Pattern Recognition Letters 29 (2008) 1893–1900

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Skew detection for complex document images using robust borderlines in both text and non-text regions

Hong Liu [a,b,*], Qi Wu [a], Hongbin Zha [a], Xueping Liu [c]

[a] National Laboratory on Machine Perception, Peking University, Beijing 100871, China
[b] Shenzhen Graduate School, Peking University, Beijing 100871, China
[c] Ricoh Co., Japan

ARTICLE INFO

Article history:
Received 10 September 2006
Received in revised form 31 March 2008
Available online 20 June 2008

Communicated by H. Sako

Keywords:
Document analysis
Skew detection
Robust borderline
Linear filter
Optimization

ABSTRACT

A new skew detection method for complex document images based on robust borderlines extracted from both text and non-text regions is proposed in this paper. First, borderlines are extracted from the borders of large connected components in a document image by using a run length based method. Second, after filtering out non-linear borderlines, a fast iteration algorithm is applied to optimize each linear borderline's directional angle. Finally, the weighted median value of all the directional angles is calculated as the skew angle of the whole document. Experiments on 2000 various skew document images are implemented. Total correct rate is 95.2%, and the detecting time on average is less than 0.2 s for each document. The proposed skew detection method is efficient for complex documents with horizontal and vertical text layout, three kinds of linguistic characters in English, Japanese and Chinese, especially for documents with predominant non-text regions or sparse text regions.

1. Introduction

When converting paper documents into its electronic form by using scanners and digital cameras, it is almost inevitable to make the inborn document images a little skew. In actual applications, document images without any skew are usually required. Because document analysis such as character recognition and image segmentation are very sensitive to page skew. Therefore, skew detection becomes an important task in document analysis.

Due to its essentiality, researchers made great efforts on skew detection and many methods were presented in literatures during the last two decades.

Projection profile based methods first discretize angle space, and then histograms are calculated at each possible angle. The skew angle corresponds to the histogram which shows the maximum variation. Some efforts of this category are presented in (Akiyama and Hagita, 1990, Ishitani, 1993).

Cross correlation based method is first proposed by Yan (1993). The main idea of this method is that pixels on the same text line have the maximum statistical correlation. Therefore, this method uses a correlation function to find the amount of vertical shift v for a certain horizontal distance d, and then v/d denotes the slope of the image. In Chen and Ding's improved method (Chen and Ding, 1999), an enhanced correlation function is applied on automatically selected regions until the selected region is a text region, which is judged by a peak-valley value analysis algorithm. It makes the method more robust and reduces computational cost.

The above categories of methods use statistical features to detect skew angles of document images. The statistical features are generally extracted from big text regions which should contain many text lines. Therefore, performances of these methods are affected by sparse text distribution. Moreover, if the non-text regions are predominant in the image, performances will be hard to be guaranteed. Another shortage of these methods is the conflict between accuracy and computational cost. The discretization of angle space brings an inherent system error. On the other hand, shortening the interval of the discretization to reduce the system error will increase the computational cost.

Hough Transform, which can detect parameters of straight lines in images, is another popular method for skew detection. Aiming at the inherent high computational cost of Hough Transform, some modified methods are proposed. Hinds et al. (1990) adopt horizontal and vertical run length algorithm and (Le et al., 1994) take advantage of bottom pixels to sample the candidate objects in order to reduce data scalar and accelerate their method. Methods of this category usually implement Hough Transform on big text regions, but Hough Transform can also be used for small text regions such as a separate text line. Amin and Fischer (2000) try to group

* Corresponding author. Address: National Lab on Machine Perception, Peking University, Beijing 100871, China. Fax: +86 10 62755569.
E-mail address: hongliu@pku.edu.cn (H. Liu).

0167-8655/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2008.06.008

connected components with similar bounding rectangle size and small distance, and then implement Hough Transform only on the bottom connected components of each group. Shivakumar et al. (2005) use a boundary growing method to smear text lines and implement Hough Transform on all the lowermost, uppermost coordinates and centroids of characters in text lines. One thing should be mentioned is that if text lines can be partitioned and extracted well, Hough Transform can be replaced by least squares method due to its accuracy and rapidness. As discretization of angle space is a necessary step in Hough Transform, the conflict mentioned above is also unavoidable in Hough Transform based methods.

Nearest-neighbor based methods, which generally depends on local features of text, are the most popular methods in recent years, which is first proposed by Hashizume et al. (1986). In their method, connected components should be detected first, and then nearest-neighbor connected components are clustered. Each clustering indicates a skew angle candidate and the skew angle of the whole image corresponds to the median or mean of the candidates. O'Gorman (1993) generalizes a method to cluster *k*-nearest-neighbor connected components. Nearest-neighbor based methods can calculate skew angles in any text region which contain more than one connected components of characters. Therefore, these methods can reduce the affection of non-text regions most. However, as these local features could be affected by noise and various shapes of neighbor characters, the detecting accuracy is usually not very high. Several methods are proposed to improve the accuracy. Cao et al. (2003) utilize eigen-points instead of cancroids to denote the connected components and least squares method to fit straight lines of clusters. In Lu and Chew's method (Lu and Tan, 2003), a size restriction idea is applied to wipe off noises and non-text regions and to form nearest-neighbor chains as long as possible. Then the chains long enough are selected to determine the skew angle of an image. A novel method is proposed by Amin and Wu (2005). First a CC (connected components) grouping method is used to identify the objects or regions of interest, and then a minimum bounding box method is applied to detect the skew angles of the objects or regions. It can deal with the non-text regions in numerous situations, but it is necessary to discretize the angle space, therefore the conflict between accuracy and computational cost is a limitation of this method.

