Recognition of Handwritten Characters from Trend Codes

Carlos Ditmeyer
and
Rajjan Shinghal

Department of Computer Science
Concordia University
1455 de Maisonneuve Blvd. W.
Montreal, Quebec
CANADA H3G 1M8

Abstract

A procedure to recognize handwritten characters is proposed. Trend codes encoding the major changes in the shape of the character are used as features. Experiments have shown the procedure to give good results.

KEYWORDS: Optical character recognition, features, compass codes, trend codes.

1. Introduction

Optical character recognition (OCR) of handwritten characters has been an area of active research for many years. In this, binary images of alphanumeric characters are fed into a computer, which then follows some procedure to recognize the character classes of the patterns fed in. Different researchers have adopted different procedures for OCR. Berthod and Maroy [1] encoded a character into a string of five primitives using angular segmentation for on-line recognition. Granlund [4] used Fourier transformation to extract features for classifying a pattern. Lam and Suen [6] extracted convex polygons and line segments as geometric primitives for a decision tree. Loy and Landau [7] employed segmentation in feature extraction for on-line recognition. Mantas and Heaton [8], and Pavlidis and Ali [10] analysed the pattern's shape by polygonal approximation. Persoon and Fu [11] developed a distance measure based on Fourier descriptors. Sarvarayudu and Sethi [12] described a boundary scalar transformation technique. Shen-Pei Wang [13] discussed concepts of recognizability, learnability and ambiguity, and used Freeman's [3] chain code for on-line handwritten character recognition. Shridhar and Badreldin [14] described a two-stage OCR, which first converted the contour points of a character's shape into a Fourier series, and then used topological descriptors to resolve the ambiguity of rotational invariance. Toussaint and Donaldson [15] based their recognition on Mason and Clemens [9] extremum points in the contour of a pattern.

In this paper, we describe a technique of extracting compass codes from the contour of a pattern. The compass codes are condensed into trend codes, which give the major directional trends in the shape of a character. These trend codes are then used as features to classify a pattern. In Section 2, we describe our feature extraction scheme. Section 3 describes the recognizer, which uses the features extracted. Our experiment results are given in Section 4. Concluding remarks are given in Section 5.

2. Feature Extraction

A. Locating Paramount Pixels

The pattern is enclosed in a rectangle called the characteristic rectangle such that neither the topmost row, the bottom-most row, the leftmost column, nor the rightmost column in the characteristic rectangle is entirely blank. Thus the characteristic rectangle is the smallest rectangle that encloses the pattern, there being no blank rows or columns surrounding the pattern.

The rows in the characteristic rectangle are numbered from top to bottom as 1 to **H**, and the columns are numbered from left to right as 1 to **W**, where **H** is called the height, and **W** the width of the characteristic rectangle. Thus, for example, **H** is 23, and **W** is 20 in Figure 1.

We begin by selecting twelve pixels: L_1, L_2, and L_3 along the left edge of the pattern; T_1, T_2, and T_3 along the top edge of the pattern; R_1, R_2, and R_3 along the right edge of the pattern; and B_1, B_2, and B_3 along the bottom edge of the pattern. These twelve pixels are called paramount pixels and their purpose will be discussed later. Below, we delineate the sequence in which these pixels are labeled. The values obtained from the computation in the formulas mentioned below were all rounded off.

(1) The paramount pixels L_1, L_2 and L_3 are the leftmost black pixels on, respectively, the $(H/4)^{th}$, $(H/2)^{th}$, and $(3*H/4)^{th}$ rows of the characteristic rectangle. Accordingly in Figure 1, we note that L_1, L_2 and L_3 are respectively on the 6^{th}, 12^{th}, and 17^{th} rows.

(2) The paramount pixels R_1, R_2 and R_3 are the corresponding black rightmost pixels on the rows that contain L_1, L_2 and L_3.

(3) The portion of the pattern from its topmost row to the row containing L_2 and R_2, is known as the top sector of the pattern. Thus in Figure 1, the black pixels in rows 1 to 12 constitute the top sector of the pattern. Let the leftmost (rightmost) nonblank column of the top sector of the characteristic rectangle be A_L (A_R). Then the paramount pixels T_1, T_2 and T_3 are the topmost black pixels in, respectively, the $((3*A_L + A_R)/4)^{th}$, $((A_L + A_R)/2)^{th}$, and $((A_L + 3*A_R)/4)^{th}$ columns. Accordingly in Figure 1, where $A_L = 4$ and $A_R = 20$, we note that T_1, T_2 and T_3 are respectively on the 8^{th}, 12^{th} and 16^{th} columns.

