Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 1825–1839
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

# Text line extraction from multi-skewed handwritten documents

S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri*, D.K. Basu

*Computer Science and Engineering Department, Jadavpur University, Kolkata 700032, India*

Received 8 November 2005; received in revised form 18 August 2006; accepted 2 October 2006

**Abstract**

A novel text line extraction technique is presented for multi-skewed document images of handwritten English or Bengali text. It assumes that hypothetical water flows, from both left and right sides of the image frame, face obstruction from characters of text lines. The stripes of areas left unwetted on the image frame are finally labelled for extraction of text lines. The success rate of the technique, as observed experimentally, are 90.34% and 91.44% for handwritten Bengali and English document images, respectively. The work may contribute significantly for the development of applications related to optical character recognition of Bengali/English text.


*Keywords:* OCR; Multi-skewed documents; Text line extraction; Connected component labelling; Skew angle detection; Touching line segmentation

## 1. Introduction

Text line extraction from optically scanned document images is one of the major problems of optical character recognition (OCR) of *printed/handwritten* text. Appearance of *skewed lines* in the text makes the problem complex. The problem becomes compounded if the lines in a text image are skewed with different orientations. Such lines are called *multi-skewed lines*. Appearance of multi-skewed lines in text images is common to both printed and handwritten texts for various reasons.

Lines in a text image get skewed mainly for two reasons. *Firstly*, a few degrees of misalignment of the document with respect to the scanner or copier bed is unavoidable at the time of scanning. This makes all text lines in the document image uniformly skewed as illustrated with a sample text image in Fig. 1(a). *Secondly*, text lines in the original document may be skewed differently for either some individual's *handwriting style* or some special *design choice*. Images of such documents always consist of multi-skewed lines. Figs. 1(b)–(d) and 2(b) show four sample images of multi-skewed lines. In one of these images, shown in Fig. 1(b), skewness of each text line is different from that of the others. For the rest of the two images, shown in Fig. 1(c)–(d), skewness of one part of each individual text line differs from that of some other parts of the same line. The text line extraction technique to be presented here can deal with all sorts of skewness described above.

### 1.1. The problem of text line extraction

The problem of text line extraction from optically scanned document images is simple under the *ideal situation*. In such situation, document images contain unskewed text lines, i.e., all the text lines therein have parallel orientations with some edge of the image frame. Text lines from these images can be easily extracted just by identifying valleys of *horizontal pixel density histograms* of the text lines as shown in Fig. 2(a). But this technique fails for document images with skewed lines, i.e., for all practical situations. One such document image is shown in Fig. 2(b). A straightway solution in such situation may be one which suggests for skew correction first and then line extraction with horizontal pixel density histograms of document images. But it does not work with complex cases of skewness of text lines. Special techniques are necessary to deal with these cases. In some of these techniques, skewed text lines are first extracted from

* Corresponding author. Tel.: +91 3324146766.
*E-mail address:* nasipuri@vsnl.com (M. Nasipuri).

0031-3203/$30.00 
doi:10.1016/j.patcog.2006.10.002

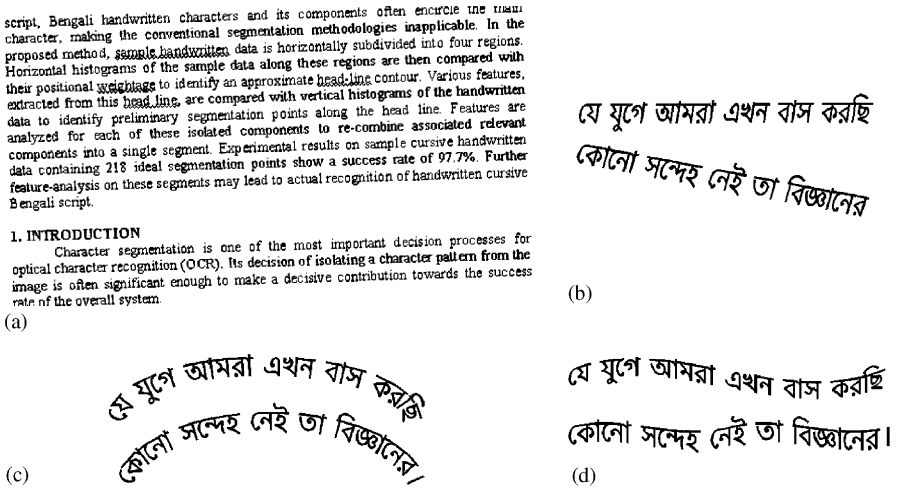



Fig. 1. Document images show different types of skewness: (a) a uniformly skewed image of a text document; (b) an image of text lines with different angles of inclinations; (c) an image of curved text lines; (d) an image of wavy text lines.

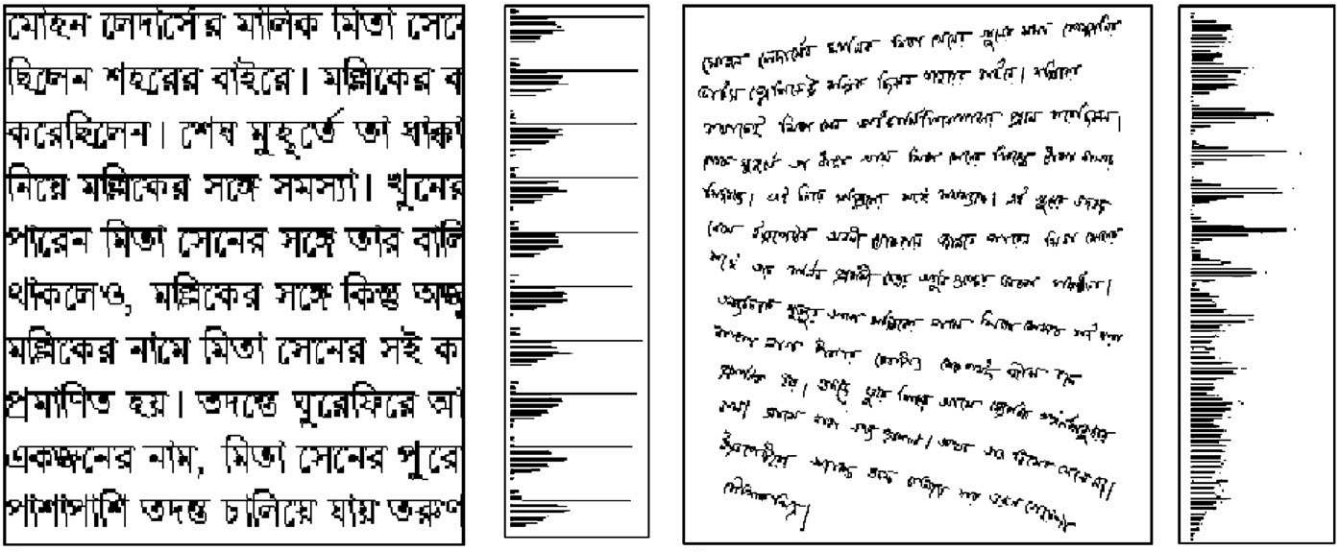



Fig. 2. Difficulty of text line extraction using horizontal pixel density histograms of skewed document images is illustrated: (a) horizontal pixel density histograms of a document image with unskewed text lines; (b) horizontal pixel density histograms of a document image with skewed text lines.

document images and then performing correction to these lines becomes a trivial problem.

Solutions so far devised for the text line extraction problem can be grouped into *three categories*. Each category of these solutions targets certain *kind of skewness* of the text lines in document images for certain particular *script*, either *handwritten* or *printed*. The first category of solutions deals with the uniformly skewed text lines in an image of a document page, which is similar to one shown in Fig. 1(a). The second category of solutions deals with nature of skewness shown in Fig. 1(b). And the third category of solutions deals with complex nature of skewness of text lines, similar to those shown in Fig. 1(c), (d).

### *1.2. The first category of solutions*

For text images of *printed Roman script*, most skew correction and line extraction techniques predominantly deal with a single skew angle for an entire document page [1–3]. For dealing with *handwritten text* of *Roman script*, the techniques, as described in Refs. [4–6], all principally determine a skew angle from a page of text lines on the basis of the *base lines* of the text words before skew correction. To extract text lines and words from document images of *handwritten English text* lines, a work described in Ref. [7] uses *horizontal and vertical histogram* values of the same.

A work is described in Ref. [8] for performing skew correction to document images of *printed Bengali and Devnagri scripts*, in which all text lines are equally skewed. With each image of one such document page, it computes a *skew angle* so that the document image can be subsequently rotated in a suitable direction by an angle same as the computed skew angle, for necessary *skew correction*. The technique is based on identification of digital straight line segments (DSLs) along the '*shirorekha*' or '*Matra*' of the scripts and then computation of the skew angle of the document image from the inclinations of the DSLs. The *shirorekha* is an important feature of Bengali text. It is also known as '*Matra*' or *head line*. A 'Matra' is a horizontal line touching the upper parts of the most of the characters of Bengali script. 'Matras' of consecutive characters in a word are joined to form a common 'Matra' of the word.

