Ram Sarkar*, Nibaran Das, Subhadip Basu, Mahantapas Kundu and Mita Nasipuri

Extraction of Text Lines from Handwritten Documents Using Piecewise Water Flow Technique

Abstract: A novel piecewise water flow technique for text line extraction from multi-skewed document images of handwritten text of different scripts is presented here. The basic water flow technique assumes that the hypothetical water flows from both left and right sides of the image frame. This flow of water fills up the gaps between consecutive objects (texts) but faces obstruction if any object lies in the path of the flow. All unwetted regions in the document image are then labeled distinctly to extract the text lines. However, the technique fails when two neighboring text lines touch each other, as water gets obstructed by the touching segment(s). To get rid of this difficulty, we have modified the basic water flow technique by iteratively applying the same over the vertically segmented document images. The main purpose of this vertical segmentation is to localize the text line segment(s) where two text lines get joined. These segments are then horizontally fragmented, and each fragment is placed suitably to the text line in which it actually belongs to. This way, the probable data loss during isolation of the touching text line segment is minimized. Both the techniques (current and basic ones) have been tested on three different databases, viz., *CMATERdb*1.1.1, *CMATERdb*1.1.2, and ICDAR2009 handwritten segmentation contest pages, respectively. The test results show that the present technique outperforms the basic one for all three databases.

Keywords: Text line extraction, piecewise water flow technique, touching text line, handwritten multi-script document, optical character recognition.

***Corresponding author: Ram Sarkar,** Department of Computer Science and Engineering, 188, Raja S.C. Mallick Road, Jadavpur University, Kolkata 700032, West Bengal, India, e-mail: raamsarkar@gmail.com
Nibaran Das, Subhadip Basu, Mahantapas Kundu and Mita Nasipuri: Computer Science and Engineering Department, Jadavpur University, Kolkata, India

1 Introduction

Extraction of handwritten text lines from digitized document pages is one of the major challenges of any optical character recognition (OCR) system. In an unconstrained handwritten document, text lines are usually skewed with different angles of inclination. Even for a single handwritten text line, skewness may vary from one part of the text line to another. All these make extraction of such text lines from handwritten documents a difficult task. The problem becomes compounded if any two neighboring text lines touch each other. These text lines are called multiskewed touching text lines.

The problem of text line extraction from optically scanned document images is trivial in case of unskewed text lines, i.e., the text lines oriented parallel with the horizontal edges of the text pages. Such text lines can be easily extracted from the document image by identifying the valleys of horizontal pixel density histograms as shown in Figure 1A. However, for most practical situations where text lines are complexly skewed and touching each other in successive text lines, this technique fails. One such document image, with the corresponding horizontal pixel density histogram, is shown in Figure 1B.

2 Previous Work

However, researches on extraction of unconstrained handwritten text lines from digitized document pages are limited in the literature [1–3, 5–7, 9, 11]. In one of our earlier works [1], we had reviewed contemporary

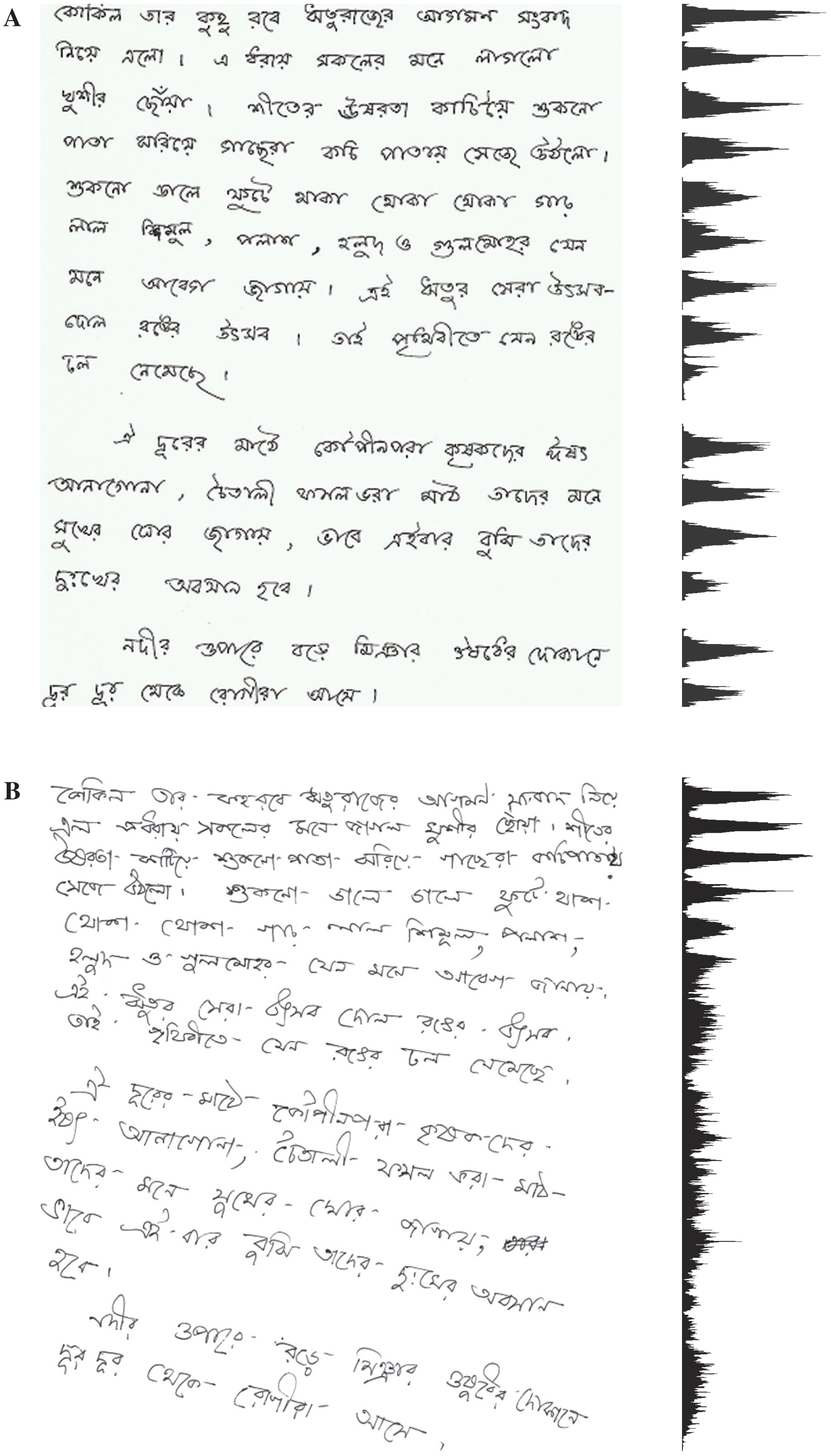

Figure 1. Horizontal Pixel Density Histograms of Handwritten Document Images Containing (A) Unskewed and (B) Multi-Oriented Text Lines.

research contributions related to extraction of handwritten/printed text lines from digitized document pages. In the following section, some more recent developments in this area of research are discussed along with the motivations behind the current research.

Current research contributions related to the extraction of unconstrained handwritten text lines from digitized document pages may be classified into several categories, viz., Hough transformation-based techniques [3, 6, 7], statistical approaches using minimal spanning tree (MST), probability distribution function, etc. [2, 5, 11, 12], and typical morphological approaches using run length encoding, water flow technique, etc. [1, 9]. The work presented in this article falls in the third category of solutions.

