A Novel Approach to Skew Detection and Character Segmentation for Handwritten Bangla Words

A.Roy[*], T.K.Bhowmik[†], S.K.Parui[#] and U.Roy[*]

[*] *Dept. of Computer and System Sciences, Visva-Bharati, Santiniketan 731235, INDIA.*
ananda2111@rediffmail.com, uroyin@yahoo.co.in
[†] *IBM Global Services Pvt Ltd, Embassy Golf Link, Bangalore - 560 071, INDIA.*
tbhowmik@in.ibm.com
[#] *Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, 203, B. T. Road, Kolkata-700 108, INDIA.*
swapan@isical.ac.in

Abstract

Character segmentation is a necessary preprocessing step for character recognition in many handwritten word recognition systems. The most difficult case in character segmentation is the cursive script. Fully cursive nature of Bangla handwriting, the natural skewness in words poses some challenges for automatic character segmentation. In this article a novel approach to skew detection, correction as well as character segmentation has been presented for handwritten Bangla words as a test case. Segmenting points are extracted on the basis of some patterns observed in the handwritten words. With these segmenting points a graphical path (hereafter referred to as a candidate path) has been constructed. The handwritten words contain some consistent and also inconsistent skewness. Our algorithm can cope with both types of skewness at a time. Further the method is so direct that with the help of a candidate path one can handle both skew correction and segmentation successfully. the algorithm has been tested on a database prepared for laboratory use. The method yields fairly good results for this database.

1. Introduction

We consider here the task of automatic segmentation of handwriting for Bangla, the second-most popular language and script in the Indian subcontinent and the fifth-most popular language in the world. There are 50 basic characters (11 vowels and 39 consonants) in Bangla apart from the numerals. Obviously the structure of handwritten document is more complicated than the text document. Moreover the task considered here is much more difficult than segmentation of handwritten Roman script, as the Bangla scripts are very much cursive in nature. A number of handwritten samples covering a reasonably large spectrum of handwriting style have been collected form people of various levels of age, literacy and profession. A set of names of small, medium and large towns of West Bengal is the database concerned. The work has two main parts as—

1. Skew detection and correction.
2. Character segmentation.

Prior to segmentation skew estimation and correction is another important step in any document analysis and recognition system. A majority of the algorithms used for segmentation and character recognition depend upon upright images. Therefore skewed images severely degrade the performance of such systems. A wide variety of skew detection algorithms have been proposed in literature. A majority of the algorithms are based on Hough transform [1] and analysis of projection profiles [2]. Algorithms based on feature point distribution [3], run length analysis [4] have also been proposed in literature.

Character segmentation is one of the decision processes in a system for Optical Character Recognition (OCR) that seeks to decompose an image of a sequence of characters into sub-images of meaningful individual patterns/symbols. The decision of OCR that a pattern isolated from the image is a

Proceedings of the Digital Imaging Computing: Techniques and Applications (DICTA 2005)
0-7695-2467-2/05 $20.00 © 2005 IEEE

character or some other identifiable unit may be right or wrong. If it is wrong in considerable amount then the system may incur high error rate. That is why segmentation is an important part of any printed or handwritten OCR system. Character segmentation task will be more challenging as well as difficult when the script is handwritten and cursive. The letters in cursive writing are often connected, moreover the individual letters in a cursive word are often written so as to be unidentifiable as isolated characters. The variation in writing styles is also crucial. The basic segmentation algorithms can be classified into three main categories: region-based [5], contour-based [6] and recognition-based [7] algorithms. Although many methods for handwritten character segmentation have been published in the literature for different scripts [8,9], to the best of our knowledge only two reports are available on Bangla handwritten scripts [10,11].

In this study we initially do not consider skew estimation in isolation from segmentation. We gather information about skewness during extraction of segmenting points in the word. The actual skew detection and correction is done before segmentation and the segmenting points are further refined according to the corrected image obtained after skew correction.

