Ethiopic Character Recognition Using Direction Field Tensor

Yaregal Assabie and Josef Bigun

Assabie Y, Bigun J. Ethiopic Character Recognition Using Direction Field Tensor. In: 18th International Conference on Pattern Recognition, 2006. ICPR 2006. IEEE; 2006. p. 284-287.

DOI: http://dx.doi.org/10.1109/ICPR.2006.507
Copyright: IEEE

Ethiopic Character Recognition Using Direction Field Tensor

Yaregal Assabie and Josef Bigun
School of Information Science, Computer and Electrical Engineering
Halmstad University, SE-301 18 Halmstad, Sweden
{Yaregal.Assabie, Josef.Bigun}@ide.hh.se

Abstract

Many languages in Ethiopia use a unique alphabet called Ethiopic for writing. However, there is no OCR system developed to date. In an effort to develop automatic recognition of Ethiopic script, a novel system is designed by applying structural and syntactic techniques. The recognition system is developed by extracting primitive structural features and their spatial relationships. A special tree structure is used to represent the spatial relationship of primitive structures. For each character, a unique string pattern is generated from the tree and recognition is achieved by matching the string against a stored knowledge base of the alphabet. To implement the recognition system, we use direction field tensor as a tool for character segmentation, and extraction of structural features and their spatial relationships. Experimental results are reported.

1. Introduction

The history of Ethiopian written literature dates back to the 5th century B.C.[4]. Ethiopic alphabet is used for writing by languages such as Amharic, Geez, Tigrinya, and others. The alphabet is used by multiple languages spoken inside as well as outside of Ethiopia, and the current estimate of its users is over 80 million which include the 73 million inhabitants of Ethiopia.
Studies of automatic recognition of Ethiopic script have been presented in conferences only recently [3], [5], and no Ethiopic OCR software is developed to date. The scarce results are linked to the increased relational complexity of graphical elements when representing vocals. Therefore, we suggest a generic recognition system for Ethiopic characters by extracting the smallest graphical units called primitive structures and their spatial relationships. A major advantage is, for a given character in Ethiopic alphabet, the primitives and their spatial relationships remain similar for different font types, styles, and sizes. The efficiency of the recognition system is highly dependent on the computational effectiveness of the algorithms used for extraction and classification of structural features. We use direction field tensor [2] to efficiently segment characters, extract and classify structural features, and build their spatial relationships.

2. Ethiopic Character Recognition System

The most common Ethiopic alphabet has 34 basic characters and six modifications making a total of 238 characters. The alphabet is conveniently written in a tabular format of seven columns as shown in Table 1 where each column corresponds to vocal sounds in the order of *ä*, *u*, *i*, *a*, *e*, *ə*, and *o*.

Table 1. Part of the Ethiopic alphabet

ä order	**u** order	**i** order	**a** order	**e** order	**ə** order	**o** order
ሀ hä	ሁ hu	ሂ hi	ሃ ha	ሄ he	ህ hə	ሆ ho
ለ lä	ሉ lu	ሊ li	ላ la	ሌ le	ል lə	ሎ lo
ሐ hä	ሑ hu	ሒ hi	ሓ ha	ሔ he	ሕ hə	ሖ ho
⋮	⋮	⋮	⋮	⋮	⋮	⋮

The nature of prominent structural features in Ethiopic characters gives a way to employ structural and syntactic techniques to efficiently design a generic recognition system that invariably works for different font types, sizes and styles. Primitive structures and their spatial relationships for each character are handled by a special tree structure which, by traversing with a defined rule, generates a unique set of string patterns for each character. The string pattern for a character is matched with each record in a stored knowledge base of the alphabet, and a similarity measure is computed for each comparison. The recognition process is completed by accepting or rejecting the most similar pattern based on a threshold value of similarity. In the next subsections, we describe components of the recognition process. Further details are given in [1].

2.1. Primitive Structures in the Alphabet

An in-depth analysis of characters leads us to discover 7 primitive structures in the Ethiopic alphabet.

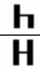
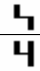
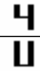
The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

All characters can be represented as a combination of two or more primitive structures. Primitive structures are identified using their structure type, relative length, orientation, and spatial position. The classes of primitive structures (with example characters) are: *long vertical line* (ዘ), *medium vertical line*(ሰ), *short vertical line* (ሐ), *long forward slash* (ጊ), *medium forward slash* (ፋ), *backslash* (ሌ), and *appendages* (ፒ).

