front measurement data are shown by the fire segments below the red arrows. When the UAV is at location P1, its field of view covers a continuous segment of the fire front. When the UAV is at location P2, its field of view covers two segments of the fire front due to the limited field of view. When the UAV is at location P3, its field view cover no fire front. We note that due to the noise factors, the observation data at each position is not the same as the fire segment within the corresponding field of view due to the measurement noise that are added. Also note that at location P3, the observation data has no fire front data.

4 PARTICLE FILTER-BASED DATA ASSIMILATION

Particle filter (PF) is utilized as the data assimilation method in our work. The PF algorithm is a sample-based data assimilation method used on non-linear system calibration based on Bayesian inference and stochastic sampling techniques. The purpose of using PF is to estimate posterior density for all particles based on system states and observation data then make the best selection on the particles. PF includes sampling, weight calculation function, resampling as the main steps. In our work, sampling , weight calculation functions and resampling are all related to wildfire simulation and UAV observation data. In the sampling phase, system transition model is applied for transiting the fire states with process noise to generate diversity of fire states. The weight calculation function is applied for calculating and assigning the importance weights based on the fire states and UAV observation data. PF proceeds resampling based on the importance weights to select and keep offspring of particles of the sampled fire states.

PF is processed by steps based on a specific time interval. In every PF time interval, PF uses all sensor data within the time interval to process the data assimilation. If sensor data is retrieved constantly, the longer the time interval is, the more sensor data is within the time interval. However, this comes an issue that the sensor data within a time interval does not belong to the same time snapshot. Thus, the existence of time lag issue in the sensor data impacts the PF performance. To properly utilize the UAV observation data, a special treatment on the PF time interval is approached by significantly reducing the PF step time interval that the PF model has a much higher frequent adjustment of the fire shapes. The UAV observation data in a very short PF time interval could be considered approximately within the same time snapshot. Therefore, the PF model works more reasonably on the UAV observation data and reduces the impact of time lag issue.

The weight calculation function in PF is utilized to compare the system states with the observation data and provides importance weights for particles. Trajectory similarity methods, which are used to measure the overlap of two geography paths, are effective to assess the difference between UAV observation data and the fire front. One Way Distance (OWD) trajectory similarity algorithm is utilized in the weight calculation function. The OWD is defined as the integral of the distance from discrete points of a trajectory T_1 to another trajectory T_2 [26], which is equivalent to the area shaped by the discrete points of T_1 to T_2. Figure 4 illustrates how the OWD works. In the figure, the black line represents the trajectory T_1; red points p_1, …, p_5 represent the discrete points that are averagely sampled from T_1. Area s_1, …, s_5 represents the area that shaped by the discrete points of T_1 with trajectory T_2 in different segments (represented in different colors), which are averagely divided to the same number with the discrete points in trajectory T_1. The OWD trajectory similarity from T_1 to T_2 is the summation of area $D_{owd}(T_1, T_2) = \sum_i^5 s_i$.