### 1.3. The second category of solutions

The concept of skew angle computation on the basis of DSLs has been extended in Ref. [9] to deal with images of *printed Bengali text* lines with different orientations. In this work, the *upper envelope* of each printed text line is used for identifying DSL segments. DSL segments so identified are then clustered on the basis of normal distances of all DSL segments from the longest one. All the DSL segments, which are grouped into a single cluster, represent the parts of a single text line, to be extracted from a document image of differently skewed text lines.

A work presented in Ref. [10] involves cut text minimization (CTM) for segmentation of text lines from handwritten English text documents. In doing so, an optimization technique is applied which varies the cutting angle and start location to minimize the text pixels cut while tracking between two text lines.

As a part of a digitization project of cultural heritage manuscripts, a production system [11] is applied for text line segmentation in handwritten textual documents of English script. In this work, a textual document is considered as a set of objects with spatial relations between them. These objects represent text lines and connected components in the document. Since a graph is a natural choice for representing relations between objects, the global database of the production system is represented with a graph under the work. The nodes of this graph represent connected components and the text lines of the document, and the edges of the graph represent adjacency relations between objects. Each edge in the graph is weighted by the gap measure between the two objects in the document image.

### 1.4. The third category of solutions

The technique described in Ref. [12] can deal with complex skewnesses of *printed text* lines of *Roman script*. It can accept printed pages of *non-rectangular layouts* with various skewnesses of text lines. The technique performs *thinning* on backgrounds of text images. The background being so thinned produces loops around various textual portions of the input document image. Irrelevant loops are removed by using some predetermined distance and width thresholds. These thresholds are computed on the basis of the average line width.

To deal with complex types of skewness of *printed Bengali text* lines, as shown in Fig. 1(c), (d), *water reservoir principle* is applied in Refs. [13,14]. Left, right, top and bottom reservoirs are used for detection of both isolated and touching word components in text images. Word components thus detected are finally so clustered into groups that each group contains all word components belonging to a single text line. The techniques described above have limited applications only on images of printed texts of Indian scripts, Bengali or Devnagri. Such a technique mostly considers the *upper envelope* of a text line in determining skewness of the same. It becomes possible because text lines of Bengali or Devnagri script are featured with prominent *head lines or* '*Matras*'. The upper envelope in the Roman script does not necessarily represent the slope of the text lines. Therefore, the upper envelope based skew correction and line extraction techniques [9,13,14] cannot be applied for Roman scripts due to the absence of prominent headlines.

A technique 'extended linear segment linking' (ELSL) is described in Ref. [15] to extract both of the multi-skewed and curved text lines from images of printed documents. It is also applicable to document images containing both horizontal and vertical text lines on the same page. Such text lines are found in documents of Japanese language. In ELSL technique, a document is split into some small sub-regions and local orientations of text lines in all the sub-regions are then estimated. The consecutive sub-regions with the same orientation are finally connected to extract the text line therein. The technique is applicable to text documents with characters of pre-fixed maximum and minimum sizes.

### 1.5. Motivation

From the above discussions, it is clear that there is a need for developing a general technique for extraction of text lines from multi-skewed document images, prepared with Roman or Bengali script, either handwritten or printed, as demanded by some specific applications. Keeping this in mind, the present technique has been developed.

Ideas, which have motivated the work, are as follows. All text lines in a document are separated from each other with uniform or non-uniform spacings depending on nature of skewness of the lines. To access these lines, all line spacings in the document are to be labelled first. Each of the unlabelled stripes of text left in the document image is then to be labelled distinctly to identify different text lines in the document. The technique so conceived can work irrespective of nature of skewness of text lines in the document image.

## 2. The present work

To get over the hurdles for implementation of the above idea, some technique is *firstly* required to label all line spacings in the document irrespective of their degrees of uniformity. *Secondly*, a technique is also required to identify separately all unlabelled stripes left after labelling of line spacings in it.

To develop a technique for labelling all line spacings in the document image, the present work hypothetically assumes a flow of water in a particular direction across the image frame in a way that it faces obstruction from the characters of the text lines. In this hypothetically assumed situation, water flowing across the image frame does not wet those areas, which are guarded by such surfaces. Simply speaking, a hypothetical water flow across the image frame is expected to fill up the gaps between consecutive text lines. The unwetted areas left on the image frame ideally lies under the text lines.

Speaking very specifically, in Bengali or Roman script, one or more components of textual characters may enter into an adjacent line spacing making the task for identifying the line spacing from the text line difficult. It is illustrated in Fig. 3. In such cases, a text line is to be so extracted that a strip of the adjacent line spacing covering the extended portion(s) from the text line can be included in the same. But the water flow technique as described so far cannot do this. Identification of a text line by labelling it as a unwetted stripe on the image frame of a text document is not possible in such cases as water passes freely through a line spacing until it is blocked by the extended portion of textual character entering therein. To isolate such text lines, line spacings are to be identified first by labelling the stripes of areas wetted with water flowing from two opposite directions across the image frame. The stripes, which remain unwetted or wetted by water flowing from just one particular direction, are to be labelled then differently to isolate text lines in the document image.

Once labellings of the document image are thus completed, the entire image is divided into two different types of stripes, one containing text lines and the other containing line spacings. The former category of stripes will be referred to as *white stripes* and the latter category of stripes as *dark stripes* in the subsequent discussion.

Unlike the dark stripes, pixels in each white stripe are not all initially labelled uniquely. This is needed for maintaining the difference between text and non-text pixels in the white stripe.

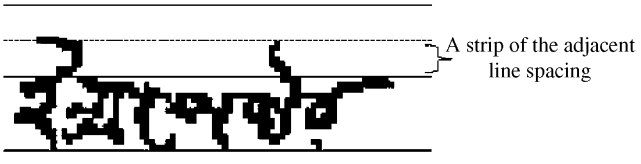



Fig. 3. Extended portions of text characters entering into the upper adjacent line spacing are shown.

Water hypothetically flowing from two opposite directions across the image frame divides it into a number of wetted and unwetted stripes. To ensure precise wetting of all line spacings in a document image, a parameter called the *flow angle* is to be controlled depending on the nature of skewness observed in the document image. All these are discussed in detail in the subsequent section.

Identification of wetted stripes in a document image is not sufficient for extraction of text lines from the same. All unwetted stripes in the document image are to be labelled distinctly before text line extraction. Connected component labelling algorithm [16] is applied for this purpose. This is in a nutshell how the technique developed under the present work is applied to extract multi-skewed text lines from handwritten document images.

### 2.1. The technique of labelling line spacings

It has already been mentioned that line spacings in a document image is labelled here by wetting them under a hypothetical situation. Fig. 4(a)–(d) illustrates how the surface of an image frame is wetted under hypothetical water flows from different directions. It is the *flow angle*, shown in Fig. 5, that determines the wetted areas on an image frame. To adjust the flow angle to a desired value, the flow of water from a hypothetical nozzle behind the left most column of the image frame is to be controlled.

The value of the flow angle chosen for a document depends on the average skew angle, the average line spacing and also the average word spacing in the document. Taking these three factors into consideration, it is to be so chosen that water can flow freely along line spacings without entering into inter-word gaps in text lines. Usually, the higher is the skew angle the greater should be the flow angle chosen. But for a document with wider word spacings, a high value of the flow angle may allow the hypothetical water flow to enter into text lines creating problems in labelling of line spacings. So the value of the flow angle need not always be high for extracting lines from a highly skewed document. For a large line spacing, a comparatively small value of the flow angle can do the job even with a high value of the skew angle. It is however safe to choose a value for the flow angle, which should at least be greater than the maximum of the skew angles of the text lines in a document of interest.

