A Simple and Effective Approach for Border Noise Removal from Document Images

Faisal Shafait[1] and Thomas M. Breuel[2]
[1]Image Understanding and Pattern Recognition (IUPR) research group
German Research Center for Artificial Intelligence (DFKI GmbH)
D-67663 Kaiserslautern, Germany
[2]Department of Computer Science, Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
faisal.shafait@dfki.de, tmb@informatik.uni-kl.de

***Abstract*— When digitizing bound material like books or magazines, marginal noise appears along the page border. This noise consists of undesired text parts from the neighboring page and/or speckles that result from the binarization process. When a keyword based search is performed in a digitized collection, textual noise in particular poses problems since the returned search results might correspond to textual noise instead of actual contents of the page. Manually removing marginal noise for each page is not feasible in large scale digitization projects. In this paper, we present a simple and effective approach for removing both textual and non-textual noise by finding borders of noise regions using projection profile analysis. We demonstrate the effectiveness of our approach by evaluating it quantitatively on the widely used University of Washington (UW3) dataset. The results show that our approach reduces the noise ratio from 70% to 20% while retaining more than 99% of actual page contents. Comparison with state-of-the-art approaches shows that our algorithm performs comparable to them, while being simple to understand and easy to implement. We also provide an open source implementation of our method as part of the OCRopus OCR system.**

I. Introduction

Scanners have traditionally been used as the main source of capturing document images, i.e. converting a paper document into a digital image. The captured document is then processed through an optical character recognition (OCR) system to extract text from the document. The extracted text can then be used to store the document into an editable format like Microsoft Word. Besides, this text can be used to enable search in the document image. This makes keyword based retrieval possible in large collections of digitized books like in Google Book Search [1].

When a page of a book is scanned, textual noise (extraneous symbols from the neighboring page) and/or non-textual noise (black borders, speckles, ...) appear along the border of the document. Different amount of noise can be present along the border of a document image depending on the position of the paper on the scanner. In general, marginal noise along the page border can be classified into two broad categories based on its source: non-textual noise (black bars, speckles, ...) resulting from the binarization process [2], [3], and textual noise coming from the neighboring page.

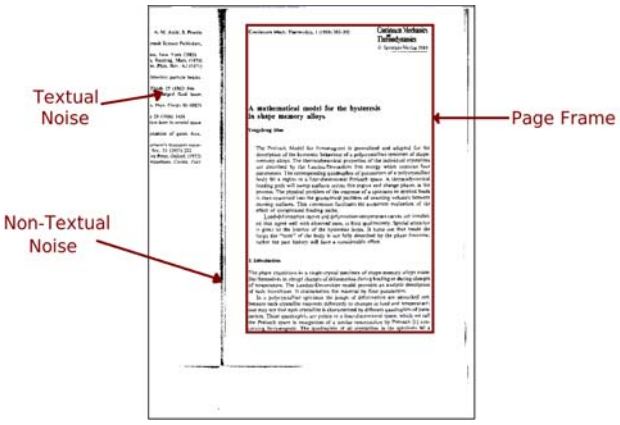


Fig. 1. Example image showing textual and non-textual noise along the page border

An example image showing textual and non-textual noise along the page border is shown in Figure 1.

Presence of non-textual noise in the image makes further processing of document like page segmentation a difficult task [4]. The most common approach to deal with non-textual noise is to perform document cleaning by filtering out connected components based on their size and aspect ratio [5], [6], [7]. This usually works out quite well in removing black bars and isolated specks. However, when characters from the adjacent page are also present, they cannot be filtered out using this approach. Therefore, state-of-the-art page segmentation algorithms report a number of false alarms originating from textual noise regions [8]. When these textual noise regions are fed to a character recognition engine, extra characters appear in the output of the OCR system along with the actual contents of the document. These extra characters in the OCR output result in inaccurate retrieval results, since the keywords that the user entered might match some text from the textual noise instead of the actual document contents.

Most of the marginal noise removal approaches reported in literature focus only on removal of non-textual noise.

