Document image binarization using background estimation and stroke edges

Article *in* Document Analysis and Recognition · December 2010

DOI: 10.1007/s10032-010-0130-8 · Source: DBLP

3 authors:

Shijian Lu
Nanyang Technological University
162 PUBLICATIONS **3,208** CITATIONS

SEE PROFILE

Bolan Su
Institute for Infocomm Research
31 PUBLICATIONS **1,014** CITATIONS

SEE PROFILE

Chew Lim Tan
National University of Singapore
425 PUBLICATIONS **9,618** CITATIONS

SEE PROFILE

IJDAR (2010) 13:303–314
DOI 10.1007/s10032-010-0130-8

ORIGINAL PAPER

Document image binarization using background estimation and stroke edges

Shijian Lu · Bolan Su · Chew Lim Tan

Received: 6 December 2009 / Revised: 26 July 2010 / Accepted: 1 October 2010 / Published online: 21 October 2010

Abstract Document images often suffer from different types of degradation that renders the document image binarization a challenging task. This paper presents a document image binarization technique that segments the text from badly degraded document images accurately. The proposed technique is based on the observations that the text documents usually have a document background of the uniform color and texture and the document text within it has a different intensity level compared with the surrounding document background. Given a document image, the proposed technique first estimates a document background surface through an iterative polynomial smoothing procedure. Different types of document degradation are then compensated by using the estimated document background surface. The text stroke edge is further detected from the compensated document image by using L1-norm image gradient. Finally, the document text is segmented by a local threshold that is estimated based on the detected text stroke edges. The proposed technique was submitted to the recent document image binarization contest (DIBCO) held under the framework of ICDAR 2009 and has achieved the top performance among 43 algorithms that are submitted from 35 international research groups.

S. Lu (✉)
Department of Computer Visoin and Image Understanding,
Institute for Infocomm Research, 1 Fusionopolis Way,
#21-01 Connexis, Singapore 138632, Singapore
e-mail: slu@i2r.a-star.edu.sg

B. Su · C. L. Tan
Department of Computer Science, School of Computing,
National University of Singapore, Computing 1,
13 Computing Drive, Singapore 117417, Singapore
e-mail: subolan@comp.nus.edu.sg

C. L. Tan
e-mail: tancl@comp.nus.edu.sg

Keywords Document image analysis · Document image binarization · Document background estimation · Polynomial smoothing

1 Introduction

Document image binarization is often performed in the preprocessing stage of different document image processing related applications such as optical character recognition (OCR) and document image retrieval. It converts a gray-scale document image into a binary document image and accordingly facilitates the ensuing tasks such as document skew estimation and document layout analysis. As more and more text documents are scanned, fast and accurate document image binarization is becoming increasingly important.

Though document image binarization has been studied for many years, the thresholding of degraded document images is still an unsolved problem. This can be explained by the difficulty in modeling different types of document degradation such as uneven illumination, image contrast variation, bleeding-through, and smear that exist within many document images as illustrated in Fig. 1. The recent document image binarization contest (DIBCO)[1] held under the framework of International Conference on Document Analysis and Recognition (ICDAR) 2009 particularly addresses this issue by creating a challenging benchmarking dataset and evaluating the recent advances in document image binarization. The contest received 43 algorithms from 35 international research groups, partially reflecting the current efforts on this task as well as the common understanding that further efforts are required for better document image binarization solutions.

[1] http://users.iit.demokritos.gr/~bgat/DIBCO2009/benchmark.

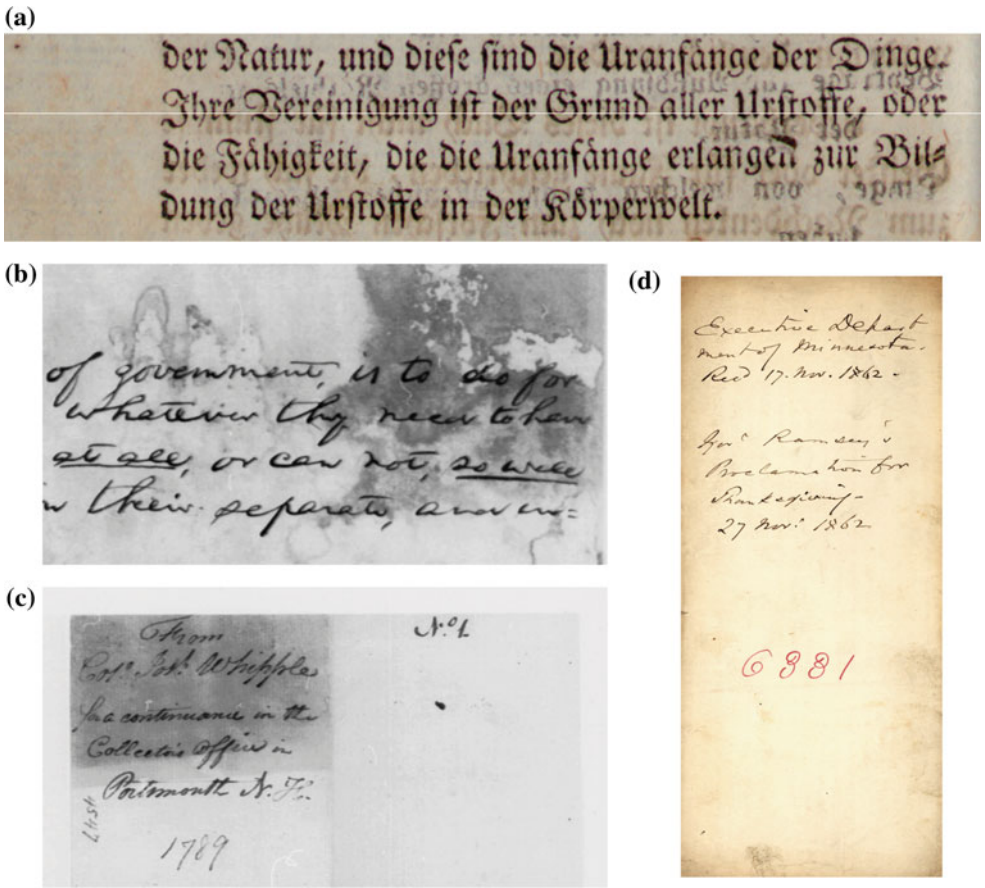

Fig. 1 Example document images in DIBCO's dataset that illustrate document degradation including bleeding-through in (**a**), image contrast variation and smear in (**b**) and (**c**), and uneven illumination in (**d**)

A large number of document image thresholding techniques [1,2] have been reported in the literature. For document images of a good quality, global thresholding [3–6] is capable of extracting the document text efficiently. But for document images suffering from different types of document degradation, adaptive thresholding, which estimates a local threshold for each document image pixel, is usually capable of producing much better binarization results. One typical adaptive thresholding approach is window based [7–12], which estimates the local threshold based on image pixels within a neighborhood window. However, the performance of the window-based methods depends heavily on the window size that cannot be determined properly without prior knowledge of the text strokes. At the same time, some window-based method such as Niblack's [11] often introduces a large amount of noise and some method such as Sauvola's [12] is very sensitive to the variation of the image contrast between the document text and the document background.

