# Musical Symbol Recognition using SOM-based Fuzzy Systems

Mu-Chun Su
Department of Computer Science and Information Engineering, National Central University, Taiwan, R. O. C.
**E-mail:** muchun@csie.ncu.edu.tw

Chee-Yuen Tew, and Hsin-Hua Chen
Department of Electrical Engineering, Tamkang University, Taiwan, R. O. C.

**Abstract**

Over the last two decades a large number of research activities have been undertaken to investigate into Optical Music Recognition (OMR). OMR involves identifying musical symbols on a scanned sheet of music and transforming them into a computer readable format. In this paper we propose an efficient method based on SOM-based fuzzy systems to recognize musical symbols. A database consisting of 9 kinds of musical symbols were used to test the performance of the SOM-based fuzzy systems.

**Keywords**: neuro-fuzzy systems, pattern recognition, neural networks, optical music recognition.

## 1. Introduction

Applications of optical music recognition (OMR) include providing an automatic transposition, extracting parts for individual instruments, and performing an automated musicological analysis of the music. The earliest attempts at OMR were made by Pruslin [1] and Prerau [2]. Many recognition methods have been actively researched since 70's [3]-[15]. The research literature for the period 1966 through 1990 has been critically examined by Blostein and Baird [16] and a survey of current work is given in [17]. Like Optical Character Recognition (OCR), OMR is a problem that has yet to be solved in a satisfactory manner because there are many obstacles to OMR. For example, imperfections introduced during the printing and digitizing process that are normally tolerable to humans can often complicate the segmentation process and the recognition process. Obstacles in OMR are addressed in [17]. Although the first commercial product appeared in 1993 [18] the slow adoption of OMR as a music acquisition method attests to the immaturity of the technology.

A typical OMR system consists of the following stages: acquisition, preprocessing (e.g. noise reduction and rotation), identification of staff lines, segmentation, musical symbol recognition, contextual postprocessing, and generation of a symbolic representation. The presented approach belongs to the topic of musical symbol recognition. Especially, we only address the recognition for note heads (except the whole notes). Several methods have been proposed to solve the recognition problem. Horizontal search method based on the staff position to extract node heads was proposed in [5], [6]. The method proved to be suffered from the influence due to the staff distortion. In addition, it is difficult to extract node heads which are located out of the staff regions. Some methods needed a set of complex If-Then rules and subtle adjustment of parameters through experiments [5]-[19]. Recently, neural networks and genetic algorithms (Gas) were proposed to recognize node heads [20]-[21]. However, multilayer perceptions using the backpropagation algorithm or GAs usually require a lot of time to converge.

In the paper, we propose to use SOM-based fuzzy systems to recognize musical symbols. This paper is organized into 4 sections. In the following section, we briefly review the properties of the SOM-based fuzzy systems. Experimental results are given in section 3. Finally, section 4 concludes the paper.

## 2. The SOM-based Fuzzy System

One of the challenges that arise in designing a fuzzy system is the trade-off between computational efficiency and performance. Basically, the more rules, the more powerful the fuzzy system becomes. However, the price paid for the high performance is that the computational load becomes extremely large. The class of SOM-based fuzzy systems provides an appealing and easy solution to solve the dilemma [22]. The SOM-based fuzzy system uses the Kohonen's self-organizing feature map (SOM) algorithm, not only for its vector quantization feature, but also for its topological property. The vector quantization feature of feature maps is used to search a good supply of most representative cluster centers.

Then the topology-preserving feature is fully utilized to select a set of most influential rules so as to contribute to the computation of system outputs. By behaving this way, the SOM-based fuzzy system provides an appealing solution to the trade-off between computational efficiency and performance.

### 2.1 The structure of the neuro-fuzzy system

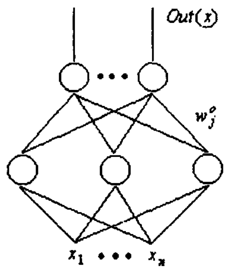


Figure 1: The general structure of conventional neuro-fuzzy systems.

