A Study of Representations for Pen based Handwriting Recognition of Tamil Characters

C. S. Sundaresan
Computer Science Automation
Indian Institute of Science
Bangalore 560012, India
E-mail:css@csa.iisc.ernet.in

S. S. Keerthi
Computer Science Automation
Indian Institute of Science
Bangalore 560012, India
E-mail:ssk@csa.iisc.ernet.in

Abstract

In this paper we study the important issue of choosing representations that are suitable for recognizing pen based handwriting of characters in Tamil, a language of India. Four different choices, based on the following set of features are considered: (1) a sequence of directions and curvature; (2) a sequence of angles; (3) Fourier transform coefficients; and (4) wavelet features. We provide arguments in support of the representation using wavelet features. A neural network designed using these features gives excellent accuracy for recognizing Tamil characters.

Keywords: *Character representations, OLCR, Neural networks, Pen computing, Wavelets*

1. Introduction

The design of better human-computer interfaces has become an important area of Computer Science. Speech recognition and on-line handwriting recognition are two important and promising approaches towards better human-computer interfaces because they provide more efficient and natural communication between human and machine. In an on-Line Character Recognition (OLCR) system the machine recognizes the character written by the user. The user writes on a special device called *graphic tablet* which captures the handwriting of the user.

OLCR is attractive for languages with large number of characters, since it is necessary to type two or more keys in an English key board to input one character in such a language. In this paper we develop an on-line character recognizer for an Indian language called Tamil. Tamil characters are difficult to recognize. This is because there are more characters and there are many characters which look very similar. i.e., two or more Tamil characters look very similar and the difference is minute. Tamil characters ஒ, ஓ, ச், ச, சு, க, க், த், த, ந, ந், கி, சி, கீ, சீ, ஐ, ஜ, எ, ஏ, ை, ன,ள are some examples. Moreover these characters will be written in the same way and they differ only at the end of the writing.

In this paper we focus on preprocessing and feature extraction. For Tamil character recognition we started our experiments with a character representation which was used previously for recognizing digits and uppercase English characters [1]. After analyzing the character in time and frequency domains and doing experiments we found that wavelet feature representation actually simplifies the classifier and and significantly improves the classification accuracy of the system.

2. Preprocessing

In this section we discuss about the preprocessing steps necessary for on-line character recognition. The main purpose of preprocessing is to reduce the unnecessary variabilities present in the raw signal (x-y coordinate sequence) obtained from the graphic tablet. The user may write any character in any size. Hence there will be lot of difference between the x-y coordinate sequences of same character but of different size. We can eliminate this variability due to size, by resizing the character. Size normalizer transforms the handwritten character so that it fits into a standard size box. These raw x-y coordinates obtained from the graphic tablet is equidistant in time.The number of x-y coordinates for a character will vary if the writing speed is varied. Re-sampling converts a sequence, which is equidistant in time to a sequence which is equidistant on the x-y plane. Special care has to be taken while re-sampling the characters with pen lifts; After size normalization and re-sampling the character will have standard size. This sequence will still have a small amount of noise introduced by size normalization and re-sampling. To reduce the noise we employ a moving average filter.

3. Feature extraction

The preprocessed x-y coordinates can be used as the input to the neural network. But this is not generally preferred because to get good classification accuracy the classifier has to be more complex and needs more data. Feature extraction or representation of on-line handwritten characters is very important in recognizing the characters written, since good features selected for representation will simplify the recognizer. In the following sections we will discuss about three different representations used for doing experiments with on-line Tamil character recognition.

