Skew angle detection using texture direction analysis

Jaakko Sauvola and Matti Pietikäinen

Computer Laboratory
Department of Electrical Engineering
University of Oulu
FIN-90570 OULU, Finland
e-mail: jjs@ee.oulu.fi

Abstract

In a document analysis system the skew error detection is one of the most crucial parts. This paper describes a method that takes advantage of texture orientation analysis in order to find the skew error from the source document. The source image is first shrunk to melt the characters in together, reduce noise and decrease the amount of computing. After pre-processing a texture direction analysis method is applied. This procedure produces an unambiguous angle in degrees as the skew angle.

1. Introduction

Skew detection and correction is a very important problem when an OCR algorithm tries to detect and interpret characters from the source document. Many different solutions have been developed. The stage in which the skew detection takes place also varies from the first to the very last one before the OCR phase.

The document structure analysis and character recognition are usually done in several phases: scanning and thresholding, image enhancement, skew detection and correction, segmentation, classification and character recognition. Each step must be completed well enough for the performance of the sequence and result to be successful. Steps that follow the skew correction are inefficient if the correction fails.

Skew correction may depend on many different approaches. Methods based on Hough transform and run-length smearing [1], cross-correlation [2] and projection histogram [3] have been proposed, as well as methods that rely on white space versus black areas (text/picture) and their placing. Most of the previous methods are strongly dependent on the amount of textual coverage on the page. Their efficiency usually drops if the text content in page diminishes.

This paper proposes a new approach based on texture orientation analysis, earlier used for other purposes [4] and [5], which does not require a text oriented source in order to define the skew error: a few lines of text or pictures that usually have the same orientation as the text are sufficient. Keeping in mind that although a text document from a texture point of view is quite complex, it is possible to create a texture-like surface so that the image can be dealt with as a texture.

The texture analysis phase produces an unambiguous number as the skew error. In our experiments, this value is quantized at one degree intervals. In OCR, the skew error must typically be less than two degrees in order to gain satisfactory results: the smaller the error the better result.

2. Skew angle detection

The skew angle detection is performed using texture direction analysis as the primary method. First, the image is blurred in order to create a texture-like surface, in which the areas searched for orientation appear as almost homogeneous grey areas. The same effect can be seen when looking at a textual image from a distance: letters seem to merge to look like a united grey area (Fig. 1).

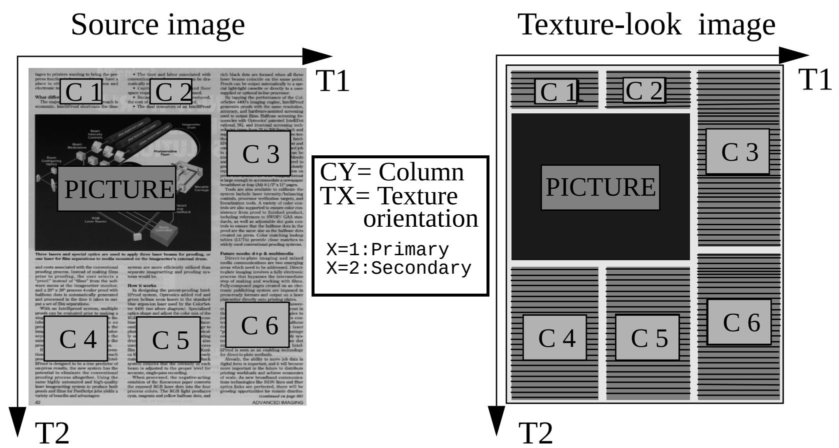


Figure 1. Blurring source image to texture-look grey regions.

This effect can be achieved in many ways. Methods based on grey-scale morphology or connectivity analysis can be used, for example. In our method three properties are combined in one in order to blur the image, filter the noise and decrease the amount of image data size used for the direction analysis. The source image is shrunk using bilinear interpolation techniques by a factor that depends directly on the image size. The bigger the image the bigger the factor. A factor of one fifth was used in all of our experiments.

