

Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling

Majid Hojati¹[0000-0001-7350-0055] and Colin Robertson²

¹ Department of Geography & Environmental Studies, Wilfrid Laurier University, Waterloo, Ontario, Canada mhojati@uwaterloo.ca

² Department of Geography & Environmental Studies, Wilfrid Laurier University, Waterloo, Ontario, Canada

Abstract. With new forms of digital spatial data driving new applications for monitoring and understanding environmental change, there are growing demands on traditional GIS tools for spatial data storage, management and processing. Discrete Global Grid System (DGGS) are methods to tessellate globe into multiresolution grids, which represent a global spatial fabric capable of storing heterogeneous spatial data, and improved performance in data access, retrieval, and analysis. While DGGS-based GIS may hold potential for next-generation big data GIS platforms, few of studies have tried to implement them as a framework for operational spatial analysis. Cellular Automata (CA) is a classic dynamic modeling framework which has been used with traditional raster data model for various environmental modeling such as wildfire modeling, urban expansion modeling and so on. The main objectives of this paper are to (i) investigate the possibility of using DGGS for running dynamic spatial analysis, (ii) evaluate CA as a generic data model for dynamic phenomena modeling within a DGGS data model and (iii) evaluate an in-database approach for CA modelling. To do so, a case study into wildfire spread modelling is developed. Results demonstrate that using a DGGS data model not only provides the ability to integrate different data sources, but also provides a framework to do spatial analysis without using geometry-based analysis. This results in a simplified architecture and common spatial fabric to support development of a wide array of spatial algorithms. While considerable work remains to be done, CA modelling within a DGGS-based GIS is a robust and flexible modelling framework for big-data GIS analysis in an environmental monitoring context.

Keywords: DGGS, Discrete Global Grid System, Cellular Automaton, Wildfire.

1 Introduction

With increasing sources of geographically referenced data such as growing satellite-based imaging archives, GPS tracking data, and transactional data stores; new challenges are emerging that relate to storing, accessing, analyzing, visualizing, and sharing big data as components of Geographical Information Systems (GIS) [1, 2].

One of the key challenges is the development of new data models capable of integrating data from different sources with varying levels of accuracy and uncertainty [3]. The term “Digital Earth” embodies a novel vision for managing and manipulating digital spatial information in the big data era [3–5]. Different architectures and data models have been proposed to realize a digital earth vision, including data cubes [6] and discrete global grid systems (DGGS) [7–9]. However, much work remains to be done to realize a digitally integrated ‘earth double’ system which can be used for understanding global and local change processes [10].

In most big spatial data architectures, data are assigned into discrete locations or cells in a grid system which is indexed and accessible to generic analytic algorithms. Each cell has a specific resolution based on the cell size which represents the spatial measurement uncertainty [3, 4]. DGGS implement this discrete conceptualization of space as a multiresolution representation of the earth, in which nested tessellations of the earth's surface form a hierarchy of possible resolutions for storing geographic data [7]. A DGGS can be characterized by the tessellation geometry, the method of indexing cells, and the quantization strategy and associated mathematical functions [8]. In a DGGS, cells can have different shapes such as hexagons, squares, and rectangles. Hexagonal grid shapes have attracted more attention due to their compact division of the space, isotropic and equidistant neighbors and smaller average error in quantizing the plane [11–13]. Having a multiresolution data structure requires methods for identifying parent and child relations between resolutions. Each parent can be constructed from a set of child cells, the ratio of which is a key characteristic of DGGS known as its aperture [14]. In hexagonal grid systems, there are six types of apertures, with 1-to-3, 1-to-4 and 1-to-7 being the most common [14]. There are several studies addressing the different indexing methods used for DGGS such as two-dimensional grid addressing (e.g. hexagonal coordinate system developed by [11]) or space-filling curve-based indexing (e.g. Z space-filling curve indexing method developed by [15]). **Fig. 1** shows different examples of characteristics of DGGS systems including cell shape and apertures.

Current studies in this field are largely focused on data model development for handling spatial and temporal data (e.g., [3], [15] and [7]), or proposing different tessellation systems for DGGS (e.g., [16], [3]). In recent years practical applications based on DGGS data models have been proposed, such as one using the rHealpix system [17], ohsomeHex which aimed to visually explorer the history of OSM data [18], PlanetRisk International DGGS [19], Fixed Rank Kriging (FRK) R package [20] and so on. However, most applications tend to focus on data storage, retrieval, and visualization over spatial analysis and modeling techniques. There remains significant work to evaluate the potential of using a DGGS-based data model for applying spatial and temporal analysis [10].

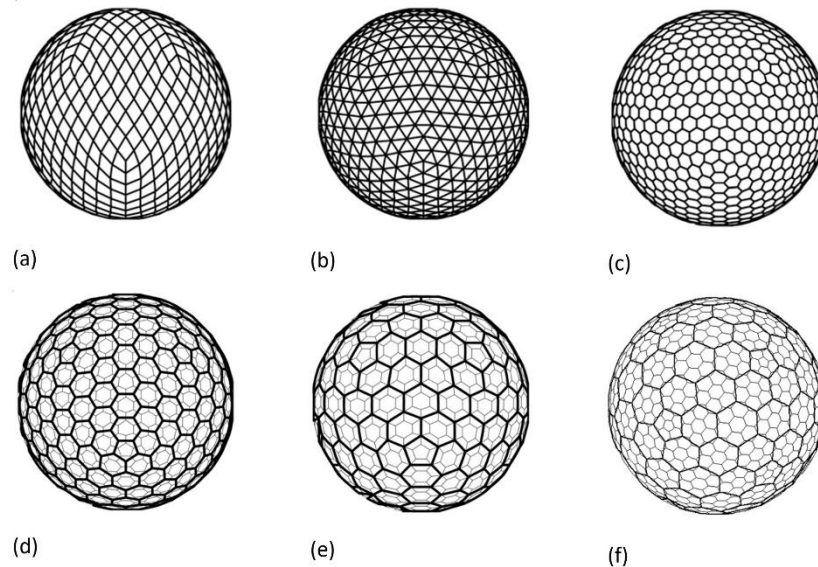


Fig. 1. - Different shapes and aperture for a DGG system a) Square shape grid, b) a Triangular grid, c) hexagonal grid, d) An aperture 3 hexagonal grid, e) an aperture 4 hexagonal grid and f) an aperture 7 hexagonal grid.

1.1 Background

2 Methodology

With the advent of big data, GIS has also entered into the big data GIS era [21]. Big geo-data handling can be broken down into data storage and organization methods, processing, analysis, as finally, insights and applications [1]. DGGs as a data model for big geo-data is able to store, manage and manipulate the large volume of heterogeneous data, including combining both vector and raster data into one spatial representation [8, 17].

In-Database Analysis was first proposed in the mid-1990s but did not get much attraction by mid-2000s [22]. In-database processing lets users blend and analyze large amounts of data without moving data out of the database to analytic applications. In a geospatial context, in-database approaches were developed separately for vector and raster data. With vector data, spatial databases evolved geometry data types which are now widely used to represent OGC/WKT spatial classes. Spatial database platforms such as PostGIS [23], Oracle Spatial [24] and ESRI Spatial Database Engine [25] have become integral parts of enterprise GIS infrastructure. In-database raster processing and analysis has also been developed

though is less common for analysis. Typically, raster data are stored as image or binary data types, with variable degrees of analytical methods available. Xie et al. [26] describes an in-database implementation of map algebra for Oracle's GeoRaster data type. An in-database DGGS data structure provides potential advantages over vector and raster approaches in that single data model can support a wide array of analytic algorithms with simple datatypes.

