Pattern Recognition Letters 33 (2012) 24–32

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

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

Shape based local thresholding for binarization of document images

Jichuan Shi, Nilanjan Ray *, Hong Zhang

Department of Computing Science, University of Alberta, Edmonton, Canada

ARTICLE INFO

Article history:
Received 14 September 2010
Available online 2 October 2011
Communicated by G. Borgefors

Keywords:
Document binarization
Image threshold
Shape analysis

ABSTRACT

This paper presents a novel local threshold algorithm for the binarization of document images. Stroke width of handwritten and printed characters in documents is utilized as the shape feature. As a result, in addition to the intensity analysis, the proposed algorithm introduces the stroke width as shape information into local thresholding. Experimental results for both synthetic and practical document images show that the proposed local threshold algorithm is superior in terms of segmentation quality to the threshold approaches that solely use intensity information.

1. Introduction

Documents are a ubiquitous medium in our daily life. Importing documents into a computer calls for a mechanism of converting handwritten and printed characters into an electronic form. Characters are often captured optically, such as using scanners or cameras, to create images. Successfully extracting characters from an image background is a necessary first step before further analysis. This process is known as document binarization.

Image thresholding is a dominant approach for document binarization. Compared with the global thresholding algorithms (Otsu, 1979; Liu and Srihari, 1997), local ones are superior in terms of selecting threshold values according to local intensity variation. Specifically in the document binarization field, this local intensity variation appears quite often and results from various factors such as uneven illumination, stains, and texture of paper. Local thresholding algorithms, which aim to deal with the difficulty caused by the intensity variation, select threshold values primarily based on local variance methods (Niblack, 1986; Sauvola and Pietikainen, 2000; Rangoni et al., 2009; Yan et al., 2005) and center-surround schemes (Giuliano et al., 1977; Kamel and Zhao, 1993; Ye et al., 2001; Yang and Yan, 2000; Ntirogiannis et al., 2009). With respect to the local variance methods, threshold values are calculated for each pixel from the mean and standard deviation of the intensity in its neighborhood, while for local center-surround schemes, threshold values are locally determined by the contrast between the average intensities of a central window as well as several others surrounding the central window.

Kim et al. (2002) propose a tailor-made watershed algorithm for character extraction from document images. Badekas and Papamarkos (2003) propose a system for document binarization that combines several independent methods in a fuzzy clustering framework. Gatos et al. (2006) propose an adaptive, user-parameter free document binarization method with an image processing pipeline that consists of a low-pass filter, foreground estimation, background surface computations and their meaningful combinations. In a recent work, Pai et al. (2010) apply image block-based document binarization method. They extract blocks of variable sizes by gray scale histogram analysis. Next, a global threshold is applied to each block. Zhu et al. (2006) propose an approach based on stroke neighborhood enhancement with various cues – gradient, foreground–background pixel distance, etc. In a more recent development, given a document image Lu et al. (2010) first estimate a document background surface through an iterative polynomial smoothing procedure. Their technique ranked first in the Document Image Binarization Contest (DIBCO) held under the framework of ICDAR 2009 (www.iit.demokritos.gr/~bgat/DIBCO2009/).

For document binarization, if we observe the process of writing, strokes are trajectories of a pen tip. Thus, the stroke width of handwritten characters should be consistent. This observation can also be extended to printed characters as long as they share similar font size and style. Due to the consistency of stroke width, we would like to exploit it as shape information in addition to intensity information that has been utilized, in determining local threshold for document binarization.

In previous investigations researchers have utilized shape information in document binarization. For example, in Logical Level Threshold (LLT) method (Kamel and Zhao, 1993) stroke width is used as a pre-defined parameter. In adaptive Logical Level Threshold (ALLT) method (Yang and Yan, 2000), stroke width is determined automatically before the binarization process and used

* Corresponding author. Address: Department of Computing Science, 2-21 Athabasca Hall, University of Alberta, Edmonton, Alberta, Canada T6G 2E8. Tel.: +1 780 492 3010; fax: +1 780 492 1071.
E-mail address: nray1@ualberta.ca (N. Ray).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.09.014

subsequently to determine the size of the local windows. However, the process of local threshold selection still depends on the intensity analysis. We are interested in whether shape information on stroke width can further improve the binarization quality.

Based on a similar idea, Liu and Srihari (1997) present an algorithm that calculates a global threshold value with shape information. The method measures the stroke width by run-length histogram. A run in an image is a group of connected pixels containing the same grey-scale value. The number of pixels in the run is defined as the run length, and a run length histogram reflects the variation of objects in terms of shape and texture (Rahman et al., 1999). In (Liu and Srihari, 1997), run length of a binary image is used to select the threshold. Run lengths are computed along x and y directions separately and, as a result, their method is not rotation invariant. Also, their method chooses a single global threshold. Unfortunately, in a lot of cases one global threshold cannot segment all the characters from background with varying intensity. In this paper, we use local threshold selection based on stroke/character width, which is rotation invariant.

In our proposed method for local threshold selection, we capture stroke width information by distance transform (Borgefors, 1986), rather than run length histogram. Our algorithm consists of two stages: training and testing. In the training stage, a training image, which is a small image patch containing typical and clear characters, is binarized by a threshold algorithm, such as Otsu threshold (Otsu, 1979). For this binarized image we compute distance transform. The distance transform essentially generates information about the stroke width of the handwritten or printed characters. A training (normalized) histogram is formed by the distance values. In the testing stage, an image to be binarized is divided into several local regions adaptively. For each local region, a set of threshold candidates is produced by searching for the dominant valleys in the intensity histogram, as is motivated by Liu and Srihari's work (Liu and Srihari, 1997). For each prospective threshold value, a normalized histogram is formed with the distance transform. The similarity between this distance histogram and the training histogram is computed. This way a similarity score is computed for every candidate threshold value. For each local window, we choose a threshold value that maximizes the similarity score. Thus, every local region is assigned one optimal threshold value. Next, a threshold surface adapting to the intensity variation is generated by the Thin Plate Splines (TPS) algorithm (Bookstein, 1989) with these selected local threshold values as supporting points.

