The Effect of Border Noise on the Performance of Projection-Based Page Segmentation Methods

Faisal Shafait and Thomas M. Breuel

Abstract—Projection methods have been used in the analysis of bitonal document images for different tasks such as page segmentation and skew correction for more than two decades. However, these algorithms are sensitive to the presence of border noise in document images. Border noise can appear along the page border due to scanning or photocopying. Over the years, several page segmentation algorithms have been proposed in the literature. Some of these algorithms have come into widespread use due to their high accuracy and robustness with respect to border noise. This paper addresses two important questions in this context: 1) Can existing border noise removal algorithms clean up document images to a degree required by projection methods to achieve competitive performance? 2) Can projection methods reach the performance of other state-of-the-art page segmentation algorithms (e.g., Docstrum or Voronoi) for documents where border noise has successfully been removed? We perform extensive experiments on the University of Washington (UW-III) data set with six border noise removal methods. Our results show that although projection methods can achieve the accuracy of other state-of-the-art algorithms on the cleaned document images, existing border noise removal techniques cannot clean up documents captured under a variety of scanning conditions to the degree required to achieve that accuracy.

Index Terms—Document page segmentation, OCR, performance evaluation, border noise removal, document cleanup.

1 Introduction

THE goal of document image analysis is to convert a scanned document image into an editable electronic representation. One of its key steps is to locate the position of text lines or zones in the page image. This task is achieved by page segmentation, for which several algorithms have been proposed in the literature over the years [1], [2]. These algorithms can be categorized into two broad classes based on their ability to handle border noise along the page boundary: those that are sensitive to border noise and those that are robust against the presence of border noise. One of the pioneering algorithms that is still widely used for page segmentation is the X-Y Cut algorithm by Nagy et al. [3], [4], [5]. This algorithm cannot handle border noise and belongs to the first category. Representative algorithms for the second category are the Docstrum algorithm by O'Gorman [6], the Voronoi algorithm by Kise et al. [7], and the constrained text-line finding algorithm by Breuel [8].

A recent comparison of page segmentation algorithms [9], [10] on skew-corrected Manhattan layout documents from the University of Washington (UW-III) data set [11] showed that the latter category of algorithms [6], [7], [8] showed better performance than the former one [3], [12]. This might be attributed to border noise. The document images in the UW-III data set contain border noise that varies in amount, size, and shape across all images in the data set. These results raise two important questions [13], [14] related to the performance of the X-Y Cut algorithm in particular:

1. How much gain in the performance of the X-Y Cut algorithm can be achieved if document images are preprocessed with a state-of-the-art border noise removal algorithm?
2. Is the lower performance of the X-Y Cut algorithm due to border noise, or is the algorithm still outperformed by other state-of-the-art algorithms when border noise is removed in a preprocessing step?

In this paper, we find answers to these questions by extensive experiments on the UW-III data set cleaned with different border noise removal algorithms.

The X-Y Cut algorithm is based on a recursive analysis of the projection profile of a document image. The original approach in [3], [4], [5] proceeds by computing horizontal and vertical projection profiles of a scanned page image, which are obtained simply by counting the number of black (foreground) pixels in each row/column. The projection profile is then thresholded to obtain a binary string. These strings are analyzed using a block grammar to identify locations where the string can be subdivided into two strings, corresponding to a segmentation of the page image in the horizontal or vertical direction. This process is recursively applied to the blocks obtained by the segmentation to obtain a final segmented page. This algorithm is called X-Y Cut [1], [9], [10], [15], [16], [17] due to its ability to "cut" a page in the X and Y directions.

Different modifications of the original X-Y Cut algorithm have been proposed. Ha et al. [15] presented a variation based on projection profile analysis of bounding boxes, i.e., the smallest rectangular boxes which circumscribe connected components. The technique was applied to segment words, text lines, and paragraphs from a document image. Sylwester and Seth [16] presented a trainable algorithm for column segmentation from technical journals using projection profile analysis in horizontal and vertical directions. Their algorithm produces an X-Y tree representing the columnar structure of a page in a single pass through the binary image.

Projection profile-based techniques are also widely used for other preprocessing tasks like skew correction. The key idea in these approaches is to compute the projection profile along each candidate skew angle, and then select the skew angle that maximizes a given objective function. Baird [18] used the midpoint of the bottom side of the bounding boxes of connected components to compute the projection along the candidate skew angles. A similar approach was used by Kanai and Bagdanov [19], [20] to detect the skew of compressed document images directly. The points to be projected are selected from runs of black pixels that have no neighboring black pixel in the lower row. The rightmost pixel of such a black run is chosen as the point to be projected.

