Simultaneous Document Margin Removal and Skew Correction Based on Corner Detection in Projection Profiles

M. Mehdi Haji, Tien D. Bui, and Ching Y. Suen

Centre for Pattern Recognition and Machine Intelligence, Concordia University, 1455 de Maisonneuve Blvd. West, Montreal, Quebec, Canada, H3G 1M8
{m_haji,bui,suen}@cs.concordia.ca

Abstract. Document images obtained from scanners or photocopiers usually have a black margin which interferes with subsequent stages of page segmentation algorithms. Thus, the margins must be removed at the initial stage of a document processing application. This paper presents an algorithm which we have developed for document margin removal based upon the detection of document corners from projection profiles. The algorithm does not make any restrictive assumptions regarding the input document image to be processed. It neither needs all four margins to be present nor needs the corners to be right angles. In the case of the tilted documents, it is able to detect and correct the skew. In our experiments, the algorithm was successfully applied to all document images in our databases of French and Arabic document images which contain more than two hundred images with different types of layouts, noise, and intensity levels.

Keywords: Document margin, layout analysis, projection profile, corner detection, skew correction.

1 Introduction

Document processing technologies are concerned with the use of computers for automatic processing of different kinds of media containing text data. Examples of the applications are Optical Character Recognition (OCR), digital searchable libraries, document image retrieval, postal address recognition, bank cheque processing and so on. In most of these applications, the source of data is an image of a document coming from a scanner or a photocopier. During the process of scanning or photocopying, an artifact, which we simply refer to as margin, is added to the image. These black margins are not only a useless piece of data and unpleasant when the page is reproduced (reprinted), but also can interfere with the subsequent stages of document layout analysis and page segmentation algorithms. Therefore, it is desirable or necessary to remove these margins before any subsequent stages in a document processing application. Despite its practical significance, this problem is often overlooked or not discussed thoroughly in papers. There are only a few studies which have addressed

P. Foggia, C. Sansone, and M. Vento (Eds.): ICIAP 2009, LNCS 5716, pp. 1025–1034, 2009.

the problem of document margin removal. Manmatha and Rothfeder in [1] have proposed a novel method using scale spaces for segmenting words in handwritten documents wherein they have used the basic technique of projection profiles for the detection of document margins. It is easy to obtain the margins from the projection profiles when the document is not tilted and the page is a perfectly straight rectangle. But as shown in Fig. 1, this is not always the case. The page may be skewed, and it may not be a perfect rectangle. Also, any of the four margins may be present or not. The basic technique discussed in [1], is not able to handle these cases. Peerawit and Kawtrakul in [2] have proposed a marginal noise removal method based upon edge detection. They have used the edge density property of the noise and text areas to detect the border between them. This method is designed to remove left and right margins only, and is incapable of handling skewed pages.

(a) (b) (c)

Fig. 1. Examples of documents images with margins

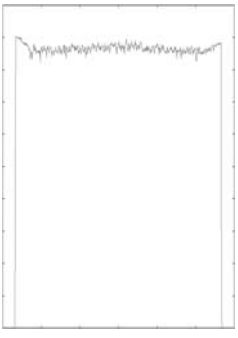
In [3], Fan et al. have proposed a top-down approach to margin removal. Firstly, the image is divided by locating possible boundaries between connected blocks. Next, the regions corresponding to marginal noise are identified by applying some heuristics based upon shape length and location of the split blocks and finally these regions are removed. Fan et al.'s algorithm is able to remove marginal noise from skewed pages, but it can not correct the page skew. Moreover, it does not find the page borders, i.e. it only removes the marginal noise and if portions of a neighboring page are present in the image, they will not be removed.

In [4, 5], Shafait et al. have used a geometric matching algorithm to find the optimal page frame. Their method is based on extraction and labeling of connected components at the first stage. Text lines and text zones must be identified prior to margin detection. However, extracting text lines from a page is a challenging task, especially for unconstrained handwritten types of documents [6, 7]. In fact, Shafait et al.'s algorithm is designed for machine printed documents. Moreover, it assumes the page frame is an axis-aligned rectangle (i.e. again, it can not handle skewed pages).

