J. Indian Inst. Sci., May–Aug. 2002, **82**, 161–175

Skew angle detection of a cursive handwritten Devanagari script character image

RAJIV KAPOOR,[*$] DEEPAK BAGAI,[$] T. S. KAMAL[‡]
[$]Division of Electronics Engineering, Punjab Engineering College, Chandigarh 160 012, email: raj_himani@hotmail.com, Ph: 91-172-737221. email: dbagai@yahoo.com, Ph: 91-172-600590.
[‡]Division of Electronics Engineering, Sant Longowal Institute of Technology, Longowal, Punjab (India) 148 106. email: tskamal@yahoo.com, Ph: 91-1672-83930, Fax: 91-172-540133.

Abstract

This paper proposes an accurate and exhaustive approach to detect the skew angle of the images of words/characters of cursive Devanagari script. This approach was applied to 235 writing samples and a total collection of around 6000 samples. It is efficient in terms of time and is a simpler process as compared to the existing ones. The method is an extension to the work of Pal and Chaudhuri [B. B. Chaudhuri and U. Pal, Skew angle detection of digitized Indian script documents, *IEEE Trans*. **PAMI-19**, 182–186 (1997)]. Heuristic approach has been applied to detect the skew angle. The inherent dominating features of the structure of the Devanagari script have been used to accurately calculate the skew of the Devanagari word.

Keywords: Character recognition, DSL detection and Shiro-Rekha.

1. Introduction

Offline handwritten character recognition is a process where the computer understands automatically the image of a handwritten script. Offline handwriting is therefore distinguished from online handwriting, where a device such as a digitizing tablet measures the path of the pen. Online text capturing process provides more features as compared to the offline process but most of the applications must have offline capturing. Moreover, age-old documents lying in offices have to be fed to the computer in text form. A number of applications can be developed including document transcription, automatic mail routing, and machine processing of forms, cheques and faxes [1]–[4]. Many other systems have been developed to recognize offline handwriting, but these are limited to digit recognition or printed material. This work makes the Devanagari text skew-free. Moreover, the handwritten material is generally not in straight line, hence, it is extremely important to straighten the written word.

This process starts with the scanning of the initial image from paper to the output of machine-readable text. The recognition system can be conveniently divided into the same broad sections of preprocessing, recognition and postprocessing as is common in most other handwriting recognition systems. The processing starts with data acquisition and ends up with recognition process (Fig. 1).

*Author for correspondence.

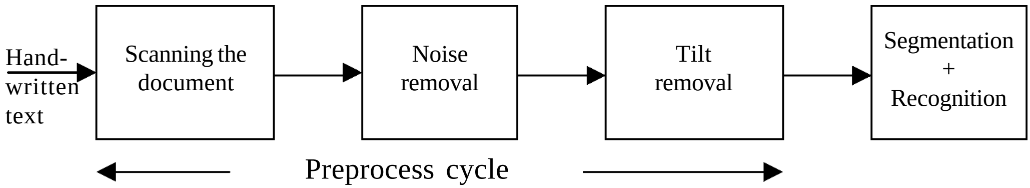


FIG. 1. Character recognition process flow diagram.

To capture the data from a handwritten document, a conventional flatbed scanner has been used. The scanner image must be segmented into separate words but made tilt-free before segmentation. Subsequently, a series of image-processing operations are carried out to normalize the image, so as to make it invariant to distortion [2]. The writer used a black dot-pen for writing. Each sheet was scanned at 300 dpi resolution to create a file. Once the skew has been detected and removed thereafter, a simple segmentation algorithm [3] can be used to separate identifiable characters. In the next section, the literature on the subject is reviewed.

2. Existing methods

Kim and Govindaraju [5] have employed chain code method of image representation. Chain code is a linear structure that results from quantization of the trajectory traced by the centres of adjacent boundary elements in an image array. Each data node in the structure represents one of the eight grid nodes that surround the previous data node. Tracing of chain code components was carried out for slant angle correction using a pair of one-dimensional filters. Each filter is a five-element array of different weights. Coordinates of the start and end points of each vertical line extracted provide the slant angle. Global slant angle is the average of all the angles of the lines, weighted by their length in the vertical direction since the longer line gives more accurate angle than the shorter one.

Another technique depends on the projection profile of the handwritten document where the horizontal or vertical profile is only a histogram of the information pixels scanned horizontally or vertically [6]. For a script with horizontal text lines, the horizontal projection profile will have peaks at text line position and troughs at positions in between successive text lines. The projection profile taken in such a manner is computed at a number of angles. And for each angle, peak and trough height difference is measured. The maximum difference corresponds to the best alignment in the direction of the tilt, which in turn determines the tilt angle. Baird [7] proposed a modification for quick convergence of this iterative approach. Akiyama and Hagita [8] described an approach where the document is partitioned into vertical strips. The horizontal projection profiles are calculated for each strip and the angle of tilt is determined from the correlation of the profiles of the neighboring strips. The method is fast but less accurate. Pavlidis and Zhou [9] proposed a method, which is based on vertical projection profile of horizontal strips, which works well when the angle of tilt is small. Yan [10] proposed another method, which uses cross-correlation between the lines at a fixed distance. An analysis shows projection profile methods are suitable for tilt angles of less than 10.

