Recognizing on-line hand written alphanumeric character Using Structural Approach

Dr. Hala Bahjat Abdual Wahab
Department of Computer Science
University of Technology
Baghdad-Iraq.
Hala_bahjat@yashoo.com

Abstract

Speed, accuracy, and flexibility are crucial to the practical use of on - line handwriting recognition. Besides, extensibility is also an important concern as we move from one domain to another which requires the character set to be extended. This research, produce a study for simple yet robust structural approach for recognizing on - line handwriting. In this approach is designed to achieve reasonable speed, high accuracy and sufficient tolerance to variations. At the same time, it maintains a high degree of reusability and hence facilitates extensibility. Experimental for study results show that the recognition rates are 98.60% for digits, 98.49% for the uppercase letters, 97.44% for lowercase letters, and 97.40% for the combined set. When the rejected cases are excluded from the calculation, the rates can be increased to 99.93%, 99.53%, 89.55% and 98.07%, respectively.

***Keywords*:** Structure extraction, structural primitives; Flexible structural matching, and On-line handwritten character recognition.

تمييزالكتابة الفورية للأرقام والحروف الأنكليزية بأستخدام الطريقة المهيكلة

هالة بهجت عبدالوهاب
مدرس
قسم علم الحاسبات
الجامعة التكنولوجية
بغداد / العراق

ملخص

السرعة ، الدقة والمرونة هي النقاط الحاسمة في الاستخدامات العملية في عملية تمييز الكتابات الخطية ، بالاضافة الى ذلك قابلية التمدد هي ايضا لها اهمية . في هذا البحث نطرح دراسه حول طريقةلتمييز تعتمد على استخدام الطريقة الهيكلية لتمييز الكتابات الخطية وقد صممت هذه الطريقة للوصول الى سرعة ودقة عالية وكفاءة على مقاومة التغيرات ،حيث كانت معدلات التمييز التي تم الحصول عليها هي ٩٨.٦٠ % للارقام ، ٩٨.٩٤ % للاحرف الكبيرة ، ٩٧.٤٤ % للاحرف الصغيرة ،و ٩٧.٤٠ % لمجوعة تدمج الانواع السابقة.

1. Character recognition

1-1 Introduction

The recognition of objects is a perceptual process and is one of the aspects on intelligence, thus machine's recognition of patterns is a natural application of artificial intelligence. Pattern recognition in digital image began to be investigated in the early days of artificial intelligence. The patterns encountered can fall into two categories: abstract and concrete. Example of abstract items includes ideas and arguments. Examples of concrete items include characters, symbols, pictures, biomedical images, three dimensional physical objects, etc. Approaches for pattern recognition may be grouped mainly into two categories: the statistical approach and the syntactic or structural approach. For some extreme problems, the statistical or structural approach is most suitable, where as for other extreme problems the syntactic app-roach is most suitable .the selection of approach depends primarily on the nature of the data set involved in a problem. For those problems where structural information is rich, it might be advantageous to use the syntactic method to show its power for problem description.

If the data involved in the problem are better expressed in terms of statistical relationships and at the same time structural information about the pattern is not considered important, the statistical method is recommended for its classification. In response to conflict between the use of the syntactic or statistical method, a third approach (the hybrid approach) appeared. The hybrid approach deals with the many applications falling half way between the two extreme cases mentioned above [1].

2. Recognition on - line Hand Written Alphanumeric Characters through Flexible Structural Matching.

2.1 Introduction

Automatic recognition of on - line handwriting provides on of the most natural ways for human beings to interact with computers without having to learn any extra skill (e.g., typing) research in this area has been active for more than three decades[1].Different approaches, such as statistical, syntactic and structural, and neural network approaches, have been proposed. Characters consist of line segments and curves. Different spatial arrangements of these elements form different characters. In order to recognize a character, we should first find out the structural relationships between the elements, which make up the character. However, in practice, the syntactic and structural approach [2, 3] suffers from several drawbacks. One of the major concerns is the need for robust extraction of primitives [4].In order to make on-line handwriting recognition feasible, at least three issues have to be considered: speed, accuracy, and flexibility [5]. High speed and accuracy are always the key characteristics of any system, while flexibility is also important here due to diverse writing styles of different writers. In other words, an ideal handwriting recognizer should be able to quickly and accurately recognize a reasonably wide range of handwriting input. most, if not all, handwriting recognition systems in existence are designed to recognize predefined character sets of finite size. Typically, they include digits, uppercase letters and lowercase letters. However, when we move from one application to another, some characters may have to be added to the character set. However, many existing methods require that complete retaining from scratch be conducted and hence previous effort wasted [6,7,8]. It would certainly be desirable to reuse what is already available as much as possible.

