A Novel Visualization Environment to Support Modelers in Analyzing Data Generated by Cellular Automata

Philippe J. Giabbanelli(✉), Guru Jagadeesh Babu, and Magda Baniukiewicz

Department of Computer Science, Northern Illinois University, Dekalb, IL, USA
giabba@cs.niu.edu, {z1784615,z1791304}@students.niu.edu

Abstract. In the 'big data' era the attention is often on deriving models from vast amounts of routinely collected data, for example to lear about human behaviors. However, models themselves can produce a large amount of data which has to be analyzed. In this paper, we focus on visually exploring data produced by a type of discrete simulation models known as 'cellular automaton' (CA). In particular, we visualize two-dimensional CA with square cells, which can intuitively be thought of as a grid of colored cells. This type of CA is usually visualized using a slider to display the whole grid at each time of the simulation, but this can make it challenging to see patterns over the whole simulations because of change blindness. Consequently, our new visualization framework uses a temporal clock glyph to show the successive states of each cell on the same display. This approach is illustrated for three classical models using CA: an epidemic (a human health model), sandpiles (a self-organized dynamical system), and fire spread (a geographical model). Several improvements to the framework are discussed, in part based on feedback collected from trained modelers.

1 Introduction

In the 'big data' era, a lot of attention is devoted to processing massive datasets about humans (e.g., Medicare data, hospital discharge data, police calls), by using machine learning or by calibrating and validating digital human models. These models also produce massive datasets to analyze. In particular, they typically produce time series capturing changes from baseline to the end of a hypothetical intervention. While only the last point is seen as the "final result", both modelers and field experts often need to pay close attention to trends in the series. This can inform modelers of potential bugs in the implementation (e.g., identical consecutive pairs may indicate that results are mistakenly registered twice), while informing experts about the human dynamics (e.g., by observing cycles).

Research funded by the Department of Computer Science and the College of Liberal Arts and Sciences at Northern Illinois University.

V.G. Duffy (Ed.): DHM 2016, LNCS 9745, pp. 529–540, 2016.
DOI: 10.1007/978-3-319-40247-5_53

Consequently, many interactive visualizations have been developed for time series generated by simulations (Fig. 1). In this setting, time tends to be either linear (i.e., an ordered collection of time points) or branching (e.g., a simulation splits into 'branches' when there are several possible outcomes or competing hypotheses) [1]. While sliders can straightforwardly navigate through time, they lead to issues such as change blindness (i.e., some differences from one time point to another may be missed). Pixel visualizations [2] or glyphs [3] allow to visualize multiple time series on the same space. Several temporal glyphs have been designed (Fig. 2) and experimentally evaluated for tasks such as detecting peaks of trends [4]. While such innovative visualizations have adopted for simulations in engineering [5] (e.g., automotive, flows), there is a relative paucity of visualization environments for data generated by digital human models.

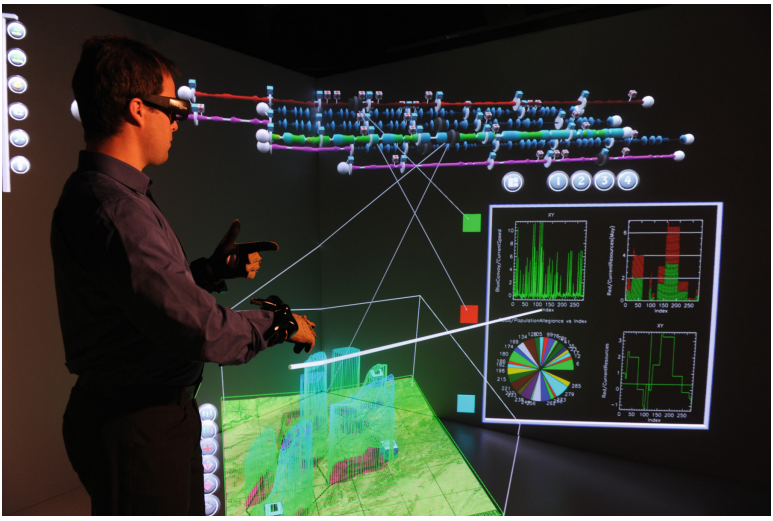

Fig. 1. The EXP V2 environment [6] allows to explore a simulation by hand gestures. *Reproduced with permission from Defense Research and Development Canada, who holds all intellectual rights.*