2.2. Spatial Relationships of Primitives

The way primitives are connected to each other describes their spatial relationship, and gives unique structures to characters. A primitive structure can be connected to another at one or more of the following regions: *top* (t), *middle* (m), and *bottom* (b). There may be one, two or three connections between two primitives. The first connection detected as one goes from top to bottom is considered as the *principal connection*. There are 9 principal connection types between two primitives. This is represented by the set: {(t,t), (t,m), (t,b), (m,t), (m,m), (m,b), (b,t), (b,m), (b,b)}. Other additional connections, if they exist, are considered as *supplementary connections*. Together with supplementary connections, there are 18 types of connections between primitives of Ethiopic alphabet. This finding is summarized in Table 2.

Table 2. Connection types between primitives

Principal Connection	Supplementary Connections					
	None	(m,b)	(b,m)	(b,b)	(m,m)+(b,m)	(m,m)+(b,b)
(t,t)						
(t,m)						
(t,b)						
(m,t)						
(m,m)						
(m,b)						
(b,t)						
(b,m)						
(b,b)						

2.3. Representation and Pattern Generation

By analyzing how primitives are connected to each other, a special tree structure is designed which effectively represents, stores, and retrieves primitives and their spatial relationships [1]. The tree has three left nodes and four right nodes as shown in Fig.1. Each node in the tree stores data about the type of the primitive itself, the type of connections with its parent primitive, and the types of primitives connected to it.

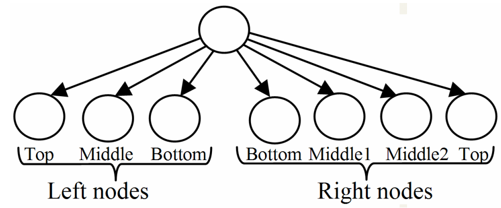


Figure 1. General tree structure of characters

To build a consistent primitive tree structure for each character, a primitive which is spatially located at the left top position of the character is used as the root primitive. Based on principal connections with their parent, child primitives are then appended in the order of {*left*{t,m,b}, *right*{b,m1,m2,t}}. To implement this rule, a recursive algorithm is developed [1]. Examples of primitive tree of characters are shown in Fig. 2.

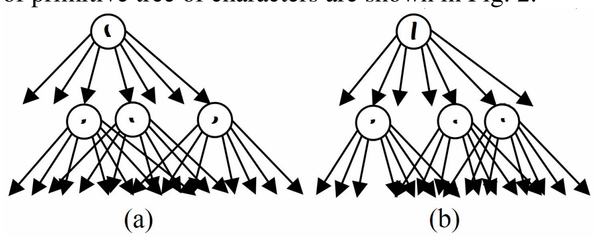


Figure 2. Tree structure for (a) ጺ , (b) ቲ

A pattern of primitives and their relationships is generated from the tree by using a recursive algorithm similar to the one used for building the tree. *In-order traversal* of the tree, i.e., traversing the tree in the order of {left{t,m,b}, parent, right{b,m1,m2,t}}, generates a unique pattern for each primitive tree of characters.

2.4. Alphabet Knowledge Base

For each character in the alphabet, a pattern of their possibly occurring primitives and relationships is stored. In the implementation, a numerical code is given to the 7 primitives and 18 connection types [1]. Each character in the knowledge base is represented by a set of patterns built based on this coding.

3. Direction Field Tensor

The ordinary way of estimating the local direction of pixels in an image neighborhood is to compute the gradient field. However, during the computation of the gradient field, it is not possible to analyze whether a pixel fits to a line or not. Thus, the results obtained from the gradient field are not optimal solutions to extract linear patterns in images. In addition, a linear structure will have two groups of opposite directions in the gradient field, and averaging them over a window gives large deviations due to cancellation effects.

A double angle representation of the direction of pixels avoids the problem of averaging the resultant

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

gradient field. Therefore, to take the advantage of this property, we use direction field tensor. Direction field tensor also suppresses image intensity differences that do not fit to a line in a local neighborhood and amplifies those that fit to a line. Since the directional features are observed along lines, the local direction is also called Linear Symmetry. For a local neighborhood *f(x,y)* of an image *f*, the direction tensor, also called the structure tensor S, is computed as a 2x2 symmetric matrix using derivative operators D_x and D_y.

