TraCCA - A Complex Cellular Automata based Particle Tracking Framework

Davide Spataro, Paola Arcuri, Alessio De Rango,
William Spataro and Donato D'Ambrosio
Department of Computer Science
University of Calabria
Rende, Italy
Email: spataro@mat.unial.it

Alice Mari
Department of
University of Edinburgh
Edinburgh, Scotland (UK)
Email: mailalice

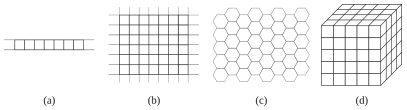

Fig. 1: Example of cellular spaces: (a) one-dimensional, (b) two-dimensional with square cells, (c) two-dimensional with hexagonal cells, (d) three-dimensional with cubic cells.

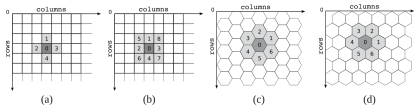

Fig. 2: Examples of von Neumann (a) and Moore (b) neighborhoods for two-dimensional CA with square cells. Examples of (Moore) neighborhoods are also shows for hexagonal CA, both for the cases of horizontal (c) and vertical (d) cells orientation. Central cell is represented in dark gray, while adjacent cells are in light gray. A numerical identifier can optionally be assigned to a cell within a neighborhood. Note that the central cell can optionally belong to its own neighborhood.

Keywords—IEEEtran, journal, LaTeX, paper, template.

I. Introduction

1. problem statement 2. lavori esistenti 3. contributo di questo lavoro

II. Complex Cellular Automata

A. Cellular Automata

Cellular Automata are parallel computing models, whose evolution is defined by local rules. A cellular automaton can be thought as a d-dimensional space, called *cellular space*, subdivided in regular cells of uniform shape and size. Each cell embeds a *finite automaton*, one of the most simple and well known computational models, which can assume a finite number of states. At time $t = 0$, cells are in arbitrary states and the CA evolves step by step by changing the states of the cells at discrete time steps, by applying the same local rule of evolution, i.e. the cell's *transition function*, simultaneously (i.e. in parallel) to each cell of the CA. Input for the cell is given by the states of a predefined (usually small) set of neighboring cells, which is assumed invariant in space and time (cf. Figures 2 and 3 for examples of 2D and 3D neighborhoods, respectively).

Despite their simple definition, CA can exhibit very interesting complex global behaviors. Moreover, from a computational point of view, they are equivalent to Turing Machines. This means, in principle, that everything that can be computed can also be by means of a cellular automaton (Church Turing thesis). Thanks to their *computational universality*, CA gained a great consideration among the Scientific Community and were employed for solving a great variety of different complex problems. Cellular Automata are formally defined as a quadruple:

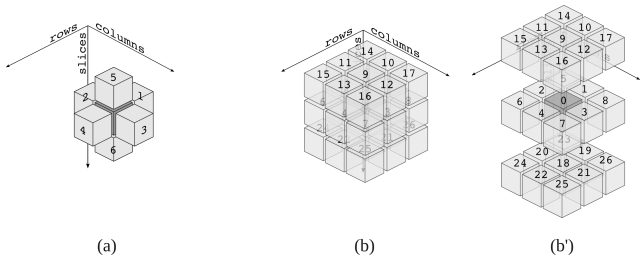

Fig. 3: Examples of von Neumann (a) and Moore (b, b') neighborhoods for three-dimensional CA with cubic cells. Central cell is represented in dark gray, while adjacent cells are in light gray. As for the case of two-dimensional neighborhoods, a numerical identifier can optionally be assigned to a cell within a three-dimensional neighborhood.

$$A =< R, X, Q, \sigma >$$

where:

- $R = \{i = (i_0, i_1, ..., i_{d-1}) \mid i_k \in \mathbb{Z} \ \forall k = 0, 1, ..., d-1\}$ is the set of points, with integer coordinates, which defines the d-dimensional cellular space;
- $X = \{\xi_0, \xi_1 ...\xi_{m-1}\}$ is the finite set of m d-dimensional vectors

$$\xi_j = \{\xi_{j_0}, \xi_{j_1}, ...\xi_{j_{d-1}}\}$$

that define the set

$$N(X,i) = \{i+\xi_0, i+\xi_1, ..., i+\xi_{m-1}\}$$

of coordinates of cells belonging the the central cell's neighborhood. In other words, X represents the geometrical pattern that specifies the neighborhood relationship;

- Q is the finite set of states of the cell;
- $\sigma : Q^m \rightarrow Q$ is the cell's transition function. Note that, a cell i is said to be in a *quiescent state*, $q_0 \in Q$, if all its neighbouring cell are in the same state and $\sigma(q_0, q_0, \ldots, q_0) = q_0$.

