A Segmentation-free Approach to Recognise Printed Sinhala Script

H. L. Premaratne
University of Colombo School of Computing, Sri Lanka
Lalith.Premaratne@ide.hh.se hlp@mail.cmb.ac.lk

J.Bigun
School of Information Science, Computer and Electrical Engineering
Halmstad University, S-301 18 Halmstad, Sweden
Josef.Bigun@ide.hh.se

Abstract
Majority of character recognition algorithms such as the use of ANNs needs segmentation of the script prior to recognition. Contrast to Western scripts, Brahmi descended South Asian scripts such as Sinhala consist of modifier symbols, which make the segmentation a difficult task that needs to be addressed as a separate issue. Further, the change of shape of the basic character (by violating modification rules) in the modification process makes some modified Sinhala characters impossible to segment. The proposed method, which uses Linear Symmetry to examine a co-relation between characters in the script with the testing alphabet, recognises characters directly within the image of the script. A similar method is used to resolve confusing characters. Experiments show highly favourable results not only for the basic characters of the alphabet but also for the modifier symbols. A novel but simple method using Linear Symmetry for skew correction has also been proposed.
Key Words: *Linear Symmetry, Recognition, Segmentation, Skew Correction*

1. INTRODUCTION

1.1 Alphabet and the Modification Process

The Sinhala script used by over 80% of the 18.4 million population in Sri Lanka has been descended from the ancient Brahmi script and evolved independently over many centuries. The Sinhala language is unique to Sri Lanka and the Sinhala characters that are generally round in shape differ from all the other Brahmi descended scripts in South Asia. The Sinhala alphabet consists of 18 vowels, 41 consonants and 17 modifier symbols. A vowel may appear only as the first character of a word and a consonant is modified using one or more of the modifier symbols to produce the required vocal sound. The total number of different modifications from the entire alphabet including the basic characters is nearly 400. Although each character possesses a distinct characteristic shape to distinguish from the others, some characters resemble with one or more of the other characters by their appearance. Some examples are given in Figure 1.

Modification of a character is carried out by simply adding one or more modifier symbols before/after/above/below the character without affecting its general shape.. However this rule is violated for a specific subset of the alphabet numbering to 10 characters, in most of the printed scripts, to give a better appearance (Figure 2). Also, in some modifications, the joint between the character and the modifier symbol is smoothed to make the modified character appear as a single unit of symbol.

1.2 Characteristics of the Script

A single line of script is organised in three horizontal layers. The middle layer contributing to approximately 50% of the total line height, mainly include fifteen (15) basic characters and Nine (9) modifier symbols. Twenty two (22) other basic characters occupy the middle layer and the upper layer, with approximately 75% and 25% of the total height of each character in each layer respectively. The middle and the lower layers include the remaining eight (8) characters, with approximately 75% and 25% of the total height of each character in each layer respectively. Four (4) modifiers occupy the upper layer while the remaining five (5) modifiers are assigned to the lower layer. The upper and the lower layers are of equal height each having 25% of the total line height. (Figure 4).

1.3 The OCR Technology and Recent Developments

Optical Character Recognition (OCR) is the process of converting typed or printed documents into machine-readable code. The original typed or printed documents scanned to form an image file would be the input to the OCR software system. The result is a picture represented as light intensities on a rectangular grid of points, which do not yet identify individual characters. The OCR will in turn, recognise each character or symbol in the image file and make them available in a suitable text editor, which could either be edited or modified.

Most of the OCR systems use Artificial Neural Networks (ANN's) as the major tool. In addition to the features identified in a rectangular grid of a matrix that encloses a single character, other features of the character such as the curvature features and transition counts are also used. In the case of handwriting recognition, some common approaches are the ANN's, mathematical morphology, shape analysis and hidden Markov model (HMM). Each of the above approaches has its own strengths and weaknesses. Researchers have achieved a significant improvement in performance by combining two or more of the above methods. Majority of alphabets consists of confusing characters that resemble to each other to a greater extent. Resolving this problem especially in the case of handwriting recognition is a critical issue.

The research on the south and the South-East Asian scripts lag behind that on European scripts due to various reasons. The main reason is the complexity of a script. In Asian alphabets, the number of characters in the alphabet is high and the generation of a vocal sound by modifying a character using modifier symbols is complex. Extensive research has been done on a few scripts used by a very large population of the community. Some of such research has been initiated in developed countries due to the high exposure to such research.