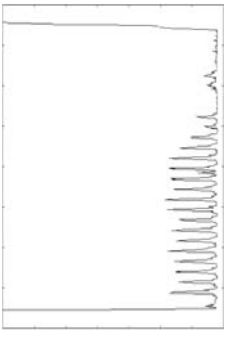
In [8], Stamatopoulos et al. have proposed a border detection algorithm for camera document images. Their method is based upon projection profiles combined with a connected component labeling process. But again, it needs the document skew to be corrected prior to margin removal.

There are several other published works concerning the problem of margin removal [9-12], but to the best of our knowledge, the algorithm we present in this paper is the first to address the problem of margin removal in presence of document skew.

(a) Input Document Image



(b) Vertical Projection Profile



(c) Horizontal Projection Profile

Fig. 2. A document image with margin and the corresponding vertical and horizontal projection profiles

2 Document Margin Removal and Skew Correction Using Projection Profiles

In this section we explain the margin removal algorithm, starting with the case of straight pages and then generalizing it to handle skewed pages.

2.1 Margin Removal for Straight Pages

The basic function of the algorithm is to find the corners which correspond to the page margins from the projection profiles of the input image. For a straight page, the left-most and right-most sharp corners in the horizontal profile of the image correspond to the left and right margins, and the left-most and right-most sharp corners in the vertical profile of the image correspond to the upper and lower margins (Fig. 2). Carrying out this task may appear simple, however the difficulty of implementation lies in corner detection, which is one of the most studied and open problems in computer vision. But in our case, by searching for the corners in 1-D projection profiles, rather than a 2-D image, we encounter a problem which can be easily solved.

Much research has been conducted upon the subject of corner detection in computer vision literature. This research can be broadly classified into two categories: grey-level and boundary-based [13]. In the first category, corners are found by using corner templates or computing the gradient at edge pixels. In the second category, corners are found by analyzing the properties of boundary pixels. For our case, we have chosen a boundary-based approach because we want to obtain corners from 1-D profiles which correspond to the document boundaries. We use a modification of the *K*-Cosine measure presented in [13] which is a new and robust algorithm for position, orientation, and scale invariant boundary-based corner detection for 2-D images.

The K-Cosine measure for a set of boundary points S = { $P_i \mid i = 1, 2, \ldots, m$ } is defined for each point i as follows:

$$c_i(K) = \cos\theta_i = \frac{\vec{a}_i(K).\vec{b}_i(K)}{\| \vec{a}_i(K) \| . \| \vec{b}_i(K) \|} \tag{1}$$

Where $\vec{a}_i(K) = \vec{P}_{i+K} - \vec{P}_i$ and $\vec{b}_i(K) = \vec{P}_{i-K} - \vec{P}_i$ are the two vectors connecting the point i to the K^{th} point before and after it, and θ_i denotes the angle between these two vectors. Therefore, K-cosine provides a measure of the curvature of boundary points over a region of support specified by K.

The overall performance of the 2-D corner detection algorithm based on the K-Cosine measure greatly depends on K. In [13], a careful analysis and a method of choosing a proper value for K is discussed, which is based on some geometric properties of the input set of boundary points. But, in our simplified 1-D version of the problem, where we are looking for corners in 1-D profiles, even a fixed value of K will work fine. Because, firstly, the corners of interest are almost right angles, secondly, they are located near the left and right ends of the boundary (i.e. projection profile), thirdly, there is only zero or one corners at each end (depending on whether or not the margin is present).

