10th Symposium on Neural Network Applications in Electrical Engineering, NEUREL-2010
Faculty of Electrical Engineering, University of Belgrade, Serbia, September 23-25, 2010

# Farsi and Latin Script Identification using Curvature Scale Space Features

Malike Khoddami and Alireza Behrad

***Abstract*—Script recognition is a necessary process before OCR algorithm in multilingual systems. In this paper, a novel method is proposed for identifying Farsi and Latin scripts in bilingual document using curvature scale space features. The proposed features are rotation and scale invariant and can be used to identify scripts with different fonts. We assumed that the bilingual scripts may have Farsi and English words and characters together; therefore the algorithm is designed to be able to recognize scripts in the connected components level. The output of the recognition is then generalized to word, line and page levels. Experimental results show that the proposed method has good accuracy especially in word and connected component levels.**

***Index Terms*—Curvature scale space, Script identification, Optical character recognition**

## I. Introduction

STORAGE of documents in digital form has too many advantages and we often prefer to work with electronic documents instead of paper documents. That is why many printed or paper document are converted to digital form. But sometimes it's necessary to retrieve required information from such digital documents. Therefore optical character recognition systems has been developed which is a technique for converting text images into computer readable/editable texts. In multilingual documents, script recognition is a main stage before applying OCR algorithms. Different methods have been presented for script identification which can be divided into two well-known groups including algorithms based on global features and algorithms based on local features. The first group is based on global features of the text such as textures. For example, in [1] co-occurrence matrix and wavelet analysis are used to extract texture features for differentiation between eight languages. In [2] and [3] recognition of scripts is performed in word-level using texture analysis through Gabor filter. The second group of script identification algorithms is based on local analysis which relies on local features and mostly acts on word and connected components levels. Features in these methods are further divided into four categories including 1- Structural or geometric features, 2- Morphological features, 3- Statistical features and 4- Spatial spread features Structural features include features such as physical size of the blocks, black pixels density, width, height, aspect ratio, area and etc [4], [5]. Morphological features are general description of text, according to script and connected components, regardless of its physical structure. The morphological features such as number of holes and upward concavities are related to inherent shape features of each script [5], [6]. Statistical features are simple mathematical approaches dealing with distribution of pixels in the character image. This set of structural features are mostly based on connected components inside the specified image blocks; such as mean and variance of the width, height, ratio and the area of connected components.

Extraction of structural features usually is performed in connected component level. If these features are extracted at higher levels such as words, lines and text blocks, they are called statistical properties. For example, instead of aspect ratio of components, the mean aspect ratio of components of a word may be used [5], [6]. Spatial spread features provide information about scattering of ascenders and descenders, regional pixels concentration and characters density. The ratio of black pixels in one region to the total number of black pixels and the number of characters per unit area are samples of these features [3], [4]. In Latin base languages that have similar texture characteristics, identification is performed using word and character shape coding. Word shape coding convert the word images into shape codes using shape characteristics [7], [8].

In this paper, we have proposed a new algorithm for Farsi and Latin texts identification. We assumed that the bilingual scripts may have Farsi and English words and characters together; therefore we used local features based on curvature scale space (CSS) representation to identify scripts in different levels. The proposed algorithms can operates in different levels. First, recognition process is applied on connected component level. At the next level the number of Latin and Farsi components in a word is considered for identification in word level. For the scripts that a complete line or page contains characters of one language, the algorithm can be extended to identify scripts in line or page levels by checking the number of recognized Farsi and Latin word in the line or the number of the recognized Farsi and Latin lines in the page. Therefore identification may be used in word, line and page levels.

The structure of this paper is as follows: in the next section we review the fundamentals of curvature scale space representation. The proposed algorithm for script identification will be described in section 3. Experimental

Alireza Behrad is with the Faculty of Engineering, Electrical Engineering Department of Shahed University Tehran, Iran. (phone: (+98-21) 51212036; e-mail: behrad@ shahed.ac.ir).

Malike Khoddami is Msc. Student of Faculty of Engineering, Shahed University Tehran, Iran (e-mail: khoddami@shahed.ac.ir).

978-1-4244-8819-3/10/$26.00 ©2010 IEEE

results are given in section 4 and conclusions appear in section 5.