Let $C = \{c|c : R \rightarrow Q\}$ be the set of global configurations of the CA, and $c(i)$ the state of the cell i in the configuration c. The CA *global trnasition function*, which applies the cell transition function to all the cells in R, is defined as:

$$\begin{aligned} \tau : C &\longrightarrow C \\ c &\mapsto \tau(c) \end{aligned}$$

where

$$\tau(c(i)) = \sigma(c(N(X,i))) = \sigma(c(i+\xi_0), c(i+\xi_1), \ldots, c(i+\xi_{m-1}))$$

and the global evolution of the CA is obtained by applying the global transition function τ step by step.

B. Extended Cellular Automata

Extended Cellular Automata [**?**] represents an extension of the original CA computational paradighm. XCA were firstly applied to the simulation of basaltic lava flows in the 80's [**?**] and many subsequent examples of application shown that the approach behind XCA can greatly make more straightforward the modeling of some complex systems.

Informally, XCA, compared to classical CA, are different because of the following reasons:

- The cell's state is decomposed in *substates*, each of them representing the set of admissible values of a given characteristic assumed to be relevant for the modeled system and its evolution (e.g., lava temperature, lava thickness, etc, in the case of a lava flow model). The set of states for the cell is simply obtained as the Cartesian product of the considered substates.
- As the cell's state can be decomposed in substates, also the transition function can be split into *elementary processes*, each of them representing a particular aspect that rules the dynamic of the considered phenomenon. In turn, *elementary processes* can be split into *local interactions*, which refer to rules that deal with interactions among substates of the cell with neighbor ones (e.g., mass exchange with neighbors) and *internal transformations*, defined as the changes in the values of the substates due only to interactions among substates inside the cell (e.g. the solidification of the lava inside the cell due to the temperature drop).
- A set of *parameters*, commonly used to characterize the dynamic behaviors of the considered phenomenon, can be defined.
- Global operations can also be allowed (e.g. to model external influences that can not easily be described in terms of local interactions, or to perform reductions over the whole, or a subset of, the cellular space). They are often referred as *steering* operations.

Formally, a XCA is defined as a 7-tuple:

$$A =< R, X, Q, P, \sigma, \Gamma, \gamma >$$

where:

- R is the d-dimensional cellular space.
- $\Gamma \subseteq R$ is the region over which steering is applied.
- X is the geometrical pattern that specifies the neighborhood relationship; $m = |X|$ represent the number of elements in the set X, i.e. the number of neighbors for the central cell.
- $Q = Q_0 \times Q_1 \times .... \times Q_{n-1}$ is the set of cell's states, expressed as Cartesian product of the n considered *substates* $Q_0 \times Q_1 \times .... \times Q_{n-1}$.
- $P = p_0, p_1, ..., p_{p-1}$ is the set of CA *parameters*.They can allow a fine tuning of the XCA model, with the purpose of reproducing different dynamical behaviors of the phenomenon of interest.
- $\psi : Q^{|R|} \rightarrow Q^{|R|}$ is the global function which define the initial conditions of the system.
- $\sigma : Q^m \rightarrow Q$ is the cell's transition function. It is split in s *elementary processes*, $\sigma_0, \sigma_1, ..., \sigma_{s-1}$, each one describing a particular aspect ruling the dynamic of the considered system.
- $\gamma : Q^{|\Gamma|} \rightarrow Q^{|\Gamma|} \times \mathbb{R}$ is the (global) steering function.

The initial conditions of the system are obtained by preliminarly applying the ψ initialization function. Then, if τ represents the function which applyes the transition function σ to all the cells in R, as for the CA case, and γ the XCA steering function, the XCA evolution is obatained by applying the $\Phi = \tau \circ \gamma$ XCA global transition function at discrete time steps.

III. Tracking Framework

A. Preprocessing

B. Segmentation

C. Tracking

IV. Bacteria Tracking

V. Conclusion

Appendix A
Images Filters

A. contrast stretch

Acknowledgment

The authors would like to thank...

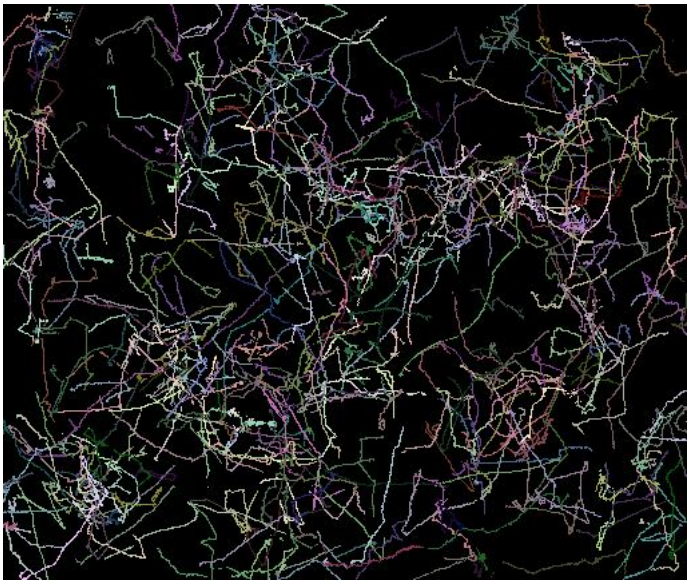

Fig. 4: Trajectories

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.