To select the local regions adaptively, we resort to binary partitioning of the input image recursively and greedily. An image is partitioned into two equal halves (in the direction of the longer of the two dimensions) provided at least one of the halves provides a better shape histogram matching score than its parent partition. A child partition with a better matching score is further subdivided. Advantage of this greedy and recursive binary partition is manifold. First, often it takes only a few partitions to quickly binarize a large document image. Second, our entire binarization method hardly has any user tunable parameter. The only parameter is the smallest allowed size of a partition, beyond which recursion stops forcefully.

Our method can be viewed as an user interactive document binarization scheme, where the user selects a tiny portion of the document to extract the stroke width distribution of handwritten/printed characters. We have noticed that as small as a single word or a few characters can serve well in terms of user interaction. Our experiments demonstrate that with respect to both handwritten and printed character images, which contain noise resulting from uneven illumination or background texture, the proposed local threshold algorithm is superior to the other local threshold methods in terms of segmentation quality.

The rest of this paper is organized as follows. Section 2 describes the general framework of the proposed local thresholding algorithm, and then illustrates the process of the shape information extraction, the local threshold estimation, and the threshold surface generation in detail. Section 3 presents the experimental results for both synthetic and real document images and compares the proposed algorithm with other competing local thresholding methods. Conclusions are drawn in Section 4.

2. Local thresholding based on shape information

The proposed local thresholding algorithm is separated into two parts: an offline or user interactive training stage and an online testing stage. In the training stage the shape information (distance histogram) is learned. In the testing stage, for an input grey-scale image, the shape information is utilized to select thresholds in adaptively computed local partitions of the image.

2.1. Shape information extraction

Since the stroke width distribution of characters is the only information required for training, we crop a small part of a document with a few characters, and choose the optimal threshold value by Otsu method to produce a training binary image. Next, distance transform (Borgefors, 1986) is computed on the binarized training image and a normalized training histogram $h_{training}$ of the distance values is created. A simple algorithmic description below defines this process.

2.1.1. Training histogram

(**S1**) User selects a small portion of the input document image.
(**S2**) Binarize the selected small image by Otsu threshold.
(**S3**) Compute shape/distance histogram $h_{training}$ for the binarized image patch.

We assume that the intensity in the background is 1 (“on”) while character/stroke is 0 (“off”). For an “off” pixel, the distance transform value is the distance to its nearest “on” pixel. Thus, a histogram of the non-zero distance transform values characterizes the width of the “off” connected components – a description suitable for pen stroke/character width. Theoretically, one can choose any distance metric for this distance transform. We choose Chebyshev distance because it produces integer distances. One can also choose rotation invariant Euclidean distance, for which the computation is slightly more expensive.

From a training image (Fig. 1(a)), the distance transform is applied to generate a distance map as illustrated in Fig. 1(b). In the distance map, a darker pixel value represents a higher distance value, or equivalently, a wider stroke. Although the stroke width is consistent, it may still vary in a small range. We use the histogram of *non-zero* distance values (Fig. 1(c)).

2.2. Local threshold estimation

The test grey-scale image is recursively partitioned into rectangular local regions. For each local region, a set of gray-scale threshold candidates is produced by a simple analysis of the image intensity histogram. Since a homogeneous region in an image often forms a peak in the intensity histogram, valleys in the histogram tend to indicate threshold values separating homogeneous regions. Thus, we choose the dominant valleys as threshold candidates (Liu and Srihari, 1997). To avoid non-dominant valleys resulting from noise, a smoothing process is applied in the intensity histogram analysis. Parzen window-based probability density estimation

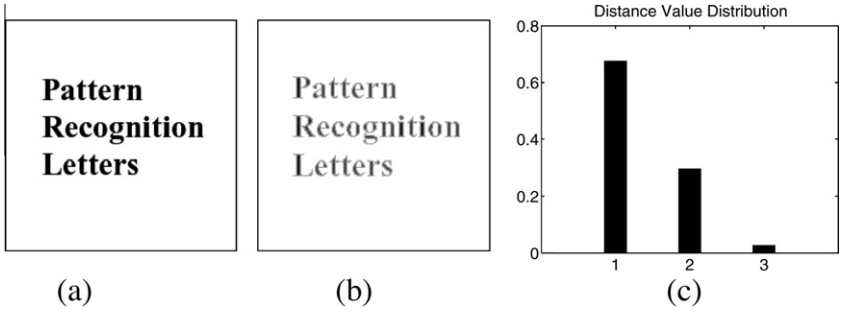

Fig. 1. Distance transform: (a) binary training image, (b) distance map of training image, (c) distance value distribution $h_{training}$.

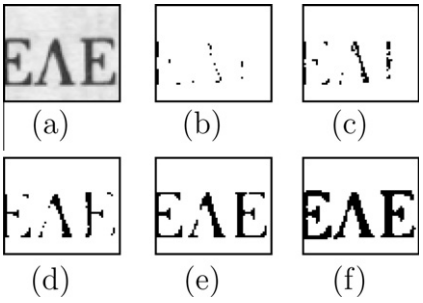

Fig. 2. Segmentation generated by threshold candidates: (a) input grey-scale image, (b) threshold 58, BC value 0.8303, (c) threshold 65, BC value 0.9203, (d) threshold 78, BC value 0.9707, (e) threshold 106, BC value 0.9830, (f) threshold 161, BC value 0.9989.

method (Silverman, 1986) is adopted. This method automatically calculates the size of the smoothing window and generates a continuous distribution from discrete histogram data. The probability density function is estimated by Silverman (1986):

$$H(x) = \frac{1}{nw}\sum_{i=1}^{n} K\left(\frac{x - X_i}{w}\right), \tag{1}$$

where X_i in our case is a pixel intensity value in a rectangular local image region. n is the total number of pixels in the local region, and K is the Gaussian kernel. The size of the smoothing window $w = w_{opt}$ is calculated from (2), in which σ is the standard deviation and IQR is the interquartile range of the image intensity values in the local region (Silverman, 1986):

$$\begin{aligned} w_{opt} &= 0.9A * n^{-1/5} \\ A &= min(\sigma, IQR/1.34). \end{aligned} \tag{2}$$

Once the intensity histogram is estimated, its valleys (minima) are found and considered as threshold candidates. We denote the set of threshold candidates by T. For each candidate threshold $t \in T$, we obtain a binary image patch in the local region, and a histogram $h(t)$ of non-zero distance values is computed.

