Pattern Recognition 43 (2010) 3177–3187

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Adaptive thresholding algorithm: Efficient computation technique based on intelligent block detection for degraded document images

Yu-Ting Pai, Yi-Fan Chang, Shanq-Jang Ruan [*]

Department of Electronic Engineering, National Taiwan University of Science and Technology, Taiwan

A R T I C L E I N F O

Article history:
Received 28 August 2008
Received in revised form
16 March 2010
Accepted 20 March 2010

Keywords:
Document image analysis
Document image binarization
Adaptive thresholding
High speed
Low computational cost

A B S T R A C T

Document image binarization involves converting gray level images into binary images, which is a feature that has significantly impacted many portable devices in recent years, including PDAs and mobile camera phones. Given the limited memory space and the computational power of portable devices, reducing the computational complexity of an embedded system is of priority concern. This work presents an efficient document image binarization algorithm with low computational complexity and high performance. Integrating the advantages of global and local methods allows the proposed algorithm to divide the document image into several regions. A threshold surface is then constructed based on the diversity and the intensity of each region to derive the binary image. Experimental results demonstrate the effectiveness of the proposed method in providing a promising binarization outcome and low computational cost.

1. Introduction

Capable of distinguishing characters from background regions, document image binarization involves converting gray level images into binary images. Among its many applications as a character location method include optical character recognition (OCR), automatic bank check processing, and signature verification [1,2]. Binarization has received considerable intense over the past three decades. Although many algorithms have been proposed for the uniform images, binarization is rather complex when processing degraded document images. Many degradations, including complex backgrounds, non-uniform intensity, and shadows, are attributed to the acquisition source type and imaging environment. To resolve these problems, several binarization algorithms have been developed for degraded document images.

In current techniques, binarization can be classified into global and local thresholding approaches. In global thresholding methods, a single threshold is estimated and applied to the whole image. While comparing more than 20 global thresholding algorithms using uniformity or shape measures, Sahoo et al. [3] demonstrated the effectiveness of the methods proposed by Otsu [4], Tsai [5], Johannsen [6], and Kapur [7] in yielding adequate binarization results. Nevertheless, these methods failed to obtain acceptable results when the source image contains various background patterns or inhomogeneous backgrounds. To overcome this limitation, Liu and Srihari [8] devised a thresholding algorithm based on texture features. That work obtained a set of candidate thresholds by using Otsu's algorithm. Texture features were then measured from each threshold image based on the best threshold selected. In 1998, Cheriet [9] developed a recursive procedure by extending Otsu's algorithm. The recursive process continues until one object is left in the image. Despite these enhance methods have been proposed, difficulties persist when the foreground of the histogram overlaps with the background. Thus, many investigators tried to overcome this problem by using local methods.

In local thresholding methods, the threshold is determined for each pixel based on its own gray value and the gray values of its neighborhood. Hence, these approaches have also been called the adaptive thresholding algorithms. Trier and Jain [10] evaluated how eleven well-established local thresholding methods performed, indicating that Niblack's [11] method is apparently the most effective. However, Niblack's method is ineffective when the background contains light texture. In 2000, Sauvola [12] devised a method that modifies Niblack's method by hypothesizing on the gray values of text and background pixels. In 2004, Sezgin and Sankur [13] compared 40 binarization methods, indicating that the local based approach of Sauvola and the method of Trier [14] perform the best in terms of document binarization methods. Recently, Gatos et al. [15] devised another local binarization method for document images of poor quality. By using Sauvola's method, that work detected foreground parts and followed this by several post-processing steps to improve the final binarization

[*] Corresponding author. Tel.: +886 2 27376411; fax: +886 2 27376424.
E-mail address: sjruan@mail.ntust.edu.tw (S.-J. Ruan).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.03.014

results. Despite offering satisfactory results for degraded documents, the above local approaches have a high computational cost because each pixel must be estimated to its threshold. Therefore, these local methods expend a prohibitively high amount of processing time when thresholding the document images.

Document image binarization is extensively adopted in portable devices such as personal digital assistant (PDA), digital cameras and mobile camera-phones [16,17]. Given the limited memory space and computational power of portable devices, reducing the computational complexity in a process or a system is of priority concern. For instance, an OCR software can be developed for mobile camera phones in which image sizes are at least one megapixels. To recognize characters from a camera-captured document image, binary image quality is critical to the application performance. However, no method can provide acceptable results efficiently. Given the merits and limitations of each method, document image binarization is a trade-off between processing complexity and binarization quality.

As mentioned earlier, most binarization methods have been developed with respect to quality only, regardless of their complexity. This work presents an efficient and effective binarization algorithm with intelligent block size detection to process document images efficiently. Initially, the character region and the background region in the image are separated automatically. Then, according to the height of each region, the image is divided into several blocks with various sizes. Finally, a threshold surface is constructed based on each block to obtain the binary image. Experimental results demonstrate that the proposed method has a low computational complexity and achieves promising results.

The rest of this paper is organized as follows. Section 2 introduces pertinent literature on image binarization algorithms. Section 3 then presents the proposed method. Next, Section 4 summarizes the experimental results. Conclusions are finally drawn in Section 5, along with recommendations for future research.

2. Review of binarization methods

This section describes a global thresholding method and three local thresholding methods. The binarization methods are either developed recently or promising experimental, explaining why they are still compared in current approaches.

2.1. Otsu's algorithm

Let the pixels of a given image be represented in L gray levels $[1,2,\ldots,L]$. The number of pixels at level i is denoted by z_i and the total number of pixels can be expressed as $Z=z_1+z_2+\cdots+z_L$. The gray-level histogram is normalized and regarded as a probability distribution:

$$p(i)=z_i/Z, z_i \geq 0, \quad \sum_{i=1}^{L} z_i=1. \tag{1}$$

The image pixels are separated into two classes C_0 and C_1 (e.g. objects and background) by a threshold T_o. Where C_0 denotes pixels with levels $[1,\ldots,T_o]$, and C_1 represents pixels with levels $[T_o+1,\ldots,L]$. The probabilities of class occurrence and the class mean levels can then be written as

$$\omega_0=\sum_{i=1}^{T_o} p(i)=\omega(T) \tag{2}$$

$$\omega_1=\sum_{i=T_o+1}^{L} p(i)=1-\omega(T) \tag{3}$$

and

$$\mu_0=\sum_{i=1}^{T_o} \frac{ip(i)}{\omega_0} \tag{4}$$

$$\mu_1=\sum_{i=T_o+1}^{L} \frac{ip(i)}{\omega_1} \tag{5}$$

$$\mu_T=\sum_{i=1}^{L} ip(i)=\omega_0\mu_0+\omega_1\mu_1 \tag{6}$$

where ω_0 and ω_1 refer to probabilities of object class and background class, respectively. Besides, μ_0, μ_1, and μ_T denote the average gray levels of the object, the background, and the whole image. The class variances are given by

$$\sigma_0^2=\sum_{i=1}^{T_o} \frac{(i-\mu_0)^2 p(i)}{\omega_0} \tag{7}$$

$$\sigma_1^2=\sum_{i=T_o+1}^{L} \frac{(i-\mu_1)^2 p(i)}{\omega_1} \tag{8}$$

$$\sigma_W^2=\omega_0\sigma_0+\omega_1\sigma_1 \tag{9}$$

$$\sigma_B^2=\omega_0(\mu_0-\mu_T)^2+\omega_1(\mu_1-\mu_T)^2 \tag{10}$$

$$\sigma_T^2=\sum_{i=1}^{L}(i-\mu_T)^2 p(i) \tag{11}$$

where σ_0^2 denotes the variance of the object class and σ_1^2 refers to the variance of the background. Additionally, σ_W^2, σ_B^2 and σ_T^2 represent the within-class variance, the between-class variance, and the total variance of levels, respectively. In the discrimination analysis, the separable degree η of the class is

$$\eta=\max_{1 \leq T \leq L} \frac{\sigma_B^2}{\sigma_W^2}. \tag{12}$$

Finally, the threshold T_{ot}, resulting in maximal η is selected as the optimal threshold.