2.1 Texturizing an image and reducing the amount of data

When a grey-scale of a b/w-picture is shrunk, it is possible to gain the same effect as in a morphological region growing process: the small objects or pixel areas are merged in together, while the small noise-like defects are removed. The problem with morphology is that noise and the other disturbing factors will increase as well, and this is computationally quite expensive. Region growing or connectivity analysis with certain masks create the same melting effect, but the noise remains and the size of the source image data does not change.

Shrinking the source image to a fifth of its original size decreases the size of the input data and the calculation time when the texture direction analysis is performed. When the shrunken image is looked at closer, the text lines seem to be connected to the lines of black pixels. This helps to simplify and order the result of the texture orientation calculation process. The resulting image is a filtered version of the original picture, in which the small discontinuities and the other faults that affect the general direction of the texture are diminished. The shrinking process does remove details from the image, like very thin lines or noise-like defects. Removing this type of objects is not a problem, because the shrinking factor used with this algorithm is small enough not to cause any loss to the texture-look of textual contents in the image.

2.2 Texture direction analysis

The actual texture direction analysis is based on the directional evidence accumulation presented by Chaudhuri [4] and Rao [5]. In this method, the edge image is formed first. After that the local dominant orientations are computed for non-overlapping subwindows and used to increment a histogram of orientations. Finally, the maximum peaks are detected from the histogram, in which the highest peak defines directly the skew of the original image.

The edge image is formed first [4]. Let the gradient vector of magnitude be G and the angle of a gradient be ϕ. The resulting image is called the edge image E. Consider now two masks:

$$h_x(i,j) = \frac{2i}{\sigma^2} \times e^{\left(-\left(i^2+j^2\right)/\sigma^2\right)} \tag{1}$$

$$h_y(i,j) = \frac{2j}{\sigma^2} \times e^{\left(-\left(i^2+j^2\right)/\sigma^2\right)}, \quad -s < i, j < s\,, \tag{2}$$

where the mask size is defined as (2s+1)x(2s+1). The parameter σ defines the smoothing factor for a gaussian filter. The mask size s can be defined using Bergholms [6] solution (3):

$$s \approx \sigma \cdot \sqrt{-2\log\sigma - \log(0.005)}\,. \tag{3}$$

We used a predefined value of 2.0 for σ in our tests. Therefore s gets a value of approximately 2.6, and the mask size is 6x6 pixels. The two masks, (1) and (2), are convolved with the source image I (4)-(5), to get the gradient magnitude and direction (6)-(7):

$$G_x = h_x \bullet I \qquad (4)$$

$$G_y = h_y \bullet I \qquad (5)$$

$$G = \sqrt{G_x^2 + G_y^2} \qquad (6)$$

$$\phi_{ij} = \tan^{-1}\left(\frac{G_y}{G_x}\right). \qquad (7)$$

The gradient vector (G,ϕ) is computed on a mask of size (2s+1)x(2s+1) resulting in the image E.

We performed tests with two different methods of determining the local dominant direction. In Rao's method, the dominant direction θ_D can be calculated as follows (8):

$$\theta_D = \frac{1}{2}\left(\tan^{-1}\left(\frac{\sum_{i=1}^{m}\sum_{j=1}^{m} G^2(i,j)\sin\left(2\phi_{ij}\right)}{\sum_{i=1}^{m}\sum_{j=1}^{m} G^2(i,j)\cos\left(2\phi_{ij}\right)}\right)\right), \qquad (8)$$

where m*m is the size of the window W.

In Chaudhuri's method, on the other hand, the contribution from all pixels in a window can be written in an accumulator as (9):

$$A^W{}_\theta = \sum_{(i,j)\in W} G(i,j)\cos^2\left(\theta - \phi_{ij}\right), \qquad (9)$$

where θ goes from 0 to 179 degrees and
W is m*m pixel subimage of image E.

Now the bin with the maximum value in the accumulator represents the dominant direction in the given window.