## II. Curvature Scale Space Representation of Planar Curves

### A. Curvature of a Planar Curve

The curvature of a curve is defined as

$$k(s) = \lim_{h \to 0} \frac{\phi}{h} \tag{1}$$

where $\phi$ is the angle between $\mathbf{t}(s)$ and $\mathbf{t}(s + h)$ in which $\mathbf{t}$ represents the tangent vector and $s$ is the arc length parameter. Planar curves $\Gamma$ in an image can be represented as parametric vector equation as follow:

$$\Gamma = r(u) = \big(x(u), y(u)\big) \tag{2}$$

where $u$ is an arbitrary parameter. The curvature of the parametric curve in equation 2 can be expressed using the following equation [9]:

$$k(u) = \frac{\dot{x}(u)\ddot{y}(u) - \dot{y}(u)\ddot{x}(u)}{\left(\dot{x}(u)^2 + \dot{y}(u)^2\right)^{3/2}} \tag{3}$$

To make the curvature values robust against the curve noises, the evolved version of the curve is used to calculate the curvature. The evolved version of a curve is calculated by convolving the curve with a 1D Gaussian kernel of width $\sigma$ as follows:

$$Y(u,\sigma) = y(u) \otimes g(u,\sigma) \tag{4}$$

$$X(u,\sigma) = x(u) \otimes g(u,\sigma) \tag{5}$$

$$\Gamma_\sigma = R(u,\sigma) = (X(u,\sigma), Y(u,\sigma)) \tag{6}$$

where $\otimes$ represents 1D convolution. The curvature of the evolved curve is calculated using the following equation:

$$K(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{\left(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2\right)^{3/2}} \tag{7}$$

In the above equation the first and second derivatives of $X$ and $Y$ are estimated using the first and second derivative of Gaussian function as follows:

$$X_u(u,\sigma) = x(u) \otimes g_u(u,\sigma) \tag{8}$$

$$X_{uu}(u,\sigma) = x(u) \otimes g_{uu}(u,\sigma) \tag{9}$$

$$Y_u(u,\sigma) = y(u) \otimes g_u(u,\sigma) \tag{10}$$

$$Y_{uu}(u,\sigma) = y(u) \otimes g_{uu}(u,\sigma) \tag{11}$$

where $g_u(u,\sigma)$ and $g_{uu}(u,\sigma)$ are the first and second derivative of 1D Gaussian function with standard deviation of $\sigma$.

### B. The CSS Image

The function defined by the equation, $K(u,\sigma)=0$, is called the CSS image of a planar curve $\Gamma$ [9]. Fig. 1 shows a CSS image for a typical Persian word in which the horizontal and vertical axes are $u,\sigma$ values respectively and shows zero crossing in curvature values, $K(u,\sigma)$.

To obtain CSS image, the input curve, $\Gamma$, is convolved with Gaussian function of different width $\sigma$. The output curve, $\Gamma_\sigma$, is called evolved version of $\Gamma$. Different $u$ and $\sigma$ values defining the equation $K(u,\sigma)=0$, form a binary image that is called CSS image. In [9], [10] different properties of CSS image are discussed. It is shown that renormalized CSS image is more robust against curve noise. In renormalized CSS image the evolved curve $\Gamma_\sigma$ is reparametrized by its normalized arc length parameter $\omega$ as follows:

$$\omega = \phi_\sigma(u) = \frac{\int_0^u |R_v(v,\sigma)|\,dv}{\int_0^1 |R_v(v,\sigma)|\,dv} \tag{12}$$

$$\hat{X}(\omega,\sigma) = X(\phi_\sigma^{-1}(\omega),\sigma) \tag{13}$$

$$\hat{Y}(\omega,\sigma) = Y(\phi_\sigma^{-1}(\omega),\sigma) \tag{14}$$

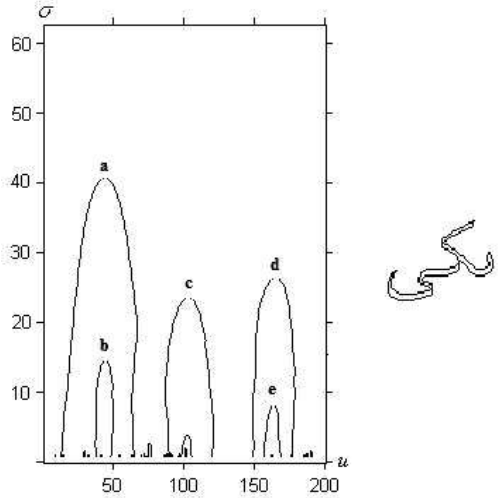


Fig. 1. The contour of a Persian word (right) and its CSS image (left).