Some adaptive document thresholding methods [13–18] have also been reported that make use of the document-specific domain knowledge. In particular, one adaptive document thresholding approach is to first estimate a document background surface and then estimate a thresholding surface based on the estimated background surface. For example, Gatos et al. [13] estimate the document background surface based on the binary document image generated by Sauvola's thresholding method [12]. Moghaddam et al. [14] instead estimate the document background surface through an adaptive and iterative image averaging procedure. In addition, some adaptive document thresholding methods make use of the image edges that can usually be detected around the text stroke boundary. For example, Chen et al. [16] propose to first detect and close image edges and then obtain a primary binary document images based on the determined edge information. Moghaddam et al. [15] instead make use of the edge profile to locate the text region and accordingly estimate the local image threshold. Su et al. [17] also attempt to locate the text stroke edges by using an image contrast that is evaluated based on the local maximum and minimum.

This paper describes our algorithm submitted to the DIBCO 2009 that has achieved the top performance among the 43 submitted algorithms [19]. The submitted algorithm makes use of both the document background and the text stroke

edge information. In particular, it first estimates a document background surface through an iterative polynomial smoothing procedure. The variation of the image contrast resulting from document degradation such as shading and smear is then compensated by using the estimated document background surface. The text stroke edges are then detected based on the local image variation within the compensated document image. After that, the document text is extracted based on the local threshold that is estimated from the detected text stroke edge pixels. At the end, a series of post-processing operations are performed to further improve the binarization results.

One characteristic of our proposed method is that it first estimates a document background surface through an one-dimensional iterative polynomial smoothing procedure [20]. Compared with the document background surface estimated in [13,14], the document background surface estimated through polynomial smoothing is smoother and closer to the real document background surface. Therefore, it is more suitable for the compensation of the variation of the document image contrast that often results from certain document degradation such as uneven illumination and smear. In addition, the proposed method makes use of the text stroke edges to estimate the local threshold and accordingly overcomes the limitations of many existing adaptive thresholding methods such as those window-based methods [7–12] that often falsely detect text pixels from the document background. Furthermore, it makes use of L1-norm image gradient that is often more subitable (compared with the traditional edge detector and the edge profile used in [15,16]) for the text stroke edge detection based on our empirical observations.

The rest of this paper is arranged as follows: Section 2 first presents the proposed document binarization method in detail. Experimental results are then described and discussed in Sect. 3. Finally, some concluding remarks are summarized in Sect. 4.

2 Proposed method

The section presents the proposed document image binarization method. In particular, we will divide this section into five subsections, which deal with polynomial smoothing, document background estimation, text stroke edge detection, local threshold estimation, and post-processing, respectively.

2.1 Polynomial smoothing

The proposed technique makes use of a document background surface that is estimated through an iterative polynomial smoothing procedure. We therefore first give a brief introduction of smoothing and polynomial smoothing. Smoothing is a process by which signals are weighted within a local neighborhood window. For a series of signals $[s_1, s_2, \ldots, s_n]$, the new series of signals $[f_1, f_2, \ldots, f_n]$ after the smoothing can be represented as follows:

$$f_k = \sum_{-n<i<n} w_i s_{k+i} \tag{1}$$

where w_i denotes the weighs and n denotes the size of the local neighborhood window. Therefore, the smoothed signal f_k is actually a weighted combination of the original signal s_k and its neighbors within a neighborhood window.

The polynomial smoothing (also called Savitzky–Golay smoothing [21]) aims to fit a least square polynomial function to the signals within a local neighborhood window. The smoothed signal f_k is estimated as the value of the fitted polynomial function at the same coordinate. Given a set of data within a local neighborhood window, the smoothing polynomial function of order d can be represented in Eq. 2 as follows:

$$f(x) = \sum_{i=0}^{d} a_i s^i \tag{2}$$

where $[a_d, \ldots, a_0]$ refer to the coefficients of the smoothing polynomial function, which can be estimated from the signals within the neighborhood window as follows:

$$A = (S^T \cdot S)^{-1} \cdot S^T \cdot I \tag{3}$$

where I refers to the signal within the local neighborhood window and the matrix S is constructed as follows:

$$S = \begin{pmatrix} 1 & s_1 & s_1^2 & \cdots & s_1^d \\ 1 & s_2 & s_2^2 & \cdots & s_2^d \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & s_n & s_n^2 & \cdots & s_n^d \end{pmatrix}$$

where n refers to the number of signals within the local neighborhood window.

2.2 Document background estimation

We estimate the document background surface through polynomial smoothing. Polynomial smoothing has been used in many different applications for the background surface estimation. For example, Krzysztof et al. [22] make use of the polynomial smoothing to estimate the fingerprint background where a local two-dimensional polynomial surface is fitted by using fingerprint pixels within a sliding window. Seeger et al. [23] make use of the local polynomial smoothing to estimate the background surface of the pre-detected document text regions. In addition, we also studied the document background estimation through two-dimensional polynomial smoothing as reported in [24].

We implement the polynomial smoothing in a different way. First, we estimate the document background surface

through one-dimensional polynomial smoothing [20] that is usually much faster (up to ten times) and also more accurate than the two-dimensional polynomial smoothing [24]. Second, we perform the global polynomial smoothing, which fits a smoothing polynomial to the image pixels within each whole document row/column and therefore requires no pre-detection of the text regions. As the text documents usually have a background of the same color and texture, the global smoothing polynomial is usually capable of tracking the image variation within the document background accurately. Third, we perform the polynomial smoothing iteratively that updates the polynomial order and the data points adaptively after each round of smoothing. The iterative smoothing further improves the accuracy of the estimated document background surface.

In the proposed polynomial smoothing, a set of equidistant pixels are first sampled from a document row/column. The signal at each sampling pixel is estimated by the median intensity of the document image pixels within a local one-dimensional neighborhood window. The initial smoothing setup can be specified as follows:

$$\begin{aligned} x_i &= k_s \times i \\ S_i &= f_{mdn}([I(x_{f_{rnd}(i-k_s)}), \ldots, I(x_{f_{rnd}(i+k_s)})]), \\ &\quad i = 1, \ldots, N \end{aligned} \tag{4}$$

where functions $f_{mdn}(\cdot)$ and $f_{rnd}(\cdot)$ denote a median and a rounding functions, respectively. x_i and s_i refer to the position of the i-th sampling pixel and the sampled image intensity at that sampling pixel. The sampling index i changes from 1 to N where N refers to the number of the image pixels sampled from the document row/column under study. Parameter k_s denotes the sampling step. Our experiments show that the document thresholding performance changes little when k_s changes from 1 to 6.

