Skew Detection via Principal Components Analysis

Tal Steinherz **Nathan Intrator**
School of Mathematical Sciences
Tel-Aviv University
Ramat Aviv 69978, Israel
(talstz,nin)@math.tau.ac.il

Ehud Rivlin
Department of Computer Science
Technion
Haifa 32000, Israel
ehudr@cs.technion.ac.il

Abstract

Skew detection via principal components is proposed as an effective method for images which contain other parts than text. It is shown that the negative of the image leads to much more robust results, and that the computation time involved is still practical.
Keywords: Document Analysis, Skew Detection, Principal Components.

1 Introduction

Skew detection has become very important as document processing tools are making a great progress. Most applications require a prior skew detection to be applied before any recognition step is executed.

In this paper a method for skew detection based on Principal Components is presented. We assume that the text in the input image share a common orientation. This assumption is necessary in order to define the skew angle of a full page, and it is used in other methods. Obviously the lines can be at any angle with respect to the horizontal axis. We begin with a short review on previous work, and briefly go over the main approaches and give a few examples. In the following section we present the proposed method. We conclude with experimental results which demonstrate the performance of our method on different documents.

2 Previous Work

Several approaches have been proposed as alternatives for skew angle detection of document images. All of them require a dominant text area to be present in order to work properly. Unlike graphic zones, text areas possess a well known characteristic structure – one or more (separate) lines of printed or handwritten words, sharing a common direction. Main approaches for skew detection include: Hough transform, nearest neighbor, correlation and projection profile.

Hough transform is a popular method for skew detection [11], [6], [9], [8] and [12]. It is capable of locating fragmented lines in a binary image. Therefore given a group of black pixels, one can find the imaginary line or lines that go through the maximum number of these pixels. Given a binary image with a dominant text area, the detected lines will most probably go along the whole middle zone of the textual lines. Hence these lines have approximately the same skew as the reference lines of the text which define the skew of the whole page. Whenever the Hough transform is used, there is always a tradeoff between accuracy and speed. The more accurate the angles of the detected lines are, the more computation is required. In addition the computation time depends on the number of pixels in the image. Therefore most methods ([6], [8], [12]) make some efforts to reduce the number of pixels by compressing a group of pixels into one representative.

In the nearest neighbor approach [5], a connected component analysis is required first of all. Then several iterations take place in order to connect each component to its nearest neighbor in recursion. This process results in several chains representing the textual lines. A line that goes through the central mass of the components in a chain approximates the skew of the associated textual line. Alternatively, one can calculate the skew by averaging the skews of the lines that connect neighboring components in a chain. This method is based on the fact that inter character and inter word spaces between two consecutive characters or words respectively, are usually smaller than spaces between such neighboring elements that belong to different lines. Hence the resulting chains

correspond to single textual lines and do not contain components of more than one line.

The correlation approach attempts to find the best correlation between two or more profiles of the image, taken from vertical (horizontal) cuts of the image [1], [4]. When the image contains text organized in a few lines, one can deduce from the correlation between a pair of profiles the amount of vertical (horizontal) shift the lines were exposed to. The skew angle of these lines can then be calculated, knowing the distance between the cuts from which the profiles were taken. The simplest profile is achieved by taking a single column of the image at the position of the cut. The resulting profile is a binary vector.

Given a binary image, a projection profile of the image on an axis w, is a function $f(z)$ where z is a point along the axis, and $f(z)$ is the number of black pixels on a line perpendicular to the axis at that point.

The projection profile tool was used by many to help determine the skew angle of a document image [10], [3]. Based on the assumption that there exists a text area, one can take advantage of the fact that most of the black pixels appear in the middle zone of each text line. In this case a unique projection profile is accepted when the image is projected on an axis that is perpendicular to the direction of the text lines, because the columns along which the pixels are accumulated are aligned with the text lines. Several functions were proposed for the evaluation of the projection profile pattern. All of these functions share the fact that their global maximum is reached when the projection profile at the skew angle is given. As a result, a projection profile based skew detection algorithm computes the projection profile of the document image at various angles and chooses the one in which the evaluated pattern is maximized. Some heuristics were proposed in order to converge to the correct angle quickly. For example the method proposed by Baird in [3].