At present, the OCR software for the languages such as Sindhi, Bengali and Thai are available as commercial products. The research on Devanagari and Tamil languages has achieved a tremendous progress. To the best of our knowledge, there have been no or a very little research done on the recognition of printed Sinhala script.

2. RECOGNITION PROCESS

2.1 Theory

The theory used in the recognition process is the orientation field tensor which has been used effectively in many applications over the past few years. A local neighbourhood with ideal local orientation is characterised by the fact that the gray value only changes in one direction. In all other directions it is constant. Since the gray values are constant along lines, local orientation is also denoted as linear symmetry [1]. The linear symmetry is also represented in the form a vector. Since the direction of a simple neighbourhood is different from the direction of a gradient, which is strictly cyclic, representation of the linear symmetry needs the doubling of the angle of orientation. The vector that represents the linear symmetry is composed of two quantities. One is the orientation angle and the other is the certainty measure.

2.1.1 Mathematical representation

The local orientation is determined using the following three steps [1].

i. Select a local neighbourhood from the image using a window function
ii. Fourier transform the windowed image
iii. Determine the local orientation by fitting a straight line to the spectral density distribution.

When fitting a straight line, the sum of the squares of the distances of the data points are minimised.
Since the minimisation of d_i is same as the maximisation of S_I, the equation (2) is obtained.

The orientation is obtained as the eigen vector of the largest eigen value of J. J can be rotated so that it is diagonalised. The rotation matrix is in fact the eigen vector matrix given in equation (1).

Comparison of the diagonal elements on both sides of the equation (1) gives

$$\lambda_1 + \lambda_2 = J_{xx} + J_{yy} \;;$$

$$\lambda_1 - \lambda_2 = (J_{xx} - J_{yy})\mathrm{Cos}2\phi + 2\,J_{xy}\,\mathrm{Sin}2\phi$$

$$= (J_{xx} - J_{yy}\;, +2J_{xy}) \begin{pmatrix} \mathrm{Cos}2\phi \\ \mathrm{Sin}2\phi \end{pmatrix}$$

$$= < I_{20}, \begin{pmatrix} \cos 2\phi \\ \sin 2\phi \end{pmatrix} > = < I_{20}, I_{20} / \|I_{20}\| > = \|I_{20}\|$$

$$\therefore \quad \lambda_1 - \lambda_2 = \|I_{20}\|$$

Define $\nabla f = \partial f/\partial x + i(\partial f/\partial y)$

then I_{20}

$$=\int(\nabla f)^2=\int((\partial f/\partial x)^2-(\partial f / \partial y)^2 + 2I(\partial f/ \partial x).(\partial f / \partial y))$$

$$=\int [(\omega_x + i\omega_y)^2(\omega_x - i\omega_y)^0|F|^2] = (\lambda_1 - \lambda_2)\exp(2i\phi)$$

$$I_{11} =\int[(\omega_x + i\omega_y)^1(\omega_x - i\omega_y)^1|F|^2] = \int (\omega_x^2 + \omega_y^2)|F|^2$$

$$= \int ((\partial f/ \partial x)^2 + (\partial f / \partial y)^2 = \lambda_1 + \lambda_2$$

Angle of I_{20} represents the (2 x angle) where the angle is the inclination angle of the fitting orientation if the linear symmetry exists, and I_{11} represents the sum of the best and the worst total errors.

The Linear Symmetry algorithm that extracts the tensor is characterised by the fact that it delivers a dense orientation field along with certainties. In case of high confidence on the existence of orientation, the linear orientation represents the least change of gray values in one direction and maximal change in the orthogonal direction. Hence a Linear Symmetry Tensor for an image is constructed by averaging the orientation of the local neighbourhood, for each pixel of the image.

2.1.2 Implementation

The LS Tensor for an image is built as explained in the following steps.

Four 1-D derivative filters dx (Gaussian kernal), dy (= - dx') and gx (Gaussian kernal), gy (= gx') are generated.

The two derivative convolutions dxf (= convolution(gy, convolution(dx, Image)) and dyf (= convolution(gx, convolution(dy, Image)) of the original image with respect to x and y are constructed using the above pair of filters.

