

Comparison of elastic matching algorithms for on-line Tamil handwriting recognition

Niranjan Joshi, G Sita, A G Ramakrishnan
and Sriganesh Madhvanath

Dept. of Electrical Engg., Indian Institute of Science, Bangalore, India
HP Labs India, Koramangala, Bangalore, India
e-mail: joshi,sita,ramkiag@ee.iisc.ernet.in,
srig@hp.com

Corresponding Author:

G. Sita
Department of Electrical Engg.
Indian Institute of Science
Bangalore - 560 012, INDIA.
E-mail: sita@ee.iisc.ernet.in

Comparison of elastic matching algorithms for on-line Tamil handwriting recognition

Niranjan Joshi, G Sita, A G Ramakrishnan¹, Sriganesh Madhvanath²

¹ Dept. of Electrical Engg., Indian Institute of Science, Bangalore, India

² HP Labs India, Koramangala, Bangalore, India

e-mail: {joshi,sita,ramkiag}@ee.iisc.ernet.in,

srig@hp.com

Abstract. We present a comparison of elastic matching schemes for writer dependent on-line handwriting recognition of isolated Tamil characters. Three different features are considered namely, preprocessed x-y co-ordinates, quantized slope values, and dominant point co-ordinates. Seven schemes based on the three features and dynamic time warping distance measure are compared. Comparison is done with respect to recognition accuracy, recognition speed, and number of training templates. Along with these results, possible grouping strategies and error analysis is also presented in brief.

1 Introduction

On-line handwriting recognition means machine recognition of the writing as user writes on a sensitive screen. Here the writing is stored as a sequence of points as against an image in case of off-line handwriting recognition. For a good review of on-line handwriting recognition, see [1, 2]. On-line handwriting recognition is especially very relevant in Indian Scenario, because it eliminates the need to learn to any complex key stroke sequences and handwriting input is faster compared to any other text input mechanism for Indian languages. Given the complexity of entering the Indian scripts using a keyboard, handwriting recognition has the potential to simplify and thereby revolutionize data entry for Indian languages.

The challenges posed by Indian languages are different from English as not only the script size is very large but also the closeness between some of the characters call for sophisticated algorithms. In addition, unlike English and other oriental scripts, Indian scripts have compound symbols resulting from vowel-consonant combinations and in many cases, consonant-consonant combinations. Hence, the recognition strategy to be adopted must take into account the size of the basic character set and the permitted compound symbols to choose between a stroke based analysis and a character based analysis. As a preliminary attempt, we use character based recognition for on-line handwriting recognition of Tamil which is a very popular south Indian language. It is also one of the official languages in countries such as Singapore, Malaysia, and Sri Lanka apart from India.

There are 247 distinct symbols in Tamil of which 156 symbols are commonly used now-a-days. Of the latter, 12 are pure vowels, 23 are pure consonants, and

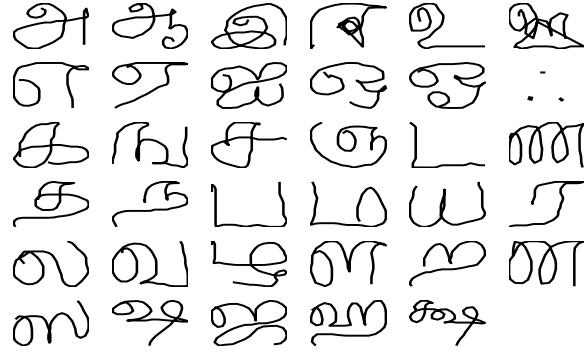


Fig. 1. Basic Tamil character set

the remaining are vowel-consonant combinations. Pure vowel and consonant set is shown in Fig. 1. To the best of our knowledge, very few attempts for Tamil on-line handwriting recognition have been reported. Keerthi et. al. [3] consider the problem of representation of Tamil characters. Deepu [4] considers subspace based on-line recognition of Indian scripts which includes Tamil.

