Tamil Characters Recognition and Retrieval

Abdol Hamid Pilevar
Language Engineering Lab, Computer Engineering Department, Bu Ali Sina University, Hamedan, Iran

Keywords: Feature Analysis, Text Retrieval, Character Recognition, Natural Language Processing.

Abstract: In this paper the shape of the vertical projection curves are considered. The behavior of the edges of vertical projection curve is selected for creating the feature vectors of the characters. The edges of the vertical projection curve traced and the direction of the movement in the edges has been mapped by Eleven Direction Method (EDM) method .The direction codes have been extracted and saved as features vectors of the characters. The method is tested on the Tamil printed text documents. The testing data are collected from various legal documents. The test documents contain alphabet, special characters. A technique named EDM is used to search and retrieve the characters from Tamil text databases. The effectiveness and performance of the proposed algorithm have been tested with 10 separate sample data of 6 different fonts. The experiments shows that more than 97% of the Tamil characters are recognized correctly therefore, the proposed algorithm and the selected features perform satisfactorily.

1 INTRODUCTION

Incorrect segmentation of merged characters is one of the main causes for errors in the recognition. As segmentation errors induce recognition errors, the performance of segmentation is crucial for the whole OCR process. For segmenting the merged characters in a work which is called cut-off point or an inflexion, the point with the smallest interior angle is detected and the whole stroke is split into two adjacent curves by this point (Lu et al., 2008). Character recognition based on pixel distribution probability of character image (Wang, 2008). A method for segmentation of touching italic characters (Li et al., 2004). A segmentation technique for touching Thai type written (Watch, 2004). A segmentation of machine printed Gurmukhi text (Davessar, 2003). A segmentation technique far recognizing the touching Thai type written (Electronic et al., 2004). A lossy/lossless compression method for printed typeset bi-level text images is proposed for archiving purposes (Grailu et al., 2009). A recognition method of line-touching characters without line removal (Hotta et al., 2008). Links between landscape aesthetic theory and visual indicators (ode et al., 2008). A technique for identification and segmentation of Bengali printed characters (Sattar et al., 2007). Vertical and horizontal text lines are detected without prior assumption. The touching characters belonging to different lines are detected (Faure et al., 2009). A method for segmentation of touching italic offered (Li et al., 2004). A two pass algorithm for the segmentation and decomposition of Devanagari composite characters-symbols into their constituent symbols (Bansal et al., 2002). A recursive segmentation algorithm for segmenting touching characters (Liang et al., 1994).

Furthermore, since segmentation errors often raise chain effects, the performance of segmentation is crucial for the whole OCR process. In this paper, we address this issue by proposing a robust method for detecting the touching characters in Tamil text document images. We also present a technique for recognition of isolated Tamil character images using simple features.

2 PREPROCESSING

In the present system, character images have been obtained by optical scanning of the character images on plain paper. The input data obtained by scanning of printed text is contaminated with noise and contains redundant information. Preprocessing includes noise removal, elimination of redundant information as far as possible, segmentation, and scaling. The segmentation started by scanning the

Hamid Pilevar A..
Tamil Characters Recognition and Retrieval.
DOI: 10.5220/0004030504870493
In *Proceedings of the 7th International Conference on Software Paradigm Trends* (ICSOFT-2012), pages 487-493
ISBN: 978-989-8565-19-8

page images then continued by horizontally detecting the Text lines in each scanned page. Frequency of black pixels in each row is counted in order to construct the row histogram.

The position between two consecutive lines, where the number of pixels in a row is zero denotes a boundary between the lines. After a line has been detected, it is scanned vertically. In order to find the column histogram, the number of black pixels in each column is counted. If there are *n* consecutive vertical scans that find no black pixel, we denote those columns to be a marker between two words. The value of *n* is decided experimentally. To segment the individual character in a word, the column histogram is found; number of black pixels in each column is calculated. If there are more than one consecutive scans that has no black pixels, then the region is decided to be a marker between characters, and again if it is more than *n*, then considered as marker between two words. The isolated characters have been normalized so that size invariant recognition is possible. Though the recognition is size invariant, better result is obtained when the size of the characters is assumed to be within a specific range. The performance of our algorithm has been tested, by using the fixed size characters in creating our characters training sets.

For the developed system, 6pt - 36pt character sizes have been selected. The characters are scaled to a standard size (40 x 40) using an efficient scaling algorithm (Kumar et al., 1997). Scanning commonly introduces noisy cavities in the character images. These distortions detrimentally affect the shape of the characters. Single pixel components of noise are removed from the character images, before feature extraction.

