A New Feature Extraction Technique for On-Line Recognition of Handwritten Alphanumeric Characters

Basabi Chakraborty [a], Goutam Chakraborty [a]

[a] *Faculty of Software and Information Science, Iwate Prefectural University, 152-52 Aza Sugo, Takizawamura, Iwate 020-0193, Japan*

Abstract

In this work a new method of feature extraction for an interactive and adaptive recognizer for on-line handwritten alphanumeric characters has been proposed. The system is suitable for use in conjunction with magnetic pen based devices for inputting data to a data processing system or a computer terminal. The features are extracted from dynamically changing locations of the writing device. The new method of feature extraction is simple, computationally light and fast enough for adaptive on-line use. Extracted features are robust with respect to all possible distortions like shape, size, and orientation. For simulation experiment, numerals 0 to 9 are used. A single hidden layer feed forward neural network trained by *Quickprop* algorithm, a variation of error back propagation is used for recognition. Very high recognition rates, even for highly distorted samples have been achieved confirming high generalization capability of the extracted feature set.

Key words: On-line recognition, Handwritten character recognition, Feature extraction, Artificial neural network, Quickprop learning algorithm.

1 Introduction

With the technological advancement of sophisticated magnetic pen based devices used as a more compact and comfortable input interface than keyboard, on-line handwritten character recognition is becoming an active research area of practical interest. Improvement of on-line handwriting recognition technique is considered a necessary step to provide hand written interface to different small portable machines (where keyboard is too big to attach) as well as to computers for a more natural way of communication. On-line recognition refers to the recognition mode in which the machine recognizes instantaneously while the user writes using some special writing device e.g. magnetic pen on some writing tablet. The user can easily detect and correct misrecognized character and a close interaction between the user and the machine is possible. This is extremely important in commercial development of personal digital assistants (PDA), digital cellular phones or real time signature verifiers. The advantage of on-line recognition is that the dynamic or temporal information of the pen movement can be captured in contrast to the static images in case of off-line recognition. This information leads to better and easier recognition algorithms.

Numerous methods and approaches for on-line character recognition problem have been proposed till to date and they have already summarized in a few exhaustive survey papers [1], [2], [3]. On-line or real-time character recognition consists of the following main steps.

- Pre processing of the data
- Feature extraction
- Classification

For designing real time reliable recognizer, we need fast algorithm for feature extraction as well as classification. Extracted features must possess high discriminatory characteristics for better recognition with simpler classification algorithms. There are variety of problems that need on-line hand writing recognition with different standard and set of requirements. Some of the examples are pen based computers or hand held devices, signature verifiers and various development tools. Depending on the objective of the individual problem there are different algorithmic approaches which in turn make use of different set of discriminatory features. Many alphanumeric character recognition algorithms have been developed to use geometrical or topological and statistical features based on character shape which provides the most discriminatory information. Feature extraction is the most crucial step for fast and efficient recognition. In case of real time hand-written character recognition it is easy to capture the movement of the writing device with respect to time. Recently researchers are trying to use features from the dynamic information collected from the pen-tip movement [4]. The pen trajectories on a digitizing surface are generally

normalized and local dynamical features such as number, direction and order of the strokes, dominant points etc. are extracted.

In the classification stage lots of statistical approaches from curve matching [5] to markov modelling [6] have been developed. Artificial neural networks, a class of computationally simple adaptive trainable system, are now becoming popular tools for pattern recognition problems. A number of neural network models [7], [8], [9] have been developed for off-line handwriting recognition, where the recognition is done after the handwriting has been completed and no dynamic information is available for recognition process. Generally the scanning of the image is done and either pixel values or some static shape features extracted from the images are used as the input information. Le Cun et al. [10] [11] have also developed neural network recognizers for on-line recognition where dynamic informations in time domain as well as the pictorial informations are used for fast recognition. Guyon et al. [12] used a restricted neural network architecture for on-line character recognition where sequential as well as local geometric informations from the pen trajectory has been utilized.

In this work we focus on extracting efficient features from the time sequential signal generated by the pen movement on writing pad for reliable real time recognition of alphanumeric characters required for better man machine interface for personalized use. We propose a method for feature extraction which is simple and computationally light. The extracted features are robust with respect to different distortions like size, shape, orientation and noise present in the sample and therefore have a high generalization capability. For recognition, we have used a single hidden layer feed forward artificial neural network (ANN) with *Quickprop* [14] , a variation of error back propagation, as the learning algorithm. Though simple, the proposed alphanumeric character recognition algorithm is fast and adaptive enough to act in real time with moderately high recognition rate in an interactive mode. The performance has been tested with a generated set of hand written data.