The background surface of the document row/column under study can thus be estimated through an iterative polynomial smoothing procedure specified in Algorithm 1.

As described in Algorithm 1, we pre-define a threshold to stop the iterative polynomial smoothing procedure. In our implemented system, the pre-defined threshold is set at 10 because the intensity difference between the document text pixels and the document background pixels is usually much larger than 10. In addition, we set the initial polynomial order d_o at 6 based on the observation that the polynomial of order 6 in the initial iteration is usually sufficient to track the image variation within the document background. Furthermore, we increase the polynomial order adaptively (after each smoothing iteration) as follows to estimate the document background surface accurately:

$$d_n = d_o + f_{rnd}(k_t \cdot n) \tag{5}$$

Algorithm 1 Polynomial smoothing of one row/column of a document image

Require:
One row/column document image pixels
Ensure:
A smoothing polynomial of the background of the document image row/column under study
1: Sample the image data from the document row/column under study as specified in Eq. 4.
2: Fit a smoothing polynomial of the initial order d_0 to the sampled image data.
3: Evaluate the maximum fitting error between the sampled data and the fitted smoothing polynomial. Remove the sampling point with the maximum fitting error if the maximum fitting error is larger than a pre-defined threshold (to be discussed next).
4: Refit a smoothing polynomial of a higher order d_n (to be discussed next) to the remaining data points;
5: Repeat the previous two steps iteratively until the maximum fitting error is smaller than the pre-defined threshold or the number of the remaining data points is smaller than d_n.
6: **return** The final smoothing polynomial

where n denotes the iteration number and $f_{rnd}(\cdot)$ refers to a rounding function. d_o and d_n denote the order of the initial smoothing polynomial and the smoothing polynomial at the n^{th} iteration, respectively. Parameter k_t specifies the increase speed of the polynomial order that can be set between 0.1 and 0.2.

The blue graphs in Fig. 2a, b show the image pixel intensity within the document row labeled in the document image in Fig. 1a. The black graphs show the fitted initial and the final smoothing polynomials as described in Algorithm 1. As Fig. 2a shows, the initial smoothing polynomial does not track the document background variation properly but the one after multiple rounds of smoothing iterations tracks the document background much more accurately. Figure 3a further shows the document background surface estimated through the row-by-row smoothing procedure. As Fig. 3a shows, the background of most image rows is estimated accurately except a small number of image rows.

We therefore further perform a column-by-column smoothing procedure to correct the estimation error that is introduced through the row-by-row smoothing procedure. The column-by-column smoothing is very similar to the row-by-row smoothing as described in Algorithm 1. The only difference is that the image data are sampled not from the original document image but from the document background surface estimated in the row-by-row smoothing stage. Figure 3b shows the document background surface that is estimated in the column-by-column smoothing procedure. As Fig. 3b shows, the column-by-column smoothing properly corrects the error that is introduced in the row-by-row smoothing procedure. The estimated document background surface will be used in document image normalization as well as post-processing to be described next.

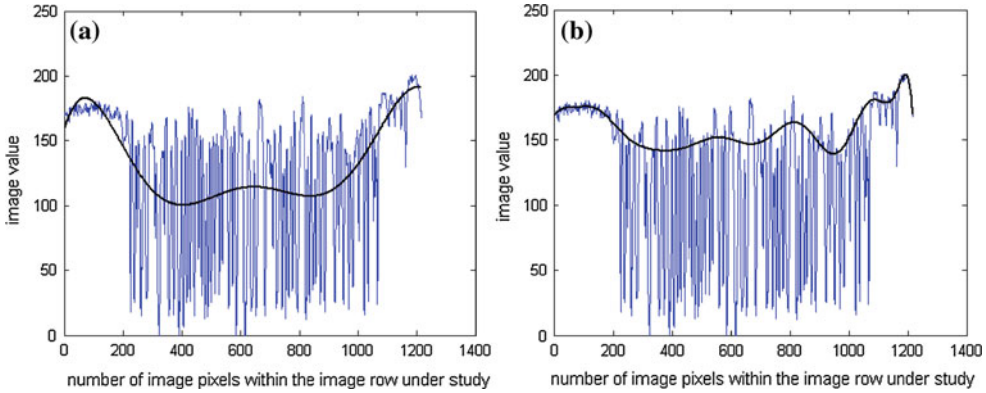

Fig. 2 Iterative polynomial smoothing: **a** The intensity of one image row (*blue graph*) labeled in Fig. 1a and the fitted initial smoothing polynomial (*black bold graph*); **b** The final smoothing polynomial (*black bold graph*) after multiple rounds of smoothing of the image row labeled in Fig. 1a

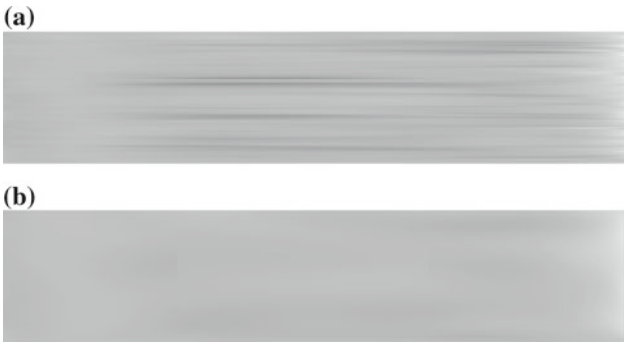

Fig. 3 Document background surface estimation: **a** The document background surface that is estimated through the row-by-row polynomial smoothing procedure; **b** The final document background surface that is estimated after the column-by-column smoothing procedure

2.3 Text stroke edge detection

The stroke edge as a strong text indicator has been used for document image thresholding [15,16]. But for degraded document images, stroke edges may not be detected properly due to various types of document degradation. In particular, certain amount of non-stroke edges may be detected due to the high variation such as noise within the document background. At the same time, certain amount of real text stroke edges may not be detected because of the low image contrast that often results from different types of document degradation such as uneven illumination or document smear.

We detect the text stroke edges based on the local image variation. Before the evaluation of the local image variation, the "global" variation of the document image contrast (often resulting from document degradation such as uneven illumination and smear) is first compensated so that the text stroke edges can be better detected in the ensuing operations. The document contrast compensatation is performed by using the estimated document background surface described in the last subsection as follows:

$$\overline{I} = \frac{C}{\mathrm{BG}} \times I \tag{6}$$

where C is a constant that controls the brightness of the compensated document images. In our implemented system, it is set at the median intensity of the document image under study to preserve the original document brightness. BG stands for the estimated document background surface. The image variation within the document background can therefore be compensated because the compensation factor, i.e., $\frac{C}{\mathrm{BG}}$, will be large at the dark document regions due to the relatively small BG but will be small at the bright document regions because of the relatively large BG.

