Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images

Faisal Shafait[a], Daniel Keysers[a], Thomas M. Breuel[b]

[a]Image Understanding and Pattern Recognition (IUPR) Research Group
German Research Center for Artificial Intelligence (DFKI) GmbH
D-67663 Kaiserslautern, Germany
faisal@iupr.dfki.de, keysers@iupr.dfki.de

[b]Department of Computer Science, Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
tmb@informatik.uni-kl.de

ABSTRACT

Adaptive binarization is an important first step in many document analysis and OCR processes. This paper describes a fast adaptive binarization algorithm that yields the same quality of binarization as the Sauvola method,[1] but runs in time close to that of global thresholding methods (like Otsu's method[2]), independent of the window size. The algorithm combines the statistical constraints of Sauvola's method with integral images.[3] Testing on the UW-1 dataset demonstrates a 20-fold speedup compared to the original Sauvola algorithm.

Keywords: Document binarization, adaptive thresholding, local thresholding, integral images

1. INTRODUCTION

Document binarization is the first step in most document analysis systems. The goal of document binarization is to convert the given input greyscale or color document into a bi-level representation. This representation is particularly convenient because most of the documents that occur in practice have one color for text (e.g. black), and a different color (e.g. white) for background. These colors are typically chosen to be high-contrast so that the text is easily readable by humans. Therefore, most of the document analysis systems have been developed to work on binary images.[4] The performance of subsequent steps in document analysis like page segmentation[5] or optical character recognition heavily depend on the result of binarization algorithm. Over the last three decades many different approaches have been proposed for binarizing a greyscale document.[1,2,6–9] Color documents can either be converted to greyscale before binarization, or techniques specialized for binarizing colored documents directly[10–12] can be used.

In this paper we focus on the binarization of greyscale documents because in most cases color documents can be converted to greyscale without losing much information as far as distinction between page foreground (text) and background is concerned. Some exceptions to this case are advertisements and some magazine styles. The binarization techniques for greyscale documents can be grouped into two broad categories: global binarization and local binarization. Global binarization methods like that of Otsu[2] try to find a single threshold value for the whole document. Then each pixel is assigned to page foreground or background based on its grey value. Global binarization methods are very fast and they give good results for typical scanned documents. However if the illumination over the document is not uniform, for instance in the case of scanned book pages or camera-captured documents, global binarization methods tend to produce marginal noise along the page borders.[13] Another class of documents are historical documents in which image intensities can change significantly within a document. Local binarization methods[1,7,8] try to overcome these problem by computing thresholds individually for each pixel using information from the local neighborhood of the pixel. These methods are usually able to achieve good results even on severely degraded documents, but they are often slow since the computation of image features from the local neighborhood is to be done for each image pixel.

This paper presents a fast approach to compute local thresholds without compromising the performance of local thresholding techniques. Our approach uses the concept of sum-tables[14] that were made popular in the computer vision community by Viola and Jones.[3] Using this approach we are able to achieve binarization speed close to the global binarization methods with the performance as good as that of local binarization schemes. The following section (Sec. 2) describes our approach for combining integral images with the local adaptive thresholding techniques. The evaluation of our approach is described in Sec. 3, followed by conclusion in Sec. 4.

