

## **AN EFFECTIVE PROPOSAL DISTRIBUTION FOR SEQUENTIAL MONTE CARLO METHODS-BASED WILDFIRE DATA ASSIMILATION**

Haidong Xue  
Xiaolin Hu

Computer Science Department  
Georgia State University  
Atlanta, GA 30303, USA

### **ABSTRACT**

Sequential Monte Carlo (SMC) methods have shown their effectiveness in data assimilation for wildfire simulation; however, when errors of wildfire simulation models are extremely large or rare events happen, the current SMC methods have limited impacts on improving the simulation results. The major problem lies in the proposal distribution that is commonly chosen as the system transition prior in order to avoid difficulties in importance weight updating. In this article, we propose a more effective proposal distribution by taking advantage of information contained in sensor data, and also present a method to solve the problem in weight updating. Experimental results demonstrate that a SMC method with this proposal distribution significantly improves wildfire simulation results when the one with a system transition prior proposal fails.

### **1 INTRODUCTION**

Assimilating observation data into simulation models of complex systems is an effective method to improve simulation results. For most of complex systems, their infinite states, non-Gaussianity or non-linearity excludes the usage of optimal tools, such as Hidden Markov Models (HMM) or Kalman Filter (KF) based methods. Among the approximation tools, Sequential Monte Carlo (SMC) (Cappe, Godsill, and Moulines 2007; Doucet and Johansen 2011) methods have no assumption on involved distributions and transition functions; also, the computation load does not increase with time, and the system state estimation error decreases with  $O(1/N)$ , where  $N$  is the number of particles. With these features, SMC methods then become an effective and efficient data assimilation tool set and succeed in many applications, for example, Salamon and Feyen (2009), van Leeuwen (2010), and Matgen et al. (2010), and our previous work, wildfire data assimilation as in Xue, Gu, and Hu (2012).

In wildfire SMC data assimilation, the proposal distribution (also referred as “proposal” for short) is often set the same as the system transition distribution (also referred as “system prior”), as in the bootstrap filter (Gordon, Salmond, and Smith 1993) and condensation filter (Isard and Blake 1998). This choice largely simplifies the weight updating; however, it limits the effectiveness of the data assimilation when a system prior is far away from the optimal proposal distribution. The density values of proposal distributions and system transition distributions are both needed to correct importance weights for a SMC method, and when choosing the transition distribution as the proposal distribution, they cancel each other; as a result, only likelihoods are needed in weight updating steps. Moreover, it is sometimes the only choice since a wildfire model is usually so complex that its analytical expressions are unknown, leading to difficulties to obtain density values from those distributions. The problem coming with using a system prior as a proposal is that observation data do not play roles in a sampling step, that is, a SMC method

loses the chance to use observation data to early generate particles with high likelihoods. When a real fire is largely different from a simulated result, this SMC data assimilation then fails.

In this work, we present a proposal distribution that sufficiently utilizes sensor readings in a sampling step, and is then much more effective than the system prior proposal. In this proposal distribution, a state is firstly generated from the system prior, and each sensor cluster then generates a local state for its monitored area based on its readings and the system prior generated state. A final sample is combined from those generated states according to their confidence. In weight updating step, we employ kernel estimation as in Rosenblatt (1956) to estimate the needed density values. Identical twin experiments are performed to test the effectiveness of this proposed proposal distribution.

## 2 RELATED WORK

The optimal proposal distribution of a SMC method is identified in literature. It is the one minimizes the variance of importance weights as discussed in Cappe, Godsill, and Moulines (2007) and Doucet and Johansen (2011). According to the core algorithm of a SMC method, one can apparently see the following proposal minimize the importance weight variance, so it is the optimal proposal distribution:

$$q^{opt}(s_t|s_{t-1}, m_t) \propto p(s_t|s_{t-1})p(m_t|s_t). \quad (1)$$

If Equation (1) can be evaluated, a Markov Chain Monte Carlo (MCMC) method can be employed to draw samples from it like the “smoothing MH step” in van der Merwe et al. (2000), and “bootstrap filter with parameter regeneration” in Cappe, Godsill, and Moulines (2007). However, the large number of burn-in iterations becomes the weakness as stated in Godsill and Clapp (2001). In certain cases, it is even hard to know when a burn-in phase is finished. Although directly sampling from the optimal proposal distribution is often impossible, it sets the guideline of the design of proposal distributions.

The general methods to construct proposal distributions utilizing observation data are Kalman Filter based proposals. The extended Kalman Filter and unscented Kalman Filter are two typical choices for a SMC method as presented in De Freitas (2000) and van der Merwe et al. (2000). For each SMC particle, an extended or unscented Kalman Filter is maintained. In each sampling step of a SMC method, the Kalman Filter is first updated, and samples are then drawn from this filter in the SMC sampling step. Since the unscented Kalman Filter has relatively smaller errors than the extended Kalman Filter, it is often considered as a better choice than the other. However, in both of them, the proposal distribution is approximated by a Gaussian distribution, and this approximation is not always appropriate, especially in the context of wildfire simulations where the optimal fire proposal distribution is usually with strong non-Gaussianity. Even if a Gaussian approximation is sometimes acceptable, the Jacobians of extended Kalman Filter is hard to be obtained from a non-analytical wildfire simulation model; meanwhile, to update an unscented Kalman Filter proposal for one SMC particle,  $2n+1$  ( $n$  is the number of space dimensions) times of executions of a wildfire simulation are needed, and it is computationally not practical for wildfire simulations since the dimension is often super high and each execution consumes significant computing resources.

