International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:6, 2008

An Efficient Feature Extraction Algorithm for the Recognition of Handwritten Arabic Digits

Ahmad T. Al-Taani

***Abstract*—**In this paper, an efficient structural approach for recognizing on-line handwritten digits is proposed. After reading the digit from the user, the slope is estimated and normalized for adjacent nodes. Based on the changing of signs of the slope values, the primitives are identified and extracted. The names of these primitives are represented by strings, and then a finite state machine, which contains the grammars of the digits, is traced to identify the digit. Finally, if there is any ambiguity, it will be resolved. Experiments showed that this technique is flexible and can achieve high recognition accuracy for the shapes of the digits represented in this work.

***Keywords*—**Digits Recognition, Pattern Recognition, Feature Extraction, Structural Primitives, Document Processing, Handwritten Recognition, Primitives Selection.

I. Introduction

IN areas of automatic document analysis and recognition, the correct interpretation of digits is very important. Automatic recognition of on-line handwriting has a variety of applications at the interface between man and machine. The performance of any system for handwriting recognition can be evaluated by several factors, such as size of the alphabet, independence of the writing style, and speed of recognition.

Automatic recognition of handwritten digits is difficult due to several reasons, including different writing styles of different persons, different writing devices, and the context of the digit. This leads to digits of different sizes and skews, and strokes that vary in width and shape.

Since the past few decades a number of researchers have investigated the problem of handwritten digit (character) recognition and many methods have been developed.

However, no system to date has achieved the goal of system acceptability. Researchers in this field have proposed different approaches, such as statistical, structural, and neural network approaches [1, 2]. The main primitives that form digits are line segments and curves. Different arrangements of these primitives form different digits. To recognize a digit, we should first find out the structural relationships between the features which make up the digit. The syntactic and structural approaches require efficient extraction of primitives [3-5]. A review of the computer based systems that have been developed for handwritten digit (character) recognition is given.

Ahmad T. Al-Taani is with the Department of Computer Sciences, Yarmouk University, Irbid, Jordan. (phone: +962-7-77438520, fax: +962-2-7211128, e-mail: ahmadta@yu.edu.jo).

Verma [6] proposed a method for cursive handwriting recognition. He used a contour code feature and a rule-based segmentation in the recognition process. A heuristic segmentation algorithm is used to over segment each word. Then the segmentation points are passed through the rule-based module to discard the incorrect segmentation points and include any missing segmentation points**.**

You et al. [7] presented an approach for segmentation of handwritten touching numeral strings. They designed a neural network to deal with various types of touching observed frequently in numeral strings. A numeral string image is split into a number of line segments while stroke extraction is being performed and the segments are represented with straight lines. Segmentation points are located using the neural network by interpreting the features collected from the primitives.

Olszewski [8] designed a structural recognition approach for extracting morphological features and performing classification without relying on domain knowledge. This system employs a statistical classification technique to perform discrimination based on structural features is a natural solution. A set of morphological features is suggested as the foundation for the development of a suite of structure detectors to perform generalized feature extraction for structural pattern recognition in time-series data.

Chan et al. [9] proposed a syntactic (structural) approach for the analysis of on-line handwritten mathematical expressions.

The authors used definite clause grammar (DCG) to define a set of replacement rules for parsing mathematical expressions. They also proposed some methods to increase the efficiency of the parsing process. The authors tested the proposed system on some commonly seen mathematical expressions and they claimed that their method has achieved satisfactory speedup.

Chan et al. [10] discussed a structural approach for recognizing on-line handwriting. The recognition process starts when getting a sequence of points from the user and then by using these points to extract the structural primitives. These primitives include different types of line segments and curves. The authors demonstrated their approach on 62 character classes (digits, uppercase and lowercase letters). Each class has 150 different entries. They stated that experimental results showed that the recognition rates were 98.60% for digits, 98.49% for uppercase letters,

97.44% for lowercase letters, and 97.40% for the combined set.

