Iterative Multimodel Subimage Binarization for Handwritten Character Segmentation

Amer Dawoud and Mohamed S. Kamel

***Abstract*—Existing binarization methods are categorized as either global or local. In this paper, we present a new category, where the image is considered a collection of subimages. Each subimage provides a statistical model for the handwritten characters that can be used to optimize the binarization of other subimages based on gray-level and stroke-run features. The proposed method uses these multimodels to iteratively arrive at the optimal threshold for each subimage. It can be applied to different types of documents where prior knowledge about the noisiness of the subimages is not available. Experimental results showed significant improvement in the binarization quality in comparison with other well-established algorithms.**

***Index Terms*—Document binarization, document processing, handwritten character segmentation, subimage binarization.**

I. Introduction

EXISTING binarization techniques can be categorized as either global or local. Global thresholding algorithms use a single threshold, while local thresholding algorithms compute a separate threshold for each pixel (or group of pixels) based on a neighborhood of the pixel. Many binarization methods were proposed over the last 20 years. Sahoo *et al.* [8] compared the performance of more than 20 global thresholding algorithms using uniformity or shape measures. The comparison showed that Otsu's method [7] gave best performance. Trier and Jain [10] evaluated the performance of 11 well-established local thresholding algorithms. In their evaluation, algorithms of Niblack [6] and Yanowitz and Bruckstein [11] produced high recognition rates. The following algorithms are examples of algorithms that were developed recently. Liu and Srihari's algorithm [5] uses texture features to select an optimal threshold from a set of candidate thresholds. Solihin and Leedham's algorithm [9] selects an optimal threshold using histograms modified by integral ratio techniques. Zhao's *et al.* method [13] uses multiple window size operation to select local thresholds. Cheriet's *et al.* method [2] uses a recursive technique that extends Otsu's approach. Dawoud and Kamel [3] proposed an iterative model-based binarization technique. The method we propose in this paper can be considered as a generalization of the methods reported by Dawoud and Kamel [3] and Liu and Srihari's algorithm [5].

We propose a new category of image binarization where the image is divided into subimages. Let us assume that a document image is divided into subimages: Image (1), Image $(2), \ldots,$ Image (M), where M is the total number of subimages. Fig. 1 shows an example of such a document image that contains five subimages, $M = 5$. We will also assume that the levels of background noise in the subimages are independent and unknown, and that the characters in all subimages have been written by the same pen. The underlying principle of our approach is that since handwritten characters of different subimages come from the same source and are captured under the same conditions, they possess similar characteristics, regardless of their background type.

The objective of our approach is to find an optimal threshold, $T(x)$, $x \in \{1, 2, \ldots, M\}$ for each subimage that would eliminate background noise, while preserving as much handwritten stroke data as possible.

The rest of the paper is organized as follows: Section II presents the multimodel binarization algorithm and how to optimize its criteria for stopping the iterations. Comparative performance evaluation results are presented in Section III. The conclusions of this work are stated in Section IV.

II. Iterative Multimodel Binarization Algorithm

We will use Fig. 2 as an example to demonstrate the application of steps of the algorithm. As a preprocessing step, noncharacter structures, like baselines and frame boxes, are cropped. These structures usually have different gray-level and stroke-run characteristics than those of the handwritten characters; therefore, they should be cropped so that they are not confused with handwritten strokes. This preprocessing step, as shown in Fig. 2, is performed before doing the iterative feature extraction.

The subimages are then binarized at a sequence of candidate thresholds CT_i, where CT_1 is the lowest possible threshold in gray-scale histogram. The difference between two successive CTs was chosen to be eight gray-levels, which we found to be satisfactory. As the iteration number i increases, the handwritten strokes become thicker and more connected, but, at the same time, there will be a higher risk of background noise interference, because the CT_i will also be closer to the background region of gray-scale histogram. Our objective is to find the optimal threshold for each subimage that will extract as much handwritten strokes as possible without allowing the background noise interference.

Manuscript received January 20, 2003; revised December 11, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aly A. Farag.

A. Dawoud is with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada, and also with the Department of Electrical and Computer Engineering, University of South Alabama, Mobile, AL 36688 USA (e-mail: dawoud@watfast.uwaterloo.ca; adawoud@usouthal.edu).