There are application-specific approaches presented in literature to construct proposal distribution by utilizing observation data. For video tracking problems, Pan and Schonfeld (2008) developed an algorithm to choose variances for proposals, and Lao, Zhu, and Zheng (2009) used video measurement confidence to adjust proposals; also, Zhai et al. (2009) employed multiple transition models and state partition to help construct proposals; Kyriakides, Morrell, and Papandreou-Suppappola (2008) incorporated motion constraints into proposals for multiple target tracking problems; Saha and Gustafsson (2012) utilized noise dependency to derive the optimal proposal for signal processing applications.

In this work, we propose an effective proposal distribution for wildfire data assimilation; we also propose to use the kernel estimation method to solve the density value estimation problem in weight updating steps. Potentially, this method can be applied to other complex systems similar to wildfires.

### 3 SMC BASED WILDFIRE DATA ASSIMILATION

#### 3.1 Sequential Monte Carlo Methods

The goal of a Sequential Monte Carlo (SMC) method for filtering is to approximate  $p(s_t|m_{0:t})$  for a dynamic-state model, in which the system states and their measurements are defined as:

$$\begin{aligned} s_t &= f(s_{t-1}, u_t), \\ m_t &= g(s_t, v_t), \end{aligned}$$

where  $s_{t-1}$  is the system state at time  $t - 1$ ,  $s_t$  is the system state at time  $t$ ,  $m_t$  is the measurement of  $s_t$ ,  $f$  is the system transition function,  $g$  is the measurement function, and  $u_t$  and  $v_t$  are random variables in  $f$  and  $g$ . As a result, system states evolve with time, forming a sequence  $\{s_t, t \in \mathbb{N}\}$ , and for each element of the state sequence there exists a measurement forming a measurement sequence  $\{m_t, t \in \mathbb{N}\}$ .

With the statistic kernel of  $u_t$ , given  $s_{t-1}$ , the next state  $s_t$  then becomes a random variable and is described by a probability distribution (usually referred as *system transition distribution*):

$$p(s_t|s_{t-1}).$$

At the same time, with the statistic kernel of  $v_t$ ,  $m_t$  is described as a probability distribution conditioned on  $s_t$  (usually referred as *measurement distribution*):

$$p(m_t|s_t).$$

When these two distributions are available, starting from an initial sample set of  $s_0$  (usually drawn from  $q(s_0|m_0)$ ), a SMC method iteratively draws samples and updates their importance weights to approximate the posterior distribution following the Bayesian inference through Sequential Importance Sampling (SIS) algorithm:

$$\begin{aligned} w'(s_t) &= w(s_{t-1}) \frac{p(s_t|s_{t-1})p(m_t|s_t)}{q(s_t|s_{t-1}, m_t)}, \\ w(s_t) &\approx \frac{w'(s_t)}{\sum_{i=1}^N w'(s_t^{(i)})}. \end{aligned} \tag{2}$$

where  $w$  is the normalized importance weight function,  $w'$  is the unnormalized importance weight function, and  $q(s_t|s_{t-1}, m_t)$  is the proposal distribution.

To solve the sample degeneracy problem, the resampling step (Rubin 1988) is introduced to SIS as in Gordon, Salmond, and Smith (1993), and forms the Sequential Importance Sampling with Resampling (SISR) algorithm as shown in Algorithm 1.

---

#### Algorithm 1 Sequential Importance Sampling with Resampling (SISR) for approximating $p(s_t|m_{0:t})$

---

##### 1. Initialization

Draw  $\{s_0^{(i)} | 1 \leq i \leq N\}$  from  $q(s_0|m_0)$ ;

**Repeat** for  $i = 1$  to  $N$

$$w'(s_0^{(i)}) \leftarrow \frac{p(s_0)p(m_0|s_0)}{q(s_0|m_0)},$$

**Repeat** for  $i = 1$  to  $N$

$$w(s_0^{(i)}) \leftarrow \frac{w'(s_0^{(i)})}{\sum_{i=1}^N w'(s_0^{(i)})},$$

##### 2. Iterative Sampling, Weight Updating and Resampling

**Repeat** for  $j = 1$  to  $t$

**Repeat** for  $i = 1$  to  $N$

###### Sampling (Prediction)

Draw  $s_j^{(i)}$  from  $q(s_j|s_{j-1}, m_j)$ ;

###### Weight Updating (Correction)

$$w'(s_j^{(i)}) \leftarrow w(s_{j-1}^{(i)}) \frac{p(s_j|s_{j-1})p(m_j|s_j)}{q(s_j|s_{j-1}, m_j)},$$

###### Resampling (Selection)

Draw  $\{s_j^{(i)} | 1 \leq i \leq N\}$  from  $\frac{w'(s_j^{(i)})}{\sum_{i=1}^N w'(s_j^{(i)})} \sum_{i=1}^N \delta(s_j - s_j^{(i)})$ ;

Set  $\{w(s_j^{(i)}) = 1/N | 1 \leq i \leq N\}$ ;

**Return**  $\{< s_t^{(i)}, 1/N > | 1 \leq i \leq N\}$ .

---

### 3.2 The DEVS-FIRE Model

The DEVS-FIRE model plays the role of system transition function ( $f$ ) in our wildfire data assimilation. It is a wildfire spread and suppression integrated model built on Discrete Event System Specification (DEVS) formalism (Zeigler, Kim, and Praehofer 2000). More details can be found in Hu, Sun, and Ntaimo (2012), and we only briefly introduce its wildfire spread modeling mechanism here.

