IJDAR (2001) 4: 109–114

International Journal on Document Analysis and Recognition IJDAR
© Springer-Verlag 2001

A fast algorithm for skew detection of document images using morphology

A.K. Das[1], **B. Chanda**[2]

[1] Computer Science and Technology Department, Bengal Engineering College, Howrah 711 103, India
[2] Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700 035, India

Received August 27, 2000 / Revised November 15, 2000

Abstract. Any paper document when converted to electronic form through standard digitizing devices, like scanners, is subject to a small tilt or skew. Meanwhile, a de-skewed document allows a more compact representation of its components, particularly text objects, such as words, lines, and paragraphs, where they can be represented by their rectilinear bounding boxes. This simplified representation leads to more efficient, robust, as well as simpler algorithms for document image analysis including optical character recognition (OCR). This paper presents a new method for automatic detection of skew in a document image using mathematical morphology. The proposed algorithm is extremely fast as well as independent of script forms.

Keywords: Document processing – Mathematical morphology – Skew detection – OCR – Text segmentation

1 Introduction

Document processing has gained the attention of researchers and industries in recent times due to its immense potential in commercial applications. One of the major tasks of document image analysis is the segmentation of its different components, namely text, graphics, and half-tones. This is done for subsequent processing of the text portion by OCR, vectorization of graphics, and compression of the half-tones using suitable methods. Many document processing algorithms expect a de-skewed document. Meanwhile, we know, that a few degrees of skew (tilt) is inevitable in a scanned document. This is true for manual as well as automatic handling of the digitization process. Acting on a skewed document, the processing algorithms may not be able to produce a satisfactory result. Usually, a more complex algorithm is required to achieve satisfactory output, for the simple reason that a de-skewed document allows a more compact representation of its components, particularly text objects, such as words, lines, and paragraphs. The compact representation leads to more efficient, robust, as well as simpler algorithms for document image analysis. A number of good skew estimation algorithms are available in the literature. However, the time required to estimate the skew angle is still an important issue.

Correspondence to: A.K. Das

1.1 Existing methods

Most of the algorithms for skew angle estimation presented in the literature may be classified into three groups: (i) the projection profile method; (ii) clustering of nearest neighbors; and (iii) Hough transform. In the projection profile technique [9] the document is projected at different angles. Then peaks (due to text line positions) and troughs (due to inter-line gap positions) are identified. The angle which gives the maximum difference between the peaks and troughs is accepted as the skew angle. The method, being computationally expensive, has been improved by Baird [3] by proposing a quick convergence of this iterative approach. In another approach Akiyama and Hagita [2] partitioned the image vertically into a number of strips. The horizontal projection profile is then calculated for each strip where, from the correlation of the profiles of the neighboring strips, the skew angle is estimated. This yields fast but not so accurate result. Pavlidis and Zhou [18] have proposed a method where the image is striped horizontally and then a vertical profile of each horizontal strip is computed; the method works well for small skew angles. Cross correlation between lines at a fixed distance 'd' shifted by a variable amount 's' in the vertical direction is used by Yan [23]. The accumulated values of the correlations between all pair of lines is computed and the shift for which the maximum takes place is used for skew determination. Projection profile methods are, in general, well suited to estimate skew angle within ±10 degrees.

Nearest neighbor clustering has been used by Hashizume [7] and Jiang et al. [10]. Hashizume computed the directions of the nearest neighbors of all the pixels in each connected component of the image. A his-

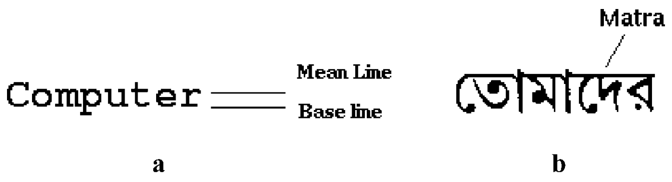


Fig. 1a. Mean line and base line **b** 'Matra' joining the characters of a 'Bangla' word

togram of the directions is constructed which indicates the skew angle from its peak. This method is generalized by O'Gorman [14].