Different text stroke edge detection methods have been reported such as those using the traditional edge detector [16] and edge profile [15]. However, we empirically observed that many edge pixels detected by either the edge profile or the traditional edge detector do not correspond to the real text stroke edges within document images. Instead, the text stroke edge pixels can be better detected from the ones that have the maximum L1-norm image gradient in either horizontal or vertical direction as follows:

$$\begin{aligned} V_h(x, y) &= |\overline{I}(x, y+1) - \overline{I}(x, y-1)| \\ V_v(x, y) &= |\overline{I}(x+1, y) - \overline{I}(x-1, y)| \end{aligned} \tag{7}$$

where $\overline{I}$ denotes the normalized document image under study. We therefore first detect a number of candidate text stroke edge pixels by the ones that have the maximum L1-norm image gradient in either horizontal or vertical direction.

The local image variation at each candidate text stroke edge pixel is then evaluated by combining the L1-norm image gradient in horizontal and vertical directions as follows:

$$V(x, y) = V_h(x, y) + V_v(x, y) \tag{8}$$

where $V_h(x, y)$ and $V_v(x, y)$ denote the L1-norm image gradient in horizontal and vertical direction as defined in Eq. 7. For the sample document image in Figs. 1a, 4a shows the

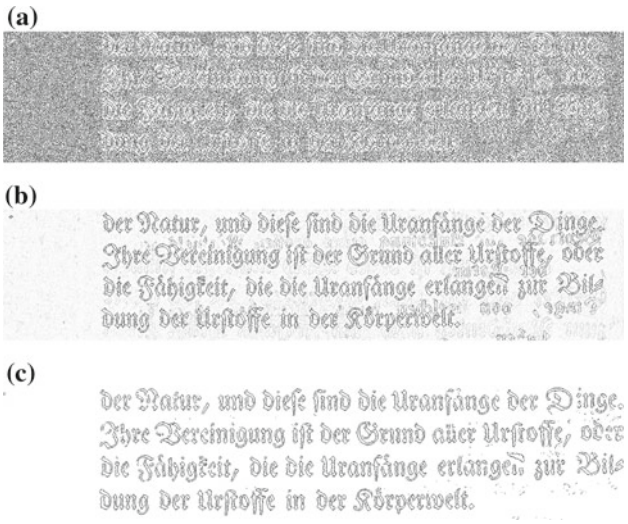

Fig. 4 Text stroke edge pixel detection: **a** The candidate stroke edge pixels detected from the sample document in Fig. 1a; **b** The local image variation of the candidate text stroke edge pixels in Fig. 4a; **c** The text stroke edge detected through the thresholding of the image variation histogram in Fig. 5

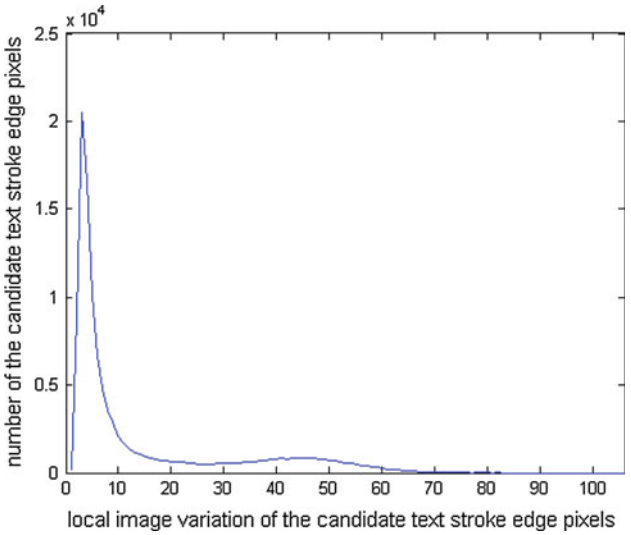

Fig. 5 The histogram that is built based on the local image variation (defined in Eq. 8) of the detected candidate text stroke edge pixels shown in Fig. 4b

candidate text stroke edge pixels that are detected by the ones having either the maximum $V_h(x, y)$ or the maximum $V_v(x, y)$. Figure 4b shows the local image variation of the detected candidate text stroke edge pixels as evaluated in Eq. 8.

The histogram of the local image variation of the detected candidate stroke edge pixels usually has a bimodal pattern. In particular, the local image variation of the real stroke edge pixels is much larger than that of the non-stroke edge pixels such as those detected around the bleeding-through shown in Fig. 4a, b. Figure 5 shows the histogram of the local image variation of the candidate text stroke edge pixels shown in Fig. 4a. As Fig. 5 shows, the peak on the left formed by the non-stroke edge pixels has a small local image variation whereas the one on the right formed by the real text stroke edge pixels has a much larger local image variation. The real text stroke edge pixels can therefore be detected by using Otsu's global thresholding method [3] based on such bimodal histogram pattern. For the detected candidate stroke edge pixels in Fig. 4a, c shows the finally determined text stroke edge pixels where most stroke edge pixels are determined properly.

2.4 Local threshold estimation

Once the text stroke edges are detected, the document text can be extracted based on the observation that the document text is surrounded by text stroke edges and also has a lower intensity level compared with the detected stroke edge pixels. The document text is extracted based on the detected text stroke edges as follows:

$$R(x, y) = \begin{cases} 0 & N_e \geq N_{\min} \quad \&\& \quad x\overline{I}(x, y) \leq E_{\text{mean}} \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

where $\overline{I}$ refers to the normalized document image under study. N_e refers to the number of the detected stroke edge pixels within a local neighborhood window. $N_{\min}$ denotes a threshold that specifies the minimum number of detected stroke edge pixels (within the neighborhood window) that is required to consider the image pixel under study as a possible text pixel. E_{mean} refers to the mean image intensity of the detected stroke edge pixels within the local neighborhood window that can be determined as follows:

$$E_{\text{mean}} = \frac{\sum_{\text{neighbor}} \overline{I}(x, y) * (1 - E(x, y))}{N_e} \tag{10}$$

where E refers to the determined stroke edge image shown in Fig. 4c. As Eq. 9 shows, the image pixel will be classified as a text pixel if N_e is larger than $N_{\min}$ and $\overline{I}(x, y)$ is smaller than E_{mean}. Otherwise, it will be classified as a background pixel.

