Generating Transition Rules of Cellular Automata for Urban Growth Prediction

N. Laila Ab Ghani[1], Siti Z.Z. Abidin[1] and M. Zamani Z. Abiden[2]
[1]Faculty of Computer and Mathematical Sciences,
[2]Faculty of Architecture, Planning and Surveying,
University Technology MARA,
40450, Shah Alam, Selangor
MALAYSIA
lailaghani@tmsk.uitm.edu.my, sitizaleha533@salam.uitm.edu.my, mzabiden009@gmail.com

***Abstract*—** Urban growth prediction can be simulated using digital maps. The growth of a non-built area can be detected through the change of pixels in a temporal imagery data. A built area usually affects the growth of its surrounding area as similar to Cellular Automata theory. Cellular Automata (CA) is a system consists of grid cells where each one is in finite number of states. The basic components of CA are cells, states neighborhood and transition rules. This research is mainly about obtaining a set of transition rules that detect the pattern of urban growth based on digital maps. The datasets are in the form of satellite images of the study area, the district of Subang Jaya, one of the most rapid urban growth areas. It is difficult to specify equation-based transition rules due to complex geographical processes in the urban growth. Most of the available transition rules are defined statically. This research proposes a different approach using deterministic and pixel-based method by experimentally identifying the unique pattern of surrounding cells on every pixel in the map. Then, the unique patterns are used to generate the transition rules. The rules are implemented as a prototype engine and the accuracy of the rules are tested by comparing predicted results with the original satellite images. Due to the rapid urbanization process in Malaysia, it is important to have a system that has the ability to predict the future growth of an urban area. Excellent accuracy will lead to better monitoring system to cater future livings.

***Keywords*—** Urban growth, Urban pattern, Prediction, Cellular automata, Multitemporal satellite images, Remote sensing, Transition rule.

I. Introduction

URBAN growth prediction can be simulated using digital maps. It is possible to model the urban growth by using a deterministic approach that considers the change of pixels in a temporal imagery data. A built area can be detected based on the change of its surrounding area as similar to Cellular Automata theory.

Cellular Automata (CA) is a system consists of grid of cells where each one is in finite number of states. The basic components of CA consist of cells, states, neighborhood and transition rules. The primary issue of CA is to define the transition rules, which can be represented in many forms [1]. Recently, CA has been applied to numerous applications especially in the simulation of urban growth. Studies have shown that CA is suitable for simulating complex geographical process [1].

This research focuses to obtain a set of transition rules that detect the pattern of urban growth based on digital maps. The datasets are in the form of satellite images of the study area, the district of Subang Jaya, one of the most rapid urban growth areas in Malaysia (a developing country in South East of Asia). It is difficult to specify equation-based transition rules due to complex geographical processes in the urban growth. Most of the available transition rules are defined statically. This research has different approach using deterministic and pixel-based method by experimentally identifying the unique pattern of surrounding cells on every pixel in the map.

There are two digital maps required to classify the pixel location with change or unchanged value. For each pixel on a map, there are eight pixels surrounding it. The pattern of the surrounding cells is determined according to the location in relative to the center pixel. Since the value of a pixel is either 0 or 1, the pattern consists of 8 binary digits. It is possible that there are many occurrences of similar 8 binary digits. While reading all the pixel values from the map, these 8 binary digits are stored in a file. Working with 8 surrounding pixels is found to be not achieving the expected satisfactory results, therefore, the number of surrounding pixels are extended to 24 (2 pixels distance away from the center cell).

With this pixel-based approach, it is possible to detect and predict any location as long as the correct digital map format is provided. Even though the details on the real geographical factors are not known in particular, it is still possible to investigate the pattern of growth for urban areas. In fact, it is the common problem for most of the developing countries. With the limited information gathered, the detection and prediction of urban growth is still possible. When the focus is made on the change or growth of an area, the method is similar to the transition rules in a CA. The rules are implemented as a prototype engine and the accuracy of the rules are tested by comparing predicted results with the original satellite images.

For the case study, the comparison of results is performed on three digital images of three different years. The results are very promising with the accuracy values of 66%, 91% and 88%. The key contributions of this research are as follows:

- the new approach (deterministic) in detecting and predicting urban growth based on pixel,
- the new mathematical approach in determining transition rules,
- the use a prototype engine that can accept digital maps of any location on earth for monitoring urban growth.

