Wildfire Propagation Simulation Tool using Cellular Automata and GIS

Washington Velásquez , Andres Munoz-Arcentales , Thomas Michael Bohnert , Joaquin Salvachúa

***Abstract*—This paper states the research done in building a Wildfire Propagation Simulation tool able to present graphically, how the fire will be propagated in case of a wildfire. The core of this tool was developed using Cellular Automata (CA) relayed on mathematical foundations for modeling the propagation and the transition rules of the CA. Also, the mathematical model was combined with a geographical information system provided by Google Maps API, allowing to conduct a simulation that considers the kind and density of vegetation in a specific zone of interest. Finally, we define and perform a full simulation providing information for managing and predicting wildfires and, the future works derived from this research.**

***Keywords*—Cellular Automata; Emergency; GIS; Google Maps; Simulation; Wildfire**

I. Introduction

Nowadays, wildfires are considered a global phenomenon that affects not only the forest area but also the whole ecosystem (human lives, wildlife, vegetation, physical infrastructures, etc.). According to Matthew P. Thompson Dave and E. Calkin in ”Uncertainty and risk in Wildland fire management: A review”[1], beyond the unpredictability of wildfire behavior, one of the most sources of uncertainty is generated for inaccurate and missing data and incomplete scientific understanding of fire behavior. Due to this fact, the research in this area has been growing in the last years.

Moreover, in recent years, many countries have been affected by wildfires generating substantial economic losses and even the loss of animals and human lives. Catastrophes like the California wildfire in December of 2017 [2] and the wildfires that have been affected Portugal during the last decade [3] made as to realize that we need to study and develop strategies for preventing, controlling and managing this type of emergencies.

Currently, many researchers have been working on how to predict and determine the propagation of fire inside the forest using a lot of environmental and composition variables according to the study zone [4], [5], [6]. Thus, as an extension in this topic, we present in this paper a wildfire propagation simulation tool, using cellular automata and Geographical Information systems (GIS) for modeling and showing the wildfire propagation with the aim of providing relief agency of a tool that can be used in ambient environments and allow safeguarding the surrounding cities. The simulation tool is the first piece of a set of work items with the aim to build and deploy a full stack of components that allow to monitor and predict fires.

This paper is divided into six sections. Section II covers all state of the art about the concepts and technologies used for building the simulation tool. Section III focuses on providing the mathematical foundations for making the cellular automata used in this work. In section IV, we present the simulation environment, providing information about the variables that the simulation tool uses and the algorithm developed. Section V is dedicated to showing the results and tests of the tested simulation. Finally, section VI states the discussions about the effects and the future works on this topic.

II. State of the Art

In this section, we overview the literature about wildfires model simulation and different topics considered into the scope of this paper.

A. Cellular Automata

Let's assume that we can see the earth's surface from above, and we can separate them into multiple rows and columns, i.e., we create a matrix with $m * n$ elements. If we call each of these elements a cell, we will have, on the surface, $m*n$ cells. Each cell will have a coordinate (x, y) in a Cartesian space or similar, depending on the coordinates used in the GIS or in the surface system to model. In our case, it will be GPS Location using Google maps $(latitude/longitude)$.

A cellular automata (CA) is a mathematical model that evolution through discrete events in a dynamic system [7] using a net or matrix with n elements. At each location on the network, each cell has a state, and the state of the cell in the following time transition will depend on this state and the states of the cells in the neighborhood. A CA has the following properties:

- *Infrastructure*: The network dimension can be like (ECA, Elementary Cellular Automata, a cell line), or (a two-dimensional network or ”grid”) or higher. The cells are usually rectangular or square, but they can also have other topologies but with finite dimension.
- *Cell states*: S is a finite non-empty set of states. Every cell in the CA has one state S.
- *Cell's neighborhood*: Each cell will have a neighborhood scheme so that the next state, cell state will depend on its

current state and the states of the cells belonging to that neighborhood. fig. 1 shows the Von Neumann and Moore neighborhoods for two dimensions, those generally used in CA models for the simulation of the extent of forest fires.

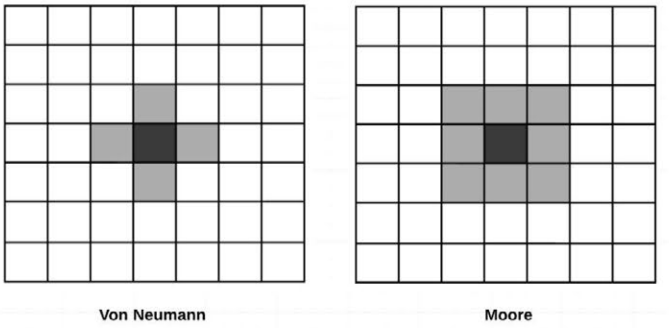


Fig. 1. Schemes of Cells used in a Cellular Automata

- *Transition States Rule*: f rule is applied locally to each cell, so that the cell state changes in each discrete step of the transition and that next state depends on the current cell state and the current state of the cells in its neighborhood. Thus, $S_{(i,j)}(t+1) = f(S_{(i,j)}(t), S_{neighbor(i,j)}(t))$. All the CA cells will apply, locally, the function f, so that at time $(t+1)$ the global configuration of the CA will be the set of the states of each cell of the CA at instant $(t+1)$.