As described earlier, the performance of the proposed document image binarization using the text stroke edges depends on two parameters, namely the size of the neighborhood window and the minimum number of the text stroke edge pixels within the neighborhood window $N_{\min}$. Both parameters are closely related to the width of text strokes within the document image under study. In particular, the size of the neighborhood window should not be smaller than the text stroke width. Otherwise, the text pixels in the interior of the text strokes will not be extracted properly because there may not be sufficient text stroke edge pixels within the local neighborhood window. At the same time, the threshold number of the text stroke edge pixels $N_{\min}$ (within the local neighborhood

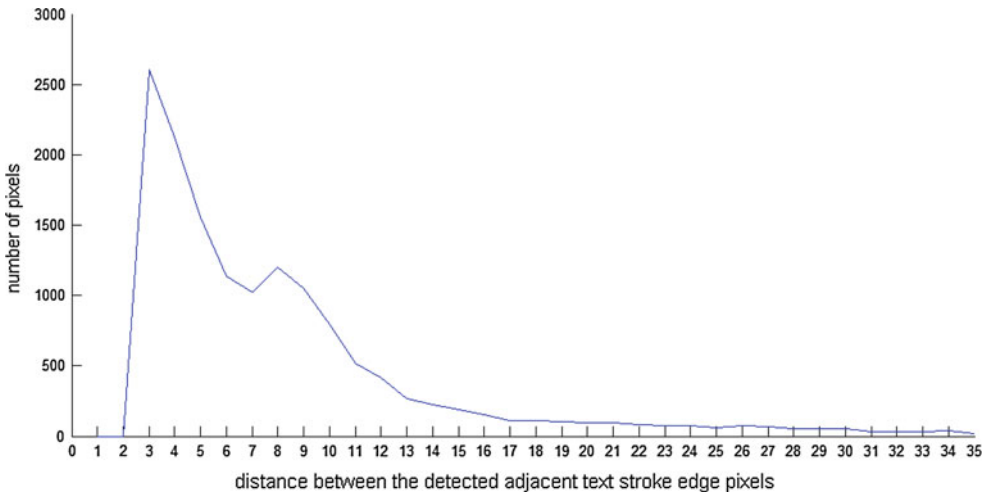

Fig. 6 The histogram that records the frequency of the distance between the detected adjacent text stroke edge pixels shown in Fig. 4c

window) should be more or less larger than the window size (if the window size is larger than the text stroke width) due to the double-edge structure of the text strokes.

The text stroke width therefore needs to be estimated before the document image thresholding. We estimate the text stroke width based on the detected text stroke edges shown in Fig. 4c. In particular, we scan the stroke edge image row-by-row and record the distance between all adjacent stroke edge pixel pairs in each row. The stroke width is then estimated based on the recorded text stroke edge distance as follows:

$$W = \underset{i}{\text{argmax}}\, H \tag{11}$$

where H denotes a histogram that accumulated the frequency of the distance between two adjacent text stroke edge pixels. Therefore, the text stroke width is estimated by the most frequent distance within the built distance histogram. Such estimation is based on two observations. First, the proposed text stroke edge detection method is able to detect most text stroke edges properly as illustrated in Fig. 4c. Second, for most scanned text document images, the most frequent distance between adjacent stroke edges exactly corresponds to the stroke width. For the detected text stroke edge pixels shown in Fig. 4c, Fig. 6 shows the constructed edge distance histogram where a global peak can be easily located.

The size of the thresholding window can therefore be determined based on the estimated stroke width. Generally, the document thresholding performance is not so sensitive to the window size when the window size is bigger than the real stroke width. Our experiments that change the window size from 0.5 to 6 times of the estimated stroke width show that the thresholding performance changes little when the window size changes from 1.5 to 4.5 times of the estimated stroke width. The window size can therefore be set at 2–4 times of the estimated stroke width in practice. The edge number threshold N_{min} can be set around the same as the estimated text stroke width based on the double-edge structure of the text strokes. For the document image in Fig. 1a, Fig. 7 shows the resultant binary document image determined by Eqs. 9 and 10 where the window size and the $N_{\min}$ are set at 2 and 1 times of the estimated stroke width, respectively.

2.5 Post-processing

Document image thresholding often introduces a certain amount of error as illustrated in Fig. 7 that can be corrected through a series of post-processing operations. We correct the document thresholding error by three post-processing operations based on the estimated document background surface and some document domain knowledge. In particular, we first remove text components (labeled through connected component analysis) of a very small size that often result from image noise such as salt and pepper noise. Based on the observation that the real text components are usually composed of much more than 3 pixels, we simply remove the text components that contain no more than 3 pixels in our system.

Next, we remove the falsely detected text components that have a relatively large size. The falsely detected text components of a relatively large size are identified based on the observation that they are usually much brighter than the

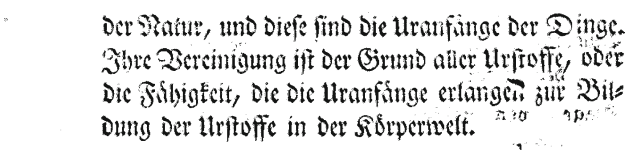

Fig. 7 Document image thresholding result: the binarization result of the document image in Fig. 1a based on the detected text stroke edge pixels in Fig. 4c

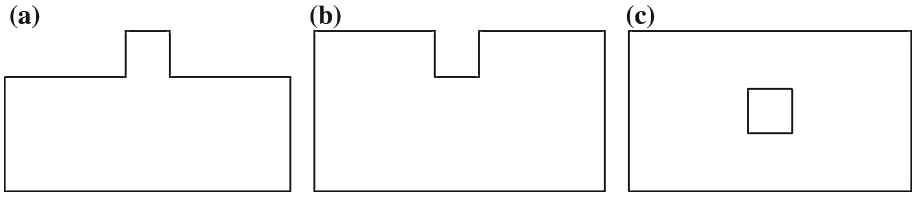

Fig. 8 Single-pixel artifacts along the text stroke boundary after the document thresholding: **a–b** The single-pixel concavity and convexity along the text stroke boundary where only upward single-pixel artifacts are plotted for the illustration purpose; **c** The single-pixel hole that is often detected from the interior of the text strokes due to image noise

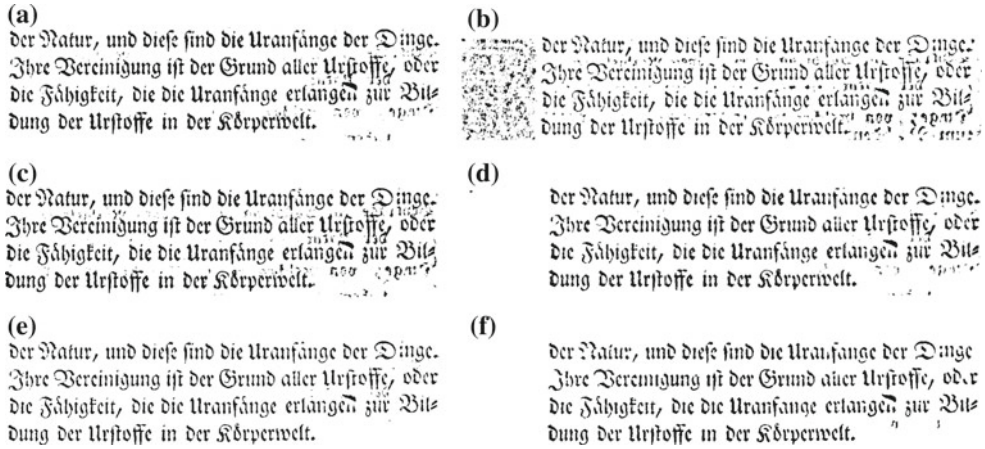

Fig. 9 Binarization results of the sample document image in Fig. 1(a) by using Otsu's method in (**a**), Niblack's method in (**b**), Sauvola's method in (**c**), Gatos's method in (**d**), Su's method in (**e**), and our proposed method in (**f**)

surrounding real text strokes. We capture such observation by the image difference between the labeled text component and the corresponding patch within the estimated document background surface as follows:

$$\mathrm{Diff}(c) = |\mathrm{BG}_c - I_c| \tag{12}$$

where I_c and BG_c denote the intensity of the text component under study and the value of the corresponding document background region, respectively. In our system, we first determine the median of the image difference of all labeled text components. Based on our empirical experiments that the image difference of the real text components is usually much larger than 0.4 of the median image difference, the falsely detected text components of a relatively large size can therefore be identified and removed if their image difference is smaller than 0.2–0.4 of the median image difference (set at 0.3 in our system).

