

# Lexicon-based Offline Recognition of Amharic Words in Unconstrained Handwritten Text

Yaregal Assabie and Josef Bigun

*School of Information Science, Computer and Electrical Engineering  
Halmstad University, Halmstad, Sweden  
{yaregal.assabie, josef.bigun}@hh.se*

## Abstract

*This paper describes an offline handwriting recognition system for Amharic words based on lexicon. The system computes direction fields of scanned handwritten documents, from which pseudo-characters are segmented. The pseudo-characters are organized based on their proximity and direction to form text lines. Words are then segmented by analyzing the relative gap between subsequent pseudo-characters in text lines. For each segmented word image, the structural characteristics of pseudo-characters are syntactically analyzed to predict a set of plausible characters forming the word. The most likelihood word is finally selected among candidates by matching against the lexicon. The system is tested by a database of unconstrained handwritten Amharic documents collected from various sources. The lexicon is prepared from words appearing in the collected database. Experimental results are reported.*

## 1. Introduction

Amharic is the official language of Ethiopia, currently having a population of over 80 million. The language is believed to be evolved from Geez, and today Amharic has become the second most widely spoken Semitic language in the world, next to Arabic. Along with dozens of other Ethiopian languages, Amharic uses Ethiopic script for writing. The Ethiopic script used by Amharic has 265 characters including 27 labialized (characters mostly representing two sounds, e.g. ሀ for ሀሀ) and 34 base characters with six orders representing derived vocal sounds of the base character. The alphabet is written in a tabular format of seven columns where the first column represents the base characters and others represent their derived vocal sounds. Part of the alphabet is shown in Table 1.

**Table 1.** Part of the Ethiopic alphabet.

	Base Sound	Orders					
		(ä)	(u)	(i)	(a)	(e)	(o)
1	h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ
2	l	ለ	ሉ	ሊ	ላ	ላ፡	ላ፡፡
3	h	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
34	v	ቨ	ቩ	ቪ	ቫ	ቬ	ቭ

Recognition of handwritten text has been studied for scripts such as Latin, Chinese, Arabic, etc., and several approaches have been proposed, with the most common techniques being neural networks, HMM, elastic matching, stroke analysis, and combinations of multiple classifiers [3]. The trend is now to use language models such as lexicon and part-of-speech tagger to improve the results [4], [5]. In this work, we use structural and syntactic information of segmented words as a basis for recognition of handwritten Ethiopic text with the support of Amharic lexicon to optimize their recognition.

## 2. The recognition system

The proposed handwritten Amharic word recognition system analyses sequences of primitive strokes in pseudo-characters of segmented words to generate possible combinations of characters. Each character is generated by matching pseudo-characters against a knowledge base which stores possibly occurring sequences of primitives and their spatial relationships for Ethiopic characters. Recognition of words is finally achieved by selecting the most likelihood combination of characters with respect to Amharic lexicon. The recognition system is illustrated in Fig. 1, where dotted line boxes are iterative tasks, and the details are presented below.

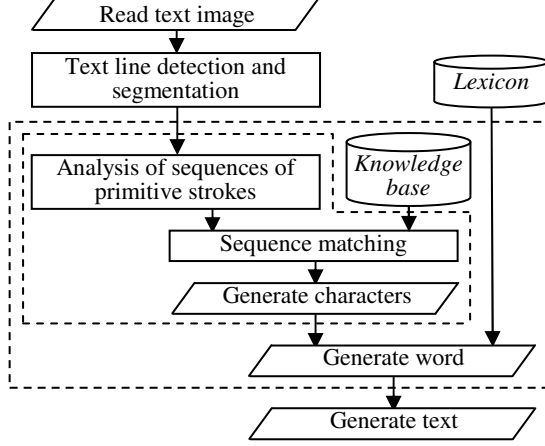


Figure 1. Amharic word recognition system.

### 3. Text line detection and segmentation

The recognition system requires text lines, words and pseudo-characters to be segmented for analysis. We developed an algorithm for such segmentation tasks using direction field image. Pseudo-characters represent two or more physically connected characters, but hereafter we simply refer to them as characters.