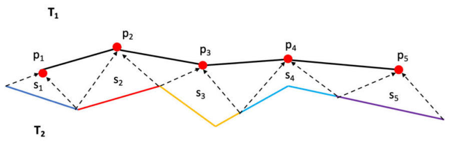

Figure 4: Illustration of OWD trajectory similarity

Similar to the process noise, the weight calculation function is also applied the special treatment due to local-spatial property of UAV observation data. Angle-based division of the fire shape can also be applied on calculating the trajectory similarity. Due to the UAV observation has a property of being shifted, there might be pieces of UAV observation exist and forms a cluster within an angle scope. For example, in Figure 5 red lines represent UAV observation data, and in angle scope $\Delta\theta_1$ there are several pieces of UAV observation being captured. Therefore, we sample the centroid of the clusters of UAV observation in each angle scope. The trajectory similarity D_{owd} in each angle scope is the area that formed by the centroids of UAV observation clusters and the corresponding fire front segments. In each valid angle scope $\Delta\theta_i$, trajectory similarity D_i is calculated as area s_i. Figure 5 as an example, in angle scope $\Delta\theta_1$ the sampled centroid point of UAV observation cluster is c_1; trajectory similarity D_1 is calculated as area s_1. Similarly in $\Delta\theta_n$ the sampled centroid is c_n and trajectory similarity D_n is calculated as area s_n. For each particle i, the fire front shape is divided by n angle scopes $\boldsymbol{\theta^i} = [\Delta\theta_1 \dots \Delta\theta_n]$, the trajectory similarity $\boldsymbol{D_{owd}}^{\boldsymbol{i}} = [s_1 \dots s_n]$ represents a set of areas that calculated in each angle scope and it is utilized to generate OWD-based independent weights $\boldsymbol{w^i_\theta} = \left[w_{\theta_1} \dots w_{\theta_n}\right]$ based on gaussian distribution. The particle's importance weight is calculated as a cumulative multiplication weight $w^i = \prod_0^n w_{\theta_n}$.

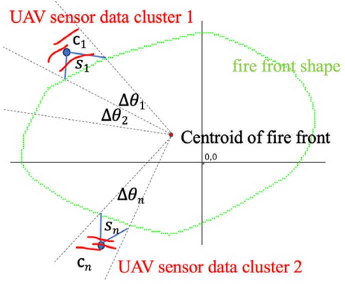

Figure 5: Illustration on the OWD calculation based on UAV observation data

The description of the proposed PF algorithm is provided in the Table 1.

Table 1: Implementation of PF algorithm

Algorithm 1: Particle filter in wildfire simulation using UAV observation data

```
1. Initialization, time step $k = 0$
    • For $i = 1, \dots, N$, sample $fire_0^{(i)}$ from DEVS-FIRE as the initial fire states.
    • Increment the time step by setting $k = 1$
2. Sampling:
    • For each fire state in $\{fire_{k-1}^{(i)}\}_{i=1}^N$, draw a sample using the system transition model described
      in section 3 including DEVS-FIRE model and process noise model, and generate noise fire
      states, donated by $\{\overline{fire}_{k-1}^{(i)}\}_{i=1}^N$.
3. Importance weight calculation:
    • For each noise fire states $\{\overline{fire}_{k-1}^{(i)}\}_{i=1}^N$, calculate its importance weight $\{\overline{w}_k^{(i)}\}_{i=1}^N$, according
      to UAV observation data using OWD.
    • Normalize the importance weights: $w_k^{(i)} = \frac{\bar{w}_k^{(i)}}{\sum_{j=1}^{N} \bar{w}_k^{(j)}}$
4. Resampling
    • Resample noise fire states $\{\overline{fire}_{k-1}^{(i)}\}_{i=1}^{N}$ according to $\{w_k^{(i)}\}_{i=1}^{N}$.
5. Update the time step and loop the previous steps (except initialization step):
    • Set time step $k$ to $k + 1$, then go to step 2.
```

5 EXPERIMENT DESIGN AND EXPERIMENT RESULTS

To evaluate the performance of PF, identical twin experiment [23] is applied in our work. Identical twin experiment is a methodology testing the simulation outputs between a perfect system and a system with imperfect data. An identical twin experiment includes a set of two experiments: a real experiment and a twin experiment. These two experiments run under different system parameters and generate different system states. A data assimilation process is applied only on the twin experiment to process states adjustment and generates filtered states. In our work, true states, twin states, and filtered states indicate real fire states, twin fire states, and PF fire states respectively. PF is aimed to integrates the UAV observation data to adjust and re-predict the fire states to improve the simulation results when system error appears.