Clustering of pixels along the mean line and base line (see Fig. 1) has been successfully used by Pal and Chaudhuri [16] where component labeling is used to compute the average height of the characters. Components with a small number of pixels (like the dot over the i, commas, etc.) as well as tall characters (i.e., capital letters and characters with ascenders and descenders, e.g., d, t, g, y, etc.) whose heights are more than the average are discarded. This leads to fairly accurate identification of the pixels lined up along the mean line. An average of the skew value computed from the mean line skew values is the skew of the document. This algorithm is faster than many other reported algorithms. The algorithm proposed by Ma and Yu [13] is also based on detection of the base line.

Several techniques are available [8,12,22] which use Hough transform to find out the skew angle. The major problem with Hough transform is the massive computational cost. Modifications of the Hough transform approach are also proposed in [8,12] that discard irrelevant pixels in order to obtain accurate transform peaks and reduce the processing time. For example, in Le et al. [12] only the bottom pixels of each of the connected components are used for a subsequent Hough transform to find out the skew.

Some other techniques also use special characteristics of the alphabets and the text lines of a particular language [17]. For the same reason every skew detection algorithm may not be applicable to all script forms. For example, the nearest neighbor clustering method may not produce correct results for documents containing *Devnagari or Bangla* scripts where the characters in a word are mostly connected through a head line called *matra* (see Fig. 1b). For the same reason Le's [12] method is not applicable for such scripts,as the subsequent Hough transform with very few bottom pixels may not produce correct peaks. Note that in printed *English or Roman* script the characters do not touch each other. Here we have presented a morphology-based fast algorithm for detection of the skew angle of the document image; the algorithm has the additional advantage of being script independent. It works with the assumption that the text lines are supposed to be oriented horizontally, i.e., the reading direction is either from left to right or vice versa. This is true for most of the scripts as well as for all other algorithms as referred to in this paper. It may be noted that morphology has already been used in finding out the skew of document pages [4]; however, the algorithm is not that fast compared to the proposed one. Skew angle is determined based on well-defined structures of the text portion in a document. Our algorithm, like others, expects documents mostly filled with text lines. However, it can handle documents with a moderate amount of graphics. The algorithm has some steps to neutralise the effect of graphics in skew detection. The proposed algorithm works well for small skew angles (less than 10 degrees). Although skew detection methods without such restrictions are reported in the literature [15,16],in reality the skew seldom exceeds ± 5 degrees [19].

The paper is organized as follows: Sect. 2 describes the proposed algorithm and the speedup techniques. Section 3 reports the experimental results and comparison.

2 Proposed algorithm

In this work we have used mathematical morphological operations, i.e., dilation, erosion, opening, and closing [21] to smear the text lines to solid black bands and to remove the bumps in the bands. We have also adapted the common binary image attributes like connectivity, connected components, etc., from [20]. Note that the morphological operations are inherently slow when executed in sequential machines. We have used fast algorithms as proposed in [1]. We have also formulated special algorithms for dilation (erosion) and component labeling for further speedup of execution. The algorithm in steps is given below:

1. Close the image with line structuring element (SE) to form solid black bands corresponding to each text line. Some portion of the graphics may form large blobs due to this operation.
2. Open the image (closed) with small square SE to remove bumps formed due to the presence of ascenders and descenders, etc. Lines and arcs of graphics are also removed, consequently.
3. Scan the opened image vertically to register all 1 to 0 transitions i.e., base-line pixels of the text lines. Some additional small curves/lines may appear due to blobs retained after the previous step.
4. Use component labeling to select lines whose lengths are greater than a threshold. In addition, remove the lines which are not straight [20].
5. Identify two points in those lines to prune some portions from the beginning and the end, respectively.
6. Find out the skew of all those lines and declare the median value as the skew of the entire document.

Note that lines generated in step 3, due to the presence of graphics, are not considered in step 6 because of steps 4 and 5.