2.2 Picture Description Language

The picture description language (PDL) [9] is one of the earliest attempts to describe pictorial patterns using a formal language. In PDL, each primitive has two connection points, i.e., head and tail. Four binary operators, denoted by +,-, + and *, are defined to combine a pair of PDL expressions. On the other hand, the unary operator ~ is used to reverse the head and tail of a primitive or a PDL expression. Fig. 1 shows how primitives are connected using these four binary operators. By definition, a PDL expression is either.

* a primitive, or

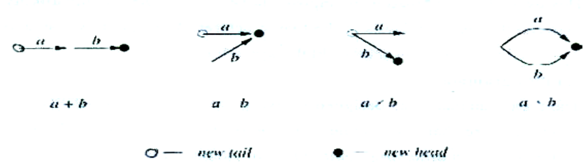

Fig. 1. Four binary operations in PDL.

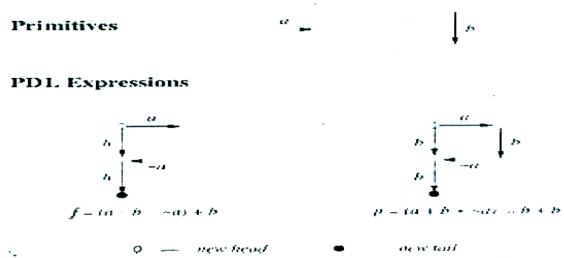

Fig. 2. Two example PDL expressions.

* A PDL expression preceded by a unary operator, or .
* Two PDL expressions connected by a binary operator.

One of the advantages of PDL is that pictorial patterns can be represented by strings, i.e., PDL expressions. Fig. 2 shows two PDL expressions which describe the structures of two characters, 'F' and 'P' . However, PDL has its limitation that all connections between primitives or complex pattern are allowed at only two points. also, the same pattern can sometimes be described by more than one PDL expression.Intuitively, the 'P' in fig. 2 could simply be represented as the concatenation of the PDL expression for 'F' and the incompatibility 'b'. however, this combination is impossible because of the incompatibility between the heads and tails of the two structures. As a result, we need to describe the letter 'P' by another PDL expression.

2.3 Plex Grammar

The plex grammar [10] was designed to overcome the limitation of PDL by allowing more than two connection points. The basic unit of the plex grammar is an n-attaching point entity, or simply nape. An identifier and a list of attaching points represent it. Structures formed by connecting the napes together are called plex structures. A plex structure consists of three components:

- A list of napes.
- A list of internal connections between napes, and
- A list of attaching points which can be used for joining the plex structure with other napes or plex structures.

Fig. 3 shows the structures of characters 'F' and 'P' in plex grammar. Three napes, h, v2 and v3, are used to

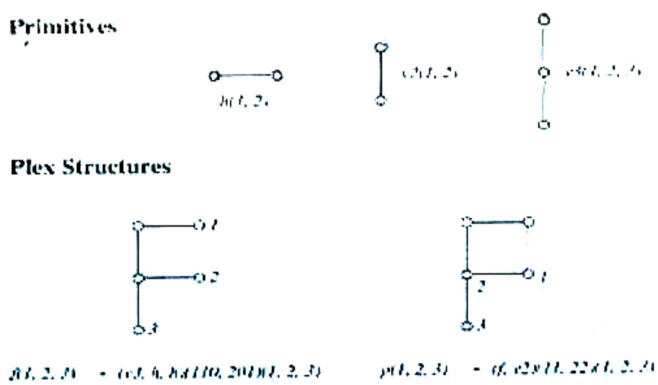

Fig. 3. Two example plex structures.