In this paper, we focus on digital human models implemented as cellular automata (CA). Intuitively, a cellular automaton is a collection of coloured cells on a grid that updates over a period of discrete, fixed time steps based on certain rules defined around neighbouring cells [7]. CA grids and cells can be of different types and shapes. Square and hexagonal cells are most common. CA models are generally one-dimensional (1D), two-dimensional (2D), and three-dimensional (3D), but can have more dimensions as well. There is a vast quantity of research using CA, as it can be applied to study any situation where individual units or cells affect others surrounding them [8]. In this paper, we focus on two-dimensional cellular automata with square cells, while noting that the same

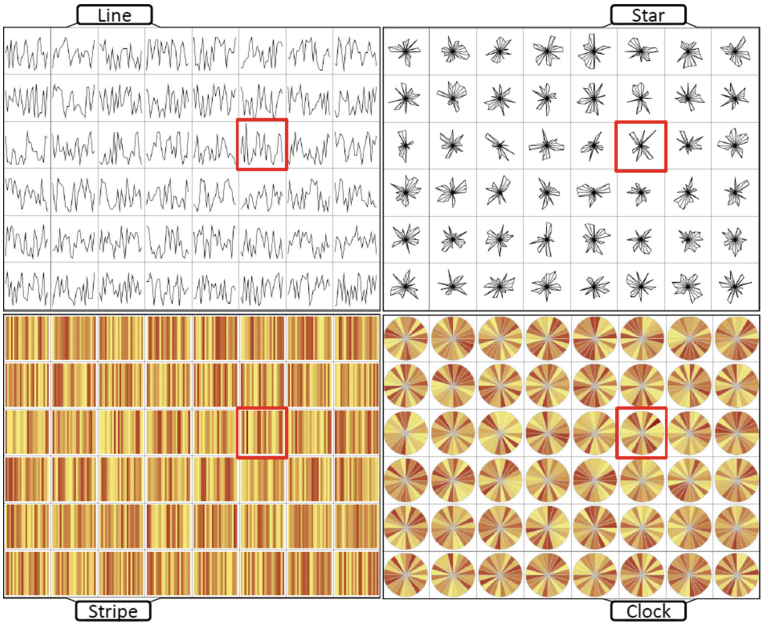

Fig. 2. Fuchs and colleagues compared different temporal glyphs for a dataset with continuous values [4].

principles would apply to other shapes of cells. Such CA are typically visualized by using a slider to move through the grid of states at different time steps (Fig. 3).

Our main contribution is the design and prototype implementation of temporal glyphs for cellular automata. This allows to see multiple time steps rather than going through each one via a slider. Our hypothesis is that this new visualization environment can contribute to providing better analytical capabilities, particularly when designed for the specific needs of modelers.

This paper is organized as follows. In Sect. 2, we introduce our visualization environment and explain how the data is rendered. In Sect. 3, the environment is illustrated for three well-known simulations (i.e., epidemics, sandpile, burning forest). Our hypothesis regarding the usefulness of this framework for modelers is discussed in Sect. 4 based on the feedback obtained from trained modelers. Finally, concluding remarks are provided in Sect. 5 together with a brief overview of future work.

2 Designing the Visualization Environment

Since CA have categorical values, line or star glyphs (Fig. 2; top) are not suitable. Either the stripe or clock glyphs could be used. They encode data values through colours (Fig. 2; bottom), and differ only in their encoding of time as either position (stripe glyph) or angle (clock glyph). Recent experiments found that the

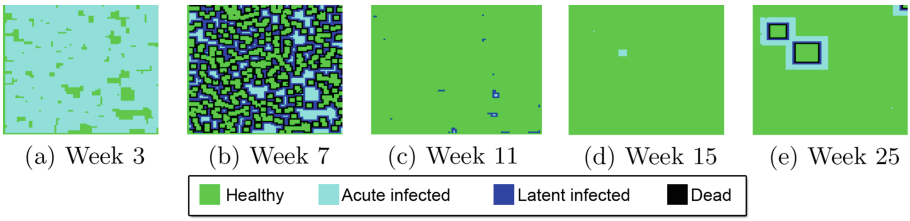

Fig. 3. Cells within the body were modeled via a two-dimensional cellular automaton to study the spread of HIV [9]. Each figure corresponds to CA at a specific time step. To visually explore disease progression, the modeler would use a slider and go through the weeks, displaying the CA of each week one after the other.

