

## Recognition of Hand Printed Characters Based on Simple Geometric Features

Sami M. Halawani and Ibrahim A. Albidewi

Department of Computer Science, Faculty of Computing and Information Technology,  
King Abdulaziz University, Jeddah, Kingdom Saudi Arabia

---

**Abstract: Problem statement:** The use of computers in information processing technology nowadays is one of the main trends of office automation. For more than four decades, information from the outside world is transferred into computers in a traditional way by keying in these raw data with the help of keyboard. Most of these data are in hand printed form and very large; therefore the use of automatic recognition technique of hand printed character by machine is essential to achieve an efficient way of high speed man-machine communication. **Approach:** Handwriting recognition had always been tough problem because of the handwriting variability, ambiguity and illegibility. This study described a simple approach involved in offline hand printed recognition. **Results:** The cost of computing had been reduced a lot by introducing a simple mechanism yet effective to extract the minimal number of geometric features for recognition's purpose. **Conclusion:** The experiment had shown that when hand printed characters are drawn clearly according to specified rules and models; it is then possible to perform recognition perfectly.

**Key words:** Hand printed character recognition, feature extraction, character dictionary, bit masking operation

---

### INTRODUCTION

Machine simulation of human writing is one of the most challenging research areas. It has been the subject of intensive research especially in unconstrained handwriting recognition. The interest devoted to this field is not explained only by the exciting challenges involved, but also the huge benefits that a system, designed in the context of a commercial application, could bring (Morita *et al.*, 2003). Two classes of recognition systems are usually distinguished: online systems (Tappert *et al.*, 1990; Namboodiri and Jain, 2004; Liu *et al.*, 2004; Bahiraie *et al.*, 2009) for which handwriting data are captured during the writing process, which makes available the information on the ordering of the strokes and offline systems (Steinherz *et al.*, 1999) for which recognition takes place on a static image captured once the writing process is over. With the increase in popularity of portable computing devices such as PDAs and handheld computers, non-keyboard based methods for data entry are receiving more attention in the research communities and commercial sector. Large number of symbols in some natural languages (e.g., Kanji contains 4,000 commonly used characters) making keyboard entry even a more difficult task.

**Problem background:** Handwriting recognition has always been tough problem. Recognition of handwritten characters by computer poses serious problems because of the high variability in the character shapes written by individuals. As people tend to adjust their handwriting style to personal preferences, the resulting variability of handwriting styles often makes reading difficult even for humans. This problem becomes even more complicated when the writer is unknown (Powalka, 1995). Moreover, pairs of characters can be formed which are ambiguous, both for human and machine recognition, for instance U-V, C-G, Q-G, D-O, F-P.

There is extensive study in the field of handwriting recognition and a number of reviews exist. General methodologies in pattern recognition and image analysis are presented in (Mantas, 1986). Character recognition is reviewed in (Suen *et al.*, 1980; Govindan and Shivaprasad, 1990; Vinciarelli, 2002; Koerich *et al.*, 2003) for off-line recognition and in (Nouboud and Plamondon, 1990; Plamondon and Sargur, 2000) for online recognition.

Numerous techniques for handwriting recognition have been investigated based on four general approaches of pattern recognition, as suggested by (Jain *et al.*,

---

**Corresponding Author:** Sami M. Halawani, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia

2000): template matching, statistical techniques, structural techniques and neural networks. Template matching operations determine the degree of similarity between two vectors (groups of pixels, shapes, or curvatures) in the feature space. Matching techniques can be grouped into three classes: direct matching (Gader *et al.*, 1991), deformable templates and elastic matching (Dimauro *et al.*, 1997) and relaxation matching (Xie and Suk, 1988; Mitoma *et al.*, 2005).

