URBAN GROWTH SIMULATION
USING REMOTE SENSING IMAGERY AND NEURAL NETWORKS

Sharaf Alkheder   and   Jie Shan

Geomatics Engineering, School of Civil Engineering, Purdue University,
550 Stadium Mall Drive, West Lafayette, IN-47907, U.S.A – (salkhede, jshan)@purdue.edu

ABSTRACT:

This paper presents a methodology for simulating urban growth phenomenon through utilizing remote sensing imagery and neural network (NN) algorithms. Historical satellite images of Indianapolis city, IN were used. All images were rectified and registered to Universal Transverse Mercator (UTM) NAD83 zone 16N. Supervised classification was used to classify the images to different land use categories. Seven classes were identified: water, road, residential, commercial, forest, pasture grasses and row crops. Image fusion was tested to examine its effect on the classification results. Overall, the classification accuracy using original images was better than the results from the fused ones. To implement NN algorithms to simulate the urban growth; focus was directed to the residential and commercial classes and their growth. The boundaries of these areas were extracted for each of the growth years. Radial extent of the boundary at specified angles was measured using different city centres. Radial distances and growth years were used as inputs to train the neural network algorithms. Two NN algorithms were used to simulate the urban growth: simple linear NN and back propagation (BP). Each algorithm was trained using the available data as well as interpolated data produced through NN function approximation algorithm. Short and long term urban boundary predictions were performed. Results showed that both algorithms after increasing the volume of the dataset succeeded in simulating the growth trends with better results achieved using the simple linear NN. Visual check and similarity of simulated and real growths were tested.

1. INTRODUCTION

1.1 Neural Networks and Urban Growth

Urban growth is a complex process that encounters a number of sophisticated parameters that interact to produce the urban growth pattern. These parameters included - but not limited to - land use suitability, city development level and economical phase. Due to the complexity of urban growth process dynamics; defining a mathematical model to describe such a process is extremely difficult. Artificial intelligence tools that include, but not limited to, Neural Network (NN), Fuzzy Logic systems and cellular automata are gaining increasing popularity in simulating urban growth. Many situations – such as urban growth – in which the phenomena involved, are very complex and not well understood and for which modelling principles are not possible, the Artificial intelligence tools are the best choice (Tsoukalas and Uhrig, 1997). Li and Yeh (2002) presented a method for integrating NN, GIS and Cellular Automata for the purpose of simulating different development patterns based on the planning objective. Backpropagation NN was used to simplify the structure of the model and to facilitate parameter values determination. Training of the Backpropagation NN algorithm was performed using historical remote sensing data. Li and Yeh (2001) implemented NN to determine the Cellular Automata simulation parameters through importing the parameters values from the training of NN into the cellular automata models. Liu (2000) adapted a new method to detect the change from non-urban to urban land use through using artificial neural network (ANN). Through using Levenburg-Marquart algorithm, the three-layered Backpropagation network was able to identify the changes of interest with an overall accuracy of 93.10%. Artificial Intelligence technologies (artificial neural network and fuzzy logic) have shown potential ability as a part of the artificial intelligence tools to be applied for solving natural resources assessment and spatial data processing in recent years as a part of the development in the information technology (Smith, 1984).

Neurocomputing principles are the core of the new neural spatial interaction models. These models are related to unconstrained spatial interaction models of the gravity type which can be expressed as special case of feedforward neural network models with a single hidden layer and sigmoidal transfer functions (Fischer, 1998).

In urban growth, (Pijanowskia et al., 2002) integrate both the artificial neural network and geographic information systems for the purpose of forecasting the change in land use. The scheme of their work starts with design of the neural network and identify the inputs using a historical data, using subset of the inputs the network was trained, then neural network testing was performed using the full data set of inputs and the final stage was using the information from the neural network to forecast changes.

The neural network learning model approach constituted a new technique, complementary to dynamic systems analysis, for gaining insight into the problems of structure evolution (White, 1989). White developed a neural network model and applied it to urban structures. Dougherty (1995) reviewed a variety of neural network models that have been applied to the field of transportation studies. Rodrique (1997) provided a conceptual overview of parallel computer process modelling and neural networks.