Certain techniques based on Hough transform are also popular to detect the angle of tilt [11]–[13]. Hinds *et al.* [11] modified the Hough transform method to reduce the amount of data to be processed. Le *et al.* [12] found connected components in a script and considered only the bottom pixels of each component for Hough transform, which reduces the amount of information data. Pal and Chaudhuri [14] have improved this approach.

Fourier transform-based methods have also been used to detect the angle of tilt. Postl [15] proposes an approach where the direction of the density of Fourier transform is the largest gives an estimate of the angle of tilt.

Hashizume *et al.* [16] propose the nearest neighbor clustering approach to detect the angle of tilt. He found all the connected components in the document and computed the direction of its nearest neighbor for each component. Thereafter, a histogram of the direction angle is computed. Its peak indicates the document angle of tilt. O'Gorman [17] suggested 'docstrum analysis'. These approaches are not limited to any range of tilt angle.

It is clear that the component nearest clustering methods will not work well for Devanagari script because the characters are connected by the headline. Modified Hough transform method is also unsuitable for Devanagari script because of the connecting nature of the word. Some pixels at the bottom of each word will make the data very sparse and its peak will be quite flattened.

Now, from the above, it is clear that the angle of tilt can be found easily only if some inherent characteristic of the script can be used and this is always so. Based on this characteristic, Chaudhuri and Pal [18] suggested a skew detection approach for the printed characters of Devanagari script. This approach is based on detecting the inherent feature (head line) in the Devanagari word. Here, the author has detected the word in a skewed document by using the method of connected component labeling. Then, the upper envelope of the selected components is isolated. Finally, a new approach has been used to detect the skew based on the detection of digital straight line (DSL) segments from the upper envelope. DSL (digital arc) have been represented by chain code. Certain points deserve attention:

a) This method was applied to printed characters.
b) The documents were tilted by an angle ranging between 0 and $\pm 45^{\circ}$.
c) Minimum average execution time is 17.8 s.
d) The angular resolution used in Hough transform is 1°.

The method proposed by us also exploits the inherent head line (property) of the Devanagari script but does not need the overhead of preprocessing and has wide range of angle detection. It is also comparatively efficient.

3. Proposed method

The method is applicable to the whole document image. The method is based on Shiro-Rekha as inherent feature of Devanagari. If there is no Shiro-Rekha then the document will not be inspected based on the same criterion. The method assumes at the beginning that the document contains Shiro-Rekha. If the method is not able to detect the Shiro-Rekha and the line feature as explained later on then the document will be inspected as an ordinary image but not as a special case of

Devanagari. The following steps have been implemented and tested. The method gives fairly good accuracy and this work stresses on the exploitation of the Shiro-Rekha feature. For this reason, the skew in the character image without Shiro-Rekha can be removed only if the following steps are followed faithfully.

1. Extract the word from the document image (the method followed is discussed later on).
2. The extracted word is fitted in a standard frame (while testing, the size was calculated by taking the mean length and height from the sample word documents) to ensure the precision factor.
3. The image is projected by rotating (explained in Section 4) it from -89^{o} to $+89^{o}$.
4. Calculate the Wigner–Ville distribution (a joint function of time and frequency) of the projections taken at various angles.
5. Choose the angle, which presents the maximum intensity after applying WVD, and hence the word skew is corrected.

3.1. *Detecting document without skew*

Scan the whole image horizontally and count the number of pixels in each row. Prepare a table with two columns of row number and number of pixels. We find out from the table whether some consecutive lines fulfill the constraint mentioned below. Based on this, we define a feature which once extracted will prove that the character image has a tilt or not. That feature has been named as 'line' which means that a feature 'line' has been detected when scanned row gives the number of information pixels that are at least 90% of the total number of pixels possible horizontally in an image frame (Fig. 2).

$$H_{LINE} \geqslant 0.9 * L \text{ (number of pixels)} \tag{1}$$

where H_{LINE} is the number of pixels in a row while scanning horizontally.

3.1.1. The X-axis in the graph shows the row number and the Y-axis the number of pixels. The graph showing the peak meeting of the constraint mentioned above will be termed as 'line detection' and once detected has to be confirmed through vertical scanning.

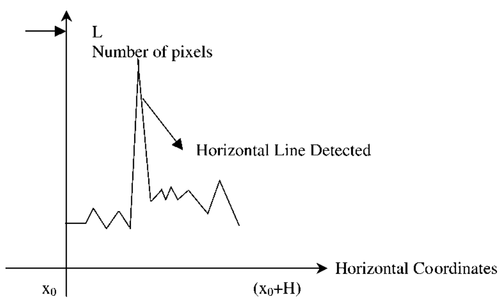


FIG. 2. Horizontal projection.

