Document Skew Detection Using Minimum-Area Bounding Rectangle

Reza Safabakhsh and Shahram Khadivi
Computer Engineering Department
Amirkabir University of Technology, Tehran, Iran
Email: safa@ce.aku.ac.ir , khadivi@ce.aku.ac.ir

Abstract

Detection of document skew is an important step in document image analysis. This paper presents a new method for calculation of document skew. The method forms large connected components by a smoothing algorithm and calculates the document skew by finding the orientation of the minimum-area bounding rectangle of one or several connected components. Connection of text to non-text in the smoothing step does not degrade the performance of the method. The smoothing parameters are determined automatically and no manual adjustment is necessary. The method is not limited in the range of detectable skew angles and the achievable accuracy. Experimental results show the high performance of the algorithm in detecting document skew for a variety of documents with different levels of complexity.

1. Introduction

Computer analysis of documents seeks to decompose a document image into its different structural units such as text, graphics, tables, images, etc., and to specify the logical relations between units to indicate the order of reading the document regions.

Document image analysis generally requires that the document images do not include any skew. Producing skew-free images through scanning, however, is a difficult task and introduction of small skews in the image seems to be inevitable. As a result, a document skew detection and correction is required before the actual analysis of the document starts. This is an important step because otherwise, the document skew will adversely affect the analysis steps that follow.

Researchers have spent a good deal of effort on the problem of fast and robust document skew detection. These efforts may be classified into the following groups:

1. Projection-based Methods: The main idea in these methods is that the projection profiles show more variations when the profile is in the direction of document skew. Thus, a function of the profile magnitude variations for different angles is used to determine the direction for the maximum variations. Among the work presented in this category are [1], [2], [3], and [4].
2. Hough-Transform-based Methods: The document skew is actually equal to the skew of the document text lines. As a result, these methods use the Hough algorithm on the text lines in the document image to determine the direction of document skew. Several efforts of this category are [5], [6], and [7].
3. Nearest-Neighbor-based Methods: In these methods the angle of each connected component and its nearest neighbor(s) is calculated and a histogram of these angles is formed. The document skew is then determined from the largest peak in the histogram. Examples of this approach are discussed in [8], and [9].
4. Cross-Correlation-based Methods: These methods calculate the document skew by finding the amount of vertical shift needed to maximize cross-correlation between pairs of narrow vertical columns of the document. Of these efforts are [10], [11], and [12].
5. Other Methods: Other efforts for the detection of document skew include using the Fourier Transform [13], using morphological transforms [14], and using local region complexity [15].

In this paper, we present a new technique for document skew detection. The method is based on finding the direction of minimum area bounding rectangle of connected components of the text. After the image is binarized, large connected components are formed by a smoothing algorithm. A number of these components with largest length-to-height ratios are then selected, and the document skew is determined from the angle of minimum-area bounding rectangle of the selected component(s).

In section 2, we discuss the proposed algorithm in detail. Section 3 presents the experimental results obtained, and section 4 concludes the paper.

2. The proposed algorithm

The basic idea of the minimum-area bounding rectangle method is that the area of the bounding rectangle of each large connected component of a text line has the minimum area when the component has no skew with respect to the rectangle major axis. As such, close letters or words are first connected together to form large connected components. Then, the angle of the bounding rectangle of each component with the minimum area is found and the document skew is determined. The proposed algorithm consists of the following steps:

1. Conversion of the document image to binary,
2. Run length smoothing of the binary image,
3. Selection of connected components,
4. Finding the angle of the bounding rectangle which has the smallest area for selected connected components,
5. Determining the document skew angle.

2.1. Image binarization

Since the method is not sensitive to minor details in the words or graphics, the binarization technique chosen is not very critical. We used the iterative technique given by Parker [16].

2.2. Smoothing

In run-length smoothing of the document, the image is scanned and any string of white pixels shorter than a certain threshold is converted to black. This way, close black components are connected to form a larger component. Scanning the image can be horizontal with a threshold value T_h or vertical with a threshold value T_v. Due to the inter-line spacing in the vertical direction, the above two thresholds are usually different.