Projection profile based methods and Cross Correlation based methods use statistical features in big text regions and nearest-neighbor based methods use local features of adjacent characters. Besides above categories, several other methods that focus on single text line are proposed. In these methods, a dilation operation, which uses morphology or run length algorithm, is first implemented to smear the characters in the same text line. After the operation, each text line is expected to be a connected component. Second, the connected components of text lines are extracted to calculate a skew angle. Finally, the mean or median skew angle of the text lines is selected as the skew angle of the whole image. Methods proposed by Das and Chanda (2001), Shi and Govindaraju, (2003), Safabakhsh and Khadivi, (2000) and Dhandra et al., (2006) belong to this category. This category can be considered as a congruity of the above categories. It reduces the affection of non-text regions and noises easily at a low computational cost. Furthermore, it does not need to discretize angle space in most situations. However, there are some problems left in the method. First, it is difficult to determine orientation of images (horizontal or vertical) and thresholds of dilation, which are important to form solid black block of a text line. If the assumption of horizontal orientation and a fixed threshold are applied, it will limit the adaptability of the methods. Another problem is how to calculate the skew angle of a text line. Enough sampled points must be extracted, and then a fast and accurate algorithm, such as least squares method should be implemented. However, which points should be extracted to stand for the text line, head ones, middle ones, or bottom ones? The answer is complex for different characters of languages and fonts. The adaptability of the methods will also be limited if only a fixed strategy is applied for complex documents.

To develop a skew detection method with improved accuracy and strong adaptability, a method based on robust borderlines, which are extracted from both text and non-text regions, is proposed in this paper. First, the borderlines of inherent large connected components, which are considered as non-text regions, are extracted, and then these connected components are removed. Second, a run length algorithm is applied to smear the characters of the same text lines. The orientation and the run length threshold are determined automatically. Then borderlines are extracted from the formed large connected components. Third, robust borderlines are selected from all the extracted borderlines by a borderline filter and an iteration algorithm is applied to calculate the skew angle of each robust borderline. Finally, a weighted median idea is presented to determine the skew angle of the whole image.

2. Motivation of our method

In most previous methods, two common assumptions are considered as the basis of their skew detection ideas for all document images. First, text lines are always parallel or vertical to skew directions of the images. Second, non-text regions are always obstacles for skew detection. Therefore, skew features are extracted only from text regions. Non-text regions are detected and then generally abandoned so as not to influence performance of skew detection. However, there are several problems left for complex document images. First, distinguishing text from non-text regions is rather difficult for many mixture regions. Second, computational complexity is very high to extract detailed features from all the text lines. In fact, it is not reliable to select randomly text sub-regions to speed up skew detection for complex documents with sparse text regions. Third, if text regions are sparse or non-text regions are predominant, insufficient enough skew features can be extracted from the text to support global skew detection for their statistical limitations.

To solve the above problems, the assumption that non-text regions are always obstacles for skew detection should be argued. Considering how human beings detect the skew angle of document images such as Fig. 1a, most people will select the page header to determine the skew angle of the whole image because the header is an obvious natural straight line parallel to the skew direction. Of course most methods can deal with this image well due to the

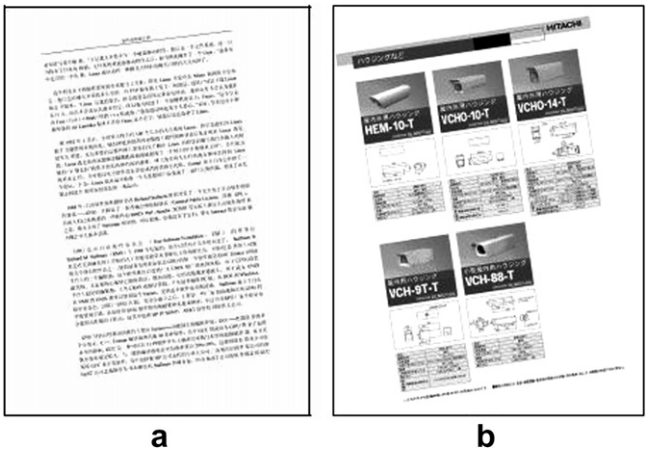

Fig. 1. Illustration of document images.

predominant text areas, although non-text areas in this image also provide useful information for skew detection. Furthermore, how to deal with the images without predominant text areas such as Fig. 1b? People may quickly detect the skew angle from the borders of the pictures, but most met hods will be inefficient in this situation since they only extract skew features from text regions.

In fact, the objects, which contain reliable skew features, not only exist in text regions, but also in non-text regions, such as page headers, page footers, borders of tables, frames of pictures, and so on. These linear features are parallel or vertical to the skew direction to a great extent in most cases and even more robust than many text lines. Moreover, edges of pictures and graphs are also suitable to detect skew angles in some cases. If these features can be used in skew detection, the adaptability will be improved, especially for images with predominant non-text regions.

To develop skew detection methods with strong adaptability, language independence is an important property that should be considered. It is researched and reported that text lines of English characters have bottom lines statistically (Das and Chanda, 2001), Indic text lines have inherent top lines (Chaudhuri and Pal, 1997), and both top and bottom lines of Chinese characters are good for skew detection due to their square property. Although it is not certain which border is more reliable for all kinds of languages, most characters of the above languages have at least one border suitable for skew detection.

It s found that good skew features usually exist in the borders of objects, either in text regions or in non-text regions from above analysis. Therefore, a new concept of robust borderline is proposed and a fast skew detection method is presented in this paper. To satisfy the adaptability demands, borderlines are extracted from all the four borders of objects (top, bottom, left, and right), and from both text regions and non-text regions. However, not all extracted borderlines are robust for estimating skew angles, because "bad" skew features such as rugged edges of pictures and headlines of English text lines, which can not represent skew directions accurately, may also be extracted in the process of borderline extraction. According to our observations, a distinct difference between reliable borderlines and "bad" ones is their linearity. Reliable ones have better linearity than "bad" ones in most situations. Therefore, a borderline filter, which can measure the linearity of borderlines, is applied to select the ones with good linearity as robust candidates for skew estimation. Then, to gain an better performance, an iteration algorithm, which can reduce the affection of wavy pixels by changing the weight of each pixel iteratively, is applied to optimize the candidate borderlines.

