



Yoshihiro Kobayashi

sunp@ucla.edu
The Department of Architecture
and Urban Design
School of Art and Architecture
University of California, Los Angeles

Kostas Terzidis

kostas@ucla.edu
The Department of Architecture
and Urban Design
School of Art and Architecture
University of California, Los Angeles

Extracting the Geometry of Buildings from Satellite Images

Using Fuzzy Multiple Layer Perceptrons

Abstract

This paper presents Computer Aided Architectural Design (CAAD) system utilizing the technologies of artificial intelligence (AI) and image processing. The goal is to create a CAAD system that detects buildings from satellite images and produces computer city models allowing the system's users to manipulate the models utilizing machine learning technology. The flexibility and usability of the system was evaluated with case studies. Soft computing technologies including neural networks and fuzzy systems are mainly applied and tested as the system's methodology.

Resumen

Este trabajo trata sobre un sistema de Diseño Asistido por Computadora (CAAD) que utiliza las tecnologías de Inteligencia Artificial (AI) y procesamiento de Imágenes. El objetivo es crear un sistema CAAD que detecta los edificios en las imágenes de satélite y genera modelos computarizados de la ciudad permitiendo que los usuarios del sistema manipulen los modelos utilizando tecnología de "machine learning" o de aprendizaje de máquina. La flexibilidad y aplicabilidad del sistema fue evaluada en casos de estudios. Tecnologías de computación suave que incluyeron redes neurales y sistemas borrosos son aplicados y evaluados como la metodología del sistema.

I. Introduction

Currently it has become important for designers and planners in architecture and urban planning to create computer city models in order to visualize, simulate, and estimate their design plans (Liggett and Jepson, 1995). However, in most of cases when designers and planners need 3D computer city models, they create city objects such as buildings, streets, and houses step by step manually because required level of detail is different in each case. From the technology side in image processing, the study of building extraction from aerial and satellite images in image processing has been researched and developed since the 1980's and the application of image understanding (IU) has become practical due to the improvement in computer speed and price. However, it has been little applied in practical architectural and urban designing tasks. For example, most of designers and planners prefer to use the current commercial Computer Aided Design (CAD) and Computer Graphics (CG) systems, because those systems have a sufficient ability to create complex shapes and realistic scenes even though it takes a lot of time and labor to create objects by combining a lot of functions and commands. Therefore, it is important to implement a new CAAD system that reduces the labor of designers and planners by integrating IU technologies in creating computer city models and automating as many stages of the process as possible.

In the study field of image understanding, there are basically two approaches, rule based approach and soft computing approach. Traditional rule based UI approach in building extraction are limited because they cannot flexibly adapt enough to various situations that designers and planners require, though some projects, (Chellappa, 1994), (Hsieh, 1996), (McKeown, 1990), have tried to apply expert systems in order to implement the adapting ability with a collection of <If ~, then ~> rules. This approach is not user friendly because it is possible for the system developers to adapt the system by adding rules in the expert systems, but it is impossible for the system's users to allow the opportunities to adapt the system. On the other hand, soft computing approach has the advantage of adapting the systems by the system users, and several techniques have been proposed and developed since the beginning of 1990's.

One of the soft computing approaches, *neuro-fuzzy system*, are very applicable in allowing system's users to adapt the system, because neuro-fuzzy system has both advantages of neural networks and fuzzy systems. In short, the system has the ability of learning of neural networks and representing the rules in linguistic forms, which is the ability of fuzzy systems. Therefore, the system's users can make the system adapt to the expecting situations and can understand how the system learns. Several projects applying this approach have been reported in pattern recognition (Pal and Mitra, 1999).

Given the advantages described above, this project uses neuro-fuzzy systems into a CAAD system integrating IU techniques. In other words, this project is theoretically links two



