

A SURVEY ON DOCUMENT IMAGE SKEW DETECTION

Mihai Zaharescu^{1*}
Costin-Anton Boiangiu²
Ion Bucur³

ABSTRACT

This paper addresses the research area of Document Image Skew Detection and Correction. The problem is of critical importance in the automated content conversion systems domain, making libraries digitalization projects possible. The paper at hand consists in a comparison between the main types of skew detection algorithms and presents the reader with a study on their advantages and disadvantages, as well as proposed improvements.

Keywords: *Skew detection, deskew, automatic content conversion, Hough transform, projection profile, nearest neighbor, cross correlation, OCR, digitalization process.*

1. INTRODUCTION

1.1 The need

In recent years, the digitalization of libraries has become a priority, as the volume of information worldwide grows in an exponential rate.

Furthermore, companies and organizations are making the transition from paper to electronic documents, a process during which they need to integrate large amounts of paper documents into the new electronic archive, in order to preserve important data.

Thus, there is a growing need for efficient document digitalization techniques, which use complex image preprocessing technologies and Optical Character Recognition systems.

1.2 The digitalization process

The digitalization process consists of multiple stages, among which are: scanning, image preprocessing for noise reduction and color correction, binarization, skew correction, page segmentation, character recognition, language detection and dictionary based correction.

After the library worker scans the document, the scanned image undergoes a preprocessing step, which is necessary in order to forward an optimal input image to the following processing stages.

The majority of OCR readers need a binary image as input, so binarization process usually follows.

^{1*} Corresponding author. Engineer, mihai.zaharescu@cs.pub.ro, "Politehnica" University of Bucharest, 060042 Bucharest, Romania

² Associate Professor PhD Eng., costin.boiangiu@cs.pub.ro, "Politehnica" University of Bucharest, 060042 Bucharest, Romania

³ Associate Professor PhD Eng., ion.bucur@cs.pub.ro, "Politehnica" University of Bucharest, 060042 Bucharest, Romania

Following this process, a page layout algorithm identifies and separates text and picture areas. This helps in building a hierarchy of page elements, which the system can exploit in order to generate automatic summaries, indexes used for fast searching and metadata useful for digital libraries. Various computational geometry methods identify the geometric characteristics of the input text (font size and type, line spacing, etc.) which fine-tune the OCR process in order to retain the original aspect.

Only now, can the OCR receive text information from images cut from the original.

Finally, a text correcting stage compares the extracted data against a dictionary. Of course, the systems that are more complex can use a feedback algorithm which, depending on the number of errors in the result, can repeat the last stages with refined input parameters.

The entire complex processing systems described above use the results from the preprocessing stages. Any error in the first stages will propagate and generate poor results.

One of the preprocessing stages that can influence severely the result is the skew correction.

2. CAUSES FOR SKEW AND RESULTING PROBLEMS

2.1 Causes for document skew appearance

Skew errors are introduced in the first stage of the digitalization process: the scanning. Errors can occur from two sources: either from physical misalignment of the scanned paper on the scanning surface, or by scanning a document that already has skew.

2.2 Problems introduced by skewed documents

Skewed images not only present problems for the digitization process, but also for the human users: many digital libraries present pages in PDF format that contain both the reconstructed text and the original image as background. Without deskewing, this would present misaligned content. For machine reading, skewed images present a number of problems that impact the majority of algorithms related to image layout analysis, text characteristics identification and text recognition. Skew images make the algorithms more error prone and ultimately lead to poor results in the OCR process. This is because many algorithms are based on the detection of straight runs of space or groups of vertically or horizontally aligned characters or pictures.

The solution requires an automatic skew detection and correction system that can work with minimal human intervention. At most, it should require initial parameters regarding the type of documents, language and other characteristics that might influence the skew detection algorithm.



Fig. 1. Example of a skewed Image

2.3 Impact on image processing algorithms

One of the first processing stages in the digitalization process is Page Segmentation.

The run-length smearing algorithm (RLSA) [13] works on binary images, by connecting black regions closer than a given threshold, on both X and Y axes. If the input image is skewed, the algorithm will most likely end up connecting lines or joining different columns of text.

Breuel [6] proposed a layout analyzing system for page segmentation based on Whitespace Cover that starts with finding maximal whitespace rectangles. These are later used as column separators. In a skewed image, however, even small amounts of rotation can prevent thin columns separators from appearing as vertical white lines, making the approach useless.

These are not the only algorithms that require skew free image as input in order to obtain accurate results. The majority of algorithms assume that images have text grouped in horizontal lines.