3.1. Sequence of directions and curvature

To begin, we used the features used for on-line recognition of digits and upper case English letters. For more description about the recognizer see [1]. In this representation each point on the character is represented by seven features. The seven features are penup(n), $\tilde{x}(n)$, $\tilde{y}(n)$, $\cos\theta(n)$, $\sin\theta(n)$, $\cos\phi(n)$, $\sin\phi(n)$, where: penup(n) tells the status of the pen, i.e., whether the pen is touching the tablet or not; $\tilde{x}(n)$, $\tilde{y}(n)$ are preprocessed x-y coordinates; $\theta(n)$ represents the local direction of the curve at n^{th} point; and $\phi(n)$ represents the local curvature of the curve at n^{th} point. Here direction of a stroke is encoded by calculating the direction cosines of the tangent to the curve at point n. These parameters can also be thought of as discrete approximations to the first derivatives with respect to the arc length, $\frac{dx}{ds}$ and $\frac{dy}{ds}$ where $ds = \sqrt{dx^2 + dy^2}$. The local curvature is measured by calculating the angle between the two elementary segments. This angle is encoded by its cosine and sine.

One important drawback in using the sequence of directions and curvature features is its high dimension. If we have n points after re-sampling, the character will be represented by $(7 * n)$ numbers. Because of its high dimension the neural network requires large number of hidden layers and connections and hence we require large amount of training patterns for good generalization.

3.2. Sequence of cosine of angles

In this representation the cosine of the angle between two consecutive lines formed by three consecutive points in the character was calculated. This angle information is sufficient for recognizing the handwritten characters since we can reconstruct the character with the help of this angle alone. This is possible because in preprocessing we have taken the distance between any two consecutive points to be constant.

The sequence of angle features looks appropriate for representing the hand written character. But if the x-y coordinates of a character are close and noisy it will not give good results since presence of small variation will cause big difference in the extracted feature. At the same time if we reduce the number of x-y coordinates while re-sampling we will loose information (corners) about the character which will lead to poor classification accuracy.

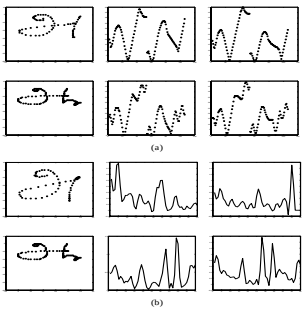

Figure 1. Wavelet and angle feature representation.

3.3. Wavelet features

The basic idea behind this representation is that, many times the given signal makes more sense in the frequency domain than in the time domain. In our case the signal is the sequence of x-y coordinates and we can use Fourier transform to get the frequency domain representation of the signal. The problem here is that the frequency domain information obtained by Fourier coefficients depend on the entire time sequence. Hence small change in the time domain sequence in a small interval of time will be spread out over the entire frequency domain. This is undesirable because, there are many characters which differ among themselves in a small portion of the time sequence. Example: ன, ண, ணை, னை, னி, னீ, ணி, ணீ. For these characters Fourier coefficients will not vary much, which makes the classification problem tough.

3.3.1. Analysis of characters in Fourier domain

We used Fourier transform for analyzing the handwritten characters. Figure 2 gives an analysis of a Tamil characters in Fourier domain. In this figure the preprocessed character is shown at the top. Below that the x and y coordinate sequences of that character are plotted separately. Then we show the frequency components present in the coordinate sequences, separately. The approximated frequency components of the character is shown next i.e., high frequency components are made zero. Next the x-y coordinate sequence which is reconstructed from the approximated frequency components is shown. The plot in the last row of

the figure shows the character without any high frequency components. If we plot the same for different characters, we can observe that inter class variance for frequency domain signals is poor. i.e., we cannot easily distinguish two characters belonging to different classes on seeing the frequency domain information. This is due to the fact that each frequency domain information obtained by Fourier coefficients depend on the entire time sequence. The calculation of each frequency component involves the entire time sequence. Hence small change in the time domain sequence in a small interval of time will be spread over entire frequency domain which leads to small inter class variance.

From the FFTs of x-y coordinate sequences we can observe that low frequency components are prominent. Here low frequency components reflect the basic shape of the character and high frequency components reflect the finer details. This can be verified by making all the amplitudes of high frequency components zero and reconstruct the character from the approximated FFTs, since the basic shape of the reconstructed character is same as the original one, we can say that the low frequency Fourier coefficients represents the basic shape. See figure 2. Moreover low frequency Fourier coefficients are less sensitive to the writing styles, since basic shape will not change much with different writing styles or with different users.