978-1-4244-4873-9/09/$25.00 ©2009 IEEE

Such approaches include the work of [9], [10] and [11]. Cinque et al. [12] propose an algorithm for removing both textual and non-textual noise from greyscale images based on image statistics like horizontal/vertical difference vectors and row luminosities. However, their method is not suitable for cleaning binary images. The approach in [13] tries to identify borders of noise regions based on an analysis of the projection profiles of the edges in the image. Their technique is based on the observation that non-textual marginal noise areas have much higher density of edges than normal text. However, if the only noise present in the document is textual noise, this approach can not find the page borders. Such a scenario frequently happens in the case of thin books, since the binarization algorithm might not produce non-textual noise at all.

Instead of identifying and removing noisy components themselves, some methods focus on identifying the actual content area or the page frame of the documents [14], [15]. The page frame of a scanned document is defined as the smallest rectangle that encloses all the foreground elements of the document image as shown in Figure 1. These methods find the page frame of structured documents (journal articles, books, magazines) by exploiting their text alignment property. This is done in two steps. First, a geometric model is built for the page frame of a scanned document. Then, a geometric matching method is used to find the globally optimal page frame with respect to a defined quality function. Although these methods work quite well in practice, they require prior extraction of text lines and zones from the document images, which makes them both slow and hard to implement.

In this paper we present a simple and effective approach for border noise removal from scanned documents. Our approach is based on an analysis of projection profile of the document image. The approach is described in detail in Section II. We define different error measures to quantify different aspects of the performance of a border noise removal algorithm (Section III). Then, we present our experimental setup and results of performance evaluation in Section IV. Finally, we conclude the paper in Section V.

II. Border Noise Removal

Our algorithm for border noise removal works in three steps:

1) Black filter
2) Connected component removal
3) White filter

Each of these steps are illustrated in the following.

A. Black Filter

The black filter finds large black areas that come as a result of photocopying or scanning and removes them. It looks for these black areas only at the margins of the image so that it does not affect the text or halftones in the center of the image. It uses a rectangular window which moves in these parts of the image, calculating the ratio of black pixels under it at any position and comparing it with a threshold.

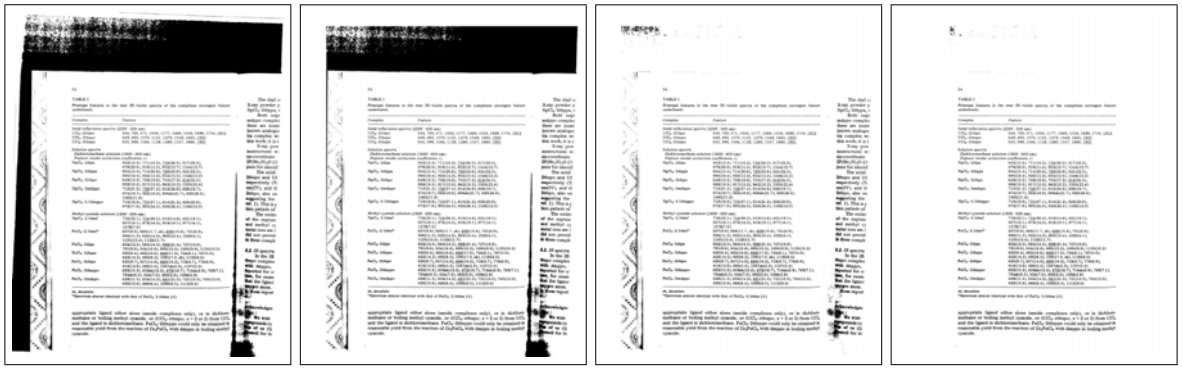
The rectangular window runs up to 1/3rd of the width or height of the image along the four margins. It starts with the left margin, starting from the x-coordinate = 1/3rd of the image width. The width of the rectangular window is specified by a parameter (default set to 5 pixels). The length or the height of the rectangular window is same as the height of the image. It counts the total number of black pixels under it at any position divided by the total number of pixels under it (equal to width of the rectangular window multiplied by its height) which gives it the ratio of black pixels. If this ratio is greater than the threshold (default set to 0.70) then it removes everything to the left of itself including itself, and also goes directly on to scanning the next margin(right margin in this case). Else, it moves leftward by the parameter called x-step (default set to 5 pixels) and continues in the same way until it reaches the left border. The rectangular window runs similarly on the right edge starting from 2/3rd of the $x$coordinate and running up to the right border. It then scans the bottom and the top borders, but while scanning the top and the bottom borders the length of the rectangular window is total width of the image minus the points where it met the threshold while scanning the left and right margins. For example, if the width of the image is 3300 pixels and while scanning the left border starting at $x = 900$ it met the threshold and removed (painted white) the entire left margin from $x = 0$ to $x = 900$, and while scanning the right border it did not meet the threshold anywhere, so while scanning the top and bottom edges it would scan between $x = 900$ and $x = 3300$. The length of the rectangular window for scanning top and bottom edges is chosen like this as the noise beyond the length of the bar has already been considered. When the black pixels ratio goes beyond the threshold while scanning top or bottom edges, it removes the part above or below along the entire width of the image.