(4) The portion of the pattern not in the top sector is said to be the bottom sector of the pattern. Thus in Figure 1, the black pixels in rows 13 to 23 constitute the bottom sector of the pattern. Let the leftmost (rightmost) non-blank column of the bottom sector of the characteristic rectangle be D_L (D_R). Then the paramount pixels B_1, B_2 and B_3 are the bottommost black pixels in, respectively, the $((3*D_L + D_R)/4)^{th}$, $((D_L + D_R)/2)^{th}$, and $((D_L + 3*D_R)/4)^{th}$ columns. Accordingly in Figure 1, where $D_L = 1$ and $D_R = 18$, we note that B_1, B_2 and B_3 are respectively on the 5^{th}, 10^{th} and 14^{th} columns.

As will become clear from our discussion later, the paramount pixels serve to monitor the directional trend of the outer contour of the pattern. Note that the L's, T's, R's and B's are distributed on their corresponding edges as evenly as possible. The top sector and the bottom sector of a pattern were considered separately to locate the T's and the B's, because quite often patterns of handwritten characters have been observed to be inclined: in other words, the bottom sector is often not directly below the top sector. Moreover, considering the top sector separately from the bottom sector is useful for patterns whose upper width is different from the lower width; for example the pattern of the character "T".

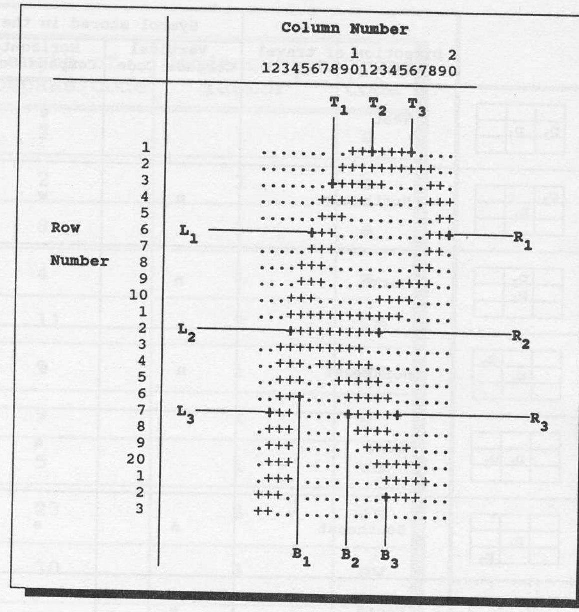
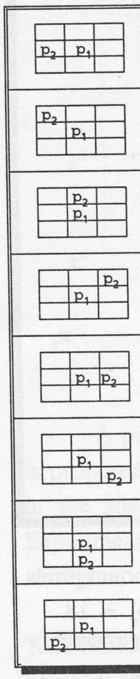


Figure 1. A specimen pattern and its paramount pixels. Note that $A_L = 4$, $A_R = 20$, $D_L = 1$ and $D_R = 18$.

B. Compass Codes and Trend Codes

To obtain compass codes for the segment between any two paramount pixels, say R_2 and L_2, we trace clockwise along the outer contour [2] of the pattern from R_2 to L_2. All the pixels in the outer contour are flagged. Tracing from a pixel p_1 on the contour to another pixel p_2 on the contour, we can be travelling in any one of the eight directions; west, northwest, north, northeast, east, southeast, south, and southwest.

This direction of travel is recorded in two types of codes called the horizontal and the vertical compass codes, as shown in Figure 2. Thus for example, if we



	Symbol stored in the	
Direction of travel from p_1 to p_2	Vertical Compass Code	Horizontal Compass Code
West		w
Northwest	n	w
North	n	
Northeast	n	e
East		e
Southeast	s	e
South	s	
Southwest	s	w

Figure 2. The symbols stored in the vertical and compass codes depending on the direction of travel while tracing clockwise along the outer contour the pattern. We are travelling from p_1 to p_2.

travel northeast, we record the symbol **n** in the vertical compass code and the symbol **e** in the horizontal compass code; if however we travel, say, west, we record the symbol **w** in the horizontal compass code and nothing in the vertical compass code. We similarly record the symbol **s** for travelling south, and the symbol **w** for travelling west.