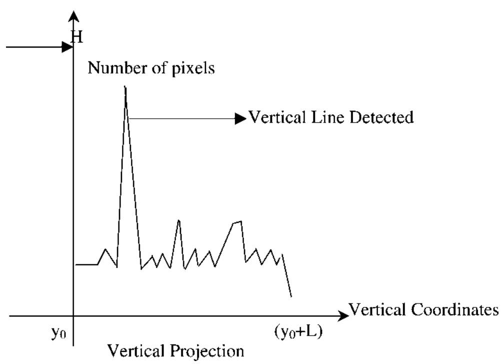


FIG. 3. Vertical projection.

3.1.2. Even if the line has been detected after horizontal scanning, it still has to be confirmed through vertical scan and no further processing is needed if the same result is obtained. Vertical projection and finding the line feature will also require the same effort as in the horizontal process. The same has been demonstrated in Fig. 3.

$$V_{LINE} \geqslant 0.9 * H \text{ (number of pixels)} \quad (2)$$

where V_{LINE} is the number of pixels in a row while scanning horizontally.

3.1.3. Once the 'line' has been detected, there is no further need to process the image to find the tilt. This means that the image is either exactly horizontally straight, i.e. at 0^o, 180^o or it is exactly vertical at 90^o or 270^o. The distinction between the 0^o and 180^o images will depend on the direction of the scan (top-left or top-right) and the detection of the line feature. Similar process is applied to have distinction between 90^o and 270^o. If the 'line' is not confirmed during the vertical scan then that means the image is tilted and we need to find the tilt angle and remove it. Hence move to step 3.2.

3.2. Devanagari characters are formed in a structured manner. Whereas most of the language scripts are written with reference to the central axis, the Devanagari script is written w.r.t. the central axis along with an axis parallel to the central axis. This is the inherent property of writing Devanagari script. Once the angle of the axis w.r.t. the reference axis has been detected, thereafter the word can be rotated perfectly horizontally. This dominating feature of Devanagari script has been used in this paper for detecting the tilt angle. This can be better explained with an example (Fig. 4). Scannable area is nothing but the cover of the word when it is viewed from four directions. But the informative scan area is the projected view ON_1 from left side of the character image and ON_2 is the projected view from topside of the image. If the word image is enclosed in a rectangular frame then, we easily get the length and the height of the image, which has been represented by L and H, respectively. In the case of handwritten characters, the top line is generally not straight; therefore, we need to select the smoothest portion of the top line. Tilt angle has been represented by ζ. Figure 5 shows the skewed handwritten document being used to demonstrate the skew detection process.

It is clear from the document that the handwritten document words of a line do not have exactly the same skew. Even the headline is also not exactly straight. The document shown above

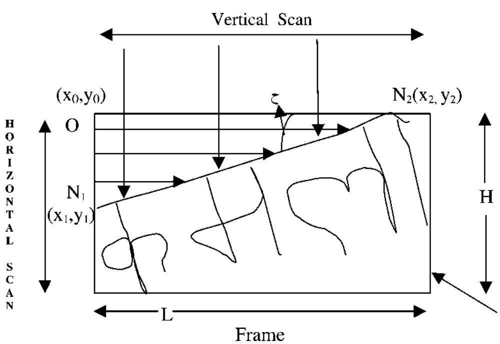
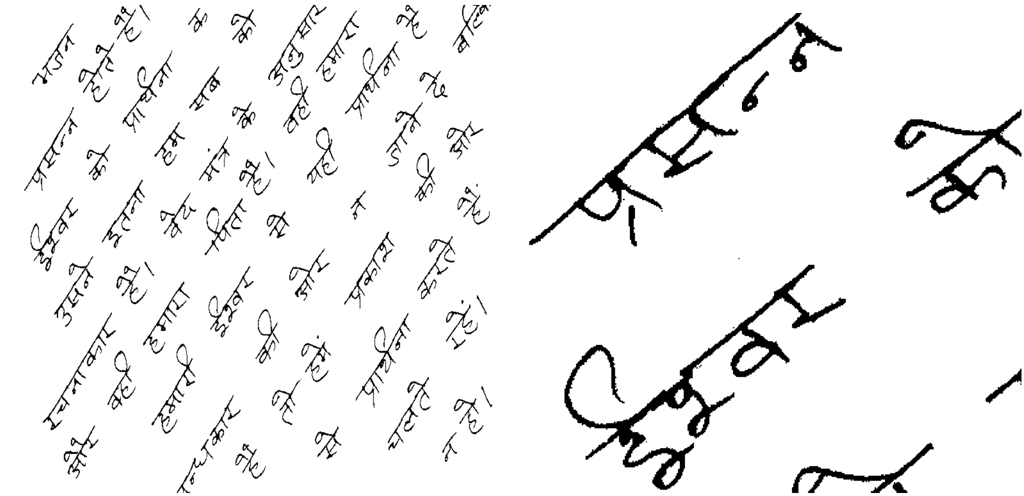


FIG. 4. Handwritten Devanagari word sample.