B. Connected Component Removal

Connected component analysis first extracts all connected components from the image after applying black filter on it. All components that are very close to the border of the image are considered noise and hence removed from the image. The default border margin set for this purpose is 25 pixels. Hence all connected components whose bounding boxes either start or end within 25 pixels of page border are removed from the image. For scanned documents, this small threshold does not affects components within the page contents area due to the white margin that is typically always present along the border of actual page contents.

C. White Filter

The white filter is very similar to the black filter, the difference being that it removes everything up to the border if it finds a big white block. White filter is run on the image returned after running black filter on the original image and doing a connected component filtering. It uses a different threshold and it runs on slightly different areas of the image. Just like the black filter, white filter also runs on all four margins of the page, but for the left and right margins it



(a) Original Image (b) Black Filter Result (c) Concomp Filter Result (d) White Filter Result

Fig. 2. An example images from the UW3 dataset to demonstrate different steps of the algorithm.

starts from xcoordinate equal to 1/5th and 4/5th of the image width compared to 1/3rd and 2/3rd for the black filter. For the top margin it starts from 24/25th of the image height and for the bottom margin from 1/50th of the image height. These thresholds are chosen very small in order to prevent the page-footers from being removed as they can be very close to the bottom border. The threshold used for the white filter is 0.995, so that if the number of white pixels are more than 99.5% of the total pixels under the rectangular window, only then the portion is wiped out.

The result of applying these filter on a sample image are shown in Figure 2.

III. Error Measures

The goal of border noise removal algorithm is to remove as much of border noise as possible while retaining the actual contents of the page image. To evaluate different aspects of noise removal algorithm presented in this paper, we present the following error measures.

A. Hamming Distance

In order to measure the overall performance of the border noise removal algorithm, we compute the Hamming distance between the ground-truth image and the cleaned version of it. Since we are dealing with binary images, both ground-truth image I_{gt} and cleaned image I_c can be represented as simple one-dimensional strings with the Hamming distance given by:

$$D = I_{gt} \oplus I_c \tag{1}$$

where $\oplus$ represents the exclusive OR (XOR) operator. The distance D tells us how close or similar the cleaned image is to the ground-truth image.

B. Noise Ratio

In order to quantify the amount of border noise in a document image, the noise ratio of a document image is defined as in [15]:

$$\text{Noise ratio} = \frac{n_{pb}}{n_p} \tag{2}$$

Where n_{pb} is the number of foreground pixels outside the ground-truth page frame, and n_p is the total number of foreground pixels in actual page content area of a document image. Noise ratio tells us how much of border noise still remains in the document image relative to the page contents. This measure evaluates how well the algorithm performed in removing the border noise but does not penalize the actual page contents that were removed by the algorithm.