A fire area is considered as a two-dimensional cellular automata consisting of rectangular fire cells whose dimensions depend on the resolution of GIS, fuel and terrain data. Each cell is coupled with its eight adjacent neighbors, and its fire spread behavior is modeled by the Rothermel model (Rothermel 1972). Similar to Finney (1998), an elliptical decomposition method is used to calculate the spread rates towards the eight neighbor cells as shown in Figure 1. Given a fire state and environmental parameters, DEVS-FIRE predicts a fire state for a future time point:

$$fire_{t+\Delta t} = DEVSFIRE(fire_t, \theta_t, \Delta t),$$

where  $fire_t$  and  $fire_{t+\Delta t}$  are the fire states at time  $t$  and  $t + \Delta t$ ,  $\theta_t$  is a vector containing all the other model inputs (information of terrain, weather and so forth),  $\Delta t$  is the time duration.

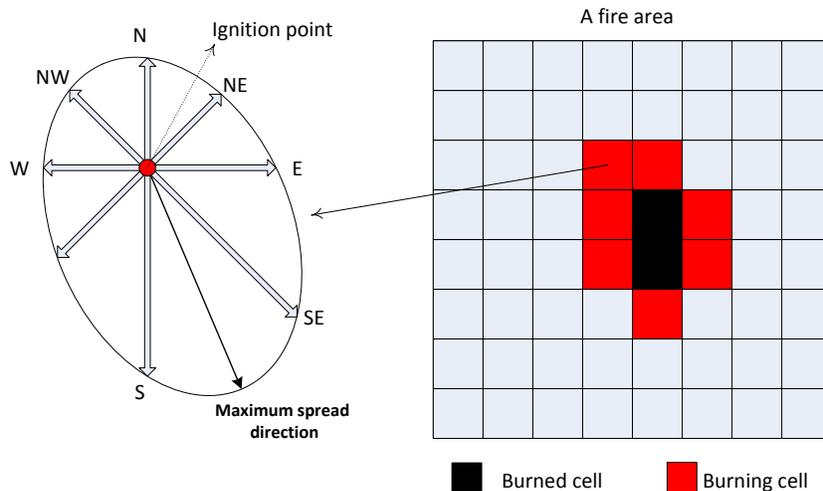


Figure 1: Fire spread decomposition of DEVS-FIRE.

### 3.3 SMC Based DEVS-FIRE Data Assimilation

Ground temperature sensors are assumed to be deployed in a real wildfire area. Considering their readings as measurements, and fire states as system states, a SMC method is then employed to improve the simulation results by approximating  $p(fire_t | m_{0:t})$ . When a set of sensor readings arrives, particles are updated through an iteration of Algorithm 1 as illustrated in Figure 2 (it comes from Xue, Gu, and Hu (2012)). Details of a bootstrap filter (a SMC method, where the proposal distribution is the same as the system transition distribution) based DEVS-FIRE data assimilation can be found in Xue, Gu, and Hu (2012), and its SMC data assimilation framework is the same as the one in this work.

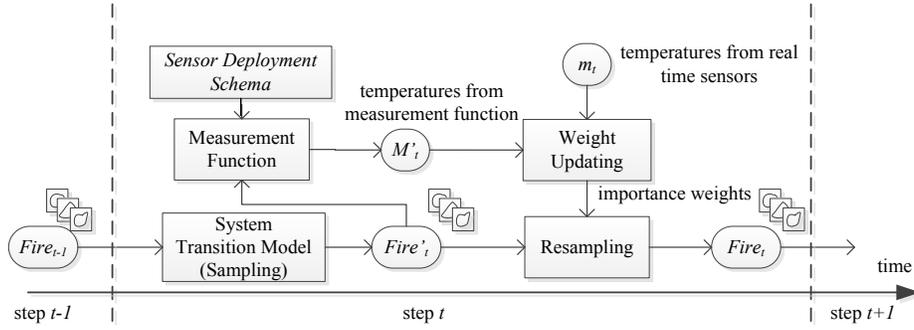


Figure 2: Architecture of SMC based DEVS-FIRE data assimilation.

Given a fire state in DEVS-FIRE, and the locations of each temperature sensor (that is, the sensor deployment schema), the reading of each sensor is modeled by a measurement function. When there is only one burning cell, the temperature on a sensor is

$$T = T_c e^{\frac{-d^2}{\sigma^2}} + T_a,$$

where  $T_c$  is the temperature rise above ambient temperature of the fire cell;  $T_a$  denotes the ambient temperature;  $d$  denotes the distance from the sensor to the burning cell;  $\sigma$  is a constant and is set to 50 (m) in our work. This model is based on the work of Mandel et al. (2008). A sensor temperature may be affected by multiple cells, and it records the highest temperature among the ones it receives. To compute  $T_c$  for a burning cell, a formula from Van Wagner (1973, 1975) is used:

$$T_c = 3.9FI^{\frac{2}{3}}/h,$$

where  $FI$  is the fireline intensity of the burning cell, and  $h$  is the height above ground. The fireline intensity is calculated from DEVS-FIRE.

Given this measurement function, the temperature on each sensor can be calculated, indicated as  $m'_t$ , and we model its difference to the real sensor readings ( $m_t$ ) as a multivariate distribution:

$$(m'_t - m_t) \sim MN(0, \Sigma),$$

where  $MN(0, \Sigma)$  is a multivariate Gaussian distribution with a zero mean vector and a covariance matrix  $\Sigma$ . It can be rewritten into:  $(m_t \sim MN(m'_t, \Sigma))$ . The measurement density in the form of multivariate Gaussian distribution is then given as:

$$p(m_t | fire_t) = \frac{\exp(-\frac{1}{2}(m'_t - m_t)' \Sigma^{-1} (m'_t - m_t))}{(2\pi)^{n_r/2} |\Sigma|^{1/2}}.$$

where  $n_r$  is the number of sensors. With this measurement distribution and a DEVS-FIRE based system transition model, if using a bootstrap filter, observation data of a wildfire can be assimilated to enhance the simulated prediction of a wildfire, but it suffers low performance in some cases, and we propose a more effective proposal distribution than the DEVS-FIRE based system transition distribution proposal.