2. Characteristics of Bangla handwriting

A detailed study of the Bangla words has shown that most of them have a long horizontal run called a '*Matra' (Headline)* and somewhat identifiable '*Baseline*'. Hence if we can detect the Matra and the baseline, we can immediately segregate the entire image into three zones, namely, *upper, middle* and *lower zones*. The Matra represents the boundary between upper and middle zones, and the baseline indicates boundary between middle and lower zones. The correct detection of Matra facilitates the segmentation process, because in Bangla words segmentation is done mostly along the Matra line. However the problems with handwritten words are that the writer may not include such a long Matra, so the Matra can only be approximated by a curve instead of a straight line. This situation is shown in Figure 1. In that case it is difficult to detect the Matra line. So we omit the detection of actual Matra, directly by examining long runs.

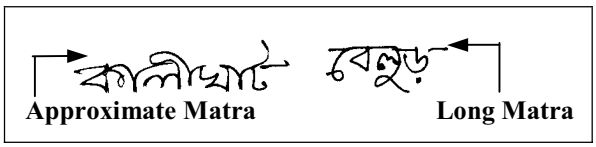


Figure 1. Sample Bangla words with Matra

Generally in Bangla handwritten words any two consecutive characters are connected at the upper portion of the word. Furthermore the connection between two consecutive characters is through an approximate straight-line segment (which may be very short). It is observed that in most of the cases, at the intersection region of a character and its connecting line with the consecutive character; one of the following patterns (shown in Figure 2) may arise along lower contour of the word.

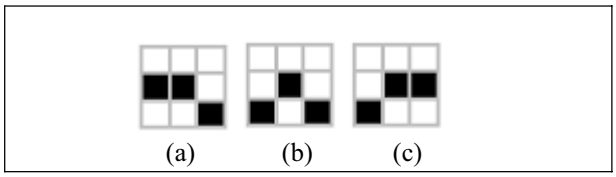


Figure 2. Pattern observed in Bangla words

These patterns are the main observations, by which we find the segmented regions as well as the segmenting points throughout the word image.

Concerning the skew of a word it is observed that in a word the characters may be skewed at different angles and directions independent of the skewness present in the adjacent characters. This observation introduces inconsistency in skewness, which is common to handwritten words. Moreover consistent skewness due to incorrect scanning process may also be present. This phenomenon is shown in Figure 3.

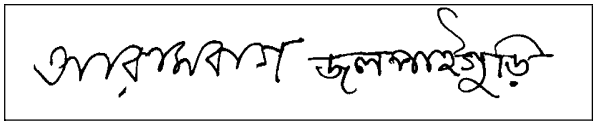


Figure 3. Examples of skewed Bangla words

3. Proposed segmentation technique

In the following sections, the algorithms described are tested on binary word images. The black pixels in the binary images represent the handwriting (foreground pixels) whilst the white pixels are used to denote the background.

3.1. Preprocessing

Prior to segmentation it is necessary to preprocess all word images. Initially the images are in gray-level format. The gray images are median filtered and then Otsu's thresholding algorithm [12] is used to binarize

the images. The binary images are then filtered to obtain smooth images.

3.2 Feature extraction

The segmenting points occur at the lower contour of the strokes in a word. So for feature extraction the lower contour of the word image is traced anticlockwise. Let (x_t, y_t) be the contour of the black pixels. The index t increases with the progress of tracing. The left and right touching points of the word image with its bounding box are taken as the starting and ending points of tracing respectively.

Considering all the observed patterns (Figure 2) during tracing, a complete tracing process along lower contour gives a set $S_L=\{(x_1, y_1), (x_2, y_2),\ldots,(x_L, y_L)\}$ containing all the points $(x_1,y_1),(x_2,y_2), \ldots, (x_L,y_L)$ at which any one of the above-described patterns are followed. Among every three successive points (along lower contour) $(x_{k-1}, y_{k-1}), (x_k, y_k), (x_{k+1}, y_{k+1})$ participate to form a pattern. We however consider the middle point (x_k, y_k) to be included in S_L. The elements of S_L are considered as the segmenting points of the word. It is clear that the set S_L may contain several points at which the image should not be segmented originally.