M. S. Kamel is with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: mkamel@watfast.uwaterloo.ca).

Digital Object Identifier 10.1109/TIP.2004.833101

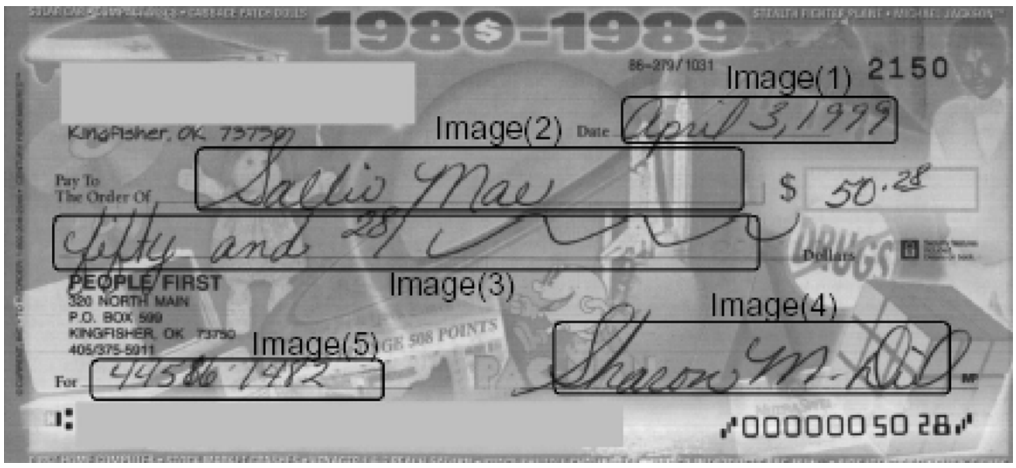

Fig. 1. Document image containing five subimages.

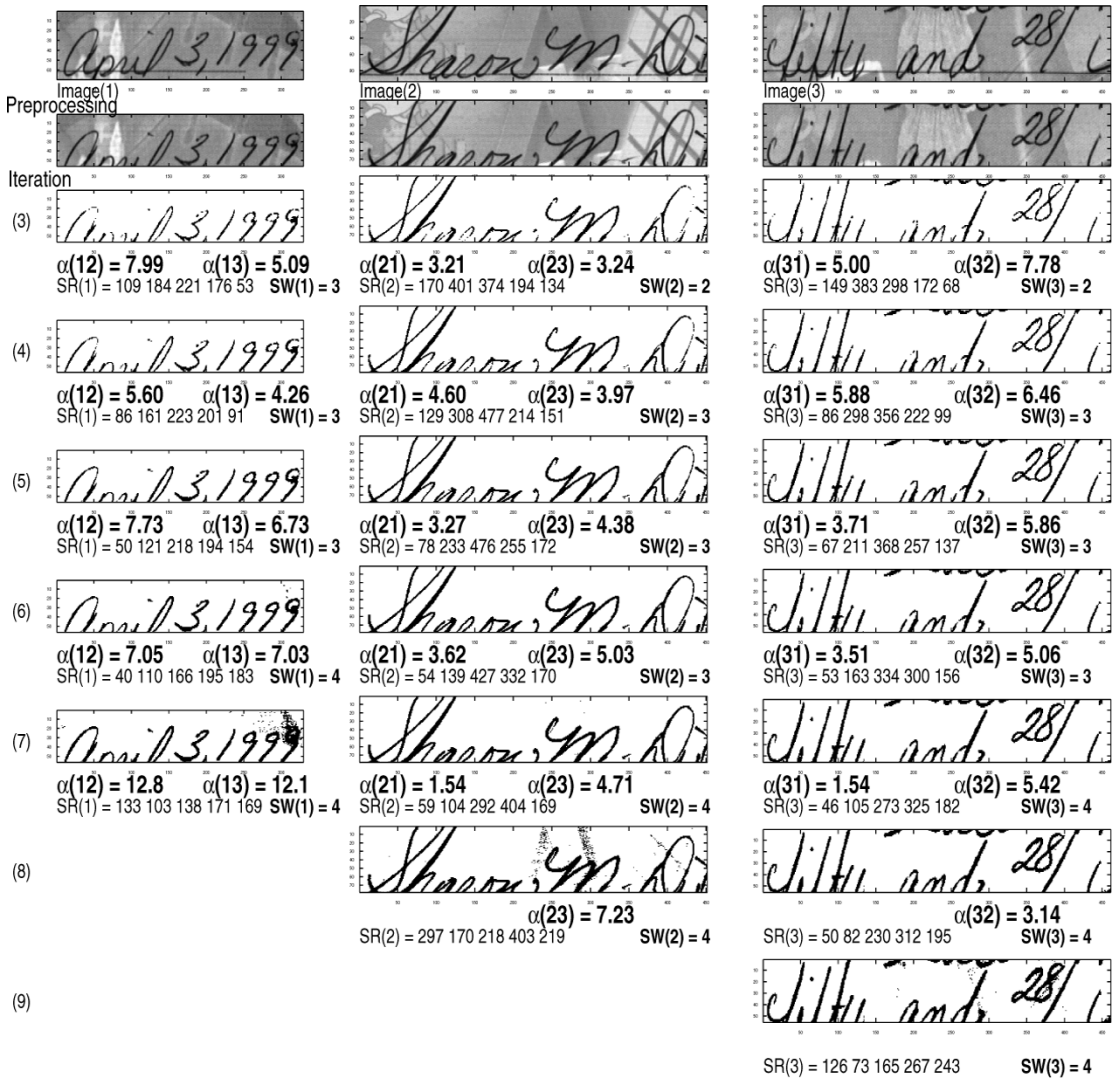

Fig. 2. Three subimages and their binarization at a sequence of candidate thresholds.