This paper is organized as follows: Section 2 discusses on the related work of urban sprawl and cellular automata. Section 3 presents the design and implementation of the transition rule. Section 4 illustrates the results, before the concluding remarks in Section 5.

II. RELATED WORK

A. Urban Sprawl

Historically, Malaysia has experienced a rapid urbanization growth due to the industrialization and related residential growth [2]. As population increase in an area of city, the boundary of the city expands to accommodate the growth and becoming sprawl.

Sprawl can be defined as the increase in built-up and paved area with impact on loss of agricultural land, open space and ecologically sensitive habitats [3]. The main cause of sprawl includes population growth, economy and the pattern and provision of infrastructure [4]. In [5], sprawl development can occur in three ways which are low-density (radial) sprawl, ribbon sprawl and leapfrog sprawl.

Radial sprawl is the consumptive use of land for urban purposes along the margins of existing metropolitan areas. It is highly affected by provision of public infrastructures such as water, sewer, power and roads. Unlike radial sprawl, a ribbon sprawl is development along the major transportation network that connects urban areas. Lands alongside the network are developed but those without direct access to the network remains rural or undeveloped. Differently, a leapfrog sprawl is a discontinuous pattern of urbanization in which the developed lands are widely separated from each other. This is the most costly form of development with respect to providing urban services such as water and sewage.

The inherent causal factor and dynamics involved in the rapid changes of land-use due to urban sprawl is considered as a fit case to apply Cellular Automata models for simulating future scenarios [4].

B. Cellular Automata

Cellular automata (CA) can imply that population density of an area is related to the population density at the adjacent areas, but close areas have s stronger influence than more remote areas. The Cellular Automata (CA) model was introduced by Ulam in 1940s and soon used by Von Neumann in 1950s. It is a system consists of grid of cells where each one is in finite number of states, updated in a discrete time steps according to a set of transition rules. A typical CA system comprises of four components which are cells, states, neighbourhood and transition rules. The state of a cell can change only based on transition rules, which are defined in terms of neighbourhood function [4].

According to Mavraudi [6], the principle can be stated in most general form as, "If something happens in the neighborhood of the cell, Then some other things happen to the cell". Typically in CA, the neighbourhood and transition rules are important for automaton to initiate state transitions in a cell [4]. The core of CA is how to define the transition rules, which can be represented in many forms [7]. One of the effectiveness of CA application is defining the CA transition rules in each specific case. A compositional approach to define CA transition rules resulted in more complex nature as compared to traditional approach [8].

In most CA models, the automata are influenced only within Moore's neighbourhood, which are the eight adjacent neighbourhood cells. Recently, CA has been applied to numerous application including the simulation of urban growth [3,4,6,9-12], land use change [4,13], population expansion [14] and forest fire spread prediction [15]. It can be concluded that CA is suitable for simulating complex geographical process [16]. CA has many advantages for geographic modelling including its capability to support a very large parameter spaces for simulation. For instant, a one-dimensional CA with a binary state set and 13 cells has 2^{13} possible combinations. A two-dimensional version of the same CA had 2^{169} possible combinations and so on [17]. It is possible to combine CA with artificial neural network technique [18] to predict urban growth.

III. DESIGN AND IMPLEMENTATION

In this section, the design and implementation of the proposed technique are discussed. The multitemporal satellite images are pre-processed using ENVI software in order to obtain a bitmap format. Fig. 1 below shows the major cities of Subang Jaya, an area of the case study.

Fig. 1. Overall view of the study area

Datasets available for this case study are for the year 1988, 1994, 2000, 2001 and 2003. It is in the form of images (Fig. 2) with the size of 491 pixels width and 238 pixels height. Each dataset consist of 116858 (491 x 238) data containing information of both built and non-built area.

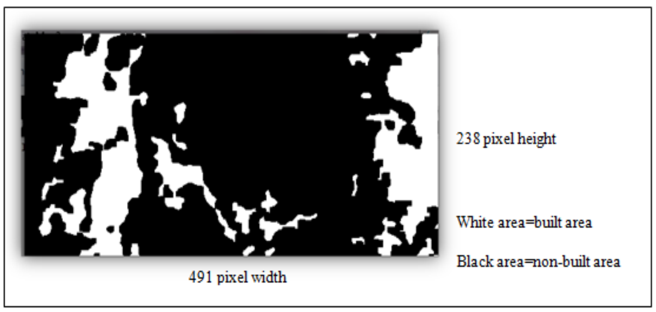


Fig. 2. Binary image of the study area

There are four main steps involved in obtaining the transition rules: image conversion, data filtration, data comparison, and rules conversion.