For the purpose of the document skew detection, horizontal smoothing alone may suffice in many cases. However, in order to reduce sensitivity of smoothing to the threshold values and reduce the possibility of the connection of close text and non-text regions, Wong et al. proposed a four-step algorithm [17] which was later improved by Shih et al. [18] into the following two step algorithm:

1. A vertical smoothing is applied to the original document image by a threshold T_v.
2. If the run-length of white pixels in the horizontal direction of the original image (denoted by RL) is greater than T_h, then set the corresponding pixels in the output of step 1 to white pixels. If $RL \le T_h'$, then set the corresponding pixels in the output of step 1 to black pixels. If $T_h' < RL < T_h$ and the run-length of horizontally consecutive white pixels in the output of step 1 is less than or equal to T_h', then set the corresponding pixels in the output of step 1 to black.

The threshold values have to be determined for each document independently. These values can be selected experimentally or automatically by the system. For automatic determination of these values, we devised a technique based on a statistical method that has been used by Avanindra et al. [11] to decrease the time required by the basic time-consuming cross-correlation method given in [10]. For automatic determination of smoothing thresholds, we select N windows of $M_1 \times M_2$ size from the image randomly. In each window, the mean value of white runs in the horizontal and vertical directions are calculated and called w_h and w_v, respectively. The estimated mean of the white runs are then taken as the median of the nonzero values of w_h and w_v estimated from the N windows as:

$$W_h = \text{median}_i\ w_h(i), \qquad W_v = \text{median}_i\ w_v(i). \tag{1}$$

Since the median of the mean values is used in (1), the method is appropriate for documents in which more than 50% of the document includes text. The number of windows, N, can be determined from the probability of obtaining true estimate of white runs w_h and w_v [11]:

$$P = \sum_{i=0}^{N/2} \binom{N}{i} p^{N-i} (1-p)^i \tag{2}$$

in which p is the ratio of the text to entire document areas. For example, for $P \ge 0.99$ and $p = 70\%$, the number of windows should be at least equal to 30.

To determine the threshold values, we set M_1 to $1/12$ of the document height and M_2 to $1/12$ of the document width. Then by experimenting with a large number of documents, we obtained the following relations for the threshold values:

$$T_h = 4W_h,\ T_v = 7W_v, \text{and } T_h' = W_h \tag{3}$$

The threshold values required for smoothing any document will be calculated automatically using the above relations. Thus, no manual parameter adjustment is necessary in the algorithm.

2.3. Connected components specification

Specifying the connected components of the smoothed image is the next step in the proposed algorithm. The technique used here is the sequential algorithm given in [19] with some modifications. The modifications include changing the 4-neighborhood of the algorithm to 8-neighborhood and implementing it so that only one time

scanning of the image is necessary. As such, the algorithm is as follows:

1. Scan the image from left to right, top to bottom.
2. If the pixel is black, then
   a) If only one of its neighbors has a label, then copy the label.
   b) If all neighbors have the same label, then copy the label.
   c) If there are at least two neighbors with different labels, copy the label of the first neighbor (the upper and left-most neighbor) and equalize the label of the other neighbors to that of the first neighbor.
   d) Otherwise assign a new label to this pixel.
3. If there are more pixels to consider, then go to step 2.
4. Delete the unused labels.

The neighbors of a pixel used in this algorithm are the upper and left, the upper, the upper and right, and the left neighbors. Each label L, is specified by a four-tuple L = (*label, newLabel, p1, p2*), where *p1* and *p2* are the two non-adjacent corners of the bounding rectangle of pixels with the same label, and *newLabel* may be set to a new label in the process of equalizing labels. Equalizing a label to another consists of setting the *newLabel* of the current label to that of the target label and extending the rectangle of target label to include the bounding rectangle of the current label. The role of *newLabel* is to make links between the equalized labels. The unused labels are those which are equalized to other labels. Note that in equalizing a label to a target label, the link of the target label is followed until a label which is not equalized to another one is found. With the above approach, instead of an additional scanning of the image for equalizing labels as given in [19], we scan the array of labels, which is much smaller than the image.