The renormalized CSS representation is robust with respect to scale, noise and change in orientation. A rotation of the curve usually causes a circular shift on its representation. Note that the effect of a change in the starting point of a curve is also the same. Due to arc length normalization, scaling does not change the representation, and noise may create some small contours in the CSS image, but the main contours and therefore the corresponding maximums remain unaffected.

Another property of the CSS image is that it retains the local properties of the shape. Every contour of the CSS image corresponds to a concavity or a convexity of the shape. A local deformation of the shape mainly causes a change in the corresponding contour of the CSS image.

## III. The Proposed Algorithm

We assumed that the bilingual scripts may have Farsi and English words and characters together; therefore we proposed an algorithm to recognize scripts in the connected components level. The output of recognition is then generalized to word identification. For the scripts that a complete line or page contains characters of one language, the algorithm can be extended to identify scripts in line or page levels by checking the number of recognized Farsi and Latin word in the line or the number of the recognized Farsi and Latin lines in the page. Therefore the proposed algorithm can be used in connected component, word, line and page levels. The proposed algorithm has the following steps:

- Apply required preprocessing to extract lines,

words and connected components.
- Represent each connected component as planar curve and extract required features.
- Classify the features and recognize script language in connected component level
- Recognize script language in higher levels

### A. Preprocessing

The first stage of the proposed algorithm is the identification based on connected components. To identify connected components we first smooth the input image and apply threshold to convert it to binary image. We used Otsu method [11] to calculate threshold. We then extract connected components and remove small components such as dots and commas which are common in both Farsi and Latin language. To obtain the threshold for removing small connected components, we calculate the number of pixels for largest component in the text and divide it by 13 to calculate the required threshold. The number was obtained experimentally. To recognize script in higher levels we need to extract word and line in the script. To extract lines in the input script, the image profile in horizontal directions are calculated. When the lines are extracted the vertical profile of each line is used to extract words.

### B. Feature Extraction

To obtain proper features for the classification of connected components, we first extract the outer contour of connected components and represent them as planar curve $\Gamma=r(u)=(x(u), y(u))$ . To extract outer contours, we fill holes in the connected components and find pixels in the boundary of connected components. To normalize the arc length, the extracted curves are resampled and represented by 200 equally distant samples.

We used renormalized CSS representation to extract features from curves. As it is shown in Fig. 1, CSS image contains several contours with different local maximums. We used local maximums of CSS contours to extract proper features for script identification. CSS contours with local maximum of less amplitude (less than $0.2\sigma_{max}$) are related to noise or small ripples of the curve. In order to avoid noisy feature we remove CSS contours with maximum amplitude value of less than 0.2 of the largest maximum value in CSS image.

Each local maximum in CSS image is specified with amplitude value $\sigma$ and the renormalized curve index $\omega$. Although the general shape of CSS image doesn't change with curve scale and rotation, the rotation and change in the starting point of a curve causes a circular shift in $\omega$ and the CSS representation. The change in curve scale also scales the amplitude of CSS representation. To cope with this problem we used the difference of $\omega$ and the ratio of $\sigma$ values for two successive local maximums as features. We used 5 largest local maximums of CSS image to extract 10 CSS based features including 5 $\omega$ differences and 5 $\sigma$ ratios. It is important to note that considering the circular shift property of CSS image, the fifth local maximum is compared to the first local maximum.

Removing CSS curve with less amplitude, may result in CSS images with less than five local maximums, in this case we fill the empty elements with a constant number larger than $\omega_{max}$ and $\sigma_{max}$.

For CSS image with only one local maximum, it is also impossible to calculate the difference and ratio. In this situation, we find the w values for $\sigma=0.5\sigma_{max}$ and the difference of the obtained w values is used as feature. The $\sigma$ ratio for this case is assumed to be one.

Curvature scale space representation produce no local maximum for sphere like connected components such as "o" letter in Latin and " ه" letter in Farsi, in this case the feature vectors are similar for both letters. To handle this problem we added five other shape based and structural features to our feature vector as follows:

- The ratio of vertical profile maximum after filling holes to the height of connected component (VPMXHR).
- The ratio of vertical profile mean after filling its holes to the height of connected component (VPMNHR).
- Height to width ratio of connected component (HWR).
- The number of holes in connected component (NH).
- The number of local maximums extracted from CSS image (NLM).

Table 1 shows the summary of the proposed feature vector for the recognition of connected component language.