3. CLASSES OF ALGORITHMS AND GENERAL ASSUMPTIONS

During the past 20 plus years, a few major classes of skew detection algorithms have been developed:

- Projection profile;
- Hough transforms;
- Clustering of nearest neighbors (connected component clustering);
- Correlation between lines (cross correlation).

Other methods like white spaces parallelograms covers, wavelet decomposition, moments, morphology, subspace based line detection and Fourier analysis are worth mentioning.

3.1 General assumptions

All these methods work on black and white (monochromatic) images and are generally optimized for documents in which text is predominant and arranged in parallel straight lines, in one or more columns and containing intercalated images. In addition, they assure best performance when the text contains only Latin characters. There are types of scripts that present unique features that prevent most of these algorithms from performing well. For them, dedicated algorithms had to be developed. Examples are Indian scripts (Devnagari, Bangla) [9] and Arabian scripts [26].

During a normal scan operation, one can expect skew errors of up to 15° . Although there are algorithms which can deal with angles in the range $[-90^\circ, 90^\circ]$ [31], many algorithms take into consideration the assumption that the skewing angle is rather small.

3.2 Summary for the 4 main approaches

The Hough transform based algorithms and the Projection profile based methods are different mathematical representations of the same idea. Both presume that the lines of a text are horizontal and parallel, and work by detecting the angle of the dominant parallel lines of pixels.

The algorithms based on the Hough transform start by projecting the image from the XOY space into the Hough space, and then computing which lines are dominant (which lines contain more black pixels). Although they can achieve very good precision, the disadvantage of the method lies in the computation time that increases with resolution.

The projection profile techniques also rely on the detection of parallel straight lines. The program projects the image's pixels on the OX and OY axes, generating histograms. By rotating the source image, it generates different projections. The angle at which the most peaks appear is the angle at which the most lines are perpendicular to the projection axes. As with the Hough transform based methods, this method is computational intensive.

Nearest neighbor clustering based approaches, exploit the fact that the majority of documents contain text, which contains aligned and close-by characters. Using a bottom-up process, they start from connected components, group them, and use their spatial relationships to compute the document skew.

Cross correlation approaches, assume that textual zones inside a skew free image will have a homogeneous structure. Using the deviation from this benchmark, they can estimate the skew angle of the input image.

4. METHODS BASED ON THE HOUGH TRANSFORM

This class of methods uses the Hough transform to find the predominant straight lines in a document, which give the image's skew angle.

Generally, an algorithm based on Hough transform goes through the following steps:

1. Select a set of points to use as input points for the transform
2. Estimate input parameters, regarding the resolution and the search size
3. Initialize the accumulation buffers used by the algorithm

4. Apply the Hough transform on the selected input points and fill the accumulation buffers
5. Select the dominant values from the buffers
6. Compute the skew angle

4.1 The Hough transform

Although the Hough transform can find different types of shapes, the focus is on line detection.

The most common approaches are based either on the Cartesian or Polar representations.

The Cartesian system uses the line's slope and the intersection with an axis in Euclidian space:

$$y_0 = y - x \cdot m \quad (1)$$

The Polar representation uses the line's angle and distance from origin.

$$r = x \cdot \cos \theta + y \cdot \sin \theta \quad (2)$$

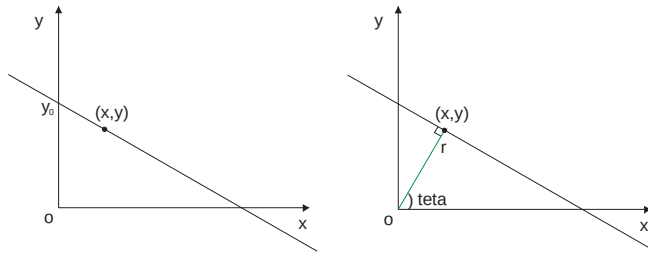


Fig. 2. Cartesian and Polar coordinate systems

This equation describes a sin curve:

$$\sin(\alpha + \theta) \sqrt{x^2 + y^2} \quad (3)$$

The focus will be set now on the line detection procedure using the polar representation. Points that lie on the same line will generate sinusoidal curves that intersect at a given location. This location defined by the coordinate r and θ will uniquely define the line on which all those points rest, and thus, retrieving the angle for that set of points.

