A Fast Orientation and Skew Detection Algorithm for Monochromatic Document Images

Bruno Tenório Ávila
Universidade Federal de Pernambuco
Recife – PE, Brazil
bta@cin.ufpe.br

Rafael Dueire Lins
Universidade Federal de Pernambuco
Recife – PE, Brazil
rdl@ufpe.br

ABSTRACT

Very often in the digitization process, documents are either not placed with the correct orientation or are rotated of small angles in relation to the original image axis. These factors make more difficult the visualization of images by human users, increase the complexity of any sort of automatic image recognition, degrade the performance of OCR tools, increase the space needed for image storage, etc. This paper presents a fast algorithm for orientation and skew detection for complex monochromatic document images, which is capable of detecting any document rotation at a high precision.

Categories & Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: I.4.3 [Enhancement]: *Filtering*.

General Terms

Algorithms, Design, Experimentation.

Keywords

Orientation and skew detection, monochromatic document image.

1. INTRODUCTION

Digitalisation of large amounts of paper documents inherited from a recent past is the way organisations use to move into the electronic document era. This allows bridging over the gap of past and present technologies, organizing, indexing, storing, retrieving directly or making accessible through networks, and keeping the contents of documents for future generations.

Nowadays, scanners tend to be of widespread use for the digitalization of documents. Very often documents are not always correctly placed on the flat-bet scanner either manually by operators or by the automatic feeding device. This problem yields either incorrectly oriented or skewed images. For humans, either badly oriented or skewed images are difficult for visualisation and reading. In machine processing, arises a number of problems that range from needing extra space for storage to making more error prone the recognition and transcription of the image by automatic OCR tools. Thus, orientation and skew correction are present in any environment for document processing. Although being two decades old already, faster and more accurate solutions to these problems remain a matter of interest, today.

Table 5 at the end of this paper shows nine different classes of skew estimation algorithms, according to a comprehensive review by Cattoni and his colleagues [9] updated by ourselves. All the methods work on monochromatic input images. Furthermore, the major part of the skew detection schemes assumes to deal with documents with a clearly dominant skew angle, and only a few methods can deal with documents containing multiple skews (the very recent paper [30] addresses this problem in the context of correcting the image produced by an uneven feeding device to a scanner, for instance).

The projection profile approach [5][6][12][19][26] is considered the fastest method for skew detection. It is simple to implement and works in complex documents. Skew detection algorithms based on projection profile have as underlying assumption that documents have text arranged along parallel straight lines and that text represents most of the document image; the performance of most projection profile algorithms often decay in the presence of other components like graphics or pictures. Hence, the variation of the projection profile when the document has no skew is maximized. However, there is a tradeoff between the range angle detection and its precision. Whenever the range is limited to ±45º, the precision is greater than 0.5º. On the other hand, whenever the range is limited to ±15º the precision is 0.05º.

The Hough transform is a well-known method to detect lines [18][27]. For each black pixels of the image, it transforms its coordinates (x, y) to the polar coordinates (r, θ) and increments its value in a bi-dimensional array. The peak value of the array corresponds to the line that has more collinear black pixels. In the skew detection problem, it is widely used to detect the angle of the baseline of a text-line or a line of a table [17][22][25][31][33]. It can be used in complex documents and it is a recommended method for skew detection of forms document images. However, those algorithms claim for a large amount of memory. The range angle detection and precision increases the use of memory. Besides that, it is a slow method since it has to apply the transformation to all black pixels and to all range angle detection. There are several variations of this approach that try to minimize the number of transformations [3][20] performed.

The nearest neighbor clustering approach [16][23][25][29] is a bottom-up process. All methods in this group start by component labeling to group together the connected black pixels in blocks. After that step, they group the blocks with similar characteristics

DocEng'05, November 2–4, 2005, Bristol, United Kingdom.

in larger components, e.g. text-lines. Finally, they try to estimate the skew based on the formed text-lines. One should remark that is intrinsic to this approach the ability to detect the landscape/portrait orientation, since text-lines can be grouped in any direction. Skew detection based on nearest neighbor clustering methods is slow due to the component labeling phase that is performed bottom-up and has quadratic time complexity $O(n^2)$ with the number of blocks formed.

Orientation detection methods are less frequent and can be used to detect landscape/portrait and up-down orientations. They are also summarized in appendix A (Table 6). The landscape/portrait orientation detection algorithm [2] uses the same idea as the projection profile approach. The largest variation determines the document orientation. This method uses global variation of all the text-lines of the document. Hence, there are components in the document images that can alter the results, e.g. large black blobs. Thus, a local method [20] was developed to solve the problem. The up-down detection algorithms [7][8] use the fact that ascending characters are more frequent in a text than descending ones. Thus, if the number of strokes below a case line is greater than over it there is a high probability that the document is upside-down. Thus, these algorithms are limited to documents with texts written with the Latin alphabet with salient strokes.

This paper presents a fast algorithm to detect both orientation and skew angle for monochromatic document images. It uses the nearest neighbor clustering approach. It works with complex documents and has a range angle detection of 0º to 360º with precision of 0.1º. It was tested on 270 document images rotated of several angles reaching over 22,000 images. The proposed algorithm was compared with Baird's method [6] to prove its effectiveness and efficiency.