3. Borderline extraction

3.1. Run length for large connected components

For a testing binary document image, very small connected components are removed as noise first. Then, the four neighbors of each foreground pixel are set up to enhance the document image. Currently, large connected components, whose width or height is larger than the threshold min_*size*, will exist in the image. These large connected components generally exist in non-text regions, and occasionally in text regions if the layouts are too compact. In our experiments, these inherent large connected components are considered as non-text regions for the convenience of statistic calculation. Because there are reliable skew features in these large connected components and their existence may affect later processing, borderlines of these connected components should be extracted, and then these components and small ones their contained should be removed.

Now in the document image, small connected components which come from text regions are predominant. To form large connected components, a run length based algorithm is applied. The run length orientations, horizontal or vertical, should be judged first. Two intuitionistic assumptions are used here. First, the distance of text lines is larger than the distance of characters. Second, the run length orientation corresponds to the orientation at which there are much more acceptable run lengths. The steps of run length orientation judgment are presented as follows:

Step 1: Create two arrays $h\_dis[]$ and $v\_dis[]$. Initialize two parameters: $h\_sum = 0$, $v\_sum = 0$.

Step 2: Traverse the document image horizontally. For each foreground pixel, find the next foreground pixel in the same line and calculate their distance dis. If $1 < dis < rl\_th$ (Here rl_th is a given threshold), $h\_dis[h\_sum] = dis, h\_sum = h\_sum + 1$.

Step 3: Calculate the mean and standard deviation of horizontal run length.

Step 4: Same as Step 2 and 3, traverse the document image vertically. v_sum, mean and standard deviation of vertical run length are calculated, respectively.

Step 5: If $h\_sum > 2 \times v\_sum$, the orientation is horizontal, else if $v\_sum > 2 \times h\_sum$, the orientation is vertical, else the orientation corresponds to the smaller mean run length.

After the judgment of run length orientation, the threshold rl_th can be updated. It is equal to mean plus twice standard deviation at the run length orientation. Then this threshold is used to form large connected components as follows:

Traverse the document image. For each foreground pixel, find the next foreground pixel at the run length orientation and calculate their distance. If it is smaller than the updated threshold rl_th, set up all the pixels between the two pixels. After the traverse, most text lines are form large connected components.

3.2. Borderline extraction

Borders are defined for each connected component in four directions: top, bottom, left and right. According to the previous analysis for different languages, it is not certain that which border is suitable for skew detection, but at least one of the four borders is good for most situations. Therefore, in order to deal with various situations, our method extracts borderlines of each connected component large enough from their top, bottom, left and right. The steps are presented as follows (take extracting borderlines from top for example):

Step 1: For each connected component, compute its width. If the width is larger than min_*size*, go to Step 2, else go to Step 6.

Step 2: Pick up the top border of the connected component. For each column of the connected component, pick up the pixel with minimum vertical coordinate value of y.

Step 3: Sample the pixel with minimum y in every ten picked pixels from left to right. If there is more than one pixel with y, sample the leftmost one.

Step 4: These sampled pixels are denoted as $p_i(x_i, y_i)$, $i = 1, 2, \ldots, n$, p_{i-1} is the left sampled pixel of p_i. Initialize two variables *start* and *end* equal to 1.

Step 5: Traverse the sample pixels. For each i, if $(i = n)$ or $(dv = |y_i - y_{i+1}| > \max\_fall)$, let $end = i$. Then, if $(end - start) > \min\_length$, save the pixels from start to end as a borderline and let $start = i + 1$.

Step 6: End.

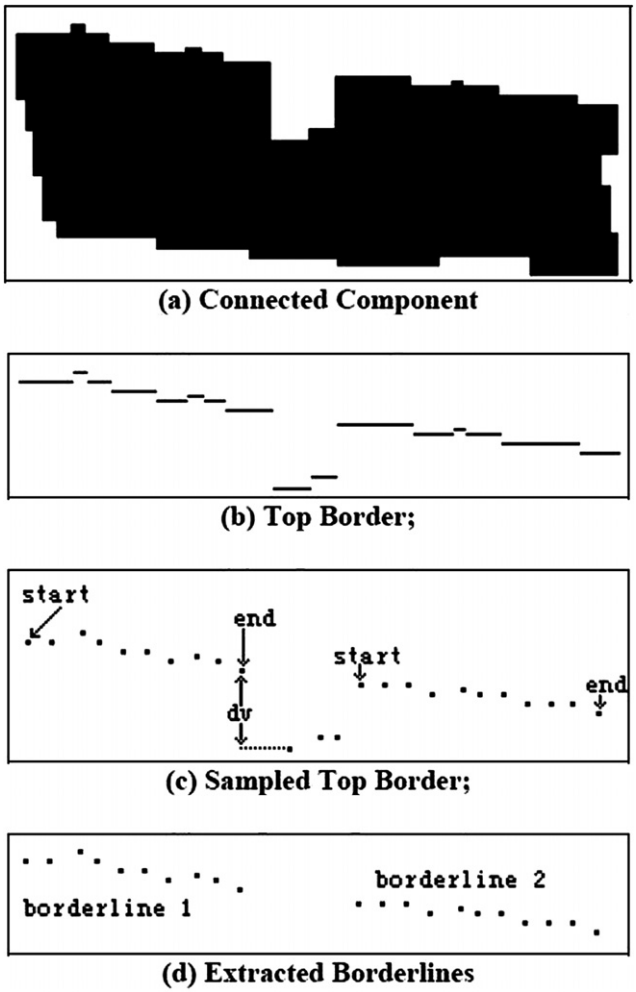

Fig. 2. Borderline extraction.