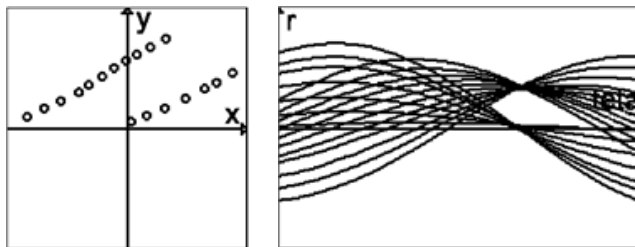


Fig. 3. Two sets of collinear points inside the Cartesian and Polar systems.

Accumulation buffers are used in order to determine the intersection of the sines, in the Hough plane. These buffers are a discrete matrix capable of storing a limited interval and resolution. For each point in the image, the algorithm “plots” its sine in the buffer, incrementing the corresponding cells, thus generating a peak where several sines meet.

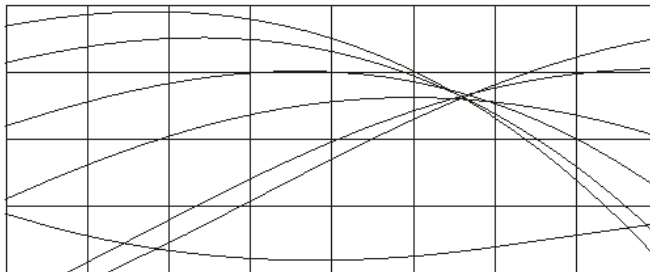


Fig. 4. Zoom in on the accumulation buffers. At cell (2, 5) several sine curves intersect, thus creating a peak in the accumulation buffers array

The highest values indicate the most predominant original lines, which results in the overall skew angle estimation.

4.2 Complexity

1. For each black point selected from the image, generate a number of lines intersecting that point, at different angles.
2. Plot the generated lines in the Hough space
3. Filter the accumulation buffer in order to obtain the highest values
4. Select the resulting lines and compute the skew angle

The complexity is linear with respect to the number of input points and to the desired angular resolution.

4.3 Sources of errors and drawbacks

As the Hough transform relies on the detection of aligned pixels, serious errors occur if the input page contains pictures. Thus, it is important that the algorithm receives input points selected only from text areas.

However, the main drawback of the Hough transform approach is its complexity. Without a proper optimization stage, the run time of the algorithm is prohibitive for large images. On the other hand, optimizations in the number of input points or resolution affect the precision so much, that the results can become inaccurate.

4.4 Improvements

An idea for reducing the input points is to use a boundary growing technique for detecting characters and forming lines of text. The centroids of the components are the input points which generate the best results [28].

A continuation to this idea [4] is to apply the standard Hough based algorithm two times after selecting the centroids: firstly, using a low resolution, in order to compute a low accuracy angle but in a very short time and finally, with a reduced Hough space, with a fine resolution to compute the precise angle.

The drawback of these methods is that the initial reduction from all points to centroids can have a negative impact in images containing primarily straight lines, such as lines from tables or from non-Latin scripts.

Software optimizations (pre-computed operations, smart use of pointers for 2D arrays and integer calculations) can reduce run-time without losing accuracy [7]. This optimization is up to ten times faster than other Hough transform based methods, giving errors below 0.5° even for large skew angles.

A different approach is to create a grayscale burst image in which each pixel value represents the vertical run-length of a column in the original image [15]. These important grayscale pixels are the bottommost pixels of each run-length line, thus emphasizing the bottom contour of text lines. Then the algorithm increments the values in the accumulation buffers not by a constant value, but with a value given by the grayscale value of the pixels. Considering, for example, only run-lengths between 1 and 25 pixels, for 75 dpi images it reduces errors introduced by non-text elements. This approach not only saves time by reducing the amount of input data, but also improves accuracy by minimizing the effects of noise and non-textual input from the image. One extra advantage of this method is that using the run-lengths one can also indicate the orientation of the document (portrait or landscape).

Le, Thoma and Wechsler [19] implemented a similar method, selecting only the last black run-lengths. Their algorithm also incorporates a page orientation detection system (portrait or landscape) based on projection profiling. The authors report errors below 0.5° for skew detection and 0.1% for page orientation.

5. PROJECTION PROFILING

Projection profiling methods compute a histogram on the vertical or horizontal axis, containing the number of black pixels contained in the respective line.

In its simplest form, the algorithm rotates the input image at different angles and projects the pixels on the vertical axes. The angle at which the projection profile gives the most peaks and vales in the histogram is the opposite of the skew. This is because, as seen in the figures following, in a skew free image, lines of text produce peaks in the histogram of the projection profile, while white spaces between lines produce vales.