The flow angle is measured between two lines which intersect each other at an endpoint of an obstacle as shown in Fig. 5. One of these two lines passes across the width of the image frame and the other passes over the borderline of the wetted area hypothetically created before the obstacle. For a specific flow angle, labelling of wetted areas on an image frame starts from the left most column of the frame for left to right water flow. More specifically speaking, it begins from the top left pixel position of the column and is continued up to the bottom most pixel position of the same one by one. In doing so, if the current pixel is found to be

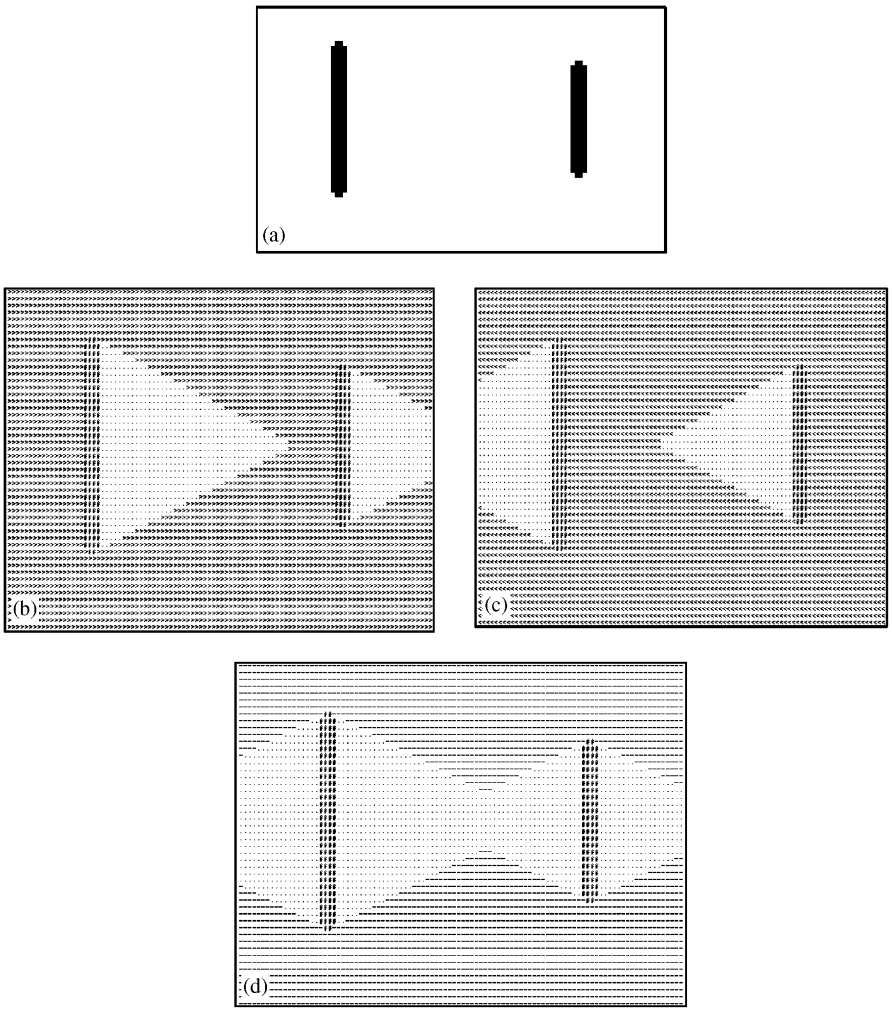



Fig. 4. Wetting of the 2-D surface of an image frame under hypothetical water flows from different directions is illustrated: (a) two obstacle surfaces are shown in cross-sections with dark bars in a 2-D image frame; (b) unwetted regions in the image frame are shown in white after water flows hypothetically from left to right over the same; (c) unwetted regions in the image frame are shown in white after water flows hypothetically from right to left over the same; (d) unwetted regions in the image are shown in white after water flows hypothetically from two opposite directions across the same.

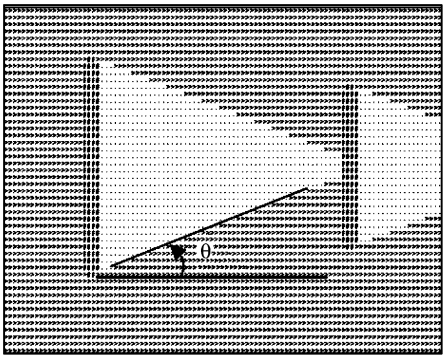



Fig. 5. An illustration of the flow angle ($\theta$).

a black one, i.e., a part of an obstacle, then it is left as it is. Otherwise, it is labelled with a special pixel value denoting wet areas. This special value is denoted by the symbol '>'.

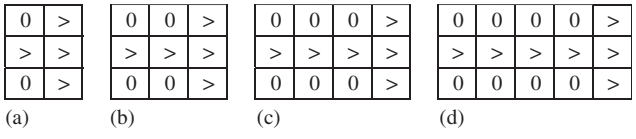



Fig. 6. Left to right labellings of images with the symbol '>' for different values of the maximum flow angle ($\theta$) are illustrated: (a) $\theta = 45°$; (b) $\theta = 26.6°$; (c) $\theta = 18.4°$; (d) $\theta = 14°$.

Starting from a pixel position, how far wetting will be continued along the image frame depends on the value of the flow angle chosen for the work. For values of the flow angle chosen as 45°, 26.6°, 18.4° and 14°, it will be continued up to 1, 2, 3 and 4 consecutive columns, respectively. The labellings produced for these operations are illustrated in Fig. 6(a)–(d). It is assumed in the illustrations that water flows from only one pixel position of the left most column.

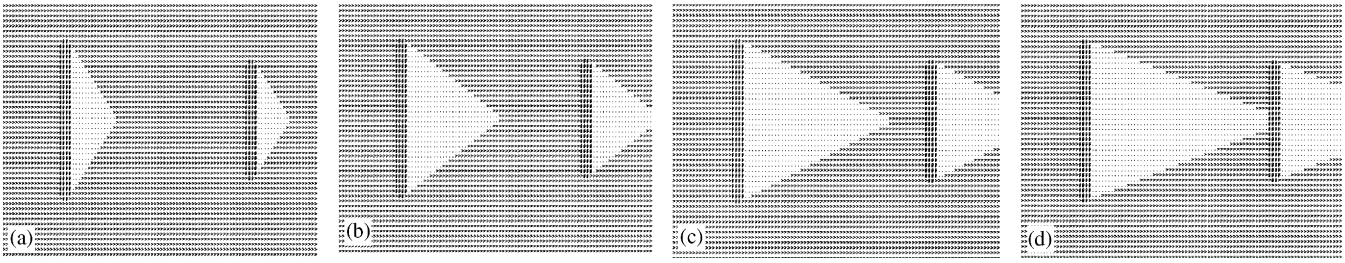



Fig. 7. Unwetted regions produced from left to right water flows of different $\theta$ values are shown in white: (a) $\theta = 45°$; (b) $\theta = 26.6°$; (c) $\theta = 18.4°$; (d) $\theta = 14°$.

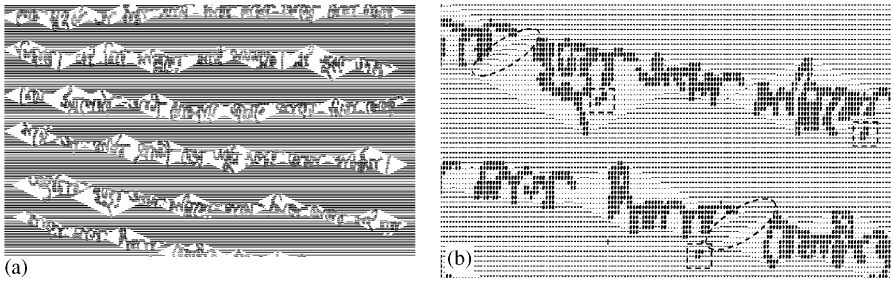



Fig. 8. How isolated character components appear within white stripes and white stripes within word spacings is shown: (a) a sample image in which line spacings are labelled with 'dark stripes' is shown; (b) two periods appearing within dark stripes and one within a white stripe and two wetted inter-word spacings are shown in the same image.

After performing labelling from the left most column of the image frame, it is continued with the next column following the same technique. In doing so if a pixel position, to be labelled with '>', is already found labelled with the same then it need not be labelled any more. But this pixel position is to be considered as a source of some hypothetical water flow for the subsequent columns on the image frame. This is to simulate the water flow from one point to others on the image frame. The simulation is effected by marking all appropriate pixel positions in the subsequent columns with the symbol '>'. Again up to which column this process of marking will be continued from a specific pixel position depends on the flow angle chosen for the work. The specific pixel positions to be labelled with '>' in the subsequent columns also depend on the same. How these pixel positions are to be determined are already shown in Fig. 6(a)–(d). The entire process described so far is continued up to the last but one column from the right most column of the image frame. It is so because the hypothetical water flows from each column reaches at least its next column for all possible values of the flow angle. Fig. 7(a)–(d) show how by repeating left to right labellings, already shown in Fig. 6(a)–(d) for different $\theta$ values, unwetted regions are formed beside obstacles.

This is for simulating a left to right water flow on the image frame through labelling of appropriate pixel positions on it with '>'. For simulating a water flow in the reverse direction, labelling starts from the right most column of the image frame and is continued up to the last but one column in a way same as before. In order to make labels indicative of the direction of water flow, the symbol used as the label in this case is chosen as '<'. To find the common area of the image frame, which is wetted by water flows from both left and right, the pixel positions labelled with both the symbols, '>' and '<', are to be relabelled each with some special symbol '-'.