(a) (b)

Fig. 5. (a) Test sample document and (b) portion of the document.

has tilted words and simply scanning the document horizontally and marking boundary patterns made the segmentation of document into isolated words. While scanning the document, the patterns are marked so as to recollect the coordinates of the marked locations. Figures 6 and 7 show the left and the right scan covers of the document image. A portion of the bigger document image (Fig. 5 b) has been used for demonstrating the determination of the scan direction, which in this case comes out to be LT scan (left-top scan).

The skew angle has been detected without separating the word from the document image. This helps in the recognition process by providing sufficient details to segment even the individual character. To give a demonstration of the work without showing the whole document, Devanagari word USKI has been assumed as shown below having various skew positions on which the method has been applied. Chaudhuri and Pal [18] have computed b_m and b_s of the image bounding box, as mean and standard deviation, respectively, to isolate the false figures but this is not required in our case as the false figures get filtered out while detecting the skew of the

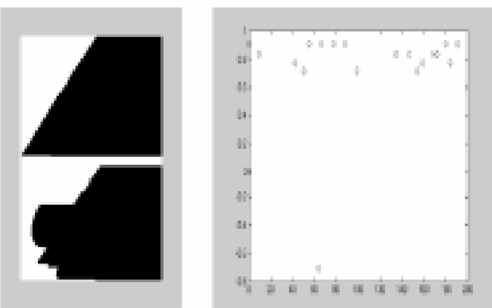


FIG. 6. (a) Left cover of the test sample of Fig. 5(b) and (b) differential chart of the left cover.

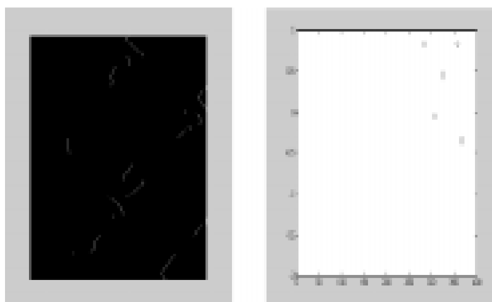


FIG. 7. (a) Right cover of the document sample of Fig. 5(b) and (b) differential chart of the right cover.

individual word. This shows that the skew of every word in the document may not be the same in the case of handwritten word documents. In addition, we do not isolate the cover of the word; rather our method gives the value of the tilt without isolating the cover from the word by collecting the first strike information from the already determined scan directions and hence is more efficient.

The above task of false figure detection has been handled along with the task of determining the skew angle. We have exposed the figure to any of the four pair-scan-directions and the results of the differential figure give the clear indication of a false figure. In the proposed method, we need not bother how the document has been scanned. For example, if the document has been kept in such a manner that the headline of the words is almost vertical, our proposed method still works better as we are determining the scan directions at the beginning. In the case of Chaudhuri and Pal [18], even when there is no tilt in the document, we need to follow all the steps.

Keeping this in mind, the foremost objective is to determine the scanning direction like left-top scan, top-right scan, left-bottom scan or right-bottom scan (Fig. 8). This will help us to determine the quadrant in which the handwritten word has been written. Remember that this method is applicable only when the skew angle of all the words throughout the document is almost similar but may not be exactly. This job is accomplished in the following steps:

a) The document is scanned from all the four sides for recording the coordinates of the first information pixel encountered along with the demarcation of the word boundaries.

b) First-order differential of the coordinate data gives the spatial-level curve. The curve is normalized to adjust the abnormalities due to handwriting simply by thresholding of levels.

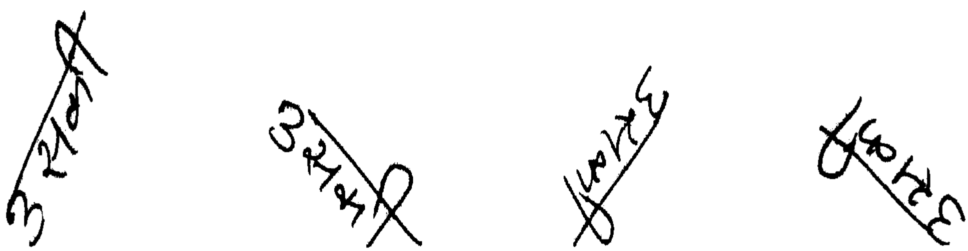
c) The individual levels result in the regions by using the nearest neighborhood algorithm and the mean of the biggest region is recorded out of all the four scans.

d) The biggest levelregion out of the four gives the scan direction.

e) Once the scan direction of the documents is known, the word has to undergo the following steps.

It is important to ensure the following before going for vertical or horizontal scan.

3.3. It has been confirmed that the document has skew and also that the skew is not 0, 180, 270 or 90. Along with this, the boundaries of the words have been demarcated.



(a) (b) (c) (d)

FIG. 8. Devanagari word with various tilt angles. (a) LF, (b) TR, (c) RB and (d) LB scans.

