A New Rotation Algorithm for Monochromatic Images

Bruno T. Ávila, Rafael D. Lins, and Lamberto Oliveira

Abstract — **The classical rotation algorithm whenever applied to monochromatic images degrades them by introducing white holes in black areas, making edges uneven and disconnecting neighboring elements. Several algorithms in the literature address only the white hole problem. They ignore both edge ripples and do not guarantee the connectivity of neighboring areas. This paper proposes a new algorithm that solves those three problems, producing better quality images and smaller compressed file size than its predecessors. It works by detecting and rotating the critical points of the each element of the image, connecting them using a line drawing algorithm and flood-filling each region.**

Index Terms — **Monochromatic images rotation, document image analysis, skew correction, critical points, white holes, uneven edges, disconnection of the elements.**

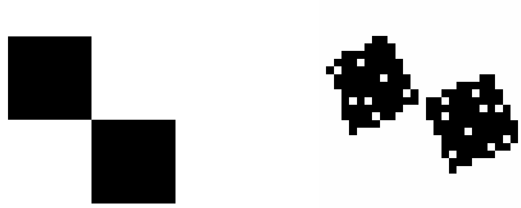
I. Introduction

Orientation and skew detection algorithms are very important pre-processing steps to document image analysis and recognition systems. They try to take advantage of some properties of the elements of documents (such as text-lines, alphabet used, etc.) to determine the orientation (up-down and landscape/portrait) and the skew angle (±45º). Several methods have been proposed in literature and can be classified as: Projection Profile [2][20], Hough Transform [1][12][14], Nearest Neighbor [11][17] and Median Average [15].

After the estimation of the document skew, the pixels of the image must be actually rotated by the detected angle. The classical rotation algorithm [3] works as a linear transformation applied to pixels in the image. First, the image is translated from the center point to the origin, rotated by the input angle and then, translated back. The classical rotation formula is:

$$\begin{aligned} x' &= \lfloor (x-c_x)\times\cos\theta-(y-c_y)\times sen\theta+c_x \rfloor \\ y' &= \lfloor (y-c_y)\times\cos\theta+(x-c_x)\times sen\theta+c_y \rfloor \end{aligned} \quad (1)$$

Where c_x is half of the image width and c_y is half of the image height. To minimize the rounding off errors in the arithmetic operations with trigonometric functions the use of double precision floating point arithmetic is recommended. However, a representation problem arises in this application: the pixels in the image must have integer coordinates thus the resulting values have to be approximated to the next integer in some way. As a consequence, three problems arise: white holes appear within flat black areas, smooth edges become uneven and become full of ripples and neighboring areas become disconnected. Fig. 1 below shows on its left hand side (Fig. 1a) an image of two connected squares often used to benchmark such rotation algorithms. The image obtained by the rotation of the original image (Fig. 1a) of about 25 degrees counterclockwise is shown on the right hand side of Fig. 1b, where one may observe the three damaging aspects mentioned above simultaneously, causing a great degradation to the original image to the point of making it almost unrecognizable.



(a) (b)

Fig. 1. (a) Two squares 10 by 10 pixels; (b) image *a* rotated of about 25º.

The use of the classical rotation algorithm in filled regions creates white holes between two neighbor pixels because many pixels are mapped into the same resulting pixel after rotated and rounded. The uneven edge and the disconnection of the elements problem are a consequence of the white hole problem, since it rounds any point including the edges. These problems arise in monochromatic, gray-level and color images because they do not depend on the value of the pixel, but just on its coordinates. It was observed that, for images of small dimensions, these problems occur for only for large angles. However, for larger images, these problems may appear for smaller angles (1).

There are several rotation algorithms for monochromatic images proposed in the literature that only solves the white hole problem. Paeth [19] proposed a three-pass algorithm in which matrix rotation is decomposed into three shearing matrices. Cheng and his colleagues [4] observed that holes appear whenever the Euclidean distance between two neighboring pixels after rotation is either 2 or square root of 5

Bruno T. Ávila, Rafael D. Lins and Lamberto Oliveira, Departamento de Sistemas e Eletrônica, Centro de Geociências e Tecnologia, Universidade Federal de Pernambuco, Recife, Pernambuco, Brasil, e-mails: bta@cin.ufpe.br, rdl@ee.ufpe.br, laon@cin.ufpe.br.