'clock metaphor' helps with chronological orientation, thus proving better than linear layouts to detect temporal locations, and triangular shapes performed better than rectangular ones to encode colours [4]. This suggests that temporal glyphs with a 'clock' ordering and triangular shapes have potential to support visualizing CA. Since users of CA are particularly familiar with square cells, we used square cells as clock glyphs (Fig. 3).

To understand the design challenges in using clock glyphs, we can think of the well-known pie chart. Having too many slices in a pie chart turns it from a meaningful visualization into an abstract pattern. The number of slices is thus best kept small, for example by collecting small slices within an 'other' category. A clock glyphs with too many data points would thus be like a pie chart with too many categories. This problem is particularly salient in our situation, since each glyph would have to represent the successive states taken by a cell over all time steps of the simulation, and there may be more time steps than could even fit within a circle (i.e., 360 slices). To address this problem, we limited the clock glyph to have either 4, 8 or 16 equal-sized partitions and each partition attempts at displaying the most relevant state within the corresponding segment of the data. This is illustrated in Fig. 4. In Fig. 4(a) we use 8 partitions and there happen to be exactly 8 time steps in the simulation, so each value is mapped to one partition. In Fig. 4(b) there are more values than partitions, so each partition represents the most frequent[1] state among multiple data points. Consequently, the visualization depends on the number of partitions (4, 8, or 16) and on the aggregation method (e.g., most frequent value). Both are set by the user in the current prototype.

Research suggests that "multiple views are particularly helpful in analyzing time-oriented data" [1]. Consequently, another design consideration was to allow working across multiple data representation. Given that the glyphs need a significant amount of space to display each cell, our goal was to have a complementary representation that takes limited space and provides a higher level of

[1] If the top frequency is found in multiple states, then ties are solved by picking the first one. For example, if there are 1 'dead', 2 'susceptible', and 2 'infected' then 'susceptible' would be picked.

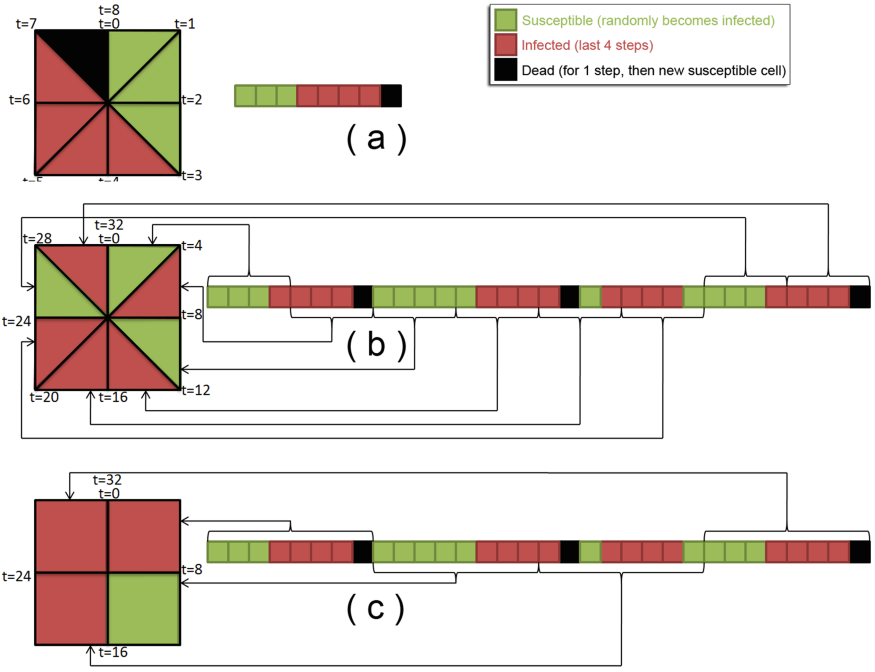

Fig. 4. Data produced by a disease model (top-right: states and transitions) may be entirely visualized if the model is ran for a few time steps (a). As the number of time steps grows, they are aggregated into each of the 8 (b), 4 (c) or 16 possible segments.