Despite the widespread use of projection methods, these techniques share some common limitations when used for page segmentation [1].

1. Projection methods can only segment pages that have a Manhattan layout (e.g., those of typical technical journals and books). However, pages having more complex layouts like newspapers and magazines cannot be segmented with projection methods.
2. If skew correction is not performed as a preprocessing step, projection-based page segmentation fails on document images that have a significant amount of skew.
3. Projection methods are sensitive to the presence of border noise and require black border removal for achieving good performance [13], [14].

- *F. Shafait is with the Multimedia Analysis and Data Mining (MADM) Competence Center, German Research Center for Artificial Intelligence (DFKI GmbH), D-67663 Kaiserslautern, Germany. E-mail: faisal.shafait@dfki.de.*
- *T.M. Breuel is with the Computer Science Department, Technical University of Kaiserslautern, D-67663 Kaiserslautern, Germany. E-mail: tmb@informatik.uni-kl.de.*

Manuscript received 7 Oct. 2009; revised 28 May 2010; accepted 3 Sept. 2010; published online 20 Oct. 2010.
Recommended for acceptance by E. Saund.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-10-0666.
Digital Object Identifier no. 10.1109/TPAMI.2010.194.

Due to these limitations, several alternative page segmentation algorithms have been proposed in the literature over the years. The most notable ones are the Docstrum algorithm by O'Gorman [6] and the Voronoi algorithm by Kise et al. [7]. These algorithms are not only capable of segmenting non-Manhattan layouts, but are also robust to the presence of both skew and border noise. However, the X-Y Cut algorithm has some unique characteristics that make it the algorithm of choice for several application scenarios where target documents have limited variability (for instance, bank statements, business letters, books, ...).

1. The algorithm is simple to understand and easy to implement. The effect of changing different parameters can be easily understood by nonexperts. Hence, engineers without an image processing or document analysis background can tune it on target documents.
2. It can segment several pages per second on modern computers, making it suitable for large volume applications like incoming mail digitization.
3. One can specify a complete grammar of splits to obtain a desired segmentation. This is very difficult to achieve with more complex algorithms.

In the light of these strengths and weaknesses, this work focuses on finding out how much gain in the performance of the X-Y Cut algorithm can be achieved by combining it with state-of-the-art border noise removal algorithms.

The rest of this paper is organized as follows: Section 2 briefly describes the X-Y Cut algorithm, its parameters, and the optimization algorithm used for parameter tuning. Section 3 outlines different methods used for border noise removal, followed by the evaluation protocol in Section 4. Experimental results are discussed in Section 5. The paper is concluded in Section 6.

2 X-Y Cut Algorithm for Page Segmentation

The X-Y Cut algorithm recursively subdivides a page image into regions by recursively analyzing its projection profile until a stopping criterion is satisfied. Since the horizontal projection is computed by projecting all the black pixels onto the y-axis, and the vertical projection is obtained by projecting all the black pixels onto the x-axis, the subscript y will denote the parameters related to the horizontal direction and x will represent those for the vertical direction.

2.1 Algorithm Description

First, the projection profile of a page image is computed in both horizontal and vertical directions. These profiles are then thresholded to convert them into binary strings by comparing them against two noise removal thresholds T_x^n and T_y^n. Then, the largest zero-valleys (consecutive runs of background pixels) in both the horizontal and vertical directions are identified. The widths of these valleys, v_x and v_y, are compared against two other thresholds T_x^c and T_y^c. If $v_x \geq T_x^c$ and $v_y \geq T_y^c$, the page is split into two zones at the center of the larger of the two valleys, i.e., either a horizontal or a vertical split is done. If only one of the valleys is larger than the corresponding threshold, the page is split in that direction. This process is recursively repeated on the zones obtained as a result of the split until no more splitting can be done, i.e., both $v_x < T_x^c$ and $v_y < T_y^c$. Note that the noise removal thresholds T_x^n and T_y^n are relative to the width and height of the page and are therefore scaled corresponding to the width and height of individual zones.

Note that the algorithm presented above is a slight modification of the original approach published in [4], [5] because it does not use document-source-specific block grammars as in [4], [5]. Yet, this modified algorithm is commonly used in practice [1], [9], [10] instead of the original version due to its ability to work on a heterogeneous collection of documents in the absence of a priori knowledge of the document structure.