In our experiments, both methods were accurate enough for detecting the text orientation from the source image. The calculation time was the only factor that was different. Rao's method proved to be faster, even though the peaks in the histogram were wider horizontally. The tests presented in this paper were performed using Rao's method.

After completing the local orientation determination, the construction of a histogram for obtaining the global skew angle is the next stage. In our experiments, the direction was quantized at one degree intervals, but smaller intervals are also possible if needed for reliable character recognition. An array of 180 bins is formed, and a vote gained from every subimage is added to the corresponding bin, until the whole image has been processed.

Only one vote comes from a single subimage W. The size of a subimage affects the locality of a direction vote.

In our experiments, 10x10 pixel windows were used.‘Empty’ subimages i.e. the windows that include only white background are ignored in the histogram construction. Examination of the overall contents of a window W could also be used to pre-classify regions to text, graphics and backgound, so that only the desired areas are searched for skew. This feature exists in our system, but is beyond the scope of this paper. The votes obtained from local dominant directions are accumulated in a direction histogram $H_{\theta D}$, formula (10):

$$H_{\theta D} \leftarrow H_{\theta D} + 1 . \tag{10}$$

The maximum estimate value of the dominant direction $H_{\theta D}$ directly defines the text orientation i.e. the skew angle of a source image.

If the skew angle of the source image is big or vertical edges dominate over the horizontal ones, the maximum peak of A may be the wrong one. Phase shift between the two biggest peaks is then usually 90 degrees, which indicates that the document was scanned in a landscape position. If the wrong (vertical) peak dominates, the second biggest peak (with 90 degrees shift to maximum) is the right one (Fig. 2). This is why the two biggest peaks must be detected from the dominant direction histogram.

Usually one can make an assumption that the skew error is less than 45 degrees. If this limitation can be made, only the biggest maximum under 45 degrees must be found.

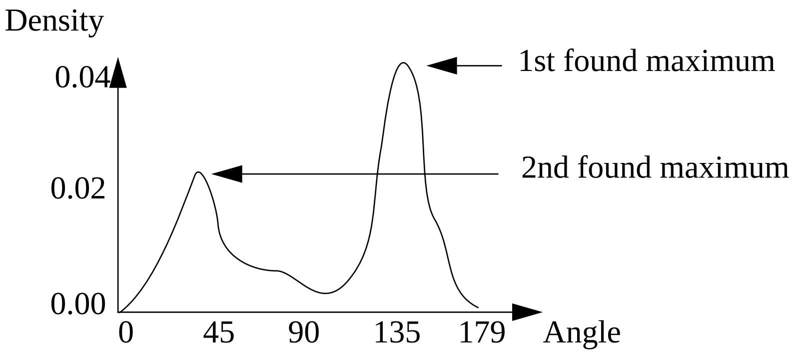
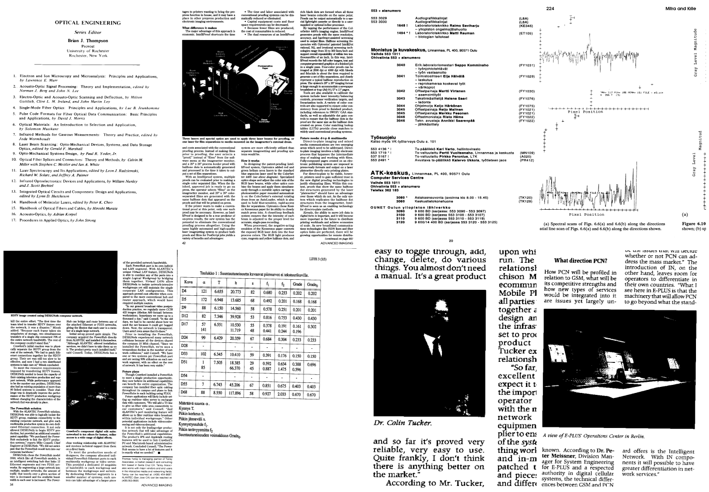
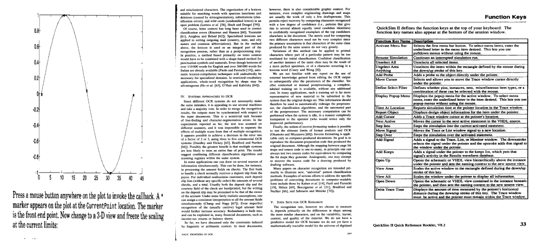


Figure 2. Angle histogram for a sample image.