abstraction. This was fulfilled by using a flow diagram as secondary view. Flow diagrams are the most common depiction of cellular automata models; that is, a modeler using CA would immediately recognize and know how to interpret a flow diagram. In short, a flow diagram shows each state, and possible transitions between states. Formally, a flow diagram is a directed graph where each state corresponds to a node, and an edge exists from node a to node b if a cell can transition from state a at time t to state b at time $t + 1$. Our prototype automatically generates the flow diagram from the trace file (i.e. dump of simulation data).

3 Application to Classical Cellular Automata Models

3.1 Epidemics

In compartmental modeling, the population is divided into several groups or 'compartments', and then transitions or 'rules' specify the flows. The underlying mathematics are described in details by Hethcote [10]. Compartmental models of epidemics are typically named after the transitions between compartments: for example, in the SIS model an individual starts susceptible, can become infected,

and eventually becomes susceptible again; similarly, in the SEIR model, an individual starts susceptible, can be exposed to a disease, then becomes infectious, and eventually recovers.

While a compartmental model represents a population, it can be ran on a cellular automaton where each cell stands for an individual. In this case, the rules reflect how infections can be passed on between neighbouring cells. This approach has been widely used. For example, there are cellular automata models of the classical SIR model [11] or SIS model [12]. In this example, we used the SIR model where an infected cell has a probability $p_i = 0.4$ of transmitting the disease to a healthy cell, and an infected cell has a probability $p_r = 0.5$ of recovering.

Visualizations of simulation traces from this model are shown in Fig. 5, with different grid size (10 by 10 or 25 by 25) and different number of segments per cell (4, 8, or 18). The flow diagram is automatically generated by the visualization environment, and names or colours can be changed by the user. A consequence of displaying the most frequent state within each segment is that, as the number of segment decreases, some transitions are not visible. For example, we can see that cells get infected multiple times with 18 segments (Fig. 5-c), less so with 8 segments (Fig. 5-b), and not at all with 4 segments (Fig. 5-a). Similarly, some states may not be visible: using 4 segments (Fig. 5-d) instead of 8 (Fig. 5-e) would tend to under-estimate the spread of the disease as peripheral cells that were recently infected do not yet display this infection.

3.2 Sandpile

The sandpile model is a vehicle to illustrate the theory of Self-Organized Criticality (SOC), that is, the idea that large interactive systems self-organize into a critical state and that small perturbations in this state trigger chain reactions. Informally, one can build a pile of sand by adding one grain at a time, until reaching a critical point where adding a single more grain causes an avalanche. This model was introduced by Bak and colleagues in 1987 [13]. We implemented the Sandpile model as described by Athanassopoulos and colleagues [14]. There is only one parameter p which applies when two grains are above two empty cells: the configuration either remains as such (with probability p), or both grains fall in the cells below (with probability $1 - p$). In our example, we used $p = 0.5$.

Visualizations of simulation traces from this model are shown in Fig. 6, with different grid size (10 by 10 or 25 by 25) and different number of segments per cell (4, 8, or 18). All visualizations display that grains gradually fall and a stack of filled cell increases from the bottom. This would appear to be too simplified with 4 segments, and perhaps excessively detailed with 18 segments. The visualization with 8 segments could thus offer an interesting trade-off.