2. BINARIZATION USING INTEGRAL IMAGES

Comparison of different techniques for document binarization has received some attention in the past. Trier et al.[15] evaluated eleven different locally adaptive binarization methods for gray scale images with low contrast, variable background intensity and noise. Their evaluation showed that Niblack's method[8] performed better than other local thresholding methods. Badekas et al.[16] evaluated seven different algorithms for binarizing old Greek documents. They found that Sauvola's binarization method[1] - which is an improvement over Niblack's method - works best among the local thresholding techniques; whereas Otsu's binarization method[2] outperforms other global binarization techniques. Overall, Sauvola's binarization method works slightly better than Otsu's technique in their experiments. Another comprehensive evaluation of thresholding techniques is done by Sezgin et al.[17] They have evaluated 40 different thresholding methods in the applications of non-destructive testing and document analysis. In the case of document analysis, they synthetically degraded document images using Baird's degradation model.[18] They also concluded that Sauvola's binarization method works better than other local binarization techniques. Bradley et al.[19] have recently proposed real-time adaptive thresholding using mean of a local window, where local mean is computed using integral image. However, for thresholding document images, local mean alone does not work as good as considering both local mean and local variance.[17] Based on these findings, we chose Sauvola's binarization technique as a representative state-of-the-art technique for binarization of greyscale documents. Results of running the Otsu and Sauvola binarization algorithms on a camera captured document are shown in Figure 1. Illumination over the original greyscale image (Figure 1(a)) is not uniform, changing from a bright region near the right border to a much darker region near the left border. Hence the grey value of the pixels near the left border is lower than the global threshold computed by the Otsu's method, resulting in a thick black bar near the left side of the image loosing all the information in that region (Figure 1(b)). Sauvola's method computes a local threshold for each pixel individually taking into account image intensities in the local neighborhood of the pixel only (Section 2.1). This results in a much better output (Figure 1(c)) at a cost of a 30-fold increase in computation time as compared to Otsu's method. Our proposed modification (Section 2.2) achieves the same result as that of Sauvola (Figure 1(d)) with a 10-fold decrease in computation time for this image.

2.1 Local Adaptive Thresholding Using Sauvola's Technique

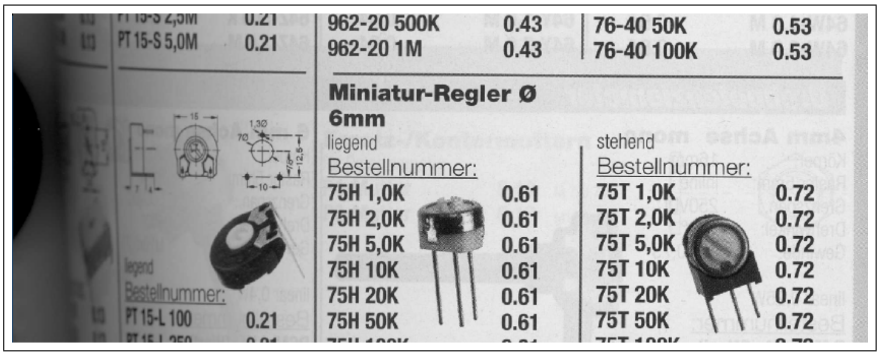
Consider a greyscale document image in which $g(x, y) \in [0, 255]$ be the intensity of a pixel at location (x, y). In local adaptive thresholding techniques, the aim is to compute a threshold $t(x, y)$ for each pixel such that

$$o(x,y) = \begin{cases} 0 & \text{if } g(x,y) \le t(x,y) \\ 255 & \text{otherwise} \end{cases} \tag{1}$$

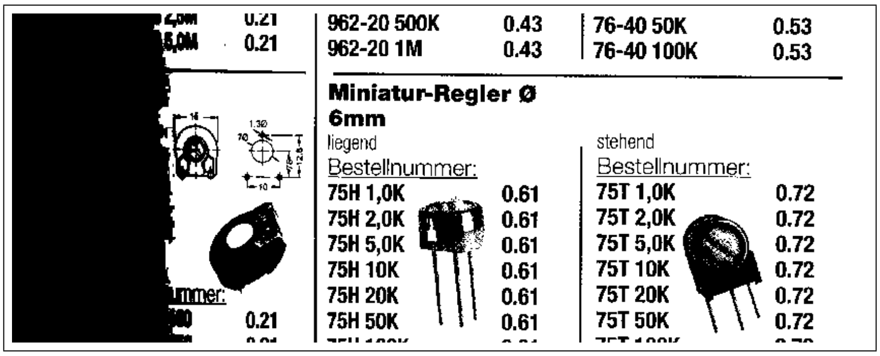
In Sauvola's binarization method, the threshold $t(x, y)$ is computed using the mean $m(x, y)$ and standard deviation $s(x, y)$ of the pixel intensities in a $w \times w$ window centered around the pixel (x, y):

$$t(x,y) = m(x,y) \left[1 + k \left(\frac{s(x,y)}{R} - 1 \right) \right] \tag{2}$$