#### 4 TEMPERATURE SENSOR READING ENHANCED PROPOSAL DISTRIBUTION

As illustrated in Equation (1), the optimal proposal distribution is proportional to  $p(s_t | s_{t-1})p(m_t | s_t)$ . Based on this fact, we propose a two-stage proposal distribution sampling algorithm: a sample is first drawn from a DEVS-FIRE based system transition distribution; then each sensor cluster generates a local state for its monitored area. Based on their confidence, a sample is finally combined from those states.

##### 4.1 The DEVS-FIRE Generated Fire States

After the randomness of DEVS-FIRE parameters is modeled, a system transition distribution ( $p(fire_t | fire_{t-1})$ ) is then constructed (for example, in Xue, Gu, and Hu (2012)), the randomness is

modeled as a fire front noise). The first step of the proposed sampling algorithm is to draw a sample from this distribution.

In this work, to clearly illustrate the proposed method, we assume the randomness is only caused by the random move of wind speed and wind directions. Consequently, to sample a fire state  $fire_t^{(i)}$  from  $p(fire_t|fire_{t-1})$ , we first draw samples for wind speed move and wind direction move as:

$$SpeedMove \sim N(Speed_{t-1}, var_{speed}),$$

$$DirectionMove \sim N(Direction_{t-1}, var_{direction}),$$

where  $N$  is the Gaussian distribution,  $Speed_{t-1}$  and  $Direction_{t-1}$  are the wind speed and direction at time  $t - 1$ ,  $var_{speed}$  and  $var_{direction}$  are the variances of these two Gaussian distributions. With the samples of wind speed move and direction move, samples for wind speed and direction at time  $t$  can be obtained, and  $fire_t^{(i)}$  is then calculated through the DEVS-FIRE model.

#### 4.2 Sensor Cluster Generated Local Fire States

A sensor cluster is defined as the sensors having overlapped monitored area. Given a sensor cluster with temperature readings, we first choose a set of “hot” sensors that are the ones have a higher temperature than a predefined threshold  $T_{hot}$  and a set of “cool” sensors that have a temperature lower than a predefined  $T_{cool}$ . Within a radius  $r_{hot}$ , each “hot” sensor defines a round shape, their union defines the “hot” area of this sensor cluster; similarly, with  $r_{cool}$ , “cool” sensors define the “cool” area of the cluster. Subtracting the “cool” area from the “hot” area, a possible fire area is then obtained.

We consider  $r_{hot}$  and  $r_{cool}$  as random variables, and draw samples from their distributions before the generation of possible fire areas. The mean of  $r_{hot}$  is proportional to the temperature of a sensor; the mean of  $r_{cool}$  is predefined.

To sample a fire state for the monitored area of a sensor cluster, both the possible fire area and the cool area are used. Given a possible fire area, each sensor ignites the intersection of its local hot area and the possible fire area. When igniting, the DEVS-FIRE generated system state plays an important role: if a sensor’s local possible fire area is not ignited in the DEVS-FIRE generated state, the sensor ignites all its local possible fire area; if the area has already been ignited, the sensor keeps it unchanged. Given a cool area, all the inside burning cells are turned off.

---

#### Algorithm 2 Sensor enhanced proposal distribution sampling algorithm for wildfire data assimilation

---

##### 1. Sampling from DEVS-FIRE

Draw  $fire_t^{(i)}$  from  $p(fire_t|fire_{t-1})$ ;

##### 2. Sampling from sensors, and selecting local states

**Repeat** for each sensor cluster  $k = 1$  to  $n_s$ , and its monitored area  $D_k$

Separate  $fire_{t,D_k}^{(i)}$  from  $fire_t^{(i)}$

Construct  $fire_{t,D_k}^{\prime\prime(i)}$  by igniting the possible fire area and turning off the cool area;

Draw  $fire_{t,D_k}^{\prime\prime\prime(i)}$  from  $\frac{c_{DEVS-FIRE}}{c_{sen}+c_{DEVS-FIRE}} \delta(fire_{t,D_k} - fire_{t,D_k}^{\prime(i)}) + \frac{c_{sen}}{c_{sen}+c_{DEVS-FIRE}} \delta(fire_{t,D_k} - fire_{t,D_k}^{\prime\prime(i)})$ ;

**Return**  $fire_t^{(i)} \leftarrow \{fire_{t,D_k}^{\prime\prime\prime(i)}\}_{k=1}^{n_s} \cup fire_{t,D_{non}}^{\prime(i)}$ . ( $D_{non}$  is the area not monitored by sensors)

---

#### 4.3 Sensor Enhanced Proposal Distribution and Density Estimation

Although the sensor cluster generated local fire states may be closer to the real fire than the DEVS-FIRE generated state, it is not always true especially when the fire model is accurate and the sensor readings are with large uncertainty. For the monitored area of each sensor cluster, we predefine a sensor confidence

