# A Template Matching Distance for Recognition of On-Line Mathematical Symbols

*Fotini Simistira*[1,2]*, Vassilis Katsouros*[1] *and George Carayannis*[1,2]
[1] Institute for Language and Speech Processing of Athena - Research and Innovation Center in Information, Communication and Knowledge Technologies, Athens, Greece
[2] National Technical University of Athens, Greece
{fotini, vsk, gcara}@ilsp.gr

## Abstract

*An on-line handwritten character recognition technique based on a template matching distance is proposed. In this method, the pen-direction features are quantized using the 8-level Freeman chain coding scheme and the dominant points of the stroke are identified using the first difference of the chain code. The distance between two symbols results from the difference of the respective chain codes of the variable speed normalization of dominant points weighted by the local length proportions of the strokes. The proposed technique was tested on two datasets and showed a recognition rate of 92 % in the top 1 choice.*

**Keywords**: on-line character recognition, template matching distance, Freeman chain code, dominant points, variable speed normalization.

## 1. Introduction

In this paper we propose a system that recognizes online handwritten mathematical symbols. The task of designing a mathematical recognizer becomes difficult as the number of symbols that the system has to process becomes large. The database of such a system varies from symbols including Latin and Greek letters and numerals to more specialized mathematical symbols like summation, integral, gradient etc. In addition, mathematical symbols vary in sizes (e.g. the sum operator is very large) and even the same symbol appear in different sizes (e.g. subscripts). Furthermore, there is a great variance in the writing style of each writer. As a result, the task of mathematical symbol recognition involves too many comparisons and becomes time consuming as the system has to process a large number of symbol classes [1].

There has been a significant amount of research in the field of recognition of online mathematical symbols [2, 8] using features based on the Freeman chain code and many more in the field of online handwritten character recognition [1, 4, 7, 9]. For the comparison of chain code sequences there have been proposed approaches that are based on elastic matching techniques [9] or techniques [2] that aim to modify the strokes under consideration so that they are represented by the same number of elements. Both approaches are applied on the chain code sequences resulted from the original number of points of the strokes and have an increased complexity compared to methods that are applied to a reduced features sequence. Approaches that overcome the problem of the complexity involve the use of Hidden Markov Model (HMM) [10] or Neural Networks [6] for the comparison of feature sequences.

In order to reduce the complexity of the recognition process, we propose a representation that is based on the features extracted only from the trajectory of the dominant points of the symbol, i.e. points that include the most important information regarding the shape of the symbol. Before applying the distance measure between two sequences, we apply a merging technique that is equivalent to a type of variable writing speed normalization of the strokes in order to produce sequences of equal number of elements.

We have obtained experimental results comparing the recognition rates between the full features sequences and the reduced features sequences on two datasets. We have concluded that the proposed method increases significantly the throughput by having only a slight effect on the recognition rate.

The organization of the presentation is as follows. In Section 2 we present the feature representation of a symbol. Section 3 deals with the variable speed normalization of the symbols and their comparison. Section 4 discusses the experimental results and in Section 5 we present our conclusions.

## 2. Feature extraction

The input data is handwritten mathematical symbols that are written using the pen on the digitizer of a Tablet PC. Each symbol consists of one or more strokes and for each stroke a series of pen-coordinates through time is stored. Some necessary preprocessing steps are being carried out, such as elimination of the repeated points, smoothing, etc.

Let $S = \left\{ S^m, m = 1,...,M \right\}$ denote an on-line input symbol, where $S^m$ represents the $m$th stroke and $M$ the number of strokes. A stroke is represented as a time-series of points, $S^m = \left\{ p_n^m, n = 1,..., N_m \right\}$, where $p_n^m = (x_n^m, y_n^m)$ are the pen-coordinates of the $n$th point of the stroke and $N_m$ the total number of points of the stroke. The pen-direction feature at the $n$th point of the stroke results from the angle of the slope of the segment between the $n$th and the $(n+1)$th point, i.e.,

$$\theta_n^m = \tan^{-1}\left( \frac{x_{n+1} - x_n}{y_{n+1} - y_n} \right). \tag{1}$$

Using the 8-level Freeman code, the resulted slope angle at the $n$th point is quantized to an integer from 0 to 7. A stroke is now represented as a sequence $S^m = \left\{ c_n^m, n = 1...N_m - 1 \right\}$ of $c_n^m \in \left\{ 0,1,...,7 \right\}$ direction codes.