Cellular automate (CA) models, which have been used for dynamic modelling in GIS for decades, have not seen implementation within in-database systems. CA models with DGGS can provide a combination of space-time models in addition to providing the ability to generate complex spatial patterns using simple transition rules, resulting in advanced spatial modelling and simulation capabilities [27]. Classic examples of CA applications in GIS environments include wildfire modeling [28], land use change [29], and invasive species modelling [30]. Despite their wide use, conventional CA models have problems in defining simulation parameter values, transition rules and model structures [31]. Integration of CA models with big data and AI/machine learning models, has been proposed as a potential solution to these issues in recent studies such as neural network-based cellular automate by [31] or [32]. Developing data-mining models for extraction and evaluation of CA rules needs a platform to integrate data, and ability to handle heterogeneous and multiresolution geographical data. We propose here that combining a CA model and DGGS data model may enable the use of machine learning and other AI algorithms for rule set definition. In this paper, CA as a classic GIS modelling framework is implemented within a DGGS data model to examine the potential of using this data model for the analyzing large scale heterogeneous spatial data. This paper aims to couple a CA model with a DGGS data model in a distributed database system, to realize an in-database approach with associated computational advantages.

The main objectives of this paper are to: (i) investigate the possibility of using DGGS for running complex spatial analysis and environmental models rather than just storing and data integration, and to show the possibility of using non-geometric data types to address the spatial data related models, (ii) evaluate CA as a generic data model for dynamic phenomena modelling within DGGS data model and finally (iii) evaluate an in-database approach to CA modelling. For the purpose of creating and evaluating our in-database DGGS CA model, a recent wildfire in Canada was selected as a case study for wildfire spread prediction.

2.1 Cellular Automata

Cellular automata was first proposed by John Von Neumann and Stanislaw Ulam in early 1950s [33, 34]. CA are defined by a discrete dynamical system and a mathematical model, which consist of a very large number of simple elements, that operate in parallel and have a local interaction with neighboring elements [33]. The fundamental properties of a CA model are synchronicity which could be defined as the updating all the elements' state at the same time, uniformity meaning using the same set of updating rules for all the elements, and locality meaning the effect of rules are based on the local neighbors of each element [34]. It worth noting that due to the

use of a hexagonal DGGS grid, the equations and definitions that follow are different from those of a square lattice system.

Cells on a hexagonal grid.

A cellular automaton consists of a set of discrete elements or cells. These cells could be arranged in a one, two or n-dimensional grid and different shapes and characteristics, but homogeneity is always assumed in grid [35]. In environmental applications homogeneity can be defined as a set of cells with equal area [36, 37], where each cell points to a specific location on the earth. A cell (i, j) on a two-dimensional grid is a member of a finite grid of cells denoted as following:

$$(i, j) \in \{ (I, J) | 1 < I < I_{max} \text{ and } 1 < J < J_{max} \} \quad (1)$$

Where I_{max} and J_{max} are the maximum number of cells in grid and (i, j) is a cell on that grid.

Neighborhood.

In a CA grid system, each cell is surrounded by a set of neighboring cells. Neighboring cells could be adjacent cells to the central cell or they can be specified by an offset distance of the central cell [38]. A cell in each grid has a different number of neighbors depending on the tessellation geometry. For instance, in hexagonal grids each cell has 6 touching neighboring cells, which are equal distance from the central cell. On a square grid, each cell has eight neighbors with different distances between the central cell and its orthogonal and diagonal neighbors. For the purpose of this paper an aperture 3 DGGS hexagonal grid is used. In such a grid system based on different class of the grid, each cell (i, j) has six neighboring cells with the following indexes [19, 39]:

Class 1 neighboring cells for cell (i, j)

$$N_{(i,j)} \in \{ (i-1, j+1), (i-2, j-1), (i-1, j-2), (i-1, j-1), (i+1, j-1), (i+2, j+1), (i+1, j+2) \} \quad (2)$$

And for the class 2 neighboring cells are as following:

$$N_{(i,j)} \in \{ (i, j+1), (i-1, j), (i-1, j-1), (i, j-1), (i+1, j), (i+2, j+1), (i+1, j+1) \} \quad (3)$$

Where (i, j) is the cell coordinates, and $N_{(i,j)}$ is a set of neighboring cells around each cell. The class of data could be defined based on the resolution of the input data and construction of the neighborhood lookup table can be done in one-time process at the input data process step which is explained in section **Model implementation in Database..**

State of each cell.

Each cell at each discrete time step has one and only one state [35]. This state can change based on the neighboring cells' state in next time step. The state of a cell (i, j) at time t is a member of a set of states' S which can be finite or infinitive and can be defined as follows:

$$S_{(ij)}^t = \{s \mid s \in s_1, s_2, \dots, s_n\} \quad (4)$$

Where S_i is state of cell. In this work the state of cells is a finite set.

Local transformation rules.

In each time step, each cell updates its state based on the state transition function and a set of rules. The input of local transformation function is the state of the central and neighboring cells at time step t, and its output is a new state for central cell at time step equal to (t + 1). A local state transition function is denoted as $f: S_{ij}^t \rightarrow S_{ij}^{t+1}$ (5), which depending on the neighboring cells could be defined as following for class 1 neighborhood:

$$S_{(i,j)}^{t+1} = f(S_{(i-1,j+1)}^t, S_{(i-2,j-1)}^t, \dots, S_{(i+1,j+2)}^t) = f(N_{(i,j)}) \quad (5)$$

And the same for class 2:

$$S_{(i,j)}^{t+1} = f(S_{(i,j+1)}^t, S_{(i-1,j)}^t, \dots, S_{(i+1,j+1)}^t) = f(N_{(i,j)}) \quad (6)$$

Where $S_{(i,j)}^{t+1}$ is the state of the cell (i, j) at time (t+1), and $N_{(i,j)}$ is set of neighbours.

The update process must be *synchronous* and the function is applied to all cells at the same time [35, 38]. These transformation rules are defined based on each model's application and defining their form is the main scientific task of CA model-development.

Discrete time.

In CA model time is defined in a discrete manner. Time steps are integer clock ticks which starts from t=0, which is reserved for the initial state of the cells and increases by one at each iteration. The state of the cells in a grid only changes at discrete moments in time [34, 38].

Model implementation in Database.

The CA model reported here was implemented using an in-database approach. To do so, a relational data model was developed to store data in DGGs structure. Each DGGs cell is stored as a 4-tuple data row including DGGID, key, value and time. DGGID is a unique identifier which is calculated based on a space-filling curve [19]. A key/value structure was used for attribute storage. Due to vectoral nature of some environmental data such as wind, a long table structure is used to store data with direction. As the nature of CA model is iterative, where each state is related to the

previous cell's state, as well as its neighbors, it is necessary to store each iteration's result in the database. The results of each iteration are stored as a long table with different key and time values based on each iteration.

Data and Software Availability.

To run the CA model several software packages were used; R [40]; Dplyr [41] and dgggridR [42]. **Table 1** also shows the different datasets, which were used for wildfire spread modelling. These data were converted into the DGGS data model and stored in the database table structure. A Netezza IBM database engine was used as big geo data storage platform, however any relational database system could be used. Currently, due to security-related issues it is not possible to share any connection to this database application. For this reason, a small portion of the data, which is used to run the model, is stored as CSV data format with a working script, which are accessible in the following GitHub repository: <https://github.com/am2222/AGILECA>

Table 1. The data sets used for case study on wildfire.