A. Feature Extraction

Notice that CT_7 in Fig. 2 failed to eliminate the background noise in Image (1). We want to infer such failure by comparing features of the binarized Image (1) with those of the other subimages. We used the gray-level and stroke-run features, which mainly reflect the underlying characteristics of the pen used in writing and are invariant to handwriting style or content. Now, because CT_7 failed to eliminate Image (1) noise, we

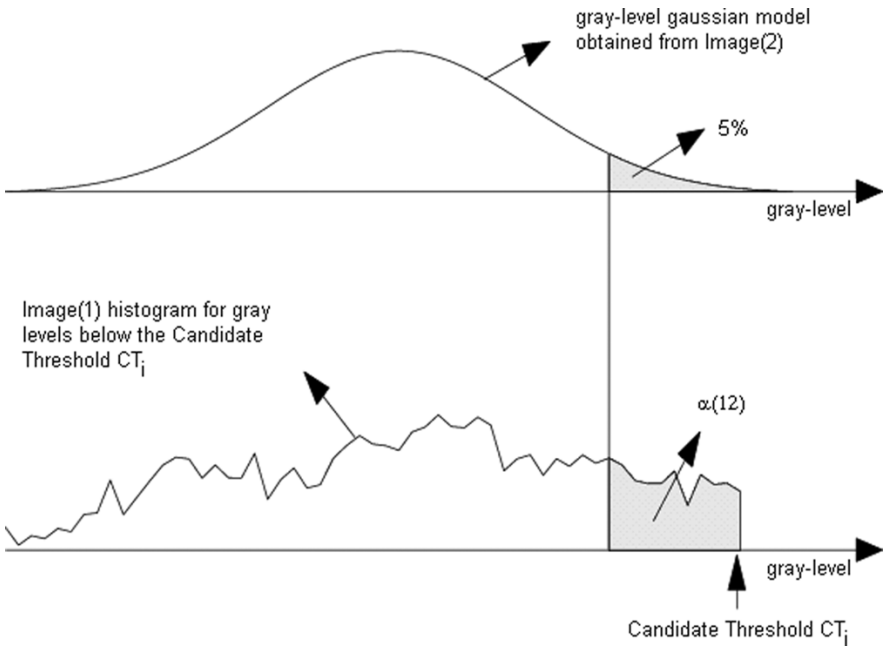

Fig. 3. Hypothesis test.

conclude that the CT_6 (the one that precedes the failure) is the optimal threshold; it preserved as much handwriting stroke data as possible without allowing the background noise interference.

1) Gray-Level Feature: We will show how to optimize the binarization of Image (y) using information from other another subimage, Image (x). Suppose that all subimages are binarized at a CT_i. If the binarized subimage Image (x) is noise free, then its gray-level statistics of the extracted pixels (pixels with gray-level lower than CT_i only) can be used to estimate the parameters of a Gaussian distribution $N_x \sim (\mu_x, \sigma_x)$. An empirical study done by Dawoud and Kamel [3] showed that the Gaussian distribution is the best among other distributions in representing the handwriting gray-level population. If the binarized subimage Image (y) is noise free also, then we expect its gray-level statistics of its extracted pixels (μ_y) to be similar to N_x. When CT_i fails to eliminate Image (y) noise, these statistics will become different from N_x. Experimentally, we found this expectation to be true, regardless of the noise's pattern or characteristics. So, by testing the following null and alternative hypotheses, we can infer whether or not a CT_i eliminated background noise from Image (y)

$$H_0 : \mu_x - \mu_y = 0 \tag{1}$$

$$H_1 : \mu_x - \mu_y \neq 0. \tag{2}$$

The hypothesis test is conducted by calculating the gray-level that corresponds to 5% of Image (x) noise-free Gaussian distribution. Then, as shown in Fig. 3, we calculate the number of pixels above this gray-level as a percentage of the total number of extracted pixels in the Image (y). This percentage is interpreted as $\alpha(yx)$, the probability of error associated with accepting the hypothesis that the gray-level statistics of the Image (y) obey the Gaussian model obtained from the Image (x), where $x, y \in K$ and $K = \{1, 2, \ldots, M\}$. As long as CT_i eliminates Image (y) background noise, this error will be small (around 5%), given that Image (x) is noise free also. However, this error will significantly increase when the CT_i fails to eliminate the noise, which is illustrated in the increase of Image (1)'s $\alpha(12)$ and $\alpha(13)$ at iteration 7 and Image (2)'s $\alpha(23)$ at iteration 8. The gray-level feature for all subimages will be a matrix in the following form:

$$\begin{bmatrix} - & \alpha(12) & \ldots & \alpha(1M) \\ \alpha(21) & - & \ldots & \alpha(2M) \\ \vdots & \vdots & \vdots & \vdots \\ \alpha(M1) & \alpha(M2) & \ldots & - \end{bmatrix}. \tag{3}$$

For example, the first row of this matrix is the probabilities of error associated with accepting the hypothesis that the gray-level statistics of the Image (1) obey the Gaussian models obtained from other subimages. The decision to accept or reject H_0 is reached by comparing $\alpha(yx)$ with a predetermined parameter called the gray-level rejection criterion [(GRC) the selection of GRC is discussed in Section II-B]. If $\alpha(yx)$ error exceeds GRC, H_0 is rejected; otherwise, it is accepted. When noise interferes in Image (z) at a certain CT_i, then Image (z) can no longer be used as a valid model to judge other subimages. Therefore, we remove it from the list of valid subimages used to calculate $\alpha(yz)$ at the following iterations, where $y \in K$, $K = \{1, 2, \ldots, M\}$.

