An Enhanced Skew Angle Estimation Technique for Binary Document Images

Huiye Ma Zhenwei Yu
Beijing Graduate School of China University of Mining & Technology
D11 Xue-Yuan Road , Beijing , 100083 ,P.R. CHINA

Abstract

In this paper , an improved skew angle estimation technique is presented. Skew angle detection is an important part of any Optical Character Recognition (OCR) and document analysis system. The method considers the lowermost pixels of some selected characters of the text, which maybe subject to Hough transform for skew angle detection. A fast approach is also proposed which works almost as accurately as Hough transform for skew angle detection. Experimental results are presented and compared with results on several other skew detection methods.

1. Introduction

In document analysis the first step is to acquire a digitized raster image of the document using a suitable scanning system. Then it is followed by page layout analysis and character recognition. Before the structure of the text is obtained, a test is carried out to find out whether the document is skewed. As a part of the development of a successful OCR and document structure system we have proposed a method, based on the lower pixels of some selected characters, for the estimation of the skew angle of the recorded digital document.

Many methods have been developed for the correction of skewed document images. There exist five classes of skew estimation techniques. They are based on (1) Projection profile, (2) Fourier method, (3) Nearest neighbor clustering, (4) Correlation, (5) Hough transform technique respectively. In the first method, the projection profile is computed at a number of angles and a measure of difference of peak and trough height is made for each angle. The maximum difference corresponds to the best alignment with the text line direction, which, in turn, determines the skew angle. In the second method, the direction for which the density of the Fourier space is the largest gives the skew angle. And very often for a document image, the largest density direction of the Fourier space is on a vertical line and the true density direction may not be the largest. This makes the skew detection/search difficult. Hashizume et al. proposed nearest neighbor clustering to skew detection. He found all the connected components in the document and computed the direction of its nearest neighbor for each component. A histogram of the direction angle is computed, the peak of which indicates the document skew angle. O'Gorman generalized the approach in so-called 'docstrum' analysis. Yan [4] introduced a method for determining the skew angle of an image using cross-correlation between lines at a fixed distance. It is based on the observation that the correlation between two vertical lines in an image of a skewed document is maximized in general if one line is shifted relatively to the other line such that the character base line levels for the two lines are coincident. Hough transform has been used by Srihari and Govindaraju for skew detection. The basic method consists of mapping points in Cartesian space (x , y) to sinusoidal curves in (ρ , θ) space via the transformation: ρ =xcos θ + ysin θ . Each time a sinusoidal curve intersects another at a particular value of ρ and θ , the likelihood increases that a line corresponding to that (ρ , θ) coordinate value is present in the original image. An accumulator array is used to count the number of intersection at various ρ and θ values. The skew is then determined by the θ values corresponding to the highest number of counts in the accumulator array.

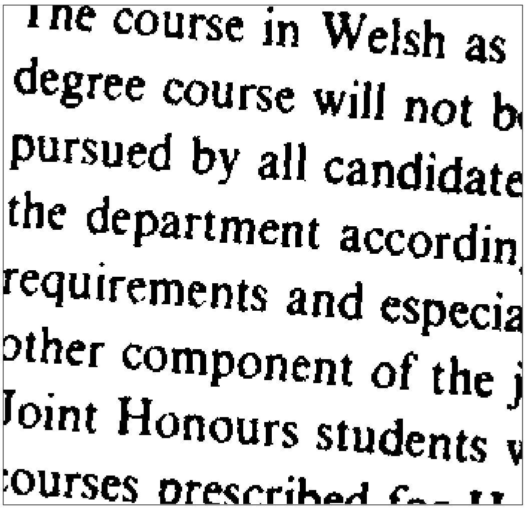

Fig.1. An example of English skewed text.

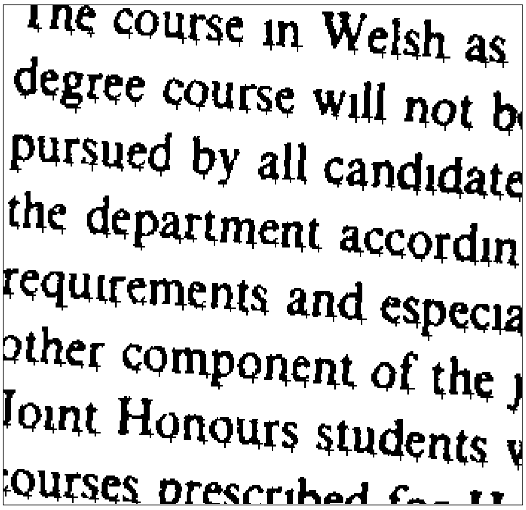

Fig.2. The dots of i and j, and punctuation marks like comma etc. are removed and selected lowermost base line pixels are marked by +.