To the best of authors' knowledge, this is the first algorithm capable of detecting the rotation of a document in any angle. Hence, this is the main contribution of this paper is to present an algorithm capable of detecting any rotation of a complex document with an accurate precision in a fast way.

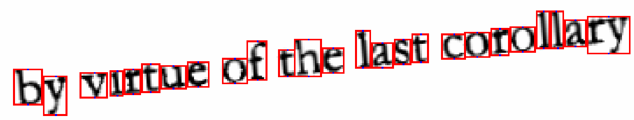
2. THE ALGORITHM

This section explains the basic idea, the main steps and the features of the algorithm.

2.1 The Basic Idea

The algorithm starts by boxing each block of black pixels by using component labeling (Figure 1a). Applying the least square method to the middle top point and middle bottom point separately of all blocks of the text-line, it forms two lines located at strategic points (Figure 1b). The bottom line is located at the baseline of the text-line and crosses the descending characters. The top line is located above the text-line and crosses the ascending characters.

This happens because there are more non-salient characters (e.g. a, c, e, o and u) in the Latin alphabet than ascending and descending ones. Besides that, non-salient characters are more frequent in texts than the others. Thus, the bottom line calculated by the least square method tends to be very close to the bottom of the non-salient characters, which is the baseline of the text-line. The same argument applies to the top line.



(a)

(b)

Figure 1. (a) Blocks of a text-line; (b) the two lines formed after the calculation of the least square method.

This observation lead us to conclude that it is viable to detect the skew angle and the landscape/portrait orientation by accumulating in a histogram the angles of each line formed and, also, to detect the up-down orientation by counting the numbers of ascending characters that crosses the top line and the numbers of descending characters that crosses with the bottom line. The peak at the histogram determines the document skew and landscape/portrait orientation. If the number of descending stokes is greater than the ascending one then the document is upside-down.

2.2 The Main Steps of the Algorithm

The main steps of the algorithm whose basic idea was presented above are:

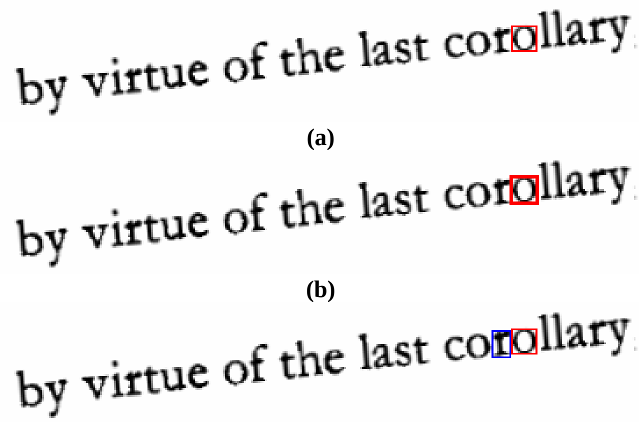
1. Component Labeling;
2. For each block B do:
    - 2.1. Locate the nearest neighbor block N of block B;
    - 2.2. Group a text-line starting from blocks B and N;
    - 2.3. Detect skew angle and landscape/portrait orientation of the text-line;
    - 2.4. Detect up-down orientation of the text-line;
3. Detect total document rotation;

The initial step of a bottom-up process takes as input a document image, applies the 8x8 component labeling algorithm [11][13][14] yielding as output a list of blocks and a map containing the unique identification of each block.

It may be necessary to remove noise from some documents as a pre-processing step to improve the performance and to provide a more accurate result. In practice, this could be done after the component labeling process and ignoring the blocks classified as noise by analyzing their dimensions.

The second step groups all neighboring blocks with the same properties. This represents the final step of the bottom-up process: the blocks are merged and form the text-lines. In order to accomplish this, the second step is divided in four phases. The first and the second phases (2.1 and 2.2) build a text-line and the next two phases (2.3 and 2.4) detect the skew angle and the orientation of the text-line.

In phase 2.1 the nearest block of B must be found. Starting from the boundaries of block B (Figure 2a), it searches for the first black pixel. If not found the radius is incremented (Figure 2b) and the search is repeated. If a black pixel is found (Figure 2c) then it corresponds to the nearest neighbor of B (block N). A text-line is classified as *horizontal* if the block N was found at the left or right side of the block B; it is classified *vertical* if found at the top or bottom side. In figure 2c, the letter *o* (boxed) is the block B and the letter *r* (boxed) is the block N which was found at the left side of block B, thus, the text-line is classified as horizontal.



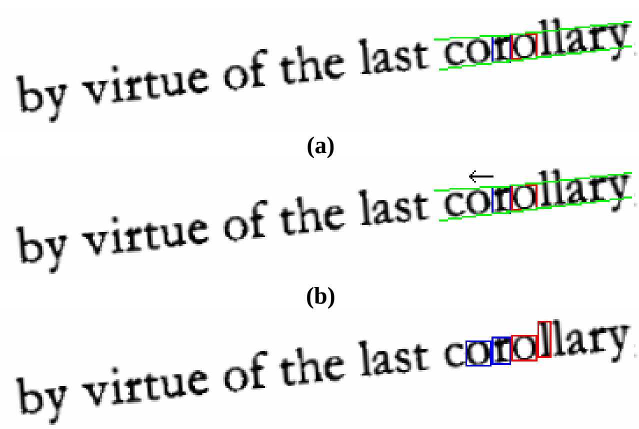
(a)

(b)

(c)

Figure 2. Phase 2.1 locates the nearest neighboring block.