The LS Tensor (complex) is then given by
LS = (dxf +j*dxy)^2 where $j = \sqrt{(-1)}$

The correlation between the character being tested with the image is calculated using the formula
absolute(convolution(conjugate(LS Tensor of Character), LS Tensor of Image)).

2.2 . Determination of Skew Angle

Almost all the recognition algorithms need the text lines in the input image to be horizontal. Therefore, any skew associated with the input image needs corrections prior to recognition. Experiments show that the recognition algorithm proposed in this thesis tolerates a skew of $+1^0$ to -1^0. The accuracy of recognition deviates considerably with the increasing skew. Therefore a robust method for skew correction needs to be incorporated.

Careful observation of a line of Sinhala script shows that the boundary between the upper and the middle layers and the boundary between the middle and the lower layers (fig. 8) possess the highest amount of energy in the horizontal direction. The horizontal projection of a sample script clearly agrees with this concept. This is due to the fact that any character in the alphabet should touch either at least one or both of these boundaries. Therefore, tracing the appearance of one of these boundaries in a skewed script could be used to determine the skew angle. Although any straightforward method to detect a boundary line could have been used, a more appropriate method using the Linear Symmetry (LS) tensor has been proposed.

The Linear Symmetry tensor [1] which gives information for each pixel of the image, on how it is organised with respect to the orientation within a local neighbourhood, could effectively be used to determine the orientation of the script. In general, the orientation angle of the resultant vector of all the vectors representing the LS for each pixel of the image would provide a near approximation to the skew angle. In order to improve the accuracy, the interference to the final result from the following components should be elimination.

i. Edges of the image
ii. Background of the image, which consists of pixels having random orientations of low confidence.
iii. Other pixels (within the text area) having orientations of low confidence.

The results obtained for the LS tensor derived in section 3.3.2 yield the skew angle within $+1^0$ to -1^0 accuracy, which is well within the required accuracy for the recognition algorithm.

2.3 Recognition Procedure

2.3.1 Testing Database.

The recognition process is based on the examination of the correlation of characters in the script with each character of the alphabet through a filtering operation. The testing alphabet which consists of all the characters (including the modifier symbols), is built by extracting characters from an LS tensor. Each character in the testing alphabet is filtered (one at a time) through the LS tensor of the script in order to identify its occurrences in the entire script. The plot of correlation at each pixel (Fig. 10) shows that, each occurrence of the character being tested gives a strong correlation. A suitable threshold that separates the required character from the rest of the characters in the script, is then determined. This procedure is conducted for each and every character of the alphabet. During this process, it has been observed that a total number of 35 characters amounting to 60% of the alphabet separates from all the other characters with a clear threshold (Fig. 10(a)) while the balance 40% confuse with one or more characters with similar shapes (Fig. 10(b)). Eight (8) such confusing groups have been identified.

Once all the different confusing groups are identified, another level of filtering is carried out to separate each character within the confusing group. The secondary level of filtering is performed to examine the correlation of a distinct segment from one character with all the members in the group (Fig. 11). A suitable (secondary) threshold that separates each character from the rest is then determined. A further level of filtering is carried out if the confusion still occurs.

The structure of the testing database is as follows.

Character Identifier
LS Tensor of character
Primary Threshold
Flag to indicate confusing status
Secondary Threshold (for confusing characters)
Tertiary Threshold (for confusing characters)

2.3.2 Recognition.

The image is initially pre-processed to remove the background noise. The image is then scaled (if necessary) to match the average height of a character to that of the testing alphabet.

Recognition of a script is performed by filtering the LS tensor of each character of the testing alphabet with the LS tensor of the script. In each filtering cycle, all the occurrences of the character being tested are identified. If the testing character is a confusing one, the secondary level of filtering is carried out in order to determine the acceptance or rejection of the identified character. A tertiary level of filtering is carried out similarly.

It has been observed that, in addition to the highest value of correlation produced usually at the centre of the character, a few more relatively high values are also produced around the neighbouring pixels. This is due to the fact that the template of the testing character nearly coincides with the neighbouring pixels around its centre. This will result in recognising the same character in the image more than once. Therefore, once the filtering has been performed, non-maximums in a small neighbourhood (e.g. 3x3) are suppressed in order to eliminate the multiple acceptance of the same character.