Create new plex structures. For example, the plex structure of the nape, f (1, 2, 3), is as follows :

(V3, h, h) (110,201) (one, 2, 3)

The first part is the list of napes used in the structure. The second part gives information about the two internal connections:
* The first value, 110, means that a connection exists
Between point 1 of v3 and point, 1 of the first h. note that the second h is not involved in this connection, and hence the third digit is 0.
* The second value, 201, means that there exists a connection between point 2 of v3 and point 1 of the second h. similarly, a value of 0 for the second digit shows that the first h is not involved in this connection.
The last part is the list of points, which can be used to form connections with other napes or plex structures.
The last part is the list of points, which can be used to form connections with other napes or plex structure. the plex grammar suffer from the same problem as PDL in that a pattern may be represented by several different plex structures corresponding to different orders in listing the napes.

2.4 Berthod and Maroy's Encoding Scheme

In Berthod and Maroy's encoding scheme, there are five primitives:

- straight line , denoted by T,
- positive (counter – clockwise) curve, denoted by P,
- minus (clockwise) curve , denoted by M,
- pen – lift , denoted by L,
- Cusp, denoted by R.

every characters can be represented by a string of primitives. Fig 4 shows some examples of characters, which are encoded using Berthod and Maroy's scheme. Note that some resulting strings are ambiguous. For examples, the code P has three interpretations {C, L, U}

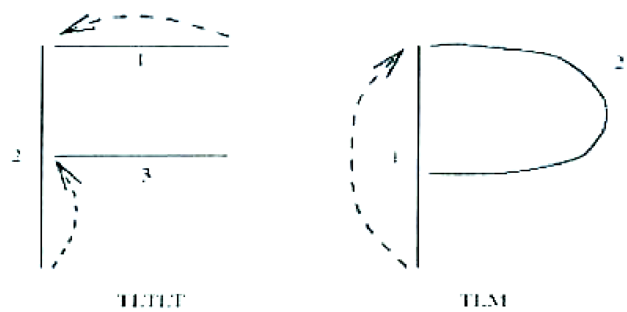

Fig. 4. Examples of some characters encoded using Berthod and Maroy's scheme.

and the code TLTLT has eight { A, F, H, I, K, N, Y, Z }. IN YHE LATTER CASE, SOME OF THE CHARACTERS ARE NOT REALLY SIMILAR AT ALL, E.G., 'H' has two vertical and one horizontal strokes while 'I' has one vertical and two horizontal strokes. the scheme is thus not very informative.

2.5 Freeman's chain code and some extended schemes

freeman's chain code is a widely used chain coding method. it consists of eight values, 0,7, which indicate how the current point is connected to the next one. fig. 5 shows the direction values in Freeman's chain code.

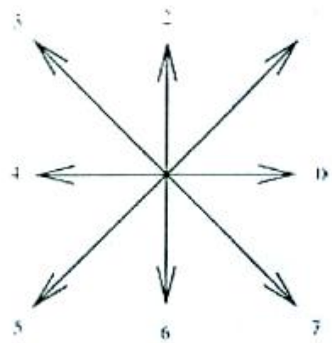

Fig. 5. Direction values used in Freeman's chain code.

the extended freeman and the improved extended freeman encoding schemes both use the eight direction values in freeman's chain code as primitives. During the preprocessing stage, all values that are the same as their preceding ones in the chain code are removed so that the resulting chain code can be shorter and at the same time preserve all the critical information. The extended freeman-encoding scheme pays special attention to the pen - lifting action by assigning a direction value (with a bar on it) to the pair of points between

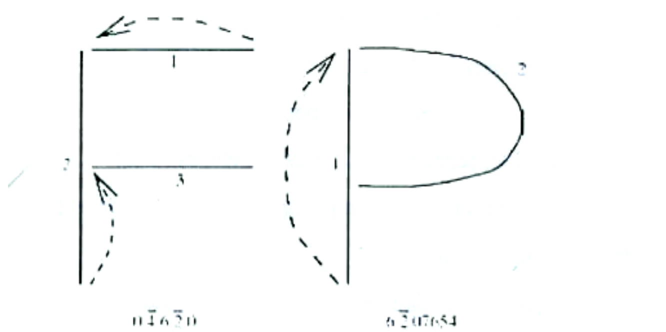

Fig. 6. Some example characters encoded using the extended Freeman encoding scheme.

