International Journal on Document Analysis and Recognition (IJDAR) (2018) 21:123–135
https://doi.org/10.1007/s10032-018-0299-9

ORIGINAL PAPER

Binarization of degraded document images based on contrast enhancement

Di Lu[1] · Xin Huang[1] · LiXue Sui[1]

Received: 24 September 2015 / Revised: 31 January 2018 / Accepted: 27 March 2018 / Published online: 6 April 2018

Abstract

Because of the different types of document degradation such as uneven illumination, image contrast variation, blur caused by humidity, and bleed-through, degraded document image binarization is still an enormous challenge. This paper presents a new binarization method for degraded document images. The proposed algorithm focuses on the differences of image grayscale contrast in different areas. Quadtree is used to divide areas adaptively. In addition, various contrast enhancements are selected to adjust local grayscale contrast in areas with different contrasts. Finally, the local threshold is regarded as the mean of foreground and background gray values, which are determined by the frequency of the gray values. The proposed algorithm was tested on the datasets from the Document Image Binarization Contest (DIBCO) (DIBCO 2009, H-DIBCO 2010, DIBCO 2011, and H-DIBCO 2012). Compared with five other classical algorithms, the images binarized using the proposed algorithm achieved the highest F-measure and peak signal-to-noise ratio and obtained the highest correct rate of recognition.

Keywords Degraded document · Binarization · Quadtree · Contrast enhancement · Local threshold · Document image analysis

1 Introduction

As the optical character recognition (OCR) techniques have become widely available, a crucial first step for OCR remains document image binarization. Although document image binarization has been studied for many years, extracting clear characters from degraded document images is still a challenging problem.

Image binarization sets the gray values of pixels to 0 or 255, creating a black and white image [1]. Wen et al. [2] divided binarization methods into three major categories: clustering-based, threshold-based, and hybrid methods. In clustering-based methods, the image pixel's gray levels are partitioned into two clusters according to model-based features. Fuzzy classification [3–5] is a typical clustering-based method. Because clustering is an iterative process, the key to clustering-based methods is the selection of the initial value, clustering criterion, and termination condition. If this selection is effective, the binary image will have clear characters and little noise, but if the selection is poor, some background will be clustered in the foreground mistakenly. In addition, the binary image will lose a large amount of the foreground information. Clustering-based methods perform well on the images with non-uniform illumination. However, for images with bleed-through, clustering-based methods are usually unable to select an ideal clustering criterion nor produce a high-quality binary image. Recently, many researchers have presented hybrid algorithms for image binarization. For example, Chou and Lin's method [6] combined SVM with Otsu's threshold. Mesquita et al. [7] combined K-means with Otsu's threshold. However, most of the hybrid methods have a trade-off between the ability to reduce noise and the complexity of processing. If the algorithm has a strong ability to reduce noise, its complexity will be high and the running time will increase, which is not conducive to practical applications. However, threshold-based methods have been researched widely because of their briefness, efficiency, and easy comprehension. These methods include two subcategories: global and local. Global binarization methods [8–11] segment well the images for which the gray value

✉ Xin Huang
scorpion_hx@163.com

Di Lu
ludizeng@hrbust.edu.cn

LiXue Sui
13263509301@163.com

[1] Harbin University of Science and Technology, Harbin, China

distributions between foreground and background are uniform and the deviations are small. However, for degraded document images, global binarization methods will produce numerous mistakes. Nevertheless, local binarization methods [12–18] are more suitable for document image binarization because they use windows or blocks to determine a local threshold. However, the selection of window or block will affect the application of the local threshold method. Windows or blocks that are too small will produce a large amount of noise. However, if they are too large, texts will be lost.

Most of the current techniques achieve a good binarization effect on a specific degradation. However, there are many reasons for degradation, and a good binarization algorithm should be able to deal with a variety of situations. Based on this, this paper proposes a local binarization algorithm that can handle images with uneven illumination, bleed-through, and variable background.

This paper is organized as follows: Section 2 reviews binarization methods. Section 3 describes the proposed binarization method. Section 4 presents an analysis of experimental results. Section 5 presents a discussion of the results and concludes the paper.

2 Review of binarization methods

The key of threshold-based methods is how to select the threshold. Researchers usually use histograms to determine a global threshold for images that have a clear distinction between the foreground and background. Classical global binarization methods (such as Otsu's method [8]) use maximum variance between foreground and background to determine the threshold dynamically. Thanks to its adaptability, it is still one of the most commonly used methods of image segmentation. For some images degraded by noise, uneven illumination, or low contrast, local analysis for images can overcome the influence of degradation to some extent. For these kinds of degraded images, local binarization methods generally gain better results. This section mainly reviews local threshold-based methods.

Niblack's method [13] is a commonly used local adaptive binarization algorithm. It calculates the threshold based on the local mean and standard deviation. Threshold T for pixel $f(x, y)$ is defined as:

$$T(x, y) = m(x, y) + k \cdot s(x, y) \tag{1}$$

where $m(x, y)$ and $s(x, y)$ refer to the mean and variance of the gray values in a neighborhood, respectively. The neighborhood should be moderate in size such that it can both contain local details and suppress noise. Hence, the value of k and the neighborhood's size are chosen as in [18]. According to Gatos et al. [19], Niblack's [13] method cannot handle background with light texture.

Sauvola and Pietikainen [14] proposed an improved algorithm based on Niblack's method, which has become a standard for local threshold methods. Sauvola and Pietikainen's algorithm uses the current pixel as the center of a neighborhood. It dynamically calculates the threshold on the basis of the grayscale average and standard deviation of the current pixel in a neighborhood. Threshold T is as follows:

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

where $m(x, y)$ and $s(x, y)$ are the same as those in Niblack's method, R is the dynamic range of the standard deviation, and k is the correction factor, which ranges from 0 to 1. Sauvola and Pietikainen's method can handle degraded document images with variable illumination, resolution variation, and noise, but fails for very light or very dark backgrounds [18].

Bernsen's method [12] is a typical local threshold algorithm. It calculates the threshold using mean and contrast information over a local region. The threshold is calculated as:

$$T(x, y) = \frac{Z_{\text{low}} + Z_{\text{high}}}{2} \tag{3}$$

where Z_{low} and Z_{high} are the lowest and highest gray levels, respectively, in an $r \times r$ region. Bernsen selects $r = 15$. This method produces a large amount of background noise, especially for degraded document images with blank backgrounds.

Singh et al. [18] proposed a new adaptive binarization method. Their method has four steps: contrast analysis, contrast stretching, thresholding, and noise removal. Singh et al.'s method works well on degraded document images. However, it fails when the document does not contain dense text or suffers serious bleed-through. Furthermore, it is more time-consuming and sensitive to parameter changes.

For degraded document images, current local binarization methods are affected by the size of the window or block to some extent. This means that they are unable binarize images that have bleed-through or little text. Hence, this paper presents a new binarization method for degraded document images based on contrast enhancement. It mainly solves the problem of document images degraded by uneven illumination, bleed-through, and variable background.

3 Proposed algorithm

For both humans and computers, when identifying text, the basis of distinction between the foreground and background is the obvious difference in gray values at the edges of the characters. For degraded document images, some areas have obvious grayscale contrast, but others do not. Therefore, for degraded document images that have obvious differences in grayscale contrast in different areas, a single binarization method cannot achieve a good result. To address this issue, this paper presents a binarization method that uses different contrast enhancements for areas with different grayscale contrasts.

