Improved Document Image Segmentation Algorithm using Multiresolution Morphology

Syed Saqib Bukhari[a], Faisal Shafait[b], and Thomas M. Breuel[a]

[a] Technical University of Kaiserslautern, Kaiserslautern, Germany,
[b] German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany

ABSTRACT

Page segmentation into text and non-text elements is an essential preprocessing step before optical character recognition (OCR) operation. In case of poor segmentation, an OCR classification engine produces garbage characters due to the presence of non-text elements. This paper describes modifications to the text/non-text segmentation algorithm presented by Bloomberg,[1] which is also available in his open-source Leptonica library.[2] The modifications result in significant improvements and achieved better segmentation accuracy than the original algorithm for UW-III, UNLV, ICDAR 2009 page segmentation competition test images and circuit diagram datasets.

Keywords: Page Segmentation, Text/Non-Text Segmentation, Multiresolution Morphology

1. INTRODUCTION

Text/non-text segmentation is an important initial step in most document analysis systems. The goal of text/non-text segmentation is to separate text and non-text elements in a document image. Optical character recognition (OCR) systems are trained to recognize text elements, if a segmentation algorithm fails to correctly segment text and non-text elements, the character recognition module in a OCR system produces a lot of garbage characters because of the presence of non-text elements. Non-text elements can be composed of following classes: halftone, drawing, graphics, logo, speckle, ruling etc.

Several approaches for text/non-text segmentation have been proposed in the literature. Wong et al.[3] presented a classical page segmentation approach by smearing. Bloomberg's approach[1] separates halftone elements from document images by using multiresolution morphology. Block or zone based classification approaches[4,5] classify the blocks detected by a zone segmentation algorithm into text and non-text elements. Pixel-based classifiers[6] try to classify each pixel either as text or non-text pixel. An approach based on discriminative learning over connected components[7] is introduced recently, which tries to classify each connected component either as text or non-text component.

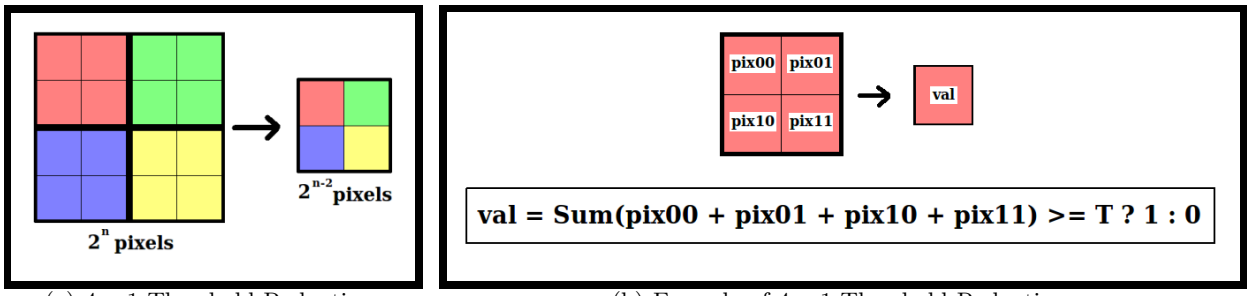
Zone based classification approaches require a preprocessing step of zone segmentation and heavily depend on the accuracy of zone segmentation results. In general, the accuracy of classification based text/non-text segmentation approaches, either pixel-based or connected component based, heavily depends on training samples. Bloomberg's multiresolution morphology based approach works on the assumption that non-text components are bigger than text components.

Bloomberg's text/non-text segmentation algorithm is easy to adapt for the task in a variety of scripting languages. It was specifically designed for separating halftones from document images. It is a simple approach and performs well for halftone mask generation, but most of the time fails to correctly segment other types of non-text elements, such as drawing, graphics, maps etc. An open source implementation is available as part of the Leptonica library.[2] Here, we present an improved version of Bloomberg's text/non-text segmentation algorithm, which can also segment drawing type non-text component in document images.

The paper is organized as follows. Section 2 describes in detail the Bloomberg's text/non-text segmentation algorithm. Section 3 explains our modifications in Bloomberg's algorithm. Section 4 deals with the experimental results and Section 5 discusses our conclusions.

bukhari@informatik.uni-kl.de,faisal.shafait@dfki.de and tmb@informatik.uni-kl.de

Document Recognition and Retrieval XVIII, edited by Gady Agam, Christian Viard-Gaudin,
Proc. of SPIE-IS&T Electronic Imaging, Vol. 7874, 78740D · © 2011 SPIE-IS&T
CCC code: 0277-786X/11/$18 · doi: 10.1117/12.873461



(a) 4×1 Threshold Reduction (b) Formula of 4×1 Threshold Reduction

Figure 1. Definition of multiresolution morphology based threshold reduction operation: (a) each 2×2 block of four pixels is subsampled to one pixel (4×1 Reduction). (b) the value of subsampled or reduced pixel is '1' if sum of the values of four binary pixels within a 2×2 block is greater than or equal to the threshold (T), otherwise '0'. The threshold can be set between one and four.

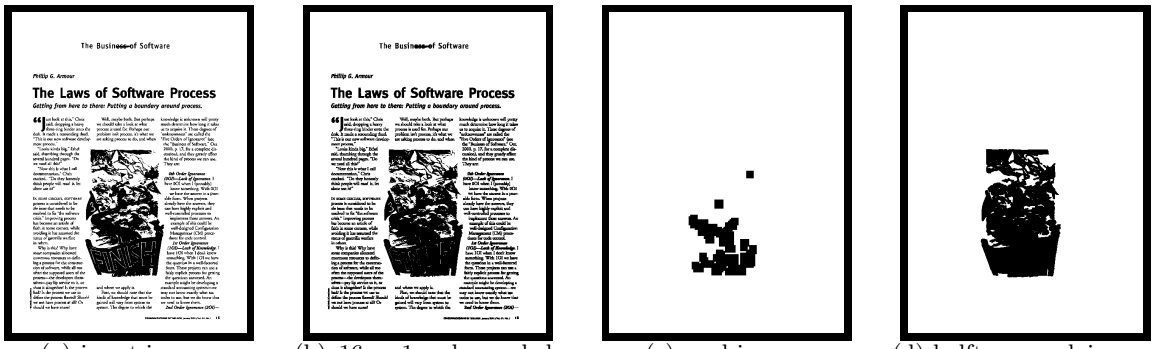
2. BLOOMBERG'S TEXT AND HALFTONE IMAGE SEGMENTATION

Bloomberg's page segmentation algorithm[1] is based on the idea of multiresolution morphology. Bloomberg first proposed the outline of a text/non-text segmentation algorithm using basic morphological operations before presenting his multiresolution morphology based algorithm, such as: i) an image can be closed with a sufficiently large structuring element intending to solidify halftone components, ii) then the image can be opened with an even larger structuring element intending to remove text blobs and to preserve some residual portions of halftone elements of the image, iii) the residual portions or seeds of halftone elements can be used for generating a halftone mask for the original image. He has highlighted the importance of the multi-scale image representation by emphasizing that it can be used for efficient analysis of image contents as well as to speed up image processing operations, like morphology. He updated the aforementioned basic outline of the text and halftone segmentation algorithm using multi-scale image representation such that: i) an image can be closed or dilated before subsampling in order to combine neighboring pixels, ii) the resulting image can be opened or eroded before further subsampling intending to preserve big size elements, like halftone. As it is expensive to use large structuring element at full or high image resolution, he introduced the concept of *"threshold reduction"* for the implementation of his subsampling based text/non-text segmentation algorithm. The threshold reduction criteria is defined as follows.

Threshold Reduction: consider a binary image where each foreground pixel is represented by '1' and each background pixel is represented by '0'. The image is tiled into 2×2 pixel blocks. Each 2×2 block of four pixels is replaced by a single pixel in a subsampled image. The value of each subsampled pixel is either '1' or '0' depending on the chosen threshold, that ranges between one and four. The subsampled pixel value is '1' if the sum of the values of four pixels is greater than or equal to the threshold, otherwise '0'. This subsampling operation with threshold equal to one mimics the dilation of an image with 2×2 structuring element followed by the subsampling of upper-left pixel of each 2×2 pixels block. Similarly, subsampling with threshold equal to four mimics the subsampling after erosion. Threshold can be set equal to two or three as well. The threshold selection criteria is also referred as *threshold convolution* or *rank order filter*. Bloomberg referred the combination of threshold convolution followed by subsampling as *threshold reduction*. After a single threshold reduction (also called 4×1 reduction) operation, the number of an image pixels is reduced from 2^n to 2^{n-2}. The concept of threshold reduction is illustrated in Figure 1. Bloomberg's text/non-text segmentation algorithm using the concept of threshold reduction operation proceeds as follows.

2.1 Algorithm

Bloomberg's text/non-text algorithm is based on the concept of threshold reduction and basic morphological operations. It also uses a 1×4 expansion operation in which each pixel is enlarged to 2×2 pixels block. Bloomberg's algorithm is described as follows. A binary document image is first processed by two threshold reduction operations with thresholds equal to one. After these two threshold reduction operations, the input image is subsampled from 2^n to 2^{n-4} pixels and at the same time the density of all foreground pixels are preserved



(a) input image (b) 16×1 subsampled image (c) seed image (d) halftone mask image

Figure 2. Snapshots of the Bloomberg's text/non-text segmentation algorithm for an example image with text and halftone components.

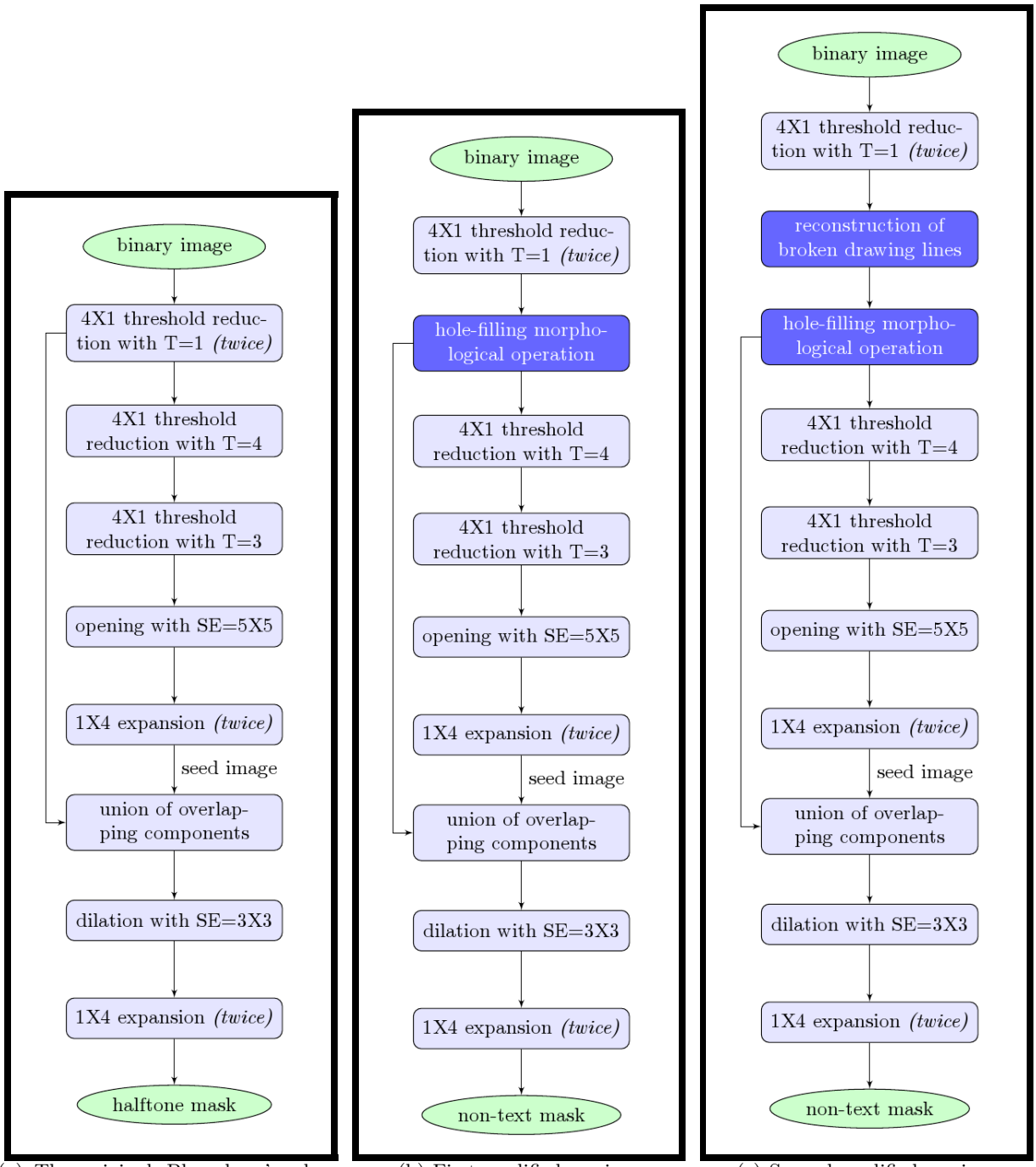
as before. This subsampled image can also be referred to as a 16×1 subsampled image, as shown in Figure 2(b). The 16×1 subsampled image is further reduced by two threshold reduction operations with thresholds equal to four and three respectively and then processed by morphological opening with 5×5 structuring element. These further threshold reduction operations with morphological opening of the 16×1 subsampled image are intended to preserve some portions of big size (halftone) components and to remove comparatively small size (text) components, as shown in Figure 2(c). The image in Figure 2(c) is called a seed image. The seed image is enlarged by using two 1×4 expansion operations, due to which its size becomes equal to the 16×1 subsampled image. Finally, a halftone mask image is generated by selecting only fully or partially overlapping components between the 16×1 subsampled image (Figure 2(b)) and the seed image (Figure 2(c)). The resulting mask image is processed by morphological dilation with 3×3 structuring element and then enlarged by using two 1×4 expansion operations, so that it becomes equal to the size of the input image. The final halftone mask image is shown in Figure 2(d). The data flow diagram of Bloomberg's text/non-text segmentation algorithm is shown in Figure 3(a).

3. MODIFICATION TO BLOOMBERG'S ALGORITHM

Bloomberg's text/non-text segmentation algorithm is specifically designed for generating the non-text halftone mask form a document image. It usually fails to segment thin-line drawing and graphics types components as non-text components from a document image. Here we first describe why Bloomberg's algorithm is unable to identify drawing type non-text elements, then we introduce some modifications to Bloomberg's text/non-text segmentation algorithm to improve its performance for drawing type non-text components.

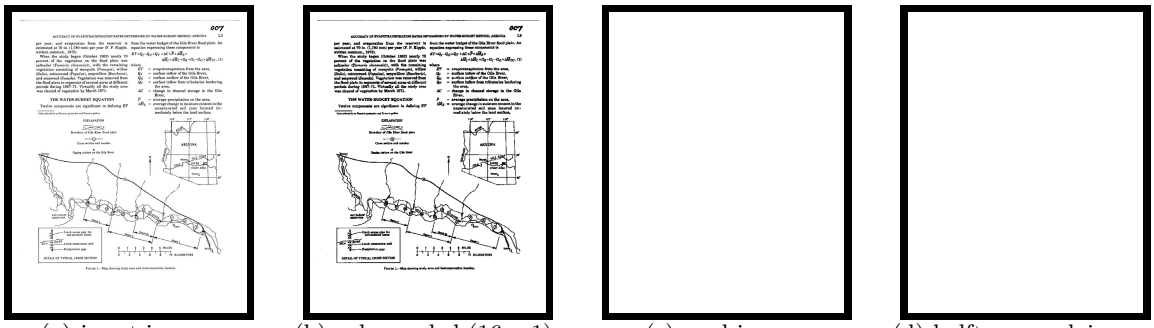
It is clear from the Bloomberg's text/non-text segmentation algorithm that, it fails to separate those non-text elements which are not preserved in the intermediate seed image. The seed image is generated by using four consecutive threshold reduction operations; first two with low threshold values and other two with high threshold values. The threshold reduction operations with high value threshold drop fine or minor foreground image details. Usually in a document image, halftone elements are composed of lots of neighboring pixels as compared to text and drawing type non-text elements. Therefore, the resulting seed image of a document image only composed of the residual portions of thick elements mostly like halftone. Due to which, Bloomberg's algorithm often unable to segment thin-line drawing and graphics elements as non-text elements which is shown in Figure 4.

First Modification - Hole-Filling Morphological Operation: we have observed that non-text components, such as drawings, maps, graphs and even halftones, are often composed of thin-line hollow contours of geometric and irregular shapes, as shown in Figure 4(a). As discussed earlier, the threshold reduction operations with high value thresholds remove these hollow contours. If thin-line hollow contours of a document image are filled before high value threshold reductions, then there are more chances of getting residual portions of different types of non-text elements in the seed image. For this purpose, a well-known *"hole-filling"* morphological operation can be used, which is briefly described here for completeness, (i) the inverted image of an input document



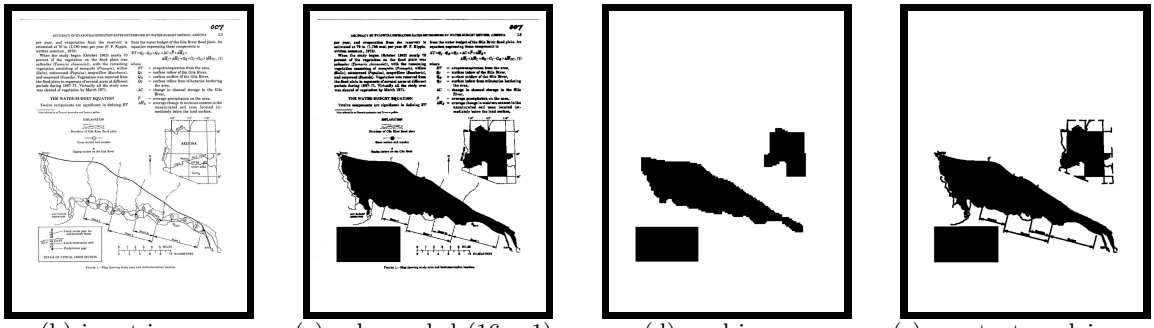
(a) The original Bloomberg's algorithm (b) First modified version (c) Second modified version

Figure 3. Data flow diagrams of the original Bloomberg's text/image segmentation algorithm and its modified versions ('T': threshold; 'SE': structuring element).



(a) input image (b) subsampled (16×1) image (c) seed image (d) halftone mask image

Figure 4. Snapshots of the Bloomberg's text/image segmentation algorithm for an example document image containing text and drawing components, where it fails to segment drawing components as non-text components.



(b) input image (c) subsampled (16×1) image followed by hole-filling operation) (d) seed image (e) non-text mask image

Figure 5. Snapshots of the first improved version of the Bloomberg's text/image segmentation algorithm. Hole-filling based improved version (Figure 3(b)) produces correct non-text mask for drawing type components as compared to the result of the original Bloomberg's algorithm shown in Figure 4(d).

image is used as a mask image, ii) a filled-image is initialized with the top-left pixel is equal to '1' and all the rest are equal to '0', iii) the filled-image is dilated using a 3×3 structuring element, iv) after dilation, all of the pixels that are '0' in the mask image are set to '0' in the filled-image, v) dilation followed by resetting of the filled-image's pixels is repeated until no more changes are made to the filled-image.

The data flow diagram of the original Bloomberg's algorithm is shown in Figure 3(a). Our first modification in the original algorithm is defined as folows: the hole-filling operation is first applied to the 16×1 subsampled image and then the resulting image is processed by further high value threshold reduction operations. A brief data flow diagram of the first modified version of Bloomberg's text/non-text segmentation algorithm is shown in Figure 3(b). For an example image, which composed of text and drawing, the text/image segmentation results of the original Bloomberg's algorithm and the first improved version are shown in Figures 4 and 5 respectively for comparison. These figures show that, unlike the original Bloomberg's algorithm the improved version correctly segment drawing type non-text components.

Second Modification - Reconstruction of Broken Drawing Lines: we have also observed that, sometimes non-text components consist of broken drawing lines either by choice or because of document digitization errors (like poor binarization because of uneven illumination). For a document image, components with closed contours of unbroken lines are filled by hole-filling operation, but components with broken lines are remained unfilled. Therefore, the first improved version of Bloomberg's algorithm still fails to segment broken-line drawing type components, as shown in the top row of Figure 6. We can further improve the segmentation accuracy of

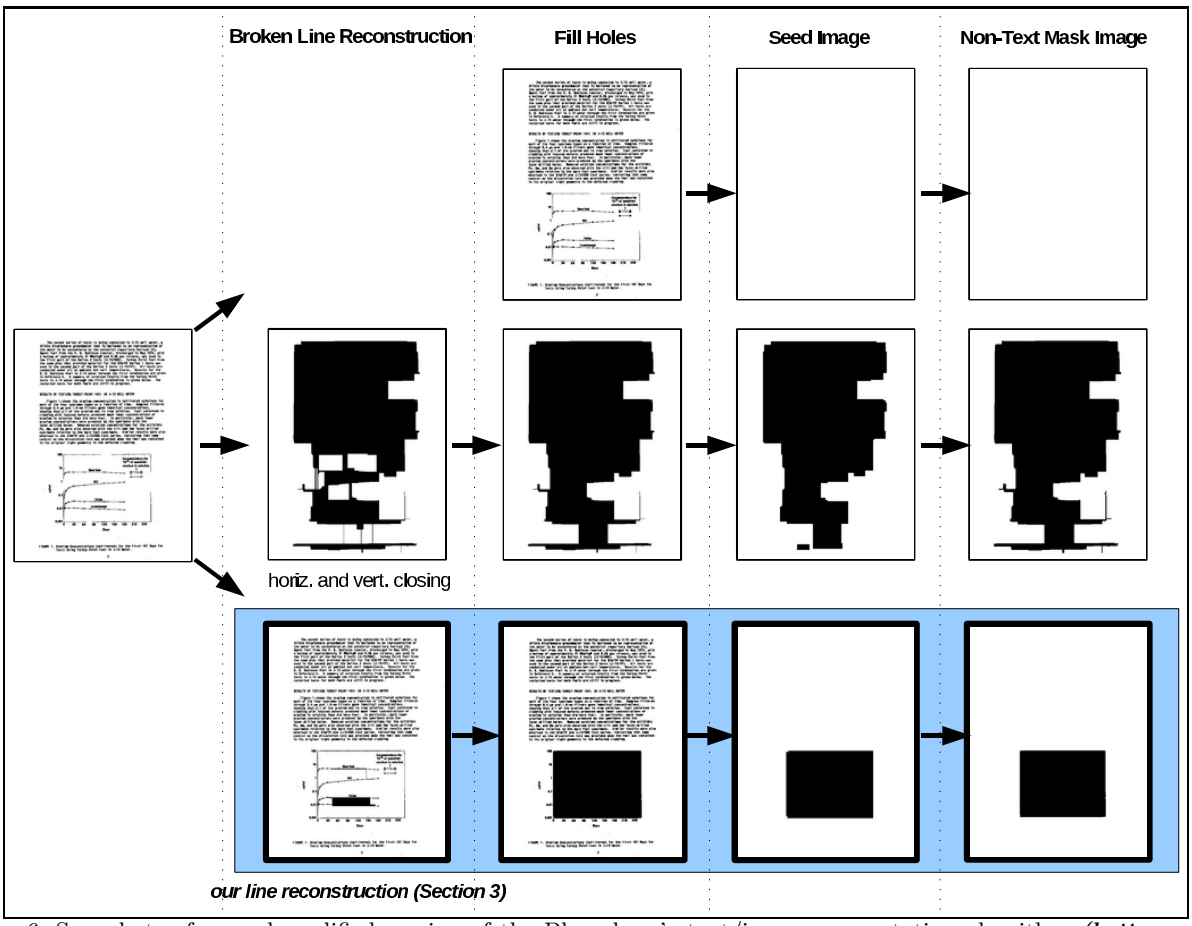


Figure 6. Snapshots of second modified version of the Bloomberg's text/image segmentation algorithm ***(bottom row)***. **Top Row:** hole-filling operation does not fill broken drawing lines component and therefore first modified version fails to segment it as non-text component. **Middle Row:** horizontal and vertical closing based drawing line reconstruction produces a garbage image and a garbage result for non-text mask. **Bottom Row:** our line reconstruction method for broken drawing lines, as described in Section 3, generates closed contour drawing shape which help in producing a correct non-text mask.

Bloomberg's algorithm for such cases by reconstructing broken drawing lines before the hole-filling operation. At first, one might consider using a morphological closing operation with oriented (like horizontal and vertical) structuring elements for drawing lines reconstruction. However, a morphological closing operation is not a suitable solution for the reconstruction of broken drawing lines and it produces worse effect on the final non-text mask, as shown in the middle row of Figure 6.

Here we present an efficient and easy to implement algorithm for reconstructing horizontal and vertical broken drawing lines. It can also be generalized for a variety of different orientations. Our drawing line reconstruction algorithm is described as follows: i) morphological thinning is applied to an input image, ii) then horizontal lines are identified using morphological hit-miss transform with horizontal structuring element, iii) the horizontal lines are then smoothed by using anisotropic Gaussian smoothing with $\sigma_x > \sigma_y$, which fills small gap between neighboring horizontal lines, iii) then smoothed image is binarized using global thresholding, iv) finally broken horizontal lines in the original image are reconstructed with respect to the joined lines in the binarized image, v) the same procedure is repeated for reconstructing vertical drawing lines by using vertical structuring element for hit-miss transform and $\sigma_x < \sigma_y$ for anisotropic Gaussian smoothing. The second improved version of Bloomberg's algorithm, i.e. reconstruction of broken drawing lines before hole-filling operation, is shown in Figure 3(c). This modification further improved the performance of Bloomberg's algorithm especially for correctly segmenting broken drawing lines type of non-text components, as shown in the bottom row of Figure 6.

Table 1. Performance evaluation of the original Bloomberg's text/image segmentation algorithm and our modified versions on UW-III[8] and ICDAR 2009[10] page segmentation competition test datasets.

	UW-III [a]			ICDAR-2009 [b]		
	Original	1st version	2nd version	Original	1st version	2nd version
non-text classified as non-text	95.36%	99.39%	99.51%	85.62%	91.44%	98.41%
text classified as text	99.79%	99.28%	99.19%	100%	99.11%	99.42%
segmentation accuracy	97.58%	**99.34%**	**99.35%**	92.81%	**95.28%**	**98.92%**

[a]Total 95 document images were selected from UW-III dataset which mainly composed of text and halftone components.
[b]ICDAR 2009 page segmentation test dataset consist of 8 document images; mostly with non-Manhattan layout.

4. EXPERIMENTS AND RESULTS

We have compared the performance of Bloomberg's original text/non-text segmentation algorithm and its modified versions by using following standard datasets: UW-III,[8] UNLV,[9] ICDAR-2009[10] page segmentation competition test images and our own private circuit diagrams images. The main reason of using different datasets is to compare the text/non-text segmentation accuracy of the original and modified versions of Bloomberg's algorithm on different types of document images with a variety of text and non-text components. A total of 95 documents mainly composed of text and halftone components were selected from the UW-III dataset. 100 documents comprising of text, halftone, graphs, drawings, and maps were selected from UNLV Magazine Sample 2 (category Z). Our own circuit diagrams dataset composed of 10 images having text and drawing components. ICDAR 2009 dataset contains 8 test images with non-Manhattan layout, unlike Manhattan layout of other selected datasets. For each dataset, pixel-level ground-truth images were generated using zone-level ground truth information. Each pixel in a ground-truth image contains either a text or a non-text label.

Different types of metrics were used for the performance evaluation of text/non-text segmentation algorithm, as defined below:

1. **non-text classified as non-text:** percentage of intersection of non-text pixels in both segmented image and ground-truth image with respect to the total number of non-text pixels in the ground-truth image.
2. **text classified as text:** percentage of intersection of text pixels in both segmented image and ground-truth image with respect to the total number of text pixels in the ground-truth image.
3. **segmentation accuracy:** average percentage of non-text classified as non-text and text classified as text accuracy.

Based on the metrics defined above, the performance evaluation results of the original Bloomberg's text/non-text segmentation algorithm and its modified versions are shown in Table 4 and 5. It is clearly visible in these tables that the modified versions of Bloomberg's algorithm have achieved better segmentation accuracy than the original algorithm.

5. DISCUSSION

In this paper, we have introduced the improved version of Bloomberg's[1] text/non-text segmentation algorithm in terms of better performance than the original algorithm. Bloomberg's original algorithm is one of a simple, easy to implement and fast page segmentation approach. Unlike classification based text/non-text segmentation approaches,[4–7] Bloomberg's algorithm can be equally applicable to a verity of different scripts document images. It gives correct text and non-text segmentation results, unless a document image contains very large text element(s) and/or very small non-text element(s). As shown in Table 4, it achieved good segmentation accuracy for UW-III dataset which mainly contains text and halftone elements. On the other hand, It achieved poor segmentation accuracy for UNLV and circuit diagrams datasets which mainly composed of different types of non-text elements,

Table 2. Performance evaluation of the original Bloomberg's text/image segmentation algorithm and our modified versions on UNLV dataset[9] and Circuits dataset (10 document images).

	UNLV [a]			CIRCUITS [b]		
	Original	1st version	2nd version	Original	1st version	2nd version
non-text classified as non-text	18.48%	72.98%	79.39%	0%	89.11%	90.31%
text classified as text	99.98%	97.97%	97.48%	100%	100%	96.67%
segmentation accuracy	59.23%	**85.48%**	**88.45%**	50%	**94.56%**	**93.49%**

[a] A subset of 100 images were selected from UNLV dataset comprising text, halftone, graphs, drawings, and maps.
[b] Our own samples of 10 document images which contain only text and drawings.

like drawings, graphics, maps and halftones, as shown in Table 5. This is because, Bloomberg's page segmentation algorithm is mainly designed for text and halftone segmentation and usually fails to segment thin-line drawing type non-text elements.

Here we have presented two simple modifications to the original Bloomberg's algorithm for transforming it in a general text and non-text image segmentation approach, where non-text components can be halftones, drawings and graphics. First modification is based on well-know hole-filling morphological operation and second modification is based on Gaussian smoothing and morphology based line reconstruction method for broken drawing lines. Our modification can perform better than the original algorithm, unless a text block is surrounded by a closed border, in such a case that text block may be segmented as non-text element. Our modifications achieved better segmentation accuracy than the original algorithm for different datasets containing a variety of text and non-text elements, as shown in Tables 4 and 5. These table show the performance results for both the modifications, so that one can compare the effects after each modification.

REFERENCES

[1] Bloomberg, D. S., "Multiresolution morphological approach to document image analysis," in [*Proceedings International Conference of Document Analysis and Recognition, (ICDAR '91)*], 963–971 (1991).
[2] Bloomberg, D. S., "Leptonica: An open source C library for efficient image processing and image analysis operations." http://code.google.com/p/leptonica/.
[3] Wong, K. Y., Casey, R. G., and Wahl, F. M., "Document analysis system," *IBM Journal of Research and Development* **26**(6), 647–656 (1982).
[4] Wang, Y., Haralick, R., and Phillips, I., "Improvement of zone content classification by using background analysis," in [*Proceedings 4th IAPR International Workshop on Document Analysis Systems, (DAS '00)*], (Dec. 2000).
[5] Keysers, D., Shafait, F., and Breuel, T. M., "Document image zone classification- a simple high-performance approach," in [*Proceedings 2nd International Conference Computer Vision Theory and Applications*], 44–51 (Mar. 2007).
[6] Moll, M. A., Baird, H. S., and An, C., "Truthing for pixel-accurate segmentation," in [*Proceedings 8th IAPR International Workshop Document Analysis Systems, (DAS '08)*], 379–385 (Sep. 2008).
[7] Bukhari, S. S., Shafait, F., and Breuel, T. M., "Document image segmentation using discriminative learning over connected components," in [*Proceedings 9th IAPR International Workshop on Document Analysis Systems, (DAS '10)*], 183–190 (2010).
[8] Guyon, I., Haralick, R. M., Hull, J. J., and Phillips, I. T., "Data sets for OCR and document image understanding research," in [*Handbook of character recognition and document image analysis*], Bunke, H. and Wang, P., eds., 779–799, World Scientific, Singapore (1997).
[9] http://www.isri.unlv.edu/ISRI/OCRtk.
[10] Antonacopoulos, A., Pletschacher, S., Bridson, D., and Papadopoulos, C., "ICDAR 2009 page segmentation competition," in [*Proceedings 10th International Conference on Document Analysis and Recognition, ICDAR '09*], 1370 –1374 (July 2009).