Thus, for example, in Figure 1, we obtain the following codes between R_2 and L_2: the horizontal compass code contains the thirty symbols

wweeeeweeeewwwwwwwwwwwwwwwwwee

and the vertical compass code the thirty nine symbols

sssssssssssnnnnnnnnnnssssssssssnnnnnnnnnn.

We say that between R_2 and L_2 the length of the horizontal compass code is 30, and the length of the vertical compass code is 39.

We define a variable **K** to be the <u>scaling index</u>. If **L** is the length of a code, then we define the <u>condensation factor</u> **C** for that code to be:

$$\max(1, \text{rounded\_off}(L/K)).$$

We scan the compass code, and if the code contains **C** or more occurences of the same symbol, we store a single occurence of that symbol in the trend code. If the compass code contains fewer than **C** occurences of the symbol, then we store nothing in the trend code. Suppose the scaling index **K** is 8. Then the condensation factor for the horizontal compass code between R_2 and L_2, whose length as mentionned above is 30, will be

$$C = \max(1, \text{rounded\_off}(30/8)) = 4.$$

Accordingly, the corresponding horizontal trend code is **ew**.

To get an intuitive understanding for obtaining the trend code, let us examine the above horizontal compass code. It contains sequentially 2w's, 4e's, 1w, 4e's, 17w's and 2e's. Nothing is stored in the horizontal trend code for the first two **w**'s (its occurences are fewer than C = 4). They are in effect ignored. For the next sequence of **e**'s, we store a single **e** in the trend code. So on we proceed. Thus, the horizontal trend code **ew** tells us that to travel clockwise from R_2 to L_2, we essentially travel first toward east, and then toward west. Thus the trend code gives the major trends of the contour between R_2 and L_2, ignoring any minor directional fluctuations. Similarly, the vertical trend code between and R_2 and L_2 is **snsn**.

The value of the scaling index is a heuristic; it is obtained by experimentation. The larger the value of **K**, the more is the detail we are considering in the code. The value of **C**, which is a function of **L** and **K**, governs the magnitude of the noise in the form of minor directional fluctuations that we ignore in the contour of the pattern.

It should be clear that one complete travel around the outer contour of a pattern will give us the horizontal and vertical compass codes between whatever pairs of paramount pixels we want. From the compass codes we can then obtain the corresponding trend codes. As wil become apparent in Section 3, a horizontal trend code between two paramount pixels, say R_2 and L_2, constitutes one feature extracted from the pattern. Similarly a vertical trend code between the two paramount pixels constitutes another feature of the pattern. Thus, Figures 3 and 4 show, for example, 24 features extracted from the pattern of Figure 1.

Segment	Horizontal Compass Code	Length of compass code	Condensation factor	Trend Code
L_3L_1	eee	3	1	e
L_1T_1	ee	2	1	e
T_1T_3	eeeeeeee	8	1	e
T_3R_1	eeee	4	1	e
R_1R_3	wwwwwwwweee	11	2	we
R_3B_3	weeeewwww	9	1	wew
B_3B_1	wwwwwwwww	9	1	w
B_1L_3	wwwwe	5	1	we
L_2R_2	eeeeeeeeeeeeeeeewwwwww w	23	3	ew
R_2L_2	wweeeeweeeeewwwwwwwwwwww wwwwwwee	30	4	ew
T_2B_2	eeeeeeeewwwwwwwwwweeeew eeeewwwwwww	34	4	ewew
B_2T_2	wwwwwwwweeeeeeeeeee	19	2	we

Figure 3. The horizontal compass and trend codes for the pattern of Figure 1, when the scaling index is 8.

Segment	Vertical Compass Code	Length of compass code	Condensation factor	Trend Code
L_3L_1	nnnnnnnnnn	10	1	n
L_1T_1	nnn	3	1	n
T_1T_3	nn	2	1	n
T_3R_1	sssss	5	1	s
R_1R_3	sssssssssss	11	2	s
R_3B_3	sssss	5	1	s
B_3B_1	nnnnnnnnnsss	12	2	ns
B_1L_3	sssssssnnnnnn	13	2	sn
L_2R_2	nnnnnnnnnnssssssssssss	22	3	ns
R_2L_2	sssssssssnnnnnnnnnnsss sssssssnnnnnnnnnn	39	5	snsn
T_2B_2	sssssssssssssssssssssn nnnn	26	3	sn
B_2T_2	nnnsssssssssssnnnnnnnnn nnnnnnnnnnnnn	35	4	nsn

Figure 4. The vertical compass and trend codes for the pattern of Figure 1, when the scaling index is 8.

