Segmentation-Free Word Recognition with Application to Arabic

Badr Al-Badr Robert M. Haralick
The Intelligent Systems Laboratory, FT-10
University of Washington, Seattle, WA 98195, USA
{badr, haralick}@cs.washington.edu

Abstract

This paper describes the design and implementation of a system that recognizes machine-printed Arabic words without prior segmentation. The technique is based on describing symbols in terms of shape primitives. At recognition time, the primitives are detected on a word image using mathematical morphology operations. The system then matches the detected primitives with symbol models. This leads to a spatial arrangement of matched symbol models. The system conducts a search in the space of spatial arrangements of models and outputs the arrangement with the highest posterior probability as the recognition of the word.

The advantage of using this whole word approach versus a segmentation approach is that the result of recognition is optimized with regard to the whole word. Results of preliminary experiments using a lexicon of 42,000 words show a recognition rate of 99.4% for noise-free text and 73% for scanned tex.

1 Introduction

OCR of machine-printed Arabic text is considerably harder then that of Latin text due to the cursiveness of Arabic writing. Arabic also has more shape classes and combined characters (ligatures). Moreover, characters within a word might overlap vertically (without touching). For a survey of Arabic character recognition, the reader is referred to [1, 2].

Figure 1 demonstrates some of the features of Arabic on a typeset sentence consisting of seven words. Reading from right to left, the first word is a ligature made up of two characters, the second word consists of three characters and two main connected components. The short strokes at the top of text are diacritic marks. The ligature in the middle of the figure consists of three vertically stacked characters. The text's baseline is between the two horizontal lines.

This paper describes a system for recognizing Arabic words without segmenting in advance words into characters. It searches for the best recognition of a word taking into account the whole word image, and not only local regions of the image as with segmentation-based approaches (e.g., [4, 5]). It is also designed to be taught new fonts automatically. To teach the system a new font the input that is required is an ideal (noise-free) image of all the character shapes and ligatures. The system uses a noise model so that it can recognize degraded instances of the characters.

The system recognizes an input word by detecting a set of shape primitives on the word. It then matches the regions of the word (represented by the detected primitives) with a set of symbol models. A spatial arrangement of symbol models that are matched to regions of the word, then, becomes the recognition of the word. Since the number of potential arrangements of all symbol models is combinatorially large, the system imposes a set of constraints that pertain to word structure and spatial consistency. The system searches the space made up of the arrangements that satisfy the constraints and tries to maximize the a posteriori probability of the arrangement. Figure 2 shows a diagram of the system. In the figure, the boxes designate data structures and the ovals designate processes.

2 Recognition strategy

The input to the word recognition system is an image of a machine printed Arabic word. Before a word is recognized, the baseline of the word is detected. The baseline corresponds to the row of the image with the highest density of black pixels; it is detected using the horizontal projection profile.

The system starts by detecting a predefined set of shape primitives on the isolated image of the word. Instances of primitives are found by applying the erosion morphological transforms on the image, with the shape of the primitive as the structuring element. In

0-8186-7128-9/95 $4.00 © 1995 IEEE

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

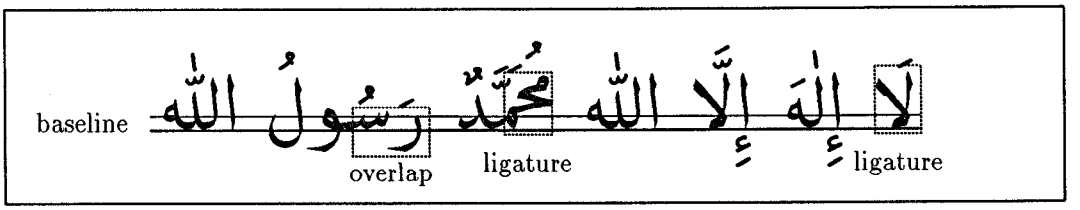

Figure 1: A sample of written Arabic showing some of its characteristics.

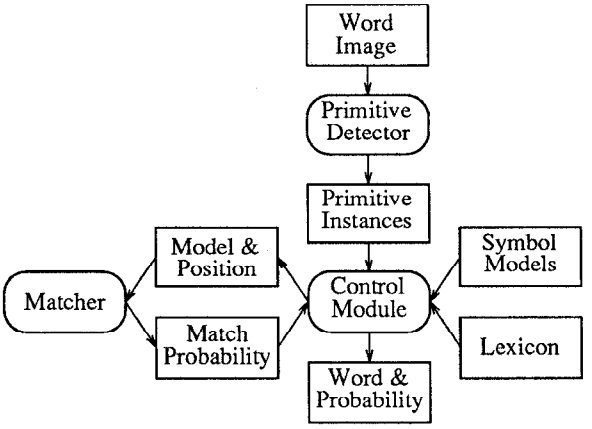

Figure 2: Block Diagram of the recognition system.

the erosion operation, the structuring element K may be visualized as a probe that slides across the image A. Whenever K translated to x is contained in A, x belongs to the erosion of A by K.

