Using GIS data in a m:n-AC^k cellular automaton to perform an avalanche simulation

Pau Fonseca i Casas[1], Santiago Rodríguez Fontoba[1]

[1]Barcelona School of informatics computing laboratory, Campus nord, Ed B6, Jordi Girona 1-3, 08034
Barcelona Spain
Tel. (+34)934017732 Fax (+34)934017040
pau@fib.upc.edu, e3449838@est.fib.upc.edu

1. Introduction

The purpose of this paper is to represent, in a virtual reality environment, the simulation of avalanche phenomena. We use GIS data to define the different elements that take part in the calculations. All these data can be modified and maintained in a common GIS. To use these data in the simulation engine a m:n-AC^k cellular automata is used. m:n-AC^k is a generalization of the classical cellular automaton allowing the use of different layers in a single cellular automata.

The paper first describes briefly the m:n-ACk cellular automaton structure, next we describe the GIS data used to define the avalanche model, and finally some results are presented.

2. Cellular automata

Cellular automata are discrete dynamical systems whose behaviour is completely specified in terms of a local relation (Emmeche C., 1998). Cells represent automaton space; time advances in discrete steps following "the rules", the laws of "automaton universe", usually expressed in a small look-up table. At each step every cell computes its new state in function of its closer neighbours. Thus, system's laws are local and uniform. Next figure shows one-dimensional cellular automaton initial state and successive two states after rules application.

Figure 1: One-dimension cellular automata.

3. Multi:N-Dimensional Cellular Automata (M:N-AC)

A multi:n-dimensional cellular automaton (m:n-AC) is a generalization of a cellular automata defined as follows (Fonseca et al. 2005):

Definition 1: m:n-AC^k

A multi n dimensional cellular automaton is a cellular automaton generalization composed by m layers with n dimensions each one.

The representation is:

$m{:}n{-}AC^k$

Where

---

m: is the automaton number of layers.

n: is the different layers dimension.

k: is the number of main layers (1 by default). A layer in a m:n-ACk is a main layer if a transition function Λ is defined in order to modify its state. A m:n-AC automaton only presents one main layer, while m:n-ACk automaton presents k main layers.

Since multiple layers belong to a single automaton, its state is defined as follows.

Definition 2: $E_m[x_1,..,x_n]$, layer m state in $x_1,..,x_n$ position

E_m is a function describing cell state in position $x_1,..,x_n$ of layer m.

E_m function allows state representation for each cell in the different layers of the automata, but this is not the global state of the automata. This state is represented by the *EG* function.

Definition 3: $EG[x_1,..,x_n]$, automata status in $x_1,..,x_n$ position.

EG returns automata global state in position georeferenced by coordinates $x_1,..,x_n$.

The global state of cellular automata depends on EG function in all automata positions.

Combination functions Ψ is represented by equation:

$$\Psi(E_1[x_1..x_n],.^{m-2)},E_m[x_1..x_n]) = EG[x_1..x_n] \tag{1}$$

In a common cellular automaton, evolution function allows global automata state change through cells value modification.

In a m:n-AC^k vectorial layers use makes necessary to generalize the neighbourhood and later define a new function that determines something similar to cell size (nucleus function).

Definition 4: Evolution Function Λm

Function defined for the layer m to modify its state through the state of others layers using combination function Ψ, and vicinity and nucleus functions.

Intuitively evolution function allows the representation of the modifications in this layer (modifications in nucleus area of a point $x_1.x_n$), using the state of other layers with combination function Ψ, and the vicinity area.

For more information about the vicinity and nucleus area and the vicinity and nucleus function, (Fonseca et al. 2005) can be reviewed.