C. Loop Detection

Detecting the number of loops in a pattern can be useful in limiting the number of character classes to which the pattern is likely to belong. For example, "C" and "G" contain zero loops, "A" and "D" contain one loop each, and "B" and "8" contain two loops each.

To detect the loops in a pattern, we scan the pattern rowwise left to right starting from the topmost row. If we meet any transition from an unflagged black pixel to white pixel (as mentioned earlier, all pixesl in the outer contour had been flagged), then we have reached a loop, so we trace along the loop. All pixels along the loop are also flagged. We repeat this procedure until we find all the loops of the pattern. If the length of the inner contour of a loop is smaller than $1/7^{th}$ (value derived experimentally) the outer contour length, then the loop is ignored as noise.

3. Formulating The Recognizer

Having extracted the number of loops in a pattern, we know the possible character classes to which the pattern may belong. These are called the candidate classes for the pattern. Then, we use the trend codes in a Bayesian classifier to recognize a given pattern.

A. The Bayesian Classifier

From each pattern we extract horizontal trend codes between $\mathbf{m} \geq 1$ pairs of paramount pixels, and similarly we extract $\mathbf{m}$ vertical trend codes. These constitute a $2\mathbf{m}$ element feature vector $X = (x_1, x_2, \ldots, x_{2m})$. The probability of recognizing an unknown X is maximized by selecting that character class θ, which maximizes the a posteriori probability $P(\theta|X)$. By Bayes's rule

$$P(\theta|X) = \frac{P(X|\theta)\ P(\theta)}{P(X)}$$

Since the patterns we were recognizing were those of isolated characters, we assumed the prior occurrence of all characters as equiprobable. Moreover, since P(X) is not a function of θ, we can say from the above equation that maximizing $P(\theta|X)$ is equivalent to maximizing $P(X|\theta)$.

To simplify the evaluation of $P(X|\theta)$, we assumed conditional independence among x_1 to x_{2m}; that is

$$P(X|\theta) = \prod_{j=1}^{2m} P(x_j|\theta)$$

We found experimentally that our classifier could recognize most patterns correctly. Nevertheless there were two pairs of characters which could be confused for each other. For the candidates classes with no loops, confusion could occur between a "V" and a "Y", and for candidate classes with one loop, confusion could occur between "0"(zero) and a "O"(oh). So, if our recognizer output was any of these four characters, we incorporated further tests as described below.

To distinguish a "V" from a "Y" we observed the relative position of the rows that contain the paramount pixels $\mathbf{L_2}$ and $\mathbf{T_2}$. If the row containing $\mathbf{T_2}$ was above the row containing $\mathbf{L_2}$, then it was a "Y"; otherwise it was a "V". To get an intuitive reason for this criterion, see Figure 5.

To distinguish a "0"(zero) from an "O"(oh), we examined the writing habits of people. We found that "oh's" are more circular than zeros (see Figure 6). So if the **(H/W)** of the character was greater than a threshold then it was a zero, else an "oh". Experimentally, we found that this threshold to be 2.

4. Experimental Results

We tested our recognizer on sets of characters ("A" to "Z" and "0" to "9") handwritten unconstrained by 18 students of Concordia University in Montreal, each student writing one sequence of characters from "A" to "Z" and "0" to "9". After the characters were digitized by converting into binary patterns, we preprocessed the characters to remove any salt or pepper noise [5].

