Skew Angle Detection of Digitized Indian Script Documents

B.B. Chaudhuri and U. Pal

Abstract—Skew angle detection of scanned documents containing most popular Indian scripts (*Devnagari* and *Bangla*) is considered. Most characters in these scripts have horizontal lines at the top, called *head lines*. The character head lines mostly join one another in a word and the word appears as a single component. In the proposed method the components are at first labeled. The *upper envelope* of a component is found by columnwise scanning from an imaginary line above the component. Portions of upper envelope satisfying the properties of *digital straight line* are detected. They are clustered as belonging to single text lines. Estimates from individual clusters are combined to get the skew angle. Apart from accuracy and efficiency, an advantage of the method is that character segmentation and zone detection can be readily done from head line information, which is useful in Optical Character Recognition approaches of these scripts.

Index Terms—Document processing, skew detection, optical character recognition (OCR), document structure analysis, digital library.

1 Introduction

DIGITAL document processing is gaining popularity for its application potentials in office and library automation, bank and postal services, publishing houses, data compression and communication technology etc. One important task of document processing is automatic reading of text in the document. To read a complex document containing text, graphics, logo, tables etc. at first it is necessary to distinguish the text regions of the document. Segmentation of text regions can be achieved by document layout analysis [4], [16]. The text regions are then subject to the character recognition algorithm. Machine reading of optically scanned text is usually called optical character recognition (OCR). OCR technology for scripts like Roman is at an advanced stage and some commercial OCR systems with accuracy in the high 90% range are available in the market.

When a document is fed to the optical sensor either mechanically or by a human operator, a few degrees of skew (tilt) is unavoidable. Skew angle is the angle that the text lines in the digital image makes with the horizontal direction. Skew estimation and correction are important preprocessing steps of document layout analysis and OCR approaches.

This paper deals with skew estimation of a class of scripts that includes two major Indian scripts namely *Devnagari* and *Bangla*. Languages like *Hindi*, *Nepali*, *Sanskrit* and *Marathi* are popularly written in *Devnagari* while *Bangla*, *Assamese* and *Manipuri* languages are written in *Bangla* script. Moreover, *Hindi* and *Bangla* are the national languages of India and Bangladesh, respectively. *Hindi* is the secondmost and *Bangla* is the fourthmost popular language in the world. To the best of our knowledge this is the first report on automatic skew angle detection of documents containing major Indian language script forms.

One of the popular skew estimation techniques is based on projection profile of the documents. The horizontal/vertical projection profile [8] is a histogram of the number of black pixels along horizontal/vertical scan lines. For a script with horizontal text lines, the horizontal projection profile will have peaks at text line positions and troughs at positions in between successive text lines. To determine the skew of a document, the projection profile is computed at a number of angles and for each angle, a measure of difference of peak and trough height is made. The maximum difference corresponds to the best alignment with the text line direction which, in turn, determines the skew angle. Baird [2] proposed a modification for quick convergence of this iterative approach. Akiyama and Hagita [1] described an approach where the document is partitioned into vertical strips. The horizontal projection profiles are calculated for each strip and the skew angle is determined from the correlation of the profiles of the neighboring strips. This method is fast but less accurate. Pavlidis and Zhou [12] proposed a method based on vertical projection profile of horizontal strips which works well if the skew angle is small. Another method using cross correlation between the lines at a fixed distance has been proposed by Yan [17]. The projection profile methods are, however, well suited for skew angle less than ±10°.

Techniques based on Hough transform are also popular for skew estimation [7], [9], [15]. Hinds et al. [7] made some modification of Hough transform method to reduce the amount of data to be processed. Le et al. [9] found connected components in a script and considered only bottom pixels of each component for Hough transform which also reduces the amount of data. An improvement of this approach is proposed by Pal and Chaudhuri [11]. One general limitation of Hough transform is that correct identification of peak in Hough space may be difficult for documents containing sparse text.

Fourier transform has been used for the detection of skew angle. Postl [13] proposed an approach where the direction for which the density of Fourier transform is largest gives an estimate of skew angle.