The recognition algorithm is as follows:

```
Input image
Input database-of-characters
*/Alphabet/*
Pre-process image
Perform Horizontal-projection
Extract Line-data
ConstructLS-tensor
Read character
While not-end-of-alphabet do
    Filter characte with the LS Tensor

        */ Primary Filtering /*
    Supress non-maximums
    While not-end-of-image do
        Segment occurrences above threshold
        If confusing-charcater
            Determine relative rhreshold
            Perform secondary-filtering
          /* and tertiary-filtering if necessary*/
        End-If
        Store image-coordinates of -each
        occurrence
    End-While *** not-end-of-image ***
    Update output array
    /* with ASCII Value, row, column no, .*/
    Read character
End-While *** not-end-of-alphabet***
Sort output on Column No. within the Row No.
```

Since a character is identified directly within the image of the script, the need to segment individual characters does not arise. Symbols such as comma, full stop, question mark are also recognised with the same accuracy.

3 EXPERIMENTAL RESULTS

Some experimental results obtained in determining the skew angle are given in Table 1 below.

Results of the recognition of individual characters during the initial stage show that while the characters with unique distinct shapes consist of 60% of the alphabet are recognised between 93 to 100 percent, the confusing characters are recognised at an average rate of 76%. 96% of the confusing characters are distinctly recognised at the secondary level of filtering. The experiments conducted for various images of the same font and size carrying 600 to 1200 characters per image yield approximately 92% accurate results. Nearly forty images carrying widely used fonts with varying sizes contributed to as high as 84% accurate results. It has been observed that the variation of size is tolerated within a 5-10% margin. Only 60% of the modifier symbols used in the upper and lower zones of the script are recognised with 70% accuracy. On the average, basic characters recognise at the rate of nearly 92%, modifier symbols used within the upper and lower zones recognise only at a rate of 70%. The quality of images varied from more noise to less noise and some of the images were captured from two year old newspapers of low quality prints.

4. CONCLUSIONS AND FUTURE WORK

In the industrialized world, OCR for a language plays a key role in document processing. In addition to facilitating the processing of texts available as images in documentation, an OCR will encourage the use of archives more effectively and efficiently. This thesis presents the outcome of an investigation carried out towards an OCR for the Sinhala alphabet. The main three stages in the OCR development namely the skew correction, segmentation and recognition are presented in details.

The principle feature used in the recognition process is the linear symmetry. Results of the experiments show that the feature itself can contribute to the recognition process very effectively. Therefore, it is hoped to continue the research to its highest depth in this direction and then accommodate other available techniques such as ANN's, statistical methods, language rules and an on-line dictionary to bridge the system to achieve the maximum accuracy and efficiency.

It has been observed that the correlation produced by a confusing character with another original character (having the same dimensions) is almost always less than that produced by the original character with itself. Therefore, more accurate results could be achieved by processing a confusing group as a single unit rather than processing individual characters separately within a group. In such a case, final acceptance of a confusing character will be determined only after processing all the members in the confusing group.

The testing alphabet constructed from an LS tensor of the same font and the size performs fairly well for slightly different fonts. The behaviour of the proposed method for highly different fonts will be investigated closely and methods such as geometric correction to deal with such fonts will be explored. Construction of a larger database of the alphabet, and the methods to generate dynamic relative threshold for separation will be investigated in the next stage. Application of properties characteristic to the Sinhala script (e.g. a vowel is used only as the first character of a word) and the statistical data such as the frequency of characters in a typical script will be considered. The possibility of word-level recognition using the on-line dictionary, in order to deal with false rejections will also be explored.

Acknowledgement: Support of SIDA (Swedish International Agency for Development Co-operation) is gratefully acknowledged.

REFERENCES

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

[2] Nucharee Premchaiswadi, Wichian Premchaiswadi and Seinosuhe Narita, Segmentation of Horizontal and Vertical Touching Thai Characters. pp 987-995, ITC-CSCC'99.