3.1 Area partition

Regional division directly influences whether an adaptable contrast enhancement method can achieve the best results in the corresponding region. Hence, it is crucial to find a suitable region division method. The contrast of pixels can be used as the basis for dividing the areas. Let $F(x, y)$ with 256 gray levels be the grayscale image of an input document of size $M \times N$, where M is the number of lines and N is the number of pixels per line in the image. The grayscale contrast C for pixel $f(x, y)$ is defined as:

$$C(x, y) = \max[C_h(x, y), C_v(x, y)] \tag{4}$$

$$C_h(x, y) = |f(x+1, y) - f(x, y)| \tag{5}$$

$$C_v(x, y) = |f(x, y+1) - f(x, y)| \tag{6}$$

where $C_h(x, y)$ and $C_v(x, y)$ are the absolute contrasts along the horizontal and vertical directions, respectively. For an image with a white background and black foreground, considering that there may be many background areas without characters in the image, it will save a lot of computation if this background region can be removed directly. For the target areas with characters, due to the fact that the gray contrast between regions (e.g., bright and dark regions) may be significantly different, they may need to be divided again. By dividing the area repeatedly, the areas can be fully partitioned. The whole image can be divided into not-significant areas, significant areas, and comparatively significant areas.

3.1.1 Coarse region division

The proposed algorithm utilizes a quadtree to divide areas on the basis of grayscale contrast, as shown in Fig. 1. After first division, the image is divided into four subregions named A, B, C, and D. If the maximum grayscale contrast in subregion B is less than k_1 times the maximum grayscale contrast of the whole area as follows:

$$C_{B\max}(x, y) \leq k_1 \cdot C_{\text{entire}\max}(x, y) \tag{7}$$

where $C_{B\max}(x, y)$ is the maximum grayscale contrast of subregion B after the first division, $C_{\text{entire}\max}(x, y)$ is the maximum grayscale contrast of the whole image, and k_1 is the partition coefficient between the foreground and background, the grayscale is defined to change indistinctively. Hence, this subregion is defined as background without characters and output directly. In this step, large areas of background can be eliminated. This step notably reduces the computation. The rest of the target areas with characters can then be divided sequentially.

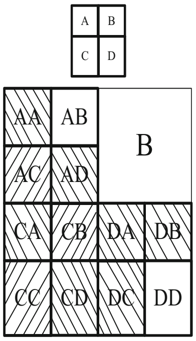


Fig. 1 Division diagram. Empty box Background, box with right side stripe Areas with significant grayscale contrast, and box with left side stripe Areas with comparatively significant grayscale contrast

3.1.2 Fine region division

After coarse division, areas that do not satisfy Eq. 7 are regarded as target areas with characters. In the example shown in Fig. 1, A, C, and D are target areas with characters. For a degraded document image, there may also be a significant difference in the gray contrast among the remaining regions. Hence, further subdivision needs to be done for the rest of the regions.

For instance, subregion A (note that subregions C and D have the same division rules) is divided for the second time after coarse division. If the maximum grayscale contrast in subregion AB is less than k_1 times the maximum grayscale contrast of the former division as follows:

$$C_{AB\max}(x, y) \leq k_1 \cdot C_{A\max}(x, y) \tag{8}$$

there is no significant variance in this subregion. Hence, this subregion is also background and output directly.

If the maximum grayscale contrast in subregion AA is more than k_2 times the maximum grayscale contrast of the former division as follows:

$$C_{AA\max}(x, y) \geq k_2 \cdot C_{A\max}(x, y) \tag{9}$$

this subregion has significant variance. Hence, weak contrast enhancement is used in this subregion. If the maximum grayscale contrast in subregion AC is between k_1 and k_2 times the maximum grayscale contrast of the former division as follows:

$$k_1 \cdot C_{A\max}(x, y) \le C_{AC\max}(x, y) \le k_2 \cdot C_{A\max}(x, y) \tag{10}$$

where $C_{AB\max}(x, y)$,$C_{AA\max}(x, y)$, and $C_{AC\max}(x, y)$ are the maximum grayscale contrasts of subregions AB, AA, and AC after the second division, respectively, $C_{A\max}(x, y)$ is the maximum grayscale contrast of subregion A after the first division, and k_2 is the partition coefficient between the significant and comparatively significant areas, the grayscale variance is comparatively significant. A strong contrast enhancement is used in this subregion.

In this study, the ranges for $k_1$and k_2 were empirically determined to be $k_1 \in [0, 0.4]$ and $k_2 \in [0.7, 1]$, respectively. Two divisions were also empirically found to be the optimal number for determining the property of grayscale variance. Too many divisions will lead to a large amount of calculation, generate mistakes between the noise and target, and cannot handle noise well. At the same time, too few divisions will reduce calculation but will also lose detail.

3.2 Grayscale contrast enhancement

Section 3.1 divides areas into not-significant areas, significant areas, and comparatively significant areas. Usually, a document image has a black foreground and white background. Therefore, for not-significant areas, the gray values of pixels within this area are set to:

$$ff(x, y) = 255 \tag{11}$$

For significant areas, weak contrast enhancement [20] is used to modify gray values as follows:

$$ff(x, y) = (n - 1) \times \frac{f(x, y) - f_{\min}(x, y)}{f_{\max}(x, y) - f_{\min}(x, y)} \tag{12}$$

For comparatively significant areas, this paper proposes a strong contrast enhancement mode that further widens the contrast between pixels within that region. The gray values of the pixels are modified as follows:

$$ff(x, y) = (nn - 1) \times \left(\frac{f(x, y) - f_{\min}(x, y)}{f_{\max}(x, y) - f_{\min}(x, y)}\right)^2 \tag{13}$$

In Eqs. 12 and 13, $f(x, y)$ is the gray value of the original grayscale, $f_{\max}$ and $f_{\min}$ denote the maximum and minimum gray levels in the original document image, respectively, and n and nn denote the number of gray levels modified.

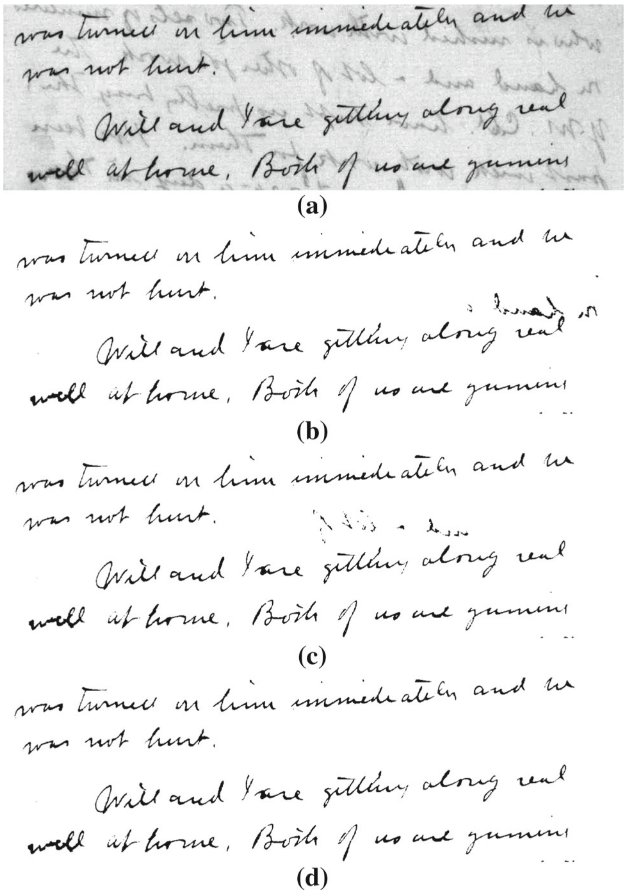


Fig. 2 **a** Original degraded image. Binarized output of the proposed algorithm using, **b** weak contrast enhancement only, **c** strong contrast enhancement only, and **d** both weak and strong contrast enhancements

The essence of gray contrast enhancement methods is contrast extension. The object is to extend the contrast of target areas to a larger range of gray levels and to suppress gray-level changes in the background. The reason why two enhancement methods are used is that strong contrast enhancement inevitably produces noise in significant areas. Another reason is that, for comparatively significant areas, weak contrast enhancement may not separate clear characters because it does not have enough capacity to widen the contrast between pixels. Meanwhile, for degraded document images, a single contrast enhancement method may create a two-tone image. Therefore, two types of contrast enhancement are indispensable. Figure 2 compares the binarized images of strong and weak contrast enhancement. This illustrates the necessity of using different contrast enhancements for different grayscale variance areas.