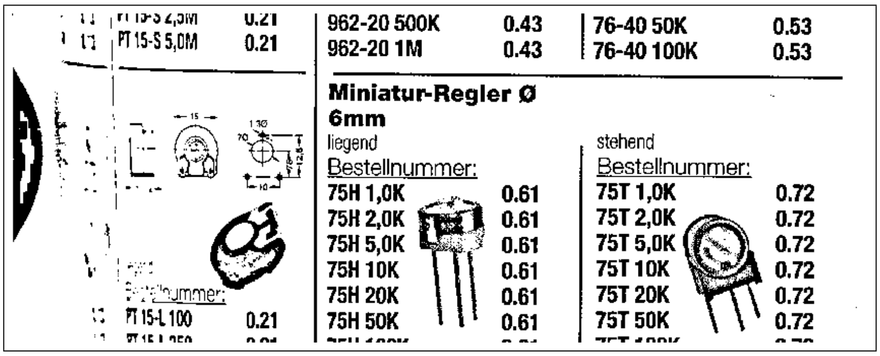
where R is the maximum value of the standard deviation ($R = 128$ for a greyscale document), and k is a parameter which takes positive values in the range $[0.2, 0.5]$. The local mean $m(x, y)$ and standard deviation $s(x, y)$ adapt the value of the threshold according to the contrast in the local neighborhood of the pixel. When there is high contrast in some region of the image, $s(x, y) \approx R$ which results in $t(x, y) \approx m(x, y)$. This is the same result as in Niblack's method. However, the difference comes in when the contrast in the local neighborhood is quite low. In that case the threshold $t(x, y)$ goes below the mean value thereby successfully removing the relatively dark regions of the background. The parameter k controls the value of the threshold in the local window such that the higher the value of k, the lower the threshold from the local mean $m(x, y)$.



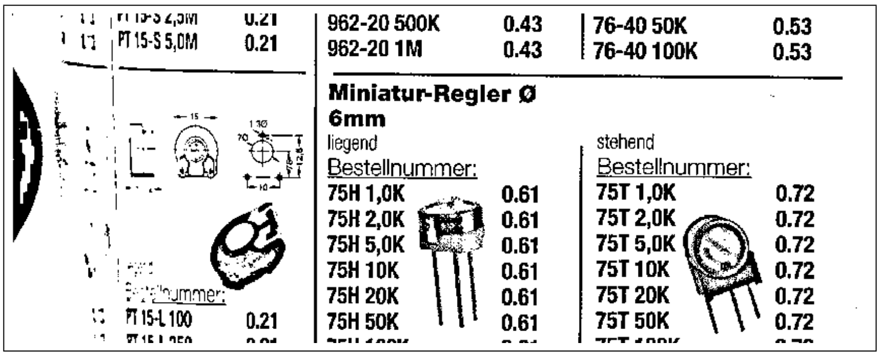
(a) Input Image (842 × 324)



(b) Otsu's result ($t = 16$ msec)



(c) Sauvola's result ($t = 484$ msec)



(d) Our algorithm's result ($t = 47$ msec)

Figure 1. Result of applying the Otsu[2] and Sauvola[1] binarization algorithms on a camera-captured document. Note that Sauvola's method achieves better results at a cost of a 30-fold increase in computation time as compared to Otsu's method. Our proposed modification achieves the same result as that of Sauvola with a 10-fold decrease in computation time.

The statistical constraint in Equation 2 gives very good results even for severely degraded documents. However in order to compute the threshold $t(x, y)$, local mean and standard deviation have to be computed for each pixel. Computing $m(x, y)$ and $s(x, y)$ in a naive way results in a computational complexity of $O(W^2N^2)$ for an $N \times N$ image. In order to speed up the computation, Sauvola et al. propose computing a threshold for every nth pixel and then using interpolation for the rest of the pixels. This speeds up the computation by some factors at the cost of reduced accuracy of determining the threshold. In addition, the computational complexity is still a quadratic function of the local window dimension. In the following, we propose an efficient way of computing local means and variances using sum tables (integral images) such that the computational complexity does not depend on the window dimension anymore.