We use Bhattacharyya Coefficient (BC) (Bhattacharyya, 1943) to measure the similarity between the training histogram $h_{training}$ and test histograms $h(t)$. All histograms are normalized. BC is widely adopted to evaluate the degree of closeness of two probability density functions. Among all the threshold candidates derived from one local region, the threshold value resulting in the maximum BC value is selected for the local region, i.e.,

$$t^* = \arg\max_{t \in T}[BC(h(t), h_{training})]. \tag{3}$$

As an example, in Fig. 2, the threshold 161 is selected as the optimal value for this specific local region because it leads to the maximum BC value among the five threshold candidates. The following algorithm summarizes this process.

$(\boldsymbol{t}^*, \boldsymbol{s}^*) \leftarrow \boldsymbol{BEST\_THRESHOLD}(\boldsymbol{I})$.

(**S1**) Generate candidate threshold values for input image I.
(**S2**) For each threshold value t, binarize I, obtain shape histogram $h(t)$ for the binary image, and compute Bhattacharya coefficient between $h(t)$ and $h_{training}$.
(**S3**) Select best threshold value t^* and corresponding best BC value s^*.

As already mentioned, our algorithm adaptively partitions the image into rectangular regions. The partitioning and threshold value selection for each partition occur in a recursive algorithm as follows.

2.2.1. Recursive subdivision of input image I

(**S1**) $(t,s) \leftarrow BEST\_THRESHOLD(I)$.
(**S2**) If height of I is greater than its width, rotate I by 90°.
(**S3**) Divide I into left and right halves; call them I_l and I_r.
(**S4**) $(t_l, s_l) \leftarrow BEST\_THRESHOLD(I_l)$.
(**S5**) $(t_r, s_r) \leftarrow BEST\_THRESHOLD(I_r)$.
(**S6**) If $s \leqslant s_l$ and dimensions of I_l are larger than w_{min}, then recursively subdivide I_l.
(**S7**) If $s \leqslant s_r$ and dimensions of I_r are larger than w_{min}, then recursively subdivide I_r.

Note that the subdivision algorithm retains the best threshold values selected for each leaf node (a rectangular local region) in the binary tree. We have omitted these details to keep the above algorithmic description simple. w_{min} is the only user input parameter in our algorithm that prevents further subdivision of a local region once the height or width of the region goes below w_{min}. In all our experiments, we have chosen $w_{min} = 16$. Fig. 3(a) shows a handwritten document image (taken from document binarization competition web site: www.iit.demokritos.gr/~bgat/DIBCO2009/). It also shows an overlaid rectangle, which is the user chosen area to obtain training histogram. Fig. 3(b) shows user selected area binarized by Otsu threshold. Fig. 3(c) shows $h_{training}$. Fig. 3(d) shows rectangular partitions generated by our proposed recursive subdivision algorithm. Fig. 4(a) through (d) show the exact same process for another document image obtained from www.iit.demokritos.gr/~bgat/DIBCO2009/.

2.3. Threshold surface generation

Once we obtain a set of optimum local thresholds by (3) for all the local regions, we could directly apply these threshold values in their respective local regions to generate the binary document. However, this approach would lead to discontinuities at the