Last, document image thresholding often introduces a certain amount of single-pixel holes, concavities, and convexities along the text stroke boundary. Figure 8 illustrates the patterns of these single pixel defects where Fig. 8a, b just show the pattern of upward convexities and concavities, respectively. These single pixel defects are actually artifacts, which can be removed by using certain logical operators that can be simply set according to their neighborhood patterns as illustrated in Fig. 8. For the binary document image in Fig. 7, Fig. 9f shows the final binarization result after the post-processing where most thresholding error is corrected properly.

3 Experiments and discussion

3.1 Experiment setup

The described document image thresholding method has been tested on the document images used in the DIBCO 2009 [2] that suffer from different types of representative document degradation shown in Fig. 1. In addition, we also compare our method with five state-of-art document image binarization methods including Otsu's global thresholding method [3], Niblack's, Sauvola's, Gatos's, and Su's adaptive thresholding methods [11–13,17]. The parameters of the adaptive thresholding methods such as the window size, the weights of local mean, standard variation, and dynamic range of standard variation used in [11–13,17] are all set according to the recommendations within the reported papers.

[2] http://users.iit.demokritos.gr/~bgat/DIBCO2009/benchmark.

3.2 Experimental results

The evaluation measures are adapted from the DIBCO report [19] including F-measure, peak signal-to-noise ratio (PSNR), negative rate metric (NRM), and misclassification penalty metric (MPM). In particular, the F-measure is defined as follows:

$$\mathrm{FM} = \frac{2 * \mathrm{RC} * \mathrm{PR}}{\mathrm{RC} + \mathrm{PR}} \tag{13}$$

where RC and PR refer the binarization recall and the binarization precision, respectively. This metric measures how well an algorithm can retrieve the desire pixels. The PSNR is defined as follows:

$$\mathrm{PSNR} = 10 \log \left(\frac{C^2}{\mathrm{MSE}} \right) \tag{14}$$

where MSE denotes the mean square error and C is a constant and can be set at 1. This metric measures how close the result image to the ground truth image. The NRM is defined as follows:

$$\mathrm{NRM} = \frac{\frac{N_{\mathrm{FN}}}{N_{\mathrm{FN}}+N_{\mathrm{TP}}} + \frac{N_{\mathrm{FP}}}{N_{\mathrm{FP}}+N_{\mathrm{TN}}}}{2} \tag{15}$$

where N_{TP}, N_{FP}, N_{TN}, N_{FN} denote the number of true positives, false positives, true negatives, and false negatives respectively. This metric measures pixel mismatch rate between the ground truth image and result image. The MPM is defined as follows:

$$\mathrm{MPM} = \frac{\sum_{i=1}^{N_{\mathrm{FN}}} d_{\mathrm{FN}}^{i} + \sum_{j=1}^{N_{\mathrm{FP}}} d_{\mathrm{FP}}^{j}}{2D} \tag{16}$$

where d_{FN}^{i} and d_{FP}^{j} denote the distance of the ith false negative and the jth false positive pixel from the contour of the ground truth segmentation. The normalization factor D is the sum over all the pixel-to-contour distances of the ground truth object. This metric measures how well the result image represents the contour of ground truth image.

Experimental results are shown in Table 1. As Table 1 shows, our proposed method achieves the highest score in F-measure, PSNR, and NRM and its MPM is only slightly lower than Su's method. This means that our proposed method produces a higher overall precision and preserves the text strokes better. In addition, our proposed method also outperforms the 43 document thresholding algorithms submitted to the DIBCO 2009 [19]. Figures 9, 10, 11, and 12 further compare the binarization results of the four example document images in Fig. 1 by using the six document binarization methods. As the four figures show, our proposed method extracts the text properly from the four document images that suffer from different types of document degradation. On the other hand, the performance of the other five methods is generally more or less lower compared with the proposed method.

Table 1 Experimental results of Otsu's, Niblack's, Sauvola's, Gatos's, Su's methods and our proposed method submitted to the DIBCO 2009

Methods	F-measure (%)	PSNR	NRM ($\times 10^{-2}$)	MPM ($\times 10^{-3}$)
Otsu's	78.72	15.34	5.77	13.3
Niblack's	55.82	9.89	16.40	61.5
Sauvola's	85.41	16.39	6.94	3.2
Gatos's	85.25	16.50	10	0.7
Su's	91.06	18.50	7	0.3
Our proposed method	91.24	18.66	4.31	0.55

In addition, experiments over DIBCO's test dataset show that the average execution time of the proposed method is 24 s (implemented in Matlab). In particular, most computation of the proposed method is spent on the document background estimation that involves an iterative polynomial smoothing procedure. The ensuing text stroke edge detection is computational light because it just evaluates the L1-norm image gradient within a 3×3 local neighborhood window. The thresholding from the detected stroke edge pixels is computational light as well because it only evaluates the image mean at the text region that has a certain amount of text stroke edge pixels around. As a comparison, the proposed technique is much slower than Otsu's global thresholding method. However, it is comparable to Niblack's, Sauvola's, and Su's methods and much faster than than Gatos's method.

3.3 Discussion

As described in Sect. 2, the proposed technique involves a number of parameters. In particular, the document background estimation in Sect. 2.2 makes use of several parameters including the sampling step k_s, the initial polynomial order d_0, and the order increase step k_t. Generally, the estimated document background surface has little variation when d_0 is set between 4 and 6 and k_t is set between 0.1 and 0.2. The sampling step k_s has slight effects on the document thresholding when it lies between 1 and 6. In most cases, a slight better performance can be achieved when k_s is set at a small number such as 1 and 2 with the sacrifice of a high computation cost. In addition, the local threshold estimation in Sect. 2.4 makes use of several parameters as well including the thresholding window size and the number of the edge pixels $N_{\min}$ within the thresholding window. In our implemented system, we set the thresholding window size and $N_{\min}$ at 2 and 1 times of the estimated stroke width, respectively, based on our empirical observations and the double-edge structure of the text strokes as described in Sect. 2.4.