B. Fire Simulation Models

Considering the CA as the basic model (Van Neumann and Moore), the different algorithm to model wildfires are going since a simple simulation where they don't take into account variables like the land topography, wind speed, elevation of the zone (uphill or downhill) until some very complexity that indeed those considering changes of states in discrete events and kind of vegetation. Among them are:

- **Quarterie** [8], It's a very simple model that uses the neighborhood of Moore, four states per cell, the transition rule (p) depends on how many cells are burning and unfortunately the model does not include meteorological and topographical variables.
- **Ljiljana** [9], Model a little more complex because this incorporates variables of CORINE Land Cover[1] to adapt Geographical Information System (GIS) to the model.
- **Karafyllidis** [10], It uses weighting factors to simulate the slope of the terrain and wind speed and direction. This is based on a neighborhood of Moore and The initial state of each cell depends on the burned area of the cell with respect to its total area. It starts from a matrix in which, to each cell, this initial state is assigned, depending on the conditions of the terrain.
- **Yassemi** [11], A complex model that uses real-time weather and topographic variables together with a geometrically tuned state transition rule, applying it to a GIS-based simulation.
- **Lopes** [12] , Very complex model with two variants, simple and complex, with the advantage of real-time data and the ability to use the simulator for small areas or for predicting large areas.
- **Alexandridis** [13], Model to calculate fire in surface with GIS and spotting effect (Generating new ignition points away from the perimeter of advance of the fire due to falling material burning out of the perimeter area, which have been transported by convection currents created by the fire).

Hence, the models start from a CA by varying their geometric layout, but, they include topographic and meteorological information, there are even some that have a simulator to perform several practices with regard to fires [14].

C. Geographical Information System (GIS)

The use of meteorological and topographical variables help to get a better precision in how it will be the fire propagation. In our case, we obtain the following:

- Latitude (lat), longitude (long), and land elevation are obtained using the Google Maps API [15].
- Wind speed and direction in a specific lat/long point is obtained using Weather API[2]

III. Mathematical Model

In 2008, Alexandridis et al.[13] presented a mathematical model to predict or infer wildfires with a specific studied case: "Spetses Island in 1990", the accuracy about this model was around 98% compared to the historical map of the island. In the model, it considers the following factors:

- *Moore neighborhood*
- *Kind of vegetation*: The vegetable fuel is what will burn, it has chemical characteristics that will determine the amount of heat released in its combustion.
- *Density of vegetation*: The distribution of each type of vegetation on each unit area of land, whether measured in density, either in a number of trees or shrubs per unit area, will depend on how the spread of fire.
- *Vegetation humidity*: Quantity of humidity that contains the fuel.
- *Wind speed and direction*
- *Land topography*: In this variable is considered: elevation of each cell, terrain barriers (rivers, street, rocks, etc)

A. Cellular automata with Moore

A two-dimensional network is used that covers a geographical area of the study. Each cell represents a specific point of lat/long in square meters. Moore neighborhood consists of a central cell with its 8 neighbors and as shown in fig. 2. In the same way, the arrows represent the possible propagation directions depending on the central cell.

[1] https://www.eea.europa.eu/data-and-maps/data/clc-2000-vector-6

[2] https://openweathermap.org/api

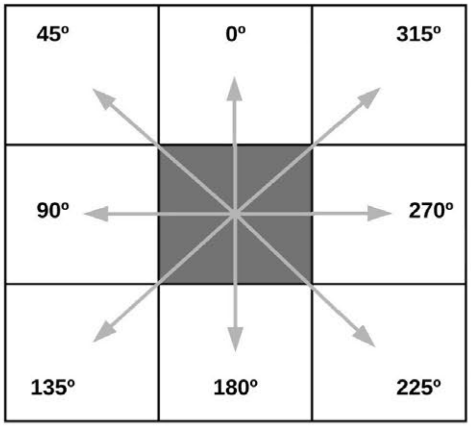


Fig. 2. Schemes of Cells used in a Cellular Automata

B. Cell states

Each cell has a finite number of states, which evolves in discrete time. Possible states are:

- 1. The cell does not contain vegetable fuel and, therefore, it can not burn.
- 2. It contains fuel (vegetation) that has not ignited.
- 3. It contains burning vegetation.
- 4. It contains vegetation that has burned completely

C. State transition rules

In each simulation step t (the steps are discrete time), given the cell (i, j), then:

- *Rule 1*: If state of $(i, j, t) = 1$, then state of $(i, j, t+1) = 1$. That is, the state of the cells that do not contain fuel will always remain the same, since these cells can not burn.
- *Rule 2*: If state of $(i, j, t) = 3$, then state of $(i, j, t+1) = 4$. That is, if the cell burns in the current step of the simulation, the next time step will be considered burned.
- *Rule 3*: If the state of $(i, j, t) = 4$, then state of $(i, j, t+1) = 4$. That is, if the cell is burned, in the next step remains the same.
- *Rule 4*: If state of $(i, j, t) = 3$, then state of $(i \pm 1, j \pm 1, t+1) = 3$, with a certain probability of fire. That is, at time t we consider the central cell (i, j).