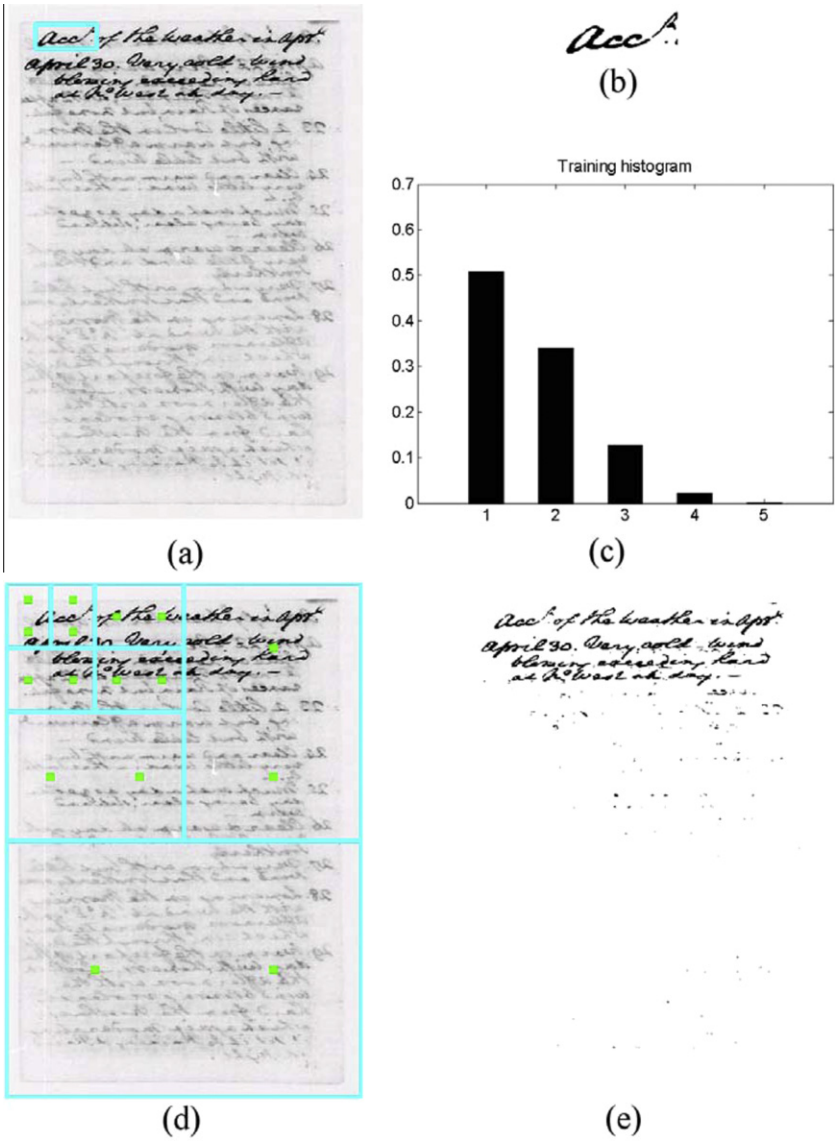

Fig. 3. Recursive subdivision on a handwritten character image.

boundaries where local regions meet each other. We take the route of an interpolating surface to overcome this problem. To generate the final threshold surface, Thin Plate Splines (TPS) interpolation is utilized. TPS is a 2D interpolation scheme for arbitrarily placed supporting points in a plane. It produces a smooth surface passing through these supporting points by minimizing an energy function (Bookstein, 1989). Here the primary advantage of TPS is that it can generate a threshold surface without any constraint in terms of the quantity or position of supporting points in the plane.

A TPS surface is given by the following equation (Bookstein, 1989):

$$f(x,y) = \sum_{j=1}^{n} a_j E(\|(x - x_j, y - y_j)\|) + b_0 + b_1 x + b_2 y, \quad (4)$$

where (x_j, y_j) are the supporting points and E is defined by $E(r) = r^2 \log(r^2)$. The TPS surface (4) must pass through its supporting points (x_i, y_i, z_i), where (x_i, y_i) are the coordinates of the center of the ith local window and z_i is the optimum threshold value there. The coefficients a_j and b's are found by solving a set of linear equations (Bookstein, 1989). In our approach this TPS surface (4) serves as the final threshold surface for the test image.

In Fig. 3(d) we show two points (square shaped) in each local region (leaf node of the binary partitioning tree) that are support points for TPS. Notice that for both points belonging to a partition, the z_i value is the best selected threshold value for that partition. In our experiments, we have taken two points of equal value from each local region as opposed to a single point, purely based on our observation that the former leads to better binarization results. Once the TPS surface is obtained we binarize the input image with this TPS. The final binarized images are shown in Figs. 3(e) and 4(e).

To illustrate how the TPS surface adapts with intensity variation, one row of pixels are extracted from the synthetic document image (Fig. 6) and plotted in Fig. 5. It gives a one dimensional illustration of intensity surface, threshold surface (TPS), and threshold surface formed by the optimal threshold in each local region. From this figure we can see that the TPS surface adapts to the intensity variation while it follows the values of the local thresholds selected with shape information.

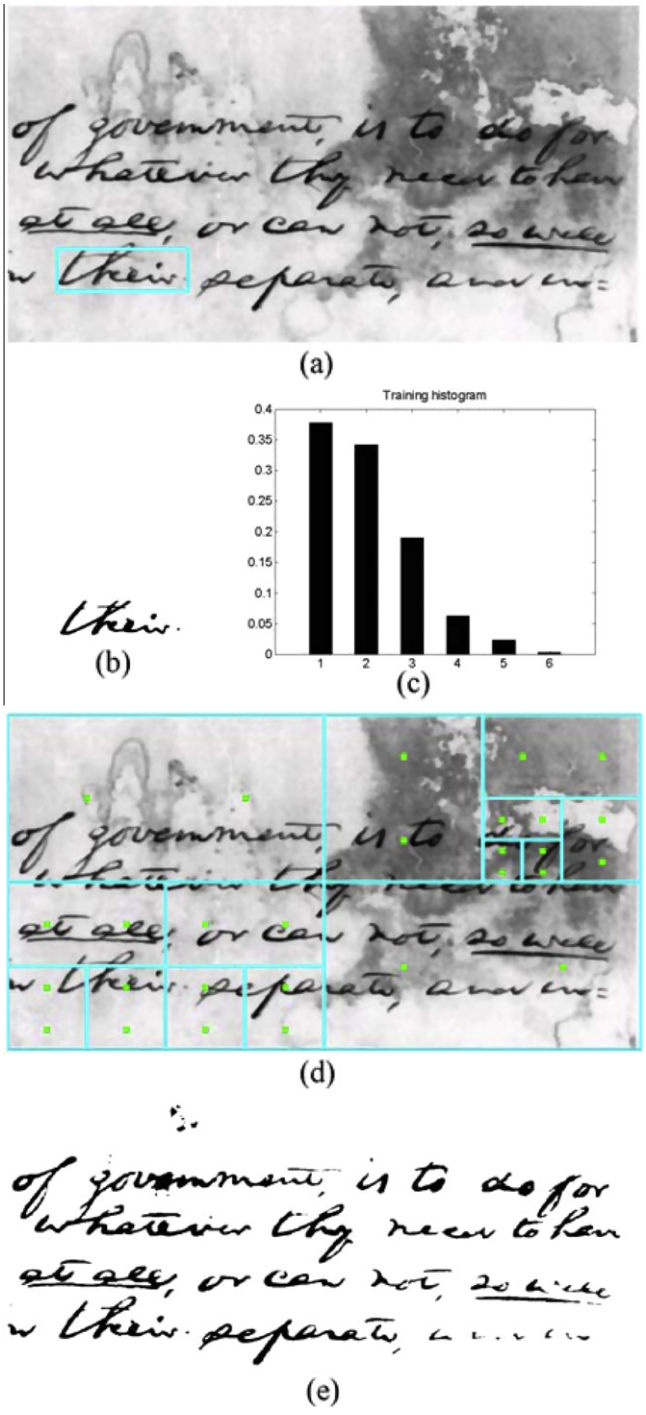

Fig. 4. Recursive subdivision on another handwritten image.