The proposed algorithm uses morphological closing operations to smear the text lines to black bands using a line SE of length l pixel, say. Depending on the font size and scanner resolution, in our experiment. we have used $l = 12$. The black bands will have bumps above the mean lines and below the base lines due to the presence of ascenders and descenders in characters. To eliminate these bumps the closed image is opened with small rectangular SE of size 5×5. This choice of SEs for closing

Table 1. % of characters with ascenders and descenders in English text

Ascenders		Descenders	
Single	Pairwise	Single	Pairwise
19.40	14.89	4.00	0.71

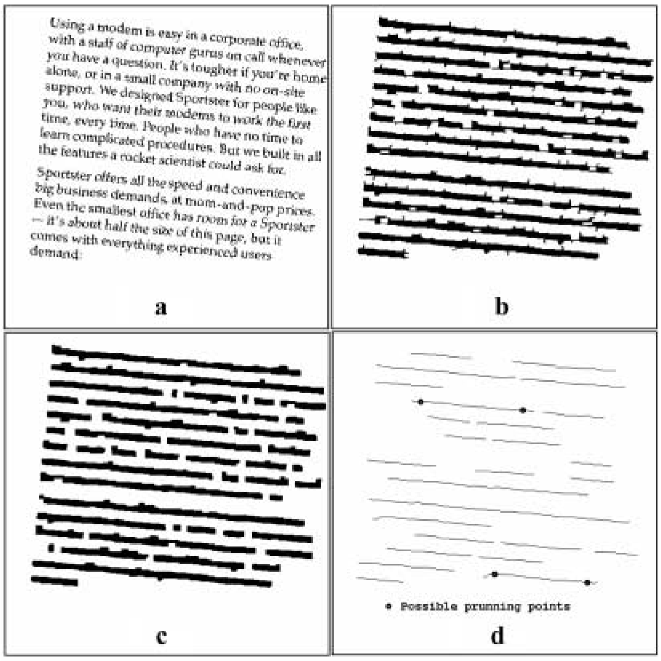
and opening give good results for commonly used fonts like 10pt–16pt and 300 dpi scanner resolution. After closing and opening we get black bands of (approximately) uniform thickness separated with inter-line gaps. A vertically scanned 1 to 0 transition is mapped as a new image. This gives the outline of the bottom portion of each black band (smeared text line) as line elements (1-pixel-thick lines). There exists discontinuity in the lines due to the presence of projected rectilinear portions (due to the occurrence of multiple characters with descenders) in the lower side of the bands, but on an average we get lines oriented to the skew of the text. Note that the presence of bumps in the resultant black bands cause breaks in the bottom profile (i.e., base lines) of the text lines. This may lead to short segments and inaccurate skew angles. The bottom portion (i.e., the base line) is selected deliberately as we have seen less undulation in the bottom portion because of the following reasons:

1. The number of descenders in a text file is relatively less than the number of ascenders. The pecentage occurrence of ascenders and descenders present in text files is statistically studied by examining 50 text files of various sizes. We have also computed the same for repeated appearance (twice) of possible ascenders and descenders. Table 1 corroborates the said phenomenon.
2. In many scripts (including English) the characters are aligned along the base line.

We intend to select large lines as they give better estimation of the skew. This is done in step 4 by component labeling. Lines that are not so straight are also filtered out [20]. To achieve further accuracy we first prune the lines up to a length equal to the width of the SE from the start and end of each line, as we have seen distortions in those parts because of curvatures in the character body. We then take only those lines whose lengths are greater than a predefined threshold (at least 4× the width of SE) and compute the skew of each line. We finally compute the skew by taking the median of the skew values computed for the large lines selected and processed in step 4 and 5. The whole process is shown in Fig. 2.

2.1 Speedup with specially formulated dilation and component labeling

The time to compute the skew can be substantially reduced by using the operations dilation and erosion in lieu of closing and opening, respectively. Performance will not be degraded in the context of the said algorithm. For example, the objective of closing here is to smear the text characters and words to form the black


Using a modem is easy in a corporate office, with a staff of computer gurus on call whenever you have a question. It's tougher if you're home alone, or in a small company with no on-site support. We designed Sportster for people like you, who want their modems to work the first time, every time. People who have no time to learn complicated procedures. But we built in all the features a rocket scientist could ask for.