D. Propagation calculation

The probability (P_{burn}) that each cell is burning, is given by the equation 1 [13], [16]:

$$P_{burn} = P_0(1 + P_{veg})(1 + P_{den}) * P_w * P_{ele} \tag{1}$$

Where,

- The P_0 factor is empirically developed by Alexandridis, which indicates the probability that a neighboring cell is burning and garment in the next step of the simulation under conditions of absence of fire and elevation difference between the central cell and neighboring. ($P_0 = 0.58$)
- Vegetation density (P_{den}). It can have the values of (limited = -0.4, medium = 0, high = 0.3) [13]
- Kind of vegetation (P_{veg}). It depends on the kind of vegetation in the zone.
- Wind speed (P_w). Θ is the angle between the direction of propagation of the fire (the direction in which the neighboring cell has been with respect to the center) and the wind direction. Let V be the wind speed. Then, the factor P_w is modeled as:

$$P_w = exp(c_1 V) f_t \tag{2}$$

where,

$$f_t = exp(V c_2(\cos\Theta - 1)) \tag{3}$$

And, c_1,c_2 are constants defined empirically by Alexandridis [13] with values of 0.045 and 0.131 respectively.

- Land elevation (P_{ele}). Fire spreads faster if it goes uphill than downhill. To model, this phenomenon is considered the difference in height between the center cell and the neighboring cell to which the fire can spread. Let E_1 be the elevation of the neighboring cell and E_2 the elevation of the central cell. So,

$$P_{ele} = exp(a\Theta_s) \tag{4}$$

So, a is constant with value of 0.078 [13].
Θ_s has different formula according give cells disposed adjacent or diagonal. So, if the cells are adjacent (0°, 90°, 180°, 270°), Θ_S is:

$$\Theta_s = \arctan(\frac{E_1 - E_2}{l}) \tag{5}$$

Or, if the cells are diagonal (45°, 135°, 225°, 315°) is:

$$\Theta_s = \arctan(\frac{E_1 - E_2}{l\sqrt{2}}) \tag{6}$$

And, l is the distance between those two cells.

IV. Simulation Environment

In this section, we detail the simulation environment like the algorithm used to calculate the propagation using cellular automata, the use of Google maps API to get lat, long, and elevation of each point in the studied surface. Also, how the density and kind of vegetation of a coverage area can affect to the wildfire propagation and so on.

A. GIS data

The first step in the simulation is getting the Google Tile[3] and to set the different Cartesian plane of each one of them with matrices. So, Using Google maps API [15], to get a specific monitoring zone as shown in fig. 3, to predict wildfires in this area. Although, how the monitoring area affects the simulation is out of the scope of this paper.

Using a central point ($lat = 40.136693, long = -8.693093$) to get an image that contains $5x5$ Google Tile of

[3] https://developers.google.com/maps/documentation/javascript/coordinates

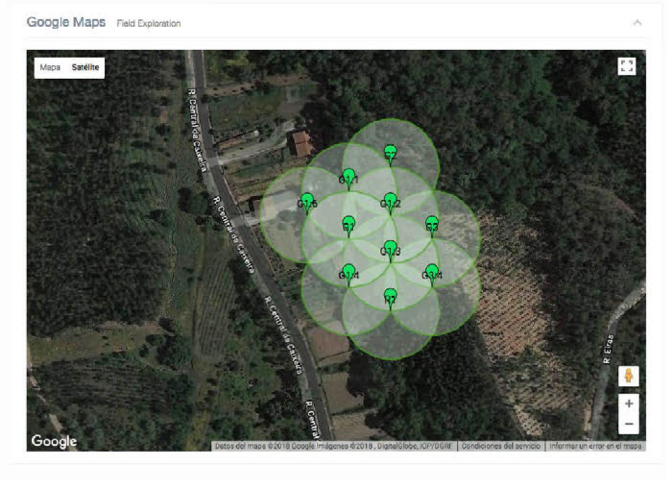


Fig. 3. Monitoring area of the simulation

$256px$*$256px$ each one, i.e., each Google Tile is a subsection of our final image as shown in fig. 4.