The dominant points of the stroke are located by finding the elements of the direction code sequence, for which the first difference is a positive number. The first difference of the chain code is defined by counting the code changes between successive elements of the chain code sequence [3]. Let $K_m$<$N_m$ be the number of dominant points of the stroke. Instead of the original sequence of chain codes, we are now focusing on the chain codes of the dominant points (Figure 2).

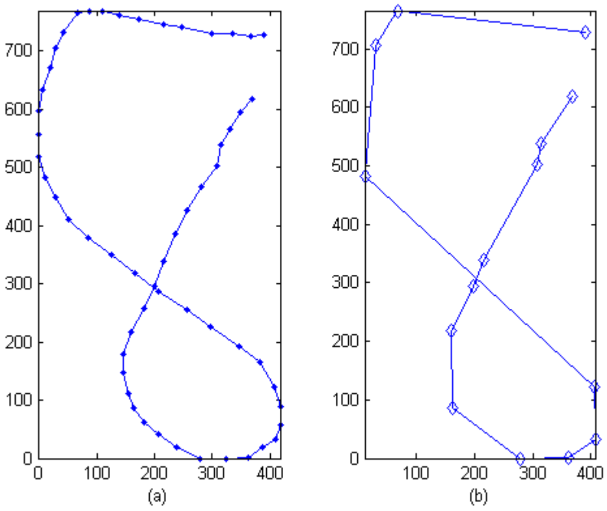

**Figure 1**. (a) Original and (b) dominant points of a symbol.

In addition to the quantized direction features of the dominant points, we also calculate the proportion of the length of the segment between the $k$th and the $(k+1)$th dominant points with respect to the total length of the stroke, as follows:

$$\lambda_k = \frac{\Delta l_k}{\sum_{i=1}^{K_m - 1} \Delta l_i}, \tag{2}$$

where $\Delta l_k$ is the Euclidean distance between the $k$th and the $(k+1)$th dominant points. As a result, a stroke is now represented as a sequence of the 2-tuple of the local direction and the local length proportion between successive dominant points of the stroke, that is, $S^m = \left\{ (c_k^m, \lambda_k^m), n = 1...K_m - 1 \right\}$, with $c_k^m \in \left\{ 0,1,...,7 \right\}$ and $\lambda_k^m \in [0,1]$. This representation of a stroke can be read as "at each dominant point towards which direction and for how long in comparison to the total length of the stroke one has to move in order to reproduce the stroke" (Figure 2).

