# Recognition of Modification-based Scripts Using Direction Tensors

Lalith Premaratne Yaregal Assabie Josef Bigun
School of Information Science, Computer and Electrical Engineering
Halmstad University, S- 301 18 Halmstad, Sweden
hlp@ucsc.cmb.ac.lk Yaregal.Assabie@ide.hh.se Josef.Bigun@ide.hh.se

## Abstract

*The research on the OCR technology for the Latin-based scripts has been successful in achieving the status of image scanners with built-in OCR facility. But, a majority of modification-based scripts such as Brahmi descended South Asian or Ethiopic scripts are still progressing to achieve this status. This indicates the difficulties in adopting the recognition methods that have been proposed so far for the Latin-based scripts to modification-based scripts. In this paper we propose a novel method that can be adopted to recognise modification-based printed scripts consisting of a large character set, without the need for prior segmentation. The major strength of this method is that, the direction features that are used as the main principle for recognition, are further used in the separation of confusing characters, detection of skew angle, segmentation of script and graphic objects which substantially improves the computation efficiency. Algorithms developed initially for the Brahmi descended Sinhala script used in Sri Lanka, have been extended successfully for the Ethiopic script which has been evolved in a different geographical region, yielding consistently accurate results. Together, these two scripts are used by a population of ninety million.*

## 1. Introduction

The research on the Optical Character Recognition (OCR) for Latin-based scripts has reached a successful state leading to the availability of image scanners with built in OCR facility. In contrast, little research on a majority of modification-based scripts such as Brahmi descended South Asian and Ethiopic scripts have been published.

Although there are citations of research publications towards the OCR for the Brahmi descended scripts such as Devanagari, Gurmuki, Sindhi, Bengali, Tamil, Malayalam, Thai and Telugu, many of these scripts are yet to achieve the commercial OCR products. This clearly indicates the challenges faced by the researchers. Features such as horizontal and vertical segments, loops and junctions are specific to the characters in many of these scripts and therefore, the recognition techniques developed so far are characteristic to the relevant script. Hence, any option for extending a method proposed for one script to another is very slim.

Studies in the application of OCR techniques to Ethiopic characters were initiated at the Addis Ababa University in 1997. However, a research in Ethiopic OCR has been presented in a conference only recently [4], and still, much effort is needed to come up with a working system. The Sinhala script, which has been descended from the ancient Brahmi script over a two millennia, is used by over 75 percent of the 17 million population in Sri Lanka. Apart from [5] and [6], very little published research have been observed in the recognition of the Sinhala script.

In this paper we present an effective algorithm that uses direction features for the recognition of Brahmi descended Sinhala script and the Ethiopic script, which have been evolved in two different geographical regions. In addition to the recognition process of the script, which is the core step of an OCR system, issues such as detection of skew, segmentation of text from the graphic symbols have also been addressed. As proposed in this paper, the use of direction features in skew detection and graphic object segmentation not only provides accurate results but also gains improved computational efficiency. Further enhancements to the recognised script are achieved by using the Lexicon and Hidden Markov Models (HMMs) to improve the accuracy at word level.

## 2. Characteristics of the Alphabets and Scripts

The Sinhala alphabet consists of 16 vowels, 38 consonants and 16 modifier symbols (Figure 1). A consonant is modified using one or more modifier

symbols to produce the required vocal sound. The Sinhala script is organised in a three zonal horizontal line from left to right, with the middle zone occupying a major component of a character. Sinhala characters are generally rounded in shape and therefore, presence of a vertical or a horizontal straight-line segment is extremely rare.

