CAndbox: Modeling Fire Spread on Real-Time Interactive Terrain

David Robinson[1,2], Miles Currie[1,2], Bryan Quaife[1,2], and Kevin Speer[1]

[1]Geophysical Fluid Dynamics Institute
[2]FSU Department of Scientific Computing

Abstract

Modeling the spread of forest fires via cellular automata is an attractive approach for its efficient computational implementation, and the surprising realism of deceptively simple algorithms in simple geometries. However, applying this method to a real wildland fire environment with complex fuel, topography, and wind requires substantial parameterization. An established approach is to use pixel-by-pixel rates of spread derived from the Rothermel model, but these models do not integrate over the contributions of multiple ignited pixels and only allow a discrete number of states. We present a new method which allows for semi-discrete states by summing over the contribution of all ignited pixels on a combustable pixel within a given neighborhood. Folding together the effects of terrain and wind speed, we employ an extended Rothermel model to form an "effective spread ellipse" defined by the rate of spread an ignited pixel contributes in any direction. Furthermore, we tune the model to real fire data using Bayesian inference. Because of the computational efficiency of this method, we are able to apply it to a "sandbox", an augmented reality projection system. Initially designed to demonstrate fluid flow on an evolvable surface, this system allows the fire model to facilitate real-time adjustments to topography and burning regions. Additionally, the system handles complex initial fire shapes while remaining intuitive to the user. The outcome is a tool that is ideal for exploring the effect of topography on the spread rates of fire in a real-time environment.

1 Introduction

This is the introduction

2 Model

2.1 Modification to Previous Model

2.1.1 Previous Model

2.1.2 Our Model

- Based on rothermel model

- explain classical approach

- explain fire bucket method and how it differs

We use the Rothermel model given by

$$R(\theta) = R_0 \frac{1 - \overline{E}}{1 - \overline{E}\cos(\theta)} \tag{1}$$

where θ is the angle measured from the direction of maximum spread, R_0 is the rate of spread with no wind or terrain effects, and $\overline{E}$ is the ellipse eccentricity, defined as

$$\overline{E} = \sqrt{1 - 1/LW^2} \tag{2}$$

where LW is the ellipse length-to-width ratio calculated as

$$LW = 0.936e^{0.2566U_{eq}} + 0.461e^{-0.1548U_{eq}} - 0.397 \tag{3}$$

where the equivalent wind speed U_{eq} $(\frac{m}{s})$ is the wind speed that alone would produce the combined effect of actual wind and terrain slope.

Our CA method uses the Rothermel model to draw an ellipse around a single ignited cell, where the cell in question is one of the foci. $R(\theta)$ gives the rate of spread around the cell as an ellipse.

Outline of code:

- the code is based on a grid system:

- grid for terrain (elevation), wind direction, wind speed, a state grid, a bucket grid, and a fuel grid

- explain each of these

- explain states

- we have timesteps:

- for each timestep, we find the cells in the "I", or burning state

- for each of the cells in the "I" state, we first check to see if they can be extinguished– how?

- then, several steps:

- for a burning cell, we find the elevations of its eight neighbors and calculate the slope between each neighbor and the cell of interest

- we then find the rate of spread from the Rothermel model with inputs of fuel, wind magnitude, and wind direction

- then calculate modifications to the rate of spread given by effects of the terrain and wind (phi values, write equations)

- treat the ros values as contributions to the neighboring cells (maybe show example in a figure?)

- add these values to bucket grid

- check where on the bucket grid the buckets are "filled up". ie we have some threshold, in the initial case unity, where we call the buckets full and then change the corresponding cell on the state grid to the "I" state.

- do these steps for each timestep

2.2 Grid System

Our model is based on the use of a grid system: All information is stored in separate grids that describe the terrain, wind magnitude, wind direction, fuel, the state of each cell, and a means to store the contributions of burning cells on others.

2.2.1 Terrain Grid

To describe the terrain, our model requires a grid of elevation values at each cell. The elevation values are used to calculate a slope between a cell of interest and its neighbors. The slope information contributes to the rate of spread.

2.2.2 Wind Grids

Although wind can be described by a single grid of vectors, we found it is simpler to describe wind in two separate grids, one for magnitude and the other for direction. This works better with the software we use for projecting this model onto our sand box because each texture requires a single value for each cell, which would not work for a vector. (David, is this right? Please correct me if I got it wrong.)

2.2.3 Fuel Grid

Each cell is able to carry a different value for the type of fuel present in it. This makes it easy to later add a fuel model in to further compare to other fire models.