Among the first category of solutions, the technique described in [6] is based on three steps, viz., preprocessing, Hough transform, and postprocessing. The preprocessing step partitions the connected components (CCs) of the handwritten digitized document page into three subsets. Subset 1 consists of the components of the majority of characters. Subset 2 contains all large CCs, and Subset 3 contains small characters, punctuation marks, etc. In the second step, Hough transform takes into consideration a subdomain of the CCs of the image. Finally, in postprocessing, a merging technique is applied to rectify some false acceptance cases (i.e., separation of touching text lines). In [7], the handwritten text lines are extracted using three steps. In the first step, image binarization, enhancement, CC extraction, etc. are done. In the second step, a block-based Hough transformation technique is used for detection of text lines, and finally, the text lines that are not separated in the previous step are identified. In [8], text line segmentation is achieved by applying Hough transform on a subset of the document image CCs, and a skeletonization technique is used to separate the vertically CCs that were not separated by Hough transform. After segmenting the text lines, word segmentation is done hierarchically in a Gaussian mixture modeling framework. The methodology uses suitable performance measures to compare the text line and word segmentation results against the corresponding ground truth annotation.

Among the statistical approaches, MST-based clustering technique [11] with distance metric learning, is used for text line segmentation purpose of Chinese documents. In this technique, each text line is viewed as a cluster of stroke pixels or CCs. The MST algorithm then clusters the CCs into text lines based on the distance between two neighboring components in the same text line. The distance metric between CCs is evaluated by supervised learning. The MST is generated using the distance metric having the characteristic that the neighboring components of the same text line are connected and each text line corresponds to a subtree. Another text line segmentation technique for handwritten documents is described in [2] using Mumford–Shah (MS) model. Here, text line segmentation is achieved by minimizing the MS energy. Different morphing techniques are also used to remove the overlaps between neighboring text lines, and to connect the broken text lines. The method uses level set representation to morph the segmented regions. However, as the MS energy function is not convex, the minimization result depends heavily on the initial conditions. In [5], density estimation and level-set methods are used for the extraction of handwritten text lines from digitized document pages. Because the distribution of black pixels in any document image is not uniform, they have estimated a probability map where each pixel represents the probability of its belongingness to a text line. As the text lines are assumed to have horizontally elongated shapes, they have used region-growing technique to determine the text line boundary. To extract the text lines from a handwritten Chinese document, MST clustering with distance metric learning is used in [12]. Given a distance metric, the CCs are grouped into a tree structure, from which text lines are extracted by dynamically cutting the edges using a new hypervolume reduction criterion and a straightness measure. Supervised learning technique is used here.

The technique used in [9] is based on morphological operations and run length smoothing algorithm (RLSA). Here, RLSA has been applied to get individual word as a component. Then, the foreground portion of the image has been eroded to get some seed components. Using positional information of the seed components and boundary information, the text lines have been segmented.

In one of our earlier works [1], we developed a novel technique for the segmentation of multi-oriented handwritten text lines using hypothetical water flow technique. Here, water flows at a specific flow angle from both sides of the document image. The technique effectively separates unconstrained handwritten text lines of Roman or Bangla script. A brief discussion of the technique is given in the next subsection.

2.1 Basic Water Flow Technique

In an unconstrained handwritten document, all text lines are separated from each other with uniform or nonuniform spacing depending on the nature of skewness of the text lines. The skewness of the text lines, in such documents, varies not only from one text line to another but also along the horizontal stretches of the text lines. To extract such text lines, all text line spacings in the document are to be labeled first. Each of the unlabeled stripes of the text, left in the document image, is then to be labeled distinctly to identify different text lines in the document.

For labeling all text line spacing, a hypothetical flow of water in a particular direction across the image frame is considered in our earlier work [1]. Figure 2 illustrates the basic water flow technique. In this figure, a starting pixel shows how the water spreads at a particular flow angle from left to right direction. In doing so, it fills up the gaps between consecutive objects but faces obstruction if any object lies in the path of the flow. In this hypothetical situation, water flowing across the image frame does not wet those obstructed areas. The flows from two opposite directions across the image frame result in a number of common wetted stripes as shown in Figure 3. These wetted stripes generally correspond to the spacings between consecutive text lines if the technique is applied to a text page. To ensure precise wetting of all text line spacings in a document image, a parameter called flow angle is to be controlled depending on the nature of the skewness of the text

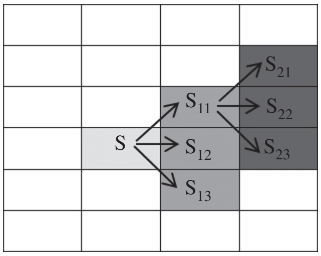

Figure 2. Pictorial Illustration of Water Flow Technique.
Let Flow Start at Pixel S, then in the First Iteration, S_{11}, S_{12}, and S_{13} are the Wetted Pixels; Similarly, in the Next Iteration, if S_{11} is the Starting Point, then Pixels S_{21}, S_{22}, and S_{23} are Wetted, and the Technique Proceeds in this Way.

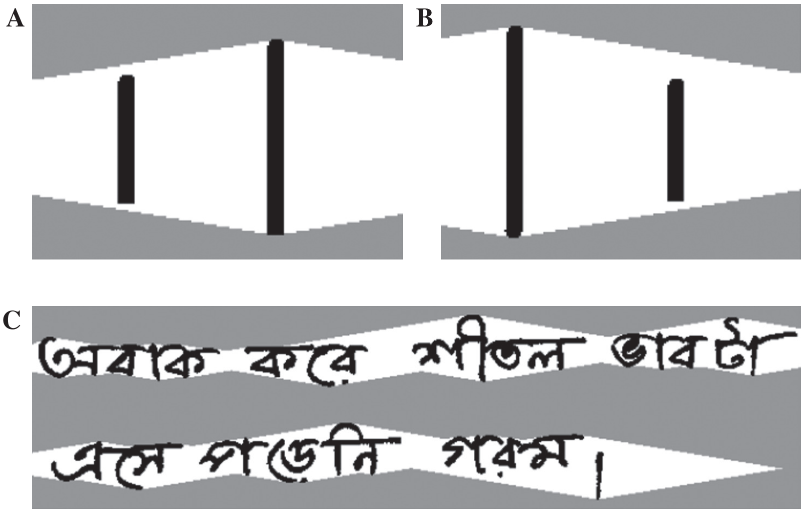

Figure 3. Common Wetted Area of Any Image Frame Under Hypothetical Bidirectional (Right-to-Left and Left-to-Right) Water Flow Technique.
Gray Shades Signify Wetted Regions. Examples of Water Flow Scenarios When (A and B) Two Vertical Bars of Unequal Heights are Placed in Different Sides and (C) Two Text Line Segments, Written Bangla Script, are Exposed to the Technique.

lines in the document image. The flow angle is assumed to be at least greater than the maximum of the skew angles of the text lines in a document of interest.

Identification of wetted stripes in the document image is not sufficient for extraction of text lines from the same. All unwetted stripes in the document image are to be labeled distinctly before the text line extraction. Connected component labeling algorithm [3] is applied for this purpose.

2.2 Motivation Behind the Current Work

The main limitation of our earlier technique [1] was its inability to separate touching text lines in a convincing way. This is so because hypothetical water flows from either side of the document pages are obstructed by the touching part(s) of any two neighboring text lines. A possible solution, as proposed in our earlier work, is to extract the touching text lines together and subsequently segment them using pixels density histograms along the skew angle of the topmost touching text line. However, in case of multiskewed touching text lines, the technique often fails to identify the ideal segmentation line using pixel density histogram. In addition, the performance of the overall technique was heavily dependent on the choice of the flow angle. To overcome these limitations, we are motivated to develop a novel piecewise water flow technique as discussed in the subsequent sections.