One should refer to the following surveys by Hull ([7]), and Bagdanov et al. ([2]) for further review of previous work.

3 Skew detection via Principal Components Analysis

The deviation of a projection profile was also mentioned as an alternative evaluation function in which the global maximum is usually achieved at the skew angle. Furthermore the proposed method finds the exact angle at which the projection profile has the maximum deviation, within a single iteration!

Given a binary image, where stroke pixels are black and background pixels are white, one can map any black pixel in the image to a two dimensional vector and vice versa. For example, a black pixel located in raw y and column x in the image is mapped to the vector (x, y). The projection profile of the image on any axis, is equal to the projection profile of the set of associated vectors on a unit vector that points to the same direction as the axis does. In this case one can find the unit vector that maximizes the projection profile deviation as an equivalent skew detection method. The desired unit vector is called the Principal Component of the given vector set, and it is equal to the eigen vector that is associated with the largest absolute eigen value of this set. This property is important because it implies that the skew angle can be found within one iteration! Therefore the proposed algorithm is the following. Given binary document image, we create an isomorphic set of two dimensional vectors, where each vector corresponds to a black pixel in the image. The transformation maps a pixel to a vector of the same coordinates. Then the Principal Component of the resulted vector set is found: first the eigen values and corresponding eigen vectors of the vector set are found, then the vector associated with the largest absolute eigen value is picked. The direction of the Principal Component found is perpendicular to the direction of the text lines in the image, which is referred to as the skew angle. Hence the skew angle is achieved within a single iteration.

Most of the computation required for this algorithm concerns the eigen value and vector calculation phase. This procedure is done in two steps. First the covariance matrix of the vector set is found. Then the eigen values and corresponding vectors are calculated. In this case the covariance matrix dimensions are 2×2 because it represents vectors of only two dimensions, and therefore computing its eigen values and vectors is immediate. Hence the only significant computation left concerns the creation of the covariance matrix. In this case it requires the multiplication of two matrices of $2 \times N$ and $N \times 2$ dimensions respectively, where N is the number of black pixels in the image. The first matrix contains the set of vectors ordered in columns, the second matrix is the transform of the first.

We have discovered that often it is better to use

the negative image of the document instead of the regular one as the number of sample points is greatly increased and thus the sensitivity to different types of characters and possible non-text parts is greatly reduced. When a dominant text is present this will have almost no influence on the final result, yet it will be slightly improved. The projection profile pattern at the skew angle remains very similar, but with one difference: peaks and valleys are exchanged one with another. However the total number of peaks (valleys) is kept and so is the averaged difference between a peak and the valley that follows or precedes it. From similar arguments there will be no significant change in the projection profile pattern at other angles and therefore the global maximum of the evaluation function remains at the same point, i.e. the projection profile at the skew angle.

In case of an image of mostly pictures and other kinds of graphical areas – where the distribution of black and white pixels might be considered random – the existence of a few text lines can make a great difference. Note that it requires only a few text lines that might be located at various positions on the page except that they should all share the same direction. Since real documents always contain some portion of graphics (a picture, graph, company logo etc.) using the negative image improves the final results enormously, and extends the range of documents that the system can handle with respect to skew correction.

4 Experimental Results

Our results refer to experiments that detected skew on full pages. The Principal Component method was used on each image twice: in its natural mode and when the negative was used instead. Our method and all the surrounding processing was implemented in MATLAB and tested on a Unix machine. We focus on the improvements in speed and the more accurate results achieved by using the negative image. Therefore we used a small database of document images that present a variety of text and graphics mixture. This database of 12 fax images that are used for compression algorithm evaluation purposes was found to be representative for skew detection scenarios as well. One of the images (indexed *ccitt*8) was disqualified in advance because it contains two text blocks in different orientations (horizontal and vertical). Table 1 summarizes the results achieved. For each image we present the detected skew, number of black pixels in the image and amount of computation required in flops – for both the natural and negative images. Note that the detected skew was normalized according to the manually observed skew of the image.

One can see that the results achieved by the negative images are usually superior than the ones achieved by the regular ones. The only exception in the testing set is image indexed *test*3. In this image a number of vertical lines produce too much noise, so the projection profile of maximum deviation is aligned in this direction. However note that the great difference between the skew detected in the regular and negative images respectively, is a fair cause to disqualify both results. In the other 10 images the averaged error of the results achieved on the negative images is 0.6306 in comparison with 11.2959 for the regular ones. In addition we would like to point out that the difference between the detected skews of the natural image and its negative, increases in accordance with the amount of graphics in the image.