Amin [11] reviewed the state of Arabic character recognition research throughout the last two decades. The author summarized all the work accomplished in the past two decades in off-line systems in an attempt to pin-out the different areas that need to be tackled.

Behnke et al. [12, 13] proposed a case study on the combination of classifiers for the recognition of handwritten digits. Four different classifiers are used and evaluated; Wavelet-Preprocessing Classifier, Structural Classifier, Neural Networks Classifier, and Combined Classifier.

Previous work reviewed in this section has shown that systems concerned with digit recognition were very limited and they did not give promising results. No system to date has fully recognized the handwritten digits.

In this paper, we propose an efficient approach for extracting features for handwritten digits recognition. First, I will give an overview of structural pattern recognition. After introducing the normalization and slope estimation method used in this paper, I will discuss the feature extraction algorithm used to extract the primary and secondary features. Then, I will give an overview of the proposed recognition approach. I will also illustrate how to resolve ambiguities in some digits. Finally, I will present and discuss the experimental results and draw some conclusions.

I. Structural Pattern Recognition

There are two fundamental approaches to implement a pattern recognition system: statistical and structural. Each approach employs different techniques within the description and classification tasks which constitute a pattern recognition system. Statistical pattern recognition [17-19] draws from established concepts in statistical decision theory to discriminate among data from different groups based upon quantitative features of the data. The quantitative nature of statistical pattern recognition, however, makes it difficult to discriminate among groups based on the morphological (i.e., shape based or structural) sub patterns and their interrelationships embedded within the data. This limitation provided the impetus for the development of a structural approach to pattern recognition.

Structural pattern recognition [3, 4, 16] sometimes referred to as syntactic pattern recognition due to its origins in formal language theory, relies on syntactic grammars to discriminate among data from different groups based upon the morphological interrelationships (or interconnections) present within the data. Structural pattern recognition systems have proven to be effective for data which contain an inherent, identifiable organization such as character or digit recognition. The usefulness of structural pattern recognition systems, however, is limited as a consequence of fundamental complications associated with the implementation of the description and classification tasks. The description task of a structural pattern recognition system is difficult to implement because there is no general notion for extracting structural features, commonly called primitives, from data. The lack of a general approach for extracting primitives puts designers of structural pattern recognition systems in an awkward position: feature extractors are necessary to identify primitives in the data, and yet there is no established methodology for deciding which primitives to extract.

The result is that feature extractors for structural pattern recognition systems are developed to extract either the simplest and most generic primitives possible or the dominant application specific primitives that best support the subsequent classification task. Neither scheme is optimal. Simplistic primitives are domain independent, but capture a minimum of structural information and postpone deeper interpretation until classification. At the other extreme, dominant application specific primitives can be developed with the assistance of a domain expert, but obtaining and formalizing the necessary domain knowledge, called knowledge acquisition, can be problematic [20].

To avoid the overhead of knowledge acquisition, structural pattern recognition systems rely on morphological features that have been established in the literature as being particularly effective for the domain under analysis.The classification task of a structural pattern recognition system is difficult to implement because the syntactic grammars embody the precise criteria which discriminate among groups and, therefore, they are by their very nature dominant application specific. Grammar inference techniques can be used to construct automatically a grammar from examples, but these methods can fail in the most general cases such as when the target grammar is context free. Consequently, existing structural pattern recognition systems are primarily applied to domains where the syntactic grammars required for classification can be constructed by hand.

III. The Proposed System

The proposed method for recognizing online handwritten digits is shown in Figure 1. The process can be divided into five main stages:

1. Normalization and Slope Estimation.
2. Feature Extraction: Primary Primitives and Secondary primitives.
3. Sorting Primitives.
4. Identifying the Digit.
5. Distinguish Ambiguous Digits.

A. Normalization and Slope Estimation

The user draws the digit on a special window using a mouse. Then the coordinates (x, y) of the pixels representing the drawn digit are saved on a file. These coordinate values are used for calculating and normalizing slope values. Successive slope values are then used to record the change of direction which used to estimate the slope [14].