Sportster offers all the speed and convenience big business demands, at mom-and-pop prices. Even the smallest office has room for a Sportster — it's about half the size of this page, but it comes with everything experienced users demand:

Fig. 2a. Original document in binary **b** closing original binary image with 4 × 12 SE **c** opening the closed image with 5 × 5 SE **d** after transition and rejection of small lines operation

bands. Dilation can be used in this context as we get the same effect with elongation of components in the horizontal direction. As we later strip off the start and end of each line, dilation does not introduce elongation of end pixels horizontally irrespective of the skew. Opening to smoothen the black bands may be replaced with erosion. Speedup is achieved by formulating a special dilation module suitable to the line structuring element used for smearing. The method is now described.

We assume that the origin of SE is at its left-most pixel. Now, instead of dilating all the foreground pixels in the horizontal direction we align the origin of the SE with a foreground pixel of the input image, and all the pixels along the linear SE are changed to foreground pixels in the dilated image. Then, starting from extreme right pixel of the SE, we traverse towards the left and stop at the first foreground pixel of the input image. Mark this pixel as the next foreground pixel to be dilated if it is not the already considered one; if it is, follow the raster scan order to find the next foreground pixel to be dilated. This expedites the whole process as in the text area occurrence of the foreground pixel is frequent and by the said operation we are directly joining two foreground pixels situated at the farthest points within the line SE. Erosion is dual to dilation, so it is carried out using the same module and complement operation.

To achieve further speedup, the component labeling process is reformulated as we are doing the operation on 8-connected 1-pixel-wide lines. For component labeling we use a vertical scan to detect the left-most pixel of an unlabeled component. Then the remaining portion of the component is labeled as follows: we start with the nearest right pixel because its presence has the highest probabil-

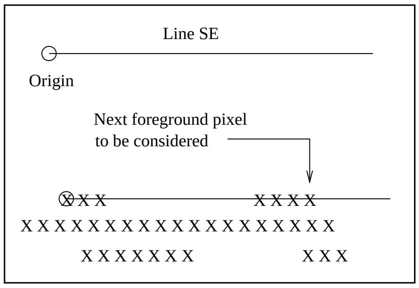


Fig. 3. Dilation with line SE to join two foreground pixels

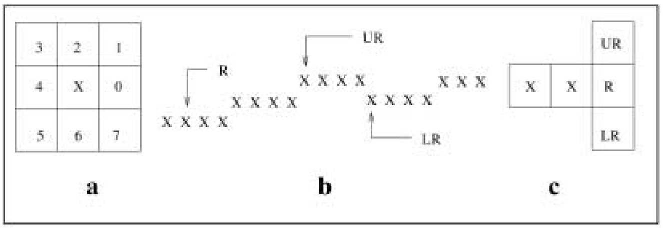


Fig. 4a. 8-neighboring pixels (0–7) around a foreground pixel 'x' **b** representation of a skewed line **c** 3 possible neighbors (R, UR, and LR) of a foreground pixel 'x' of a 1-pixel-wide line object

ity. Next we check the upper-right or lower-right pixel. Note that it is a single scan 3 neighborhood case. The component labeling process also stores the co-ordinates of the end points of each line as well as the points up to which we will prune the lines (see Fig. 2d).