Phase 2.2 attempts to merge together similar blocks forming a text-line. At this step, the basic idea of the new algorithm is implemented. First, the least square method is applied to form one line from the middle top points and another line from the middle bottom points of both blocks (Figure 3a). Only the steps for text-lines in the horizontal direction (±45º) are described here; for vertical text-lines (45º-135º; 225º-315º) the points used are middle left and middle right and the rest of the algorithm follows in a similar way. Then, two lines are drawn, making a window to limit the search for others blocks. The algorithm scans between the two lines starting from the left block boundary to its left (Figure 3b). It also sweeps to the right in a similar way to find another block and then merges them together (Figure 3c). This process is repeated until no further blocks are found.



(a)

(b)

(c)

Figure 3. Phase 2.2 merges blocks together stepwise.

Two criteria are used to control block merging. The first criterion is related to the dimensions of the blocks and uses a parameter defined as the average dimension of the blocks of the text-line. Only the blocks in a range of half of the parameter and three times larger than it can be merged to the line-text. The other criterion is related to the distance between the blocks and uses another parameter defined as the average Euclidean distance between central points of two neighboring blocks. This measure is used to limit the search window drawn. Only blocks that are at a Euclidean distance equal or less than three times the parameter should be merged together. Both parameters are updated after each block merge.

The following step determines the skew angle and the landscape/portrait orientation of the actual text-line (phase 2.3). This step accumulates the square of the number of blocks merged in the actual text-line at the two histograms of angles ranging from ±90º. The difference lies in the precision: one has 1º and the other 0.1º. The bin of the histogram to update corresponds to the angle of each line formed with all blocks of this group.

The next phase detects the up-down orientation of the text-line (phase 2.4). Using the same last bottom-line calculated in the previous phase, for each block of the text-line, the square of the maximum distance of a black pixel of the block to the bottom-line is accumulated in a variable called "*count_descending*". It is worth mentioning that the black pixel should correspond to the outermost salient part of the character (Figure 4). The same calculation is done in respect to the top-line and stored in the variable "*count_ascending*".

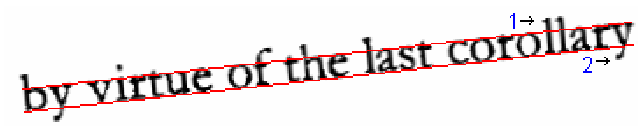


Figure 4. 1 and 2 indicates the outermost black pixels that should be used for each block in up-down orientation.

The output of the second step is two histograms with 0.1º and 1º precision and the variables *count_ascending* and *count_descending*. The last step of the algorithm calculates the total rotation of the document based on these four variables. First, it determines the peak of the histogram with 1º precision which corresponds to a coarse angle. Next, it determines the peak of histogram with 0.1º precision just only in the range of ±1º of the coarse angle. Finally, if the variable *count_descending* is greater than *count_ascending* then the resulting angle is added to 180º.

2.3 Algorithm Features

All characteristics of the algorithm are listed and its behavior is discussed. In practice, all of them are proved in the test results.

2.3.1 Parameter-Freedom

Users need not to feed parameters of the document image to be processed. There are only two internal parameters that are self-adaptive to each text-line and calculated by the algorithm. The only pre-requisite of the algorithm is that the document image must contain a minimum number of text-lines. Since the algorithm uses them as the reference point to define the global rotation of a document. The greater the number of text-lines a more precise and accurate result is obtained

It was observed that the distance between two characters of a word is smaller than the distance between two adjacent text-lines. This justifies the need for the distance parameter and also explains the way text-lines are grouped (Phase 2.1 and 2.2). The distance parameter plays two roles: (1) to limit the search for neighboring blocks and, as a consequence, it speeds up the algorithm and; (2) to avoid merging far away blocks (that could be from another text-line of another column).

2.3.2 Layout Independence

The algorithm presented groups the text-lines in the document without taking into account either their position in the image or their direction. It calculates each text-line rotation and the dominant rotation will define the global document rotation.

This means that the document image may have text-lines in different directions within the same document, that the algorithm will detect the dominant direction. Besides that, this also means that the document image may have tables, line-arts, figures and others graphical elements. The distance parameter allows documents to have multiple columns.

2.3.3 Language Dependence

It was observed that most cultures around the world organize their texts with text-lines in a structured way. The algorithm was implemented for documents written in languages with horizontal text-lines. For languages that structure the text-lines vertically (e.g. Japanese), the reference line must be rotated of 90º and thus, the changes needed to the algorithm is minimal.

The up-down orientation detection process described herein only works with languages that make use of the Latin alphabet, as the frequency of descending and ascending strokes are used for that.