This work of on-line Tamil handwriting recognition is based on template based elastic matching algorithms. Advantage of elastic matching algorithms is that they do not require a relatively large amount of training data, making them suitable for writer dependent recognition. Many such methods of elastic matching have been reported [5-7]. However these algorithms have the disadvantage that as training samples go on increasing, their classification time increases linearly. This may prove problematic particularly for real time applications. In this paper we present the results of our experimentation on three different features and seven different recognition schemes based on dynamic time warping (DTW). Rest of the paper is organized as follows. Section 2 gives the details of the database that we have collected. Section 3 describes preprocessing methods. Section 4 describes three different features that we have used. Dynamic time warping is introduced in brief in Section 5 along with seven different recognition schemes based on it. Section 6 gives experimental results and our discussion on it. Finally, Section 7 summarizes few important observations and concludes the paper.

2 Database

The iPAQ pocket PC (ARM processor running on Win CE operating system) is used for data collection. Sampling rate for this device is 90 points per second. A database of 20 writers, each writing all 156 characters ten times, is collected. Each character is written in separate box, which obviates the need for character segmentation. In this work we are attempting writer dependent recognition. Therefore, out of the ten data samples of a particular class for a given writer, seven data samples are used as training set to model a class and the remaining three data samples are used for testing.

3 Preprocessing

The initial number of pen-down points recorded by the digital Tablet varies roughly between 50 and 200 points for different characters. Raw data is passed

through series of preprocessing operations namely, smoothing, centering, size normalization, and re-sampling. Re-sampling is done to obtain a constant number of points uniformly sampled in space rather than the input data which is uniformly spaced in time. The total length of trajectory is found out and divided by the number of intervals required after re-sampling. The re-sampled characters are represented as a sequence of points $[x(n), y(n)]$, regularly spaced in arc length. Uniform re-sampling results in 60 points for all characters. Special care has been taken for re-sampling characters with multiple strokes in order to preserve the proportion of the strokes. In such cases, the re-sampling for any given stroke is done as a fraction of the length of the stroke to the total length of the character.

4 Feature extraction

As mentioned earlier, we experiment on three different features.

- **Preprocessed x-y co-ordinates:**

x-y co-ordinates can be used as features after preprocessing. Let P be the pre-processed character with 60 points.

$$P = \{p_i\}, \quad \text{where, } i = 1, \dots, 60 \quad \text{and,}$$

$$p_i = (x_i, y_i)$$

Note that feature dimension is 120 for this case.

- **Quantized slopes:**

This feature is also referred to as direction primitives. In order to get these features, we first find the slope angle of the segment between two consecutive points of P as,

$$\theta_i = \tan^{-1} \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right)$$

The slope angle obtained is quantized uniformly into 8 levels. Let Q be the set of quantized slope values corresponding to given P , then

$$Q = \{q_i\}, \quad \text{where, } i = 1, \dots, 60 \quad \text{and } q_i \in \{0, \dots, 7\}$$

The feature dimension for this feature is 60.

- **Dominant point co-ordinates:**

Dominant points of a character P are those points where quantized slope values q_i change noticeably. More specifically a point p_i of P is said to be a dominant point if following two conditions are satisfied simultaneously:

$$(q_{i+1} - q_i + 8) \% 8 \geq FI \quad \text{and} \quad (1)$$

$$(q_i - q_{i+1} + 8) \% 8 \geq FI \quad (2)$$

where, FI is *Flexibility Index*, $2 \leq i \leq 59$, and $\%$ is *modulo* operator. FI is the threshold set for deciding any point as a dominant point; it can take any value from the set $\{0, \dots, 4\}$. By default, first and last points of P are considered as dominant points. Fig. 2 illustrates this concept. As FI increases, structure of the character becomes more coarse and vice versa. The dimension for this feature will vary from one template to other, according to inherent complexity of that character and how it has been written. It is also interesting to note that this derived feature implicitly uses slope information along with the explicit use of x - y co-ordinate information.

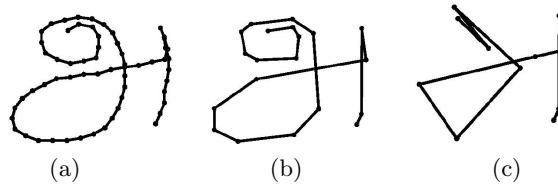


Fig. 2. Dominant points of a character at different FI values. (a) $FI = 0$, (b) $FI = 1$, (c) $FI = 2$.