[3] .S.M.S.Rajasekaran and B.L.Dekshatulu, Recognition of Printed Telugu Characters. Computer Graphics Image Processing 6, 1977

[4] G.Siromoney, R.Chandrasekaran and M. Chand-rasekaran. Machine Recognition of

Printed Tamil Characters, Pattern recognition 10, 1978.
[5] A.K.Dutta, A Generalised Formal Approach for Description and Analysis for Major Indian Scripts. J.IETE 30, 1984.
[6] S.N. Srihari, High-Performance Reading Machines, Proceedings of the IEEE, 80(7), July 1992, 1120-1132.
[7] Rene Sennhauser, Improving the recognition accuracy of text recognition systems using typographical constraints, Electronic Publishing, Vol. 6(3), pp273-282, Sept. 1993.
[8] Y. Wang, I. Phillips and R. Haralick, Statistical-based Approach to Word Segmentation, pp 555-558, ICDAR'2001.
[9] P.J.Burt and E.H.Adelson, The Laplacian Pyramid as a Compact Image Code. IEE Trans. COMM, 31:532-540, 1983.
[10] G.S. Lehal and Chandan Singh, A Gurmuki Script Recognition System, pp 557-560, ICDAR'00.
[11] Abhujit Dutta, Santanu Chaudhury, Bengali Alpha-Numeric Character Recognition using Curvature Features, Pattern Recognition Vol 26 No. 12,pp1757-1770.
[12] Scott D. Connel , R.M.K. Sinha and Anil K Jain, Recognition of Unconstrained Devanagari Characters, pp 368-371, ICDAR'00.
[13] Yi-Kai Chen, Jhing-Fa Wang, Skew Detection and Reconstruction Based on Maximisation of Variance of Transition-Counts, pp 195-208,Pattern Recognition, 33 (2000).
[14] C.L.Tang and Z.heng Zhang, Text Block Segmentation using Pyramid Structure, pp 297-306, SPIE Document Recognition and Retrieval VIII.
[15] Y.Zhou and C.L.Tang, Hough Technique for Bar-chart Detection and Recognition in Document Images, pp 605-608, ICIP 2000.
[16] Ching Y. Suen, Jinho Kim, Kyekyung Kim, Qizhi Xu and Louisa Lam, Handwriting Recognition - Last Frontiers, pp 1-10, ICIP 2000.
[17] Bernd Jähne, Digital Image Processing – Concepts, Algorithms and Scientific Applications, Springer, 1997.

