



Proceedings of the Second International Workshop on Sustainable
Ultrascale Computing Systems (NESUS 2015)
Krakow, Poland

Jesus Carretero, Javier Garcia Blas
Roman Wyrzykowski, Emmanuel Jeannot.
(Editors)

September 10-11, 2015

Labeling connected componets in binary images based on cellular automata

BILJANA STAMATOVIC

University of Donja Gorica, Montenegro
biljana.stamatovic@udg.edu.me

Abstract

This short paper introduce an algorithm for labeling connected components in n -dimensional binary images based on cellular automata, $n \geq 2$. Here is presented tree-dimensional binary images algorithm. The algorithm code was implemented in NetLogo programming environment. The algorithm is local and can be efficiently implemented on data-flow parallel platforms with an asymptotic complexity of $O(L)$ on an $L \times L \times L$ bynary image.

Keywords labeling connected components, cellular automata, data-flow

I. INTRODUCTION

Labeling of connected components has been intensively studied. This is one of the most fundamental operations in pattern analysis, pattern recognition, computer (robot) vision, and machine intelligence. Many algorithms have been proposed for addressing this issue [1, 2, 3, 4, 5, 6, 7, 8]. The labeling algorithms transform a binary image in an image in which groups of connected cells (pixels) are grouped in disjoint components with unique labels. So, the analysis of the image can be performed on a higher level than the pixel level.

The cellular automata (CA) can be considered as an alternative way of computation based on local data flow principles. Also, the CA can be seen as a computing machine in the sense that it is able to transform an input configuration, embedded in its initial state, to an output configuration. Here, we suppose that initial data, pixels of image, are loaded into CA cells. A CA can be informally represented as a set of regularly and locally connected identical elements. The elements can be only in a finite set of states. The CA evolves in discrete time steps, changing the states of elements according to a local rule, which is the same for all elements. The new state of each element depends on its previous state and on the state of its neighbourhood. The characteristic properties of CAs are therefore locality, discreteness

and synchrony. More definitions about connectivity and CAs can be found in [9], [10].

II. DEFINITIONS AND ALGORITHM

We consider a CA as a 3D lattice network of unite cubes (cells) whose centres are in integer lattice. For simplicity, we suppose that the lattice has $N = L \times L \times L$ cells $c_{i,j,k}$ with positions determined by indices $i, j, k = 1, \dots, L$ in x, y and z directions, respectively, with $L \geq 3$. Each cell can exist in a finite number of states marked by colors. The cells can change their states at the end of time-steps that are discrete moments in time after the computing of time-steps is completed. The state of the cell $c_{i,j,k}$ in a time-step t is denoted by $c_{i,j,k}(t)$ and the state of all lattice cells by $C_t, t \geq 0$.

The initial configuration C_0 is represented by a 3D grid of cube cells and each cell can exist in two different states denoted by white or black colors. Boundaries of the grid are black. All the remaining cells of the grid, are coloured (labeled) white with the probability p and black with probability $1 - p$. The probabilities are independent for each cell. Example of an initial configuration is in Figure 1.

Two cells $c_{i,j,k}$ and $c_{l,m,n}$ are 6-neighbours if $|i - l| + |j - m| + |k - n| = 1$. An ordered sequence of cells c_1, \dots, c_n is called 6-path if each cell c_i is a 6-neighbour

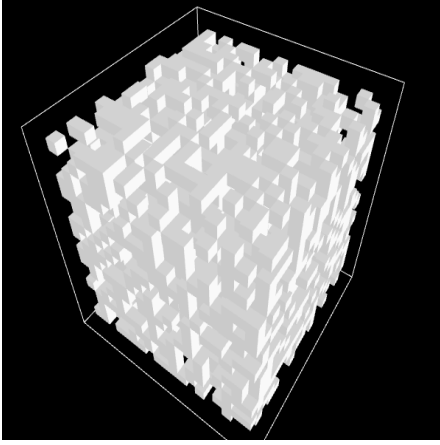


Figure 1: Initial grid configuration generated with $p = 0.4$.

of the next cell c_{i+1} in the sequence, $i = 1, \dots, n - 1$, $n \geq 2$. Set of cells is 6-connected, if for any two cells there exists a 6-path of cells, within the set, between them. A maximum 6-connected set of white cells is called a *component*. In [7] can be find definitions for 18 and 26 connectivity in 3D. For short notation, the relative positions of the cells $c_{i+1,j,k}$, $c_{i-1,j,k}$, $c_{i,j+1,k}$, $c_{i,j-1,k}$, $c_{i,j,k+1}$, $c_{i,j,k-1}$ to the cell $c_{i,j,k}$ are denoted by E, W, N, S, U, D, respectively.