#### 3.1 Computation of direction field image

Direction field tensor  $S$  is a  $2 \times 2$  matrix which computes the optimal direction of pixels in a local neighborhood of an image  $f$  [2]. It is computed as:

$$S = \begin{pmatrix} \iint (D_x f)^2 dx dy & \iint (D_x f)(D_y f) dx dy \\ \iint (D_x f)(D_y f) dx dy & \iint (D_y f)^2 dx dy \end{pmatrix} \quad (1)$$

The integrals are implemented as convolutions with a Gaussian kernel, and  $D_x$  and  $D_y$  are derivative operators. The local direction vector is the most significant eigenvector modulated by the error differences (the difference of eigenvalues). This vector field is also known as the *linear symmetry* (LS) vector field and can be obtained directly by use of complex moments. The latter are defined as:

$$I_{mn} = \iint ((D_x + iD_y)f)^m ((D_x - iD_y)f)^n dx dy \quad (2)$$

where  $m$  and  $n$  are non-negative integers. Among other orders, of interest to us are  $I_{10}$ ,  $I_{11}$ , and  $I_{20}$  derived as:

$$I_{10} = \iint ((D_x + iD_y)f) dx dy \quad (3)$$

$$I_{11} = \iint |(D_x + iD_y)f|^2 dx dy \quad (4)$$

$$I_{20} = \iint ((D_x + iD_y)f)^2 dx dy \quad (5)$$

In a local neighborhood of an image,  $I_{10}$  computes the ordinary gradient field;  $I_{11}$  measures gray value changes (the sum of eigenvalues of  $S$ ); and  $I_{20}$  gives a complex value where its argument is the optimal direction of pixels in double angle representation and

its magnitude is the local LS strength (the difference of eigenvalues of  $S$ ). Pixels with low magnitude are said to be lacking LS property. As shown in Fig. 2,  $I_{10}$  and  $I_{20}$  images can be displayed in color where the hue represents direction of pixels with the red color corresponding to the direction of zero degree.

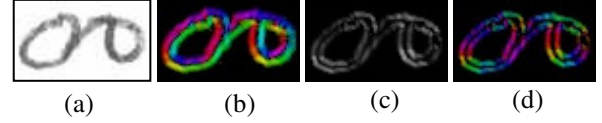


Figure 2. (a) Handwritten መ, (b)  $I_{10}$ , (c)  $I_{11}$ , (d)  $I_{20}$  of a.

#### 3.2 The segmentation process

Segmentation and text line detection is done on the direction field image ( $I_{20}$ ) in two passes. In the first pass, the image is traversed from top to down and pixels are grouped into two as *blocked* (character) and *open* (background) regions. A pixel is recursively classified as open if it:

- is in the first row of the direction field image,
- lacks LS and one of its immediate top and/or sideways neighborhood is open.

The remaining are grouped as blocked pixels. In the second pass, the  $I_{20}$  image is traversed from left to right grouping each segmented character into appropriate text lines based on character's proximity along *global* (average direction of the text line) and *local* (direction at the head of the text line) directions. Segmented characters that do not fit into the existing text lines form a new text line. The directions of a text line help to predict the direction in which the next member character is found during traversal, which is essential especially in skewed documents and non-straight text lines. Figure 3 shows segmentation and text line detection for handwritten Amharic text skewed by  $15^\circ$ . Words are then segmented based on the relative gap  $R$  between characters within a text line, defined as  $R_i = G_i - G_{i-1}$ , where  $G_i$  is the horizontal gap between the  $i^{\text{th}}$  character and its predecessor. Although the horizontal gap between consecutive characters varies greatly, the relative gap suppresses those variations and a threshold segments words fairly well.

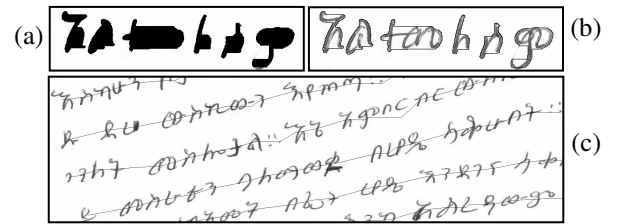


Figure 3. (a) Character regions separated from the background, (b) Character segmentation mapped onto the original text (c) text line detection.

## 4. Word recognition

### 4.1. Analysis of primitive strokes

We use a set of primitive strokes and connectors as a basis for recognition of characters and words. Primitives in handwritten Ethiopic text are formed from vertical and diagonal lines and end points of horizontal lines, whereas connectors are defined as horizontal lines between two primitives. Primitives are further classified hierarchically based on their orientation/ structure type, relative length with in the character, and relative spatial position. This classification scheme results in 15 types of primitives, each of which are assigned with three-digit numbers (each ranging from 6 to 9) where the digits represent orientation/structure type, relative length, and spatial position of primitives, respectively. Details of the classification are presented in [1] and summarized in Table 2 below.