To validate the effectiveness of the PF-based data assimilation using UAV observation data on wildfire simulation, we design the experiments such that the twin weather data is different from the real weather data. The experiment results demonstrate the effectiveness of the proposed PF approach in aspects of dealing with both UAV observation noise and system error. In the two sets of experiments, we used DEVS-FIRE-based UAV sensing model to simulate and generate the UAV observation data. We applied coordinated single UAV path planning that the UAV always follows the fire front to retrieve the fire front information. The experiments are deployed on a 200*200 cell space with each cell is in size of 30m. The cell space is filled up with 3 different types of fuel which are evenly distributed in strips that the fire spreading rate are different on the 3 fuels.

In the first set of experiments, we validate that the proposed PF approach has great effect when the UAV observation data contains noise. The fire is ignited at cell (150, 170) for 13000 seconds of simulation time. Having the north direction as 0 degree, the real wind data is in speed of 22 ± 5 m/s and in direction of 43 ± 25 degree, and the twin wind data is in speed of 20 ± 5 m/s and in direction of 57 ± 25 degree. Figure 6 (a) and (b) show the real fire front (green) and the twin fire front (yellow). The UAV starts being deployed at 240s after the fire is ignited and data retrieving range is at 3 cells radius with a 1 second reporting rate.

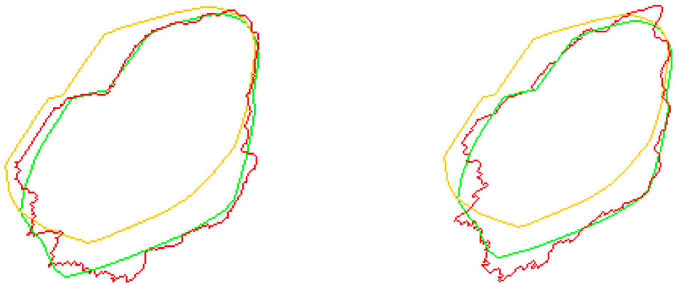

Figure 6: Experiment set 1 (a) Fire front shapes for UAV data with small noise; (b) Fire front shapes for UAV data with large noise. (Green: Real fire front; Yellow: Twin fire front; Red: PF fire front)

The experiment is designed to demonstrate effectiveness of the proposed PF approach on various UAV observation noise. We validated two sets of UAV observation data in small observation noise and big observation noise, which are shown in pink cells in Figure 7 (a) (b) respectively. We applied a gaussian noise with smaller standard deviation and a small rotation on the UAV for small noise observation; while a larger gaussian standard deviation and rotation for big noise observation. We applied 200 particles with a 30s PF time interval on the twin experiments. The visualized experiment results are provided in Figure 6 that the PF filtered fire fronts are shown in red. The results shows that in both experiments, the twin fire front shapes are properly adjusted and the PF filtered fire front shapes have a good match with the real fire front shape. However, due to the larger noise variance, the PF filtered fire shape based on large UAV sensor noise is less accurate and has more mismatched fire segments on the real fire shape.

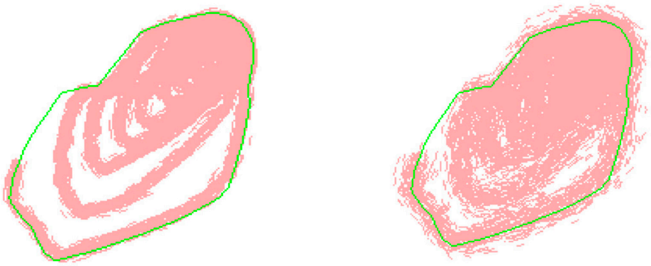

Figure 7: Experiment set 1: (a) UAV observation data with small noise; (b) UAV observation data with big noise. (Green: Real fire front shape; Pink: UAV observation data)