C. Page Contents Removal

While noise ratio quantifies the amount of noise present in the image, the purpose of measuring the percentage of ground-truth pixels, that is actual page content, removed from the image is to find the damage done by the noise removal algorithm to the page content area. This measure is defined as:

$$\text{GT Removal} = \frac{n_p - n_c}{n_p} \tag{3}$$

Where n_p is the total number of foreground pixels in actual page content area of a document image, and n_c is the total number of foreground pixels in the cleaned image that match pixels in the ground-truth image. Since noise removal does not introduce new foreground pixels, this difference is equal to the number of foreground pixels in the actual page contents (ground-truth) that were removed by the noise removal algorithm.

IV. Experiments, Results, and Discussion

The evaluation of our border noise removal algorithm was done on the publicly available University of Washington III (UW-III) database [16]. The database consists of 1600 English document images and is widely used in the document analysis community. The document images in the dataset contain a lot of noise, making it quite suitable for our experiments. Some example images from the dataset are shown in Figure 3.

The dataset comes with manually edited ground-truth of bounding boxes for page frame, text and non-text zones, text-lines and words. While the bounding boxes of zones, text-lines, and words tightly enclose their contents, this is not the same for the ground-truth page frame bounding box

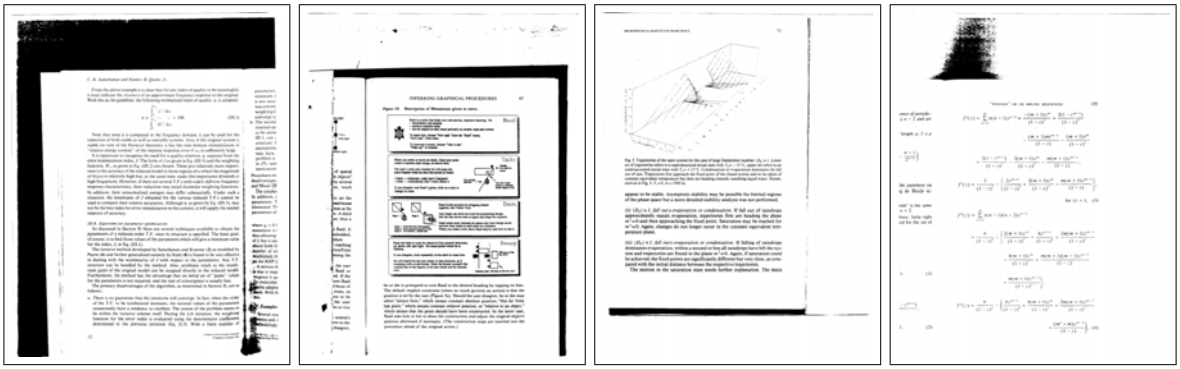


Fig. 3. Some example images from the UW3 dataset showing the variability of border noise in the dataset.

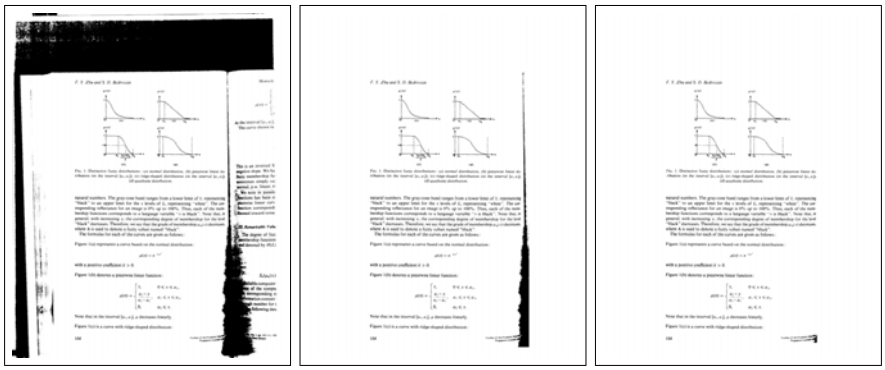


Fig. 4. An example images from the UW3 dataset and its cleaned version using ground-truth page frame (middle image) and using ground-truth zones (right image). Cleaned images of UW3 using ground-truth zone information were used as *ground-truth images* for the purpose of evaluation.