4. Avalanche overview

An avalanche is a massive slide of snow, ice, rock or debris down a mountainside. Can be provoked by an earth tremor, extreme precipitation or man-made disturbances (such as a mountain skier). The impact of the falling material and the winds produced by the flow can cause extensive damage to anything in its path. In the next picture the avalanche fatalities in IKAR countries is shown (the IKAR countries considered are Germany, Norway, Slovakia, France, U.S.A., Austria, Switzerland, Italy and Canada). In the case of a snow avalanche, the new snow that accumulates on top of another heavy layer of snow can begin to slide down the mountainside. The risk of an avalanche can be reduced by building a snow shed — a barrier made of rocks, soil and other materials — or by triggering a controlled avalanche at a time when no one is on the mountain.
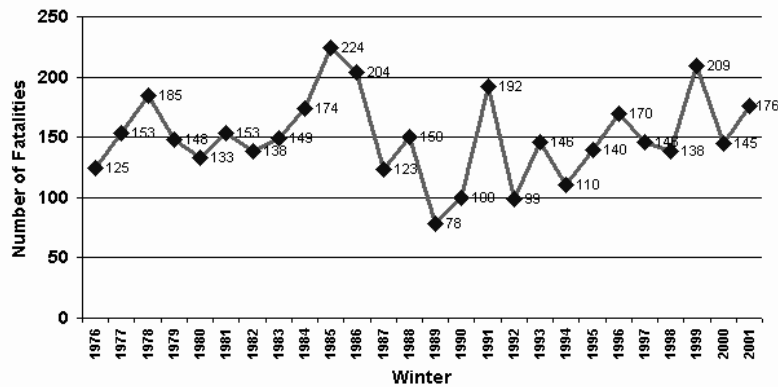
Figure 2: Avalanche fatalities in IKAR Countries (Avalanche 2007)

Minimal requirements for the occurrence of an avalanche are snow and an inclined surface, usually a mountainside. Most avalanches occur on slopes between 30 and 45°. Two basic main types of avalanches are recognized according to snow cover conditions at the point of origin.

1. A loose-snow avalanche originates at a point and propagates downhill by successively dislodging increasing numbers of poorly cohering snow grains, typically gaining width as movement continues down slope. This type of avalanche commonly involves only those snow layers near the surface. The mechanism is analogous to dry sand.

2. The second type, the slab avalanche, occurs when a distinct cohesive snow layer breaks away as a unit and slides because it is poorly anchored to the snow or ground below. A clearly defined gliding surface as well as a lubricating layer may be identifiable at the base of the slab, but the meteorological conditions, which create these layers, are complex.

In this model the slab avalanche is the only kind of avalanche considered.

5. Avalanche Model: 6+N:2-AC^{4+N} on Z^2.

$N \in [0,5]$, represents the maximum number of obstacles typologies that can be added to the model. Since all the parameters of the simulation are represented in these layers, is easy to perform different simulations representing different alternatives or situations. The problem to perform a simulation is reduced to find the layers that represent the area we want to simulate. No modifications in the simulation engine must be performed.

These data is stored following the IDRISI32 file format. To perform the avalanche simulation the data can be obtained from the ICC (Catalonia cartographic institute, http://www.icc.cat/portal/), from the Creaf (Center for Ecological Research and Forestry Applications, http://www.creaf.uab.es/mcsc/mms/index.htm) and from the Meteocat (http://www.meteocat.com/marcs/marc_muntanya.html). In the next table, the different layers used in the automata, and the source, are shown.

Name	Type	Description	Qtt	Source	Modifiable
Height	Raster	Layer representing the height of the environment.	1	ICC	No
Thickness of the snow	Raster	Represents the thickness of the "slab snow"	1	Meteocat	Yes
Floor features	Raster	Represents the kind of surface (rocks, sand, snow, ice,..). Each surface has his own specific rough parameter.	1	Meteocat Creaf	No
Snow that causes the slab features	Raster	Density, compactness of the snow.	1	Meteocat	Yes
Obstacles	Raster	Represents the obstacles that have the environment (small rocks, big rocks, houses, trees,…)	N	Creaf	Yes
Crack	Vectorial	Line representing the breakdown of the ice.	1	Input data	Yes, at beginning.
State of the snow	Raster	Shows the state of the terrain, **empty** (without snow that causes the windslab), **static** (contains snow that is stable) and **dynamic** (contains snow that in the next iteration is moving)	1	Meteocat	Yes

Table 1: GIS data used in the simulation model

5.1 Vicinity, nucleus and evolution functions

The Moore neighbourhood is used over Z^2. The figure 3 shows the representation of the cells relations. The vicinity and nucleus functions allowing the definition of the evolution function are based in the discrete topology over Z^2. The vicinity functions represents, in Z, what is the cells that must be taken in consideration to perform a calculation for the cell defined by (x_1,x_1), while the nucleus function represents the cell that must be modified due to the calculus.