and proposed that the midpoints are filled in to eliminate holes and to maintain the connectedness regions. Danielsson and Hammerin [9] also proposed a three-pass algorithm in which the image is shifted three times each using a different matrix. Shah and Aggrawal [21] proposed an inverse mapping method in which the pixel value is determined by inversely mapping all pixels of the rotated image to the original image. Jiang et al. [13] optimize the inverse mapping method by making a mapping table, obtained in advance from the skew detection algorithm, and that can transform 16 pixels at once. In 1997, Chien and Baek [5] proposed a fast black run algorithm that for each black run in the image it rotates the starting and ending pixels and draws a line between the two rotated points. In order to eliminate holes, if the black run has an adjacent bottom black run then it draws a new line below the current line. This method is faster than the classic rotation algorithm. In 2001, Chien and Baek [6] proposed a fast hierarchical block matching method which defines coarse and fine blocks to extract bit patterns of the original image and calculate their pre-drawn mapping patterns using the given angle. In this way, the classical rotation algorithm can be replaced by simple matching of bit patterns of blocks. This method is faster than the fast black run method.

However, the uneven edge problem has not been treated by any of these methods. In fact, this problem is ignored and the edges are degraded resulting in poor image quality. As a consequence, none of the algorithms reviewed guarantees to maintain the connection between the elements.

However, there are other methods in the literature that improve the edge quality, such as, the K-fill algorithm [18]. This method is an extension to the salt-and-pepper algorithm [18] and is capable to smooth the ripples of uneven edges. Since the white holes can be treated as noise, they can be removed by this algorithm. But, it cannot guarantee to keep the connectivity between neighboring elements. In the case of halftone images, the use of the K-fill algorithm is inadequate, because it blurs flat the halftone patterns.

This paper presents a new rotation algorithm for monochromatic images. It solves the white hole, uneven edges and disconnection between elements problems and generates better quality images. It was also verified that the compressed file size is smaller than its preceding algorithms. In order to assess the quality of resulting images, a novel comparison quantitative method is presented.

II. The Algorithm

The algorithm proposed herein attempts to decrease the number of points rotated by applying equation 1 by performing a kind of vectorization over the edge of the image. To construct the rotated image, it draws the rotated vectors and flood-fill them. The basic steps of the algorithm are:

1. Vectorization:
   1.1. Find the edges;
   1.2. Find the critical points;
   1.3. Construct a graph of vectors;
   1.4. Remove redundant critical points and vectors;
2. Find filled polygons;
3. Rotate only the critical points using (1);
4. Draw the vectors of the rotated image;
5. Flood-fill the detected filled polygons;

The first step is vectorization of the image, which is performed as a bottom-up process from the raster pixels into a graph of the abstract representation of the image.

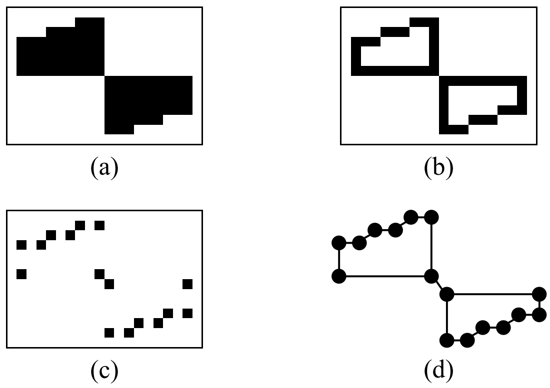


Fig. 2. (a) Original image; (b) edges; (c) critical points; (d) vector graph.

Step 1.1 attempts to find the edges of the components of the image (Fig. 2b). It is accomplished by using an edge mask table and applying it to all black pixels. The next step (1.2) determines the critical points (Fig. 2c), the minimal set of points necessary to rotate the image. It also uses a critical point mask table and applies to all edge pixels.