Moreover, to assess the efficiency of the proposed features for on-line character recognition, we have compared our method with other similar reported methods. Among the reported works on on-line character recognition utilizing time sequential signals we have selected the work of Guyon [12] and Li [5] for comparison. The same set of generated handwritten data has been used for simulation.

The next three sections describe the main three steps of the on-line character recognition algorithm, preprocessing (collection) of the data followed by our proposed feature extraction algorithm followed by classification. Section 5 describes simulation experiments and their results. The following section represents the results of comparison with other similar reported methods. Final

section contains discussion and conclusion.

2 Preprocessing of the data

The characters are inputted by mouse or written on writing pads by magnetic pen. The sequential positions of the writing device i.e. the dynamic positions are noted. The sequence of the position coordinates of the writing device are used for generating the data. Thus here we get information for not only the position of online pixels, but also their sequential order of generation. The time sequential signal from the pen movement on the writing pad, collected by sensor attached to the pad, is described as a sequence of consecutive points on the $x - y$ plane. Thus μth sample of ith character class C_i is expressed by the time sequence of co-ordinates as:

$$C_i^\mu = \{(x_{i0}^\mu, y_{i0}^\mu), (x_{i1}^\mu, y_{i1}^\mu), \ldots, (x_{ik}^\mu, y_{ik}^\mu), \ldots, (x_{in}^\mu, y_{in}^\mu)\} \tag{1}$$

Figure 1 represents the generation of the data for a sample of numeric character class 2 in which (x_0, y_0) and (x_n, y_n) denote the initial and the final point of the generated time sequence of co-ordinates for the particular sample.

The sampling rate of the signal is considered fixed for all the samples for all the classes of characters. Thus the number of points n in the series of co-ordinates for a particular sample is not fixed and depends on the time taken to write the sample on the pad. As the number of points n in actual trace of the characters are generally large and varies greatly due to high variation in writing speed, a fixed lesser number p of points, regularly spaced in time are selected for further processing depending on the particular problem.

The i th point from the series of n points is selected according to the following equation

$$i = i \times (n/p) \tag{2}$$

The value of i is successively taken as $i = 0, 1, 2, \ldots, p$, and right hand side of the Equation 1 is rounded to the nearest integer value to get all the p successive selected points.

Thus the time sequential signal for μth sample of any character class C for further processing is expressed by the simple time sequence of co-ordinates as:

$$C^\mu = \{(x_0^\mu, y_0^\mu), (x_1^\mu, y_1^\mu), \ldots, (x_k^\mu, y_k^\mu), \ldots, (x_p^\mu, y_p^\mu)\} \tag{3}$$

The subscript i used in Equation 1 to denote sample from a particular character class C_i is also dropped in Equation 3 to avoid complicacy in representation and from now on we will write x_k^μ and y_k^μ to denote x_{ik}^μ and y_{ik}^μ respectively .

3 Feature Extraction

The method of feature extraction proposed here from the pen movement on the writing pad exploits the sequential and dynamical information present for on-line recognition. The idea and algorithm is described below.

3.1 Idea and Algorithm for Feature Extraction

The idea proposed for feature extraction is at present applicable only for real-time inputs. This is because it uses the time dependent positions of the writing device, though the correlation of these sequential data is not explicitly exploited in the classification stage at present.

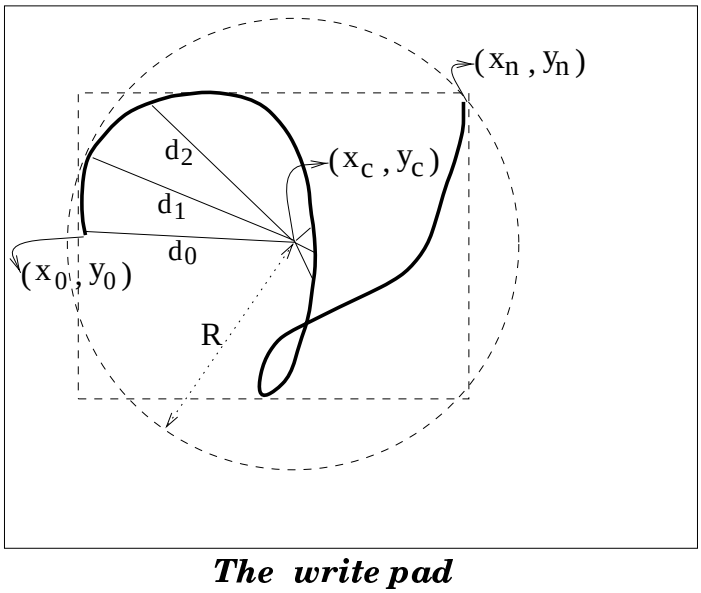

Fig. 1. Feature extraction from a sample of character class 2

Now to make the features independent of size and orientation, a modified sequence is generated from the original sequence according to the following algorithm which is also shown in Figure 1.