1.2 Statement of the Problem

Indianapolis city in Indiana State exhibited accelerated rates of urban growth over the last three decades. Being the main city in

Indiana, Indianapolis undergoes a lot of economical activity and development in term of urban growth and the rapid expansion of infrastructure. Such a growth makes Indianapolis, after it was composing a small part of Marion County in seventies, to expand and cover whole Marion County and parts of neighbour counties in 2003. The main objective of this paper is to implement the Neural Network (NN) concepts through utilizing two well-known NN algorithms to simulate the urban growth boundaries of Indianapolis city over a study period of more than three decades. Two neural network algorithms (Simple Adaptive Linear NN (SALNN) and Back Propagation BPNN) were trained using two datasets collected assuming two different locations of the city centre to test the effect of the centre location on the similarity of simulation results. The two algorithms were implemented using real and interpolated data for each dataset. Using both of the algorithms short and long term predictions were made and the simulation results were compared to the real boundaries visually. The number of hidden layers and weights for each algorithm were selected to best fit the growth pattern.

2. RAW DATA PROCESSING

This section presents a complete description for all the urban growth simulation using NN algorithms starting from the very beginning steps through study area images pre-processing, image classification, data set preparation and ending with NN algorithms implementation.

2.1 Data Preparation

2.1.1 Study Area: Indianapolis City in Marion County located in State of Indiana (Figure 1) was selected for the purpose of this study. Indianapolis located on latitude 39° 44' N and longitude of 86° 17' W. The city is considered as the largest and main economical capital of the State of Indiana. It has encountered during the last three decades a rapid urban growth movement associated with the increase in population and industry infrastructure. This undergoing growth resulted in the expanding of Indianapolis city to cover parts of other counties while it was before only part of Marion County. This arises the need of understanding the growth pattern in Indianapolis to help municipalities for planning infrastructure development and services accordingly.
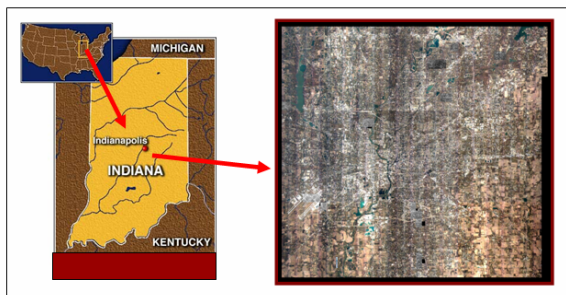

Figure 1. Indianapolis city (map and image)

2.1.2 Data: six historical satellite images that cover Indianapolis and surrounding areas at different years were obtained; one Landsat MSS 60-meter image (1973) and five Landsat TM 30-meters image (1982, 1987, 1993, 2000 and 2003). The selection of these images covers an urban growth period of thirty years over Indianapolis area.

2.1.3 Data Pre-processing: The historical images were geometrically rectified and registered to the same projection namely, Universal Transverse Mercator (UTM) NAD 1983 to lay them over each other. Second order polynomial transformation function was used to resample the new projected images to the same resolution of 30 meters. Figure 2 shows the rectification and registration results as an example for the 1973 and 1982 images. Registration errors were far less than one pixel.
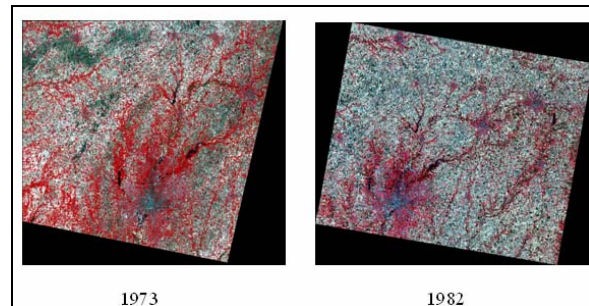