When the noise interference starts at the same CT_i for two subimages Image (x) and Image (y), there is a chance that neither $\alpha(yx)$ nor $\alpha(xy)$ will increase to reflect this noise interference. In this case, this feature may fail to detect the interference.

2) Stroke-Run Feature: Another feature that was used to infer a CT_i's failure to eliminate noise in a subimage is the stroke-run histogram, referred to as $SR(j)_i[f]$, $j \in K$, and

TABLE I
SUMMARY OF STATISTICAL DECISIONS

Statistical decision	True state of null hypothesis	
	H_0 True	H_0 False
Reject H_0	Type I error	Correct
Do not reject H_0	Correct	Type II error

$K = \{1, 2, \ldots, M\}$, where M is the total number of subimages, and $f = \{1, 2, \ldots, L\}$, where L is the longest run to be counted. Our statistical study on stroke-run histograms of 4 200 check images (250 ppi) showed that the mean of the longest stroke run was about five pixels and the thinnest is about three pixels; therefore, we set $L = 5$. This histogram is created by scanning the binary images both horizontally and vertically to compensate for the variations in handwriting direction, and it records the frequency at which the run lengths occur. We noticed that a CT's failure to eliminate noise is always accompanied by a sharp increase in the frequency of stroke run of length one. This increase is demonstrated at CT_7 for Image (1), CT_8 for Image (2), and CT_9 for Image (3) of Fig. 2. This increase was noticed even when the background produced stroke-like patterns or patterns similar to handwriting. This is true because, regardless of the shape or pattern of background noise, it initially appears as scattered dots.

The stoke-run analysis calculates unit-run feature $UR(j)$

$$UR(j)_i = \frac{SR(j)_i[1] - SR(j)_{i-1}[1]}{SR(j)_{i-1}[1]} \tag{4}$$

where i is the number of iteration. When $UR(j)$ exceeds a predetermined parameter called the stroke rejection criterion [(SRC) the selection of GRC is discussed in Section II-B], we decide that the background noise has interfered and therefore we stop the iterations for that subimage. The selection of SRC should distinguish between the UR's moderate increase caused by normal formation of the characters and the sharp increase caused by noise interference. Unlike the gray-level feature, the calculation of this feature does not depend on information from other subimages. Therefore, it will not be affected by the noise interference in other subimages. In the situation where all subimages are equivalently complex and difficult, this feature will detect the background interference better than the gray-level feature.

B. Setting Rejection Criteria

In deciding whether a CT_i has succeeded or failed to eliminate a subimage noise, two types of error could be committed: type I error, which occurs when a CT succeeds and we decide that it failed, and type II, which occurs when a CT fails and we decide it that succeeded. These statistical decisions are summarized in Table I. Setting the rejection criteria GRC and SRC is critical part in optimizing this decision. If the rejection criteria are set at low values, type I error will increase, and if they are set at high values, type II error will increase. The costs of these errors depend on how these errors affect OCR readability of the extracted characters. Type I error results in terminating the algorithm; therefore, its cost will be high when the formation of characters is not complete. In the worst case, this error might cause the extracted characters to be broken and, as a result, difficult to be read by OCR. Type II error allows background noise to be extracted; therefore, its cost will increase with the increase of extracted noise. As we can see, the cost of these errors actually depends on growth of the extracted characters. To minimize the cost of these errors, we need to measure this growth, and then to let GRC and SRC be decreasing functions of that measure. We found the stroke-width feature (SW) to be a true measure of the characters' thickness and connectivity. SW is defined as the stroke run with the highest frequency in the stroke-run histogram of a subimage. That is

$$SW(j) = \arg \max_{f=1,\ldots,5} SR(j)[f]. \tag{5}$$

This feature reflects the average width of the extracted characters in Image (j).

The literature of handwritten character recognition contained many methods for the estimation of stroke width. For example, Arica *et al.* [1] introduced the two-scan procedure. The first scan calculates the stroke-run histogram by counting the black horizontal runs. Then, the mean width is taken as an upper limit. The second horizontal scan discards those strokes whose run length is greater than maximum width. Finally, stroke width is estimated as the average width of the strokes in the second scan. Ye *et al.* [12] proposed a stroke model to depict the local features of the characters as double edges in a predefined size. Their model detected thin connected components selectively, while ignoring large relatively large backgrounds. Our method to estimate the stroke width of the characters is simpler. The effect of background is eliminated by the iterative binarization in the first place.

Now, to reduce the cost of Type I and II errors, GRC and SRC are set to high values at lower SW, and then reduced with the increase of SW. CTs with $SW = 1$ are skipped (GRC and SRC are omitted for $SW = 1$), because the characters will be in their early stages of formation.