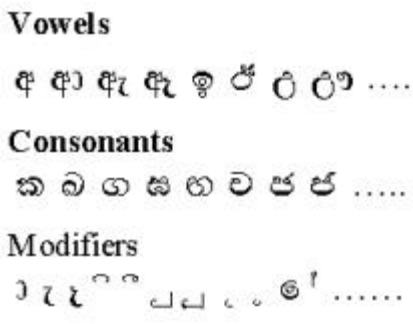


Figure 1. Some Sinhala Characters

Ethiopic is a name coined for Ethio-Semitic languages of Amharic, Geez, Tigrinya, and others, which are spoken by the major populations of Ethiopia and Eritrea. These languages use Ethiopic script as a mode of writing. The modern Ethiopic alphabet consists of 34 base characters and each base character has 6 other orders corresponding to different vocal sounds (Figure 2). The Ethiopic script is organised in a single zone in a horizontal line. A prominent feature within the script is the presence of vertical straight-line segments.

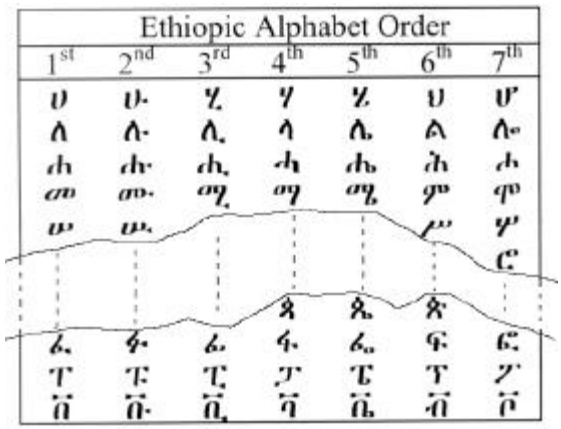


Figure 2. Some Ethiopic script core characters

In the modification-based scripts, a consonant, which is used as the base character, is generally modified using various modifier symbols. In some cases, the initial shape of the base character is not changed during the modification process and therefore, the process is reversible. This means, the base character and modifier symbols could be recognised separately. But often this is not the case. The initial shape of the base character is changed and a totally different symbol (composite character) is formed. In some modifications, a modifier symbol is placed inside the frame of the basic character. In such circumstances, the modified character needs to be treated as a different symbol in the character set. Since the modification process generates additional symbols of characters, the Sinhala character set in the script will consist of more than 120 different symbols and the Ethiopic character set will consist of 238 different symbols.

As observed in many other scripts, Sinhala and Ethiopic scripts too have similar characters that

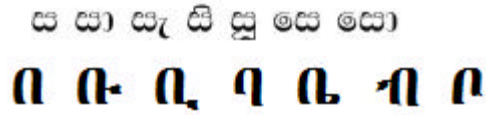

Figure 3. Modification of a Consonant (above- Sinhala, below - Ethiopic)

confuse with each other in the recognition process.. This problem has also been addressed successfully.

## 3. Theory

The orientation field tensor (Bigun[3]) is used as the main tool in the recognition process. A local neighbourhood with ideal local orientation is characterised by the fact that the gray value only changes in one direction. In the orthogonal direction the gray value is constant. Since the gray values are constant along lines, local direction also denoted as the direction of linear symmetry (LS).

The linear symmetry (LS) property can be extracted by using local spatial filtering [3] through separable gaussian filters and gaussian derivative filters. The LS vector field is a three dimensional tensor field and could conveniently be represented as one complex number called $I_{20}$ and one real number called $I_{11}$. The complex number is obtained via:

$$I_{20} = < \text{Grad}(f), \text{Grad}^*(f) >$$

where Grad = (Dx + iDy) is a complex operator, and < > is the usual scalar product, having a gaussian as a kernel.

The real valued $I_{11}$ is obtained via:

$$I_{11} = < |\text{Grad}(f)| , |\text{Grad}^*(f)| >$$

In the implementation, the Linear Symmetry (LS) Tensor of the image of the script to be recognised, is initially built. Each pixel of $I_{20}$ , which is represented as a vector will be of the form x + jy where $j = \sqrt{(-1)}$. The length of the vector is a measure of the local 'LS strength' and the argument is the estimated local orientation.