3 Present Work

For extraction of text lines, in the current work, the basic water flow technique [1] is first used over the complete document image. As the document gets segmented into number of wetted and unwetted regions, the average heights of both such regions are estimated. The touching text line segmentation threshold (*Th*) is then estimated from the calculated average heights of the wetted and unwetted regions. At this point, on vertical scanning of the whole document image, if any unwetted stripe/line segment height (*L*) is found to be greater than *Th*, it can be inferred that the said document contains at least one touching text line segment. Otherwise, the text lines in the document are considered to be extracted perfectly. In the former case, i.e., when the document contains at least one touching text line segment, the current piecewise water flow technique is used for text line extraction. This is done by first segmenting the document image into a number of vertical partitions and then using the water flow in each such partition/document sub-image.

For this, in each such document sub-image, if the hypothetical water flows identify any potential touching text line segments, then such segments are analyzed and split into their component characters and associated to the text lines they actually belong. The overall document image is then further partitioned vertically into more number of sub-images to validate the effectiveness of the touching text line segmentation technique used in the earlier step. This validation process is required to avoid any possible under-segmentation of the touching text lines, especially along the sub-image boundaries (vertical). This image-partitioning scheme continues until no such touching text line segment is found in any document sub-image.

The details of the image partitioning scheme, estimation of touching text line segmentation threshold, separation of potential touching text line segment, and validation of overall segmentation decisions are discussed in the following subsections. The basic steps involved in the piecewise water flow technique are described in Algorithm 1.

Algorithm 1:

1. Input binarized document images.
2. Apply water flow technique on whole document without any partitioning.
3. Calculate average text line height and average text line spacing to design a threshold (*Th*) to identify touching text lines.
4. If any touching text line segment is observed, partition the document image vertically into halves of equal widths.

5. Apply the water flow technique on each partition separately.
6. Re-evaluate *Th* to localize and separate the touching text line segments.
7. If touching text line segment is present in any of the partitions, then increment the number of partitions by one and repeat steps 5–7 until all the text lines are extracted properly or number of partitions reaches to a predefined threshold.
8. Exit.

3.1 Image Partitioning Scheme

If all the text lines in any document page are not extracted by the hypothetical water flow on the complete document at the beginning, the document page is divided into two vertical stripes of equal width. This partitioning scheme is applied on the document image to localize the touching text line segments in an effective way. In each such vertical stripe, water is hypothetically flown from two opposite directions. This method divides each vertical image partition into a number of wetted and unwetted regions. This is also illustrated in Figure 4.

In many cases, even this initial partitioning structure fails to separate all the text lines within the document image. This is mainly due to the presence of touching text lines therein. In such cases, a document image is needed to partition into more number of vertical stripes. The piecewise water flow technique is repeated for the new partition structure with number of partition one more than the earlier one. This process of partitioning the document image into more number of vertical stripes is repeated until all the text lines are effectively extracted from the document image or the number of vertical stripes reached a certain limit. In any given partitioning structure, the text line extraction methodology primarily depends on identification of touching text line segments in the document image.

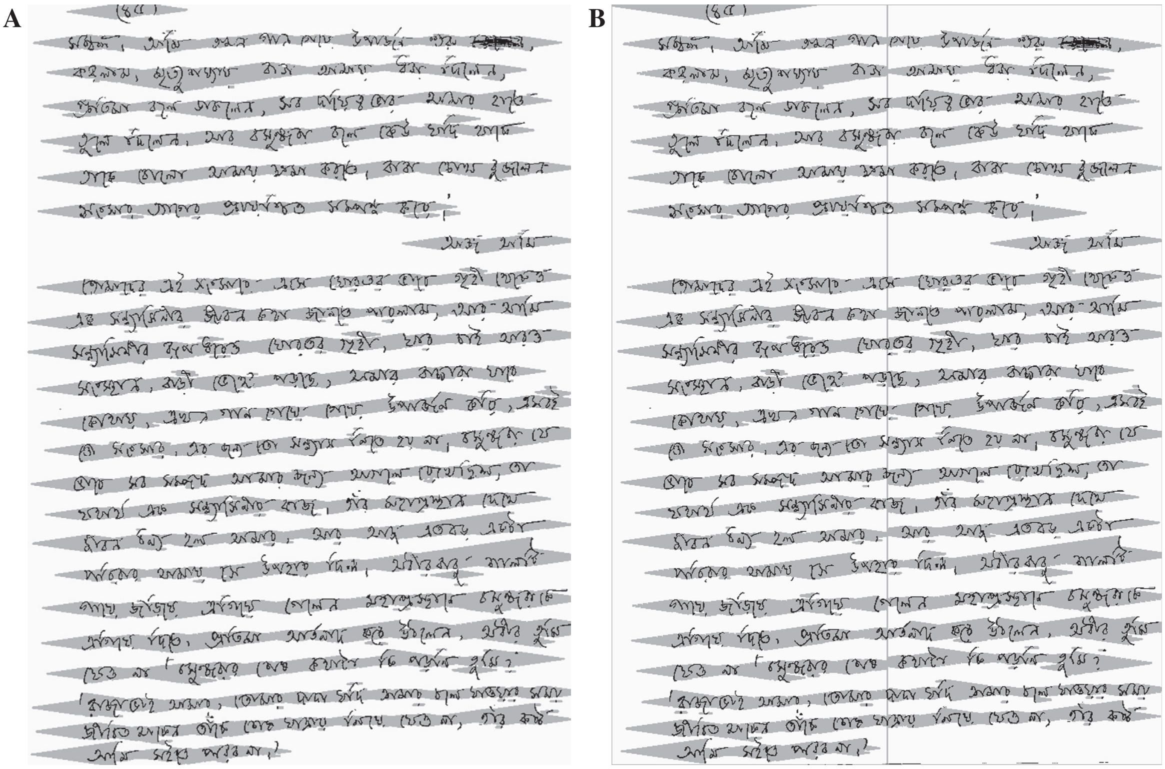

Figure 4. Hypothetical Water Flow in a Document Image.
(A) Water Flow in the Whole Document. (B) Water Flow in Individual Vertical Partitions.

3.2 Estimation of Touching Text Line Segmentation Threshold

The piecewise water flow technique, discussed above, generates a number of wetted and unwetted regions in each vertical stripe. In such vertical stripes, unwetted regions signify text line segments and wetted regions represent the background. However, as mentioned earlier, because of the presence of the touching text lines, some of the text line segments in any vertical stripe are having abnormal line heights that need to be identified. To do so, the height of each unwetted text line region (L_{ht}) is calculated. For this purpose, a top-to-bottom vertical scan is performed for every column of the vertical stripes. In each such document, the transition from wetted to unwetted pixel (i.e., background to text line) is considered as the starting row of the text line region, and the transition from unwetted to wetted pixel is considered as the ending row of the text line region. The difference between the two row markers is considered as the height of each text line region along a column in a vertical stripe. The average of these line heights for all the columns in a particular stripe is then estimated.

In many cases, the unwetted text line regions are of very negligible heights at the left and right ends of any text line. At the same time, due to touching text lines, large text line heights may also be possible. These two scenarios, as illustrated in Figure 5, may mislead the estimation of average text line height in a document image.