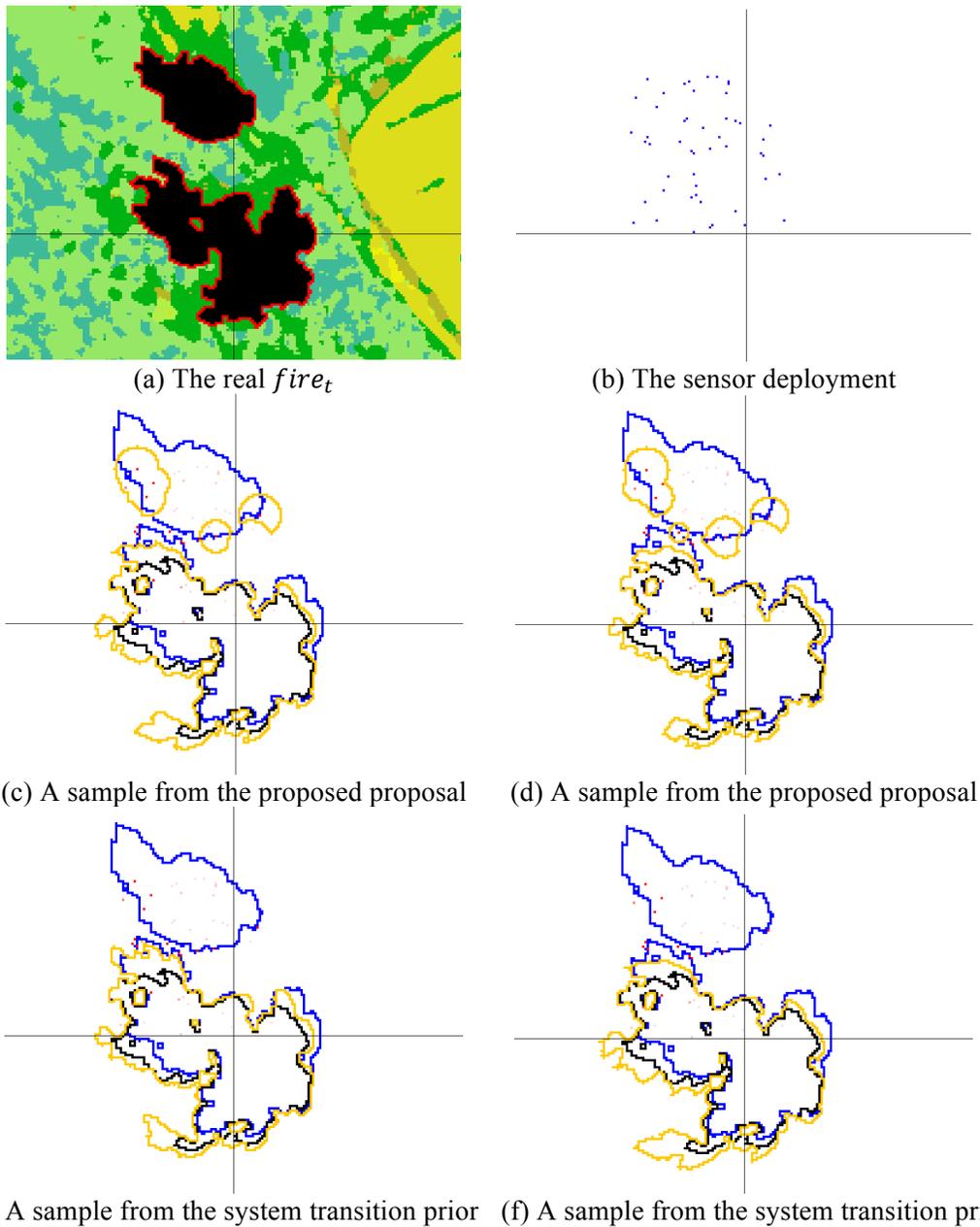


Figure 3: Samples drawn from the sensor enhanced proposal distribution and system transition prior. (The real fire state is shown in (a), and the sensor deployment is shown in (b). (c) and (d) are samples from the proposed sensor enhanced proposal. (e) and (f) are samples from  $p(fire_t|fire_{t-1})$ . In (c)-(f), the real fire front is shown in blue, the fire front of the previous fire state is shown in black, the sampled fire front is shown in yellow, and sensors with temperatures are shown in red dots (the redder, the higher temperature they have)).

$c_{sen}$  and a DEVS-FIRE confidence  $c_{DEV\text{-}FIRE}$ , and the local state is chosen based on these confidence, that is, draw a sample from:

$$\frac{c_{DEV\text{-}FIRE}}{c_{sen}+c_{DEV\text{-}FIRE}}\delta(fire_{t,D} - fire'_{t,D}{}^{(i)}) + \frac{c_{sen}}{c_{sen}+c_{DEV\text{-}FIRE}}\delta(fire_{t,D} - fire''_{t,D}{}^{(i)}),$$

where  $D$  is a sensor cluster monitored area,  $fire_{t,D}$  is the local fire state on  $D$ ,  $fire'_{t,D}^{(i)}$  is the DEVS-FIRE generated local fire state, and  $fire''_{t,D}^{(i)}$  is the sensor cluster generated local fire state. The sampling algorithm is summarized in Algorithm 2. In Figure 3, an example of this proposal distribution is shown; for comparison, system transition proposal (the proposal of the bootstrap filter) results are also displayed.

When updating importance weights through Equation (2), the density values of both the system transition prior and the proposed proposal distribution are needed. Although we do not have the analytical expressions of these distributions, their sampling algorithms are available, so we employ the kernel estimation method (Rosenblatt 1956), that is, when estimating, we draw  $M$  extra samples ( $\{fire_i\}_{i=1}^M$ ) through these sampling algorithms, and apply the kernel function estimation to approximate those densities:

$$p(fire) = \frac{1}{Mh} \sum_{i=1}^M K\left(\frac{fire-fire_i}{h}\right).$$

## 5 EXPERIMENTAL RESULTS

### 5.1 Settings

We employ the identical twin paradigm to perform the wildfire simulation data assimilate experiments. In each experiment, we choose a DEVS-FIRE simulation as the “real” system, and another DEVS-FIRE simulation with added errors as the “simulated” system. We assimilate data from the “real” system to the “simulated” system to illustrate the effectiveness of data assimilation methods.