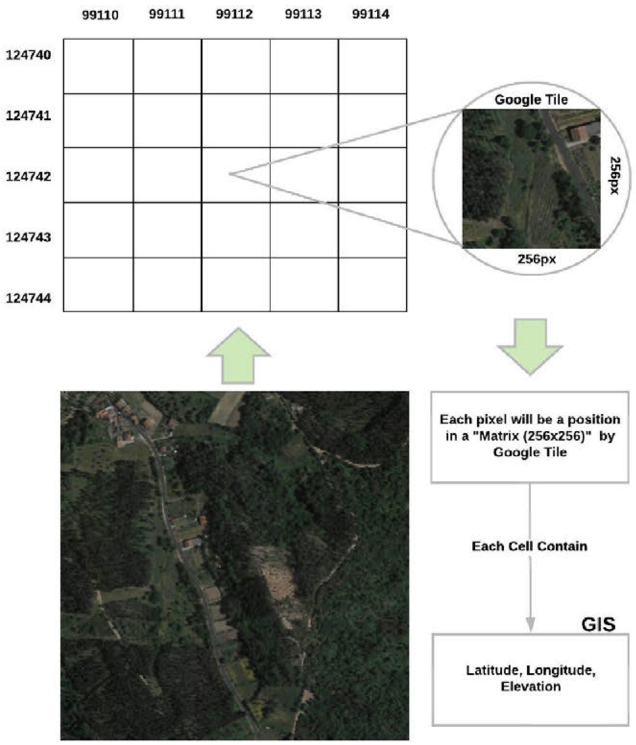

99110
99111
99112
99113
99114
124740
124741
124742
124743
124744
Google Tile
256px
256px
Each pixel will be a position in a "Matrix (256x256)" by Google Tile
Each Cell Contain
GIS
Latitude, Longitude, Elevation

Fig. 4. Model to assemble the various matrices involved in the simulation

B. Kind and Density of Vegetation

To get these values in a specific zone, we use each *Google Tile* image; depending on the position lat/long in Google maps, this position is transformed to (x, y) position in an image (256x256)px. Then, RGB Color is obtained in (x, y) position to send it to a RGB Sorter. As shown in the fig. 5

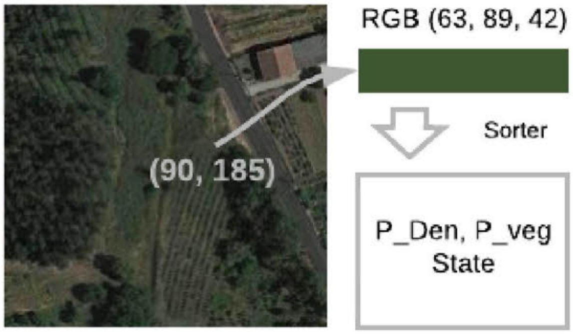


Fig. 5. RGB Color Sorter using the position in the Google Tile image

So, *State* can be four values $(1 : Free,\ 2 : Vegetable - Fuel,\ 3 : Burning,\ 4 : Burned)$. This state is used in each matrix and count simulation step through time.

C. Distance between lat/long points

This value (l) is used in the equations 5 and 6 to calculate Θ_S, and, to get the value of P_{ele}. But, before to use the distance formula derived from the Pythagorean theorem, it has to transform (x, y) or $(lat/long)$ position to meters.

We use the globalMercator[4] (Google Tile) class to do this transformation, This class use conversion given $(lat/long)$ in WGS84 Datum to XY in Spherical Mercator EPSG:900913.

D. Starting point fire

Before to start the simulation, first, we have to set the starting point fire using the Google Maps API, as shown in fig. 6. Also, In this location $(40.13611883734981, -8.693508742397285)$ using the Weather API, we get in real-time the wind speed and direction $(speed : 5.7m/s, direction : 310^\circ)$.

E. Algorithm

In this section, we introduce a brief description of the algorithm used in each step of the simulation, as shown in Algorithm 1. The main function receives certain parameters that are obtained before, for which it assumes the following:

- The RGB Color for each Google Tile image cell is got.
- All matrices are configured with latitude, longitude and elevation for each one Google Tile.
- The starting point fire has been set.

V. Results

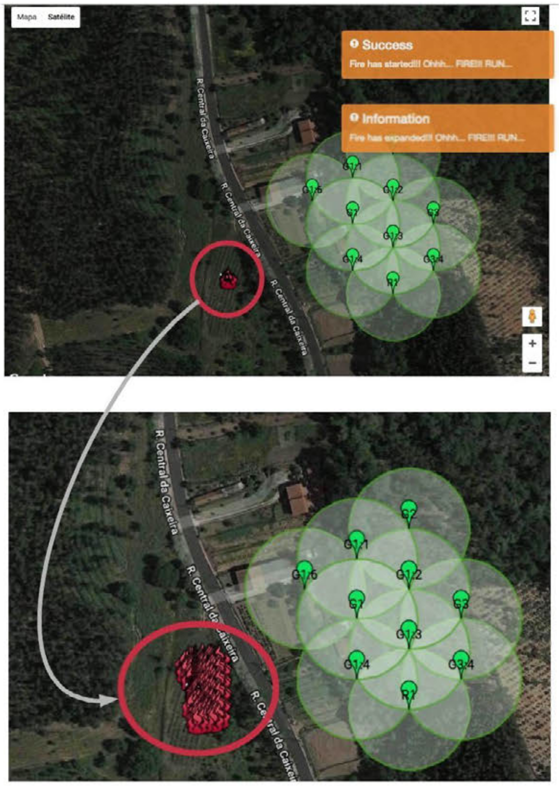
In this section, we present the simulation and how the fire propagation goes through a coverage area. In the fig. 6, It is shown as is the spread of fire, from a starting point until it gradually expands. The initial data are: $latitude \rightarrow 40.13611883734981$, $longitude \rightarrow -8.693508742397285$, wind speed → $7.7m/s$, wind direction → 340° and Google Tile → $(124741 - 99112)$