The proposed grayscale enhancement method can effectively adjust the pixel gray values of an image with non-uniform illumination, bleed-through, and variable background. As a result, these three issues in image binarization can be solved. For degraded images caused by bleed-through or non-uniform illumination, the ink bleed-through area and lighter or darker areas can be classified as comparatively significant areas as they have little difference between the foreground and background. For this situation, the strong contrast enhancement in Eq. 13 can be used to separate clear characters. For areas that are only slightly degraded, the weak contrast enhancement of Eq. 12 can be used to reduce the effect of noise, as the areas have a significant difference between the foreground and background. For degraded images with variable background, the variety of contrast in the background is far less than it is between the background and foreground. Therefore, if the region only consists of large contrast background without any characters in it, Eq. 11 is used to remove it. If the region has both characters and variable background, two kinds of contrast enhancement are used to widen the contrast between pixels so that the foreground can be separated from the background.

3.3 Local threshold estimation

The foreground can be distinguished from the background intuitively after the grayscale values have been modified. By analysis of the document image, in general, most of the document image character pixels are less than those in the background. Only a very small number of characters are more than the background pixels. Hence, the gray values of the background and the foreground can be determined by accumulating the number of pixels corresponding to the gray value in a histogram. For an image after contrast enhancement (which is $p \times q$ in size), we search for the gray value n_{halfsize} for which its accumulation is closest to $\frac{p \times q}{2}$. In addition, the highest frequency gray value is regarded as the foreground $ff_{\text{foreground}}$ in $0 \sim n_{\text{halfsize}}$. In $n_{\text{halfsize}} \sim n$, the highest frequency gray value is regarded as the background $ff_{\text{background}}$. In some cases, there is more than one gray value that is the highest frequency for the background or foreground. We then choose the largest or smallest of these in value to represent the background or foreground. Threshold T is defined to be the mean of $ff_{\text{foreground}}$ and $ff_{\text{background}}$:

$$T = \frac{ff_{\text{foreground}} + ff_{\text{background}}}{2} \tag{14}$$

Finally, the binarized image $g(x, y)$ is obtained as:

$$g(x, y) = \begin{cases} 1, & ff > T \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

4 Experiments and discussion

4.1 Experimental environment and test datasets

All algorithms were implemented on a MATLAB (r2011a) compiler and run on an Intel Core i3-3240 CPU 3.40 GHz processor with 4.00 GB RAM and Windows 7 operating system. In our experiments, we used the Document Image Binarization Contest (DIBCO) series datasets (DIBCO 2009, H-DIBCO 2010, DIBCO 2011, and H-DIBCO 2012) [21–24] that include 50 handwritten and printed images.

4.2 Testing segmentation results

The proposed approach for binarization was compared with five recent and benchmark binarization methods: Otsu, Niblack, Sauvola and Pietikainen, Bernsen, and Singh et al. The experimental results of all the methods in Fig. 3 are shown in Figs. 4, 5, 6, 7 and 8. Because Niblack's, Bernsen's, Sauvola and Pietikainen's, and Singh et al.'s methods all have the problem of parameter selection, this study set the parameters according to the references [12–14,18]. For Niblack's, Bernsen's, and Sauvola and Pietikainen's methods, five experiments were done with the window sizes 5 × 5, 15 × 15, 25 × 25, 35 × 35, and 50 × 50. For Singh et al.'s method, we used the block sizes 32 × 32, 64 × 64, 128 × 128, 256 × 256, and 512 × 512. The proposed method randomly selected five groups k_1, k_2 from $k_1 \in [0, 0.4]$, $k_2 \in [0.7, 1]$ using 0.1 as the interval to perform the tests. The above experiments all selected the binary image with the best F-measure value as the final result.

4.3 Visual evaluation

4.3.1 Experiment 1

As can be seen from Figs. 4 and 6, for images with bleed-through, Otsu's method and Sauvola and Pietikainen's method inevitably produce a little noise. Niblack's method mistakes noise caused by bleed-through as foreground. Bernsen's method produces a large amount of background noise. In addition, Singh et al.'s method also introduces noise in the background areas. However, in these images, the proposed algorithm can intelligently select the target area and non-target background area, avoiding the interference of noise.

4.3.2 Experiment 2

It can be seen from Figs. 5 and 7, that for images with variable background, Otsu's method and Sauvola and Pietikainen's method can separate characters without noise. However, the characters have clear broken strokes in weak contrast areas

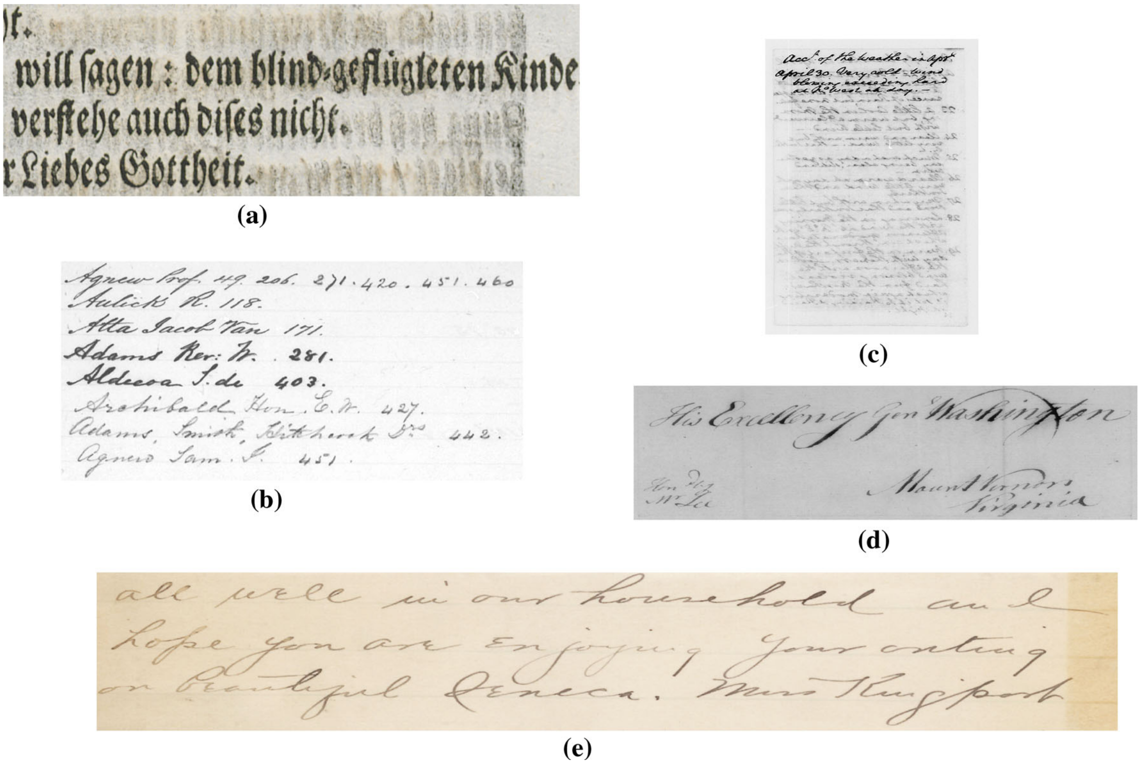


Fig. 3 Example document images in the DIBCO datasets that illustrate document degradation, consisting of bleed-through in **(a)** and **(c)**, image contrast variation in **(b)** and **(d)**, and uneven illumination in **(e)**