We prepare two cases as listed in Table 1 to apply SMC data assimilation. In each case, we perform an experiment using the system transition prior as the proposal distribution (it is then a bootstrap filter), and one using the proposed method as the proposal distribution. The errors of the “simulated” system are in wind speed, wind direction, and ignitions. In Case 1, there is a significant wind direction error and a moderate wind speed error; in Case 2, other than the moderate wind speed and direction errors, a rare event happens: in the “real” system, the cell (100, 150) is ignited at 4800s.

Table 1: Settings of “real” and “simulated” fire systems for identical twin experiments.

	“Real” System	“Simulated” System
Case 1	Wind: 5m/s, 125 degrees	Wind: 4m/s, 305 degrees
Case 2	Wind: 5m/s, 125 degrees; a new ignition (100, 150) added at 4800s	Wind: 6m/s, 105 degrees

The fire cell space dimension is 200×200 and the cell size is 15 (m). The airborne LiDAR (Light Detection and Ranging) (Wagner et al. 2004) raster-based terrain data are used as the GIS data. The fuel data were obtained from a multispectral QuickBird (DigitalGlobal) image (Muge et al. 2008). Those data were acquired from Huntsville area, Texas, during the leaf-off season in March 2004 by M7 Visual Intelligence of Houston, Texas.

We define the error as the number of mismatched cells of two fire states (That is, the cells with the same coordinates but not both in the same state as *unburned*, *burning* or *burned*). Other settings are as follows: initial ignition point is at (120, 18); in the measurement distribution, sensor readings are considered as independent and with a standard deviation of 200; when sampling,  $var_{speed} = 1$ ,  $var_{direction} = 15$ ,  $T_{hot} = 150$ ,  $T_{cool} = 20$ ,  $c_{DEV\text{-}FIRE} = 0.8$ , and each  $c_{sen} = 0.8$ ; the simulation length is 3 hours; the data assimilation interval is 1200 seconds; 1000 sensors randomly deployed in the whole fire area.

### 5.2 Experimental Results

Sensor reading data are then assimilated from the “real” fire to the “simulated” fire for each case using SMC methods with the two different proposal distributions. We refer to the bootstrap experiments as “System Prior SMC”, and the one with our proposed method as “Sensor Enhanced SMC”.

In Case 1, the wind direction of the simulated fire is largely wrong, so the simulated fire spreads to a fully wrong direction as in Figure 4: the simulated fire front (in red) goes to Southeast, and the real fire front (in blue) goes to Northwest. In this situation, System Prior SMC fails to approach the real fire as the result in Figure 4(a), it just limits the growth of the fire, but does not provide a good approximation of the real fire. At the same time, Sensor Enhanced SMC still works as the results shown in Figure 4(b), and it still generally keeps track of the real fire. Moreover, from Figure 5, it can be seen that the Sensor Enhanced SMC filtered fire always has much smaller errors than the one of System Prior SMC.

Case 2 is an example of rare events. At 4800s a new fire is ignited at (100, 150), this kind of new fire could happen but with a low probability. When it happens, even if the simulation model is generally accurate, the simulation will still fail by a large chance. As shown in Figure 6, with a rare event (the new ignited fire), System Prior SMC only keeps track of the initial fire, but have no prediction on the new fire.

On the contrary, the filtered fire of Sensor Enhanced SMC generates a separated fire approximately covering the new fire. From Figure 7, one can observe that, after 7200s (when the size of the new fire becomes significant), Sensor Enhanced SMC has much better results than System Prior SMC.

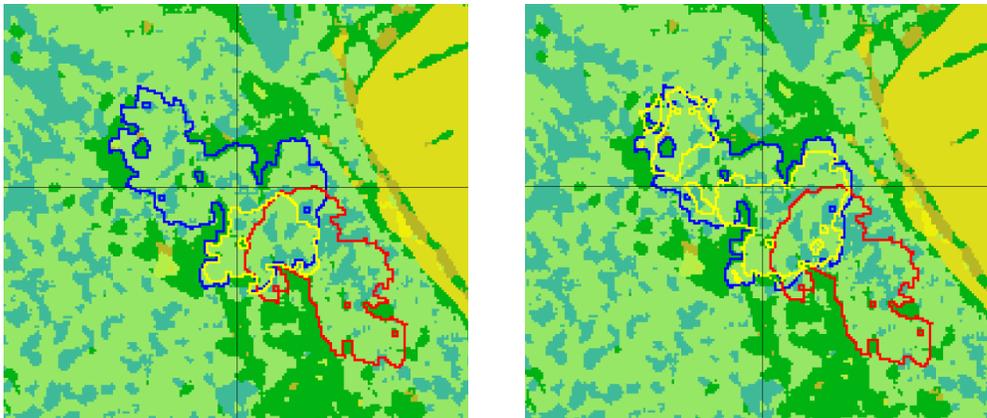


Figure 4: Results of Case 1: the left one is the results of System Prior SMC (b) the right one is the results of Sensor Enhanced SMC. (The SMC filtered fire front is in yellow; the real fire front is in blue; the simulated fire front is in red).