Another class of approaches is based on component nearest neighbor clustering. Hashizume et al. [6] proposed nearest neighbor clustering for skew detection. They found all the connected components in the document and for each component computed the direction of its nearest neighbor. A histogram of the direction angle is computed, the peak of which indicates the document skew angle. O'Gorman [5] generalized the approach in so called "docstrum" analysis. In principle, these approaches are not limited to any range of skew angle.

It should be pointed out that success of some of the above methods depends on the characteristics of the alphabet and text lines of a particular script form. For example, the component nearest neighbor clustering methods [5], [6] will not work well for documents containing *Bangla* and *Devnagari* script. This is so because the characters in a word of these scripts get connected through *head lines* (described in Section 2) and a complete word can be a single component unlike in English (Roman) where usually each character is a separate component. Since word length can vary to a great extent, the nearest neighbor of a word (computed in terms of centroid) may not be another word to its left or right and thus the direction of the neighbor may not indicate the skew angle properly.

As another example, consider the modified Hough transform method due to Le et al. [9]. This technique is also unsuitable for *Bangla* or *Devnagari* script because of the connecting nature of a word. A few bottom pixels of each word will make the data very sparse and the peak of the Hough transform will be quite flattened.

The above examples suggest that it may be possible to find skew angle easily and quickly if some inherent characteristics of the script form is employed. In this article we show that this is

• *The authors are with Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, 203, B.T. Road Calcutta-35 India. E-mail: bbc@isical.ernet.in.*

Manuscript received July 26, 1995; revised Sept. 27, 1996. Recommended for acceptance by M. Mohiuddin.
For information on obtaining reprints of this article, please send e-mail to: transpami@computer.org, and reference IEEECS Log Number P96108.

indeed so for *Bangla* and *Devnagari* script. The idea is to detect the head lines automatically and estimate the skew angles of these head lines. The detection of head lines is necessary for script zone (see Fig. 3 and Section 2) separation and character segmentation, as described in Section 4. So, head line based skew detection and correction helps in saving computation during zone detection and character segmentation, which is another motivation of using this approach.

2 PROPERTIES OF *BANGLA* AND *DEVNAGARI* SCRIPT

All major Indian scripts are mixtures of syllabic and alphabetic scripts. Around 250 BC two different scripts took shape in India. They are Indo-Bactrain or Kharosti script and the Brahmi script. Both *Bangla* and *Devnagari* script is derived from the Brahmi script through various transformations.

The modern *Bangla* script alphabet consists of 11 vowel and 39 consonant characters. These characters may be called *basic* characters. Out of 49 basic characters of *Devnagari* script, 11 are vowels and 38 are consonants. The basic characters of both the scripts are shown in Fig. 1. Writing style in both the scripts is from left to right. The concept of upper/lower case is absent in both *Bangla* and *Devnagari*.

From Fig. 1 it is noted that most of the characters have a horizontal line at the upper part. The characters of words in these scripts are connected mostly by this line called *head line*. In *Bangla* language this line is called *matra* while in *Devnagari* (Hindi) language it is called *shirorekha*.

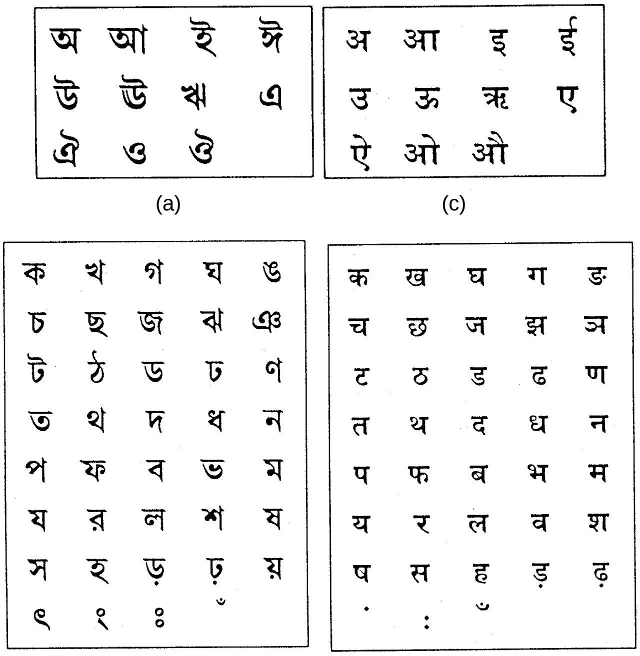

Fig. 1. (a) Vowels of Bangla script; (b) Consonants of Bangla script; (c) Vowels of Devnagari script; (d) Consonants of Devnagari script.

