Skew Correction for Text document Using Fourier Domain Analysis
Chang, Wen-Teng
Department of Electrical Engineering and Computer Science,
Case Western Reserve University, Cleveland, OH, Email: wxc31@cwru.edu

Abstract
Skew estimation and correction is an important step in any document analysis and recognition system. This project presents an algorithm for skew recognition and implements it by Fourier domain analysis. The error estimation and process time are also presented in this project. We suppose the skew angles are within $\pm 45^\circ$ and then analyze the deviation between the corrected one by rotating a known angle.

KEYWORDS
Skew correction, Fourier transform, Frequency domain

INTRODUCTION
Skew estimation and correction is an important step in any document analysis and recognition system. It is quite common when we are digitizing a copy of an original document of text which is skewed by unknown angle. The skewed angle sometimes will severely degrade the performance of the systems.

A wide variety of methods of skew detection algorithms have been proposed in the literature. Basically they can be summarized in five categories [1] (1) project profile, (2) Hough transform technique [2] (3) Fourier method (4) nearest-neighbor clustering and by (5) correction.

Humans can determinate the document orientation without recognizing the text or contents of the document. This idea, however, suggests orientation can be implemented from the global description of the document without looking into the local details of the documents. An algorithm by Fourier transform can exactly extract the predominant direction which is derived from the edge of photo, text or diagram in the document. This project proposes a method based on Fourier domain analysis.

OVERVIEW
This project uses 2 images downloaded from EECS490 website, one is simply text and the other is hand written script with vertical and horizontal mesh that are shown as figure 1 and figure 2, individually.

By rotating an image by a known angle (‘imrotate’), we use Fourier transform to analyze its rotation angle. Here have several methods to find out the angle. One is to find its maximum histogram among the slopes; the other is to find its mean or median value. The rotation angle is converted from slope rate and then got the upright image. Finally, the un-skewed document will need padding its board because the board is surrounded with all black. Figure 3 is the block diagram of this correction process.
**ALGORITHM**

This section will discuss the process flow of orientating a skew document. All the algorithms used for skew correction are based on estimating the predominant orientation of the text in the spatial domain. When we read an image or a document, the digitized readings combine an M by N matrix with unsigned 1 to 256 gray level values.

1 **Fourier Transform**

When we transform the images shown upright and skewed one from spatial domain image of figure 2 to frequency domain, the images show upright and skew Fourier spectrum as figure 4 (a)~ 4 (d).

![Figure 4 (a)]() ![Figure 4(b)]()

![Figure 4(c)]() ![Figure 4(d)]()

Figure 4 (a) is an upright document whose frequency domain is upright as figure 4(b); figure 4(c) rotates 10 degree and its frequency domain rotates also.

The discrete Fourier transform of 2D image $f(x,y)$ of size MxN is given by[2]

$$
F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-2\pi j \frac{ux}{M} \frac{vy}{N}}
$$

(1)

2 **Set threshold**

In order to utilize the frequency domain to obtain skew angle, we convert it to binary image and set a threshold that is optimized to eliminate bright point that will help shorten calculation time in finding slopes. Median and maximum value are used in this algorithm that can assure the threshold a reasonable value which simplifies to find real rotation angle.

![Figure 5 Binary image of the skew document in frequency domain](figure)

3 **Slope analysis**

There are several ways to find out optimal slope of figure 5. One is get a histogram of the slopes that are distributed on figure 6, the slope with the most distributed bright point is desired slope rate.

![Figure 6 Histogram of slope distribution](figure)

Alternatively, mathematic methods can be used to get desired value. Function such as “mean” and “median” can get similar answers. The advantage of using histogram is more accuracy can be acquired by dividing more bins on horizontal axis.
4 Un-skew correction

We can easily convert desired slope rate to angle and rotate this angle in opposite direction. There might be some deviation after correction. The latter part will discuss this question. However, this algorithm will limit skew angle within ±45° because we regard document should orientate toward the other direction if angle is over 45°.

Figure 7 Unskew picture by correcting angle on frequency domain

5 Board padding

To fix the picture on a given box, we need to pad the black board of figure 7. The padding result is shown on figure 8. Note the padding is not completed because the corrected image is not exactly upright.

Figure 8 padding result of figure 7

RESULT DISCUSSION

In this section, we will evaluate the gap between corrected angle and true angle from this algorithm. We also calculate the required process time. The process time includes the time of Fourier transform, image presentation of skew and un-skew, histogram and board padding [see appendix]. This analysis uses figure 1, 512x512 pixels gray level image. Table 1 is the summary of the result.

From the result, we can find that the gap between corrected angle and true rotation angle roughly increases along with rotation angle, so the corrected angle may be not satisfying if the skew angle is too large. One way to resolve this problem is to do another Fourier transform. However, it will take approximate double process time according to the result of table 1. Meanwhile, the process time is highly correlated with rotation angle: the higher skew degree requires more time to process the correction and the required time is roughly symmetric to central zero degree.

