Comparison of Two Different Feature Sets for Offline Recognition of Handwritten Arabic Words

Mario Pechwitz, Volker Maergner, Haikal El Abed

Comparison of Two Different Feature Sets for Offline Recognition of Handwritten Arabic Words

Mario Pechwitz[1*], *Volker Maergner, Haikal El Abed*
Technical University Braunschweig
Institute of Communications Technology (IfN)
38092 Braunschweig, Germany
{maergner,el-abed}@ifn.ing.tu-bs.de

Abstract

Normalization is a very important step in automatic cursive handwritten word recognition. Based on an offline recognition system for Arabic handwritten words which uses a semi-continuous 1-dimensional HMM recognizer two different feature sets are presented. The dependencies of the feature sets from normalization steps is discussed and their performances are compared using the IFN/ENIT - database of handwritten Arabic words. As the lower and upper baseline of each word are part of the ground truth (GT) of the database, the dependency of the feature set from the accuracy of the estimated baseline is evaluated.

Keywords: Handwriting recognition, Arabic word recognition, feature extraction, baseline evaluation, HMM

1. Introduction

Automatic recognition of handwritten words is still a challenging task. Especially for the automatic recognition of Arabic handwritten words a lot of work has to be done (see e.g. [1] and [2]). It goes without saying that the most important requirement for the development and comparison of recognition systems is a large database together with ground truth (GT) information. However, in the past most databases were not available to the public. The *IFN/ENIT* - database, published at the CIFED 2002 conference [10] is a database of handwritten Arabic words (26459 words, 411 different writers). This database is available for free, for noncommercial use (www.ifnenit.com). At ICDAR 2003 [12] we presented first results using this database for training and testing of a HMM based approach for handwritten Arabic word recognition. This work has shown that the used feature set depends strongly on the accuracy of the normalization step, especially on the accuracy of the position of the estimated baseline. At ICDAR 2005 a first competition of Arabic handwriting recognition [9] based on the *IFN/ENIT* - database has been carried out. The results have shown that HMM methods performed best on the data set used. All systems used different methods for normalization and feature extraction. It is a great advantage of the *IFN/ENIT* - database that not only the coded word but also the position of the lower baseline (baseline) and in the version 1.0p2 also for a part of all words the upper baseline (topline) is included. This allows - for the first time - an evaluation of the estimated baselines. This paper gives a comparison of two different feature sets used to recognize Arabic handwritten words. The evaluation is done using the results obtained with the *IFN/ENIT* - database.
Section 2 provides an introduction to the recognition system, a detailed description of the preprocessing, normalization, and feature extraction methods, and a short introduction to the HMM based recognition process. Section 3 follows with results using the different feature sets.

[1*] now with: Paradatec GmbH, Mittelweg 7, 38106 Braunschweig, Germany (mp@ifnenit.com)

2. Recognition system

Figure 1 shows a block diagram of the recognition system. In the following subsections we describe normalization, feature extraction, and HMM-based recognizer in more detail.

2.1. Preprocessing

After scanning a document some basic preprocessing tasks like image binarization, word segmentation, and noise reduction have to be performed. Due to the fact that we use the cropped binary word images coming from the *IFN/ENIT* - database, some of these fundamental preprocessing tasks were already done during the database development.
Using the binary image we perform a connected component analysis, extract a contour representation of the image, and perform a noise reduction filtering synchronously. For estimating normalization parameters as well as for performing normalization, a skeletonization of the word image is useful. The skeletonization is performed on the contour representation resulting directly in a graph representation of the skeleton [7]. The advantages of this method are the following: The data structure is easily accessible, the pen dependent line width is removed, and the connected components are easily to obtain.