2.2.4 State Grid

The state grid stores the present state of each cell. Each cell can only be one of four states:

- U: the cell is unburnable

- N: the cell is flammable, but not currently ignited

- I: the cell is flammable and currently ignited, however the fuel is not yet consumed

- C: the cell is not currently ignited and all fuel has been consumed

Unburnable (U) cells can include objects like rocks or bodies of water and are forbidden to ignite in our simulations. Flammable cells also carry fuel information in the corresponding fuel grid (Section 2.2.3).

2.2.5 Bucket Grid

Heavily tied to the state grid, the bucket grid carries the information that tells the state grid when to switch a cell in the "N" state to the "I" state. For a given cell in the "I" state, we determine the "energy" contribution of that cell to each of its eight nearest neighbors by employing the Rothermel model (Equation 1). This process is further described in Section 2.3.2 and Figure 1.

2.3 Rate of Spread Calculations

For each iteration (timestep) of our model, several steps are implemented to propagate the fire wall. We begin by obtaining all cells in the "I" state. Cells in any other state are skipped over because the fire is not able to propagate to them. For all cells in the "I" state, we first check if they will be extinguished, and if not, we continue with the process of calculating the rate of spread for these cells.

2.3.1 Extinguishing Cells

Following the suggestion of [?], a cell is only moved from the "I" state to the "C" when all eight of its nearest neighbors are in some combination the "I", "C", and "U" states. In other words, if all eight of its neighbors have no potential to light on fire, the cell is moved to the "C" state.

In each timestep, the first action performed after finding all cells in the "I" state is determining if any of those cells are switched to the "C" state using this algorithm.

2.3.2 Fire Propagation

For all "I" cells which pass the extinguishing algorithm still in the "I" state, we begin calculating individual rate of spread ellipses using the Rothermel model. The Rothermel model requires several input parameters, which the are stored in the aforementioned grid system. R_0 directly corresponds to the type of fuel present in each cell and is found in the fuel grid (Section 2.2.3). $\overline{E}$ is more complicated, however it can be calculated using Equations 2 and 3, with U_{eq} coming from the corresponding value in the wind manitude grid. θ is the angle of a cell's neighbor measured from the direction of maximum spread, or wind direction, which is found in the wind direction grid.

Because the Rothermel model in Equation 1 does not include a modification for terrain effects and an arguably weak modification for wind speed (this may not be true), we introduce two new parameters that are used in the FARSITE fire model [], Φ_s and Φ_w, which describe these effects further:

$$\Phi_s = 5.275\beta^{-0.3}\tan^2(\phi) \tag{4}$$

we need to check on the tan value. not sure whether ϕ is squared or $\tan(\phi)$ is squared

Where is the squared and why? What is ϕ? Can Φ_s ever be negative? If not, the ROS ellipse will always grow regardless of the slope which is incorrect.
- Rothermel 1972 says that it is the whole thing squared, i.e. $(\tan(\phi))^2$
- Why? Because it fits the data. (Couldn't find a better explanation.)
- $\tan(\phi)$ is the slope, not ϕ
- ϕ is the ratio of change in elevation to change in pixel distance, not an angle
- Φ_s can never be negative because Rothermel never tested downward slopes. He took data for upward slopes and fit it accordingly.
- I think we need to find something new for downward slopes. Surely there must be a paper about this somewhere...

$$\Phi_w = C(3.28U)^B \left(\frac{\beta}{\beta_{op}}\right)^{-E} \tag{5}$$

Φ_w is left out of my code at the moment

where β is the packing ratio of the fuel bed, β_{op} is the optimum packing ratio (references for βs are in FARSITE paper), ϕ is the slope (radians), U is the midflame windspeed ($\frac{m}{s}$), and the C, B, and E coefficients are functions of the fuel particle sizes in the fuel bed [](Burgan 1987; Rothermel 1972).

Table 1 summarizes all input parameters for the Rothermel model and where the algorithm finds them. We input these parameters in the Rothermel model to obtain rate-of-spread valuse for the eight nearest neighbors of a given cell. We treat these values as energy values, filling up the energy buckets in the neighboring cells. Figure 1 shows an example of this with an ellipse of eccentricity 0.5. The cell in the direction of maximum spread gets the most energy from

0.77	1	0.77
0.5	1	0.5
0.36	0.33	0.36

Figure 1: An example of a 0.5 eccentricity rate-of-spread ellipse's energy contribution on neighboring cells. These values are calculated from Equation 1 with $R_0 = 1$ and $\overline{E} = 0.5$ (eccentricity). In this example, and with a combustion threshold of 1, the cell directly to the north will change states from "N" to "I" in this timestep. It will be able to make contributions to its respective neighbors in the next timestep. The remaining values are saved and new rate-of-spread values will be added to them in subsequent timesteps until they pass the combustion threshold.

this ellipse, thus it fills up faster than the rest of its neighbors. The biggest advantage to this method is that neighboring cells also in the "I" state can contribute to the ignition of "N" cells which share the same neighborhood as other "I" cells.