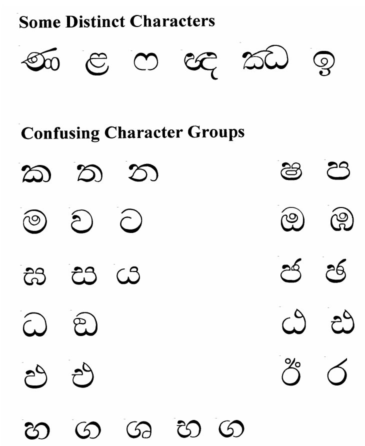


Figure 1. Distinct and similar shapes of characters

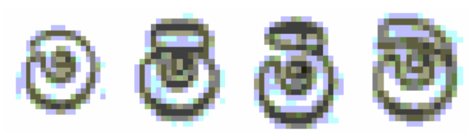


Figure 2. Violation of modification rule by changing the shape of a character

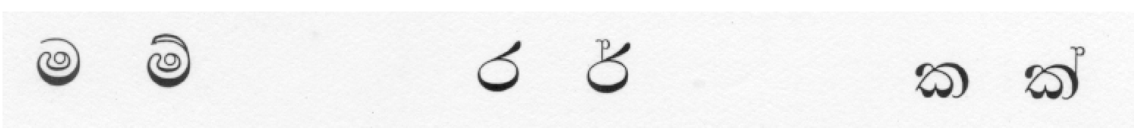


Figure 3(a). Same phonetic sound but two different modifier symbols

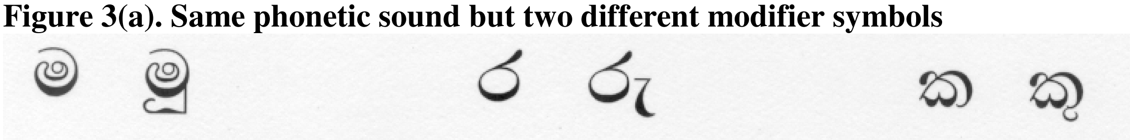


Figure 3(b). Same phonetic sound but three different modifier symbols

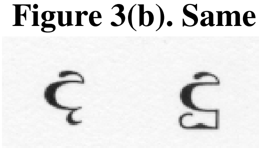


Figure 3(c). Modification by changing the original shape of character

Figure 3(d). Modofier symbol placed inside the frame of the basic character

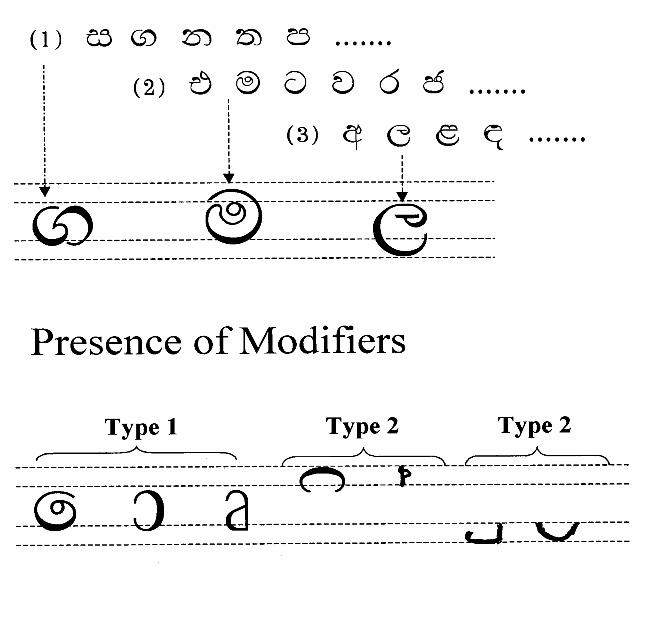


Figure 4. Three-layered structure of a line

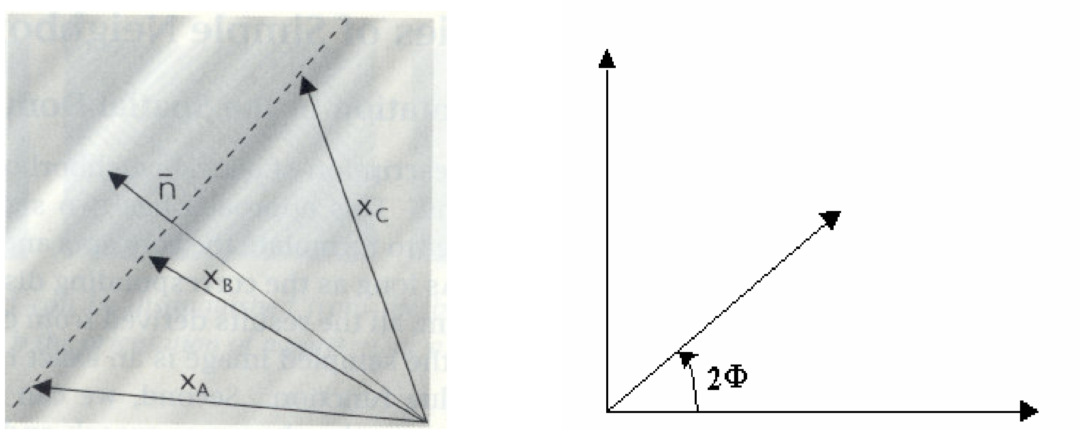
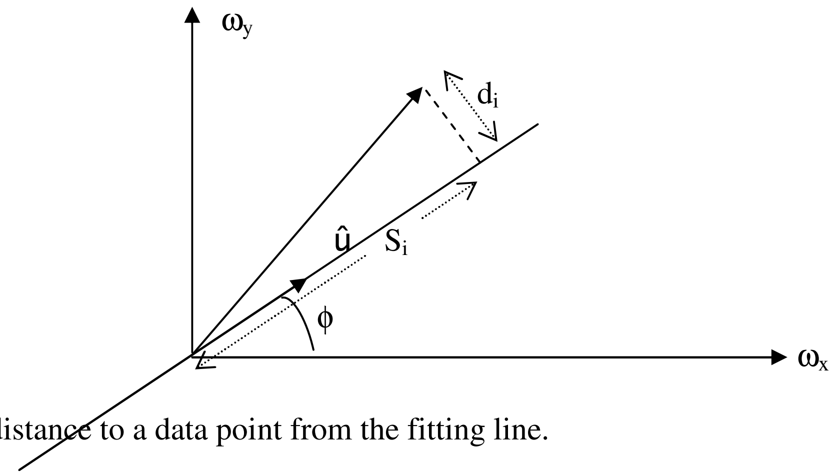
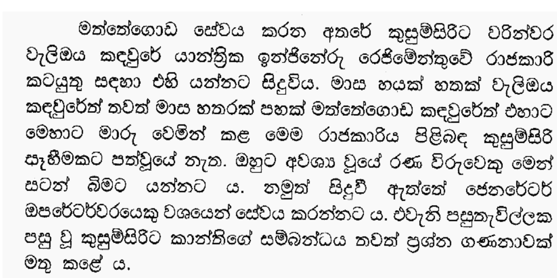


Figure 5(a) Representation of linear symmetry (left) [17]**, Vector representation (right)**



d_i – distance to a data point from the fitting line.