### 2.2. Erosion of the dark stripes

Some isolated components, mostly dots, from the text written in Roman or Bengali may appear outside the white stripes, i.e., in the dark stripes, as illustrated in Fig. 8(a), (b). Such components are missed when text lines are extracted from the document images by marking all white stripes therein separately. To prevent textual components from being so missed, the dark stripes in document images need to be eroded morphologically. It is also required to wipe out water hypothetically wetting large inter-word spacings, shown in Fig. 8(b). How much morphological erosion [17] is required for a document image depends on the writing style followed therein. To tune the degree of erosion with the writing style, a circular structuring element of a variable radius $k$ is considered here. How such measure can help in including isolated textual components in white stripes and wiping out water from larger inter-word spacings are illustrated in Fig. 9 by showing the sample of document image same as one previously shown in Fig. 8(a).

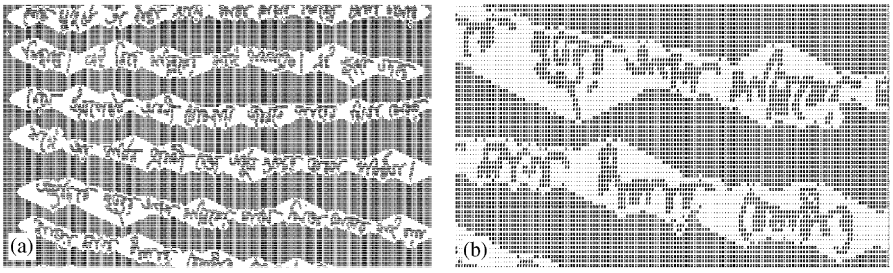



Fig. 9. The images illustrate how iterative thinning of dark stripes helps to include two periods, as isolated character components, in the white stripes and wipe out water wetting large inter-word spacings: (a) the document image produced after thinning of black stripes; (b) a close-up view of the same image.

| | | |
|---|---|---|
| $q_0$ | $q_1$ | $q_2$ |
| $q_3$ | $p$ | X |
| X | X | X |

Fig. 10. 8 neighbors of the center pixel $p$ are shown.

### 2.3. Text line extraction

Text line extraction from a striped document image simply requires extraction of all white stripes from the image one by one. To do so, all white stripes appearing in the document image are to be labelled uniquely so that each stripe can be copied pixel by pixel into a separate file. Once a stripe is thus copied, the text written therein is isolated from the rest of the document. To make the labelling of such stripe unique compared to labellings of other white stripes of the image, pixels belonging to the same white stripe are labelled with same symbols whereas pixels belonging to different white stripes are labelled with different symbols. This is required for making labellings of separate text lines different after all white stripes are labelled in the document image. In doing so, all text and non-text pixels within a white stripe are to be considered same. After the stripe is copied into a file, text pixels are to be restored in the copy from the original document image.

To label text and non-text pixels in a white stripe with same symbols, the connected component labelling algorithm [16] is applied here to trace connectivities among all pixels of the same white stripe. For this, the algorithm is to be modified so that it can find connectivities among all text and non-text pixels of the same stripe. In doing so, it assumes pixel values of 1,0 and '-' for text, non-text and 'dark' pixels, respectively.

To label all connected pixels in an image identically, the connected component labelling algorithm scans the image pixel by pixel from left to right and from top to bottom. During scanning, it considers all eight neighbours of each pixel $p$ as shown in Fig. 10. Out of the eight neighbours, four are specially marked as q0, q1, q2 and q3 in the Fig. 10. The pixel $p$ is labelled if it has a pixel value '0' or '1'. Otherwise it is skipped during the scan. To decide about a symbol for labelling the pixel $p$, pixel values of $p$ and its four neighbours are to be compared. If all four neighbours of $p$ have values same as '-' then $p$ is to be labelled with a new symbol. If only one of the four neighbors has a value '0' or '1' then $p$ is to be labelled with the label of any such neighbour. The labels of the other neighbours of values '0' or '1', if found different from the label assigned to $p$, are to be stored in an equivalent table. This is about the first pass of the connected component labelling algorithm. In the second pass of the algorithm the equivalent pairs are grouped into equivalent classes by applying the property of transitivity. For each of the equivalent classes, all its member labels occurring in the image are to be replaced by a single distinct symbol. This is to complete labellings of the connected pixels in the image finally.

The above algorithm, when applied to a white stripe containing a text line touching another text line in the next white stripe, labels the two consecutive stripes as a single component. This is so because a hypothetical water flow from either side of the image frame is obstructed by the segment connecting the two text lines. The white stripes formed at the initial stage of labelling of such text lines contain two text lines. The situation is depicted in Fig. 13 with a sample of document image containing a pair of touching text lines.

The present work also employs a technique to separate such text lines of handwritten Bengali script. But it requires computation of skew angle from the upper envelope of the topmost one of the touching text lines in a white stripe. The technique is described below.

### 2.4. Skew angle detection

The skew angle detection technique employed here for text lines of handwritten Bengali script [18] is based on detection of the upper envelop of the text line. This is so because from the angle of inclination of the common 'Matra', formed with the 'Matras' of individual characters of a text line, the skew angle of a line is detected under the present technique. But the problem with handwritten Bengali text is that the common 'Matra' therein is not very well formed compared to that of printed Bengali text. For this reason the upper envelope of a text line is considered here before its skew angle detection, as shown in Fig. 11.

An upper envelope vector, $E = (e_1, e_2, \ldots, e_n)$, is computed by calculating the height of the upper envelope pixel in

each column from the bottom most row. The skew angle detection technique presented here starts by filtering out upper envelope pixels of a text line that do not lie on the 'Matra' of the text written therein. This is to exclude the portions of certain Bengali characters extended over the common 'Matra' and the portion of the upper envelope of a lower text line visible through an inter word spacing. To filter out certain upper envelope pixels as mentioned here, a difference vector, $D = (d_1, d_2, \ldots, d_{n-1})$, is calculated from the upper envelope of the handwritten text. For each column $i$, $d_i$ is calculated by computing the difference of successive column values of upper envelope vector, $E$:

$$\text{i.e.,}\ \ d_i = (e_i - e_{i-1}); \quad i = 1, \ldots, n-1.$$

If the absolute value of $d_i$ for any column is more than a heuristically chosen threshold $\partial$, then the $i$th pixel of the upper envelope is discarded, as it is not a part of the headline.

The complete upper envelope is then divided into heuristically chosen $N$ intervals of equal widths. Within each interval, sum of all $d_i$'s is calculated. Thus, the slope of the upper envelope within the $j$th interval $S_j$, $\forall j \in N$, is calculated as follows:

$$\text{i.e.,}\ \ S_j = \left(\sum_i d_i\right) \Big/ \text{width of } j\text{th interval},$$

$$\text{therefore}\ \ \tan\alpha_j = S_j,$$

$$\text{therefore}\ \ \alpha_j = \tan^{-1}(S_j),$$

Fig. 11. The image shows a text line and its upper envelope.

where $\alpha_j$ = slope angle of the topmost text line within the $j$th interval.

Once slope angles for all the intervals are calculated, initial mean slope angle, $(M_s)$, and standard deviation, $\sigma_s$, are calculated.

$$\text{i.e.,}\ \ M_s = \left(\sum_j \alpha_j\right) \Big/ N,$$

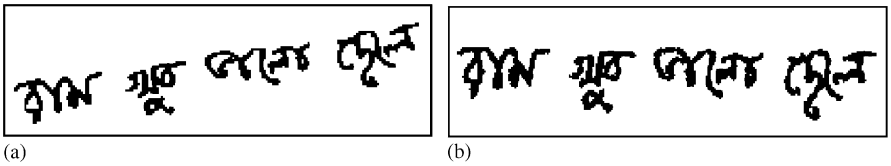
$$\sigma_s = \sqrt{\sum_j (\alpha_j - M_s)^2}.$$

Now, to iteratively refine the mean slope angle $M_s$, slope angles $\alpha_j$ for every interval is again compared with $M_s$. If the absolute difference between $\alpha_j$ and $M_s$ is greater than $\sigma_s$, then the contribution of $j$th interval to the formation of the head line is considered to be insignificant. Hence this interval is discarded. Mean slope and standard deviation are recomputed for the rest of the intervals. The process of refinement of $M_s$ is repeated as before to identify any further insignificant interval. It is continued until no further insignificant interval is detected. This final value of $M_s$ is considered as the estimated value of the skew angle, $\alpha_s$, of the present text line.

For applications requiring skew correction, the extracted text line needs to be rotated by an angle $-\alpha_s$ using simple rotation transforms. This is to realign the text line across the image frame, as illustrated with sample images in Fig. 12(a), (b).