Therefore, to find the columns, that have significant contribution to the true text line height, the mean (μ_{ht}) and standard deviation (σ_{ht}) of all the text line heights in all column positions are calculated. Text line heights (L_{ht}) within the range $(\mu_{ht} - \delta \star \sigma_{ht}) < L_{ht} < (\mu_{ht} + \delta \star \sigma_{ht})$ are only considered for the refined average height calculation, where δ is a tuning parameter for accurate estimation of text line height. Using this criteria, the average height of the text line segment ($\mu_{htrefined}$) and the standard deviation ($\sigma_{htrefined}$) are calculated for the overall document image.

However, to estimate the text line segmentation threshold more accurately, the average text line height information is not always enough because, in different document pages, as the shape of the particular character/alphabet or height of the text line may vary (due to writing style etc.), the text line spacing or the gap between two consecutive text lines may also vary at the same time, as illustrated in Figure 6.

Therefore, to identify the potential segmentation point for splitting, the text line spacing information needs to be considered. For this, the text line spacing between the consecutive text lines is estimated using the same technique applied for estimating average text line height. In each such document, the transition from unwetted to wetted (i.e., background to text line) pixel is considered as the starting row of the text line spacing region, and the transition from wetted to unwetted pixel is considered as the ending row of the text line spacing region. The difference between the two row markers is considered as the height of each text line

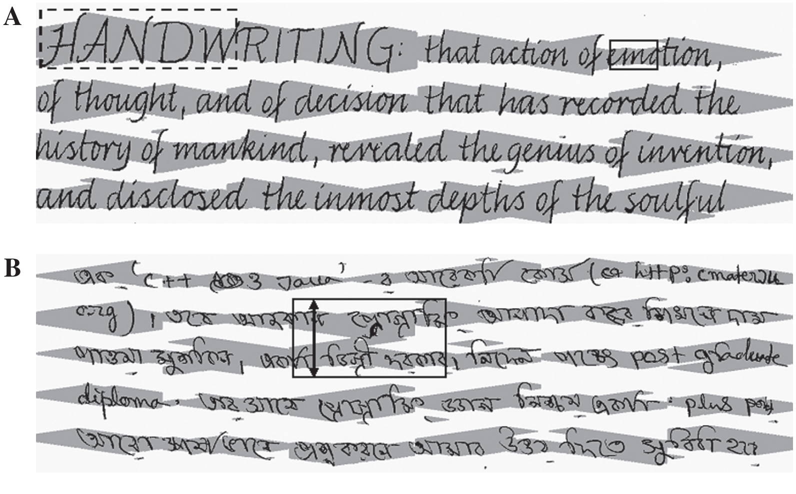
A
HANDWRITING: that action of emotion,
of thought, and of decision that has recorded the
history of mankind, revealed the genius of invention,
and disclosed the inmost depths of the soulful
B

Figure 5. Huge Variations in Text Line Heights May Be Observed After the Water Flow Technique Is Used in the Whole Document.
(A) The Small Rectangle Indicates Lesser Text Line Height (Especially at the Start/End of Flow Regions). (B) The Big Rectangle Indicates Larger Text Line Height.

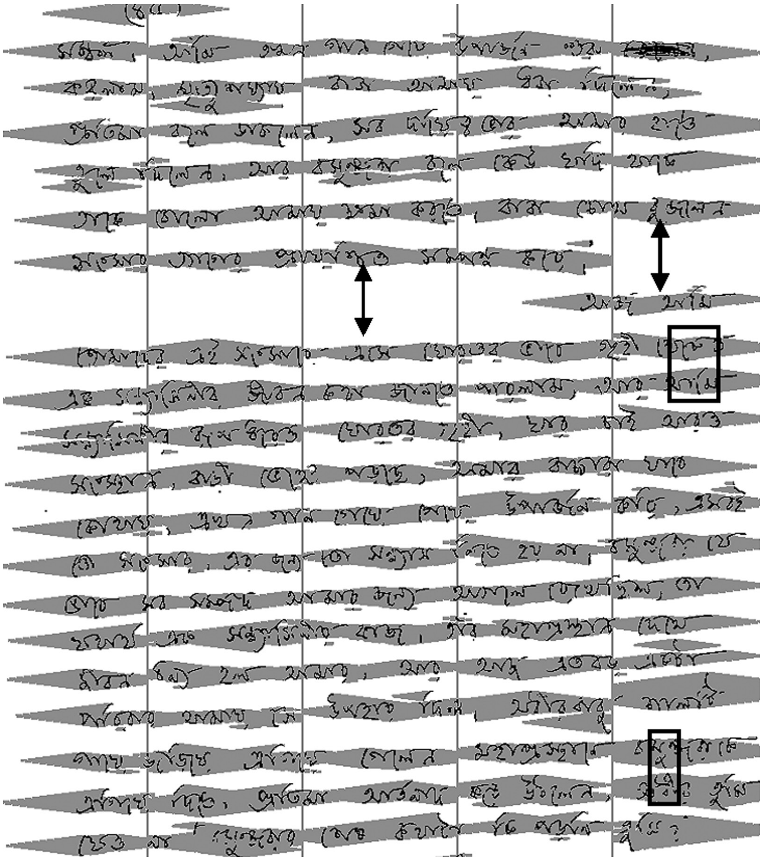

Figure 6. Variations in Text Line Spacing in a Handwritten Document Are Shown. Boxes Indicate Small Gaps and Arrows Indicate Large Gaps.

spacing region along a column in a vertical stripe. The average of these text line spacing values for all the columns in a particular stripe is then estimated.

However, in many cases, there will be no consecutive text line segments in any particular vertical partition at all. This is also illustrated in Figure 6. Due to this type of discontinuity of the appearance of the text line segments, the above-mentioned technique for calculating the average text line spacing estimation sometimes may give erroneous result about the average text line spacing. Therefore, to find the columns that have significant contribution to the true text line spacing, the mean (μ_{ls}) and standard deviation (σ_{ls}) of all the text line spacings in all column positions are calculated. Text line spacings (L_{ls}) within the range $(\mu_{ls} - \delta{\star}\sigma_{ls}) < L_{ls} < (\mu_{ls} + \delta{\star}\sigma_{ls})$ are only considered for the refined average spacing calculation, where δ is a tuning parameter for the accurate estimation of such spacing. Using the above-mentioned criteria, the average spacing of the text line segment ($\mu_{lsrefined}$) and the standard deviation ($\sigma_{lsrefined}$) are calculated for the overall document image.

Th for any document image is computed using refined means and standard deviations of the L_{ht} and L_{ls}, respectively for the said document. This is represented as

$$Th = (2{\star}(\mu_{htrefined} + \sigma_{htrefined}) + (\mu_{lsrefined} + \sigma_{lsrefined})){\star}\eta. \tag{1}$$

where η is a heuristically chosen tuning parameter for effective estimation of the *Th*.

3.3 Separation of Potential Touching Text Line Segments

The height of each unwetted text line region and text line spacing between consecutive text lines are calculated using the above technique. It is said earlier that in any column, if the height of the any such region is found to be greater than an estimated *Th*, then the unwetted text line region is said to contain at least a pair of touching text lines. More specifically, if the L_{ht} of any unwetted text line region along a given column is found to be greater than *Th*, i.e., $Th = (2*(\mu_{htrefined} + \sigma_{htrefined}) + (\mu_{lsrefined} + \sigma_{lsrefined}))*\eta$, then the text line region under consideration is said to contain vertically connected component(s) belonging to at least two successive text lines. These components are to be split along a given row to associate the separated components into the text lines where they belong. To do this, the middle of the start and the end rows of the unwetted segment is identified as the potential segmentation point. To make the separation prominent, one row above and one row below the middle row are also treated as segmentation points to enable smooth flow of water through these points, separating the touching text line segment. This is illustrated in Figure 7.

