BENGALI HANDWRITTEN CHARACTER RECOGNITION USING MODIFIED SYNTACTIC METHOD

Mohammad Badiul Islam, Mollah Masum Billah Azadi, Md. Abdur Rahman and M. M. A. Hashem***

Southtech Limited, Dhaka, Bangladesh
**Grassroots IT Ltd, Dhaka, Bangladesh*
***Department of Computer Science and Engineering*
Khulna University of Engineering & Technology, Khulna, Bangladesh

Abstract:
Several approaches have been proposed to recognize hand written Bengali characters using different curve fitting algorithms and curvature analysis. In this paper, a new algorithm to identify various strokes of a handwritten character is developed. This curve-fitting algorithm helps recognizing various strokes of different patterns (line, quadratic curve) precisely. This reduces the error elimination burden heavily. Implementation of this Modified Syntactic Method demonstrates significant improvement in the recognition of Bengali hand written characters.

Keywords: *Syntactic method, curve fitting, thinning algorithm, curvature, and regression analysis*

1.0 INTRODUCTION

With the rapid growth and advancement of the use of computers in Bangladesh, the use of our mother tongue, Bengali in computers is being much talked about and much research is being done. As we find in many cases, the problem of input of Bengali characters to computers is time consuming and error prone. One answer to the problem would be the development of a practical Bengali character recognition method towards which the efforts of this project is directed. The objective of this research is the development of a method for the recognition of hand written Bengali characters using the syntactic method.

After several decades of research activities, hand written character recognition still retains its appeal as an exciting and growing field of scientific investigation. Handwritten Character Recognition (HCR) [20] is the reading of a scanned image of a hand written character along with an associated lexicon. The domain of hand written character verification varies form applications in reading handwritten addresses in post mail, reading Bank checks, or extracting some data from diverse forms. Depending on the chosen language different techniques can be applied for different types of characters. The Syntactic method [5,13,14] is a well developed approach to recognize the characters of different Latin-based languages while suffers from the drawback of recognizing Bengali hand written characters properly. The largest difficulty of handwritten Bengali character recognition is the immense variation of the way handwriting. Many factors can account for the diversity in Bengali handwriting styles, among others the regional origin of the writers; their educational level and their profession In fact different circumstances such as stress; fatigue; and hurry affect the handwriting of any individual. In this paper, a new algorithm to identify various strokes of handwritten characters is proposed. It is a curve-fitting based algorithm, which helps to recognize various strokes of different patterns (line, quadratic curve) with high accuracy. Figure.1 depicts our proposed Modified Syntactic Method (MSM) system, which consists of four components. In the first component the strokes are generated. We differentiate between curve and line stroke, as we will discuss later. In the reference characters database components, character elements are entered at a time in scanned style, pre-processed, and then stored in a reference character database. In then curvature analysis to differentiate curves and lines and string generation component, a numeric string type code is generated which uniquely identifies the character from the structural representation of a character. In the last and final component, we get the result depending on the previous 'code' (Reference Component) the machine takes decision to the class belonging of the character. A dictionary of code is searched to find a matching code corresponding to the code generated from the input character.

2.0 SYNTACTIC METHOD

The syntactic approach to pattern recognition provides a capability for describing a large set of complex patterns by using small sets of simple pattern primitives [5,14,34] and of grammatical rules. Of course, the practical utility of such an approach depends on our ability to recognize the simple pattern primitives [5,14,34] and their relationships, represented by the composition operations. Atypical pattern primitives [5,14,34] and the corresponding structure of digit 9 are shown in Figure 2.

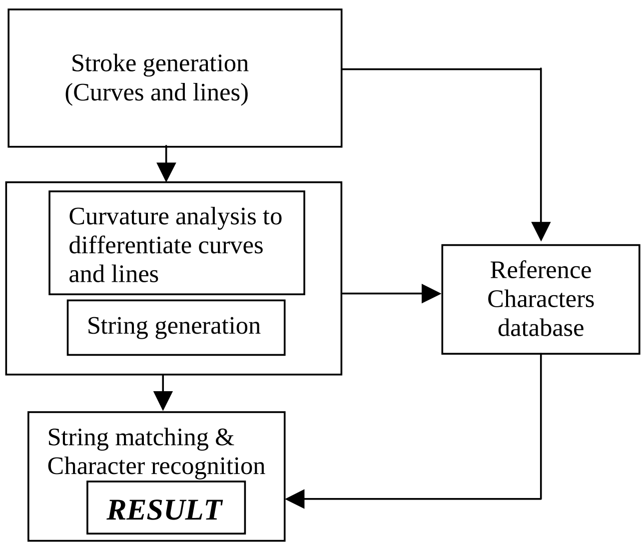


Figure 1: System organization

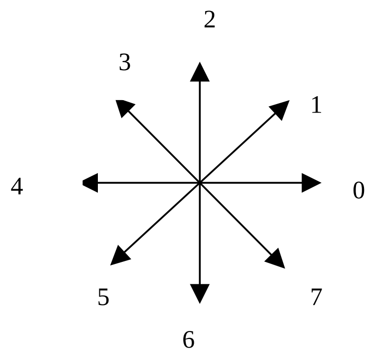


Figure 3: 8 Directional chain code.