3.3 Fire Spread

The mathematical principles of fire spread were summarized by Rothermel in 1972 [15]. Cellular automata have since been abundantly used to model fire

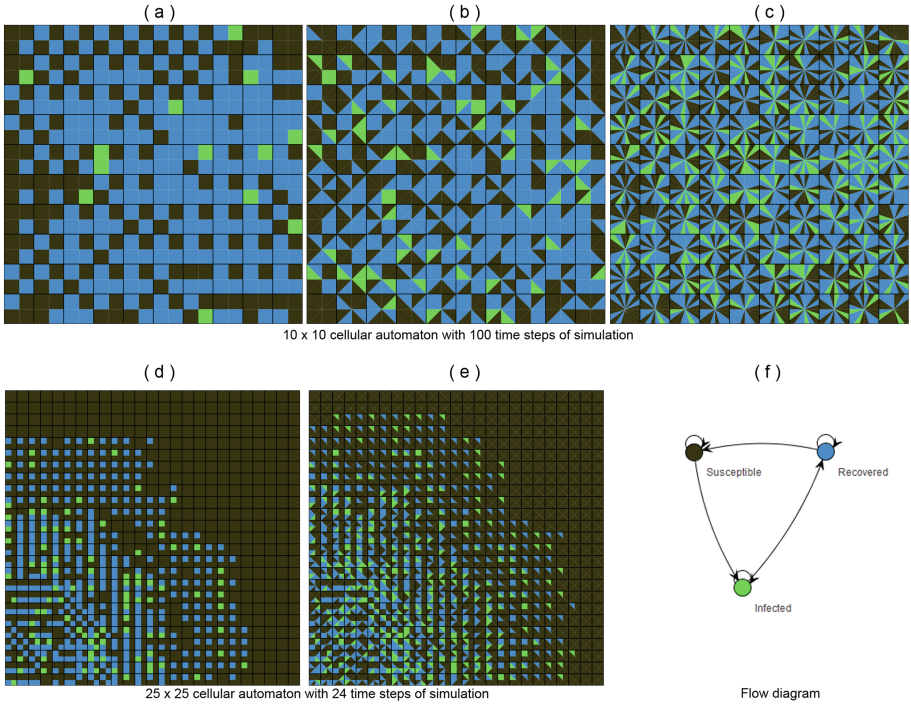

Fig. 5. Visualization of an epidemic over a 10×10 CA where each cell has 4 segments (a), 8 segments (b) or 18 segments (c). The same model is visualized over a 25×25 CA with 4 (a) or 8 (b) segments. The same flow diagram applies to all simulations and is automatically generated (f).

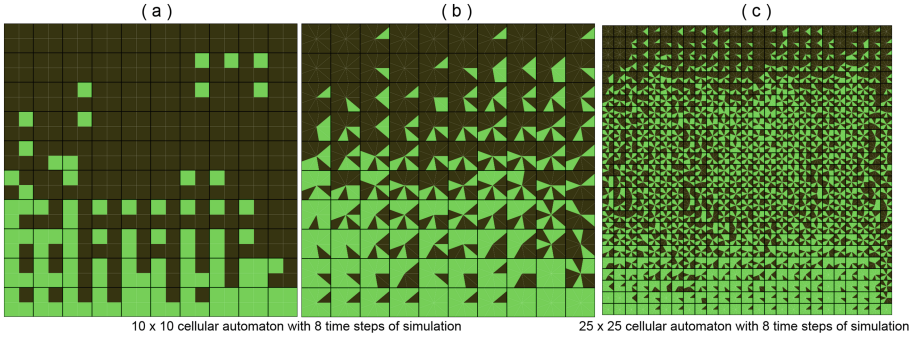

Fig. 6. Visualization of an sandpile over a 10×10 CA where each cell has 4 (a) or 8 (b) segments. The same model is visualized over a 25×25 CA with 8 (c) segments.

spread within a spatial context, captured using either square [16,17] or hexagonal cells [18,19]. In its simplest version, each cell has three possible states: empty, tree, or burning. Initially, each cell is empty with probability $p = 0.3$ or a tree with probability $1 - p = 0.7$; the fire is started by picking one cell as burning. At each time step, a tree burns if at least one neighbour is burning, or has a probability 0.001 of spontaneously burning. A burning tree turns into an empty cell after 1 time step, and an empty cell can turn into a tree with probability 0.1. Visualizations of simulation traces from this model are shown in Fig. 7, with a grid of 25 by 25 and different number of segments per cell (4, 8, or 18). In this simulation, no large component formed, thus there were random sporadic and isolated fires. Since the fire lasts only time step and only the most frequent state is displayed, the fire is never visible. Thus, it is implicit that a cell transitioned from light green (tree) to dark green (empty) because fire occurred. Having the most segments (i.e. 18) shows that almost all cells have been occupied by a tree at some point, which is gradually lost as the number of segments decreases.