Table II summarizes the range of rejection criteria values that we used. As SW increases, GRC and SRC are decreased by 1/4. These ranges were determined experimentally, and we recommend using values within these ranges. Using values outside this range will increase either Type I or Type II errors.

C. Selecting Subimages Optimal Thresholds

To decide that a CT_i has failed to eliminate the background noise in a subimage, at least one feature of that subimage has to exceed its corresponding rejection criteria value. When this happens, we stop the iterations for that subimage, and we also stop using its gray-level statistics as a model to judge other subimages by removing it from the list of valid subimages at the following iterations. The iterations are also terminated when f$SW(j) = 5$, which means that extracted characters in

TABLE II
SUMMARY OF THE RANGE OF REJECTION CRITERION PERCENTAGES AS FUNCTIONS OF SW

stroke-width feature SW	1	2	3	4	5
gray-level rejection criterion GRC	NA	18-12	14-9	11-7	8-5
stroke rejection criterion SRC	NA	25-15	19-11	14-8	10-6

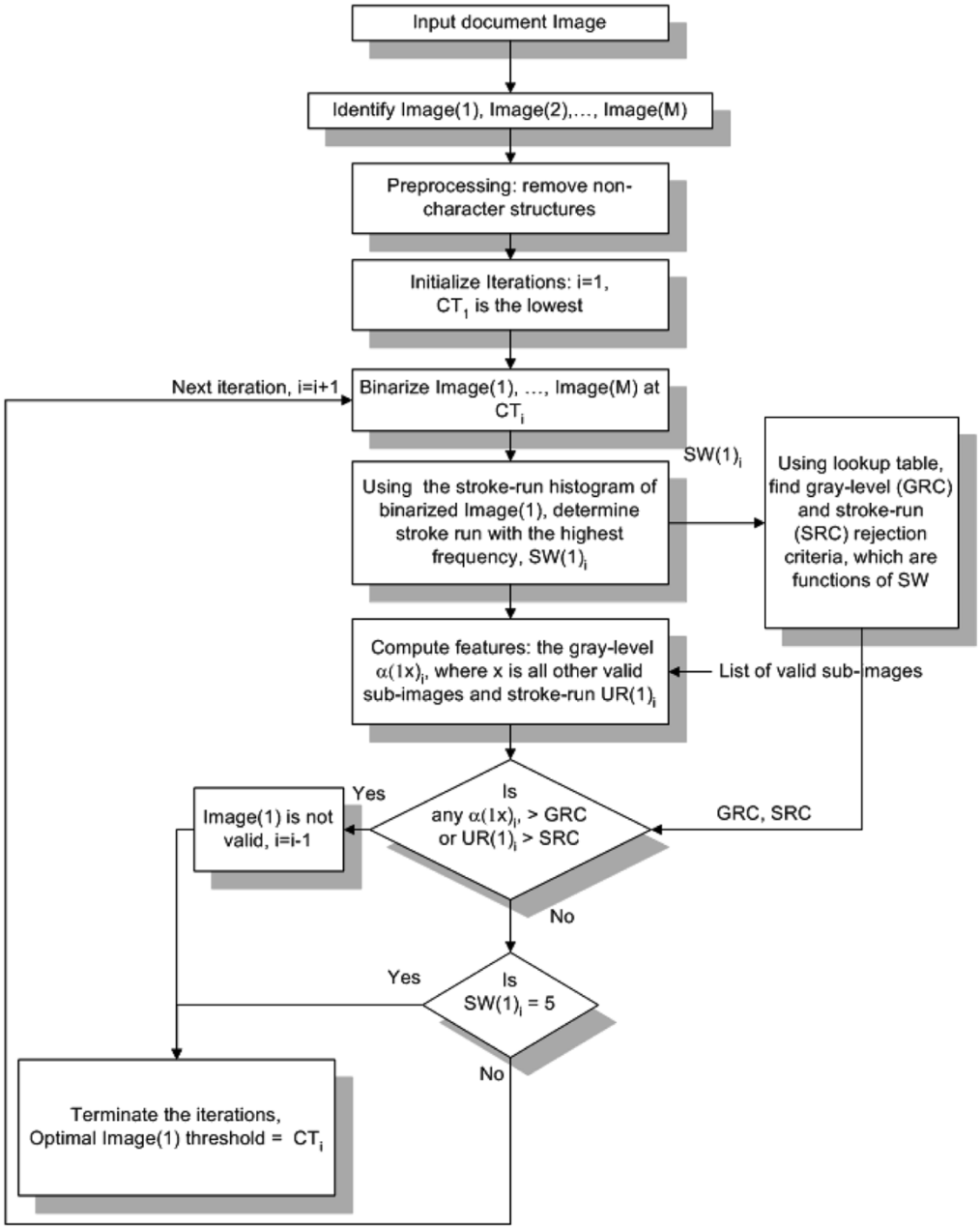

Fig. 4. Flowchart of multimodel approach applied on one subimage.

that subimage have reached full growth. In all cases, the optimal threshold for a subimage is chosen to be the highest CT_i that eliminated its background noise. When this iterative search was applied to the subimages shown in Fig. 2, the following decisions were made.