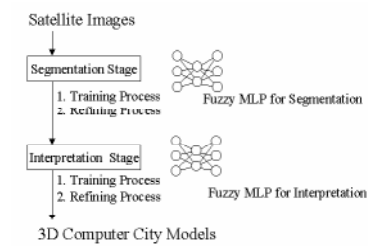


figure 1 - Outline Process

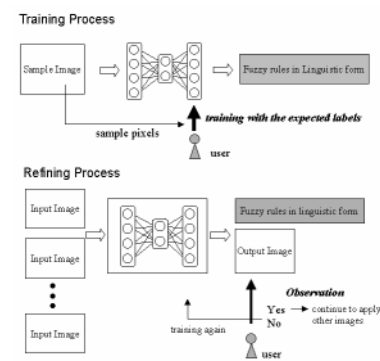


figure 2 - Training and Refining Process



Figure 3 - Interface Image

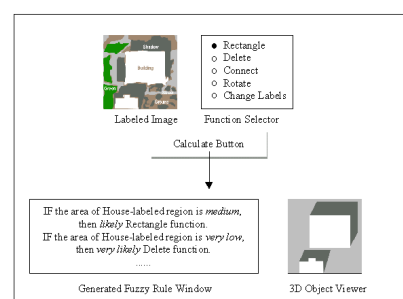


figure 4 - Diagram of Interpretation Stage

research fields. One is the study of image understanding (IU) in image processing, especially the study of monocular building extraction, an application of IU. The other related research field is the study of machine learning (ML) in AI, especially the study of how to make computer systems gain architectural and urban design knowledge. Therefore, the goal of this research can be defined as the integration of these two technologies in order to create 3D computer city models for architectural and urban design.

2. Methodology

2-1. Model

In this project, the model of fuzzy multiple layers perceptron (MLP) (Pal, 1992) is used and tested. Fuzzy MLP is one of the neuro-fuzzy systems. It is the model incorporating fuzziness into the neural net framework, and has the ability of representing the complex relations that the weight matrices of neural networks learned in linguistic forms with fuzziness. For example, the model can generate the rules such as "IF feature 1 is very low and feature 2 is high, then the result is very likely Case 2". The structure of neural networks belongs to feedforward type that has three kinds of layers, input, output, and hidden layers. The backpropagation algorithm is used for learning.

2-2. Stages

In order to generate 3D computer city models, this system requires two stages, which are segmentation and interpretation. The segmentation stage involves the assignment of pixels in images to specific labeled classes and the generation of regions by connecting the similar neighboring pixels. The interpretation stage involves, on the other hand, the process of determining what the labeled regions are after the segmentation stage. In other words, it is the stage where we apply each region to predefined function and generate 3D objects.

Inside each stage described above, there are two processes: the training and the refining process. The training process is the first process to supervise networks manually when there is no previous experiment. The refining process is to rearrange the networks that have refined or not by observing the generated fuzzy rules and output images. The details of these processes are shown in the figure 1 and 2.

2-3. Interface

The interface of application program in this project has mainly three components: a training screen, a diagram screen, and an information screen. In the training screen, the system's users can load a satellite image, train the images into several categories, and see the visual output images. In the diagram screen, the users can observe the weights of the neural networks. In the information screen, the user can see the information of samples in training process and the fuzzy rules in linguistic forms after calculating the relations of inputs and outputs. The interface of this program is shown in the figure 3.

2-4. Scenario

The first stage to run this system is segmentation stage. Initially, the user of this system can load a satellite image, and then the image is set in the training screen automatically. Next, the user can pick sample pixels one by one and select one category corresponding to each picked pixel manually. The user can also observe the information of the picked pixel in the information screen. After 1% or more pixels are categorized, the user can calculate the neural networks. In the first calculation, the weights of neural networks are generated randomly and supervised with the picked sample data. Then, the categorized output image is seen in the training screen and the generated fuzzy rules in the information screen. The diagram screen provides the condition of neural networks. If the output image is close to what the user expects, the training has finished. Otherwise, the user observes the generated fuzzy rules and find what kind of information is necessary and lacking of. Then, the user categorizes more pixels by evaluating the fuzzy rules in order to adapt the system to what the user requires.

The next stage is interpretation stage. In this stage, the output image produced in the segmentation stage is applied. The user can load the output image and apply several geometric functions in order to generate 3D objects. First, the user picks sample pixels one by one and select one function corresponding to the picked segment manually. After 1% or more pixels are categorized, the user can calculate the neural networks as the same as the previous stage. If the user is satisfied with the output, the training has finished. Otherwise, more training process is repeated. The diagram of this stage is shown in the figure 4.

After the above two processes, the neural networks for both segmentation and interpretation are well trained. Therefore, using these well-trained neural networks, the user can load different satellite images and generate 3D objects automatically without any training.