3. Test material and results

Material used for testing the algorithm was chosen carefully in order to gain enough variety from different types of documents. Every single source document was tested with several different skew angles from 0 to 20 degrees. Testing was performed in a Unix Khoros environment where the actual development work was done. Some of the test images are shown in (Fig. 3).





Images 1-11, from left to right, top to bottom.

Figure 3: Test image set.

When using the texture orientation method for skew detection we obtain quite an accurate and robust estimate for skew angle as shown in Table 1.

The final accuracy of the skew angle estimate is determined by the quantization level, the amount of text and the properties of the general texture in the image.

From Table1, we can see that the half degree skew errors (3.5 degrees and 10.5 degrees) are classified to either the nearest higher or lower quantized even number.

This behaviour can be explained with the different contents of each image, as far as the texture properties are concerned. Otherwise detected values seem to follow the given skew error angle accurately.

Table 1. Test results for images in figure 3.

IMG No \ Skew	3.5 dgr	6 dgr	10.5 dgr	15 dgr	20 dgr
Image 1	3	6	10	15	20
Image 2	3	6	10	15	20
Image 3	3	6	10	15	20
Image 4	3	6	10	15	20
Image 5	3	6	10	15	20
Image 6	3	6	10	15	20
Image 7	4	6	11	15	20
Image 8	3	6	10	15	20
Image 9	3	6	10	15	20
Image 10	4	6	11	16	21
Image 11	3	6	10	15	20

4. Conclusion

In this paper, we have introduced an efficient method for skew detection. The method makes use of texture orientation determination in a new context. The source document image must be pre-processed first. The texture algorithm must be able to see the text in an image as a texture. This is done by shrinking the source image to one fifth of the original size. The shrinking process unifies the text areas to uniform textures and lessens the amount of computing time radically compared with corresponding original images. Since texture analysis is computionally quite demanding, the shrinking process mostly eliminates this defect. The texture orientation analysis produces an unambiguous answer to the skew correction procedure.

The accuracy of the produced angle depends on the quantization level of the histogram in the texture analysis phase and the contents of the original page. The present method can be used for b/w or grey-scale images. The benefits of this method are the reduced amount of text needed in a document to produce an accurate skew estimate and the vast angle range it can be applied to.

References

[1] Hinds S. C., Fisher J. L. and D’Amato D. P. (1990) A Document Skew Detection Method Using Run-Length Smearing and the Hough Transform. In: 10th International Conference on Pattern Recognition, Vol. 1, pp. 464-468.

[2] Yan H. (1993) Skew Correction of Document Images Using Interline Cross-Correlation. CVGIP: Graphical Models and Image Processing, Vol. 55, No. 6, pp. 538-543.

[3] Ciardiello G., Scafuro G., Degrandi M. T., Spada M. R. and Roccotelli M. P. (1988) An Experimental System for Office Document Handling and Text Recognition. In: 8th International Conference on Pattern Recognition, pp. 739-743.

[4] Chaudhuri B.B., Kundu P. & Sarkar N. (1993) Detection and Gradation of Oriented Texture. Pattern Recognition Letters. Vol. 14, pp. 147-153.

[5] Rao A. (1990) A Taxonomy for Texture Description and Identification. Springer, New York.

[6] Bergholm F. (1988) Detection of Edges and Busy Regions Using Canny’s Edge Detection. TRITA-NA-E8472, Royal Institute of Technology, Stockholm, Sweden.