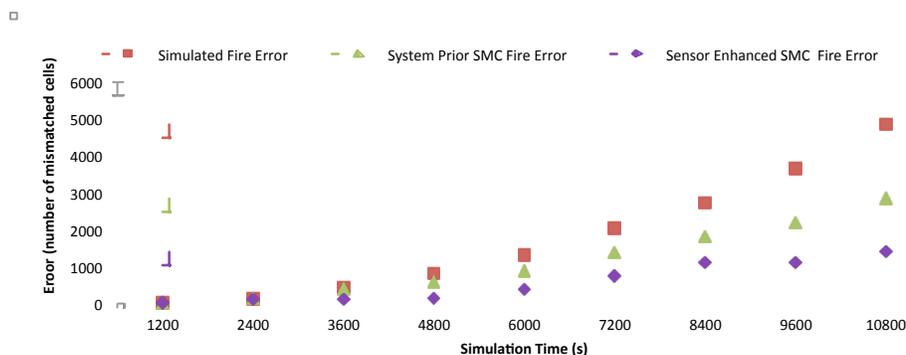


Figure 5: Error comparison of Case 1.

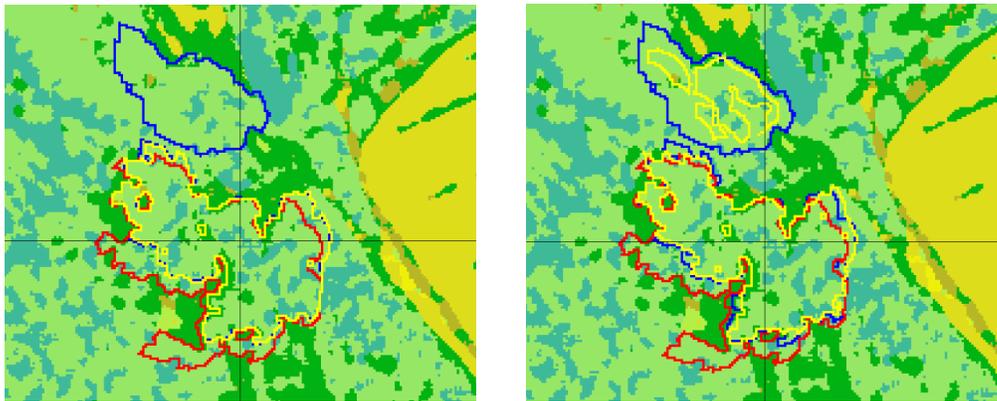


Figure 6: Results of Case 2: the left one is the results of System Prior SMC (b) the right one is the results of Sensor Enhanced SMC. (The SMC filtered fire front is in yellow; the real fire front is in blue; the simulated fire front is in red).

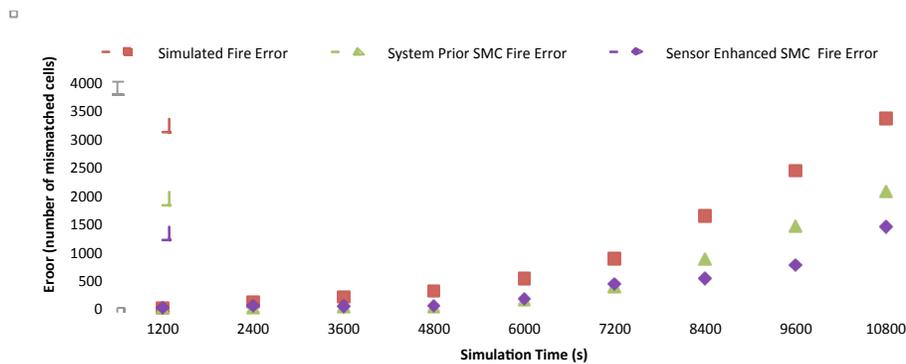


Figure 7: Error comparison of Case 2.

## 6 CONCLUSION

In this article, we present an effective proposal distribution for SMC based wildfire data assimilation. In the sampling step, each sensor monitored area generates two local states: one from the wildfire simulation model, and one from the sensor readings. With the early utilization of sensor information, the proposed method achieves significant better experimental results than a bootstrap filter. This method can be potentially generalized to apply to other complex simulation models.

In the current sensor enhanced sampling algorithm, the confidence of sensor based state and the one of model based state are predefined. Generic methods may be developed to automatically generate them. When updating weights, the kernel estimation method generates approximations of real densities, and its impact on the data assimilation results may be explored in the future.

## REFERENCES

- Cappe, O., S. J. Godsill, and E. Moulines. 2007. "An Overview of Existing Eethods and Recent Advances in Sequential Monte Carlo." In *Proceedings of the IEEE* 95 (5):899-924.
- De Freitas, J. F. G., M. A. Niranjan, A. H. Gee, and A. Doucet. 2000. "Sequential Monte Carlo Methods to Train Neural Network Models." *Neural Computation* 12 (4):955-993.
- Doucet, A., and A. M. Johansen. 2011. "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later." In *The Oxford Handbook of Nonlinear Filtering*, edited by D. Crisan and B. Rozovski. Oxford University Press.