(1) The minimum rectangular area that cover μth sample of a particular

character class is found from x^{μ}_{min}, x^{μ}_{max} and y^{μ}_{min}, y^{μ}_{max} where

$x^{\mu}_{min} =$ minimum of all x^{μ}_{k}s, $x^{\mu}_{max} =$ maximum of all x^{μ}_{k}s

$y^{\mu}_{min} =$ minimum of all y^{μ}_{k}s, $y^{\mu}_{max} =$ maximum of all y^{μ}_{k}s.

(2) The centre of the rectangular area $(x^{\mu}_{c},\ y^{\mu}_{c})$ and the radius of the smallest circle R^{μ} that could enclose the μ^{th} sample character is found as shown in Figure 1.

(3) A new sequence is created from the sequence given in Equation 3 in terms of the distances of the points

$(x^{\mu}_{0},\ y^{\mu}_{0});\ (x^{\mu}_{1},\ y^{\mu}_{1}); \ldots,\ (x^{\mu}_{k},\ y^{\mu}_{k}); \ldots,\ (x^{\mu}_{p},\ y^{\mu}_{p})$

from the center $(x^{\mu}_{c},\ y^{\mu}_{c})$ and normalized with respect to the corresponding radius R^{μ}. So, we have a sequence of distances from the center $(x^{\mu}_{c}, y^{\mu}_{c})$ as,

$$D^{\mu} = \{d^{\mu}_{0},\ d^{\mu}_{1}, \ldots,\ d^{\mu}_{k}, \ldots,\ d^{\mu}_{p}\} \tag{4}$$

where

$$d^{\mu}_{k} = \frac{\text{distance between } (x^{\mu}_{k},\ y^{\mu}_{k}) \text{ and } (x^{\mu}_{c},\ y^{\mu}_{c})}{R^{\mu}} \tag{5}$$

Thus this distance feature (i.e. the D^{μ} vector) becomes invariant of the size and orientation of the sample character.

(4) Similarly the sequence of angles to capture the angular movement of the writing device is created as shown in Figure 1. For this the reference line has been taken as the line joining $(x^{\mu}_{0}, y^{\mu}_{0})$ (the first point in the sequence given in Equation 1) and the centre $(x^{\mu}_{c}, y^{\mu}_{c})$. The succession of angles for the rest of the points in the sequence is found as the angles between the line joining the point $(x^{\mu}_{k}, y^{\mu}_{k})$ and the center $(x^{\mu}_{c}, y^{\mu}_{c})$ and the reference line. The sequence of angles thus generated is,

$$\Theta^{\mu} = \{\ \theta^{\mu}_{0},\ \theta^{\mu}_{1},\ \theta^{\mu}_{2}, \ldots,\ \theta^{\mu}_{k}, \ldots,\ \theta^{\mu}_{p}\} \tag{6}$$

Here θ^{μ}_{k} is the angle between the lines joining the pair of points $\{(x^{\mu}_{c}, y^{\mu}_{c}), (x^{\mu}_{0}, y^{\mu}_{0})\}$ and $\{(x^{\mu}_{c}, y^{\mu}_{c}), (x^{\mu}_{k}, y^{\mu}_{k})\}$ and θ^{μ}_{0} is 0 . This feature is also invariant to the size and orientation of the sample character.

Now the sequences of signals represented in Equation 4 and Equation 6 are used as the features for classification of different characters.

In addition to using time sequences D^{μ} and Θ^{μ} for μth sample of a particular character class presented in Equation 4 and Equation 6 as the feature vectors, we also used Fourier Transform of those time sequences as two other feature vectors for inputting to the next stage of recognition process i.e. to the neural network classifier. Let Fourier Transforms of D^{μ} and Θ^{μ} are represented as DF^{μ} and ΘF^{μ}.

4 Character Classification

We have used artificial neural network(ANN) for the classification stage of the characters as they have become the most popular classifier for pattern recognition problems, mainly due to following reasons.