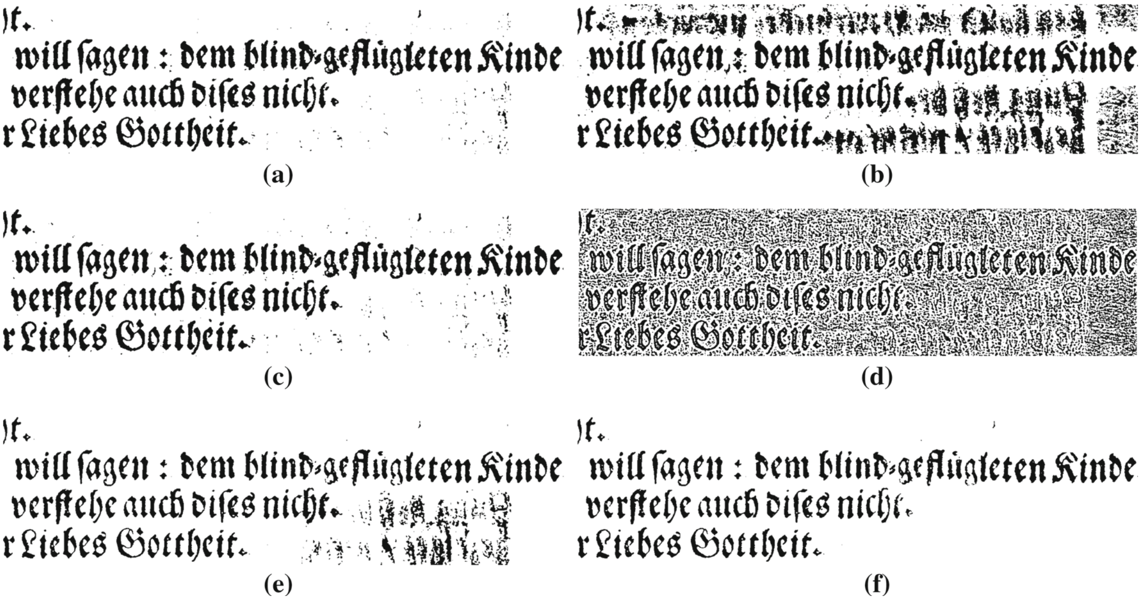


Fig. 4 Binarization results of the sample document images in Fig. 3a using **a** Otsu's method, **b** Niblack's method, **c** Sauvola and Pietikainen's method, **d** Bernsen's method, **e** Singh et al.'s method, and **f** proposed method

so they are unable to provide a reliable basis for subsequent character recognition. Although Niblack's method can isolate clear characters in both strong and weak contrast areas, at the same time, it detects a large number of black blobs in the non-target areas. The noise generated by Bernsen's method almost covers the target area and cannot distinguish the background area and target area. Singh et al.'s method still does not work well in degraded images with non-dense text. However, the proposed method can separate clear characters in both the strong and weak contrast areas without noise.

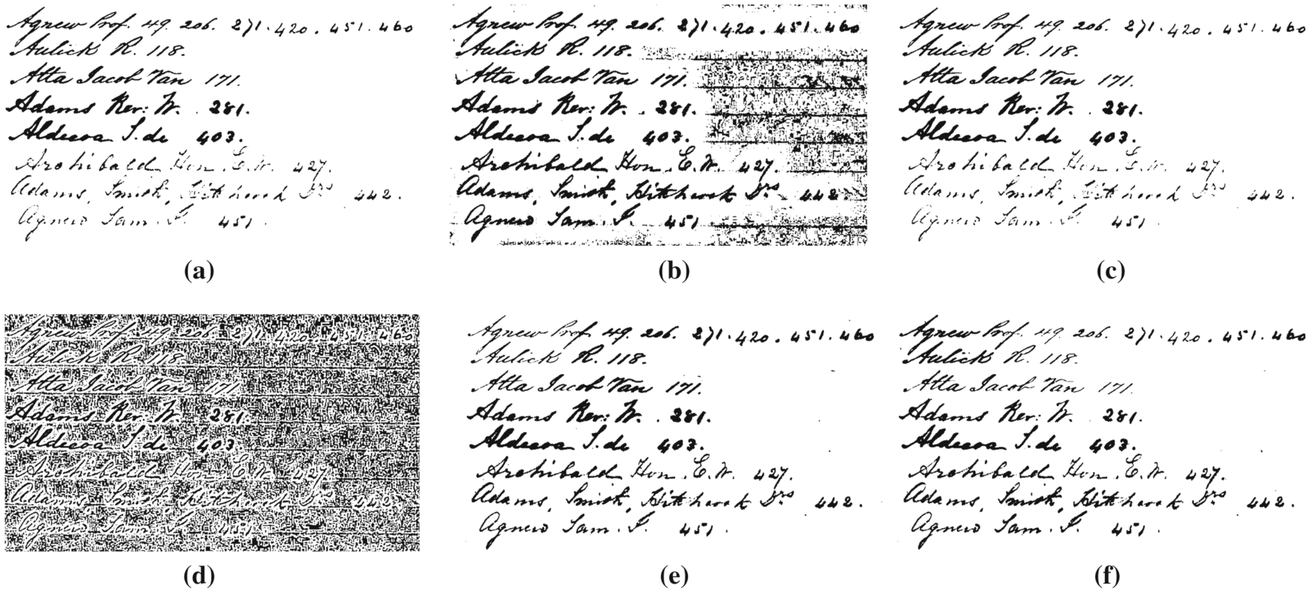


Fig. 5 Binarization results of the sample document image in Fig. 3(b) using **a** Otsu's method, **b** Niblack's method, **c** Sauvola and Pietikainen's method, **d** Bernsen's method, **e** Singh et al.'s method, and **f** proposed method

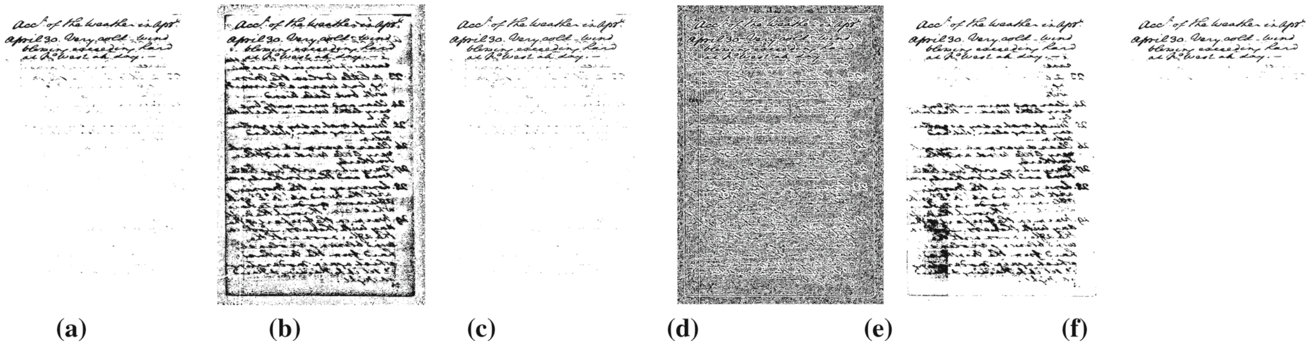


Fig. 6 Binarization results of the sample document image in Fig. 3(c) using **a** Otsu's method, **b** Niblack's method, **c** Sauvola and Pietikainen's method, **d** Bernsen's method, **e** Singh et al.'s method, and **f** proposed method

4.3.3 Experiment3

It can be seen in Fig. 8, that for images with uneven illumination, Otsu's method, Niblack's method and Bernsen's method all cannot eliminate the influence of the dark background. Although Sauvola and Pietikainen's method can handle noise, it loses many characters in the lighter and darker areas. Nevertheless, Singh et al.'s method and the proposed method can restore more complete characters with minimal noise.

4.4 Ground-truth-based evaluation measures

Higher F-measure, higher PSNR, and lower negative rate metric (NRM) are the essential conditions for a high-quality binarized image [16]. F-measure is calculated as:

$$\mathrm{FM} = \frac{2 \times \mathrm{RC} \times \mathrm{PR}}{\mathrm{RC} + \mathrm{PR}} \tag{16}$$

where PR and RC refer to the binarization recall and the binarization precision, respectively. Table 1 shows the F-measure of the results of various algorithms on the DIBCO datasets.

