Journal of Multimedia Processing and Technologies

Print ISSN: 0976-4127
Online ISSN: 0976-4135

DLINE JOURNALS

JMPT 2024; 15 (1)
https://doi.org/10.6025/jmpt/2024/15/1/1-8

An Efficient Adaptive Local Binarization Algorithm for Extracting Text from an Image with a Complex Background

Antoaneta Popova
Faculty of Telecommunications at Technical University of Sofia
8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria
antoaneta.popova@tu-sofia.bg

ABSTRACT

The adaptive local threshold algorithm is described in this paper. It has the same quality as Sauvola and is as fast as global threshold methods. The adaptive local threshold calculation is independent of the operator window size. It combines the advantages of Wiener filter prior processing, the integral images, and local threshold calculation.

Received: 19 August 2023
Revised: 28 November 2023
Accepted: 6 December 2023

Keywords: Binarization, Local Thresholding, Integral Images, Wiener Filter

1. Introduction

The binarization converts the input grayscale or color image into a bi-level representation. Most document analysis systems use this approach. The next steps in document analysis like Optical Character Recognition (OCR) heavily depend on the result of binarization algorithm. Several different methods for image thresholding have been proposed the last decade.

In this paper we focus on the binarization of color or grayscale images with superimposed text. There are two groups of binarization methods: global binarization and local binarization. Global binarization methods like that of Otsu, Iso Data, K-Means define a single threshold value for the whole image and are very fast [1]. They give good results for typical scanned documents, but are not appropriate for our goal to develop an efficient and fast thresholding for text extraction from image with complex background: for example Web pages or camera-captured documents; images with not uniform illumination; noised / degraded images; images with significant intensity changes.

Local binarization methods [2, 3, and 4] try to overcome problems of the complex background by computing thresholds individually for each pixel using information from the local neighborhood of the pixel. The achieved results are good, but they are often slow because the local neighborhood is computed for each pixel.

This paper presents a modified approach with a good performance for computing of adaptive local thresholds with speed close to the global thresholding methods.

The suggested algorithm has considerably lower computational complexity. The proposed adaptive threshold level computation gives smooth, faster image binarization and often it has better noise robustness. The "coarse to fine" approach combining Wiener filter [5], adaptive local thresholding and integral images [6, 7] is applied.

The following section (Sec. 2) describes the modified approach for combining integral images with the local adaptive thresholding techniques. The evaluation of the suggested algorithm is described in Sec. 3, followed by conclusion in Sec. 4.

2. Adaptive Local Binarisation Algorithm and Block Diagram

Integral image-based representation is applied in the modified algorithm, because it allows very fast multi-scale image processing.

An integral image $I(x, y)$ of an input grayscale image $I_g(x, y)$ 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:

$$I(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{x} I_g(x, y) \tag{1}$$

The integral image is computed in a single pass. A cumulative row sum $S(x, y)$ is used to calculate the integral image $I(x, y)$:

$$S(x, y) = S(x, y-1) + I_g(x, y) \tag{2}$$

$$I(x, y) = I(x-1, y) + S(x, y).$$

In Figure 1 a region A of the integral image can be computed using the following 4 array references:

$$A = (A_1 + A_4) - (A_2 + A_3) \tag{3}$$

If the integral image is the sum of pixels in a given area and it can be represented as two addition and one subtraction operations.