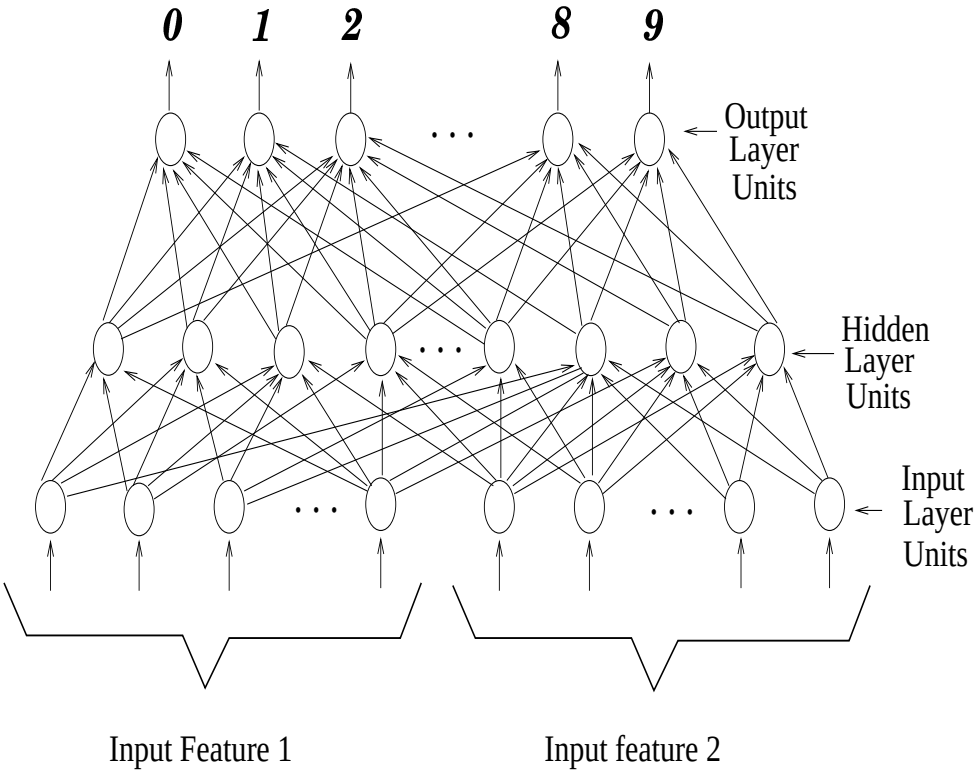

Fig. 2. Artificial Neural Network Model used for classification

(1) Learning of the neural network can be simply done from the samples.
(2) Once learning is complete, recognition is done very fast.
(3) They have good generalization capability.

Though ANN's are proved to be efficient classifiers, the proper selection of their architecture still remains a hard task and seems dependant on the particular problem. Here we have used multilayer feedforward perceptron (MLP), the most popular architecture for classification task, for the classification of alphanumeric characters with the proposed extracted features. The structure of the neural network used is shown in Figure. 2. The extracted feature vectors for μth sample character D^{μ}, Θ^{μ} pair or DF^{μ}, ΘF^{μ} pair are fed to the units of the input layer as the input feature 1 and input feature 2. The output units correspond to the class of the recognized characters and thus the number of neurons in the output layer is taken same as the number of classes. The number of neurons in the input layer is taken as $2 \times p$, double the number of points in the time sequences.

At the time of training, the training samples for a particular class are fed to the input and the actual outputs are noted. The sum of errors calculated as the deviation of the actual outputs from the ideal ones is back propagated to train the network with the objective of getting actual outputs reaching their ideal values as close as possible. After training, the network is tested with unseen samples and the samples are classified according to the output with highest value.

In between this input and output layer, one hidden layer is used. Nodes are connected only in the forward direction i.e. from input units to hidden units and from hidden units to output units as shown in Figure 2. We used a fully connected network, where an upper layer node is connected to all the lower layer nodes (though all connections are not shown in the Figure 2). The connection weight between the i^{th} node in the upper layer from the j^{th} node in the lower layer is denoted as w_{ij}. The input to the i^{th} node is the summation of activations from nodes of the previous layer multiplied by the corresponding connection weights i.e. $\Sigma_j w_{ij} \times a_j$, where a_j is the activation of the j^{th} node. A sigmoidal like activation function produces the output from the hidden and output nodes.

Usually error back propagation [13] learning rule is used to train MLPs. To speed up training we used here *Quickprop* training algorithm [14] . *Quickprop* uses information about the curvature of the error surface. This requires the computation of the second order derivatives of the error function during training. *Quickprop* assumes the error surface, a function of connection weights, to be locally quadratic (i.e. a parabola) and attempts to jump in one step from the current position directly into the minimum of the parabola. *Quickprop* computes the derivatives in the direction of each weight. After computing the first gradient as in regular backpropagation, a direct step to the error minimum is attempted by changing the weight as

$$\Delta w_{ij}(t+1) = \frac{S(t+1)}{S(t) - S(t+1)} \Delta w_{ij}(t)$$

where $\Delta w_{ij}(t+1)$ is actual weight change at $(t+1)$ th step, S(t+1) and S(t) are the partial derivatives of the error function with respect to w_{ij} at $(t+1)$ th and previous (t th) step.