3. Experimental results

The proposed algorithm is compared with several local thresholding algorithms including Niblack's method (Niblack, 1986) and Multistage Adaptive Thresholding (MAT) (Yan et al., 2005). The global thresholding method utilizing the run-length histogram (Liu and Srihari, 1997) and the Otsu method (Otsu, 1979) are also included for comparison to show that one global threshold value is not suitable for document images with intensity variation. Both the proposed algorithm and competing algorithms are applied to synthetic and real document images. An evaluation metric designed by Badekas and Papamarkos (2005, 2008) is used to compare these thresholding methods.

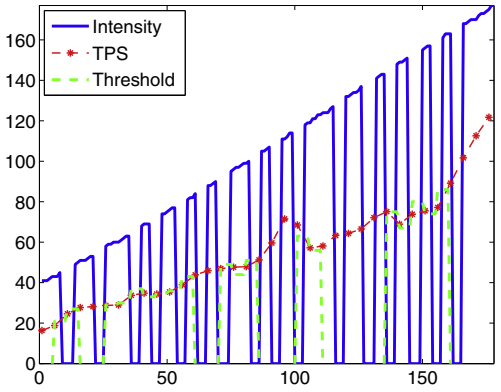

Fig. 5. One dimensional illustration thresholding surfaces for a grey-scale image (a line across the word "transaction" in a synthetic document image of Fig. 6.)

For a test synthetic image shown in Fig. 6(b), a binary image (Fig. 6(a)) is utilized as the training sample. Both images have the same size of 178 × 178 pixels. The characters in the training and test images are different from each other but share the same font size and style. The proposed algorithm learns the shape information from the training image and incorporates it in threshold selection. To show the superiority of the proposed algorithm, an unevenly illuminated test image (Fig. 6(b)) is considered. We found that Niblack's method is sensitive to noise, especially in regions containing no characters. The MAT algorithm produces a better segmentation than Niblack's. The result of MAT still contains, however, many unwanted fragments. Global thresholding with run-length histogram and Otsu produce broken and partially missing characters. Compared with these thresholding algorithms, the proposed local threshold algorithm is clearly superior.

The binarization results derived from a real document image are presented in Fig. 7. A local region (100 × 120 pixels) of the document image (400 × 400 pixels) is binarized by the Otsu algorithm to produce a training sample, before the proposed algorithm is applied to the whole grey-scale image. As shown in Fig. 7(b), the grey-scale image includes a light background texture in the background that is common in some scanned images. From the segmented results we can see that Niblack's method and the run-length histogram produced poor results due to the intensity variation. The MAT method, the Otsu method, and the proposed local threshold algorithm successfully separate the characters from a complex background and produce a clean segmentation.

To evaluate the binarization quality quantitatively, a criterion by Badekas and Papamarkos (2005, 2008) is adopted. This criterion compares a binarized image with the ground truth at the pixel level where the segmentation process can be considered as a binary classification. A pixel is labeled as either object or background. According to Badekas and Papamarkos (2005), the Chi-square test indicates the segmentation quality by balancing precision and recall.

To obtain the Chi-square value, a ground truth image is required as the reference. Pixels are classified into four categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). For a pixel, if it is labeled as object in both ground truth and segmented images, it will be considered as TP, whereas if it is classified as object in the segmented image but categorized as background in the ground truth, it will be considered as FP. TN and FN are defined similarly. The Chi-square value is calculated as

$$\chi^2 = \frac{(TPR - Q)\cdot((1 - FPR) - (1 - Q))}{(1 - Q)Q}, \tag{5}$$

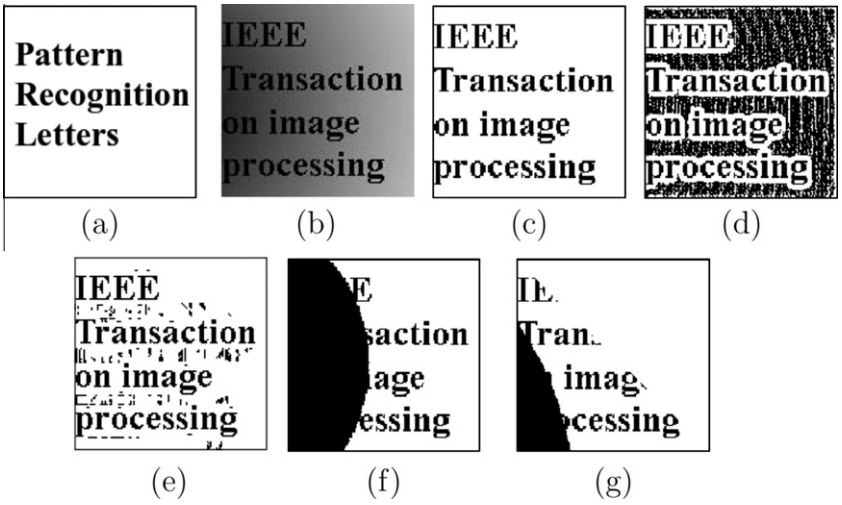

Fig. 6. Thresholding results for a synthetic document image: (a) training image (b) grey-scale image (c) proposed local thresholding based on shape information (d) Niblack (e) MAT (f) Otsu (g) runlength histogram.