Vicinity function: $vn(x_1,x_1) = \{(x_{1-1},x_{2-1}), (x_{1-1},x_2), (x_{1-1},x_{2+1}), (x_1,x_{2-1}), (x_1,x_2), (x_1,x_{2+1}), (x_{1+1},x_{2-1}), (x_{1+1},x_2), (x_{1+1},x_{2+1})\}$

Nucleus function: $nc(x_1,x_1)= \{(x_1,x_1)\}$

The evolution function (Λ) is based in the analysis of the global state ($EG(x1,x1)$) for a selected cell, through the combination function that merges the data contained in each one of the different automata layers ($\psi(E_1[x_1,x_1],E_2[x_1,x_1],..,E_{4+N}[x_1,x_1])$). The figure 4 shows a state machine representing the evolution function. Is not a complete description of the evolution function, but represents the three main states in witch each cell can be.
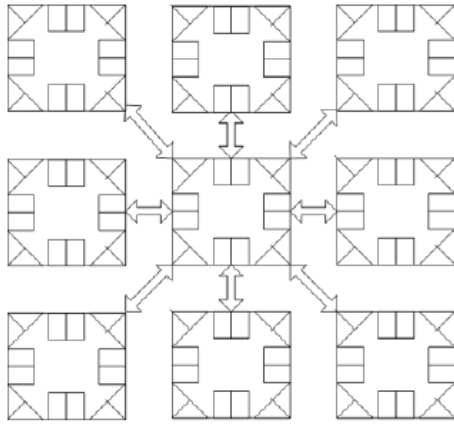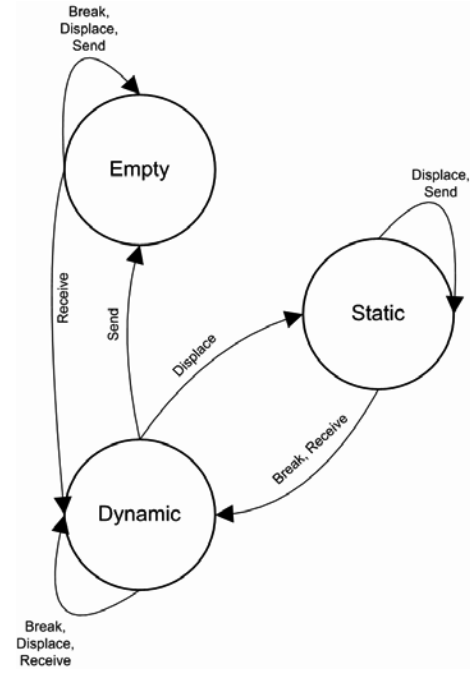
Figure 3: Moore neighbourhood

Figure 4: evolution function (Λ)

6. Results

The simulations of five different avalanche models have been conducted, obtaining data that can be compared with real avalanches data. In addition, a graphical tool, useful to understand the avalanche evolution has been developed. In the picture, a representation of an avalanche is shown:
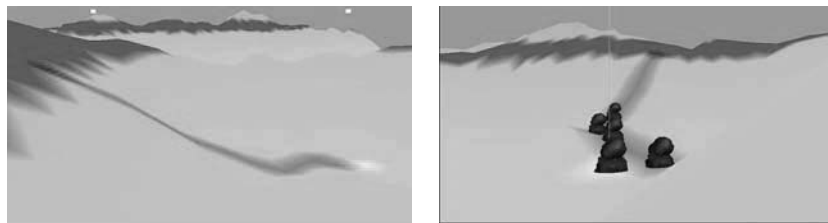