Figure 2. Examples of the rectified images

2.2 Image Classification: In order to classify the rectified images, seven classes of interest were specified in the images namely, water, road, residential, commercial, forest, pasture grasses and row crops. These classes were identified using sets of high resolution orthophotographs over the area and USGS land classification map as ground references. Two classification schemes were implemented to select the one with higher classification accuracy results: fused images classification and rectified only classification. The 2003 image 30-meter resolution was fused using the multiplicative method with a panchromatic image 15-meter resolution with the same extent. The fused 2003 image and the original 2003 images were classified. To provide same comparison conditions we used the same training samples locations on both images through creating the training samples on the rectified image and import them back to the fused image. The number of training pixels increased by four times for the fused image to match the increase in the resolution ratio (1:4). Figure 3 shows the results of fused and rectified images classification. It indicates that classification results from the original 2003 image was overall better than those of the fused one; some areas in the fused images were classified better than the original image such as forest class but other classes were deteriorated extensively such as commercials. It is decided to classify the original images without fusion.
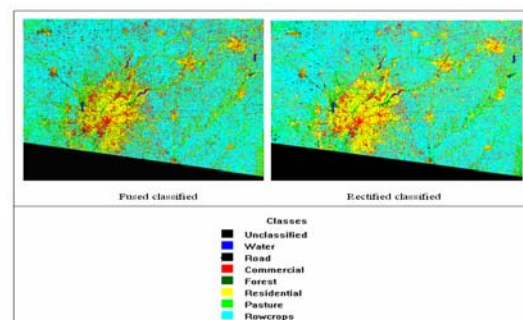

Figure 3. Fused vs. original (rectified) image classification

The overall testing accuracy matches with our expectation. For the classification of the original image, it is 89.14%, while it was 84.00% for the fused classified image.

3. NEURAL NETWORK GROWTH SIMULATION

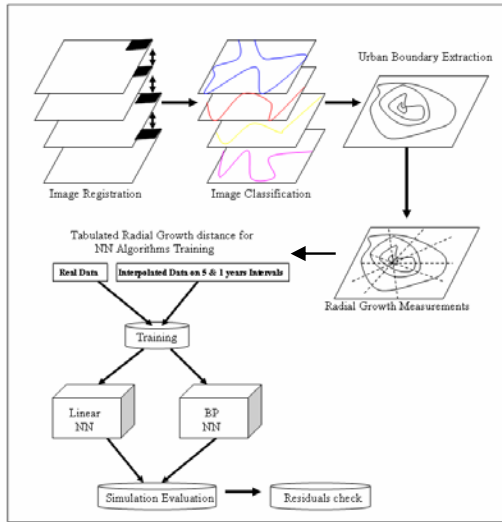The scheme of analysis we followed for the purpose of urban growth simulation is shown in Figure 4.

Figure 4. Urban growth simulation scheme using NN

3.1 Training Datasets Preparation

The six classified images were overlayed over each others and a new image was produced using ArcGIS containing the six years historical urban growth boundaries of Indianapolis area (Figure 5). Once these growth boundaries were identified, we created a circle over the boundaries image which was bisected to 360 degrees at an interval of 3 degrees. Two circles centred at two different locations were plotted over the boundary image in order to prepare two different datasets for each centre configuration (Figure 6). The purpose of selecting two different city centres was to test if there is any effect caused by such centre shift on the simulation results.
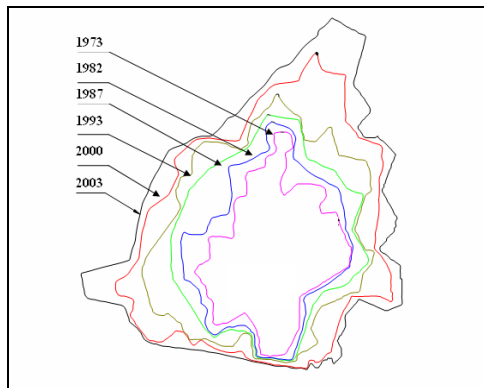