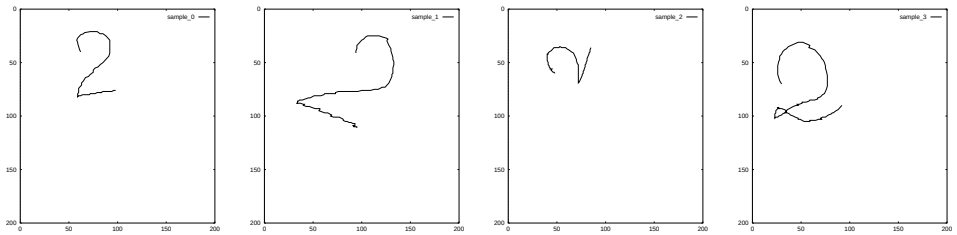

Fig. 3. Sample images of numeral "2"

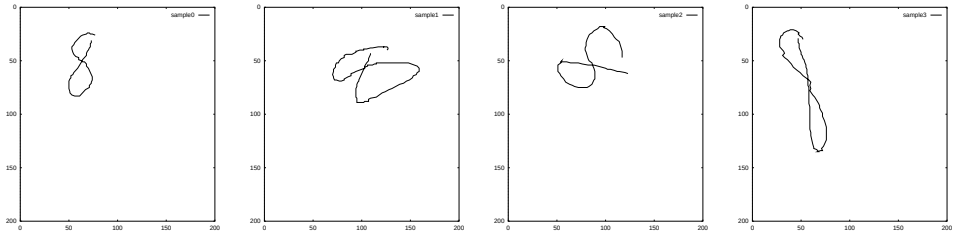

Fig. 4. Sample images of numeral "8"

5 Simulation and Results

5.1 Data Generation

For our present simulation we have used only numerals though alphabets can also be used with our algorithm. C_i^μ sequences as expressed by Equation 1 for digits $i = 0, 1, \ldots, 9$ are generated from pentip movements as explained before. The numerals were written by different writers and they are of different shape, size and orientation. A total of 2000 samples (200 samples for each digit) were generated. Four typical samples each for digit "2" and "8" are shown in Figure 3 and Figure 4 respectively to show the amount of distortions present in our sample set. The modified sequences, D^μ and Θ^μ, have been generated according to the procedure described in the previous section. The $D^\mu(t)$ and $\Theta^\mu(t)$ of the four digit "2" samples of Figure 3 and digit "8" of Figure 4 are shown in Figure 5. The corresponding Fourier transforms DF^μ and ΘF^μ were computed and used as the extracted features for on line digit recognition. DF^μ, ΘF^μ for the same samples of digit "2" and digit "8" are shown in Figure 6.

5.2 Simulation

We simulated our ANN using SNNS simulation package with the features from generated data as described above. To keep the size of the neural network small we used only 21 points, selected from the original sequence of Equation 1 with the help of Equation 2 to generate the sequence given in Equation 3 with value