The goal of our algorithm is to assign a unique label to each of the components. The algorithm is similar for 18 and 26 connectivities, for 2D labeling connected components (4 and 8 - connectivity) and also can be generalized for n-dimension, $n > 3$.

III. ALGORITHM AND EXPERIMENTAL RESULTS

Let $i, j, k \in \{0, 1, 2, \dots, L - 1\}$ and $t \in \mathbb{N}$, $t \geq 0$. Denote by $c_{(i,j,k)}(t)$ the state of a cell (i, j, k) in time step t and by $c(t)$ an argument of a local transition function, which is an ordered collections of 6-neighbours cell's states in time step t .

We use the terminology of colors. A cell is in state "m" if it is coloured by color m . 0-color is white, 1-color is black, m -color is a color with code m , $m > 1$, e.g. in RGB implementation.

Let C_0 be an initial configuration. We will define two CAs A_i by their local transition functions φ_{A_i} , $i = 1, 2$.

Step 1: Each component has the lowest down, right and south cell. The CA A_1 makes unique labeling, by changing states of white cells whose E, S, D 6-neighbours are black. The unique colors can be determined from the cells' positions and must be different from white and black. The color we denoted by $col(i, j, k)$ for the cell (i, j, k) . On this way all the lowest down, right and south cells in all components are labeled by unique colors. Also, some other cells change their states. The local transition function is defined by $\varphi_{A_1}(c_{i,j,k}(0)) = c_{i,j,k}(1)$, where

$$c_{i,j,k}(1) = \begin{cases} col(i, j, k), & c_{i,j,k}(0) = 0 \wedge c_{i+1,j,k}(0) = \\ & = c_{i,j-1,k}(0) = c_{i,j,k-1}(0) = 1 \\ c_{i,j,k}(0), & otherwise \end{cases}$$

For the labeling the CA A_1 makes one step. On figure 2 is the configuration on the end of the step for the initial configuration from figure 1.

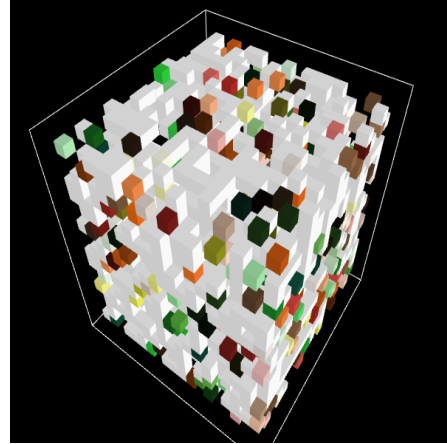


Figure 2: Grid configuration from Figure 1 after first step of the algorithm.

Step 2: Local transition function of the CA A_2 is defined by $\varphi_{A_2}(c_{i,j,k}(t)) = c_{i,j,k}(t + 1)$, $t \geq 0$, where

$$c_{i,j,k}(t + 1) = \begin{cases} \max(c(t)), & c_{i,j,k}(t) \neq 1 \\ c_{i,j,k}(t), & otherwise \end{cases}$$

CA A_2 will color every white component with different color (not black or white) using iteration, until the state of the lattice becomes constant.

Using the described CAs we have algorithm 1 for the labeling connected components.

Data: Initial configuration C_0

Result: Connected components labeling for C_0

φ_{A_1} ;

while exist cell with change **do**

 | φ_{A_2} ;

end

Algorithm 1: Algorithm for labeling 6-connected components

On Figure 3 is the configuration on the end of the algorithm for the initial configuration from Figure 1.

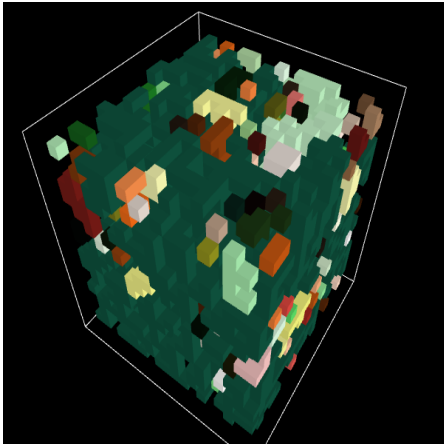


Figure 3: Grid configuration from Figure 1 on the end of the algorithm.