### 2.5. Separating touching text lines

For separating touching text lines in a white stripe (Fig. 13), a straight line making an angle $\alpha_s$ with a horizontal line across the image frame and passing through either the right most pixel point or the left most pixel point of the upper envelope of the top most touching line is to be drawn. If $\alpha_s$ be positive then the top right corner point is chosen. Otherwise, the top left corner point is chosen. The line so drawn is parallely shifted pixel by pixel towards the bottom of the image frame. After every time it is shifted, the number of text pixels, through which it passes, is to be counted. The shifting operation stops if the count goes below certain threshold ($\eta$). Segmentation is finally performed along that position of the line. A sample image of text lines so segmented is shown in Fig. 14. At this point of



(a) (b)

Fig. 12. The images illustrate skew correction through rotation of a text line by an angle-$\alpha_s$: (a) a text line before skew correction; (b) the text line after skew correction.

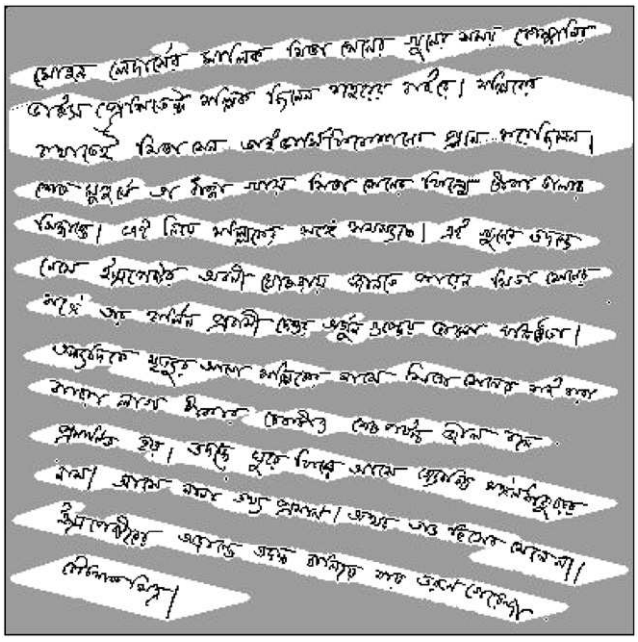


Fig. 13. A document image containing a pair of touching text lines is shown.

discussion, it is noteworthy to mention that the technique runs the risk of improper segmentation of the character linking two text lines. But with this risk, almost all the contents of two touching lines can be restored.

## 3. Results and discussion

To conduct experiments with the technique described here, various samples of English and Bengali documents have been collected from different sources. The documents are digitized using a flatbed scanner at a resolution of 300 dpi. The digitized documents so prepared are finally binarized simply through *thresholding*.

The performances of the present technique are tested separately with samples of printed and handwritten text lines. Two data sets, one with *printed text* lines and the other with *handwritten text* lines, are developed for this purpose. Detailed descriptions of the two sets are given in Table 1(a), (b).

Each sample line of printed text used here consists of 8 words on average and that of handwritten text consists of 12 words on average. A few samples of document images taken from Data Set #1 and Data Set #2 are shown in Figs. 1(a), (c), 15(a), (b), and 17.

Table 1
Descriptions of (a) Data Set #1 and (b) Data Set #2

(a) *Data Set #1 consists of 360 lines of printed text*

| Language | Number of text lines | |
|---|---|---|
| | Uniformly skewed | Complexly skewed |
| English | 72 | 100 |
| Bengali | 91 | 97 |

(b) *Data Set #2 consists of 1191 lines of handwritten text*

| Language | Number of text lines |
|---|---|
| English | 456 |
| Bengali | 735 |

Experiments are conducted to test the performances of the present technique on uniformly skewed and complexly skewed samples of Data Set #1 separately. Each of these two types of samples comprises of images of both English and Bengali text lines. To test performances of this technique on images of handwritten documents of Data Set #2, samples of English and Bengali text lines are considered separately.

Choices for the values of the *flow angle* ($\theta$) and the *radius of the structure element* ($k$) involve the most critical decision related to successful application of the present technique. Strictly speaking, these decisions require case-to-case basis consideration of document images. However, for the *uniformly skewed printed documents* considered here, the skew angles vary between $-14°$ and $+14°$. Such documents are reasonably considered here as the amount of skewness occurring due to misalignment of document pages with the scanner or copier bed usually lies within this range. The line spacings in these documents vary from 2 to 8 mm. Considering all these, the values of $\theta$ and $k$ are chosen as 14° and 3, respectively. With these values, the present technique shows 100% success rate in extracting all the 163 text lines from sample images of uniformly skewed documents of Data Set #1. Fig. 16(a) illustrates successful extraction of text lines from one such document image, already shown in Fig. 1(a). The extracted text lines in Fig. 16(a) are highlighted with shadings. Because of the uniformly spaced words of samples of printed documents used here, and the variation of skew angles of the sample documents written within a fixed range, it has been possible to select some common values of $\theta$ and $k$ for all

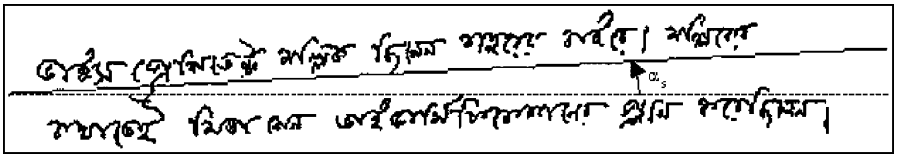



Fig. 14. The line drawn at an angle $\alpha_s$ separating a pair of touching text lines is shown with $\eta = 2$.

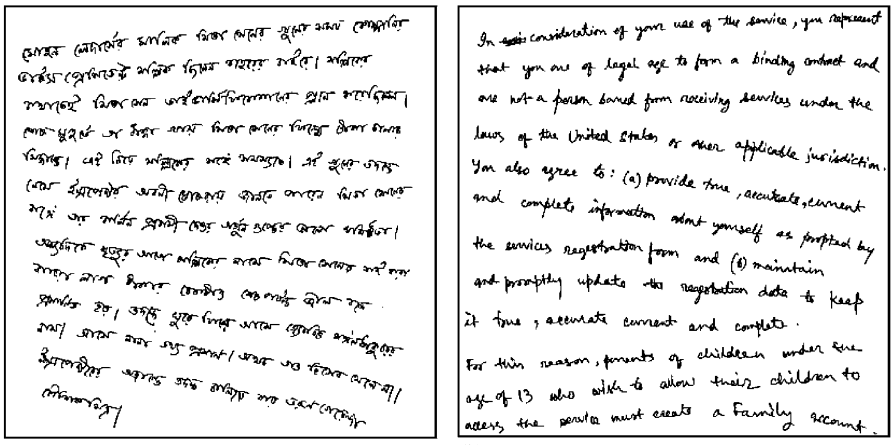


Fig. 15. Two samples of document images taken from the experimental dataset are shown: (a) a sample document image of handwritten Bengali text; (b) a sample document image of handwritten English text.

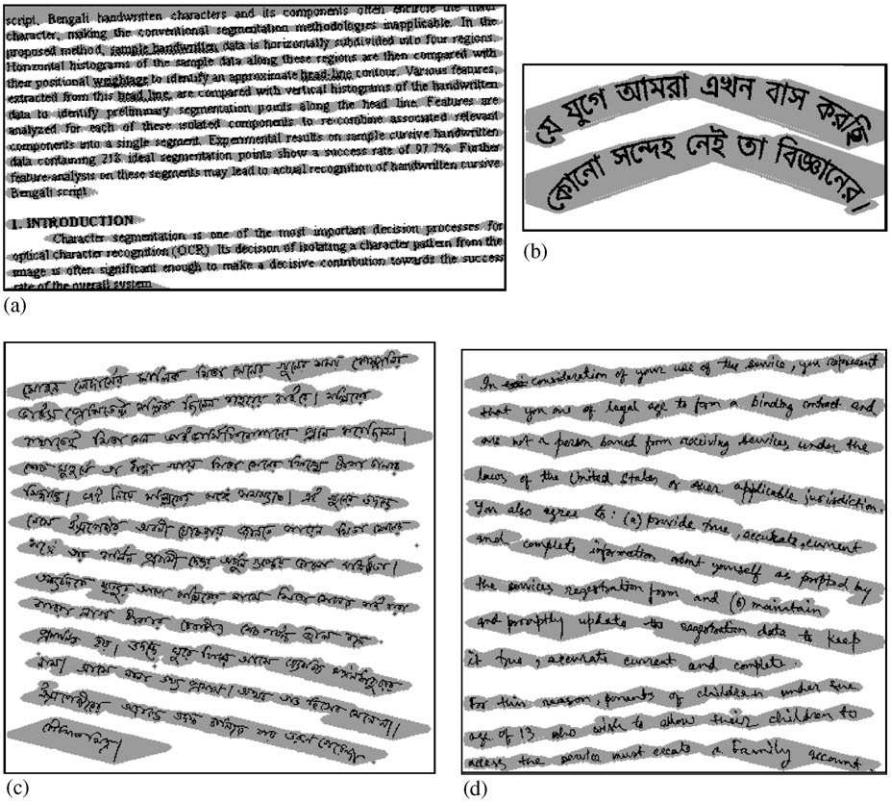


Fig. 16. (a)–(d). Extracted text lines are highlighted with shadings in the images formed as a result of applying the present technique on the document images of Figs. 1(a), (c) and 15(a)–(b).

the documents so appropriately that the 100% success rate has become achievable for the technique. Compared to 100% success rate of the present technique on uniformly skewed printed documents, the technique described in Ref. [14] has shown 100% success rate only in 41.6% cases of 'Single-oriented' printed Bengali documents.

