One Dimensional Representation Of Two Dimensional Information For HMM Based Handwritten Recognition

Nafiz Arica
Dept. of Computer Engineering,
Middle East Technical University,
Ankara,Turkey
nafiz@ceng.metu.edu.tr

Fatos. T. Yarman-Vural
Dept. of Computer Engineering,
Middle East Technical University,
Ankara,Turkey
vural@ceng.metu.edu.tr

Abstract

In this study, we introduce a new set of one-dimensional discrete, constant length features to represent two dimensional shape information for HMM (Hidden Markov Model), based handwritten optical character recognition problem. The proposed feature set embeds the two dimensional information into a sequence of one-dimensional codes, selected from a code book. It provides a consistent normalization among distinct classes of shapes, which is very convenient for HMM based shape recognition schemes. The new feature set is used in a handwritten optical character recognition scheme, where a sequence of segmentation and recognition stages is employed. The normalization parameters, which maximize the recognition rate, are dynamically estimated in the training stage of HMM. The proposed character recognition system is tested on both a locally generated cursively handwritten data and isolated number digits of NIST database. The experimental results indicate high recognition rates.

1. Introduction

Hidden Markov Model (HMM) is a very powerful tool for modeling and recognition problems of one-dimensional signals. However, the extension of the HMM into two-dimensional image processing applications are not as successful as the one-dimensional cases. This is, basically, because of the requirement of large number of parameters to large amount of training data. On the other hand, the use of the one-dimensional HMM in the two-dimensional problems requires embedding the two dimensional information in the one-dimensional representation. In these applications, the major limitation is the difficulty of expressing the neighborhood relation in the observation sequences. Therefore, the power of the Markovian property is not effectively reflected in the neighborhood relations.

Optical character recognition of cursive handwriting is an attractive field of the shape recognition problems, which is intensively studied in the last decade and yet has still a long way to achieve the final goal of fluent machine reading. Among many others, the most popular character recognition systems are the HMM based recognizers. There are many studies, which employ hundreds of features on various types of HMM, for solving the handwritten recognition problem. However, none of the representations are sufficient to express full characteristics of handwriting. A good source of references in recent developments in hand-written character recognition by HMM can be found in [1] [2] and [3].

This study proposes a one-dimensional normalized representation of the image grid, which is very convenient for HMM based handwriting recognition. The proposed method extracts a set of directional sparse skeletons of the binarized character shapes by scanning the image grid in various directions and extracting the skeletons of the digital image in each direction. Then, the directional skeletons are appended one after the other. Finally, the coordinates of the skeleton pixels are coded by the assigned code of the regions where the sparse skeleton pixels lie.

In section 2, a summary is given for the overall handwritten recognition system of this study. After an initial segmentation of the connected characters, a normalization process reduces the character size variations and provides a meaningful comparison platform for observation probabilities of HMM, in section 3. The proposed feature extraction scheme effectively embeds the two-dimensional information