Dataset	Resolution (spatial/temporal)	Retrieved parameters	Source/ License
Nasa Active fire data (VNIIRS)	approximate spatial resolution of 350m/ daily	Active fire data used for starting fire points	DOI: 10.5067/FIRMS/VIIRS/VNP14IMGT.NRT.001.
Copernicus Climate data	spatial resolution of these data is 0.1 degree / 1 hour	climate data: wind speed and direction	DOI: 10.24381/cds.e2161bac
Canada Dem	0.0002-degree spatial resolution	Elevation	https://open.canada.ca/data/en/dataset/7f245e4d-76c2-4caa-951a-45d1d2051333
National land cover dataset Landsat 8	0.0002-degree spatial resolution	Land Cover	Open Government License - Canada https://www.nrcan.gc.ca
(2016/05/03-2016/05/05-2016/05/12)	30 meters / 16 days	True color band composition to extract fire boundary	https://www.usgs.gov/landsat Landsat-7 image courtesy of the U.S. Geological Survey

3 Case Study: CA model for wildfire spread modelling

3.1 Case Study Overview

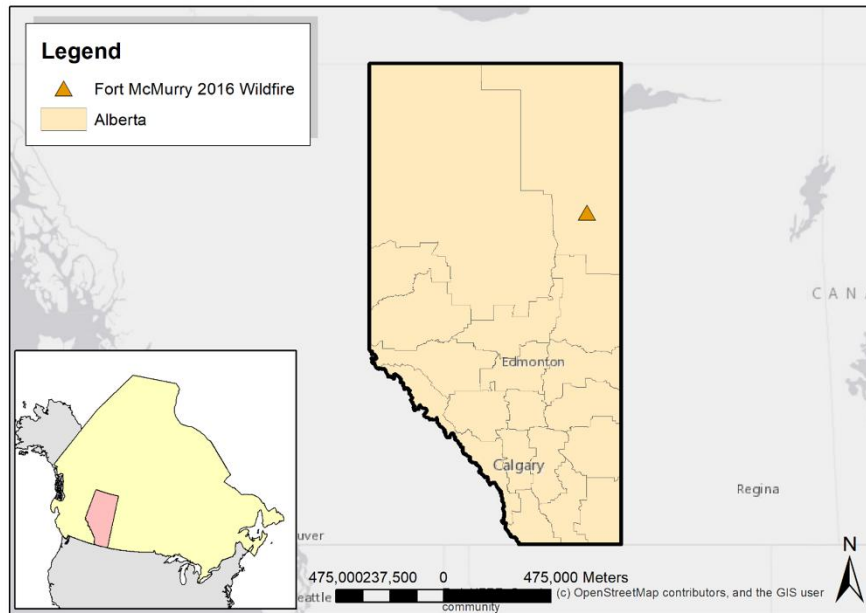
Wildfire modelling is an important information needed in many parts of the world due to its social, economic and environmental effects [43–45]. The history of wildfire spread modelling systems in Canada goes back to the models developed by J.G Wright in the 1920's aimed to keep daily track of the forest fires [46]. Later the forest fire weather index (FWI) which was developed by [27] mainly used meteorological variables including wind speed and rainfall to measure the forest fire danger in Canada. FWI is one of the first wildfire spread prediction models, and is categorized

as empirical, which uses fuel combustion and surface weather variables aimed to predict the fire spread speed and size of the flames [47]. During the evolution of empirical models scientists started to use more parameters such as elevation (e.g. work done by [48]) and tried to simplify the combustion and heat transfer of wildfires [46, 49, 50], which resulted in using more details from fire, combustion chemistry, heat transfer and fluid dynamics [51]. Using these parameters in wildfire modelling is known as physical modelling of fire behaviour [47, 52]. The third generation of fire models includes a combination of physical, empirical and other models such as those that simulate interaction between fire and environment, which causes changes in fire behavior over time [47].

Fire spread models from a mathematical approach are classified into cellular automata (CA) and vector models [37, 38]. The latter assumes that the fire spread front follows the growth law [37]. Cellular automation models were developed by Von Neumann, and for fire spread modelling applications scientists try to model fire fronts based on the behavior of the fires on a grid. Some CA models used to predict wildfire spread, are based on probability models (e.g. works by [53] and [54]) and some are based on fractals and historical data for model optimizations [55].

Study area.

The Fort McMurry wildfire (approximate location of 56°42'N 111°23'W) was one of the largest wildfires in Canada, in recent years [**Fig. 2 (a)**]. This fire started May 1, 2016 from southwest of Fort McMurry, Alberta, Canada. On May 3rd the evacuation of the area started and the provincial state of emergency started from May 4th. The fire was fully extinguished on August 2, 2017. It is estimated that the fire destroyed approximately 590,000 hectares, with an estimated damage of C\$9.9 billion, the most expensive natural disaster hazard in Canadian history of insurance providers [56]. **Fig. 2 (b)** shows the fire evolution from May 1st to May 12th extracted from Landsat 8 data as well as the study area and the location of the fire. As this figure shows, the fire first spreads toward west and northwest of its source and then spreads toward the Clearwater river to the east and east [56].



(a)



(b)

Fig. 2. - (a) the study area, (b) the fire evolution during the first 12 days of Alberta wildfire. (Natural Color Composition, Landsat 7)

CA model for wildfire spread modelling.

Fire behavior is a result of interaction between fire, land cover, vegetation attributes, climate attributes, landscape and ignition patterns [57]. Different studies have shown that some of the multi-scale climate parameters such as wind, relative humidity, air temperature, solar radiation and soil moisture can limit wildfire growth. They also show that there is constant pattern in many wildfires limiting fire growth unless particular environmental thresholds are met [58, 59]. Governing environmental parameters for wildfires have a spatial-temporal impact on their growth and treating a wildfire with a single-scale does not show a complete picture of fire behavior [59, 60]. The main variables determining the course of a wildfire could be categorized into

climate attributes, fuel characteristics and topography [59]. The spatial pattern of the fire is affected by the spatial heterogeneity of flammable fuels within the burn perimeter, wind direction and speed, and terrain slope.

Fig. 3 shows the overall CA model structure used in this work. The state transition function is defined based on each cell's characteristics and its neighbors. The fire spread speed is affected by the wind, slope and land cover. The spread speed for each cell is defined as follows:

$$R = R_0 \times W \times SI \quad (7)$$

where R_0 is defined as the initial spread speed as equation 9:

$$R_0 = LandCoverWeight \quad (8)$$

and W , SI and $fuel$ are wind and slope related indices. In this section first W and SI parameters are explained, and then the state transition function for the model is described.

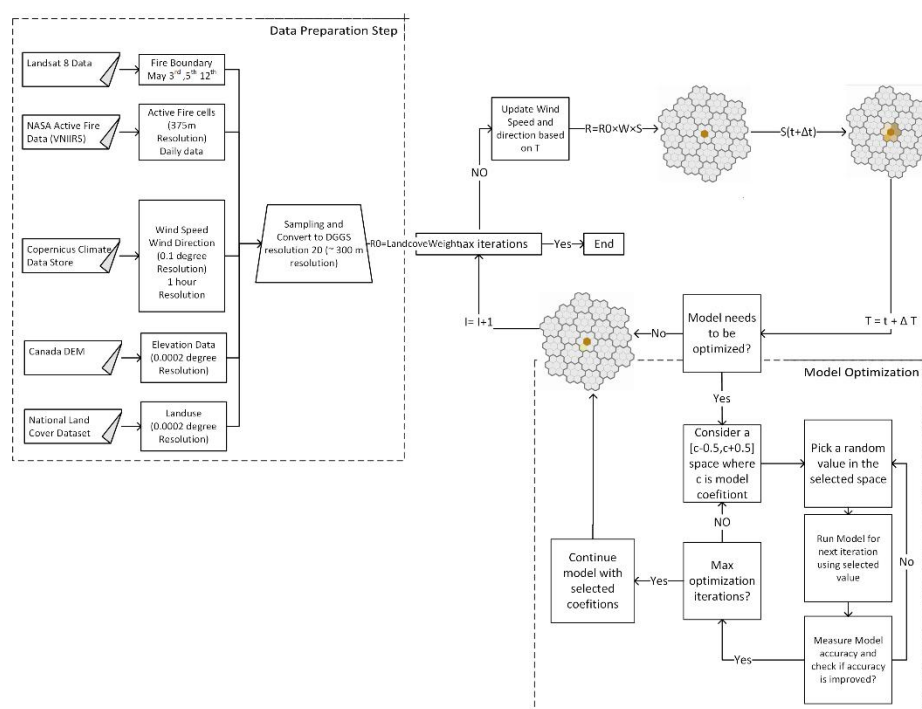


Fig. 3. - CA Model flowchart. T is time of Wind data and t is the time of CA model.

Wind speed/direction.