Pen - up and its successive pen-down. As a result, the number of primitives is increased to 16. Note that this extended scheme still creates some ambiguous cases, but the degree is much less than that of Berthod and Maroy's scheme. Fig. 6 shows some example characters encoded using the extended freeman-encoding scheme. The improved extended freeman encoding scheme reduces the number of primitives back to eight by representing 01, 11, 2, 71 simply as 0, 1, 2, 7. The classification result remains the same as the one in the extended freeman-encoding scheme but its primitives are comparatively simpler.

2.6 Recognition Process

2.6.1 Structural primitives

Characters are composed of line segments and curves. Every line segment or curve can be extended along a certain direction. A curve that joins itself at some point forms a loop. Hence, in this representation, we will use as primitives different types of line segments and curves with some directional information. Note that a single stroke my consist of several primitives. There are five types of primitives:

- Line.
- Up (curve going counter – clockwise) ,
- Down (curve going clockwise),
- Loop (curve joining itself at some point) , and
- Dot (a very short segment which may sometimes be just noise; we, however, cannot simply ignore it since it may be part of a character, like in 'I' and 'j').

To represent the directional information, we also employ freeman's chain code, which consists of eight

Values, i.e., 0-7.

The grammar, G, which expresses the proposed structure in a string representation, is a 4-tuple G= ($VT_{,}$ $V_{N,}$ P, S) where

* $V_{T,}$ = { line , up, down, loop , dot, '{','}',',',0,1,2,3,4,5,6,7}, with 0-7 denoting the direction values in Freeman's chain code.

* V_N = {Character, stroke set, stroke, primitive set, primitive, line type, direction},

* P is a set of production rules as follows:

P= {

Character → {stroke set}
Stroke set → stroke
Stroke set → stroke, stroke set
Stroke → primitive set
Primitive set → primitive
Primitive set → primitive, primitive set
Primitive → {line type, direction}
Primitive → loop
Primitive → dot
Line type → line | up | down
Direction |1|2|3|4|5|6|7

}

And

The start symbol, S, is Character.

Note that no directional information is associated with isolated dots.

In addition, for simplicity, we do not associate

Fig. 7. Examples of the representations for some characters.

Directions with loops at this stage. The direction of a line or a curve depends on the starting and ending points.Fig.7 shows some examples.

2.6.2 Model set

Before we can perform recognition, we need to define models for the corresponding character classes. Note that all models should be distinct. In other words, there exist no two models with the same number of strokes and the same sequence of primitives in each stroke. Also, different character classes may have different numbers of models depending on the complexity of their structures. Fig. 8 shows some examples. As shown in fig. 8 , some models look very similar but have different numbers of strokes.

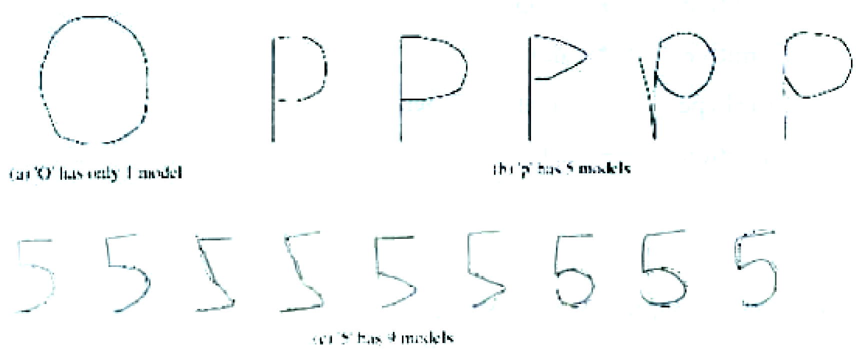

Fig. 8. Examples of the models for some character classes.

2.6.3 Recognition process

After we have written character on the digitizer, what we get is only a sequence of points. In order to recognize the character, we must first extract the structural primitives from the point sequence to form a preliminary structure. Occasionally, some kinds of reconstruction may be required if certain conditions are met. With the finalized structure, we then compare it with the models in the model set and try to find a match. Here we apply flexible structural matching in order to increase the chance of finding a match. Fig. 9 summarizes the major stages of the recognitions process.