The signs of the slope values, the zero values, and the infinity values are saved and used in the feature extraction

step. Figure 2 shows an example, the representation of the digit 2.

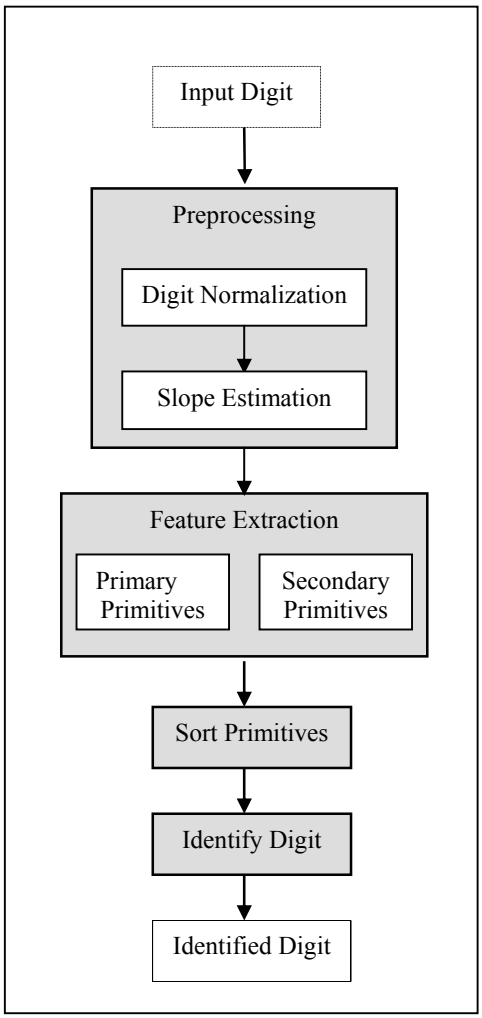

Fig. 1 Block Diagram of the Proposed System

Now, all primitives representing each digit are extracted. These primitives are identified by locating break points in the digit. Two types of break points are identified: Primary Break Points (PBP): slope values of infinity (∞) and Secondary Break Points (SBP): slope values of zero.

After the primitives are extracted, the digit is normalized according to the zero and infinity values. The purpose of this step is to eliminate any distortion that might occur during the drawing process and to ease the primitive identification process and guarantee accurate identification. Two steps of normalization are done:

1. Removing redundant break points: Eliminating adjacent reference points of the same type, except the first one. This step is required to record the change of the signs; only one break reference point is needed.

2. A threshold value is used to determine the distance (number of slope value signs) between any two recursive reference points of the same type.

Figure 3 shows an example of these two steps for the digit 1. In step 1, the adjacent break points are removed, then in step 2 the threshold value was used to remove more

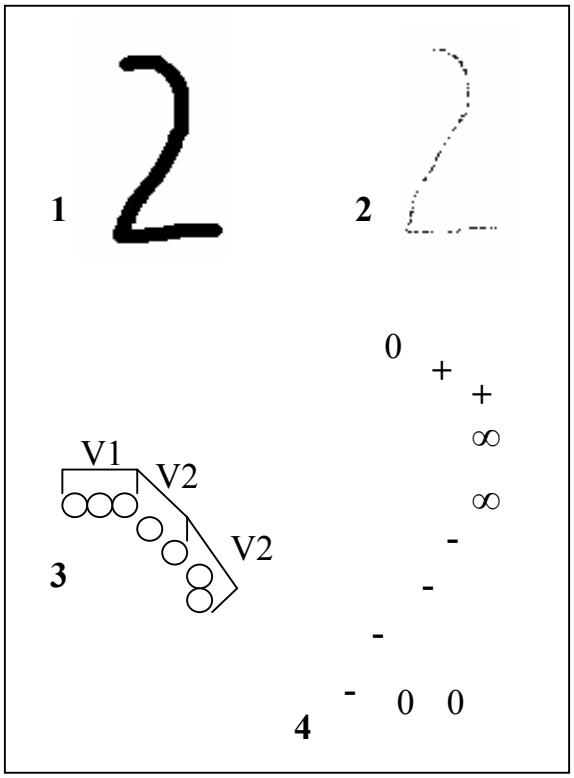

Fig. 2 Representation of the digit 2

redundant break points since the distance is less than the threshold value. The result is only one break point that will be used in the primitive identification process.