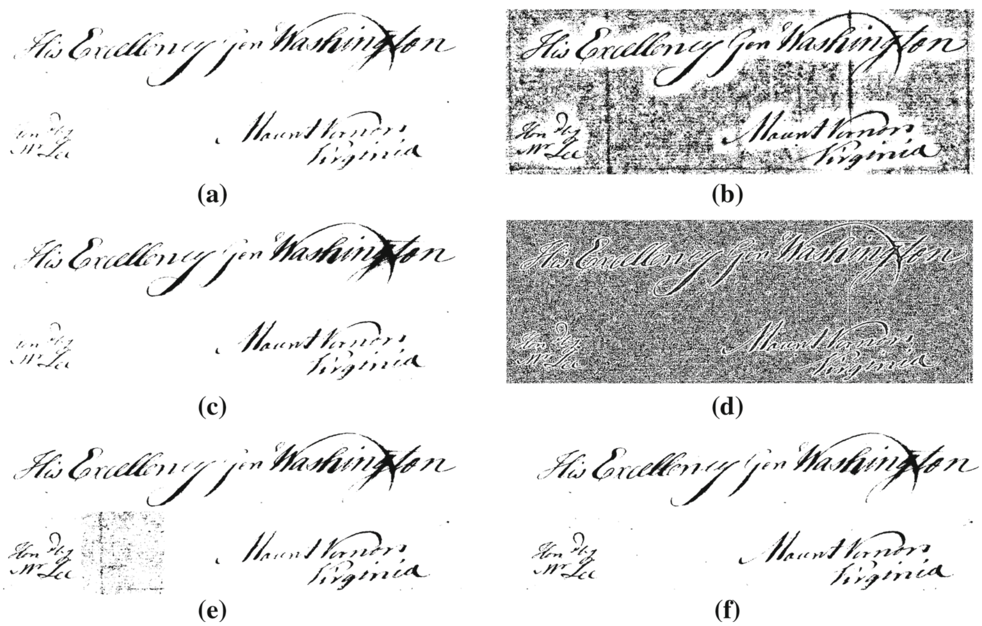
PSNR is calculated using

$$\mathrm{PSNR} = 10 \log \left(\frac{C^2}{\mathrm{MSE}} \right) \tag{17}$$

where MSE denotes the mean square error. Table 2 shows the PSNR of the results of various algorithms on the DIBCO datasets.

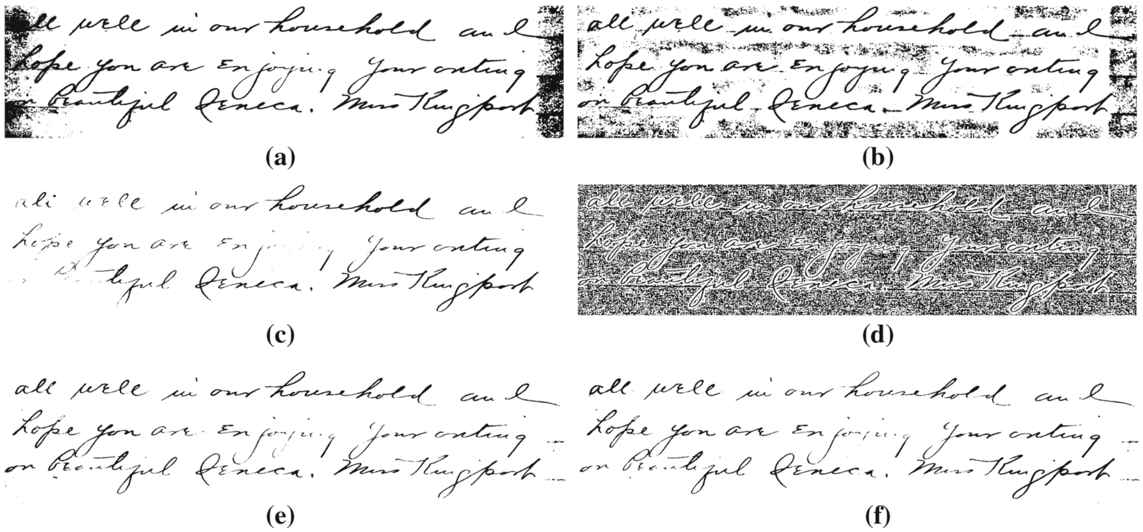
Finally, NRM is calculated as:

$$\mathrm{NRM} = \frac{\frac{\mathrm{FN}}{\mathrm{FN+TP}} + \frac{\mathrm{FP}}{\mathrm{FP+TN}}}{2} \tag{18}$$



(a) (b) (c) (d) (e) (f)

Fig. 7 Binarization results of the sample document image in Fig. 3(d) using **a** Otsu's method, **b** Niblack's method, **c** Sauvola and Pietikainen's method, **d** Bernsen's method, **e** Singh et al.'s method, and **f** proposed method



(a) (b) (c) (d) (e) (f)

Fig. 8 Binarization results of the sample document image in Fig. 3(e) using **a** Otsu's method, **b** Niblack's method, **c** Sauvola and Pietikainen's method, **d** Bernsen's method, **e** Singh et al.'s method, and **f** proposed method

where TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively. Table 3 shows the NRM of the results of the various algorithms on the DIBCO datasets. Tables 1, 2, and 3 illustrate that the binarized images using the proposed algorithm have the highest F-measure (4% higher than Otsu's method), the highest PSNR (5% higher than Sauvola and Pietikainen's method) and a higher NRM. The following explanation shows why the proposed method has a slightly higher NRM. Table 4 lists the metrics for the binary image in Fig. 8 for various algorithms.

Here, FP represents the number of pixels that are black in the binarized image but white in the ground truth, FN is the number of pixels that are white in the binarized image but black in the ground-truth image, TP represents the number of pixels that are white in both the binarized and ground-truth images, and TN represents the number of pixels that are white in both the binarized and ground-truth images. In addition, FP + FN represents the total number of pixels in error. According to Eq. 13, NRM counts the average of the proportion of background pixels for which the foreground pixels have mistakenly been regarded as background and foreground pix-

Table 1 Comparison of F-measure for six algorithms

F-measure (%)	2009	2010	2011	2012	Mean
OTSU	78.48	**83.69**	74.71	74.35	77.81
Niblack	52.70	46.16	49.88	47.15	48.97
Bernsen	32.45	22.11	30.16	21.63	26.59
Sauvola	80.66	63.37	**79.27**	68.42	72.93
Singh	69.81	74.92	70.36	66.92	70.50
Proposed	**82.75**	83.48	76.55	**80.98**	**80.94**

Bold values indicate the best experimental result compared with similar experiments

Table 2 Comparison of PSNR for six algorithms

PSNR(dB)	2009	2010	2011	2012	Mean
OTSU	15.17	**16.85**	13.86	14.40	15.07
Niblack	8.08	8.19	7.76	8.34	8.09
Bernsen	4.55	4.14	4.50	3.98	4.29
Sauvola	15.16	15.34	**15.22**	14.83	15.14
Singh	12.99	14.99	13.69	13.28	13.74
Proposed	**15.86**	16.81	14.69	**16.25**	**15.90**

Bold values indicate the best experimental result compared with similar experiments

Table 3 Comparison of NRM for six algorithms

NRM ($\times 10^{-2}$)	2009	2010	2011	2012	Mean
OTSU	**5.84**	**8.87**	**6.90**	**7.78**	**7.35**
Niblack	9.69	9.96	11.31	9.26	10.05
Bernsen	26.12	30.12	27.38	30.79	28.60
Sauvola	5.93	22.21	9.00	19.03	14.04
Singh	12.22	12.01	18.99	13.91	14.28
Proposed	9.49	9.64	14.11	10.05	10.82

Bold values indicate the best experimental result compared with similar experiments