As the value of K is fixed in our application, we modify the definition of the K-Cosine measure in order to make sure that the corner detection scheme is robust against profile noise. We simply use a low-pass filtering which can be implemented as an averaging operation. More precisely, for each point of a projection profile, now we take the average of the K-Cosine measure over a local neighborhood of K. This new curvature measure is defined as follows:

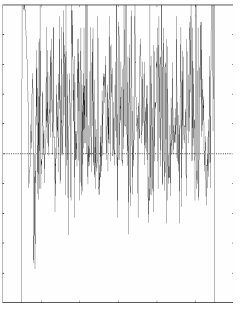
$$C_i(K) = \frac{1}{K} \sum_{k=K/2}^{3K/2} c_i(k) \tag{2}$$

Having defined the curvature measure, we apply it to all points of the projection profile to obtain the corresponding Averaged K-Cosine Curvature Curve (AKC2). Now, the first zero-crossings of AKC2, scanning from left to right and right to left, correspond to the left and right corners of the projection profile. This is due to the fact that K-Cosine values vary between $-1 = \cos(\pi)$ and $1 = \cos(0)$, and thus the AKC2 curve has to cross the axis at the left and right rising edges of the corresponding profile. Please note that, even if the projection profile is not an exact rectangle function (i.e. it does not have 90-degree corners), the AKC2 curve still has two zero crossings which correspond to the left and right (or top and bottom) margins. Fig. 3 shows the document image of Fig. 2 with the corresponding AKC2 curves which determine the four margins of the image and the final result of margin removal.

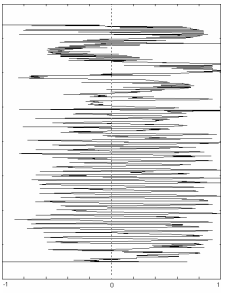
2.2 Margin Removal for Skewed Pages

For skewed pages we observe that horizontal and vertical projection profiles have an isosceles trapezoidal shape as shown in Fig. 4. In this case, we need to estimate the base angle of the corresponding trapezoid to be able to correct the page skew. Let $T_{vpp}(\mathrm{I})$ and $T_{hpp}(\mathrm{I})$ denote the trapezoids corresponding to the vertical and horizontal

(a) Input Document Image



(b) Vertical AKC2 Curve



(c) Horizontal AKC2 Curve

(d) Result of Margin Removal Algorithm

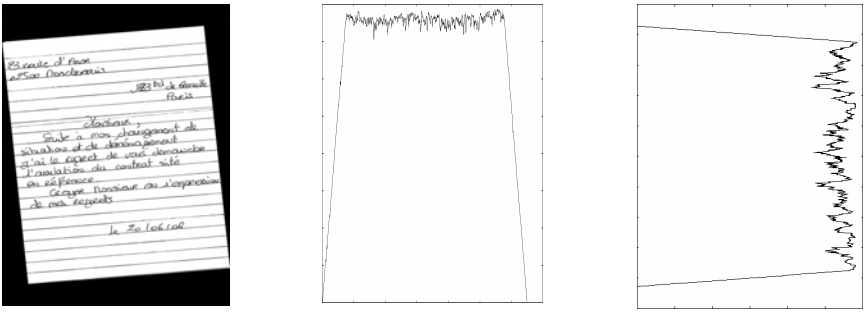
Fig. 3. A document image with margin and the corresponding AKC2 curves and the result of margin removal algorithm

projection profiles of the input document image I respectively. The base angle of T_{vpp} which is the angle that the two non-parallel sides of it make with vertical axis, or equivalently, the base angle of T_{hpp} which is the angle that the two non-parallel sides of it make with horizontal axis is equal to the page skew angle.

In order to estimate the base angle, we use the same technique discussed in the previous section for finding corners in projection profiles. However in this case, we need all the four corners (i.e. the four vertices of the corresponding trapezoid).