Wind vectors affect the spatial-pattern of the fire, and are one of the most important parameters in wildfire spread modelling [61]. Wind has a vectoral nature which incorporates both direction and magnitude. In square grids wind vectors are always decomposed to 4 main directions and 4 sub-directions which are defined based on the distance between the central cell and its neighbors, and each direction has a specific weight based on its distance to the central cell. In a hexagonal DGGs grid there are two main differences. First of all, all the 6 neighbors have the same central distance, so it is not possible to utilize the same model as the square grids, and all the cells in a hexagonal grid are considered with equal weight. Another difference of a DGGs grid is wind decomposition into six directions. In a DGGs system each cell has a degree of rotation based on its position on the globe's surface. Due to this rotation before decomposing each wind vector into 6 directions a rotation value is also applied to each cell. Let's consider V as the vertices in each hexagon as the following:

$$V_i = \{v \mid v \in v_1, v_2, \dots, v_6\} \quad (9)$$

Where i is the number of vertices. In each DGGs cell, generated using the `dggridR` library, first two vertices (v_1, v_2) construct an edge which is labeled as direction 1, and the edge constructed by the (v_2, v_3) are considered as the direction 2 and so on until we label all six edges of each hexagon. With bearing in mind that the edge 1 has general direction in each DGGs quad, the bearing value of this edge and the North direction (V component of wind) determines rotation of the DGGs cell in its quad. This value is calculated using the v_1 and v_2 coordinates as follows,

$$\theta = \text{atan2}(\sin \Delta\lambda \cdot \cos \varphi_2, \cos \varphi_1 \cdot \sin \varphi_2 - \sin \varphi_1 \cdot \cos \varphi_2 \cdot \cos \Delta\lambda) \quad (10)$$

where φ_1, λ_1 is the start point, φ_2, λ_2 the end point ($\Delta\lambda$ is the difference in longitude). Having θ as each cell's rotation value, it is possible to decompose wind v and u components into the six directions. **Fig. 4** shows the wind decompose method for each DGGs cell. The wind parameter in the final CA model is considered as W as follows:

$$W_i = e^{C_w \times \tan(\text{wind_magnitude}_i)} \quad i = 1, 2, \dots, 6 \quad (11)$$

Where C_w is a coefficient used to control the effect of the wind parameter on the entire model, and i is the wind direction. The method for storing wind decompositions in the database is explained in previous section.

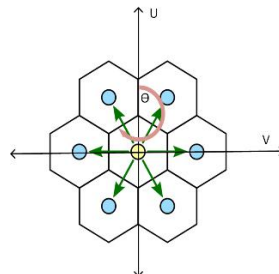


Fig. 4. - Wind decomposition into six main directions for a DGG cell, Θ is the bearing between wind north vector and first edge of each hexagon.

Slope.

Slope is one of the topographical parameters which impacts fire spread shape and direction. Fire shows a higher rate of spread uphill and a lower rate downhill [61]. To apply the topographic parameter on the CA model an elevation difference for each cell direction is calculated as follows:

$$\Delta h = (h_n - h_c) \quad (12)$$

Where h_n is the elevation of a neighboring cell in a specific direction and h_c is the central cell's elevation. The sign of Δh shows the uphill and downhill state in each direction. The slope parameter in the final CA model is considered as Sl as follows:

$$Sl = e^{C_s * \tan(\Delta h)} \quad (13)$$

Where C_s is defined as a coefficient. Many studies have used wind and slope as separate vectors with a different direction for each one, and the angle between two vectors is also considered in the calculations. In this work due to the basics of the grid, they are decomposed into six directions and the effect of the angle between slope and wind direction is considered as 0. Fire spread rate has a vectoral nature, which has direction. CA models consider a fire spread rate direction in each side of the central hexagon [62].

Land cover also has an effect on the fire spread speed and its spatial pattern. For example areas with urban development provide less flammable fuel than forests and as a result they are less flammable [63]. There are a number of studies that show the relation between different land covers and their impact on the fire spread speed. For instance, Vollmar [64] shows the highest correlation between fire spread speed and landcover is with shrub lands, then evergreen forests and the lowest correlation is with grass. In **Table 2**, the initial weights for each land cover is shown. This weight is used as initial parameter, which affects the fire spread speed inside a cell. In the current model the landcover effect is only considered in the initial fire spread speed, and does not change during the model. For urban areas and wetlands a weight equal to 0 is considered, and for the other land cover types a value between 0 and 1 is considered. A higher weight increases the spread speed within a cell in each iteration.

Table 2. Different initial weight for each land cover type adapted from [64]

Land cover	Weight
Urban area, Large Water bodies	0
Broadleaf evergreen forest	0.75
Broadleaf deciduous forest, Broadleaf deciduous forest, Mixed forest	0.5
Needleleaf forest, Grassland, Shrubland	
Others	1

State transition function.

Each cell's state is updated based on the following equation [53]:

$$S_{i,j}^{t+\Delta t} = S_{i,j}^t + \frac{(R_{i-1,j-1}^t + \dots + R_{i+1,j+1}^t)}{R_{max}} (R_{i,j}^t = 0) \quad (14)$$

where (i,j) are indices of each cell in the DGGS grid, and R_{max} is the maximum spread speed in the current iteration. Δt is time step, and is defined as the time, which takes fire to spread one cell's length by the speed of R . Δt is always equal to the time that it takes for at least one cell fully burns.

During the fire spread process, 5 different states for each cell are defined. The first state is 0, which is defined for unburned cells, these cells have enough fuel to start to burn. The next state value is 1 and is defined for cells in their early stage of burning. These cells are not able to spread flames to their neighbors. The next state is 2 and is defined as a cell, which is fully burned, and is able to spread fire to its neighbors. After this stage, a cell's state changes to 3, which means the cell is in its early stages of extinguishing. The next cell state is 4 meaning that the cell is fully extinguished, and it is not able to be burned again [53].

Fig. 5 shows the state transition rules, and the model evolution during different time steps. The model starts with a set of points, which are defined as fire ignition cells. The state of these points is 1, and the other cells are 0. In the next time step ($t_{t+1} = t + \Delta t$) the value of $S_{(i,j)}^{t+1}$ for each cell is calculated, and if it is higher than a threshold the cell state changes from 1 to 2, and it gains the ability to ignite its neighbours. This threshold is used to determine minimum percentage of each cell which is must be burned to consider that cell as a potential cell to transfer flames. In the next time step ($tt+1 = t + 2\Delta t$) the state of the cells with state of 0 changes to 1 (early burning) if they have at least a neighbor with state higher than 1. The state equal to 1 is considered as a connection between the status of neighbors and the central cell. If a cell has neighbors with state of 1 this cell keeps burning until the state of all its neighbors changes to a number higher than 2. In the same time other cells' state with the state of 2 changes to 3, and starts to extinguish. Transferring from the cell state 2 to the state of 3 is controlled by the number of burning cells around it. If a cell with the state of 2 has any early stage burning cells around it, with the state of 1, its state does not change until those cells fully burn (their state change to 2 and above). This controls the effect of burning cells in each neighborhood. At the next time step ($tt+1 = t + 3\Delta t$) the cells which their state is equal to 3 changes to 4 and fully burn. This step is used to slow down the burning process. The entire process of the burning a cell from state 0 to 4 can finish in 4 steps.

For the non-combustible cells (e.g. urban areas, large water bodies and etc.) the land cover spread weight is set to 0, resulting in a value of $S_{(i,j)}^{t+1}$ equal to 0, since this value is lower than the threshold so the state of cell always remains 0. The state of the cells at the beginning of the fire is retrieved from MODIS active fire data.