3. Algorithm

3-1. Input Data

The input data for fuzzy MLP is represented as a set of feature values. Each feature is represented in terms of some combination of membership values in the linguistic property sets *low* (L), *medium* (M), and *high* (H). For example, when the j th input feature value F_j is small, the feature is represented as,

$F_j = \{\mu_{low}(F_j), \mu_{medium}(F_j), \mu_{high}(F_j)\} = \{0.95, 0.5, 0.05\}$. The input is represented as a set of these features, $\mathbf{F} = [F_1, F_2, \dots, F_n]$, where n is the number of features. The number of input neurons must be $3n$, because each feature F_j needs three input neurons such as high, medium, and low in fuzzy MLP. This project applied π -function with the input value, the mean value, and the crossover value for getting the three membership values. (figure 5)

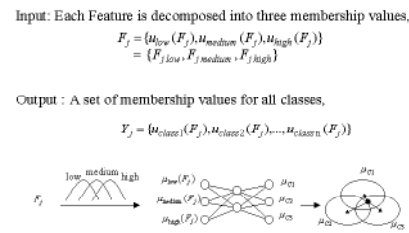


figure 5 - Input and Output of Fuzzy MLP

3-2. Output Data

The output data from fuzzy MLP is represented as a set of membership values for predefined categories. In addition to this output, the factor to estimate the belief of the results is applied to this project. The factor, bel_j^H , indicates how salient the result is comparing to the other choices, and is defined as $bel_j^H = y_j^H - \sum_{i \neq j} y_i^H$.

This value is used to choose linguistic forms among *very likely*, *likely*, *more or less*, *not unlikely*, and *unable recognize*. The chosen forms are applied to modify then-clauses in the generated fuzzy rules. The following shows the decision rules.

1. *Very likely* for $0.8 \leq bel_j^H \leq 1$
2. *Likely* for $0.6 \leq bel_j^H < 0.8$
3. *More or Less* for $0.4 \leq bel_j^H < 0.6$
4. *Not unlikely* for $0.1 \leq bel_j^H < 0.4$
5. *Unable to recognize* for $bel_j^H < 0.1$

3-3. Generating Rules (Justification)

Justification is the process to find the maximum weighted paths in the networks and generate fuzzy rules based on [Mitra, 1994]. This section explains how to find the paths

from the neuron j in the output layer H . First, select one output neuron j with $bel_j^H > 0$. Then, the weights $w_{ji}^{H-1} > 0$ connected to the neuron j are selected in the lower layer $H-1$. These neurons i connected to the neuron j are described as m^{H-1} , a set of indexes a_k^{H-1} . The maximum weight wet_i^{H-1} in the layer $H-1$ is $w_{ji}^{H-1} > 0$. The algorithm to find the maximum weighted paths is carried out in a top-down manner. For the hidden layers h , the maximum weight wet_i^h is calculated by the sum of the predefined maximum weight in the upper layer and the weights in the lower layers, and is defined as

$$y_i^h > 0.5, \quad wet_{a_k^{h+1}}^h = \max_{a_k^{h+1}} [wet_{a_k^{h+1}}^{h+1} + w_{a_k^h}^h]$$

where a_k^{h+1} indicates the index number of neurons in the upper layer $h+1$, $wet_{a_k^{h+1}}^h$ is