2.2. Niblack's algorithm

Niblack [11] proposed an algorithm that calculates a pixelwise threshold by shifting a rectangular window across the image. This method varies the threshold over the image, based on the local mean and local standard deviation. Let the local area be $b \times b$. Also, the threshold $T_{ni}(x,y)$ at pixel $f(x,y)$ is determined by the following equation:

$$T_{ni}(x,y)=\mu_{ni}(x,y)+k_{ni}\cdot\sigma_{ni}^2(x,y) \tag{13}$$

where

$$\mu_{ni}(x,y)=\frac{1}{b^2}\left[\sum_{j=y-\frac{b}{2}}^{y+\frac{b}{2}}\left(\sum_{i=x-\frac{b}{2}}^{x+\frac{b}{2}} f(i,j)\right)\right] \tag{14}$$

$$\sigma_{ni}^2(x,y)=\frac{1}{b^2}\left\{\sum_{j=y-\frac{b}{2}}^{y+\frac{b}{2}}\left[\sum_{i=x-\frac{b}{2}}^{x+\frac{b}{2}}(\mu_{ni}(x,y)-f(i,j))^2\right]\right\}. \tag{15}$$

As in the definition of Eqs. (14) and (15), $\mu_{ni}(x,y)$ and $\sigma_{ni}^2(x,y)$ are the local mean and standard deviation values of a local area. The size of the local window, b, should be small enough to accurately reflect the local illumination level and adequately large to include both objects and the background. Trier and Jain [10] recommend taking a 15×15 neighborhood and $k_{ni}=-0.2$. Thus, Niblack's algorithm cannot be applied to the varying resolution input images.

2.3. Sauvola's algorithm

Sauvola and Pietikainen [12] devised a method that solves Niblack's problem by hypothesizing on the gray values of text and background pixels, resulting in the following formula for the threshold:

$$T_{sa}(x,y)=\mu_{sa}(x,y)+\left(1-k_{sa}\left(1-\frac{\sigma_{sa}^2(x,y)}{R}\right)\right) \tag{16}$$

where $\mu_{sa}(x,y)$ and $\sigma_{sa}^2(x,y)$ are the same as in the previous method, R denotes the dynamics of the standard deviation fixed to 128 and k_{sa} refers to a fixed value usually set to 0.5.

2.4. Gato's algorithm

Gatos et al. [15] developed a method using a combination of existing approaches. By using a pre-processing Wiener filter, that work attempted to eliminate noisy areas and contrast enhancement between background and text areas. The results after Wiener filter are denote as

$$I_{ga}(x,y)=\mu_{ga}+\frac{(\sigma_{ga}^2-\upsilon_{ga}^2)(f(x,y)-\mu_{ga})}{\sigma_{ga}^2} \tag{17}$$

where $f(x,y)$ refers to the gray level source image, μ_{ga} denotes the local mean, σ_{ga}^2 represents the variance at a 3×3 neighborhood around each pixel and υ_{ga}^2 denotes the average of all estimated variances for each pixel in the neighborhood.

After the Wiener filter process, Sauvola's approach is adopted to roughly estimate foreground regions. According to Eq. (16), the binarization results of Sauvolas's method are denoted as

$$R_{sa}(x,y)=\begin{cases}1 & \text{if } f(x,y)>T_{sa}(x,y)\\ 0 & \text{otherwise}\end{cases} \tag{18}$$

where $R_{sa}(x,y)=0$ denotes the estimated foreground. A background surface $B_{ga}(x,y)$ is then computed by interpolating neighboring background intensities. The background surface $B_{ga}(x,y)$ is calculated as

$$B_{ga}(x,y)=\begin{cases}I_{ga}(x,y) & \text{if } R_{sa}(x,y)=0\\ G_{ga}(x,y) & \text{if } R_{sa}(x,y)=1\end{cases} \tag{19}$$

where

$$G_{ga}(x,y)=\frac{\sum_{i=x-d_x-1}^{x+d_x}\sum_{j=y-d_y-1}^{y+d_y}(I_{ga}(i,j)(1-R_{sa}(i,j)))}{\sum_{i=x-d_x-1}^{x+d_x}\sum_{j=y-d_y-1}^{y+d_y}(1-R_{sa}(i,j))}. \tag{20}$$

The interpolation window of size $2d_x\times 2d_y$ is defined to cover at least two image characters.

Thereafter, the binary image $R_{ga}(x,y)$ is given by integrating the background surface $B(x,y)$ with the pre-processed image $I(x,y)$ as follows:

$$R_{ga}(x,y)=\begin{cases}1 & \text{if } B_{ga}(x,y)-I_{ga}(x,y)>T_d(B_{ga}(x,y))\\ 0 & \text{otherwise}\end{cases} \tag{21}$$

where $T_d(\cdot)$ refers to the threshold that must be modified according to the gray-scale value of the background surface $B_{ga}(x,y)$. Finally, the enhanced binary image is obtained using three post-processing steps to upgrade the quality of character regions and preserve stroke connectivity.

2.5. Analysis of binarization methods

Since the threshold T_{sa} and T_{ni} must be inside a fixed square block, Niblack's and Sauvola's methods are scale dependent. Thus, the two methods do not work properly if the character size varies significantly along the document image. Moreover, the two methods are time consuming because they must calculate the intensity mean and variance for each pixel. In Gatos's method, no explicit scale factor is available, explaining why this method is more feasible for normal circumstances. However, computation for constructing the threshold surface is highly complex. Although selecting a single threshold is simpler and easier in Otsu's global method than in Gatos's method, the former fails to yield acceptable results if the document images contain degradations.

To compensate for the limitations of these four methods, a method must be developed to achieve a balance between low computational complexity and high quality of document image binarization. To achieve this objective, the binarization algorithm should be based on the global trends and the local details. Thus, the block size must be estimated intelligently according to the document image characteristics.

3. Methodology

As discussed in Section 2.5, limitations in memory space and computational power of portable devices necessitate the development of a binarization algorithm with a low computational complexity and a high quality document image. Therefore, Section 3.1 introduces a simple and efficient binarization algorithm based on intelligent block size detection. Section 3.2 then analyzes the computational complexity of the proposed method and other typically methods to demonstrate the efficiency of the proposed method. Moreover, the proposed algorithm is described by assuming that $f(x,y)$ refers to an original gray level image with size $M\times N$, where (x,y) represents the coordinate of each pixel.

3.1. Intelligent block based image binarization algorithm

The proposed method combines the advantages of global (high speed) and local (high accuracy) methods. According to Fig. 1, this method consists of two major steps, i.e. block detection and image binarization. The former step divides the document image into a number of blocks of varying sizes that reflect local adaptive characteristics in the document image. Following image segmentation, the binary image can be obtained by using the global binarization method. The detailed treatments for these two steps are described below.