Figure 5. Indianapolis boundaries for the years 1973, 1982, 1987, 1993, 2000 and 2003 from the classified images
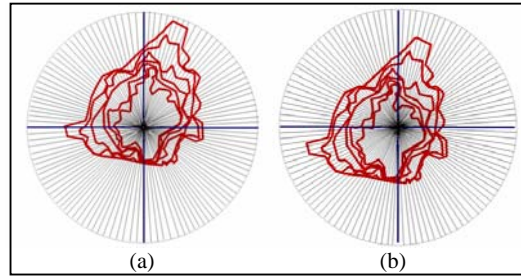

Figure 6. Two centres configuration for training data collection: (a) City center_1 location: (x, y) = (572052.517, 4401994.196) meters and (b) City center_2 location: (x, y) = (576594.305, 4409088.584) meters

For every centre configuration and in order to prepare the training dataset, at every angle starting from zero to 360 degrees six measurements were recorded representing the radial urban growth at 1973, 1982, 1987, 1993, 2000 and 2003 respectively. A matrix of 120 by 6 measurements was prepared. This matrix represented real data training condition for the NN algorithms. Another two more data sets were prepared using the Radial Basis Function algorithm (RBFN) algorithm described in Figure 7 (Tsoukalas and Uhrig, 1997). The RBFN used the years of available data (X1) and the radial distances measured at these years (X2) as two inputs to approximate a Gaussian function of this relationship. The data in-between was interpolated at intervals of every 5 and 1 years which was used beside the real data to train the NN algorithms. RBFN algorithm adapted the connected weights (wNi) automatically to produce exact fit of the approximated function to the real data points in order to interpolate in-between data with high accuracy. Output interpolated value (ON) will be controlled by the value σ - which defined as the radial basis function width for individual pattern unit - for high accuracy interpolation results. Exact radial basis option was used for the algorithm with 5.4 spread value in order to design a radial basis network with zero error on the design vectors to get a high accuracy interpolation of the simulation data.
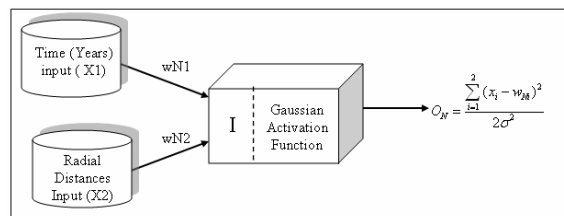

Figure 7. Gaussian kernel activation function for RBFN algorithm

The results for each centre configuration were three data sets including the real data only in one set and the real data combined with each of the interpolated data on 1 and 5 years intervals basis for the other two datasets.

3.2 Neural Network Algorithms

As mentioned earlier two of the well-known NN algorithms were trained using the three prepared datasets for every center configuration to simulate the urban growth. The adaptive linear NN as well as BP algorithms were used for the purpose of

predicting the radial distance as a function of angular distribution for short and long term predictions scenarios. The algorithms were presented with data of radial extent of the urban outline. Before starting the NN algorithms training the produced data sets were normalized to avoid any numerical problems in the training process.

3.2.1 Simple Adaptive Linear Neural Network (SALNN)

The first NN algorithm used for training through using the three data sets in order to predict the urban growth boundary pattern was the simple adaptive linear NN (SALNN). The structure of this algorithm used in this simulation is shown in Figure 8. As can be seen, the algorithm takes the vector of input variables X which represents the radial distances and growth years input parameters to train the network. A single input linear layer with one neuron structure was used. Three delay units were used to predict the new growth observation value. Weights were updated automatically based on the error difference between the input radial distance and the target value according to the following equation (Tsoukalas and Uhrig, 1997):

$$\Delta wi = \eta.E.Xi \tag{1}$$

where Δwi = change in weight after specified epoch
η = learning coefficient (0.05 in our case)
E = XT - X = error term for updating the weight after every epoch