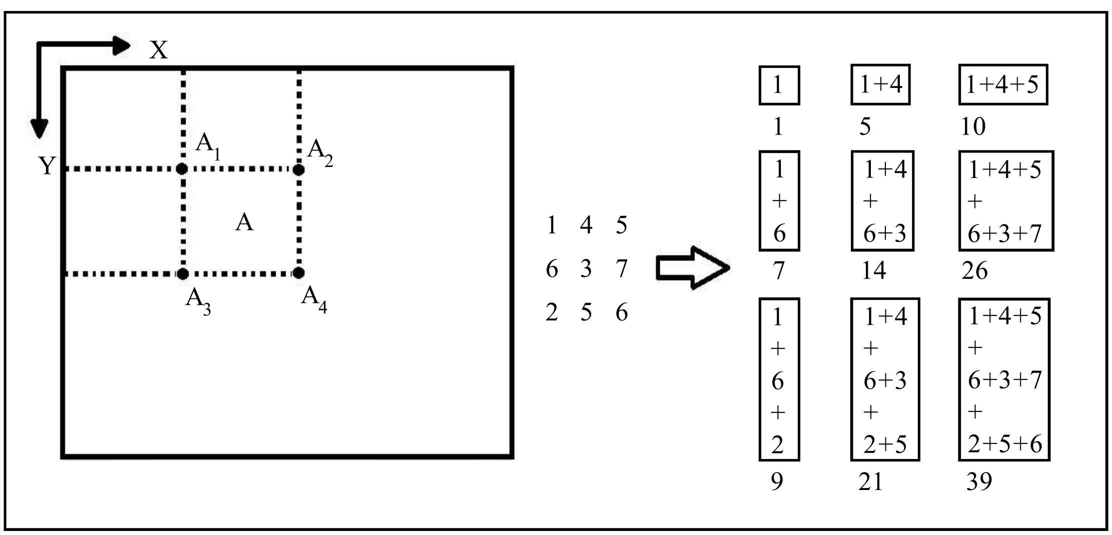


Figure 1. Integral Image Feature Computation

For example the sum of the 4 right-down pixels (3, 7, 5, 6) of the input grayscale image block is calculated faster from the elements of the integral image (right block in Fig. 1) without using the top row and the left column of the input image:

$$3 + 7 + 5 + 6 = (1 + 39) - (9 + 10)\ . \tag{4}$$

Then the local mean m(x, y) for any window size w is computed by using two addition and one subtraction operations instead of the summation over all pixel values in the window:

$$m(x, y) = \left(\begin{array}{c} (I(x + \frac{w}{2}, y + \frac{w}{2}) + (I(x - \frac{w}{2}, y - \frac{w}{2}) \\ - (I(x + \frac{w}{2}, y - \frac{w}{2}) - (I(x - \frac{w}{2}, y + \frac{w}{2}) \end{array} \right) / w^2 \tag{5}$$

Similarly, the local variance δ is computed very efficiently, independent of the local window size w:

$$\delta^2(x, y) = \frac{1}{w^2} \sum_{i=x-w/2}^{x+w/2} \sum_{j=y-w/2}^{j=y+w/2} I_g^{\,2}(i, j) - m^2(x, y). \tag{6}$$

An adaptive local binarization algorithm block diagram is presented in Figure 2.

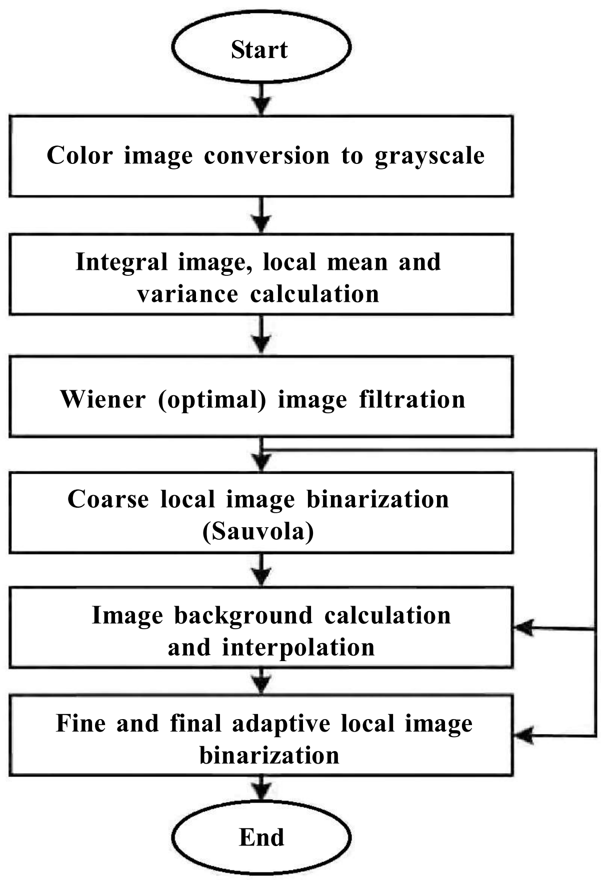


Figure 2. Local Binarization Algorithm Block Diagram

The main algorithm steps are the following:

step 1: *A color image conversion to grayscale image I_g is done, allowing color depth 2/24/32 bits:*

$$I_g = 0{,}3.R + 0.59.G + 0.11.B. \tag{7}$$

step 2: *Integral image, local mean and variance calculation is applied according to the above description.*

step 3: *Wiener (optimal) image filtration is selected for a preprocessing stage as more effective and faster, with low complexity and giving high accuracy of the OCR results. The Wiener filtered image is calculated:*

$$I_{wf}(x, y) = \frac{m + (\sigma^2 - \sigma^2_n) \, . \, (I_g(x, y) - m)}{\delta^2}, \tag{8}$$

where δ_n is a standard noise variance, defined from the whole image.

step 4: *Coarse local image binarization (Sauvola) is applied first, calculating a threshold for each pixel:*

$$\theta(x, y) = m(x, y)\left[1 + k\left(\frac{\delta(x, y)}{R} - 1\right)\right], \tag{9}$$

where k parameter has a value 0,2 and R is the defined in advance dynamic range of the standard variance δ. This algorithm uses the mean and the variance around the pixel in a local area, but threshold is better adaptively calculated for the background with changed intensity.

A pixel S(x, y) = 1 is accepted as part of the text object if its grayscale intensity I_g is less than the threshold θ:

$$\begin{aligned} &S(x, y) = 1 \quad \text{if} \quad I_g(x, y) < \theta(x, y) \\ &S(x, y) = 0, \quad \text{else} \end{aligned}, \tag{10}$$

else the pixel is part of the background S(x, y) = 0.

step 5: *Image background B_g calculation and interpolation is performed to obtain an image with background only – the previously selected text pixels now are interpolated with the closest intensity of the background pixels. The pixels in the text object are only changed in this step. The purpose is to achieve an equalized background for a more successful following finer binarization:*

$$B_g(x, y) = \begin{cases} I(x, y) & \text{if} \quad S(x, y) = 0 \\ \dfrac{\sum_{i=x-dx}^{x+dx} \sum_{j=y-dy}^{y+dy} (I(i, j) \, . \, (1 - S(i, j)))}{\sum_{i=x-dx}^{x+dx} \sum_{j=y-dy}^{y+dy} (1 - S(i, j)} & \text{if} \quad S(x, y) = 1 \end{cases}, \tag{11}$$

Journal of Multimedia Processing and Technologies Volume 15 Number 1 March 2024

where 2dx and 2dy are the size of the operator window (typically 40x40 px. to cover minimum two characters). If the pixel is part of the background S(x, y) = 0 it has the same unchanged intensity I(x, y).

step 6: *Fine and final adaptive local binarization over the grayscale input image is applied, taking in account the calculated background image. The subtraction between the background and the input image ($B_g(x, y) - I(x, y)$) gives an image with the replaced position parts of the text and background (darker) and the image can be assumed to be with a constant background.*

For the background the pixels $B_g(x, y)$ and $I(x, y)$ are equal and the background pixel intensities are not considered. The equation for achieving the fine and final binarized image is:

$$If(x, y) = \begin{cases} 1 & if \ (B_g(x, y) - I(x, y)) > d(B_g(x, y)) \\ 0 & else \end{cases} \tag{12}$$

where d(Bg(x, y)) is a local threshold for pixel (x, y).