0-8186-8821-1/98 $10.00 © 1998 IEEE

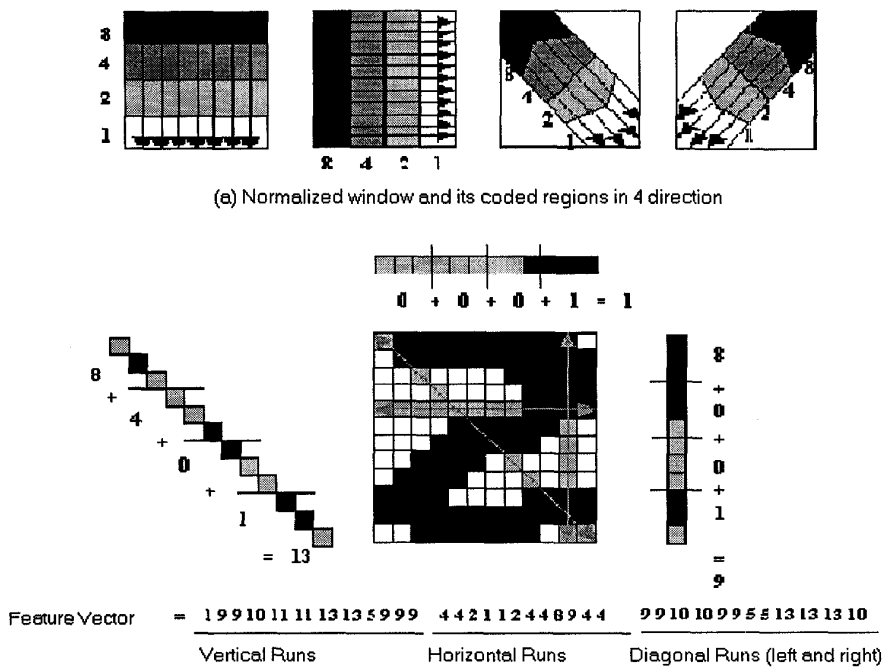


Figure 1. Feature Extraction Process

into a one-dimensional string representation, in section 4. Section 5 explains the HMM recognition of candidate characters. In section 6, the candidate characters obtained from the HMM recognition are resolved in a final stage of recognition based segmentation, where the HMM probabilities are used as a new set of features in the arcs of a graph. Finally, in section 7, experiments are performed on a local database of cursive handwriting and on the number digits of the NIST database. It is observed that, the features proposed in this study, is insensitive to noise. Very high recognition rates are achieved, even for considerable variations in writing styles and sizes.

2. System Overview

The overall system is a complete scheme for off-line handwritten connected character recognition, which employs a sequence of segmentation and recognition algorithms. The system employs the gray scale and binary information in a mixed way to extract the maximal amount of information for both segmentation and recognition stages. A two-step segmentation algorithm, proposed by the authors of this paper [5], finds the nonlinear character segmentation paths in free style handwriting. This is a search process, applied to the initial segmentation regions on the gray scale image for determining the imperfect segmentation boundaries between characters.

Then, the recognition is accomplished in two stages: In the first stage, a new set of features is extracted from the segmented regions, which mostly corresponds to a character. The features are fed to a left-right Hidden Markov Model. Training stage of HMM involves dynamical adjustments of the parameters of the normalized feature set. Finally, a recognition based segmentation algorithm resolves handwriting strings by maximizing a cumulative information measure, computed from the HMM probabilities.

3. Normalization

The characters, extracted from the segmentation algorithm, are normalized to a fixed size window. Then each normalized window is divided into equal regions in selected directions. The size of the window, the number of scanning directions and the number of regions in each

scanning direction are the normalization parameters. Initially, the parameters are taken as variables. The normalization process is not only for making the system size invariant, but also, for making the recognition probabilities of the characters comparable. The normalized gray scale segment is binarized for feature extraction.

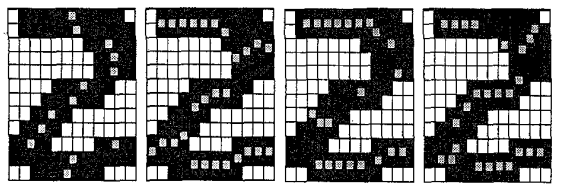
The normalized window is scanned in various directions. The number of directions depends on the window size. For relatively small size, only the horizontal and vertical directions are considered. As the size of the window is increased, two or more diagonal directions are included into the scanning procedure. The scan lines in all directions are divided into equal regions. The number of the regions also, depends on the size of the normalized window. Small windows are divided into two regions. As the size of the window grows, more regions are required to represent each character.

The size of the window, the number of scanning directions and the number of regions are estimated in the training stage of the HMM, together with the window size. Once, the size of the normalized window and the corresponding scanning directions and regions are estimated, it is fixed for the recognition stages. Figure 1, indicates a binarized and normalized sample character, obtained as the output of the normalization stage. For this particular training set, the optimal normalized window size is 12x12 pixels, with four-directional scan lines, each of which is divided into four regions. Each region is coded by the powers of 2 as indicated in Figure 1.a.

4. Feature Extraction

For feature extraction, the medians of the black runs are identified, in each scan line. The code of the region, wherein the median of that run is located, represents that particular run. Each scan line is, then, represented by the sum of the codes of the regions where the medians of the runs are located (Figure 1.b). The size of the normalized window and the number of the regions of the scan lines are selected in such a way that there exists, at most, one run in each region. For the example of Figure 1, the summation of the codes generates integers between 0 and 15 (8+4+2+1). A character candidate is represented by a concatenated integer string of length 36 (12 values for columns and 12 values for rows, 6 values for each diagonal, as depicted in Figure 1).