Figure 4: Scanned ka(K)

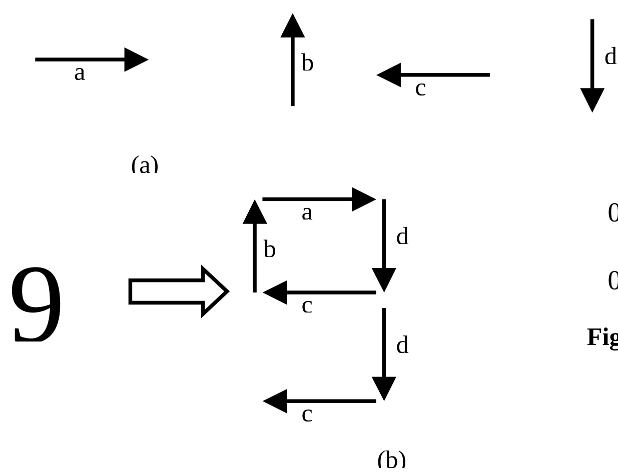


Figure 2: (a) pattern primitives (b) pattern 9 with its *primitives.*

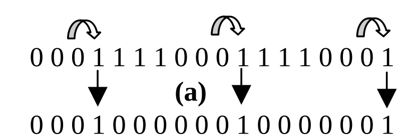


Figure 6: (a) Before contour tracing,
(b) After contour tracing.

Recognition of a primitive can also be made on the basis of a logical decision from the comparisons of feature measurements taken from the input component with a set of pre-specified thresholds. For example straight lines can be classified into different classes by computing their lengths or slopes or directions. The eight directional chain code for straight lines is shown in Figure 3.

3.0 RECOGNITION PROCESS

3.1 Inputs and Pre-processing

The scanned file of a handwritten character (the character is scanned using a regular scanner) is either enhanced/compressed or restored to form a 48x48 pixel size image. (It is not mandatory to produced 48x48 dimension pixels. We use this dimension because produced reference database is in 48x48 pixel. We can also use any dimension for operational flexibility.) This process proceeds by scanning alternately from left to right and top to bottom along various rows and columns of the input array. This continues until a black byte is encountered. A byte is taken as black if it has more than one black points. In Figure 4 scanned image of the letter ***Ka(K)*** is shown. To recognize a given character, at first it is to be digitized into a matrix i.e. to be transformed into a binary form for ease of handling by the computer. The digitizer is a device, which converts a physical sample to be recognized into a pattern vector. It is often convenient for recognition purpose to arrange the pattern in the form of a pattern vector:

$$X=[X1 \quad X2 \quad X3 \ldots\ldots Xn].$$

Where n is the number of measurements component Xi of the vector X assumes the value 1 or 0 depending on the state of the ith position for a particular input.

The program operated on a representation of one character at a time. The representation is in the form of a matrix, whose entries have values 0(Zero) or 1(One) corresponding to white and black in the original picture. The digitization process and digitized output of ka are shown in Figure 5(a) and Figure 5(b) respectively.

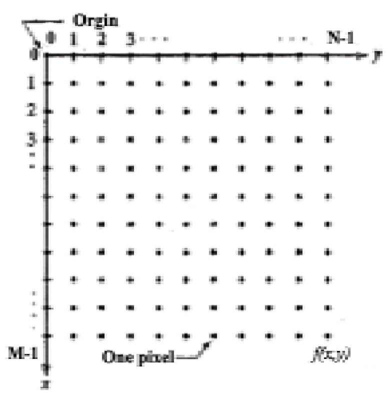


Figure 5(a): Digitization Process

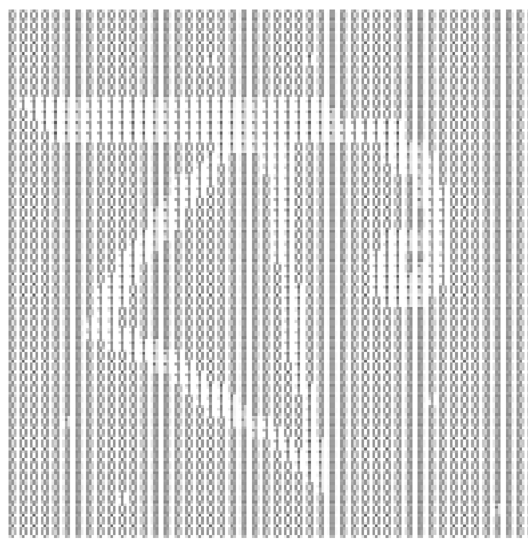


Figure 5(b): After digitized Ka (K).