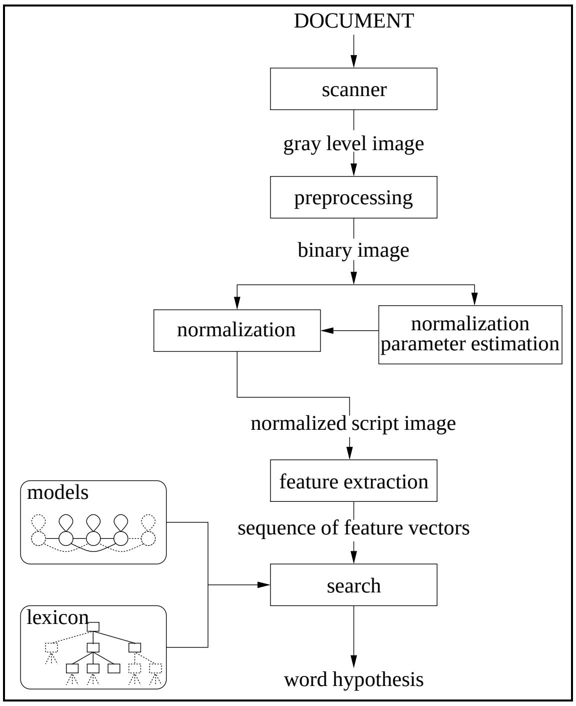

Figure 1. Block diagram of the HMM based recognition system

2.2. Writing lines estimation

In order to reduce writing style variability, we use several normalization steps. These steps involve writing line skew normalization, and horizontal word width scaling according to the number of characters. The writing line, which we call baseline is essential for subsequent steps, e.g. normalization and feature extraction, because the baseline represents an important orientation in a word.

2.2.1 Baseline estimation

In our previous work we turned our attention to the baseline estimation problem. Approaches based on horizontal projection histograms are widely used for Arabic printed text. The projection method is robust and easy to implement but it needs straight lines and long words, which is often not the case for single handwritten words. We implemented this approach transforming the binary word image into a Hough parameter space, where the dark regions indicate line directions with many black pixels on a straight line in the word image. Figure 3 gives an example.
Alternatively we developed a method completely based on polygonally approximated skeleton processing [11].

The main idea of this approach is to calculate robust features from the skeleton and use these features for classifying the obtained connected components into baseline relevant and baseline irrelevant objects. In a subsequent step a regression analysis of points of the relevant objects is done to estimate the final baseline position.
Due to the fact that the baseline is part of the GT of the

Table 1. Evaluation of baseline estimation methods

baseline quality (average pixel error)	Skeleton method	Horizontal projection
good($\leq$5)	76.7%	70.7%
acceptable($\leq$7)	87.5%	82.8%

IFN/ENIT - database, an evaluation of baseline detecting algorithms is possible.

With the skeleton based approach we reached better results[12]. In Table 1 the results using the proposed skeleton based method in comparison with the simple horizontal projection method are shown. The baseline quality is estimated by calculating the baseline error. This error is calculated as the area between the ground truth baseline and the estimated baseline in pixel. Figure 2 gives an example of the calculation of the baseline error. The two thresholds of 5 and 7 as good and acceptable are the result of visual tests on words of the database.

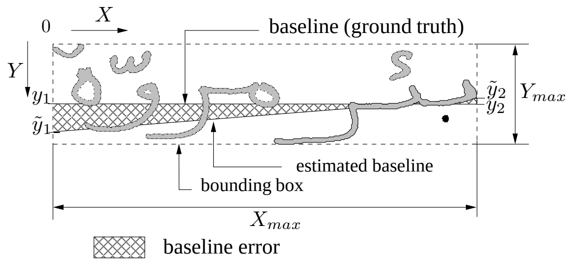

Figure 2. Visualization of the baseline error definition

2.2.2 Topline estimation

The horizontal projection histogram is often used to estimate the upper baseline (topline) too. We again use the Hough space but now we calculate the vertical gradient of the Hough space to determine the topline by selecting the maximal gradient value within a search area above the position of the baseline.