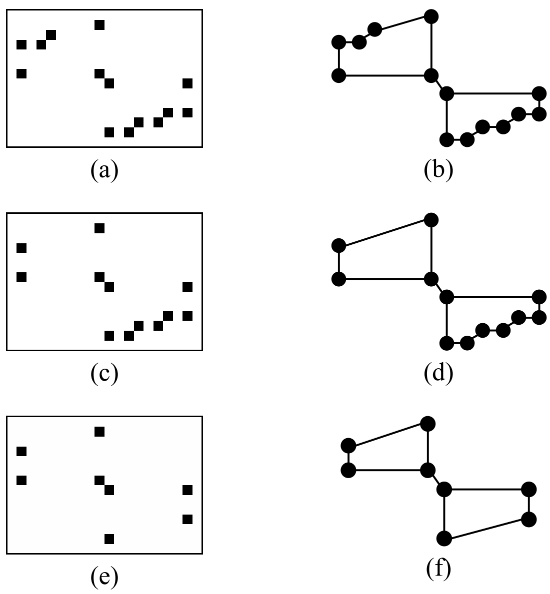


Fig. 3. Steps to remove the redundant critical points: (a, c, e) critical points being removed; (b, d, f) vector graphs without redundant points.

Step 1.3 consists of creating a graph of vectors, where the nodes represent the critical points and the edges are represented as vectors (Fig. 2d). A vector represents two neighboring critical points connected through an edge line defined in step 1.1. It works as follows: for each critical point, run through the edge line until finding a new critical point, then create a vector with both points and add it to the graph.

In many situations, there are intermediate critical points in which only two of them could represent the whole edge line. Hence, step 1.4 consists in removing those redundant critical points and merging the collinear vectors into one (Fig. 3). This step helps reducing the numbers of critical points. It works as follow: (a) for each three consecutive vectors, make a line between the starting point of the first vector and the ending point of the third vector and (b) verify if the two intermediate points belong to this line; (c) if yes, then merge the three vectors into one, remove the two intermediate critical points, get the next two vectors and go back to step a; (d) if not, just jump to step a for another vector.

In order to complete the full representation of the original image, the second step identifies the vectors that belong to a closed and filled region. It works as follows: (a) the 4x4 component labeling algorithm [7][8][10] is executed; (b) using the map of edges and critical points and the map made by the component labeling, extract all vectors that belongs to each filled region. The detected vectors will be used in the last step to flood-fill the region that they represent.

The third step simply rotates all critical points using (1) with a given angle (Fig. 4a). The fourth step draws a line for each rotated vector on the rotated image buffer (Fig. 4b). Finally, the fifth step uses the vectors identified at step 2 and applies the scan flood-fill algorithm [3] (Fig. 4c) to the region delimited by them.

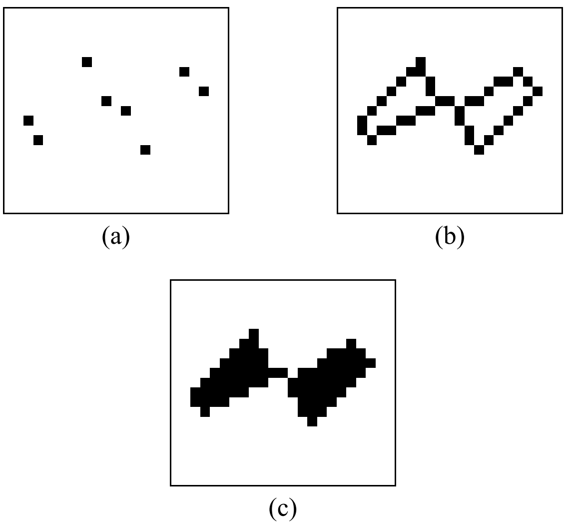


Fig. 4. (a) Critical points rotated about 30º; (b) vectors drawn; (c) flood-filled regions and final rotated image.

The white hole problem is solved by the flood-fill algorithm, because it fills every pixel inside a region bounded by a set of vectors. The uneven edge problem is solved as a consequence of the fourth step, because the line drawing algorithm creates a uniform edge for each vector. The elements are maintained connected, because the vectors graph created also represents the link between two elements, hence, these vectors are drawn at the fourth step.

Since every rotation algorithm degrades the original image, the first step of the algorithm may be replaced by any other vectorization algorithm [22][23], including lossless and lossy types. In this way, it is feasible to accept an error tolerance at step 1.4b. However, it was not implemented in this work. In order to improve further the quality of the resulting image, future implementations may include the vectorization of arcs as curved edges. In this paper, only straight lines were vectorized.

In the next section, a novel method to measure the quality of rotated images is presented.

