Stroke-like Pattern Noise Removal in Binary Document Images

Mudit Agrawal and David Doermann
Institute of Advanced Computer Studies
University of Maryland
College Park, MD, USA
{*mudit, doermann*}*@umd.edu*

Abstract—**This paper presents a two-phased stroke-like pattern noise (SPN) removal algorithm for binary document images. The proposed approach aims at understanding script-independent prominent text component features using supervised classification as a first step. It then uses their cohesiveness and stroke-width properties to filter and associate smaller text components with them using an unsupervised classification technique. In order to perform text extraction, and hence noise removal, at diacritic-level, this divide-and-conquer technique does not assume the availability of accurate and large amounts of ground-truth data at component-level for training purposes.**

The method was tested on a collection of degraded and noisy, machine-printed and handwritten binary Arabic text documents. Results show pixel-level precision and recall of 86% and 90% respectively for noise-pixels.

Keywords-**speckle removal; noise; degraded ruled-line removal; salt-n-pepper; stroke-like pattern noise; low-density languages**

I. Introduction

Observed images often deviate from the ideal images that were produced by the source. These deviations may manifest themselves on the physical document in or after production or during scanning, transmission, storage or conversion from one form to another and are collectively refered to as noise. Irrelevant content, such as rule-lines, can also be viewed as noise, making the problem of detection and removal very much application dependent.

Document analysis algorithms such as page segmentation and character recognition typically work best on clean documents and often rely on connected components as basic units, which unfortunately are sensitive to various types of noise.

Types of Noise:: Document noise such as rule-lines [1], [2], bleed-through [3], stray-marks or clutter [4], [5] may be present before the scanning process, while many other types of document noise are introduced at later stages.

Clutter noise [5], [6] may also appear during the scanning process, due to the improper alignment of the document paper with the scanner bed or due to generic thresholding applied after the scanning process. Similarly, bleed-through can also appear during the scanning of thin paper and as a result of light reflecting off the scanner's backing.

Salt-n-pepper has been one of the most prevalent kind of noise in document images. Also known as bipolar noise, it is an impulsive noise which appears as randomly distributed small components over an image formed due to dithering binarization [7] and can be composed of one or more pixels. However, by definition, they have been assumed to be much smaller than the size of wanted content and, therefore, the most prominent techniques for removing salt-n-pepper noise use a small median filter [7], [8], kFill window [9] or a morphological operator of size 3X3 or smaller [10].

In this paper, we will consider noise types in binary documents which are of magnitude (size) similar to that of text-diacritics and tend to directly affect text in the foreground in irregular ways, as shown in Figure 1. We call such noise as Stroke-like Pattern Noise (SPN) [11]. The challenge is to detach and preserve text components (consonants and diacritics) and eventually remove noise from the document.

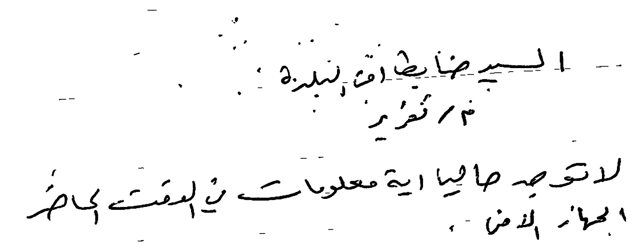
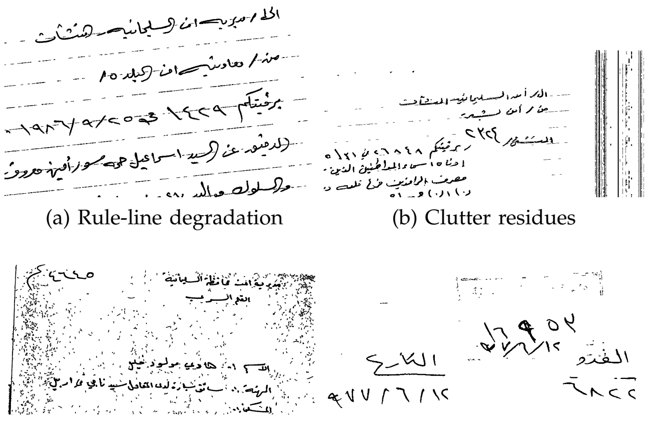


Figure 1: Stroke-like Pattern Noise, resembling diacritics, present around text components

In this paper we present a script-independent two-phased content based technique to clean stroke-like pattern noise from binary handwritten or printed documents using a minimal set of training samples. This paper is organized as follows. Section 2 gives an overview of the SPN and its removal challenges.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.13


(a) Rule-line degradation (b) Clutter residues

(c) Marks (d) Degraded Background

Figure 2: Examples of Stroke-like Pattern Noise

Section 3 outlines the proposed content understanding approach and describes its two phases. This is followed by evaluation in Section 4 and conclusion in Section 5.

II. SPN Definition and Removal Challenges

Definition:: SPN is independent noise with respect to the content [5]. In general, it is independent of location, size or other properties of text data in the document image. Recorded images having this type of noise, can be expressed as the sum of true image $I(i, j)$ and the noise $N(i, j)$ as $R(i, j) = I(i, j) + N(i, j)$. In spite of being independent, due to its similarity to diacritics, its presence near textual components can change the meaning of a word, especially in Arabic documents.

This noise is formed primarily due to the degradation of underlying page rule lines that interfere with the foreground text. These degraded rule lines are severely broken, not straight and interact significantly with text. (Figure 2a). Another major source of formation are the blurring edges of clutter noise [4], [5] which remain after clutter removal approaches (Figure 2b). Stray marks in handwritten documents, some highly degraded and unperceivable background content can be other sources of such noise, as shown in Figures 2c, 2d.

Removal Challenges:: As degraded rule-lines, these line components are broken and degraded to a degree that they cannot be perceived in straight lines even by the human eye. This makes techniques like Hough transform, projection profiles not suited for their removal. Their shape and size similarity to smaller text components, prohibits morphological processing based removal approaches because the successive erosion and dilation steps needed, tend to degrade text. Their similar spatial frequency to text renders median filtering approaches ineffective. With tremendous amount of research being done for salt-n-pepper noise and rule-line removal, this type of noise has thus been neglected as either aberrations or too degraded to model.

Document cleaning can be performed in two fundamental ways. One approach is to detect and remove noise from an image, and the other approach is to extract the information content from an image, leaving non-content as noise behind. The former approach is prefered when the noise can be differentiated from text using its independent set of features. For example, clutter [4], [5], rule-lines [12], [13], salt-n-pepper noise [9], [14], [15] and marginal noise [6] exhibit properties quite different from the textual content. On the other hand, noise like SPN cannot be removed without apriori knowledge of the textual content. This leads to the latter approach which aims at understanding content.

There has been a lot of work on extracting text components from a document image. However, the majority of the work has been focused on extracting text from colored documents or from background patterns. Using depth as an added dimension, all these algorithms benefit from gray-scale or color histogram analysis in order to differentiate text from background patterns [16], [17]. There has not been much work in differentiating handwritten (or printed) text in binary document images from stroke-like pattern noise (SPN).

Classifying all the text components and SPN in one step using a binary classifier entails using an extensive set of features capturing both shape and context information at component-level. Apart from generating a detailed feature-set, this approach suffers from script-specific associations of smaller text components to the bigger ones. In order to cover all the recognizable units, across scripts, systems need a much larger training set. Limited amounts of annotated data at component or pixel level for many low density languages and complex interaction between strokes prompt for new ways to bootstrap systems to perform similar tasks.

III. Noise Removal using Content Understanding Technique

Intuitively, text has the following distinguishing characteristics: 1) text possesses certain frequency and orientation information; 2) text shows *spatial cohesion* - a set of strokes appear together to form words or phrases [16]. At component-level, many of these stroke components, in cohesion, contain prominent textual features like length, critical points, cusps, arcs and curves. Such text components with independent features are called prominent text components (PTC). PTCs can be identified as text components individually and do not require any neighboring context. However, many smaller components, like diacritics, use their positions and stroke-widths with respect to PTCs to identify themselves as textual content. These two properties of the smaller text components (called non-PTCs) are tightly coupled with the PTCs (Figure 3).

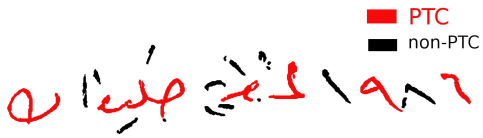


Figure 3: Red (dark gray) and black components depict PTCs and non-PTCs respectively

We use the above listed text properties to devise a two-phased component-based divide-and-conquer approach to extract text components from a noisy binary document image using a minimal set of training samples. In the first phase, we classify prominent text components (PTCs) using a supervised classification approach. Aiming at the script-independent features of text strokes, a generalized feature set is devised to classify the PTCs using a limited training dataset. Later, based on the stroke-width and cohesiveness properties of these components, non-PTCs are filtered out from the noise components using unsupervised k-means clustering.

A. *Supervised Prominent Text Component Classification*

Prominent text components exhibit script-independent and context-independent properties to distinguish themselves from other types of content in a binary image. Apart from area, perimeter, convex-area of each component, orientation of the fitted ellipse, it's major and minor axis lengths and eccentricity, four more feature descriptors are defined as follows in order to measure the independent shape properties [18].

1) FilledArea: Number of foreground pixels in the bounding box of the component with all holes filled in
2) Extent: Ratio of the pixels in the component to the pixels in the bounding box
3) Solidity: Ratio of the pixels in the smallest convex polygon that are also in the component ($=Area/ConvexArea$)
4) EquivDiameter: Diameter of the circle with the same area as the region ($=sqrt(4*Area/pi)$)

These features are normalized by the average size of the connected components and scaled to the range $[0,1]$. The components are labeled as PTCs and others (includes non-PTCs and noise) on a limited set of training samples, and sent to the feature extraction module. LibSvm library [19], is then used to classify the two set of classes. A selective number of features used over a large number of components ($|features| \ll |instances|$) implied using an RBF Kernel for classification in order to nonlinearly map data to a higher dimensional space.

After classification, the results are sent to the second-phase to selectively remove noisy components from the image.

B. *Unsupervised Small-Component Classification*

In order to filter non-PTCs from a pool of non-PTCs and SPN, we compute two characteristics of all components - their stroke-width and cohesiveness with respect to PTCs. These are computed efficiently using a distance transform approach [5]. The distance transform labels each pixel of the image with the distance to the nearest pixel of different gray-value. For a binary image, foreground distance transform, D_I, labels each pixel with its nearest distance to the background pixel, thus producing a distance map with increasing distances from the edge of each component to it's center. Similarly, $D_{I'}$ is defined as the background distance transform of image I, where background pixels are labeled by their distance to the closest foreground boundary and all foreground pixels are labeled 0. The distance transform can be computed efficiently with a two pass algorithm presented in [20].

1) Stroke-width: In order to compute this efficiently, we perform a foreground distance transform. Maximum distance value associated with each connected component (CC) defines its stroke-width (sw_{CC}).

$$sw_{CC} = \max(D_I(p)),\ \forall\ p\ \in\ \{CC\} \tag{1}$$

Mode (highest frequency) of stroke-widths for PTCs gives the average stroke-width of the document sw_{avg}.

2) Cohesiveness: First, an image with only PTCs is created (I_{PTC}). Performing a background distance transform on that image ($D_{I'_{PTC}}$) assigns each background pixel a minimum distance to the nearest PTC. Cohesiveness (co_{CC}) for each non-PTC is then defined as the minimum distance value associated with the underlaid background pixels.

$$co_{CC} = \min(D_{I'_{PTC}}(p)),\ \forall\ p\ \in\ \{CC\} \tag{2}$$

Average distance between each nearest pair of PTC (co_{avg}) is calculated using a distance adjacency matrix.

Figure 4b shows the classified non-PTCs and noise components overlaid the distance transform map of PTCs for our test image in Figure 4a. K-means clustering (k = 2) is applied based on the defined features

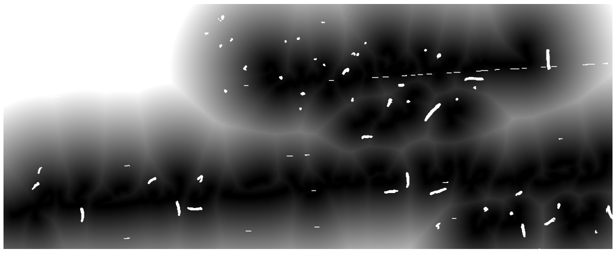


Figure 4: Image shows classified non-PTCs and noise components overlaid the distance transform map of PTCs. Components nearer the darker regions are closer to the PTCs and vice versa

($|features| = 2$). A further verification step is performed with the following rule:

$$if\ sw_{CC} >= sw_{avg}\ \&\ co_{CC} <= cc_{avg},$$
$$classify\ CC\ as\ text - component$$

The non-PTCs are filtered out leaving the noise components behind. The final result after the second phase is shown in Figure 5.

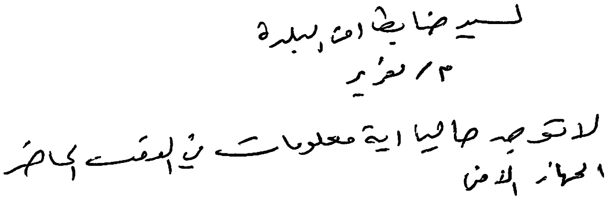


Figure 5: SPN removal result of the test image in Figure 1. The noise components are successfully removed

IV. Evaluation

A. Datasets

The dataset consists of printed and handwritten Arabic binary documents. Manual ground-truthing being a laborious job at the pixel-level, we use a representative set of 50 document images containing Stroke-like Pattern Noise (SPN) from the four sources described in Figure 2. Only 2 document images are used to train the SVM for PTC classification to validate our minimal training requirement.

B. Metrics

Pixel-based evaluations are performed in order to assess the accuracy of our approach. SPN being of the size similar to that of smaller text components, and PTCs being much bigger in size, SPN occupies 16% of the pixels in the dataset. We calculate precision and recall of our SPN removal algorithm using the following metrics to evaluate the effective gain in accuracy.

$$Precision_N = \frac{Noise\ Pixels\ Removed}{Total\ Pixels\ Removed} = \frac{TN}{TN + FN}$$
$$Recall_N = \frac{Noise\ Pixels\ Removed}{Total\ Noise\ Pixels} = \frac{TN}{TN + FP}$$

C. Results

We achieved precision and recall accuracy of 86% and 90% respectively for noise-pixels (Table I). In this component-based noise removal approach, even a few misclassified text-components tend to increase the pixel-level precision error rate due to their comparitively larger component sizes than noisy ones. Using the pixel distribution in Figure 6, we also report the precision and recall accuracy for the remaining text-pixels after noise removal as 98% and 97% respectively. The results of sample documents of each type are shown in Figure 7.

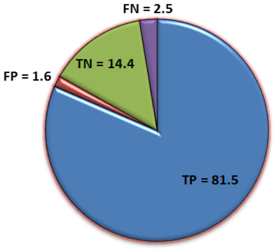


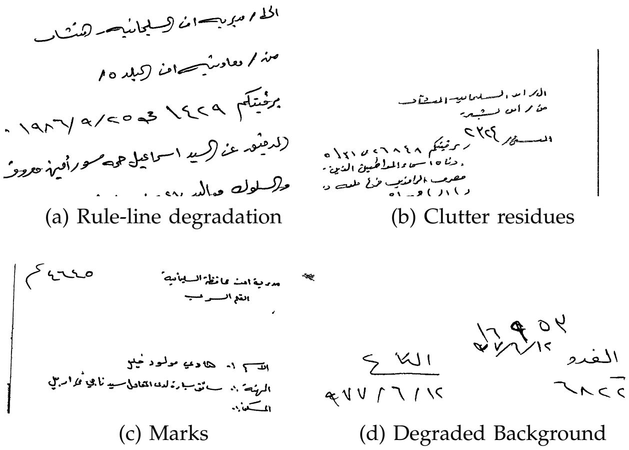
Figure 6: Pixel Distribution

	Noise
Precision	86%
Recall	90%

Table I: Accuracy

V. Conclusion

We have presented a novel approach to stroke-like pattern noise (SPN) detection and removal for binary



(a) Rule-line degradation

(b) Clutter residues

(c) Marks

(d) Degraded Background

Figure 7: Results of SPN Removal Algorithm

document images. Our two-phased approach aims at understanding the script-independent prominent text component features as the first step in a supervised classification approach. SVM with an RBF kernel is used to classify these components from the rest using a minimal set of training samples. Later, based on the cohesiveness and stroke-width features of these components, smaller text components are filtered out using k-means clustering. The novelty of this approach is that it does not aim at script or character recognition in order to perform text extraction at diacritic level. It also does not depend on a sufficient number of representative ground-truth samples at component level for training. Instead, it uses generic script features to divide-and-conquer components into prominent and dependent ones to achieve noise removal. SPN noise removal was tested on a set of Arabic machine-printed and handwritten documents and precision and recall rates of 86% and 90% respectively were reported.

We would like to extend this approach to other scripts and documents with mixed content. The idea of separating prominent text components first and using their properties to perform context analysis can be utilized in many other domains of document processing, like rule-line removal, line extraction and word segmentation.

Acknowledgment

The partial support of this research by DARPA through BBN/DARPA Award HR0011-08-C-0004 under subcontract 9500009235, the US Government through NSF Award IIS-0812111 is gratefully acknowledged.

References

[1] Z. Shi, S. Setlur, and V. Govindaraju, "Removing rule-lines from binary handwritten arabic document images using directional local profile," in *20th Int'l Conf. on Pattern Recognition (ICPR)*, 2010, pp. 1916 –1919.

[2] W. Abd-Almageed, J. Kumar, and D. Doermann, "Page rule-line removal using linear subspaces in monochromatic handwritten arabic documents," in *10th International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 768 –772.

[3] Q. Wang and C. L. Tan, "Matching of double-sided document images to remove interference," *Proc. Comp. Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I–1084–I–1089 vol.1, 2001.

[4] B. T. Ávila and R. D. Lins, "A new algorithm for removing noisy borders from monochromatic documents," in *Proc. ACM Symposium on Applied computing (SAC)*. New York, NY, USA: ACM, 2004, pp. 1219–1225.

[5] M. Agrawal and D. Doermann, "Clutter noise removal in binary document images," in *10th Int'l Conf. on Document Analysis and Recognition (ICDAR)*, 2009, pp. 556 –560.

[6] K.-C. Fan, Y.-K. Wang, and T.-R. Lay, "Marginal noise removal of document images," *Proc. 6th Int'l Conf. Document Analysis and Recognition (ICDAR)*, pp. 317–321, 2001.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Addison-Wesley, 2008.

[8] L. O'Gorman and R. Kasturi, *Document Image Analysis*. Computer Society Press, 1995.

[9] N. Premchaiswadi, S. Yimgnagm, and W. Premchaiswadi, "A scheme for salt and pepper noise reduction and its application for ocr systems," *W. Trans. on Comp.*, vol. 9, pp. 351–360, April 2010.

[10] J. Serra, *Image Analysis and Mathematical Morphology*, 3rd ed. Academic Press, 1983.

[11] Y. Liu and S. Srihari, "Document image binarization based on texture features," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 540 –544, May 1997.

[12] Y. Zheng, H. Li, and D. Doermann, "A model-based line detection algorithm in documents," *Proc. 7th Int'l Conf. on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 44–48, Aug 2003.

[13] Y. Zheng, C. Liu, X. Ding, and S. Pan, "Form frame line detection with directional single-connected chain," *Proc. 6th Int'l Conf. Document Analysis and Recognition (ICDAR)*, pp. 699–703, 2001.

[14] M. Ali, "Background noise detection and cleaning in document images," *Proc. 13th Int'l Conf. Pattern Recognition, (ICPR)*, vol. 3, pp. 758–762 vol.3, Aug 1996.

[15] K. Chinnasarn, Y. Rangsanseri, and P. Thitimajshima, "Removing salt-and-pepper noise in text/graphics images," *Asia-Pacific Conf. on Circuits and Systems (IEEE APCCAS)*, pp. 459–462, Nov 1998.

[16] V. Wu, R. Manmatha, and E. M. Riseman, Sr., "Textfinder: An automatic system to detect and recognize text in images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1224–1229, 1999.

[17] Y. Solihin and C. Leedham, "Integral ratio: a new class of global thresholding techniques for handwriting images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 761 –768, Aug. 1999.

[18] *Matlab Image Processing Toolbox: Connected component properties*, 2001, http://www.mathworks.com/help/toolbox/images/ref/regionprops.html.

[19] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[20] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *Journal of the Assoc. for Comp. Mach.*, vol. 13, no. 4, pp. 471–494, 1966.