A quantitative result is provided to analyze the performance of PF over time in Figure 8. As a comparison, we applied convex hull algorithm as a base method. Without system or states information, convex hull straightforwardly connects the centroid of UAV observation data within PF time intervals and reconstructs the fire front. For both PF and convex hull, after a fire front is reconstructed, the fire front is assigned *{burning}* states and the cells within the fire front are assigned *{burned}* states. Mismatched cell rate is used as the error metric after the fire front is reconstructed to describe the number of mismatched *{burning/burned}* cells divided by the real fire's *{burning/burned}* cells. Figure 8 (a) shows that PF is effective dealing both small and big observation noise that the two experiment results have similar decreasing trends. The mismatched cell rates constantly decrease over time and eventually reduce to around 10%. Figure 8 (b) shows that with the simulation time increases, the number of mismatched cells increase at the beginning as the fire are spreads larger but eventually become convergent. Comparing PF and convex hull, when the observation noise is small, convex hull performs better but shows an unstable behavior with the time increases. When the observation noise is large, PF shows an obvious advantage on the stable performance and accuracy. Plus, PF can show stable performances in both noise scenarios. Since convex hull simply just takes the maximum coverage of the observation data as the fire shape, the more precise the observation data is, the more accurate convex hull can reconstruct the fire shape. However, if the observation data gives more noise information, convex hull no longer well estimates the coverage of fire shape; while PF can still consistently and stably outperform in adjusting the fire states during time.

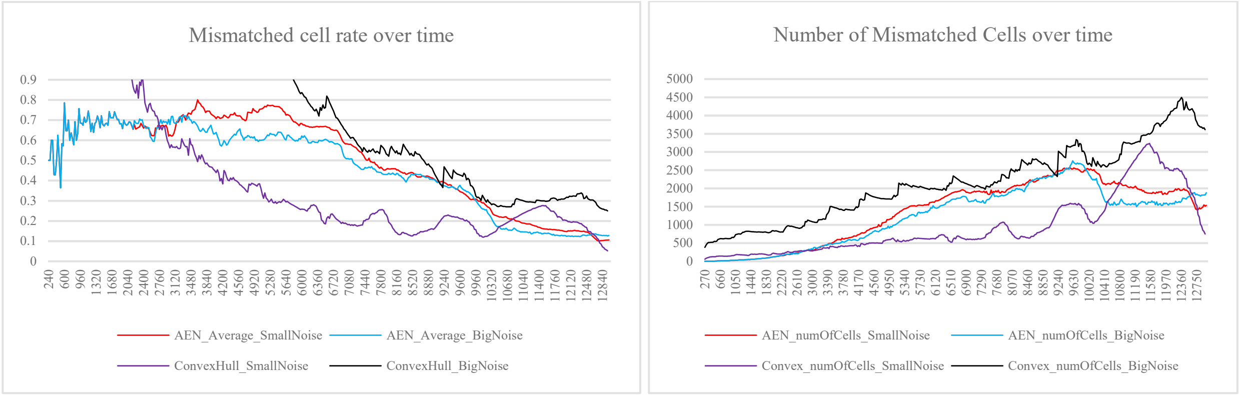

Figure 8: Experiment 1 quantitative results: (a) Mismatched cell rate; (b) Number of Mismatched cells

In the second sets of experiments, we validate the effectiveness of the proposed PF approach on different system error. In our experiments, the weather difference between real fire and twin fire is considered to be the system error. In experiment 2, we increase the weather difference between real fire and twin in order to generate larger system errors. We design 3 sets of experiment to demonstrate that the proposed PF approach is effective in small, middle and large weather data error incorporating noise UAV observation data. In the 3 sets of experiments, the fire is ignited at cell (150, 50) for 12000 seconds. The real wind data is set up in speed of 22 ± 7 m/s and in direction of 132 ± 25 degree. For the twin wind speeds, small, middle, large weather error is set in speed of 20 ± 5 m/s, 18 ± 5 m/s and 16 ± 5 m/s respectively. The twin wind directions are all set to be the same in 147 ± 25 degree. The real fire front and the twin fire shapes are shown in Figure 9 (a). A single UAV is deployed at 300s after the fire is ignited the UAV observation range is also at 3 cells radius with also a 1 second reporting rate. The UAV observation noise we applied as the big noise corresponding to experiment 1 and is shown in Figure 9 (b).