The $I_{20}$ vector is constructed by filtering the original image with two derivative filters (created as gaussian kernels), in the x-direction and in the y-direction respectively and by combining the two resulting images (dxf and dyf) as:

$$ls=(dxf+j*dyf)^2 ,$$

and by averaging the ls tensor. (The image `ls' is called the linear symmetry tensor).

In the implementation, four 1-D Gaussian kernels dx, dy, gx and gy are created and used as derivative filters, to convolve the image. The resulting images dxf and dyf are obtained by the filtering operations gy*(dx*img) and gx*(dy*img) respectively, where img is the original image representing the text of a script.

In the process of identification of a certain character, the following principle is used.

$$\max |(b^*)^t.a| \quad |b|=1$$

where $b$ is of the form c + jd, represents the $I_{20}$ component of the LS Tensor of the character and $a$ is of the form c' + jd', represents a frame (small rectangle) within the $I_{20}$ component of the direction tensor of the text to be recognised. The dimensions of the frames of $a$ and $b$ are the same. c, d, c' and d' are real values.

In other words, the scalar product $(b^*)^t.a$ (referred to as 'correlation' in the future) is maximum if the elements of $b$ are in the complex conjugate direction of the elements of $a$. Therefore, when the Tensor component of the character which is being recognised coincides with an occurrence of the same character in the LS Tensor of the image, the product $(b^*)^t.a$ will attain a relatively high value. (Note: $b$, which is the LS Tensor component of the character being recognised is extracted in advance from the LS Tensor of an image of a similar script).

## 4. Recognition

### 4.1 Correction of Skew

As mentioned in Section 1, the local direction features of the document image are used also to detect both the skew and the graphic objects in the document. Figure 4 shows two texts with two different skew angles of Sinhala and Ethiopic documents.

The complex moment component $I_{20}$ which represents the information of orientation angle is used to detect the skew angle of the document.

The resultant value of the $I_{20}$ in the LS Tensor of the entire script is a close enough approximation to the skew angle of the overall document. The accuracy is improved by removing the contribution to the resultant value from the four edges and by taking into account the value of $I_{20}$ of the pixels having high confidence.

A filter at a higher level (in this case level 4) in a (octave based) Laplacian pyramid responds strongly to the dominant orientation of the lines in the script.

### 4.2 Segmentation of Text and Graphic Objects

Segmentation of the area of text from the graphic objects is a prior requirement for the recognition of the script. [Bigun 1] and [Bigun and de Buf 2] present the use of orientation features in texture segmentation. In this work, experimental results of the successful segmentation of 7 distinct textures in 16 different patches in the image have been presented. The texture contrast between a graphic object and the script in our images, is coarser than those used in [1] and [2]. Therefore, the parameters used in the low-pass filtering were changed in order to make the lines more prominent by blurring the characters inside the lines of script in the image. The feature image is built using the $i_{20}$ and $i_{11}$ (section 3) in a 3-level Laplacian pyramid. The two-class segmentation clearly separates the text area from the graphic objects, in rectangular frames.

### 4.3 Recognition of Text

Recognition of text is performed by examining the strength of the relationship with respect to the direction features' (referred to as 'correlation') between each symbol in the script with each character in the character set. This is simply done by filtering the linear symmetry tensor (LS tensor) of the script with that of a character. When the LS tensor template of a character being tested, coincides with an occurrence of the same character in the script, the outcome is a relatively high value, which is easily separated from the outcome of the rest of the characters in the script using a suitable threshold. The plot of this correlation for a selected character is given in the figure 4.