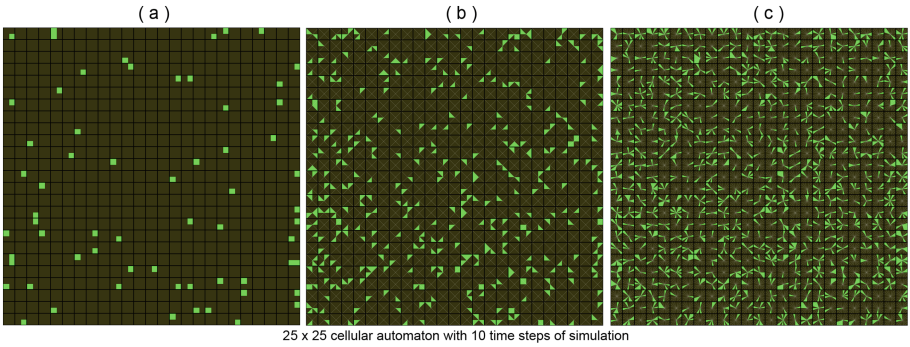

Fig. 7. Visualization of an forest fire over a 25×25 CA where each cell has 4 (a), 8 (b) or 18 (c) segments.

4 Feedback from Modelers

Two trained modelers were contacted to provide feedback on the prototype. On the positive side, the overall idea of avoiding a slider was well-liked. One modeler stated: "I like it a lot because it simplifies visualization of states across timesteps". On the negative side, a modeler reported that it became quite hard to read with a large number of cells or segments per cell. Having 8 segments was considered the most readable version. Several improvements were suggested, falling into three categories. First, even if the motivation for this visualization was to avoid sliding through time, a slider was deemed useful to allow modelers to narrow the range of time steps that are displayed. In other words, modelers agreed that a slider to move through a *single* time step was not as effective as our visualization, but they recommended being able to move through a *range* of time steps. This is in line with the visual information-seeking mantra of starting

with an overview, and then having details on demand: narrowing the time range would increase the detail of the cells since their segments would represent a narrower set of values. Using range sliders to narrow the data of interest was also done in the 2-d matrix-based interactive visualization by Song *et al.* [20].

Second, modelers appreciated the flow diagram and offered several way to better link it with the main visualization. For example, hovering over a state or transition in the diagram should highlight all the cells that include that state of transition. Conversely, selecting a cell or group of cells should update the diagram to show only the transitions relevant to the selected cell(s). The idea that selections in one view would affect another view is known as 'brushing', and is essential to work across multiple data representations. Other (interactive) data representations were suggested, such as a stacked bar chart showing the number of cells in each state, which would also update when selecting specific cells.

Finally, the visualization currently displays a summary of the successive values within each cell but leaves it to the user to find relationships between these values. One modeler suggested going further by displaying trends among the values as additional features: "the simulation will generate a temporal data streams (each cell will end up generating a stream of data points), so change analysis (shape, direction and velocity of changes) can be performed to understand how the whole system has impacted the individual cells". Since the colour of the segment already encodes information (i.e. the most frequent state) and all segments must have equal width (as they represent the same amount of time steps), the main possibility to encode additional information is to use the segment's length. This is illustrated in Fig. 8.

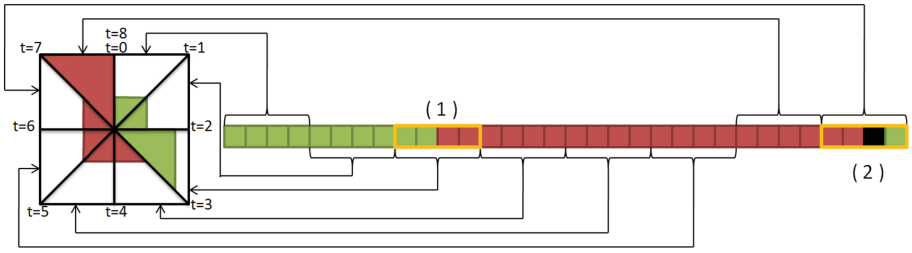

Fig. 8. Segments could have a different length to represent another feature of the data. In this simple example, the length encodes the number of different states present within each segment. All segments are low (since they only have one state) but segments (1) and (2), which encode 2 and 3 states respectively. This encoding helps finding where changes happen in the data.