2.3.4 Font and Size Independence

The first step of the algorithm is component labeling which yields a list of blocks only storing their position and dimension. The next step merges neighboring blocks together using the distance and block-size parameters as criteria. Therefore, no assumption on the contents of blocks is made. This means that the blocks may correspond to a Japanese symbol, an English character, a machine or handwritten character of font styles or size. The algorithm merges blocks of different fonts, but of similar sizes.

2.3.5 Range Angle Detection and Precision

The range angle detection of the algorithm proposed is 0º to 360º. No other algorithm described in the literature is as general. Today, many applications have to implement at least two different algorithms to reach this flexibility, thus increasing the implementation complexity and degrading the overall performance.

The algorithm is designed to have a precision of 0.1º using two different precision histograms. This value is reasonable. A higher precision brings no impact, neither on human visualization, nor on document recognition. The first histogram gives a coarse angle and second one gives a more accurate skew angle.

A small difference from the other methods must be noted: for each text-line two lines are used to detect the skew angle and landscape/portrait orientation. This helps in providing a more accurate and safer detection whenever there is a small number of text-lines in a document image.

2.3.6 Time Complexity

Most methods of the Nearest Neighbor Clustering approach have quadratic time complexity $O(n^2)$, where n is the number of blocks after the component labeling process. The proposed algorithm has a linear time complexity $O(n)$, since, for each block, it scans the neighboring region to search for another block and blocks already merged in a text-line are not processed again.

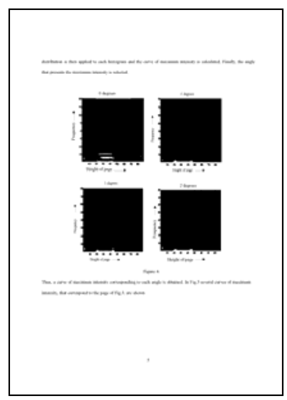
3. TEST RESULTS

A number of comparative tests were made in order to assess the quality of the results of the algorithms.

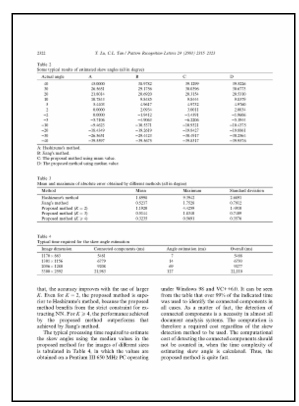
3.1 Methodology

The execution environment was a computer with processor Pentium IV of 2.4GHz and 512MB of RAM. The algorithm was implemented using the ANSI-C language compiled with Microsoft Visual C++ 6.0.

The proposed algorithm is compared with Baird's method [6]. His algorithm fits in the Projection Profile group. Tests made by Amin and Fisher [3] with algorithms [6][17][20][23][26], showed that Baird's method is the fastest of all and that it provided the second smallest error average. The choice of comparing with Baird's method is justified by: (1) both algorithms need no any input parameter, thus there is no bias; (2) both were designed to detect angles with 0.1º precision; (3) both work with monochromatic document images at 200dpi. A problem occurs because Baird's method has only an angle range detection of ±15º. Thus, only the skew angle detection will be compared.




(a) (b)

(c) (d)

Figure 5. Examples of document images used in Test 01: (a) sparse text-lines; (b, c) two columns with skewed text-lines; (d) tabular document.

Test 01 analyses 270 complex document images (Figure 5) with paper size A4 extracted from digital scientific articles, written in English with the objective of comparing skew angle detection, precision and processing time with Baird's algorithm. Documents were rotated in 49 different angles between ±15º, totaling 13,230 images.

Test 02 takes the same 270 document images used at Test 01 only with the proposed algorithm to find the whole angle range detection. The document images were rotated in 82 different angles between 0º and 360º, totaling 22,140 images.

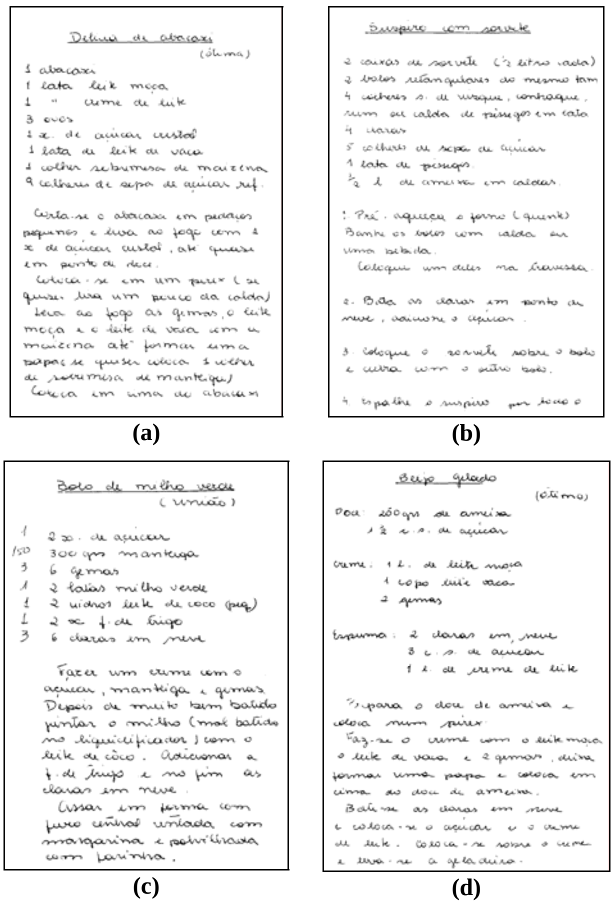


Figure 6. Examples of manuscripts used in Test 03.