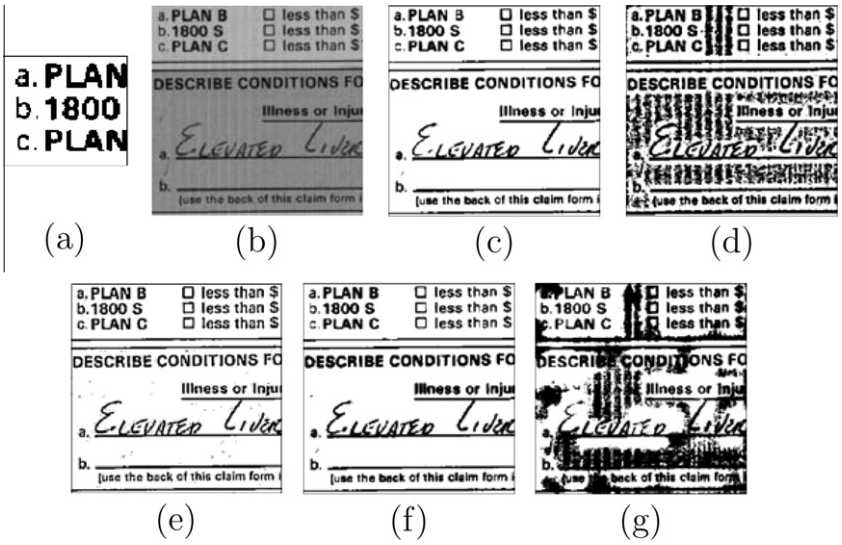

Fig. 7. Thresholding results of a practical document image: (a) training image (b) grey-scale image (c) proposed local threshold based on shape information (d) Niblack (e) MAT (f) Otsu (g) runlength histogram.

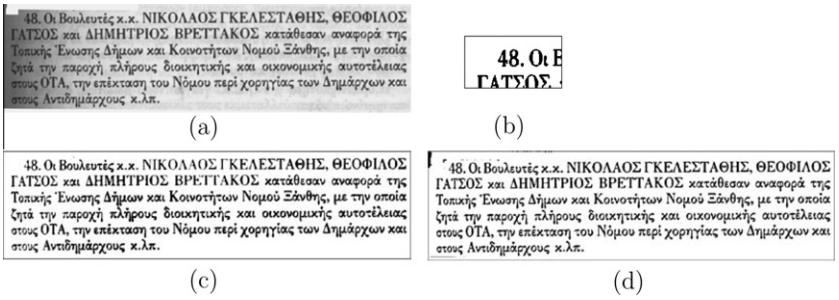

Fig. 8. Segmentation results of Greek letters: (a) grey-scale image (b) training image (c) local thresholding based on shape information (d) ALLT Badekas and Papamarkos, 2005.

where *TPR* and *FPR* are the TP ratio and FP ratio, respectively, and $Q = TPR + FPR$. Thus, this criterion evaluates the segmentation quality with respect to the precision and recall at the pixel level. A higher χ^2 value indicates a better segmentation.

Through the Chi-square test value the proposed algorithm is compared with all the threshold algorithms listed in (Badekas and Papamarkos, 2005) including (Sauvola and Pietikainen, 2000; Bernsen, 1986; Niblack, 1986; Chi et al., 1996; Otsu, 1979; Yang

Table 1
F-measures (in percentages) on DIBCO2009 images.

Images	Proposed	MAT	Niblack	OTSU	Runlength
P01	90.75	80.26	68.24	89.30	34.46
P02	95.39	70.03	94.09	96.21	39.95
P03	90.72	69.76	73.54	51.63	41.84
P04	90.39	68.60	62.69	82.71	39.63
P05	85.47	73.13	71.87	85.41	30.10
H01	90.12	59.68	63.73	90.46	8.58
H02	87.67	27.76	18.50	86.45	65.89
H03	85.60	73.71	71.39	84.52	53.34
H04	78.65	54.07	46.85	41.05	29.62
H05	59.80	30.45	27.21	28.17	35.83

Table 2
PSNR scores on DIBCO2009 images.

Images	Proposed	MAT	Niblack	OTSU	Runlength
P01	16.50	13.20	9.53	15.56	5.71
P02	17.03	9.82	15.91	18.09	4.36
P03	14.73	10.21	9.49	9.52	5.12
P04	16.81	11.04	9.12	13.80	5.15
P05	13.67	10.97	9.68	13.72	3.95
H01	19.00	10.54	11.20	19.12	3.52
H02	22.64	9.55	7.30	22.00	19.24
H03	15.21	12.25	11.12	14.65	11.99
H04	15.77	9.94	7.82	6.82	12.17
H05	16.50	10.55	6.95	7.31	10.70

Table 3
NRM values ($\times 10^{-2}$) for DIBCO2009 images.

Images	Proposed	MAT	Niblack	OTSU	Runlength
P01	5.13	10.98	6.66	3.19	33.16
P02	1.68	21.82	2.27	2.80	38.36
P03	3.48	19.91	10.14	32.58	32.54
P04	3.90	12.28	7.59	4.33	18.91
P05	7.98	15.29	8.77	8.69	40.46
H01	7.24	5.77	4.25	6.80	55.75
H02	4.78	6.41	10.68	3.69	23.48
H03	3.45	9.50	4.55	3.47	31.51
H04	8.88	14.03	9.28	11.86	41.31
H05	12.34	52.60	10.97	11.80	22.53

Table 4
MPM scores ($\times 10^{-3}$) on DIBCO2009 images.