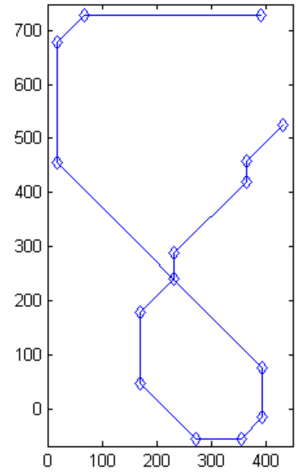

**Figure 2**. Quantized symbol representation using the 8-level Freeman chain code.

## 3. Classification

The classification is based on the nearest neighbor of a template based matching distance. Let us denote by $S_T = \left\{ S_T^m, m = 1,..., M_T \right\}$ the test symbol, where $M_T$ is the number of strokes and $S_T^m = \left\{ (c_k^m, \lambda_k^m), k = 1...K_T^m - 1 \right\}$ the representation of the respective constituent strokes. Similarly, let us denote by $S_R = \left\{ S_R^l, l = 1,..., M_R \right\}$ one of the reference symbols, where $M_R$ is the number of strokes of the reference symbol and $S_R^l = \left\{ (c_k^l, \lambda_k^l), k = 1...K_R^l - 1 \right\}$ the respective constituent strokes. We consider the case where the two symbols have the same number of strokes, i.e. $M_T = M_R = M$. Although the number of strokes is the same, the stroke correspondence is not always the same, since the writing order of the strokes may vary. Let us define a one-to-one correspondence of the strokes by the ordered pairs $\left\{ (m,l),\ m,l = 1,..., M \right\}$ consisted of the $m$th stroke of the tested symbol and the $l^{\text{th}}$ stroke of the reference symbol. In Figure 3a, the vertical stroke of the plus symbol precedes the crossing horizontal stroke, whereas in Figure 3b the vertical stroke follows the crossing horizontal stroke.

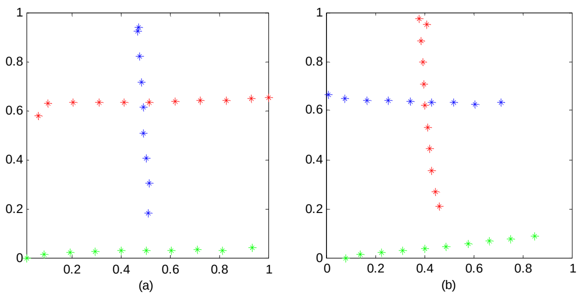

**Figure 3**. Stroke correspondence between test and reference symbol.

The correct stroke correspondence is found by minimizing the sum of the distance between all combinations of stroke pairs of the test and reference symbols. Therefore, the distance between the test and reference symbols is calculated using the following formula

$$D(S_T,S_R)=\min_{\{(m,l)\}}\left(\sum_{m=1}^{M} d\left(S_T^m,S_R^l\right)\right), \tag{3}$$

where $d\left(S_T^m,S_R^l\right)$ is the distance between two strokes. In most of the cases, the two strokes under consideration have a different number of elements, i.e. $K_T^l \neq K_R^m$. For this purpose we apply the following variable speed normalization technique by merging the two sequences. Starting from the sequence of the local relative length proportions, we define two new sequences as follows

$$\left\{\Lambda_k^T=\sum_{i=1}^{k-1}\lambda_i^m, k=1,\dots,K_T^m\right\}, \tag{4}$$

and

$$\left\{\Lambda_k^R=\sum_{i=1}^{k-1}\lambda_i^l, k=1,\dots,K_R^l\right\}, \tag{5}$$

that correspond to the cumulative local length up to the $k^{\text{th}}$ dominant point of the test and reference strokes respectively. Note also that both sequences start with 0 and the end points are equal to one, i.e., $\Lambda^T_{K_T^m}=\Lambda^R_{K_R^l}=1$. We then merge the two sequences defined above into a new sequence $\left\{\Lambda^{T,R}_{r(k)}, r=1,\dots,K_T^m+K_R^l-2\right\}$ by preserving the ascending order of both $\left\{\Lambda_k^T\right\}$ and $\left\{\Lambda_k^R\right\}$. The new sequence will have at most $K_T^m+K_R^l-2$ distinct elements, should the individual sequences coincide only at the start and end points. We re-define the stroke sequences for the test and the reference strokes as follows

$$\hat{S}_T^m=\left\{(z_r^m,\lambda_r^{m,l}), r=1\dots K_T^m+K_R^l-3\right\} \tag{6}$$

and

$$\hat{S}_R^l=\left\{(z_r^l,\lambda_r^{m,l}), r=1\dots K_T^m+K_R^l-3\right\} \tag{7}$$