3.2 Contour Tracing and Filtering

Contour tracing is the process of finding a series of black points on the boundary of a black region in a white field. The white to black transition points of the letter are obtained by scanning each time of the digitalized character by device from left to right and from top to bottom and storing them in a memory location. In Figure 6(a*)* and Figure 6(b*)* the zero (0) to one (1) transition and contour traced output are shown. In Figure 7 contour traced *Ka (K).* Next task towards pattern recognition is to filter out all isolated 1's (stray points) present in the contour map of the character to be recognized. A '1' is considered as to be isolated if there is no '1' at any one of its fourteen neighbor points as shown in Figure. 8. Filtering out is performed by 'And'-ing each byte with the mask of its top and bottom bytes and

‘Or’-ing these two results. The result is stored in the byte location for the first and last line only the bottom and top byte is taken respectively. After filtering out all stray points the contour of figure becomes as shown in Figure 9.

3.3 Extraction of Strokes (Morphs)

To extract the strokes [15] all the x and y co-ordinates of the pixels from the filtered file is saved. Scanning starts from top left corner and proceeds bottom for finding continuous points. Points are assumed continuous when three bits left or three bits right of its immediate upper line black. If a discontinuous occurs, scanning proceeds to the next line with the assumption that the void in the current line is due to improper scanning. If the next line gives continuity, a black point is assumed in the previous line and its coordinates are saved. Otherwise, the stroke is assumed to have terminated. All black points (1) of the stroke [15] are terminated out to ‘0’ during the time the coordinates are storing so that these are not considered again when scanning starts from the upper left corner for the next stroke. The process of scanning strokes continues until no new strokes can be found. The Figure. 10 shows the extraction of strokes from the filtered Ka (*K*). The result is to produce the collection of stroke coordinates.

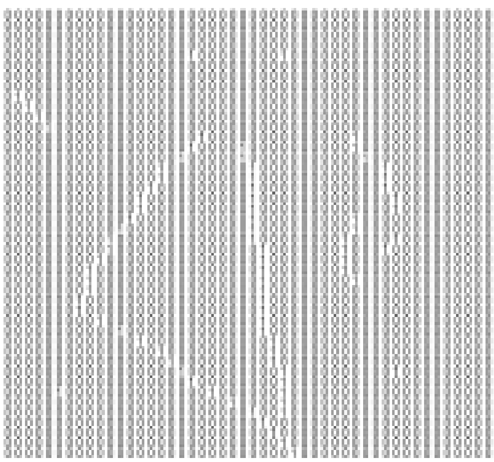


Figure 7: Contour traced Ka *(K)*.

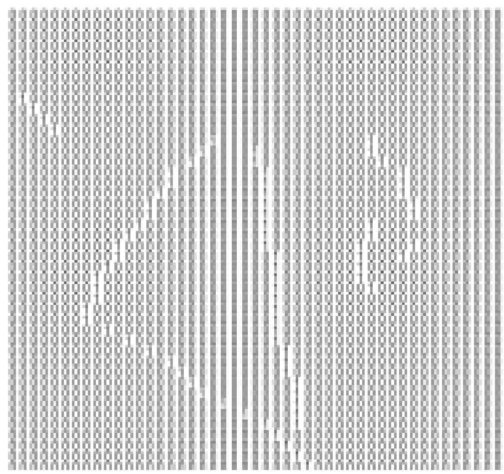


Figure 9: After filtering the Ka (K *)*.

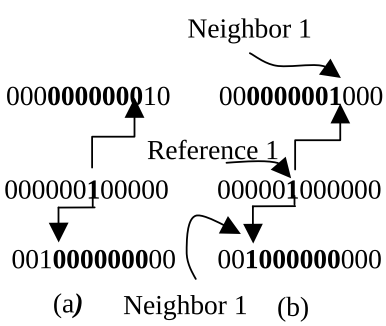
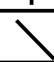


Figure 8: (a) An isolated ‘1’
(b) A ‘1’ having neighbor points

Table 1: Strokes Used for Bangla Character Recognition.

Stroke	Characteristics	Numeric Code
	Horizontal line	1
	Positive sloping line	2
	Vertical Line	3
	Negative sloping line	4
	Vertical concave curve	5
	Vertical convex curve	6
	Rejected stroke	0

3.4 Encoding of Strokes

For the purpose of recognition, it is desired to generate a numeric code for each character. The first step in this procedure is the selection of strokes and the generation of code for each stroke in a character.

3.4.1 Selection of Strokes

In syntactic method each input pattern is to be resolved into primitive structural elements called strokes [15]. The first step in designing a syntactic model is the selection of morphs in terms of which the patterns of interest can be represented. In selecting strokes care must be taken so that they are simple to recognize and they are minimum in number. For our analysis six simple strokes [15] are chosen to represent all 50 Bengali characters, which are shown in Table 1. A wide range of deviation is allowed to each of six strokes is shown dramatically in Figure.11.

3.4.2 Finding Stroke Code

The coordinates for a particular stroke are now analyzed to find the code of that stroke. The stroke under consideration is either a curve or a straight line with the theme that any alphabet in any language can be represented by combination of curves or straight lines. The challenge in this process lies in what degree we are breaking an alphabet. Later the co-ordinates of continuous strokes are fitted for finding numeric codes.