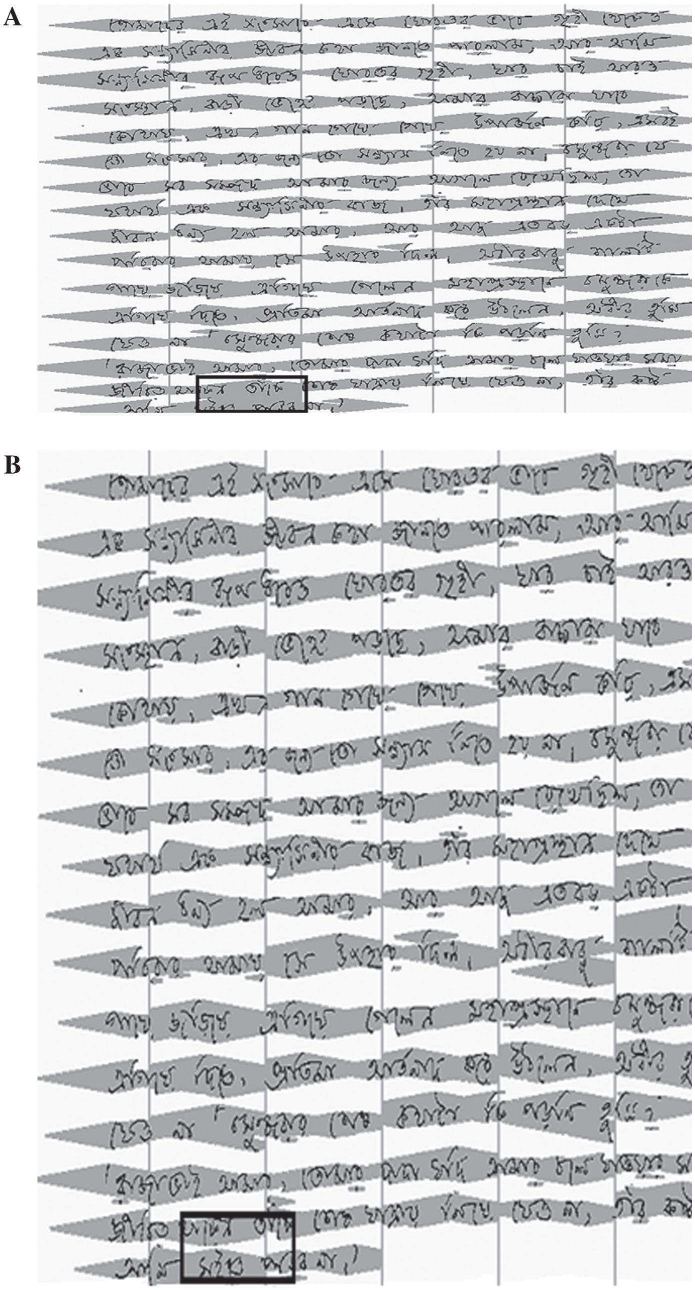

Figure 7. Accurate Localization and Subsequent Separation of Overlapping Text Lines Using the Current Technique Are Shown Within the Highlighted Rectangular Areas.
(A) Successfully Localized Overlapping Text Lines (Marked by Bounding Box). (B) Successfully Separated Text Lines.

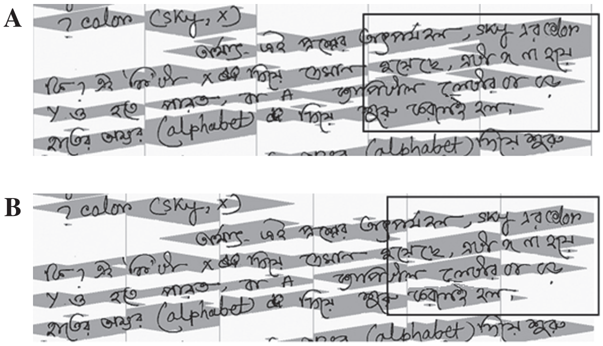

Figure 8. Identified Touching Text Line Segment in a Partition Boundary Shown in (A) Are Separated Almost Accurately in the Following Iteration with Different Partition Structure as Shown in (B).
(A) Water Flow in Five Vertical Partitions Fails to Extract the Text Line Segments (See the Rectangular Box for Easy Reference).
(B) Water Flow in Six Vertical Partitions Almost Extracts the Text Line Segments Correctly (See the Rectangular Box for Easy Reference).

3.4 Validation of Overall Segmentation Decision

To design an effective stopping criterion for the developed technique is a challenging issue. This is because hypothetical water flows in multiple vertical partitions often mislead the text line segmentation routine along the partition boundaries. More specifically, individual vertical partitions may not contain any text line height greater than *Th*, but the text region along the vertical partition boundary may contain such text line segments with $L_{ht} > Th$. To overcome this problem, the text line extraction results for two consecutive partitioning schemes are estimated. For example, if the text lines in a document image are extracted for *n* vertical partitions, the result is validated for $(n + 1)$ vertical partitions as well. If in the said document all the text line heights are less than *Th* for both *n* and $(n + 1)$ vertical partitions, then further partitioning of the document image is stopped. The intersection of the unwetted regions of these two partitioning structures signifies the text line boundaries within the document images.

Figure 8 shows a sample document image with identified touching text line segment along the partition boundary of the vertical stripe. The overlapped partition structure in the successive iterations validates the selection of text line boundaries and justifies the design of the stopping criteria for the iterative partitioning scheme for the current work.

4 Experimental Results

To conduct text line extraction experiments with the piecewise water flow technique described here, various samples of Bangla, Roman, and Bangla mixed with Roman, French, German, and Greek scripts/languages handwritten documents have been used. Three different data sets, namely, *CMATERdb*1.1.1 (Data Set 1), *CMATERdb*1.2.1 (Data Set 2), and ICDAR2009 handwritten segmentation contest's training pages (Data Set 3) [4], are used to evaluate the present technique. *CMATERdb* [10] is a database repository developed in CMATER Lab, Jadavpur University, Kolkata, India. In the current experiment, two different databases from this repository have been used. The first one is the *CMATERdb*1.1.1, which contains 100 digitized handwritten document pages written in Bangla script only. Likewise, *CMATERdb*1.2.1 is a collection of 50 document page images where the textual content is written in both Bangla and Roman scripts. These databases are freely available at http://code.google.com/p/cmaterdb/. Data Set 3 contains document pages written in four different European languages, viz., English, French, German, and Greek.

The performance of the present text line extraction technique, based on piecewise water flow technique, is tested separately with samples of text pages of the three data sets mentioned above. A brief description of these data sets is given in Table 1.

Table 1. Descriptions of the Data Sets Used for the Current Work.

Data set	Language/script used	Number of document pages	Number of text lines
Data Set 1	Bangla	100	2163
Data Set 2	Bangla, Roman mixed scripts	50	1232
Data Set 3	English, French, German, and Greek	100	2249

For the estimation of the success rate of text line extraction technique, two types of errors, viz., under-segmentation error and over-segmentation error are considered. If two or more text lines are identified as a single text line, then it is considered as an under-segmentation error, and both/all the extracted text lines are treated as wrongly extracted text lines. Similarly, if a single text line component is erroneously allocated to two or more text lines, then this text line is also considered as wrongly extracted text line due to over-segmentation. The total number of under- and over-segmented text lines is reflected in the estimation of the success rate (SR_L) of the text line extraction technique. More specifically,

$$SR_L = (T_L - (U_L + O_L))/T_L, \tag{2}$$

where U_L is number of under-segmented text lines, O_L is the number of over-segmented text lines, and T_L is the number of actual text lines present in the document page.