2.4. The minimum-area rectangle

The minimum-area bounding rectangle can be found by calculating the area of the bounding rectangle at different orientations. The algorithm is applied to an input connected component twice. An estimation of the rectangle orientation is calculated with a large range and step size during the first iteration of the algorithm. In the second iteration, the final orientation of the rectangle is found with any desired angular resolution in a small neighborhood about the estimated angle. The algorithm is as follows:

1. Initialize *rotAngle* to zero, *minArea* to the area of the bounding rectangle of the connected component at angle *rotAngle,* and $\alpha_{\min}$, $\alpha_{\max}$, and $\Delta\alpha$ to their desired values. Consider the origin at the center of the rectangle.
2. For different values of α from $\alpha_{\min}$ to $\alpha_{\max}$ with a resolution of $\Delta\alpha$ and $\alpha \neq 0$ repeat steps 3 and 4.
3. The area of the bounding rectangle in direction α can be calculated from
$$area = (h_1 - h_2)\cdot(h_3 - h_4)\cdot\left|\cos\alpha\cdot\sin\alpha\right| \qquad (4)$$
where h_1 and h_2 are the maximum and minimum intercepts from origin of the lines having a slope of $\tan\alpha$ and passing through any boundary pixel of the connected component; and h_3 and h_4 are the maximum and minimum intercepts from origin of the lines having a slope of $-1/\tan\alpha$ and passing through any boundary pixel of the connected component.
4. If *area*<*minArea* let *minArea* = *area* and *rotAngle* = α.
5. The rotation angle of the connected component equals to *rotAngle*.

The boundary pixels of each connected component required for steps 3 are detected by the boundary-following algorithm given in [19]. Figure 1 shows the formation of the bounding rectangle for a connected component.

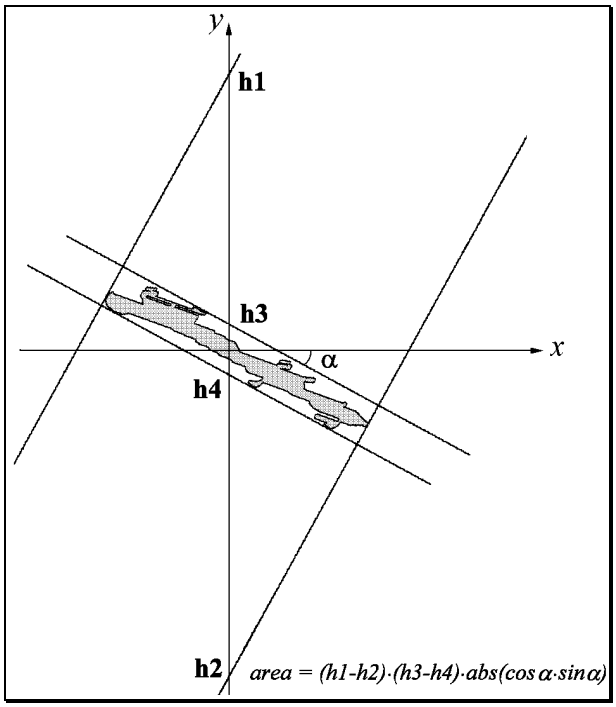

Figure 1. The geometry of calculating the minimum-area bounding rectangle

In order to evaluate the accuracy and speed of the above algorithm, we also calculated the orientation of connected components by finding the axis of the least inertia through [19]:

$$\theta = \frac{1}{2}\tan^{-1}\frac{2\sum_{i=1}^{n}\sum_{j=1}^{m}(x-\bar{x})(y-\bar{y})B[i,j]}{\sum_{i=1}^{n}\sum_{j=1}^{m}(x-\bar{x})^2 B[i,j]-\sum_{i=1}^{n}\sum_{j=1}^{m}(y-\bar{y})^2 B[i,j]} \quad (4)$$

in which $\bar{x}$ and $\bar{y}$ are coordinate of connected component centroid, and $B[i,j]$ is 1 for pixels on the boundary of the connected component and 0 otherwise. The results of two methods were compared.