For images with a constant background the value of the minimal threshold between text and background is:

$$d = q.\delta_f \tag{13}$$

where q is a weight coefficient, and δ_f is a mean distance between the background and the text and it is calculated as follows:

$$\delta_f = \frac{\sum_x \sum_y (B_g(x, y) - I - (x, y)}{\sum_x \sum_y S(x, y)} . \tag{14}$$

For achieving different adaptive thresholds for different contrasts between the text and background, the mean value b of the intensity of the grayscale pixels $B_g(x, y)$, belonging to the coarse binarized text, is calculated using the equation:

$$b = \frac{\sum_x \sum_y (B_g(x, y)(1 - S(x, y)))}{\sum_x \sum_y (1 - S(x, y))} . \tag{15}$$

The threshold for the given pixel (x, y) is equal to $q.\delta_f$ if the intensity of the image $B_g(x, y)$ is bigger than p1.b. Else the threshold is equal to p2.q. δ_f. The coefficients p1 and p2 are in the range [0; 1]. The above is achieved using the function:

$$d(B_g(x, y)) = q.\delta_f \left(\frac{1 - p2}{1 + exp\left(\frac{4B_g(x, y)}{b(1 - p1)} + \frac{2(1 - p1)}{1 - p1} \right)} + p2 \right) \tag{16}$$

3. Experiments and Results

The conducted tests include images with text over the picture, with darker background parts, and different background artifacts.

The algorithm implementation is done using program language C. During the experimental testing the selection of the window size w is done in Wiener filter (3 px.) for preserving the character contours, in the local area for Sauvola (15 px.) and in the local area for background calculation (40 px.).

In order to demonstrate the advantages (speed and ORC accuracy after text image binarization) of our proposed algorithm, 5 existing binarization algorithms (Iso Data, Otsu, Local Mean, Niblack, Sauvola) were tested.

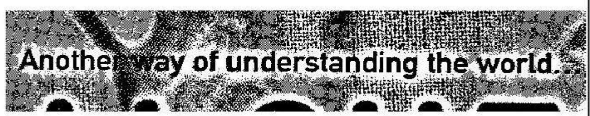
Used algorithm	Binarized image and Recognized text
Original	
Otsu	 Am 5. of understanding the world...
Niblack	 idini FI; - .___ ii: F
Sauvola	 Another ay of understanding the world...
Suggested algorithm	 Another way of understanding the world...

Figure 3. Original image, binarized by Otsu, Niblack, Sauvola and the suggested algorithm with the corresponding recognized texts

In Figure 3 is presented the original image with superimposed text on a complex background, the binarized images comparing the 4 methods - Otsu, Niblack, Sauvola and the suggested algorithm in this paper, and the recognized characters from OCR system (shown under the binarized images). After the Otsu global binarization 9 text symbols cannot be recognized. Using the Niblack binarization many black pixels pass from the background to the text object. This is the worst case and no one of 36 characters/ symbols are recognized. As it is shown in the Sauvola algorithm, binarization results in the threshold being higher than the optimal and 3 characters are almost unreadable in the binary image, but the OCR system helps and only one symbol is not recognized. In the suggested *by us adaptive local binarization algorithm, the text is maximally readable and all 36 characters are recognized.*

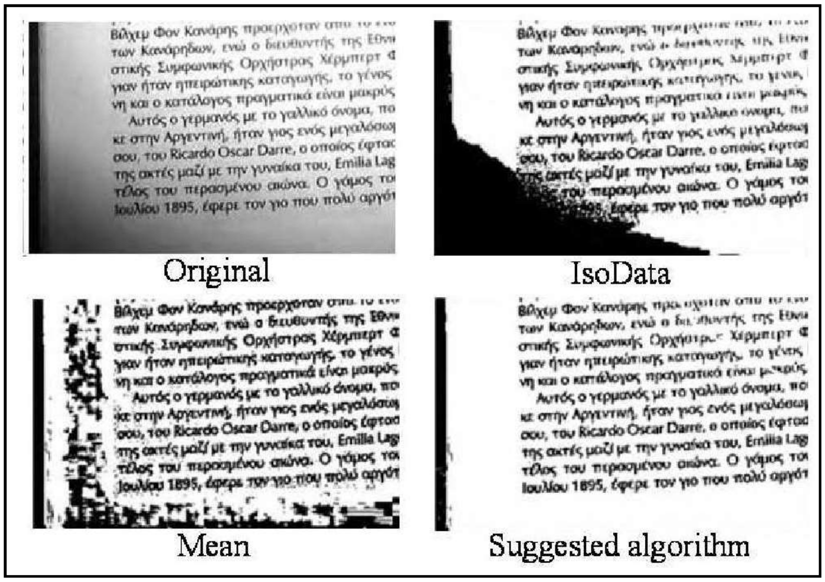


Figure 4. Original image, binarized by IsoData, Mean and the suggested algorithm

The application of the proposed algorithm, as well as IsoData and Mean algorithms, on test images with dark areas due to bad scanning shows that the suggested approach better cleans such backgrounds (Figure 4).