5 Discussion

There is a growing interest in using visualizations at different stages of the modeling process, ranging from the early conceptual stage [21] to experimentation [6]

and the analysis of results. There has been a particular interest for visualizations for cellular automata [22], as it is a widely used modeling approach. In this paper, we focused on the analysis of data generated by a two-dimensional cellular automaton. We presented a visualization in which the successive time steps of the simulation are aggregated and displayed all at once. The resulting visualization allows to see key properties of the models, such as grains of sand falling in a sandpile, or an epidemic spreading. Nonetheless, the visualizations had a number of shortcomings, mostly stemming from the aggregation method and/or the number of segments used within each cell. Two approaches should be explored in future work.

First, we could introduce a customizable weighting, allowing to under- or over-weight certain states for display. For example, consider an epidemic in which individuals start as healthy, get infected, and either recover by being healthy again or die. This scenario has three states (healthy, infected, dead) but they may not be equally important. Indeed, if we are concerned with the spread of the disease, we may want to underweight healthy individuals, give a neutral weight to infected individuals, and over-weight dead individuals. Similarly, in a forest fire, we may be less interested in seeing empty spaces than we are in seeing burning trees. In addition to allowing users to customize the weighting, an interesting research avenue would be to automatically set the weights based on the dynamics of the data. The simplest way would be to perform the equivalent of a histogram normalization, where very frequent states are under-weighted while rare states are over-weighted. However, finding a weighting scheme that best helps modelers understand the dynamics would require performing change analysis on the data as well as structural analysis on the flow diagram.

Second, we could create a large databank of visualizations in which each dataset is visualized using different aggregation methods and number of segments. Then, modelers would assign a score to each visualization based on how informative they find it for a given task. Tasks would be chosen by their relevance to modeling, and by their heterogeneity in terms of the perceptual notions involved. Example of tasks could include identifying cells whose final state is the initial one, localizing a spread, finding clusters of cells in the same state, etc. For example, consider the epidemics described in Sect. 3.1: that same dataset may be rendered with 2 different aggregation methods and 3 different number of segments. For each of the $3 \times 2 = 6$ visualizations, modelers would assign a score from 0 (least useful) to 5 (most useful) expressing the usefulness of the visualization for localizing disease spread. This would generate a relational database consisting of properties of the dataset (e.g. number of states), aggregation method, number of segments, and mean score per specific task. To understand how visualization parameters (i.e. aggregation method and number of segments) affect task performance for a given dataset, we could then mine the relational database by building classifiers [23–25]. We acknowledge that assembling a dataset where modelers judge a large number of visualizations is labour intensive. Nonetheless, having the target audience evaluate the visualizations for a set of task is a routinely performed procedure.