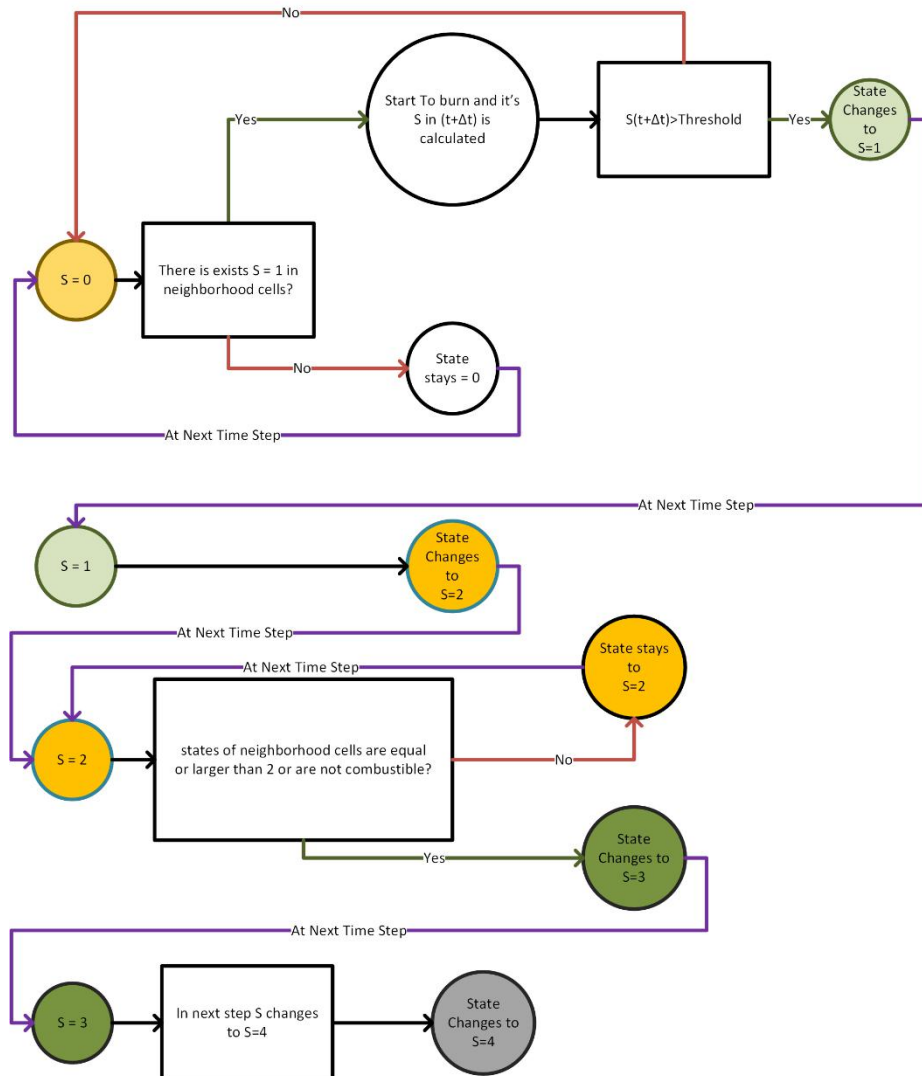


Fig. 5. - State transition rules and their change from one state to another. Red arrows are No state, Green arrows are Yes state and Purple arrows means moving to the next stage of the fire.

Mapping CA time interval to the actual fire's time interval.

As mentioned earlier, the time in the CA model is relatively defined by the cell state changes. For ideal CA modelling having less discrepancy between observation and iteration intervals is necessary [65]. The climate data for fire modelling are based on data and time of the day, so in order to use these data in dynamic wildfire modelling there must be a mapping method to connect the CA time into climate data time. To do

so, given the daily active fire points and the distance between the current CA model's boundary and existing active fires, the calculated distance is divided by the time difference between the current climate data's time and the active fire's time. The result is the temporal ratio for each CA iteration, which means after this number of iterations the value of time must be increased by one unit. This method assumes that in each iteration the fire boundary moves one cell toward the active fires in area.

Test Cases.

In order to analyze the sensitivity of the model a set of predefined test cases are designed, and the model is applied on these test cases with different values for each coefficient for each parameter. In each test case only one of the parameters is changed and the rest of parameters are remained constant. In this section the results of each test case are described:

Land cover (homogeneous and heterogeneous land covers).

For this test case other parameters such as wind and elevation are omitted. It is assumed that the wind value for all directions are constant (no wind) and the elevation does not change in the area. Then an area with two different land covers is used to run model. As **Fig. 6** (a) shows in the area with a homogeneous land cover (right side of fire-starting point) the fire spread speed is constant and fire spreads as a hexagon, but on a heterogeneous land cover the fire spread speed is affected by land cover type.

Wind.

In this test case a homogeneous land cover on a flat surface in addition to a wind in only one of six directions is assumed. **Fig. 6** (b) shows the model behavior for the different number of iterations (50,70 and 100). The Wind coefficient value is changed between 0 and 1, and shows a wind coefficient equal to 0 the effect of the wind is entirely cancelled, and the fire spreads in a hexagonal manner, while having a higher value of wind coefficient results in a longer fire front.

Elevation.

Elevation change between two neighboring cells has a high impact on fire spread speed. In this test case a homogenous land cover with no wind is assumed. The fire moves faster up-hill and has a slower spread speed downhill. **Fig. 6** (c) illustrates the effect of different values of slope coefficient on 50 iterations on an up-hill area. As it shows the fire spread speed in up-hills is faster than downhill and its speed and shape is controlled by the relative coefficient.

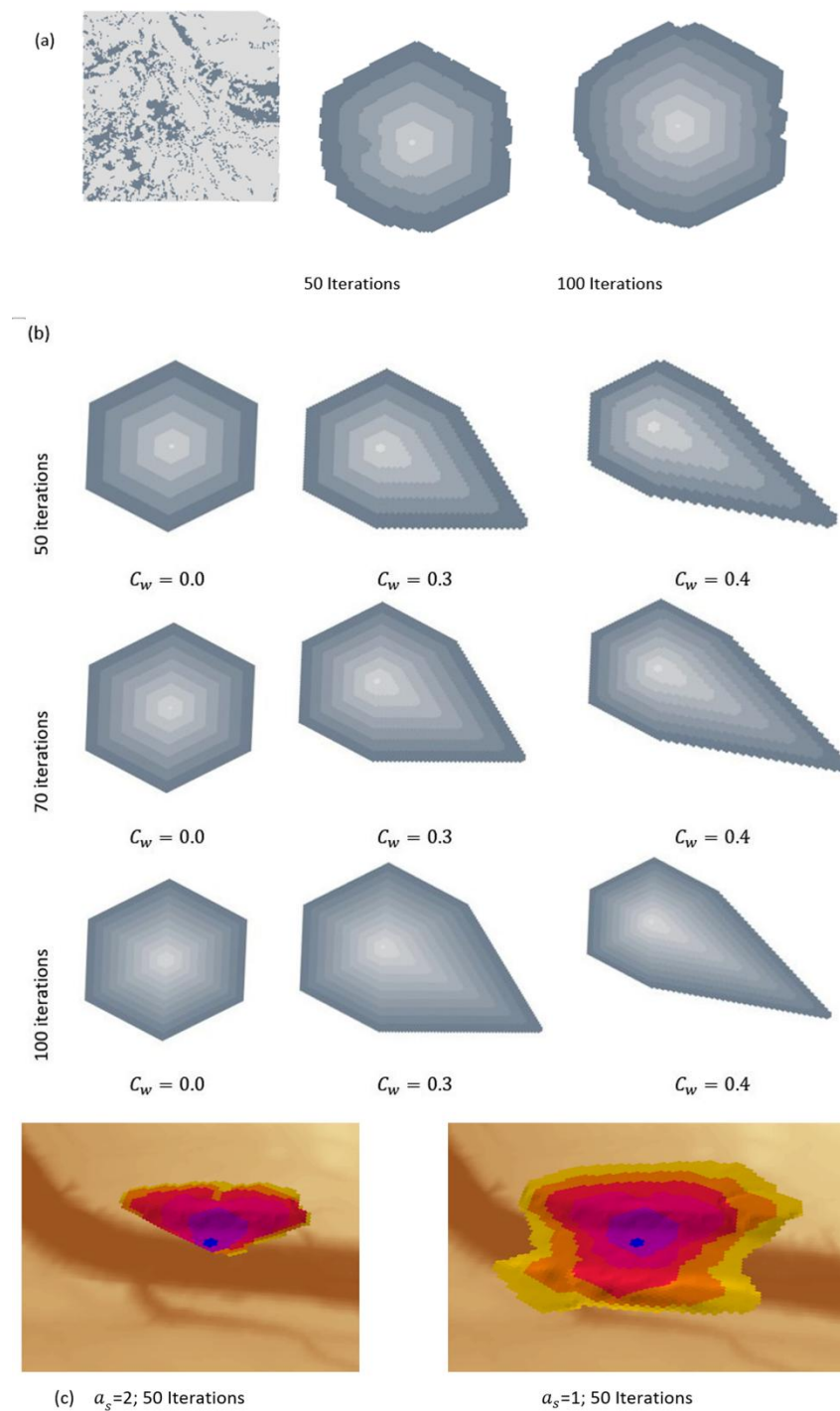


Fig. 6. - The different test cases and their effect on the model. (a) homogeneous and heterogeneous land covers. (b) the effect of the wind coefficient on one directional wind in different iterations (50,70,100). (c) the effect of up and down hill on fire spread speed in 50 iterations.