In the above feature extraction method, the median in each scanning direction represents a sparse directional skeleton of the character. It is well known that a skeleton of a character preserves almost all the shape information of a character. The directional skeletons, proposed in this study are much more convenient than the classical skeletons for representing the shape information of the characters in HMM based recognition schemes: First of all, concatenated directional skeletons preserves almost all the skeleton information of the characters. By increasing the number of scanning directions, it is possible to obtain curse to fine representation of the complicated skeletons. More importantly, in the proposed directional skeleton representation, there are no cross points and branch points as in the classical skeletons. These qualitative features form inconvenient bases for HMM recognizers. In the proposed directional skeleton representation there are no such points (see Figure 2).



(a) (b) (c) (d)

Figure 2. Directional Skeletons obtained from a) vertical, b) horizontal, c) left and d) right diagonal scanning

5. The HMM Training and Recognition

The discrete HMM for each digit can be represented by $\lambda_l = \{\Delta, \Theta, S, T\}$, where $\Delta = \{\delta_{ij}\}$ is state transition probability matrix for transition from state i to state j, $\Theta = \{\theta_i(k)\}$ is the observation probability matrix of observing the code k in i^{th} state for $1 \le i, j \le S$, $i \le j$, S is the number of states in the model, T is the number of distinct code symbols per state, $k \in V$ and $V = \{0,\ldots\ldots,M\}$ is the set of codes obtained in the feature set [6]. For the example of Figure 1, M=15, and T=36.

A training set is formed by using the correctly segmented characters. This set is used to estimate the model parameters, as well as, the three normalization parameters. For this purpose, a tree is formed with the nodes in the first level representing various sizes of the window. The nodes in the second level represent various numbers of scanning directions for each window size and the third level represents the number of regions for each number of scanning directions. Then, the problem is to find the optimum path from the root to the leaf, which maximizes the recognition rate. In other word, for

each path, Δ and Θ parameters of HMM model for every character class is estimated via the Baum-Welch method [6]. Then, the characters in the training set are recognized via forward-backward algorithm [6]. The path, which gives the maximum recognition rate for the training set, is taken as the optimum path. During this process, the number of states of the HMM model is increased proportionally with the size of the normalized window, from S=10 to S=30. The normalization parameters are fixed for the recognition stage.

In the recognition stage, the probability of observing the unknown character by every HMM is calculated. Then, the observed string $O = \{o_1,\ldots,o_T\}$ is labeled with the class which maximizes the probability $P(O/\lambda_l)$. If the preliminary segmentation algorithm yields a false partition, then the HMM classifier is forced to make an incorrect assignment for that particular segment. However, this assignment, mostly, has a low observation probability. Because; the training set does not contain a similar sample.

6. Recognition Based Segmentation

Candidate character segmentation paths, which are found in the segmentation stage, are not always the correct character boundaries. In the final stage, the candidate character segmentation paths and log $P(O|\lambda_l)$ are used as new features to resolve the recognition results. The HMM recognition results, and log $P(O|\lambda_l)$ measure are represented as arcs and segmentation points are represented as nodes, in a graph. The optimal character segmentation paths can be confirmed by maximizing the cumulative log $P(O|\lambda_l)$ measure, searching the longest path in the graph (Figure 3).

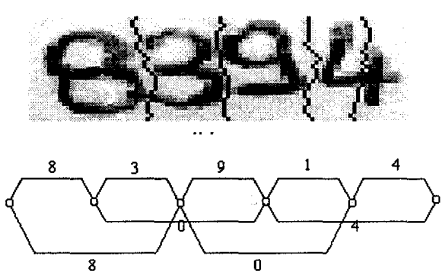


Figure 3. Recognition Based Segmentation

7. Results

The experiments are performed in UNIX environment, under C programming language. The proposed scheme is, first tested on a local database with 4000 handwritten connected digits, collected from 20 persons. For each digit, 50 samples are used in the training stage. In the experiments, the normalization parameters are estimated by gradually increasing the parameters and calculating the corresponding recognition rates.

The smallest window size is taken as 10x10 and is increased until 22x44. The increments are, also, applied in the length-to-width ratios of the windows. At each squared size, length-to-width ratio is started from 1 and gradually increased to 2. It is interesting to note that the optimal length-to-width ratio is a function of the window size and ranges between 1.2-1.7, for this particular database. The recognition rates are small for the square shapes and gradually increases until length-to-width ratio reaches to the optimal value. Then, it decreases again. These local maxima keeps increasing at each optimal length-to-width ratio, as the window size increases, until the window size reaches to 20x26. Then, its starts decreasing. In the experiments, the number of states is increased from 10 to 22, proportionally with the window size. Table 1 indicates the results of recognition for the local database with two directional (vertical and horizontal) runs.