The superior performance of the proposed method can be explained by several factors. First, the proposed method

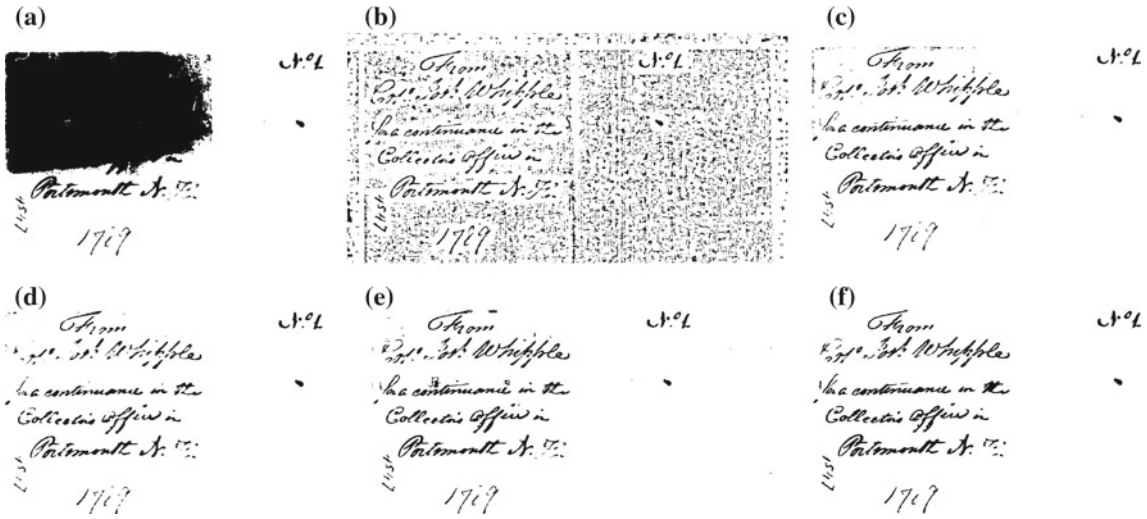

Fig. 10 Binarization results of the sample document image in Fig. 1b by using Otsu's method in (**a**), Niblack's method in (**b**), Sauvola's method in (**c**), Gatos's method in (**d**), Su's method in (**e**), and our proposed method in (**f**)

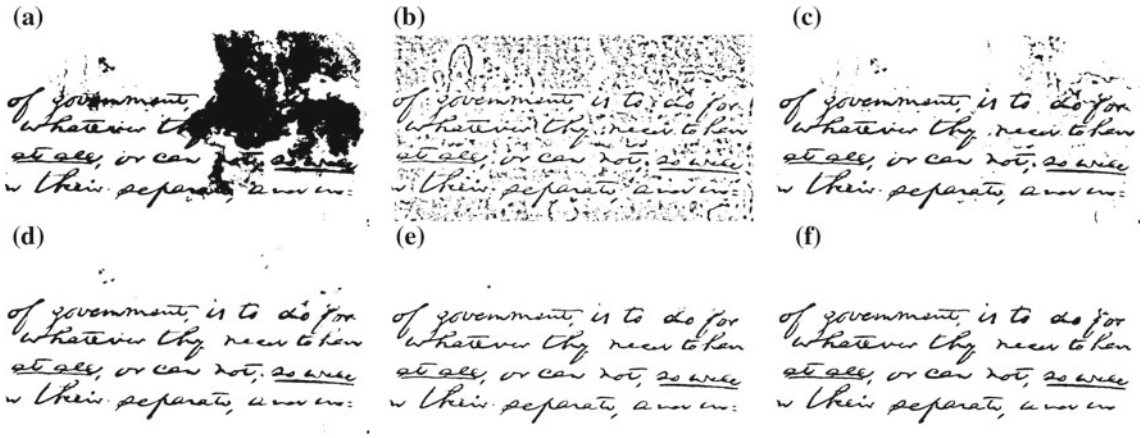

Fig. 11 Binarization results of the sample document image in Fig. 1c by using Otsu's method in (**a**), Niblack's method in (**b**), Sauvola's method in (**c**), Gatos's method in (**d**), Su's method in (**e**), and our proposed method in (**f**)

makes use of a document background surface that helps to compensate the variation of the document background properly. In particular, the compensation greatly improves the detection of the text stroke edge pixels. In addition, the estimated document background surface also helps to remove those falsely detected non-text components in the post-processing stage. As a comparison, global thresholding such as Otsu's method [3] requires a bimodal histogram pattern and so cannot handle the document images with severe background variation as illustrated in Figs. 9, 10, 11, and 12a. Adaptive thresholding such as Niblack's and Sauvola's [11,12] methods may either introduce a certain amount of noise or fail to detect the document text with a low image contrast shown in Figs. 9, 10, 11, and 12b, c. At the same time, the document background surface estimated through polynomial smoothing is also much smoother compared with the ones in [13,14] and so more suitable for the document degradation compensation.

Second, the proposed method estimates the local threshold based on the detected stroke edge pixels. The use of the text stroke edges improves the document thresholding as the document text usually has a sharp and different intensity level compared with the surrounding document background. Therefore, those document regions without text stroke edges will not be classified during the document thresholding process. As a comparison, many reported methods [3,11,12] often improperly classify a certain amount of text pixels from the document background as illustrated in Figs. 9, 10, 11, and 12. In addition, the proposed method detects the text stroke edges from the pixels with the maximum L1-norm image gradient. Our empirical experiments show that the L1-norm image gradient usually outperforms the traditional edge detector and edge profile [16,15] in text stroke edge detection.

Third, the superior performance of our proposed method is also partially due to the three post-processing operations