III. MEASURING IMAGE QUALITY

All rotation algorithms in the literature degrade the original images during rotation. Visual inspection of the quality of the resulting image is useful only in the case of simple images such as the one presented in Fig. 1 above, used by most algorithms to show their results. Real document images are far more complex than the "connected-squares" (Fig. 1a). The algorithm proposed herein besides filling in the white hole introduced by rotation, it guarantees to maintain the disconnection of neighboring elements and keeps the smoothness of edges, features that its predecessors are not able to offer. Visual inspection of the rotated images provides a weak qualitative assessment of the performance of the algorithms under comparison. Therefore, it is necessary to quantify by how much these algorithms degrade the original image in the rotation process. Thus, a quantitative method to measure the degradation level of the rotation algorithms in monochromatic images is introduced here.

The methodology for the quantitative analysis is presented as follows. In the first step, the original image is rotated clockwise of an angle θº and then, it is rotated counterclockwise to the original position by -θº, using the same rotation algorithm both ways. Then, the number of pixels that differs from the original image to the final rotated image is counted, starting from the midpoint of the image. It is important to remark that rotation alters the original image causing a "frame expansion", the rotated image is larger than the original one. Each rotation algorithm produces an image of different size. Therefore, a quantitative comparison is far from being a trivial task and has not been presented by any other paper in the literature. PSNR (Peak Signal to Noise Ratio) analysis, which was successfully used in comparing images with losses [16], is not effective in this context, either. It is difficult to align images to establish a fixed reference point in order to count the non matching pixels. To solve this problem, the original image has to be cropped to remove the white margin before and after each rotation. If the final rotated and cropped image has different dimension size when compared to the original cropped image, then the exceeding pixels rows or columns are counted as incorrect (non-matching). Thus, the percentage between the number of incorrect pixels and the total number of pixels is used as a measure of image quality. The smaller the obtained percentage, the better is the quality of the image, consequently the better the algorithm.

An example is given to exhibit the proposed measure method by comparing the results from the proposed rotation algorithm and the classical rotation algorithm. Fig. 2a was cropped resulting in Fig. 5a. Using the proposed rotation algorithm, Fig. 5a was rotated by 30º (Fig. 5b), then it was rotated back of -30º and cropped yielding Fig. 5c. The same steps were performed using the classical rotation algorithm (Fig. 5d,e).

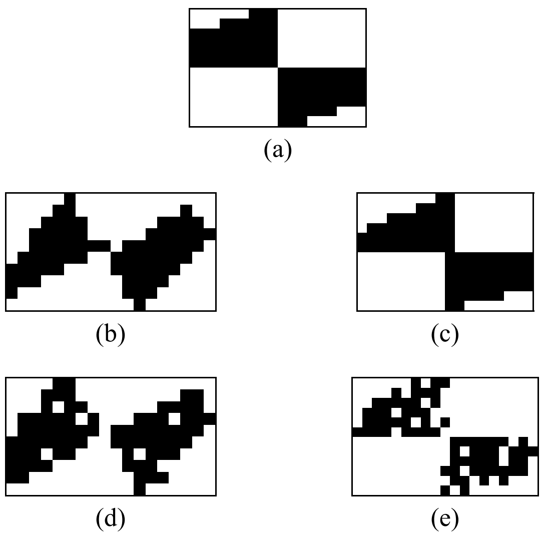


Fig. 5. An example to measure the quality of the image produced by proposed algorithm and the classical algorithm.

Fig. 5c has 18x12 pixels and has the same size of the original cropped image (Fig. 5a). However, Fig. 5e has 19x12 pixels and, consequently, it will result in a greater number of incorrect pixels. The total number of pixels in Fig. 5c is 216, thus there are 16 incorrect pixels indicating a 7.4% of degradation. The total of pixels in Fig. 5e is 228, of which 40 are considered incorrect yielding 17.5% of degradation. These results are summarized on table I. In other words, the proposed rotation algorithm produces better image quality than the classic rotation algorithm.

TABLE I
RESULTS FROM THE PROPOSED MEASURE METHOD

Algorithm	Total of pixels	Incorrect Pixels	Degradation
Proposed	216	16	7.4%
Classical	228	40	17.5%

The results of the proposed quantitative method are consistent with the qualitative results obtained by visual inspection. In the next section, the proposed rotation algorithm is compared and measured with other rotation algorithms using over 2,000 document images.