The X-Y Cut algorithm is known to be quite sensitive to the values of its parameters $(T_x^n, T_y^n, T_x^c, T_y^c)$. Choosing too high cutting thresholds, T_x^c, T_y^c may result in undersegmentation, whereas choosing too low values may result in oversegmentation. Similarly, if the values of the noise removal thresholds T_x^n, T_y^n are too small, the algorithm is not able to segment a page having even small amounts of speckle noise. On the other hand, large values for noise removal thresholds may result in the removal of parts of the actual page content. Therefore, these parameters need to be tuned for the target data set to obtain the most accurate results.

2.2 Parameter Optimization

To choose the most suitable parameter values of the X-Y Cut algorithm for each target data set, we use the Nelder-Mead Simplex optimization algorithm with standard parameter values $(\alpha = 1, \beta = 0.5, \gamma = 2, \sigma = 0.5)$ as in [9]. The objective function to be minimized is the mean error rate of the algorithm on the training set. The error rate is measured as the percentage of ground-truth (GT) text lines G that are not identified correctly:

$$\rho = \frac{|C \cup S \cup M|}{|G|}, \tag{1}$$

where C, S, M denote the ordered sets of missed, split, and merged text lines, respectively. A ground-truth text line is considered missed if it does not overlap significantly with any segmented text line. A split/merge error occurs when a ground-truth/segmented text line significantly overlaps with more than one segmented/ground-truth text line. Significance is determined using two length thresholds in number of pixels. The thresholds control the tolerance level along the horizontal and vertical directions such that differences in overlap less than the threshold in that particular direction are ignored.

Note that the union operator in the numerator ensures that if a line is split and a part of it is also merged with another line, it is still counted as one error. Therefore, the error rate is in the range $[0, 1]$. This error measure is the same as used in [9], [10] for measuring text-line segmentation accuracy of different page segmentation methods.

3 Algorithms for Border Noise Removal

When a page of a book is scanned or photocopied, textual noise (text parts from the neighboring page) and nontextual noise (black bars, speckles) may appear along the page border, as shown in Fig. 1. Border noise varies from image to image depending on the scanning process, the material of the scanned page, and the preprocessing methods (e.g., binarization) used to prepare the image for page segmentation or optical character recognition. This variability in the location, size, and shape of the noise components renders removal of the border noise a challenging task. Several algorithms for border noise removal were proposed in the last few years. We selected six representative algorithms for our experiments. Different capabilities of these algorithms are summarized in Table 1, and a brief description of the main ideas of these algorithms is given in the following sections:

3.1 Projection-Based Cleanup

The algorithm in [21] identifies page borders by scanning the page with narrow rectangular windows spanning the width/height of the image. The key idea is to find black bars that usually occur along the page border in scanned books due to nonuniform illumination. The density of the black pixels in these border regions gives a clue about the end of the page content area. A

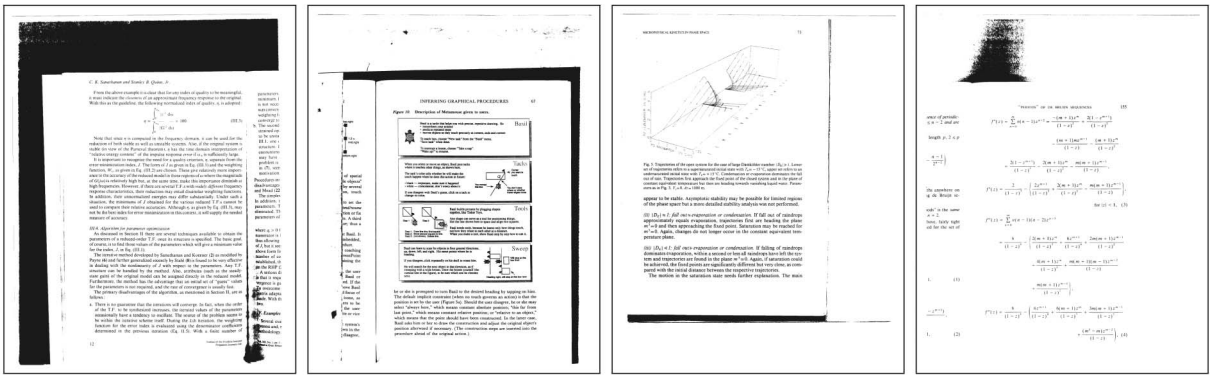

Fig. 1. Samples of document images from the UW-III data set showing the variability of border noise in the data set.

rectangular window scans the image from all four directions to identify boundaries of border noise. All of the black pixels outside this border are removed. Then, connected component analysis is performed to identify and remove large components close to the border. As a final step, the image is scanned again from all four directions to locate white regions that mark the page content boundary, and all black pixels outside this boundary are removed.

An open source implementation of this algorithm from the OCRopus OCR system [27] was used in this work.