[4] http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/

ALGORITHM 1: Get the neighboring cells in a position

```
Input: latitude, longitude, windSpeed, windDegree, step
Output: An array of cells representing the neighbors at position
        $(latitude, longitude)$, with information for each cell:
        $(id_{mesh}) \rightarrow$ Identify which mesh you are working with,
        $(degree) \rightarrow$ Which neighbor cell has been calculated,
        $(x, y) \rightarrow$ Position in the Google Tile Image,
        $(lat, long) \rightarrow$ Position in Google Maps, $(ele) \rightarrow$
        Elevation in lat/long position, $(RGB) \rightarrow$ RGB Color
        considered in the calculation, $(Tile_x, Tile_y) \rightarrow$ Which
        Google Tile has been considered in the calculation,
        $(Step) \rightarrow$ In which discrete event was considered that
        cell, $(P_{burn}) \rightarrow$ What is the probability that cell is on
        fire, $(State) \rightarrow$ State of this cell.
$neighborhood$ = [];
$gn$ = GlobalMercator();
$tile_x, tile_y$ =
  gn.GoogleTileFromLatLong(latitude,longitude,zoom=18);
$id_{mesh}$ = getIdMeshMatrizbyTile($tile_x, tile_y$);
$value_{center_x}, center_x, value_{center_y}, center_y$ =
  getPositionImgLatLong(latitude,longitude,zoom=18);
$meters_{center_x}, meters_{center_y}$ =
  gn.LatLonToMeters($lat = value_{center_x}, lon = value_{center_y}$);
$P_o = 0.58;$
$a = 0.078;$
$c_1 = 0.045;$
$c_2 = 0.131;$
Position for each Cell;
${}_{0}Degree_x = center_x - 1$; ${}_{0}Degree_y = center_y$;
${}_{45}Degree_x = center_x - 1$; ${}_{45}Degree_y = center_y - 1$;
${}_{90}Degree_x = center_x$; ${}_{90}Degree_y = center_y - 1$;
${}_{135}Degree_x = center_x + 1$; ${}_{135}Degree_y = center_y - 1$;
${}_{180}Degree_x = center_x + 1$; ${}_{190}Degree_y = center_y$;
${}_{225}Degree_x = center_x + 1$; ${}_{225}Degree_y = center_y + 1$;
${}_{270}Degree_x = center_x$; ${}_{270}Degree_y = center_y + 1$;
${}_{315}Degree_x = center_x - 1$; ${}_{315}Degree_y = center_y + 1$;
repeat
    $m_x, m_y \rightarrow$ Values in meters for each cell, using the
      globalMercator class;
    $radio \rightarrow$ Calculation of the distance between the center cell
      with each one neighborhood cell;
    $P_{den}$ and $P_{veg} \rightarrow$ Depends on the RGB Sorter;
    $P_w \rightarrow$ Weather API (Wind Speed and Direction);
    $P_{ele} \rightarrow$ We obtain the elevation of the cells, and calculate the
      distance $l$ between these two points;
    $P_{burn} \rightarrow$ The probability of a cell is on fire, using lat, long,
      and elevation of each cell. Even, the RGB Color in that
      position;
    if $State$ == $2$ then
        When a cell has vegetable fuel. It is checked if that cell
          can be turned (fire)
    else
        if $State$ == $3$ then
            This cell is burning and It has to change the state to
              burned.
        else
            Cell is not considered in the simulation step because it
              doesn't have vegetable fuel or was burned before.
        end
    end
until Each cell in a different position has been evaluated;
```

This is for each position, but, it is important to say that if the fire goes beyond the limits of a Google Tile, new matrices must be evaluated based on a new latitude and longitude.

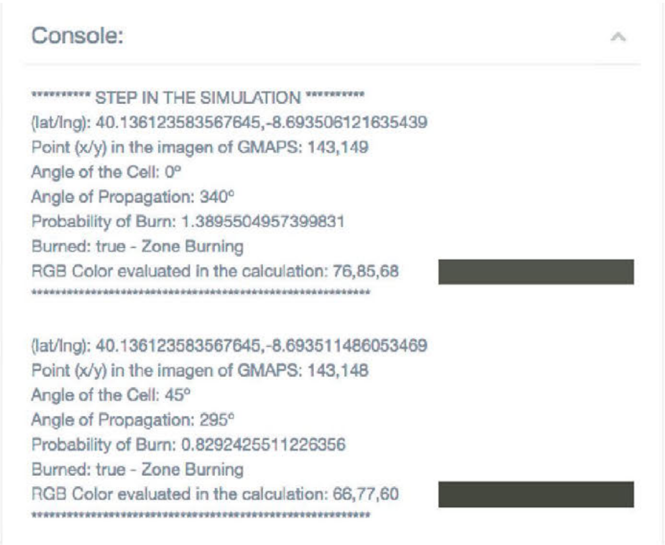


Success
Fire has started!!! Ohhh... FIRE!!! RUN...