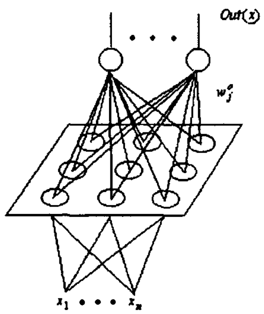


Figure 2: The structure of the proposed SOM-based fuzzy systems.

The general structure of most conventional neuro-fuzzy systems adopts a one-dimensional structure to arrange their hidden nodes, as shown in Fig. 1. The relationship between rules is not revealed from their placements. Therefore, all rules are usually simultaneously fine-tuned during the training procedure and contribute to the computation of the final crisp output. A side effect is a high computational load.

In our SOM-based fuzzy system, we use a two-dimensional structure to arrange rule nodes so that neighboring rule nodes will respond to similar inputs. The SOM algorithm is used to achieve the topology preserving property. The architecture of the SOM-based fuzzy system is shown in Fig. 2. The output of the neuro-fuzzy system is computed by the following equations:

$$Out(\underline{x}) = \sum_{j \in S_{j^*}} c_j m_j(\underline{x}) \tag{1}$$

where

$$m_j(\underline{x}) = \exp\left( - \frac{\|\underline{x} - \underline{w}_j\|^2}{2\sigma_j^2} \right) \tag{2}$$

$c_j$ is the connection weight from the $j$ th hidden node to the output node, $\underline{w}_j$ is the weighting vector of the $j$ th hidden node, $j^*$ denotes the winner that most responds to the input $\underline{x} = (x_1, \ldots, x_n)^T$, $S_{j^*}$ denotes the neighboring set of the winner, and $\sigma_j$ is a regulating parameter which controls how fast the membership function, $m_j(\underline{x})$, decreases. Here we want to emphasize that only the corresponding winner and its neighbors can contribute to the computation of the overall output of the neuro-fuzzy system.

### 2.2 The hybrid training algorithm

Training of the SOM-based fuzzy system is carried out in the following three phases.

**Phase 1 (unsupervised phase):**

Using the SOM algorithm to cluster the input data into $M \times N$ groups. In order to accelerate the training procedure, the initialization scheme proposed in [23] is utilized here. Note that the size of the network is predetermined by the user.

**Phase 2 (imitating phase):**

A so-called "imitating" learning scheme is adopted to initialize the connection weights (i.e. $c_j$) and regulating parameters (i.e. $\sigma_j$). The basic idea is very simple. When an inexperienced learner tries to learn some particular technique, he or she will probably observe the instructor first and then imitate how the instructor executes the technique. This motivated us to initialize the corresponding parameters in the following way. We present every input pattern to the trained SOM and then select the corresponding winner e.g. neuron $j^*$. In the following, the corresponding parameters of the winner are updated as follows:

$$c_{j^*}(new) = \frac{P_{j^*}(old)}{P_{j^*}(old)+1} c_{j^*}(old) + \frac{1}{P_{j^*}(old)+1} d(\underline{x}) \tag{3}$$

$$\sigma_{j^*}(new) = \max(\sigma_{j^*}(old), \|\underline{x} - \underline{w}_{j^*}\|) \tag{4}$$

and

$$P_{j^*}(new) = P_{j^*}(old) + 1 \tag{5}$$

where $d(\underline{x})$ denotes the desired output for the present input $\underline{x}$ and $P_{j^*}$ represents the winning times of the neuron $j^*$ during the competing procedure. The initial value for these three variables, $w_{j^*}$, $\sigma_{j^*}$, $P_{j^*}$, are all zero. Note that if some $\sigma_j$'s are still zero after presenting the whole input data set to the SOM, we will set them to be some small-valued numbers, e.g. 1's.