A close study of Bangla word images reveals that a character tracing originates from its Matra and diverges from its starting position as indicated in Figure 2(a). On the other hand at the end tracing it converges to the Matra, and thus ending points follow pattern identical to Figure 2(c). The pattern in Figure 2(b) is a special case, which occurs when the exact sharpness in tracing persists. However this situation is very rare as the word images are smoothed up, so most of the time the pattern in Figure 2(b) converges either to 2(a) or to 2(c). However both the patterns 2(b) at the beginning and at the ending points do not converge to same pattern because of the structure of the word. A complete character can thus be described by combining two pattern 2(a) or 2(b) and 2(c) or perhaps 2(a) and 2(c) or 2(b). Thus in the next step we consider all pairs $\{(x_i,y_i),(x_j,y_j)\}$ where j is next to i, (x_i,y_i) following pattern in Fig 2(a) or 2(b) and (x_j,y_j) following pattern in Fig 2(c) or 2(b). When considering these pairs we also take into account the positional information of each point. While dealing with Bangla handwritten words we have observed earlier that segmentation is mostly done along the approximate Matra line. So a point is rarely a segmenting point if it lies at the lower portion of a character in the word image. We consider the ratio of the distances from the segmenting point to the upper to lower boundaries of a character. As the actual height of a character is difficult to determine, we consider the width of the bounding box of the word as representative of height of each character.

Considering (x_i,y_i) as the segmenting point arises in a character, the ratio ρ_i is defined as—

$$\rho_i = \begin{cases} \dfrac{D_{upper}^{(i)}}{D_{lower}^{(i)}} & \text{if } D_{lower}^{(i)} \neq 0 \\ D_{upper}^{(i)} & \text{otherwise} \end{cases}$$

where,

$D_{upper}^{(i)}$ = distance of (x_i,y_i) with respect to upper portion of bounding box.

$D_{lower}^{(i)}$ = distance of (x_i,y_i) with respect to lower portion of bounding box.

The line through the point (x_i,y_i) and parallel to X-axis is taken as the boundary of the two portions of bounding box.

This situation is described in Figure 4 for a particular character.

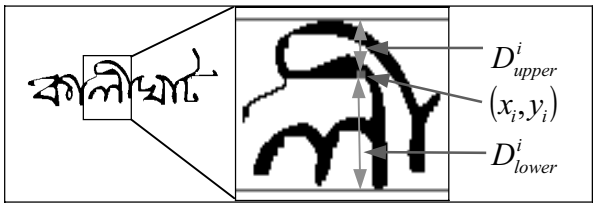


Figure 4. Distances to calculate ρ_i for (x_i,y_i)

Normally the ratio ρ_i has the property that $0 \le \rho_i << 1$ for all relevant segmenting points (x_i,y_i). However in some cases of skewed words the value of ρ_i for a relevant segmenting point (x_i,y_i) in a character may be close to or even a little greater than unity. Those characters itself, appears at the lower portion of the bounding box for the word. So we further extract another feature τ_i, for each point (x_i,y_i), which is the ratio of total number of black pixels in a rectangle defined for the upper part of bounding box, to the number of black pixels in a rectangle defined for the lower part of bounding box. Formally—

$$\tau_i = \begin{cases} \dfrac{M^{(i)}_{upper}}{M^{(i)}_{lower}} & if\ M^{(i)}_{lower} \neq 0 \\ M^{(i)}_{upper} & otherwise \end{cases}$$

where,

$M^{(i)}_{upper}$ = Number of pixels in $2h \times D^{(i)}_{upper}$ rectangle.

$M^{(i)}_{lower}$ = Number of pixels in $2h \times D^{(i)}_{lower}$ rectangle.

h = Threshold to define length of row of rectangle as - (position of x_i + h) - (position of x_i - h)

The Figure 5 represents the upper and lower rectangles as shaded portion.

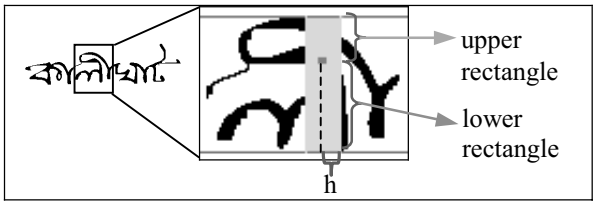


Figure 5. Rectangles to calculate τ_i for (x_i, y_i)

For a given segmenting point (x_i, y_i) that has ρ_i value close to or greater than unity we must have $\tau_i < \xi$ for a given threshold ξ.