<table>
<thead>
<tr>
<th>True angle (°)</th>
<th>Detected angle (°)</th>
<th>Required time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-30</td>
<td>-27.474</td>
<td>3.609</td>
</tr>
<tr>
<td>-20</td>
<td>-18.434</td>
<td>3.515</td>
</tr>
<tr>
<td>-10</td>
<td>-8.972</td>
<td>2.844</td>
</tr>
<tr>
<td>-5</td>
<td>-4.398</td>
<td>3.219</td>
</tr>
<tr>
<td>-3</td>
<td>-2.726</td>
<td>3.891</td>
</tr>
<tr>
<td>-1</td>
<td>-1.273</td>
<td>2.766</td>
</tr>
<tr>
<td>1</td>
<td>1.735</td>
<td>2.719</td>
</tr>
<tr>
<td>3</td>
<td>3.366</td>
<td>2.843</td>
</tr>
<tr>
<td>5</td>
<td>5.527</td>
<td>2.813</td>
</tr>
<tr>
<td>10</td>
<td>9.782</td>
<td>2.906</td>
</tr>
<tr>
<td>20</td>
<td>18.434</td>
<td>3.266</td>
</tr>
<tr>
<td>30</td>
<td>26.474</td>
<td>3.906</td>
</tr>
</tbody>
</table>

Table 1 Test result of (1) corrected angle and (2) process time at different rotation angle

Figure 9 Rotation angle versus detected angle ranged from -30° to 30°. The square points denote the corrected angle; the line indicates true angle.

SUMMARY

The required time by using frequency domain to detect skew angle is highly correlated to skew angle. Besides, the higher skew degree yields roughly larger degree deviation, however, even it cannot obtain a satisfying corrected angle at first correction, the second correction process can reach as good accuracy at least less one degree. For example, if the rotation angle is 30° in this case, we only have to repeat the same procedure so that the accuracy can be further improved to 0.3° deviation at the second time. Nevertheless, comparing to other literature, it may take relatively long
time [4], which may be a disadvantage on the application such as on OCR, which requires fast recognition.

In the result, we find the corrected angle increases roughly along with rotation angle, so there still is limitation for high angle rotation angle. However, Fourier transform can recognize the skew angle without knowing the detailed context; the predominant direction will be extracted form the edge of the text. This is a major feature and advantage of using this algorithm.

REFERENCES


Image De-skewing Using Fourier Transform

Dmitriy Goldman
Department of Electrical Engineering and Computer Science,
Case Western Reserve University, Cleveland, OH, Email: dxg68@cwru.edu

ABSTRACT
When the document is scanned it can be placed on the scanner at arbitrary angle so that it would appear on computer monitor at the same angle. The goal of this project is to detect the angle and correct for it so that the document would appear on computer monitor straight regardless of scanned angle.

KEYWORDS
Fourier, skew, image.

INTRODUCTION
There are number of different methods for image de-skewing available in literature. This method I developed myself and it is based on Fourier transform. The approach is to transform input image from spatial domain to frequency domain and look at direction of frequency distribution. Usually text represents letters arranged in horizontal rows and those rows are stacked one under another. Because of that most of the energy in frequency domain should be along the rows of letters and perpendicular to them. The following experiment confirms this assumption.

IMPLEMENTATION
Fig. 1 shows the input text scanned at some arbitrary angle. In this case the angle is known for test purposes and is equal to 30 degrees. I used the geometric transformation to rotate an input image by 30 degrees. So that if my de-skewing algorithm works correctly I know that the resulting de-skewing angle has to be very close to 30 degrees. Because of rounding errors the resulting angle can’t be exactly 30 degrees but if it is close, for example the error is less than 0.1 degrees, then from human point of view it will be completely de-skewed.

Fig. 1: Input Image Scanned at arbitrary angle

Fig. 2 shows the spectrum of the input image and as it was expected most of the energy is distributed along the axis parallel and perpendicular to text lines and greed.

Fig. 2: Spectrum of the Input Image
Next as shown on Fig. 3 the spectrum of the input image was segmented in four quadrants corresponding to quadrants of Cartesian coordinate system.

Fig. 3: Spectrum of the Input Image Segmented in four Quadrants.

Clearly the brightest points of the spectrum make up the lines that make the same angles with x and y axis as the original document placed on the scanner. The spectrum image was segmented this way because each quadrant processed by itself should result in right de-skewing angle. And to increase accuracy and minimize possibility for error an average of for resulting errors can be found. Also as it shown in Fig. 3 there is little black square in the corner of each quadrant. It was done to mask the pixel in the middle of the spectrum image and some pixels around it because the following algorithm is based on finding the brightest points in each quadrant and the middle point is the brightest since it is an average value but this point doesn’t help in finding the angle so it needs to be ignored.

The de-skewing algorithm works as follows. Independently for each quadrant I find 20 brightest points and their coordinates. Next I fit a straight line through those points and determine the angle between the line and x axis. When all four quadrants are evaluated in the same way and four angles (one for each quadrant) are found the de-skewing angle is determined as an average of four angles found above.

RESULTS AND DISCUSSION
Fig. 4 shows the de-skewed text after applying the rotational transformation to the input image at determined above angle.

Fig. 4: De-skewed Input Image

The MATLAB code shown in Appendix 1 after being run resulted in de-skewing angle of 30.0182 degrees. This is a very good result since error is so small that it is absolutely invisible. The image in Fig. 4 was processed using rotational transformation by the found above angle. And it appears perfectly straight.

Fig. 5 shows the de-skewed image after fitting it into some text boundaries 1000 x 1000 pixels in this example but they can be user selected.

Fig. 5: De-skewed Input Image inside the Text Box
SUMMARY
The described above algorithm definitely does its job. It works with great precision and is easy to implement. The only computationally intensive part is a Fourier transform of the input image. The inverse Fourier transform is not required. Rest of the algorithm is significantly less computationally intensive. Overall this algorithm can successfully be used for text de-skewing.

REFERENCES