2.1 Word recognition

To identify words, the recognition system uses a set of symbol models. A *symbol* is a shape or glyph of a character. A *symbol model* is a description of the symbol in terms of model primitives. The primitives of a symbol model represent of as the primitives detected on an ideal (noise-free) image of the symbol and the deformations expected under the noise model. Symbol models are generated during a training stage that is done in advance of recognition.

The goal of the system is to find a spatial arrangement of symbol models with the maximum a posteriori probability. This spatial arrangement of models specifies a string of symbols, or a recognition of the word. The system achieves its goal by matching symbol models with local regions in the word image (described below).

The system uses state-space search to find the best spatial arrangement of symbol models. A state in the search space specifies a set of matched symbol models, each at a certain translation relative to the coordinates of the word image, and associated with each model is a computed probability of match. As such, a state specifies a string of characters that is a (partial) solution to the recognition problem.

To reduce the number of searched states, we incorporate a number of assumptions about word structure and the spatial arrangement of models. The constraints on word structure include: (1) the font type and size are consistent within a word; (2) within a word, all adjacent symbols are compatible (e.g., a ending-form must not be followed by a beginning-form); and (3) a word begins with a beginning form and ends with an ending form. To further reduce the search space, the system uses a lexicon for Arabic words. The system searches only the states that lead to a lexicon word, and, hence, recognizes words in the lexicon only.

The spatial constraints govern the location of the bounding boxes of symbol models relative to one another and to the word's bounding box. The spatial constraints specify that the baseline of each symbol must coincide with the word's baseline. Further, bounding boxes of symbols have minimal overlap with one another, have only small gaps between them, and are almost totally enclosed in the bounding box of the word.

When recognizing a word, the system starts at the right-most region of the word and proposes an initial set of translated symbol models as the recognition of the first symbol of the word. Structural filtering keeps only symbol models that can occur at the beginning of words. Spatial filtering removes translations that take a large portion of the symbol model outside of the word's bounding box and translations that put the model's baseline far apart from the words baseline.

Each of the remaining translated models is passed to the matcher, which computes the probability of

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

match. The system makes each of the matched models a state and expands the state with the best match probability. Expansion of a state means that the recognition system resumes recognition with the contents of the state (the used region and primitives) and attempts to recognize the remaining portion of the word.

The recognition procedure continues in this manner, with earlier models restricting the selection of future models. At all states whose left-most model represents a glyph that can terminate a word—i.e., isolated- and end-forms—the system evaluates the state as a recognition for the word by computing the a posteriori probability of the word.

The recognition system keeps track of the word with the highest probability, and uses the highest probability to limit the expansion of states whose probability is below the maximum. The expansion of a state terminates when all the proposed models are filtered out or have a match probability below the maximum. By the time the search is completed, all states have either been pruned because of their low probability or have been evaluated. The system then outputs the state with the maximum a posteriori probability as the solution to the recognition problem.

2.2 A posteriori word probability

Let M be a symbol model, and let t be a translation of the model onto the word image coordinates. A spatial arrangement of n symbol models is represented as $\{\langle M_1, t_1\rangle, \ldots, \langle M_n, t_n\rangle\}$. We assume that the models are ordered by the x coordinate of their centroid. Further, let S denote the subset of primitives detected on the word that have not been used in the matching, let I be the image of the word, and let the name of the character represented by model M be $l(M)$. Then the string of characters $l(M_1)\ldots l(M_n)$ is a proposed recognition (solution) for the unknown input word.

The probability of a spatial arrangement is computed from the match probabilities of its constituent models. The probability of each symbol matching the word region onto which it have been translated is determined by the matcher, and is returned on an independent call to the matcher. Hence, we take the probability of each symbol's match to be independent of the other symbols matches. So, the a posteriori word probability distribution is

$$P(\{\langle M_1, t_1\rangle, \ldots, \langle M_n, t_n\rangle\}, S \mid I) = \left(\prod_i^n P(\langle M_i, t_i\rangle \mid I)\right) \times P(S \mid I). \quad (1)$$

The first term on the right-hand side is the product of model match probabilities and is discussed below. When the system decides that it has a word, the remaining unmatched detected primitives are considered to be extraneous primitives. The probability of the extraneous primitives (the second term) is product of the probabilities of the individual extraneous primitives. It can be thought of as a penalty for matches that do not account for all detected primitives. The extraneous primitives have an exponential parametric distribution similar to the distribution of matched primitives.