In this paper we propose a simple and fast algorithm for determining the skew angle of a document image. It can select the data automatically so that only relevant pixels corresponding to the base line (see Fig.1 and Fig.2) can be used for skew estimation. It also proposes a quick method of skew estimation using these relevant pixels.

2. Skew angle estimation algorithm

Since the characters of a word are mostly separated, the characters in a document can be detected by the method of component labeling. When we begin with component labeling, the extreme (both in x coordinate and y coordinate) pixel coordinates are calculated for each labeled object and the bounding box is defined. We discard those components whose box height is very small so that the dots of the characters like i and j and punctuation marks, like full stop, comma, hyphen, etc. are removed. For example, the lowermost pixel is chosen from each selected component in Fig.2. If more lowermost pixels are there then the rightmost among them is selected. Thus, we get the pixels that lie on base lines of the text image.

To reduce the processing time further, we propose a quick approach that is to cluster pixels corresponding to individual base lines. For example, we put the encircled pixels of Fig.3 into one cluster because they belong to the same base line. The slope of line MN gives an estimate of the skew for these encircled pixels. In this way pixels of each base line are clustered and an estimate of the skew angle is obtained from them. An average of these skew angles is considered to be the estimate over the base lines.

In order to cluster the pixels of each base line we start with a line called first-line. In this line, we will search all its pixels and divide them into two groups. The pixels taken from characters like a,b,c,d,e etc. are put into above level group, while those taken from characters like g,y,p etc. are put into under level group. Then we will compute the skew angles according to two furthest pixels in each group separately and get the average skew angle. The pixels are scanned row-wise and a 2W neighborhood of the first pixel encountered is considered (see Fig.4). The neighborhood of 2W is considered because the spacing between two characters (hence the spacing between two pixels) is usually less than W. Now for a pixel A we get its nearest neighbor pixel B that stays within a circle with diameter 2W and center A, and the slope of the line AB has the restriction of not greater than ±45° . In order to confirm that A and B is in the same level, we should do as follows. (1) Search for a nearest neighbor pixel of the current B, (2) if pixel B lies above the line AC in the triangle ABC obviously, put pixel B in above level group, (3) if pixel B lies under the line AC in the triangle obviously, put pixel B in under level group, (4) if pixel B lies on the line AC, put B to the same level group as A and C. (5) continue to do the same thing to the next pixel. This process will be repeated until there does not exist a nearest neighbor pixel of the current pixel. Thus we can get two groups of this line, one of which contains above pixels and the other contains under pixels. In each group, we find their furthest pixels M and N. The slope of line MN gives an estimate of the skew for these encircled pixels. After averaging these two skew angles, we will get the estimate of the skew for the first-line.

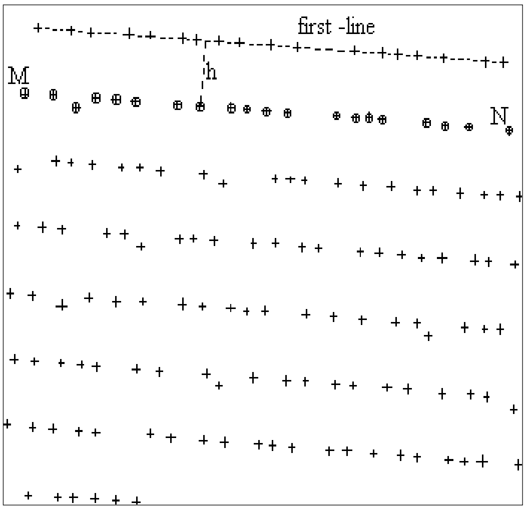

Fig.3.Clustering of pixels of base lines belonging to a single text line. (Encircled pixels are at a distance h+d or h-d from the first-line, where h is the constant distance and d is a small quantity.)

Then, for an unprocessed pixel x in base lines its normal distance to first-line is computed. If this distance is within ±d (where d is a very small quantity) of the distance of any existing bin pixel which has nearly constant normal distance to first-line, x is placed in that bin for clustering. Otherwise, a new bin is created and x is placed in the new bin. In the end, k bins and k clusters are produced if there are k text lines in the document.