3 TAMIL ALPHABET

Tamil has many letters, but most of them are derived from the 12 vowels and 18 consonants. The other 216 letters are made by combining the sounds of a vowel and a consonant. Even in writing, they are viewed as the addition of a vowel and consonant.

From more than 300 Tamil fonts which some of them are very stylish too, six of the simplest fonts are selected. For providing the test documents, the Azhagi editor is used. The samples of the six selected fonts are shown in Figure 1.

Chanakya	பணிவே சக்தி
Ilango	அன்பே சிவம்
Kamal	அன்பே சிவம்
Kannadasan	பணிவே சக்தி
Nambi	அன்பே சிவம்
Padma	பணிவே சக்தி

Figure 1: Six different fonts "Chanakya, Ilango, Kamal, Kannadasan Nambi, and Padma" are used in test data.

4 FEATURE SELECTION AND EXTRACTION

The features are extracted, and vectors are selected in the follows stages:

1. Project the pixels Vertically, and then calculate the number of pixels in each column(VPc).
2. Find the Directional Vectors (DV).
3. Calculate the Angles of Directional Vectors (ADV) with the X axis.
4. Based on discrete directional table (DDA) values, convert the directional vector angels (ADV) to eleven connected chain (EDM) codes.

4.1 Vertical Projection Count (VPc)

The vertical projection of all the Tamil characters (consonant "க, ங, ச, ஞ ..." , borrowed consonants "ஜ, ஷ, ஸ, ஹ, க்ஷ", vowels isolated form "அ, ஆ, இ ...", vowels compound form "க், க, கா ...") were found, and the black pixels for each projected column are count.

4.2 Directional Vector (DV)

The variations of the VPC values from one column to the next column (in the right side) are found and the directions of the variation are respectively registered as directional vectors.

4.3 Angle Calculation of Directional Vectors (ADV)

The angle of directional vectors are calculated with,

$$\alpha_i = \text{atan}(y_{i+1} - y_i) * \frac{360}{2\pi}$$

where: i = 1 …n-1

n is the number of features in the feature vector

$\pi = 3.1416\ldots$

y_i is y coordination of starting point of directional vector i.

y_{i+1} is y coordination of ending point of directional vector i.

For example: the graphical depict of a directional vectors are shown in Figure 2.

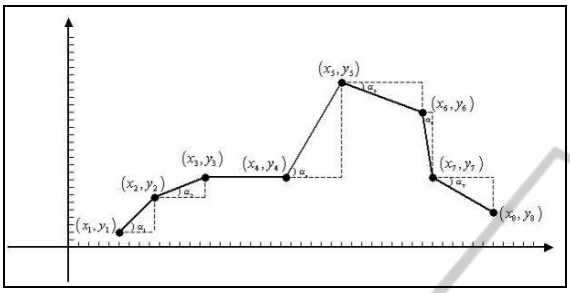


Figure 2: The graphical representation of a directional vector.

4.4 Eleven Connected Chain (ECC)

In Freeman's coding method characters are usually encoded either with 8-connected or 4-connected chain codes. For the work described in this paper, 11-connected chain codes were used (Figure 3). Our experiments show that eleven directions method has better results than 4 and 8 or other numbers of directions. The technique is named EDM coding system.

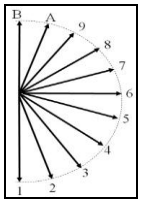


Figure 3: Eleven connected chain coding.

4.5 Discrete Directional Angle (DDA)

In our recognition system, templates are presented by strings of normalized discrete directional angle values. The angle values are classified as demonstrated in Table 1.

The number of the classes is selected to be eleven classes for being compatible with ECC-codes.

The number of the classes is selected to be eleven classes for being compatible with ECC-codes.
The number of the classes is selected to be eleven classes for being compatible with ECC-codes.

Table 1: normalized discrete directional angle values

Direction	Angle >	Angle <=
1	-99	-81
2	-81	-63
3	-63	-45
4	-45	-27
5	-27	-9
6	-9	9
7	9	27
8	27	45
9	45	63
A	63	81
B	81	99

4.6 EDM-Code

If the calculated discrete directional angles (DDA) string for a given directional vector in stage (d) are: -88, -36, 14, 53, and 86 then its EDM-code by referring respectively to Table1 is 1479B.

The EDM-codes for all of the Tamil characters (consonant "க, ங, ச, ஞ ..." , borrowed consonants "ஜ, ஷ, ஸ, ஹ, க்ஷ", vowels isolated form "அ, ஆ, இ ...", vowels compound form "க், க, கா ...") are calculated and saved in EDM-Table.