5 Discussion

A novel projection profile that is based on Principal Components analysis for skew detection was presented. It enables one to find the angle at which the projection profile deviation is maximized within a single iteration. This is considered a major improvement since computation time can be reduced enormously using this method. In Table 1, one can see that finding the skew of a given image requires $8 \times N + c$ flops, where N is the number of black pixels in the image and c is a constant $\ll N$. Classical projection profile as well as Hough transform based methods require approximately $3 \times N$ flops for every tested angle, and there are usually a few dozens of tested angles i.e. iterations. Though various speedups were suggested before, they usually produce less accurate results. We used the negative image of documents in order to increase the number of pixels participating in the estimate, and thus improve the detected skew accuracy. It was shown that by using this alternative, there were new document images with dominant graphic parts that their skew could now be found.

Moreover, it was shown that by running skew detection on both the regular and the negative images, one can determine the amount of graphics in the image according to the level of agreement between the two.

Index	Natural			Negative		
	Detected Skew	No. of Black Pixels	Flops	Detected Skew	No. of Black Pixels	Flops
ccitt1	-4.8881	15987	127933	0.2686	413325	3306637
ccitt2	34.9943	19135	153117	-1.1421	410177	3281453
ccitt3	-22.8014	34870	278997	2.1258	394442	3155573
ccitt4	1.8147	52685	421517	-0.2244	376627	3013053
ccitt5	-11.5372	32444	259589	0.7034	396868	3174981
ccitt6	12.3951	20927	167453	-0.4494	408385	3267117
ccitt7	0.3879	30296	242405	-0.1722	317277	2538253
test1	-9.4868	8184	65509	0.2212	408840	3270757
test2	-8.7228	17474	139829	0.5133	471742	3773973
test3	5.0097	33202	265653	-89.8460	603209	4825709
test4	5.9308	56611	452925	-0.4857	374237	2993933

Table 1: Skew detection results of natural images and their negative version. The difference between the detected skew and the observed skew is given, as well as the corresponding number of floating point operations.

References

[1] Avanindra and S. Chaudhuri. Robust detection of skew in document images. *IEEE Transactions on Image Processing*, 6(2):344–349, 1997.

[2] A. D. Bagdanov and J. Kanai. Evaluation of document image skew estimation techniques. In *Proceedings of the SPIE - Document Recognition III*, pages 343–353, 1996.

[3] H. S. Baird. The skew angle of printed documents. In *Proc. Conf. of the Society of Photographic Scientists and Engineers*, pages 14–21, May 1987.

[4] B. Gatos, N. Papamarkos, and C. Chamzas. Skew detection and text line position determination in digitized documents. *Pattern Recognition*, 30(9):1505–1519, 1997.

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

[6] S. C. Hinds, J. L. Fisher, and D. P. D'Amato. A document skew detection method using run-length encoding and the Hough transform. In *Proceedings Int. Conf. on Pattern Recognition*, pages 464–468, June 1990.

[7] J. J. Hull. Document image skew detection : survey and annotated bibliography. In *Document Analysis Systems II*, pages 40–64, 1998.

[8] D. S. Le, G. R. Thoma, and H. Wechsler. Automated page orientation and skew angle detection for binary document images. *Pattern Recognition*, 27(10):1325–1344, 1994.

[9] Y. Nakano, Y. Shima, H. Fujisawa, J. Higashino, and M. Fujiwara. An algorithm for skew normalization of document image. In *Proceedings Int. Conf. on Pattern Recognition*, pages 8–13, June 1990.

[10] W. Postl. Detection of linear oblique structures and skew scan in digitized documents. In *Proceedings Int. Conf. on Pattern Recognition*, pages 687–689, October 1986.

[11] S. N. Srihari and V. Govindaraju. Analysis of textual images using the Hough transform. *Machine Vision and Applications*, 2:141–153, 1989.

[12] B. Yu and A. K. Jain. A robust and fast skew detection algorithm for generic documents. *Pattern Recognition*, 29(10):1599–1630, 1996.