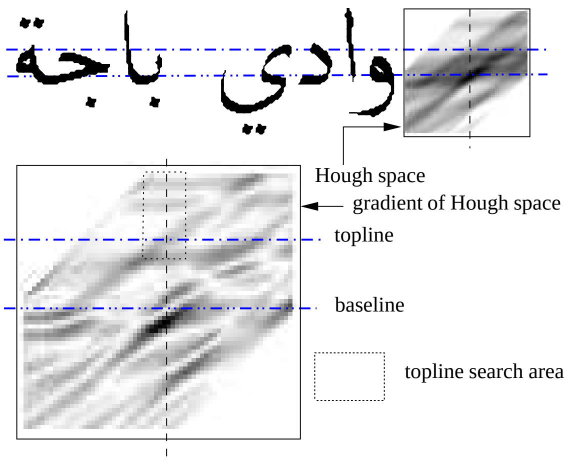

Figure 3. Topline estimation approach using the vertical gradient of the Hough parameter space (For this example printed text was used)

Table 2. Evaluation of topline finding methods

topline quality (average pixel error)	Hough space method	Simple method
good($\leq$5)	46.3%	56.8%
acceptable($\leq$7)	59.6%	74.3%

Figure 3 gives an example of the topline estimation approach. Due the fact that the topline is part of the *IFN/ENIT* - database (*set a*), an evaluation of the topline detecting algorithm was performed in the same manner as for the baseline. The result is disappointing (Table 2). Only for about 60% of the handwritten Arabic words an acceptable topline was determined. The comparison with the approach reported in [12] shows that even this straight forward method provides better results. In this approach the topline was set parallel to the baseline in a distance of 40% of the distance between baseline and the top of the word. These experiments have shown that the topline estimation should not be used within an Arabic handwritten word recognition process. In the following two different word normalization methods and two different feature sets are presented with low and without any dependency of the estimated topline position.

2.3. Normalization and feature extraction method A

The first normalization step is a rotation of the word resulting in a horizontal baseline. Subsequently a vertical height normalization is done using a linear characteristic between topline and baseline and a nonlinear characteristic elsewhere, this results in constant heights for ascender and descender regions and thus in a fixed total height of the resulting skeleton graph. Next a horizontal width normalization with a linear characteristic is performed, yielding a word with constant average character width. The line thickness is normalized during the generation of the skeleton. Finally a rethickening is done by a Gaussian filtering of the normalized skeleton image, resulting in a gray level image. Figure 4 shows an example of a normalized word image.
The feature extraction method *A* is directly based on an image representation of the script using pixel values as basic features. A rectangular window is shifted in respect to the Arabic writing direction from right to left across the normalized gray level script image and generates a feature vector (frame). Figure 4 gives an example of the sliding window feature extraction method. The three columns of the sliding window are concatenated to a feature vector. This is done at each position of the window.

2.4. Normalization and feature extraction method B

The normalization steps in method *A* and *B* are widely identical. However, instead of the vertical height normalization used in *A* we now enlarged virtually the word

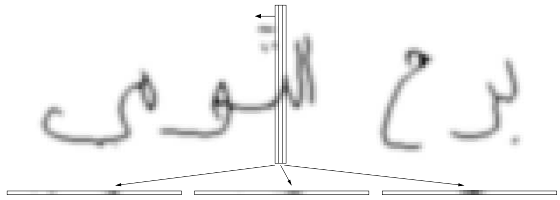

Figure 4. Extraction of pixel features using a sliding window with three columns (method **A**)

skeleton graph image with some blank lines to bring the baseline of the word into the middle of the image. A topline estimation is not needed. The normalized word skeleton graph is now used for the feature extraction process.