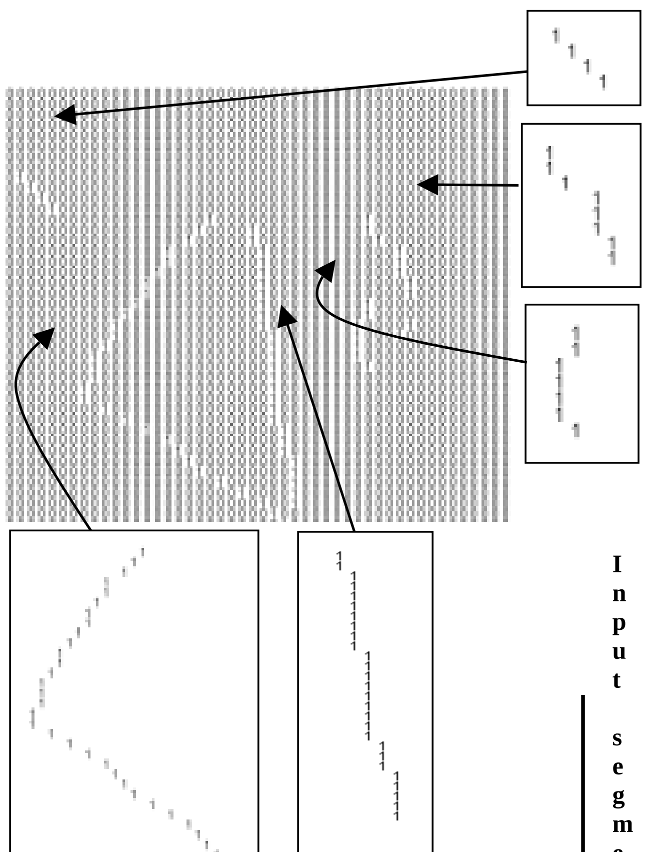


Figure 10: Filtered Strokes

Reference segments → / ↓ Input segments	$Rseg_1$	…… …….	$Rseg_k$
$Iseg_1$	$S_{1,1}$		$S_{1,k}$
$Iseg_2$	$S_{2,1}$		$S_{2,k}$
. . .	. . .	…… …….	. . .
$Iseg_n$	$S_{n,1}$		$S_{n,k}$

Figure 12: Matching Matrix

Table 2: Relative matching scores for specified primitive codes

Reference Segments →

Input segments ↓

	1	2	3	4	5	6
1	100	50	5	50	5	5
2	50	100	50	0	70	0
3	5	50	100	50	10	10
4	50	0	50	100	0	70
5	5	70	10	0	100	0
6	5	0	10	70	0	100

The equation of curve fitting [7] is- $x = a+by+cy^2$.
Where a, b, c are as follows-

$$a = \frac{\begin{vmatrix} \sum x & \sum y & \sum y^2 \\ \sum xy & \sum y^2 & \sum y^3 \\ \sum xy^2 & \sum y^3 & \sum y^4 \end{vmatrix}}{\begin{vmatrix} N & \sum y & \sum y^2 \\ \sum y & \sum y^2 & \sum y^3 \\ \sum y^2 & \sum y^3 & \sum y^4 \end{vmatrix}} \quad b = \frac{\begin{vmatrix} N & \sum x & \sum y^2 \\ \sum y & \sum xy & \sum y^3 \\ \sum y^2 & \sum xy^2 & \sum y^4 \end{vmatrix}}{\begin{vmatrix} N & \sum y & \sum y^2 \\ \sum y & \sum y^2 & \sum y^3 \\ \sum y^2 & \sum y^3 & \sum y^4 \end{vmatrix}} \quad c = \frac{\begin{vmatrix} N & \sum y & \sum x \\ \sum y & \sum y^2 & \sum xy \\ \sum y^2 & \sum y^3 & \sum xy^2 \end{vmatrix}}{\begin{vmatrix} N & \sum y & \sum y^2 \\ \sum y & \sum y^2 & \sum y^3 \\ \sum y^2 & \sum y^3 & \sum y^4 \end{vmatrix}}$$

Where N= Total number of points on the stroke.

The point on the stroke which has maximum curvature (κ) [3] is calculated by using the formula.

$\kappa = x_2/(1+x_1^2)^{3/2}$ Here $x_1 = dx/dy$ & $x_2 = d^2x/dy^2$

After extensive investigation and curve analysis [3] (We get several samples for this purpose. We get different κ for the samples. Thus we found for above 95% samples κ distinguished the curve and straight lines.) We have found that if $\kappa >= 0.9$ then the stroke is assumed to be a curve while $\kappa < 0.9$ will assume the stroke a straight line. This new approach in differentiating the curve and straight line will result in significant amount of error minimization.
Again code 5 or 6 is given according to the value of constant c whether positive or negative. If we looked at the Figure 10 we can get the logical explanation for the code 5 and 6. From Figure 10 we find it that we get three curves. Two of them are Vertical concave curve (code 5) because their slope is positive. We get one negative slope curve. For that purpose its code is 6 (Vertical convex curve)

The slope [7] of the line is calculated from the equation – $x = a+by$
where a, b are as follows-

$$a = \frac{\begin{vmatrix} \sum x & \sum y \\ \sum xy & \sum y^2 \end{vmatrix}}{\begin{vmatrix} N & \sum y \\ \sum y & \sum y^2 \end{vmatrix}} \quad b = \frac{\begin{vmatrix} N & \sum x \\ \sum y & \sum xy \end{vmatrix}}{\begin{vmatrix} N & \sum y \\ \sum y & \sum y^2 \end{vmatrix}}$$

If the length of strokes (N) is less than or equal to 4, the stroke code is assumed to be 0. It is assumed that a stroke with code ‘0’ is created due to noise present in the surrounding of the original letter or for some other reasons.