A. Image conversion

In this phase, the bitmap image is converted to binary format. Process includes substituting each pixel in image file format into binary (0 or 1). Substitution is done using the algorithm shown in Fig. 3.

```
if (black_pixel)
   current_pixel = 0; //non-built
else if (white_pixel)
   current_pixel = 1; //built
```

Fig.3. Values determination of a binary image

The resulting image file to binary format might best be described from Fig. 4 where the black areas are represented by value 0 and value 1 for white.

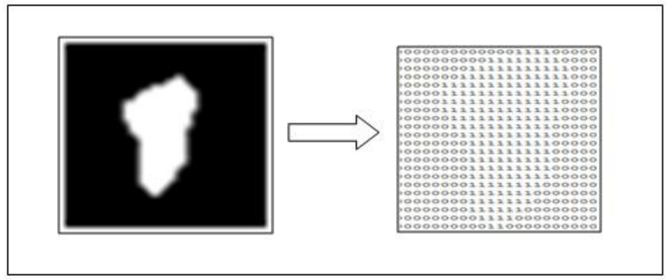


Fig. 4. Binary representation of a pre-processed image

B. Data filtration

Filtration includes comparing two consequence datasets by replacing the cell that have a condition from built to non-built (1 to 0). This condition is checked to avoid the wrong condition where the urban area is changed into rural. Replacement of the correct values is performed by using the following filtration algorithm (Fig. 6).

```
if (center_val_year88 = 1 and center_val_year94 = 0)
   center_val_year94 = 0;
```

Fig. 6 Algorithm for correcting the wrong condition

The example of such condition is illustrated in Fig. 7. With the surrounding cells of value 1 cannot produce the value 1 in the next transition.

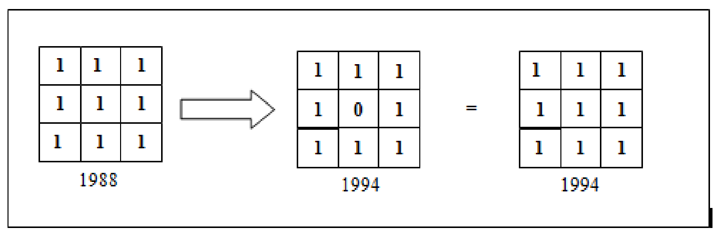
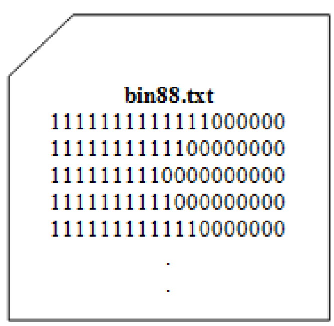


Fig. 7

After the conversion and filtration processes, the obtained data are stored to a text file for later processing. Fig. 8 below is the sample file of a map for the year 1988.



bin88.txt
111111111111000000
111111111100000000
111111110000000000
111111111000000000
111111111110000000
.
.

Fig. 8 Binary pattern of image as in textfile.

C. *Data comparison*

The purpose of this process is to get the unique pattern of states change from non-built to built state (0 to 1). It starts by comparing only two datasets to get the pattern of 8-neighborhood pixels. Other datasets will only be used for testing purpose. Initially, the comparison consists of two conditions; the state from non-built to built and state of non-built that remains unchanged. The algorithms are as depicted in Fig.9.

```
if (center_pixel88 = 0 and center_pixel94 = 1
    extract information of neighborhood pixel of
        center_pixel88

if (center_pixel88 = 0 and center_pixel94 = 0)
    extract information of neighborhood pixel of
        center_pixel88
```

Fig. 9. Pseudo codes for checking values for center pixel

Fig. 10 shows how the neighborhood information is captured and stored in a text files. The position of cells is determined by the order of their existence.