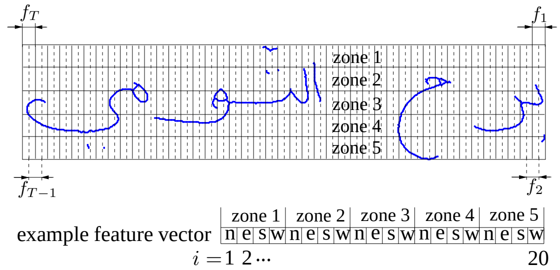

Figure 5. Extraction of skeleton direction features in 5 zones using overlapping frames (method **B**)

The feature extraction process starts with splitting the word image into a set of frames with fixed width in vertical direction. Each frame has an overlap of 50% with it's neighbors. Each frame is split horizontally into 5 zones with equal height (see Figure 5). The choice of 5 zones is intuitively, yielded the best recognition results, and corresponds with the work in [5]. The length of all lines within a zone of a frame calculated in the four directions north, east, south, and west are used as features. A normalization of these values with the height of the zone ensures the invariance of the height of the word without the need of a topline estimation. Each frame is represented by a 20-dimensional feature vector. To overcome the problem of very different value ranges of each feature i a normalization is performed (see Eqs. 1 to Eqs. 6). Where i is the number of the feature in the feature vector j and $x(i)_j$ is the value of the feature vector of frame j. $\bar{x}(i)$ and $s(i)$ are calculated using a sample set of words. w_{min} is the minimum cut off value and w_{max} is the maximum cut off value. V_σ is a parameter and set to a value of 0.5. The function $\hat{x}(i)_j$ is hard limited based on the previously calculated values of w_{min} and w_{max}. The value of $\hat{x}(i)_j$ is finally adjusted to a scale of 0 to 255. At last the value $\check{x}(i)_j$ gives the normalized value of $x(i)_j$.

$$\bar{x}(i) = \frac{1}{N} \cdot \sum_{j=1}^{N} x(i)_j \tag{1}$$

$$s(i) = \sqrt{\frac{1}{N-1} \cdot \sum_{j=1}^{N} (x(i)_j - \bar{x}(i))^2} \quad (2)$$

$$w_{min}(i) = \begin{cases} \bar{x}(i) - V_\sigma \cdot s(i) & \text{if } z_1 \\ \min(x(i)_j) \ \forall \ 1 \leq j \leq N & \text{else} \end{cases} \quad (3)$$
$$with \ z_1 := \min(x(i)_j) < \bar{x}(i) - V_\sigma \cdot s(i)$$

$$w_{max}(i) = \begin{cases} \bar{x}(i) + V_\sigma \cdot s(i) & \text{if } z_2 \\ \max(x(i)_j) \ \forall \ 1 \leq j \leq N & \text{else} \end{cases} \quad (4)$$
$$with \ z_2 := \max(x(i)_j) > \bar{x}(i) + V_\sigma \cdot s(i)$$

$$\hat{x}(i)_j = \begin{cases} w_{min}(i) & \text{if } \ x(i)_j < w_{min}(i) \\ x(i)_j & \text{else} \qquad \forall \ 1 \leq j \leq N \\ w_{max}(i) & \text{if } \ x(i)_j > w_{max}(i) \end{cases} \quad (5)$$

$$\check{x}(i)_j = \frac{\hat{x}(i)_j - w_{min}}{w_{max} - w_{min}} \cdot 2^8 \qquad \forall \quad 1 \leq j \leq N \quad (6)$$

2.5. HMM recognizer

As the basic HMM scheme we use Semi Continuous HMMs (SCHMM) as described in detail by Huang [8]. In our tests we used 7 states for a character but this is still a parameter to be optimized. For each state we use 3 transitions: a self-transition, a transition to the next state, and a transition which skips a single state.
Basically the standard Viterbi Algorithm is used for training and recognition. In the training process a segmental k-means algorithm is performed. In each iteration only the state-vector assignments resulting from best path obtained from applying the Viterbi Algorithm are used to re-estimate model parameters. The initialization is done by a Dynamic Programming clustering procedure.
The recognition is done by applying a frame synchronous network Viterbi search algorithm together with a tree structured lexicon representing valid words. The number of hypotheses generated at each step is controlled by a constant score threshold relative to the current best score and a maximum number of allowed hypotheses to be activated at the same time [3, 4].