Statistical techniques are concerned with statistical decision functions and a set of optimal criteria, which determine the probability of the observed pattern belonging to a certain class. The statistical scheme is receiving increasing attention in recent years (Liu *et al.*, 2004). Statistical techniques use concepts from statistical decision theory to establish decision boundaries between pattern classes (Jain *et al.*, 2000). In structural techniques the characters are represented as unions of structural primitives. It is assumed that the character primitives extracted from handwriting are quantifiable and one can find the relationship among them. Basically, structural methods can be categorized into two classes: Grammatical methods (Shridhar and Badreldin, 1986; Aly and Abuelnasr, 2010) and graphical methods (Kim and Kim, 1999).

From an application's point of view, a recognition method can be writer-dependent or writer-independent. Writer-independent recognition is more challenging due to the diversity of writing styles. On the other hand, writer-dependent recognition allows stable recognition due to the relative stability of personal writing styles (Liu *et al.*, 2004). The representation schemes of input pattern and model database are of particular importance since the classification method depends largely on them.

### MATERIALS AND METHODS

The approach taken here is by combining elements from several methods such as globally and locally computed features to give unique feature descriptors. Initially, local information is computed from low level features, i.e., end-point and T-point, in the relative locations of the features within the character. An end-point is basically defined as a point where the stroke ends, while a T-point is described as a point where more than two strokes meet. A frame of four quadrants is then mapped onto a thinned image to determine its feature quadrant location. For example, character 'A' (Fig. 1) as to consist of two end-point; each in quadrant 3 and 4 respectively and two T-points; each in quadrant 2 and 4 respectively.

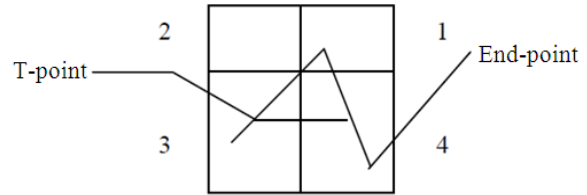


Fig. 1: Quadrants used in feature location

Table 1: Expected quadrant locations of endpoint and T-point of each letter

Letter	End-point	T-point
A	[3][4]	[1 4][2 3]
B	None	[2 3][1 4]
C	[1][4]	None
D	None	None
E	[1][4][any]	[2 3]
F	[1][3][any]	[2 3]
G	[1] (1 4)(2 3)]	[4]
H	[1][2][3][4]	[1 4][2 3]
I	[1][2][3][4]	[1 2][3 4]
J	[1][2][3]	[2 3]
K	[1][2][3][4]	None
L	[2][4]	None
M	[3][4]	None
N	[1][3]	None
O	None	None
P	[3]	[2 3]
Q	[4][4]	None
R	[3][4]	[2 3][1 4]
S	[1][3]	None
T	[1][2][3 4]	[1 2]
U	[1][2][4]	[1]
V	[1][2]	None
W	[1][2]	None
X	[1][2][3][4]	[any]
Y	[1][2][3]	[1 4]
Z	[2][4]	None

While it might be argued that the character bounding frame should be divided into a large number of sectors instead of four quadrants to give higher degree of uniqueness. It is found the division into four quadrants is enough to provide good performance and a low degree of computational complexity. After a low level feature has been identified (as an end-point or a T-point), its quadrant location is determined and this information is added to the feature descriptor.

The resulting features in feature descriptor with regard to their quadrant locations are matched to the unique stored features of standard character in the developed character dictionary. The prototype that matches closely provides recognition. Prior to this, the expected quadrant locations of end-point and T-point of each standard letter ('A' to 'Z') are determined and kept in the suggested character dictionary as a standard database (Table 1).