Table 5 Comparison of execution time for six algorithms

Execution time(s)	2009	2010	2011	2012	Mean
OTSU	0.463	0.274	0.215	0.192	0.286
Niblack	50.204	102.658	46.378	52.483	62.931
Bernsen	36.103	72.793	32.899	36.994	44.697
Sauvola	**0.116**	**0.189**	**0.123**	**0.137**	**0.141**
Singh	140.710	324.297	264.225	273.458	250.673
Proposed	0.774	1.150	0.655	0.712	0.823

Bold values indicate the best experimental result compared with similar experiments

els for which the background pixels have mistakenly been regarded as foreground. For instance in the proposed method and Otsu's method, the proposed method has a higher F-measure, but $\mathrm{NRM}_{\mathrm{otsu}} = \frac{\frac{765}{765+45,216}+\frac{60,069}{60,069+691,103}}{2} = 0.0483$ and $\mathrm{NRM}_{\mathrm{proposed}} = \frac{\frac{8963}{8963+37,013}+\frac{3180}{3180+747,992}}{2} = 0.0996$ This indicates that the binary image of Otsu's method has a large FP because of a great deal of noise. However, the image binarized by the proposed method effectively avoids the noise, and hence, it has a small FP. However, when widening the contrast for fuzzy character edges, the pixels in these edges may be divided into background mistakenly because of their minor contrast enhancement. This is why the FN of the proposed method is larger, and hence, $\mathrm{NRM}_{\mathrm{proposed}} > \mathrm{NRM}_{\mathrm{otsu}}$. This shows that a smaller error partition of the pixels in the output binary image may lead to a larger NRM. It will not affect the subsequent recognition work as long as the character strokes are not extremely fine. At the same time, it also illustrates that the algorithm is not suitable for blurry images with slender characters.

4.5 Execution time-based evaluation

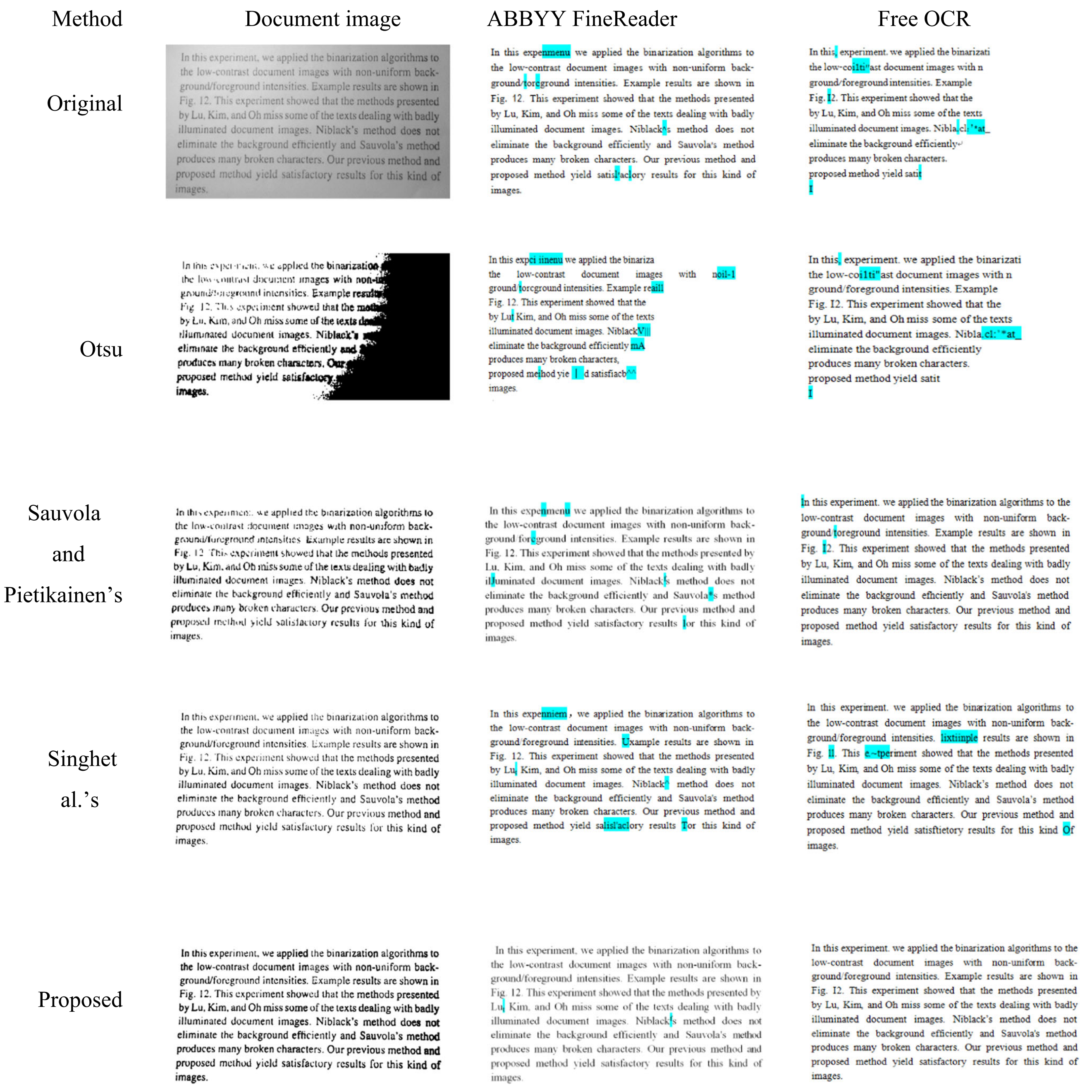
It is shown in Table 5 that the relatively high complexity of our algorithm is the reason why the average execution time by proposed method is not the fastest. It will inevitably lead to a longer execution time. But our algorithm in the MATLAB platform can still be completed within 1 second, and this can fully meet the needs of practical applications.

4.6 OCR-based evaluation

OCR-based comparison is one of the most acceptable methods for the quantitative evaluation of binarization algorithms

Table 4 Parameters for binary image by various algorithms

	FP	FN	TP	TN	FP + FN	F-measure (%)	NRM
OTSU	60,069	765	45,216	69,1103	60,834	59.78	0.0483
Niblack	90,053	247	45,734	661,119	90,300	50.32	0.0626
Bernsen	35,1743	7104	38,877	399,429	358,847	17.81	0.3114
Sauvola	15,090	3318	42,663	736,082	18,408	82.25	0.0461
Singh	3376	8853	37,128	747,796	12,229	85.86	0.0985
Proposed	3180	8968	37,013	747,992	12,148	85.90	0.0996