IV. TEST RESULTS

This section analyses the space performance and the quality of the images obtained by using the new algorithm, the classical rotation algorithm [3] and the Fast Black Run algorithm [5]. No pre or post-processing was performed on any image, except for the cropping of the white margin required by the proposed measurement method. All algorithms were implemented in ANSI-C language and the tests were executed on an Intel Pentium IV 3 GHz and 512MB RAM.

An image database was created with 2,000 monochromatic document images extracted from books, thesis and papers. It includes digital and scanned document images with textual and non-textual elements, such as, halftone pictures (Fig. 6). All images are 300dpi and stored using the TIFF file format using the compression algorithm CCITT G4.

A. Image Quality

This section evaluates the quality of rotated images by visual inspection and by measuring the degradation level for all rotation algorithms, according to the methodology proposed in the last section.

In order to measure the degradation level, all images were cropped, rotated of 45º, cropped again, rotated back of -45º and cropped for the last time. The results are summarized on table II.

The results show that the proposed rotation algorithm degrades less than any other rotation algorithm. The classical rotation algorithm resulted as the worst rotation algorithm. This result is compatible with the visual inspection of images.

TABLE II
MEASURE OF THE 2,000 DOCUMENT IMAGES QUALITY

Algorithm	Degradation Level
Classical	3.50%
Fast Black Run	3.01%
Proposed	2.52%

To provide to the reader a more realistic example of the effects of the rotation algorithms on images, the word "Using" that was cropped from the twice rotated version of the Fig. 6a for each algorithms was zoomed and presented in Fig. 7.

The resulting image of the classical algorithm (Fig. 7b) presents the white hole and the uneven edges problems, producing a kind of texture inside the letters. The Fast Black Run algorithm (Fig. 7c) solves the white hole problem, as expected. However, edges have lost their smoothness. The proposed rotation algorithm solves both problems effectively and produces better quality image.

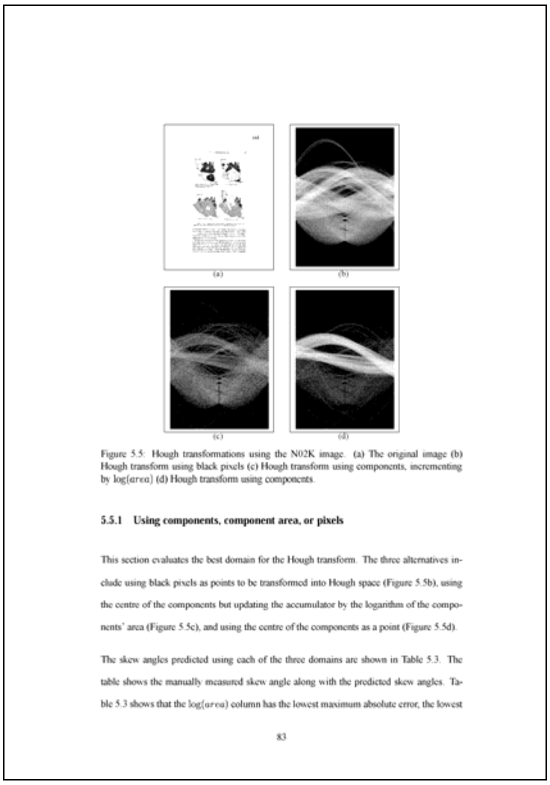


Figure 5.5: Hough transformations using the N02K image. (a) The original image (b) Hough transform using black pixels (c) Hough transform using components, incrementing by log(*area*) (d) Hough transform using components.

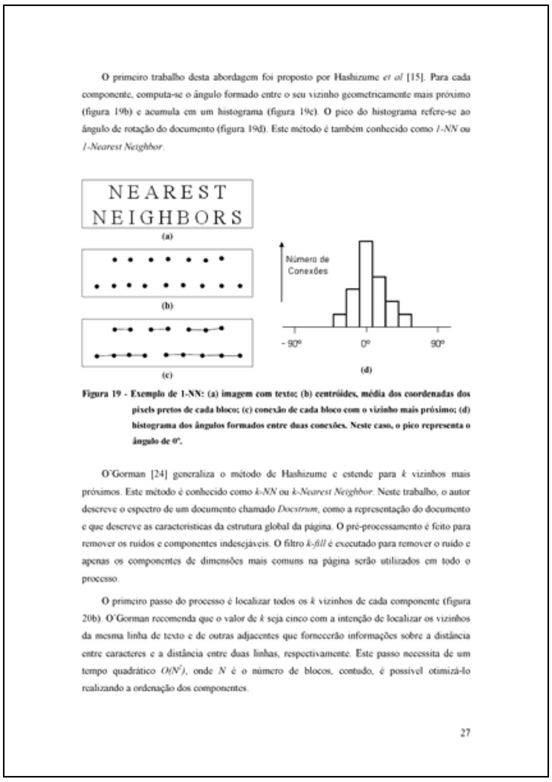
5.5.1 Using components, component area, or pixels