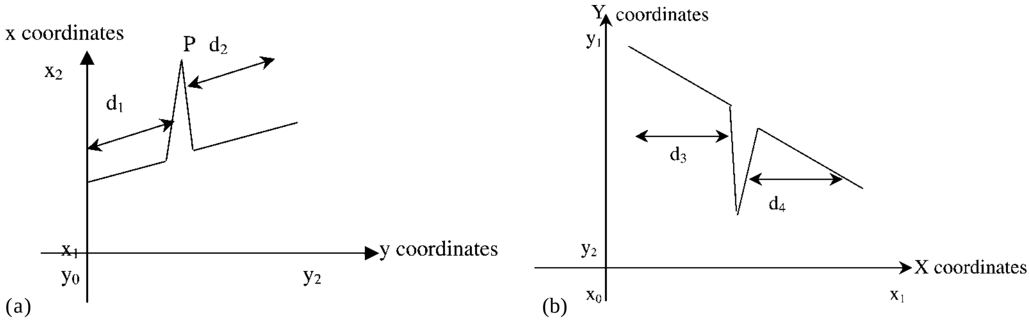


FIG. 9. (a) Vertical and (b) horizontal scans–ambiguous/nonambiguous data points.

3.3.1. *Jitter detection due to Matras*

The process has been explained by taking a sample word (Fig. 4). Scanning is performed from top to bottom direction as shown below and the coordinates of the location wherever the information comes first on the way is recorded. Now our data contain the set of coordinates where vertical scan strikes first. The following points need attention:

a) The image has to be noise free.

b) There should not be big difference in the (a, x) coordinates of the information points recorded either from left top to right top or right top to left top. This big change shows the ambiguity and is not information for us. This has been shown in Fig. 9 (a) (peak P).

c) In case of ambiguity, the linear part (d_1 and d_2) of the curve should be considered for finding the angle of tilt.

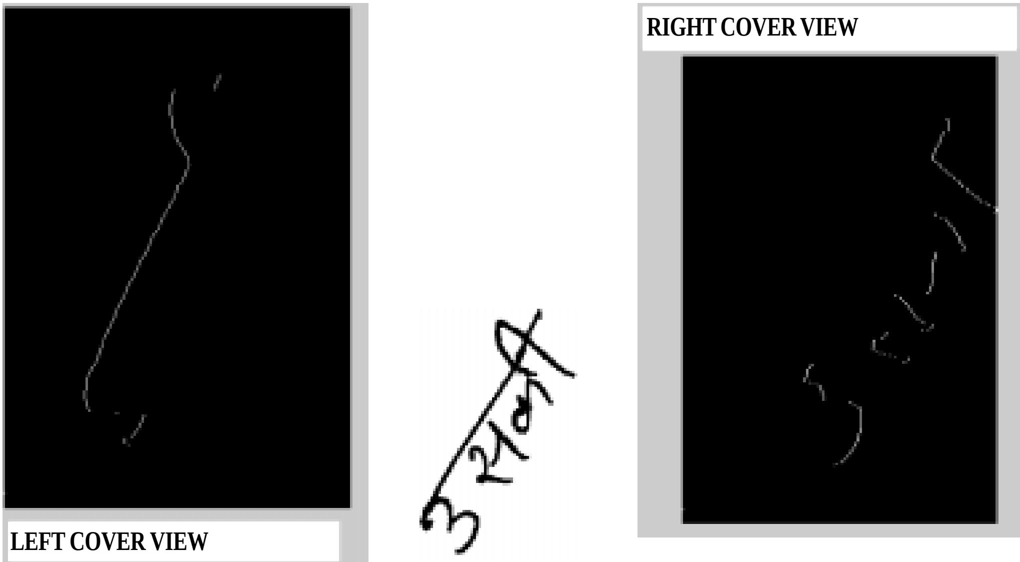
d) Then move to step 3.4.

3.3.1.1. This task is performed in a similar manner as during the vertical scan. All precautions similar to vertical scan should be taken in the case of horizontal scan also. The linear portion of Fig. 9 (b) (d_3 and d_4) should be considered.

3.4. Once the scanned data has been recorded then an analysis needs to be performed to detect the angle of tilt. A model has been devised to take care of all the tilt angles and to exploit the heuristics of Devanagari characters. Figure 10 shows the left and the right cover view of the Devanagari word USKI. Here the left cover indicates clearly the isolation of Shiro-Rekha from the rest of the word. It also indicates that the handwritten word will never have the straight Shiro-Rekha exactly and hence we have considered in this paper the most straight part of the Shiro-Rekha for skew determination. Let us define some of the terms being used.

3.4.1. *Weights*

Once the data (information) has been extracted, the tilt angle finding model should assign the following weights. Weights have been assigned as the utility of information collected through the vertical and horizontal scans will depend on the angle of tilt. Hence these weights need to be changed depending upon the angle of tilt. This seems difficult to implement because we are still



(a) (b) (c)

FIG. 10. Original segmented image and its cover views. (a) Left cover view, (b) original word image, and (c) right cover view.

in the process of finding the angle of tilt. In this paper, we follow a simple principle: of the vertical and horizontal scan graphs, the one with lesser ambiguities (transitions) will be given more weight and the graph having more ambiguities will be given lesser weight. Table I shows the weights assigned to the ambiguities.

a) Weight assigned to the angle calculated by the vertical scan.

b) Weight assigned to the angle calculated by the horizontal scan.