3. Experimental results

The *IFN/ENIT* - database for Arabic handwritten words together with the mentioned HMM recognizer was used to test the two different feature sets. For the HMM recognizer we used character models. Due to the fact that Arabic characters might have several different shapes depending on their relative position in a word, we generate a HMM-model for each character shape. Also ligatures as well as characters with additional marks are labeled and modeled separately. Therefore we got up to 160 different character shape HMM-models, resulting from 28 Arabic characters. These HMM-models are combined to valid word models using a lexicon with all 937 different Tunisian town/village names. Some town/village names occur in the database with slightly different writing. From this it follows that our lexicon consists of about 2100 valid entries. A word is recognized correctly if the recognizer output matches the post code given by the GT. Therefore we report recognition rates on word level.
Four distinct sets (a,b,c,d) are predefined in the database for training and testing recognition systems. We use three of them for training and one set for testing our system.
The goal of the accomplished tests is to point out, which of the two normalization and feature extraction methods is more robust against baseline detection errors. Due to the fact that the database comes with baseline GT, we first used these baselines. The results[1] using normalization and feature extraction method *A* together with pixel features are shown in Table 3. In comparison Table 4 shows the results achieved with normalization and feature extraction method *B*. Although the features differ considerably, the results are very similiar. It follows that in the case of optimal writing lines there are only slight differences between both methods.

Table 3. Recognition results using the ground truth baseline together with pixel features

TEST	training sets	test set	1-best	2-best	10-best
1	a,b,c	d	91.8	95.5	98.5
2	b,c,d	a	92.3	95.3	98.4
3	c,d,a	b	91.7	95.1	98.6
4	d,a,b	c	92.1	95.1	98.2

Table 4. Recognition results using ground truth baseline together with skeleton direction features

TEST	training sets	test set	1-best	2-best	10-best
1	a,b,c	d	90.7	95.7	98.1
2	b,c,d	a	91.2	95.4	97.4
3	c,d,a	b	91.0	94.9	97.9
4	d,a,b	c	90.8	94.7	97.4

Table 5 and Table 6 show the results we reached using the two different normalization and feature extraction methods but now using the estimated baseline position. A decrease of the recognition rate from about 91% to 87% and to 89% respectively can be observed. The reduction of the recognition rate is in both cases significant, but especially in the case of using the normalization and feature extrac-

[1] The *IFN/ENIT* - database in version 1.0 patch level 2 (v1.0p2) was used.

tion method B the results are noticeable better. The result shows us that the feature extraction method B is less dependent from a writing line estimation. The HMM recognizer still can be optimized to the used feature set.

Table 5. Recognition results using baseline estimation with skeleton based method and pixel features

TEST	training sets	test set	1-best	2-best	10-best
1	a,b,c	d	87.4	91.0	95.2
2	b,c,d	a	87.3	90.8	94.9
3	c,d,a	b	86.8	89.9	94.4
4	d,a,b	c	87.1	90.0	94.8

Table 6. Recognition results using baseline estimation with skeleton based method and skeleton direction features

TEST	training sets	test set	1-best	2-best	10-best
1	a,b,c	d	89.1	91.7	95.9
2	b,c,d	a	88.9	91.5	95.6
3	c,d,a	b	88.7	90.8	95.3
4	d,a,b	c	89.0	91.5	96.4

Additionally we can say that our system, with respect to the different data sets used, in comparison with the work of Dehghan et al. [5] provided better recognition results, even using similar features. Dehghan used a discrete HMM with chain-code histogram features, clustered with the Kohonen self-organization map on Farsi (Arabic) handwritten word recognition. They generated models on word level and achieved a recognition rate of 65.0% by using a lexicon of size 198. But also the competition winning system at ICDAR 2005 [6], which used a HMM recognizer too with similiar line based features achieved on the same data but with a reduced lexicon a minor recognition rate than our system with skeleton directional features (see Table 7). These differences in recognition rate may be caused by differences in the feature extraction method or the baseline estimation algorithm.