Coefficient Optimizations.

In order to optimize the coefficients, which are described in section 3-1 for each specific fire, in each iteration a 0.5 space around each coefficient is considered, and 100 random samples are selected from this space. Then the CA model is applied on the next iteration for each of 100 samples and every time model accuracy is measured. Using this method, the value of each coefficient is optimized during next iterations. The charts in **Fig. 7** show the optimization process of coefficients for wind and elevation in one of the iterations.

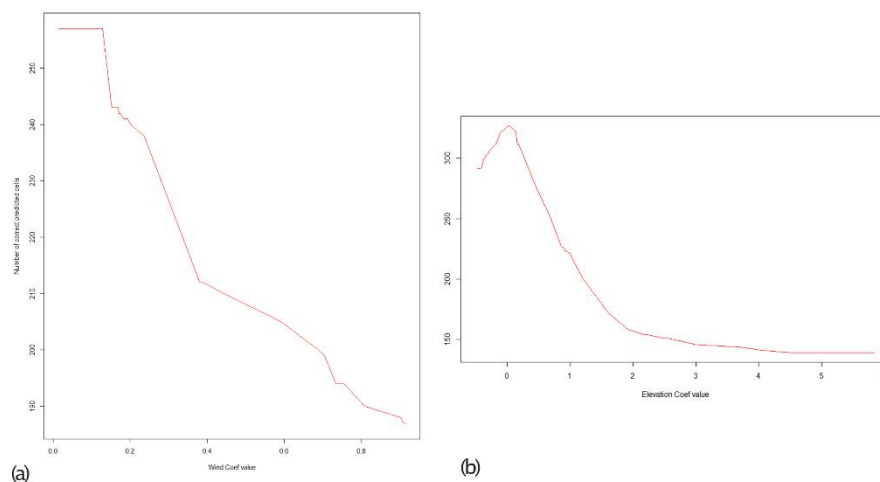


Fig. 7. - (a) Number of correct predictions for cells based on different wind coefficient values for 100 iterations in one time step, and (b) the number of correct predictions for cells based on different elevation coefficient values for 100 iterations in one time step

Model evaluation.

Fig. 8 shows the model result for first three days after starting the wildfire in Alberta, CA in 2016. In order to validate results, border of the fire is compared with the boundary extracted from Landsat images. In order to evaluate the model results, a confusion matrix is used. Two main classes of the burned area and unburned area are used to construct confusion matrix. For the reference data a boundary data extracted from the Landsat 8 imageries in conjunction with the fire boundaries, provided by the Alberta fire department is used. The confusion matrix generation is done in database using IBM Netezza analytic functions.

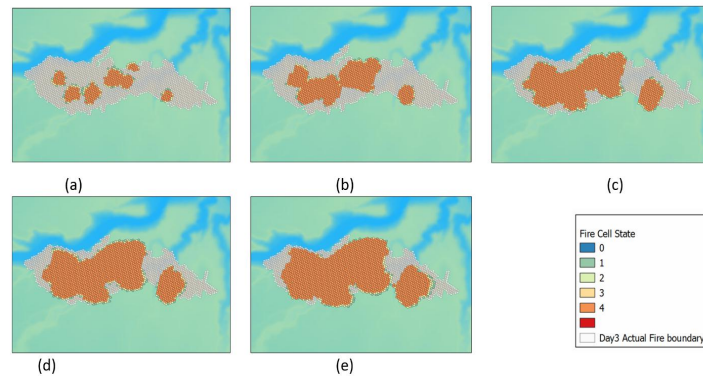


Fig. 8. - CA model output for Alberta wildfire in 2016. (a) after 20 iterations, (b) after 30 iterations, (c) after 40 iterations, (d) after 50 iterations and (e) after 60 iterations

4 Discussion

The DGGS framework provides the ability of using in-database approach for running dynamic models such as CA models for dynamic environmental phenomena. Using this approach, it is possible to perform complex modelling, sensitivity analysis and also apply optimization models inside a database without transferring data out of the database.

In traditional CA models which are used for environmental modelling the focus has always been on improving the ruleset definition. In recent years the focus of such models is moved from a regular CA model into use of AI and machine learning methods for ruleset definitions [31, 66]. Considering the ability of some database engines to run in-database analytic functions such as decision trees, regression models and so on, using a DGGS data model provides an ability to use such functions to train models and define the rulesets inside the database without the need of complex spatial algorithms.

In this work Alberta's wildfire is used as a case study to show the possibility of coupling in-database CA models with a DGGS data model. For this case study some sources of uncertainty can be identified as following: (i) the resolution of the climate data is relative to the resolution of the wildfire, which can cause a high level of uncertainty in running the model due to the effect of wind on wildfire behavior. Having high resolution climate data not only can solve this issue, but also can overcome the effect of the microclimate, which is generated by the wildfire. Another source of uncertainty is the lack of accurate fire boundaries. This can directly affect the model accuracy assessment, and the parameter optimization process. The other source of uncertainty is the method which is used for the mapping of time. In this paper it is assumed that in each CA iteration fire moves one cell toward the active fire points. This assumption must be treated with caution and can cause mismatches in

input data's time and CA time, resulting in inaccurate prediction of fire spread direction.

Regarding to the complexity of applying spatial functions on a DGGS grid, several issues that can be addressed. The first one is data quantization. In the process of data quantization, it is required to consider input data's uncertainty level. This step can cause lose of data and accuracy based on the selected method for quantization or the nature of data itself. The second noticeable issue raises based on desecrate nature of DGGS. Since DGGS grids are a collection discretized cells so there might be limitation in working with continues data and even representing such spaces like fuzzy space. The latest issue raises when performing spatial functions, all the recent spatial functions are optimized to work with coordinates and as a result dealing with desecrate space requires to develop desecrate spatial functions. As an example, an approach to apply buffer function is to consider buffer distance as discrete cells instead of a regular Euclidian distance.

Regarding to the second objective of this work as it is shown in this work using DGGS indexes it is possible to perform neighborhood relations for each cell and develop CA models based on these relations. However, it is required to have some lookup tables to store neighborhood relations in order to increase the model's performance. Such lookup tables can get larger by increasing the resolution of DGGS grid, and generating them is a time-consuming process. In addition due to the nature of CA based geographical models and their functionality, they have always been used in the platforms other than database engines, by using DGGS data model it is possible to use these models in a database. A DGGS grid is capable to integrate different datasets required for CA models and provides a backbone for dynamic geographic models.