3.2 Projection with Smearing

The method for border noise removal presented in [22] is also based on projection profile analysis for border noise detection. The algorithm first uses the run-length smearing algorithm [12] with a small threshold to smooth the image. Then, the limits of text regions (page contents) are computed by horizontal and vertical projection profile analysis of the smeared image. Finally, connected component labeling is performed. In the cleanup stage, all of the black pixels that belong to a connected component with at least one pixel lying outside the detected page content limit are transformed to white. A similar procedure is applied for textual noise detection and removal.

Stamatopoulos et al. [22] provided us with an executable of their algorithm which we used in our experiments.

3.3 Unpaper

Unpaper [23] is an open source postprocessing tool for scanned sheets of paper, especially for scans of photocopies of book pages. It tries to clean scanned images by removing dark edges resulting from scanning or photocopying. The algorithm first removes small sized isolated components by applying a "noisefilter" and a "blurfilter". Then, the border noise regions are detected by applying a "blackfilter" which scans the image with a virtual bar and fills the bar with white pixels if the amount of black pixels under the bar exceeds a predefined threshold.

We used version 0.2 of the software that comes with Ubuntu Linux 8.04 distribution. This program was run with default parameter settings in our experiments.

TABLE 1
An Overview of Capabilities of Different Border Noise Removal Algorithms w.r.t. Handling Textual Noise, Regular-Shaped Nontextual Noise (e.g., Black Bars), and Irregular-Shaped Noise Blocks that Might Appear, for Instance, Due to Torn-Off Documents

Method	Textual Noise	Non-Textual Noise Bars	Irregular-Shaped Noise Blocks
Projection [21]	YES	YES	NO
Projection with Smearing [22]	YES	YES	NO
Unpaper [23]	NO	YES	NO
Page frame detection [24]	YES	YES	YES
Edge Density [25]	YES	YES	NO
Resolution Reduction [26]	NO	YES	YES

3.4 Page Frame Detection

The method presented in [24], [28] performs document image cleanup by detecting the page frame (i.e., the actual page contents area), ignoring the margin noise along the page border. The page frame is modeled as a rectangular region tightly enclosing all page content. A geometric matching algorithm is applied to find this rectangular region by maximizing a quality function. This quality function is defined in such a way that it increases with the number of text lines touching the boundary of the rectangular region. The method works well for structured documents (journal articles, books) due to left or right aligned text in these documents. After detecting the page frame, all black pixels outside the page frame are removed to clean up the image.

We used an open source implementation of this algorithm with the default parameter setting from the OCRopus OCR system [27].

3.5 Edge Density

The method presented by Peerawit and Kawtrakul [25] detects border noise in document images by inspecting the projection of the edges instead of the page contents. The key idea behind this algorithm is that text areas have a low density of edges, while border noise areas have a high edge density. The algorithm consists of three steps. First, Sobel edge detection is performed and a vertical projection profile of the edge image is computed. Then, sharp peaks in the projection profile are detected using a so-called critical density filter. These sharp peaks correspond to the boundary between page content and border noise. Finally, border noise is discarded by a coarse-to-fine removal step.

3.6 Resolution Reduction

Fan et al. [26] presented an interesting approach for border noise removal by reducing the resolution of an image. Noise regions are detected by first removing text regions from the image using a reduction rate equal to the average size of the characters in the image. The resulting downscaled imagey consists of onl black borders and half-tones. Since these regions might be connected due to overlap between the border noise and the page contents, a block splitting step is performed to split connected components by computing their run-lengths in the reduced image. For any two neighboring runs, the shorter run is removed (resulting in a split) if the length ratio between the shorter run and the longer run is smaller than a cutting threshold. The segmented components are then classified into border noise components and nonborder noise components based on their size, position, and neighborhood. To remove noise regions, a polygonal boundary of each noise block is

established in the original image and all the foreground pixels that lie within this boundary are removed from the image.

4 Error Measures

The evaluation scheme employed in this work consists of two major parts. The first part (Section 4.1) deals with evaluating border noise removal algorithms directly. This evaluation will help in measuring the individual performance of a border noise removal algorithm. The second part (Section 4.2) presents an error measure for estimating the segmentation errors made by the X-Y Cut algorithm on target documents. The main purpose of this evaluation scheme is to identify which characteristics (see Table 1 for an overview) of the border noise removal algorithms are crucial for improving the performance of the X-Y Cut algorithm.