For application of the present technique on document images of *complexly skewed* text lines of Data Set #1, the values of $\theta$ and $k$ are to be chosen after considering the document images individually. For one such document image, shown in Fig. 1(c), the values of $\theta$ and $k$ are chosen as 26.6° and 3, respectively. The extracted lines from the document image are highlighted with shadings in Fig. 16(b). With the choices of values of $\theta$ and $k$ on a case-to-case basis, the present technique successfully extracted all the 197 complexly skewed printed text lines from the document images

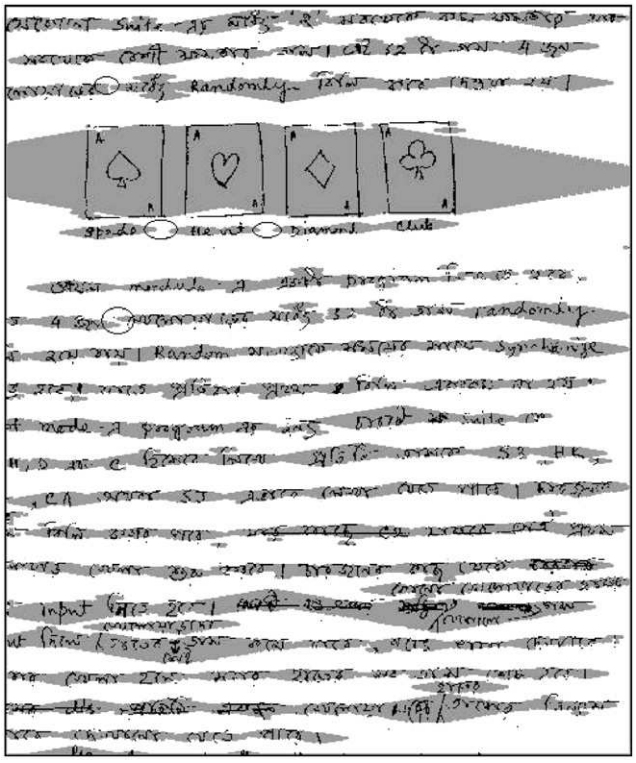


Fig. 17. A sample image of handwritten Bengali text with handwritten figures, scratches, English words and numbers ($\theta = 9.5°, k = 2$).

of Data Set #1. The success rate of the technique as recorded here is again 100%. Compared to this, the technique described in Ref. [14] has shown 100% success rate in 35.73% of 'Multioriented documents' and 32.45% of 'Curved documents'.

Skew angles in handwritten document images of Data Set #2 vary between $-14°$ to $+14°$. This variation is normal except for specially designed documents. The line spacings in such document images vary from 5 to 12 mm and the word spacings from 4 to 16 mm. Considering all these, the values of $\theta$ and $k$ are chosen as 14° and 4°, respectively, for application of the present technique on handwritten document images of Data Set #2. The average success rates as observed from this experimentation are 91.4% and 90.34% for English and Bengali text lines, respectively. Two of the document images of Data Set #2 are already shown in Fig. 15(a), (b). Successfully extracted lines from these images are shown Fig. 16(c), (d).

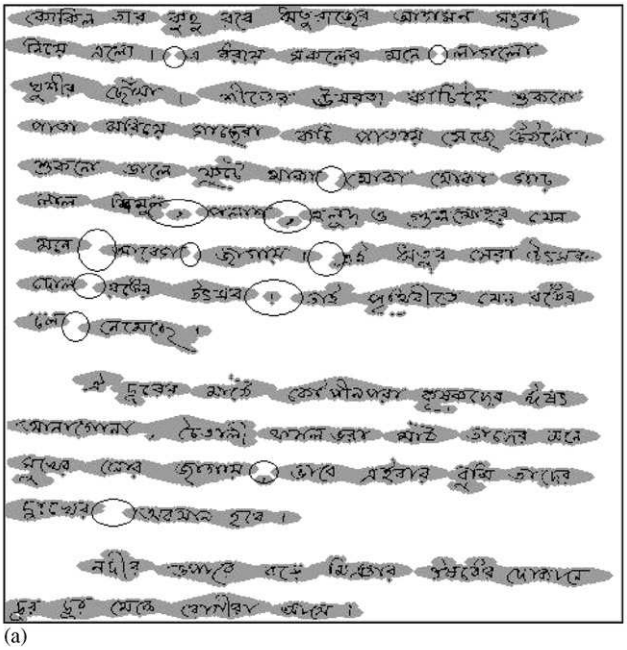
A document image sample of handwritten Bengali texts with handwritten figures, scratches, English words and numbers is shown in Fig. 17. All the text lines from one such complex document are extracted successfully excepting at four positions, shown encircled in the figure. Out of these four positions, two are below the handwritten figures in the document. All the four captions of these figures ought to be extracted as a single text line. The technique has failed to extract them in two places possibly because of the entry of the hypothetical water flow in the word spacings due to very small discontinuities in the bottom most lines of the corresponding figures. In the two other encircled places in the sample document image of Fig. 17, the technique has failed because of comparatively larger word spacings and misalignment of consecutive words.

The average success rate of the present technique as observed on documents of handwritten Bengali text is slightly lower than that of handwritten English text. It is because of the fact that Bengali script has much larger numbers of ascendants and descendants compared to English script. All these *ascendants* and *descendants* appearing in the line spacings make the problem of line extraction more difficult for Bengali documents, by creating obstruction to hypothetical water flows across the document images.

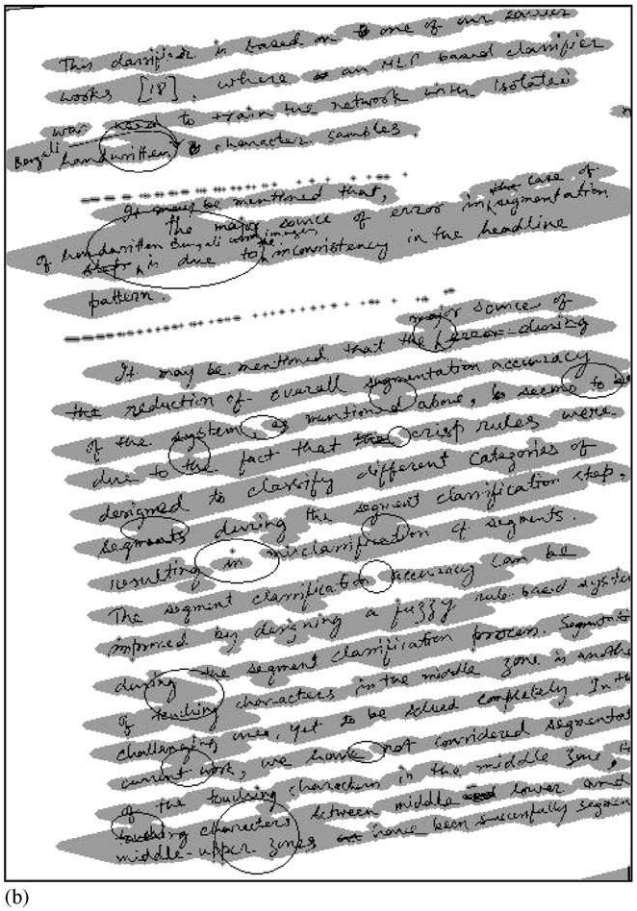
Two of the document images, for which the present technique fails to extract certain text lines, are shown in Fig. 18(a), (b). The failure occurs where more than one line is extracted together or where a single text line is divided into two or more parts. The portions, where such failure occurs, are shown encircled in Fig. 18(a), (b). The lines which are extracted together from a document image are either narrowly spaced or connected. Certain text lines of the document images shown in Fig. 18(a), (b) are divided into parts because of choices of a high value for $\theta$, compared to the average skew angle of each of these documents. This may also happen for large word spacings in the documents. In such cases, user intervention is necessary to tune the values of $\theta$ and $k$. Improved results are obtained on the document images of Fig. 18(a), (b) with readjustment of values of $\theta$ and $k$. It is shown in Fig. 19(a), (b).