3.3.2. Wavelet transforms for OLCR

As we already mentioned, if we consider each character as a sequence of x-y coordinate sequence, we can use wavelet transform to get wavelet features [2] which can be used for representing the handwritten character. In our case each character has 50 x-y coordinates after re-sampling. We considered this sequence of x-y coordinates as two time sequences. One is $\mathbf{X}$ which is the x sequence of the original x-y coordinate sequence and the other is $\mathbf{Y}$ which is the y sequence of the original x-y coordinate sequence. We applied wavelet transform on $\mathbf{X}$ and $\mathbf{Y}$ separately. For our experiments we used only the first approximation of the original sequence. To get the first approximation of $\mathbf{X}$ and $\mathbf{Y}$ we convolved these sequences with a set of discrete filter coefficients. We used second order Daubechies filter coefficients to get $\mathbf{W_x}$ and $\mathbf{W_y}$ and used them as the input for the neural network. We will discuss results on the application of wavelet transform to OLCR in section 5.

4. Comparison of features

In this section we compare the sequence of cosine of angle feature with wavelet features. The angle between two consecutive lines formed by two consecutive points in the character is calculated as described in section 3.2. We believe that this angle information is sufficient for recognizing the handwritten characters since we can reconstruct the character with the help of this angle alone. This is possible because after preprocessing the distance between any two consecutive points is constant. Even though this feature has good qualities like *compactness* and *rotation invariance*, it did not give good results with neural networks. To find out the reason for its bad performance, we plotted the cosine of angle against the index (feature vector). See figure 1. In figure 1(b) the first column shows 2 original characters and the figures in each row except the first one, shows the angle feature plotted against the index for two different instances of the same character. We selected two examples (per class) and plotted the features for both characters but shown only one original character in the first column. We can see high variations in features of characters that belong to the same class. i.e., there is lot of difference among the figures in the same row. These variations are the cause for the poor performance of the networks when the sequence of angle feature is used. As already mentioned these variations are due to the presence of noise in the x-y coordinate sequence, but unfortunately presence of small amount of noise will lead to a big variation in the extracted feature. This is because the x-y coordinates are very close. But if increase the distance between successive points i.e., reduce the number of points in a character (by re-sampling) then we may loose some useful informations like corners present in the character.

As we explained in section 3.3 we extracted wavelet features from the preprocessed x-y coordinate sequence. Since wavelet features gave very good performance with neural networks, we were interested in comparing the wavelet features with the angle feature. Hence we plotted wavelet features also similarly. See figure 1(a). From these two figures we observe that there is not much variation in features of characters that belong to same class. Moreover there is lot of variation between the features of characters that belong to different class.

5. Experimental results and discussion

In this section we will discuss about the results of the experiments conducted with the character representations for on-line Tamil character recognition problem. Tamil language has 247 alphabets with 135 distinct symbols. To study the problem closely, initially we considered only 12 Tamil characters for recognition. We conducted many experiments with the representations discussed in section 3 and found that wavelet features are most suitable for on-line Tamil character recognition problem. After finding a good character representation (wavelet features) and neural network architecture, we focussed on the complete set of Tamil characters. Then we developed a system for recognizing all Tamil characters. From the experimental results

it was found that, among all representations wavelet features perform the best. Below we briefly describe our experiments. Full details with analysis will be presented in a later paper.

To start with, we used the representation for recognizing digits and upper case English letters, as suggested by Guyon et. al [1], for our problem. The preprocessing was done as described in section 2. We used TDNN [3] for classification. The performance of TDNN for Tamil character recognition was not good. The recognition accuracy was only 84.95 %. As already mentioned earlier, the main reasons for TDNN's poor performance are: (a) Tamil characters are similar, (b) High input dimension and (c) Presence of large number of weights in the network

Next we used the sequence of angle features for representing the Tamil characters. We used the same preprocessing technique as described in section 2. We used a single hidden layer network for classification. The performance of this system was also poor. As we explained in section 4, the main reason for the poor performance is that the presence of noise in the x-y coordinates, which affects the extracted features greatly.

