New Skew Detection and Correction Algorithms

Ricky Shiu and Adnan Amin
School of Computer Science and Engineering
University of New South Wales, Sydney N.S.W. 2052
Australia
Email: amin@cse.unsw.edu.au

Abstract

Document image processing has become an increasingly important technology in the automation of office documentation tasks. Automatic document scanners such as text readers and OCR (Optical Character Recognition) systems are an essential component of systems capable of those tasks. One of the problems in this field is that the document to be read is not always placed correctly on a flat-bed scanner. This means that the document may be skewed on the scanner bed, resulting in a skewed image. This skew has a detrimental effect on document analysis, document understanding, and character segmentation and recognition. Consequently, detecting the skew of a document image and correcting it are important issues in realizing a practical document reader.

In this paper, we describe two new algorithms - one for skew detection and one for skew correction. The new skew correction algorithm we propose has been shown to be fast and accurate, with run times averaging under 1.5 CPU seconds and 30 seconds real time to calculate the angle on a 5000/20 DEC workstation. We have tested with more than 100 images with extremely promising results.

1. Introduction

Over the past few years many researchers have explored methods on detecting the skew angle of a page document. The main motivation for this effort has come from the need to have an unskewed digitized document image for the subsequent stages of processing such as segmentation and optical character recognition of a document image analysis system. Efficient and reliable method for detecting the skew angle of a composite page document containing mixed text, graphics, line segments and other forms together is becoming more important in this evolving research area.

Many methods have been proposed to detect the skew angle of a page document. For example, Srihari and Govindaraju [1], Le et al [2] and Hinds et al [3] and utilized the Hough transform approach while Baird [4], Akiyama and Hagita [5] and Pavlidis and Zhou [6] used the projection profile technique. Moreover, Postl [7] used two-dimensional Fourier transform and Bessho et al [8] used the correlation and regression method and Ishitani [9] used the complexity variance. Although the Hough transform is computationally expensive, it is accurate, robust, reliable and noise insensitive for skew detection.

In this paper, we present a new algorithm for skew angle detection by using the Hough transform. The literature presented techniques based on the Hough transform process an excessive amount of data. The speed of processing is proportional to the amount of data required. The new algorithm reduces the amount of data required for detecting the skew angle by using the bottom most connected components thus the speed for skew detection can be enhanced while preserving the accuracy in determining the skew angle.

The algorithm can be divided into four major components. The first step is digitisation in which the original image is transformed into a gray level image utilising a 300 dpi (dot per inch) scanner. Second, an appropriate threshold is chosen to separate the foreground objects such as characters and the background image such as noise. Third, a skew detection algorithm is used in which the dominant skew angle of the page document is detected. Finally, a skew correction technique is applied for which the skewed image is rotated to give an upright image. Figure 1 illustrates the block diagram of the system.

Figure 1: System diagram of the skew detection and correction processes
2. Digitisation and Thresholding

Each page document is converted to a digital image using a 300 dpi scanner and stored as a gray level image. An appropriate threshold value must be found for the binarization of a gray level image. At this point it is desirable to preserve the character, and a small amount of scattered noise is acceptable. The thresholding technique is applicable to a document image with several colors.

The program generates a histogram from all pixel values and smooths it by a sliding average of the number of pixels of each gray scale intensity. Then it checks for local maxima. The two maxima with the greatest number of pixels are then chosen to represent the gray scale levels for the foreground objects such as characters and background image (M₁ and M₂, respectively). The local maxima between these two points are then checked, and if the number of pixels of one of these gray scale values is greater than 1/N times the number at value M₁, where N is a predetermined value, this level (M₃) is chosen to represent an average value of the third dominant color of the image. We have found that N = 5 works well with the data without detecting spurious noise thresholds. To find the cutoff level for the background, Otsu’s algorithm [10] is then applied to find a threshold between M₃ and M₂, if M₃ exists, otherwise between M₁ and M₂. We assume that distinct and widely separated peaks can be found in most documents, thus Otsu’s global non-parametric thresholding technique is adequate.

This skew detection and correction system is designed to handle gray level images. As gray-scale are becoming increasingly prevalent, digitized document images are in gray-scale, there is a growing need to handle these documents. Apart from the gray level images, the algorithm also works on binary and color images with appropriate modifications to the digitisation and thresholding stages. The skew detection algorithm can be similarly modified.

3. Skew Detection

In this paper, we present a new algorithm to determine the skew angle of the document. The skew detection process can be divided into four stages: 1) creation of connected components; 2) extraction of bottom most connected components; 3) the Hough Transform and 4) determination of skew angle.

3.1 Creation of Connected Components

Connected components are rectangular boxes bounding together regions of 8-connected black pixels. The objective of the connected component analysis of an image is to form rectangles around distinct components on the page, whether they be textual elements such as characters or non-textual elements such as images. These bounding rectangles then form the skeleton for all future analysis on the page. An illustration of an image and its connected component skeleton is shown in Figure 2.