In both classes of script a vowel following a consonant takes a modified shape, which depending on the vowel is placed at the left, right(or both) or bottom of the consonant. These are called *modified* characters (allographs). For example, see Fig. 2. If the first character of the word is a vowel then it retains its basic shape. For two consecutive vowels in a word, the second one retains its basic shape. A consonant or vowel following a consonant sometimes take a compound orthographic shape which we call as *compound* character.

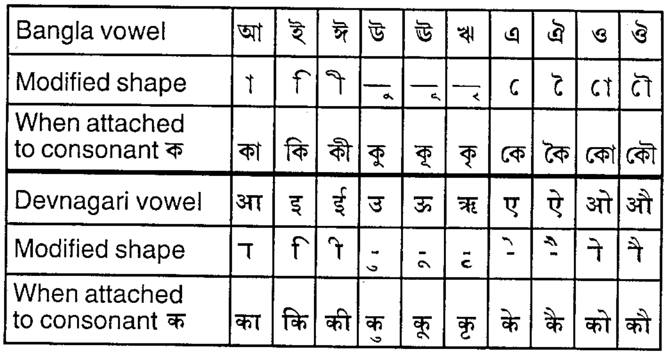

Fig. 2. Shapes of Bangla (Devnagari) vowel modifiers when attached to the consonant ক (क).

More formally speaking, modifiers are those symbols which do not disturb the shape of the basic characters (in middle zone) to which they are attached. If the shape is disturbed at the middle zone, we call the resultant shape as compound character shape. Compound characters can be combinations of consonant and consonant, as well as consonant and vowel. In both the script forms there are about 250 compound characters, very few of which are formed by consonant-vowel combination. Compounding of three consonants is also possible. The vowel and consonant modifiers may be attached to the compound characters, as usual.

A text line in both scripts may be partitioned into three horizontal zones. The upper zone denotes the portion above the head line, the middle zone covers the portion of basic (and compound) characters below head line and the lower zone is the portion where the modifiers can reside. A typical zoning is shown in Fig. 3.

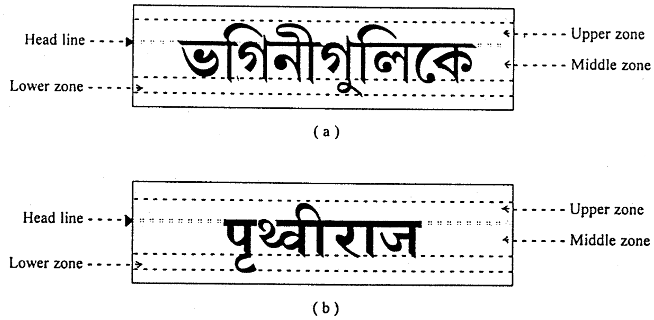

Fig. 3. (a) Different zones of Bangla script word; (b) Different zones of Devnagari script word.

3 PROPOSED SKEW DETECTION APPROACH

As mentioned, our approach is based on the detection of head lines of document words. If several consecutive characters of a word have head lines they make a continuous straight line which, if properly detected, shows the orientation of the document. The estimate of orientation (hence skew) is better if the head line is longer.

To test the likelihood of head line in a word we note that out of the 50 basic characters in *Bangla* there are 32 characters with head lines while in *Devnagari* out of 49 basic characters 42 characters have head lines. We have computed character occurrence statistics in *Bangla* language [3] which confirms that out of twelve most frequent characters only one character has no head line. So, it is likely that most *Bangla* words will have a head line. We can make a better quantitative analysis of the presence of head line, as follows.