2.5. The document skew angle

The document skew angle can be specified in one of the following ways:

1. Set the skew angle equal to the rotation angle of one of the selected components,
2. Set the skew angle equal to mean value of the rotation angles of the selected components,
3. Set the skew angle equal to median value of the rotation angles of the selected components,

This concludes the proposed algorithm. In the next section, we present the results obtained from implementation of the method.

3. Experimental results

A number of documents with different levels of complexity from newspapers, journals, and books were selected to carry out experiments. These documents include diagrams, images, formulas, and single and multi-column English, Arabic, or Farsi text. The documents were rotated at different angles between $[-10^o, 10^o]$ and were digitized at 100 dpi. After binarization, the images were smoothed by the proposed algorithm, the result of which for one of the documents is shown in Figure 2.

In the first experiment, ten connected components were selected from each smoothed image. The basis for this selection was the ratio of the length-to-height of the connected components. The ten components with largest length-to-height ratios were individually used to determine the document skew. The skew angle range, the large and small steps sizes, and the neighborhood radius were taken as $[-10^\circ, 10^\circ]$, 1°, 0.1°, and 0.9^o, respectively. The algorithm has no limitation concerning these values and the angular resolution can be modified as desired. The computed skew angles are shown in Table 1. In addition, the mean, median, absolute maximum error, and the absolute error of the mean and median values for the ten components of each document are shown in the table. The following conclusions were drawn from this experiment:

1. The skew angles calculated from any of the connected components with larger length-to-height ratio are generally good estimates of the actual skew angle. The connected component with the largest ratio, except for rare cases, gives very good results.
2. In the 80 measurements given in the table, only 4 cases have an error larger than 1° and 6 cases have errors larger than 0.5°. Most of errors are observed in skews of documents 6 to 8 which occur due to the nature of these documents which are Arabic and Farsi documents, and not English.
3. To make the method more robust, we should calculate the skew based on several connected components. In this case, the median of the skew angles computed for the connected components can be declared as the document skew. Although the median values sometimes include slightly higher errors than the mean, they produce more robust results.
4. The algorithm will work properly, even if some text sections are connected to non-text sections by the smoothing algorithm.

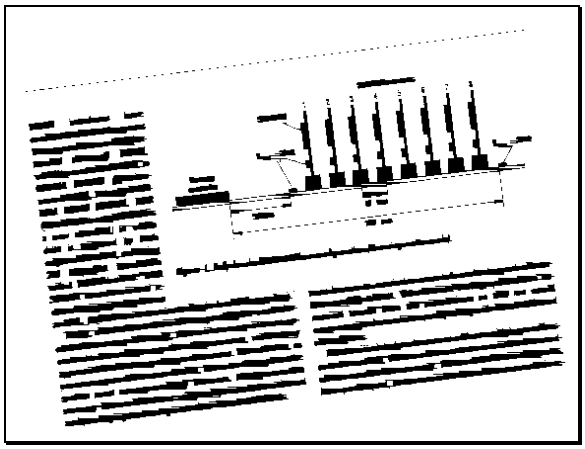

Figure 2. The connected components of an input document

In the second experiment, we considered groups of connected components and determined the document skew from the mean and median values in each group. Table 2 indicates the results. The conclusions drawn from this experiment are:

1. The mean and median values calculated from cc1 to cc5 skew angles are often the best estimates calculated through the mean and median values of the groups of connected components.
2. The median value calculated from cc1 to cc3 skew angles is often a good and robust estimate of the document skew angle.
3. For various documents, the sum of absolute error of the document skew calculated from the median of

cc1 to cc5 is less than the sum of absolute error for other cases.

4. Considering the above points, the median value of cc1 to cc5 seems to be a proper choice for the document skew angle.

The above experiments were repeated for the two methods given in section 2 for finding the skew angle of the connected components. The results showed that calculating the skew angle using the minimum-area bounding rectangle algorithm, except in rare cases, produces better results with no significant increase in total processing time and is preferable.

The typical processing time required to find the skew angles for different cases are given in Table 3. These values are obtained on a 233Mhz Pentium machine operating under Windows 95. The developed software was written in Visual C++5. In all cases, about 90% of the indicated time was used by the smoothing algorithm and specifying connected components. It can be observed that when a single connected component is used, the algorithm is quite fast, and taking the median values only slightly increases the required time. The accuracy and speed of document skew detection by the proposed method is shown to be very acceptable. Additional experiments show that if only horizontal smoothing is used, the required processing time will decrease about 10%.