Considering the above facts several point pairs are now extracted from the set S_L. These point pairs constitute a set of candidate points C = { p_1 , p_2 , … , p_n }

where

$$p_i = \{(x_i, y_i), (x_{i+1}, y_{i+1})\} \quad 1 \le i \le n, 1 \le n \le L$$

with a criteria for acceptance of a point (x_i, y_i), defined as—

$$(x_j, y_j) \begin{cases} \in C & if\ \rho_i < \varepsilon\ or\ (\rho_i > \varepsilon\ and\ \tau_i < \xi) \\ \notin C & otherwise \end{cases}$$

where ε is a predefined threshold in the range $[0,1)$. In practice we take $\varepsilon > 0.5$

However the set C may contain several nonsegmenting points. For an example a point (x_i, y_i) may appear above the Matra line but satisfying the criteria of a segmenting point ($\rho_i < \varepsilon$), though the point is nonsegmenting. The refinement of these unwanted segmenting points has been performed before actual segmentation. Such procedure has been described in section 3.6.

Taking the set C in hand we consider possible paths through the points p_i available in C. These paths go through the word horizontally and thus covering the characters. Relevant features for each path are computed. This process constitutes a feature vector for each path, defined as —

$$f = (\theta_{\max}, \theta_{avg}, N, J_{\max}, pos_{\max}, W)$$

where,

$\theta_{\max}$ is the maximum of angles between two successive points of a path.

θ_{avg} is the average of angles between two points of a path.

N is the number of points in a path.

$J_{\max}$ is the maximum of distances between two successive points along the width of the word.

$pos_{\max}$ is the highest of the positions of a point p_i involved in the path with respect to height.

W is the length of the path.

3.3. Optimal path extraction

From the possible paths the candidate path C_p is chosen by optimizing the set of feature vectors subject to the constrains—

$$\theta_{\max} \in [-10\quad, +10\quad]$$

$$\theta_{avg} < \varepsilon_1$$

$$N \to n * \varepsilon_2$$

$$J_{\max} \le \varepsilon_3$$

$$pos_{\max} < \varepsilon_4$$

and $W \to width\ of\ word$.

where $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ are predefined thresholds. Typically—

ε_1 has empirical value on the basis of the database.

$\varepsilon_2 \in (0,1]$ in practice we take $\varepsilon_2 \to 1^-$

and $\varepsilon_3 = width\ of\ corresponding\ character$

the value of ε_4 depends on ε and ξ previously defined. Formally $\varepsilon_4 = \Psi(\varepsilon, \xi)$, i.e. ε_4 is a function of (ε, ξ).

The factor $\theta_{\max}$ defines the maximum skewness present in a path. We however assume that the maximum skew angle should not exceed ± 10 .
This candidate path spread from the beginning to the end of the word image, because the length of the candidate path W is maximized to become as long as the width of the word. Further due to the fact that we make total number of points forming a path (N) close to the total points in the space (n), the candidate path takes care of all the points related to the characters (not necessarily all) of the handwritten words. To consider all possible characters in the word we refine this fact; in essence this helps to achieve more accurate skew of the handwritten word under consideration. We exploit the property of the feature $J_{\max}$ for the above refinement process. The higher value of $J_{\max}$ will incorporate the points wide enough along the considered word. In that case there is a high chance that the candidate path may omit some of characters in between. So in principle $J_{\max}$ should be a smaller as well as moderate value to consider all the characters of the handwritten word. Finally as the $pos_{\max}$ has considered minimized it is quite reasonable that candidate path should pass almost along the approximate Matra as discussed earlier. As most of the candidate points lie at the upper portion of the handwritten word, the probability of occurring the candidate path at the lower portion of the word is negligible. However for the case of peculiar as well as bad handwriting this case may arise which we do not consider. We notice the property that $C_p \subseteq C$, so we have extracted in candidate path only those points which we need for skew detection purpose in next phase.
Figure 6 below shows a word image with the candidate path. It can be easily verified that although the original image doesn't include a long Matra, the approximate Matra can be realized from the candidate path.