4.7 Feature Vector

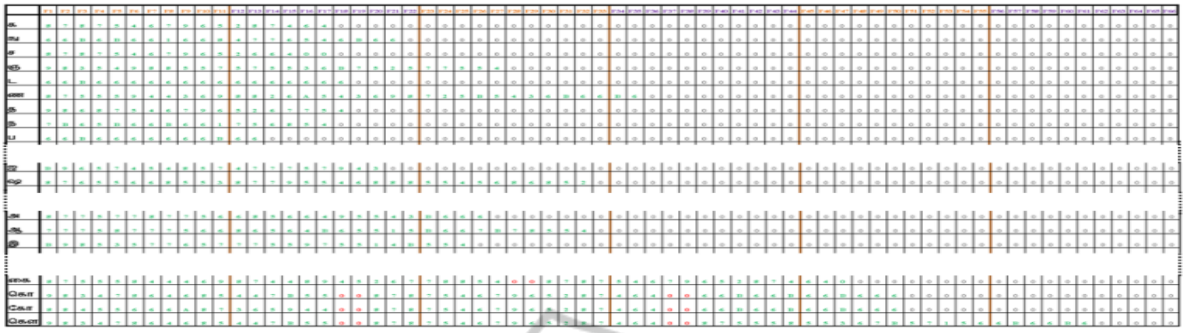
The smallest extracted features vector belongs to "ர" and "ப" with 13 features and the largest features vector belong to "கௌ" with 61 features in it. Our scaling template size is eleven features and the feature vectors are classified based on the number of fitted templates (NFT) in them. For example NFT of "ர" and "ப" are 2, and the NFT of "கௌ" is 6.

The EDM-codes are selected as characters feature vectors and saved in the feature vectors table (Table 2).

5 CHARACTERS RECOGNITION

Template-matching algorithms can recognize touched and broken versions of character templates without major difficulty. The reason for this phenomenon is simple: The difference between a broken image and the ideal one is relatively small.

Table 2: characters Feature vectors table



For template- matching, if a gap is introduced in an image and it does not change the interpretation of the image, it increases the distance from the image to all templates so the recognition is not affected. Statistical classifiers trained to recognize the most common broken characters, and in this context a broken character is better seen as the result of a feature extractor anomaly.

In contrast, most structural approaches to classification are confused by broken characters, since a broken piece may drastically change the representation of the character structure.

Our approach allows recognition of broken characters by a template matching technique: the representations of a character with and without a gap look alike modulo the graph matching. In other words, the representation does not decide if the gap corresponds to a missing part or to a real separation of strokes, but allows both possibilities to coexist until the matching with models decides which the best interpretation of the gap is.

This property for gap representation is just an extension of our paradigm that all singularities on the image should appear on the input graph, but only in the context of the models is the interpretation of the singularities given.

Thus, gaps should be represented as character parts that may be missing or not from the ideal image. Gaps are now positive features with ambiguous interpretations. This brings up two problems: first, how to identify all gaps, in other words, how to define a gap, and, second, how to consider only possible useful ones, since we want to increase the number of edges in the input graph as little as possible.

Figure 4 illustrates the segmentation procedure of the characters "ஸ க்ரு". The optimal Cutting Points (CP) can be found after several forward and backward cutting iterations. The segmented patterns are best matched to the prototypes. For the characters containing two or more parts, mistake may happen in the sequences of the recognized characters. The broken characters will be rejoined in the merge process using layout context information.

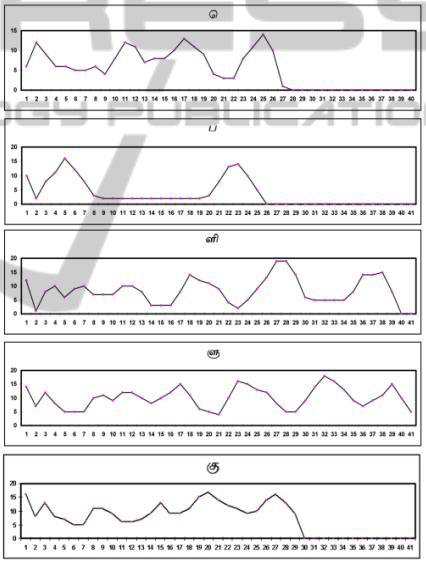


Figure 4: Graphical representation of directional vectors for the characters "ெ ப ளி ஸ க்ரு".