The patterns were split into two disjoint sets. One set, called the training set, with 432 patterns was used to train the recognizer. The other set, called the testing set, with 216 patterns was used to test the performance of the recognizer on patterns different from those it was trained on. We conducted fourteen experiments. For each experiment we selected different pairs of paramount pixels to extract the trend codes. See Figure 7 for the results of these experiments. Beside each experiment, we pictorially show the pairs of

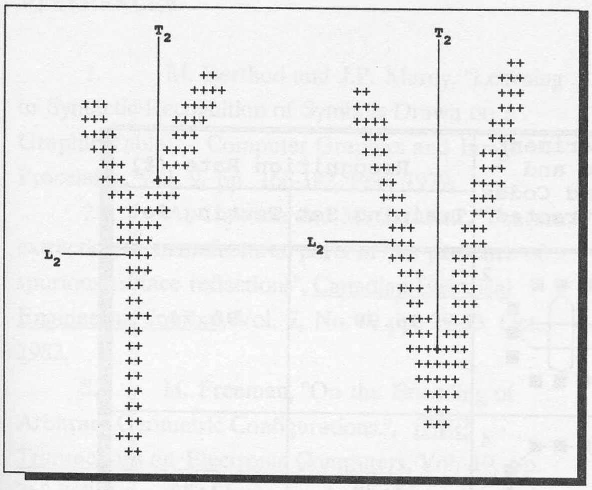


Figure 5. In the pattern of a "Y", the row containing T_2 is above the row containing L_2. In the pattern of a "V", the row containing T_2 is lower than the row containing L_2.

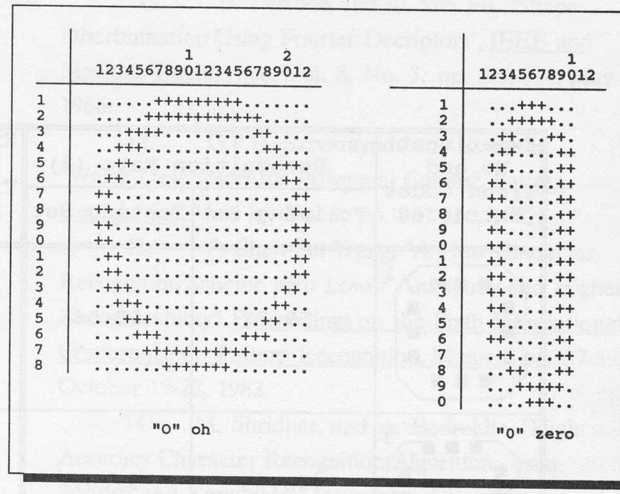


Figure 6. According to the observed writing habits of people, the "O"(oh) as shown on the left above is usually more circular than the "0"(zero) as shown on the right above. Hence **H/W** ratio for "oh" was larger than that for zero.

paramount pixels we selected to extract the trend codes. To illustrate, for Experiment 14, we selected 12 vertical trend codes and 12 horizontal trend codes between the paramount pixels as listed in Figures 3 and 4. For this experiment, with a scaling index of 16, the recognition rate for the training set was 100%, and for the testing set 99.54%. As mentioned in Section 2, the larger the scaling index, the more is the detail we are considering in the directional trends of the contour. The level of detail considered can affect the recognition rate. For example, with a scaling index of 6, the recognition rate for the training set was 99.07%, and for the testing set 97.69%. On an PC-XT running at 4MHz, we were able to classify one pattern in 0.62 sec. On faster machines we expect the time to drop considerably.

We notice that it is useful to have some overlap in the trend codes extracted as can be seen in Experiment 14 and some of the other experiments given in Figure 7. A trend code between paramount pixels that are close to each other gives local trends, whereas the trend code between paramount pixels that are distant from each gives somewhat overall trends. Both the local and the overall trends are useful features in recognizing a pattern. This leads to an overlap between trend codes.

5. Concluding Remarks

We have described our technique to extract features in the form of trend codes. Experimental resuls have shown that our technique gives a good reconition rate for unconstrained handwritten characters.