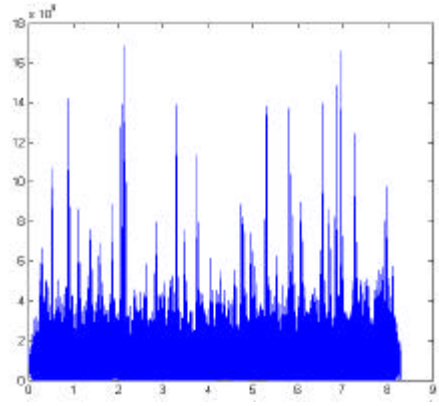

Figure 4. Correlation of a character with the script

The plot in figure 4 shows that in addition to the high correlation values depicted in each sharp spike, there are a few other relatively higher values corresponding to a cluster of spikes around each peak value. This is due to the fact that, in addition to the highest value that corresponds to the overlapping of the character template at the centre of an occurrence of the character, high values are generated with overlapping around the centre of the occurrence. Therefore, suppression of non-maxima will leave only the value corresponds to the required occurrence.

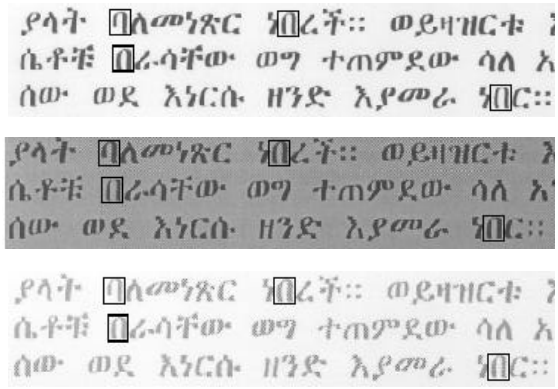

Figure 5. A character (shown in the rectangular frame) recognised using the same threshold in the same text with three different intensities

The experiments show that the primary threshold that separates a character from the tensor depends on the intensity of the image. In order to form a uniformity between images of different intensities, the tensor represented by ls./i11 is used for filtering in the recognition process. The tensor ls./i11 of any image produces the same correlation for a selected character, hence a single threshold could be used to recognise the character.

### 4.3.1 Building the Confusion Matrix and Separating Confusing Characters.

The first step in separating the confusing characters is the identification of confusing character groups within the character set. The confusion matrix was built by conducting experiments for a script consists of 30 occurrences of each character. The confusion matrix shows the uniquely recognised characters and the clusters of confusing characters. Further, it provides a quantitative measure of the intra-cluster confusion and the inter-cluster confusion. The numbers of confusing groups identified in the Sinhala and Ethiopic scripts are 11 and 13 respectively. Another characteristic derived from the confusion matrix is that, given the two characters A and B, character A might confuse with character B but not the vice versa. This property is important in arranging the order of characters to be recognised.

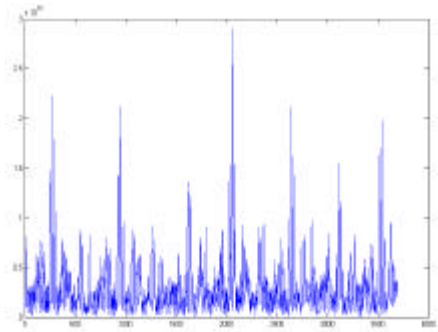

Figure 6. Correlation of a distinct segment of one character within the 7 members of confusing group (highest peak corresponds to the required character)

The same principle, which is used to recognise a character in the script, is used to separate a confusing character from the rest of the members in its confusing group. That is, the examination of correlation between each LS tensor of the confusing character with a distinct segment of the LS tensor of the character being tested. When the distinct segment coincides with the same frame inside a character, it generates a relatively high correlation, which could be separated using a suitable threshold. Hence, a secondary level filtering process is carried out to separate confusing characters. The number of filtering steps at secondary level will depend on the number of members in the confusing group.

### 4.3.2 Recognition of different fonts with different sizes

Since the characters in the recognising alphabet are extracted from one or more of the arbitrarily selected images, the best results for recognition are obtained only for the images with same font and size. Therefore, it is necessary to generalise the recognition process in order to accommodate different fonts with different sizes. The average line height in both the Sinhala and the Ethiopic scripts is a reasonably accurate parameter, which could be used for image size adjustments. The experiments show that a majority of widely used fonts which preserve the general curvature features of characters perform accurately when the size of characters is matched to that of the alphabet.