![Figure 2: Example of digitized image of business card and the corresponding connected components](image)

The algorithm used to obtain the connected components is a simple iterative procedure which compares successive scanlines of an image to determine whether black pixels in any pair of scanlines are connected together. Bounding rectangles are extended to enclose any groupings of connected black pixels between successive scanlines, Figure 3 demonstrates this procedure.

![Figure 3: The process of building connected components from image scanlines](image)

Each scanline in Figure 3 is 14 pixels in width. Each pixel is represented by a rectangular box. The bounding rectangles in Figure 3a show the enclosure of the black pixels of that scanline. For each successive scanline, the size of bounding box increases to enclose the black
pixels connected to the previous scanline. Figure 3c shows that a bounding box stops growing in size only when there are no more black pixels on the current scanline joined onto black pixels of the previous scanline.

3.2 Extraction of Bottom most Connected Components

To determine the dominant skew angle of a digitised document image, the bottom most connected components are extracted for skew detection process. There are three major steps in the extraction process. First, the connected components are saved as a list of rectangles at the stage of formation of connected components. Second, the page is divided into vertical segments of width equal to the average width of connected components. Non-textual data such as noise and images are the major source of skew error. In order to minimize the skew error, connected components with width and height between 1 and 63 pixels and area between 4 and 3907 pixels$^2$ are considered as characters while the other connected components are considered as noise or images. These parameters are the empirical measurements and given by [3]. The non-textual connected components are excluded in the following projection process. The lower bounds are effective in excluding the scattering salt-and-pepper noise so that this process allows the algorithm to work with unclear document images.

Third, every textual connected component is projected into the appropriate bin slot. If another connected component already exists in the bin location, the connected component closest the baseline is allowed to stay in the bin slot while the other connected component is discarded. Until all connected components are processed, this process allows us to store primarily the bottom row(s) of text in the page document.

Figure 4a shows the original digitized image and Figure 4b shows the extracted bottom most connected components of the page document. We can see that the image shown on Figure 4a is distorted by noise. The scattering salt and pepper noise, at the bottom part of the original image, is discarded in this extraction process in order to minimize the errors in the detection of skew angle.

3.3 The Hough Transform

We then apply the Hough Transform [11] to the point at the centre of the bottom of each rectangle, therefore mapping the point from the (x,y) domain to a curve in the (p,θ) domain according to the equation:

$$p = x \cos \theta + y \sin \theta$$

for $0 \leq \theta < \pi$

The Hough Transform has several interesting properties:
1. Points in the (x,y) domain map to sinusoidal curves in the (p,θ) domain.
2. Points in the (p,θ) domain map to lines in the (x,y) domain, where p is the perpendicular distance of the line from the origin, and θ is the angle from the horizontal of the perpendicular line.

3. Curves that cross at a common point in the (p,θ) domain map to collinear points in the (x,y) domain.

3.4 Determination of Skew Angle

In determining the skew angle of the page, the Hough Transform is applied to each point and the resulting Hough accumulator array is examined to find the two points at which most curves cross. The Hough accumulator array is examined once again and all the cell array elements of which the values cross are greater than one-half of the second maxima are added together based on their angle. The maximum of these values corresponds to the dominant skew angle of the page document.
4. Skew Correction

In order to achieve the preciseness, a new technique has been developed. We use a new method inspired by [12] to calculate the correct gray value for a pixel in the skew corrected image. We calculate the correct value for a pixel at location \((x,y)\) in the skew corrected image, then we determine the true original position \((X_{skewed \rightarrow \text{skewed}}, Y_{skewed \rightarrow \text{skewed}})\) in the skewed image.

\[
X_{skewed} = x \cos \alpha + y \sin \alpha
\]

\[
y_{skewed} = y \cos \alpha + x \sin \alpha,
\]

where \(\alpha\) is the calculated skew angle of the image. Since \((X_{skewed \rightarrow \text{skewed}}, Y_{skewed \rightarrow \text{skewed}})\) is generally non-integral, a weighted average of gray values of the four surrounding pixels is calculated to determine the gray value of pixel \((x,y)\). The contribution of each pixel is the product of the relative vertical and horizontal distances between \((X_{skewed \rightarrow \text{skewed}}, Y_{skewed \rightarrow \text{skewed}})\) and the corresponding surrounding pixel. Figure 5 (refer to the last page) shows the relationship between the position of pixels in the skewed image and the skew corrected image. The corresponding contribution of the four surrounding pixels \((X_{int \rightarrow \text{int}}, Y_{int \rightarrow \text{int}}), (X_{int+1 \rightarrow \text{int}}, Y_{int+1 \rightarrow \text{int}})\) and \((X_{int \rightarrow \text{int}}+1, Y_{int \rightarrow \text{int}}+1)\) are \((1-\delta x)^*(1-\delta y)^*f(x_{int \rightarrow \text{int}}, y_{int \rightarrow \text{int}}), 1-\delta x)^*(1-\delta y)^*f(x_{int \rightarrow \text{int}}+1, Y_{int \rightarrow \text{int}}), (1-\delta x)^*\delta y^*f(x_{int \rightarrow \text{int}}, Y_{int \rightarrow \text{int}}+1)\) and \(\delta x^*\delta y^*f(x_{int \rightarrow \text{int}}+1, Y_{int \rightarrow \text{int}}+1)\) respectively where \(f(x,y)\) is the gray-scale value of the pixel \((x,y)\).