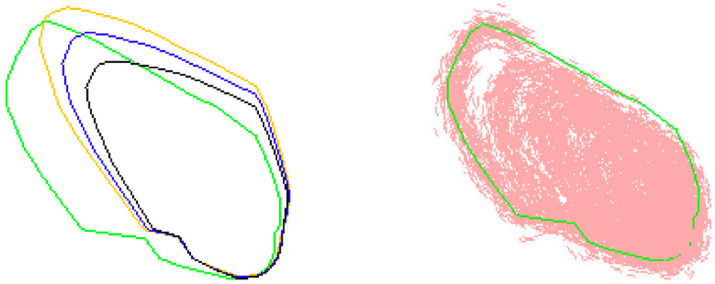

Figure 9: Experiment set 2: (a) Fire front shapes in different weather conditions. Green: Real weather; Yellow: Small weather error; Blue: Middle weather error; Black: Big weather error. (b) UAV observation data with noise (Green: Real fire front; Pink: UAV observation data).

We also applied 200 particles with 30s PF time interval for each twin experiment. The visualized results of each PF fire front are shown in Figure 10 in red. The results shows that PF is effective adjusting the fire front in all 3 twin weather conditions but with the weather error increases the PF fire front shows in more unstable shapes. The quantitative results are shown in Figure 11. From Figure 11 (a) and (b) we can see that the PF has great performance of reducing the mismatch cell rate over time in all weather conditions. However, due to the reason that when the weather error increases, PF needs to process larger state noise for the noise fire states to accommodate the weather difference then have a potential to reduce the state-to-observation difference. Thus, results of small weather error and middle weather error are both good since these two weather errors are considered relatively small. The results for big weather error shows that a big weather error has impact on reducing the PF performance.

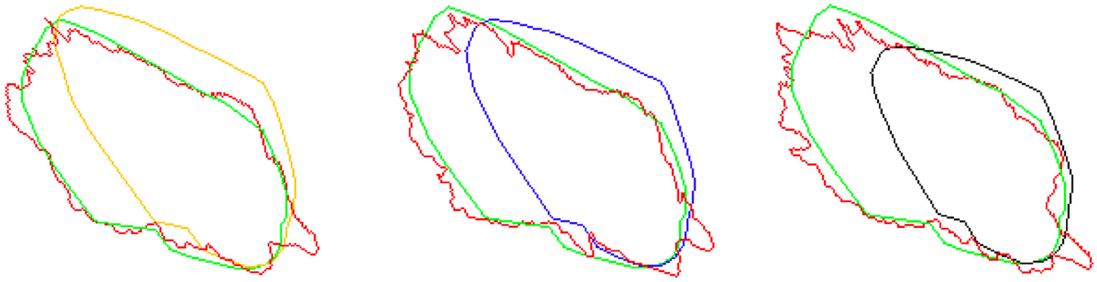

Figure 10: Experiment set 2 PF fire front: (a) Twin small weather error; (b) Twin middle weather error (c) Twin big weather error. (Green: Real fire front; Red: PF fire front; Other: Twin fire fronts)