Test 03 worked with 50 handwritten documents (as the ones presented in Figure 6) of page size *letter* written in Portuguese to compare the performance between Baird's and the new algorithm with handwritten documents. The images were rotated in 49 different angles in the range of ±15º, totaling 2,450 images.

Test 04 preliminary analyzed the suitability of the proposed algorithm to work with printed document images in Japanese. Images of the test set are presented in Figure 7 and correspond to pages of paper size *A4* written horizontally from left to right to test the effectiveness of Baird's and the new algorithms with others languages. The document images were rotated in 49 different angles in the range of ±15º, totaling 2,450 images.

In the next section, the results of the tests performed for the average processing (CPU) time, average angle of rotation error, maximum error found in angle detection and degree of confidence in the output of the algorithms are provided.

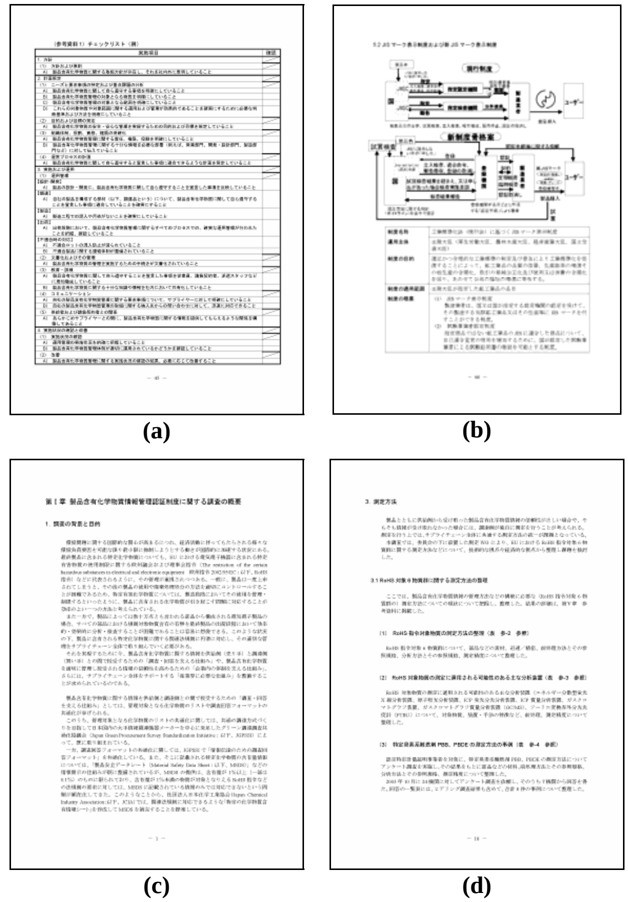


Figure 7. Examples of Japanese document images used in Test 04.

3.2 Test Results

The results of Test 01 (Table 1) shows that Baird's algorithm is faster, however, the proposed algorithm is capable of detecting a angle range 12 times larger than Baird's method and increasing only 21% the average time. The proposed method offers a small improvement at the skew angle detection with 0.001º of average error and the precision is proved to be 0.1º reaching 100% of detection when the tolerance is 0.1º. This test also proves that both algorithms work very well on complex documents.

Table 1. Results of Test 01.

Method	Average Time	Average Error	Degree of Confidence		
			0.0º	0.1º	0.2º
Baird	95 ms	0.004º	96.02%	99.97%	100%
Proposed	115 ms	0.001º	98.60%	100%	100%

In Test 02, the up-down orientation detection was miscalculated in only 12 (0.05%) of the 22.140 images proving the effectiveness of the proposed algorithm. Those 12 errors had a high impact in the resulting value as each wrong result adds up an error of 180º.

Thus, the same test was repeated with the up-down orientation detection disabled. The results on Table 2 show that the elapsed-time overhead to detect up-down orientation is minimal. This test also proves that the proposed algorithm is effective in the accurate skew angle detection for angles between 15º and 45º. Besides that, the landscape/portrait orientation was detected without errors. All images in this test were processed within 48 minutes.

Table 2. Results of Test 02.

Method	Average Time	Average Error	Degree of Confidence		
			0.0º	0.1º	0.2º
Proposed	131 ms	0.022º	98.29%	99.73%	99.94%
Proposed with up-down detection disabled	131 ms	0.002º	98.34%	100%	100%

The results of Test 03 (Table 3) show that both methods work reasonably well with handwritten documents. The highest average error in the proposed algorithm occurs because the blocks formed by the component labeling may represent a whole word, since the letters are connected. In order to improve the accuracy, a letter segmentation pre-processing step is recommended. However, the degree of confidence with 1.0º tolerance reaches 91.18% and 95.67% for Baird's and the proposed algorithm, respectively.

Table 3. Results of Test 03.