with

$$\lambda_r^{m,l}=\Lambda_{r+1}^{T,R}-\Lambda_r^{T,R} \tag{8}$$

and

$$z_r^m=\begin{cases} c_k^m, \text{ if } \Lambda_r^{T,R}\in\left\{\Lambda_k^T, k=1,\dots,K_T^m-1\right\} \\ c_{k+1}^m, \text{ if } \Lambda_r^{T,R}\in\left\{\Lambda_k^R, k=1,\dots,K_R^l-1\right\} \end{cases} \tag{9}$$

and

$$z_r^l=\begin{cases} c_k^l, \text{ if } \Lambda_r^{T,R}\in\left\{\Lambda_k^R, k=1,\dots,K_R^l-1\right\} \\ c_{k+1}^l, \text{ if } \Lambda_r^{T,R}\in\left\{\Lambda_k^T, k=1,\dots,K_T^m-1\right\} \end{cases}. \tag{10}$$

The new sequences $\hat{S}_T^m$ and $\hat{S}_R^m$ have the same number of elements and share the same relative normalized local length of the segment between consecutive dominant points.

Let us give an illustrative example of the above technique. Consider the feature sequences of the reference and test symbols as shown in Table 1. The sequences of the cumulative local length proportions $\Lambda_k^R$ and $\Lambda_k^T$ result from the application of Eqs. 4 and 5.

**Table 1**. Feature sequences and cumulative local length proportions of the reference and test symbols.

| Reference Symbol | | | Test symbol | | |
|---|---|---|---|---|---|
| $c_k^R$ | $\lambda_k^R$ (%) | $\Lambda_k^R$ (%) | $c_k^T$ | $\lambda_k^T$ (%) | $\Lambda_k^T$ (%) |
| - | - | 0 | - | - | 0 |
| 3 | 0,63 | 0,63 | 4 | 15,35 | 15,35 |
| 4 | 13,59 | 14,22 | 5 | 3,32 | 18,67 |
| 5 | 2,18 | 16,40 | 6 | 10,60 | 29,27 |
| 6 | 13,50 | 29,90 | 7 | 25,28 | 54,55 |
| 7 | 14,37 | 44,27 | 6 | 4,31 | 58,86 |
| 0 | 1,93 | 46,20 | 5 | 2,68 | 61,54 |
| 7 | 6,77 | 52,97 | 4 | 3,89 | 65,43 |
| 5 | 3,78 | 56,75 | 3 | 6,84 | 72,27 |
| 4 | 9,23 | 65,98 | 2 | 6,21 | 78,48 |
| 3 | 4,54 | 70,52 | 1 | 4,13 | 82,61 |
| 2 | 8,38 | 78,90 | 2 | 2,22 | 84,83 |
| 1 | 14,45 | 93,35 | 1 | 8,87 | 93,70 |
| 2 | 2,97 | 96,32 | 2 | 1,79 | 95,49 |
| 1 | 1,44 | 97,76 | 1 | 4,51 | 100,00 |
| 1 | 2,24 | 100,00 | | | |

Note that the two sequences of the cumulative local length have common the first and the last elements. If we merge the two sequences into a new one by preserving the ascending order, we obtain the sequence $\Lambda_{r(k)}^{T,R}$ that is shown in Table 2, where in bold-face we have denoted the elements that come from the reference symbol. By taking the first difference of the above sequence we obtain from Eq. 8 the merged sequence of local length proportions $\lambda_r^{m,l}$ . Using Eqs. 9 and 10, we obtain the normalized chain codes for the reference and test symbol respectively.