1) Iterations CT_{1-2}: skipped because $SW = 1$.

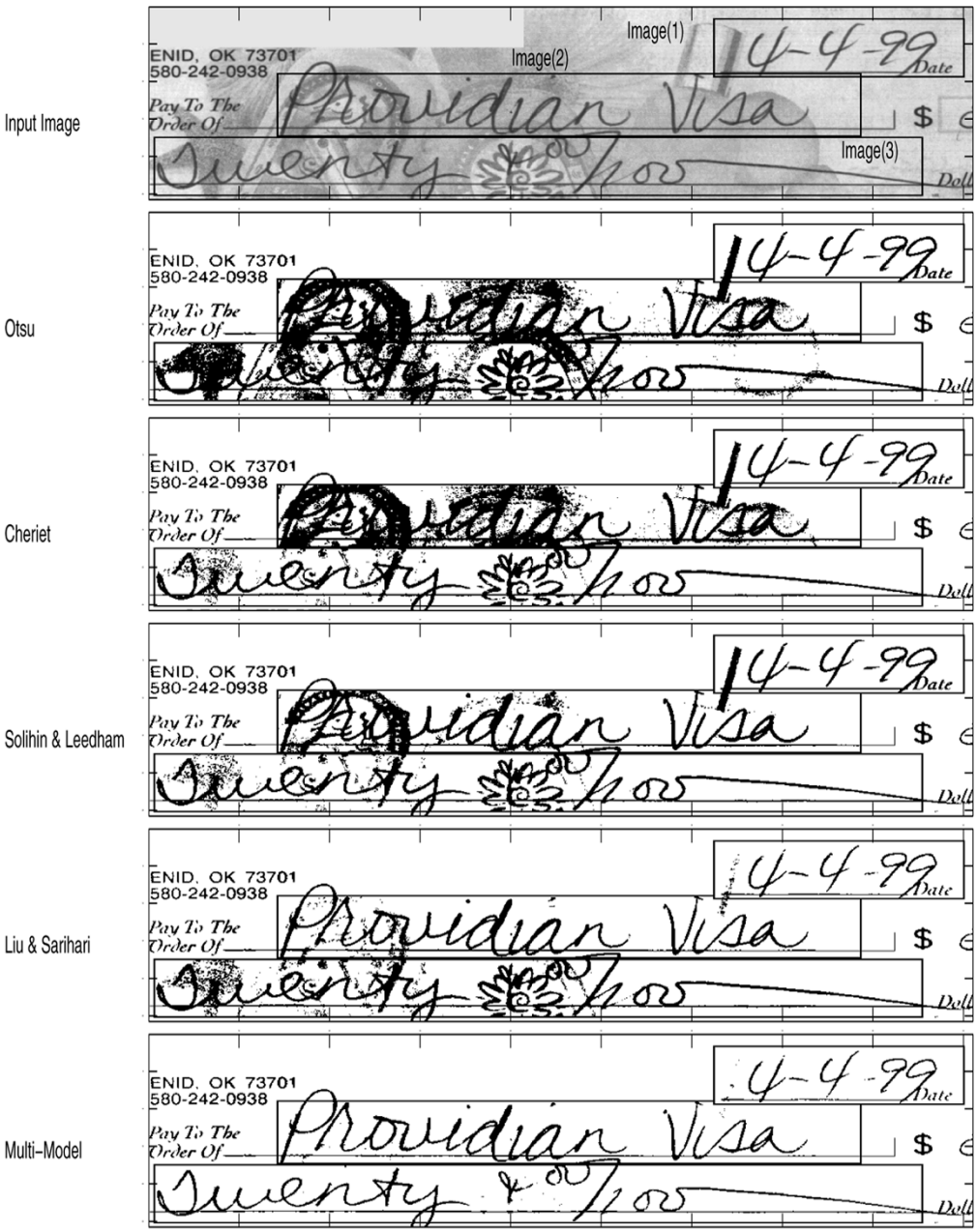

Fig. 5. Comparison result.

2) Iterations CT_{3-6}: background noise was not detected in any subimage.
3) Iteration CT_7: background noise was detected in Image (1), because $\alpha(12), \alpha(13) > \mathrm{GRC}_{SW(1)=4}$, as well as $UR(1) > \mathrm{SRC}_{SW(1)=4}$. Therefore, CT_6 is the optimal threshold for Image (1). Also, Image (1) is removed from the list of valid subimages at following iterations.
4) Iteration CT_8: background noise was detected in Image (2), because $\alpha(23) > \mathrm{GRC}_{SW(2)=4}$, as well as $UR(2) > \mathrm{SRC}_{SW(2)=4}$. Therefore, CT_7 is the optimal threshold for Image (2). Also, Image (2) is removed from the list of valid subimages at following iterations.
5) Iteration CT_9: background noise was detected in Image (3), because $UR(3) > \mathrm{SRC}_{SW(3)=4}$. Therefore, CT_8 is the optimal threshold for Image (3).