According to our statistics [3], the average length of *Bangla* words is about six characters. About 30%-35% of characters are vowel modifiers which, being small in size, contribute very little to the head line of the word. Most basic characters are consonants, as vowels in basic form can appear at the beginning of the word or when two vowels appear side by side. But in practical situation, the latter case is very rare. Also, we verified that [3] compound characters are very infrequent, occurring in about 5% of the cases only.

Thus, for simplicity, we assume that on an average four basic characters only contribute to the head line of the word. We also assume that each character is equally likely in a word. In *Bangla* 41 characters can appear in the first position of a word. Out of these 41 characters 30 characters have head lines. Hence probability of getting a character with head line in the first position of a word is $P_1 = \frac{30}{41}$. Then probability of getting a character without head line in the first position is $1 - P_1 = \frac{11}{41}$. As argued above, the characters which can contribute to the head line in the other positions of a word are mostly consonants. Since 28 out of 39 *Bangla* consonants have head lines, the probability of getting a consonant with head line for other positions in a word is $P_2 = \frac{28}{39}$. Then probability of getting a character without head line in other positions is $1 - P_2 = \frac{11}{39}$.

Thus, probability of all four characters without head line in a 0word is $(1 - P_1)(1 - P_2)^3 = 0.00601$ (assuming that all characters are equally likely and independently occurring in a word). Hence, probability that a word will have at least one character with head line is $1 - 0.00601 = 0.99399$. Analyzing in the same way we get for *Devnagari*, the corresponding probability of 0.99702. The practical situation is better than these estimates since characters are not equally likely in a word and most frequently used characters have head lines.

Since the characters in a word are usually connected through head line, a complete word in a (skewed or unskewed) document can often be detected by the method of connected component labeling. For skew angle detection, at first the connected component labeling in the document is done. At the time of component labeling, for each labeled component its bounding box (minimum upright rectangle containing the component) is defined. The mean b_m and standard deviation b_s of the boundary box width are also computed. Next, components having boundary box width greater than or equal to b_m and less than $b_m + 3b_s$ are retained. By thresholding at b_m the small components like dots, punctuation marks, isolated characters and characters without head line are mostly filtered out while by thresholding at $b_m + 3b_s$ big components that may represent graphs and tables are also filtered out. Because of these filtering processes the irrelevant components can not create error in skew estimation. Also, longer head lines of the words are often retained by this process. See Fig. 4b and Fig. 5b for the selected components of Fig. 4a and Fig. 5a respectively, obtained using this approach.

Next, we find *upper envelope* of the selected components. Consider a component with label *G*. From each pixel of the uppermost row of the bounding box, we perform a vertical scan and as soon as the pixel labeled *G* is encountered, we convert its label to *U*. The set of pixels labeled *U* obtained in this way denotes the upper envelope of the component. See Fig. 4c and Fig. 5c where all the upper envelopes of components of Fig. 4b and Fig. 5b, are shown. From Fig. 4c and Fig. 5c, note that in most of the cases the upper envelope contains the head line. In this way we could filter out the irrelevant data for further processing. Hough transform technique may be applied on the upper envelopes for skew estimation, but this is a slow process. We have used the following approach which is fast, accurate and robust.