3.1.1. Block detection

For degraded document images, the first stage separates the region of the character from the background. Assume that all input images are rotated to straight document image by pre-processing. Additionally, the characteristic of input images can be extracted using horizontal projection [18]. The horizontal histogram $H(y)$ is denoted as

$$H(y)=\sum_{x=0}^{M-1}f(x,y). \tag{22}$$

Fig. 2 shows an example of horizontal histogram. The gray color histogram refers to $H(y)$. For a document image, the horizontal histogram can be regarded as a non-period square wave. The character pixel value must normally be lower than the background pixel value in a document image. Thus, the valleys and the peaks of the non-period square wave $H(y)$ are the

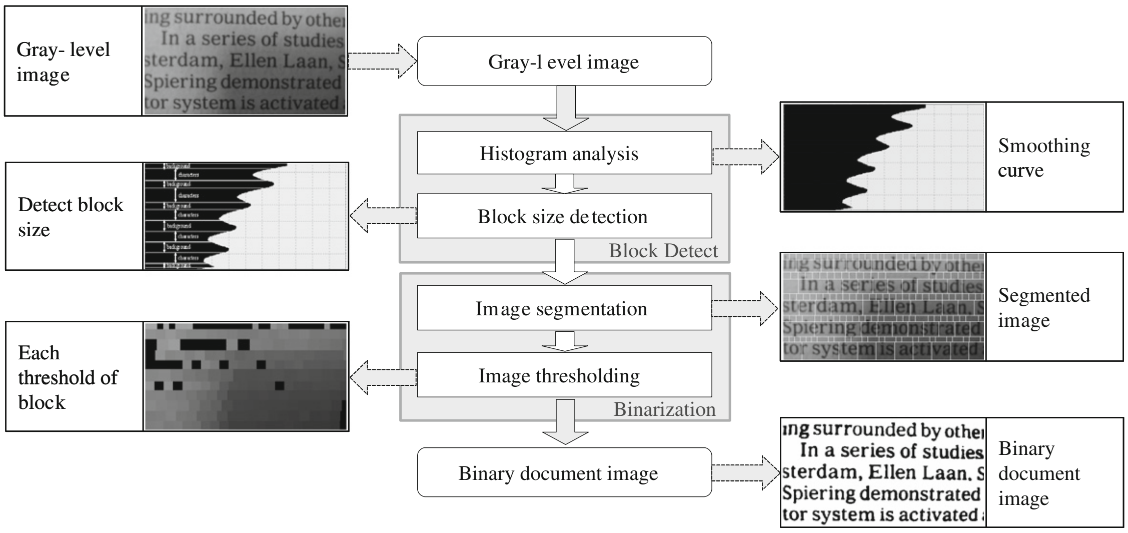


Fig. 1. Flowchart of the proposed method.

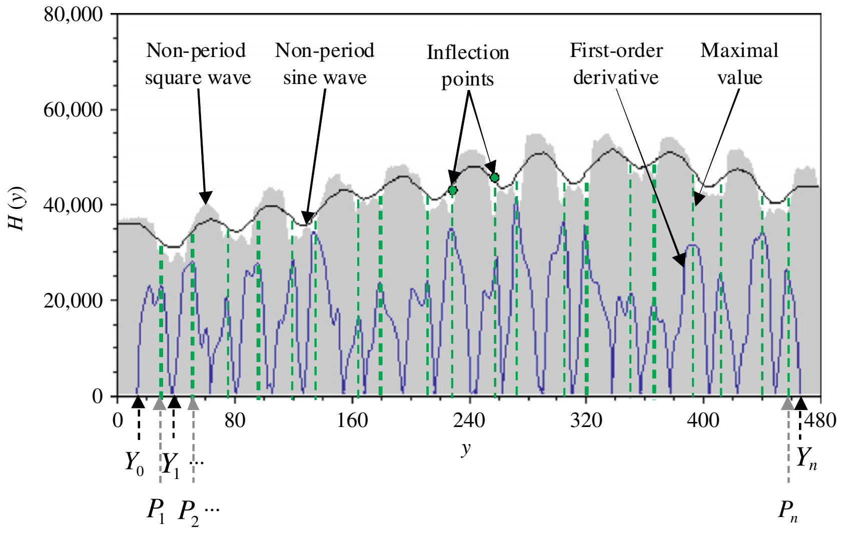


Fig. 2. The methodology of the proposed blocks detection.

accumulation of the character and the background pixel values, respectively. Obviously, the character region can be separated from the background region. However, the document image is either occasionally poorly contrasted or contaminated by noise from various sources that cause many noises in the non-period square wave. Consequently, the separation processing fails when the noise is intense. To eliminate these noises and preserve the critical information, a smoothing filter [19] is adopted to transform the non-period square wave into the non-period sine wave $H_s(y)$ as given below

$$H_s(y) = \frac{1}{m}\left(\sum_{k=y-\lfloor m/2\rfloor}^{y+\lceil m/2\rceil-1} H(k)\right) \tag{23}$$

where m denotes the size of the smoothing window. The window size is defined by determining the approximate period τ of non-period square wave $H(y)$ based on the following steps:

1. Let $R_\tau(y)$ be the periodic square wave with period τ and can be denoted as

$$R_\tau(y) = \max[H(y)] \cdot \operatorname{sgn}\left[\sin\left(\frac{2\pi y}{\tau}\right)+1\right]. \tag{24}$$

2. The variance of $R_\tau(y)$ and $H(y)$ is given by

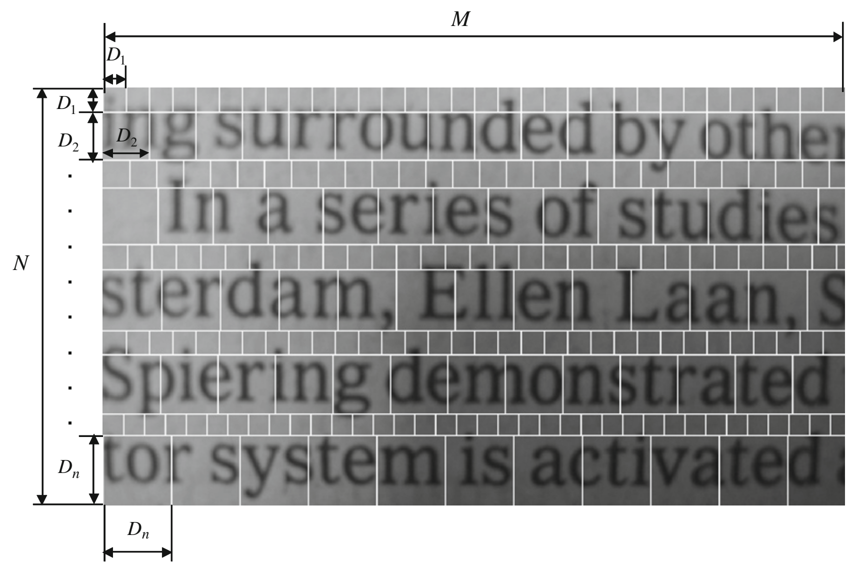
$$\sigma_\xi^2 = \frac{1}{N}\sum_{y=0}^{N-1}[R_\tau(y) - H(y)]^2 \tag{25}$$

where $\xi = \frac{1}{2}\tau$ and $0 < \xi < N/2$.

3. The window size m can be defined as the value ξ which results in the minimal variance σ_ξ^2.

According to Fig. 2, the non-period sine wave is smoother than the non-period square wave.