Figure 6. Representation of linear symmetry in the frequency domain



මත්තේගොඩ සේවය කරන අතරේ කුසුම්සිරිට වරින්වර වැලිඔය කඳවුරේ යාන්ත්‍රික ඉන්ජිනේරු රෙජිමේන්තුවේ රාජකාරි කටයුතු සඳහා එහි යන්නට සිදුවිය. මාස හයක් හතක් වැලිඔය කඳවුරේත් තවත් මාස හතරක් පහක් මත්තේගොඩ කඳවුරේත් එහාට මෙහාට මාරු වෙමින් කළ මෙම රාජකාරිය පිළිබඳ කුසුම්සිරි සෑහීමකට පත්වූයේ නැත. ඔහුට අවශ්‍ය වූයේ රණ විරුවෙකු මෙන් සටන් බිමට යන්නට ය. නමුත් සිදුවී ඇත්තේ ජෙනරේටර් ඔපරේටර්වරයෙකු වශයෙන් සේවය කරන්නට ය. එවැනි පසුතැවිල්ලක පසු වූ කුසුම්සිරිට කාන්තිගේ සම්බන්ධය තවත් ප්‍රශ්න ගණනාවක් මතු කළේ ය.

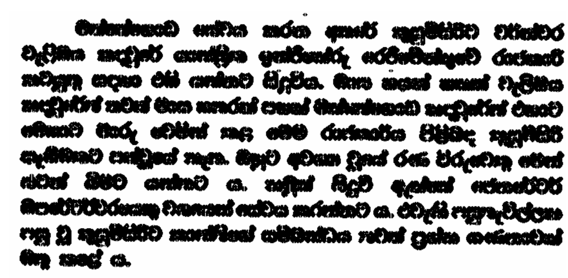


Figure 7. Original image and Linear Symmetry tensor

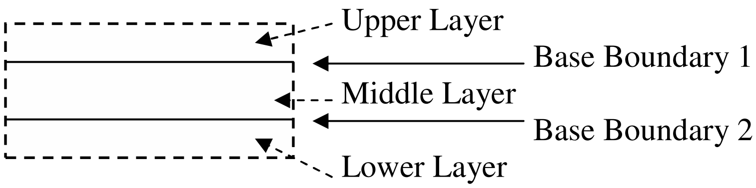
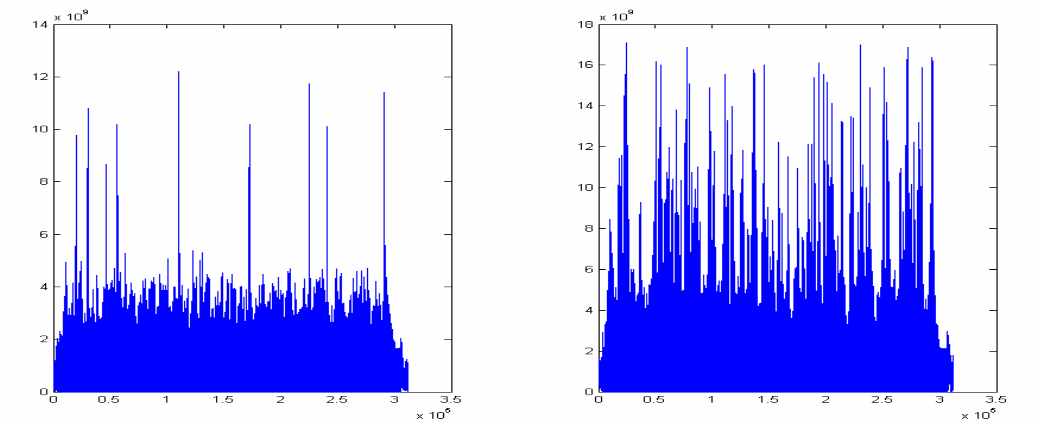