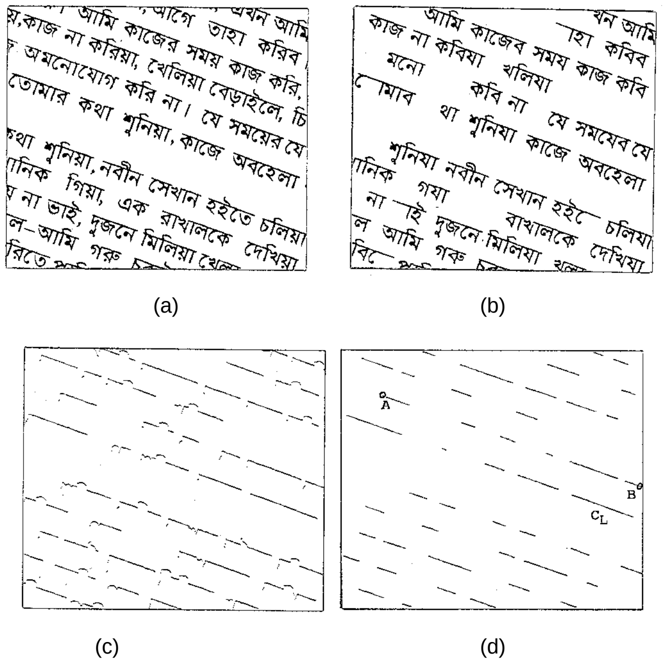

Fig. 4. Skew detection approach (Bangla). (a) An example of Bangla skewed text; (b) Selected components from Fig 4a.; (c) Upper envelope of selected components of Fig. 4b; (d) SDSL components of Fig. 4c.

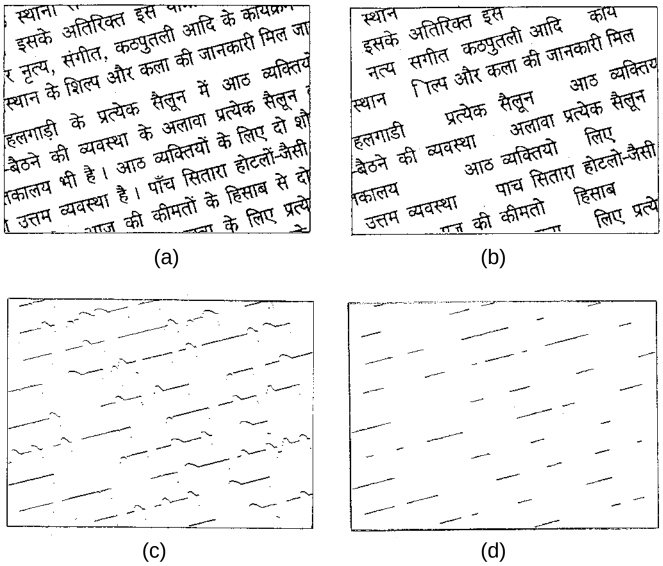

Fig. 5. Skew detection approach (Devnagari). (a) An example of Devnagari skewed text; (b) Selected components from Fig 5a.; (c) Upper envelope of selected components of Fig. 5b; (d) SDSL components of Fig. 5c.

The idea is based on the detection of *Digital Straight Line* (DSL) segments from the upper envelope. Although the upper envelope image contains head line information, it may contain non-linear parts due to some modified and basic characters. To delete these non-linear parts we find and select DSL segments in the upper envelope image. A DSL is a digital arc which result from digitization of a straight line segment. There are two ways of digital arc representation namely, *crack code* and *chain code*. In a chain code representation, two digital arc pixels can be 8-neighbors while in crack code representation two digital arc pixels can be 4-neighbors. We consider chain code representation of the upper envelope digital arcs.

TABLE 1
MEAN AND STANDARD DEVIATION (SD) OF ESTIMATED SKEW ANGLES OBTAINED BY DIFFERENT METHODS

True skew angle (in deg.) (manual)	Mean and SD of estimated skew angles using method					
	A_1		A_2		A_3	
	mean	SD	mean	SD	mean	SD
40	40.396	0.285	40.034	0.256	39.889	0.301
20	20.174	0.439	20.049	0.3162	20.047	0.242
10	10.271	0.393	10.166	0.201	10.112	0.323
5	5.064	0.458	4.962	0.213	5.188	0.233
2	1.986	0.396	2.151	0.234	2.054	0.307

For each true skew angle the statistics is computed over 20 document images.
A_1: *Hough transform over total image.*
A_2: *Hough transform over SDSLs of upper envelope.*
A_3: *Proposed quick method.*

The following conditions are necessary for straightness of a chain code digital arc [14].

- There exist runs of pixels in at most two directions which differ by 45°.
- For runs of two directions, the run lengths in one of the directions is always one.
- The run length in the other direction can have at most two values differing by unity (e.g., n and n + 1).

An example of a DSL is shown in Fig. 6. Here the angle between two directions d_1 and d_2 is 45° and run length in the direction d_2 is one whereas run lengths in d_1 direction are two (n) or three (n + 1) occurring alternately.