The final step in the normalization and slope estimation is to compute three additional values in order to complete the recognition process. These values are: The X and Y positions for the middle pixel which its neighbors used to calculate the slope, and the sign of ΔY.

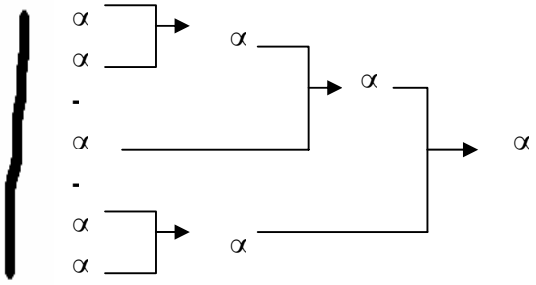

Fig. 3 Removing Redundant Break Points

The sign of ΔY is used to determine the direction of writing (drawing) the digit (upward or downward). The final representation of the digit will be a list of vectors, V_1, V_2, …,Vn each vector V_i contains the following data: V_1= (slope value, Y position, X position, ΔY sign).

B. Extracting Primary Primitives

The final representation of the digit is used to extract primary primitives. Figure 4 shows these primitives (a, b, c,

d, e, or f). These primitives are called primary primitives because the primary break points (PBP) are used to identify them.

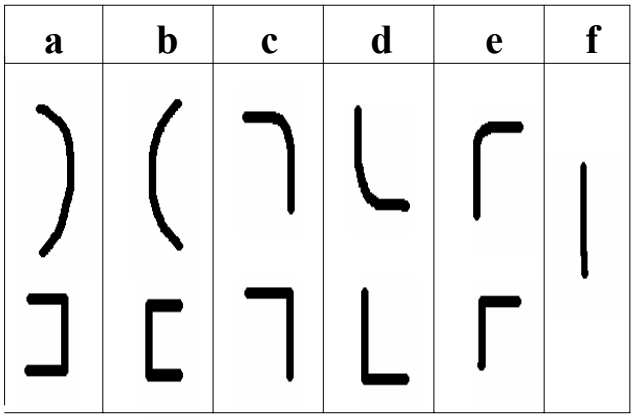

Fig. 4 Primary Primitives

Figure 5 explains this step for the digit 2. Assume that the user draw the digit downward. In this case, ΔY is greater than zero for all points, so the algorithm proceeds as follows:

1. Take the first PBP1.
2. Now, primitive "a" is recognized.
3. Find the next PBP.
4. Take the next PBP (PBP2).
5. Now, primitive "b" is recognized.
6. No More PBP, end.

Now the vector contains the primitives "ab". If the user draws the digit from bottom to top, the vector will contain the primitives "ba". We need also to extract starting and ending points for the primitives "a" and "b". This step is necessary to resolve ambiguity which will be discussed in section F.

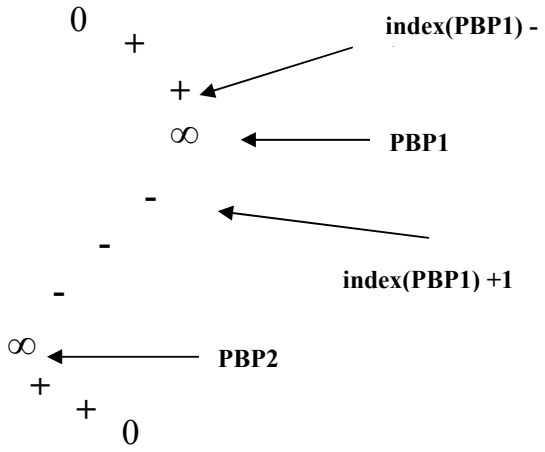

Fig. 5 Signs and Break Points for the digit 2

Each digit has different patterns which captured by the set of primitives that are described. These patterns are represented in figure 4.