Method	Document image	ABBYY FineReader	Free OCR
Original	In this experiment, we applied the binarization algorithms to the low-contrast document images with non-uniform background/foreground intensities. Example results are shown in Fig. 12. This experiment showed that the methods presented by Lu, Kim, and Oh miss some of the texts dealing with badly illuminated document images. Niblack's method does not eliminate the background efficiently and Sauvola's method produces many broken characters. Our previous method and proposed method yield satisfactory results for this kind of images.	In this expenmenu we applied the binarization algorithms to the low-contrast document images with non-uniform background/torcground intensities. Example results are shown in Fig. 12. This experiment showed that the methods presented by Lu, Kim, and Oh miss some of the texts dealing with badly illuminated document images. Niblack^s method does not eliminate the background efficiently and Sauvola's method produces many broken characters. Our previous method and proposed method yield satisl'aclory results for this kind of images.	In this, experiment. we applied the binarizati the low-coi1ti"ast document images with n ground/foreground intensities. Example Fig. I2. This experiment showed that the by Lu, Kim, and Oh miss some of the texts illuminated document images. Nibla.cl:'*at_ eliminate the background efficiently↵ produces many broken characters. proposed method yield satit I
Otsu	In this experiment, we applied the binarization the low-contrast document images with non-u ground/foreground intensities. Example results Fig. 12. This experiment showed that the meth by Lu, Kim, and Oh miss some of the texts dea illuminated document images. Niblack's eliminate the background efficiently and produces many broken characters. Our proposed method yield satisfactory images.	In this expci iinenu we applied the binariza the low-contrast document images with noil-1 ground/torcground intensities. Example reaill Fig. 12. This experiment showed that the by Lut Kim, and Oh miss some of the texts illuminated document images. NiblackV|| eliminate the background efficiently mA produces many broken characters, proposed meihod yie | d satisfiacb^^ images.	In this, experiment. we applied the binarizati the low-coi1ti"ast document images with n ground/foreground intensities. Example Fig. I2. This experiment showed that the by Lu, Kim, and Oh miss some of the texts illuminated document images. Nibla.cl:'*at_ eliminate the background efficiently produces many broken characters. proposed method yield satit I
Sauvola and Pietikainen's	In this experiment, we applied the binarization algorithms to the low-contrast document images with non-uniform background/foreground intensities. Example results are shown in Fig. 12. This experiment showed that the methods presented by Lu, Kim, and Oh miss some of the texts dealing with badly illuminated document images. Niblack's method does not eliminate the background efficiently and Sauvola's method produces many broken characters. Our previous method and proposed method yield satisfactory results for this kind of images.	In this expenmenu we applied the binarization algorithms to the low-contrast document images with non-uniform background/forcground intensities. Example results are shown in Fig. 12. This experiment showed that the methods presented by Lu. Kim. and Oh miss some of the texts dealing with badly ilJuminated document images. Niblack!s method does not eliminate the background efficiently and Sauvola*s method produces many broken characters. Our previous method and proposed method yield satisfactory results lor this kind of images.	In this experiment. we applied the binarization algorithms to the low-contrast document images with non-uniform background/toreground intensities. Example results are shown in Fig. I2. This experiment showed that the methods presented by Lu, Kim, and Oh miss some of the texts dealing with badly illuminated document images. Niblack's method does not eliminate the background efhciently and Sauvola's method produces many broken characters. Our previous method and proposed method yield satisfactory results for this kind of images.
Singhet al.'s	In this experiment, we applied the binarization algorithms to the low-contrast document images with non-uniform background/foreground intensities. Example results are shown in Fig. 12. This experiment showed that the methods presented by Lu, Kim, and Oh miss some of the texts dealing with badly illuminated document images. Niblack's method does not eliminate the background efficiently and Sauvola's method produces many broken characters. Our previous method and proposed method yield satisfactory results for this kind of images.	In this expenniem, we applied the binarization algorithms to the low-contrast document images with non-uniform background/foreground intensities. Uxample results are shown in Fig. 12. This experiment showed that the methods presented by Lu; Kim, and Oh miss some of the texts dealing with badly illuminated document images. Niblack^ method does not eliminate the background efficiently and Sauvola's method produces many broken characters. Our previous method and proposed method yield salisl'aclory results Tor this kind of images.	In this experiment. we applied the binarization algorithms to the low-contrast document images with non-uniform background/foreground intensities. lixtiinple results are shown in Fig. ll. This e.~tperiment showed that the methods presented by Lu, Kim, and Oh miss some of the texts dealing with badly illuminated document images. Niblack's method does not eliminate the background efficiently and Sauvola's method produces many broken characters. Our previous method and proposed method yield satisftietory results for this kind Of images.
Proposed	In this experiment, we applied the binarization algorithms to the low-contrast document images with non-uniform background/foreground intensities. Example results are shown in Fig. 12. This experiment showed that the methods presented by Lu, Kim, and Oh miss some of the texts dealing with badly illuminated document images. Niblack's method does not eliminate the background efficiently and Sauvola's method produces many broken characters. Our previous method and proposed method yield satisfactory results for this kind of images.	In this experiment, we applied the binarization algorithms to the low-contrast document images with non-uniform background/foreground intensities. Example results are shown in Fig. 12. This experiment showed that the methods presented by Lu, Kim. and Oh miss some of the texts dealing with badly illuminated document images. Niblack!s method does not eliminate the background efficiently and Sauvola's method produces many broken characters. Our previous method and proposed method yield satisfactory results for this kind of images.	In this experiment. we applied the binarization algorithms to the low-contrast document images with non-uniform background/foreground intensities. Example results are shown in Fig. I2. This experiment showed that the methods presented by Lu, Kim, and Oh miss some of the texts dealing with badly illuminated document images. Niblack's method does not eliminate the background efficiently and Sauvola's method produces many broken characters. Our previous method and proposed method yield satisfactory results for this kind of images.

Fig. 9 Recognition results of the binarization image handled by each algorithm in OCRs

[25]. To test the recognition effect of various algorithms in OCR, this experiment tested images including all printed images in DIBCO datasets and an image randomly captured under non-uniform illumination. We selected an image as a representative. And we selected four algorithms with the best F-measure in Table 1 to process the degraded image, testing their recognition rate in ABBYY Fine Reader 12 [26] and Free OCR [27]. Figure 9 shows the recognition results of the binarized image handled by each algorithm in OCRs.

4.6.1 Qualitative analysis for Fig. 9

In Fig. 9, the first image in the upper left corner is the original gray image. It can be seen that the original image has a lighter background in the top left corner and a darker background in the bottom right corner because of its non-uniform illumination. Here, Otsu's method does not work well. Because it is a global threshold method, it cannot separate clear characters using the same threshold both in lighter and in darker backgrounds at the same time. Sauvola and Pietikainen's method,

Table 6 Recognition rate of various algorithms by two OCR programs

Correct rate of recognition (%)	ABBYY FineReader (%)	Free OCR (%)
Original	97.60	64.19
Otsu	63.97	62.45
Sauvola	98.47	99.34
Singh	96.29	96.94
Proposed	99.56	100

Singh et al.'s method, and the proposed algorithm are independent of non-uniform illumination and can separate the characters.

4.6.2 Quantitative analysis for Fig. 9

The CRR (correct rate of recognition) by OCR is defined as:

$$\mathrm{CRR} = \frac{N_{\mathrm{crc}}}{N_{\mathrm{total}}} \times 100\% \qquad (19)$$

where N_{crc} is the number of correctly recognized characters and N_{total} is the total number of characters.

Table 6 shows the CRRs for the original gray image and binary images of the four algorithms in ABBYY FineReader and Free OCR.

The combination of Fig. 9 and Table 6 shows that the original gray image with its non-uniform illumination has a very low rate of recognition in Free OCR. The binary image created by Otsu's method only has a 63% correct rate in the two OCRs because of its black area on the right side of the image. Although the images binarized by Sauvola and Pietikainen's and Singh et al.'s methods segment the characters, there is a problem in that the characters are stuck together, are incomplete, or have broken strokes. Hence, the images binarized by Sauvola and Pietikainen's and Singh et al.'s methods have a high error rate for both OCRs. However, the images produced by the proposed method have clear and complete characters that are easy to identify. The OCR programs can achieve a more than 99.5% recognition rate.

4.6.3 Quantitative analysis for images in datasets

At the same time, in order to further test the generality of proposed algorithm, this paper tested all printed images in datasets. Table 7 and Table 8 show the CRRs of the four algorithms in ABBYY FineReader and Free OCR. From the tables, it can be seen that the proposed algorithm gains the highest average CRR, and it is about 4.5% higher than the second highest one by original. The reason for this phenomenon is that other three algorithms have very low correct rates for some individual images, and it leads the average CRRs are pulled down. For example, the 2011-PR7 processed by Otsu's and Singh et al.'s methods, the 2011-PR6 processed by Sauvola and Pietikainen's and Singh et al.'s methods, the CRRs in above images are all 0 percentage. However, this is the case because many characters binarized by other three algorithms were broken in strokes. For instance, *e* was binarized as *c*, *m* was binarized as ni, etc. The images with broken strokes can still have a higher *F*-measure and PSNR. But, OCR will recognize a wrong character. So, it occurs with a higher *F*-measure, a higher PSNR, and a lower CRR. This illustrates that the above three algorithms have limitations for some images and may not have a good recognition