5 Character Recognition schemes

This section describes seven different recognition schemes based on DTW. It is an elastic matching technique and hence it allows to compare two sequences of different lengths. This is especially useful to compare patterns in which rate of progression varies non-linearly which makes similarity measures such as euclidean distance and cross-correlation unusable. Here, time alignment or warping is carried out using dynamic programming concepts. For further description of DTW refer to [8].

5.1 Basic schemes

The following four schemes employ single stage procedures for recognition purposes and hence we refer to them as *basic schemes*. The choice of these schemes is motivated by their possible use in real time applications.

- **Scheme 1:**

This scheme uses preprocessed x - y co-ordinates as features. Euclidean distance is used as *cost measure* of dissimilarity between two points of feature vector. DTW distance between test pattern and templates is used for classification purposes for which nearest neighbor classifier is used. Computational complexity of this method is $O(M^2N)$ where M is length of sequence and N is total number of training templates across all the classes.

- **Scheme 2:**

This scheme uses quantized slope values as features. A fixed cost matrix is used to find out *cost measure* of dissimilarity between two quantized slopes. DTW distance measure and nearest neighbor classifier is used for classification purposes. Even though the computational complexity is same as that for *scheme 1*, Euclidean distance calculation is replaced by the simple table look-up operation and hence overall speed is expected to improve.

- **Scheme 3:**

This scheme uses dominant point x - y co-ordinates as features. Flexibility index is set to 1. Rest of the procedure is same as that for *scheme 1*. Computational complexity for this scheme is $O(\hat{M}^2N)$ where $\hat{M} < M$. As a result, considerable improvement in the recognition speed is expected.

- **Scheme 4:**

This scheme uses preprocessed x - y co-ordinates as features and Euclidean distance as *cost measure*. However, the important difference between this scheme and *scheme 1* is that here warping path is forced to follow diagonal path of the warping matrix. Hence this is a one to one or *rigid* matching scheme. Nearest

neighbor classifier is used for classification purposes. Computational complexity of this scheme is $O(MN)$. Hence this is the fastest scheme among four basic schemes.

5.2 Hybrid schemes

Next three schemes are combinations of the above four basic schemes. Each of the following three *hybrid schemes* accomplishes the recognition task in two stages. First stage acts as a *pre-classification* stage with low computational complexity and selects top 5 choices as its output. Second stage selects the output from these 5 choices and provides *post-classification*.

- **Scheme 5:**

This scheme uses quantized slope based classifier described in *scheme 2* at the pre-classification stage and preprocessed x - y co-ordinate based classifier described in *scheme 1* at the post-classification stage.

- **Scheme 6:**

This scheme uses dominant point co-ordinates based method described in *scheme 3* at both of its stages. In the pre-classification stage, FI is set to 2 and in the post-classification stage, it is reduced to 1. We also refer to this scheme as *hierarchical* dominant point based scheme, where dominant points are selected by gradually reducing FI value. At high values of FI , computational complexity is low but structure of the character is also coarse and hence they suit the first of two stage classification.

- **Scheme 7:**

This scheme uses rigid matching scheme based on preprocessed x - y co-ordinates (*scheme 4*) at its pre-classification stage. Post-classification stage uses elastic matching scheme based on preprocessed x - y co-ordinates (*scheme 1*). Higher computational complexity of DTW is due to its elastic matching capability. Hence idea here is to gradually switch from rigid matching to elastic matching.