TABLE I
THE SUMMARY OF THE PROPOSED FEATURE VECTOR

| Feature Title | ω Differences | σ Ratios | VPMXHR | VPMNHR | HWR | NH | NLM |
|---|---|---|---|---|---|---|---|
| Number of elements | 5 | 5 | 1 | 1 | 1 | 1 | 1 |

### C. Classification

We used improved KNN[1] classifier for the language identification in connected component level. In KNN classifier the distance between input feature vector and feature vectors in database are calculated. Then K nearest feature vectors are selected from the database. Finally the output class is determined using a voting algorithm. The calculation of distance is the most important part of the KNN algorithm and affects the accuracy of algorithm. Our experimental results showed that the Euclidian distance is not a proper distance measurement method for the proposed feature vector. This is mainly because of difference in the type and the number range of feature elements in the feature vector. To handle this problem we used a new algorithm to calculate distance for two feature vector as follows:

- Build a vector with the same size of the proposed feature vector which is called distance vector.
- Calculate the difference between the first five elements of two feature vectors. The first five elements in the distance vector are set to zero if

[1] K nearest neighbor

the calculated difference is less than 6, otherwise they are set to one. The value of 6 has been obtained experimentally.

- Calculate the difference between the second five elements of two feature vectors. The second five elements in the distance vector are set to zero if the calculated difference is less than 0.5, otherwise they are set to one. The value of 0.5 has been obtained experimentally.
- The remaining elements of the distance vector are filled with squared difference of remaining elements in two vectors.
- The distance between two feature vectors is defined as the sum of distance vector elements.

## IV. Script Identification at Higher Levels

In addition to script identifications in connected components level, it is possible to recognize script language in higher levels like word, line or page levels where the identification in word level is more important. In script identification in line and page levels it is assumed whole words in line and page belong to one language. The proposed algorithm for script identification in higher levels has the following steps:

1. The lines and words in the input image are extracted using image profile in horizontal and vertical directions.
2. The number of connected components forming the word is calculated. If the majority of components belong to Latin language, the word is Latin; otherwise it is a Farsi word. If the number of Farsi and Latin components is the same, the word will be recognized as Farsi word experimentally.
3. The words of each line are extracted.
4. If the majority of words in a line belong to Farsi language, the line is considered as Farsi, otherwise it is Latin.
5. Similarly according to the number of Farsi and Latin lines in a page, identification in page level is carried out.

## V. Experimental Results

We implemented the proposed algorithm using a MATLAB program and tested with the provided data set. Our data set contains 99 pages including 62 Farsi and 37 Latin pages. This data set contains documents with resolution of 300dpi that are randomly provided from Internet. Each page has an average of 28 lines. For both Farsi and Latin scripts, three fonts in different sizes and three different types of pens including natural, bold and italic pen are used.

In Latin scripts each letter is a connected component while because of the cursive nature of Farsi scripts, connected components include letters, words and sub-words. Additionally letters may appear differently at the start, end and middle of the word in Farsi scripts. Therefore we need more scripts for the classification of Farsi scripts in database.

Table II shows results of script identification in connected component level. Table III and IV show the identification results in word and line levels respectively. We obtained the recognition rate of 100% for script identification in page level.

TABLE II
The Results of Script Identification in Connected Component Level

| | Farsi | Latin | Accuracy |
|---|---|---|---|
| Farsi | 16129 | 73 | 99.55% |
| Latin | 307 | 23698 | 98.72% |

TABLE III
Identification Results in Word Level

| | Farsi | Latin | Accuracy |
|---|---|---|---|
| Farsi | 6046 | 12 | 99.8% |
| Latin | 24 | 4285 | 99.44% |

TABLE IV
Identification Results in Line Level

| | Farsi | Latin | Accuracy |
|---|---|---|---|
| Farsi | 441 | 1 | 97.52% |
| Latin | 0 | 457 | 100% |

To show the efficiency of the proposed algorithm, the results of the proposed method has been compared to the results of the method presented in [2]. This method uses a 16 channel Gabor filter in different frequencies and channels applied to fixed size 128x128 input image. The input image is constituted by repeating the connected component image. In each channel, two features are extracted from the image which results in a 32 dimensional feature vector. We tested the mentioned method using our database. Table V shows the results of script identification using this method in connected component level on our dataset. Comparing the results of table III with table V illustrates that the results of the proposed method is more satisfactory.