**Phase 3 (exploitative phase):**

A gradient-based method is employed to fine-tune the neuro-fuzzy system. We call this phase the exploitative learning phase. The goal is to minimize the error function

$$E = \frac{1}{2}(Out(\underline{x}) - d(\underline{x}))^2 \qquad (6)$$

According to the gradient decent method the updating formula for the corresponding parameters are given as follows:

$$c_j(new) = c_j(old) + \eta(d(\underline{x}) - Out(\underline{x}))m_j(\underline{x}) \quad for \quad j \in S_{j^*} \qquad (7)$$

$$\underline{w}_j(new) = \underline{w}_j(old) + \eta(d(\underline{x}) - Out(\underline{x})) * c_j(old)m_j(\underline{x})\frac{(\underline{x} - \underline{w}_j)}{\sigma_j^2} \quad for \quad j \in S_{j^*} \qquad (8)$$

$$\sigma_j(new) = \sigma_j(old) + \eta(d(\underline{x}) - Out(\underline{x})) * c_j(old)m_j(\underline{x})\frac{\|\underline{x} - \underline{w}_j\|}{\sigma_j^3} \quad for \quad j \in S_{j^*} \qquad (9)$$

It should be emphasized that we may fine-tune the neuro-fuzzy system in either of the following ways:

(1) We fix the values of $\underline{w}_j$'s and $\sigma_j$'s and update only $c_j$'s.

(2) We fix only the values of $\underline{w}_j$'s and update $\sigma_j$'s and $c_j$'s.

(3) We simultaneously update $\underline{w}_j$'s, $\sigma_j$'s, and $c_j$'s.

### 3. Experimental Results

The recognition method has been implemented in Borland C ++ Builder language using a K6-300 PC under Microsoft Window 98 environment. In the experiments, the recognition method was tested on four piano scores (e.g. the first piece of the Tchaikovsky album for the young) printed on A4 sheets. Using an image scanner, the musical score printed on A4 sheet is digitized and converted into a black-and-white bitmap. The performance of the SOM-based fuzzy systems are tested on a database consisting of 9 kinds of musical symbols (shown in Fig. 3) including rests, clefs, accidentals, etc. The database consists of 639 patterns. We split the database into two data sets – training data set consisting of 400 patterns and testing data set consisting of 239 patterns. Table I tabulates the recognition rates. On average the correct recognition rates are 96.92% and 97.86% for the training set and the testing set, respectively. Fig. 4 shows an example of experimental results.

Table I: Experimental results

| | Training (%) | Testing (%) |
|---|---|---|
| 1 | 99.50 | 97.07 |
| 2 | 97.75 | 100.00 |
| 3 | 99.50 | 100.00 |
| 4 | 93.75 | 94.14 |
| 5 | 94.75 | 97.91 |
| 6 | 96.00 | 98.33 |
| 7 | 94.25 | 98.75 |
| 8 | 98.00 | 94.98 |
| 9 | 98.75 | 99.58 |

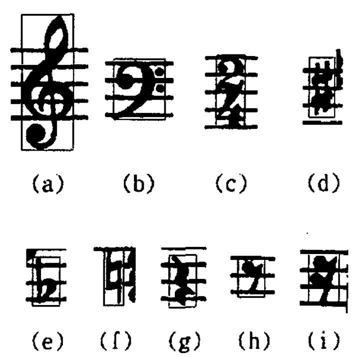


Figure 3: Nine kinds of musical symbols used in the database.

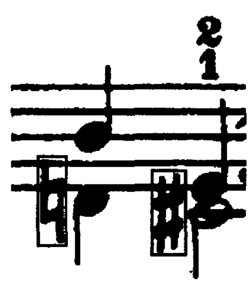

Figure 4: An example of detected musical symbols bounded by a box.

### 4. Conclusion

In this paper we propose an efficient method of recognition of musical symbols without construction of a set of complex If-Then rule based on the musical theory. A very appealing property of the SOM-based fuzzy systems is that the vector quantization feature of the SOM provides us a good supply of clusters and the topology preserving property greatly reduces the computational load.

**Acknowledgement:**

This work is supported by the National Science Council, Taiwan, R.O.C., under the Grant NSC 89-2218-E-008-034.

**References**

[1] D. H. Pruslin, "Automatic recognition of sheet music," SCD dissertation, MIT, 1967.