![Figure 5: Weighted contribution of each surrounding pixel](image)

If \((X_{skewed \rightarrow \text{skewed}}, Y_{skewed \rightarrow \text{skewed}})\) lies outside the original image, the new pixel is set to white. Figure 6a shows a portion of the skewed image and Figure 6b shows the skew corrected image. This method retains a high degree of information and there is no degradation in the quality of the characters.

5. Experimental Results

Over 100 unconstrained document images were used in the experiments. These images were chosen from a wide range of sources including technical reports, Arabic, Chinese and English magazines and business documents, business cards, Australian Telecom Yellow pages and badly degraded photocopies. These images contain a wide variety of layouts, including sparse textual regions, dense textual regions, mixed fonts, multiple columns, tabular and even for documents with very high graphical contents.

The experiment is performed by detecting the skew angle of the rotated original unskewed image from the set of testing images. Some of these testing images are rotated at several angles up to 45°. The skew detection error is the difference between the rotation angle and the detected skew angle. The time taken is measured from the stage of formation of connected components to the completion of skew angle detection (not including the skew correction stage). In order to obtain a better and consistent measurement on the processing time, we run the algorithm 10 times on each testing image. The average of these 10 periods is determined. The four most deviated data set are discarded. The average of the

![Symmetric Phase-Only of Fourier–Mellin Image Registration](image)

Figure 6: a) The skewed image (upper); and b) the skew corrected (without image degradation) images (lower)
remaining time periods is calculated again and this refers to the final average of time taken of the skew detection process of the image.

All scanned images have been processed with 100% skew correction. The skew detection process works up to 45° of skew angle. With the Hough Transform being set at 0.2° angular resolution, the algorithm is fast requiring on average less than 1.5 CPU seconds to determine the skew angle of a page on a DEC 5000/20 workstation. The average absolute skew error is 0.11°.

In addition, we have implemented well-known other methods such as [1,2,3] which used the Hough transform for comparison. The new method is substantially faster than any of these methods (Figure 7 - refer to the last page) without decreasing the accuracy in determining the skew angle. For documents with a high graphical content, our method is more accurate and robust than the algorithm presented in [3] and faster than the algorithm which appeared in [2]. However, Srihari's algorithm [1] is not included in the graph for many reasons. First, the method is very slow comparing with the other two methods since the method applies the Hough transform to every pixel of the image. Second, the method works for textual images only but many documents contain textual blocks interwined with images.

Figure 8 and Figure 9 show a chinese magazine document image with different colors skewed at 15.0° as well as some background noises and a business card image skewed at -37.0° respectively, the bottom most connected components and the skew corrected images.

**Figure 8:** A chinese magazine document image skewed at 15.0° (upper); the bottom most connected components (middle); and the skew corrected image (lower).

**Figure 9:** Digitized image of business card skewed at -37.0° (upper); the bottom most connected components (middle); and the skew corrected image (lower).

### 6. Conclusion

The Hough transform is accurate and reliable in detecting the skew angle of a page document but it is computationally expensive and requires a large amount of memory during processing. The literature presented techniques [1,2,3] based on the Hough transform process an excessive amount of data. In this paper, the new technique has been presented which reduces the amount of data required for detecting the skew angle thus the speed for skew detection can be increased while preserving the accuracy in determining the skew angle. The improvement is mainly as a result of working with the bottom-most connected components of a page rather than with pixels (which range in the order of 6-8 million for an A4 document scanned in 300 dpi) or with every
connected component (approximately 3000-5000 components). The set of bottom-most connected components contains typically about 200 connected components.

In addition, a new interpolation technique for determining the gray level of a pixel in the skew corrected image. The experiments have shown that there is no significant image degradation. This is important to the reliability and accuracy in recognizing characters by the current Optical Character Recognition systems.

In the implementation, we have projected the documents with skew angles up to 45°. If we restrict the documents with skew angle up to 15°, the time taken can further be reduced and expected to be approximately one third of the time required in the current setting.

References


A Comparison on the speed of three different algorithms

Figure 7: A comparison on the speeds of the three algorithms