2.2 Integral Images for Computing Local Means and Variances

The concept of integral images was popularized in computer vision by Viola and Jones[3] based on prior work in graphics.[14] An integral image i of an input image g is defined as the image in which the intensity at a pixel position is equal to the sum of the intensities of all the pixels above and to the left of that position in the original image. So the intensity at position (x, y) can be written as

$$I(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} g(i, j) \tag{3}$$

The integral image of any greyscale image can be efficiently computed in a single pass.[3] Once we have the integral image, the local mean $m(x, y)$ for any window size can be computed simply by using two addition and two subtraction operations instead of the summation over all pixel values within that window:[20]

$$\begin{aligned} m(x, y) \quad = \quad & (I(x + w/2, y + w/2) + I(x - w/2, y - w/2) - \\ & I(x + w/2, y - w/2) + I(x - w/2, y + w/2))\,/w^2 \end{aligned} \tag{4}$$

Similarly, if we consider the computation of the local variance

$$s^2(x, y) = \frac{1}{w^2} \sum_{i=x-w/2}^{x+w/2} \sum_{j=y-w/2}^{y+w/2} g^2(i, j) - m^2(x, y) \tag{5}$$

the first term in Equation 5 can be computed in a similar way as Equation 4 by using an integral image of the squared pixel intensities. An important hint from implementation point of view is that the values of the squared integral image get very large, so overflow problems might occur if 32-bit integers are used. Once computed the integral image of the pixel intensities and the square of the pixel intensities, local means and variances can be computed very efficiently, independent of the local window size. Using these formulas in Equation 2 reduces the computational complexity from $O(W^2N^2)$ to $O(N^2)$.

3. EXPERIMENTS AND RESULTS

In order to measure the gain in speed, we tested our approach on the University of Washington (UW-1) dataset. The dataset contains 125 greyscale images scanned at a resolution of 300-dpi. The dimensions of the images are 2530x3300 with little variations for each image. The experiment was conducted on an AMD Opteron 2.4 GHz machine running Linux. Otsu's binarization algorithm[2] was used as representative for global binarization methods, whereas Sauvola's binarization algorithm[1] was chosen as a representative local binarization technique. A comparison of mean running times is shown in Figure 2. The figure shows that by using our proposed algorithm, the execution time of Sauvola binarization came close to that of Otsu's binarization method. Mean running time for the Otsu's binarization method was 2.0 secs whereas our algorithm took a mean running time of 2.8 secs. Original Sauvola algorithm took 12.6 secs when using a local window of 15×15 and 65.5 secs when using a 40×40 local window. This amounts to a 5-fold speed gain in case of small window size (15×15) and a 20-fold speed gain in case of large window size (40×40).

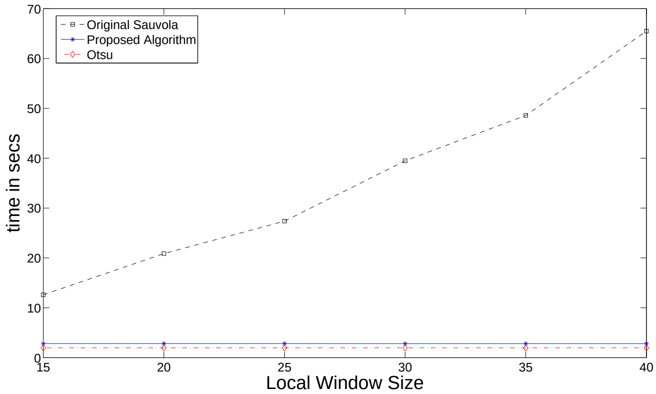


Figure 2. Comparison of mean running times of the original Sauvola binarization method,[1] the proposed algorithm, and the Otsu binarization method[2] on the UW-1 dataset. It is clear from the graph that the proposed technique achieves speed close to Otsu's binarization method, while computing the same local thresholds as Sauvola's method.