After the non-period sine wave $H_s(y)$ is derived, each inflection point is the boundary between the character and the background region. Hence, the distance between each two inflection points is the height of the detected block. The inflection point can be determined by calculating the extremum, i.e. inclusive of the maximum or minimum values, in the first-order derivative or the zero-crossing in the second-order derivative. Given that finding a zero-crossing in the second-order derivative is much complex than the extremum in the first-order derivative, the inflection point is obtained using the first-order derivative function. The first-order derivative function of the non-period sine wave is



ng surrounded by other
In a series of studies
sterdam, Ellen Laan, S
Spiering demonstrated
tor system is activated

Fig. 3. The methodology of the proposed image segmentation.

shown as

$$H'_s(y) = \frac{dH_s(y)}{dy} = \frac{H_s(y+1) - H_s(y)}{(y+1) - y} = H_s(y+1) - H_s(y). \tag{26}$$

According to Fig. 2, a sequence $\mathbf{Y} = \{Y_j | 0 < j < n\}$ is defined as the points in which the value of first-order derivative is zero. The inflection point $\mathbf{P} = \{P_j | 1 < j < n\}$ can than be obtained between Y_j and Y_{j+1} and is denoted as follows:

$$P_j = \max_{Y_j \le y \le Y_j + 1} \{|H'_s|\}. \tag{27}$$

After all the inflection points are derived from the non-period sine wave, distance $\mathbf{D} = \{D_j | 1 < j < n\}$ can be computed by

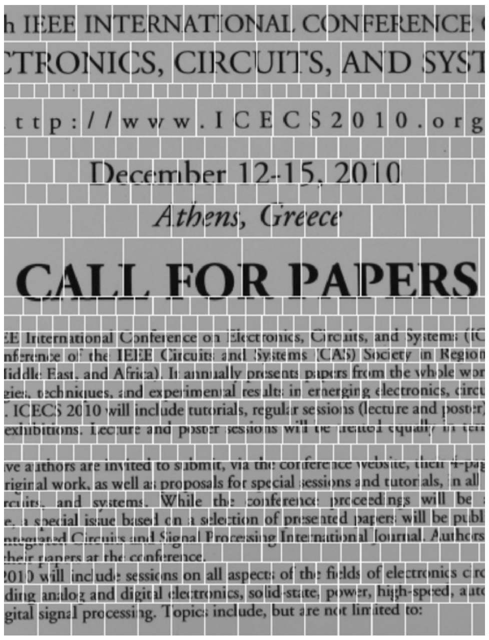
$$D_j = P_{j+1} - P_j. \tag{28}$$

Hence, distance D_j must be the height of the character region or the background region. According to the distance D_j, the document image can be segmented into several blocks. Fig. 3 illustrates the behavioral segmentation process. Let the entire image region be separated into $n-1$ strip sub-regions R_1 to R_{n-1}. Each strip sub-region consists of $D_j \times M$ pixels. Each strip sub-region is then sub-divided into a set of blocks. Each block consists of $D_j \times D_j$ pixels. Notably, the block size between each strip sub-region is varied according to the distance D_j. Fig. 4 shows another example of the document image segmentation, indicating that the document is filled with varying size characters. The document image can still be divided into various size blocks using the proposed segmentation process.

Moreover, our assumption that inputs are straight images does not imply that the input images should be processed by skew angle detection and orientation correction. According to [20], the horizontal histograms of rotated images still have many spurious local maxima and minima. Therefore, the proposed block detection can also be applied to skew images. The experiment introduced in Section 4.2 demonstrates the effectiveness of the proposed method in skew images.

3.1.2. Image binarization

After image segmentation, the binary image is obtained by thresholding the amount of blocks. The local mean μ and the local standard deviation s are measured for each $D_j \times D_j$ block and are used for determining whether the block is required to evaluate an adaptive threshold. For a lower standard deviation and a higher mean, $s < s_T$ and $\mu > \mu_T$, the block is assumed to contain only background pixels. Hence, each pixel in the block is substituted by the background. In contrast, the block is binarized using a threshold. In thresholding of the block, Otsu's algorithm is used for automatic selection of an optimal threshold value. For a pixel in the block with a higher gray level than threshold, the pixel is labeled as background; otherwise it is labeled as character.



h IEEE INTERNATIONAL CONFERENCE
CTRONICS, CIRCUITS, AND SYST
ttp://www.ICECS2010.org
December 12-15, 2010
Athens, Greece
CALL FOR PAPERS
E International Conference on Electronics, Circuits, and Systems (IC
nference of the IEEE Circuits and Systems (CAS) Society in Region
iddle East, and Africa). It annually presents papers from the whole wor
gies, techniques, and experimental results in emerging electronics, circu
. ICECS 2010 will include tutorials, regular sessions (lecture and poster)
exhibitions. Lecture and poster sessions will be treated equally in ter
ve authors are invited to submit, via the conference website, their 4-pa
riginal work, as well as proposals for special sessions and tutorials, in all
ircuits and systems. While the conference proceedings will be
e. A special issue based on a selection of presented papers will be publ
ntegrated Circuits and Signal Processing International Journal. Authors
their papers at the conference.
2010 will include sessions on all aspects of the fields of electronics c
ding analog and digital electronics, solid-state, power, high-speed, a
gital signal processing. Topics include, but are not limited to:

Fig. 4. A segmentation example of varying size words.

The local standard deviation and the local mean represent the diversity and the intensity of the block, explaining why they are regarded as the condition for judgment. Moreover, this condition can further reduce a significant amount of computation to achieve a low computational complexity. Following experimental work, for the case of document images, the parameter values of $s_T = 10$ and $\mu_T = 32$.

Table 1
Computational complexity of Otsu's method.

Procedure	**Equation(s)**	**Computation**
Histogram	(1)	$MN+L$
Separated class mean	(2)–(5)	$(3L-2)(L-1)$
Separated variance	(6)–(8)	$(4L+3)(L-1)$
Maximal separable degree	(10)–(12)	$11(L-1)$
Total	(1)–(12)	$MN+7L^2+5L-12$

Table 2
Computational complexity of Niblack's method.

Procedure	**Equation(s)**	**Computation**
Mean of neighborhood	(14)	$(b^2)(MN)$
Variance of neighborhood	(15)	$(3b^2+1)(MN)$
Denote threshold	(13)	$2(MN)$
Total	(13)–(15)	$(4b^2+3)MN$

Table 3
Computational complexity of the proposed method.

Procedure	**Equation(s)**	**Computation**
Horizontal histogram	(22)	$N(M-1)$
Smoothing window	(24) and (25)	$\frac{N}{2}(3N-1)$
Smooth $H(y)$	(23)	$N(m-1)$
Block size detection	(26)–(28)	$2N-1+\frac{N}{m}$
Otsut's operation	(1)–(12)	$MN+\frac{MN}{m^2}(7L^2+5L-12)$
Total	(1)–(12), (22)–(28)	$\frac{3}{2}N^2+(2M+m+\frac{1}{m}-\frac{3}{2})N$ $-1+\frac{MN}{m^2}(7L^2+5L-12)$

3.2. *Comparison of the proposed method with previous one in terms of computational complexity*

This section compares the arithmetic computational complexity of the proposed and previous methods to demonstrate the efficiency of the proposed method. Here, Otsu's and Niblack's approaches are considered owing to their lower computation in global and local thresholding methods, respectively. For an image of size $M \times N$ with L gray levels, Tables 1–3 describe the computational complexity required at various stages of the three methods.

Table 1 summarizes the computational complexity of Otsu's method according to the discussion in Section 2.1. The computational cost is defined as follows: MN is used for obtaining the histogram of the image. $7L^2+5L-12$ is used for estimating the probability, mean, and variance whenever the histogram is dichotomized into two classes. Therefore, the total computational complexity of Otsu's method is $MN+7L_2+5L-12$.

Table 2 displays the computational complexity of Niblack's method derived from Section 2.2. Notably b denotes the block size of neighborhood. In Niblack's method, the threshold is computed by estimating the local mean and standard deviation inside a fixed block for each pixel. Hence, each equation in Section 2.2 should be performed $M \times N$ times. The total computational complexity is expensive: $(4b^2+3)MN$.