Information
Fire has expanded!!! Ohhh... FIRE!!! RUN...

Fig. 6. Simulation progress

In fig. 7, a console is shown in the application, to know how the points and the different states are calculated (Free, Vegetation Fuel, burning, Burned).



Console:

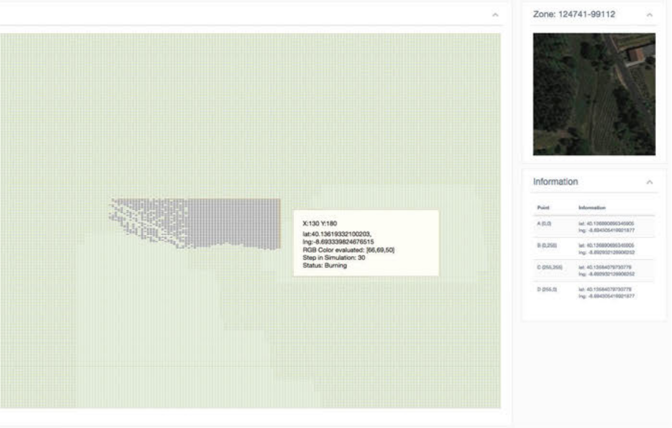
```
********** STEP IN THE SIMULATION **********
(lat/lng): 40.136123583567645,-8.693506121635439
Point (x/y) in the imagen of GMAPS: 143,149
Angle of the Cell: 0°
Angle of Propagation: 340°
Probability of Burn: 1.3895504957399831
Burned: true - Zone Burning
RGB Color evaluated in the calculation: 76,85,68
***************************************************************

(lat/lng): 40.136123583567645,-8.693511486053469
Point (x/y) in the imagen of GMAPS: 143,148
Angle of the Cell: 45°
Angle of Propagation: 295°
Probability of Burn: 0.8292425511226356
Burned: true - Zone Burning
RGB Color evaluated in the calculation: 66,77,60
***************************************************************
```

Fig. 7. Console to see how it goes the simulation progress

Maybe, in the fig. 6 cannot visualize very well how it goes the propagation, for this reason, in fig. 8, it shows the matrix for the Google Tile

$(124741 - 99112)$ with the main points of lat/long in $(0, 0) \rightarrow (40.136890695345905, -8.694305419921877)$, $(0, 255) \rightarrow (40.136890695345905, -8.692932128906252)$, $(255, 0) \rightarrow (40.13584079730779, -8.692932128906252)$ and $(255, 255) \rightarrow (40.13584079730779, -8.694305419921877)$.



Zone: 124741-99112

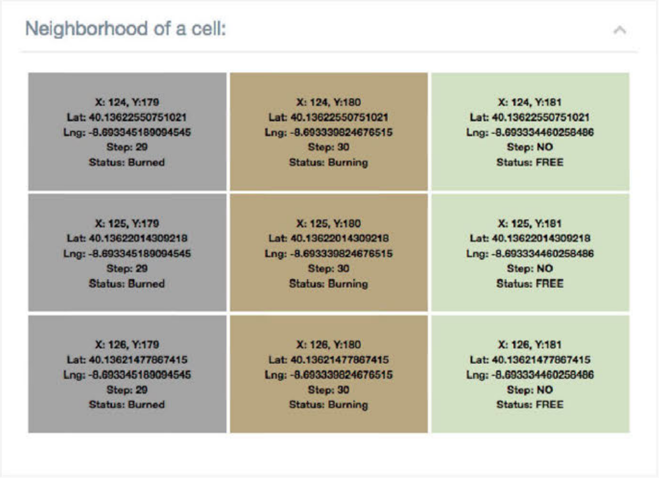
X:130 Y:180
lat:40.13619332100203,
lng:-8.693339824676515
RGB Color evaluated: [66,69,50]
Step in Simulation: 30
Status: Burning

Information

Point	Information
A (0,0)	lat: 40.136890695345905 lng: -8.694305419921877
B (0,255)	lat: 40.136890695345905 lng: -8.692932128906252
C (255,255)	lat: 40.13584079730779 lng: -8.692932128906252
D (255,0)	lat: 40.13584079730779 lng: -8.694305419921877

Fig. 8. Google Tile pixel mesh (124741-99112)

The fig. 9 shows different information about cells, e.g., The center cell is in $(x : 125, y : 180)$ with $(lat : 40.13622014309218, long : -8.693339824676515)$ and this cell was processed in the $step = 30$ and in that moment is burning. Also, It presents the neighborhood of this center cell where $3 \rightarrow burned$ in the $step = 29$, $2 \rightarrow burning$ in the $step = 30$ and $3 \rightarrow Free$, where the simulation will go in the next step depending on the wind direction.