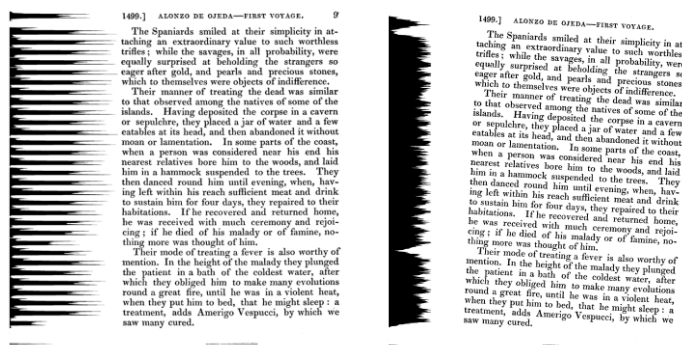


Fig. 5. Projection profiling on the OX axes for a skew free image and for an image with a 3° skew angle

The steps involved are:

1. Select resolution and skew angle interval
2. For each skew angle in the interval, at the given resolution, compute the projection profile and retain information memorize the histogram
3. Compare the histograms for different angles and select the one that resembles best a skew free image

Authors proposed different histogram comparing functions, called premium functions. The functions give greater values for greater number of vales and peaks with greater differences. They take an angle as input, and the angle that maximizes the function is the desired skew angle.

5.1 Complexity

The time required to run a projection profile based program is linear in respect to the skew angle search size, the skew angle resolution, the page size and the resolution at which the algorithm computes the projections.

5.2 Sources of errors and drawbacks

The drawback of this method is that it is very CPU intensive, as it needs to compute and compare histograms of projection profiles for a number of skew angles, along with rotating the image for each measurement.

Non-text areas generate errors because they introduce distortions in the histogram. This technique can achieve good precision for a text-only document, but if a large amount of non-text information is present, the method yields poor results.

5.3 Improvements

In [3] Baird proposes two improvements:

The first is to search for the skew angle in two steps. Firstly, the algorithm runs using a large angle interval with low resolution, in order to find the approximate value of the skew angle, without trying a large number of angles. After finding the approximate angle, the algorithm runs a second time, but now using a small search interval, at very small steps, in order to achieve high precision.

The second improvement considers a similar sampling method used for reducing the input data in the Hough transform: only the middle point of the bounding boxes of the components represent the input in the projection algorithm.

A small addition which may be also useful to determine the orientation of the image is to select only the closest points to the four borders of the bounding boxes of the groups, for analyzing the image at the four possible orientations (0°, 90°, 180°, 270°). This approach provides very fast and reasonably accurate results.

A modified projection profiling technique divides the image into straps, and applies a binary projection profiling technique afterwards, in which, for each line, the projection's value is one if there is at least a black pixel on that line, or zero if there is none [35].

Because wavelets can also show the orientation, filter the wanted frequencies and retain the signal's position, they can aid projection profiling [29]. The input of the projection profile is the higher frequency sub-band with horizontal orientation (because it follows the text better). The authors claim that this method delivers improvements in speed and accuracy compared to applying projection profiling technique on the input image itself.

Special scripts need special projection profile methods, because they may not contain horizontal lines of text. One example is the Gurmukhi Script [20], which has most characters forming horizontal lines at the top, called headlines. After deskewing the image, expected projection profiling for possible orientations will judge the orientation of the document.

6. CROSS CORRELATION

Cross-correlation approaches use projection profiling on thin vertical segments cut out from the image, in order to compute the skew angle.

The algorithm starts by selecting vertical segments from the input image, which it uses for profiling. Then it computes the projection profiles for each vertical segment. By comparing the projection profiles of adjacent segments, it can estimate the skew angle, which gives the best alignment of the two segments.

Yan [34] defines a cross correlation function $R(s)$ for pairs of vertical segments located at the same distance, d .

$$R(s) = \sum_{x,y} Seg(x + d, y + s) \cdot Seg(x, y) \quad (4)$$

This multiplies adjacent vertical segments, shifted by s , and sums the results, in order to generate a global estimate. s is a characteristic of the angle. By selecting s from the values that gave the greatest correlation the algorithm estimates the skew angle.