Table 3 describes the computational complexity of the proposed method. First, Eq. (22) presents the projection of horizontal histogram. The computational cost is $M-1$ for each row, explaining why the entire image is computed as $N(M-1)$. Second, the smoothing window m is denoted in Eqs.(24) and (25).

Table 4
Computation of Otsu's, Niblack's and the proposed methods.

Method	Image resolution 640×480	1280×960	2560×1920	3072×2304	3648×2736
Otsu	767	1689	5375	7538	10,441
Niblack	277,402	1,109,606	4,438,426	6,391,333	9,012,778
Proposed[#1]	3528	6754	19,656	27,225	37,384
Proposed[#2]	10,831	13,884	26,094	33,258	42,874
Proposed[#3]	62,319	65,268	77,064	83,984	87,196
Proposed[#4]	157,994	631,949	2,527,739	3,639,931	5,127,158

Notice 1: Proposed[#1], proposed[#2], proposed[#3], and proposed[#4] mean that $m=N/2$, $m=N/4$, $m=N/10$, and $m=30$ of the proposed method. **Notice 2**: The unit of this table is 1000.

Because the image length is N, $N/2$ kinds of square wave should be compared with $H(y)$ to derive the optimal m. Thus, the computational cost is $N/2(3N-1)$. Third, in Eq. (23), the computational cost of smooth processing is $N(m-1)$. Fourth, for obtaining the block size D_j, the computational of Eqs. (26)–(28) is $2N-1+(N/m)$. Notably we assume that the average value of D_j is approximated as m, since the average detected block size is rather close to the smoothing window m. Finally, an optimal threshold is determined using Otsu's method to obtain the binary image. Hence, the computational cost of a block is $MN+(MN/m^2)(7L^2-11L+4)$. Consequently, the total computational complexity of the proposed method is $\frac{3}{2}N^2+(2M+m+(1/m)-\frac{3}{2})N-1+(MN/m^2)(7L^2+5L-12)$.

Table 4 compares the results of the three methods with respect to arithmetical computation. For general images, the gray level L is 256. The values of M and N are defined in the first row. The discussed image resolutions are thus 0.3, 1, 5, 7, and 10 M pixels. The second and third rows show the Otsut's and Niblack's methods, respectively. The remaining four rows represent the proposed approach when $m=N/2$, $N/4$, $N/10$, and 30. Note that the unit of this table is 1000. According to this table, for the best case, where $m=N/2$, the image is divided into four entire blocks and two fragmented regions. Hence, the computational cost of the proposed method is about five times of Otsu's method. For a general case, where $m=N/4$ and $m=N/10$, the computational costs are great smaller than those of Niblack's method. For the worst case, when $m=30$, the character size is not smaller than 15 pixels for most of real document images. To cover at least 1 to 2 characters, the smallest block size is defined by 30. Obviously, for the worst case, the computational cost of the proposed method is still only 57% that of Niblack's method. Fig. 5 illustrates the behavior of Table 4, indicating that each case of the proposed method has a lower computation cost than that of Niblack's method. In the general case, when $m=N/4$ and $m=N/10$, the proposed method tends to resemble Otsu's curve more than Niblack's one does. Consequently, mathematical analysis clearly reveals that the computational complexity of our approach is between that of the global and the local methods. The next section evaluates and compares the processing times of a global, three local, and the proposed methods to verify the inference of this section.

4. Experimental results

This section summarizes the experimental results of the four document images shown in Figs.7,8,9,and 10(a). These document images are 640×480 pixels with 8-bit gray levels and are obtained from papers, newspapers and magazines, respectively. Five thresholding algorithms including Otsu's, Niblack's, Sauvola's, Gatos's and the proposed algorithms are implemented

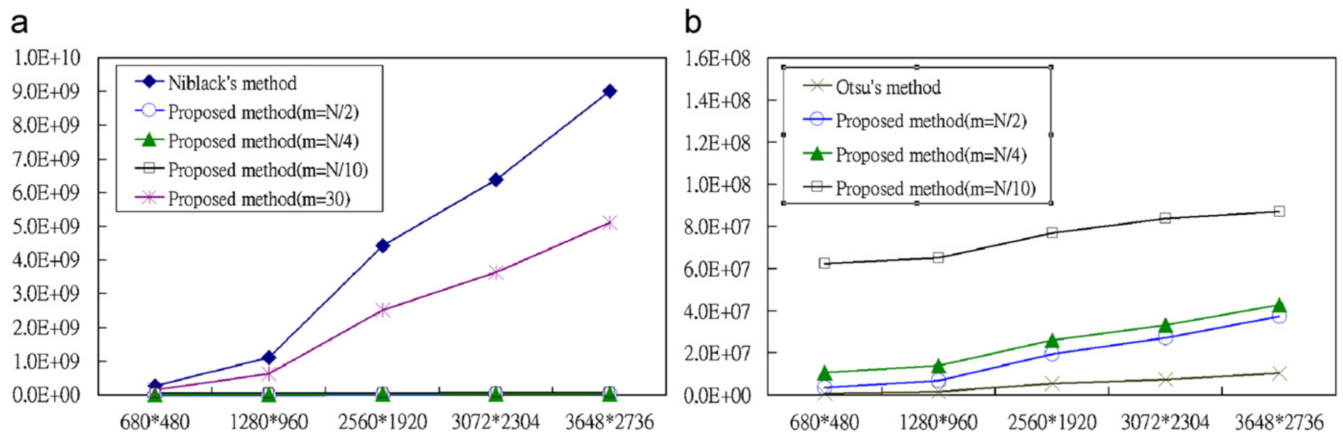


Fig. 5. (a) The tendency of computation upon various image resolutions among Niblack's and the proposed methods ($m=N/2$, $N/4$, N/10,and 30), (b) The tendency of computation upon various image resolution among Otsu's and the proposed methods ($m=N/2$, $N/4$,and $N/10$).

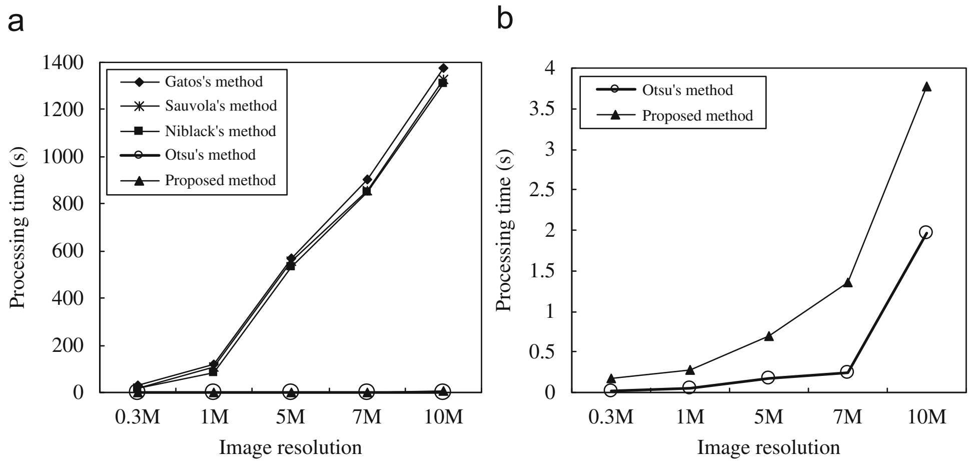


Fig. 6. (a) The tendency of processing time upon various image resolutions among Gatos's, Sauvola's, Niblack's, Otsu's, and the proposed methods, (b) The tendency of processing time upon various image resolution between Otsu's and the proposed methods.