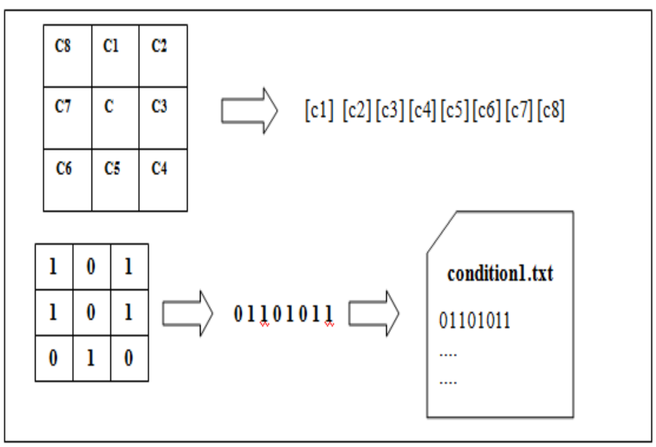


Fig. 10 Transforming values into files

Due to the intersection values between both conditions, an additional condition that consists of the same values between both conditions, is introduced. Table 1 below shows the sample data to represent the intersection values in both conditions. The values in red are the intersection values.

Table 1: Determining binary pattern of 8 cells

Changed(0 to 1)	**Intersection**	**Unchanged (0 to 0)**
01010101	**01101011**	00000000
01111011	**01001111**	**01101011**
01101011	**11001110**	**01001111**
11001010		00001010
01001111		**11001110**
11001110		

Next, the number of unique and intersection patterns are calculated in order to find the ratio of all the three conditions. Fig. 11 is the pie chart that represents the 81 percentage of the intersections. It can be concluded that for the first pixel distance (the value of distance is 1), only 17 percents unique answers can be produced, while 81 percents of the output values are still in an ambiguous state.

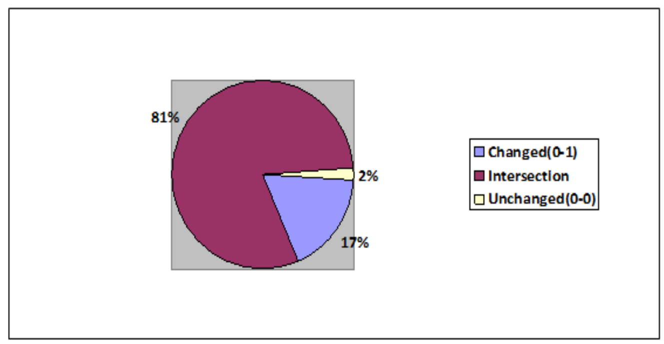


Fig. 11. Percentage of three different patterns (8 cells).

Since the percentage of intersection is the highest, the neighborhood pixel is extended to 24-neighborhood pixel to get the unique pattern for non-built state to built state for the case of intersection. Process will traverse each cell and check if its 8-neighborhood pixels are the same as the intersection values. Fig. 12 explains how the process are done:

```
if(c[1][c2][c3][c4][c5][c6][c7][c8] = intersection)
    extract information of 24-neighborhood pixel
```

Fig.12.Extend checking on 24 pixels

With extending the number of cells from 8 to 24 pixels, more accuracy is possible to be achieved. Fig. 13 below is the image representation to describe the process.

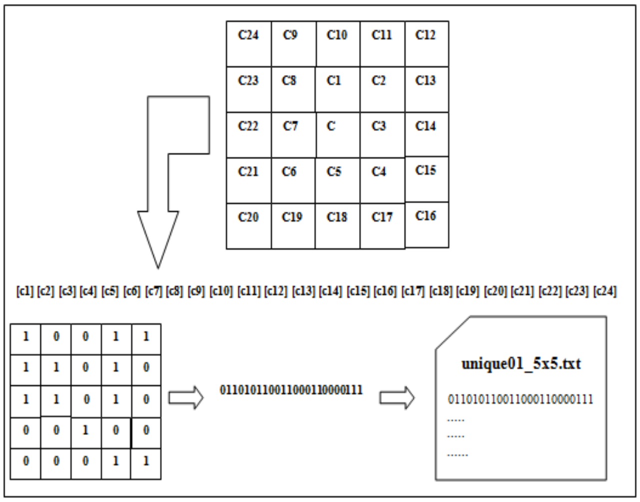


Fig. 13.Process of determining 24 surrounding cells

In this phase, it is still found that there are intersection values (ambiguous) between these two conditions (surrounding cells for 8 and 24 cells). Table 2 shows the sample data to represent the intersection values in both conditions. The values are presented in red color the intersection values.