Where slope=1/b.

3.4.3 Encoding of Characters

The representation of characters is in the form of numeric string. The numerals in the code not only indicate strokes [15], which build the characters but also show the relationship among the strokes. The strokes in the character are visited from left to right and from top to bottom along various rows and columns. All strokes [15] are traversed exactly once. The letter code consists of the stroke codes written down in the order in which they encountered. It is possible to get an idea about the structure of the character from its code.

3.5 Recognition of Characters

3.5.1 Matching Scores

The characters represented by numeric string codes are now to be recognized. A dictionary of codes called reference strings is searched to find a matching code corresponding to the code generated from the input character. The reference segment [10] and the matching scores [10,30] for the corresponding input segments are shown in Figure. 11.

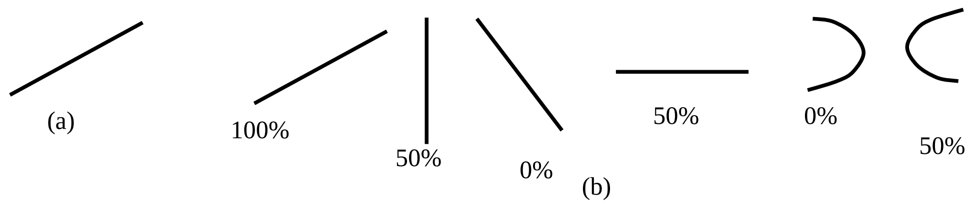


Figure 11: (a) Reference segment (b) Possible input segments and their matching scores

3.5.2 Matching Matrix

The reference pattern is thought as a sequence of segments. If the reference string [10] R is represented by, R={$Rseg_{1,}$ $Rseg_{2,}$ …………….., $Rseg_k$ } and the input string is also a list of segments represented by, I={$Iseg_{1,}$ $Iseg_{2,}$ ……….., $Iseg_n$ }. We have included a matching matrix [10,11] to assess the minimum score the input deserves. The matrix has rows reference by the input segments and columns reference by the reference segments. The intersection of i-th input segment and j-th reference segment holds the matching score between them. The matching matrix [10,11] is shown in Figure. 12**.** Where Si,j denotes the matching score. Relative matching scores for specified primitive codes in Table 2.

3.5.3 Optimal Score Computation

After we have finished building the matching matrix, our next aim is to find optimal path [10,32] through the matrix having the score maximized. This is given below-

If we are currently in $M_{(i,j)}$, we compute-

$Sl = M_{(i,j)} + \max((M_{(i+1,j)} + M_{(i+2,j)}), (M_{(i+1,j)} + M_{(i+2,j+1)}))$

$Sr = M_{(i,j)} + \max((M_{(i,j+1)} + M_{(i,j+2)}), (M_{(i,j+1)} + M_{(i+1,j+2)}))$

$Sd = M_{(i,j)} + (M_{(i+1,j+1)} + \max(M_{(i+1,j+2)}, (M_{(i+2,j+2),} M_{(i+2,j+1)})$ Where, Sl denotes shift down, Sr denotes Shift right, Sd denotes Shift diagonally.

Then according to the maximum Value achieved, we move down, right or diagonally down into the matrix. The Algorithm 1 is used for optimal score computation.

Algorithm 1:

For (i = 0 to n -1) do
    For (j = 0 to k-1) do
        Compute Sl, Sr, Sd for each $M_{(i,j)}$
    End
  Compute max (Sl, Sr, Sd) and move to the direction of maximum value.
  End

	5	3	6	5
5	100	10	0	100
3	10	100	10	10
3	10	100	10	10
6	0	10	100	0

(a)
Average matching score with Ka(K): 100%

	4	5	3
5	0	100	10
3	50	10	100
3	50	10	100
6	70	0	10

(b)
Average matching score with Ba(e): 62%

	4	5	3	5	6
5	0	100	10	100	0
3	50	10	100	10	10
3	50	10	100	10	10
6	70	0	10	0	100

(c)
Average matching score with Ra(i) 62%

Figure 13: Optimal matching score computation

3.5.4 Taking Decision

After completing the optimal score computation [10,11], we compute the average matching score of the input string with each reference string. The average matching score ($S_{av)}$ above 90% is taken for taking decision of recognition. Otherwise, it is decided that the input character is more erroneous and it can not be recognized by this project. The reference string (character) that gives the maximum S_{av} with the input string is considered as the recognized character. The Algorithm 2 is used for taking decision.

Algorithm 2: 1. Compute average matching score, S_{av}
2. If S_{av}>=90% then
Choose which score is high and print the recognized character.
Else Print cannot be recognized.