Figure 6. Word with candidate path

3.4. Skew estimation technique

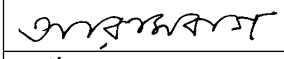
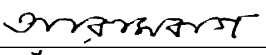
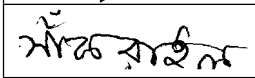
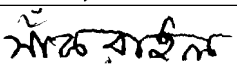
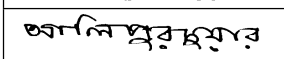
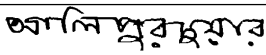
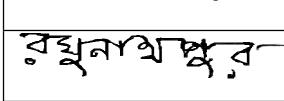
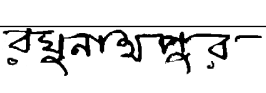
Document skew is a distortion that often occurs during document scanning or copying. This mainly concerns the orientation of text lines, and with no skew the lines are horizontal or vertical, depending on the script. This effect visually appears as a slope of the text lines with respect to the X-axis. Document skew is an unavoidable effect because of the complex structure of handwritten words and the copying/scanning process, especially when automatically digitizing huge document bulks. The conventional skew detection methods take care of detection of skew that arises during scanning process, which we refer to as consistent skew. It has been observed that in a single Bangla handwritten word the amount and direction of skew angle of individual letters normally varies, which we call the inconsistent skew. The conventional skew detection algorithms normally don't take care of inconsistent skew. Certainly, proper skew estimation and correction may be one of the most important steps for segmentation and recognition.
Normally in other methods a de-skewed image is generated from the original image and then supplied for tracing. However we at first trace the original image and then on the basis of tracing results the skew is detected and corrected. The candidate path can be used as a Matra when the corresponding points are so aligned that the path becomes horizontal with the X-axis. The skew detection algorithm utilizes the candidate path for a measurement of skewness. The modified candidate path is then used for segmentation purpose. Our skew detection algorithm relies on the fact that the angle between two successive points of the candidate path would give the amount of skew between the two points. Furthermore we claim that two successive points p_i and p_j correspond two successive characters, and thus the angle between them is the skew angle between the two characters. With a properly chosen candidate path, we can take into consideration of skewness between each pair of characters or components. In case of consistent skew the whole path itself becomes skewed with respect to X-axis. Thus our algorithm is capable of detecting consistent as well as inconsistent skew angle throughout the word.

3.5. Skew correction technique

In order to correct the skew between two successive characters the angle between them is calculated first. This is obviously the angle between the two successive points p_i and p_j corresponding the two characters. Let this angle be denoted as θ, then we can find $\tan\theta = l/b$. The base b gives the horizontal distance between two points and the perpendicular l

gives the amount of skewness. Thus in the next step the image component between the two segmenting points is rotated upwards or downwards properly to make the skew angle zero. This process is continued for all points in the set C_p. Proper *padding* ensures that we don't loose any vital object information when we rotate the image word in concerned.
Table 1 shows some of the examples of skew corrected image along with the original image placed in first coloum of the table.

Table 1. Sample handwritten words before and after skew correction

Original Words	Corrected Words
	
	
	
	

3.6. Segmentation

Now the de-skewed image is available for segmentation approach. The segmentation is performed along the so called candidate path (approximate Matra) of Bangla word as by virtue of the Bangla scripts the characters are connected at the upper portion that is along the Matra of a handwritten word.
To extract characters from a word we initially detect isolated and connected characters within a word. This process initially segments the word into connected components. Later each of the connected components are segmented into individual characters. In course of tracing for feature extraction we have collected the segmenting points in the word. During the skew detection process we consider only a subset C_p of all candidate points. This time however we need not to consider only the set C_p, instead we take into account the set of candidate points C. However all the points in the set C are not segmenting points always. We can merely take all the points along the approximate Matra to be our segmenting points. But observations show that there are some segmenting points not accurately along the Matra line, but close to the Matra. Thus in order to segment the word component as accurately as we can, we define a segmented region instead of a line. We assume that all the points in this region segment the word correctly. In the horizontal projection of a de-skewed word, we observe a sharp peak almost along the Matra line detected earlier. This peak can represent the Matra. Most of the segmentation process is done along the Matra, so a segmented region is obviously around the Matra line and hence the peak. We further observe two valleys on two sides of the peak in the horizontal projection. Figure 7 below shows a sample word with horizontal projection.