3 Experimental results and discussions

Experiments are carried out in a DEC ALPHA 3000/300X workstation running at 175 MHz under OSF/1 operating system. The scanner used is a HP Scan-jet 3P with a resolution of 300 dpi. All algorithms are written in C and are developed using similar coding sophistication. This is important as we have carried out performance comparisons on a real-time basis instead of a theoretical complexity analysis. We have tested our algorithm on real as well as synthetic documents with varying degrees of tilt angles. Real images are obtained from image databases UW-I and UW-II [19] as well as by scanning documents with a variety of, deliberate, tilts (both clockwise and anti-clockwise directions). Synthetic documents are also created without any tilt using graphics drawing packages. They are then rotated at various, preselected, angles in clockwise and anti-clockwise directions. In order to ascertain the applicability of our algorithm irrespective of scripts, in addition to English, we have used 'Bangla' and 'Devnagari' text documents as well as handwritten documents too. Once the algorithm returns the estimated skew angle we evaluate its performance as follows: for the synthetic documents as well as UW-I and UW-II images with known values of tilts, we have carried out immediate comparisons. For unknown tilt angles, the document is rotated by the negative of the estimated angle. Then horizontal lines are drawn on the de-skewed images using a graphics package and evaluation is made carefully by human beings.

For English script, comparison is done with four other methods. For scripts where letters in words are connected (this includes handwritten scripts) only the Hough approach is used for comparison. Test results for synthetic English script with three different skew angles are given in Table 2.

Table 2. Mean and standard deviation for five different methods (skew angle is given in degrees)

Skew angle	Mean	Mean	Mean	Mean	Mean
3	3.15	3.26	3.51	2.90	3.11
5	4.83	5.21	5.63	5.38	5.14
10	10.32	9.82	10.64	10.53	10.36
Skew angle	SD	SD	SD	SD	SD
3	0.193	0.342	0.562	0.498	0.318
5	0.219	0.390	0.301	0.517	0.212
10	0.118	0.218	0.827	0.631	0.229

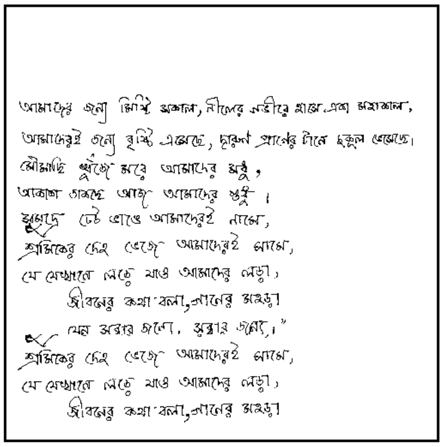
A Hough Transform method **B** Pal and Chaudhuri's method **C** O' Gorman's (Docstrum) method **D** Le's method **E** Proposed morphology-based method

Each class is tested with 25 images using five different methods. Mean and standard deviation are computed. We did not implement any projection profile method as they are in general slow and work well only with low skew angles. Timing comparison is given in Table 3 where the average skew angle of 512 × 512 pixels images are computed by different methods. The unit of time is seconds. The effectiveness of the proposed algorithm in the context of script independence is shown in Fig. 5. The figure shows a 512 × 512 image containing a handwritten 'Bangla' script. The image is rotated at various preselected angles in clockwise and anti-clockwise directions and tested with Hough and the proposed algorithm. The results are given in Table 3. The average time of execution for Hough and the proposed algorithm for the hand printed image is 22.0 and 0.9 s, respectively.

Experimental results are satisfactory both in terms of performance and computational cost. We notice that the results produced by our method are quite comparable in terms of accuracy with that of method A and method B, while it is superior to method C and D. However, our method is extremely fast in comparison to Method A and about 10% faster than method B. Thus, in terms of execution speed, method B is comparable to the proposed method. However, method B has the following drawbacks. As indicated earlier, method B is script dependent. It also depends on correct formation of 'ini-line' [16] for proper clustering of pixels along the base or mean

Table 3. Time required to find out the skew

A	B	C	D	E
33.54	0.984	2.34	5.04	0.887



আমাদের জন্য মিষ্টি সকাল, নীলের গভীরে মাঝে এত মহাকাল,
আমাদেরই জন্য বৃষ্টি এসেছে, দুরন্ত প্রাণের টানে দুকূল ভেসেছে।
মৌমাছি খুঁজে মরে আমাদের মধু,
আকাশ ডাকছে আজ আমাদের শুধু।
সমুদ্রে ঢেউ ওঠে আমাদেরই নামে,
শ্রমিকের দেহ ভেজে আমাদেরই ঘামে,
যে যেখানে লড়ে যাও আমাদের লড়া,
জীবনের কথা বলা, গানের মহড়া
যেন সবার জন্যে, সবার জন্যে।"
শ্রমিকের দেহ ভেজে আমাদেরই ঘামে,
যে যেখানে লড়ে যাও আমাদের লড়া,
জীবনের কথা বলা, গানের মহড়া