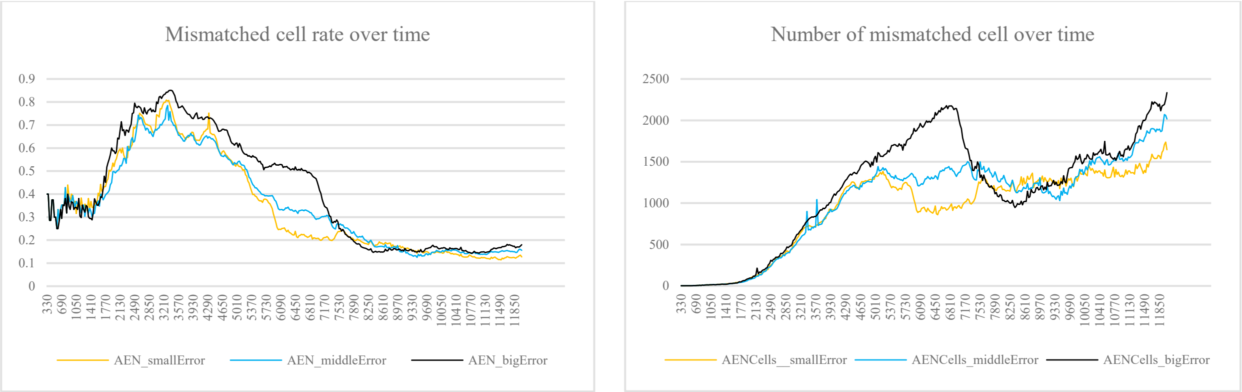

Figure 11: Experiment 2 quantitative results: (a) Mismatched cell rate; (b) Number of Mismatched cells

6 CONCLUSION AND DISCUSSION

In conclusion, in this paper our contribution includes formulating the wildfire simulation problem with state transition and UAV-based observation, and proposing a particle filtering approach which effectively assimilating the UAV based observation data to make adjustment on the fire states. The proposed particle filter approach properly handles with the constraints of UAV observation data being noisy and partially and utilizes the advantage of the high collection frequency to enhance the PF performance. The proposed particle filter approach also demonstrates the effectiveness when the simulation contains system error.

The possible future study needs to be focus on solving the constraints of the current PF approach, including different deployments of UAV data collection and improvement on PF performance. For example, when UAV observation data are collected in different path plannings, it is possible to happen that UAV data is collected with a big time gap or even with data loss. In this way, system transition model needs to be revised to deal with the scenarios and weight generation function needs to be properly developed to evaluate the particles if there is no UAV observation. Different fire front division strategies, for example space-grid based division, become feasible approaches to be applied on process noise and weight calculation function which might generate more precise evaluations, while the computational cost and the complexity of implementation might become tradeoffs. PF parameter estimation and particle selection both become potential directions help improve the PF performance to lead more accurate and more stable experiment results, which is also a popular topic in general PF studies.

REFERENCES

[1] National Interagency Fire Center, https://www.nifc.gov/.

[2] Ntaimo, L., Hu, X., & Sun, Y. (2008). DEVS-FIRE: Towards an integrated simulation environment for surface wildfire spread and containment. *Simulation*, *84*(4), 137-155.

[3] Hu, Xiaolin, and Lewis Ntaimo. "Integrated simulation and optimization for wildfire containment." *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 19.4 (2009): 1-29.

[4] Finney, M. A. (1998). *FARSITE, Fire Area Simulator--model development and evaluation* (No. 4). US Department of Agriculture, Forest Service, Rocky Mountain Research Station.

[5] Andrews, P. L. (2007, October). BehavePlus fire modeling system: past, present, and future. In *Proceedings of 7th symposium on fire and forest meteorology* (pp. 23-25). Boston MA:: American Meteorological Society.

[6] Rothermel, R. C. (1972). *A mathematical model for predicting fire spread in wildland fuels* (Vol. 115). Intermountain Forest & Range Experiment Station, Forest Service, US Department of Agriculture.

[7] Xu, Guang, and Xu Zhong. "Real-time wildfire detection and tracking in Australia using geostationary satellite: Himawari-8." *Remote Sensing Letters* 8.11 (2017): 1052-1061.

[8] Sifakis, Nicolaos I., et al. "Wildfire detection and tracking over Greece using MSG-SEVIRI satellite data." *Remote sensing* 3.3 (2011): 524-538.