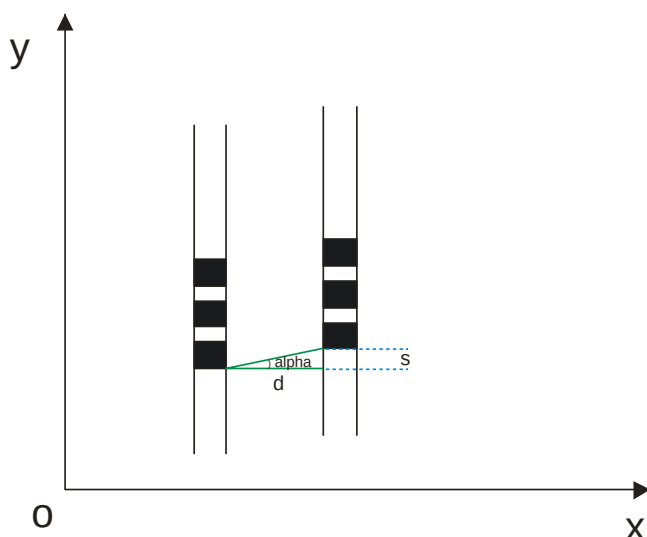


Fig. 6. Cross correlation of vertical segments containing text lines

An addition is to use a Monte Carlo sampling technique for determining the regions where to apply cross correlation [1].

6.1 Complexity

The complexity is similar with that of projection profiling methods. The main advantage of this class of algorithms is that, given a good initial estimate of the skew angle and a good selection for the width of the columns, they can obtain a drastic reduction of processed data, without compromising accuracy. That means both less computing time and good precision.

6.2 Sources of errors and drawbacks

Methods based on correlation are well suited only for documents that present a homogeneous horizontal structure.

The major drawback of this method is that it requires text-only input images and the vertical segments used for cross correlation need to have the text sampled from the same columns of text. Otherwise, if the columns cut through different lines of text (for example, lines from columns with different font size) errors rates rise sharply. If one segment has text and another segment contains portions of images, the correlation between them will yield wrong results, and deteriorate the accuracy of the final skew angle. One way to resolve this issue is to run image segmentation algorithms first, which select text regions from the image. However, because most of these algorithms require skew free input samples, this method becomes restricted to text predominant images.

7. NEAREST-NEIGHBOR CLUSTERING

Nearest-neighbor clustering methods use the coherency found in the alignment of blocks of characters, in order to determine the skew angle.

They rely on discovering connected components, which represent characters, and compute

the skew angle by finding the dominant slope between them, considering their centroids or their bottom pixels.



Fig. 7. Characters represented by their bounding boxes, clustered together. The line at the bottom is the identified skew line

A simple implementation of this idea is to apply a series of filters and a clustering algorithm in order to identify lines. The bottom pixels of each connected component in the cluster generate a local skew angle. The skew angle is the median value of all the found angles. The method obtains good results for skew angles of at most 20 degrees.

Other similar idea is to group the components by their neighbor [14]. The angles between each connected component contribute to a histogram. The peak in the histogram is the skew angle.

Another idea, after the connected components identification step, the centroids and corners generate K-nearest-neighbor chains. The longest chains of nearest neighbors give the skew angle of the document [21]. The algorithm works both on horizontal-flowing text and on vertical-flowing text documents (Chinese and Japanese documents for example).

An approach based on focused nearest-neighbor clustering is to use the centroids of the connected components which represent the feature points of the input image [17]. A feature point and its K nearest neighbors (K is an input parameter, with the minimal value of 3) generate a local skew angle. This initial skew angle gives a better approximation for a local cluster, which in turn generates a second skew angle. This second skew angle contributes to a histogram, from which the algorithm chooses the global skew angle.

A solution that can be included in this category of algorithms, with a novel approach, is to filter noise from the input image. Then to use a run-length algorithm, which incorporates a way to determine if text flow is horizontal or vertical, to generate connected components. A proposed borderline extraction algorithm selects the borderlines (the clusters' contours) in the image. The best borders pass through another filter that optimizes them and separates discontinuous borders. As each borderline contains pixels that are distant from the computed straight line, a weight function gives more importance to pixels that are closer to the computed line. Finally, each border generates a local skew angle. The weighted mean of local angles gives the global angle, where the weight is directly proportional to the number of pixels corresponding to each border.

7.1 Complexity

Because of the different implementations for this class of algorithms, a general complexity measurement for this class cannot be presented.

Generally, these algorithms consist of three steps: connected components identification, clustering and the skew angle computation.

The first and third steps are generally linear. Thus, the main factor that influences the speed is the clustering method used.

7.2 Sources of errors and drawbacks

One of the main advantages of nearest-neighbor based methods is the fact that they are generally less error prone to errors introduced by non-text areas.

However, some scenarios can degrade the performance of these algorithms. The main problems are created by clusters containing ascenders and descenders (characters like j, p, q, y), which generate lines that are not straight even in the original skew free image. The second problem is that most clustering algorithms will not produce perfect lines, and thus, offer an erroneous base for computing the skew angle.