TABLE I
EVALUATION OF THE PROPOSED ALGORITHM ON UW3 DATASET. ALL VALUES ARE IN [%]

Method	Hamming Distance	Noise Ratio	Page Contents Removal
Original (no cleanup)	3.40	71.74	0.00
Unpaper [17]	2.15	7.57	8.78
Page frame detection [15]	0.79	11.98	2.45
Proposed Method	1.00	20.60	0.64

provided with the data. Instead, there is a margin between the page contents and the ground-truth page frame. In order to prepare ground-truth images for document cleanup task, the ground-truth image might still contain a small portion of border noise if the provided ground-truth is used for cleanup. Therefore we generated the ground-truth documents by using ground-truth zone information. All foreground pixels in the documents that were not contained in any of the ground-truth zones, were removed from the image. An example image demonstrating an original UW3 document, its cleanup version using the provided ground-truth page frame, and the cleanup version using ground-truth zones is shown in Figure 4. In the following, whenever we mention ground-truth image, we mean the cleaned version using ground-truth zone information.

Results of noise removal on some sample images from UW3 are shown in Figure 5. Comparative evaluation results of our proposed method with the state-of-the-art algorithms are given in Table I.

The results show that overall our algorithm performs better than the unpaper method [17] as pointed out by the Hamming distance metric, but has a slightly lower accuracy than the page frame detection technique [15]. However, the proposed algorithm does not require any pre-processing of the document or extraction of text-lines and zones which makes the proposed algorithm easy to understand and implement. Noise ratio measure shows that unpaper utility removes a larger amount of noise as compared to the page frame detection method or the proposed method. However, this removal comes at an expense of erroneous removal of actual page contents. The large percentage of actual page contents removal by unpaper might make it unsuitable for many practical document analysis applications. The proposed method, on the other hand, retains more than 99% of the actual page content while reducing the noise ratio from 70% to 20%. It should also be noted that both unpaper and page frame detection algorithms were run with their default settings.