After feature extraction process, we need to identify these features. The primitives which are extracted in the previous phase represent a certain string which is a production of a certain grammar. Each digit can be described by a specific string. In order to identify the digit we have to determine to which grammar the string belongs. According to the patterns in figure 4, we can write various sorted primitives. This grammar is used to identify the input digit.

C. Extracting Secondary Primitives

Secondary Break Points (SBP) are used to identify the secondary primitives which shown in figure 6. Secondary Primitive "c' " is the same as the primary primitive "c" but here there is no PBP and the primitive makes a cute angle with SBP. For example, to identify the secondary primitive "c' " we have to move forward in the representation list where the signs are negative and the X value for each new point increases, after moving a certain number of steps and the previous conditions applied, we then identify this primitive.

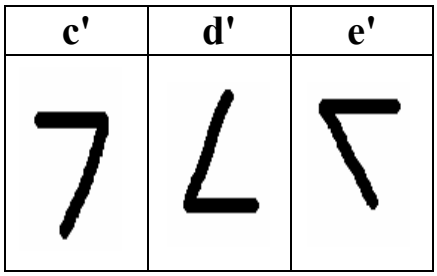

Fig. 6 Secondary Primitives

D. Sorting Primitives

At this stage, the primitives' vector contains primary and secondary primitives. The order of these primitives depends on the drawing style. For example, if the drawing style was downward then the primitives' vector will contain "ab", on the other hand if the drawing style was upward, it would contain "ba". This is confusing and increases the number of patterns for the digit.

The order is very important in translating the primitives into digits. So we need a standard order to be used in sorting all primitives. The order of the identified primitives must be independent from the drawing style and from the order of drawing the primitives. The standard order used here is the Y position for the break points; they are sorted in increasing order. After collecting all primitives, they are reordered according to the Y position for the break points that used to identify them.

E. Identifying the Digit

Each digit has different patterns which captured by the set of primitives. These primitives represent a specific string which is a production of a certain grammar. Each digit can be described by a specific string. In order to identify the digit we have to determine to which grammar the string belong. This grammar is used to identify the input digit.

According to the digit patterns, we can write various sorted primitives as shown in figure 7. This grammar is used

to identify the input digit.

Digit	Possible Representations			
0	ab	ba		
1	f			
2	ab	ad'	c'd'	
3	aa	aba	ad'a	
4	fd'	df		
5	ba	ea	e'a	
6	ba	bc	da	dc
7	c'	c	c'c'	c'e'
8	baba	baab	abba	abab
9	ba	cb	bf	

Fig. 7 Possible Digits Patterns

F. Distinguish Ambiguous Digits

As we can see in figure 7, there are multiple digits which have the same string of primitives, for example the string "ab" is common for digits 0 and 2. In this phase this ambiguity is removed, and more constrains on some digits are applied to guarantee the correct result. The key element that helps in resolving this ambiguity, is the ending points of the primitives "a" and "b", since the ambiguity results from the two strings " ab" and " ba". This step will be investigated more in future work to improve its efficiency.

II. Experimental Results and Conclusions

A new online structural pattern recognition approach has been discussed. This approach recognizes the handwritten digits; the primitives are determined by identifying the changes in the slope's signs around the zero and the infinity values (break points). This technique is independent of the type of drawing (upward or downward). A special grammar has been used to match the string of primitives to the corresponding digit.

The method is tested on an on-line dataset representing the digits 0-9 collected from 100 users. On the average, the recognition rate was about 95%. Future work considers testing the method on a larger data set to improve the effectiveness of the method.