Figure 8. Zone boundaries

Figure 9. Stages in De-skewing



(a) A character with a unique shape (b) A character confusing with similar shapes
Figure 10. Correlation of a character with the script

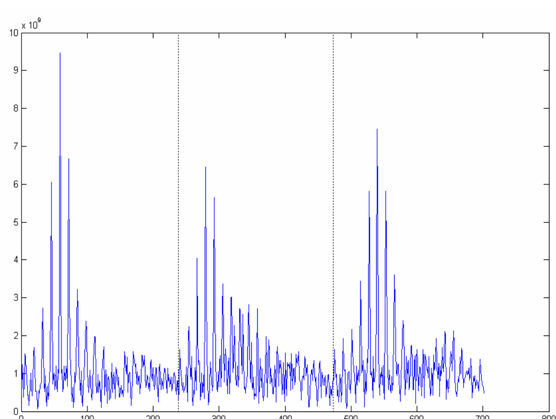
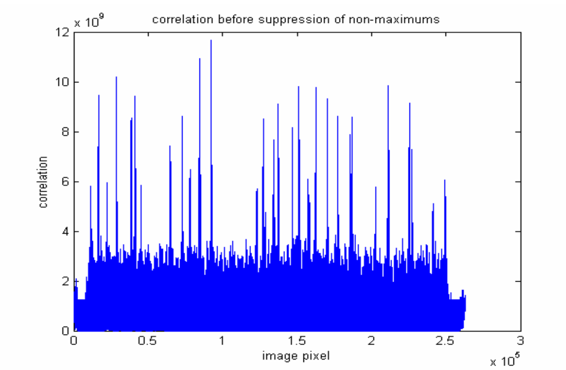


Figure 11. Correlation of a distinct segment from one character with two similar characters



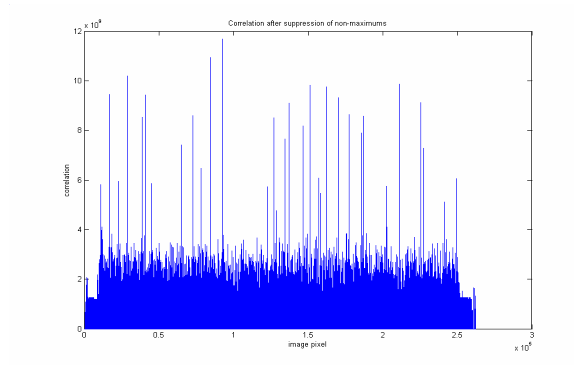


Figure 12. Suppression of non-maxima (before and after)

Table 1. Experimental Results - Determination of Skew Angle

Skew Angle	Experimental Value	Rounded Value	Percentage Error
17 (R)	17.6043	18	5.9
38 (R)	37.8446	38	0
57 (R)	57.2716	57	0
7 (R)	7.2683	7	0
4 (R)	3.7416	4	0
85 (R)	84.4144	84	1.2
27 (L)	25.8401	26	3.8
63 (L)	62.9361	63	0
16 (L)	15.0740	15	6.2
3 (L)	3.0030	3	0
6 (L)	6.0457	6	0
17 (L)	17.8092	18	5.9

(R) - Right Skew , (L) - Left Skew

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{Cos}\phi & \mathrm{Sin}\phi \\ -\mathrm{Sin}\phi & \mathrm{Cos}\phi \end{bmatrix}}_{u^T} \underbrace{\begin{bmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{bmatrix}}_{J} \underbrace{\begin{bmatrix} \mathrm{Cos}\phi & -\mathrm{Sin}\phi \\ \mathrm{Sin}\phi & \mathrm{Cos}\phi \end{bmatrix}}_{u} \quad \text{——— (1)}$$

Equation (1)

$$\min_{\phi} \iint d_i^2(\omega)d\omega_x d\omega_y = \max_{\phi} \iint S_i^2(\omega)d\omega_x d\omega_y$$

$$= \max_{\|u\|=1} u^T \left[\underbrace{\iint \begin{bmatrix} \omega_x^2 & \omega_x\omega_y \\ \omega_x\omega_y & \omega_y^2 \end{bmatrix} d\omega_x d\omega_y}_{J} \right] u \quad \text{——— (2)}$$

Equation (2)