This section evaluates the best domain for the Hough transform. The three alternatives include using black pixels as points to be transformed into Hough space (Figure 5.5b), using the centre of the components but updating the accumulator by the logarithm of the components' area (Figure 5.5c), and using the centre of the components as a point (Figure 5.5d).

The skew angles predicted using each of the three domains are shown in Table 5.3. The table shows the manually measured skew angle along with the predicted skew angles. Table 5.3 shows that the log(*area*) column has the lowest maximum absolute error, the lowest

83

(a)



O primeiro trabalho desta abordagem foi proposto por Hashizume *et al* [15]. Para cada componente, computa-se o ângulo formado entre o seu vizinho geometricamente mais próximo (figura 19b) e acumula em um histograma (figura 19c). O pico do histograma refere-se ao ângulo de rotação do documento (figura 19d). Este método é também conhecido como *1-NN* ou *1-Nearest Neighbor*.

Figura 19 - Exemplo de 1-NN: (a) imagem com texto; (b) centróides, média dos coordenadas dos pixels pretos de cada bloco; (c) conexão de cada bloco com o vizinho mais próximo; (d) histograma dos ângulos formados entre duas conexões. Neste caso, o pico representa o ângulo de 0º.

O'Gorman [24] generaliza o método de Hashizume e estende para *k* vizinhos mais próximos. Este método é conhecido como *k-NN* ou *k-Nearest Neighbor*. Neste trabalho, o autor descreve o espectro de um documento chamado *Docstrum*, como a representação do documento e que descreve as características da estrutura global da página. O pré-processamento é feito para remover os ruídos e componentes indesejáveis. O filtro *k-fill* é executado para remover o ruído e apenas os componentes de dimensões mais comuns na página serão utilizados em todo o processo.

O primeiro passo do processo é localizar todos os *k* vizinhos de cada componente (figura 20b). O'Gorman recomenda que o valor de *k* seja cinco com a intenção de localizar os vizinhos da mesma linha de texto e de outras adjacentes que fornecerão informações sobre a distância entre caracteres e a distância entre duas linhas, respectivamente. Este passo necessita de um tempo quadrático $O(N^2)$, onde N é o número de blocos, contudo, é possível otimizá-lo realizando a ordenação dos componentes.

27

(b)

Fig. 6. Examples of the image database: (a) halftone images; (b) text with art lines.

(a)

(b)

(c)

(d)

Fig. 7. The word "Using" of fig. 6a rotated by 45º and -45º for each algorithm: (a) original; (b) classical; (c) fast black run; (d) proposed.

B. Space Performance

This section evaluates the space performance of all rotation algorithms applied to the original set of images using TIFF file format using compression algorithm CCITT G4.

The total compressed size of the original set of images has 105.944.920 bytes. The compressed image size of the first rotation of 45º and the second rotation of -45º obtained by the rotation algorithms are summarized on table III.

The total size of the set of compressed images of the classical rotation algorithm is the largest. This occurs because of the white holes and the uneven edges that produce more and shorter black runs which degrade the compression algorithm CCITT G4. The result in size of the Fast Black Run algorithm is smaller than the classical one, because it fills in the white holes within black areas, thus reducing the number of black runs. However, the total image size is bigger than using the proposed algorithm, because the remaining uneven edges will shorten black runs. The proposed rotation algorithm yielded the smallest compressed image size, because it generated the smallest number of black runs.

TABLE III
RESULTS OF THE SIZE, IN BYTES, OF COMPRESSED IMAGES

Algorithm	Compressed image size	
	1st rotation	2nd rotation
Classical	281.370.126	273.545.102
Fast Black Run	152.947.758	150.904.026
Proposed	129.710.828	104.225.298

The results above indicated a correlation between the total sizes of the compressed files with the image degradation level.