### 4.3.3 Preparation of Alphabet Data File and the ordering of characters

The main task of the recognition process is the preparation of the alphabet data, which will contain the data needed for recognition for each character. The

structure of the alphabet data file with the essential fields is given below.

i. ASCII value of character
ii. Character type
iii. LS Tensor of character
iv. Primary Threshold
v. Flag indicating the status of confusion
vi. LS Tensor of a distinct segment I
vii. Secondary Threshold I
viii. LS Tensor of a distinct segment II
ix. Secondary Threshold II

If the relevant character is unique which does not need the secondary level filtering as indicated in field v, values for the fields vi and vii are irrelevant. For a character, which needs more than one step of secondary filtering, these fields will have appropriate values.

Order is primarily arranged on the descending order of the LS Tensor template size of the character. This will avoid the template of a small character coinciding with a similar segment within a larger character. The order is then rearranged to give a preference to non-confusing or less-confusing characters.

### 4.3.4 Pre -processing of Image - Noise Removal, Size Adjustment and Threshold Adjustment

The first step in pre-processing is the skew correction, which is followed by the separation of script from the graphic objects. A horizontal projection is carried out to determine the average line height which is then used to adjust the size of the image to match the average character size of the script to that of the characters in the alphabet.

### 4.3.5 Recognition

The recognition algorithm is now performed for the pre-processed image. During this step, the LS Tensor of the image is built and each character is filtered through the LS Tensor of the image. Suppression of non-maxima within a 3x3 neighbourhood is performed. A secondary level filtering with one or more steps is carried out for each confusing character. For each accepted character, its row and column co-ordinates are recorded along with the character identifier and its ASCII value.

The efficiency of the filtering process is improved substantially, by limiting the filtering space only to the area that covers the script. Since the upper and lower line boundaries are determined during the horizontal projection, the filtering could be carried out inside the line boundaries.

### 4.3.6 Post-processing of Recognised Script

The output of the recognition algorithm is a structure array of recognised characters in the order of the characters in the alphabet. The structure of the array is as follows.

i. Character Identifier ii. Row Number
iii. Column Number iv. Character Type
v. ASCII value

The output array is primarily sorted on row number and each row is then sorted on column number. Characters in the sorted array are now arranged into words using the word boundaries. Paragraphs could also be separated during this process.

The false rejections are identified only at this stage. Such missing (vacant) character positions could be detected by comparing each distance between the centres of two characters in a word starting from the left boundary of the word, with average character width. The lexicon is now used to detect the possible missing characters.

## 5. Experimental Results

### 5.1 Skew Detection

Experiments were conducted to test the proposed algorithm to detection skew angles for a variety of skewed images. Some results are shown in Table 1.

**Table 1. Detection of Skew Angle**

| Angle | Sinhala | Ethiopic |
|---|---|---|
| -2 | -1.70 | -1.75 |
| +2 | +1.82 | +1.77 |
| -31 | -30.60 | -30.81 |
| +31 | +30.51 | +30.33 |
| -78 | -77.61 | -77.45 |
| +78 | +77.30 | -77.21 |

### 5.2 Segmentation of Text

Various images containing graphic objects such as human faces, sceneries along with text were used in the experiments for segmentation. Apart from a few cases where the white space between text paragraphs was included in the cluster of graphic object, all the other images were segmented correctly. In each case, the rectangular boundaries of the text area have been detected clearly (Figure 7).