Fig. 4 summarizes the flowchart of the multimodel algorithm's application on one subimage, and it should be applied on the remaining subimages also.

This method is actually a generalization of Liu and Srihari [5] and Dawoud and Kamel [3] algorithms. In Liu and Srihari's algorithm, candidate thresholds are generated by the iterative application of Otsu's method. The stroke-run histogram at each candidate threshold is used to extract texture features that will evaluate the binarization at these candidate thresholds. The unit run feature is one of these features. An optimal threshold is then selected from these candidate thresholds to achieve certain desirable binarization qualities. Its main disadvantage is that the difference between two candidate thresholds is relatively large. The beginning of noise interference may not be detected when it happens at a threshold that lies between these

TABLE III
SUMMARY OF CHARACTER RECOGNITION AND ERROR RATES

Binarization method	correct	substituted	deleted	inserted
Multi-model	69.1	24.7	6.2	20.6
Yanowitz and Bruckstein	64.5	28.3	7.2	28.1
Liu and Srihari	64.0	30.7	5.3	37.3
Cheriet	62.5	29.2	8.3	35.1
Otsu	60.2	32.2	7.6	39.8
Solihin and Leedham	58.1	35.9	6.0	29.3

widely spaced candidate thresholds. In the multimodel algorithm, the difference between two candidate thresholds is limited to eight gray-levels; therefore, the beginning of noise interference is guaranteed to be detected. In addition to that, Liu and Srihari's algorithm relied only on features extracted from the stroke-run analysis, while the multimodel algorithm also relied on the gray-level features. As we will see in Section III, the multimodel algorithm performed significantly better than Liu and Srihari's algorithm in suppressing background noise.

Dawoud and Kamel's algorithm [3] optimizes the binarization of a part of a document image that suffers from noise interference, called the target subimage, using information extracted from another noise-free part of the same image, the model subimage. It requires prior knowledge about the noisiness of such subimages. The multimodel algorithm is a generalization because it does not require such information.

III. EXPERIMENTAL RESULTS

The multimodel binarization algorithm was tested using a set of 4 200 check images (250 ppi), provided by our industrial partner. These images were selected so that they would represent a wide range of background complexity, noise levels, and pen types. Based on visual judgment, the multimodel algorithm performed significantly better than other algorithms. Fig. 5 shows an input gray-level image that contains three subimages and also shows a comparison between the binarization outputs of the multimodel algorithm, Liu and Srihari [5] algorithm, Solihin and Leedham's algorithm [9], and Cheriet's *et al.* algorithm [2], and Otsu's [7] algorithm. The multimodel algorithm performed exceptionally well, even when the images were very noisy or when the background produced noise patterns similar to handwriting.

An equally important part of the research is the quantitative comparative evaluation of the performance. Performance evaluation of low-level image processing algorithms, such as binarization, segmentation, edge detection, thinning, etc., is inherently difficult. A meaningful way to compare alternative algorithms is by integrating each alternative into an end-to-end OCR system and comparing their impact on overall system performance. We compared the multimodel algorithm with five other binarization algorithms: Yanowitz and Bruckstein's method [11], Otsu's method [7], Liu and Srihari's method [5], Cheriet's *et al.* method [2], and Solihin and Leedham's method [9]. These algorithms were selected because they performed well in Dawoud and Kamel's [3] comparative study.

Here, the date area, the payee area, and the legal amount area of the check image were used as Image (1), Image (2), and Image (3), respectively. All algorithms except the multimodel algorithm were applied on each subimage separately. The multimodel algorithm was applied on the three subimages simultaneously. The binarization outputs of the three subimages were then fed into an off-line handwritten recognition engine. Then, we used scoring package [4] to automatically analyze the recognition results; the hypothesized string generated by the recognition system is reconciled with the reference (correct) string, using the dynamic string alignment algorithm. This algorithm uses dynamic programming to find the minimum distance between the two strings given penalties for character substitutions, deletions, and insertions. Table III summarizes the character recognition and error rates for total of 2 136 checks. The multimodel algorithm performed significantly better than other binarization algorithms; it resulted in the highest recognition rate and the lowest substitution rate, which means that it consistently preserved the handwritten characters. The insertion errors of other binarization algorithms were relatively high. This was due to the fact that OCR was actually trying to classify, as characters, parts of the background that were not eliminated. Our algorithm maintained low insertion rate because it was able to eliminate the background efficiently.

IV. CONCLUSION

Multimodel approach deals with the binarization of subimages; the threshold of each subimage is optimized using information extracted easily from the same subimage and from other subimages also. Simple spatial features, which capture the underlying characteristics of the strokes and are invariant to the handwriting style or content, are used to detect and eliminate the background noise interference. It can be applied to different types of documents where we do not have prior knowledge about the noisiness of the subimages.