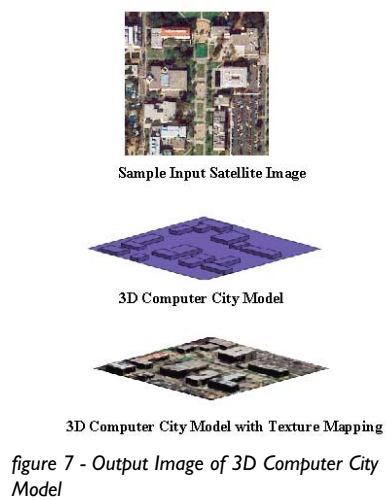


figure 7 - Output Image of 3D Computer City Model

References

- Liggett, R, Friedman, S, and Jepson, W. (1995). "Interactive Design/Decision Making in a Virtual Urban World: Visual Simulation and GIS," *Proceedings of the Fifteenth Annual ESRI User Conference*. Palm Springs, CA, (May).
- Chellappa, R., Zheng, Q., Davis, L. S., Lin, C. L., Zhang, X., Rodriguez, C., Rosenfeld, A., and Moore, T. (1994). "Site-Model-Based Monitoring of Aerial Images", *Proc. ARPA Image Understanding workshop*, vol. 1, 295-318.
- Hsieh, Y. (1996). "SiteCity: A Semi-Automated Site Modeling System," *Proc. IEEE Conference Computer Vision and Pattern Recognition*, June 1996, 499-506.
- Mckeown, D. M. (1990). "Toward Automatic Cartographic Feature Extraction from Aerial Imagery," In *Mapping and Spatial Modeling for Navigation*, L.F.Fau, (ed), NATOASI series, vol. F65, 149-180.
- Pal, S. K. and Mitra, S. (1999). *Neuro-Fuzzy Pattern Recognition*. New York: Wiley-Interscience.
- Pal, S. K., and Mitra, S. (1992). "Multi-Layer perceptron, fuzzy set and classification," *IEEE Transactions on Neural Networks*, vol.3, 683-697.
- Mitra, S. and Pal, S. K. (1994). "Logical operation based Fuzzy MLP for classification and rule generation," *Neural Networks*, vol. 7, 353-373.

the predefined weight in the upper layer; and $w_{a_i}^h$ indicates the weights of the neurons i in the lower layer h . When the maximum weighted paths reach the input layer, the connected neurons in the input layer become the candidates for if-clauses. Then the fuzzy rules are

generated among the candidate neurons m^0 until $\sum_{i_s} wet_{i_s} > 2 \sum_{i_n} wet_{i_n}$,

where i_s indicates the input neurons selected for the clause and i_n indicates the input neurons remaining from the set of m^0 as shown in the figure 6. In this case, the generated rule from the system would be "IF F1 is very medium AND F2 is high, THEN likely Class 1." (figure 6)

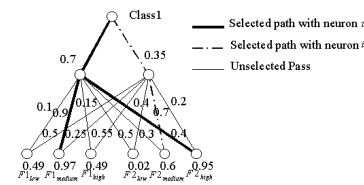


figure 6 - Example Rule Generation Scheme

3-4. Application

In this project, two individual fuzzy MLP networks are used for creating 3D computer city models from satellite images. One is for the segmentation stage, and the other is for the interpretation stage.

For the segmentation stage, the input data for fuzzy MLP is represented as a set of attributes of pixels in images such as the Red-value, the Green-value, the Blue-value, intensity, 3x3 neighbors mean, etc., and the output data of fuzzy MLP is represented as a set of class membership values. The user defines the number of classes and the labels of the classes such as "House", "Shadow", "Street", "Green", etc.

For the interpretation stage, on the other hand, the input data for fuzzy MLP is assumed as a set of attributes of the segmented regions in the segmentation stage such as the area, the maximum length, maximum width, the distance to the next region, the center position, etc. The output data of fuzzy MLP is represented as a set of the selected function membership value. The functions to manipulate the segmented region should be predefined such as "delete", "create rectangle", "combine", etc. The user trains the system by choosing one expected function corresponding to each region in order to refine the shapes and create 3D models.

4. Case Study

We used a satellite image of Los Angeles, which has 400x400 pixels. We picked 1500 pixels and trained them into 5 categories, which are "House", "Street", "Green", "Shadow", and "Ground". 300 samples are applied to neural networks for each category. The sampling was manually proceeded. The features in the segmentation stage are RGB values, intensity, 3x3 neighborhood average value. The features in the interpretation stage are the area, width, height, and length to the nearest segment. The output image of a 3D computer city model generated by this system is shown in the figure 7.

5. Conclusion

In this project, we demonstrated an implementation of a neuro-fuzzy system into a CAAD system. The most effective benefit is that the system's users can adapt the system to what they expect. Especially, fuzzy Multiple Layers Perceptron (MLP) could allow the users to observe the complex relations between inputs and outputs as a collection of fuzzy rules in linguistic forms. This makes possible for the system's users to understand how the neural networks are trained and what kind of relations are supervised.

However, there are still some problems in integrating neuro-fuzzy systems into CAAD systems. For example, it is difficult to define the optimum structure such as the number of hidden layers and the number of nodes in each hidden layer. In this project, the fuzzy MLP model has three hidden layers and ten nodes for each hidden layer, and they were fixed (hard-coded). Therefore, we did not examine how this structure may be applicable to our system. We should consider ways of finding the optimum structure for improving the system in the future. In other words, it can be concluded that the next step of research is to find the optimum structure of neural networks or control the structure adapting to given situations with keeping the ability of representing the relations of neural networks as a set of fuzzy rules.