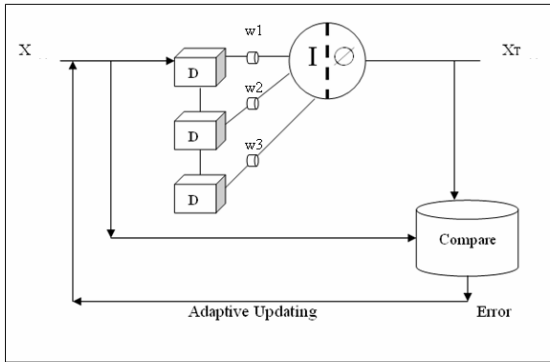The activation function $\varnothing(I)$ used for this algorithm was linear.

Figure 8. SALNN structure

The Simulink model was simulated using all simulation parameter dialog settings. A linear layer created for the purpose of training based on the simulated results returns a new linear layer with the predicted values.

The three datasets prepared in the previous sections - the real dataset, real dataset with interpolated data over 5 years interval and the real dataset with the interpolated data over 1 year interval – for each of the city center configuration were used to train the SALNN algorithm. Using the input datasets and SALNN architecture provided, a short and long term prediction were performed using the different datasets.

3.2.2 Backpropagation Neural Network Algorithm (BPNN)

This is the second algorithm used for simulating the urban growth pattern. The structure of this algorithm is more complex compared to the SALNN structure which might affect the accuracy of results in the case of small datasets. Figure 9 shows a simple representation of the BPNN algorithm used for the purpose of growth simulation. The same input variables of radial distances and growth years (X1) used in the SALNN algorithm were used here. Tan-sigmoid transfer activation function was used for the activation of hidden neurons. Linear transfer function purelin activation function was used for the output neuron activation (Tsoukalas and Uhrig, 1997).
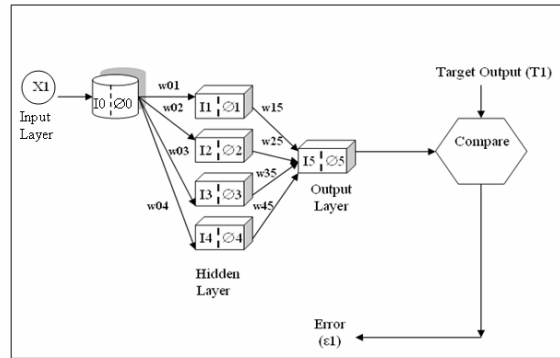

Figure 9. Backpropagation training algorithm structure

For the purpose of calculating the updated weights for the Backpropagation algorithm, delta training rule was used. All random weights initialized on the algorithm connections were updated. To update weight and based on the delta training rule we have for output element (w45 as example);

$$\Delta w_{45} = -\eta_{45}.\frac{\partial \in_1^2}{\partial w_{45}} \tag{2}$$

Since $\in_1^2 = (T1 - \varnothing 5)^2$

Then;

$$\Delta w_{45} = -\eta_{45}.\frac{\partial I5}{\partial w_{45}}.\frac{\partial \varnothing 5}{\partial I5}.\frac{\partial \in_1^2}{\partial \varnothing 5}$$

$$\Delta w_{45} = -\eta_{45}.\varnothing 4.\frac{\partial \varnothing 5}{\partial I5}.2\in_1 \tag{3}$$

$$\Delta w_{45} = -\eta_{45}.\varnothing 4.\delta 1$$

$$w^{new}_{45} = w^{old}_{45} + \Delta w_{45} \tag{4}$$

For hidden element (w01 as example) through using the same principle for output layer we have:

$$\Delta w_{01} = -\eta_{01}.\varnothing 0.\frac{\partial \varnothing 1}{\partial I1}.w_{15}\delta 5$$

$$\Delta w_{01} = -\eta_{01}.\varnothing 0.\delta 1 \tag{5}$$

$$w^{new}_{01} = w^{old}_{01} + \Delta w_{01}$$

where $\delta 1$ is a constant proportional to the error and the change in the activation function.