In addition to the benefits of an in-database approach for doing analysis, one of the main profits of using DGGS data in a database is to enable users to perform spatial analysis without requiring to support complex geometry objects by database system. It is obvious that the majority of big data platforms lack in supporting spatial objects and functions (such as the different contributions for adding support for spatial objects to Hadoop platform by [67]) or their performance for supported spatial models is not satisfying [68]. These limitations raise the importance of using such none-geometric approach to perform spatial analysis in a database. DGGS indexes are only integer values representing specific portion of space, and major number of database systems provide support for statistical analysis on integer values. This approach provides us the ability of using different platforms, which are specifically developed for big data analysis, but does not support spatial object or analysis. However, in term of running the analysis there are a certain complexity in utilizing this type of analysis in a database due to the nature of SQL queries like the number of internal functions. But it is necessary to keep in mind that the recent improvements in analytic packages for database systems (such as Netezza analytic package) aimed to help increase in-database analysis use by providing more statistical and analytical functions.

5 Conclusion

Discrete Global Grid Systems aim to provide a framework for digital earth for data integration and management. DGG systems are different gridding systems, which have an equal area and shapes. This feature can provide the potential of using DGG systems as a base of CA models. They provide a homogenous spatial domain, in addition to an ability to integrate heterogeneous data into one data model, with resultant advantages for parallel and distributed analysis, and implementation of models for different scenarios and with different data models [69]. In this work a CA model is used to predict wildfire behavior for 2016 Fort McMurray in Alberta, Canada. The model is implemented entirely using an in-database approach and shows the possibility of running spatial models in a non-spatial database using DGGs systems. Also, it is shown that the DGGs data model can be used to integrate large amounts of spatial data and use them in different models. However, there are some limitations such as algorithm complexities or the lack of configuration procedures for in-database analysis.

6 Acknowledgments

The authors thank the Global Water Futures research program for funding this work under the Developing Big Data and Decision Support Systems theme. We also thank Dr. Chiranjib Chaudhuri for assistance with the model accuracy assessment and his comments that greatly improved the manuscript.

References

1. Yao, X., Li, G.: Big spatial vector data management: a review. *Big Earth Data*. (2018). <https://doi.org/10.1080/20964471.2018.1432115>.
2. Sabeur, Z., Gibb, R., Purss, M.: Discrete Global Grid Systems SWG.
3. Mahdavi-Amiri, A., Alderson, T., Samavati, F.: A Survey of Digital Earth. *Comput. Graph.* 53, 95–117 (2015). <https://doi.org/10.1016/j.cag.2015.08.005>.
4. Goodchild, M.F.: 1.03 - Big Geodata. In: Huang, B. (ed.) *Comprehensive Geographic Information Systems*. pp. 19–25. Elsevier, Oxford (2018). <https://doi.org/https://doi.org/10.1016/B978-0-12-409548-9.09595-6>.
5. Craglia, M., Goodchild, M.F., Annoni, A., Câmara, G., Gould, M.D., Kuhn, W., Mark, D.M., Masser, I., Maguire, D., Liang, S., Parsons, E.: Next-Generation Digital Earth (Editorial). *Int. J. Spat. Data Infrastructures Res.* (2008). <https://doi.org/10.2902/1725-0463.2008.03.art9>.
6. Purss, M.B.J., Peterson, P.R., Strobl, P., Dow, C., Sabeur, Z.A., Gibb, R.G., Ben, J.: Datacubes: A Discrete Global Grid Systems Perspective. *Cartogr. Int. J. Geogr. Inf. Geovisualization*. 54, 63–71 (2019). <https://doi.org/10.3138/cart.54.1.2018-0017>.
7. Sahr, K., White, D., Kimerling, A.J.: Geodesic Discrete Global Grid Systems. *Cartogr. Geogr. Inf. Sci.* 30, 121–134 (2003). <https://doi.org/10.1559/152304003100011090>.

8. Purss, M.B.J., Gibb, R., Samavati, F., Peterson, P., Ben, J.: The OGC® Discrete Global Grid System core standard: A framework for rapid geospatial integration. In: International Geoscience and Remote Sensing Symposium (IGARSS) (2016). <https://doi.org/10.1109/IGARSS.2016.7729935>.
9. Gibb, R., Raichev, A., Speth, M.: The rHEALPix Discrete Global Grid System. In: IOP Conference Series: Earth and Environmental Science (2016). <https://doi.org/10.1088/1755-1315/34/1/012012>.
10. Guo, H., Goodchild, M.F., Annon, A.: Manual of Digital Earth. Springer Singapore, Singapore (2020). <https://doi.org/10.1007/978-981-32-9915-3>.
11. Snyder, W.E., Qi, H., Sander, W.A.: Coordinate system for hexagonal pixel. In: Hanson, K.M. (ed.) Medical Imaging 1999: Image Processing. pp. 716–727 (1999). <https://doi.org/10.1117/12.348629>.
12. De Sousa, L., Nery, F., Sousa, R., Matos, J.: Assessing the accuracy of hexagonal versus square tiled grids in preserving DEM surface flow directions. In: Proceedings of ACCURACY 2006 - 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences (2006).
13. Ben, J., Li, Y.L., Zhou, C.H., Wang, R., Du, L.Y.: Algebraic encoding scheme for aperture 3 hexagonal discrete global grid system. *Sci. China Earth Sci.* 61, 215–227 (2018). <https://doi.org/10.1007/s11430-017-9111-y>.
14. Mahdavi-Amiri, A., Harrison, E., Samavati, F.: Hexagonal connectivity maps for Digital Earth. *Int. J. Digit. Earth.* 8, 750–769 (2015). <https://doi.org/10.1080/17538947.2014.927597>.
15. Bai, J., Zhao, X., Chen, J.: Indexing of discrete global grids using linear quadtree. *Geomatics Inf. Sci. Wuhan Univ.* (2005).
16. Kimerling, A.J., Sahr, K., White, D., Song, L.: Comparing geometrical properties of global grids. *Cartogr. Geogr. Inf. Sci.* (1999). <https://doi.org/10.1559/152304099782294186>.
17. Bowater, D., Stefanakis, E.: The rHEALPix discrete global grid system: Considerations for Canada. *Geomatica.* (2018). <https://doi.org/10.1139/geomat-2018-0008>.
18. Roick, O., Loos, L., Zipf, A.: A Technical Framework for Visualizing Spatio-temporal Quality Metrics of Volunteered Geographic Information. *Geoinformatik 2012.* (2012).
19. Sahr, K.: DGGRID version 6.4 User Documentation for Discrete Global Grid Generation Software. (2018).
20. Cressie, N., Johannesson, G.: Fixed rank kriging for very large spatial data sets. *J. R. Stat. Soc. Ser. B Stat. Methodol.* (2008). <https://doi.org/10.1111/j.1467-9868.2007.00633.x>.
21. Lee, J.G., Kang, M.: Geospatial Big Data: Challenges and Opportunities. *Big Data Res.* (2015). <https://doi.org/10.1016/j.bdr.2015.01.003>.
22. Grimes, S.: In-Database Analytics: A Passing Lane for Complex Analysis.
23. PostGIS: PostGIS 3.0.1dev Manual.
24. Oracle: Spatial Topology and Network Data Models Developer's Guide.
25. ESRI: Databases and ArcGIS.
26. Xie, Q.J., Zhang, Z.Z., Ravada, S.: In-Database raster analytics: map algebra and parallel processing in oracle spatial georaster. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* (2012). <https://doi.org/10.5194/isprsarchives-xxxix-b4-91-2012>.
27. Wagner, D.F.: Cellular automata and geographic information systems. *Environ. Plan. B Plan. Des.* (1997). <https://doi.org/10.1068/b240219>.
28. Ben, J., Li, Y.L., Zhou, C.H., Wang, R., Du, L.Y.: Algebraic encoding scheme for aperture 3 hexagonal discrete global grid system. *Sci. China Earth Sci.* (2018). <https://doi.org/10.1007/s11430-017-9111-y>.