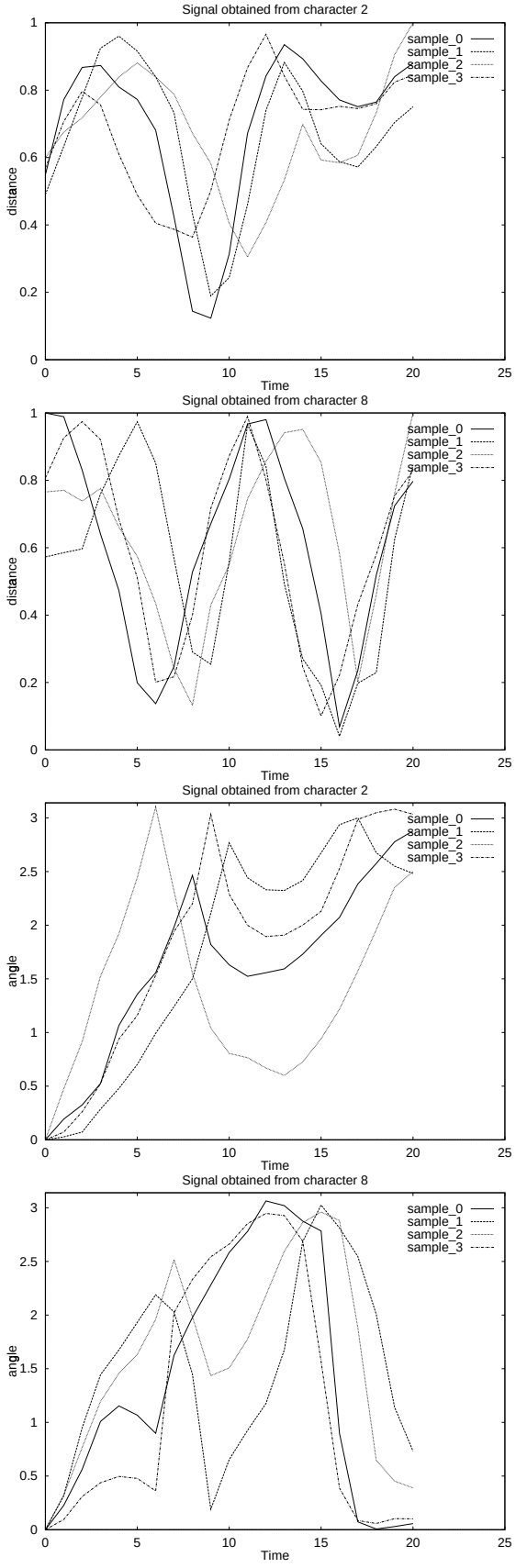

Fig. 5. Four typical examples of D^{μ} and Θ^{μ} for characters "2" and "8"

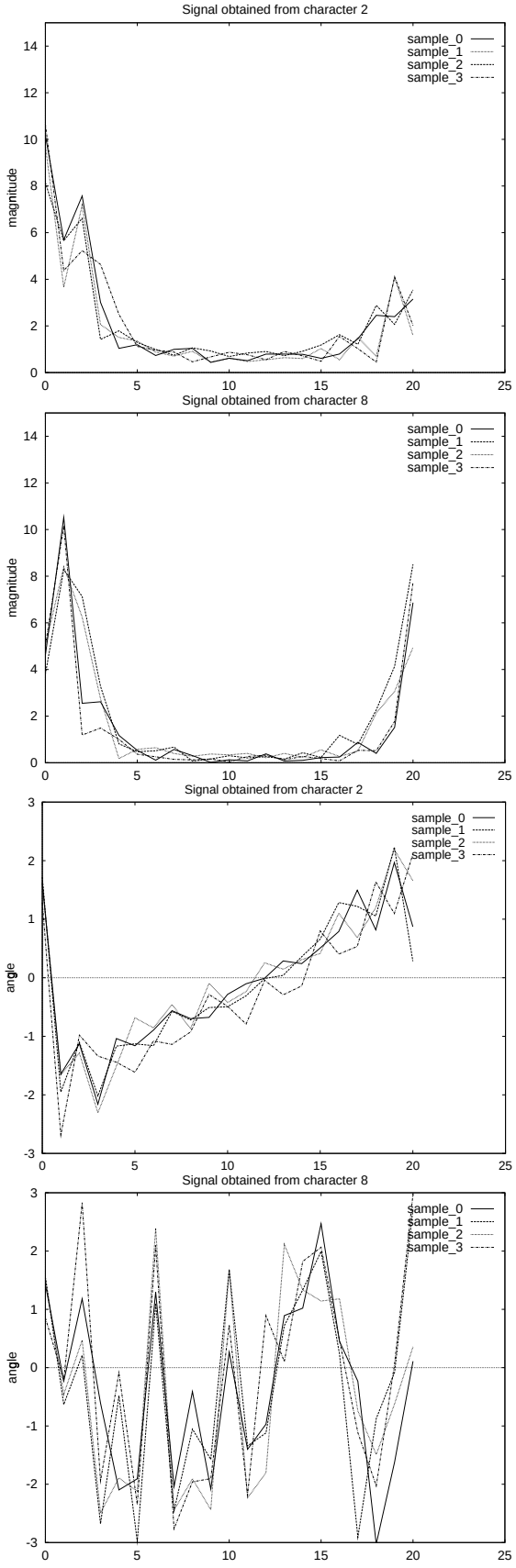

Fig. 6. Four typical examples of DF^{μ} and ΘF^{μ} for characters "2" and "8"

of $p = 21$.

Thus number of nodes in the input layer of the neural network is taken as 42 while the number of nodes in output layer is taken as 10 corresponding to the ten numerals. Experiments with different network sizes i.e. with varying number of hidden layer neurons have been done. Initially large number of hidden layer neurons are taken and are gradually decreased. The whole sample set is divided into *training set* with 100 samples and *test set* with 100 samples respectively for each digit. The training set of 100 samples is further divided into actual *training set* with 75 samples and *validation set* with 25 samples for each digit. Various networks with same number of neurons in the hidden layer with different initial weights are trained with the actual training samples for all the classes for a pre-assigned minimum allowable output error and tested for validation error with the validation set. The network having the smallest error with respect to the validation set is used for further processing. In the second phase, the number of hidden layer units are decreased and the training procedure is repeated with actual training set and validated by validation set for minimum validation error for determining the optimum number of hidden layer neurons. The final selected network is used for testing recognition rate with the test set.