For the BP algorithm, the problem was set up to consist of inputs and targets that would like to solve for the radii with the network. A two-layer feedforward network was created, the first layer had four TANSIG neurons, and the second layer had one PURELIN neuron as the transfer function. The TRAINLM network training function was used. The network was simulated and its output was determined against the targets. Afterwards the network was trained for 10000 epochs, and then again the network's output was determined. With these settings, the network was allowed to converge for all the angles for all the real and interpolated years. 1e-9 performance goal was specified.

4. RESULTS AND DISCUSSIONS

As mentioned earlier both of the NN algorithms: SALNN and BPNN were used to predict the urban growth using the three datasets per center configuration. The three datasets contain one set for real data only, one set for the real and interpolated data on a 5 years interval and one final set with the real and 1 year interval interpolated data. Two types of urban growth prediction were performed for each selected city center namely, short term and long term prediction. Using the three data sets for each center configuration, the urban growth boundary for the year 2000 was predicted using the years 1973, 1982, 1987 and 1993 for a long term prediction over 7 years. For short term prediction, years 1973, 1982, 1987, 1993 and 2000 were used to predict the urban boundary for the year of 2003. For the first city center (part a in Figure 6), the short and long term predictions results for the years 2003 and 2000 respectively can be seen in Figures 10 to 13 using SALNN and BPNN algorithms for all the three datasets. Figures 14 to 17 show the prediction results for the second center.
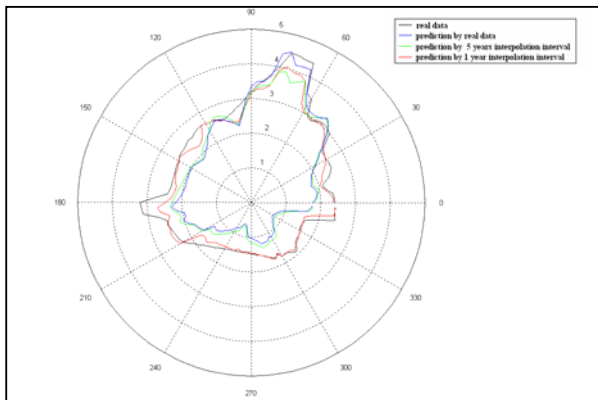

Figure 10. SALNN 2003 short term prediction for first center

Results as can be seen in the last figures showed that the best prediction using both algorithms and for the two centers was achieved using the third dataset that includes the real data plus the interpolated data per 1 year interval; this is because the size of the data is increased and hence improved the training and prediction results. In the case of both centers, the results showed that SALNN gave better results compared to the BPNN results as can be seen visually. This is due to the fact that under the limitation of the availability of the data the SALNN works better under such conditions compared to the complexity of the BPNN that needs more data for training. Considering the two
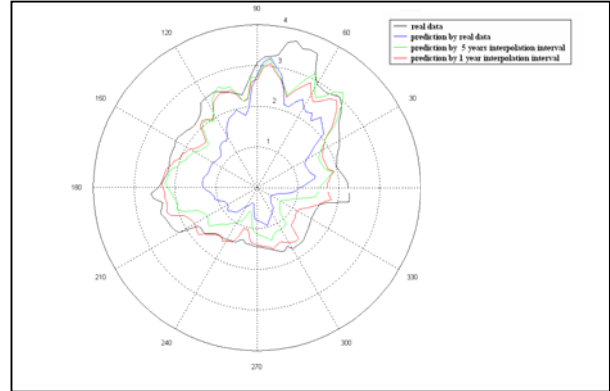

Figure 11. SALNN 2000 long term prediction for first center
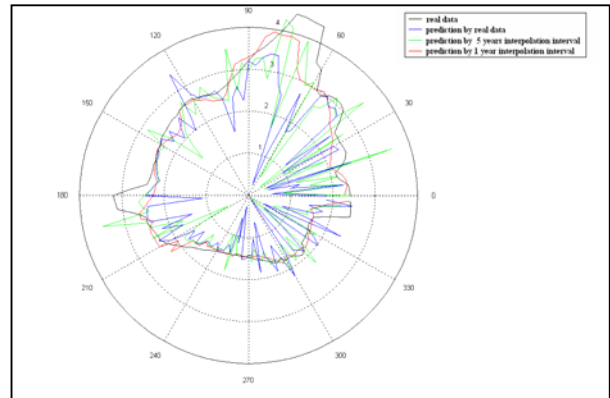