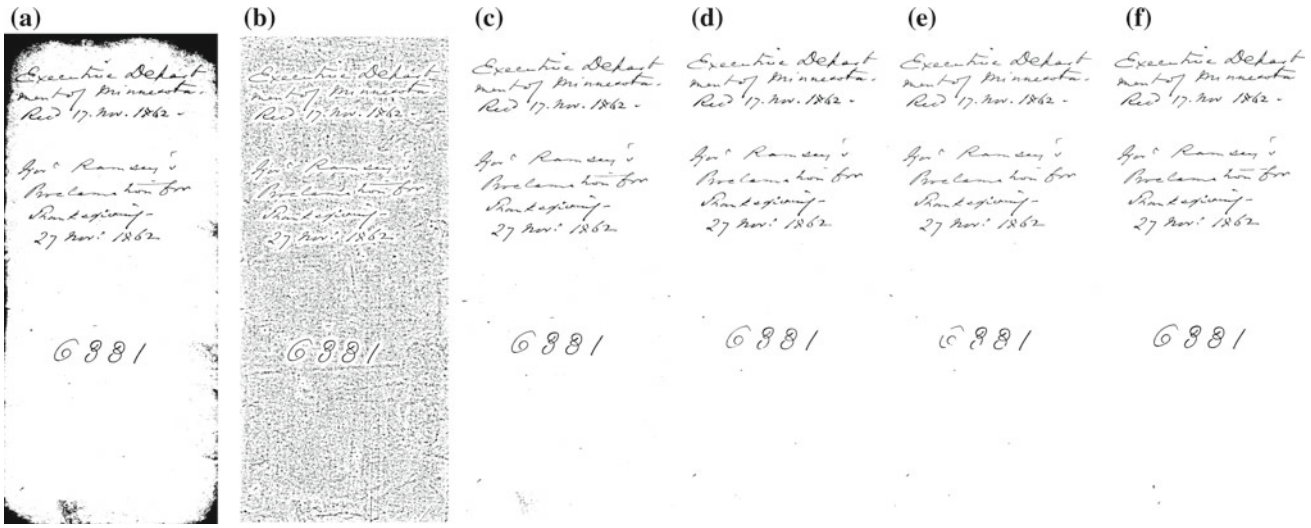

Fig. 12 Binarization results of the sample document image in Fig. 1d by using Otsu's method in (**a**), Niblack's method in (**b**), Sauvola's method in (**c**), Gatos's method in (**d**), Su's method in (**e**), and our proposed method in (**f**)

as described in Sect. 2.5. With an estimated document background surface, most bright non-text components that are falsely detected from the document background can be conveniently identified based on the image difference between the document image at each labeled text component and the corresponding background surface patch. At the same time, document thresholding often introduces a certain amount of single-pixel artifacts such as concavities, convexities, and holes along the text stroke boundary as illustrated in Fig. 8. The correction of such single-pixel artifacts more or less improves the document thresholding performance in most case.

On the other hand, the proposed document thresholding method still has several limitations. First, the proposed method can deal with the document bleeding-through shown in Fig. 1a when the back-side text is fairly brighter compared with the front-side text. But when the back-side text is as dark as or even darker than the front-side text, the proposed method cannot differentiate the two types of character strokes properly. Second, the proposed technique is designed for the binarization of scanned document images that have no or weak slanting. But for the document text captured by digital cameras that may have severe slanting, the performance of the proposed document binarization method may degrade a bit due to higher text stroke width variation resulting from the severe document slanting. Third, the polynomial smoothing is most suitable for the estimation and compensation of the smooth variation within the document background such as shading and smear of large size shown in Fig. 1b, c. But it cannot handle the sharp variation of small size within the document background such as the one resulting from the document folding. We will study these three issues in our future works.

4 Conclusion

This paper presents a document binarization technique that makes use of the document background surface and the text stroke edge information. In the proposed technique, an iterative polynomial smoothing procedure is first implemented to estimate a document background surface efficiently. The stroke edges are then detected based on the local image variation within the compensated document image by using estimated document background surface. Finally, the local threshold is estimated based on the detected stroke edge pixels within a local neighborhood window. The proposed method has been tested and compared with a number of reported document thresholding methods. Experiments show its superior performance which complies with the results of the recent DIBCO contest.

References

1. Trier, O., Taxt, T.: Evaluation of binarization methods for document images. IEEE Trans. Pattern Anal. Mach. Intell **17**, 312–315 (1995)
2. Leedham, G., Yan, C., Takru, K., Tan, J.H.N., Mian, L.: Comparison of some thresholding algorithms for text/background segmentation in difficult document images. Int Conf Doc Anal. Recogn. **2**, 859–864 (2003)
3. Otsu, N.: A threshold selection method from gray level histogram. IEEE Trans. Syst. Man Cybern. **19**, 62–66 (1978)
4. Brink, A.: Thresholding of digital images using two-dimensional entropies. Pattern Recogn. **25**(8), 803–808 (1992)
5. Kittler, J., Illingworth, J.: On threshold selection using clustering criteria. IEEE Trans. Syst. Man Cybern. **15**, 652–655 (1985)
6. Solihin, Y., Leedham, C.: Integral ratio: a new class of global thresholding techniques for handwriting images. IEEE Trans. Pattern Anal. Mach. Intell. **21**, 761–768 (1999)

7. Kim, I.-K., Jung, D.-W., Park, R.-H.: Document image binarization based on topographic analysis using a water flow model. Pattern Recogn. **35**, 141–150 (2002)
8. Yang, J., Chen, Y., Hsu, W.: Adaptive thresholding algorithm and its hardware implementation. Pattern Recogn. Lett. **15**(2), 141–150 (1994)
9. Parker, J., Jennings, C., Salkauskas, A.: Thresholding using an illumination model. International Conference on Document Analysis and Recognition, pp. 270–273. September 1993
10. Eikvil, L., Taxt, T., Moen, K.: A fast adaptive method for binarization of document images. International Conference on Document Analysis and Recognition, pp. 435–443, September 1991
11. Niblack, W.: An Introduction to Digital Image Processing. Prentice-Hall, Englewood Cliffs (1986)
12. Sauvola, J., Pietikainen, M.: Adaptive document image binarization. Pattern Recogn. **33**, 225–236 (2000)
13. Gatos, B., Pratikakis, I., Perantonis, S.: Adaptive degraded document image binarization. Pattern Recogn. **39**, 317–327 (2006)
14. Moghaddam, R.F., Cheriet, M.: Rsldi: restoration of single-sided low-quality document images. Pattern Recogn. **42**, 3355–3364 (2009)
15. Moghaddam, R.F., Cheriet, M.: Application of multi-level classifiers and clustering for automatic word-spotting in historical document images. International Conference on Document Analysis and Recognition, pp. 511–515. July 2009
16. Chen, Q., Sun, Q., Heng, P.A., Xia, D.: A double-threshold image binarization method based on edge detector. Pattern Recogn. **41**, 1254–1267 (2008)
17. Su, B., Lu, S., Tan, C.L.: Binarization of historical handwritten document images using local maximum and minimum filter. International Workshop on Document Analysis Systems, pp. 159–165. June 2010
18. Dawoud, A.: Iterative cross section sequence graph for handwritten character segmentation. IEEE Trans. Image Process. **16**, 2150–2154 (2007)
19. Gatos, B., Ntirogiannis, K., Pratikakis, I.: Icdar 2009 document image binarization contest (dibco 2009). International Conference on Document Analysis and Recognition, pp. 1375–1382. July 2009
20. Lu, S., Tan, C.L.: Binarization of badly illuminated document images through shading estimation and compensation. Int. Conf. Doc. Anal. Recogn. **1**, 312–316 (2007)
21. Hamming, R.W.: Digital Filter. Prentice-Hall, Englewood Cliffs (1983)
22. Krzysztof, M.P.M., Axel, M.: Dynamic threshold using polynomial surface regression with application to the binarization of fingerprints. Proceedings of the SPIE, vol. 5779
23. Seeger, M., Dance, C.: Binarising camera images for ocr. Proceedings of International Conference on Document Analysis and Recognition, pp. 54–58 (2001)
24. Lu, S., Tan, C.L.: Thresholding of badly illuminated document images through photometric correction. ACM symposium on Document engineering pp. 3–8. 2007