Table 2: Determining binary pattern of 24 cells

Changed(0-1)	Intersection	Unchanged(0-0
010101011111111111111111	011010110101000000000000	000000000000000000000000
011110111010101010101010	010011111111110111110111	010011111111110111110111
011010110101000000000000	110011100100001011101110	110011100100001011101110
110010101101010001111111		000010100000101010000000
010011111111110111110111		011010110101000000000000
110011100100001011101110		

The summary of results is presented in a pie chart shown in Fig. 14. With the larger distance (based on pixel), the percentage of intersection (ambiguous values) has decrease to 54 percent.

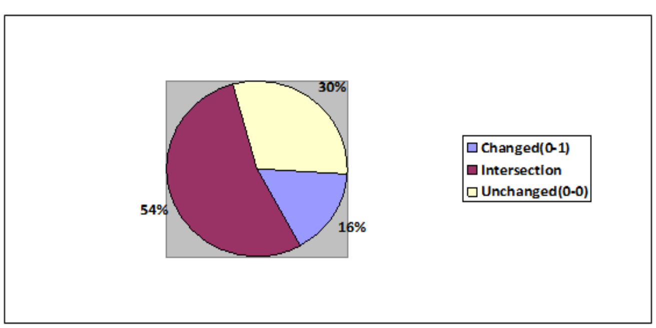


Fig. 14. Percentage of three different patterns (24 cells).

Figure 15 summarizes all the steps involved in this phase. For the case of 8-neighborhood pixel, the unique values that changed the condition from non-built to built is taken while for the case of ambiguity or intersection, the neighbourhood pixel is extended to 24-neighborhood pixel. In the end, the unique values are also extracted from the process.

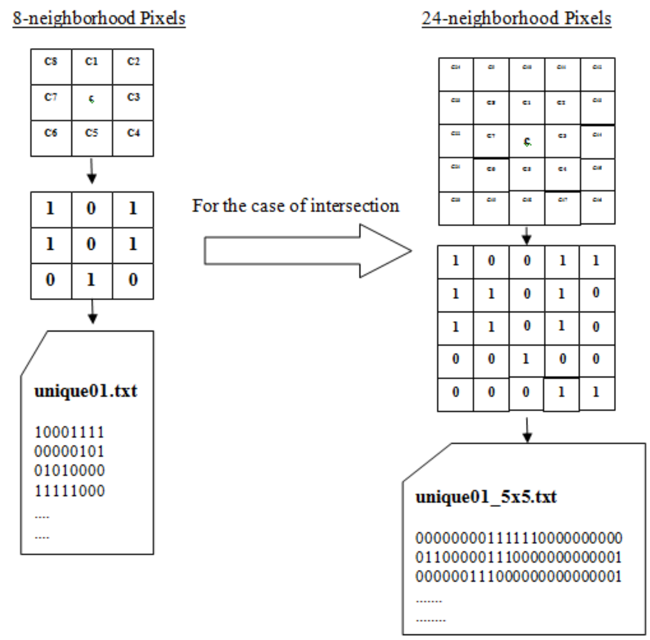


Fig. 15. Converting cells' values into text files

D. Rules conversion

Since the rules obtained are in the form of binary values, they are converted to decimal values and stored in another text file (Fig. 16).

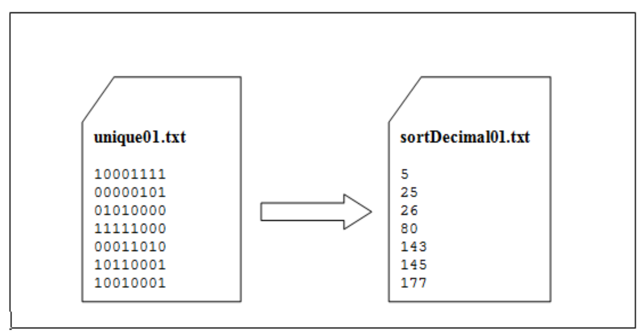


Fig. 16. Binary to decimal values

The decimal value presents the transition rules of Cellular Automata. The algorithm used to test the transition rule is illustrated in Fig. 17.

```
if (center_pixel = 0)
   if(neighborhood_pixel = sort_decimal)
      center_pixel = 1;
```

Fig.17. Change a center cell's value

The mathematical representation of the obtained transition rule is as follows:

$$C_{[1..8]}(t_n) = \begin{cases} V_1 \;:\; C_0(t_n) = 0 \;\rightarrow\; C_0(t_{n+1}) = 1 \\ V_2 \;:\; C_0(t_n) = 0 \;\rightarrow\; C_0(t_{n+1}) = 0 \\ V_3 \;:\; V_1 \cap V_2 \end{cases} \qquad (1)$$