TABLE V
Identification Results in Connected Component Level for Method Presented in [2]

| | Farsi | Latin | Accuracy |
|---|---|---|---|
| Farsi | 13513 | 603 | 95.73% |
| Latin | 400 | 7342 | 93.62% |

## VI. Testing Algorithm with Fonts Not Presented in Training Dataset

To show the efficiency of proposed features, we tested the proposed algorithm on documents with different fonts that were not presented in the training set. The test documents contain 5 Farsi documents with 5 different fonts, and 5 Latin documents with 5 different fonts. The results of this test in connected component and word levels presented in tables VI and VII respectively. The results of tables VI and VII show good accuracy especially in word level.

TABLE VI
Identification Results in Connected Component Level with Fonts Not Presented in Training Dataset

| | Farsi | Latin | Accuracy |
|---|---|---|---|
| Farsi | 3710 | 426 | 89.7% |
| Latin | 226 | 7925 | 97.23% |

TABLE VII
IDENTIFICATION RESULTS IN WORD LEVEL WITH FONTS NOT PRESENTED IN TRAINING DATASET

| | Farsi | Latin | Accuracy |
|---|---|---|---|
| Farsi | 1797 | 20 | 98.9% |
| Latin | 17 | 1557 | 98.92% |

## VII. ERROR ANALYSIS

We analyzed the source of errors for the proposed algorithm. The main error originated from some symbols that are common in two languages such as bracket, parenthesis, comma and so forth. This error mostly occurs for test images which have fonts with different sizes in a page. This makes some of small symbols not to be removed in preprocessing step and results in some errors. Another source of error that occurs mostly in Latin scripts is related to two adjacent connected components touched to each other. This contact causes the change in features extracted from CSS image as well as other features such as holes number and results in erroneous recognition. Unlike to Latin scripts, most of errors in Farsi scripts are related to the disconnection of connected components in some of Farsi fonts.

Most errors in line level belong to short lines that have few words.

## VIII. CONCLUSION

This paper presents a new algorithm for script identification in Farsi and Latin bilingual documents. The proposed method is based on features extracted from curvature scale space representation of connected components in scripts. The proposed algorithm is able to recognize script in four levels including connected component, word, line and page levels. The proposed features and algorithm are robust against font and scale change of scripts therefore we don't require having database of all font and sizes for language identification. Experimental results showed the good accuracy of proposed algorithm for script identification.

## REFERENCES


[1] Andrew Busch, Wageeh W. Boles and Sridha Sridharan, "Texture for Script Identification", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp.1720-1732, Nov. 2005.

[2] Huanfeng Ma and David Doermann, "Word Level Script Identification for Scanned Document Images", in *Proc. of Int. Conf. on Document Recognition and Retrieval*, pp.178-191, 2004.

[3] D Dhanya, A G Ramakrishnan and Peeta Basa Pati, "Script identification in printed bilingual documents", *Sadhana* Vol. 27, Part 1, pp. 73–82, February 2002.

[4] Santanu Chaudhury and Rabindra Sheth, "Trainable Script Identification Strategies for Indian Languages", *in Proc. of the ICDAR Fifth Int. Conf. on Document Analysis and Recognition (ICDAR'99)*, pp.657 - 660, 1999.

[5] I. Moalla, A. M. Alimi and A. Benhamadou, "Extraction of Arabic Words from Multilingual Documents", in Proc. Of *Artificial Intelligence and Soft Computing Conference (ASC2004)*, 2004.

[6] B.V.Dhandra and Mallikarjun Hangarge, "On Separation of English Numerals from Multilingual Document Images", *Journal of Multimedia*, vol. 2, no. 6, November 2007.

[7] A.L. Spitz, "Determination of the Script and Language Content of Document Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 235-245, Mar. 1997.

[8] Shijian Lu, Chew Lim Tan, and Weihua Huang, "Language Identification in Degraded and Distorted Document Images ", in proc. Of *21st National Conference on Artificial Intelligence*, pp. 769-774, 2006.

[9] F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature – based shape representation for planar curves", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 789-805, Aug. 1992.

[10] F. Mokhtarian, "Convergence properties of curvature scale space representations", Proc. of the 1995 British conf. on Machine vision, *vol. 1, pp. 357 – 366,1995.*

[11] N. Otsu, "A Threshold Selection Method from Gray-Level Histogram", *IEEE Trans. on Systems*, Man, and Cybernetics, vol. 9, pp. 62-66, 1976.