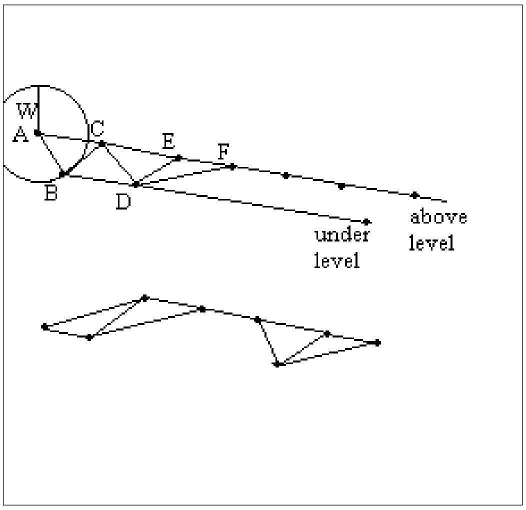

Fig.4. The approach of choosing first-line.

In brief, our quick skew detection algorithm has the following steps.

SKEW ALGORITHM

Step 1. Find connected components in the binary document image .

Step 2. Choose those connected components whose box height is very small and remove them.

Step 3. Find a lowermost pixel to obtain base lines for each component.

Step 4. Find first-line from the pixels of base lines.

Step 5. Find the normal distance of x to first-line for each pixel x in base lines.

Step 6. Cluster pixels of base lines corresponding to individual text lines.

Step 7. Find the slope of the line joining two furthest pixels of two groups for each cluster of pixels. After averaging these two angles, we will get a local estimate of the skew angle.

Step 8. Find the average of all the cluster lines as the final estimate of the skew angle.

3. Conclusion

We considered more than one hundred document images from different books, magazines and journals. They were digitized by a flatted scanner at a resolution of 300 dpi. The documents were tilted by a pre-specified angle ranging between 0 and ±45° . This angle is considered as the true skew angle. Typical results are shown in Table 1. We can see from Table 1 that the docstrum method (O'Gorman 1993) is less accurate for almost all skew angles. Hough transform on pixels selected by Li et al. method is better than the docstrum method. We noted that the accuracy of these two methods could be improved if the characters having ascender and descender are deleted, as done in our proposed method. Yan's method is quite accurate but it is time consuming. To test the time efficiency of our quick method (method A) with respect to Hough transform based on method C, we computed the time of execution in these methods. The average execution times for a document of 2100*2100 pixels on a TONTRU (with P166, and Windows 95) machine are 570 and 210 seconds for methods C and A. So our method is more accurate than Hough transform based on method C and ours is much more time economical.

Table 1

True angle	A	B	C	D
40	39.85	39.13	39.52	39.92
30	29.86	29.15	29.41	29.89
20	19.57	19.17	19.21	19.53
10	9.97	10.78	10.64	10.45
5	5.04	5.74	5.25	5.03
3	3.14	3.87	2.81	3.67

A: Our proposed quick method
B: Docstrum method (O'Gorman)
C: Hough transform on pixels selected by the Li et al. method
D: Yan's method

For a complex document containing graphics and drawings our proposed method would work well if the bounding box height is chosen in a modified way and if the text region is larger than the non-text region. However, it is better to use Hough transform because it is more robust to noise while the quick method depends on the right choice of first-line and clustering of pixels of each text line which may not be attained always for a complex document.

References

[1] Li, D.S., G.R. Thoma and H. Wechsler. Automatic page orientation and skew angle detection for binary document images. Pattern Recognition,27:1325-1344,1994.

[2] Com, U.P., B.B. Chaudhuri. An improved document skew angle estimation technique. Pattern Recognition Letters,17: 899-904,1996.

[3] Akiyma, T. and N. Hagita. Automatic entry system for printed documents. Pattern Recognition, 23:1141-1154,1990.

[4] Yan, H. Skew correction of document images using interline cross-correlation. Computer Vision, Graphics, and Image Processing, 55: 538-543,1993.

[5] Hashizume, A., P.S. Yeh and A. Rosenfeld. A method of detecting the orientation of aligned components. Pattern Recognition Letters, 4: 125-132,1986.

[6] Hinds, S.C., J.L. Fisher and D.P. D'Amato. A document skew detection method using run-length encoding and the Hough transform. Proc. 10th Internat. Conf. on Pattern Recognition, Hedgue, Vol. 1,pages 464-468,1990.

[7] O'Gorman,L. The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell, 15: 1162-1173,1993.

[8] Pavlidis, T. and J. Zhau. Page segmentation and classification. Computer Vision, Graphics, and Image Processing, 54: 484-496,1992.

[9] Srihari, S.N. and Govindaraju. Analysis of textual images using the Hough transform. Machine Vision Appl, 2:141-153,1989.