Figure 12. BPNN 2003 short term prediction for first center
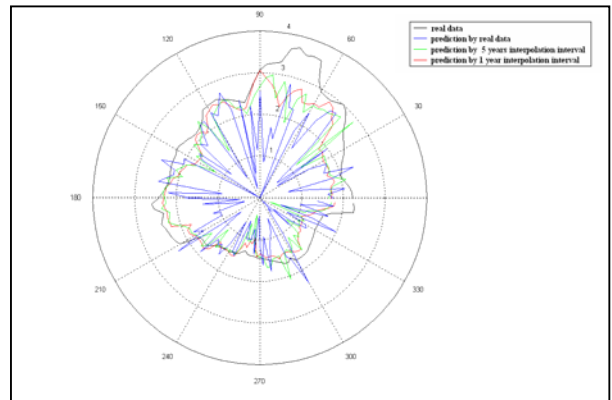

Figure 13. BPNN 2000 long term prediction for first center

centers results we can see that the results of predictions is somewhat independent on the centers location especially for the third dataset where the results were very close using both of the algorithms for the two centers. Finally it is obvious that short prediction results for the year 2003 for an interval of 3 years produce more accurate prediction compared to the year 2000 results with an interval of 7 years.

5. CONCLUSIONS

The methodology described in this paper showed the potential of implementing NN algorithms as a tool to predict the urban growth pattern based on historical satellite images. Two NN
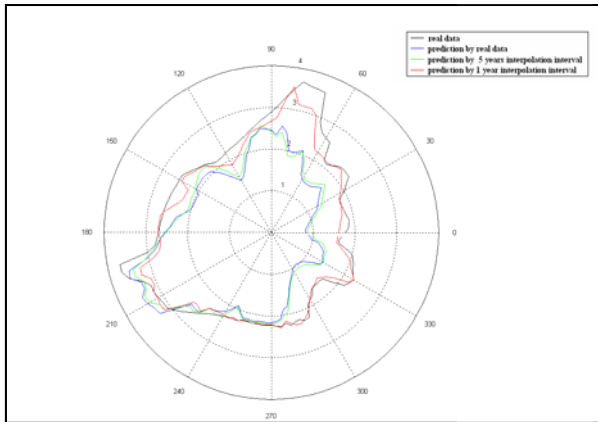
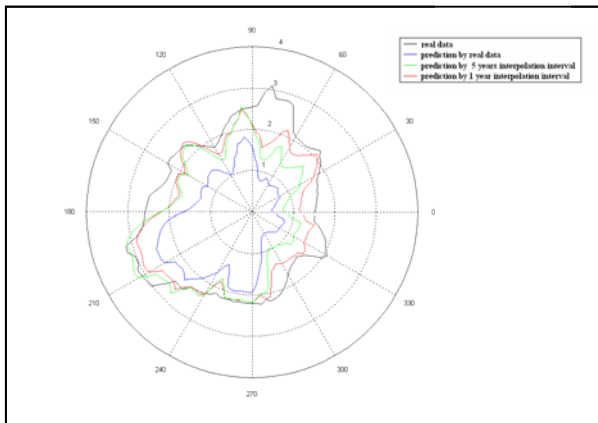Figure 14. SALNN 2003 short prediction for second center