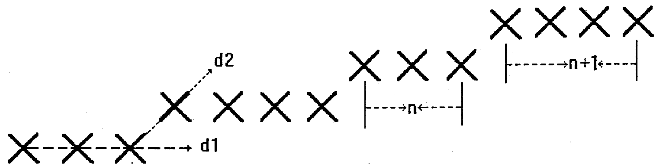

Fig. 6. Example of a digital straight line (DSL). Here, "X" denotes DSL pixel.

It should be noted that a digital arc satisfying the above three conditions may not always represent a digital straight line. The arc should satisfy the chord property for being a DSL. Arcs satisfying the above three properties belong to a superset of DSL and may be called SDSL. We considered SDSLs only because they are easy to detect and they represent straightness almost always in the data considered.

Now if skew angle estimates are taken for a reasonable number of SDSLs then a good estimate can be obtained from their aggregate. Note that better estimates of skew are obtained if the envelopes are long so that they have long head line regions. Hence, from upper envelope of each component we can find the largest digital arc which satisfy the three conditions stated above. These SDSL components are considered for skew detection (See Fig. 4d and Fig. 5d). Let the set of such components be R_1 and let the largest SDSL of all members of R_1 be C_L. For example see Fig. 4d. Rejection of other arcs makes a kind of noise cleaning.

If we take average of angles obtained from the lines joining the first and last pixel of every SDSLs with horizontal direction, we get an approximate estimate of skew angle. To get a better estimate we have clustered the members of R_1 corresponding to individual text lines. The required estimate of skew angle is obtained by averaging of angles between the horizontal line and the lines joining the first pixel of the leftmost SDSL and the last pixel of the rightmost SDSL of each individual text line (cluster).

The clustering is done as follows. From C_L or its continuation find the normal distances to a reference (say, leftmost) pixel of other SDSLs in R_1. The SDSLs of a particular text line will have nearly equal normal distances, and they can be clustered in terms of their normal distances. The coordinates of leftmost (x_{lc}, y_{lc}) and rightmost (x_{rc}, y_{rc}) pixels of each cluster C are also maintained for future convenience. Actually, (x_{lc}, y_{lc}) and (x_{rc}, y_{rc}) are the coordinates of leftmost pixel of leftmost SDSL and rightmost pixel of rightmost SDSL of cluster C.

The approach may be viewed as filling parametric space bins as in Hough transform. A bin corresponds to SDSLs having nearly equal normal distance defined above. Thus, for a SDSL N_1 of R_1 its normal distance of leftmost pixel to C_L is computed and if this distance is within $\pm H/2$ of normal distance of the leftmost pixel of the leftmost SDSL of any existing bin to C_L then N_1 is placed in that bin (H is the minimum possible distance between two text lines any document can have. Way of estimating H is given in the next paragraph). Otherwise, a new bin is created and N_1 is placed in the new bin. At the same time, the coordinates of the leftmost and rightmost pixels of each cluster are updated accordingly. For example, suppose a new SDSL q enters an existing cluster C. If the leftmost pixel (x_q, y_q) of q is to the left of the existing leftmost pixel (x_{lc}, y_{lc}) of this cluster i.e., if $x_q < x_{lc}$ then we make $x_{lc} \leftarrow x_q$, $y_{lc} \leftarrow y_q$. Else, no modification is done. The rightmost pixel of C is treated accordingly. The whole procedure can be executed in single scan over R_1. At the end, K bins and hence K clusters are usually produced if there are K text lines in the document. From each cluster C the angle of the line joining (x_{lc}, y_{lc}) and (x_{rc}, y_{rc}) with horizontal direction is found. See Fig. 4d where leftmost pixel A and rightmost pixel B of a text line (cluster) are shown. Average of such angles over all text lines gives an accurate estimate of the skew angle.

The value of H can be estimated as follows. For most documents, the character size is not smaller than six points. Then, minimum spacing between two head lines is 12 points. If the document is digitized at P DPI then the minimum distance (H) between the head lines of two text lines is $P \times 12/72$ pixels = P/6 pixels. (Since 72 points = 1 inch). For document digitized at 300 DPI we have H = 50 pixels.