The implementation of the algorithm is made in NetLogo 5.0.4. The algorithm was extensively evaluated on various test cases for different size of grids and probabilities, i.e. densities of white cells in initial configuration. We calculated the average time steps required for 100 initial configurations of $31 \times 31 \times 31$ grid for the different probabilities (Table 1). However, the current implementation of the algorithm with NetLogo is limited with grid sizes and no possibility to use variables in RGB notation.

Probability	Mean value
0.2	18.11
0.25	30.25
0.3	69.92
0.35	131.44
0.4	87.13
0.45	75.55
0.5	68.82
0.55	68.43
0.6	68.16
0.65	67.35

Table 1: Average number of time steps on the $31 \times 31 \times 31$ grid in 100 iterations with different probability of initial cell coloring

IV. CONCLUSION AND FUTURE INVESTIGATIONS

The proposed CA algorithm has several advantages, e.g. it is not limited by the number of cells, its evolution is inherently parallel, and it has a strong resemblance to the important approaches in the nature like principles of cells or elementary particles. Complexity of the algorithm depends on the component shapes but an asymptotic complexity is $O(L)$ on an $L \times L \times L$ binary image. The algorithm use iterative function and can resolve the problem of stack overflow that could appear in recursive labeling which is studied in [11]. Drawbacks of the algorithm are in using global variables for stopping CA's work. Lack of global communication, implies problems related to global synchronization, data manipulation and inability for calculation of complex mathematical operations, however, these difficulties can be resolved by dedicated hardware resources.

The heterogeneous computing, supported today with data-flow approaches, FPGAs, SoCs, GPUs and manycore systems, are promising platforms for the implementation of the efficient CA based algorithms.

REFERENCES

- [1] Lifeng He, Yuyan Chao and Kenji Suzuki, "A New Two-Scan Algorithm for Labeling Connected Components in Binary Images," in *Proceedings of the World Congress on Engineering*, London, U.K., Vol II, July 2012, p. 1141-1146.
- [2] Lifeng He, Yuyan Chao, Kenji Suzuki and Kesheng Wu, "Fast connected-component labeling," *Pattern Recognition*, vol. 42, 2009, 1977 - 1987.
- [3] T. Y. Kong and A. Rosenfeld,, *Topological algorithms for digital image processing*, Elsevier Science, Amsterdam, Netherlands, 1996.
- [4] R.E. Cypher, L. Snyder and J. L. C. Sanz, "Practical Algorithms for Image Component Labeling on SIMD Mesh Connected Computers," *IEEE Transactions on Computers*, vol. 39, 1990, 276 - 281.
- [5] Fei Zhao, Huan zhang Lu and Zhi yong Zhang, "Real-time single-pass connected components analysis algorithm," *EURASIP Journal on Image and Video Processing*, vol 2013, 2013.
- [6] Kenji Suzuki, Isao Horiba, and Noboru Sugie, "Linear-time connected-component labeling based on sequential local operations," *Computer Vision and Image Understanding*, Vol 89, 2003, p. 1-23.
- [7] Qingmao Hu, Guoyu Qian and Wieslaw L. Nowinski, "Fast connected-component labelling in three-dimensional binary images based on iterative recursion," *Computer Vision and Image Understanding*, Vol 99, 2005, p. 414-434.
- [8] Christopher T. Johnston and Donald G. Bailey, "FPGA implementation of a Single Pass Connected Components Algorithm," in *4th IEEE International Symposium on Electronic Design, Test and Applications*, DELTA 2008, Hong Kong, January 23-25, 2008, p. 228-231
- [9] Biljana Stamatovic, Gregor Kosec, Roman Trobec, Xiao Xuan and Sinisa Stamatovic, "Cellular Automata Supporting n-Connectivity," *Mathematical Problems in Engineering*, Vol 2014, 2014.
- [10] Biljana Stamatovic and Roman Trobec, "Data parallel algorithm in finding 2-D site percolation backbones," in *Proceedings of the First International Workshop on Sustainable Ultrascale Computing Systems (NESUS 2014)*, Porto, Portugal, 2014, p. 65-70.
- [11] G. Borgefors, I. Nystrom and G.S.D. Baja, "Connected components in 3D neighbourhoods," in *Proceedings of the 10th Scandinavian Conference on Image Analysis*, 1997, p. 567-572.