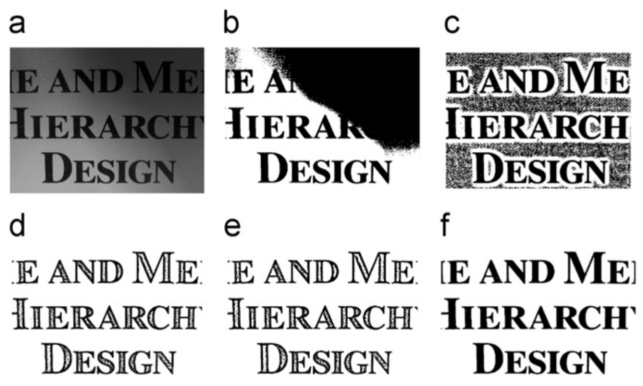


Fig. 7. Binarization of a paper document image: (a) original image, (b) Otsu's method, (c) Niblack's method, (d) Sauvola's method, (e) Gatos's method, and (f) the proposed method.

for comparison. All algorithms are implemented in Borland C++ Builder 6 and ran on a Pentium D 2.66 GHz processor with a 1 GB RAM Windows XP platform. Section 4.1 first evaluates the processing time to demonstrate the efficiency of the proposed approach. Section 4.2 then determine the recognition rate to show the high performance of the proposed method.

4.1. *Time consumption*

Table 5 lists the processing times of various methods for the four document images. The first row represents the document images *IMG1* to *IMG4*. The first column represents the three local methods, the proposed method and a global method. Since the local algorithms of Niblack's, Sauvola's, and Gatos's were reported in 1986, 2000 and 2006, they require additional processing time to binarize the images. The increasing processing time of the three local methods is attributed to their complexity in terms of quality. According to Table 5, the best case of the local methods requires 19.83 s, while the proposed method requires only 0.610 s. Thus, the proposed method is much faster than the local methods. In particular, in the case of *IMG1*, the processing time of the proposed method is 0.077 s, while that of Otsu's method is 0.013 s. Although the proposed method is slightly slower than Otsu's method, they are of the same order. This observation further demonstrates that the proposed method has the lowest computational complexity among all the adaptive thresholding algorithms, as mentioned earlier in Section 3.2. The higher speed can be explained by use of the intelligent block size, which can be computed automatically based on the image information.

In addition to the above analysis, Table 6 compares the processing times for various image resolutions by using the typical methods and the proposed method. Table 8 displays test document image *IMG5*. In the manufacturing portable

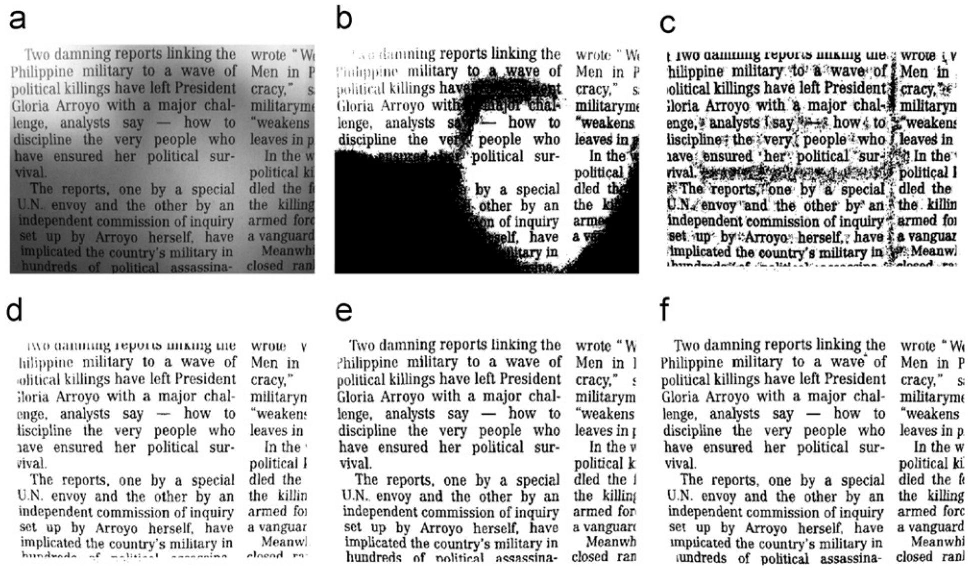


Fig. 8. Binarization of a newspaper document image: (a) original image, (b) Otsu's method, (c) Niblack's method, (d) Sauvola's method, (e) Gatos's method, and (f) the proposed method.

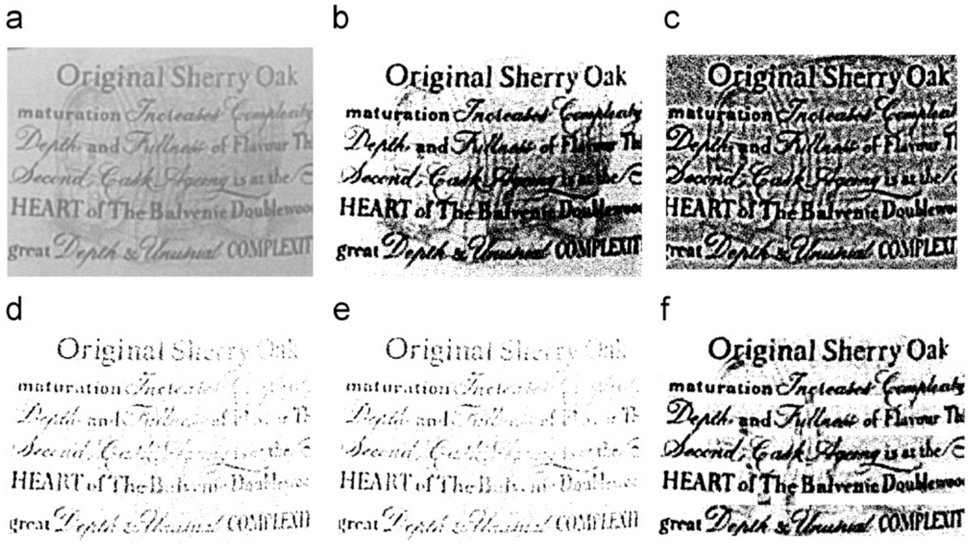


Fig. 9. Binarization of a magazine document image: (a) original image, (b) Otsu's method, (c) Niblack's method, (d) Sauvola's method, (e) Gatos's method, and (f) the proposed method.

devices, the image resolution can be classified into 0.3, 1, 5, 7, and 10 M pixels. The table clearly reveals that the processing time of the proposed method is significantly shorter than that of the local methods, regardless of which resolution is set. Additionally, with the increasing image resolution, the proposed method can significantly reduce the processing time. Fig. 6(a) schematically depicts this tendency, indicating that the proposed method requires only 3.77 s to obtain an acceptable performance with 10 M pixels resolution. Moreover, the proposed method is 347.7 times faster than that of Niblack's. Fig. 6(b) shows a closeup of the proposed method and Otsu's method. Although slower than Otsu's method, the proposed method only exceeds 1.23 s of Otsu's method in 10 M pixels resolution. We can thus infer that the processing time tendency of the proposed method is consistent with that of the global method.