The choices of the flow angle (θ), initial partitioning structure, and the text line height tuning parameters (δ and η) are very crucial for the successful application of the present technique. Strictly speaking, these decisions often require a case-to-case basis consideration of document images.

However, the current technique is designed in such a way that it works in most practical situations. Text line spacing in most of the documents is observed to vary from 2 to 8 mm. In addition, the skew angles of individual text lines in a document image usually range between –14° and 14° within any vertical partition. Considering all these, the value of the flow angle (θ) is chosen as 14° for the current work. The initial number of vertical partitions is chosen as 2 (if the document does not need further partitioning). The values of the tuning parameters δ and η are heuristically chosen as 0.5 and 0.65, respectively. It is also observed from the experiment that the choice of η is particularly significant for the overall performance of the designed system. Figure 9A–C shows a comparison of outputs of the present technique for three different values η on a sample handwritten Bangla document image. It is evident from the figures that with $\eta = 0.65$, all the text lines are extracted properly (see Figure 9B), whereas with $\eta = 0.55$ and $\eta = 0.75$, there are some cases of under-segmentations that are marked with rectangular boxes in panels A and C, respectively, of Figure 9.

It may be worth mentioning in this context that if there is no touching text line, then the process terminates in a single iteration. In the presence of touching text lines, the method terminates after k iterations. Experimentally, it is observed that in most of the cases, the value of k lies within 4–6. However, to limit the iterating procedure, the maximum value of k for the current experiment is considered as 12. This is more or less close to the maximum number of words possible in any text line. Therefore, in the worst case, nearly all the words in any text line are segmented into individual vertical partitions. With these conditions, the document pages are restricted from being over-partitioned unnecessarily by the current methodology. More specifically, it is assumed that if a document page is not segmented with this maximum value, there is no utility to iterate the process even further.

Figure 10A–C shows sample document images from Data Sets 1, 2, and 3, respectively, segmented perfectly using the current technique. These images include multi-oriented handwritten text lines written in a variety of scripts or their combinations. The current technique successfully extracts 86.45% text lines from document images of Data Set 1. This shows a 4.25% improvement over the performance on the same data set by the text line extraction technique reported on the earlier work [1]. In case of Data Set 2 and Data Set 3, text line extraction accuracies of 79.66% and 80.32%, respectively, are observed in comparison

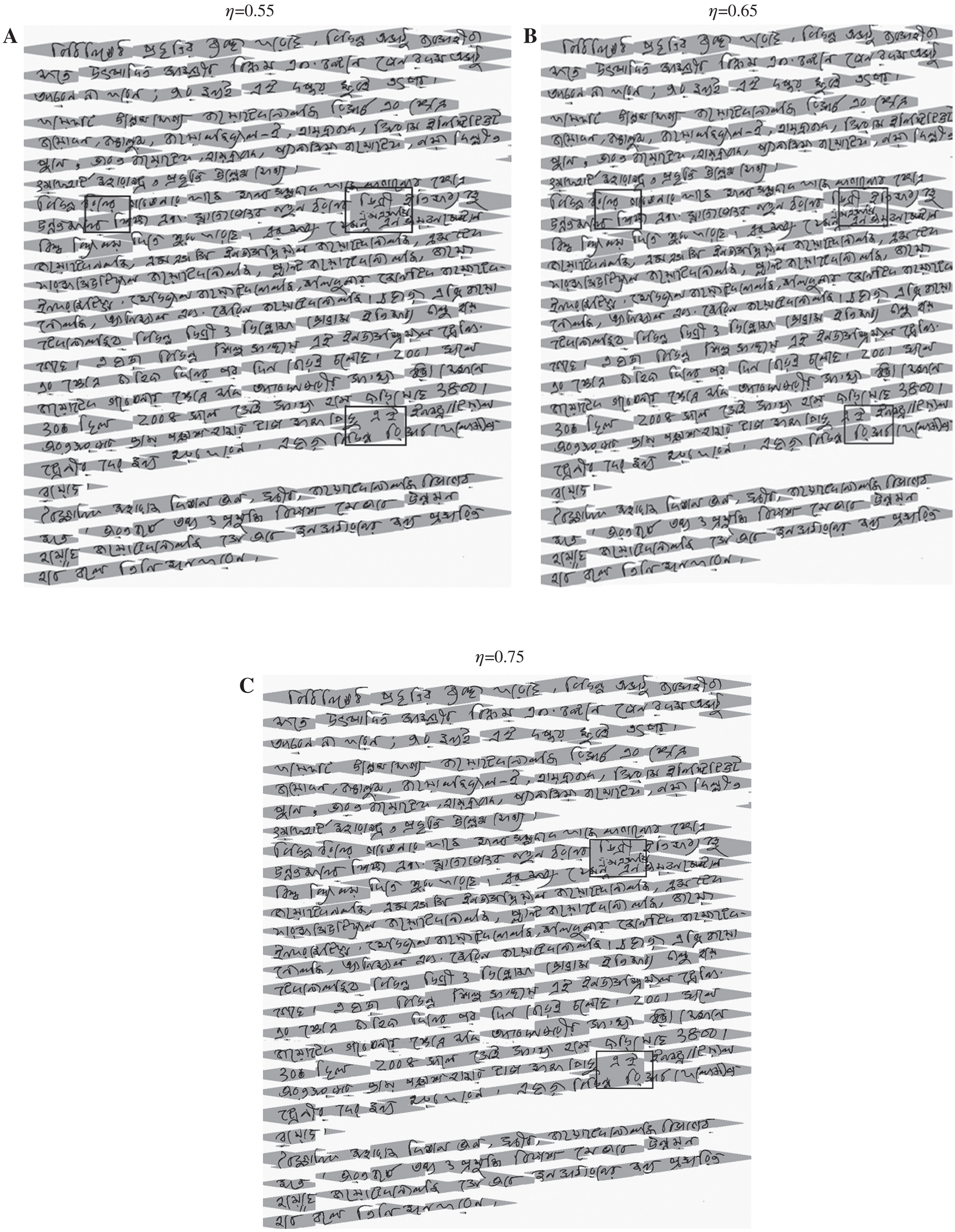

Figure 9. Variation in Output for Different Values of the Text Line Separation Parameter (η). It may be Observed that the Regions Marked in Rectangular Boxes are Better Separated with $\eta = 0.65$.

to 73.32% and 75.68%, respectively, by the earlier technique [1]. A detailed description of the results is also given in Table 2. It may be noted that despite the complexity of many document pages (as illustrated in Figure 10), the current technique successfully extracts individual text lines from such images. Figure 11A