Images	Proposed	MAT	Niblack	OTSU	Runlength
P01	0.77	17.32	145.21	1.65	36.30
P02	1.32	17.37	15.53	0.34	104.66
P03	11.27	33.56	120.71	5.00	221.58
P04	2.92	25.07	118.73	9.08	31.25
P05	4.36	12.09	124.04	4.05	181.60
H01	0.15	24.96	83.12	0.15	319.29
H02	0.30	42.81	110.03	0.54	1.70
H03	0.31	16.95	15.87	2.66	1.04
H04	0.20	29.30	98.91	102.71	0.58
H05	0.18	40.98	135.96	11.92	112.68

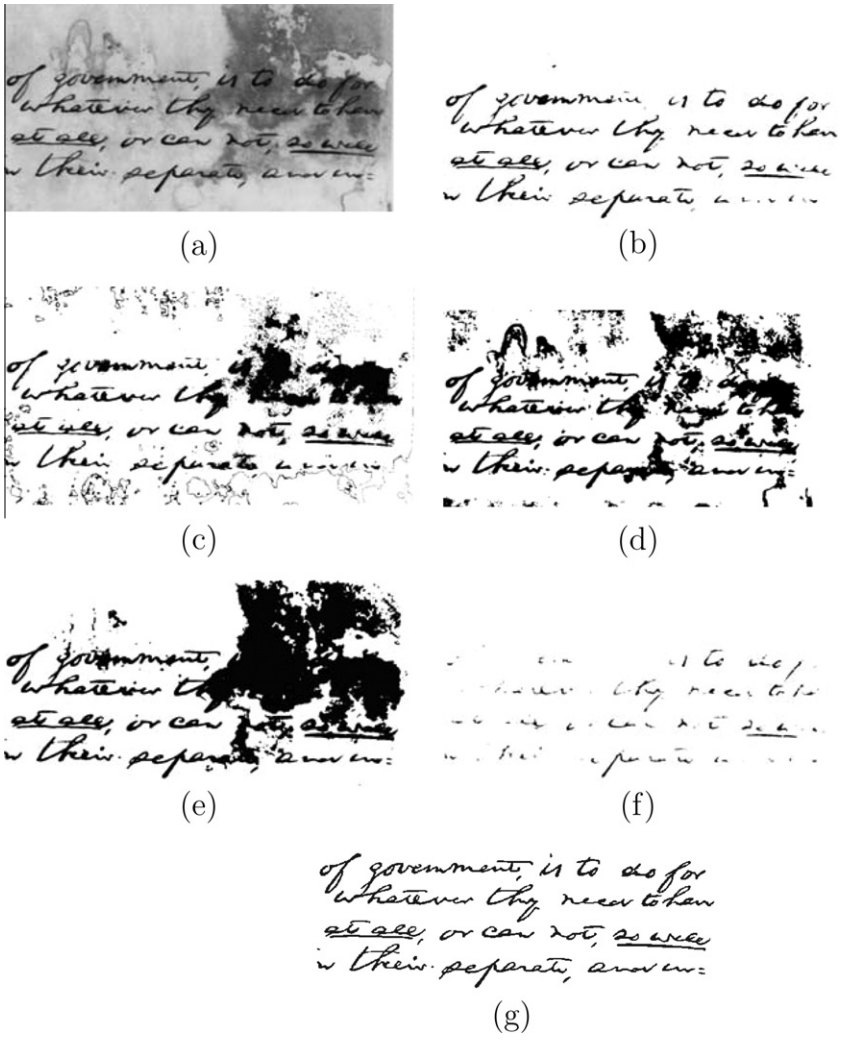

Fig. 9. Segmentation results of handwritten image 4: (a) grey-scale image (b) proposed local thresholding based on shape information (c) MAT (d) Niblack (e) OTSU (f) global runlength (g) ground truth.

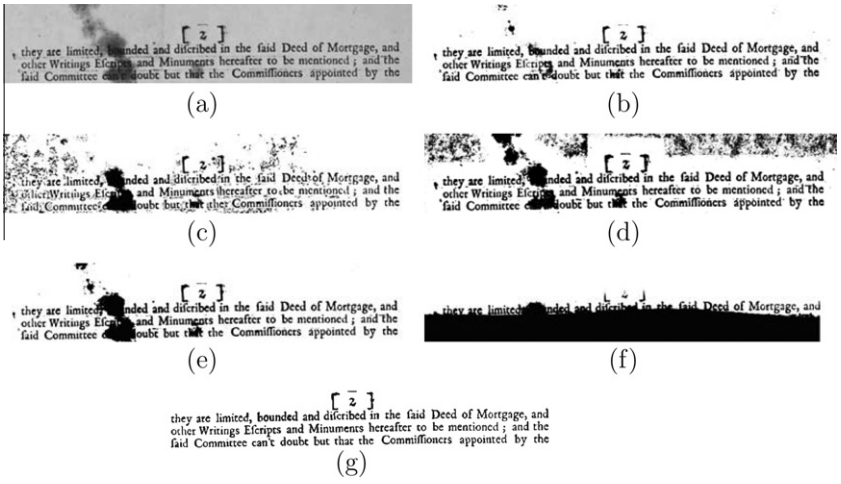

Fig. 10. Segmentation results of printed image 4: (a) grey-scale image (b) proposed local thresholding based on shape information (c) MAT (d) Niblack (e) OTSU (f) global runlength (g) ground truth.

and Yan, 2000; Trier and Taxt, 1995), as well as other two intensity analysis based methods (Yan et al., 2005; Liu and Srihari, 1997). To implement the comparison, we utilize the grey-scale image in (Badekas and Papamarkos, 2005) as a test instance (Fig. 8(a)) and produce a segmentation by the proposed algorithm (Fig. 8(c)). A Chi-square test value is derived from this segmentation. Among algorithms described above, the Chi-square value ranges between 0.972 and 0.609 while our method has the highest Chi-square value at 0.972. ALLT and Sauvola also performed well, with Chi-square value of 0.94.

For visual evaluation the proposed algorithm is compared with the ALLT (Badekas and Papamarkos, 2005). The thresholding results of the proposed algorithm (Fig. 8(c)) and ALLT (Fig. 8(d)) are visually comparable with each other, while the proposed algorithm has a better performance at the upper left corner and lower left corner of the image. In view of all the experimental results described in this section, the proposed local threshold algorithm which incorporates the shape information achieves a superior segmentation over its competitors.