Let V_1, V_2, V_3 and V_4 denote the four vertices of T_{vpp}, and H_1, H_2, H_3 and H_4 denote the four vertices of T_{hpp} as shown in Fig. 5. V_2 and V_3, and H_2 and H_3 can be found from the corresponding AKC2 curves, exactly the same way we did in the previous section. However, for H_1 and H_4, and V_1 and V_4, it should be noted that these corners may be very close to, or exactly lie on, the two ends (boundaries) of the corresponding profiles. Therefore, the AKC2 may not provide an appropriate measure of curvature to find them. We can easily handle this boundary problem by padding the profiles with enough ($> K$) number of zeros, corresponding to fictitious black margins on the four sides of the input document image.

Having obtained the coordinates of the vertices of T_{hpp} and T_{vpp}, we can calculate the absolute value of the page skew angle, but not the sign of it. As shown in Fig. 6, an axis-aligned rectangle when tilted to the left and to the right by the same skew angle θ, result in the same horizontal and the same vertical projection profiles. The



(a) Input Document Image (b) Vertical Projection Profile (c) Horizontal Proj. Profile

Fig. 4. A skewed document page with margin and the corresponding vertical and horizontal projection profiles

(a) T_{vpp} (b) T_{hpp}

Fig. 5. Trapezoids corresponding to vertical and horizontal projection profiles of a skewed document page with margin

proof is trivial by noting that the areas of the triangles $L_1L_2T_1'$ and $R_1R_2S_2'$; $L_3L_4T_3'$ and $R_3R_4S_4'$; $L_1L_4T_4'$ and $R_1R_4S_1'$; and $L_2L_3T_2'$ and $R_2R_3S_3'$ are equal by symmetry; and so are the areas of the parallelograms $T_4'L_2T_2'L_4$ and $R_1S_3'R_3S_1'$; and $L_1T_1'L_3T_3'$ and $S_2'R_2S_4'R_4$. Where T_1' is the intersection of the line segments L_1T_1 and L_2L_3; T_2' is the intersection of the line segments L_2T_2 and L_3L_4; T_3' is the intersection of the line segments L_3T_3 and L_1L_4; and T_4' is the intersection of the line segments L_4T_4 and L_1L_2; and similarly for S_1', S_2', S_3' and S_4'.

(a) (b)

Fig. 6. An axis-aligned rectangle tilted to the left and to the right by the same angle

In Fig. 6, the inner rectangle $P_1P_2P_3P_4$ can correspond to the bounding box of a page of document without skew and margin. Then, the rectangles $L_1L_2L_3L_4$ and $R_1R_2R_3R_4$ are the skewed versions of it, and the triangles $I_1L_1L_2$, $I_2L_2L_3$, $I_3L_3L_4$, $I_4L_4L_1$, $I_1R_1R_4$, $I_2R_2R_1$, $I_3R_3R_2$ and $I_4R_4R_3$ correspond to the black (dark) margins around the page.

In our problem, given the horizontal and vertical projection profiles, we want to find the page corners (i.e. the coordinates of the rectangle $P_1P_2P_3P_4$). We do this by first obtaining the coordinates of $L_1L_2L_3L_4$ and $R_1R_2R_3R_4$ and then determining the sign of the skew angle. Let V_{1x}, V_{2x}, V_{3x} and V_{4x} be the indices of the four columns of the image corresponding to the four corners of the vertical projection profile as shown in Fig. 5(a). Let H_{1y}, H_{2y}, H_{3y} and H_{4y} be the indices of the four rows of the image corresponding to the four corners of the horizontal projection profile as shown in Fig. 5(b). Now, when $L_1L_2L_3L_4$ and $R_1R_2R_3R_4$ correspond to the left-skewed and right-skewed versions of the image, from Fig. 6, we can easily see:

$$\begin{aligned} &L_{1x} = R_{4x} = V_{4x}\ ;\ L_{4x} = R_{1x} = V_{1x}\ ;\ L_{2x} = R_{3x} = V_{2x}\ ;\ L_{3x} = R_{2x} = V_{3x} \\ &L_{2y} = R_{1y} = H_{1y}\ ;\ L_{1y} = R_{2y} = H_{2y}\ ;\ L_{3y} = R_{4y} = H_{3y}\ ;\ L_{4y} = R_{3y} = H_{4y} \end{aligned} \tag{3}$$