Fig. 20 shows a sample document image of handwritten English text. It is a multioriented document with scratches. The failure points in the document are all encircled. From closer observations of these failure points, it can be analyzed that the present algorithm over segments certain text lines as it fails to deal with abrupt misalignment of consecutive words and sudden occurrences of larger word spacings at the failure points. The algorithm also under segments certain text lines in the document by extracting more than one consecutive text line as a single one. In this particular case, the failure has occurred as the hypothetical water flows are obstructed in the line spacings against large slashes and words written in the line spacing.

Fig. 21 shows the image of a sample document of handwritten Bengali text with occurrences of some English words. It is noteworthy that failure cases for this document consist of only under segmentations. Under segmented portions of the document image are all encircled in Fig. 21. From closer observations of the document image, it can be analyzed that under segmentations have occurred mainly for two reasons. *Firstly*, it is because of the touching characters appearing in consecutive lines. *Secondly*, certain spurious inclined lines have appeared in the document images due to folding marks in the original documents. These lines passing over more than one consecutive text lines are also

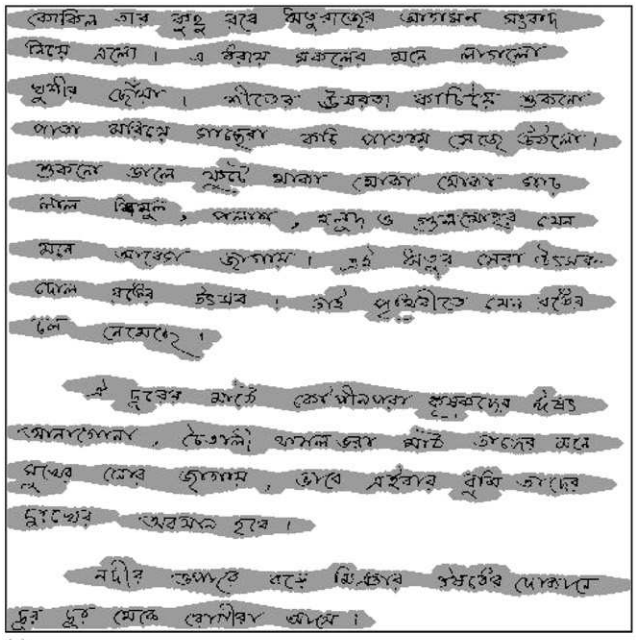


(a)

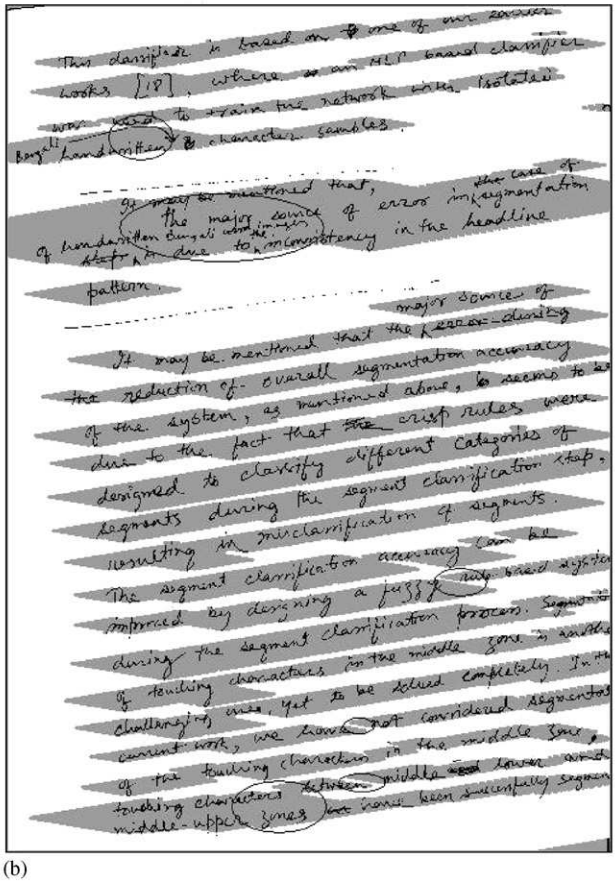


(b)

Fig. 18. The encircled portions in the two sample images of Data Set #2 show where the line extraction technique fails with $\theta = 14°$ and $k = 4$: (a) a sample image of handwritten Bengali text; (b) a sample image of handwritten English text.



(a)



(b)

Fig. 19. Much of the encircled portions, i.e., failure points, shown in Fig. 8(a)–(b) are removed by tuning of values of $\theta$ and $k$: (a) $\theta = 9.5°$ and $k = 5$; (b) $\theta = 9.5°$ and $k = 2$.

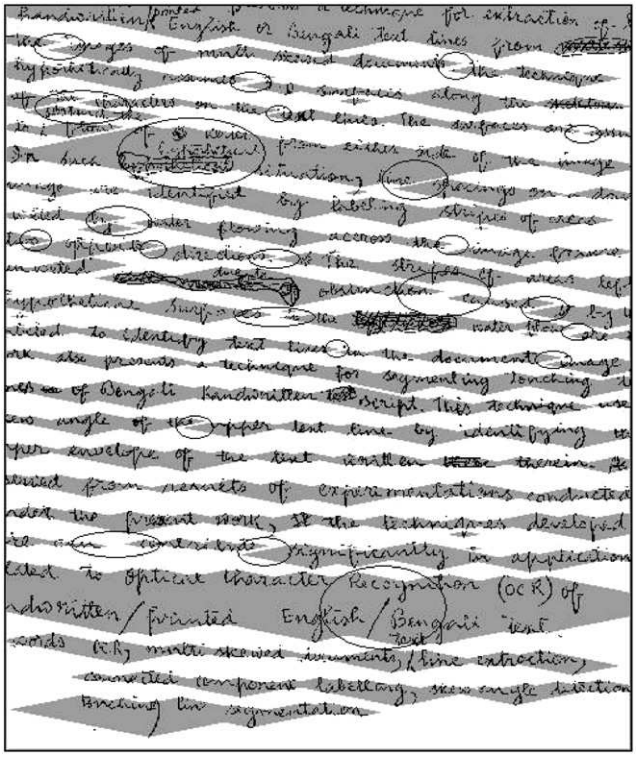


Fig. 20. A sample image of multioriented handwritten English document with scratches ($\theta = 11.3°, k = 2$).

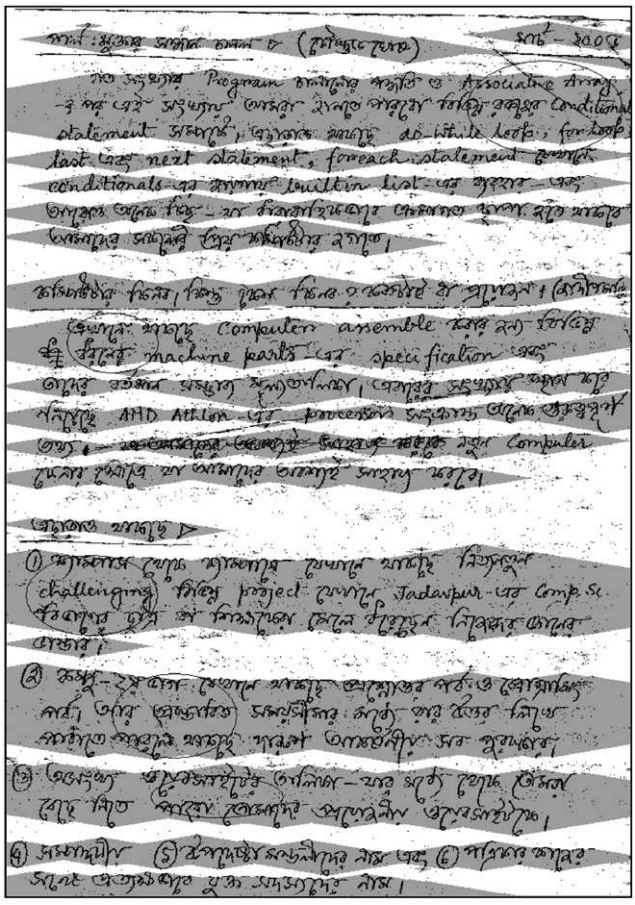


Fig. 21. A sample image of handwritten Bengali document with failure cases of only under segmentation ($\theta = 9.5°, k = 1$).

shown encircled in Fig. 21. Hypothetical water flows under the present algorithm are obstructed in the line spacings by these spurious lines resulting into under segmentation of the text lines in the document.