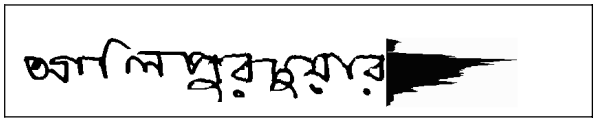


Figure 7. Word with horizontal projection

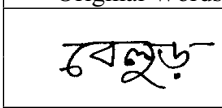
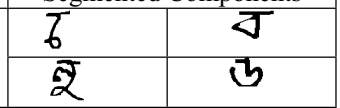
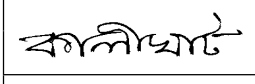
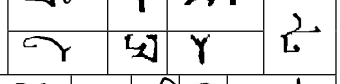
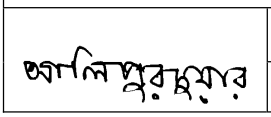
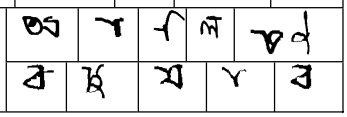
In the original word these two valleys corresponds to two boundaries just before and after the Matra line. So we can take these two valleys as boundaries of our segmented region. The segmenting points are not much further than Matra line. So this process yields correct segmented region. Now we extract those points, which belong to the segmented region. Based on those points now the connected components are further segmented. Our segmentation algorithm however segments some characters in parts. The specialty of those characters is that they contain segmenting points in them. This is due to the various patterns observed inside the characters. Our refinement process strikes off some of the incorrect segmenting points, but still some of them reside in the characters. This is frequently the case when such characters appear close to segmented zone. Moreover the components extracted from the word are not always exactly individual characters. The characters are sometimes modified by some vowels. That is the component consists of a character as well as a vowel modifier attached with the character. We accept these two phenomenons as a natural. In the recognition phase it is always possible to concatenate parts of character together. The overall algorithm is given below—

1. The gray level image of a Bangla word is median filtered and then threshold into a binary image using Otsu threshold technique.
2. Lower contour of the word image is traced anticlockwise, and the points relevant for segmentation are extracted.
3. Based on the points extracted earlier various paths are constructed. Along with this construction several features are drawn for each path.
4. By optimizing the features the best path is chosen.
5. The path is used to detect and correct the skew in the word.

6. Connected components of a word to be segmented are detected.
7. Each connected component is further segmented using the point space drawn earlier.

Some examples of successful segmentation attempts are shown in Table 2 below. However we omit small and unimportant components. Only the main components are shown.

Table 2. Examples of handwritten words before and after segmentation

Original Words	Segmented Components
	
	
	

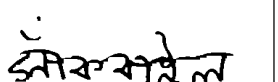
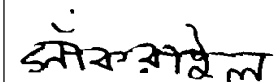
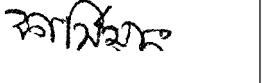
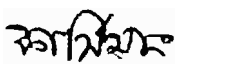
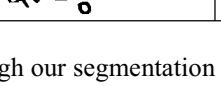
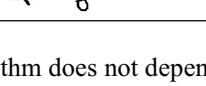
4. Results and discussions

Our moderately large database contains the variety of handwritten specimens (almost 150) for small, medium and large towns of West Bengal. We have simulated out scheme for entire database and have obtained satisfactory result. The novelty of our approach resides in the fact that it is a direct method. Once, through the procedure described earlier the candidate path is obtained, we could perform both the skew correction and segmentation with the help of candidate path. We do not need to adopt any separate method for skew correction and segmentation, thus the time consumption is quite reasonable. Majority of segmentation algorithms [5,6] discuss separate analysis of skew detection and correction.