6 Experimental Results and Discussion

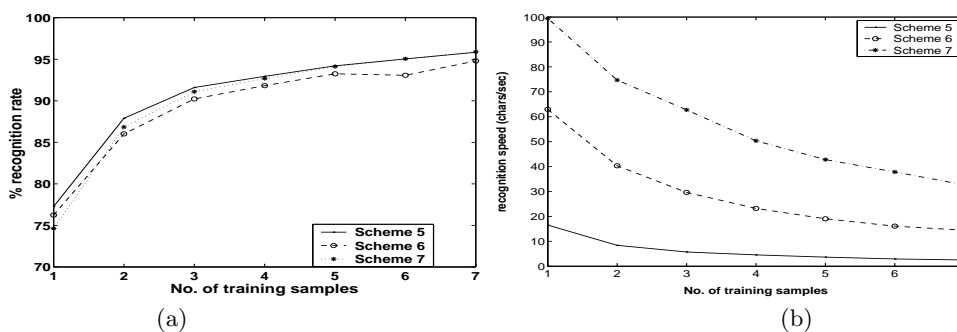
In this section, we present the results of our experimentation on the seven schemes mentioned earlier. We study the performance of these schemes with respect to three criteria: average recognition accuracy, average recognition speed, and number of training templates used per class. Average recognition accuracy is found out by dividing the number of correctly recognized test patterns by the total number of test patterns. Average recognition speed is calculated by dividing the number of test patterns recognized by the total time taken and its unit is characters per second. While the first criterion evaluates effective recognition capability of a scheme under consideration, the remaining two are important for studying effectiveness of that scheme in real time applications. All of the results are obtained on the machine with Intel Pentium IV processor and 256 MB RAM. First we present the results for the basic schemes (*schemes 1-4*). Our main intention to study these methods is to find their suitability for hybrid schemes. Therefore, average recognition accuracy is given up to top 5 choices. In this experiment, we use 7 training templates per class. Table 1 gives the results for four basic schemes. We can notice that the recognition accuracy for *scheme 1* is the highest. However, recognition speed is lowest. This fact is attributed to the

Table 1. Recognition results for four basic schemes

Scheme	top 1 (%)	top 2 (%)	top 3 (%)	top 4 (%)	top 5 (%)	Speed (chs/s)
1	96.30	99.22	99.50	99.64	99.69	1.69
2	88.22	96.21	97.85	98.51	98.87	3.31
3	94.89	98.51	98.93	99.09	99.15	5.83
4	90.65	96.37	97.69	98.20	98.49	67.39

inherent computational complexity present with DTW distance based methods. Low accuracy of *scheme 2* implies that quantized slope values by themselves do not work as a feature for the problem at hand, though they provide small gain in recognition speed. However, high accuracy and improved recognition speed of *scheme 3* suggests that implicit use of quantized slope values prove useful for recognition purposes. *Scheme 4* exhibits fairly low accuracy, which may be due to the fact that Tamil characters are more curved in nature and only rigid matching is not sufficient for their recognition. However, top 5 choice accuracy is reasonably high for all of them. Hence *schemes 2-4* can be used as the first stage of a two stage classification strategy.

Next, we present the results for the hybrid schemes (*schemes 5-7*). Fig. 3(a) shows the performance of hybrid schemes in terms of accuracy whereas Fig. 3(b) shows the performance in terms of recognition speed against number of training templates used. We notice that *schemes 5* and *7* perform equally well in the region where training templates are more, with the maximum accuracy of 95.89%. However recognition speed for *scheme 5* is much lower than of *scheme 7*. On the other hand, *scheme 6* shows somewhat lower recognition accuracies than the rest two methods. This is possibly because the process of extraction of dominant points may result into removal of some useful information, which in turn leads to increased error rate. But it has better recognition speed than *scheme 5* and hence it is more useful for real time application purposes. More-

**Fig. 3.** (a) Plot of recognition accuracy vs. number of training samples, and (b) plot of recognition speed vs. number of training samples, for three hybrid schemes

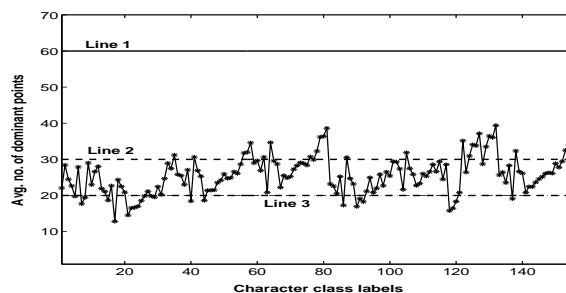


Fig. 4. Plot of average number of dominant points vs. character class

over, recognition accuracies can be improved to some extent by increasing the number of top choices selected at pre-classification level, at the cost of reduced recognition speed.

Secondly, we notice that all the three schemes show same general behavior with respect to recognition accuracy. Graphs in Fig. 3(a) do not show saturation at high values of the number of training templates. And hence it is expected that, in general, recognition accuracy will improve with the increase in number of training templates, though at the cost of recognition speed. The graphs also show remarkable decrease in recognition accuracy below 3 training templates per class. A possible reason for this may be that most of the writers adopt two or more writing styles for some characters. This variability is completely disregarded when less number of training templates are used.