4.2. *Quality measurment*

Although high-speed image binarization processing is of priority concern, the quality of the results must also be considered. Fig. 7(a) displays an original document image taken from a paper that involves large font size characters and gradations of shadows. When using Otsu's method, shown in Fig. 7(b), the characters located in the upper part were not separated and still merged with the background. Fig. 7(c) shows the binarization result of Niblack's method. We can see that characters can be separated well, but a lot of noises interference still occurs in the area without characters. Although Sauvola's method resolves Niblack's problem, the additional hypothesis caused thinner characters and holes, as shown in Fig. 7(d). Fig. 7(e) summarizes the binarization results of Gatos's method. The characters are still broken, explaining why the binarization

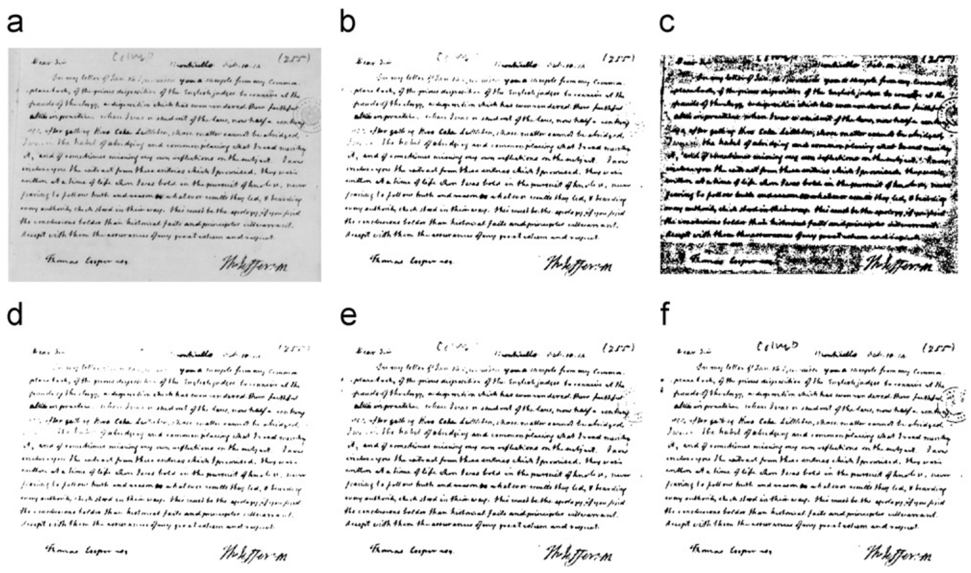


Fig. 10. Binarization of a handwritten document image: (a) original image, (b) Otsu's method, (c) Niblack's method, (d) Sauvola's method, (e) Gatos's method, and (f) the proposed method.

Table 5
Comparison of processing time(s) among Niblack's, Sauvola's, Gatos's, Otsu's, and the proposed methods.

Method	*IMG1*	*IMG2*	*IMG3*	*IMG4*	*Average*
Niblack's	19.83	19.83	19.79	20.01	20.04
Sauvola's	19.85	20.20	20.86	22.48	20.84
Gatos's	30.01	29.85	29.18	29.36	29.60
Proposed	0.077	0.610	0.159	0.561	0.351
Otsu's	0.013	0.013	0.013	0.013	0.013

Table 6
Comparison of processing time(s) upon various image resolutions by using Niblack's, Sauvola's, Gatos's, Otsu's, and the proposed methods.

	Image resolution (pixel)				
Method	640×480 (0.3 M)	1280×960 (1 M)	2560×1920 (5 M)	3072×2304 (7 M)	3648×2376 (10 M)
Niblack's	19.80	83.98	532.51	852.36	1311
Sauvola's	19.94	105.51	555.40	854.57	1331
Gatos's	29.51	119.61	569.38	904.94	1373
Proposed	0.16	0.27	0.68	1.35	3.77
Otsu's	0.01	0.04	0.18	0.25	1.965

Table 7
Comparison of recognition rate among Otsu's, Niblack's, Sauvola's, Gatos's, and the proposed methods.

Binarization method	Average recognition rate (%)
Otsu's	31.52
Niblack's	44.18
Sauvola's	83.01
Gatos's	96.70
Proposed	98.22

results remain unsatisfactory. According to Fig. 7(f), the proposed method yields the best segmentation outcome without any broken characters.

Fig. 8(a) illustrates an original document image taken from a newspaper containing small font size characters and non-uniform illumination. When using Otsu's method, shown in Fig. 8(b), the characters were partially broken in the upper left part; in addition, the characters in the lower right and left parts co-existed with the background. Fig. 8(c) shows the binarization result of Niblack's method, in which characters were segmented successfully from the background; however, noises interference still occurred in the area without characters. Fig. 8(d) and (e) display the binary document images obtained by applying Sauvola's method and Gatos's method. Under these circumstances, the characters were successfully extracted from the background and the binarization quality was also acceptable. Noteably, the results of the proposed method, shown in Fig. 8(f), were also satisfactory and resembled those of Gatos's method.

Fig. 9(a) displays an original document image taken from a magazine that includes slant characters and a complex background. According to the binarization results of Otsu's method, shown in Fig. 9(b), the characters located in the right part were not well segmented from the background due to its complexity. Fig. 9(c) shows the binarization results of Niblack's method, where a significant amount of noises interference in the area without characters. Fig. 9(d) displays the binarization result of Sauvola's method, where characters became extremely thinned and broken. In the case of Gatos's method, shown in Fig. 9(e), although an improved result was obtained over that of Sauvola's method, partially broken characters were still observed. Conversely, the proposed method yield the best segmentation results among all of the methods as shown in Fig. 9(f).

Fig. 10(a) shows the image of a historical handwritten letter. When using Otsu's method, shown in Fig. 10(b), the characters located in the upper and left part were not separated clearly. Fig. 10(c) summarizes the binarization results of Niblack's method, in which a large amount of noises interference appeared in the background area without characters. In the case of Sauvola's method and Gatos's method, despite more details than that of Niblack's method, some characters became thinned and broken, as shown in Fig. 10(d) and (e). According to Fig. 10(f), the proposed method produced the best segmentation results without any broken characters.

Table 8
Comparison of binarization results upon various image resolutions by using Gatos's and the proposed methods.

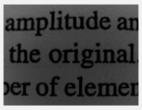
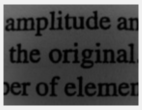
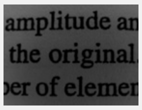
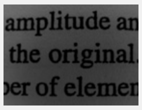
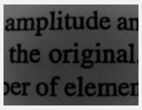
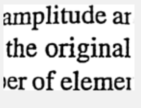
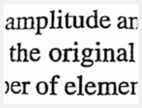
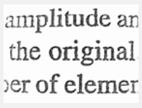
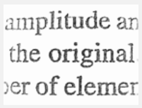
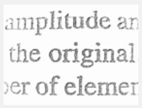
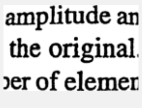
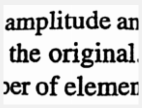
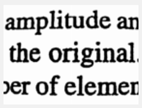
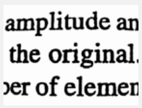
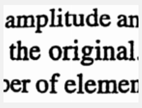
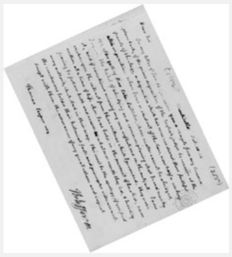
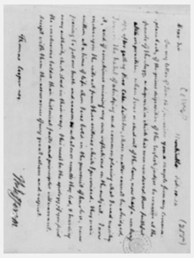
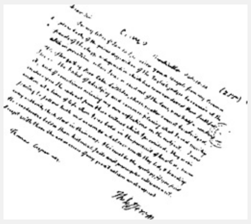
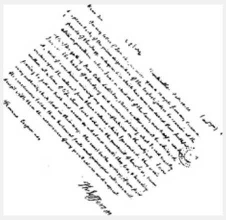
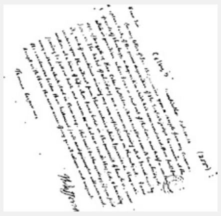
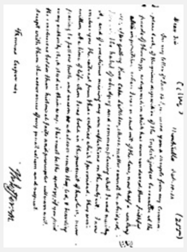
	Image resolution (pixel)				
	640 × 480 (0.3 M)	1280 × 960 (1 M)	2560 × 1920 (5 M)	3072 × 2304 (7 M)	3648 × 2736 (10 M)
Original image					
Gatos's method					
Proposed method					

Table 9
Experimental results of skew images binarization using the proposed method.