6 DYNAMIC RECURSIVE SEGMENTATION ALGORITHM

The approach presented in this section, the dynamic recursive segmentation algorithm, executes a forward segmentation or a backward merge process dynamically, based on the recognition result of the current input array and the neighboring character arrays, until the connected components are accepted by the classifier as a valid character. Instead of registering all the cutting points whose

discrimination function values exceed the specific threshold, only the cutting points determined by the recursive segmentation algorithm are recorded for additional processing with context information and spelling tools. To prevent from misclassifying the characters whose shape is similar to other characters, the minimum distance classifier utilizes multiple rejection thresholds to control the recursive segmentation process. R_1, R_2 and R_3 are respectively used for the initial input patterns (IP), the Residue input Pattern (RP) and the forward and backward input patterns (FP- BP). By properly adjusting R_1, R_2 and R_3 ($R_1 < R_2 < R_3$), the algorithm achieves the optimal results.

7 MATCHING THE FEATURE VECTORS

Given the feature vector for a word, our method finds vectors and sub-vectors that are homeomorphic to some prototype. A distance function between vectors measures the amount of distortion between vector, sub-vector and prototype. This distance function represents the minimum transformations that a vector will undergo so that the matching is possible. Thus, it is used as a measure of the quality of the matching. The cost of matching two vectors is the distance between the vectors.

The cost function and the algorithm that is used to find sub-vectors are basically the same used for recognition of segmented words**.** In order to find sub-vectors of the entire word vector that match prototypes, the recognition process is initiated on all features of the vector. The prototypes will guide the best recognition that contains the initial feature. The cost of the sub-vector matching does not include features that are not matched in the candidate, since they may belong to a different character.

8 SEARCH ALGORITHM FOR FEATURE VECTOR

A special tree search technique named EDM-Codes method is applied, the feature vectors are looked for using a tree based searching technique as displayed in Figure 4. The search process is as follows:

1-Extract the image of a word
2-Calculate the feature vector of the image
3-If the number of features is larger than 66, then it can be touching characters problem(follow the related routine: executes a forward segmentation or a backward merge process dynamically as discussed in sections 4, 5, and 6)
4-Look in the search tree for the most similar feature vector
5-If N features are matched then Match = N* 1.515
6-If Match is equal to 100 then it is exactly matched
else
if Match is less than 100 then
it can be a broken character (follow the related routine: executes a forward segmentation or a backward merge process dynamically as discussed in sections 4, 5, and 6)

Let us for example: a query image is given and its feature vector is extracted by the system, then the searching process will be followed through the nodes of the search tree as marked in Figure 5.

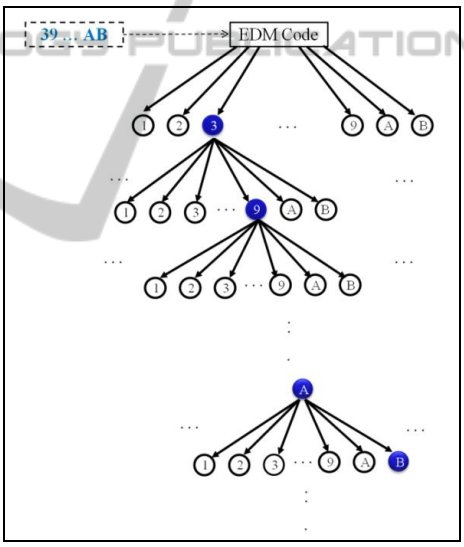


Figure 5: The search tree: each state represents a character candidate and the remaining pattern. A path from the root to a node constitutes a segmentation attempt. Search pattern for the features vector 39…AB is marked.

9 DISTANCE METRIC

The characters are classified based on the above mentioned features vectors and for similarity measurement the weight method is used (Pilevar 2005). The similarity degree S_i , between the i-th elements of the feature vectors of character images $f(x)$ and $g(x)$ is defined as:

if ($f_i == g_j$), then $S_{i=1}$

else $$S_i = \frac{\min(f_i, g_j)}{\max(f_i, g_j)}$$

Where:

f_i is the feature vector of the i-th character in the mean set of the training sets

g_j is the feature vector of the j-th character in the document.