After analyzing the characters in the Fourier domain, we concluded that wavelet features will be more appropriate for representing the Tamil characters. Again we used the same preprocessing technique as described in section 2. Wavelet features were extracted as described in section 3.3.2. A Single hidden layer network was used for classification. The network gave excellent performance. The classification accuracy is 96.54 % for 12 character problem and 94.30 % for 135 character problem. In the case of 12 character problem, there are 50 examples per class for training purpose and 200 examples for testing purpose and for 135 characters problem there are 40 examples per class for training and 20 examples for testing.

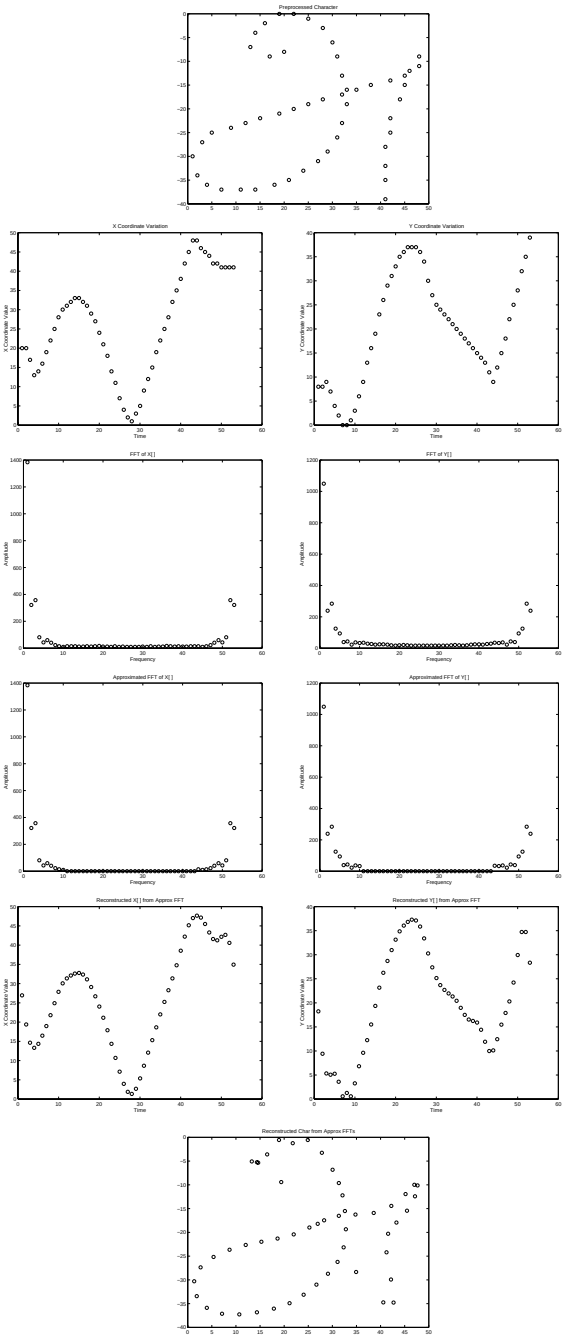

Figure 2. Analysis of character

References

[1]. I. Guyon, P. Albrecht, Y. Le Cun, J. Denker, and W. Hubbard. Design of a neural network character recognizer for a touch terminal. *Pattern recognition*, 1991.

[2]. P. Wunsch and F.Laine. Wavelet descriptors for multiresolution recognition of handprinted characters. *Pattern Recognition, Vol. 28, No. 8*, 1995.

[3]. A.Waibel, T.Hanazawa, G.Hinton, K.Shikano, and K.Lang. Phoneme recognition using time delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, March 1989.