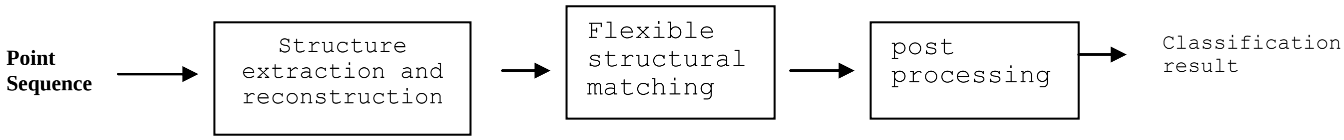

Fig9. Overview of the recognition process

2.7 Structure extraction and reconstruction

2.7.1 Structure extraction

In each character, there may be one or more strokes. Each stroke consists pen – down to pen – up in normal handwriting style. Every pair of consecutive points induces a direction. For points that follow the same direction or have a slight turn, we group them into one line segment. On the other hand, if there is a sharp turn along a stroke, we will represent it with multiple line segments. Fig. 10 shows the steps taken to extract the structure of a digit '3' in practice, some writers produce characters, which are hard to recognize. Most cases are the result of different writing habits, for example, writing strokes in a reverse or unusual order. However, some may be due to the poor quality of the hardware. As a result, zigzag line segments sometimes occur. To solve the problem, we can simply extract the mid – points of those zigzag line segments and connect them together to form a stroke. Fig. 11 shows how lines segments are eliminated during structure extraction zigzag.

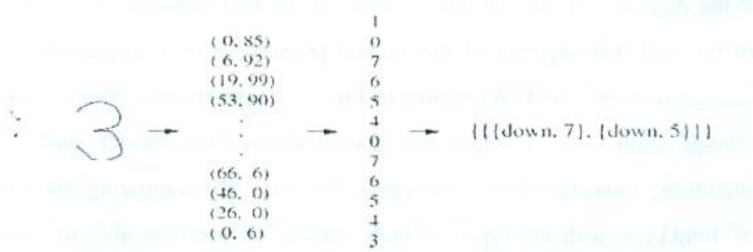

Fig. 10. Steps taken to extract the structure.

Fig. 11. Eliminating zig-zag line segments during structure extraction.

3. Conclusion

Nowadays, relatively few researchers use the structural approach for character recognition. Experiment shows that, by making use of structural information contained in a character together with a simple, flexible matching mechanism and some additional post processing procedures indeed achieve good recognition results. With this simple and robust structural approach, already have an effective and efficient on-line character recognition module.

References

[1] C.C.Tappert,C.Y.Suen,T.Wakahara,The state of the art in on-line handwriting recognition,IEEE Trans.Pattern Anal.Machine Intell.12(8)(1990)787-808.

[2] k.S. Fu,Syntactic Pattern Recognition and Applications,Prentice-Hall,Englwood Cliffs,NJ,1982.

[3] T.Pavlidis,Structural Pattern Recogntition,Springer,New York,1977.

[4] S. lucas,E. Vidal, A. Amiri, S. Hanlon, J.C. Amengual, Acomparison of syntactic and statistical techniques for off-line OCR,in : R.C. Carrasco,J. Oncina(Eds),Grammatical Inference and Application(ICGI-94), Springer,Berlin,September 199,p.168-179.

[5] C.C. Tappert, Speed, Accuracy, and flexibility tradeoffs in on-line character recognition, in: P.S.P Wang (Ed.), Character and Handwriting Recognition: Expanding Frontiers, World Scientific, Singapore, 1991, pp. 79-95.

[6] S.W. Lee, Y.J.Kim, A new type of recurrent neural network for handwritten charater recognition, Proc. 3rd Int. Conf. on Document analysis and Reconition, Montreal, Canada, 1995, pp.38-41.

[7] T.pavlidis, Structural descriptions and graph grammars, in pictorial Information Systema, Springer, New York, 1980.

[8] A. Rosenfeld, Array, tree and graph grammars, in Bunke and sanfeliu, hapter 4,pp. 85-115.

[9] A.C. Shaw, A formal picture description scheme as a basis for picture processing system, Inform. Control 14(1969) 9-52.

[10]J Feder, Plex Languages, Inform Sci. 3 (1971) 225-241.