The proposed scheme is, also, tested on 10000 isolated digits randomly selected from NIST Special Database 1 and Special Database 7. In the training stage, 100 sample different from test set are used for each digit. Due to relatively large size of the NIST database, compared to our local database, it was impractical to work with a larger size of the training sets in the laboratory environment. This yielded relatively smaller recognition rates as it is seen from Table 2. The optimal window size is 20x30, with 4 scanning directions and 5 regions in each direction. After obtaining the optimal normalization parameters for the NIST database the training stage is redone with 1000 samples for each digit. For this case the recognition rate reaches 97% for the optimal parameter set.

8. References:

[1] S. Connel, "A Comparison of Hidden Markov Model Features for the Recognition of Cursive Handwriting", Comp. Science Dept., Michigan State Univ., MS Thesis (1996)

[2] A. Atici, F.T. Yarman-Vural, " A Heuristic Method for Arabic Character Recognition, Journ. of Signal Processing, vol. 62, issue.1, 1997.

[3] N. Arica, F. T. Yarman-Vural "HMM Based Handwriting Recognition", Proc. of the 12nd Int. Symposium on Computer and Information Sciences, 242-247 (1997).

[4] M.A. Özdil, F.T. Yarman,Vural, "Optical Character Recognition Without Segmentation", Proc of the Int Conf. On Document Analysis and Recognition Germany, 483-487, (1997).

[5] N. Arica, F. T. Yarman-Vural, " A New Scheme for Off-Line Handwritten Connected Digit Recognition" ICPR'98 (accepted).

[6] B.H.Juang and L.R.Rabiner, (1990), "The Segmental K-means Algorithm for Estimating Parameters of Hidden Markov Models" , IEEE Trans. on A.S.S.P., Vol:38, No:9.

Table 1. Experimental Results for Local Database

Window Size	*# of States*	*Recognition Rate (%)*	*Window Size*	*# of States*	*Recognition Rate (%)*
10x10	10	95.1	18x18	18	97.7
10x12	10	96.2	18x20	18	98.0
10x14	10	96.9	18x22	18	98.3
10x16	10	**97.0**	18x24	18	**98.4**
10x18	10	96.5	18x26	18	96.8
10x20	10	95.9	18x28	18	97.9
12x12	12	95.5	18x30	18	97.0
12x14	12	96.5	18x32	18	97.8
12x16	12	97.3	18x34	18	95.7
12x18	12	97.3	18x36	18	96.8
12x20	12	**97.7**	20x20	20	98.3
12x22	12	97.6	20x22	20	97.8
12x24	12	95.8	20x24	20	98.5
14x14	14	96.3	20x26	20	**98.6**
14x16	14	97.0	20x28	20	97.2
14x18	14	97.0	20x30	20	97.0
14x20	14	**97.9**	20x32	20	98.2
14x22	14	97.0	20x34	20	96.9
14x24	14	97.3	20x36	20	96.7
14x26	14	96.3	20x38	20	96.6
14x28	14	96.6	20x40	20	96.2
16x16	16	97.3	22x22	22	96.5
16x18	16	97.7	22x24	22	98.3
16x20	16	97.8	22x26	22	**98.5**
16x22	16	98.0	22x28	22	97.4
16x24	16	97.9	22x30	22	97.2
16x26	16	97.6	22x32	22	97.8
16x28	16	**98.2**	22x34	22	97.6
16x30	16	96.4	22x36	22	97.7
16x32	16	97.4	22x38	22	96.3
			22x40	22	97.3
			22x42	22	95.4
			22x44	22	96.6

Table 2. Experimental Results for NIST Database (with 4 directional runs)

Window Size	*# of States*	*# of Regions*	*Recognition Rate (%)*	*Window Size*	*# of States*	*# of Regions*	*Recognition Rate (%)*
16x16	24	4	91.6	20x20	30	5	92.7
16x20	24	4	92.4	20x25	30	5	94.3
16x24	24	4	93.1	20x30	30	5	94.8
16x28	24	4	93.0	20x35	30	5	94.4
16x32	24	4	92.5	20x40	30	5	93.8
20x20	20	4	92.2	18x18	30	6	92.5
20x24	20	4	92.4	18x24	30	6	94.4
20x28	20	4	92.9	18x32	30	6	94.2
20x32	20	4	93.2	18x36	30	6	94.0
20x36	20	4	93.1				
20x40	20	4	92.8				