Once cells' energy buckets are filled up to a certain threshold, we change the state of the cell from "N" to "I" and these cells are able to contribute to their neighbors in subsequent timesteps. After reassigning these states, the process for one timestep is complete and the next timestep begins.
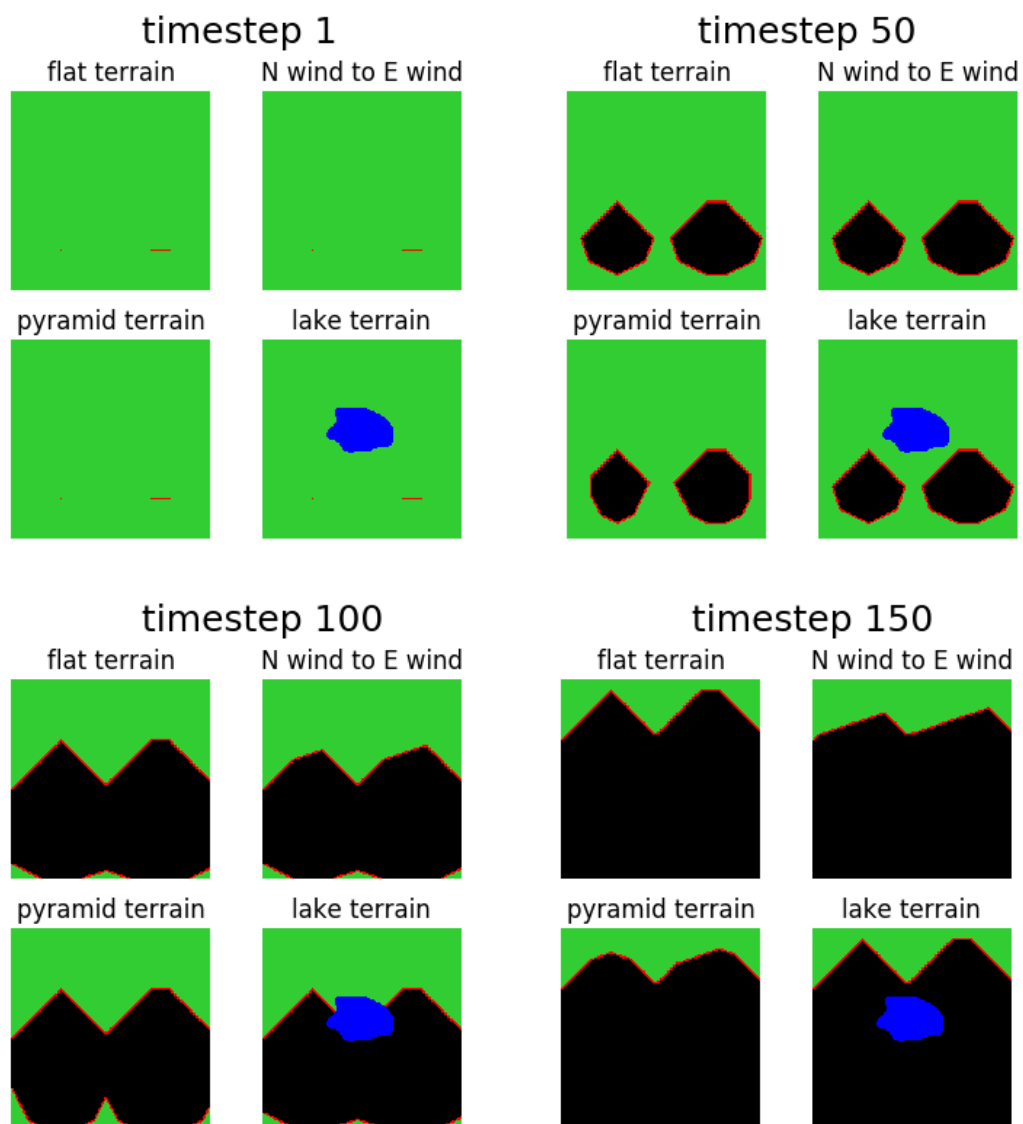
3   Sandbox

David

4   Discussion and Conclusions

Figure 2: This is the model

Table 1: Inputs Summary

Model Parameter	Corresponding Grid Parameter
R_0	Fuel Grid
	(2.2.3)
$\overline{E}$	Wind Magnitude Grid
	(2.2.2)
reference angle	Wind Direction Grid
for θ	(2.2.2)
Φ_s	Terrain Grid
	(2.2.1)
Φ_w	left out at the moment