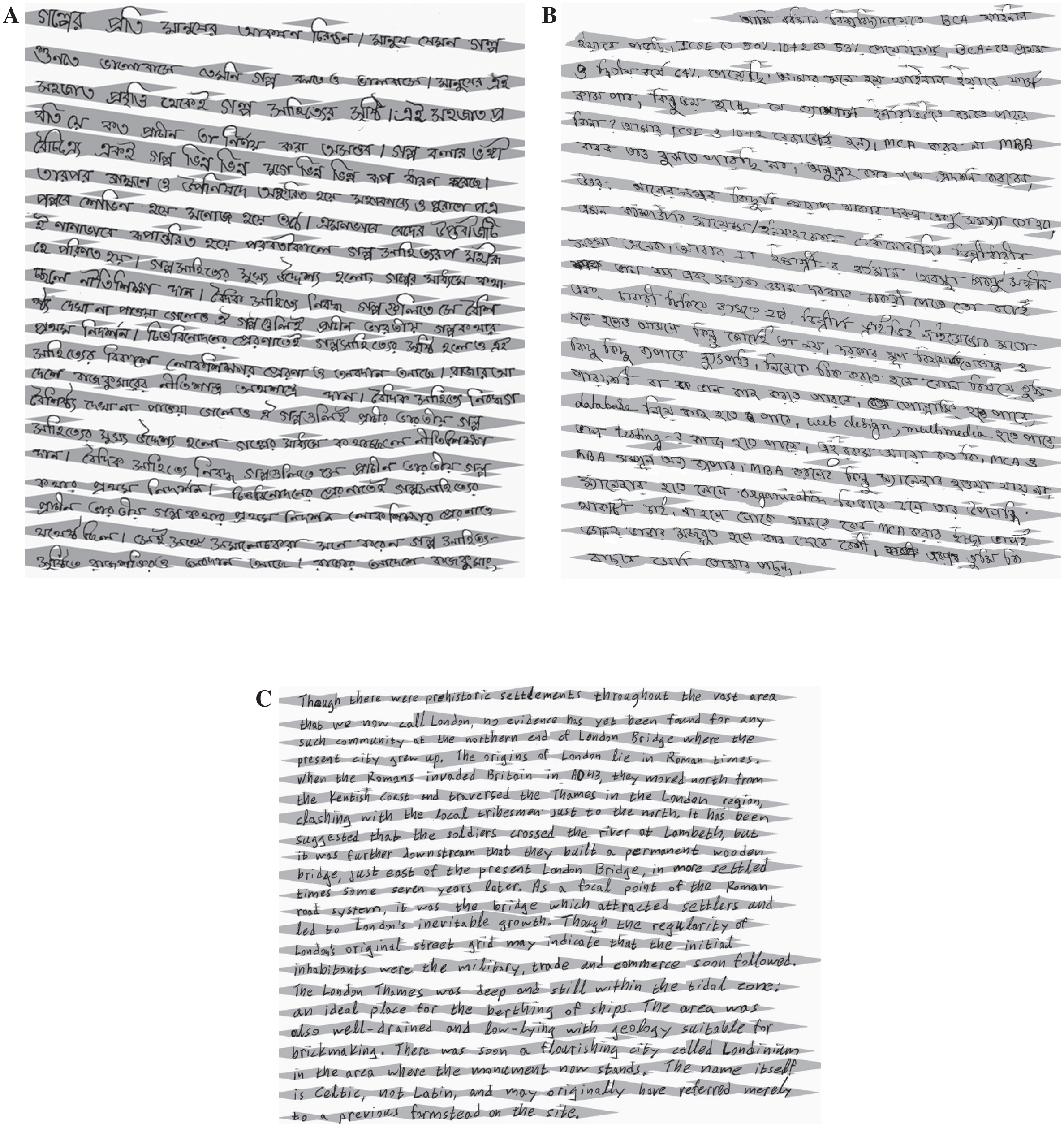

Figure 10. (A–C) Successfully Extracted Text Lines from Different Document Images.

shows a sample document image where the technique [1] fails to extract some of the text lines from the document image, and Figure 11B shows the successful result on the same document using the current methodology.

The present technique, however, fails to extract handwritten text lines in some cases because of typical handwriting styles of individuals. For example, one of the document images of handwritten Bangla text for which the present technique fails to extract certain text lines is shown in Figure 12A. In that image, there are two instances where a pair of text lines is extracted together due to very narrow spacing between them. In the figure, the portions where such failure occurs are marked with rectangular boxes. Certain text lines of the document images, shown in Figure 12B, is inherently spaced apart because of high inter-word spacing. This leads to over-segmentation of text line. This situation may occur when the value of the flow angle is

Table 2. Performance Comparison of the Piecewise Waterflow Technique with Earlier Work [1].

Data set	SR_L (%)	
	Technique described in [1]	Current technique
Data Set 1	82.20	86.45
Data Set 2	73.32	79.66
Data Set 3	75.68	80.32

substantially more than the average skew angle of the text lines of any document. In such cases, user intervention may be required to tune the values of flow angle.

Figure 12C shows the image of a sample document of handwritten Bangla text with occurrences of some Roman words. The document also contains some intersections in the spacing in between two consecutive text lines. Failure cases are all marked in Figure 12C. From closer observation of the document image, it can be analyzed that under-segmentations have occurred mainly for two reasons: (1) complex overlapping/touching characters appearing in consecutive text lines and (2) sharp changes in the text line skewness as well as in inter-line spacing. Multiple text lines are extracted simultaneously by this technique, which is also marked with rectangular boxes in Figure 12C. Hypothetical water flows under the present algorithm are obstructed in the text line spacing by these spurious text lines, resulting in the under-segmentation of the text lines in the document.

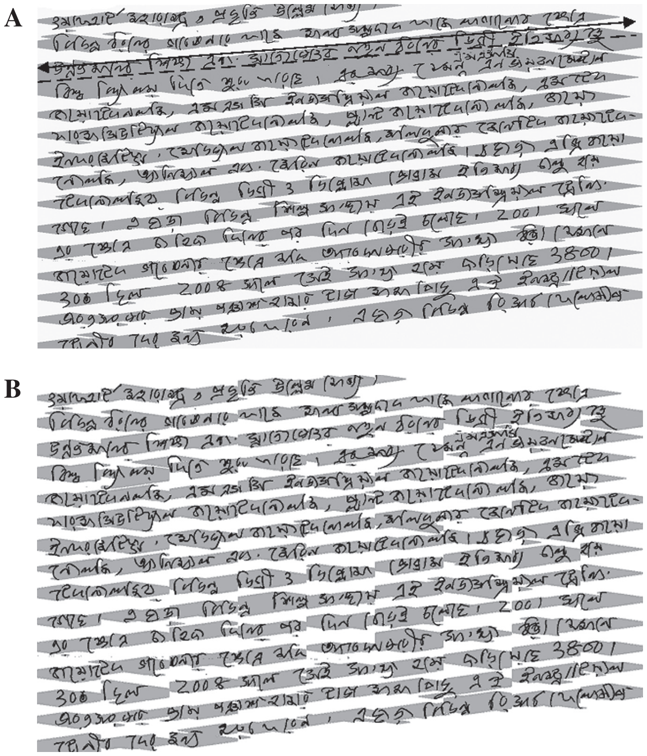

Figure 11. Comparison Between the Basic Water Flow Technique [1] and the Current Technique Shown on a Sample Bangla Document Image.
(A) Using Basic Water Flow Technique [1], with Potential Loss of Information in the Separation of the Touching Regions along the Dotted Line. The Arrow Line Signifies the Orientation of the Text Line Segment. (B) Successfully Separated Text Lines using the Current Technique.