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

The complex partial derivative operator $D_x + iD_y$ is defined as follows.

$$D_x + iD_y = \frac{\partial}{\partial x} + i\frac{\partial}{\partial y} \quad \textbf{(2)}$$

The integrals are implemented as convolutions with a Gaussian kernel:

$$g(x, y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad \textbf{(3)}$$

where σ is the standard deviation. Because of its separability property, the 2D Gaussian is more efficiently computed as a convolution of two 1D Gaussians, *g(x)* and *g(y)*, where *g* is defined as follows.

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right) \quad \textbf{(4)}$$

Linear Symmetry (LS) measures can be extracted by eigenvalue analysis of the structure tensor using complex moments of order two formulated as:

$$I_{mn} = \iint ((D_x + iD_y)f)^m ((D_x - iD_y)f)^n \, dxdy \quad \textbf{(5)}$$

where *m* and *n* are non-negative integers. From Equation (5), we can derive I_{20} and I_{11}.

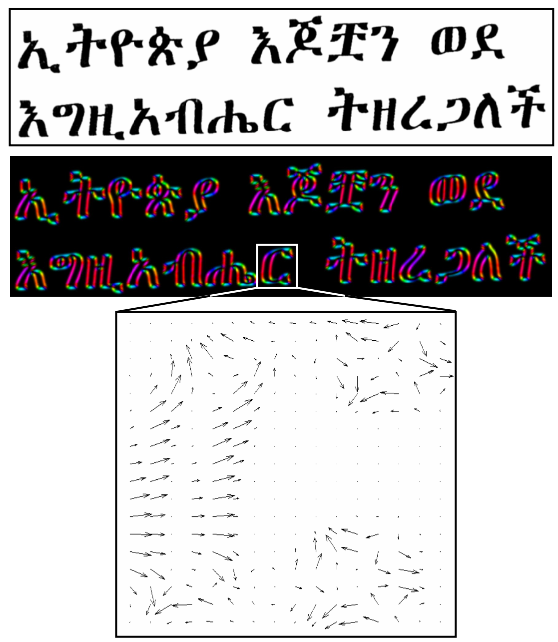
$$I_{20} = \iint ((D_x + iD_y)f)^2 \, dxdy \quad \textbf{(6)}$$

$$I_{11} = \iint \left|(D_x + iD_y)f\right|^2 dxdy \quad \textbf{(7)}$$

I_{20} is a complex number and I_{11} is a real number. The argument of I_{20} is the double angle representation of the local direction of pixels and the magnitude is a measure of the local LS strength. The scalar I_{11} measures the amount of gray value changes in a local neighborhood of pixels. The direction field, which is a 3D field tensor, can be represented as one complex number (I_{20}) and one real number (I_{11}). In the implementation, I_{20} and I_{11} are computed over the entire image by using the following the procedure.

1. Generate two 1D Gaussian kernels *gx* and *gy* and two 1D Gaussian derivative kernels *dx* and *dy*.
2. Apply convolution operations on the original image *f* to generate *dxf* and *dyf* .
   dxf = gy(dx*f)*
   dyf = gx(dy*f)*
3. Compute:
   a. $\hat{I}_{20}$ from *dxf* and *dyf* by pixel-wise complex squaring. $\hat{I}_{20} = (dxf + j*dyf)^2$ where $j=\sqrt{(-1)}$
   b. $\hat{I}_{11}$ from *abs*($\hat{I}_{20}$).
4. Compute I_{20} and I_{11} from $\hat{I}_{20}$ and $\hat{I}_{11}$, respectively by averaging with Gaussian kernels.

The complex image I_{20} can be displayed in color as shown in Fig. 3b where the hue represents direction of pixels in double angle representation. Pixels with directions of zero are shown as red color. The black area shows 'lack of LS'. Another way of displaying the I20 image is to make use of vectorial representation as shown in Fig. 3c where the arrows show direction and the magnitude shows the LS strength of pixels.



(a)

(b)

(c)

Figure 3. (a) Text image, (b) I_{20} of the text image, (c) vectorial representation of I_{20} image

3.1. Application

The application of direction field tensor for Ethiopic character recognition system is described in Fig. 4. Due to the double angle representation, the direction of pixels at the edges of primitives is close to 0 and 180 degrees and can be converted to the range of 0 to 90 degrees by $\varepsilon = abs(90-\theta)$ so that ε for primitives is consistently close to 90 degree. The value ρ shows the normalized LS strength. Since Gaussian filtering is used during the computation of direction fields, the LS strength at the orthogonal cross-section of edges is Gaussian. Therefore, the lines in the I_{20} image are reduced to skeletal form (one pixel size) by taking the point equal or closer to the mean of the Gaussian formed by the LS strength in the orthogonal direction.