Table I
Weights for horizontal and vertical scans based on predicted angle and jitters

No. of transitions		Predicted angle	Weights (W)	
Horizontal	Vertical		Vertical	Horizontal
*****	Nil	$\eta 1$	1	0
*****	1 to 5	$\eta 1$	0.9	0.1
*****	6 to 10	$\eta 1$	0.8	0.2
*****	11 to 15	$\eta 1$	0.7	0.3
*****	> 15	$\eta 1$	0.6	0.4
Nil	****	$\eta 2$	0	1
1 to 5	*****	$\eta 2$	0.1	0.9
6 to 10	*****	$\eta 2$	0.2	0.8
11 to 15	*****	$\eta 2$	0.3	0.7
> 15	*****	$\eta 2$	0.4	0.6

*$0<\eta<\pi/3$ or $2\pi/3 < \eta < 2\pi$ (Condition $\eta 1$)

*$\pi/3 < \eta < 2\pi/3$ (Condition $\eta 2$)

3.4.2. *Experimental model*

Following properties add to the model:

a) Angle detected after the horizontal scan (λ_h)

b) Angle detected after the vertical scan (λ_v)

c) Weights (w)

3.4.3. *Constraints*

a) $\zeta \approx (w_h \lambda_h + w_v \lambda_v)$, where $w_v >> w_h$ and $w_h \leq w_v$ if $0 < \eta < \pi/3$ or $2\pi/3 < \eta < 2\pi$ and η is the approximate angle predicted which in the above example is $\tan^{-1} 5.6/9 = \tan^{-1} 0.62 = 31.89^o$.

b) $\zeta \approx (w_h \lambda_h + w_v \lambda_v)$, where $w_h >> w_v$ and $w_v \leq w_h$ if $\pi/3 < \eta < 2\pi/3$.

3.5. *Final model parameters*

a) Detected angle of tilt (λ)

b) Scan direction (v/h) (Φ)

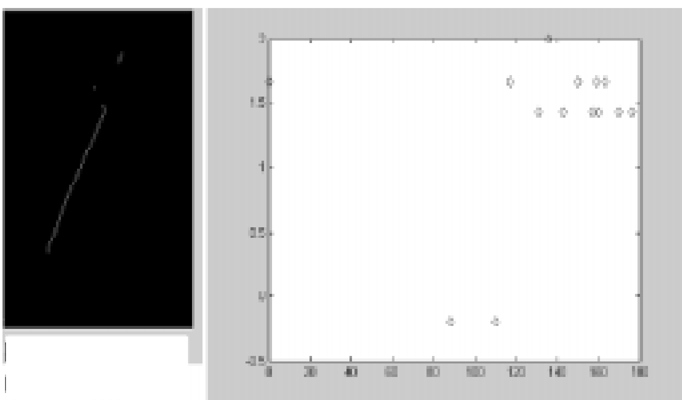
c) Weight to either of the scan (w)

Therefore, $\zeta(\lambda, \Phi, w)$ model gives the final tilt angle, provided the constraints mentioned above have been met.

Example:

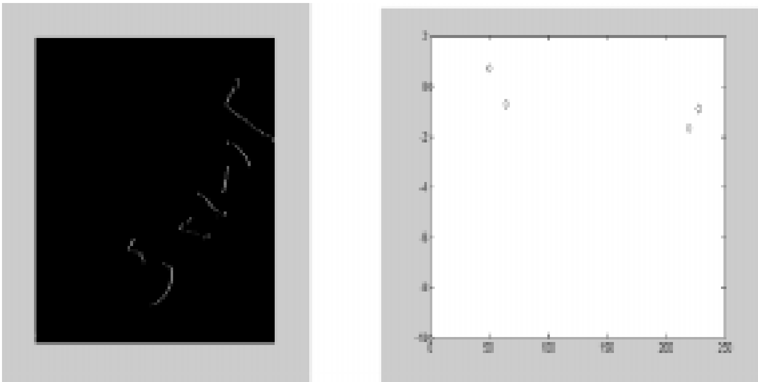
Number of ambiguities in horizontal (left/right) scan = Nil
Number of ambiguities in vertical (top/bottom) scan = Nil
$\lambda_v = \tan^{-1} 2.5/8.5 = \tan^{-1} 0.29 = 16.4^o$, $\lambda_h = \tan^{-1} 2.5/8.5 = \tan^{-1} 0.29 = 16.4^o$

$$\zeta(\lambda, \Phi, w) = \zeta((\lambda_v, v, 1.0), (\lambda_h, h, 0))$$
$$= 1.0 * 16.4 = 16.4^o$$



(a) (b)

FIG. 11. (a) Left cover of the word USKI and (b) differential of Fig. 10 (a).



(a) (b)

FIG. 12. Differential of the right cover of the word USKI. (a) Right cover of the word USKI and (b) differential of the right cover.