For $V_{[4..6]} \dot{\subseteq} V_3$,

$$C_{[1..24]}(t_n) = \begin{cases} V_4 \;:\; C_0(t_n) = 0 \;\rightarrow\; C_0(t_{n+1}) = 1 \\ V_5 \;:\; C_0(t_n) = 0 \;\rightarrow\; C_0(t_{n+1}) = 0 \\ V_6 \;:\; V_4 \cap V_5 \end{cases} \qquad (2)$$

The testing of transition rule is the following:

$$C_0(t_{n+1}) \rightarrow 1, \begin{cases} iff\; C_0(t_n) = 0 \;\wedge\; C_{[1..8]}(t_n) \in V_1 \\ iff\; C_0(t_n) = 0 \;\wedge\; C_{[1..24]}(t_n) \in V_4 \end{cases} \qquad (3)$$

where C is the set of all possible cell states, V is the set of the neighbourhood cells, and t is the time period.

IV. RESULTS

A prototype engine is developed for the purpose of testing the functionality of the transition rule. A user interface is designed in order to display the output like a form. Fig. 18 is the example of interface for the engine.

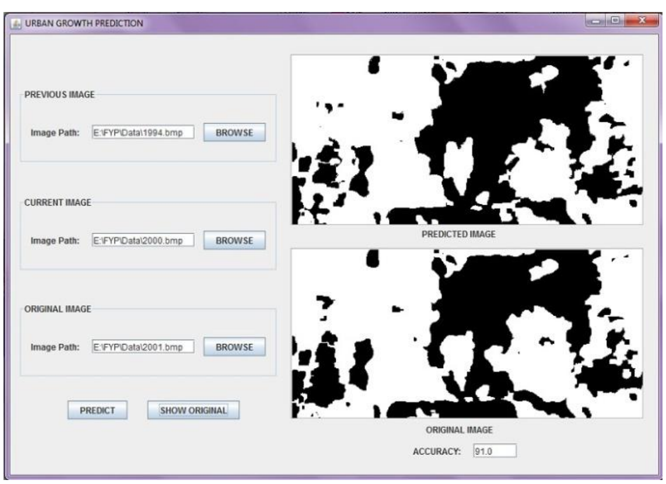


Fig. 18. A prototype engine's interface

The test datasets are binary images for the year 1988, 1994, 2000, 2001 and 2003. The image map of year 1994 is used as the input for the engine. The output or predicted image represents the prediction results for year 2000 and the accuracy of the predicted image compared to the original image as illustrated in Fig.19.

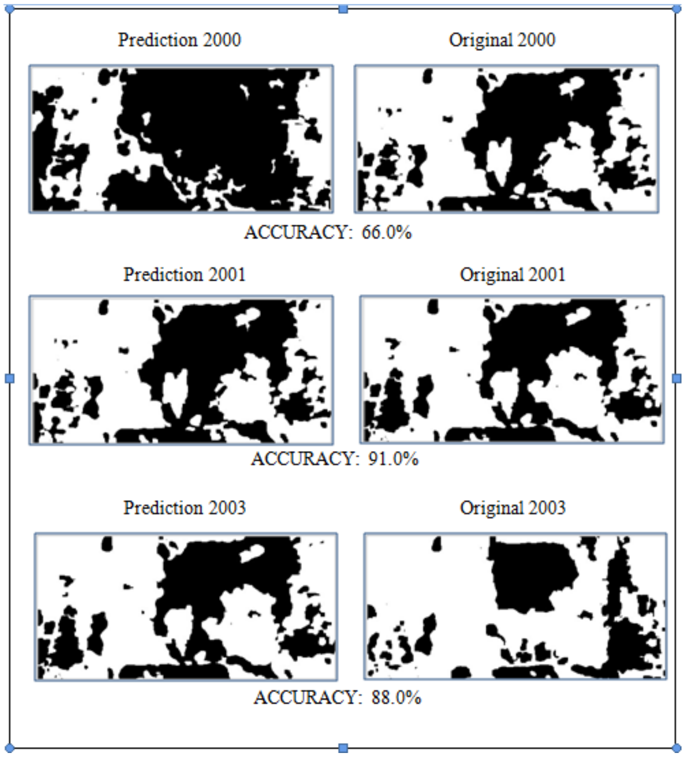


Fig. 19. Accuracy tests

Fig. 19 show the prediction and original image of the year 2000, 2001 and 2003. The accuracy of results for images of the year 2000 is 66.0 percent. The value is determined by comparing the change of pixels for predicted and original image. For the result of year 2001, the accuracy is 91.0 percent while for the image of year 2003, the accuracy is 88.0 percent. Although the accuracy for year 2000 is just 66.0 percent, but the accuracy for the later years are high. This shows that the transition rules can be used for the purpose of predicting future urban growth.