Method	Average Time	Average Error	Degree of Confidence		
			0.0º	0.1º	0.2º
Baird	24 ms	0.584º	13.30%	35.83%	52.00%
Proposed	33 ms	0.611º	11.22%	32.32%	52.85%

The result of Test 04 (Table 4) shows that Baird's algorithm performs slightly better for Japanese document images in the range of ±15º that the new algorithm presented in this paper. This shows that, although the proposed algorithm was designed and is recommended to work with documents written in the Latin alphabet, it may work suitably with documents written in other writing systems.

Table 4. Results of Test 04.

Method	Average Time	Average Error	Degree of Confidence		
			0.0º	0.1º	0.2º
Baird	86 ms	0.005º	94.20%	100%	100%
Proposed	125 ms	0.035º	71.83%	95.63%	98.48%

The plotting exhibited in Figure 8 presents the average CPU time for processing the orientation and skew detection of the documents grouped with the number of blocks in ascending order. It shows that the time of both algorithms grows linearly with the increase of the number of blocks, but the constant of proportionality is higher for the proposed algorithm. It also shows that the number of blocks in document images with page size *A4* varies between 1,000 and 5,000 blocks. Similar results are also reported by O'Gorman [23].

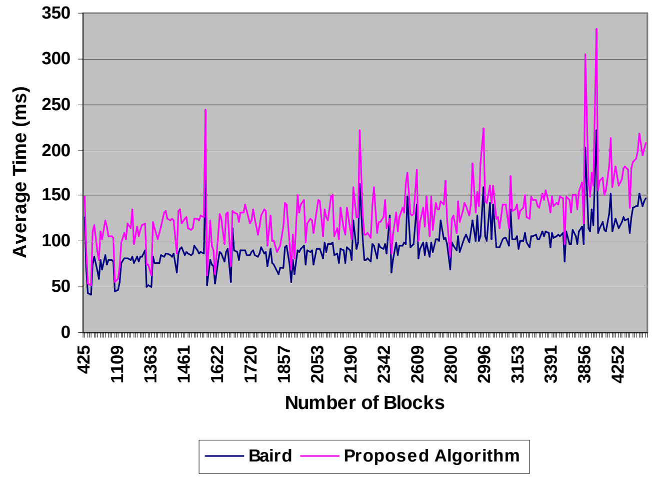


Figure 8. Time complexity of both algorithms.

As a matter of curiosity, the algorithm presented here was also tested on a small set of document images obtained with portable digital cameras. Those images were acquired with a Sony Mavica 1.3 Mpixel camera, show small lenses distortion, color, low resolution images. After binarization, the proposed algorithm was able to correctly detect the landscape/portrait orientation and skew angles (Figure 9). We expect to enlarge our test-set to check for the robustness and precision of the algorithm presented here for this purpose, as well as testing with images obtained with higher-resolution cameras.

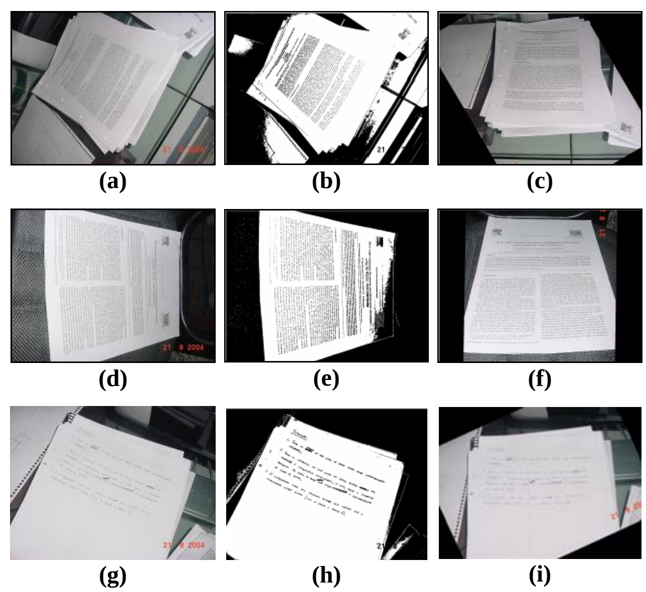


Figure 9. Document images created by a digital camera: (a, d, g) original; (b, e, h) binarized; (c, f, i) *a*, *d* and *g* rotated by detected angles of -59.6º, 89.1º, 24.8º, respectively.

4. CONCLUSIONS

This paper presents a fast algorithm for orientation and skew detection for monochromatic document images. It works with complex documents and it is able to handle documents with non-textual (figures, graphs, tables, etc.) elements and with several skews and columns. It has a range angle detection of 0º to 360º and a precision of 0.1º. The new algorithm needs no input parameters and exhibits linear time complexity.

Exhaustive comparative testing on over 20,000 images showed that the algorithm proposed is only 21% slower than Baird's, considered the fastest one [3] in literature. However, the algorithm presented herein is far more general than Baird's as it offers a range of angle detection 12 times larger than the one allowed in Baird's algorithm. Besides that, the proposed algorithm yielded a lower average error in skew angle detection.

Preliminary tests show that the new algorithm performed well also in printed Japanese documents and in document images captured by digital cameras.