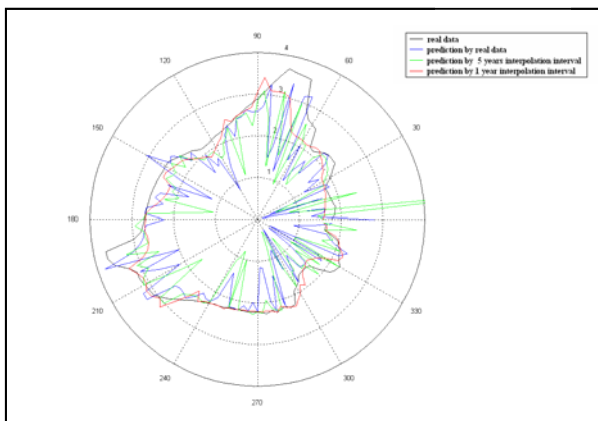Figure 15. SALNN 2000 long term prediction for second center

Figure 16. BPNN 2003 short term prediction for second center

algorithms were tested and two city centers were defined to test the effect on the simulation results. Three data sets were prepared including the real and interpolated data based on RBFNN algorithm. Results indicated that SALNN algorithm produced better results given the limited size of the available data than BPNN. The interpolation data interval has an important effect on the training performance and prediction results. The selection of the city center location has certain effect on the predicted urban growth pattern. However, looking to the third dataset results it can be seen that results are very close regardless of the center location.
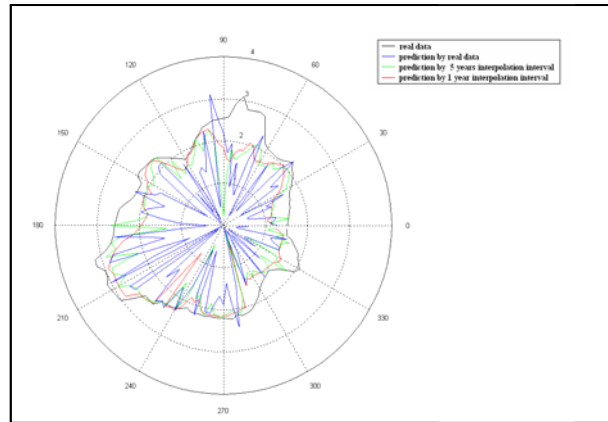

Figure 17. BPNN 2000 long term prediction for second center

One possible improvement we are working on now is to make the input dataset independent totally on the center location to improve the accuracy of the simulation.

6. REFERENCES

Dougherty M., 1995. A review of neural networks applied to transport. *Transportation Research C*, 3(4), pp. 247-260.

Fischer, M. M., 1998. Computational neural networks: A new paradigm for spatial analysis. *Environment and Planning A*, 30 (10), pp. 1873-1891.

Li, X. and Yeh, A. G. O., 2002. Urban simulation using neural networks and cellular automata for land use planning. In Proc. *ISPRS Commission IV on Geospatial Theory, Processing and Applications*, Ottawa, CA. Vol. XXXIV, part 4, ISSN 1682-1750.

Li, X. and Yeh, A. G. O., 2001. Calibration of cellular automata by using neural networks for the simulation of complex urban systems. *Environment and Planning A*, 33(8), pp. 1445-1462.

Liu, X. H., 2000. Change detection for urban growth modeling: an artificial neural network approach. In Proc. *4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs*. Banff, Alberta, CA.

Pijanowskia, B. C., Brown, D. G., Shellitoc, B. A., and Manikd, G. A., 2002. Using neural networks and GIS to forecast land use changes: a land transformation model. *Computers, Environment and Urban Systems*, 26(6), pp. 553-575.

Rodrique J., 1997. Parallel modelling and neural networks: An overview for transportation/land use systems. *Transportation Research C*, 5(5), pp. 259–271.

Smith, T. R., 1984. Artificial intelligence and its applicability to geographical problem solving. *Professional Geographer*, 36 (2), pp. 147-158.

Tsoukalas L. H. and Uhrig R. E., 1997. *Fuzzy and Neural Approaches in Engineering*. John Wiley and Sons, Inc., New York, pp. 189-289 and pp. 385-405.

White R. W., 1989. The artificial intelligence of urban dynamics: Neural network modeling of urban structure. *Papers of the Regional Science Association*, 67, pp. 43–53.