- Finney, M. A. 1998. "FARSITE, Fire Area Simulator--Model Development and Evaluation." In *Research paper RMRS, RP-4*. Ogden, UT (324 25th St., Ogden 84401): U.S. Dept. of Agriculture, Forest Service, Rocky Mountain Research Station.
- Godsill, S., and T. Clapp. 2001. "Improvement Strategies for Monte Carlo Particle Filters." In *Sequential Monte Carlo Methods in Practice*, edited by A. Doucet, N. de Freitas and N.J. Gordon.
- Gordon, N.J., D.J. Salmond, and A.F.M. Smith. 1993. "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation." *IEE-Proceedings-F* 140.
- Hu, X., Y. Sun, and L. Ntamo. 2012. "DEVS-FIRE: Design and Application of Formal Discrete Event Wildfire Spread and Suppression Models." *Simulation* 88 (3):259-279.
- Isard, M., and A. Blake. 1998. "CONDENSATION - Conditional Density Propagation for Visual Tracking." *International Journal of Computer Vision* 29 (1):5-28.
- Kyriakides, I., D. Morrell, and A. Papandreou-Suppappola. 2008. "Sequential Monte Carlo Methods for Tracking Multiple Targets With Deterministic and Stochastic Constraints." *IEEE Transactions on Signal Processing* 56 (3):937-948.
- Lao, Y. W., J. D. Zhu, and Y. F. Zheng. 2009. "Sequential Particle Generation for Visual Tracking." *IEEE Transactions on Circuits and Systems for Video Technology* 19 (9):1365-1378.
- Mandel, J., L. S. Bennethum, J. D. Beezley, J. L. Coen, C. C. Douglas, M. Kim, and A. Vodacek. 2008. "A Wildland Fire Model With Data Assimilation." *Mathematics and Computers in Simulation* 79 (3):584-606.
- Matgen, P., M. Montanari, R. Hostache, L. Pfister, L. Hoffmann, D. Plaza, V. R. N. Pauwels, G. J. M. De Lannoy, R. De Keyser, and H. H. G. Savenije. 2010. "Towards the Sequential Assimilation of SAR-Derived Water Stages into Hydraulic Models Using the Particle Filter: Proof of Concept." *Hydrology & Earth System Sciences Discussions* 7 (2):1785-1819.
- Muge, M., C. P. Sorin, S. Curt, and S. Tom. 2008. "Mapping Surface Fuel Models Using Lidar and Multispectral Data Fusion for Fire Behavior." *Remote Sensing of Environment* 112:274-285.
- Pan, P., and D. Schonfeld. 2008. "Dynamic Proposal Variance and Optimal Particle Allocation in Particle Filtering for Video Tracking." *IEEE Transactions on Circuits and Systems for Video Technology* 18 (9):1268-1279.
- Rosenblatt, M.. 1956. "Remarks on Some Nonparametric Estimates of a Density Function." *Annals of Mathematical Statistics* 27 (3):832-837.
- Rothermel, R. C. 1972. "A Mathematical Model for Predicting Fire Spread in Wildland Fuels." In *USDA Forest Service research paper INT, 115*. Ogden, Utah: Intermountain Forest & Range Experiment Station, Forest Service, U.S. Dept. of Agriculture.
- Rubin, D. B. 1988. "Using the SIR Algorithm to Simulate Posterior Distributions." In *Bayesian Statistics 3*, edited by J. M. Bernardo, M. DeGroot, D. Lindley and A. Smith, 395-402. Oxford University Press.
- Saha, S., and F. Gustafsson. 2012. "Particle Filtering With Dependent Noise Processes." *IEEE Transactions on Signal Processing* 60 (9):4497-4508.
- Salamon, P., and L. Feyen. 2009. "Assessing Parameter, Precipitation, and Predictive Uncertainty in a Distributed Hydrological Model Using Sequential Data Assimilation With the Particle Filter." *Journal of Hydrology* 376:428-442.
- van der Merwe, R., A. Doucet, N. de Freitas, and E. Wan. 2000. "The Unscented Particle Filter." Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department.
- van Leeuwen, P. J. 2010. "Nonlinear Data Assimilation in Geosciences: an Extremely Efficient Particle Filter." *Quarterly Journal of the Royal Meteorological Society* 136 (653):1991-1999.
- Van Wagner, C. E. 1973. "Height of Crown Scorch in Forest Fires." *Canadian Journal of Forest Research* 3 (3):373-378.
- Van Wagner, C. E. 1975. "Convection Temperatures above Low Intensity Forest Fires." *Canadian Forest Service Bi-monthly Research Note* 31 (2).

- Wagner, W., A. Ullrich, T. Melzer, C. Briese, and K. Kraus. 2004. "From Single-Pulse to Full-Waveform Airborne Laser Scanners: Potential and Practical Challenges" *International Archives of the Photogrammetry, Remote Sensing, and Geoinformation Sciences* XXXV (B/3) 414-419.
- Xue, H., F. Gu, and X. Hu. 2012. "Data Assimilation Using Sequential Monte Carlo Methods in Wildfire Spread Simulation." *ACM Transactions on Modeling and Computer Simulation* 22 (4):1-25.
- Zeigler, B. P., T. G. Kim, and H. Praehofer. 2000. *Theory of Modeling and Simulation*. 2nd ed: Academic Press, Inc.
- Zhai, Y., M. B. Yeary, S. Cheng, and N. Kehtarnavaz. 2009. "An Object-Tracking Algorithm Based on Multiple-Model Particle Filtering With State Partitioning." *IEEE Transactions on Instrumentation and Measurement* 58 (5):1797-1809.

#### **AUTHOR BIOGRAPHIES**

**H Aidong Xue** is a Ph.D. candidate in the Department of Computer Science at Georgia State University. His research interests include Dynamic Data Driven Applications Systems, sequential Monte Carlo based data assimilation for complex systems, and wildfire simulation. His email address is [hxue1@student.gsu.edu](mailto:hxue1@student.gsu.edu) and his web page is <http://cs.gsu.edu/~hxue1>.

**Xiaolin Hu** is an Associate Professor in the Department of Computer Science at Georgia State University, Atlanta, Georgia. He received his Ph.D. degree from the University of Arizona in 2004. His research interests include modeling and simulation theory and application, agent and multi-agent systems, and complex systems science. His email address is [xhu@cs.gsu.edu](mailto:xhu@cs.gsu.edu) and his web page is [http://www.cs.gsu.edu/?q=x\\_hu](http://www.cs.gsu.edu/?q=x_hu).