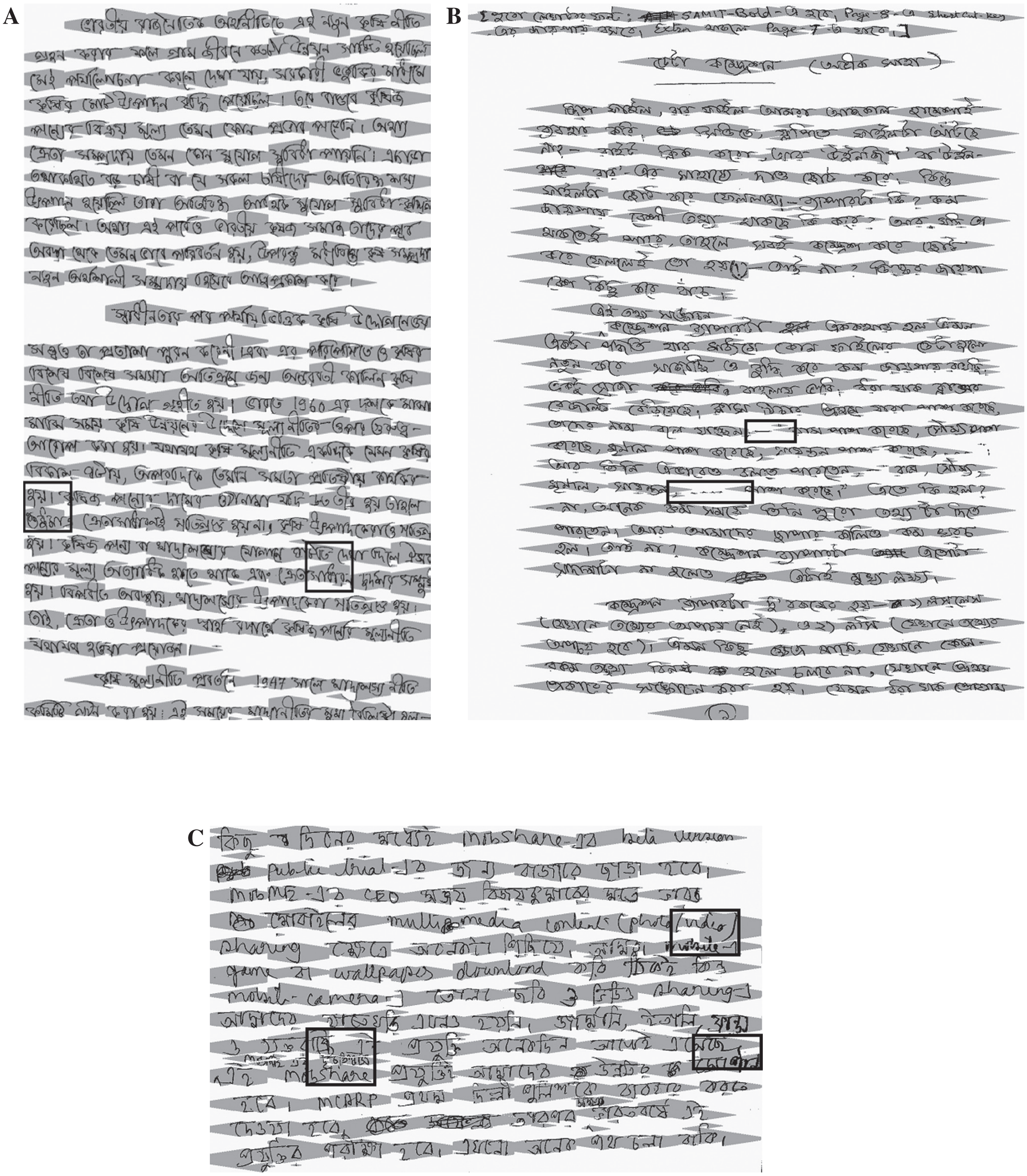

Figure 12. (A–C) Sample Document Images Where the Current Technique Fails to Successfully Separate Text Line from Document Images.
Regions Where Such Error Occurs are Highlighted in Rectangular Boxes.

As observed from experimentation, the present text line extraction technique works satisfactorily on samples of handwritten document images written in different languages/scripts or their combinations. The technique may be useful for OCR-related applications, especially involving multilingual unconstrained handwritten text. None of the samples used here for testing the performances of the present technique contain text lines with skew angles more than ±14°. Truly speaking, the technique specifically targets documents of handwritten texts without certain specifically designed skewed lines. The performance of the developed technique is, however, successfully validated on the standard multilingual data set [10], which is available freely in a public domain.

5 Conclusion

In the current work, we have developed a novel technique for the automatic extraction of handwritten text lines from digitized document pages written in different scripts or their combinations. The technique particularly improves the accuracy of our previously reported work [1] on the extraction of text lines using a water flow technique, even in presence of touching text lines. The performance of the current system is evaluated on three different data sets mentioned earlier.

The novelty of the current technique includes the use of a piecewise water flow technique for effective localization of potential touching text segments in any unconstrained document image. The loss of information, due to the separation of touching text lines, is minimized in this approach by confining the text line of separation only within a vertical partition where the touching text segment(s) is (are) present. Unlike our previous work, the hypothetical water flow angle is kept static throughout the technique. This is done because the variation of skewness of any text line segment in a given vertical partition is much lower than the variation of the overall text line.

The accuracy of the current technique may further be improved by including a connected component analysis module for logical grouping of over-segmented text line segments. The work may further be extended in developing a dynamic fluid flow model to identify curvilinear paths in between successive text lines for the segmentation of document pages consisting of complexly skewed/touching text lines.

Acknowledgments: The authors are thankful to the CMATER, Computer Science and Engineering Department, Jadavpur University, for providing infrastructural facilities during progress of the work. The work reported here has been partially funded by DST, Government of India, PURSE (Promotion of University Research and Scientific Excellence) Programme.

Received March 25, 2013; previously published online October 16, 2013.

Bibliography

[1] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri and D. K. Basu, Text line extraction from multi-skewed handwritten documents, *Pattern Recog.* **40** (2007), 1825–1839.

[2] X. Du, W. Pan and T. D. Bui, Text line segmentation in handwritten documents using Mumford–Shah model, in: *Proceedings of the International Conference in Frontiers in Handwritten Recognition (ICFHR-08)*, August 19–21, 2008, Canada, pp. 253–258, 2008.

[3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 1st ed., Prentice-Hall, India, 1992.

[4] ICDAR 2009 Handwriting Segmentation Contest, call for participation. www.cvc.uab.es/icdar2009/documents/HandSegmCont2009-CfP.pdf.

[5] Y. Li, Y. Zheng and D. Doermann, Script-independent text line segmentation in freestyle handwritten documents, *IEEE Trans. PAMI* **30** (2008), 1313–1329.

[6] G. Louloudis, B. Gatos, I. Pratikakis and C. Halatsis, Line and word segmentation of handwritten documents, in: *Proceedings of the International Conference in Frontiers in Handwritten Recognition (ICFHR-08)*, August 19–21, 2008, Canada, pp. 247–252, 2008.

[7] G. Louloudis, B. Gatos, I. Pratikakis and C. Halatsis, Text line detection in handwritten documents, *Pattern Recog.* **41** (2008), 3758–3772.

[8] G. Louloudis, B. Gatos, I. Pratikakis and C. Halatsis, Text line and word segmentation of handwritten documents, *Pattern Recog.* **42** (2009), 3169–3183.

[9] P. P. Roy, U. Pal and J. Llados, Morphology based handwritten line segmentation using foreground and background information, in: *Proceedings of the International Conference in Frontiers in Handwritten Recognition (ICFHR-08)*, August 19–21, 2008, Canada, pp. 241–246, 2008.

[10] R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri and D. K. Basu, CMATERdb1: a database of unconstrained handwritten Bangla and Bangla–English mixed script document image, *Int. J. Doc. Anal. Recog.* **15** (2012), 71–83.

[11] F. Yin and C. Liu, Handwritten text line segmentation by clustering with distance metric learning, in: *Proceedings of the International Conference in Frontiers in Handwritten Recognition (ICFHR-08)*, August 91–21, 2008, Canada, pp. 229–234, 2008.

[12] F. Yin and C. Liu, Handwritten Chinese text line segmentation by clustering with distance metric learning, *Pattern Recog.* **42** (2009), 3146–3157.