The average matching score computation of input string ***5336*** with reference strings ***5365 (*** **Ka)** **(K)**, *453* **(Ba) (e)** and *45356* ***(*** **Ra** ***)*** **(i)** is shown in ***Figure. 13***, where optimal average score *100%(>90%)* matches with reference string of character ***(*****Ka) (K)**. So the input character is *(*Ka) **(K).**

4.0 RESULTS

A test is carried out on different Bengali characters from various persons (adult) using the programming language 'JAVA' [8,6] on Pentium-IV machine. The writers were told to write each character naturally as they were writing on a notebook. The Table 3 shows our test result. Out of 400 samples (Individually 50 samples for Ka (K), Ba(e), Ra(i), Bha(f), Jha(S), Ri(F), Da(`), Dha(a) in total 400 samples) , the program recognizes 380 times correctly, 13 times incorrectly. In 7 cases it fails to give a decision i.e. it rejects the input pattern. So from Table 3, it is observed that the test gives 95% (on an average for eight characters.) accuracy. It has been found that some ambiguity may occur if a stroke in a character becomes so small as to be comparable to the width and height of the character. Problems may occur only when some of the strokes in a character are so small as to be the same order of magnitude as what is considered noise for others. The failures are due to a particular component not being analyzed, improper scanning, stroke segments appearing at different angles and strokes touching in one case but not the other. Most of the problem has occurred in the store stokes routine. The store strokes routine would sometimes make mistakes if, for example, two strokes were very close together or one stroke was divided into two parts by more than three lines. Another source of failure may be due to unavoidable noise [34] present in the character. Recognition Rates of different Characters is shown in Table 3.

5.0 DISCUSSION

5.1 Decision-theoretic and Syntactic Method

Decision-theoretic method [26] is very simple and it considers the very general and simple characteristics for pattern recognition. Syntactic method [13,14] chooses the basic and elementary (Primitives) characteristics of the pattern. So it can be used for the recognition of complex patterns (For example characters, gesture, fingerprint etc.). It's performance is higher than decision-theoretic method. For these reasons we use the method for our work.

Table 3: Recognition Rates of different Characters

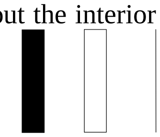
Character	No. of Samples taken (Handwritten)	No. of Samples Recognized	No. of samples wrongly recognized	No. of samples rejected	% Accuracy
Ka(K)	50	49	1	0	98
Ba(e)	50	49	1	0	98
Ra(i)	50	48	1	1	96
Bha(f)	50	48	1	1	96
Jha(S)	50	45	3	2	90
Ri(F)	50	47	2	1	94
Da(`)	50	48	1	1	96
Dha(a)	50	46	3	1	92

5.2 Primitive Generation

There are two techniques for primitive generation for character recognition in syntactic method-
(a) Boundary Tracing [4,19] and (b) Thinning method [4,19].

In boundary tracing a pixel is said to be boundary pixel if one of it's neighbor pixels is white and others are black. Tracing the boundary pixels with eliminating out the interior pixels generates the primitives. The drawbacks of this method are-

(1) Two boundaries (Outer and Inner) are obtained.
(2) Protrusions [27]can not be eliminated.
(3) There is no breaking for primitive generation.



(a) (b) (c)

Figure 14: A line (a) before boundary tracing (b) after boundary traced (c) after thinned

In the case of thinning method, the leftmost pixel is chosen as boundary pixel [27]in the continuous black pixels in the same row.

So it gives single boundary and breakings are generated for recognizing primitives easily. For these reasons, we choose this technique for our work.

5.3 Primitive Selection

There are two techniques for primitive selection for character recognition in syntactic method-
(a) Freeman's chain code generation [4,5] and (b) Slope consideration [1,20]

In the case of Freeman's chain generation primitives (lines) are selected by considering the direction and length. The drawbacks of this technique are-

1) Number of primitives are more, 2) String length is large and 3) Fixed length line is to be considered.

In the case of slope consideration [1,20] the primitives (lines) are selected by considering the slopes of lines. So the number of primitives as well as the string length becomes smaller. For this reason we choose slope consideration technique for our work.

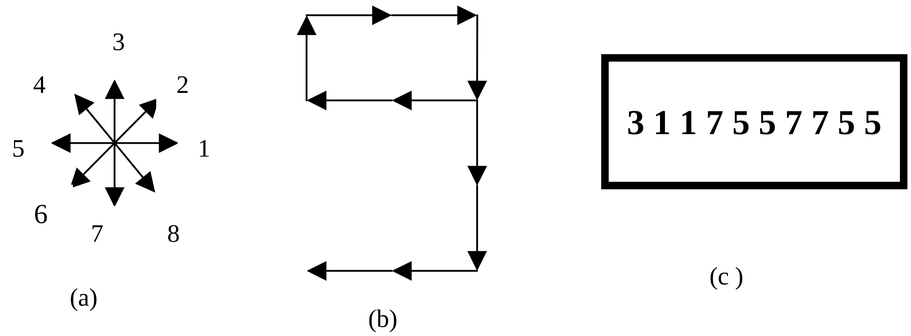


Figure 15: (a) 8 directed Freeman's chain code (b) Chain code generated digit 9 (c) Chain code generated string for digit 9.