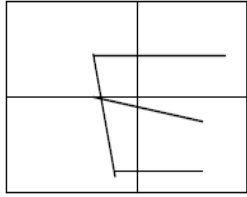


Fig. 2: Sample of hand-drawn character E

The following logical expression used with reference to quadrant 1, 2, 3 and 4. [1][2] = feature in both quadrants 1 and 2: AND, [1|2] = feature in either quadrant 1 or quadrant 2: OR, [not both] = XOR, [any] = [1|2|3|4]. It should be noted that in this implementation, the focus based features had been reduced to two, i.e. end-point and T-point. Whereas in other past implementations, more than two studied features had been based onto, yet the ambiguities were still aroused and had to be resolved by other complex computational means, such as stroke orientations, aspect ratio of the stroke, or shape curvature (Kerrick and Bovik, 1988).

The reason is that, given a set of possible input characters, even allowing for substantial variation, using only end-point and T-point features would be enough to recognize a large number of them. That is to say, a larger number of character set, considering the attribute of these features only would be enough to give their uniqueness.

However, in some cases involving very similar characters, the ambiguity cannot be resolved by only end-point and T-point features properties. The ambiguity here is referred to two characters having the same properties as far as these features are concerned. For example, character 'N' and 'S' have the same amount of end-point in the same quadrant locations. The method used for resolving this ambiguity is by using another property kept in the stroke drawn namely called corner. Corner is a single intermediate level feature (Fig. 3).

Although corner is a locally defined feature, as with low level feature, but the number of corners does exist is used as a source of global information. Corner is a rather imprecise term; by corner it means an area of sustained high stroke curvature, surrounded by area of unsustainable or low curvature on either side. Thus, a corner is less a locally defined phenomena than end-point and T-point. However, instead of determining the quadrant locations for expected corners, the possible number of corner occurrences in the pattern is checked regardless of their quadrant locations. For example, compared to character 'N', character 'S' has four expected corners, while character 'N' has only two.

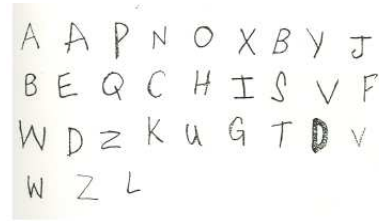


Fig. 3: Samples of off-line hand printed characters

**Recognition scheme:** Recognition process occurs as in the following sequence:

- Perform thinning algorithm on a binary image of hand drawn character
- Perform feature extraction scanning process on the skeleton image. Upon the completion of scanning process, each feature collector will has a number of quadrant locations for each collected feature and pass to pattern examiner
- Perform matching procedures: The pattern examiner matches this with the stored information in character dictionary for a targeted symbol. A matched symbol of a highly similar character is passed to a pattern examiner for corner detection to provide the recognition. The whole process is explained in what follows.

**Technique of defining character dictionary based on expected of end-point and T-point for each character of 26 letters:** Here, the concept of bit masking is used to simplify our constructed character dictionary. The concept which will be explained later, not only gives the dictionary looks simpler, but also eases the complexity of programming task especially inn feature matching logic.

The idea of bit masking is simple; for example, given a byte (8 bits) of word and masks to another word, according to logical operators used, the results are given as follows:

$$A = 10110010, B = 01001101$$

Then:

$$A \text{ and } B = 00000000 \text{ is FALSE}$$

$$A | B = 11111111 \text{ is TRUE}$$

In the implementation, the quadrant numbers are first changed to binary system, as follows:

$$\begin{array}{llll} 2 & 1 \rightarrow & 2^1 = 2 & 2^0 = 1 \\ 3 & 4 \rightarrow & 2^2 = 4 & 2^3 = 8 \end{array}$$

The character dictionary is then constructed, based on Table 1 for expected locations of end-point and T-point of 26 letters through the following operations:

- If the expected quadrant location of any specified feature is in either quadrant #1 or in quadrant #2, the possible value to be put in the dictionary for that particular feature is calculated in the following manner

$$\begin{matrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 \text{ OR} & 1 & 0 = & 1 & 0 \\ 2 & & 4 & & 2^1 + 2^2 = & 6 \end{matrix}$$