5.3 Recognition Results

The optimum number of hidden layer neurons has been chosen as 26. The results reported here are for the optimum trained network with 26 hidden units corresponding to the minimum validation error. Table 1 represents the simulation results in terms of recognition rate of the optimum trained network both with the traning sample (actual training and validation sample) and the test samples (unused during training). with two sets of extracted features D^{μ},Θ^{μ} (from the modified sequence) and DF^{μ} , ΘF^{μ} (from the fourier transform of the modified sequence)).

We used both the feature sets as the input to the NN classifier for all the digits. After training the network , we used both sets of samples, the set used for training and the set kept aside for testing, for checking the performance of the network. As is obvious, the classification result with the training samples, with the network is very high and is almost 100%. With the unseen samples set aside for testing, the result reflects the good generalization capability of the extracted features. It is seen from the table that both the classification and the generalization efficiency is better when fourier transforms of the original extracted features are used as input features than the extracted features themselves.

Table 1
Recognition rate with training and testing data and different extracted features

digit	Recognition rate in percentage			
	Using D(t)and Θ(t)		Using DF(t) and ΘF(t)	
	on Training data	on Testing data	on Training data	on Testing data
0	100.0	96.4	100.0	100.0
1	96.3	84.5	100.0	95.8
2	92.7	74.5	96.4	88.2
3	96.3	84.5	100.0	92.1
4	92.1	76.5	100.0	96.5
5	100.0	92.3	99.8	96,5
6	100.0	88.2	100.0	92.3
7	96.4	84.5	100.0	94.6
8	100.0	100.0	100.0	100.0
9	96.4	80.2	96.1	88.1

6 Comparative Performance Evaluation

Among the earlier works on online character recognition from the time sequential signals obtained from pen trajectory, the works of Guyon et, al. [12] and Li and Young [5] are the most relevant to our works in connection to feature extraction. Guyon et, al. used position coordinates of the pen trajectory after simple preprocessing as the features for on-line digit recognition with a feedforward layered network similar to time delay neural network. Li and Young extracted dominant points in strokes and the writing direction between consecutive dominat points from the time sequential signal of the pen trajectory. They then used the sequence of the dominant points and the directional information as the features for the recognition of handwritten characters. For classification, they used dynamic programming matching using the idea of bandlimited time warping.

In both the cases the extracted features as well as the the classification methods are different from our proposed work. In our experiment we have used the features reported in their works with a simple multilayer perceptron classifier trained with quick propagation algorithm for classification for the same data set and compared the results with the classification results we got with our proposed features. The simulation results are shown in the next section.

For simulation we used same samples, each sample is described by Equation 1. For extracting features according to Guyon's method the samples are resampled to obtain regularly spaced points and then rescaled to remove scale distortions. The direction and the curvature of the trajectory are also found out at each point and are added to the feature set.

Feature extraction according to Li's method involves finding out the dominant points in the trajectory and direction primitives between dominant points. The original sample signal represented by Equation 1 is used to find out the features as described in Li's paper. Here the feature set includes the sequence of dominant points and the sequence of direction primitives.

To allow effective comparison with our proposed features, in both the cases (Guyon's and Li's) a sequence of 21 points is used to represent a particular character. We used only numerals in our experiment. Multilayer feed forward perceptron of same structure (42 neurons in the input layer, 10 neurons in the output layer with one hidden layer of 26 neurons) trained with quick propagation used for classification with our extracted features has been used here. As described in sec 5.2, the total sample is divided into training and testing sample and the training set is further divided to actual training and validation set. The network is trained with quick propagation algorithm with actual training set and validated with the validation set for minimum error. The experiment is repeated for several times with different initial set up of the network. The optimum trained network is used for testing the network with test samples.

Table 2 represents the recognition rate obtained with the both the training set (actual training set and the validation set) and the testing set. Simulation results of Table 2 shows that Li's features are better than Guyon's features when classified with our MLP network.. Comparison with the recognition rates represented by Table 1 shows that our proposed features perform better. Though classification accuracy depends heavily on classifier type, it is reflected from our experiment that in the same environment with a simple classifier, our proposed features are more efficient than the features reported in other works. Analysis for the feature extraction reveals that the computational burden of extracting our proposed features are less than that of other features.