29. Sathish Kumar, D., Arya, D.S., Vojinovic, Z.: Modeling of urban growth dynamics and its impact on surface runoff characteristics. *Comput. Environ. Urban Syst.* (2013). <https://doi.org/10.1016/j.compenvurbsys.2013.05.004>.
30. Ferrari, J.R., Preisser, E.L., Fitzpatrick, M.C.: Modeling the spread of invasive species using dynamic network models. *Biol. Invasions.* 16, 949–960 (2014). <https://doi.org/10.1007/s10530-013-0552-6>.
31. Li, X., Yeh, A.G.O.: Neural-network-based cellular automata for simulating multiple land use changes using GIS. *Int. J. Geogr. Inf. Sci.* (2002). <https://doi.org/10.1080/13658810210137004>.
32. Xu, T., Gao, J., Coco, G.: Simulation of urban expansion via integrating artificial neural network with Markov chain–cellular automata. *Int. J. Geogr. Inf. Sci.* 33, 1960–1983 (2019). <https://doi.org/10.1080/13658816.2019.1600701>.
33. Wolfram, S.: Statistical mechanics of cellular automata. *Rev. Mod. Phys.* (1983). <https://doi.org/10.1103/RevModPhys.55.601>.
34. Joel, L.S.: Cellular Automata: A Discrete View of the World. (2012). <https://doi.org/10.1002/9781118032381>.
35. Berto, F., Tagliabue, J.: Cellular Automata. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University (2017).
36. Trunfio, G.A.: Predicting Wildfire Spreading Through a Hexagonal Cellular Automata Model. *Lect. Notes Comput. Sci.* (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). (2004).
37. Hernández Encinas, L., Hoya White, S., Martín del Rey, A., Rodríguez Sánchez, G.: Modelling forest fire spread using hexagonal cellular automata. *Appl. Math. Model.* 31, 1213–1227 (2007). <https://doi.org/10.1016/j.apm.2006.04.001>.
38. Quartieri, J., Mastorakis, N.E., Iannone, G., Guarnaccia, C.: A Cellular Automata model for fire spreading prediction. In: *International Conference on Urban Planning and Transportation - Proceedings* (2010).
39. Sahr, K.: Location coding on icosahedral aperture 3 hexagon discrete global grids. *Comput. Environ. Urban Syst.* 32, 174–187 (2008). <https://doi.org/10.1016/j.compenvurbsys.2007.11.005>.
40. R Core Team: R: A Language and Environment for Statistical Computing, (2019).
41. Wickham, H., François, R., Henry, L., Müller, K.: *dplyr: A Grammar of Data Manipulation*, (2019).
42. Barnes, R.: *dggridR: Discrete Global Grids*, (2018).
43. Richardson, L.A., Champ, P.A., Loomis, J.B.: The hidden cost of wildfires: Economic valuation of health effects of wildfire smoke exposure in Southern California. *J. For. Econ.* (2012). <https://doi.org/10.1016/j.jfe.2011.05.002>.
44. Gill, A.M., Stephens, S.L., Cary, G.J.: The worldwide “wildfire” problem. *Ecol. Appl.* (2013). <https://doi.org/10.1890/10-2213.1>.
45. Brunson, M.W., Tanaka, J.: Economic and social impacts of wildfires and invasive plants in American deserts: Lessons from the Great Basin. *Rangel. Ecol. Manag.* (2011). <https://doi.org/10.2111/REM-D-10-00032.1>.
46. Van Wagner, C.E.: Development and structure of the Canadian forest fire weather index system. (1987). <https://doi.org/19927>.
47. Bakhshaii, A., Johnson, E.A.: A review of a new generation of wildfire–atmosphere modeling. *Can. J. For. Res.* 49, 565–574 (2019). <https://doi.org/10.1139/cjfr-2018-0138>.
48. Brown, J.K.: Handbook for Inventorying Downed Woody Material. USDA For. Serv. Gen. Tech. Rep. (1974).
49. Byram, G.M.: Combustion of Forest Fuels. For. Fire Control Use. (1959).

50. Rothermel, R.C.: No TitleA Mathematical Model for Fire Spread Predictions in Wildland Fuels. Draft. (1970).
51. Xu, J.: Heat Transfer in Combustion Systems. In: Handbook of Combustion. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany (2010). <https://doi.org/10.1002/9783527628148.hoc005>.
52. Simeoni, A., Salinesi, P., Morandini, F.: Physical modelling of forest fire spreading through heterogeneous fuel beds. *Int. J. Wildl. Fire.* (2011). <https://doi.org/10.1071/WF09006>.
53. Alexandridis, A., Russo, L., Vakalis, D., Bafas, G. V., Siettos, C.I.: Wildland fire spread modelling using cellular automata: Evolution in large-scale spatially heterogeneous environments under fire suppression tactics. *Int. J. Wildl. Fire.* (2011). <https://doi.org/10.1071/WF09119>.
54. Gouveia Freire, J., Castro Dacamura, C.: Using cellular automata to simulate wildfire propagation and to assist in fire management. *Nat. Hazards Earth Syst. Sci.* (2019). <https://doi.org/10.5194/nhess-19-169-2019>.
55. Clarke, K.C., Brass, J.A., Riggan, P.J.: A cellular automaton model of wildfire propagation and extinction. *Photogramm. Eng. Remote Sens.* (1994).
56. Statistics Canada: Fort McMurray 2016 Wildfire Economic Impact. (2018).
57. Barros, A.M.G., Pereira, J.M.C.: Wildfire selectivity for land cover type: Does size matter? *PLoS One.* (2014). <https://doi.org/10.1371/journal.pone.0084760>.
58. Slocum, M.G., Beckage, B., Platt, W.J., Orzell, S.L., Taylor, W.: Effect of climate on wildfire size: A cross-scale analysis. *Ecosystems.* (2010). <https://doi.org/10.1007/s10021-010-9357-y>.
59. Falk, D.A., Miller, C., McKenzie, D., Black, A.E.: Cross-scale analysis of fire regimes. *Ecosystems.* (2007). <https://doi.org/10.1007/s10021-007-9070-7>.
60. Higuera, P.E., Brubaker, L.B., Anderson, P.M., Hu, F.S., Brown, T.A.: Vegetation mediated the impacts of postglacial climate change on fire regimes in the south-central Brooks Range, Alaska. *Ecol. Monogr.* (2009). <https://doi.org/10.1890/07-2019.1>.
61. Karafyllidis, I., Thanailakis, A.: A model for predicting forest fire spreading using cellular automata. *Ecol. Modell.* 99, 87–97 (1997). [https://doi.org/10.1016/S0304-3800\(96\)01942-4](https://doi.org/10.1016/S0304-3800(96)01942-4).
62. Sharples, J.J.: Review of formal methodologies for windslope correction of wildfire rate of spread. *Int. J. Wildl. Fire.* (2008). <https://doi.org/10.1071/WF06156>.
63. Pyne, S.J., Andrews, P.L., Laven, R.D.: Introduction to wildland fire. Fire management in the United States. John Wiley & Sons, Ltd, New York (1996).
64. Vollmar, P.: The influence of climate and land cover on wildfire patterns in the conterminous United States, (2014).
65. Li, X., Yeh, A.G.O.: Data mining of cellular automata's transition rules. *Int. J. Geogr. Inf. Sci.* (2004). <https://doi.org/10.1080/13658810410001705325>.
66. Usta, T., Dündar, E.B., Korkmaz, E.E.: A cellular automata based classification algorithm. *ICPRAM 2019 - Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods.* 155–162 (2019). <https://doi.org/10.5220/0007373001550162>.
67. Eldawy, A., Mokbel, M.F.: SpatialHadoop: A MapReduce framework for spatial data. *Proc. - Int. Conf. Data Eng.* 2015-May, 1352–1363 (2015). <https://doi.org/10.1109/ICDE.2015.7113382>.
68. Robertson, C., Chaudhuri, C., Hojati, M., Roberts, S.A.: An integrated environmental analytics system (IDEAS) based on a DGGS. *ISPRS J. Photogramm. Remote Sens.* 162, 214–228 (2020). <https://doi.org/10.1016/j.isprsjprs.2020.02.009>.
69. Ding, Y., Densham, P.J.: Spatial strategies for parallel spatial modelling. *Int. J. Geogr. Inf. Syst.* (1996). <https://doi.org/10.1080/02693799608902104>.