The threshold max_*fall* is applied because if the vertical distance *dv* between two neighbor sampled pixels is too long. It indicates that the two pixels belong to different text lines or unreliable borders in non-text regions. And if the amount of pixels in a borderline is too small, it will not indicate the skew angle accurately. Therefore, the threshold min_*length* is applied. Fig. 2 shows the process of borderline extraction.

4. Selection of robust borderlines

4.1. Analysis of borderline features

In document images, pixels are always denoted by two-dimensional coordinate (x,y) as it assume that every pixel has the same weight. In order to differentiate the contribution of pixels, here take the situation into account that each pixel has its own weight. Considering a group of pixels $p_i(x_i,y_i)$ and each pixel has its own weight $w_i(i = 1, 2,\ldots,n)$. The weight w_i indicates the mass of pixel $p_i(x_i,y_i)$. Thus the covariance matrix A of these pixels can be calculated by the following formula:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad a_{12} = a_{21} = \sum_{i=1}^{n} w_i(x_i - \bar{x})(y_i - \bar{y}),$$
$$a_{11} = \sum_{i=1}^{n} w_i(x_i - \bar{x})^2, \quad a_{22} = \sum_{i=1}^{n} w_i(y_i - \bar{y})^2. \tag{1}$$

Here, the centroid of these pixels is

$$\bar{x} = \frac{1}{\sum_{i=1}^{n} w_i}\sum_{i=1}^{n} w_i x_i, \quad \bar{y} = \frac{1}{\sum_{i=1}^{n} w_i}\sum_{i=1}^{n} w_i y_i.$$

This matrix is real symmetrical, and its two nonnegative eigenvalues can be calculated by the following formula:

$$\lambda_{\text{big,small}} = \frac{a_{11} + a_{12} \pm \sqrt{\Delta}}{2}. \tag{2}$$

Here $\Delta = (a_{11} + a_{12})^2 - 4(a_{11}a_{22} - a_{12}a_{21})$.

The bigger eigenvalue λ_{big} corresponds to an eigenvector $\vec{v_b} = (x_b, y_b)$ which indicates the main direction of these pixels. The smaller eigenvalue λ_{small} corresponds to an eigenvector $\vec{v_s} = (x_s, y_s)$ that indicates the direction vertical to the main direction. Then the line can be fitted by these pixels and the line equation can be presented as

$$y_b(x - \bar{x}) - x_b(y - \bar{y}) = 0. \tag{3}$$

4.2. Borderline filter

Not all extracted borderlines are reliable to estimate the skew angle of a whole document image. There are several reasons causing some borderlines hard to estimate the skew angle such as devious edges of pictures, too many ascenders in text lines and degradation caused by strong noise, etc. Therefore, it is necessary to filter the borderlines and abandon unreliable ones. Our experimental researches indicate that most reliable borderlines have a point in common of straight enough.

It is easy to get the eigenvalues of a borderline according to formulas (1) and (2). The two eigenvalues correspond to two vertical directions, and the value of each eigenvalue denotes the contribution at its direction. If $\lambda_{\text{small}}/\lambda_{\text{big}}$ is not small enough, it indicates that there are some components at borderline's vertical direction and the linearity of the borderline is not good enough. Therefore, a parameter δ is defined to represent a borderline's linearity:

$$\delta = \frac{\lambda_{\text{small}}}{\lambda_{\text{big}}}. \tag{4}$$

To detect the intrinsic linearity of each borderline, the weight of each pixel is set to 1 firstly, then the parameter δ can be calculated by formulas (1), (2) and (4). If the parameter δ is larger than a given threshold, abandon the borderline. Fig. 3 shows the process of borderlines extraction and results of borderlines filtering. Pixels of a borderline are connected to display clearly. The borderlines in Fig. 3f are the ones which pass the filter.

4.3. Borderline optimization

Though a filter is applied to select linear borderlines, some borderline pixels are left around the ideal straight lines instead of fitting the line exactly. These pixels may come from wavy borders or noise, which always deviate from exact direction of the borderlines. These pixels may affect accurate detection of skew features. To reduce the effect of these pixels, a fast iteration algorithm is applied here to optimize the borderlines.

Because the optimized borderlines have passed the filter, the amount of deviated pixels is smaller than that of other pixels. For the deviated pixels keep distance from the fitted line, the distance between a pixel and the fitted line indicates how good this pixel is. If the better pixel is evaluated by a larger weight, the borderline will be optimized. A weight function can be defined as follows:

$$w(p) = \exp(-dist). \tag{5}$$

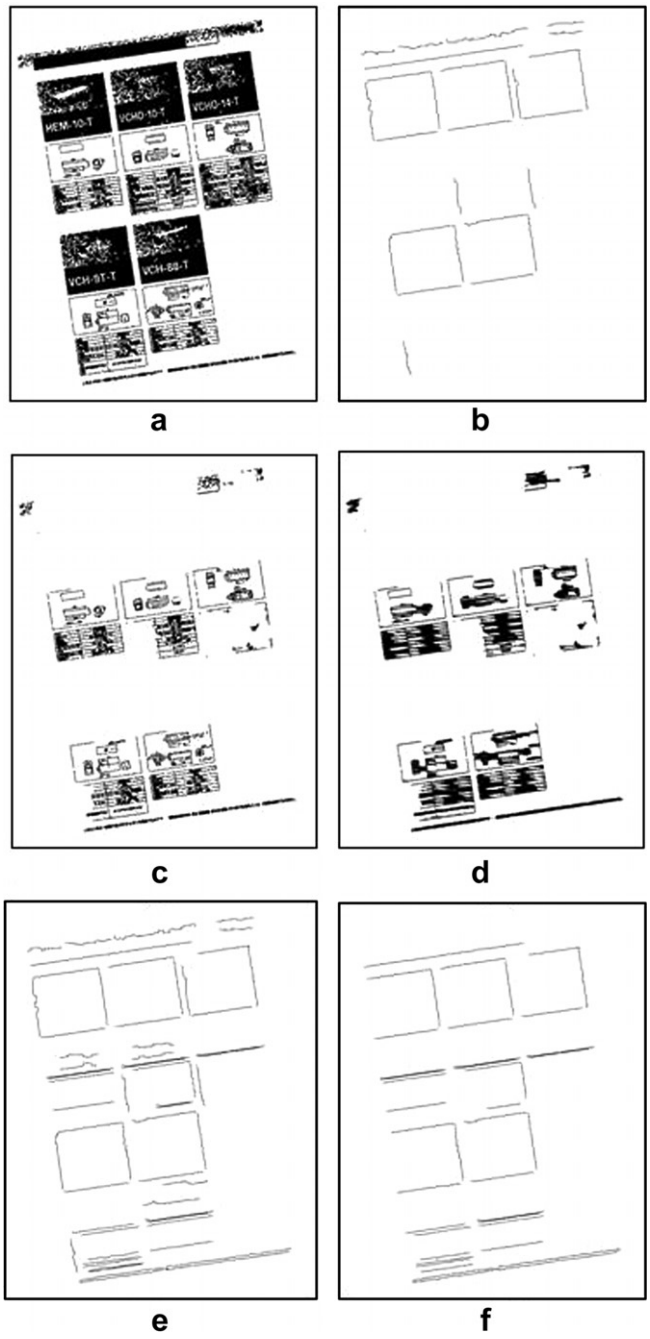

Fig. 3. Process of borderline extraction and filtering. (a) Document image after noise reduction and image enhancement; (b) Borderlines extracted from natural large connected components; (c) Document image after noise and natural large connected components removal; (d) Document image after run length algorithm; (e) Borderlines extracted from the whole image; (f) Borderlines after filtering.

Here *dist* is the distance between pixel p and the fitted line. The steps of borderline optimization are presented as follows:

Step 1: Calculate $skew\_angle_1$ of the borderline according to formula (3):

$$skew\_angle_i = \begin{cases} \tan^{-1} k, & |k| \leqslant 1, \\ \tan^{-1} k - \frac{\pi}{4}, & k > 1, \\ \tan^{-1} k + \frac{\pi}{4}, & k < -1, \end{cases} \quad i = 1, 2. \tag{6}$$

Here $k = y_b/x_b$ is the slope of the fitted line. The $Skew\_Angle_i$ is equal to 0 if $x_b = 0$.

Step 2: Update the weight of each pixel according to formula (5).

Step 3: Fit a new line and calculate $skew\_angle_2$ according to formula (1), (2), (3) and (6).

Step 4: If $|skew\_angle_1 - skew\_angle_2|$ is small enough, go to Step 5; else let $skew\_angle_1$ equal to $skew\_angle_2$, then repeat Step 2 to 4 till this iteration converge.

Step 5: The borderline's skew angle is equal to the mean of $skew\_angle_1$ and $skew\_angle_2$.

In experiments, this iteration algorithm always quickly converges. It only needs two or three iterations for most borderlines before convergence.

5. Skew angle calculation for a whole image

Now the skew angles of the optimized borderlines have been calculated. By using them to get a unique value as the skew direction of the whole document image, there are several choices:

Mean value is the simplest choice, but it is easy to be affected by a small quantity of "bad" skew features.

Weighted mean value, which is considered as a modification of mean value, is applied to calculate the skew angle of the whole image in some researches. This method differentiates the contribution of the skew features which comes from different regions according to weights. The weights are defined by the skew features of detected regions such as the length of the connected components, and the number of pixels of clusters. This method may improve the performance of mean value choice, but it may also be affected by some "bad" skew features.

Median value is another popular choice. Due to the inherent advantage of the immunity to a spot of "bad" skew features, its performance is generally better than mean value. Both the performance of mean value and median value are shown as experimental results in some literatures. The choice of median value is better than the mean value in most situations.

In our research, it can be observed that the longer the borderline is, the more reliable to estimate the skew angle of the whole image than shorter ones because the longer the borderline is, the more possible it is to be extracted from the page header, footer or a whole text lines with few waves. To take advantages of median choice and weighted choice together, a weighted median scheme is attempted in our method. The weight is the number of the sample pixels of one borderline as it has the same meaning with the length of a borderline. The weight indicates the mass of a borderline. As each optimized borderline has a skew angle and its own number of sampled pixels, the weighted median value is easy to be calculated.

Suppose there are N optimized borderlines and each borderline has a skew angle s_a_i and a mass m_i, $i = 1, 2, \ldots, N$. The weighted median skew angle can be calculated by the following steps:

Step 1: Calculate the number of skew angles, which is marked as *Angle_Sum*.

$$Angle\_Sum = \sum_{i=1}^{N} m_i \tag{7}$$

Step 2: An array *Weight_Angle* with length *Angle_Sum* is established, which contains the skew angles. The number of s_a_i in the array is m_i. Then sort the array.

Step 3: If *Angle_Sum* is an odd number, the weighted median value is the K_1^{th} value of the array, else the weighted median value will be the mean of the K_2^{th} value and the K_3^{th}

value of the array, where $K_1 = (Angle\_Sum + 1)/2, K_2 = Angle\_Sum/2$ and $K_3 = (Angle\_Sum + 2)/2$.