The quality of the image has a profound influence on the results. Connected or broken characters, noise and non-textual items, have a severe impact on accuracy. These algorithms are also dependent on assumptions upon text characteristics, needed to identify lines of text.

8. OTHER METHODS

The search for better accuracy and especially for faster computing time generated a few other approaches, from which some deserve to be mentioned.

8.1 Radon Transform

The Radon transform is similar to a projection of the image along specified directions. The bottom points of the connected components, identified as characters, are the input of the Radon transform. In a similar manner to the projection profiling technique, the skew angle is the angle at which the transform array has the largest amplitude and frequency [24].

8.2 SLIDE – Subspace-based Line Detection

This is a novel approach that not only promises faster and better results, but can also detect multiple skew angles in a single document [2]. The proposed approach transforms straight lines from the input image, into wave fronts. A signal processing method, which uses the coherency between the pixels positions in parallel lines, can enhance a subspace from the domain. For estimating the multiline angle, the authors apply a subspace fitting and array processing technique - TLS-ESPRIT. The authors claim that their method has significant advantages in terms of space complexity and computational time, compared to the methods based on the Hough transform, while maintaining a high level of accuracy. They showed that the algorithm could obtain good results on pages containing areas with different skew angles.

8.3 Skew Angle Detection of Digitized Indian Script Documents

Chaudhuri and Pal propose in [9] an algorithm that uses a particularity of the Indian scripts – horizontal lines at the top – in order to develop a skew estimation method. It detects straight lines that could be lines belonging to the same word. Then, it uses their angle in order to estimate the skew angle. The authors report a standard deviation from the skew angle of below 0.35° .

8.4 PCP & e-PCP

Like many other algorithms, PCP starts with the classical assumption that in a document, text and image objects form lines and columns, separated by white rectangles, and thus forming rectangular shapes. The algorithm presumes that in a skew free image, rectangles are enough to cover the objects; while in a rotated image, objects need parallelograms. The angle at which the parallelograms offer the best covering is the skew angle.

The algorithm works by dividing the image into columns, and then drawing inclined horizontal lines, at a selected angle. The selected angle at which the number of white spaces parallelograms created by the intersection of columns and lines is maximal is the skew angle.

e-PCP is an improvement to the original PCP algorithm that can ensure good results both on horizontal-flowing text documents and vertical-flowing text documents by using information regarding text-flow in the deskew phase. This gives an improvement from 0.5744° to 0.3580° error rate [24].

8.5 Wavelet decomposition

Several researchers tried to adapt wavelet decomposition to image skew detection, in combination with one of the classical approaches.

The anisotropic diffusion, in the direction of the text lines, blurs unimportant letter characteristics, but keeps the text line characteristics. Next, the wavelet transform decimates the frequencies. A linear regression formula, using the bounding boxes of the binarized high frequencies, taken from the transform, gives the skew angle [27].

A particular case is finding the skew angle in a scanned track image [8]. The wavelet transformation extracts and emphasizes the horizontal features of the tracks in the image. This is the input data for the least squares line fitting.

8.6 Moments

The basic idea is that entities or groups of entities in an image act as solid objects, whose orientations, relative to the OX axis, can give the skew angle.

The proposed method first uses smearing to obtain lines of text. In order to reduce computational complexity, the Freeman's Chain Code codes the blurred objects, and the moment's analysis uses only the contour. The width of the object influences positively the importance in calculating the skew angle. The authors claim to obtain error rates in the 0.01 degrees range [18].

Another way is to use boundary-tracing algorithm for determining sets of bottom pixels

from text lines. Then, an iterative method computes the skew angle based on moments, and eliminates input points that are too far off the skew angle. When modifications stop occurring, the iterations end [36].

8.7 Morphology

Mathematical morphological operations can enhance certain image attributes and form structures that contain valuable skew information (for example well-formed lines of text).

Recursive morphological closing and opening operations generate text lines. The algorithm, however, produces low-resolution results, with errors in the 0.5° range [32].

A more accurate method uses a close operation with a line structuring element which produces solid black bands. In order to remove errors caused by ascenders and descenders, an open operation filters the bands, using a small square structuring element. The base lines of these bands are the candidate lines for skew estimation. The size and the strength these lines have influence their importance. The method works well on documents that present at most 15° of skew [11].