Table 1. Document Skew Estimated for 8 Documents and 10 Connected Components Using the Proposed Algorithm (all in degrees)

	Skew angle	cc1	cc2	cc3	cc4	cc5	cc6	cc7	cc8	cc9	cc10	Mean	Median	$\lvert E_{\max} \rvert$	$\lvert E_{mean} \rvert$	$\lvert E_{median} \rvert$
Doc. 1	6.0	6.1	6.1	5.9	5.6	5.2	5.7	5.9	5.9	6.3	5.9	5.86	5.90	0.8	0.14	0.1
Doc. 2	-4.0	-4.0	-3.3	-4.1	-4.0	-2.8	-4.0	-3.7	-3.9	-4.2	-4.0	-3.80	-4.00	1.2	0.20	0
Doc. 3	7.0	7.0	6.8	6.9	7.2	7.0	6.8	6.9	6.9	7.1	7.1	6.97	6.95	0.2	0.03	0.05
Doc. 4	5.0	5.1	4.9	5.1	5.0	4.0	5.3	5.0	5.0	4.8	4.9	4.91	5.00	0.3	0.09	0
Doc. 5	-8.0	-8.0	-8.0	-8.0	-7.7	-8.0	-8.0	-8.0	-8.1	-8.1	-8.1	-8.00	-8.00	0.3	0	0
Doc. 6	-5.0	-5.0	-5.1	-5.1	-5.7	-4.7	-4.9	-5.2	-5.1	-4.4	-4.8	-5.00	-5.05	0.7	0	0.05
Doc. 7	-2	-1.7	-1.6	-2.2	-2.1	-1.9	-2.1	-1.9	-2.4	-1.8	-1.9	-1.96	-1.9	0.4	0.04	0.1
Doc. 8	3.8	3.1	3.5	3.5	3.9	3.7	2.4	3.9	5.1	3.3	5.3	3.77	3.60	1.49	0.04	0.21

Table 2. Document Skew Detected for 8 Documents and Mean or Median Values of Several Connected Components Using the Proposed Algorithm (all in degrees)

	Mean				Median			
	cc1-cc3	cc1-cc5	cc1-cc7	cc1-cc9	cc1-cc3	cc1-cc5	cc1-cc7	cc1-cc9
Doc. 1	6.03	5.78	5.79	5.86	6.10	5.90	5.90	5.90
Doc. 2	-3.80	-3.64	-3.70	-3.78	-4.00	-4.00	-4.00	-4.00
Doc. 3	6.90	6.98	6.94	6.96	6.90	7.00	6.90	6.90
Doc. 4	5.03	4.82	4.92	4.91	5.10	5.00	5.00	5.00
Doc. 5	-8.00	-7.99	-7.96	-7.99	-8.00	-8.00	-8.00	-8.00
Doc. 6	-5.07	-5.12	-5.1	-5.02	-5.1	-5.1	-5.1	-5.1
Doc. 7	-1.83	-1.9	-1.93	-1.97	-1.7	-1.9	-1.9	-1.9
Doc. 8	3.37	3.54	3.43	3.60	3.50	3.50	3.50	3.50

Table 3. Typical Time Required for the Skew Angle Detection

	Image Dimension	cc1 (sec)	cc1-cc3 Median (sec)	cc1-cc5 Median (sec)
Doc. 1	632 x 668	0.44	0.55	0.61
Doc. 2	363 x 615	0.28	0.33	0.33
Doc. 3	775 x 587	0.60	0.60	0.65
Doc. 4	514 x 811	0.66	0.72	0.77
Doc. 5	930 x 1156	2.19	2.36	2.47
Doc. 6	608 x 824	0.77	0.82	0.88
Doc. 7	764 x 1024	1.15	1.21	1.25
Doc. 8	542 x 580	0.71	0.71	0.72

4. Concluding remarks

A new method for calculation of document skew was presented in this paper. The method calculates the skew angle by finding the orientation of the minimum-area-bounding rectangle of one or several connected components. Connected components are produced by a smoothing algorithm, which does not require manual parameter adjustment and is not very sensitive to connection of text and non-text. No limitation exists in the range of detectable skew angles and the achievable accuracy. Experiments show the good performance of the algorithm. The accuracy of the estimated angle is very high and the speed of the algorithm is very acceptable.