Table 7 Recognition rate of various algorithms by ABBYY FineReader

Correct rate of recognition (%)	Original (%)	Otsu (%)	Sauvola (%)	Singh (%)	Proposed (%)
2009-P01	57.95	65.34	52.27	58.52	**65.91**
2009-P02	65.91	**80.68**	78.41	73.86	77.27
2009-P03	46.24	62.37	54.84	48.39	**66.67**
2009-P04	**87.43**	77.05	75.96	82.51	83.06
2009-P05	**75.66**	71.05	43.42	46.71	73.68
2011-PR1	40.85	33.80	32.39	28.17	**60.56**
2011-PR2	**57.98**	32.45	17.55	53.19	43.09
2011-PR3	76.50	76.50	68.00	82.50	**85.00**
2011-PR4	74.10	76.51	77.71	78.31	**86.14**
2011-PR5	27.75	22.49	18.18	25.36	**33.97**
2011-PR6	92.96	**97.18**	42.25	47.89	91.55
2011-PR7	**64.10**	0.00	10.26	23.08	56.41
2011-PR8	83.43	**87.85**	64.09	60.77	85.08
Mean	65.45	60.25	48.87	54.56	**69.88**

Bold values indicate the best experimental result compared with similar experiments

Table 8 Recognition rate of various algorithms by Free OCR

Correct rate of recognition (%)	Original (%)	Otsu (%)	Sauvola (%)	Singh (%)	Proposed (%)
2009-P01	71.59	**73.86**	39.77	49.43	46.59
2009-P02	82.95	82.95	62.50	71.59	**84.09**
2009-P03	18.28	17.20	51.61	13.98	**68.82**
2009-P04	63.39	75.96	61.75	77.05	**81.42**
2009-P05	44.08	**56.58**	5.26	18.42	40.13
2011-PR1	11.27	5.63	**47.89**	7.04	16.90
2011-PR2	25.00	19.68	16.49	30.85	**33.51**
2011-PR3	62.50	53.00	50.50	75.50	**88.00**
2011-PR4	69.88	66.87	50.60	72.89	**75.90**
2011-PR5	28.71	28.71	25.84	27.27	**41.15**
2011-PR6	**56.34**	**56.34**	0.00	0.00	16.90
2011-PR7	12.82	0.00	7.69	0.00	**15.38**
2011-PR8	77.35	**82.32**	45.30	9.39	81.77
Mean	48.01	47.62	35.79	34.88	**53.12**

Bold values indicate the best experimental result compared with similar experiments

in troubled times. However, the proposed method is universal and has a relatively good recognition accuracy for most images. Meanwhile, the average recognition accuracy by our proposed method is the highest; these OCR results show the effectives of the proposed binarization technique.

5 Conclusion

Using the difference of gray contrast between regions, the method proposed in this paper adaptively divides significant areas and comparatively significant areas. For significant areas, weak contrast enhancement is used to magnify the difference between the foreground and background. Meanwhile, weak contrast enhancement is able to reduce noise in the results. For comparatively significant areas, strong contrast enhancement is used to adjust gray values so that the method can easily distinguish between foreground and background, and clear characters can be separated. Hence, no matter the type of area (including variable background, non-uniform illumination, and bleed-through), there is always an appropriate method that will achieve satisfactory results. Furthermore, the proposed method is particularly effective for degraded document images with bleed-through and severely uneven illumination. The experimental results show that, in the results obtained from DIBCO image set processed by the six algorithms, the binarized images handled by the proposed method have clear and complete characters as well as mostly noise-free backgrounds. Meanwhile, the images binarized using the proposed algorithm achieve the highest F-measure and PSNR. When compared with the OCR results of four top binarization methods, the proposed method obtains the highest CRR.

References

1. http://baike.baidu.com/link?url=hdU9FF9dpP8-x2hi1TNXNZdvydJNlhBUmtjf3-NOli6N-OfmwvAyLRerfozUyvh9UYCTwR3MXeWb9Jga_CJxLK
2. Wen, J., Li, S., Sun, J.: A new binarization method for non-uniform illuminated document images. Pattern Recognit. **46**, 1670–1690 (2012)
3. Cheng, H.D., Chen, Y.H.: Fuzzy partition of two-dimensional histogram and its application to thresholding. Pattern Recognit. **32**(5), 825–843 (1999)
4. Cinque, L., Di Zenzo, S., Levialdi, S.: Image thresholding using fuzzy entropies. IEEE Trans. SMC **28**(1), 2–15 (1998)
5. Papamarkos, N.: A technique for fuzzy document binarization. In: Proceedings of the 2001 ACM Symposium on Document Engineering, Atlanta, Georgia, USA, ACM, pp. 152–156 (2001)
6. Chou, C.H., Lin, W.H., Chang, F.: A binarization method withl earning-built rules for document images produced by cameras. Pattern Recognit. **43**, 1518–1530 (2010)
7. Mesquita*, R.G. , Mello , C.A.B., Almeida,L.H.E.V.: A new thresholding algorithm for document images based on the perception of objects by distance. Integr. Comput.-Aided Eng. **21**, 133–146 (2014)
8. Otsu, N.: A threshold selectionmethod from gray level histograms. IEEE Trans. Syst. Man Cybern. **9**, 62–66 (1979)
9. Kapur, J.N., Sahoo, P.K., Wong, A.K.C.: A new method for graylevel picture thresholding using the entropy of the histogram. Comput. Vis. Graph. Image Process. **29**, 273–285 (1985)
10. Kittler, J., Illingworth, J.: Minimum error thresholding. Pattern Recognit. **19**(1), 41–47 (1986)
11. Abutaleb, A.S.: Automatic thresholding of gray-level pictures using two-dimensional entropy. Comput. Vis. Graph. Image Process. **47**, 22–32 (1989)
12. Bernsen, J.: Dynamic thresholding of grey-level images. In: Proceedings of ICPR'86, pp. 1251–1255 (1986)
13. Niblack, W.: An Introduction to Digital Image Processing, pp. 115–116. Prentice-Hall, Englewood Cliffs (1986)
14. Sauvola, J., Pietikainen, M.: Adaptive document image binarization. Pattern Recognit. **33**, 225–236 (2000)

15. Kim, I.K., Jung, D.W., Park, R.H.: Document image binarization based on topographic analysis using a water flow model. Pattern Recognit. **35**, 265–277 (2002)
16. Valizadeh, M., Komeili, M., Armanfard, N., Kabir, E.: Degraded document image binarization based on combination of two complementary algorithms. In: Proceedings of ICACTEA'09, IEEE, pp. 595–599 (2009)
17. Pai, Y.T., Pai, Y.F., Ruan, S.J.: Adaptive thresholding algorithm: efficient computation technique based on intelligent block detection for degraded document images. Pattern Recognit. **9**, 3177–3187 (2010)
18. Singh, B.M., Sharma, R., Ghosh, D., Mittal, A.: Adaptive binarization of severely degraded and non-uniformly illuminated documents. Proc. Int. J. Doc. Anal. Recognit. (IJDAR) **17**(4), 393–412 (2014)
19. Gatos, B., Pratikakis, I., Perantonis, S.J.: Adaptive degraded document image binarization. Pattern Recognit. **39**, 317–327 (2006)
20. Rosenfeld, A., Kak, A.C.: Digital Picture Processing, 2nd edn. Academic Press, New York (1982)
21. Users. iit. demokritos. gr/ bgat/ DIBCO 2009/ benchmark/
22. Users. iit. demokritos. gr/ bgat/ H-DIBCO 2010/ benchmark/
23. Utopia. duth. gr/ ipratika/ DIBCO 2011/ benchmark/
24. Utopia. duth. gr/ ipratika/ HDIBCO 2012/ benchmark/
25. Pratikakis, I., Gatos, B.,Ntirogiannis, K.: ICFHR2012 competition on handwritten document image binarization (H-DIBCO 2012). In: 2012 International Conference on Frontiers in Handwriting Recognition(ICFHR), pp. 813–818 (2012)
26. http://www.abbyy.cn/finereader/
27. http://www.freeocr.net/