For example, character E consists of 3 endpoints at quadrant [1], [1 or 2 or 3 or 4] and [3] respectively and 1 T-point at either quadrant 2 or 3. So, in our dictionary, it would be defined as follows:

$$\begin{aligned} \text{end-point (5)} &= \{1,15,8\} \\ \text{T-point (5)} &= \{6\} \end{aligned}$$

**Note:** Number 5 in the brackets indicates the position letter E in the list.

- Develop the dictionary of 26 letters for expected quadrant locations of end-point and T-point as described in (i) as follows:

$$\begin{aligned} \text{End-point (26)} &= \{\{4, 8, \{-1\}, \{1, 8\}, \{-1\}, \{1,8,15\}, \{1,4,9\}, \{1,8,15\}, \\ &\{1,2,4,8\}, \{1,2,4,8\}, \{1,2,4\}, \\ &\{1,2,4,8\}, \{2,8\}, \{4,8\}, \{1,4\}, \{-1\}, \{4\}, \{8,8\}, \{4,8\}, \{1,4\}, \\ &\{1,2,12\}, \{1,2\}, \{1,2,8\}, \\ &\{1,2,8\}, \{1,2,4,8\}, \{1,2,4\}, \\ &\{2,8,6,9\}\} \end{aligned}$$

$$\begin{aligned} \text{T-point(26)} &= \{\{9,6\}, \{6,9\}, \{-1\}, \{-1\}, \{6\}, \{6\}, \\ &\{8\}, \{9,6\}, \{8,12\}, \{8\}, \{6,6\}, \{-1\}, \\ &\{-1\}, \{-1\}, \{-1\}, \{6\}, \{1\}, \\ &\{9,6\}, \{-1\}, \{8\}, \{-1\}, \{-1\}, \{-1\}, \\ &\{15,15\}, \{9\}, \{-1\}\} \end{aligned}$$

**Note:** Number ‘-1’ in the curly brackets indicate none.

Suppose, character E is drawn in this shape (Fig. 2) and the quadrant locations of each end-point and T-point are shown below; Its input of end-point and T-point as a candidate input to be recognized is defined as follows:

$$\begin{aligned} \text{Input-endpoint} &= \{1,8,8\} \\ \text{Input-T-point} &= \{3\} \end{aligned}$$

Table 2: The testing sequence

Input	Standard	Comments
{1,8,8} → {1,15,8}		Using bit mask operation: 1 AND 1 is TRUE, an input of 1 is removed, 1 in standard is replaced by any negative integer (-2)
{0,8,8} → {-2,15,8}		8 and 15 is TRUE, an input of 8 is removed, 15 in standard is replaced by -2
{0,0,8} → {-2,-2,8}		8 and 8 is TRUE, An input of 8 is removed, 8 in standard is replaced by -2
{0,0,0} → {-2,-2,-2}		As no more in input, then the process of matching is terminated

The matching of those collected features with the stored ones in character dictionary is taken place in the following sequence. Firstly, an input of end-point is passed to test function to check whether or not, its input meet to those feature kept in the dictionary, starting with index number 1 (i.e. index of letter A) and it would last to the position of targeted symbol.

At this junction, let us see as to how an input element is tested one after another. Suppose, an input of end-point for character E drawn as previously defined, {1,8,8}, is taken into our consideration and it would be matched to the standard ones, {1,15,8}; stored in the dictionary. The testing sequence, is shown in the following steps (Table 2).

If the test for an input of end-point is succeeded, then an input of T-point is passed to test function to check whether or not, at particular index, its input also matches to the standard ones kept in the dictionary. If the test function returns TRUE for both inputs at the same index, then the matched symbol is checked whether it belongs to the group of uniquely characters or not. If not, the ambiguity of the involved letters will be resolved by locating the occurrences of corner in the skeleton.

## RESULTS AND DISCUSSION

More than one hundred test specimens that cover all characters (‘A’ to ‘Z’) were collected to study the effectiveness of the proposed technique (Fig. 3 shows some of the scanned hand printed characters).