Or,

$$\begin{aligned} &L_1 = (V_{4x}, H_{2y});\ L_2 = (V_{2x}, H_{1y});\ L_3 = (V_{3x}, H_{3y});\ L_4 = (V_{1x}, H_{4y}) \\ &R_1 = (V_{1x}, H_{1y});\ R_2 = (V_{3x}, H_{2y});\ R_3 = (V_{2x}, H_{4y});\ R_4 = (V_{4x}, H_{3y}) \end{aligned} \tag{4}$$

Therefore we have obtained the coordinates of the skewed versions of the page from the projection profiles of it. Now, it is straightforward to calculate the absolute value of the skew angle θ. From Fig. 6, obviously we can obtain the absolute value of θ, by computing the slope of any of the eight sides of the rectangles $L_1L_2L_3L_4$ and $R_1R_2R_3R_4$. But as we pointed out earlier, the projection profiles are noisy and the page may not be a perfect rectangle; consequently, the coordinates of the skewed rectangles that we obtain from the above set of equations are estimates and not exact. Therefore, we make use of all the eight sides of the two rectangles to obtain the Maximum Likelihood (ML) estimate for the absolute value of θ:

$$\begin{aligned} |\theta| = \frac{1}{8}\Bigg\{ &\left|\tan^{-1}\left(\frac{L_{2y}-L_{1y}}{L_{2x}-L_{1x}}\right)\right| + \left|\tan^{-1}\left(\frac{L_{3x}-L_{2x}}{L_{3y}-L_{2y}}\right)\right| + \left|\tan^{-1}\left(\frac{L_{4y}-L_{3y}}{L_{4x}-L_{3x}}\right)\right| + \left|\tan^{-1}\left(\frac{L_{1x}-L_{4x}}{L_{1y}-L_{4y}}\right)\right| + \\ &\left|\tan^{-1}\left(\frac{R_{2y}-R_{1y}}{R_{2x}-R_{1x}}\right)\right| + \left|\tan^{-1}\left(\frac{R_{3x}-R_{2x}}{R_{3y}-R_{2y}}\right)\right| + \left|\tan^{-1}\left(\frac{R_{4y}-R_{3y}}{R_{4x}-R_{3x}}\right)\right| + \left|\tan^{-1}\left(\frac{R_{1x}-R_{4x}}{R_{1y}-R_{4y}}\right)\right| \Bigg\} \end{aligned} \tag{5}$$

As we mentioned earlier, from the projection profiles we can not determine the sign of the skew angle. Therefore, we need another source of information to resolve the ambiguity of whether $L_1L_2L_3L_4$ or $R_1R_2R_3R_4$ corresponds to the true bounding box of the page. We use the fact that the local deviation of image pixels along the two sides (left and right, or up and down) of any of the four borders of the page is "high", and any border of the page corresponds to one side of the true bounding box. More precisely, the deviation of image pixels along the two sides of a line segment belonging to the true bounding box is "higher" than the other candidate line segment belonging to the other bounding box. Let $ALD_w(I, L)$ be the Average Local Deviation function

which maps an area of the image I specified by the line segment L and thickness w to an integer in [0, 255], assuming the input image is an 8-bit grayscale one. The output of the function is the average of the absolute differences of the sum of w image pixels on the left and right, or top and bottom, along the line segment. If the line slope is great than 1, meaning the line segment is more vertical than horizontal, we look at the left and right side of it for computing the local deviation. Otherwise, the line slope is less than 1, meaning the line segment is more horizontal than vertical, we look at the top and bottom of it for computing the local deviation. Let $\{ L_i \mid i = 1, 2, ..., n \}$ be the set of coordinates of the image pixels corresponding to the line segment L. We obtain these coordinates by using the Bresenham's line algorithm [14]. Now, the function $ALD_w(I, L)$ can be formally defined as follows:

$$\mathrm{ALD}_w(\mathrm{I,L}) = \frac{1}{w.n}\Bigg(H\left(\frac{\left|L_{ny}-L_{1y}\right|}{\left|L_{nx}-L_{1x}\right|}-1\right)\left|\sum_{i=1}^{n}\sum_{t=1}^{w} I(L_{iy}, L_{ix}-t) - \sum_{i=1}^{n}\sum_{t=1}^{w} I(L_{iy}, L_{ix}+t)\right| + H\left(\frac{\left|L_{nx}-L_{1x}\right|}{\left|L_{ny}-L_{1y}\right|}-1\right)\left|\sum_{i=1}^{n}\sum_{t=1}^{w} I(L_{iy}-t, L_{ix}) - \sum_{i=1}^{n}\sum_{t=1}^{w} I(L_{iy}+t, L_{ix})\right| \Bigg) \tag{6}$$

Where $H(x)$ is the Heaviside step function.

Having defined the ALD function, we can check which of the rectangles $L_1L_2L_3L_4$ or $R_1R_2R_3R_4$ corresponds to the true bounding box of the page. If $L_1L_2L_3L_4$ is the true bounding box, then $ALD_w(I, L_1L_2)$ is higher than $ALD_w(I, R_1R_2)$, and vice versa. The same proposition holds true for the other three pairs of sides: L_2L_3 and R_2R_3, L_3L_4 and R_3R_4, and L_4L_1 and R_4R_1. As we do not assume the document page must have perfectly straight borders (look at the top border of the document page of Fig. 1(c) for example), we use all the four propositions to calculate the sign of the skew angle by taking a simple majority vote. We never encountered a case of a draw in our experiments. But if it happens, for example when the ALD function for two sides of $L_1L_2L_3L_4$ is higher than the two corresponding sides of $R_1R_2R_3R_4$ and is lower for the other two sides, it is either because 1) the skew angle is too small, and so we do not need to correct the skew at all, or 2) the page borders are very jagged, in which case we can try a larger value for w, for example we can multiply it by 2, and then calculate the ALD propositions again.

Having obtained the absolute value of the skew angle and the sign of it, we can correct the page skew by rotating the image by $-\theta$ around the center of the page which is the intersection of the diagonals of the bounding box.

The coordinates of the bounding box after skew correction, P_1, P_2, P_3 and P_4 (according to the naming convention of Fig. 6), determine the page margins. We again use the ML estimates:

$$\text{left margin} = (P_{1x} + P_{4x})/2 \tag{7}$$

$$\text{right margin} = (P_{2x} + P_{3x})/2 \tag{8}$$

$$\text{top margin} = (P_{1y} + P_{2y})/2 \tag{9}$$

$$\text{bottom margin} = (P_{3y} + P_{4y})/2 \tag{10}$$

3 Experimental Results

We tested our proposed algorithm on a database containing 219 French and Arabic document images with different types of margin noise, layouts and background/foreground intensity levels. As only a small percentage of the documents were skewed (25 documents in total), we added some artificially generated skewed document images to the database by randomly selecting a set of the real documents and rotating each one by a random angle within $-\pi/6$ to $\pi/6$. There were 63 of these artificially skewed samples so we obtained an equal number of straight and skewed document images. With $K = 30$ and $w = 10$ fixed throughout all the experiments, our proposed algorithm successfully estimated the skew angle (with a standard deviation of less than 0.25 degrees) and removed margins in all cases. It should be mentioned that the algorithm performance is not very sensitive to the values of K and w. We expect the algorithm to have the same performance for a wide range of values for these two parameters.