The merit of the whole method mainly relies in the evaluation of accurate candidate path, which plays the role of Matra extracted from the Bangla handwritten words. For the evaluation of accurate candidate path the set of features f plays an important role. Moreover the method can cope simultaneously consistent and inconsistent skew of handwritten sample words under consideration. However skew detection method strongly depends on the correct construction of candidate path. Sometimes due to bad style of writing accurate candidate path cannot be constructed. This affects detection of skew.

Table 3 shows some less successful attempts to skew correction. However in case of these attempts the word becomes more skewed. This skewness can be corrected further by repeatedly applying the discussed method each time on the modified image. Sometimes the image becomes skewed by a constant angle (example in third row of table 3). In that case use of conventional skew detection algorithms [1-4] become beneficial.

Table 3: Examples of less successful attempts of skew correction

Original Words	Corrected Words
	
	
	

Although our segmentation algorithm does not depends directly on the skew detection and correction of the word, but obviously the correct detection of skew make the word upright, which facilitates refinement of the segmenting points in the segmentation phase. In case of uncommonly bad handwritten words it has been found that the candidate path appears in the middle zone of the word. Our segmentation process relies in the fact that candidate path should represent approximate Matra of the handwritten words. If somehow the segmenting points appear at the lower portion of the characters it is difficult to construct candidate path, which can be treated as the Matra. However this case is very rare in the sample database we have considered.

5. Conclusions and future research

An intelligent segmentation technique has been presented in this paper that produces fairly good results. The main characteristic of the approach is that knowledge of character structures and nature of Bangla handwriting are exploited in detail. The method can be extended to segmentation of large database of handwritten Bangla words covering a wide spectrum of type. This approach may be used for recognition technique of handwritten Bangla OCR system, which is useful for postal and office automation system. However significantly improved recognition result can be expected if this segmentation technique is combined with the recognition process in a holistic system.

6. References

[1] S.Srihari, V.Govindaraju, Analysis of textual images using the Hough transform, *Machine Vision and Applications, pp 141-153, 1989.*

[2] W. Postl, Detection of linear oblique structures and skew scan in digitized documents, *Proceedings of the 8th International Conference on Pattern Recognition, Paris, France, October, 1986, 687-689.*

[3] H.S.Baird, The skew angle of printed documents, *Proceedings of the Society of Photographic Scientists and Engineers, Rochester, New York, 1987,14-21.*

[4] Z.Shi, V.Govindaraju, Skew Detection for Complex Document Images Using Fuzzy Run length, *Proc. Of the Seventh Int. Conf. on Document Analysis and Recognition, ICDAR'03.*

[5] J. C. Simon, Off-line cursive word recognition, *Proc. IEEE 80(7), 1150-1161 (1992).*

[6] R. G. Casey and J. van Horne, Segmenting of touching characters in postal addresses, *U.S. Postal Service 5th Adv. Technical Conf. Vol. 3, pp. 743-745, Washington DC, (November 1992).*

[7] Casey, R.G. and Nagy, G., Recursive Segmentation and Classification of Composite Patterns, *Proc. Sixth Int'l Conf. Pattern Recognition, pp. 1023-1026, Munich 1982*.

[8] Casey, R. and E. Lecolinet, A Survey of Method and Strategies in Character Segmentation, *IEEE Transaction on PAMI, 18(7), pp. 690-706, 1996.*

[9] Lu, Y. and M. Shridar, Character Segmentation in Handwritten Words – An Overview, *Pattern Recognition, 29(1), pp. 77-96, 1996.*

[10] U. Pal and Sagarika Datta, Segmentation of Bangla Unconstrained Handwritten Text, *Proc of the 7th International Conference on Document Analysis and Recognition (ICDAR 2003).*

[11] A. Bisnu and B. B. Chaudhuri, Segmentation of Bangla handwritten text into characters by recursive contour following, *Proc. 5th ICDAR, pp. 402-405, 1999.*

[12] N.Otsu: A thresholding selection method from graylevel histogram, *IEEE Transactions on Systems, Man, and Cybernetics, 9 (1979) 62–66.*