The typical binarization algorithm calculates local image areas sequentially around each pixel. The calculation complexity presented as a number of operations O is:

$$C = O\,(N^2.W^2)\,, \qquad (17)$$

where N_2 is the area of the whole image in pixels, W^2 is the area of the operator window processing each pixel. We allow maximal image 4000 x 3000 px., an operator window W^2= 20 x 20 px. In this case the number of the arithmetical operation is 4.9e+9 . This speed is equal to 5 sec. for CPU with 20000 MIPS.

Using the integral image approach in the proposed algorithm the arithmetical operations are reduced to

$$C_I = 3(N^2) \qquad (18)$$

The achieved speed of the suggested algorithm with the applying integral images is 3 sec. for the same CPU, that means 1,7 times faster only for one algorithm step.

4. Conclusion

An adaptive local binarization algorithm was proposed for text images with complex background. This algorithm is appropriate for document images that are difficult to be recognized correctly directly as grayscale or after using a global thresholding like Otsu. After the applying of different binarization algorithms to the real images with text and complex background, it is easy to evaluate the results by comparing the recognized texts in images.

In this paper we presented a modified algorithm of the adaptive computing thresholds for local image binarization.

The new approach in this paper is the application of integral images in the Wiener filter in order to compute the mean and the variance in the local processing window. This results in increasing the speed of the thresholding algorithm regardless of the local processing window size. The second new approach in the presented algorithm is the computation of the Sauvola threshold function with integral images in the stage of coarse binarization in combination with the Wiener filter. The proposed algorithm includes a fine adaptive binarization technique that achieves better thresholding compared to Sauvola, resulting in a higher recognition accuracy of the image presented in Figure 3. As a result all text characters are recognized correctly after applying the described binarization algorithm. An additional advantage of the developed algorithm is the reduced time of the binarization process, compared to the other local thresholding algorithms. The decreased time is close to the fast global binarization schemes like Otsu.

Acknowledgement

This paper was supported by the National Science Fund of the Bulgarian Ministry of Education, Youth and Science (Contract – DDVU 02/13 – " Public and Private Multimedia Network Throughput Increase by Creating Methods for Assessment, Control and Traffic Optimization")

References

[1] Liang, X. (2009). Image Binarization using Otsu Method. NLPR-PAL Group, CASIA.

[2] Vonikakis, V., Andreadis, I., Papamarkos, N., Gasteratos, A. (2007). Adaptive Document Binarization: A human vision approach. In *2nd International Conference on Computer Vision Theory and Applications (VISAPP)*, Barcelona.

[3] Sauvola, J., Pietikainen, M. (2000). Adaptive Document Image Binarization. *Pattern Recognition*, 33(2), 225-236.

[4] Pavlos, S., Kavallieratou, E., Papamarkos, N. (2008). An Evaluation Technique for Binarization Algorithms. *Journal of Universal Computer Science*, 14(18), 3011-3030.

[5] Smith, E., Likforman-Sulem, L., Darbon, J. (2010). Effect of pre- processing on binarization. *In Proceedings SPIE Electronic Imaging Document Recognition and Retrieval*, 7534, 75340H-75340H-8.

[6] Derpanis. (2007). Integral image-based representations. *Department of Computer Science and Engineering*, York University.

[7] Shafait, F., Keysers, D., Breuel, T. M. (2008). Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images. In: P*roceedings of 15th International Conference on Document Recognition and Retrieval*, 6815, 81510, San Jose, CA, USA.