Table 7. Recognition results from [6] on a reduced lexicon size of 450 (out of 937)

TEST	training data	test data	rec. rate
1	a,b,c	d	87.20
2	b,c,d	a	86.10
3	c,d,a	b	86.88
4	d,a,b	c	85.45

4. Conclusion

A HMM based approach to recognize Arabic isolated handwritten words was presented. We proposed an approach using SCHMMs and we compared two different normalization and feature extraction processes in relation to the robustness against baseline detection errors. The baseline GT was essential for these tests.
We achieved recognition rates of up to 89% on word level using the skeleton based method for baseline estimation and skeleton directon features.
Currently we are working on the optimization of the feature set together with the HMM model parameters.

5. Acknowledgment

This project was in parts supported by the Volkswagen Stiftung under contract number I/75 084.

References

[1] B. Al-Badr and S. A. Mohmond. Survey and bibliography of Arabic optical text recognition. *Signal Processing*, 41:49–77, 1995.

[2] A. Amin. Off-line Arabic character recognition: The state of the art. *Pattern Recognition*, 31(5):517–530, 1998.

[3] R.-D. Bippus and M. Lehning. Cursive script recognition using Semi Continuous Hidden Markov Models in combination with simple features. In *European workshop on handwriting analysis and recognition*, Brussels, July 1994.

[4] R.-D. Bippus and V. Maergner. Script recognition using inhomogeneous P2DHMM and hierarchical search space reduction. In *Proc.of 5th. Int. Conference on Document Analysis and Recognition, ICDAR 1999*, pages 773–776, Bangalore, India, September 1999.

[5] M. Dehghan, K. Faez, M. Ahmadi, and M. Shridhar. Handwritten farsi (Arabic) word recognition: A holistic approach using discrete HMM. *Pattern Recognition*, 34(5):1057–1065, 2001.

[6] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel. Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling. In *Proc.of 8th. Int. Conference on Document Analysis and Recognition, ICDAR 2005*, pages 893–897, Seoul, Korea, August 29th - September 1st. 2005.

[7] A. Ferreira and Ubeda S. Ultra fast parallel countor tracking with application to thinning. *Pattern Recognition*, 27(7):867–878, 1994.

[8] X.D. Huang et al. *Hidden Markov Modells for Speech Recognition*. Edinburgh Universal Press, 1990.

[9] V. Maergner, M. Pechwitz, and H. El-Abed. ICDAR 2005 Arabic Handwriting Recognition Competition. In *Proc.of 8th. Int. Conference on Document Analysis and Recognition, ICDAR 2005*, pages 70–74, Seoul, Korea, August 29th - September 1st. 2005.

[10] M. Pechwitz, S. S. Maddouri, V. Maergner, N. Ellouze, and H. Amiri. IFN/ENIT - database of handwritten Arabic words. In *Proc. of Colloque Int. Francophone sur l'Ecrit et le Document, CIFED 2002*, pages 129–136, Hammamet, Tunisia, October 21-23 2002.

[11] M. Pechwitz and V. Maergner. Baseline estimation for Arabic handwritten words. In *Proc. of the 8th. Int. Workshop on Frontiers in Handwriting Recognition, IWFHR 2002*, pages 479–484, Niagara on the Lake, Canada, August 5-8 2002.

[12] M. Pechwitz and V. Maergner. HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT - database. In *Proc.of 7th. Int. Conference on Document Analysis and Recognition, ICDAR 2003*, pages 890–894, Edinburgh, Scotland, August 3-6 2003.