2.3 Model matching

The translation of a symbol model determines where to place its bounding box on the word image. Matching a symbol model to a word region requires matching each of the model primitives to the primitives detected on the image. Matching a certain model primitive requires (1) finding the detected primitives that correspond to it, (2) computing the match measurement, and (3) calculating the probability of match.

When a symbol model is translated onto a word region, the system determines the detected primitives that correspond to a model primitive by examining the correspondence region of the model primitive. The *correspondence region* for a model primitive is a region that includes all expected deformations of the primitive under the noise model. The noise model used in this work assumes that noise is more likely to occur near the boundary of text, that the intensity of noise decreases exponentially with increasing distance from the boundary, and that noise pixels tend to occur in groups [6]. The match measurement for a model primitive is the number of pixels in the intersection of the detected primitives and the correspondence region. Figure 3 shows the intersection areas in black for four shape primitives.

The probability of a match between a symbol model M at translation t and an image I is calculated as the joint probability of the matches of all the model primitives. To reduce computation, we assume that the probability of a primitive match given its indicator is independent of other primitive matches and other indicators. So:

$$P(\langle M, t\rangle \mid I) = P_M(A_1, \ldots, A_n, D_1, \ldots, D_n) = \left(\prod_{i=1}^n P_M(A_i)\right) \times P_M(D_1, \ldots, D_n). \quad (2)$$

where D_i is an indicator variable that has the value of

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

Figure 3: Matching a scanned word with symbol model 1. The left-most image shows the scanned word. The other images, in reading order, correspond to the first four primitives of Figure 5. Each image shows primitives detected on the word in light gray, the correspondence regions of the model in dark gray, and the intersection areas in black.

1 when $A_i > 0$, and is 0 otherwise.

We Assume that the probability of match between a model primitive and it matching detected primitives has the parametric form:

$$P_M(A_i) = c \cdot e^{-a\frac{|A_i - A|}{A_m}} \quad (3)$$

The constant A is the area of the ideal model primitive, A_i is the match measurement, A_m is the area of the correspondence region, and a and c are the distribution parameters. These areas are illustrated in Figure 4. The distribution parameters are determined through Bayesian estimation using a large training sample of noisy symbol images. The joint distribution of indicator variables is a discrete distribution that is simplified through the use of decomposable graphical models. The probability models are discusses in greater detail in [3].

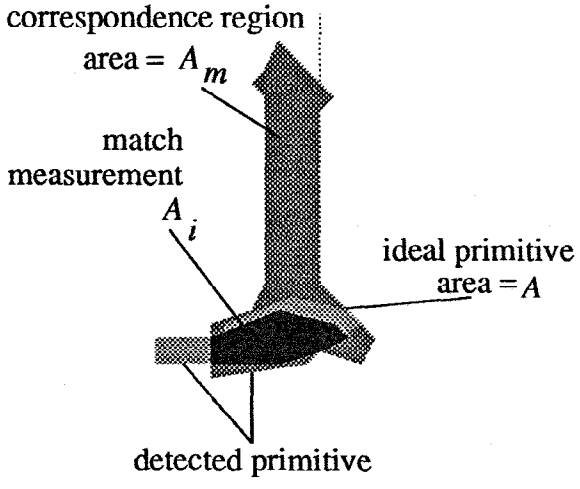

Figure 4: An illustration of the match region and measurement.

3 Experiments and discussion

The system is implemented in C and runs on a Sun Sparc 10. Here we describe some of the experiments conducted on the system. The number of symbols that the system is trained to recognize is 156. This includes the different shapes of the 28 Arabic characters, 13 ligatures, 20 punctuation marks, and 10 digits. The lexicon used for these experiments includes over 42,000 words that have been gathered from newspapers, magazines, and books. The system uses 13 primitives, which are shown in Figure 5.