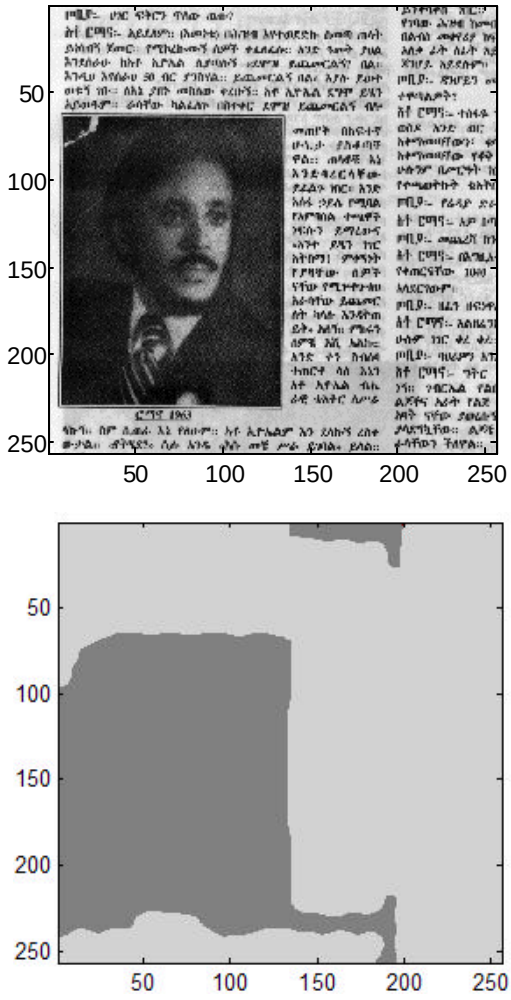

Figure 7. Segmentation of graphic objects from text

### 5.3 Recognition

Experiments covering over fifty different images of Sinhala text containing several hundred words achieve a recognition rate with approximately 90% accuracy. The same set of algorithms extended to the Ethiopic script at each level of recognition achieve similar results.(see Table 2)

**Table 2. Recognition Rate**

| Font | Sinhala | Ethiopic |
|---|---|---|
| Same | 93% | 92% |
| Similar | 85% | 82% |
| Different | 70% | Not Tested |

## 6. Conclusions and Future Work

In this paper, we have proposed an effective algorithm using direction features to recognise modification-based scripts. Although the algorithms were tested only for Sinhala and Ethiopic scripts, it appears that the same set of algorithms will effectively recognise most other scripts especially scripts with a large number of different symbols that are difficult to segment. Since each step in the OCR process is implemented at abstract level, the algorithms could be extended to a different script, with only minor modifications.

A further enhancement to improve the accuracy at word-level, using the HMMs is being tested with encouraging results. In these experiments, the Confusion Matrix built to identify intra and inter-group confusing characters, together with a State Transition Matrix constructed using the Lexicon, which represents the transition probabilities between each character in the script, are used to determine the 'most likely chain of characters' in the presence of a given recognised word.

## Acknowledgement

Lalith Premratne and Yaregal Assabie gratefully acknowledge the Swedish International Agency for Development Cooperation - Information Technology (SIDA-IT) for funding the research project.

## References

[1] J.Bigun, Frequency and orientation sensitive texture measure using linear symmetry, Signal Processing 29 pages 1-16, 1992

[2] J.Bigun and J.M. Hans du Buf, N-folded Symmetries by Complex Moments in Gabor Space and Their Application to Unsupervised Texture Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 16, No. 1, Jan, 1994.

[3] J.Bigun and G.H.Granlund, Optimal Orientation Detection of Linear Symmetry. ICCV'87, London 1987, pages 433-438, IEEE Computer Society Press, Los Alamitos, 1987.

[4] J.Cowell and F. Hussain, Amharic Character Recognition using a Fast Signature Based Algorithm. Proceedings of the Seventh International Conference on Information Visualisation pages 384-389, 2003.

[5] H.L.Premaratne and J.Bigun, Recognition of Printed Sinhala Script using Linear Symmetry, pages 813-317, Fifth Asian Conference on Computer Vision, January, 2002

[6] H.L.Premaratne and J.Bigun, A segmentation-free approach to recognise printed Sinhala script using linear symmetry, Pattern Recognition, 37, pages 2081-2089, 2004.