Another addition uses some input parameters, deduced using histograms cumulating the dimension of connected components; this helps the noise removal filters and the elimination of large entities. A fast but low-resolution estimation algorithm finds the skew angle with an accuracy of $\pm 10^\circ$, in order to have a good estimate for the structuring elements used in the morphology process. Close and open operations applied to the image form lines. These lines give the skew angle of the document. The authors claim to achieve a high level of accuracy and it works on both Latin character based documents and on other sets of character based documents, at different resolutions, and in the $\pm 90^\circ$ range [22].

Dilation with horizontal line structuring elements can also be used as the morphological operation. A region labeling technique then creates regions used in determining the skew angle [12].

8.8 Convex hull based

The convex hull of a connected component is the smallest convex polygon that completely contains the component. Convex hulls present similar areas for characters of similar size, and retain the area of an object at various skew angles. The comparison of nearby connected components, in terms of area, identifies similarities (for example, characters belonging to the same paragraph). The connected components form clusters, using the distances from their convex hull centers. Convex hull edges that are long and have perpendicular or parallel counterparts in their group are the important elements in deducing the local skew angle. A weighted histogram gives the global skew angle. The authors claim good precision and very high speed [5].

8.9 Using Contour Analysis

(Chen et al., 2013) propose a method based upon the analysis of the surrounding contours of the text and graphical elements on the page. In order to obtain the contour, they used the dilatation morphological operation and detected the edges formed by the blobs. 859

images of complex technical papers were considered for testing. Their resolution was not mentioned, but it is stated that between 14 and 24 dilatations are required in order to obtain a minimal average error.

Because the graphics generate complex random shapes, they use only the outer box of the contours, relying upon what seems to be an even-odd method for determining what is inside and what is outside.

An extra precaution is taken to eliminate graphic contours: to ignore the first 10 (from experimentation) contours generating the largest axes-oriented bounding-boxes. This seems more like a heuristic and could yield random results for images containing less than 10 elements.

However, the computation of the general skew angle itself is rather interesting: for each contour, a probability of occurrence of a tangent angle is calculated, considering the tangents in every pixel of the chain.

From the large experimental set, the authors concluded that a mean of 0.1° error is achieved for skew angles 0° - 15° and 75° - 90° and 10° errors for other skew angles.

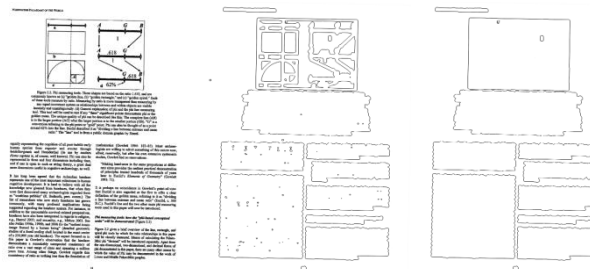


Fig. 8. Processing phases for contour detection

8.10 Fourier

The main idea is that text lines in a document appear usually at equal distances, generating a repeating pattern highly visible on the frequency spectrum, as a single line. From the properties of the Fourier transform, a rotation in the time domain generates the same rotation in the frequency domain. It is much simpler to detect the angle of a single line that traverses the center of the Fourier domain than it is to find the angle of all the lines in the document. Moreover, the more lines in the document have the same orientation, the more visible the single line in the Fourier domain will be, thus images with random shapes will not affect the detection quality, as they only generate noise.

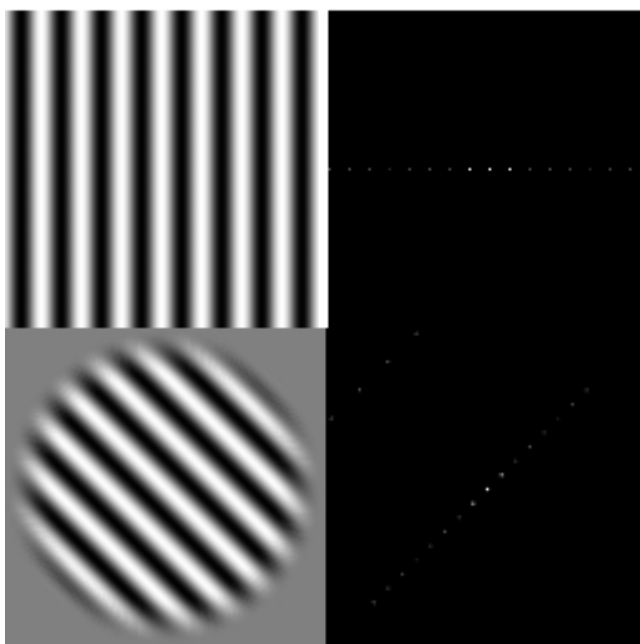


Fig. 9. Vertical and rotated image, filtered amplitude Fourier domain. *Image constructed using information from [33]*

Paper [23] presents a method for estimating the skew angle of an image document by examining the Fourier spectra of blocks of the document image, and looking for peak pairs corresponding to the skew angle. The authors claim an error within 0.5° . The main advantage of this method is that it should be more robust over a variety of document layouts, line spacing variants, font, graphics/images and different language or script documents.