In brief, our skew estimation algorithm has the following steps:
Algorithm SKEW-EST:

STEP 1: Find connected components in the binary document image and find the mean b_m and standard deviation b_s of their bounding box widths.

STEP 2: Choose the set S of connected components having bounding box width greater than or equal to b_m and less than $b_m + 3b_s$.

STEP 3: For each component in S find the upper envelope described above. From each envelope component, find the SDSLs. If more than one SDSL is found choose only the longest one and form the subset R_1. Let the longest SDSL in R_1 be C_L.

STEP 4: From the line C_L or its continuation find the normal distances to the leftmost pixel of other SDSLs of R_1.

STEP 5: Cluster the SDSLs of R_1 corresponding to individual text lines and find the leftmost and rightmost pixel of each cluster, as described above.

STEP 6: For each cluster find the angle of line joining the leftmost and rightmost pixels (e.g., A and B in Fig. 4d) with horizontal direction.

STEP 7: Average of such angles over all clusters (e.g., text lines) gives an accurate estimate of the skew angle.

4 RESULTS AND DISCUSSION

For the experiment we considered one hundred document images from different books, magazines and newspapers. Some documents contained graphics, tables and foreign texts (mainly English). Documents having both Bangla and Devnagari script lines were also chosen. They were digitized by flatbed scanner at a resolution of 300 DPI. The documents were tilted by a prespecified angle ranging between 0 and ±45°. This angle is considered as true skew angle. As a comparative study, we estimated skew angle using Hough transform over the original documents as well as using Hough transform over SDSL of upper envelope. Typical results are shown in Table 1.

To test the time efficiency of our quick method (A_3) with respect to conventional Hough transform method on original document image (A_1) as well as on SDSL (A_2), we computed the time of execution in these methods. The angular resolution used in Hough transform was 1°. The average execution times for a document of 512 × 512 pixels on a SUN 3/60 (with Microprocessor MC68020, and SUN O.S. version 3.0) machine are 620, 312, and 17.80 seconds for methods A_1, A_2 and A_3, respectively. Note that, Hough transform execution time would be higher if finer angular resolution (say 0.5°) was considered.

To test the skew angle estimation efficiency of the methods we have done a statistical testing of hypothesis using F-distribution obtained as follows. Let X_1, X_2, ... X_n be n independent observations from a Gaussian distribution with mean ϕ and standard deviation σ i.e.,

$$\sum_{i=1}^{n} \frac{\left(X_i - \phi\right)^2}{\sigma^2} \approx x_n^2 .$$

For independent true skew angles Y_j (j = 1, 2 ... n_1) let X_{ji} and Z_{ji} (i = 1, 2, ... n_2) be the independent estimated angles for *i*th observation by methods A_3 and A_2, respectively. We assume X_{ji} and Z_{ji} follow Gaussian distributions with standard deviations σ_1 and σ_2, respectively. Now

$$T1 \triangleq \frac{1}{\sigma_2^1} \sum_{j=1}^{n_1} \sum_{i=1}^{n_2} \left(X_{ji} - Y_j\right)^2 \approx \mathcal{X}_{n_0}^2$$

and

$$T2 \triangleq \frac{1}{\sigma_1^2} \sum_{j=1}^{n_1} \sum_{i=1}^{n_2} \left(Z_{ji} - Y_j\right)^2 \approx x_{n_0}^2$$

Where $n_0 = n_1 \times n_2$.

Our null hypothesis is $H_0 : \sigma_1 = \sigma_2$ under which

$$T2 = \Delta \frac{T_1}{T_2} \approx F_{n_0,\, n_0}$$

where $F_{n0,\, n0}$ is F-distribution with n_0 and n_0 degrees of freedom.

From our estimated skew angles the value of T is found as 1.24, where n_1 = 5 and n_2=20. The 5% cut off point of the F-distribution (with degrees of freedom 100 and 100) is 1.41 (obtained from F-distribution table through interpolation). Hence, the methods A_2 and A_3 are statistically equally accurate. However, as pointed out before, method A_3 takes less execution time.