Various levels are clustered using the nearest neighborhood algorithm to form various regions. The biggest region qualifies as the region of importance. The mean of the regional levels is calculated which is termed as λ. If the scan is horizontal then the mean is $\lambda_{h.}$ Similarly, the mean of the biggest region will be termed depending upon the scan direction. For example, if the scan direction determined is LT scan then we will calculate λ_h and λ_v. Figures 11 and 12 basically demonstrate the way to determine the direction of scan and the skew angle. It can be noticed that the differential of Fig. 11 shows clusters and Fig. 12 the isolated points. We have considered clustered differential for skew detection.

4. Finally, after detecting the tilt angle, consider the character image as a general image while rotating the same image to an angle detected above.

4.1. *Tilt removal process after skew detection*

Once tilt angle ζ has been detected and finalized then a simple transformation technique works perfectly fine to straighten the tilted word of Devanagari script. The transformation technique [4]

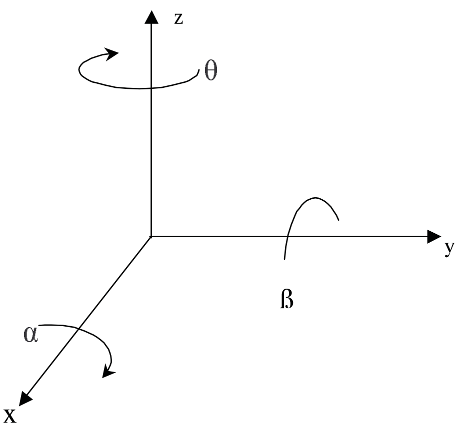


FIG. 13. Tilt angles.

mentioned below is applicable for rotating a point about the coordinate axes. To rotate a point about another arbitrary point in space requires three transformations. The first translates the arbitrary point to the origin, the second performs the rotation and the third translates the point back to its original position.

Referring to Fig. 13, the rotation of a point about the z coordinate axis by an angle θ is achieved by using the transformation

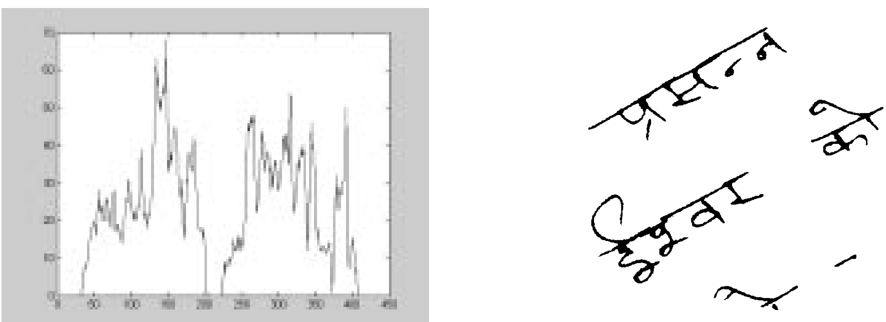
$$R\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The rotation angle θ is measured clockwise when looking at the origin from a point on the $+z$ axis. This transformation affects only the values of x and y coordinates. Rotation of a point about the x-axis by n angle x is performed by P using the transformation

$$R\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } \alpha = \zeta. \tag{4}$$

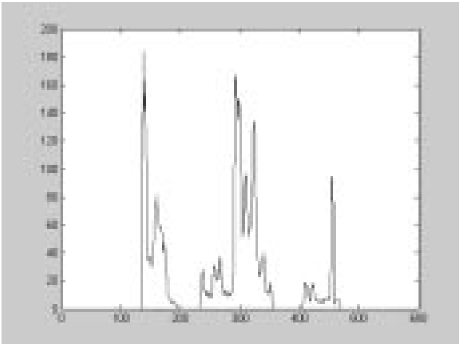
Similarly, the rotation of a point about the y-axis by an angle β is achieved by using the transformation

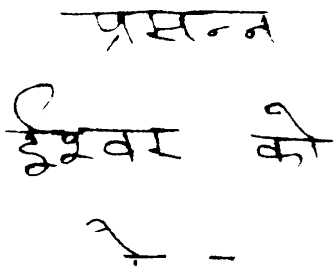
$$R\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } \beta = \zeta. \tag{5}$$



(a) (b)

FIG. 14. (a) Projection of document at angle 10° and (b) the handwritten document.





(a) (b)

FIG. 15. (a) Projection of document at angle 39° and (b) the handwritten document.

The transformation mentioned above has been applied to every information point in the Devanagari script character image. The technique works perfectly.

4.2. *Tilt removal without detecting the skew angle*

Rotating the whole document in a clock or anticlockwise direction and taking the projection of the document continuously and horizontally can also correct the skew angle. The document is analyzed for detection of line feature, which has been explained earlier. The document clearly gives the 'line' feature, which is almost at equal period. The only problem with this method is that handwritten words are generally not in a straight line but it helps to confirm the accuracy of skew detection and correction process. The method has been applied to the document after skew correction and works perfectly fine in this process and thereafter. Bit map of the rotated image is re-stored so as to apply the segmentation and character recognition algorithms. The results obtained match fairly with the results shown in Fig. 4(b). This is the case when the writer has written on an axis.