V. CONCLUSION

This paper presents a new rotation algorithm for monochromatic images and proposes a novel quantitative method to assess the quality of images produced by rotation algorithms.

Three problems were observed on rotated monochromatic images: white holes in black regions, disconnection of neighboring contiguous elements of the image and the introduction of ripples causing unsmooth edges. The white hole problem has already been addressed by several algorithms in the literature. The disconnection of elements of the image is mentioned in some of the algorithms in the literature, although, they do not guarantee to solve this problem. And finally, the unsmooth edge problem is treated for the first time by a rotation algorithm. The proposed rotation algorithm demonstrated to be effective in solving all three problems. Besides that, it is flexible, since any vectorization method can be used.

The new rotation algorithm was tested on 2,000 document images and compared with the classical rotation algorithm and the Fast Black Run algorithm. The quality of rotated images was compared by applying the new quantitative method proposed herein. The test results showed that the proposed algorithm yielded the smallest degradation level of all tested algorithms, both quantitative and qualitatively, by visual inspection. Besides that, the images rotated using the proposed algorithm generated the smallest compressed file size.

ACKNOWLEDGMENT

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

REFERENCES

[1] Amin, A. and Fisher, S. A Document Skew Detection Method Using the Hough Transform. Pattern Analysis & Applications, 2000, 3, 3, 243-253;
[2] Baird, H.S. The Skew Angle of Printed Documents. In Proceedings of the Conference Society of Photographic Scientists and Engineers, 1987, 14-21;
[3] Berger, M. Computer Graphics with Pascal. The Benjamin/Cummings Publishing Company Inc., 1st edition, 1986;
[4] Cheng, H.D., Tang, Y.Y. and Suen, C.Y. Parallel image transformation and its VLSI implementation. Pattern Recognition, 1990, 23, 1113-1129;
[5] Chien, S. and Baek, Y. A fast black run rotation algorithm for binary images. Pattern Recognition Letters, Elsevier Science Inc., New York, NY, EUA, 1998, 19, 455-459;
[6] Chien, S. and Baek, Y. Hierarchical block matching method for fast rotation of binary images. IEEE Transactions on Image Processing, 2001, 10, 3, 483-489;
[7] Christian, R. and Pierre, A.D. Connected components in binary images: the detection problem. John Wiley & Sons Inc., 1984;
[8] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. Introduction to algorithms. MIT Press, 2nd edition, 2001;
[9] Danielsson, P. and Hammerin, M. High accuracy rotation of images. CVGIP: Graphical Model and Image Processing, 1992, 54, 4, 340-344;
[10] 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;
[11] Hashizume, A., Yeh, P.S. and Rosenfeld, A. A method of detecting the orientation of aligned components. Pattern Recognition Letters, 1986, 4, 125-132;
[12] 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;
[13] Jiang, H.D., Han, C.C. and Fan, K.C. A fast approach to the detection and correction of skew documents. Pattern Recognition Letters, 1997, 18, 675-686;
[14] 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;
[15] 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, 2004.
[16] Lins, R. D. and Machado, D. S. A., A Comparative Study of File Formats for Image Storage and Transmission, vol. 13(1), pp 175-183, Journal of Electronic Imaging, Jan/2004.
[17] O'Gorman, L. The Document Spectrum for Page Layout Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993, 15, 11, 1162-1173;
[18] O'Gorman, L. and Kasturi, R. Document Image Analysis. IEEE Computer Society Executive Briefing, 1997;
[19] Paeth, A.W. A fast algorithm for general raster rotation. In Proceedings of the Graphics Interface '86/Vision Interface '86, Vancouver, British Columbia, Canada, Canadian Information Processing Society, 1986, 77-81;
[20] 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, Califórnia, IEEE CS Press, 1986, 687-689;
[21] Shah, S. and Aggarwal, J.K. Intrinsic parameter calibration procedure for a (high-distorced) fish-eye lens camera with distortion model and accuracy estimation. Pattern Recognition, 1996, 29, 1775-1778;
[22] Teh, C.H. and Chin, R.T. On the Detection of Dominant Points on Digital Curves. IEEE Trans. Pattern Anal. Mach. Intell., IEEE Computer Society, 1989, 11, 8, 859-872;
[23] Wen-Yen, W. and Mao-Jiun, J.W. Document image analysis. Detecting the dominant points by the curvature-based polygonal approximation. IEEE Computer Society Press, 1995;