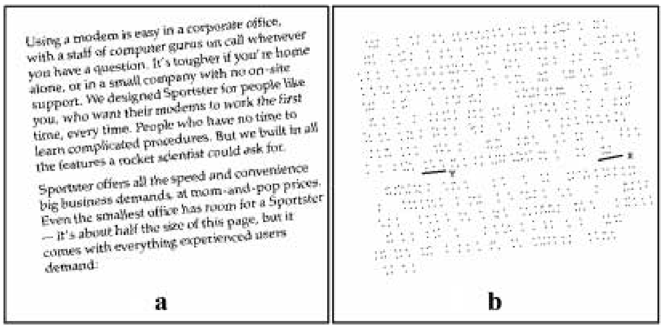
Fig. 5. A 512 × 512 handwritten image in 'Bangla' script

line. As 'ini-line' is formed by a randomly chosen seed point and two more consecutive points in the same direction, the angle between 'ini-line' and the horizontal axis (i.e., the skew of the text) may be widely different from the actual skew of the text. This leads to wrong clustering, and the estimated skew returned by the algorithm is likely to be erroneous. This happens particularly when the characters in the text lines are a bit deformed as is commonly found in worn-out documents. An example is shown in Fig. 6. The true skew angle is 5 degrees but due to wrong selection of the 'ini-line' (line X in the figure with an angle of 11 degrees) the clustering becomes faulty and the estimated skew angle returned is 10 degrees. Pal and Chaudhuri's algorithm is also tested with synthetically degraded documents. We have used the degradation model as reported in Kanungo [11]. The original document and degraded document are shown in Fig. 7 where some of the foreground pixels of the original document are deleted to simulate degradation. The degraded document is tested with our algorithm as well as Pal and Chaudhuri's algorithm. While our method returns the same skew angle for both the images, Pal and Chaudhuri's algorithm fails for degraded image and produces wrong results. The reason for the failure is quite obvious. It is due to the random breaks (discontinuities) in the characters. On the other hand, degradation has practically no effect on the proposed method, as we are smearing the text to form the black bands thereby filling all the discontinuities. It may also be noted that our algorithm can also handle documents with random additive noise, as we are opening the document with a small structuring element (step 2 of our algorithm). This step effectively erases stray foreground pixels.

We have noticed that in case of very high skew angles (say ≥ 15 degrees) the result of our method is inferior to that of other methods, e.g., Pal and Chaudhuri's algorithm can work up to a 45-degree skew. This is obvious as we approximate the skewed lines by short straight-line segments. Second, for a large skew, the line SE may create a bridge between text lines leading to the failure of the algorithm. However, for all practical purposes, the general skew angle is limited to ±5 degrees which is well within the operational limit of the proposed algorithm.

Table 4. Test results for a handwritten document in 'Bangla' script

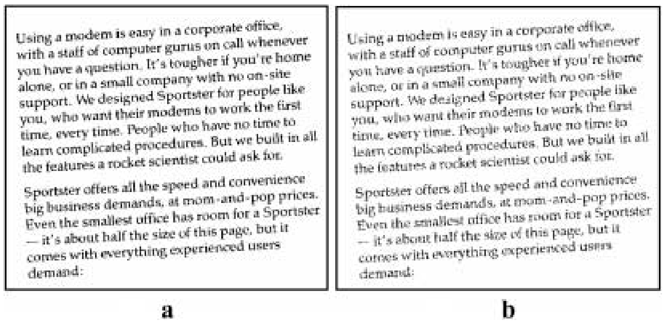
Angle of rotation	Hough transform (A)	Morphology-based (E)
0	1.0	0.7
+5	6.0	4.92
−5	+6	+5.44
+10	11.0	10.93
−10	+9.0	+9.74



Using a modem is easy in a corporate office, with a staff of computer gurus on call whenever you have a question. It's tougher if you're home alone, or in a small company with no on-site support. We designed Sportster for people like you, who want their modems to work the first time, every time. People who have no time to learn complicated procedures. But we built in all the features a rocket scientist could ask for.