The process has been demonstrated with the help of Figs 14 and 15. The highest peak showing maximum energy will be obtained when the word is rotated to the exact horizontal position.

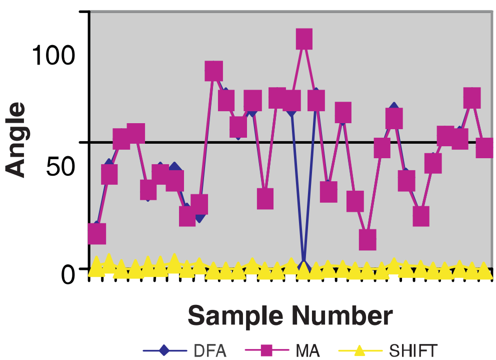


FIG. 16. Calculated vs true angle comparison.

Table II
Mean and standard deviation of estimated skew angles

True skew angle (Manual)	Calculated skew angle (Mean)	Calculated standard deviation (SD)
40	39.548	0.273
20	19.866	0.166
10	10.087	0.256
5	4.955	0.181
2	2.125	0.231

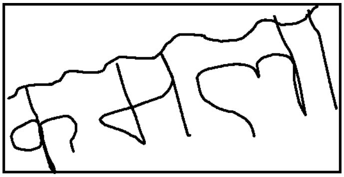


FIG. 17. Distorted sample.

5. Results and conclusion

The detection of skew and its removal thereafter worked very well with 235 samples from 36 persons of different age groups and work profile, containing words of Devanagari script (Fig. 16 and Table II). The method does not however work with samples having different kind of distortion and it is difficult to exploit the Shiro-Rekha of the Devanagari script (Fig. 17). This sample can be treated as one without the Shiro-Rekha and the skew can be detected by the steps mentioned in Section 3. Time taken for tilt removal was less than 1 ms in all the samples.

Figure 17 compares the physically measured angle and the angle measured by using the above-mentioned approach. It shows the dissimilarity only at one point where the angle of the word image is 90^{o}.

We have taken up a similar work with five different angles and the results are equally good. In addition, the document can accommodate the error of scanning faster when tested on a Pentium-IV (Dell machine) 800 MHz. The mean and the standard deviation calculated are shown in Table II. The method exploits the topline to its maximum and is valid for the handwritten documents where the words are not written on the same reference line.

References

1. H. -S. Park and S. -W. Lee, Off-line recognition of large-set handwritten characters with multiple hidden Markov models, *Pattern Recognition*, **29**, 231–244 (1996).

2. H. J. Kim and P. K. Kim, Recognition of off-line handwritten Korean characters, *Pattern Recognition*, **29**, 245–254 (1996).

3. A. W. Senior and A. J. Robinson, An off-line cursive handwritten recognition system, *IEEE Trans.* **PAMI-20**, 309–321 (1998).

4. R. C. Gonzalez and R. E. Woods, *Digital image processing*, Person Education Asia (1993).

5. G. Kim and V. Govindaraju, A lexicon driven approach to handwritten word recognition for real-time applications, *IEEE Trans.*, **PAMI-19**, 366–379 (1997).

6. H. S. Hou, *Digital document processing*, Wiley (1983).

7. H. S. Baird, The skew angle of printed documents, *Proc. Soc. Photogr. Sci. Engng*, **40**, 21–24 (1987).

8. T. Akiyama and N. Hagita, Automatic entry system for printed documents, *Pattern Recognition*, **23**, 1141–1154 (1990).

9. T. Pavlidis and J. Zhou, Page segmentation and classification, *Computer Vision Graphics Image Processing*, **54**, 434–496 (1992).

10. H. Yan, Skew correction of document images using interline cross-correlation, *CVGIP: Graphical Models Image Processing*, **55**, 538–543 (1993).

11. 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. Conf. on Pattern Recognition*, Vol. 1, pp. 464–468 (1990).

12. D. S. Le, G. R. Thoma and H. Wechsler, Automatic page orientation and skew angle detection for binary document images, *Pattern Recognition*, **27**, 1325–1344 (1994).

13. S. N. Srihari and V. Govindaraju, Analysis of textual images using the Hough transform, *Machine Vision Applic.*, **2**, 141–153 (1989).

14. U. Pal and B. B. Chaudhuri, An improved document skew angle estimation technique, *Pattern Recognition Lett.*, **17**, 899–904 (1996).

15. W. Postl, Detection of linear oblique structures and skew in digitized documents, *Proc. Eighth Int. Conf. on Pattern Recognition*, pp. 464–468 (1986).

16. A. Hashizume, P. S. Yeh and A. Rosenfeld, A method of detecting the orientation of aligned components, *Pattern Recognition Lett.*, **4**, 125–132 (1986).

17. L. O'Gorman, The document spectrum for page layout analysis, *IEEE Trans.*, **PAMI-15**, 1162–1173 (1993).

18. B. B. Chaudhuri and U. Pal, Skew angle detection of digitized Indian script documents, *IEEE Trans.*, **PAMI-19**, 182–186 (1997).