The size of the binary digitized characters was set to 25×25 pixels. The characters were hand printed with different style of shapes and variations in normal accepted handwriting styles that could easily recognizable by human being. The smoothing, thinning and reshaping algorithm had been well performed on numerous types of digitized patterns and the results are satisfactory. The experiment has shown that when hand printed characters are drawn clearly according to specified rules and models; it is then possible to perform recognition perfectly.

Table 3: Result of feature analysis, corner detection and recognition performed

Pattern	End-point	T-point	Corner	RC
1	3,4]	3,4	2	A
2	none	3,4	-	B
3	1,4	None	-	C
4	None	None	2	D
5	1,4,4]	3	-	E
6	1,3,4]	3	-	F
7	1,1,4	4	2	G
8	1,2,3,4	3,4	-	H
9	1,2,3,4	2,4	-	I
10	1,2,3	1	2	J
11	1,2,3,4	2,2	-	K
12	2,4	None	1	L
13	3,4	None	-	M
14	1,3	None	0	N
15	None	None	4	O
16	3	2	-	P
17	1	4	-	Q
18	3,4	3,4	4	R
19	1,3	None	5	S
20	1,2,4	1	0	T
21	1,2,4	1	1	U
22	1,2	None	2	V
23	1,2	None	5	W
24	1,2,3,4	1	-	X
25	1,2,4	4	-	Y
26	2,4	None	3	Z

The proposed feature extraction mechanism had been successfully developed to cater all possible type of T-point. The algorithm has been tested with even the worst shape of T-point that may occur in a skeleton. As an endpoint is initially defined as any black pixel which has no more leading black pixel, then its extraction does not cause any problem. Since, in the proposed method, corner is vital information used to resolve the ambiguity problem which involve in a highly similar character, then any applied corner finding algorithm should be able to give enough information about the existence of corner in the skeleton. Table 3 shows in detail, the results of feature analysis and recognition done for a tested symbol shown in Fig. 3.

**Recognizable Character (RC):** About 99% true recognition rates have been achieved in this experiment. It should be noted that the time taken to recognize one letter is less than 1 sec.

### CONCLUSION

In this study, a new deterministic approach which is based on the concept of using minimal character features set has been discussed. The proposed method has introduced only two easily detectable features, i.e., end-point and T-point, to perform classification. While highly similar characters in term of having same

properties of end-point and T-point is resolved by the number of corner occurrences in the skeleton. The result is satisfactory and the cost computing is less.

### REFERENCES

Aly, W.M. and M.S. Abuelnasr, 2010. Electronic design automation using object oriented electronics. Am. J. Eng. Applied Sci., 3: 121-127. DOI: 10.3844/ajeassp.2010.121.127

Namoodiri, A.M. and A.K. Jain, 2004. Online handwritten script recognition. IEEE Trans. Patt. Anal. Mach. Intell., 26: 124-30. DOI: 10.1109/TPAMI.2004.1261096

Bahiraie, A., N.A. Ibrahim, A.K.M Azhar and M.B. Ismail, 2009. Robust logistic regression to static geometric representation of ratios. J. Math. Stat., 5: 3: 226-233. DOI: 10.3844/jmssp.2009.226.233

Liu, C.L., S. Jaeger and M. Nakagawa, 2004. Online recognition of Chinese characters: The state-of-the-art. IEEE Trans. Patt. Anal. Mach. Intell., 26: 198-213. DOI: 10.1109/TPAMI.2004.1262182

Dimauro, G., S. Impedovo, G. Pirlo and A. Salzo, 1997. Automatic bankcheck processing: A new engineered system. Int. J. Patt. Recog. Artif. Intell., 11: 467-503.