**Table 2.** Hierarchical classification of primitives.

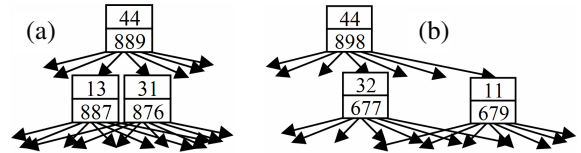
Orientation/ Structure	Length	Position	Code	Example Character
Vertical	Long	Top-to-bottom	898	ባ
	Medium	Top	889	ፈ
		Bottom	887	ሰ
	Short	Middle	876	ሳ
Forward Slash	Long	Top-to-bottom	998	ሠ
	Medium	Top	989	ረ
		Bottom	987	ረ
	Short	Middle	976	ረ
Backslash	Long	Top-to-bottom	798	ሰ
	Medium	Top	789	ሰ
		Bottom	787	ሰ
	Short	Middle	776	ሰ
Appendage	Short	Top	679	ፈ
		Middle	676	ፈ
		Bottom	677	ሰ

Pixels are grouped in to parts of primitives and connectors based on their optimal direction in the  $I_{20}$  image. After converting the double angle of  $I_{20}$  into a simple angle representation, pixels having LS properties and directions  $[0..60]$  degrees are considered as parts of primitives and those with directions  $(60..90]$  degrees are considered as parts of connectors. The extracted linear structures in the  $I_{20}$  image are mapped onto the  $I_{10}$  image to classify them into left and right edges of primitives. A primitive is then formed from the matching left and right edges. Primitives are then further classified using their direction, relative length, and spatial position.

### 4.2. Spatial relationships of primitives

Spatial relationship refers to the way primitives are connected to each other with horizontal lines. A primitive can be connected to another at one or more of the following regions of the strokes: *top* (1), *middle* (2), and *bottom* (3). The number 4 is used for cases where there is no connection. A connection between two primitives is represented by  $xy$  where  $x$  and  $y$  are numbers representing connection regions for the *left* and *right* primitives, respectively. Between two primitives of Ethiopic characters, there can also be two or three connections, and a total of 18 connection types are identified, which are listed as: 11 (ባ), 12 (ሰ), 13 (ረ), 21 (ሰ), 22 (ሰ), 23 (ሰ), 31 (ሰ), 32 (ሰ), 33 (ሰ), 1123 (ሰ), 1132 (ሰ), 1133 (ሰ), 1232 (ሰ), 2123 (ሰ), 2132 (ሰ), 2133 (ሰ), 112232 (ሰ), and 112233 (ሰ).

The spatial relationships of primitives in a character are modeled by a special tree structure called *primitive tree*. Each node in the primitive tree possibly has seven child nodes arranged in the order of **left** {top, middle, bottom} and **right** {bottom, middle1, middle2, top} nodes. Child nodes represent primitives connected to their parent primitive at different connection regions. They store information about primitive type and connection type with their parent primitive. The left top primitive is taken as a root node, and other primitives are recursively built into the tree based on their interconnections. Examples of primitive trees are shown in Fig. 4.



**Figure 4.** Primitive trees for characters (a) ሰ and (b) ረ.

### 4.3. Character Recognition

Since primitive trees are costly in terms of computation, we convert them into string data by traversing in the order of {**left**{top, middle, bottom}, **parent**, **right** {bottom, middle1, middle2, top}}, where a unique sequence of primitives and their spatial relationships is generated for each primitive tree. Recognition of unknown input is then achieved by finding the best match (with similarity above a threshold) of its primitive sequence against a knowledge base. The knowledge base stores possibly occurring sequences of primitives and their spatial relationships for each Ethiopic character. The recognition process is further exposed in detail in [1].