4. CONCLUSION

In this paper we presented a novel way of computing thresholds for locally adaptive binarization schemes. We used integral images to compute mean and variance in local windows, which resulted in an algorithm whose running time does not depend on the local window size. Using the threshold function of Sauvola, we were able to achieve the same results as those of Sauvola, but in a time close to that of global binarization schemes like Otsu.

Note that the proposed modifications not only apply to Sauvola's local thresholding technique, but to other techniques that use local mean and variance as well, e.g. Niblack's.[8]

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03).

REFERENCES

1. J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition* **33**(2), pp. 225–236, 2000.
2. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems, Man, and Cybernetics* **9**(1), pp. 62–66, 1979.
3. P. Viola and M. J. Jones, "Robust real-time face detection," *Int. Journal of Computer Vision* **57**(2), pp. 137–154, 2004.
4. R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena, "Geometric layout analysis techniques for document image understanding: a review," tech. rep., IRST, Trento, Italy, 1998.
5. F. Shafait, D. Keysers, and T. M. Breuel, "Performance comparison of six algorithms for page segmentation," in *7th IAPR Workshop on Document Analysis Systems*, pp. 368–379, (Nelson, New Zealand), Feb. 2006.
6. J. M. White and G. D. Rohrer, "Image thresholding for optical character recognition and other applications requiring character image extraction," *IBM Journal of Research and Development* **27**, pp. 400–411, July 1983.
7. J. Bernsen, "Dynamic thresholding of gray level images," in *Proc. Intl. Conf. on Pattern Recognition*, pp. 1251–1255, 1986.
8. W. Niblack, *An Introduction to Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1986.
9. L. O'Gorman, "Binarization and multithresholding of document images using connectivity," *Graphical Model and Image Processing* **56**, pp. 494–506, Nov. 1994.

10. K. Sobottka, H. Kronenberg, T. Perroud, and H. Bunke, “Text extraction from colored book and journal covers,” *Int. Jour. on Document Analysis and Recognition* **2**, pp. 163–176, June 2000.
11. C. Tsai and H. Lee, “Binarization of color document images via luminance and saturation color features,” *IEEE Trans. on Image Processing* **11**, pp. 434–451, April 2002.
12. E. Badekas, N. Nikolaou, and N. Papamarkos, “Text binarization in color documents,” *Int. Jour. of Imaging Systems and Technology* **16**(6), pp. 262–274, 2006.
13. F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel, “Page frame detection for marginal noise removal from scanned documents,” in *15th Scandinavian Conference on Image Analysis*, pp. 651–660, (Aalborg, Denmark), June 2007.
14. F. C. Crow, “Summed-area tables for texture mapping,” *Computer Graphics - Proceedings of SIGGRAPH’84* **18**(3), pp. 207–212, 1984.
15. O. Trier and T. Taxt, “Evaluation of binarization methods for document images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence* **17**, pp. 312–315, March 1995.
16. E. Badekas and N. Papamarkos, “Automatic evaluation of document binarization results,” in *10th Iberoamerican Congress on Pattern Recognition*, pp. 1005–1014, (Havana, Cuba), 2005.
17. M. Sezgin and B. Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic Imaging* **13**(1), pp. 146–165, 2004.
18. H. S. Baird, “State of the art of document image degradation modeling,” in *IAPR Workshop on Document Analysis Systems*, (Rio de Janeiro, Brazil), Dec. 2000.
19. D. Bradley and G. Roth, “Adaptive thresholding using the integral image,” *Journal of Graphics Tools* **12**(2), pp. 13–21, 2007.
20. F. Porikli and O. Tuzel, “Fast construction of covariance matrices for arbitrary size image windows,” in *Proc. Intl. Conf. on Image Processing*, pp. 1581–1584, (Atlanta, GA, USA), Oct. 2006.