[2] M. Kassler, An Essay toward the Specification of a Music-reading Machine, pp. 151-75, 1970.

[3] M. Kassler, "Optical character-recognition of printed music: a review of two dissertations," Perspectives of New Music, vol. 11, no. 2, pp. 250-54, 1972.

[4] T. et al. Matsushima, "Automated recognition system for musical Score," Bulletin of Science and Engineering Research Laboratory Waseda University No. 112, pp. 25-52, 1958.

[5] D. S. Perau, "DO-RE-MI: a program that recognizes music notation," Computers and the Humanities, vol. 9, pp. 25-29, 1975.

[6] A. Tojo, and H. Aoyama, "Automatic recognition of music score," Proceedings of the Sixth International Conference on Pattern Recognition, p. 1223, 1982.

[7] E. Sicard, "A efficient method for the recognition of printed music," 11th IAPR Int. Conference on Pattern Recognition, pp. 573-576, 1992.

[8] J. -P. Armand, "Musical score recognition: a hierarchical and recursive approach," 2nd Int. Conference on Document Analysis and Recognition, pp. 906-909, 1993.

[9] Yip-San Wong and Andrew Choi, "A two-level model-based object recognition technique," Int. Sym. On Speech, Image Processing and Neural Networks, pp. 319-322, 1994.

[10] V. Poulain d'Andecy, J. Camillerapp, and I. Leplumey, "Kalman filtering for segment detection: application to music scores analysis," 12 IAPR Int. conference on Pattern Recognition, pp. 301-305, 1994.

[11] K. C. Ng, R D Boyle, and D Cooper, "Low-and high-level approaches to optical music score recognition," IEE Colloquium on Document Image Processing and Multimedia Environments, pp. 311-316, 1995.

[12] S. Baumann, "A simplified attributed graph grammar for high-level music recognition," Proc. the Third Int. Conference on Document Analysis and Recognition, vol. 2, pp.1080-1083, 1995.

[13] B. Couasnon and J. Camillerapp, "A way to separate knowledge from program in structured document analysis: application to optical music recognition," 3 rd Int. Conference on Document Analysis and Recognition, pp. 1092-1097, 1995.

[14] K. Todd Reed and J. R. Parker, "Automatic computer recognition of printed music," Int. Conference Pattern Recognition, pp. 803-807, 1996.

[15] D. Binbridge and T C Bell, "Dealing with superimposed objects in optical music recognition," sixth Int. conference on Image Processing and its Application, pp. 756-760, 1997.

[16] D. Blostein and H. Baird. A critical survey of music image analysis. In Springer-Verlag, editor, Structured Document Image Analysis, pp. 405-434, Eds. H.S. Baird, H. Bunke, K. Yamanoto, 1992.

[17] C. Newell, MidiScan, Product information, Musitek, 1993.

[18] E. Selfridge-Field, "Optical recognition of musical notation: A survey of current work," in Computing in Musicology, Vol.9, pp.109-145, 1994.

[19] H. Kato and S. Inokuchi, "A recognition system for printed piano music using musical knowledge and constraint," in Structured Document Image Analysis, H. Baird, H. Bunke and K. Yamamoto, eds, Spring-verlag, pp. 435-455, 1992.

[20] Hidetoshi Miyao, Yasuaki NAKANO, "Head and stem extraction from printed music score using a neural network approach," pp. 1074-1079, 1995.

[21] I. Yoda, K. Yamamoto, and H. Yamada, "Automatic construction of recognition procedures for musical notes by GA," Proc. of DAS 94, pp. 203-209, 1994.

[22] M. C. Su and C. Y. Tew, "A self-organizing feature-map-based fuzzy system," in IEEE Int. Conference on Neural Networks, vol. 5, pp. 20-25, Italy, 2000.

[23] M. C. Su, T. K. Liu, and H. T. Chang, "An efficient initialization scheme for the self-organizing feature map algorithm," in IEEE Int. Joint Conference on Neural Networks, pp. 1906-1910,Washington, D. C., 1999.