V. CONCLUSION

In this research, we propose a set of transition rules of cellular automata theory obtained from identifying, classifying and calculating the value of surrounding cells of temporal digital maps. In predicting urban growth, it is possible to hybrid cellular automata with other technique for better accuracy results. The case study focuses on a rapid growth of an area in a developing country, Malaysia (in the South East of Asia). In order to test the proposed transition rules, a prototype engine is designed and developed to evaluate the effect of surrounding cells over the center cell in the digital map. The distant of one pixel (with 8 surrounding cells) and two pixels (for 24 surrounding cells) are determined and a set of algorithms are produced based of mathematical notations.

REFERENCES

[1] Fuqiang, D. (2010). Mining dynamic transition rules of cellular automata in urban population simulation. *2010 Second International Conference on Computer Modeling and Simulation.* IEEE.

[2] Samat, N. (2007). Integrating GIS and CA Spatial Model in Evaluating Urban Growth:Prospect and Challenges. *Jurnal Alam Bina,Jilid 09, No:01* .

[3] Liu X., Li X., Yeh A., He J.,Tao, J. (2008). Discovery of transition rules for geographical cellular automata by using ant colony optimization. *International Journal of Geographic Information Science* .

[4] Sudhira, H. (2004). Integration of Agent-based and CA Models for Simulating Urban Sprawl. *Indian Institute of Remote Sensing* .

[5] Sudhira H., Ramachandra T., Wytzisk. (2003). Urban Sprawl Pattern Recognitionand Modeling using GIS. *Municipal India, Map India Conference 2003* .

[6] Mavroudi, A. (2007). *Simulating Urban Growth using CA Algorithm.* UCL.

[7] Li X., Yang Q.S., Liu X.P. (2008). Discovering and evaluating urban signatures for simulating compact development using cellular automata. *Landscape Urban Plan* , vol. 86.

[8] Zamyatin A., Markov, N. (2004). Approach to Land Cover Change Modeling Using Cellular Automata.

[9] Vliet J., White R., Dragicevic, S. (2009). Modeling Urban Growth Using a Variable Grid Cellular Automaton. *Comput. Environ. Urban Syst.* , vol. 33.

[10] Mandelas E., Hatzichristos T., Prastacos, P. (2007). A Fuzzy CA Based Shell for Modeling Urban Growth:A Pilot Application in Mesogia Area. *10th AGILE International Conference for Geographic Information Science 2007.*

[11] Alkheder S., Shan J. (2005). CA Urban Growth Simulation and Evaluation:A Case Study of Indianapolis.

[12] Ding Y.C., Zhang Y.K. (2007). The simulation of urban growth applying Sleuth CA model to the Yilan Delta in Taiwan.

[13] Yang Q., Li X., Shi X. (2008). Cellular Automata for simulating land use changes. *Comput. Geosci.* , vol. 34.

[14] Huang H., Zhang L., Guan Y., Wang D. (2008). A cellular automata model for population expansion of Spartina alterniflora at Jiudansha Shoals,Shanghai,China. *Estuar. Coast. Shelf S.* , vol. 77.

[15] Alexandris A., Vakalis V., Siettos S.I, Bafas G.F. (2008). A cellular automaton model for forest fire spread prediction. *Appl. Math. Comput.* , vol. 204.

[16] Fuqiang, D. (2010). Mining dynamic transition rules of cellular automata in urban population simulation. *2010 Second International Conference on Computer Modeling and Simulation.* IEEE.

[17] Torren P. (2007). *Cellular Automata.* Encylopedia of Human Geography Lodon.

[18] Abidin, Siti Z.Z., Jamaludin, M.N.F and Abiden, M.Z.Z, (2010) *Introducing an Intelligent Computerized Tool to Detect and Predict Urban Growth Pattern*, WSEAS Transaction on Computers, Issue 6, Vol 9.