Neighborhood of a cell:

X: 124, Y:179 Lat: 40.13622550751021 Lng: -8.693345189094545 Step: 29 Status: Burned	X: 124, Y:180 Lat: 40.13622550751021 Lng: -8.693339824676515 Step: 30 Status: Burning	X: 124, Y:181 Lat: 40.13622550751021 Lng: -8.693334460258486 Step: NO Status: FREE
X: 125, Y:179 Lat: 40.13622014309218 Lng: -8.693345189094545 Step: 29 Status: Burned	X: 125, Y:180 Lat: 40.13622014309218 Lng: -8.693339824676515 Step: 30 Status: Burning	X: 125, Y:181 Lat: 40.13622014309218 Lng: -8.693334460258486 Step: NO Status: FREE
X: 126, Y:179 Lat: 40.13621477867415 Lng: -8.693345189094545 Step: 29 Status: Burned	X: 126, Y:180 Lat: 40.13621477867415 Lng: -8.693339824676515 Step: 30 Status: Burning	X: 126, Y:181 Lat: 40.13621477867415 Lng: -8.693334460258486 Step: NO Status: FREE

Fig. 9. Neighborhood of the cell (40.13622014309218,-8.693339824676515)

Similarly, in fig. 10, the number of points per state that are being processed in the simulation is shown., e.g., $(Free = 63938, Burning = 90, Burned = 1478)$ that represents the $(97.61\%, 0.14\%, 2.26\%)$ respectively of the area defined by the Google Tile $(24741 - 99112)$.

The fig. 11 shows in that direction is the spread of fire through the forest, in this case, it is 315° and 45° with respect to the central point. This behavior can also be seen in fig. 8.

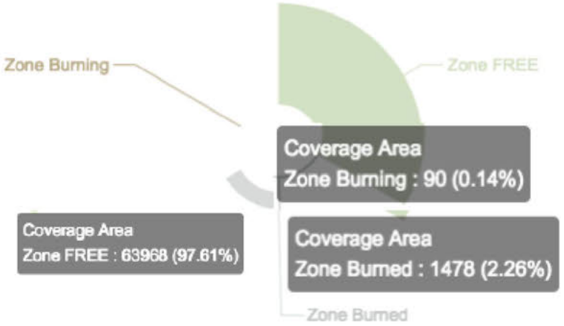


Fig. 10. Progress of the simulation (Number of points per state)

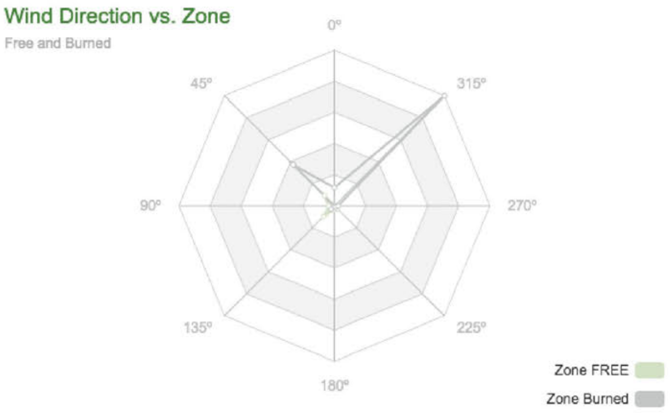


Fig. 11. Evaluation of where the fire tends to go

Finally, fig. 12 shows the step-by-step simulation process, indicating how many cells it evaluates in each step and generates a state for those cells.

VI. Discussions and Future Works

The purpose of creating a tool that allows evaluating the behavior of the fire in exteriors and even interiors (Future work), presents an opportunity for the governmental agencies in charge of this type of emergencies. The main idea about this prototype is knowing which areas could be affected in a forest fire and act immediately, as well as to identify the safe areas so firefighters can leave their belongings without fear that they may be victims of the fire. Besides, we are working on the implementation of a sensor network that allows us to monitor a specific area of a forest, and thus can more easily foresee the behavior of fires, because it will be possible to obtain information in real-time from a possibly affected area.

A critical point in the simulation process to improve is the RGB sorter, because depending on the RGB Color will be the value for P_{den}, P_{veg} and $State$, for this reason, currently, we are working to improve the RGB sorter using the work of Branson et al. [17], where they classify kind of trees with different images of Google Maps.