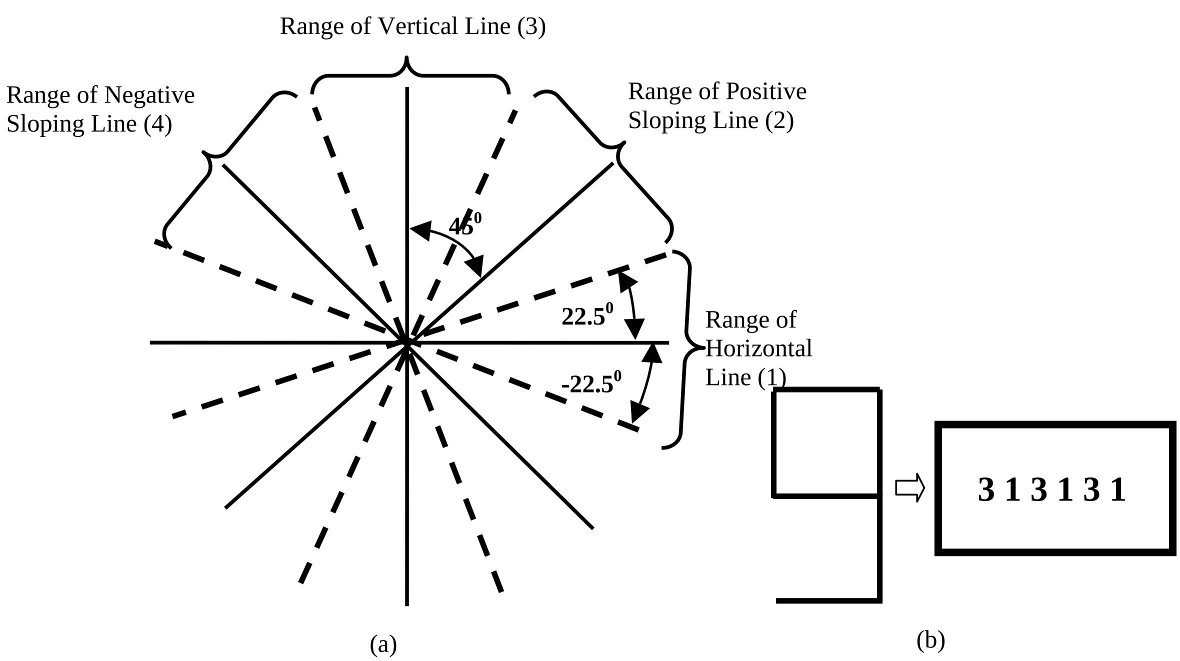


Figure 16: (a) primitives on considering slopes (b) String for digit 9.

5.4 Conclusion

For the present analysis, the basic idea of syntactic method to represent each complex pattern in terms of simple sub patterns is utilized. But the techniques developed for resolving each character into a number of stokes [18,21], their recognition lead to a new approach. However, the idea of representation of characters by numeric codes for their recognition was proposed and successfully used before in recognizing foreign letters and now also in Bengali characters.

The principal difficulty with this approach is the lack of use of a nearness criterion. If a character sample produces a code different from all known codes, it is rejected. As a partial solution to this problem, some of the characters have been assigned several codes, corresponding to common discrepancies between character samples due to a lack of uniformity in quality of handwriting. Improvements in resolution would also be expected to help.

The ideal goal of designing a hand writing recognition method with 100% accuracy is illusionary, because even human beings are not able to recognize every handwritten text without any doubt, e.g. it happens to most people that they sometimes cannot even read their own notes. Humans roughly recognize 96%[23].

By considering the curve fitting with curvature [3], the performance can be surprisingly improved. This technique is not applicable to the recognition of the handwritten characters of children or old but for the adults. Our method proved to be useful, as it was successful in recognition of about 95% of the characters.

By using this technique, Bengali numeric characters can also be recognized. With some modifications one can also recognize the compound words, which includes both characters and numerals [30]. Examples of which are postal code recognition, number plate of car recognition etc. A complete Bengali document can be recognized with better improvement of this technique. Several possibilities exist which could improve the chances of constructing a practical text recognition device [5,24].a) High standard of scanning quality. b) Stylized character [23,24,25] size. and c) Language simplification. The problem would be the development of a practical Bengali character recognition machine, toward which the effort of this project is directed. It is hope that advances in this area would provide additional incentive for work in translation devices.

5.5 Recommendations for Future Work

We have developed our system for single Bengali character recognition. Every character cannot be recognized with this technique. But with some modification this can be done. We have also the following recommendation for future work:

1. The percentage of recognition can be improved by dynamically assigning edit distances, the weights of insertion, deletion and substitution values by using the training process.
2. We did not use the positional information of the string numbers. Use of this information could prove to be beneficial.
3. This method can also be used to recognize Bengali numerals.
4. This method can also be used for the recognition of postal code, car plate number where compound structure of alphabet and numerals present.
5. Regression analysis [9] with curvature analysis can improve the performance of recognition process.
6. For an alphabet with large number of characters this method may not be effective as it was for Bengali alphabet. In those cases this method could be used as a pre-classifier to aid the actual recognition process.
7. Signatures can be recognized by improving this technique also.