To further establish the effectiveness of our document binarization method, we have used ten test images from www.iit.demokritos.gr/~bgat/DIBCO2009/, with five of them being handwritten documents and the other five being printed documents. DIBCO2009 (Gatos et al., 2009) recommends four evaluation metrics: F-measure, PSNR, negative rate metric (NRM) and misclassification penalty metric (MPM). F-measure is the harmonic mean of recall and precision. A higher F-measure value naturally indicates a better binarization. PNSR is the peak signal to noise ratio. A higher PSNR value indicates that the output image is a better approximation of the ground truth binary image. NRM is a pixel-to-pixel mismatch between the output binary image and the ground truth binary image. NRM is the mean of false positive and false negative rate. A lower NRM value indicates a better binary output. MPM evaluates the output against the ground truth on an object-by-object basis. A lower MPM value indicates better accuracies produced by the algorithm in identifying object boundaries. Tables 1–4 compare these performance measures for several algorithms including the proposed one. The image names starting with "P" and "H" respectively denote printed and handwritten document images from DIBCO2009. These tables demonstrate the superiority of the proposed method within the comparative group. The average performance metrics for the proposed method from these tables also establish that it is competitive with the top ten algorithms at DIBCO2009 competition (Gatos et al., 2009). Figs. 9 and 10 provide binarization results generated by these methods on two images taken from the DIBCO2009 database. These illustrations show that our algorithm produces significantly better results than those in the comparative group.

4. Conclusions

This paper has presented a local thresholding algorithm exploiting stroke width as shape information in improving the binarization of document images. The threshold selection is based on the stroke width consistency and our algorithm determines threshold values locally. Experiments on a variety of document images demonstrate that the proposed local threshold algorithm is superior to the other threshold methods in terms of segmentation quality.

References

Badekas, E., Papamarkos, N., 2003. A system for document binarization. In: Proc. 3rd Internat. Symp. on Image and Signal Processing and Analysis, ISPA 2003, Vol. 2, pp. 909–914.
Badekas, E., Papamarkos, N., 2005. Automatic evaluation of document binarization results. Prog. Pattern Recognition Image Anal. Appl. 3773, 1005–1014.
Badekas, E., Papamarkos, N., 2008. Estimation of proper parameter values for document binarization. In: Internat. Conf. on Computer Graphics and Imaging, track 600–037.
Bernsen, J., 1986. Dynamic thresholding of gray level images. In: ICPR'86: Proc. Internat. Conf. on Pattern Recognition, pp. 1251–1255.
Bhattacharyya, A., 1943. On a measure of divergence between two statistical populations defined by their probability distributions. Bull. Calcutta Math. Soc. 35, 99–109.
Bookstein, F.L., 1989. Principle warps: Thin-plate splines and the decomposition of deformations. IEEE Trans. Pattern Anal. Machine Intell., 567–585.
Borgefors, G., 1986. Distance transformations in digital images. Comput. Vision Graph. Image Process. 34, 344–371.
Chi, Z., Yan, H., Pham, T., 1996. Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition. World Scientific Publishing, River Edge, NJ, USA.
Gatos, B., Pratikakis, I., Perantonis, S., 2006. Adaptive degraded document image binarization. Pattern Recognition 39, 317–327.
Gatos, B., Ntirogiannis, K., Pratikakis, I., 2009. Icdar 2009 document image binarization contest (dibco 2009), In: ICDAR'09, pp. 1375–1382.
Giuliano, E., Paitra, O., Stringer, L., 1977. Electronic character reading system, US Patent.
Kamel, M., Zhao, A., 1993. Extraction of binary character/graphics images from grayscale document images. CVGIP: Graph. Models Image Process. 55, 203–217.
Kim, I.-K., Jung, D.-W., Park, R.-H., 2002. Document image binarization based on topographic analysis using a water flow model. Pattern Recognition 35, 265–277.
Liu, Y., Srihari, S.N., 1997. Document image binarization based on texture features. IEEE Trans. Pattern Anal. Machine Intell. 19, 540–544.
Lu, S., Su, B., Tan, C.L., 2010. Document image binarization using background estimation and stroke edges. Internat. J. Doc. Anal. Recognition 13, 303–314.

Niblack, W., 1986. An Introduction to Digital Image Processing. Prentice Hall, Englewood Cliffd, NJ.
Ntirogiannis, K., Gatos, B., Pratikakis, I., 2009. A modified adaptive logical level binarization technique for historical document images. In: 10th Internat. Conf. on Document Analysis and Recognition, pp. 1171–1175.
Otsu, N., 1979. A threshold selection method from gray-level histogram. IEEE Tans. Systems Man. Cybernet. 9, 62–66.
Pai, Y.-T., Chang, Y.-F., Ruan, S.-J., 2010. Adaptive thresholding algorithm: Efficient computation technique based on intelligent block detection for degraded document images. Pattern Recognition 43, 3177–3187.
Rahman, S.M., Karmaker, G.C., Bignall, R.J., 1999. Improving Image Classification Using Extended Run Length Features, Vol. 1614. Springer, Berlin.
Rangoni, Y., Shafait, F., Breuel, T.M., 2009. Ocr based thresholding. IN: IAPR Conf. on Machine Vision Application, pp. 98–101.
Sauvola, J., Pietikainen, M., 2000. Adaptive document image binarization. Pattern Recognition, 225–236.
Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. CRC Press, London, UK.
Trier, O.D., Taxt, T., 1995. Improvement of 'integrated function algorithm' for binarization of document images. Pattern Recognition Lett., 277–283.
Yan, F., Zhang, H., Kube, C.R., 2005. A multistage adaptive thresholding method. Pattern Recognition Lett. 26, 1183–1191.
Yang, Y., Yan, H., 2000. An adaptive logical method for binarization of degraded document images. Pattern Recognition 33, 787–807.
Ye, X., Cheriet, M., Suen, C.Y., 2001. Stroke-model-based character extraction from gray-level document images. IEEE Trans. Image Process. 10, 1152–1161.
Zhu, Y., Wang, C., Dai, R., 2006. Document image binarization based on stroke enhancement. In: Proc. 18th Internat. Conf. on Pattern Recognition. ICPR '06, Vol. 01. IEEE Computer Society, Washington, DC, USA, pp. 955–958.