In conclusion, superior performance was demonstrated via extensive experiments with our algorithm as compared with other algorithms. When applied to a set of images that represent wide range of background complexity and noise levels, the multimodel algorithm succeeded in eliminating the background, and in preserving the handwritten characters. As a result, higher recognition rate and lower substitution, insertion, and deletion error rates were achieved.

REFERENCES

[1] N. Arica and F. T. Yarman-Vural, "Optical character recognition for cursive handwriting," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 801–813, June 2002.

[2] M. Cheriet, J. N. Said, and C. Y. Suen, "A recursive thresholding technique for image segmentation," *IEEE Trans. Image Processing*, vol. 7, pp. 918–921, July 1998.

[3] A. Dawoud and M. Kamel, "Iterative model-based binarization algorithm for cheque images," *Int. J. Document Anal. Recognit.*, vol. 5, pp. 28–38, 2002.

[4] M. Garris and S. Janet, *NIST Scoring Package (Release 1.0)*: U.S. Dept. Commerce, Technol. Admin., Nat. Inst. Standards Technol., 1992.

[5] Y. Liu and S. N. Srihari, "Document image binarization based on texture features," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 540–533, Apr. 1997.

[6] W. Niblack, *An Introduction to Digital Image Processing*. Englewood Cliffs, N.J.: Prentice-Hall, 1986.

[7] N. Otsu, "A threshold selection method from gray-scale histogram," *IEEE Trans. Systems, Man, Cybern.*, vol. 8, pp. 62–66, Jan. 1978.

[8] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comput. Vis., Graph., and Image Processing*, vol. 41, pp. 233–260, 1988.

[9] Y. Solihin and C. G. Leedham, "A new class of global thresholding techniques for handwriting images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 761–768, Apr. 1999.

[10] O. D. Trier and A. K. Jain, "Goal-directed evaluation of binarization methods," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 1191–1201, Aug. 1995.

[11] S. D. Yanowitz and A. M. Bruckstein, "A new method for image segmentation," *Comput. Vis., Graph., and Image Processing*, vol. 46, pp. 82–95, 1989.

[12] X. Ye, M. Cheriet, and C. Y. Suen, "Stroke-model-based character extraction from gray-level document images," *IEEE Trans. Image Processing*, vol. 10, pp. 1152–1161, Aug. 2001.

[13] M. Zhao, Y. Yang, and H. Yan, "An adaptive thresholding method for binarization of blueprint images," *Pattern Recognit. Lett.*, vol. 21, pp. 927–943, 2000.

Amer Dawoud received the B.Sc. degree in electrical engineering from Yarmouk University, Jordan, in 1988, the M.Sc. degree in electrical engineering from Kuwait University, Kuwait, in 1998, and the Ph.D. degree in systems design engineering, University of Waterloo, Waterloo, ON, Canada, in 2002.

Currently, he is a Postdoctoral Research Associate and a part-time Faculty at the Electrical and Computer Department, University of South Alabama, Mobile. Dawoud's research interests are in signal/image processing, computer vision, machine intelligence, modern control and pattern recognition, with applications in automatic target tracking in infrared imagery, document processing systems, and autonomous guide vehicle systems. Also, he is interested in energy conversion in the vehicle braking systems.

Mohamed S. Kamel received the B.Sc. (Hons) degree in electrical engineering from the University of Alexandria, Alexandria, Egypt, in 1970, the M.Sc. degree in computation from McMaster University, Hamilton, ON, Canada, in 1974, and the Ph.D. degree in computer science from the University of Toronto, Toronto, ON, in 1981.

Since 1985, he has been with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON. He is currently a Canada Research Chair Professor in Cooperative Intelligent Systems and Director of the Pattern Analysis and Machine Intelligence Laboratory. From 1989 to 1990, he was a Visiting Faculty at the University of Toronto and Purdue University, West Lafayette, IN. From 1996 to 1997, he was a Visiting Professor at the ACE NASA Center, University of New Mexico, Albuquerque, and the University of Porto, Porto, Portugal. He served as a Consultant for General Motors, NCR, Diffracto, Bell & Howell, IBM, Northern Telecom, and Spar Aerospace. He is a member of the Board of Directors and Co-Founder of Virtek Vision, Inc., Waterloo. He is the Editor-in-Chief of the *International Journal of Robotics and Automation* and Associate Editor of *Intelligent Automation and Soft Computing*, *Pattern Recognition Letters*, and the *International Journal of Image and Graphics*. He has authored and coauthored over 200 papers in journals, conference proceedings, and numerous technical and industrial project reports, and he holds two patents. Over 45 graduate students have completed their degrees under his supervision. His research interests are in computational intelligence, pattern recognition, and distributed and multiagent systems.

Dr. Kamel is a member of ACM and AAAI. He is the Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS. Based on his work at the NCR (from 1980 to 1983), he received the NCR Inventor Award. He is also a recipient of the Systems Research Foundation Award for outstanding presentation in 1985 and the ISRAM best paper award in 1992. In 1994, he was awarded the IEEE Computer Society Press outstanding referee award. He was also a coauthor of the best paper in the 2000 IEEE Canadian Conference on Electrical and Computer Engineering.