**Table 2**. Normalized feature sequences of the reference and test symbols.

| $\Lambda_{r(k)}^{T,R}$ (%) | $\lambda_r^{m,l}$ (%) | $z_r^l$ | $z_r^m$ |
|---|---|---|---|
| 0,00 | | | |
| **0,63** | **0,63** | **3** | **4** |
| **14,22** | **13,59** | **4** | **4** |
| 15,35 | 1,13 | 5 | 4 |
| **16,40** | **1,05** | **5** | **5** |
| 18,67 | 2,27 | 6 | 5 |
| 29,27 | 10,60 | 6 | 6 |
| **29,90** | **0,63** | **6** | **7** |
| **44,27** | **14,37** | **7** | **7** |
| **46,20** | **1,93** | **0** | **7** |
| **52,97** | **6,77** | **7** | **7** |
| 54,55 | 1,58 | 5 | 7 |
| **56,75** | **2,20** | **5** | **6** |
| 58,86 | 2,11 | 4 | 6 |
| 61,54 | 2,68 | 4 | 5 |
| 65,43 | 3,89 | 4 | 4 |
| **65,98** | **0,55** | **4** | **3** |
| **70,52** | **4,54** | **3** | **3** |
| 72,27 | 1,75 | 2 | 3 |
| 78,48 | 6,21 | 2 | 2 |
| **78,90** | **0,42** | **2** | **1** |
| 82,61 | 3,71 | 1 | 1 |
| 84,83 | 2,22 | 1 | 2 |
| **93,35** | **8,52** | **1** | **1** |
| 93,70 | 0,35 | 2 | 1 |
| 95,49 | 1,79 | 2 | 2 |
| **96,32** | **0,83** | **2** | **1** |
| **97,76** | **1,44** | **1** | **1** |
| **100,00** | **2,24** | **1** | **1** |

An illustration of the interpolation of the elements of the test symbol into the sequence of the reference symbol and vice versa is shown in Figure 4.

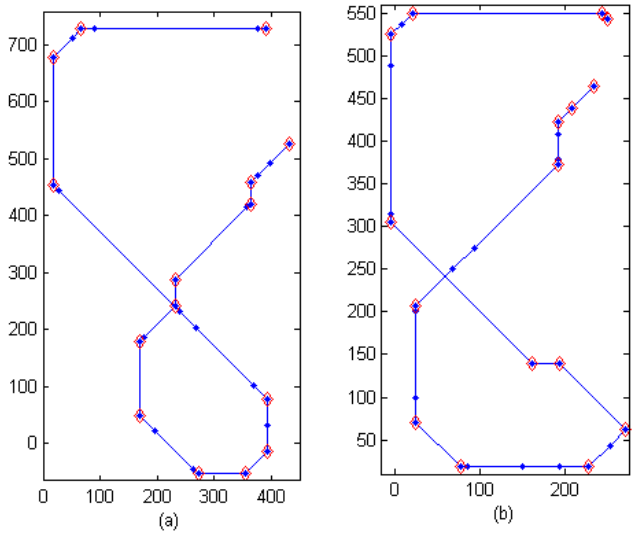

**Figure 4**. (a) Interpolation of reference symbol points (dots) into the test symbol points (diamonds) and (b) interpolation of test symbol points (dots) into the reference symbol (diamonds).

The distance between the strokes is calculated as follows

$$d\left(S_T^m, S_R^l\right) = \sum_{r=1}^{K_T+K_R-3} \lambda_r^{m,l}\left(4-\left|4-|z_r^m - z_r^l|\right|\right). \qquad (11)$$

Note that $0 \le d\left(S_T^m, S_R^l\right) \le 4$ since the difference of the chain codes is normalized to values in the range from 0 up to 4. In the example of Figure 4, the distance between the two symbols is 0.25.

## 4. Experimental Results

Two different online datasets are used for the purpose of this paper. The dataset (LVZE) in [3] consists of 48 distinct symbols written by 11 writers. Each one wrote every symbol 10 times for the training set and 12 times for the test set. The samples are taken with a Tablet PC and are encoded in the UNIPEN format. The 48 symbols of the dataset include, the characters of the Latin alphabet (*a-z*), the numerals (*0-9)*, and mathematical symbols such us the sum, the square root, the integral, the inequality symbols, the arithmetic operators and some special symbols such as the brackets and the parentheses.