It is also important to draw the readers' attention that skew detection and correction are two different phases of the problem. Skew correction is a different universe of study with particular problems. For instance, in the case of monochromatic documents, very often image rotation introduces *noise* to the image making surfaces and lines with *teeth* (unsmooth) and even separating contiguous regions. A solution for that is offered in reference [4].

5. ACKNOWLEDGMENTS

This work was partly supported by CNPq – Brazil to whom the authors are grateful.

6. REFERENCES

[1] Aghajan, H.K., Khalaj, B.H. and Kailath, T. Estimation of skew angle in text-image analysis by SLIDE: subspace-based line detection. *Machine Vision and Applications*, 1994, 7, 267-276;

[2] Akiyama, T. and Hagita, N. Automated entry system for printed documents. *Pattern Recognition*, Elsevier Science Inc., 1990, 23, 11, 1141-1154;

[3] Amin, A. and Fisher, S. A Document Skew Detection Method Using the Hough Transform. *Pattern Analysis & Applications*, 2000, 3, 3, 243-253;

[4] Ávila, B.T., Lins, R.D. and Neto, L.A.O. A New Rotation Algorithm for Monochromatic Images. In *Proceedings of the ACM International Conference on Document Engineering*, Bristol, 2005;

[5] Bagdanov, A. and Kanai, J. Projection Profile Based Skew Estimation Algorithm for JBIG Compressed Images. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*, Ulm, Germany, 1997, 401-405;

[6] Baird, H.S. The Skew Angle of Printed Documents. In Proceedings of the Conference Society of Photographic Scientists and Engineers, 1987, 14-21;

[7] Bloomberg, D.S., Kopec, G.E. and Dasari, L. Measuring document image skew and orientation. In *Proceedings of the SPIE Conference on Document Recognition II*, San Jose, California, EUA, 1995, 302-316;

[8] Caprari, R.S. Algorithm for text page up/down orientation determination. *Pattern Recognition Letters*, Elsevier Science Inc., New York, NY, EUA, 2000, 21, 311-317;

[9] Cattoni, R., Coianiz, T., Messelodi, S. and Modena, C.M. *Geometric Layout Analysis Techniques for Document Image Understanding: a Review*. ITC-IRST Technical Report #9703-09, 1998;

[10] Chen, S. and Haralick, R.M. An Automatic Algorithm for Text Skew Estimation in Document Images Using Recursive Morphological Transforms. In *Proceedings of the First IEEE International Conference on Image Processing*, Austin, Texas, 1994, 139-143;

[11] Christian, R. and Pierre, A.D. *Connected components in binary images: the detection problem*. John Wiley & Sons Inc., 1984;

[12] Ciardiello, G., Scafuro, G., Degrandi, M.T., Spada, M.R. and Roccotelli, M.P. An experimental system for office document handling and text recognition. In *Proceedings of the 9th International Conference on Pattern Recognition*, Rome, Italy, 1988, 2, 739-743;

[13] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. *Introduction to algorithms*. MIT Press, 2001;

[14] Dillencourt, M.B., Samet, H. and Tamminen, M. A General Approach to Connected-Component Labeling for Arbitrary Image Representations. *Journal of the Association for Computing Machinery*, 1992, 39, 2, 253-280;

[15] Gatos, B., Papamarkos, N. and Chamzas, C. Skew Detection and Text Line Position Determination in Digitized Documents. *Pattern Recognition*, 1997, 30, 9, 1505-1519;

[16] Hashizume, A., Yeh, P.S. and Rosenfeld, A. A method of detecting the orientation of aligned components. *Pattern Recognition Letters*, 1986, 4, 125-132;

[17] Hinds, S., Fisher, J. and D'Amato, D. A document skew detection method using run-length encoding and the Hough transform. In *Proceedings of the 10th International Conference on Pattern Recognition*, Atlantic City, NJ, 1990, 464-468;

[18] Hough, P.V.C. Methods and means for recognizing complex patterns. US Patent #3.069.654, 1962;

[19] Ishitani, Y. Document Skew Detection Based on Local Region Complexity. In *Proceedings of the 2nd International Conference on Document Analysis and Recognition*, Tsukuba, Japan, IEEE Computer Society, 1993, 49-52;

[20] Le, D.S., Thoma, G.R. and Wechsler, H. Automated Page Orientation and Skew Angle Detection for Binary Document Images. *Pattern Recognition*, 1994, 27, 10, 1325-1344;

[21] Lins, R.D. and Ávila, B.T. A New Algorithm for Skew Detection in Images of Documents. In *Proceedings of the International Conference on Image Analysis and Recognition*, LNSC, Porto, Portugal, Springer Verlag, 234-240, 2004;

[22] Min, Y., Cho, S.B. and Lee, Y. A Data Reduction Method for Efficient Document Skew Estimation Based on Hough

Transformation. In *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, IEEE Press, 1996, 732-736;

[23] O'Gorman, L. The Document Spectrum for Page Layout Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993, 15, 11, 1162-1173;

[24] O'Gorman, L. and Kasturi, R. *Document Image Analysis*. IEEE Computer Society Executive Briefing, edition, 1997;