This weighted median value is regarded as the skew angle of the whole image.

6. Experimental results and analysis

Since March 2003, a large database including 10385 document images has been established step by step in our lab to develop a document image retrieval system. Those documents are taken from practical documents including technical articles, official documents, manuals and magazines. They contain mixed contents of pictures, graphs, tables, complex directions of horizontal and vertical text, complex text with different layouts, fonts and hybrid characters in Chinese, English and Japanese. About 30% documents of those are in Chinese, 20% in English and 15% in Japanese, respectively. Others are documents with mixture of languages or predominant non-text regions. More than 60% documents contain non-text regions. The inborn document images are 200 dpi of A4 paper size. To verify our method, 2000 document images are randomly selected from the database, which are categorized into three groups: (A) text regions are predominant; (B) text and non-text regions are approximately equiponderant; (C) text regions are sparse and non-text regions are predominant. There are 1079 document images that belong to group (A), 700 and 221 images that belong to group (B) and (C), respectively. PhotoShop 7.0 is used to rotate each image with a random angle within $[-10°, 10°]$.

The values of the thresholds in our method are: $rl\_th = 25$, $\max\_fall = 10$, $\min\_length = 10$, $\min\_size = 100$ and the threshold of δ is 0.001. All of the above thresholds are obtained from experiments in a smaller training database, and they are for down sampled document images in 100 dpi. The training database is composed of 300 document images, which is randomly selected form the whole database besides the 2000 documents selected for comparison experiments. Experiments show that the thresholds can be used for the larger database without obvious performance variation.

Another important threshold ε is applied to compute correct rate. The correct rate is the percentage of the detection results whose absolute errors are less than ε. Here $\varepsilon = 0.1°$. The value of this threshold is man-made to scale the performances of the methods, since the mean and standard deviation are not important enough in some situations that the methods can not get a result. Experiments are implemented on a PC with 2.8 GHz CPU and 512 MB memory. The method is programmed by C++ language.

According to the basic characteristics of large scale training database (thousands), complex layout and multiple language characters in our database, several representative methods are selected to evaluate our method. We compare ours with three kinds of representative methods to cover the discussed methods in Section 1 to a great extent by implementing Lu's method (Lu and Tan, 2003), Das's method (Das and Chanda, 2001) and a method based on Hough Transform. First, we implement their methods step by step according to the presented literature. Second, necessary modifications are made in order to make the methods suitable for the database. To design this method based on Hough transform, the ideas of Hinds et al. (1990), Le et al. (1994) and Amin and Fischer (2000) are used for reference.

Table 1 shows the detection results of different methods. The number in the parentheses is the number of document images in the category. From the table, it can be found that the performances of our method are superior to the three other methods both in correct rate, and in accuracy of mean and SD of absolute error. From Table 1a, it can also be found that the performances of group (C) drop obviously except our method. It shows that our proposed method is more effective and robust for the document images with predominant non–text regions or sparse text distribution (see Table 1b).

Table 1a
Correct rates of different methods

	A (1079)	B (700)	C (221)	Total (2000)
Das's method (%)	61.5	63.1	57.5	61.2
Lu's method (%)	72.1	69.6	67.9	70.8
Hough Transform method (%)	76.8	71.1	62.4	73.3
Our method (%)	95.7	98.0	97.7	96.8

Table 1b
Mean and SD (standard deviation) of absolute error of different methods

	Das's method	Lu's method	Hough Transform method	Our method
Mean (°)	0.130	0.122	0.118	0.0240
SD (°)	0.620	0.611	0.405	0.0635

In our experimental document database, thousands of documents are in Chinese, which are not mainly considered and tested in most existing methods. As discussed in Section 2, there are both head lines and bottom lines in Chinese characters, different from many Western languages such as English, Indian, et al. Meanwhile, our database is combined with complex documents in not only Chinese, but also Japanese and English. It is one of the important reasons that our method gets obviously better performances than other representative methods. Comparison experiments also show that it is hard to improve performances to a great extent for such complex testing database by updating some existing thresholds only during the methods are compared.

Furthermore, to analyze the efficiency of important steps in our method, a group experiments E1–E5 are designed for comparison. As shown in Table 2, we implement four kinds of combinational experiments, including "Whether extract borderlines from natural large connected components", "Whether filter the extracted borderlines", "Whether use the iteration algorithm to optimize the borderlines" and "Whether median method or weighted median method is used to calculate the skew angle of the whole images". Therefore, E1 is a benchmark and E2 shows the effect of extracting borderlines from natural large connected components, which are likely to be formed in non-text regions. E3 shows the efficiency of the borderlines filter and E4 gives the effect of the iteration optimization. Efficiency of the weighted median method is given in the row of E5.

Table 2 also shows the correct rates of the five experiments. Performance of the benchmark E1 is not good, especially in the category (C). In fact, Since E1 does not extract borderlines from the

Table 2
Conditions and correct rates of comparison experiments

	W	X	Y	Z	A (1079) (%)	B (700) (%)	C (221) (%)	Total (2000 (%))
E1	No	No	No	Median	85.3	87.3	53.8	82.5
E2	Yes	No	No	Median	81.8	85.3	82.8	83.1
E3	Yes	Yes	No	Median	91.2	94.6	92.8	92.6
E4	Yes	Yes	Yes	Median	94.3	96.9	94.6	95.2
E5	Yes	Yes	Yes	Weighted median	95.7	98.0	97.7	96.8

W: Whether extract borderlines from natural large connected components.
X: Whether filter the extracted borderlines.
Y: Whether use the iteration algorithm to optimize the borderlines.
Z: Whether median method or weighted median method is used to calculate the skew angle of the whole images.

likely non-text areas, no skew feature can be found to detect the skew angle for 31 testing document images. However, the performance is better than the other three methods in category (A) and (B). It is caused by the following reasons. First, as the orientation and run length threshold are determined dynamically, the run length algorithm of our method is effective to smear the characters of the same text lines in various situations. It makes our method having better foundation to extract reliable skew features. Second, the idea of extracting borderlines from all the four borders of large connected components makes our method more robust for documents in different languages. Third, the fall detection of two neighboring sampled pixels, which is mentioned in Section 3.2, avoids much potential wrong extraction of borderlines. Finally, although the schemes of fitting lines according to formulas (1), (2), (3) and (6) with the same weights and the median value method to determine the skew angle of a whole image are improved in our method, they also make the method efficient to estimate the skew angle accurately.