One additional advantage of the proposed method is that the head line is readily detected during skew estimation. The characters in a word can be separated if the head line region is rubbed off. The separated characters are then subjected to the OCR recognition procedure. Moreover, detection of head line automatically separates middle zone from upper zone of a text line (see Fig. 3), thus helping in zone detection. Our experience of *Bangla* OCR [10] suggests that zone separation is beneficial in devising a good character recognition technique. Also, since our method is based on connected component labeling, document structure analysis can also be readily done from the skew corrected output.

If skew detection is the only goal then for a large document the complete document need not be subject to this technique. A small horizontal strip can be considered for the purpose. The height of the strip could be small if the expected skew angle is small. If no a priori idea about skew angle is available, we found that the strip height equal to one third of the larger side length of the document is an acceptable choice.

ACKNOWLEDGMENT

Interest in the work by Dr. S.K. Parui is acknowledged with thanks. One of the authors (Mr. U. Pal) wishes to thank CSIR, India for providing him a fellowship in pursuing his work.

REFERENCES

[1] T. Akiyama and N. Hagita, “Automatic Entry System for Printed Documents,” *Pattern Recognition*, vol. 23, pp. 1,141-1,154, 1990.
[2] H.S. Baird, “The Skew Angle of Printed Documents,” *Proc. Society of Photographic Scientific Eng.*, vol. 40, pp. 21-24, 1987.
[3] B.B. Chaudhuri and U. Pal, “Relational Studies Between Phoneme and Grapheme Statistics in Modern *Bangla* Language,” *J. Acoustical Society of India*, vol. 23, pp. 67-77, 1995.
[4] L.A. Fletcher and R. Kasturi, “A Robust Algorithm for Text String Separation From Mixed Text/Graphics Images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp. 910-918, 1988.
[5] L. O’Gorman, “The Document Spectrum for Page Layout Analysis,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1,162-1,173, 1993.
[6] A. Hashizume, P.S. Yeh, and A. Rosenfeld, “A Method of Detecting the Orientation of Aligned Components,” *Pattern Recognition Letters*, vol. 4, pp. 125-132, 1986.
[7] S.C. Hinds, J.L. Fisher, and D.P. D’Amato, “A Document Skew Detection Method Using Run-Length Encoding and the Hough Transform,” *Proc. 10th Int’l Conf. Pattern Recognition*, vol. 1, pp. 464-468, 1990.
[8] H.S. Hou, *Digital Document Processing*. New York: John Wiley, 1983.
[9] D.S. Le, G.R. Thoma, and H. Wechsler, ”Automatic Page Orientation and Skew Angle Detection for Binary Document Images,” *Pattern Recognition*, vol. 27, pp. 1,325-1,344, 1994.
[10] U. Pal and B.B. Chaudhuri, “Computer Recognition of Printed *Bangla* Script,” *Int’l J. Systems Science*, vol. 26, pp. 2,107-2,123, 1995.
[11] U. Pal and B.B. Chaudhuri, “An Improved Document Skew Angle Estimation Technique,” *Pattern Recognition Letters*, vol. 17, pp. 899-904, 1996.
[12] T. Pavlidis and J. Zhou, “Page Segmentation and Classification,” *Computer Vision Graphics and Image Processing*, vol. 54, pp. 484-496, 1992.
[13] W. Postl, “Detection of Linear Oblique Structures and Skew in Digitized Documents,” *Proc. Eighth Int’l Conf. Pattern Recognition*, pp. 464-468, 1986.
[14] A. Rosenfeld, “Digital Straight Line Segments,” *IEEE Trans. Computers*, vol. 23, pp. 1,264-1,269, 1974.
[15] S.N. Srihari and Govindaraju, “Analysis of Textual Images Using the Hough Transform,” *Machine Vision Applications*, vol. 2, pp. 141-153, 1989.
[16] K.Y. Wong, R.G. Casey, and F.M. Wahl, “Document Analysis System,” *IBM J. Res. Development*, vol. 26, pp. 647-656, 1982.
[17] H. Yan, “Skew Correction of Document Images Using Interline Cross-Correlation,” *CVGIP: Graphical Models and Image Processing*, vol. 55, pp. 538-543, 1993.