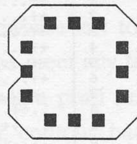
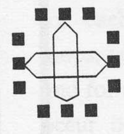
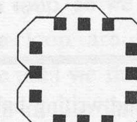
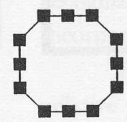
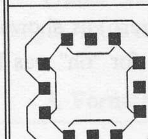
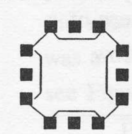
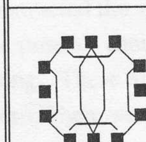
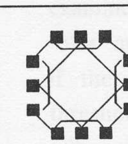
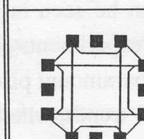
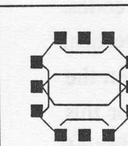
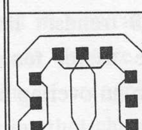
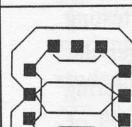
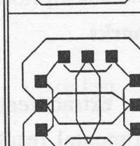
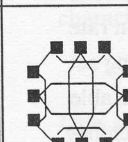
Experiment No and Trend Codes Extracted	Recognition Rate (%) Training Set	Testing Set	Experiment No and Trend Codes Extracted	Recognition Rate (%) Training Set	Testing Set
 1	59.72	55.56	 2	96.99	90.74
 3	98.15	91.67	 4	99.54	94.44
 5	99.77	97.69	 6	99.54	95.37
 7	99.54	98.15	 8	100.0	97.69
 9	99.77	98.61	 10	99.77	99.07
 11	99.77	99.07	 12	100.0	99.07
 13	100.0	99.54	 14	100.0	99.54

Figure 7. Results of the fourteen experiments. The first and the fourth column pictorially show the pairs of paramount pixels selected to extract the trend codes. Thus for the second experiment, for example, trend codes were derived between the four pairs of paramount pixels; L_2R_2, R_2L_2, T_2B_2, and B_2T_2.

REFERENCES

1. M. Berthod and J.P. Maroy, "Learning in Syntactic Recognition of Symbols Drawn on a Graphic Tablet", Computer Graphics and Image Processing, Vol. 9, pp. 166-182, Feb. 1979.

2. A. Chottera and M. Shridhar, "Feature extraction of manufactured parts in the presence of spurious surface reflections", Canadian Electrical Engineering Journal, Vol. 7, No. 4, pp. 29-33, Oct. 1982.

3. H. Freeman, "On the Encoding of Arbitrary Geometric Configurations.", IEEE Transactions on Electronic Computers, Vol. 10, pp. 260-268, June 1961.

4. G.H. Granlund, "Fourier Preprocessing for Handprinted Character Recognition", IEEE Transactions on Computers, Vol. 21, No. 2, pp. 195-205, Feb. 1972.

5. A. Güdesen, "Quantitative Analysis of Preprocessing Techniques for the Recognition of Handprinted Characters", Pattern Recognition, Vol. 8, No.4, pp. 219-227, June 1976.

6. L. Lam and Ching Y. Suen, "Structural Classification and Relaxation matching of Totally Unconstrained Handwritten Zip-code numbers", Pattern Recognition, Vol. 21, No. 1, pp. 19-31, January 1988.

7. W.W. Loy and I.D. Landau, "An On-line Procedure for Recognition of Handprinted Alphanumeric characters", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 4, No. 4, pp. 422-427, July 1982.

8. J. Mantas and A.G.Heaton, "Handwritten Character Recognition by Parallel Labelling and Shape Analysis", Pattern Recognition Letters I, North Holland, pp. 465-468, July 1983.

9. S.J. Mason and J.K. Clemens, "Character recognition in an experimental reading machine for the blind.", Recognizing Patterns, P. A. Kolers and M. Eden, Eds. Cambridge, Mass.: M.I.T. Press, 1968, pp. 156-167.

10. T. Pavlidis and F. Ali, "Computer Recognition of Handwritten Numerals by Polygonal Approximation", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 5, No. 11, pp. 61s-614, Nov. 1975.

11. E. Persoon and K. Sun Fu, "Shape Discrimination Using Fourier Decriptors", IEEE and Machine Intelligence, Vol. 8, No. 3, pp. 388-394, May 1986.

12. G.P.R. Sarvarayudu and K. Sethi, "Walsh Descriptors for Polygonal Curves", Pattern Recognition, Vol. 16, No. 3, pp. 327-336, 1983.

13. P. Shen-Pei Wang, "A New Character Recognition Scheme with Lower Ambiguity and Higher Recognizability", Proceedings on the Sixth International Conference on Pattern Recognition, Munich, pp. 37-39, October 19-22, 1982.

14. M. Shridhar, and A. Badreldin, "High Accuracy Character Recognition Algorithm using Fourier and Topological Descriptors", Pattern Recognition, Vol. 17, No. 5, pp. 515-524, 1984.

15. G.T. Toussaint and R. Donaldson, "Algorithms for Recognizing Contour-Traced Handprinted Characters", IEEE Transactions on Computers, Vol. 19, No. 6, pp. 541-546, June 1970.