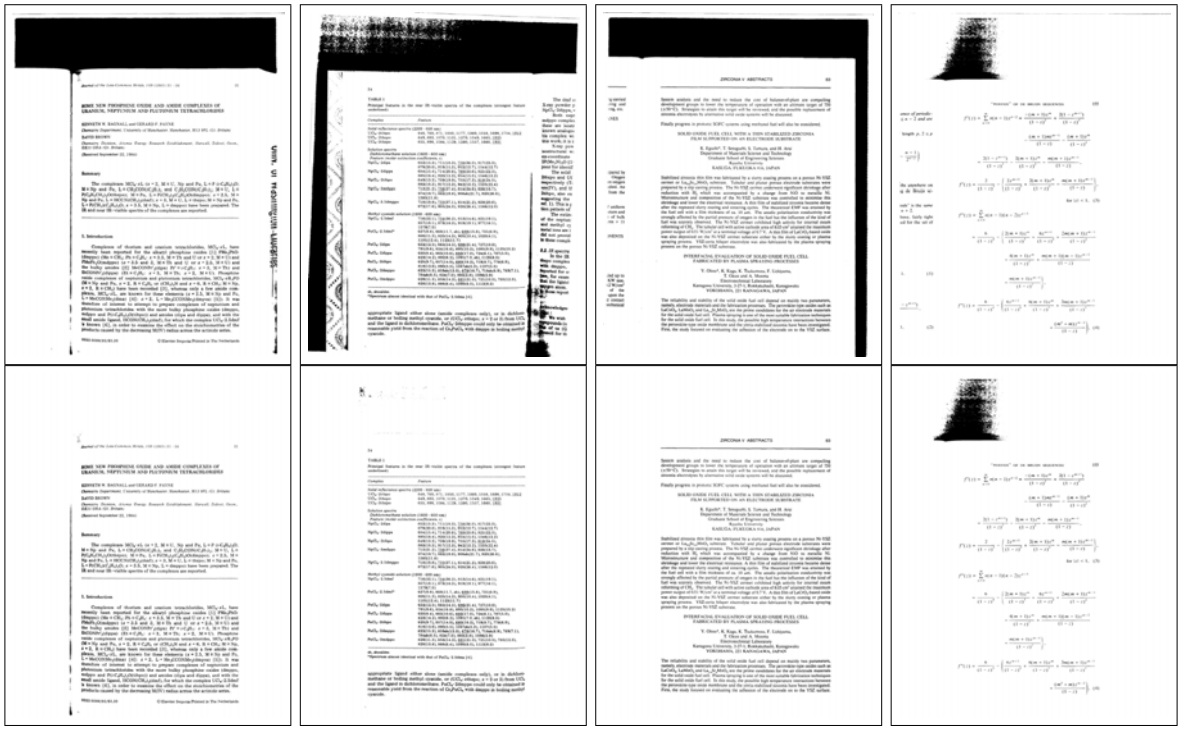


Fig. 5. Some example images from the UW3 dataset (top row) and the results of applying our noise removal algorithm on them (bottom row).

V. Conclusion

In this paper we presented a simple and efficient algorithm for border noise removal from scanned documents. The algorithm works by combining projection profile analysis with connected component removal to identify borders of noise regions. We evaluated the algorithm on the UW3 dataset and showed that while being simple to implement and understand, our method works comparable to the state-of-the-art border noise removal algorithms.

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project PaREn (01 IW 07001).

References

[1] L. Vincent. Google book search: Document understanding on a massive scale. In *9th Int. Conf. on Document Analysis and Recognition*, pages 819–823, Curitiba, Brazil, Sep. 2007.

[2] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[3] F. Shafait, D. Keysers, and T.M. Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. *Proc, of SPIE Electronic Imaging: Document Recognition and Retrieval*, 6815:81510–81510, 2008.

[4] F. Shafait, D. Keysers, and T. M. Breuel. Response to projection methods require black border removal. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(4):763–764, 2009.

[5] H. S. Baird. Background structure in document images. In H. Bunke, P. Wang, and H. S. Baird, editors, *Document Image Analysis*, pages 17–34. World Scientific, Singapore, 1994.

[6] T. M. Breuel. Two geometric algorithms for layout analysis. In *Proc. Document Analysis Systems*, volume 2423 of *Lecture Notes in Computer Science*, pages 188–199, Princeton, NY, USA, Aug. 2002.

[7] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993.

[8] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(6):941–954, 2008.

[9] D. X. Le, G. R. Thoma, and H. Wechsler. Automated borders detection and adaptive segmentation for binary document images. In *13th Int. Conf. on Pattern Recognition*, pages 737–741, Vienna, Austria, Aug. 1996.

[10] B. T. Avila and R. D. Lins. Efficient removal of noisy borders from monochromatic documents. In *Int. Conf. on Image Analysis and Recognition*, pages 249–256, Porto, Portugal, Sep. 2004.

[11] K. C. Fan, Y. K. Wang, and T. R. Lay. Marginal noise removal of document images. *Pattern Recognition*, 35(11):2593–2611, 2002.

[12] L. Cinque, S. Levialdi, L. Lombardi, and S. Tanimoto. Segmentation of page images having artifacts of photocopying and scanning. *Pattern Recognition*, 35(5):1167–1177, 2002.

[13] W. Peerawit and A. Kawtrakul. Marginal noise removal from document images using edge density. In *4th Information and Computer Engineering Postgraduate Workshop*, Phuket, Thailand, Jan. 2004.

[14] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel. Page frame detection for marginal noise removal from scanned documents. In *SCIA 2007, Image Analysis, Proceedings*, volume 4522 of *Lecture Notes in Computer Science*, pages 651–660, Aalborg, Denmark, June 2007.

[15] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel. Document cleanup using page frame detection. *Int. Jour. on Document Analysis and Recognition*, 11(2):81–96, 2008.

[16] I. Guyon, R. M. Haralick, J. J. Hull, and I. T. Phillips. Data sets for OCR and document image understanding research. In H. Bunke and P. Wang, editors, *Handbook of character recognition and document image analysis*, pages 779–799. World Scientific, Singapore, 1997.

[17] http://unpaper.berlios.de/.