[9] Syifa, Mutiara, Mahdi Panahi, and Chang-Wook Lee. "Mapping of post-wildfire burned area using a hybrid algorithm and satellite data: The case of the camp fire wildfire in California, USA." *Remote Sensing* 12.4 (2020): 623.

[10] Bailon-Ruiz, Rafael, and Simon Lacroix. "Wildfire remote sensing with UAVs: A review from the autonomy point of view." *2020 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2020.

[11] Akhloufi, M. A., Couturier, A., & Castro, N. A. (2021). Unmanned aerial vehicles for wildland fires: Sensing, perception, cooperation and assistance. *Drones*, *5*(1), 15.

[12] Liu, Y., Zheng, C., Liu, X., Tian, Y., Zhang, J., & Cui, W. (2023). Forest Fire Monitoring Method Based on UAV Visual and Infrared Image Fusion. *Remote Sensing*, *15*(12), 3173.

[13] Messina, Gaetano, and Giuseppe Modica. "Applications of UAV thermal imagery in precision agriculture: State of the art and future research outlook." *Remote Sensing* 12.9 (2020): 1491.

[14] Lee, Seunghyeon, Youngkeun Song, and Sung-Ho Kil. "Feasibility analyses of real-time detection of wildlife using UAV-derived thermal and RGB images." *Remote Sensing* 13.11 (2021): 2169.

[15] Hu, X., Ge, M., Gowravaram, S., Chao, H., & Xin, M. (2023). Prescribed Fire Simulation with Dynamic Ignitions Using Data from UAS-based Sensing. *Journal of Simulation*, 1-13.

[16] Li, X., Gao, H., Zhang, M., Zhang, S., Gao, Z., Liu, J., ... & Sun, L. (2021). Prediction of Forest fire spread rate using UAV images and an LSTM model considering the interaction between fire and wind. *Remote Sensing*, *13*(21), 4325.

[17] Hu, X., Ge, M., Gowravaram, S., Chao, H., & Xin, M. (2023). Prescribed Fire Simulation with Dynamic Ignitions Using Data from UAS-based Sensing. *Journal of Simulation*, 1-13.

[18] Djuric, P. M., Kotecha, J. H., Zhang, J., Huang, Y., Ghirmai, T., Bugallo, M. F., & Miguez, J. (2003). Particle filtering. *IEEE signal processing magazine*, *20*(5), 19-38.

[19] Xue, H., Gu, F., & Hu, X. (2012). Data assimilation using sequential Monte Carlo methods in wildfire spread simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *22*(4), 1-25.

[20] Y. Long, X. Hu, Spatial Partition-Based Particle Filtering for Data Assimilation in Wildfire Spread Simulation, *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, Volume 3 Issue 2, Article No. 5, August 2017

[21] Hu, X. (2023, May). A Tutorial on Bayesian Sequential Data Assimilation for Dynamic Data Driven Simulation. In *2023 Annual Modeling and Simulation Conference (ANNSIM)* (pp. 680-695). IEEE.

[22] X. Hu, Data Assimilation for Simulation-Based Real-Time Prediction/Analysis, Proc. 2022 Annual Modeling and Simulation Conference (ANNSIM 2022), doi: 10.23919/ANNSIM55834.2022.9859329, 2022

[23] Xiaolin Hu, Dynamic Data-Driven Simulation: Real-Time Data for Dynamic System Analysis and Prediction, World Scientific, April 2023

[24] Mandel, Jan, et al. "Data assimilation for wildland fires." *IEEE Control Systems Magazine* 29.3 (2009): 47-65.

[25] Srivas, Thayjes, et al. "Wildfire spread prediction and assimilation for FARSITE using ensemble Kalman filtering." *Procedia Computer Science* 80 (2016): 897-908.

[26] Lin, B., & Su, J. (2008). One way distance: For shape based similarity search of moving object trajectories. *GeoInformatica*, *12*, 117-142.