References

1. Aigner, W., Miksch, S., Muller, W., Schumann, H., Tominski, C.: Visualizing time-oriented data - a systematic view. Comput. Graph. **31**, 401–409 (2007)
2. Keim, D.: Designing pixel-oriented visualization techniques: theory and applications. IEEE Trans. Vis. Comput. Graph. **6**(1), 59–78 (2000)
3. Ward, M.: Multivariate data glyphs: principles and practice. In: Handbook of Data Visualization, pp. 179–198 (2008)
4. Fuchs, J., Fischer, F., Mansmann, F., Bertini, E., Isenberg, P.: Evaluation of alternative glyph designs for time series data in a small multiple setting. In: Proceedings of CHI, pp. 3237–3246 (2013)
5. Konyha, Z., Matkovic, K., Hauser, H.: Interactive visual analysis in engineering: a survey. In: Proceedings of the Spring Conference on Computer Graphics, pp. 31–38 (2009)
6. Mokhtari, M., Boivin, E., Laurendeau, D.: Making sense of large datasets in the context of complex situation understanding. In: Shumaker, R. (ed.) VAMR 2013, Part II. LNCS, vol. 8022, pp. 251–260. Springer, Heidelberg (2013)
7. Mago, V.K., Bakker, L., Papageorgiou, E.I., Alimadad, A., Borwein, P., Dabbaghian, V.: Fuzzy cognitive maps and cellular automata: an evolutionary approach for social systems modelling. Appl. Soft Comput. **12**(12), 3771–3784 (2012)
8. Pratt, S., Giabbanelli, P., Jackson, P., Mago, V.: Rebel with many causes: a computational model of insurgency. In: 2012 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 90–95 (2012)
9. Rana, E., Giabbanelli, P.J., Balabhadrapathruni, N.H., Li, X., Mago, V.K.: Exploring the relationship between adherence to treatment and viral load through a new discrete simulation model of HIV infectivity. In: Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, SIGSIM PADS 2015, pp. 145–156. ACM (2015)
10. Hethcote, H.W.: The mathematics of infectious diseases. SIAM Rev. **42**(4), 599–653 (2000)
11. White, S.H., del Rey, A.M., Sanchez, G.R.: Modeling epidemics using cellular automata. Appl. Math. Comput. **186**(1), 193–202 (2007)
12. Fuentes, M., Kuperman, M.: Cellular automata and epidemiological models with spatial dependence. Physica A Stat. Mech. Appl. **267**(3–4), 471–486 (1999)
13. Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality: an explanation of 1/f noise. Phys. Rev. Lett. **59**(4), 381–384 (1987)
14. Athanassopoulos, S., Kaklamanis, C., Kalfoutzos, G., Papaioannou, E.: Cellular automata: simulations using matlab. In: Proceedings of the Sixth International Conference on Digital Society (ICDS), pp. 63–68 (2012)
15. Rothermel, R.C.: A mathematical model for predicting fire spread in wildland fuels. In: Research Paper INT-115, U.S. Department of Agriculture, Intermountain Forest and Range Experiment Station (1972)
16. Alexandridis, A., Vakalis, D., Siettos, C., Bafas, G.: A cellular automata model for forest fire spread prediction: the case of the wildfire that swept through spetses island in 1990. Appl. Math. Comput. **204**(1), 191–201 (2008)
17. Karafyllidis, I., Thanailakis, A.: A model for predicting forest fire spreading using cellular automata. Ecol. Model. **99**, 87–97 (1997)
18. Encinas, L.H., White, S.H., del Rey, A.M., Sanchez, G.R.: Modelling forest fire spread using hexagonal cellular automata. Appl. Math. Model. **31**(6), 1213–1227 (2007)

19. Trunfio, G.A.: Predicting wildfire spreading through a hexagonal cellular automata model. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 385–394. Springer, Heidelberg (2004)
20. Song, H., Lee, B., Kim, B., Seo, J.: Diffmatrix: matrix-based interactive visualization for comparing temporal trends. In: Meyer, M., Weinkauf, T. (eds.) Proceedings of the 2012 Eurographics Conference on Visualization (EuroVis) (2012)
21. Giabbanelli, P.J., Jackson, P.J.: Using visual analytics to support the integration of expert knowledge in the design of medical models and simulations. Procedia Comput. Sci. **51**, 755–764 (2015). Proceedings of the 2015 International Conference on Computational Science (ICCS)
22. Krasnikov, G.Y., Matyushkin, I.V., Korobov, S.V.: Visualization of cellular automata in nanotechnology. Model. Artif. Intell. **3**(3), 98–120 (2014)
23. Crutzen, R., Giabbanelli, P.: Using classifiers to identify binge drinkers based on drinking motives. Subst. Use Misuse **49**(1–2), 110–115 (2014)
24. Crutzen, R., Giabbanelli, P., Jander, A., Mercken, L., de Vries, H.: Identifying binge drinkers based on parenting dimensions and alcohol-specific parenting practices: building classifiers on adolescent-parent paired data. BMC Public Health **15**(1), 1 (2015)
25. Giabbanelli, P., Adams, J.: Identifying small groups of foods that can predict achievement of key dietary recommendations: data mining of the UK national diet and nutrition survey, 2008–2012. Public Health Nutrition (2016)