#### 4.4. Lexical support

We assume that a word is a sequence of pseudo-characters called sub-words; a sub-word is a sequence of characters; and a character is a sequence of primitives and their spatial relationships. For each sub-word, a set of valid character sequences are generated by iteratively analyzing the sequences of primitives and their spatial relationships. As shown in Fig. 5, for example, the primitives of the sub-word ነቢረቸ is analyzed to form valid sets of character sequences as: {ነቢረቸ (ነቢረቸ), ነቢረቸ (ነቢረቸ), ነቢረቸ (ከገረቸ), ነቢረቸ (ከገረቸ), and ነቢረቸ (ከሀረቸ)}. In this iteration process, a word space  $\mathbf{W}$  is generated which is represented as:  $\mathbf{W} = \{\{S_{11}, S_{12}, S_{13}, \dots\}, \{S_{21}, S_{22}, S_{23}, \dots\}, \dots, \{S_{s1}, S_{s2}, S_{s3}, \dots\}\}$  where  $S_{ij}$  is the  $j^{\text{th}}$  character sequence in the  $i^{\text{th}}$  sub-word of  $\mathbf{W}$ . Then, candidate words are formed from the possible combinations of a sequence of characters from each sub-word. Now, our goal is to find a word  $\omega = \{S_{1i}, S_{2j}, \dots, S_{sp}\}$ , where  $p$  is the  $p^{\text{th}}$  character sequence in sub-word  $s$ , with  $\omega$  having an optimal *confidence value*. The confidence value of candidate words is computed as the average of *character similarity* of individual characters making up the candidate word and its *lexical similarity*. Character similarity measures how well the sequences of primitives are similar to that of the predicted characters in the knowledge base, and lexical similarity measures the similarity between candidate words and their nearest word in the lexicon. In Fig. 5, bold arrows show paths of sequences of characters forming optimal confidence value for the handwritten word.

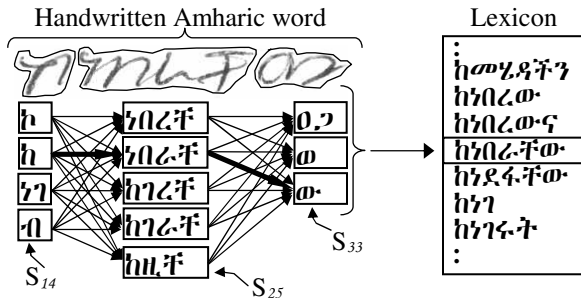


Figure 5. Word recognition network.

#### 5. Experiment

Amharic recognition of is one of the least studied areas and we were not able to get resources such as lexicon and handwriting databases available for research. Thus, we developed a database collected from 177 writers. The writers were provided with Amharic documents dealing with various real-life issues and they used ordinary pen and white papers for

writing. A total of 307 pages were collected and scanned at a resolution of 300dpi, from which we extracted 10,932 distinct words to build the lexicon. The lexicon is organized as groups of flat data each sorted by the  $i^{\text{th}}$  character of words to optimize searching. We then use binary searching algorithm to find a word containing a character  $c$  at its  $i^{\text{th}}$  position, in which the worst case complexity becomes  $O(\log_2^n)$ .

For filtering operations of scanned texts, we used a symmetric Gaussian of 3x3 pixels. Word recognition result varies greatly on the quality of the handwriting. For good quality texts (words properly separated and characters not connected to each other), we achieved a recognition rate of 73% (top-1 choice) and 87% (top-5 choices). For poor quality texts, the recognition rate drops to 36% (top-1 choice) and 58% (top-5 choices). The errors arise mainly from word segmentation and recognition of interconnected characters.

#### 6. Discussion and conclusion

We presented Amharic word recognition based on lexical support. We also proposed script-independent text line detection, and character and word segmentation algorithms based on the direction field image. The lexicon and database we developed can be used as a benchmark resource for further studies on recognition of Ethiopic script. Since we are encoding the relative size of primitives and pattern matching is based only on a knowledge base and lexicon, our recognition system does not require size normalization and training of characters or words. The recognition result can be improved by further employing statistical tools such as HMMs and neural networks. The system can be enhanced to sentence level recognition provided that the required language resources (e.g., part-of-speech tagger) are made available. It can also be further applied to other Ethiopian languages.

#### References

- [1] Y. Assabie and J. Bigun, "Writer-independent offline recognition of handwritten Ethiopic characters", In: *Proc. 11<sup>th</sup> ICFHR*, August 19-21, Montreal, 2008.
- [2] J. Bigun, *Vision with Direction: A Systematic Introduction to Image Processing and Vision*. Springer, Heidelberg, 2006.
- [3] H. Bunke, "Recognition of cursive Roman handwriting: past, present and future", In: *Proc. 7<sup>th</sup> ICDAR*, Edinburgh, Scotland, pp. 448-459, 2003.
- [4] A. L. Koerich, R. Sabourin, C. Y. Suen, "Large vocabulary offline handwriting recognition: A survey", *Pattern Anal. Applic.* 6(2): pp. 97-121, 2003.
- [5] R. Plamondon, S.N. Srihari, "On-line and off-line hand writing recognition: A comprehensive survey", *IEEE Trans. PAMI*, 22(1): pp. 63-84, 2000.