An important aspect of this approach is that it works well even in the presence of graphics.

9. CONCLUSIONS

In this paper a series of skew detection algorithms were presented, all of which have strengths and weaknesses and, as a result, may perform well under certain circumstances. None of them provides the perfect solution, but if used on appropriate documents, can yield very good results. The current paper is an extension to the one presented in [25].

The goal in the field of skew correction, at the time being, is to optimize and improve current solutions.

As later work, the focus will be to run a series of tests (on large batches of image documents), in order to establish the various algorithms performances on different types of input, and to deduce a correlation that could be helpful in creating a general deskew system that adapts itself to the type of input it receives. Such a system will be able to choose automatically a method and a set of optimal parameters for a given batch of image

documents, thus providing the lowest time and best accuracy for the type of input it receives.

ACKNOWLEDGEMENTS

The authors want to show their appreciation to Daniel Rosner and Andrei Tigora for their great ideas, support and assistance with this paper.

10. REFERENCES

- [1] Avanindra, S.C. (1997). Robust Detection of Skew in Document Images. IEEE Transactions on image processing, Volume 6, Issue 2, February 1997.
 - [2] Anghajan, H.K., Khalaj, B.H. and Kailath, T. (1994) Estimation of Skew Angle in Text-Image Analysis by SLIDE: Subspace-Based Line Detection. Machine Vision and Applications, Volume 7, Issue 4, Pages 267-276.
 - [3] Baird, H.S. (1987). The Skew Angle of Printed Documents. In Proc. of the Conference Society of Photographic Scientists and Engineers, Volume 40, Rochester, NY, May, 20-21 1987, Pages 21-24.
 - [4] Bin Yu and Anil K. Jain (1996). A Robust And Fast Skew Detection Algorithm For Generic Documents, Pattern Recognition, Volume 29, Issue 10, October 1996, Pages 1599-1629.
 - [5] Bo Yuan and Chew Lim Tan. (2007). Convex Hull Based Skew Estimation. Pattern Recognition, Volume 40, Issue 2, February 2007, Pages 456-475.
 - [6] Breuel, T.M. (2002) Two Geometric Algorithms for Layout Analysis. Document Analysis Systems, Princeton, NJ.
 - [7] Chandan Singh, Nitin Bhatia and Amandeep Kaur. (2008). Hough Transform Based Fast Skew Detection and Accurate Skew Correction Methods. Pattern Recognition, Volume 41, Issue 12, December 2008, Pages 3528-3546.
 - [8] Changyou, L. and Quanfa, Y. (2009). Skew Detection Of Track Images Based On Wavelet Transform and Linear Least Square Fitting. Information and Automation, 2009. ICIA '09. International Conference on, 22-24 June 2009.
 - [9] Chaudhuri, B.B. and Pal, U. (1997). Skew Angle Detection of Digitized Indian Script Documents. IEEE Transactions on pattern analysis and machine intelligence, Volume 19, Issue 2, February 1997.
 - [10] Chen, Y.S. and Li, P.H. (2013). Skew Detection Using Contour Analysis for a Scanned Journal Page. MVA2013 IAPR International Conference on Machine Vision Applications, May 20-23, 2013, Kyoto, Japan.
 - [11] Das A.K. and Chanda B. (2001). A Fast Algorithm for Skew Detection of Document Images Using Morphology, International Journal on Document Analysis and Recognition.
 - [12] Dhandra, B. V., Malemath, V. S., Mallikarjun, H. and Ravindra, H. (2006). Skew Detection in Binary Image Documents Based On Image Dilation And Region Labeling Approach, The 18th International Conference on Pattern Recognition.
 - [13] Leng, G.W., Mital, D.P., Yong, T.S. and Kang, T.K. (1994). A Differential-Processing Extraction Approach to Text and Image Segmentation. Engineering Application of Artificial Intelligence. Volume 7, Issue 6, Pages 639-651.
- Hashizume, A., Yeh, P.S. and Rosenfeld A. (1986). A Method of Detecting the Orientation of Aligned Components. Pattern Recognition Letters. Volume 4, Issue 2, Pages 125-132.