[25] Pal, U. and Chaudhuri, B.B. An improved document skew angle estimation technique. *Pattern Recognition Letters*, 1996, 17, 8, 899-904;

[26] Postl, W. Detection of Linear Oblique Structures and Skew Scan in Digitized Documents. In *Proceedings of the 8th International Conference on Pattern Recognition*, Los Alamitos, California, IEEE CS Press, 1986, 687-689;

[27] R.Gonzalez and P.Wintz. *Digital Image Processing*. Addison Wesley, 1995;

[28] Sauvola, J. and Pietikäinen, M. Skew Angle Detection Using Texture Direction Analysis. In *Proceedings of the 9th Scandinavian Conference on Image Analysis*, Uppsala, Sweden, 1995, 1099-1106;

[29] Smith, R. A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation. In *Proceedings of the 3th International Conference on Document Analysis and Recognition*, Montreal, Canada, 1995, 1145-1148;

[30] Spitz, A.L. Correcting for variable skew in document images. *International Journal on Document Image Analysis*, Springer Verlag, 181-189, 2003,

[31] Srihari, S.N. and Govindaraju, V. Analysis of Textual Images Using the Hough Transform. *Machine Vision and Applications*, 2, 3, 141-153, 1989;

[32] Sun, C. and Si, D. Skew and Slant Correction for Document Images Using Gradient Direction. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*, Ulm, Germany, 1997, 142-146;

[33] Vailaya, A., Zhang, H. and Jain, A. Automatic Image Orientation Detection. In *Proceedings of the IEEE International Conference on Image Processing*, Kobe, Japan, 1999;

[34] Yan, H. Skew Correction of Document Images Using Interline Cross-Correlation. *CVGIP: Graphical Models and Image Processing*, 1993, 55, 6, 538-543;

APPENDIX A – Summarization of Related Work

Table 5. Cattoni *et al*. [9] updated classification of skew detection algorithms.

Method	Authors	Input Type Resolution	Skew angle range and accuracy	Characteristics
Projection Profile	Postl 1 [26]	B/w, g.l, 160 dpi	±45º / 0.6º	Complex documents with a dominant text direction
	Baird [6]	B/w, 300 dpi	±15º / 0.05º	Most text with a dominant text direction
	Ciardiello *et al*. [12]	B/w, 300 dpi	±45º / 0.7º	Complex documents
	Ishitani [19]	B/w, 300 dpi	±30º / 0.12º	Complex documents with few text-lines
	Bagdanov and Kanai [5]	B/w, JBIG, 300dpi	±3º	Most text
Hough Transform	Srihari and Govindarasu [31]	B/w, 128 dpi	±90º / 1º	Text only documents
	Hinds *et al*. [17]	B/w, 75 dpi	±15º / 0.5º	Complex documents
	Le *et al*. [20]	B/w, 200 dpi	±15º / 0.5º	Complex documents
	Min *et al*. [22]	B/w, 300 dpi	±20º / 0.5º	Documents with tables
	Pal and Chaudhuri 1 [25]	B/w, 160 dpi	±45º / 0.2º	Complex documents with one text direction
	Vailaya *et al.* [33]	B/w, 50-75 dpi	±90º / 0.1º	Complex documents with a dominant text direction
	Amin and Fisher [3]	B/w, 200 dpi	±45º / 0.1º	Complex documents
Nearest Neighbour Clustering	Hashizume *et al*. [16]	B/w, 54-63 dpi	±90º / 5º	Simple documents
	O'Gorman [23]	B/w, 300 dpi	±90º	Text only documents
	Smith [29]	B/w, 300 dpi	±15º / 0.05º	One text direction
	Pal and Chaudhuri 2 [25]	B/w, 160 dpi	±45º / 0.2º	Complex documents with a dominant text direction
Cross Correlation	Akiyama and Hagita [2]	B/w, 200 dpi	±10º	Documents with text and graphics
	Yan [34]	B/w, g.l, color	±45º	One text direction
	Gatos *et al*. [15]	B/w 96-300 dpi	±5º / 0.05º	Complex documents with a dominant text direction
Gradient Analysis	Sauvola and Pietikäinen [28]	B/w, g.l	±20º / 1º	Complex documents with a dominant text direction
	Sun and Si [32]	G.l	±90º	Complex documents
Fourier Transform	Postl 2 [26]	B/w, g.l, 160 dpi	±45º	Dominant text direction
Morphology	Chen and Haralick [10]	B/w, 300 dpi	±5º / 0.5º	Complex documents with a dominant text direction
SLIDE	Aghajan *et al*. [1]	B/w, g.l., 144 dpi	±90º / 0.01º	Simple documents
Average Median	Lins and Avila [21]	B/w, 200-300dpi	±45º	Complex documents

Table 6. Classification of orientation detection algorithms.

Orientatio n	Authors	Method	Characteristics of documents	Comments
Landscape/ Portrait	Akiyama and Hagita [2]	Global variance of text-lines	Complex documents	Sensitive to documents with images
	Le *et al*. [20]	Local variance of text-lines	Complex documents	Attempts to solve the problems of [2]
Up-down	Bloomberg *et al*. [7]	Morphological operations	Complex document	
	Caprari [8]	Projection Profile	Faxes	