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

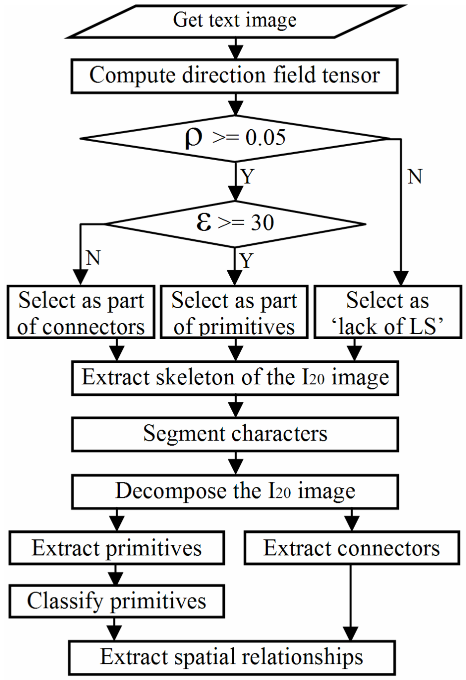


Figure 4. The application of direction field tensor

3.2. Character Segmentation

The horizontal space that lacks LS in the skeletal form of I_{20} is used to segment text lines, and the vertical space that lacks LS is used to segment characters in a line. Rectangular boxes in Fig. 5 show segmented characters for text of Fig. 3a.

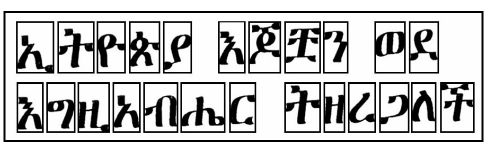


Figure 5. Character Segmentation

3.3. Extraction of Structural Features

After characters are segmented, the complex image is again decomposed into primitive and connector parts. A primitive structure will have two lines in the skeletal form of I_{20} image. Primitive structures are then formed in the I_{20} image by two matching lines as shown Fig. 6. The group information about direction of pixels in a primitive and their relative sizes, spatial positions, and structures are used to classify the primitives. Spatial relationships are finally extracted by analyzing how primitives are connected to each other.

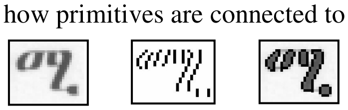


Figure 6. The character ማ, skeleton of primitives in the I_{20} image, extracted primitives

4. Experiment

A standard image database of Ethiopic text is not available for testing character recognition systems. Thus, the experiment was done on images of about thirty pages scanned from newspapers, books and clean printouts that contain characters of different fonts and sizes varying from 12 to 18. Experimental results are summarized in Table 4. The decrease in recognition accuracy of newspapers and books is linked to their degraded quality.

Table 4. Experimental Results

Type of Document	Accuracy		
	Character segmentation	Primitive Extraction	Character Recognition
Clean printouts	98%	96%	92%
Newspapers and books	93%	94%	86%

5. Conclusion and Future Work

In this paper, we presented the design of a novel Ethiopic character recognition system. The recognition system is in general tolerant of variations on the size, type and other parameters of characters. Extraction of primitive structures and their spatial relationship is the heart of the proposed recognition system, and to implement the system direction field tensor was used. Since Gaussian separable filters are used to compute direction field tensor, the computation time is minimal. Therefore, the overall research activity will help us develop efficient OCR software for Ethiopic script. The percentage of accuracy can still be improved to a higher level by working more on direction tensor and pattern matching algorithms.

References

[1] Y. Assabie and J. Bigun, "Structural and Syntactic Techniques for Recognition of Ethiopic Characters", In *Proc. Eleventh Int'l Workshop on Structural and Syntactic Pattern Recognition*, Hong Kong, 2006.

[2] J. Bigun, *Vision with Direction: A Systematic Introduction to Image Processing and Vision*, Springer, Heidelberg, 2006.

[3] J. Cowell and F. Hussain, "Amharic character recognition using a fast signature based algorithm", In *Proc. Seventh Int'l Conf. Information Visualization*, 2003, pp. 384-389.

[4] A. Gerard, *African Language Literatures: An introduction to the literary history of sub-Saharan Africa*, Three Continents Press, Washington, 1981.

[5] L. Premaratne, Y. Assabie and J. Bigun, "Recognition of modification-based scripts using direction tensors", In *Proc. ICVGIP'04*, Kolkata, 2004, pp. 587-592.

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 **IEEE**