Sportster offers all the speed and convenience big business demands, at mom-and-pop prices. Even the smallest office has room for a Sportster — it's about half the size of this page, but it comes with everything experienced users demand:

Fig. 6a. Distorted document image **b** top-left and bottom-right points of each component; two lines X and Y are drawn to show the points considered for forming 'ini-line' and a line with actual skew, respectively



Using a modem is easy in a corporate office, with a staff of computer gurus on call whenever you have a question. It's tougher if you're home alone, or in a small company with no on-site support. We designed Sportster for people like you, who want their modems to work the first time, every time. People who have no time to learn complicated procedures. But we built in all the features a rocket scientist could ask for.

Sportster offers all the speed and convenience big business demands, at mom-and-pop prices. Even the smallest office has room for a Sportster — it's about half the size of this page, but it comes with everything experienced users demand:

Using a modem is easy in a corporate office, with a staff of computer gurus on call whenever you have a question. It's tougher if you're home alone, or in a small company with no on-site support. We designed Sportster for people like you, who want their modems to work the first time, every time. People who have no time to learn complicated procedures. But we built in all the features a rocket scientist could ask for.

Sportster offers all the speed and convenience big business demands, at mom-and-pop prices. Even the smallest office has room for a Sportster — it's about half the size of this page, but it comes with everything experienced users demand:

Fig. 7a. Original document image **b** synthetically degraded image

The present algorithm works well for small skew angles (say, ±3 degrees). This is obvious as the inter-line bridge formation depends on the length of the SE, the skew, and the inter-line gap. The chance of bridge formation increases if the skew is large and inter-line gap is small.

We would like to conclude by suggesting ways to adapt the proposed method for skew detection in documents containing large amounts of graphics and half-tones. For example, half-tones can be separated out using the methods suggested in [6,5]. Component labeling may be used to delete large graphical components from the image. We understand that the texts and small elements present after component labeling in the graphics area will not be able to influence the accuracy of the proposed algorithm.

Acknowledgements. We express our indebtedness for helping us in various ways to Ms. Senty Sarkar, Mr. Nilanjan Deb, and

Mr. Pinaki Chakraborty. We appreciate very much the help rendered by Mr. U. Pal, CVPR Unit, Indian Statistical Unit, Calcutta, as he has supplied the source code for implementing several skew detection algorithms. We are also grateful to Dr. A. M. Ghosh for his support and constant encouragement.