4.1 Evaluation of Border Noise Removal Algorithms

The goal of a border noise removal algorithm is to remove as much border noise as possible while retaining the actual content of the page image. To evaluate these aspects individually, we use the following measures:

4.1.1 Noise Ratio

In order to quantify the amount of border noise in a document image, its noise ratio is defined as in [24]:

$$\text{Noise ratio} = \frac{n_{\bar{p}}}{n_p}, \tag{2}$$

where $n_{\bar{p}}$ is the number of the foreground pixels *outside* the ground-truth page frame and n_p is the number of the foreground pixels *inside* the actual page content area of a document image. The noise ratio tells us how much border noise still remains in the document image relative to its actual contents. This measure evaluates how well the algorithm performs in removing the border noise but does not penalize removal of the actual page content.

4.1.2 Page Content Removal

To quantify removal of the actual page content by a noise removal algorithm, ground-truth removal measure is used:

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

where n_p is the total number of the foreground pixels (ground-truth) in the actual page content area of a document image and n_c is the number of foreground pixels of the actual page content that remains after noise removal.

4.2 Evaluation of the X-Y Cut Algorithm

The error rate of the X-Y Cut page segmentation algorithm is measured as the percentage of text lines that have been incorrectly segmented by the algorithm as defined in (1). Accordingly, the same error measure was used as a target function for optimizing X-Y Cut's parameters (see Section 2.2).

5 Experiments and Results

We chose the UW-III data set for our experiments, which was used in the previous work on comparative evaluation of page segmentation algorithms [9], [10], [17]. The data set contains 1,600 pages of English documents obtained from various technical journals. Due to variations in the scanning process, these document images contain a large variety of border noise [14], making it suitable for our experiments. We chose the same 978 documents from the UW-III data set as in [9], [10] for our experiments. From these documents, 100 were chosen as the training set, and 878 were chosen as the test set.

Each of the border noise removal algorithms outlined in Section 3 was used to clean up the UW-III data set. In addition, two cleaned up versions of UW-III were obtained using ground-truth information. The first version was obtained by removing all of the black pixels that were lying outside the ground-truth page frame. However, the ground-truth page frame provided with the UW-III data set does not tightly enclose the foreground regions, but includes a certain amount of white border around the ground-truth zones [24]. When the border noise is very close to the page contents area, it lies partially inside the ground-truth page frame. Therefore, some of the document images cleaned using the ground-truth page frame still contain parts of border noise. To overcome this problem, a second version of the cleaned up data set was obtained by removing all foreground pixels that were not included in any of the ground-truth zones. Since the bounding boxes of ground-truth zones tightly enclose the contents of the zone, we get better cleaned up images with this approach.

TABLE 2
Evaluation of Border Noise Removal Algorithms on 878 Images from the UW-III Test Set

Method	Noise Ratio (%)	Page Contents Removal (%)
Original (no cleanup)	96.04	0.00
Projection [21]	32.59	0.67
Projection with Smearing [22]	8.38	6.96
Unpaper [23]	10.19	8.65
Page frame detection [24]	18.14	4.66
Edge Density [25]	14.84	9.59
Resolution Reduction [26]	29.38	0.17

High noise ratio means that a significant amount of noise is still present in the images after performing cleanup, whereas high percentage of page contents removal indicates that the major parts of the page contents were also removed as border noise.

5.1 Performance of Border Noise Removal Algorithms

For the purpose of evaluating border noise removal algorithms, the UW-III test set images cleaned with ground-truth zones were used as the ground-truth images. Cleaned images of UW-III from each noise removal algorithm were evaluated against corresponding images from the ground truth. Results are shown in Table 2. A closer look at the results reveals that none of the evaluated algorithms performs uniformly better than all of the other algorithms. The Projection method and the Resolution Reduction method seem to work defensively and are able to keep most of the page content intact. However, the noise ratio of the document images cleaned with them is still high. On the other hand, Unpaper and Projection with smearing methods work aggressively and are able to remove most of the noise from the document images. However, this is accompanied by a large percentage of the actual page content also being removed as noise.

5.2 Performance of the X-Y Cut Algorithm

The next step after cleaning up the UW-III data set with different approaches is to run the X-Y Cut algorithm on the cleaned up data sets. The parameters of X-Y Cut were optimized using the Simplex optimization algorithm on the training sets. Open source implementations of the X-Y Cut algorithm and the Nelder-Mead Simplex local optimization algorithms from the PSET toolkit [29] were used in this work. Note that the objective function for the X-Y Cut segmentation algorithm as defined in (1) does not necessarily have a unique minimum. Therefore, optimization can converge to a different locally optimal solution depending on the starting point.