6.0 REFERENCES

[1] Mia, M. A. K, “*Recognition of Printed Bangla Characters by Syntactic Method of Pattern Recognition*”, M. Sc. Engg. Thesis, Department of Computer Science of Engineering, BUET, 1992.

[2] Marchand-Maillet, S., and Sharaiha, Y.M., *Binary Digital Image Processing: A Discrete Approach*, Academic Press, New York, 2000.

[3] Eusuf, Md. Abu, “Differential Calculas”, Part-1 & Part-2.

[4] Jain, Ramesh. Kasturi and Rangachar. Schunck and Brain, G., *Machine Vision*, McGraw-Hill, Inc.

[5] Gonzalez, Rafael C. and woods, Richard E., *Digital Image Processing*, Pearson Education Asia.

[6] Memon, N., Neuhoff, D.L. and Shende, S, "An Analysis of Some Common Scanning Techniques for Lossless Image Coding", *IEEE Trans. Image Processing,* vol. 9, no. 11, pp. 1837-1848, 2000.

[7] Chapra, Steven C. and Canale, Raymond P., PH.D., *Numerical Methods For engineers,* McGraw- Hill Book Company.

[8] Edleman, S., *Representation and Recognition in Vision*, The MIT Press, Cambridge, Mass, 1999.

[9] Hoque, M. T. and Ali, M. M, "Noise removal algorithm from images by using breadth first search strategy", in Nat. Conf. & Info. Syst., pp156-160, Dec. 1997.

[10] Rashid, Al Mamunur & Ali, Mohammad Masroor, "Online recognition of handwritten characters using 1-Dim,1-Dim DP approach", *Proceedings of ICCIT*,1998.

[11] Bellman, R. E. , "*Dynamic Programing"*, Preinceton University Press,1957.

[12] Thomason, M.G., and Gonzalez, R.C., "Syntactic Recognition of Imperfectly Specified Patterns." *IEEE Transaction Computer*, vol. C-24, no. 1, pp. 93-96, 1975

[13] Tanaka, E., "Theoretical Aspects of Syntactic Pattern Recognition", *Pattern Recog.*, vol. 28, no. 7 pp. 1053-1061, 1995.

[14] Jaehwa Park, "An Adaptive Approach to Offline Handwritten Word Recognition", *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 24, no. 7,p. 920, 2002.

[15] Lee. H.D., Kim. T.K., Agui. T. and Nakajima, M., "On-line Recognition of Cursive Hangul by Extended DP Matching Method", Journal of KITE, Korea, vol. 26, no. 1, pp. 103-111, 1989.

[16] R.E. Bellman and S.E. Dreyfus, "Applied Dynamic Programming,", Princeton University Press, Princeton, New Jersy, 1962.

[17] Fairhurst, M. C. and Rahman, A. F, "A generalized approach to the recognition of structurally similar handwritten characters", *IEEE Proc. On Vision, Image and Signal Processing,* vol. 144, no. 1, pp, 15-22, 1997.

[18] A. K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall,1989.

[19] Sattar, M.. A. and Rahman, S. M, "An experimental investigation on Bangla character recognition system", Bangladesh Computer Society Journal, Vol. 4, PP 1-4, Dec 1989.

[20] Sattar, M. A, "Recognition of printed Bangla characters by Decision theoretic method pattern recognition ", Masters Thesis, BUET Dhaka-1000, 1990.

[21] Ali, Md. Masroor and Hossain, Md. Bellayet, " Recognition of handwritten Bangla numeral by boundary line matching", Journal of the CSI, Vol. 32 No. 2, June 2002.

[22] Rashid, Al Mamunur and Ali, Md. Masroor, "On Line Recognition of Handwritten Characters Using 1-Dim, 1-Dim DP Approach", Journal on the ICCIT, Dhaka, Dec-1998.

[23] Rahman, A.F.R. and Satter, M.A., "An Analysis of Bengali Characters: Detection of Some Characteristic Features", Journal on ICCIT, Dhaka, Dec 1998.

[24] Moshad, Md. Ali Asger and Ali, Md. Masroor, "Recognition of Handwritten Bangla Digits by Intelligent Regional Search Method", Journal on ICCIT,Dhaka 2001.

[25] Golam Rasul, "Bengali Character Recognition Using Genetic Algorithm", M. Sc. Engg. Thesis, Department of Computer Science of Engineering ,BUET,1998.

[26] Ray ,K. and Chatterjee, B., "Design of a nearest Neighbor Classifier System for Bangla Character Recognition System", Institute of Electrical Telecommunication Engineering, Vol., PP. 226- 229, 1984.

[27] Bishnu, A. and Chaudhuri, B. B., "Segmentation of Bangla Handwritten Text into Characters by recursive contour following", *In Proceedings of the Fifth International Conference on Document Analysis and Recognition,* ICDAR '99', pp. 402-405, Dec.1999.