The proposed method will be modified to deal with Arabic handwritten characters. In addition, the next important work is to add additional constraints on the primitive, for example the average length of one primitive according to another and do the primitives connected correctly or not. These constrains can guarantee an accurate results and do not directly match the resulting string of primitives to its corresponding digit unless the primitives match the digit correctly, one important note here is that, more constraints may reduce the probability of recognition. Also, a future work will be considered on solving ambiguities between digits.

Acknowledgment

The publication of this work was supported by Yarmouk University Research Council, Irbid, Jordan.

References

[1] C. C. Tappert, C. Y. Suen, and T. Wakahara, "*The state of the art in on-line handwriting recognition*", IEEE Trans. On Pattern Analysis and Machine Intelligence, vol 12, no 8, pp. 787-808, 1990.

[2] R. G. Casey and E. Lecolinet, "*Strategies in character segmentation: A survey*", In Proceedings of International Conference on Document Analysis and Recognition, pp. 1028-1033, 1995.

[3] K. S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[4] T. Pavlidis, *Structural Pattern Recognition*, Springer, New York, 1977.

[5] s. Lucas, E. Vidal, A. Amiri, S. Hanlon, and J.C. Amengual, "*A comparison of syntactic and statistical techniqes for off-line OCR*", in: R. C. Carrasco, J. Oncina (Eds.), Grammatical Inferrence and Applications (ICGI-94), Springer, Berlin, pp. 168-179, 1994.

[6] Brijesh Verma, "*A Contour Code Feature Based Segmentation For Handwriting Recognition*". Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003).

[7] Daekeun You and Gyeonghwan Kim, "*An approach for locating segmentation points of handwritten digit strings using a neural network*", Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003).

[8] Robert T. Olszewski, "*Generalized Feature Extraction for Structural Pattern Recognition in TimeSeries Data*", PhD thesis, University-Pittsburgh, 2001.

[9] Kam-Fai Chan and Dit-Yan Yeung, "*An effecient syntactic approach to structural analysis of on-line handwritten mathematical expressions*", Pattern Recognition, vol 33, pp. 375-384, 2000.

[10] Kam-Fai Chan and Dit-Yan Yeung, "*Recognizing on-line handwritten alphanumeric characters through flexible structural matching*", Pattern Recognition, vol 32, pp. 1099-1114, 1999.

[11] Adnan Amin, "*Off-Line Arabic Character Recognition: The State Of The Art*", Pattern Recognition, 31(5), 517-530, (1998).

[12] Sven Behnke, Marcus Pfisher, and Raul Rojas, "*A Study on the Combination of Classifiers for Handwritten Didit Recognition*", Proceedings of Neural Networks in Applications, Third International Workshop (NN'98), Magdeburg, Germany, pp. 39-46, 1998.

[13] Sven Behnke, Raul Rojas, and Marcus Pfister, "*Recognition of Handwritten Digits using Structural Information*", Proceedings of the International Conference of Nueral Network, Houston TX, vol 3, pp. 1391-1396, 1997.

[14] S. Madhvanath, G. Kim, and V. Govindaraju, "*Chaincode Contour Processing for Handwritten Word Recognition*", IEEE Trans. On Pattern Analysis and Machine Intelligence, vol 21, no 9, pp. 928 932, 1999.

[15] Robert Schalkoff, *Pattern Recognition: Statistical, Structural, and Neural Approaches*", John Wiley & Sons Inc. 1992.

[16] Rafael C. Gonzalez and Michael G. Thomason, *Syntactic Pattern Recognition: An Introduction*, Addison Wesley, Reading, Massachusetts, 1978.

[17] Richard O. Duda, Peter E. Hart, and David E. Stork. *Pattern Classification*. Wiley, New York, second edition, 2001.

[18] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, second edition, 1990.

[19] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. *Statistical Pattern Recognition: A Review*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):4 –37, January 2000.

[20] Robert T. Olszewski, "*Generalized Feature Extraction for Structural Pattern Recognition in TimeSeries Data*", PhD. Thesis, 2001, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.