Fig. 4 shows how the average number of dominant points varies with character classes. We note that preprocessed characters contain 60 points (represented by *line 1*) where as the number of extracted dominant points from each character is much smaller. More specifically, a given character may contain as high as about 40 and as less as about 10 dominant points. Average number of dominant points that a character contains is around 25. Since dominant points are the points with “high information” content, Fig. 4 illustrates the amount of inherent redundancy present in a character. The fairly high accuracy achieved by *scheme 3* confirms this statement. We also notice that the graph shows considerable variation along *y*-axis. And hence the number of dominant points present in a character can serve as a feature for a grouping strategy. Grouping strategy can further help to improve the recognition speed. One such possible strategy is depicted in Fig. 4. *Lines 2* and *3* divide the graph into 3 parts. Character classes from each part can be grouped together to form 3 non-overlapping groups. Fig. 5 shows an example

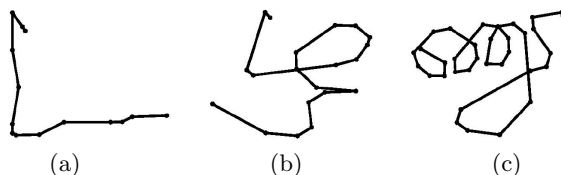


Fig. 5. Grouping of character classes based on dominant points. Number of dominant points are 14 in (a), 21 in (b), 35 in (c).

character belonging to each such group. Another possibility is that we form 3 overlapping groups instead of non-overlapping groups.

We conclude this section by listing some of the prominent confusion pairs (or triplets). They include (ஸ, ண, னை), (ஜ், ஜீ), (வ, ல), (ஶ, ஶ், ஶு) and (ஏ, ர). These errors are observed irrespective of the scheme used. While some of them look visually similar such as, (ஸ, ண, னை) and (வ, ல), others occur due to elastic matching capability of DTW viz. (ஶ, ஶ், ஶு) and (ஏ, ர). A possible solution to overcome this problem could be to extract structural features such as loops and cusps. Another possible solution could be to use these schemes in combination with other classifiers producing non-overlapping errors.

7 Conclusions

We have described implementation details of an on-line Tamil handwriting recognition system. Three different features are used in this study. A comparison of seven different schemes based on the three features and dynamic time warping based distance measure has been presented. Our results show that dominant points based two-stage scheme (*scheme 6*), and combination of rigid and elastic matching schemes (*scheme 7*) perform better than rest of the schemes, especially from the point of view of implementing them in a real time application. *Scheme 6* gives 94.8% recognition accuracy with recognition speed of 14.45 chars/sec where as *scheme 7* gives 95.89% recognition accuracy with recognition speed of 32.65 chars/sec. Efforts are underway to devise character grouping schemes for hierarchical classification, and classifier combination schemes so as to obtain a computationally more efficient recognition scheme with improved recognition accuracy. Some of our observations in this regard and possible solutions have also been presented briefly.

References

1. Charles C. Tappert, Ching Y. Suen, and Tory Wakahara, "The state of the art in on-line handwriting recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, Aug. 1990, pp. 787-808.
2. R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, Jan. 2000, pp. 63-74.
3. C. S. Sundaresan and S. S. Keerthi, "A study of representations for pen based handwriting recognition of Tamil characters," *Fifth International Conference on Document Analysis and Recognition*, Sep. 1999, pp. 422-425.
4. Deepu V., "On-line writer dependent handwriting character recognition," Master of Engineering project report, *Indian Institute of Science*, India, Jan. 2003.
5. S. D. Connell and A. K. Jain, "Template-based on-line character recognition," *Pattern Recognition*, 34(2001), pp. 1-14.
6. X. Li and D. Y. Yeung, "On-line handwritten alphanumeric character recognition using dominant points in strokes," *Pattern Recognition*, vol. 30, no. 1, pp. 31-44.
7. S. Masaki, M. Kobayashi, O. Miyamoto, Y. Nakagawa, and T. Matsumoto, "An on-line handwriting character recognition algorithms RAV (Reparametrized Angle Variations)," *IPSJ journal*, vol. 41, no. 9, 1997, pp. 919-925.
8. E. Keogh and M. Pazzani, "Derivative dynamic time warping," *First SIAM International Conference on Data Mining (SDM'2001)*, Chicago, USA, 2001.