The similarity degree between character images $f(x)$ and $g(x)$ is the sum of the similarity degrees between the corresponding n elements of the feature vectors derived from the two images, and defined as:

$$S = \sum_{i=1}^{n} \delta_i S_i \qquad \text{Where: } 0 \le S_i \le 1$$

$\sum_{i=1}^{n} \delta_i = 1$ n is the number of features , n<=66

10 EXPERIMENTAL RESULTS AND DISCUSSION

The effectiveness and performance of our algorithm have been tested on samples collected from various images of legal documents belonging to one city. For testing our method, around 200 printed Tamil text documents are scanned at 300 dpi and binarized using the two-stage method described in (Dhanya et al., 2001). For providing the text documents, the Azhagi editor is used. Six different fonts (Figure 1) are implemented for creating the documents.

The textual lines and words segments are determined from valley points in the horizontal and vertical projection profiles. A one-pixel margin is kept while detecting zone boundaries of the characters. However, it is assumed that all the characters of a text line are of the same font size. The extracted characters are normalized to the size of 40×40 pixels. After the character boxes are extracted, before starting the character recognition and recognizing the touching characters process, the documents are checked for their skew (Pilevar and Ramakrishnan, 2006). In the ultimate experiment, ten sets of separate documents given to the EDM software system, the documents are segmented into about 100000 characters, and more than 97% of characters are recognized correctly. However we couldn't find any similar work to compare ours with, but we believe that the outcome of this research is satisfactory and can be used as a base in Tamil character recognition systems in practical works.

REFERENCES

Bansal V., R. Sinha, "Segmentation of touching and fused Devanagari characters", *Pattern Recognition* 35, 875–893, 2002.

Davessar N. M., S. Madan, and H. Singh, "A Hybrid Approach to Character Segmentation of Gurmukhi Script Characters," *Pattern Recognition*, pp. 4-8, 2003.

Dhanya, D.: "Bilingual OCR for Tamil and Roman scripts. Master's thesis, Department of Electrical Engineering", *Indian Institute of Science*, 2001.

Electronics N., C. T. Center, and K. Luang, "Using Projection and Loop for Segmentation of Touching Thai Typewritten," *Analysis*, vol. 2004, pp. 504-508, 2004.

Faure, C., Vincent, N., "Simultaneous detection of vertical and horizontal text lines based on perceptual organization", *Proceedings of SPIE - The International Society for Optical Engineering*, Volume 7247, 2009.

Grailu, H., Lotfizad, M., Sadoghi-Yazdi, H, "A lossy/lossless compression method for printed typeset bi-level text images based on improved pattern matching", *International Journal on Document Analysis and Recognition*, pp. 1-24, 2009.

Hotta, Y., Fujimoto, K., "Line-touching character recognition based on dynamic reference feature synthesis", *Proceedings of SPIE - The International Society for Optical Engineering* Volume 6815, 2008.

Kumar S. and Muhammad Mashroor Ali, "An Efficient Object Scaling Algorithm for raster device", *Graphics and Image Processing*, NCCIS, 1997.

Li Y., S. Naoi, and M. Cheriet, "A Segmentation Method for Touching Italic Characters," *Pattern Recognition*, pp. 2-5, 2004.

Li Y., S., M. Cheriet, Ching Y, Suen, "A Segmentation Method for Touching Italic Characters", *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, 1051-4651/ 2004.

Liang S., M. Shridhar and M. Ahmadi, "Segmentation of touching characters in printed document recognition", *Pattern Recognition*, Vol. 27, No. 6, pp. 825 840, 1994.

Lu X., X. Liu, G. Xiao, E. Song, P. Li, and Q. Luo, "A Segment Extraction Algorithm Based on Polygonal Approximation for On-Line Chinese Character Recognition," *Japan-China Joint Workshop on Frontier of Computer Science and Technology*, pp. 204-207, 2008.

Ode, Ã., Tveit, M., Fry, G., "Capturing landscape visual character using indicators: Touching base with landscape aesthetic theory", *Landscape Research*, volume 33, Issue 1, pp. 89-117, February 2008.

Pilevar A. H, A. G. Ramakrishnan, "Inversion detection in text document images", *9th Joint Conference on Information Science*, Taiwan, 2006

Pilevar A. H., "Retrieval of signal from Biomedical Databases some new approaches", Ph D thesis, *University of Mysore*, 2005.

Sattar, Md. A., Mahmud, K., Arafat, H., Noor Uz Zaman, A. F. M., "Segmenting Bangla text for optical

recognition, *10th International Conference on Computer and Information Technology*", ICCIT, 2007.
Wang W., "Printed Chinese Character Recognition Based on Pixel Distribution Probability of Character Image," *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 1403-1407, 2008.
Watcharabutsarakham S., "Segmentation for touching thai typewrittens," *Science*, pp. 199-202, 2004.