From E2 it can be found that the performance in category (C) is improved greatly. Moreover, a skew angle can be detected for all documents in this experiment. The step of extracting skew features in non-text area makes the method robust for the document images with predominant non-text areas. The performance of E2 has a little decline for category (A) and (B) because some of the un-filtered borderlines extracted from the non-text areas may not be robust enough. When the borderline filter is used, the correct rate increase about 10% for all categories according to E3. It indicates that the filtering of the borderlines is effective and necessary. E4 shows that the iteration optimization will also contribute to improve the performance of our method. E5 shows the results of our tentative idea that uses weighted median value to define the skew angle of the whole image. The weighted value is superior to the median value in the experiment.

From experiments of E1 to E5, three points can be concluded as follows. (1) Extracting borderlines from non-text areas can provide useful skew information for many document images, especially for the document images without predominant text areas. (2) The step of borderlines filter plays a key role in our method. It supports the main idea of our method that skew features should be extracted from both text and non-text areas through filtering the bad borderlines. (3) It is a good way to adopt weighted median value if the weights are appropriate.

Table 3 shows the accuracy of our method including the mean and the SD of absolute errors of different categories.

One hundred document images are selected randomly to find the effective detecting range of the proposed method. The performance is as good as the above experimental results at the skew angle of 15°, and it begins to degenerate when skew angles increase to 20°. As the orientations of the document images are determined automatically and the run-length algorithm is only applied to the determined orientations in the proposed method, the characters of the same text line may not be smeared well and the characters of different text line may be smeared for large skew. It is a limitation of the proposed method, but the general skew angle is limited to ±5° for practical systems (Das and Chanda, 2001), which is within the well disposed range of the proposed method.

Table 3
Mean and SD (standard deviation) of absolute error of our method for different categories

	A	B	C	Total
Total number	1079	700	221	2000
Mean (°)	0.0282	0.0180	0.0225	0.0240
SD (°)	0.0801	0.0299	0.0466	0.0635

Another possible limitation is that there are several thresholds in the proposed method, which may cause adaptability problems. First, all the thresholds except ε are determined by some attempts on a much smaller database, then they are applied to the proposed method in the experiments. As the experimental results on a very large database show the determined thresholds work well in various situations where values are optimum or the problem on how to set the thresholds automatically are not considered any more. In other words, they do not work well in some situations.

The threshold ε is a man-made threshold, which is applied to scale the performances of the methods. It shows the probability that the method detect the skew angle of a document image within the error of ε. As some methods can not work out a result for some document images of the database, especially for the images of Group (C), it is not all-sided that only use mean and SD of the absolute error to scale the performances of the methods. The threshold ε is a helpful supplement which can scale the performances effectively from another side.

Some example document images are shown in Fig. 4. Fig. 4a is a document with Japanese vertical layout. Fig. 4b is a document with predominant non-text regions and text in this image is rather sparse. Fig. 4c illustrates a document with two big non-text regions and a paragraph of Chinese characters. There is no obvious frame in the figures and some of the borders of the non-text regions are bad for skew detection. Fig. 4d is a document with many non-text regions, mixed with Japanese and English characters. Most of previous methods will be ineffective for some of the above examples

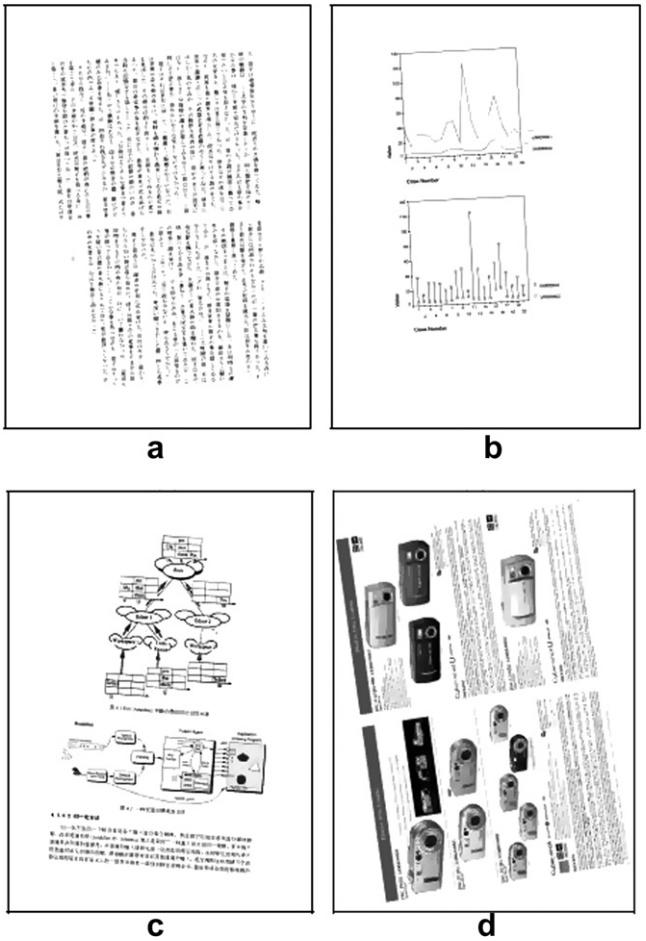

Fig. 4. Example document images.