The ILSP dataset consists of handwriting samples of 186 distinct symbols written by 50 writers and each one wrote every symbol 5 times. The distinct symbols include 24 uppercase Greek letters (Α-Ω), 25 lowercase Greek letters (α-ω, ς) , 12 uppercase Latin letters (only those that do not appear in the uppercase Greek letters, i.e. C, D, F, G, J, L, Q, R, S, U, V, W), 25 lowercase Latin letters (all except o, which is the same as the Greek omikron), 10 numerals (0-9) and 90 mathematical symbols among of which 13 operators and relational symbols (+, -, <, >, etc.), 5 logical operators (∧, ∨, etc.), 10 set operators (∩, ∪, ⊆, etc.), 5 set symbols ( ℕ, ℤ , etc.), 5 types of arrows (→, ↑, etc.), 24 functions (sin, cos, log, etc.), 5 limit symbols, 2 types of integrals, the summation and the product symbol, 2 symbols from geometry, 4 punctuation symbols, 6 symbols of parentheses, brackets and braces and 7 symbols used as modifiers such us the tilde, the hat, etc. In addition, each writer wrote 54 equations from the wide range of mathematical topics (set theory, mathematical logic, real analysis, geometry, etc.) that involve at least once the 90 mathematical symbols giving us in total 62100 symbols. The samples were collected with a Tablet PC and stored in the UNIPEN format. From the collected symbols, 27900 were used for training and 18600 for testing.

On each dataset we run two experiments. In the first experiment we applied the template matching distance on the full set of the Freeman chain code (FFCC) and in the second on the reduced (only the dominant points) chain code sequence (RFCC). The results of the experiments for the two datasets are summarized in Tables 3 and 4. Comparing the results between FFCC and RFCC, one can observe that the recognition rate in the later has an insignificant decline. In addition, the recognition rates for the LVZE dataset are comparable to the results of [5].

**Table 3**. Recognition rate (%) in top N choice on the LVZE dataset.

| | **Top 1** | **Top 2** | **Top 3** | **Top5** | **Top 10** |
|---|---|---|---|---|---|
| **FFCC** | 92.13 | 95.15 | 96.58 | 97.91 | 98.94 |
| **RFCC** | 92.00 | 95.49 | 96.85 | 97.82 | 98.81 |

**Table 4**. Recognition rate (%) in top N choices on the ILSP dataset.

| | **Top 1** | **Top 2** | **Top 3** | **Top5** | **Top 10** |
|---|---|---|---|---|---|
| **FFCC** | 92.35 | 95.15 | 96.63 | 97.94 | 98.97 |
| **RFCC** | 92.21 | 95.44 | 96.95 | 97.89 | 98.84 |

The standard deviation of the recognition rates with respect to the writers is 1.85 for the Top 1 choice of the FFCC and 2.18 for the Top 1 choice of the RFCC on the LVZE dataset. In Figure 5 we show the variation in the recognition rate for the writers of the LVZE test set.

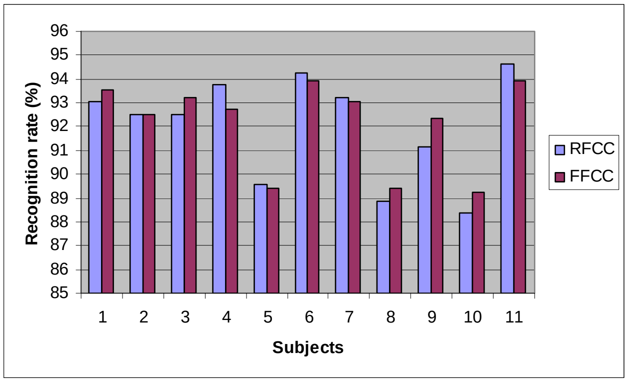

**Figure 5**. Variation of recognition rates of FFCC and RFCC among the writers of the LVZE dataset.

The RFCC sequence has much fewer elements than the FFCC sequence. In particular, the average number of elements in the LVZE dataset for a symbol with the RFCC representation is 10.79, which is 4.4 times less than the respective number of elements, that is, 47.53, for the FFCC representation. At the same time, the reduction of the recognition accuracy does not exceed the 0.15%. Figure 6 shows the average number of elements using the FFCC and the RFCC representations for each symbol class of the LVZE dataset.