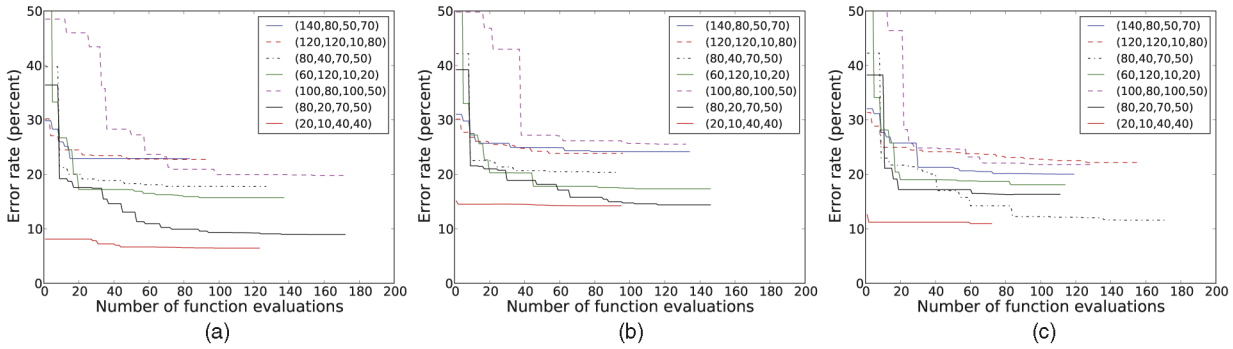

Fig. 2. Result of optimizing parameters of the X-Y Cut algorithm on the UW-III data set and its cleanup versions. Different curves correspond to different starting points (with values given in the legend) of the Simplex optimization algorithm. (a) Cleaned with ground-truth zones, (b) cleaned with Projection [21], and (c) cleaned with Unpaper [23].

To address this issue, we chose seven starting points in different regions of parameter space. Six starting points were chosen to be the same as in [9], whereas the seventh starting point was chosen based on the observation that, for cleaned documents, the optimization algorithm preferred lower values for the noise removal thresholds. The convergence curves of training for different starting points on some of the data sets are shown in Fig. 2. It can be seen that the starting point (20, 10, 40, 40) with low values of noise threshold yielded not only the best results, but also converged to this result quickly.

The optimized parameters obtained on the training set were used for evaluation on the test set using (1) as the error measure. Results of the mean training and test error rates are shown in Table 3. The running time of the X-Y Cut algorithm was less than 200 msec per page on an AMD Phenom 3.4 GHz desktop machine running Linux.

We make the following observations from these results:

1. For some algorithms, the mean error rate on the test set is lower than that on the training set. This can be explained by the fact that the cases in which the border noise overlaps with the page contents were more frequent in the training set than in the test set. Therefore, border noise removal algorithms that could not cope well with this scenario produced poorer results on the training images than those on the test images.
2. The two best performing algorithms (Unpaper and Resolution Reduction) are those that only focus on nontextual noise removal. The presence of textual noise results in a large number of false alarms produced by the X-Y Cut algorithm in those regions [10]. However, it does not influence segmentation of the actual page content, thereby not affecting the performance measured by (1). Furthermore, the Resolution Reduction algorithm performs better than the Unpaper method due to its ability to handle irregular-shaped noise regions (see Table 1).
3. The Unpaper algorithm, despite removing more than 8 percent of page contents, still leads to a low error rate for the X-Y Cut algorithm. A closer inspection revealed that in many cases, the Unpaper algorithm removed parts of several text lines in some text regions, but did not affect other text lines in the same regions. In such cases, the bounding boxes of page segments returned by the X-Y Cut algorithm mostly enclosed the removed parts of text lines as well. Therefore, these partially removed text lines were still considered as correctly segmented. Typically, the results of the segmentation algorithm are applied directly on the original image; therefore this problem would not lead to additional errors in practice.
4. A major flaw of the Edge Density method was revealed during the course of evaluation. The ruling lines found in tables or figures produce sharp peaks in the projection profiles of the edge image. Hence, they are often mistaken as the page border. If such lines are indented w.r.t. the main body of the text, all of the text lines in that image are cut at that position, resulting in high segmentation errors. In fact, the segmentation errors produced after using this cleanup method are higher than those on the original uncleaned images.
5. Although the performance of the X-Y Cut algorithm improves when used with a border noise removal algorithm, it still does not come close to the performance achieved using ground-truth information for noise removal. Reliably removing border noise is a hard problem since border noise varies in shape, size, quantity, and distance from page contents. This observation also shows that black border removal is not "simple" as suggested in [13] but rather supports the claims in [14]: "a growing literature on marginal noise removal [22], [24], [25], [26], [28], [30], [31] suggests that marginal noise removal remains a difficult problem."
6. For documents cleaned using ground-truth information, the X-Y Cut algorithm achieves very good performance

TABLE 3
Optimized Parameter Values and the Corresponding Error Rate of the X-Y Cut Algorithm on the Training and Test Sets from UW-III Data Set

Cleanup Method	Optimized Parameters	Mean Error Rate	
	$(T_x^n, T_y^n, T_x^c, T_y^c)$	Train	Test
No cleanup	(50, 10, 34, 42)	13.6	16.6
Ground-truth Pageframe	(21, 4, 41, 33)	9.1	8.4
Ground-truth Zones	(11, 3, 39, 34)	6.5	7.5
Projection [21]	(37, 4, 40, 34)	14.2	13.1
Projection with Smearing [22]	(34, 3, 42, 23)	14.9	15.8
Unpaper [23]	(40, 10, 39, 40)	11.0	11.6
Pageframe Detection [24]	(33, 7, 39, 37)	14.1	12.2
Edge Density [25]	(30, 6, 34, 38)	18.4	23.6
Resolution Reduction [26]	(37, 9, 34, 39)	8.9	11.0

Each row shows results of the X-Y Cut algorithm when the original data set was cleaned with a particular border noise removal algorithm. Note that the accuracy of the X-Y Cut algorithm obtained when used in combination with any of the evaluated border noise removal algorithms is much lower than that obtained using the ground-truth zones.

TABLE 4
Mean Text Line Detection Error Rate of the X-Y Cut Algorithm on UW-III Cleaned Using Ground-Truth Zones and with the Best Performing Noise Removal Method, Compared with that of Other State-of-the-Art Algorithms on UW-III without any Preprocessing of the Images

Page Segmentation Algorithm	Noise Removal Algorithm	Mean Error Rate	
		Train	Test
X-Y Cut	None	14.7	17.1
X-Y Cut	Resolution Reduction	8.9	11.0
X-Y Cut	Ground-truth Zones	6.5	7.5
Docstrum	None	4.3	6.0
Voronoi	None	4.7	5.5

The Results of Docstrum and Voronoi Algorithms are Taken from [10].

which is close to that of other state-of-the-art algorithms as shown in Table 4. This result supports the recommendation in [10]: "For clean documents with little or no skew, the X-Y Cut algorithm might be a good choice as it is fast and easy to implement."

Note that the error rates for Docstrum [6] and Voronoi [7] algorithms are obtained on original UW-III data set. However, since these methods are known to be robust to border noise, their performance is not expected to improve much when evaluated on cleaned documents.

6 Conclusion

This paper examined the effect of border noise removal on the performance of the X-Y Cut algorithm for page segmentation. The UW-III data set was chosen for experiments since it has page images containing a wide variety of border noise. Experimental results showed that for perfectly cleaned documents using ground-truth zone information, the X-Y Cut algorithm achieves the performance of other state-of-the-art algorithms on Manhattan layouts. However, current methods for border noise removal do not achieve the accuracy required by the X-Y Cut algorithm for competitive performance. Hence, reliable removal of border noise remains a difficult problem and further research is needed for better cleanup of documents captured under a wide variety of scanning conditions.

Acknowledgments

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

References

[1] R. Cattoni, T. Coianiz, S. Messelodi, and C.M. Modena, "Geometric Layout Analysis Techniques for Document Image Understanding: A Review," Technical Report 9703-09, http://citeseer.nj.nec.com/, 1998.
[2] G. Nagy, "Twenty Years of Document Image Analysis in PAMI," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 38-62, Jan. 2000.
[3] G. Nagy and S. Seth, "Hierarchical Representation of Optically Scanned Documents," *Proc. Seventh Int'l Conf. Pattern Recognition,* pp. 347-349, July 1984.
[4] G. Nagy, S. Seth, and M. Viswanathan, "A Prototype Document Image Analysis System for Technical Journals," *Computer,* vol. 25, no. 7, pp. 10-22, July 1992.
[5] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic Segmentation and Labeling of Digitized Pages from Technical Journals," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 7, pp. 737-747, July 1993.
[6] L. O'Gorman, "The Document Spectrum for Page Layout Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 11, pp. 1162-1173, Nov. 1993.
[7] K. Kise, A. Sato, and M. Iwata, "Segmentation of Page Images Using the Area Voronoi Diagram," *Computer Vision and Image Understanding,* vol. 70, no. 3, pp. 370-382, 1998.
[8] T.M. Breuel, "Two Geometric Algorithms for Layout Analysis," *Proc. Workshop Document Analysis Systems,* pp. 188-199, Aug. 2002.
[9] S. Mao and T. Kanungo, "Empirical Performance Evaluation Methodology and Its Application to Page Segmentation Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 3, pp. 242-256, Mar. 2001.
[10] F. Shafait, D. Keysers, and T.M. Breuel, "Performance Evaluation and Benchmarking of Six Page Segmentation Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 6, pp. 941-954, June 2008.
[11] I. Guyon, R.M. Haralick, J.J. Hull, and I.T. Phillips, "Data Sets for OCR and Document Image Understanding Research," *Handbook of Character Recognition and Document Image Analysis,* H. Bunke and P. Wang, eds., pp. 779-799, World Scientific, 1997.
[12] K.Y. Wong, R.G. Casey, and F.M. Wahl, "Document Analysis System," *IBM J. Research and Development,* vol. 26, no. 6, pp. 647-656, 1982.
[13] G. Nagy, S.C. Seth, and M. Viswanathan, "Projection Methods Require Black Border Removal," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 4, p. 762, Apr. 2009.
[14] F. Shafait, D. Keysers, and T.M. Breuel, "Response to Projection Methods Require Black Border Removal," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 4, pp. 763-764, Apr. 2009.
[15] J. Ha, R. Haralick, and I. Phillips, "Document Page Decomposition by the Bounding-Box Projection Technique," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 1119-1122, Aug. 1995.
[16] D. Sylwester and S. Seth, "A Trainable, Single-Pass Algorithm for Column Segmentation," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 615-618, Aug. 1995.
[17] F. Shafait, D. Keysers, and T.M. Breuel, "Pixel-Accurate Representation and Evaluation of Page Segmentation in Document Images," *Proc. 18th Int'l Conf. Pattern Recognition,* pp. 872-875, Aug. 2006.
[18] H. Baird, "The Skew Angle of Printed Documents," *Proc. 40th Ann. Conf. and Symp. Hybrid Imaging Systems,* pp. 21-24, May 1987.
[19] A.D. Bagdanov and J. Kanai, "Projection Profile Based Skew Estimation Algorithm for JBIG Compressed Images," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 401-405, Aug. 1997.
[20] J. Kanai and A.D. Bagdanov, "Projection Profile Based Skew Estimation Algorithm for JBIG Compressed Images," *Int'l J. Document Analysis and Recognition,* vol. 1, no. 1, pp. 43-51, 1998.
[21] F. Shafait and T.M. Breuel, "A Simple and Effective Approach for Border Noise Removal from Document Images," *Proc. 13th IEEE Int'l Multi-Topic Conf.,* Dec. 2009.
[22] N. Stamatopoulos, B. Gatos, and A. Kesidis, "Automatic Borders Detection of Camera Document Images," *Proc. Second Int'l Workshop Camera-Based Document Analysis and Recognition,* pp. 71-78, Sept. 2007.
[23] http://unpaper.berlios.de/, 2010.
[24] F. Shafait, J. van Beusekom, D. Keysers, and T.M. Breuel, "Document Cleanup Using Page Frame Detection," *Int'l J. Document Analysis and Recognition,* vol. 11, no. 2, pp. 81-96, 2008.
[25] W. Peerawit and A. Kawtrakul, "Marginal Noise Removal from Document Images Using Edge Density," *Proc. Fourth Information and Computer Eng. Postgraduate Workshop,* Jan. 2004.
[26] K.C. Fan, Y.K. Wang, and T.R. Lay, "Marginal Noise Removal of Document Images," *Pattern Recognition,* vol. 35, no. 11, pp. 2593-2611, 2002.
[27] T.M. Breuel, "The OCRopus Open Source OCR System," *Proc. SPIE Document Recognition and Retrieval XV,* pp. 0F1-0F15, Jan. 2008.
[28] F. Shafait, J. van Beusekom, D. Keysers, and T.M. Breuel, "Page Frame Detection for Marginal Noise Removal from Scanned Documents," *Proc. Scandinavian Conf. Image Analysis,* pp. 651-660, June 2007.
[29] S. Mao and T. Kanungo, "Software Architecture of PSET: A Page Segmentation Evaluation Toolkit," *Int'l J. Document Analysis and Recognition,* vol. 4, no. 3, pp. 205-217, 2002.
[30] L. Cinque, S. Levialdi, L. Lombardi, and S. Tanimoto, "Segmentation of Page Images Having Artifacts of Photocopying and Scanning," *Pattern Recognition,* vol. 35, no. 5, pp. 1167-1177, 2002.
[31] B.T. Avila and R.D. Lins, "Efficient Removal of Noisy Borders from Monochromatic Documents," *Proc. Int'l Conf. Image Analysis and Recognition,* pp. 249-256, Sept. 2004.