References

1. A.K. Das, B. Chanda: Fast algorithms for morphological operations for sequential machines. In: Proc. 3rd Int. Conf. on Advances in Pattern Recognition and Digital Techniques, Calcutta, India, pp. 258–266, 1993
2. T. Akiyama, N. Hagita: Automated entry systems for printed documents. Pattern Recognition, 23, 11:1141–1153, 1990
3. H.S. Baird: The skew angle of printed docuemnts. In: Proc. Soc. Photogr. Sci. Eng., 40:21–24, Rochester, N.Y., USA, 1987
4. S. Chen, R. Haralick, I. Phillips: Automatic text skew estimation in document images. In: Proc. 3rd Int. Conf. on Document Analysis and Recognition, pp. 1153–1156, Montreal, Canada, Aug., 14–16 1995
5. A.K. Das, B. Chanda: Text segmentation from document images: a morphological approach. J. Inst. Eng. (I), 77:50–56, 1996
6. K.C. Fan, C.H. Liu, Y.K. Wang: Segmentation and classification of mixed text/graphics/image documents. Pattern Recognition Lett., 15:1201–1209, 1994
7. A. Hashizume, P.S. Yeh, A. Rosenfeld: A method for detecting the orientation of aligned components. Pattern Recognition Lett., 4:125–132, 1986
8. S. Hinds, J. Fisher, D.P. D'Amato: A document skew detection method using run length encoding and the Hough transform. In: Proc. 10th Int. Conf. Pattern Recognition, Atlanta, Ga., USA, pp. 464–468, 1990
9. H.S. Hou: Digital Document Processing. Wiley, New York, 1983
10. X. Jiang, H. Bunke, D. Widmer-Kljajo: Skew detection of document images by focused nearest-neighbor clustering. In: Proc. ICDAR99 5th Int. Conf. on Document Analysis and Recognition, pp. 629–632, Bangalore, September 20–22, 1999
11. T. Kanungo, R.M. Haralick, I. Phillips: Global and local document degradation models. In: Proc. 2nd. Int. Conf. on Document Analysis and Recognition, pp. 730–734, Tsukuba, Japan, October, 1993
12. D.S. Le, G.R. Thoma, H. Wechsler: Automatic page orientation and skew angle detection for binary document images. Pattern Recognition, 27:1325–1344, 1994
13. H. Ma, Z. Yu: An enhanced skew angle estimation technique for binary document images. In: Proc. ICDAR99 5th Int. Conf. on Document Analysis and Recognition, pp. 165–168, Bangalore, September 20–22, 1999
14. L. O'Gorman: The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell., 15:1162–1173, 1993
15. O. Okun, M. Pietikainen, J. Sauvola: Document skew estimation without angle range restriction. Int. J. Doc. Anal. Recognition 2:132–144, 1999
16. U. Pal, B.B. Chaudhuri: An improved document skew angle estimation technique. Pattern Recognition Lett., 17:899–904, 1996
17. U. Pal, B.B. Chaudhuri: Skew angle detection of digitized Indian script documents. IEEE Trans. Pattern Anal. Mach. Intell., Accepted for publication, 1996
18. T. Pavlidis, J. Zhou: Page segmentation and classification. Comput. Vision Graph. Image Process., 54:484–496, 1992
19. I.T. Phillips, S. Chen, R.M. Haralick: Cd-rom document database standard. In: ICDAR93, pp. 478–483, 1993
20. A. Rosenfeld, A.C. Kak: Digital Picture Processing, vol. II. Academic, N.Y., USA, 1982
21. J. Serra: Image Analysis and Mathematical Morphology. Academic, London, UK, 1982
22. S.N. Srihari, V. Govindraju: Analysis of textual images using the Hough transform. Mach. Vision Appl., 2:141–153, 1989
23. H. Yan: Skew correction of document images using interline cross-correlation. CVGIP: Graph. Models Image Process., 55:538–543, 1993

Amit Kumar Das (1957) was born in Calcutta and completed his Masters and PHd degree from Calcutta University and Bengal Engineering College respectively. He is presently an Assistant Professor in the Computer Science and Technology Department, Bengal Engineering College(D.U). His field of interest includes embedded system design and document image processing.
(e-mail: amit@becs.ac.in)

Bhabatosh Chanda Born in 1957. Received B.E. in Electronics and Telecommunication Engineering and PhD in Electrical Engineering from University of Calcutta in 1979 and 1988 respectively. Received "Young Scientist Medal" of Indian National Science Academy in 1989 and "Computer Engineering Division Medal" of the Institution of Engineers (India) in 1998. He is also recepient of UN fellowship, UNESCO-INRIA fellowship and fellowship of National Academy of Science, India during his carrier. He worked at Intelligent System lab, University of Washington, Seattle, USA as a visiting faculty from 1995 to 1996. He has published more than 50 technical articles. His research interest includes Image Processing, Pattern Recognition, Computer Vision and Mathematical Morpholgy. Currently working as Professor in Indian Statistical Institute, Calcutta, India. (e-mail: chanda@isical.ac.in)