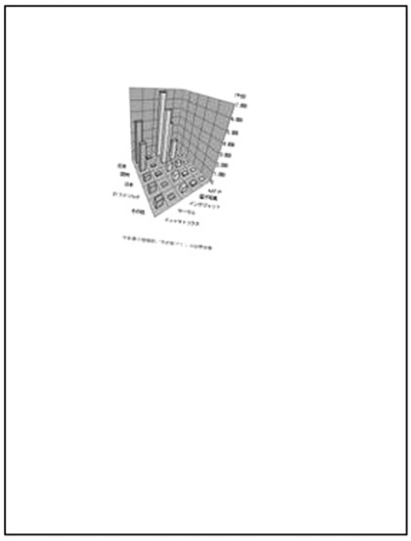

Fig. 5. An example of failed document.

due to the complexity of the images, especially for Fig. 4b. However, our method can deal with all these images well.

As skew features are extracted from both text and non-text regions, a potential weakness of our method is that it may be affected by some straight lines which are not parallel or vertical to the skew directions in non-text regions. However, it will not affect the accuracy in most factual situations. First, if the straight lines exist in the regions which have whole borders as frames, they will not be extracted because only the borders of connected components are extracted in our method. Second, even if some borderlines are extracted, the affection can be shielded if the amount is small and the lengths are short relative to good ones due to the statistical property of weighted median methods. Finally, the only situation that the straight lines would affect the accuracy is when the lines are predominant. That means the text regions are sparse and most non-text regions are unreliable for skew detection. In this situation, not only most of other methods will be ineffective, but also human beings can hardly estimate the skew angle. In fact, this situation appears rather rarely in factual document images. Fig. 5 shows an example of failed-detecting documents by proposed method.

Time performance of our method is also tested in the experiments. It takes 0.142 s on average to detect the skew angle of a document image. That will support real-time document analysis system well.

7. Conclusions

In this paper, based on the idea of extracting skew features from not only text but also non-text regions, a new concept of robust borderline is proposed to represent reliable skew features in the whole document. From observations and analysis, it can be seen that the idea is reasonable and robust borderlines popularly exist in both text and non-text regions. Experiments on lots of various document images show that combining linear filter and iteration optimization is fast and feasible to extract robust borderlines and document skew angles. Experiments also show that the skew detection method based on robust borderlines is accurate and fast for complex document images, especially for documents with sparse text and predominant non-text regions.

Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC 60675025) and the National High Technology Research and Development Program of China (863 Program, No. 2006AA04Z247). This research was also supported by the joint project between Ricoh Co., Japan and Peking University.

References

Akiyama, T., Hagita, N., 1990. Automatic entry system for printed documents. Pattern Recognition 23 (11), 1141–1154.
Amin, A., Fischer, S., 2000. A document detection method using the Hough Transform. Pattern Anal. Appl. 3, 243–253.
Amin, A., Wu, S., 2005. Robust skew detection in mixed text/graphics documents, In: International Conference on Document Analysis and Recognition, vol. 1, pp. 247–251.
Cao, Y., Wang, S., Li, H., 2003. Skew detection and correction in document images based on straight-line fitting. Pattern Recognition Lett. 24, 1871–1879.
Chaudhuri, B.B., Pal, U., 1997. Skew angle detection of digitized indian script documents. IEEE Trans. Pattern Anal. Mach. Intell. 19, 182–186.
Chen, M., Ding, X.Q., 1999. A robust skew detection algorithm for grayscale document image, In: Proceedings of 5th international conference on document analysis and recognition, pp. 617–620.
Das, A.K., Chanda, B., 2001. A fast algorithm for skew detection of document images using morphology. Int. J. Document Anal. Recognition, 109–114.
Dhandra, B.V., Malemath, V.S., Hangarge, Mallikarjun, Hegadi, Ravindra, 2006. Skew detection in binary image documents based on image dilation and region labeling approach. ICPR (2), 954–957.
Hashizume, A., Yeh, P.S., Rosenfeld, A., 1986. A method of detecting the orientation of aligned components. Pattern Recognition Lett. 4, 125–132.
Hinds, S.C., Fisher, J.L., D'Amato, D.P., 1990. A document skew detection method using run-length encoding and Hough Transform, In: Proceedings of 10th International Conference on Pattern Recognition, pp. 464–468.
Ishitani, Y., 1993. Document skew detection based on local region complexity, In: Proceedings of 2nd International Conference on Document Analysis and Recognition, pp. 49–52.
Le, D.S., Thoma, G.R., Wechsler, H., 1994. Automatic page orientation and skew angle detection for binary document images. Pattern Recognition 27, 1325–1344.
Lu, Y., Tan, C.L., 2003. A nearest-neighbor chain based approach to skew estimation in document images. Pattern Recognition Lett. 24, 2315–2323.
O'Gorman, L., 1993. The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. 15 (11), 1162–1173.
Safabakhsh, R., Khadivi, S., 2000. Document skew detection using minimum-area bounding rectangle, In: Proceedings of International Conference on Information Technology, pp. 253–258.
Shi, Z.X., Govindaraju, V., 2003. Skew detection for complex document images using fuzzy runlength, In: Proceedings of 7th International Conference on Document Analysis and Recognition, pp. 715–719.
Shivakumar, P., Kumar, G.H., Guru, D.S., Nagabhushan, P., 2005. A new boundary growing and Hough Transform based approach for accurate skew detection in binary document images, In: International Conference on Intelligent Sensing and Information Processing, pp. 140–146.
Yan, H., 1993. Skew correction of document images using interline cross-correlation. Computer Vision Graph. Image Process. 55 (6), 538–543.