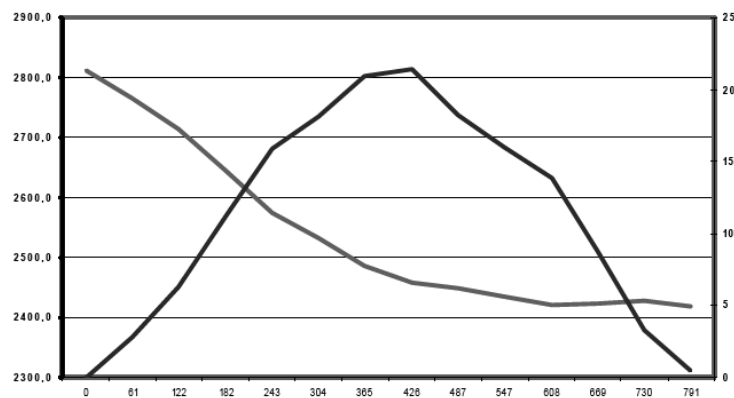Figure 4: avalanche representation

Figure 5: avalanche speed and height

In the figure 5 is shown a representation of the evolution of the snow speed in meters per second (blue line) and the height in meters (red line) for a simulation experiment.

7. Concluding remarks

An application to represent in virtual reality format the avalanche phenomena using GIS data thought the m:n-CAk cellular automaton is presented.

The comparison of the output data with studied phenomena for the speed variable shows promising results. Comparing the simulation output with (Leaf 1977; Mear 1976) we obtain similar results. In four different models, the maximum speed is about 21m/s (small or medium avalanche). In a test for a big avalanche, the maximum speed in the model is about 67m/s (little over the typical value for a big dimensions avalanche) (Leaf 1977; Mear 1976).

The structure in layers of all the elements that can interact with the evolution function of the avalanche, simplify the calculus of the evolution functions, allowing an easier implementation of the model, and a clear specification.
Since the layers represents all the model variables and are obtained from GIS data, the construction of a new simulation model is based on the definition of these layers, work that can be performed in a GIS, simplifying the modelling task.

8. References

Arregui Fernandez J., 1988, Topología Ed. UNED

Emmeche C.1998. Vida Simulada en el ordenador Ed. Gedisa.

Fonseca P.; Casanovas J., Montero J., 2004. GIS and simulation system integration in a virtual reality environment. Proceedings GISRUK 2004, 403-408

Fonseca P.; Casanovas J., Montero J 2004. A cellular automata and intelligent agents use to model natural disasters with discrete simulation. Proceedings EMS 2004.

Fonseca P.; Casanovas J. 2005. ESS205, Simplifying GIS data use inside discrete event simulation model through m_n-AC cellular automaton; Proceedings ESS 2005.

Law, A. M., Kelton, W. D. 2000. Simulation modelling and analysis. McGraw-Hill.

Leaf, Charles F., and M. Martinelli, Jr. 1977 Avalanche Dynamics: Engineering applications for land use planning. USDA For. Serv. Res. Pap. RM- 183, 51p. Rocky Mt. For. and Range Exp. Stn., Port Collins, Colo. 8052 1.

Mears, Arthur I. 1976. Guidelines and methods for detailed snow avalanche hazard investigations in Colorado. Colo. Geol. Surv. Bull. 38, 125p. Denver.

Voellmy, A. 1955. Uber die Zerstorungskraft von Lawinen. Schweiz. Bauzeitung, Jahrg. 73, S. 159-165, 212-217, 246- 249, 280-285. (In English as: On the destructive force of avalanches. 63p. Alta Avalanche Study Center, Transl. 2, 1964. On file at Rocky Mt. For. and Range Exp. Stn., Fort Collins, Colo.)

Avalanche 2007, http://www.avalanche.org/~uac/graf_world_ava.html (accessed 26/12/2006)

Biography

PAU FONSECA obtained his degree in computer engineer on 1999 from the Technical University of Catalonia. Now is full time professor of this university, teaching in statistics and simulation area. Also he works in the LCFIB (Barcelona informatics school computing laboratory) developing Simulation projects since 1998, and is member of LogiSim group dedicated to the research and simulation tools development.

SANTIAGO RODRÍGUEZ obtained his degree in computer engineer on 2006 from Technical University of Catalonia. Now he works in OS Sistemes as analyst programmer developing management and communication systems.