Compared to the performance of the present technique on handwritten document images, the technique described in Ref. [10] has shown 96% success rate on 183 text lines of handwritten English documents collected from NIST database SD19. The dataset chosen for the latter work is nearly 2.49 times smaller than that chosen for the present work. Moreover, unlike the images of sample data, shown in Ref. [10], some of the sample document images, considered for the present work, contain scratches, numbers, handwritten diagrams, touching lines and mixtures of English/Bengali printed/handwritten scripts as shown in Figs. 17–21. All these are quite natural in handwritten documents. In this respect, all the experimental observations made under the present work are more exhaustive compared to the work described in Ref. [10].

The performance of the work employing a production system for text line segmentation in handwritten textual documents of English script [11] cannot be compared to that of the present work as the former has been applied only on few 'toy problems'. Moreover the work involving a production system, in its present form, is reported to have allowed to segment only well separated text lines.

As observed from experimentation, the present line extraction technique works satisfactorily both on samples of printed and handwritten document images of English and Bengali multi-skewed texts with proper choices of values for $\theta$ and $k$. The technique will be useful for OCR related applications, especially involving handwritten English and Bengali text. None of the samples used here for testing the performances of the present technique contains lines with skew angles higher than 45°. The samples are featured with moderately wide word and line spacings. Such samples limit the application of the present technique. Truly speaking, the technique specifically targets documents of handwritten English and Bengali texts without certain specifically designed skewed lines.

As an extension of the present work, an attempt may be made to choose the values of $\theta$ and $k$ automatically prior to application of the line extraction technique on handwritten Bengali documents. Choice of a proper value for $\theta$ depends on the average skew angle of the target documents. In doing so, a sample document image may initially be segmented into candidate text lines with the choice of values of flow angle 14°. From the upper envelopes of all these candidate text lines, the average skew angle and the standard deviation may be computed. If the standard deviation exceeds certain predetermined threshold, the value of the flow angle

is to be chosen as the nearest possible one of the average skew angle. Otherwise, the magnitude of the flow angle is to be chosen as that of the average skew angle with imposition of the following restriction on water flows. For left to rightward flow from each pixel point, the water flow with a negative flow angle is to be suppressed and vice versa. This is to avoid the water flow in inter word gaps of a text document, in which all text lines are more or less uniformly skewed.

Once a value for $\theta$ is suitably chosen, the value of $k$ is to be chosen. Considering the resolution of the scanner and the normalized pixel sizes of the document images used for the present work, a suitable value of $k$ is to be chosen from a set of possible values ranging from 1 to 5. To automate the choice of a suitable value for $k$, the present line extraction algorithm is to be extended on a sample of target documents for each of five possible values of $k$ in the said range. The higher is the value of $k$, the more is the degree of erosion of the 'dark stripes' and the lesser is the number of segments finally produced. The average number of segments is to be computed from all the five runs of the algorithm. The value of $k$, which produces the number of segments closest to the average of the five runs, may be chosen finally for the work.

The technique presented here for segmentation of touching text lines of Bengali script requires to compute the skew angle of the top most touching line. The skew angle detection technique developed for this purpose is applicable on lines of Bengali text only. Further attempts are needed to extend the technique for lines of English text also. Such attempts require to consider the *lower envelope* of lower most touching line of English text. How best two touching text lines can be segmented to ensure minimal character loss is also another topic of future research in relation to this work. Horizontal histograms of pixel counts of different samples of touching text lines of a particular script need to be investigated to explore the possibility of framing general rules to find an optimal cutting point on such a histogram and also an optimal cutting angle for the touching text lines.

## Acknowledgments

The authors are thankful to the CMATER and the SRUVM project, C.S.E. Department, Jadavpur University, for providing necessary infrastructural facilities during the progress of the work.

## References

[1] O. Okun, M. Pietiekain, J. Sauvalo, Robust document skew detection based on line extraction, in: Proceedings of the 11th Scandinavian Conference on Image Analysis, 1999, pp. 457–464.
[2] O. Okun, M. Pietiekain, J. Sauvalo, Large-scale experiments with skew detection techniques, in: Proceedings of the Fifth Conference on Pattern Recognition and Information Processing, 1999, pp. 99–104.
[3] D.X. Le, G. Thoma, Document skew angle detection algorithm, in: Proceedings of 1993 SPIE Symposium on Aerospace and Remote Sensing-Visual Information Processing II, vol. 1961, 1993, pp. 251–262.
[4] P. Slavik, V. Govindaraju, Equivalence of different methods for slant and skew corrections in word recognition applications, IEEE Trans. Pattern Anal. Mach. Intell. 23 (3) (2001) 323–326.
[5] S. Madhavanath, G. Kim, V. Govindaraju, Chaincode contour processing for handwritten word recognition, IEEE Trans. Pattern Anal. Mach. Intell. 21 (9) (1999) 928–932.
[6] R.M. Bozinovic, S.N. Srihari, Off-line cursive script word recognition, IEEE Trans. Pattern Anal. Mach. Intell. 11 (1989) 68–83.
[7] A.W. Senior, A.J. Robinson, An off-line cursive handwriting recognition system, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 309–321.
[8] B.B. Chaudhuri, U. Pal, Skew angle detection of digitized Indian script documents, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 182–186.
[9] U. Pal, M. Mitra, B.B. Chaudhuri, Multi-skew detection of Indian script documents, in: Proceedings of Sixth International Conference on Document Analysis and Recognition, 2001, pp. 292–296.
[10] C. Weliwitage, A.L. Harvey, A.B. Jennings, Handwritten document offline text line segmentation, in: Proceedings of Digital Imaging Computing: Techniques and Applications, 2005, pp. 184–187.
[11] S. Nicolas, T. Paquet, L. Heutte, Text line segmentation in handwritten document using a production system, in: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition, 2004, pp. 245–250.
[12] K. Kise, O. Yanagida, S. Takamatsu, Page segmentation based on thinning of background, in: Proceeding of ICPR, 1996, pp. 788–792.
[13] U. Pal, P.P. Roy, Text line extraction from Indian documents, in: Proceedings of the Fifth International Conference on Advances in Pattern Recognition, 2003, pp. 275–279.
[14] U. Pal, P.P. Roy, Multioriented and curved text lines extraction from Indian documents, IEEE Trans. Syst. Man Cybern. Part B 34 (4) (2004) 1676–1684.
[15] H. Goto, H. Aso, Extracting curved lines using local linearity of the text lines, Int. J. Doc. Anal. Recognition 2 (1999) 111–118.
[16] R.C. Gonzalez, R.E. Woods, Digital Image Processing, first ed., Prentice-Hall, India, 1992 pp. 40–43.
[17] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis and Machine Vision, second ed., Brooks/Cole Publishing Company, 1999 pp. 565–567.
[18] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, D.K. Basu, Skew angle correction and line extraction from unconstrained handwritten Bengali text, in: Proceeding of the Fifth International Conference on Advances in Pattern Recognition, 2003, pp. 271–274.

**About the Author**—SUBHADIP BASU received his B.E. degree in Computer Science and Engineering from Kuvempu University, Karnataka, India, in 1999. He received his Ph.D. (Eng.) degree thereafter from Jadavpur University (J.U.) in 2006. He joined J.U. as a lecturer in 2006. His areas of current research interest are OCR of handwritten text, gesture recognition, real-time image processing.

**About the Author**—CHITRITA CHAUDHURI received her B.E.Te1.E. and M.E.Te1.E. degrees from Jadavpur University, in 1980 and 1982, respectively. She joined J.U. as a lecturer in 2001 and is currently working there as a Reader. Her areas of current research interest are pattern recognition, image processing, multimedia techniques, and data mining.

**About the Author**—MAHANTAPAS KUNDU received his B.E.E, M.E.Tel.E and Ph.D. (Eng.) degrees from Jadavpur University, in 1983, 1985 and 1995, respectively. Prof. Kundu has been a faculty member of J.U. since 1988. His areas of current research interest include pattern recognition, image processing, multimedia database, and artificial intelligence.

**About the Author**—MITA NASIPURI received her B.E.Tel.E., M.E.Tel.E., and Ph.D. (Eng.) degrees from Jadavpur University, in 1979, 1981 and 1990, respectively. Prof. Nasipuri has been a faculty member of J.U. since 1987. Her current research interest includes image processing, pattern recognition, and multimedia systems. She is a senior member of the IEEE, U.S.A., Fellow of I.E. (India) and W.B.A.S.T., Kolkata, India.

**About the Author**—DIPAK KUMAR BASU received his B.E.Tel.E., M.E.Tel., and Ph.D. (Eng.) degrees from Jadavpur University, in 1964, 1966 and 1969, respectively. Prof. Basu has been a faculty member of J.U. since 1968. His current fields of research interest include pattern recognition, image processing, and multimedia systems. He is a senior member of the IEEE, U.S.A., Fellow of I.E. (India) and W.B.A.S.T., Kolkata, India and a former Fellow, Alexander von Humboldt Foundation, Germany.