References

[1] H.S. Baird, "The Skew Angle of Printed Documents", Proc. Conf. Photographic Scientists and Engineers, SPIE, Bellingham, Wa., 1987, pp. 14-21.

[2] G. Ciadiella et al., "An Experimental System for Office Document Handling and Text Recognition", in Proc. 9th Int. Conf. Pattern Recognition, 1988, pp. 739-743.

[3] T. Akiyama, and N. Hagita, "Automated Entry System for Printed Documents", Pattern Recognition, Vol.23, No.11, 1990, pp.1141-1154.

[4] T. Pavlidis, and J.Zhou, "Page Segmentation by White Streams", Proc. 1st Int'l Conf. Document Analysis and Recognition (ICDAR), Int'l Assoc. Pattern Recognition, 1991, pp. 945-953.

[5] S.C. Hinds, J.L. Fisher, and D.P. D'Amato, "A Document Skew Detection Method Using Run-Length Encoding and the Hough Transform", 10th ICPR, Atlantic City, 1990, pp. 464-468.

[6] D.S. Le, G.R. Thoma, and H. Wechsler, "Automated Page Orientation and Skew Angle Detection for Binary Document Images", Pattern Recognition, Oct. 1994, pp. 1325-1344.

[7] B.Yu and A.K. Jain, "A Robust and Fast Skew Detection Algorithm for Generic Documents", Pattern Recognition, 1996, Vol.29, No.10, pp.1599-1629.

[8] A. Hashizume, P.S. Yeh, and A. Rosenfeld, "A Method of Detecting the Orientation of Aligned Components", Pattern Recognition Letters, Vol.4, 1986, pp. 125-132.

[9] L. O'Gorman [1993], "The Document Spectrum for Page Layout Analysis", IEEE Trans. on PAMI, Vol.15, No.11, 1993, pp. 1162-1173. Get from [L. O'Gorman and R. Kasturi (eds.) 1995.

[10] H. Yan, "Skew Correction of Document Images Using Interline Cross-Correlation", CVGIP Graphical Models and Image Processing, Vol.55, No.6, Nov. 1993, pp. 538-543.

[11] Avanindra and S. Chaudhuri, "Robust Detection of Skew in Document Images", IEEE Tran. On Image Processing, Vol. 6, No. 2, Feb. 1997, pp. 344-349.

[12] B. Gatos, N. Papamarkos, and C. Chamzas, "Skew Detection and Text Line Position Determination in Digitized Documents", Pattern Recognition, Vol.30, No.9, 1997, pp. 1505-1519.

[13] W. Postel, "Detection of Linear Oblique Structure and Skew Scan in Digitized Documents", Proc. 8th Int. Conf. Pattern Recognition (ICPR), IEEE Computer Society Press, Claif., 1986, pp. 639-743.

[14] S. Chen and R. M. Haralick, "An Automatic Algorithm for Text Skew Estimation in Document Images Using Recursive Morphological Transforms", in Proc. IEEE Int. Conf. Image Processing (ICIP' 94), pp. 139-143.

[15] Y. Ishitani, "Document Skew Detection Based on Local Region Complexity", 2nd ICDAR, Tsukuba, 1993, pp. 49-52.

[16] J. R. Parker, Algorithms for Image Processing and Computer Vision, Wiley Computer Pub., 1997.

[17] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document Analysis System", IBM J. Res. Develop., vol.6, Nov. 1982, pp.642-656.

[18] F.Y. Shih, and S.S. Chen, "Adaptive Document Block Segmentation and Classification", IEEE Trans. on. Sys. Man and Cyber. Part B, Vol.26, No. 5, October 1996.

[19] R. Jain, R. Kasturi, and B.G.Schunck, Machine Vision, McGraw-Hill, Inc. , 1995.