Table 2
Recognition rates with other set of features

	Recognition rate in percentage			
	Using Guyon's features		Using Li's Features	
digit	on Training data	on Testing data	on Training data	on Testing data
0	97.1	89.3	100.0	100.0
1	96.2	86.4	98.4	96.4
2	94.4	80.2	95.4	90.5
3	92.3	82.5	99.8	90.5
4	90.8	74.6	98.4	94.3
5	98.1	84.3	100.0	92.1
6	98.1	86.2	100.0	92.1
7	96.2	85.8	100.0	93.4
8	98.3	92.2	98.4	94.6
9	96.6	82.1	94.6	82.2

7 Conclusion

In this work, we have proposed a new idea for feature extraction, which is simple, computationally light, reasonably fast and suitable for designing on-line interactive and adaptive handwritten character recognition system to be used in conjunction with magnetic pen based devices for inputing data to a computer terminal or other data processing systems. Even with very distorted characters and very small training sample set, using our feature extraction technique we achieved a high degree of recognition accuracy for unseen test samples. This shows the high level of generalization capability of the proposed feature set. Our algorithm is fast enough with moderately high recognition rate perfectly suitable for designing an adaptive interactive system where the system needs the feedback capability to improve its performance such as personal digital assistants meant for personalized use.

However, there are still many areas for improvement for our algorithm. The dots that trace the character tends to concentrate more where the second derivative of the trace is larger. Also it depends on the writer and his own style. We sampled points at regular intervals of count (see Equation (5)). If the points were sampled along the trace at equal distances it would be more free from noise involved due to individual styles, though the process would cost more computation. In fact, when the ultimate aim is to personalize the handwriting recognizer for an individual, or to use for tasks as signature

identification, our approach would give better result. We would like to carry out more experiments in this regard.

Our experiments are done with a small sample set and only on numeric characters. To assesss the actual usefulness of our features in case of practical online recognition problems we need to study a far more larger sample set collected from real life applications. Another obvious area of improvement is to increase the number of inputs of the neural network classifier i,e the size of the network which will allow us to take more points per sample for better representation which we plan to do with bigger sample set. Further, in designing our algorithm we have not taken into account the inherent correlation of the sequential time signal, a possible source of information for improving efficiency of the recognition system. We would like to extend our study in this direction in near future. Though we need further experimentation with more complex problems like on-line signature verification, our proposed new features extraction algorithm has been proved significant in the limited experiment we have reported in this paper.

References

[1] Tappert, C. C.et al.,1990. The State of the Art in On-Line Handwriting recognition, *IEEE Tran. PAMI* 12(8), 787–808.

[2] Wakahara, T. et al., 1992. On-Line Handwriting Recognition, *Proc. IEEE*, 80(7),1181-1194.

[3] Plamondon, R. et al.,1999. On-Line Handwriting Recognition, *Encyclopedia of Electrical and Electronics Eng.*, 15, 123–146.

[4] Nouboud, F. and Plamondon, R. 1990. On-line Recognition of Handprinted Characters: Survey and Beta Tests, *Pattern Recognition*, 25(9), 1031–1044.

[5] Li,X. and Yeung, D. Y., 1997. On-Line Handwritten Alphanumeric Character Recognition Using Dominant Points in Strokes, *Pattern Recognition*, 30(1), 31–44.

[6] Anquetil, E. and Lorette, G.,1995. On-Line Cursive Handwritten Character Recognition Using Hidden Markov Models, *Traitement du Signal*, 12(6), 575–583.

[7] Le Cun, Y. et al.,1992. Handwritten Digit Recognition with a Back-Propagation Network, *Neural Networks, Current applications*, Chapman and Hall.

[8] Kelly, D. A., 1992. Neural networks for handwriting recognition, *Proceedings of the SPIE- The International Society for Optical Engineering*, 1709, 143–154.

[9] Fukushima, K. and Wake, N., 1991. Handwritten alphanumeric character recognition by the neocognitron, *IEEE Trans. on NN*, 2, 355–365.

[10] LeCun, Y. et al.,1993. On-Line handwriting recognition with neural networks: spatial representation versus temporal representation, *Proc. International Conference on handwriting and drawing* Ecole Nationale Superieure des Telecommunications.

[11] Bengio, Y. et al.,1995. Lerec: A NN/HMM hybrid for on-line handwriting recognition, *Neural Computation*, 7(6),1289-1303.

[12] Guyon, I. et al.,1991. P. Albrecht, Design of a neural network character recognizer for a touch terminal, *Pattern Recognition*, 24(2),105-119.

[13] Rumelhart, D. E.,et al. 1986. Learning internal representations by error propagation,*Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol 1, pp.318–362, MIT Press.

[14] Fahlman, S. E, 1989. Faster learning variations on back propagations: an empirical study, *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufman.