DOCUMENT IMAGE CLEAN-UP AND BINARIZATION *

Victor Wu and R. Manmatha
Multimedia Indexing And Retrieval Group
Computer Science Department
University of Massachusetts, Amherst, MA 01003-4610
Email:{vwu,manmatha}@cs.umass.edu
December 18, 1997

Abstract

Image binarization is a difficult task for documents with text over textured or shaded backgrounds, poor contrast, and/or considerable noise. Current optical character recognition (OCR) and document analysis technology do not handle such documents well. We have developed a simple yet effective algorithm for document image clean-up and binarization.

The algorithm consists of two basic steps. In the first step, the input image is smoothed using a low-pass (Gaussian) filter. The smoothing operation enhances the text relative to any background texture. This is because background texture normally has higher frequency than text does. The smoothing operation also removes speckle noise. In the second step, the intensity histogram of the smoothed image is computed and a threshold automatically selected as follows. For black text, the first peak of the histogram corresponds to text. Thresholding the image at the value of the valley between the first and second peaks of the histogram binarizes the image well. In order to reliably identify the valley, the histogram is smoothed by a low-pass filter before the threshold is computed.

The algorithm has been applied to some 50 images from a wide variety of sources: digitized video frames, photos, newspapers, advertisements in magazines or sales flyers, personal checks, etc. There are 21820 characters and 4406 words in these images. 91% of the characters and 86% of the words are successfully cleaned up and binarized. A commercial OCR was applied to the binarized text when it consisted of fonts which were OCR recognizable. The recognition rate was 84% for the characters and 77% for the words.

Keywords — binarization, background removal, OCR, text extraction

*This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by the National Science Foundation under grant number IRI-9619117 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

1 Introduction

Current optical character recognition (OCR) technology [1, 2] is largely restricted to recognizing text printed against clean backgrounds. Most OCR engines require that the input image be binarized before the characters can be processed. Usually, a simple global binarization technique is adopted which does not handle well text printed against shaded or textured backgrounds, and/or embedded in images.

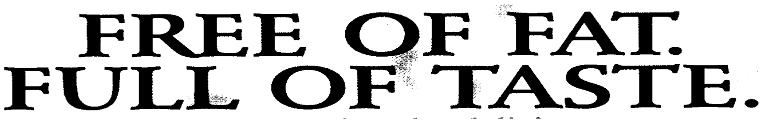
In this paper, a simple yet effective algorithm is proposed for document image binarization and clean up. It is especially robust for extracting text from images judging by the high OCR recognition rate achieved for the extracted text.

There are basically two classes of binarization (thresholding) techniques —- global and adaptive. Global methods binarize the entire image using a single threshold. For example, a typical OCR system separates text from backgrounds by *global thresholding* ([3, 4]). A simple way to automatically select a global threshold is to use the value at the valley of the intensity histogram of the image, assuming that there are two peaks in the histogram, one corresponding to the foreground, the other to the background. Methods have also been proposed to facilitate more robust valley picking [5].

There are problems with the above global thresholding paradigm. First, due to noise and poor contrast, many images do not have well-differentiated foreground and background intensities. Second, the bimodal histogram assumption is often not valid for images of complicated documents such as advertisements and photographs. Third, the foreground peak is often overshadowed by other peaks which makes the valley detection difficult or impossible. Some research has been carried out to overcome some of these problems. For example, weighted histograms [6] are used to balance the size difference between the foreground and background, and/or convert the valley-finding into maximum peak detection. Minimum-error thresholding [7, 8] models the foreground and background intensity distributions as Gaussian distributions and the threshold is selected to minimize the misclassification error. Otsu [9] models the intensity histogram as a probability distribution and the threshold is chosen to maximize the separability of the resultant foreground and background classes. Similarly, entropy measures have been used [10, 11, 12] to select the threshold which maximizes the sum of foreground and background entropies. In addition, Tsai [13] uses the threshold which best preserves the moment statistics of the binarized image as compared with the original grey-scale image. Liu and Srihari [14] uses Otsu's algorithm [9] to obtain candidate thresholds, each of which is used to produce an intermediate binarized image. Then, text features are measured from each binarized image. These features are used to pick the best threshold among the candidates.

In contrast, adaptive algorithms compute a threshold for each pixel based on information extracted from its neighborhood (local window) [15, 16]. For images in which the intensity ranges of the foreground objects and backgrounds entangle, different thresholds must be used for different regions. Domain dependent infomation can also be coded in the algorithm to get the work done [16]. Trier and Taxt [17] evaluated eleven local adaptive thresholding schemes for map images.

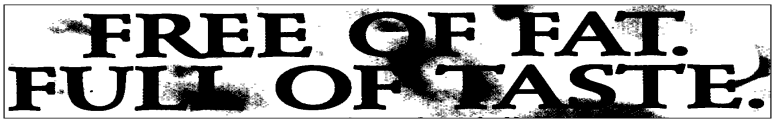
(a) Input



(b) Wu & Manmatha

(c) Tsai

(d) Otsu



(e) Kamel & Zhao

Figure 1: Example: Comparison of the four algorithms.

2 The New Algorithm

The algorithm proposed here works under the usual assumption that text in the input image or a region of the input image (called a *text chip*) has more or less the same intensity value. However, a unique feature of this algorithm is that it works well even if the text is printed against shaded or hatched background as shown in Figure 1(a) and (b). Figure 1 also demonstrates that our new algorithm performs better than Tsai's moment-preserving method [13], Otsu's histogram-based algorithm [9], and Kamel & Zhao's adaptive binarization algorithm [16].

The algorithm consists of the following steps:

1. smooth the input text chip

2. compute the intensity histogram of the smoothed chip.

3. smooth the histogram using a low-pass filter.

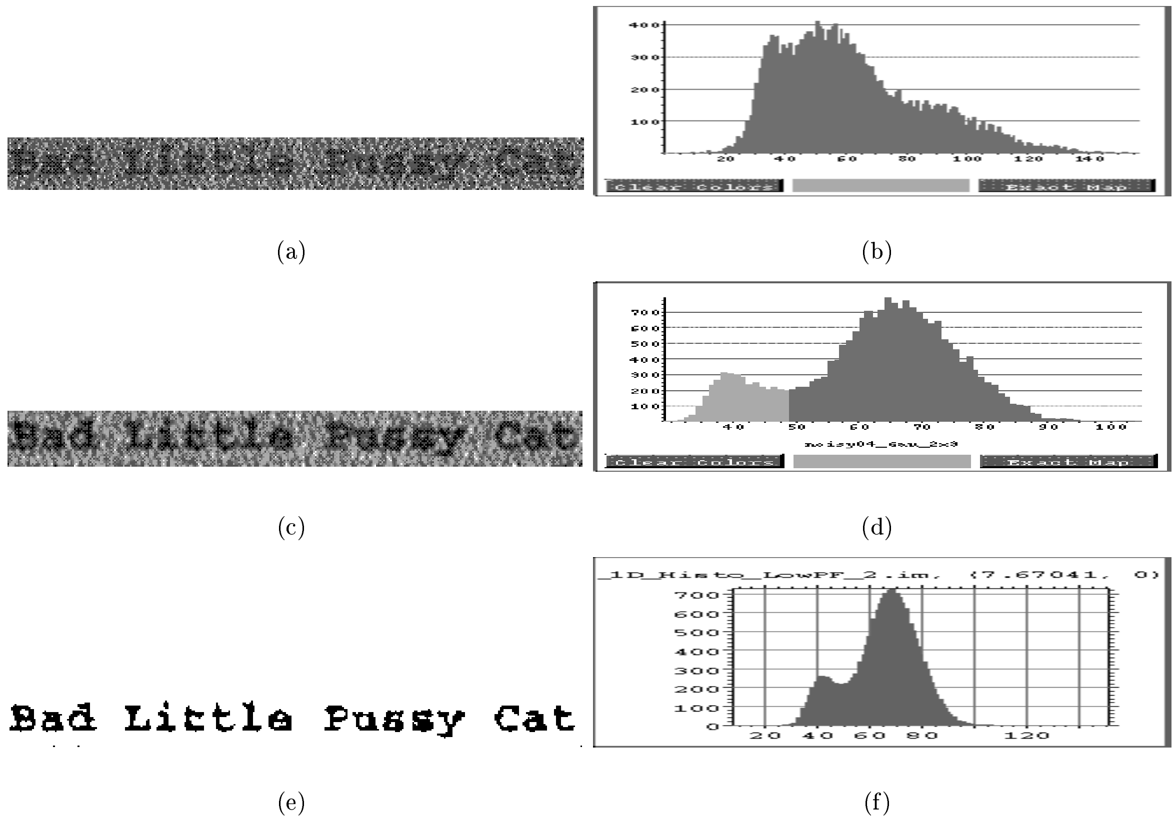


Figure 2: The Text Clean-up process. (a) Original text chip; (b) Histogram of a; (c) Smoothed version of a; (d) Histogram of c; (e) The binarization result by thresholding c using a value in the valley of f; (f) Smoothed version of d.

4. pick a threshold at the first valley counted from the left side of the histogram.

5. binarize the smoothed text chip using the threshold.

A low-pass Gaussian filter is used to smooth the input text chip in step 1. The smoothing operation affects the background more than the text because the text normally is of lower frequency than the shading. Thus it cleans up the background. Another way of looking at it is to notice that the smoothing blends the background into grey while leaving the text black, so that the separation of first two peaks of the histogram becomes more prominent as shown in Figure 2(b) and (d).

The histogram generated by step 2 is often jagged, hence it needs to be smoothed to allow the valley to be detected (see Figure 2(d)). Again, a Gaussian filter is used for this purpose.

Text is normally the darkest item in the detected chips. Therefore, a threshold is picked at the first valley closest to a dark side of the histogram. To extract text against darker background, a threshold at the last valley is picked instead.

The thresholded image is shown at bottom left in Figure 2. This has been successfully recognized by an OCR system.

Wu, Manmatha and Riseman [18] have developed a system called TextFinder which detects and generates text chips automatically. Thus, even though one threshold is used for

Table 1: Summary of the system's performance. 48 images were used for detection and clean-up. Out of these, 35 binarized images were used for the OCR process.

	Total Detected	Total Clean-up	Total OCRable	Total OCRed
Char	20788	91%	14703	12428 (84%)
Word	4139	86%	2981	2314 (77%)

the entire text chip, different threholds may be used for the input image. Therefore, this is a local, adaptive approach as opposed to global threshold methods.

3 Experiment

The algorithm has been tested using text from 48 images. Some of the test images were downloaded from the Internet, some from the Library of Congress, and others were locally scanned documents. These test images came from a wide variety of sources: digitized video frames, photographs, newspapers, advertisements in magazines or sales flyers, and personal checks. Some of the images have regular page layouts, others do not. It should be pointed out that all the algorithm parameters remain the same throughout the whole set of test images, showing the robustness of the system.

For the images scanned by us, a resolution of 300dpi (dots per inch) was used. This is the standard resolution required, for example, by the Caere OCR engine that was used. It should be pointed out that 300dpi resolution is not required by our algorithm. In fact, no assumptions are made about the resolution of the input images, since such information is normally not available for the images from outside sources, such as those downloaded from the Internet.

3.1 Text Clean-up

Characters and words (as perceived by one of the authors) detected by the TextFinder [18] were counted in each image (the ground truth). The total numbers over the whole test set are shown in the "Total Detected" column in Table 1. Then, characters and words which are clearly readable by a person after the binarization operation were counted for each image. Note that only the text which is horizontally aligned is counted (skew angle of the text string is less than roughly 30 degrees). These extracted characters (words) are called cleaned-up characters (words)[1].

As shown in Table 1, there are a total 20788 characters and 4139 words used for binarization. 91% of the characters and 86% of the words are successfully cleaned[2]. Since this evaluation is subjective, we extend our effort to obtain an objective meaure as described in the next section

[1]Due to space limitations, the table of the above results itemized for each test image is not included in this paper

[2]A word is successfully cleaned only if all its characters are clearly recognizable by a person.

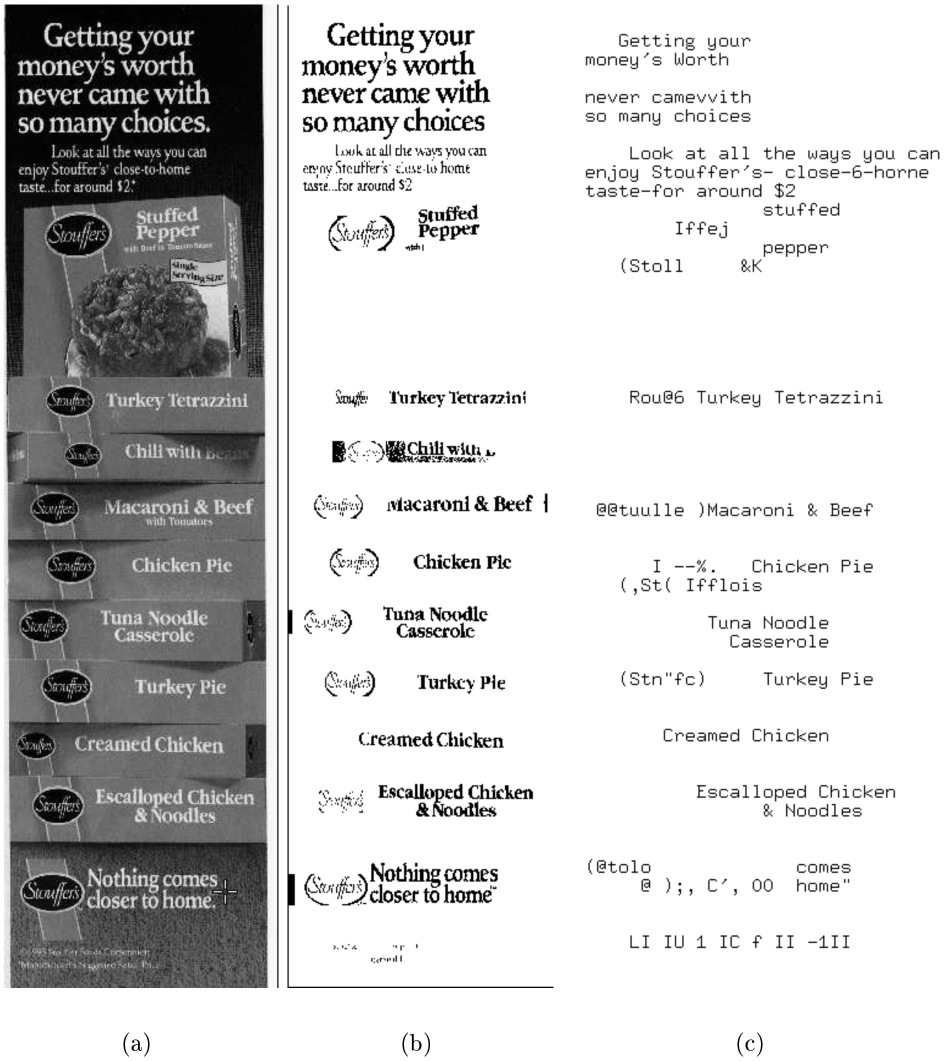


Figure 3: Example 1. (a) Original image (ads11); (b) Extracted text; (c) The OCR result using Caere's WordScan Plus 4.0 on b.

3.2 OCR Testing

In this part of experiement, Caere's WordScan Plus 4.0 for Windows was used over the binarized text to do character recognition. The recognition rate is used as an objective measure of the performance of binarization and clean-up algorithm.

35 images were reconstructed by mapping the extracted text back to the corresponding input images used in the previous experiment. In Table 1, the column "Total OCRable" shows the total number of extracted characters (words) (shown in the Total Clean-up column) that appear to be of machine printed fonts in the corresponding images (Note that only the machine printed characters are counted so that the OCR engine can be applied). The "Total OCRed" column shows the number of characters (words) in these images which are correctly recognized by the OCR engine.

As shown in the table, there are 14703 characters and 2981 printed words which are

OCRable in these images. 12428 (84%) of the characters and 2314 (77%) of the words are correctly recognized by the OCR engine. The normalized percentages are 78% and 72% respectively.

Figure 3(a) is the original image of file ads11. This is an image of an advertisement for Stouffer's, which has no structured layout. The final binarization result is shown in the middle. The corresponding OCR output is shown on the right. This example is intended to provide a feeling for the overall performance of the system by showing whole images. The drawback is that some fine details are lost due to the scaling of the images to fit the page. For example, the words of the smaller fonts and the word Stouffer's in script appear to be fragmented, although actually they are not.

The OCR engine correctly recognized most of the text of machine-printed fonts as shown in Figure 3 (c). It made mistakes on the Stouffer's trademarks since they are in script. It should be pointed out that the clean-up output looks fine to a person in the places where the rest of the OCR errors occurred.

3.3 Comparison With Other Thresholding Methods

In this section, we compare the performance of our algorithm with those of Tsai [13], Otsu [9] and Kamel and Zhao's adaptive method [16].

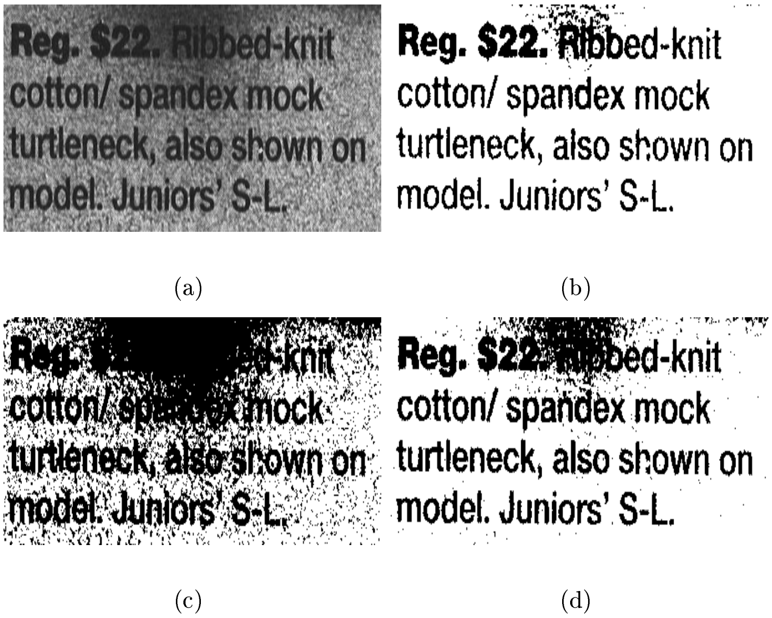
12 image chips cropped from our test image set are used for this experiment. The total number of characters/words are counted as shown in Table 2. Each of these image chips is then binarized using the four algorithms. The binarized images are then examined by one of the authors. Each output is assigned a score from 0 (worst) to 10 (perfect) based on the cleanness, readability, separability (characters should not be attached to each other) and completeness (a character should not be broken) of the characters. The result is shown in Table 2 under column "Quality".

To obtain more objective measures of the performances of the algorithms, the binarized images are fed to Caere's WordScan Plus 4.0 for OCR. The correctly recognized characters and words are then counted with the totals shown in Table 2 under columns "OCRed Char" and "OCRed Word". Note that a word is correctly recognized if and only if all its characters are correctly recognized.

As shown in Table 2, our algorithm out-performs all the rest of the algorithms with a significant margin in both subjective and objective measures. Tsai's method comes second in both categories. However, both Tsai and Otsu's methods require multi-thresholding to obtain

Table 2: Summary of performance comparison. 12 image chips were used. No OCR was performed for Kamal and Zhao's method due to the poor binarization quality.

Method	Quality (max 120)	OCRed Char (total 505)	OCRed Word (total 105)
Wu/Manmatha	108 / 90%	450 / 89%	80 / 76%
Tsai	89 / 74%	228 / 45%	34 / 32%
Otsu	58 / 48%	174 / 34%	25 / 24%
Kamal/Zhao	45 / 37%	—	—



(a) (b)

(c) (d)

Figure 4: Example: Wu/Manmatha verses Tsai. (a) Input image; (b) Binarization result using the new algorithm; (c) Binarization result using Tsai's algorithm; (d) Result by thresholding the input into three-levels using Tsai's algorithm.

their best result as shown in Figure 4, which causes problems for their automatic application. In this experiment, only the best results of these two algorithms are used. Kamel and Zhao's method is quite input-dependent, and we failed to find a set of values for its parameters which are reasonable to all the images.

Figure 5 gives another example which shows that the new algorithm out-performs Tsai's method. Even though the OCR engine made some errors as shown in 5(c), it failed completely when fed with the output of Tsai's algorithm as shown in 5(d).

4 Conclusion

Current OCR and other document segmentation and recognition technologies do not work well for documents with text printed against shaded or textured backgrounds or those with non-structured layouts. In contrast, we have proposed a simple and robust text binarization and clean-up which works well for normal documents as well as documents described in the above situations. The algorithm is histogram based. The crusial steps are that the input image is first low-pass filtered, then the histogram is computed, and finally the histogram is smoothed before the automatic threhold-picking starts.

48 images from a wide variety of sources such as newspapers, magazines, printed advertisement, photographs, and checks have been tested on the system. They are greyscale images with structured and non-structured layouts and a wide range of font styles (including certain script and hand-written fonts) and sizes (practically, the font sizes do not matter for the system). Some text has overlapping background texture patterns in the images.

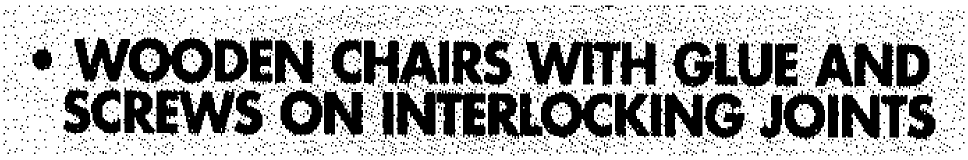
Out of the test images, there are a total 20788 characters and 4139 words used for binarization. 91% of the characters and 86% of the words are successfully cleaned. Furthermore,

(a)

(b)

```
9 WOODEN CHAIRS WITH GWE AND
 SCREWS C04 INTERLOCKING JOINTS
```

(c)



(d)

Figure 5: A Text Clean-up example. (a) Original text chip; (b) Binarization result using the new algorithm ; (c) The OCR result of b using Caere's Wordscan Plus 4.0 ; (d) Binarization output using Tsai's algorithm. No character is recognized by the OCR engine.

there are 14703 characters and 2981 printed words using fonts which are recognizable by an OCR in these images. 12428 (84%) of the characters and 2314 (77%) of the words are correctly recognized by the OCR engine. The normalized percentages are 78% and 72% respectively.

Our comparative study also shows that the new algorithm significantly out-performs Tsai's moment-preverving method, Otsu's histogram-based scheme, and Kamel and Zhao's adaptive algorithm.

The algorithm is stable and robust — all the parameters remain the same throughout all the experiments.

5 Acknowledgments

We would like to thank Bruce Croft and CIIR for supporting this work. We would also like to thank Adam Jenkins for his support in programming.

References

[1] M. Bokser, "Omnidocument Technoligies," *Proceedings of The IEEE* **80**, pp. 1066–1078, July 1992.

[2] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical Review of OCR Research and Development," *Proceedings of The IEEE* **80**, pp. 1029–1058, July 1992.

[3] C. A. Glasbey, "An Analysis of Histogram-Based Thresholding Algorithms," *CVGIP: Graphical Models and Image Processing* **55**, pp. 532–537, Nov. 1993.

[4] L. O'Gorman, "Binarization and Multithresholding of Document Images Using Connectivity," *Computer Vision, Graphics and Image Processing* **56**, pp. 494–506, Nov. 1994.

[5] A. Rosenfeld and P. D. L. Torre, "Histogram Concavity Analysis as an Aid in Threshold Selection," *IEEE Trans. Systems, Man, and Cybernetics* **SMC-13**, pp. 231–235, 1983.

[6] J. S. Weszka and A. Rosenfeld, "Histogram Modification for Threshold Selection," *IEEE Trans. on Systems, Man, and Cybernetics* **SMC-9**, pp. 38–52, Jan. 1979.

[7] J. Kittler and J. Illingworth, "Minimum Error Thresholding," *Pattern Recognition* **19**(1), pp. 41–47, 1986.

[8] Q. Z. Ye and P. E. Danielson, "On Minimum Error Thresholding and Its Implementation," *Pattern Recognition Letters* **7**, pp. 201–206, Apr. 1988.

[9] N. Otsu, "A Threshold Selection Method from Gray-Level Histogram," *IEEE Trans. on Systems, Man, and Cybernetics* **SMC-9**, pp. 62–66, Jan. 1979.

[10] A. S. Abutaleb, "Automatic Thresholding of Gray-Level Picture Using Two-Dimensional Entropy," *CVGIP: Graphical Models and Image Processing* **47**, pp. 22–32, July 1989.

[11] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram," *Computer Vision, Graphics and Image Processing* **29**, pp. 273–285, Mar. 1985.

[12] T. Pun, "Entropic Thresholding: A New Approach," *CVGIP: Graphical Models and Image Processing* **16**, pp. 210–239, July 1981.

[13] W. H. Tsai, "Moment-Preserving Thresholding: A New Approach," *Computer Vision, Graphics, and Image Processing* **29**, pp. 377–393, Mar. 1985.

[14] Y. Liu and S. N. Srihari, "Document Image Binarization Based on Texture Features," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **19**, pp. 540–544, May 1997.

[15] R. G. Casey and K. Y. Wong, "Document Analysis System and Techniques," *in Image Analysis Applications, R. Kasturi and M. M. Trivedi eds. Marcel Dekker, New York, N.Y.* , pp. 1–36, 1990.

[16] M. Kamel and A. Zhao, “Extraction of Binary Character/Graphics Images from Grayscale Document Images,” *Computer Vision, Graphics and Image Processing* **55**, pp. 203–217, May. 1993.

[17] Ø. D. Trier and T. Taxt, “Evaluation of Binarization Methods for Document Images,” *IEEE Transactions on Pattern Analysis And Machine Intelligence* **17**, pp. 312–315, March 1995.

[18] V. Wu, R. Manmatha, and E. M. Riseman, “Finding Text In Images,” *Proc. of the 2nd intl. conf. on Digital Libraries. Philadaphia, PA* , pp. 1–10, July 1997.