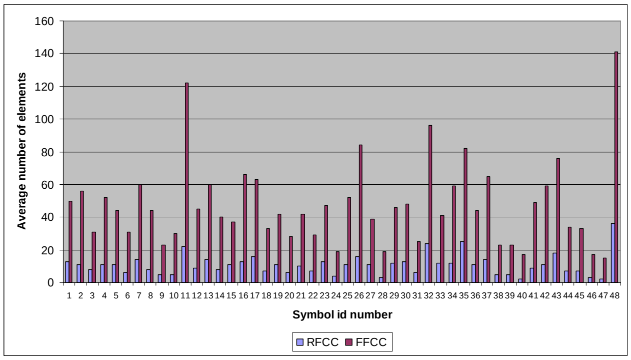

**Figure 6**. Average number of points of FFCC and RFCC for the symbols of the LVZE dataset.

Another experiment was conducted on the symbols extracted from the mathematical equations of the ILSP dataset. The overall recognition rate of the RFCC on this dataset was 94.18%. However, it must be noted that the instances of each distinct varied from symbol to symbol, e.g. the equal sign appears almost in every equation, whereas the integral symbol appears in two equations.

## 5. Conclusions

In this paper, we propose a template matching method for mathematical symbol recognition. The symbol's strokes are represented as a sequence of reduced Freeman chain codes of the dominant points of the trajectory, together with the respective local length proportions. Before comparison, the strokes are normalized with respect to a variable writing speed by multiplexing the elements of the individual feature sequences in order to produce test and reference sequences of equal size. The distance used weighs the difference of the individual chain codes by the local length proportions of the normalized sequences.

## Acknowledgements

This work was funded by the Greek Secretariat for Research and Technology and the European Community under the research program SAMS-EHG-29.

## References

[1] S.D. Connell and A.K. Jain, "Template-based on-line character recognition", Pattern Recognition, 20, 2001, pp. 1-14.

[2] U. Garain, B.B. Chaudhuri, "Recognition of Online Handwritten Mathematical Expressions", *IEEE Transactions on Systems, Man, and Cybernetics-Part B,* Vol. 34, No. 6, 2004, pp. 2366-2376**.**

[3] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Prentice Hall, 2002.

[4] N. Joshi, G. Sita, A.G. Ramakrishnan, S. Madhvanath, "Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition', *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, Tokyo, Japan, 2004, pp. 444-449.

[5] J. J. LaViola, R.C. Zeleznik, "A practical Approach for Writer-Depended Symbol Recognition Using a Writer-Independent Symbol Recognizer", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2007, Vol.29, pp. 1917-1926.

[6] M. Schenkel, I. Guyon, D. Henderson, "On-Line Cursive Script Recognition using Time Delay Neural Networks and Hidden Markov Models", *Proceedings of the International. Conference on Acoustics, Speech and Signal Processing (ICASSP-94)*, Vol. 2, 1994, pp. 637-640.

[7] J. Shin, "On-Line Handwriting Character Recognition Using Stroke Information", *LECTURE NOTES IN COMPUTER SCIENCE*, Springer-Verlag, pp. 703-714, 2002.

[8] T. Suzuki, S. Aoshima, K. Mori, Y. Suenaga "A New System for the Real-time Recognition of Handwritten Mathematical Formulas", *Proceedings of the 15th International Conference on Pattern Recognition,* 2000, pp. 515-518.

[9] S. Uchida, H. Sakoe, "A survey of Elastic Matching Techniques for Handwritten Character Recognition", *IEICE Transactions on Information and Systems,* Vol E88-D, No. 8, 2005, pp. 1781-1790.

[10] H.J. Winkler, "HMM-based Handwritten Symbol Recognition Using On-line and Off-line Features", *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP-96)*, Vol. 6, 1996, pp. 3438-3441.