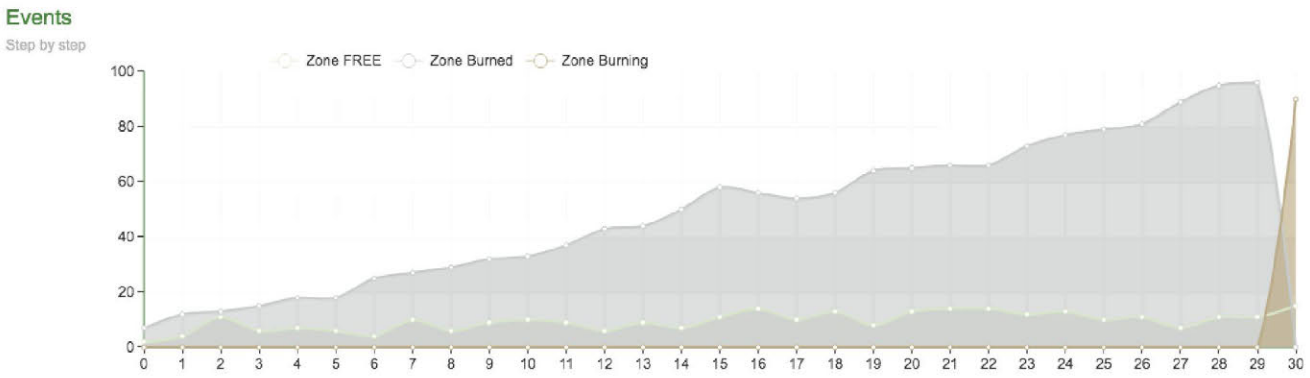


Fig. 12. Number of cells calculated in each step by simulation

ACKNOWLEDGMENT

Special thanks to the Institute of Applied Information Technology (InIT) of the School of Engineering at Zurich University of Applied Science (ZHAW) by the different equipment provided to the development of this project.

REFERENCES

[1] M. P. Thompson and D. E. Calkin, "Uncertainty and risk in wildland fire management: a review," *Journal of environmental management*, vol. 92, no. 8, pp. 1895–1909, 2011.

[2] S. ALBERTO and P. PATH, "Current 2018 forecast: Average to slightly above-average hurricane activity," 2018.

[3] M. Hosseini, V. Geissen, O. González-Pelayo, D. Serpa, A. I. Machado, C. Ritsema, and J. J. Keizer, "Effects of fire occurrence and recurrence on nitrogen and phosphorus losses by overland flow in maritime pine plantations in north-central portugal," *Geoderma*, vol. 289, pp. 97–106, 2017.

[4] R. C. Rothermel, "How to predict the spread and intensity of forest and range fires," 1983.

[5] M. A. Finney, "Farsite: Fire area simulator-model development and evaluation," *Res. Pap. RMRS-RP-4, Revised 2004. Ogden, UT: US Department of Agriculture, Forest Service, Rocky Mountain Research Station. 47 p.*, vol. 4, 1998.

[6] P. L. Andrews, "Behave: fire behavior prediction and fuel modeling system-burn subsystem, part 1," 1986.

[7] B. Chopard, *Cellular automata.* Springer.

[8] J. Quartieri, N. E. Mastorakis, G. Iannone, and C. Guarnaccia, "A cellular automata model for fire spreading prediction," *Latest Trends on Urban Planning and Transportation*, pp. 173–178, 2010.

[9] L. Bodrožić, D. Stipaničev, and M. Šerić, "Forest fires spread modeling using cellular automata approach," in *CEEPUS Summer School, Modern trends in Control*, 2006.

[10] I. Karafyllidis and A. Thanailakis, "A model for predicting forest fire spreading using cellular automata," *Ecological Modelling*, vol. 99, pp. 87–97, 1997.

[11] S. Yassemi, S. Dragićević, and M. Schmidt, "Design and implementation of an integrated gis-based cellular automata model to characterize forest fire behaviour," *ecological modelling*, vol. 210, no. 1-2, pp. 71–84, 2008.

[12] A. Lopes, M. Cruz, and D. Viegas, "Firestationan integrated software system for the numerical simulation of fire spread on complex topography," *Environmental Modelling & Software*, vol. 17, no. 3, pp. 269–285, 2002.

[13] A. Alexandridis, D. Vakalis, C. I. Siettos, and G. V. Bafas, "A cellular automata model for forest fire spread prediction: The case of the wildfire that swept through spetses island in 1990," *Applied Mathematics and Computation*, vol. 204, no. 1, pp. 191–201, 2008.

[14] E. Pastor, L. Zárate, E. Planas, and J. Arnaldos, "Mathematical models and calculation systems for the study of wildland fire behaviour," *Progress in Energy and Combustion Science*, vol. 29, no. 2, pp. 139–153, 2003.

[15] G. Svennerberg, *Beginning Google Maps API 3.* Apress, 2010.

[16] D. Vakalis, H. Sarimveis, C. Kiranoudis, A. Alexandridis, and G. Bafas, "A gis based operational system for wildland fire crisis management i. mathematical modelling and simulation," *Applied Mathematical Modelling*, vol. 28, no. 4, pp. 389–410, 2004.

[17] S. Branson, J. D. Wegner, D. Hall, N. Lang, K. Schindler, and P. Perona, "From google maps to a fine-grained catalog of street trees," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 135, pp. 13–30, 2018.