−30°	−45°	−65°	−90°
			
			

To provide an objective comparison criterion, all binary images generated by each method are fed into a commercial OCR system produced by ABBYY Finereader 7.0 Professional Edition [21]. Table 7 summarizes those results. According to this table, the recognition engine achieves a recognition rate of 98.22% when using the proposed method, 96.70% when using Gatos's method, 83.01% when using Sauvola's method, 44.18% when using Niblack's method, and 31.52% when using Otsu's method. Thus, the OCR also confirms the best results of the proposed method.

Table 8 compares the binarization results of the proposed method with those of Gatos's method for various image resolutions. This table reveals that the proposed method and Gatos's method can achieve satisfactory results for 0.3 and 1 M pixels. However, the characters of Gatos's method became broken for 5 and 7 M pixels. In contrast, the proposed method can still clearly distinguish the characters from the background. Additionally, the proposed method performs better than Gatos's method in 10 M pixels because the intelligent block detection can locate the region of the characters from the background. Therefore, the proposed method attains satisfactory binarization results regardless of the resolution.

Moreover, this work also attempts to verify the effectiveness of the posed thresholding algorithm in terms of the processing inputs with any angle, Table 9 displays the experimental results with respect to the skew images. In this table, the first row shows the angles of the document images; the second row illustrates the gray level skew images; and the last row displays the thresholding results of the proposed method. According to this table, the quality of these binarization results is satisfactory.

In summary, although the proposed method is a slightly slower than Otsu's method, the binarization results of the proposed method are much better than those of the latter one. Additionally, the proposed method is also much faster and performs better than the local methods. Therefore, the proposed algorithm can be readily implemented in embedded systems with less computational power.

5. Conclusion

This work presents a novel document image binarization algorithm with lower computational complexity and high performance. Based on the image characteristic, the document image is intelligently divided into several blocks with various sizes. A threshold surface is then constructed to derive the binary image. Importantly, the proposed method can increase the binarization quality and reduce the processing time. Experimental results also indicate that the proposed approach not only achieves a high speed as well as the global method can, but also performs considerably better than the existing methods. Consequently, the proposed algorithm can be readily adopted in embedded systems with less computing power. Given that OCR systems are widely used with less computational power, the proposed algorithm is highly promising for implementation owing to its lower computational complexity in document image binarization.

Since the proposed block detection is developed by hypothesizing on aligned letters, the detected blocks may not be able to fit the characters of irregular location. As a result, the proposed thresholding algorithm require additional processing time when the block size is much smaller than the height of characters. To overcome this limitation, we are continuing efforts to develop a floating block detection for the irregular location words in the near future.

Acknowledgments

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under Contract No. NSC 98-2221-E-001-114.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at 10.1016/j.patcog.2010.03.014.

References

[1] C.Y. Suen, L. Lam, D. Guillevic, Bank check processing system, International Journal of Imaging Systems and Technology 7 (1996) 392–403.

[2] H. Hegt, R. Haye, N. Khan, A high performance license plate recognition system, IEEE International Conference on Systems, Man, and Cybernetics 5 (1998) 4357–4362.

[3] P.K. Sahoo, S. Soltani, A.K.C. Wong, Y.C. Chen, A survey of thresholding techniques, Computer Vision, Graphics and Image Processing 41 (1988) 233–260.

[4] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man, and Cybernetics SMC 9 (1) (1979) 62–66.

[5] W. Tsai, Moment-preserving thresholding: a new approach, Computer Vision, Graphics, and Image Processing 29 (1985) 377–393.

[6] G. Johannsen, J. Bille, A threshold selection method using information measures, in: Proceedings Sixth International Conference Pattern Recognition, 1982, pp. 140–143.

[7] J.N. Kapur, P.K. Sahoo, 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 (1985) 273–285.

[8] Y. Liu, S.N. Srihari, Document image binarization based on texture features, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 533–540.

[9] M. Cheriet, J. Said, C. Suen, A recursive thresholding technique for image segmentation, IEEE Transactions on Image Processing 7 (1998) 918–921.

[10] O.D. Trier, A.K. Jain, Goal-directed evaluation of binarization methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (12) (1995) 1191–1201.

[11] W. Niblack, An Introduction to Digital Image Processing, Pretice-Hall, Englewood Cliffs, NJ, 1986.

[12] J. Sauvola, M. Pietikainen, Adaptive document image binarization, Pattern Recognition 33 (2000) 225–236.

[13] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging 13 (1) (2004) 146–165.

[14] O.D. Trier, T. Taxt, Evaluation of binarization methods for document images, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (3) (1995) 312–315.

[15] B. Gatos, I. Pratikakis, S.J. Perantonis, Adaptive degraded document image binarization, Pattern Recognition 39 (2006) 317–327.

[16] Hitachi mobile phone with ocr., 〈http://forbes.com/technology/feeds/infoimaging/2005/03/18/infoimagingasiapulse_2005_03_18_ix_4427-0236-.html〉.

[17] Quicktextscan ocr software.,〈http://jsscomputing.com/quicktextscan/index.html〉.

[18] H. Al-Yousefi, S. Udpa, Recognition of arabic characters, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (8) (1992) 853–857.

[19] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Prentice Hall, Upper Saddle River, NJ, 2002.

[20] J. Sadri, M. Cheriet, A new approach for skew correction of documents based on particle optimization, in: IEEE 10th International conference on Document Analysis and Recognition, 2009, pp. 1066–1070.

[21] Abbyy, 〈http://www.finereader.com〉.

About the Author—Yu-Ting Pai received the B.S. degree in electronic engineering from Huafan University, Taiwan (2000-2004), and Ph.D. degree in Electronic Engineering (2005-2009) from National Taiwan University of Science and Technology. He was a guest researcher at University of Dortmund, Germany from July to August in 2005-2007. He was also a researcher scholar at University of North Texas, U.S.A from July 2008 to May 2009. His current interests include low-complexity algorithms, energy-efficient computing, embedded systems, image processing, multimedia security, color image processing, and pattern recognition.

About the Author—M.S. Yi-Fan Chang received the B.S. degree in electronic engineering from Huafan University, Taiwan (2002-2006). She received the M.S. degree in electronic engineering from the Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan (2006-2008). Her research interests are in the low power design and high efficient image processing.

About the Author—Shanq-Jang Ruan received the B.S. degree in computer science and information engineering from Tamkang University, Taiwan (1992∼1995), the M.S. degree in computer science and information engineering (1995∼1997), and Ph.D. degree in Electrical Engineering (1999∼2002) from National Taiwan University, respectively. He served as an Electronic officer in R.O.C. Air Force during July 1997 and May 1999. He was also a guest researcher at University of Dortmund, German from July to August in 2001. From Sep. 2001 to May 2002, he served as a software engineer in Avant! Corporation. Since June 2002, he joined Synopsys as a software engineer. Dr. Ruan's research interests are in low power VLSI/EDA designs and energy-efficient computing for multimedia systems. He was an assistant professor at Department of Electronic Engineering of National Taiwan University of Science and Technology from 2003 to 2007. He is currently an associate processor at Department of Electronic Engineering of National Taiwan University of Science and Technology.