4 Conclusion

In this paper, we proposed an original algorithm for simultaneous skew correction and margin removal for document images, based upon the detection of document corners from projection profiles. The algorithm is fast (linear in the number of image pixels) and robust. We designed the algorithm in such a way that it can handle any type of document image without making any restrictive assumptions. The algorithm doest not need the input page of document to be a perfect and axis-aligned rectangle. The algorithm can handle skew, non-right-angled corners, and jagged page borders. We verified the efficiency of the algorithm by applying it to a collection of document images with different types of margin noise, layouts and intensity levels and the algorithm was successful in all cases.

Acknowledgments

The authors would like to thank Ms. Teresa Bowyer for proofreading a previous draft of this manuscript. They would also like to thank the anonymous reviewers for their insightful comments.

References

1. Manmatha, R., Rothfeder, J.L.: A Scale Space Approach for Automatically Segmenting Words from Historical Handwritten Documents. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1212–1225 (2005)
2. Peerawit, W., Kawtrakul, A.: Marginal noise removal from document images using edge density. In: 4th Information and Computer Engineering Postgraduate Workshop, Phuket, Thailand (2004)
3. Fan, K.-C., Wang, Y.-K., Lay, T.-R.: Marginal noise removal of document images. Pattern Recognition 35, 2593–2611 (2002)

4. Shafait, F., van Beusekom, J., Keysers, D., Breuel, T.: Page Frame Detection for Marginal Noise Removal from Scanned Documents. In: Ersbøll, B.K., Pedersen, K.S. (eds.) SCIA 2007. LNCS, vol. 4522, pp. 651–660. Springer, Heidelberg (2007)
5. Shafait, F., van Beusekom, J., Keysers, D., Breuel, T.M.: Document cleanup using page frame detection. International Journal of Document Analysis and Recognition 11, 81–96 (2008)
6. Du, X., Pan, W., Bui, T.D.: Text Line Segmentation in Handwritten Documents Using Mumford-Shah Model. In: Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition (ICFHR 2008), Montreal, Canada (2008)
7. Li, Y., Zheng, Y., Doermann, D., Jaeger, S.: Script independent text line segmentation in freestyle handwritten documents. IEEE Trans. Pattern Analysis and Machine Intelligence 30, 1313–1329 (2008)
8. Stamatopoulos, N., Gatos, B., Kesidis, A.: Automatic Borders Detection of Camera Document Images. In: 2nd International Workshop on Camera-Based Document Analysis and Recognition (CBDAR 2007), Curitiba, Brazil, pp. 71–78 (2007)
9. Le, D.X., Thoma, G.R., Wechsler, H.: Automated Borders Detection and Adaptive Segmentation for Binary Document Images. In: Proceedings of the International Conference on Pattern Recognition (ICPR 1996) Volume III-Volume 7276, p. 737. IEEE Computer Society, Los Alamitos (1996)
10. Ávila, B.T., Lins, R.D.: Efficient Removal of Noisy Borders from Monochromatic Documents. In: Campilho, A.C., Kamel, M.S. (eds.) ICIAR 2004. LNCS, vol. 3212, pp. 249–256. Springer, Heidelberg (2004)
11. Cinque, L., Levialdi, S., Lombardi, L., Tanimoto, S.: Segmentation of page images having artifacts of photocopying and scanning. Pattern Recognition 35, 1167–1177 (2002)
12. Zhang, Z., Tan, C.L.: Recovery of Distorted Document Images from Bound Volumes. In: Proceedings of the Sixth International Conference on Document Analysis and Recognition, p. 429. IEEE Computer Society, Los Alamitos (2001)
13. Te-Hsiu, S., Chih-Chung, L., Po-Shen, Y., Fang-Chih, T.: Boundary-based corner detection using K-cosine. In: IEEE International Conference on Systems, Man and Cybernetics, 2007. ISIC, pp. 1106–1111 (2007)
14. Bresenham, J.E.: Algorithm for computer control of a digital plotter. IBM Systems Journal 4(1), 25–30 (1965)