Gader, P., B. Forester, M. Ganzberger, A. Billies and B. Mitchell *et al.*, 1991. Recognition of handwritten digits using template and model matching. Patt. Recog., 5: 421-431. DOI: 10.1016/0031-3203(91)90055-A

Govindan, V.K. and A.P. Shivaprasad, 1990. Character recognition-a review. Patt. Recog., 23: 671-683. DOI: 10.1016/0031-3203(90)90091-X

Jain, A.K., R.P.W. Duin and J. Mao, 2000. Statistical pattern recognition: A review. IEEE Trans. Patt. Anal. Mach. Intell., 22: 4-37. DOI: 10.1109/34.824819

Kerrick, D.D. and A.C. Bovik, 1988. Microprocessor-based recognition of handprinted characters from a tablet input. Patt. Recog., 2: 525-537. DOI: 10.1016/0031-3203(88)90011-8

Kim, H.Y. and J.H. Kim, 1999. Handwritten Korean Character Recognition Based on Hierarchical Random Graph Modeling. In: Advances in Handwriting Recognition, Lee S.W. (Ed.). World Scientific Publishing, Singapore, ISBN: 10: 9810237154, pp: 347-356.

Koerich, A.L., R. Sabourin and C.Y. Suen, 2003. Large vocabulary off-line handwriting recognition: A survey. Patt. Anal. Appli., 6: 97-121. DOI: 10.1007/s10044-002-0169-3

- Mitoma, H., S. Uchida and H. Sakoe, 2005. Online character recognition based on elastic matching and quadratic discrimination. Proceedings of the 8th International Conference on Document Analysis and Recognition, Aug. 29-Sept. 1, IEEE Xplore Press, Seoul, Korea, pp: 36-40. DOI: 10.1109/ICDAR.2005.178
- Mantas, J., 1986. An overview of character recognition methodologies. *Patt. Recog.*, 19: 425-430. DOI: 10.1016/0031-3203(86)90040-3
- Morita, M., R. Sabourin, F. Bortolozzi and C.Y. Suen, 2003. A recognition and verification strategy for handwritten word recognition. Proceedings of the 7th International Conference on Document Analysis and Recognition, Aug. 3-6, IEEE Xplore Press, Edinburgh-Scotland, pp: 482-486. DOI: 10.1109/ICDAR.2003.1227712
- Nouboud, F. and R. Plamondon, 1990. On-line recognition of handprinted characters: Survey and beta tests. *Patt. Recog.*, 23: 1031-1044. DOI: 10.1016/0031-3203(90)90111-w
- Powalka, R.K., 1995. An algorithm toolbox for online cursive script recognition. Ph.D. Thesis. The Nottingham Trent University.
- Plamondon, R. and S.N. Sargur, 2000. Online and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. Patt. Anal. Mach. Intell.*, 22: 63-84. DOI: 10.1109/34.824821
- Shridhar, M. and A. Badreldin, 1986. Recognition of isolated and simply connected handwritten numerals. *Patt. Recog.*, 19: 1-12. DOI: 10.1016/0031-3203(86)90025-7
- Steinherz, T., E. Rivlin and N. Intrator, 1999. Offline cursive script word recognition-a survey. *Int. J. Docum. Anal. Recog.*, 2: 90-110. DOI: 10.1007/s100320050040
- Suen, C.Y., M. Berthod and S. Mori, 1980. Automatic recognition of hand printed character-the state of the art. *Proc. IEEE*, 68: 469-487. DOI: 10.1109/PROC.1980.11675
- Tappert, C.C., C.Y. Suen and T. Wakahara, 1990. The state of the art in on-line handwriting recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12: 787-808. DOI: 10.1109/34.57669
- Vinciarelli, A., 2002. A survey on offline cursive word recognition. *Patt. Recog.*, 35: 1433-1446. DOI: 10.1016/s0031-3203(01)00129-7
- Xie, S.L. and M. Suk, 1988. On machine recognition of hand-printed Chinese character by feature relaxation. *Patt. Recog.*, 21: 1-7. DOI: 10.1016/0031-3203(88)90066-0