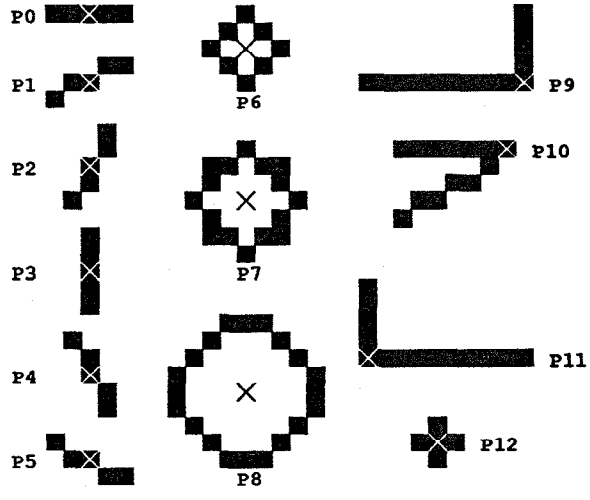

Figure 5: The shape primitives (structuring elements) used by the system. Each square corresponds to an image pixel. The origin of each shape is indicated by a cross.

The system has been trained on images of isolated symbols that have been synthetically generated and then degraded using the noise model of Kanungo *et al.* [6]. The synthetically generated images are noise-free as they are converted from Postscript to image format

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

on-line. The degradation parameters have been set at $\alpha_0 = \beta_0 = 1.0$, $\alpha = \beta = 1.5$, $c_0 = 0.0$, and $e = 3$. The training set consists of 500 degraded images per symbol, for a total of 78,000 symbols. The font type used for training and testing is the *Nadeem* Macintosh font at a size of 12 points.

The testing sets originate from seven pages selected from a news magazine. These pages were scanned at 300 dots-per-inch, in house. They were then zoned, and their text was entered by two independent typists and verified. The three test sets are: (1) the ideal set, which consists of the words of the seven document pages after converting them into noise-free documents, (2) the degraded set, which consists of the words of a synthetically degraded version of the above set, and (3) the scanned set, which consists of the words in the seven scanned pages.

With each of the test sets, the system is presented with the image of one word at a time. The recognition results are summarized in the following table:

test set	# words	rec. rate	time/word (ms.)
ideal	3787	99.39%	16,718
degraded	3522	95.60%	59,760
scanned	830	73.13%	224,593

Under the assumption that experiments are independent Bernoulli trials with identical success probability, the 95% confidence intervals for the recognition rates are ±0.1%, ±0.3%, and ±3%, respectively.

Most of the recognition errors were due to confusions in symbols that were very similar. The following table shows the most frequent confusions pairs in a left-to-right decreasing order:

true	لا	ح	لا	ئ	لا	أ	خ	»	«	ض
rec'ed	لآ	ح	لأ	ئ	لآ	ا	ح	«	»	ص

In its current state, the system is slow. Since state space search is used, the worst case time complexity for recognizing a word is exponential in the length of the word. Recognition is faster for short words, but some of the longer words might take more than 300 seconds. One way to reduce the space of the search is to examine the word to be recognized and suggest a small set of segmentation points, instead of trying out many symbol translations. The object of the search would then be to find the best subset of segmentation points. If the set of proposed segmentation points always includes the true segmentation points, then, in addition to reducing the space of the search, this approach can find the optimal solution. It remains to be investigated how this method actually performs.

In this work we have assumed that the words were already extracted from document pages and that they were not rotated. A practical recognition system must be able to robustly extract the words and align skew.

The main contribution of this work is that it optimizes the recognition of the symbols with respect to the whole word, without committing itself to a particular segmentation of the word into symbols. Other features of this system include that it has an explicit noise model, and that it is automatically trainable.

References

[1] P. Ahmed and M. A. A. Khan. Computer recognition of Arabic scripts based text — the state of the art. In *Proceedings of the 4th International Conference and Exhibition on Multi-lingual Computing (Arabic and Roman Script)*, pages 2.2.1–2.2.15, University of Cambridge, London, U.K., April 1994.

[2] Badr Al-Badr and Sabri Mahmoud. Survey and bibliography of Arabic optical text recognition. *Signal Processing*, 41(1):49–77, 1995.

[3] Badr Albadr. *A Segmentation-Free Approach to Text Recognition with Application to Arabic Text.* PhD thesis, Department of Computer Science & Engineering, University of Washington, 1995.

[4] Adnan Amin and H. B. Al-Sadoun. A new segmentation technique of Arabic text. In *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, pages 441–445, The Hague, Netherlands, September 1992.

[5] Mohamed Fakir and Chuichi Sodeyama. Machine recognition of Arabic printed scripts by dynamic programming matching method. *IEICE Transactions on Information and Systems*, 76(2):235–242, February 1993.

[6] Tapas Kanungo, Robert Haralick, and Ihsin Phillips. Global and local document degradation models. In *Proceedings of the International Conference on Document Analysis and Recognition*, Nangano, Japan, 1993.

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE