A GENERIC SYSTEM TO EXTRACT AND CLEAN HANDWRITTEN DATA FROM BUSINESS FORMS

Suen, C.Y.; Cheriet, M.

A GENERIC SYSTEM TO EXTRACT AND CLEAN HANDWRITTEN DATA FROM BUSINESS FORMS

XIANGYUN YE [1,2] MOHAMED CHERIET[1,2] AND CHING Y. SUEN[1]

1. *Centre for Pattern Recognition and Machine Intelligence*
Concordia University, Suite GM606, 1455 de Maisonneuve Blvd. West
Montréal, Québec H3G 1M8, Canada

2. *Imagery, Vision and Artificial Intelligence Laboratory*
École de Technologie Supérieure, University of Québec
1100, Notre-Dame West, Montréal, Québec H3C 1K3, Canada

E-mail: {*xyye, suen@cenparmi.concordia.ca*}, *cheriet@gpa.etsmtl.ca*

A generic system is proposed to automatically extract and clean handwritten items from business forms. Handwritten data usually touch or cross preprinted form frames and texts. Having assumed that the item-of-interest can be located roughly by existing form registration methods, we focus only on the extraction and cleaning of the filled-in items. The proposed system includes training and cleaning phases. In the training phase, a model template is generated automatically from a blank form. Features such as the position and stroke width of the preprinted entities (including form frames and instructions) are extracted. In the cleaning phase, the system registers the template to the input form by landmark alignment. The form frames are removed and the handwritings are restored by morphological operations. When the handwritings are found touching or crossing preprinted texts, morphological operations based on statistical features are used to clean them. Both subjective and objective evaluations show promising results of the proposed system.

1 Introduction

As an essential operation in many business and government organizations on telecommunication, health care, finance, insurance, and public utilities, form processing remains a labor-intensive task, and the automation of this procedure has attracted intensive research interests.

A typical automatic form processing system includes two important parts: form image analysis and character recognition. In the form image analysis part, the system captures the form structure from a blank form and extracts user-entered data from the filled forms. The quality of the extracted items usually has a substantial effect on the performance of the whole system. In this paper we will focus on the latter issue only, and leave the character recognition problems that are out of the scope of this paper.

In the literature, there exist two research directions in form image analysis. The first one is based on form structure analysis, in which the filled-in items are extracted by following a set of rules describing the form structure [1,2,3]. The second direction is the filled-in item extraction or form dropout [4,5]. Using color dropout ink is a promising approach in separating the preprinted entities from the filled-in items. Yet it is not widely adopted due to the high cost in printing, and

In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 63-72

difficulty in changing existing designs, scanning, storage space and processing time. Binary images remain the major input type in form processing systems.

According to the rigidity in the form structure, all forms can be categorized into two major types [6]. One is called “physical form”, in which the positions and sizes of the fields do not vary. Typical examples include income tax forms, various application forms, parking tickets, *etc*. This type of forms is usually described by such features as intersection points [7], rectangles [6], or the images [8]. The other type of form is called “logical (or topological) form”, whose item fields can appear in different locations, while preserving certain important topological structures. Typical examples of this type include bankchecks, payment slips, and inventory lists, in which the items-of-interest are usually directed by preprinted baseline or machine-printed characters such as ‘$’, *etc*. The most important feature in analyzing this type of forms is line. Once the lines are located correctly, the filled-in items can be extracted based on knowledge rules [1,2] or bounding rectangles [9].

Many form-processing systems have been found successful when the filled-in items are machine-printed characters. However, as discussed in [4] and [6], a number of these approaches perform the extraction of filled-in items without any attention to *field overlap* problem. This happens when the filled-in items touch or cross form frames or preprinted texts. For the approaches that can drop out form frames or straight lines [4,10], the preprinted texts remain an unsolved problem. When the filled-in items are unconstrained handwritings, this problem is more pronounced. This can happen frequently and prevent the whole processing system from functioning properly. Thinking in reverse, we found some research in removing interference marks from handwritten[11] or machine-printed texts[12] are inspiring. Cleaning handwritten data in forms can be viewed as removing handwritten or machine-printed texts from the interference marks.

In this paper, we base our work on an existing form registration system that can roughly locate the item-of-interest. Our goal is to clean the obtained item fields, *i.e.*, separating the filled-in items from not only the form frames but also the preprinted texts. The effectiveness of the system is demonstrated by both subjective and objective evaluation. The system works well on “physical forms”, and is ready to be generalized to “logical forms”.

2 Problem modeling

As mentioned in the last section, many existing systems are able to register form structures, and roughly locate the item-of-interest. Therefore we base our work on an existing system, and try to extract the filled-in data crossing or touching the form frames or preprinted texts or instructions.

A typical sub-image obtained from the item location and extraction module of an existing form registration system consists of three components [4]:

- form frames, including black lines, usually called baselines, and blocks;

- preprinted data such as logos, and machine preprinted characters;
- user filled-in data (including machine-typed and/or handwritten characters and some check marks) located in predefined areas, called filled-in data areas, which are usually bounded by baselines and preprinted texts.

These three components actually carry two types of information: preprinted entities, which give instructions to the users of the forms; and the filled-in data. In most applications, "physical forms" are used, and the preprinted entities appear at the same expected positions. In an ideal case, the filled-in items can be extracted by a simple subtraction of a registered form model from the input image [8]. However, due to the distortion, skewing, scaling, and noise produced by the scanning procedure, it is almost impossible to find an exact match between the input image and the model form.

Consider a binary blank form image $I_b = \{I_b(x,y),(x,y) \in [M_b \times N_b]\}$, and a filled form image $I_f = \{I_f(x,y),\ (x,y) \in [M_f \times N_f]\}$, as sets of foreground pixels $I_b(x,y), I_f(x,y) = 1$. Assume they have the same structures, our goal is to interpret the structure of I_b and extract the filled-in data from I_f. Figure 1 shows a typical input and the desired output of a 'cleaning' system. Since the strokes of the filled-in characters can be either attached to or located across the form frames and preprinted texts, the problem of item cleaning involves the following steps:

- Estimating the positions of preprinted entities
- Separating characters from form frames or baselines
- Reconstructing strokes broken during baseline removal
- Separating characters from preprinted texts.

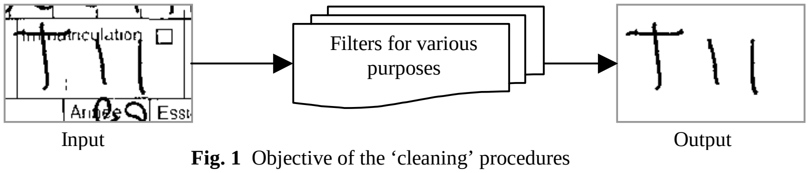

Fig. 1 Objective of the 'cleaning' procedures

3 Methodology

The key features that we used in form frame extraction are the locations of horizontal and vertical baselines. A horizontal baseline is composed of a group of 'long' horizontal line segments in the binary form image (we assume the skew angles are less than 5°). Similarly, a vertical baseline is composed of a group of 'long' vertical line segments. To extract and eliminate these baselines, the following operations are conducted:

3.1 Training

A precise analysis of the input data is a prerequisite to obtain the necessary information about the form frame (or baselines) and the item to be cleaned. In this stage, the system is exposed to an empty form field. From the training samples, we are able to collect statistical features not only from the handwritings to be extracted, but also the preprinted objects to be removed. These features include the existence, relative positions, average thickness of the baselines, *etc*. In order to process form fields that can not be described by regular shape, we propose to store a dilated form template for reference purpose[*]:

$$I_b^1 = I_b \oplus B \tag{1}$$

For the sake of simplicity, B has been chosen as a $n \times n$ square structuring element. The selection of size n depends on the precision of scanning and form registration procedures. In the experiments described in the following sections, $n = 5$.

3.2 Item cleaning and analysis

3.2.1 Smoothing

Edge smoothing presented in Suen *et al* [13] is performed to reduce the spikes and notches originated from the scanning noise. When a 3x3 window matches the pattern of Fig. 2(a), the central pixel is filled. Filling is also carried out when the window matches the pattern of Fig. 2(b) or its equivalents with 90°, 180°, and 270° rotations. Similarly, deletion of the central pixel is carried out when the 3x3 window matches either of the two patterns of Fig. 2(c) or (d), or any of the six equivalent patterns obtained with 90°, 180°, and 270° rotations.

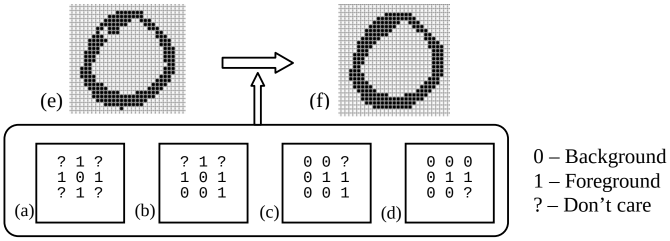

Fig. 2 ‘Smoothing’ of binary images

3.2.2 Baseline removal

Baselines are composed of line segments which are longer than a predefined threshold. They can be removed by local structural analysis [2,4], or by a set of morphological operations [14,15]. To facilitate the baseline procedure, the positions

[*] The numbers in the superscripts indicate different steps of operation; ‘*f*’ and ‘*b*’ stand for filled and blank forms.

of possible baselines can be determined by analyzing the horizontal and vertical projection histograms [16]. Baselines are therefore removed by applying morphological 'closing' operation in the surrounding region until all line segments longer than a fixed threshold are removed in the baseline regions.

We use SE_H and SE_V to denote the structuring elements that are a set of horizontally or vertically aligned pixels, whose lengths equal to the thresholds of the shortest line segments to be removed. Therefore, the frame line removal procedure can be described as

$$I_f^1 = I_f - I_f \bullet SE_H - I_f \bullet SE_V \tag{2}$$

Meanwhile, we are able to obtain the horizontal and vertical lines:

$$L_H = I_f \bullet SE_H \text{ and } L_V = I_f \bullet SE_V \tag{3}$$

In [15], we have proved that this morphological line removal method works well when the input image is in gray-level. When the input image is binary, this method is sensitive to noise and leaves some residue around the line region (Fig. 3). In the following sections, we will present a solution to this problem.

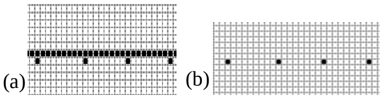

Fig. 3 Problems encountered by the morphological line removal method. (a) Input binary image with small noise caused by scanning under the line; (b) After line removal, the noise remains on the image.

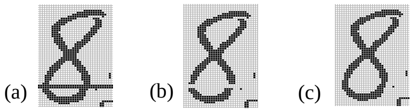

Fig. 4 Baseline removal and handwriting restoration. (a) Input binary image (partial image of a filled form); (b) baseline removed by equation (2); (c) handwriting restored by equation (4).

3.2.3 Information restoration

During the baseline removal procedure, some of the character strokes touching or crossing the baselines will be broken. This problem is likely to increase the error rate of the character recognition module.

Having observed that the morphological closing operation can act as a detector and a preserver of the information that intersects with the baselines, we applied a dynamic procedure of selecting the proper structuring element to restore the lost information. By approximating the orientations where the handwriting intersects a baseline in three directions (45°, 90°, and 135°), a dynamic kernel is able to merge the broken strokes with minimal distortion (Fig. 4) [15]. The restoration procedure can be described:

$$I_f^2 = \{I_f^2(x,y)\} \text{ , in which } I_f^2(x,y) = \begin{cases} \left(\bigcup_{k=1}^{3} I_f^1 \bullet D_k \right)(x,y) & \text{if } (x,y) \in L \\ I_f^1(x,y) & \text{otherwise} \end{cases} \tag{4}$$

in which $L = L_H \bigcup L_V$ is the region of baselines; $\{D_k, k = 1,2,3\}$ are the line-shape structuring elements in three directions (45°, 90°, and 135°), whose size is the average thickness of the baselines.

3.2.4 Model matching

As discussed in Section 2, it is difficult to find an exact match between the blank form model I_b and the filled form I_f. Due to the noise, distortion, and skewing in scanning, I_b and I_f can differ in size, position, and orientation. Theoretically speaking, Generalized Hough Transform can be used to detect a translated, rotated, and scaled version of a model object. In practice, it is not widely used in form processing due to the high computation cost. In this paper, we propose to locate some 'landmark' points in both the blank form model and the filled form, and deform the model to the filled form. The crossing points $C = \{(x, y) \in L_H \cap L_V\}$ in rectangular forms, the baselines $L_H \cup L_V$ in text-underline forms, and the machine printed '\$' or other known symbols can be used as 'landmarks'. For translation and scaling problems, we can find two landmark points (usually in diagonal directions) $P_{LT}(x_{LT}, y_{LT})$ and $P_{RB}(x_{RB}, y_{RB})$ in the model and the filled form respectively, the model form can be linearly deformed to $I_b^2 = \{I_b^2(x, y)\}$ according to the following expression:

$$I_b^2(x,y) = I_b^1((x - x_{LT}^b) * \frac{(x_{RB}^f - x_{LT}^f)}{(x_{RB}^b - x_{LT}^b)} + x_{LT}^f, (y - y_{LT}^b) * \frac{(y_{RB}^f - y_{LT}^f)}{(y_{RB}^b - y_{LT}^b)} + y_{LT}^f) \qquad (5)$$

For rotation problem, equation (5) can be replaced by affine transform incorporating more than two landmark points. In Section 3.1, we have stored a dilated blank form model in $I_b^1 = \{I_b^1(x, y)\}$, therefore, $I_b^2 = \{I_b^2(x, y)\}$ covers all possible preprinted entities in the filled form image, and the size of I_b^2 becomes the same as that of I_f^2. This step gives us the approximate positions of the preprinted entities and the filled items, and enables us to extract them in the next step.

3.2.5 Seeded region growing based on Area-Of-Interest(AOI)

In our system, the landmark points are selected as the corners of the Area-Of-Interest (AOI), which is usually a bounding rectangle surrounding the item-of-interest. Ideally, all filled-in items should appear only in AOI, and do not extrude the bounding region. However, this seldom happens in real-life applications. We propose to use a seeded region growing method to search all 8-connected components in the AOI. The seeds are chosen by

$$S = \{s(x, y), (x, y) \in rect(P_{LT}, P_{RB}) \cap I_f^2 \cap \bar{I}_b^2\} \qquad (6)$$

in which $\bar{I}_b^2$ is the set of background pixels found in the dilated and deformed model form, and $rect(P_{LT}, P_{RB})$ is a rectangle whose left-top and right-bottom points are P_{LT} and P_{RB}, respectively. This can help to discard the noise components left in Section 3.2.2, and extract the filled-in data as:

$$H^0 = \{h^0(x,y), (x,y) \in I_f^2 \cap N_8(S)\} \tag{7}$$

Here $N_8(S)$ is the set of all 8-connected component originated from seed set S. When the filled-in data is isolated from any preprinted texts, *i.e.*, $H^0 \cap I_b^2 = \Phi$, the item extraction task is fulfilled. As we have discussed in Section 3.1, the blank form model is composed of two types of preprinted entities: form frames (including horizontal and vertical baselines $L_H + L_V$) and texts T, i.e., $I_b^2 = (L_H + L_V) + T$. In Sections 3.2.2 and 3.2.3, we have solved the problem when the filled-in data touch or cross the baselines: $H^0 \cap I_b^2 \neq \Phi$ and $H^0 \cap T = \Phi$. In the next section, we are going to give one possible solution when $H^0 \cap T \neq \Phi$.

3.2.6 Preprinted text removal

One of the most important characteristics of character objects is the stroke width. For each foreground pixel in a binary image, the stroke width is defined as $SW(x,y) = \min(SW_H, SW_V)$, in which SW_H and SW_V are the distances between the two closest background pixels in horizontal and vertical directions. We have observed that in most real-life applications, the form frames and the instructions are printed in relatively small fonts. When the users fill in the forms with ball or ink pens, the stroke width of the handwritings is usually larger than that of the preprinted entities. This is confirmed by experiments on the form samples we have collected. The histograms of the handwritings and the preprinted entities in 10 form samples scanned at 300 DPI are shown in Fig. 5, which clearly shows the different distributions. This observation helps us to distinguish the preprinted frames and texts from handwritings by eliminating the pixels whose corresponding stroke width is less than a threshold. The stroke width of the handwriting (t_{hw}) and the preprinted entities (t_{pp}) can be estimated at run-time by collecting histograms of stroke widths:

$$t_{hw} = \arg\max_{i=1}^{T}\{hist[i = SW(x,y) \mid (x,y) \in I_f^2 \cap \bar{I}_b^2]\} \tag{8}$$

$$t_{pp} = \arg\max_{i=1}^{T}\{hist[i = SW(x,y) \mid (x,y) \in I_f^2 \cap I_b^2]\} \tag{9}$$

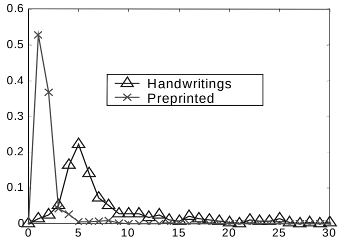

Fig. 5 Histograms of the stroke widths of handwritings and preprinted entities obtained from 10 sample form images scanned at 300 DPI.

Following a unified scheme of baseline removal and information restoration described in the previous paragraphs, we designed a set of binary morphological operators [17] at size $(t_{hw}+t_{pp})/2$ to remove the thin strokes in $H^0 \cap I_b^2$, and thus remove the connected preprinted texts from the handwriting. When t_{hw} and t_{pp} are equal, a possible solution can be found by reversing the noise removal procedure proposed in [11,12], at a high price of computational complexity. An example of the cleaning procedures and the intermediate results are illustrated in Fig. 6.

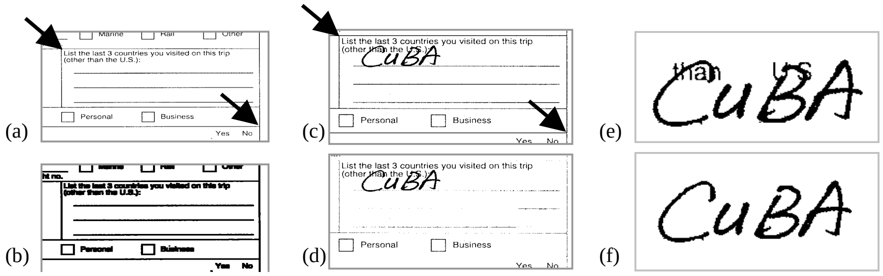

Fig. 6 An example of cleaning procedures. (a) Blank model form; (b) dilated blank form, stored as template; (c) Input filled form; (d) Baseline removal and stroke restoration; (e) seeded region growing based on AOI (the black arrows indicate the crossing points used as the landmarks to register the blank form to the filled one); (f) preprinted text removed by eliminating pixels whose corresponding stroke width is less than or equal to 3 ($t_{hw}=5$ and $t_{pp}=1$)

4 Evaluation of the cleaning methods

The evaluation of the performance and effectiveness of the proposed approach is conducted in both subjective and objective manners. The experiments used a set of 122 sub-images obtained from ‘license’ field of the parking tickets; the filled-in data include handwritten uppercase letters and digits. The system is trained on one blank form image. We assume that the size and position of the form frames are fixed in the input images, and the skew angle of the frame is less than 5°. Since the existing form processing system has taken care of the form registration and deskewing, the assumptions are validated by the testing images as practical.

4.1 Subjective evaluation

Visual inspection on the 122 original images shows that, out of a total of 735 user filled-in characters, 343 (46.7%) touch (including crossing) the preprinted entities such as form frame and texts. Digits and alphabet letters share approximately equal probabilities of touching the preprinted entities (47.5% for digits and 45.5% for letters). We observed that in cleaned images, out of 343 touching the preprinted entities, 4 are left with minor residual noise which do not distort the geometric features for recognition purpose; 4 are connected to handwriting intruding from neighboring item fields; and in one case, a stroke that overlaps entirely with the baseline is removed. On the average, 97.4% of the characters touching or crossing the preprinted entities are cleaned satisfactorily (Table 1).

Table 1 Subjective evaluation of the cleaning system

Char Type	Char Num	Touching with preprinted entities	
		Before cleaning	After cleaning
Digits	421	200 (47.5%)	5 (2.5%)
Letters	314	143 (45.5%)	4 (2.8%)
Total	735	343 (46.7%)	9 (2.6%)

4.2 Objective evaluation

The proposed approach is also evaluated objectively in a goal-directed manner, in which an image understanding module based on the results of the low-level image processing routine in question is used for quantitative evaluation. In this case, general-purpose recognizer is used. Since this recognizer is not able to segment touching character strings, we excluded 84 characters that touch neighbors from the testing. After cleaning, the recognizer can recognize 95.5% out of the remaining 651 characters (shown in Table 2). We observed that most errors are caused by abnormal writing styles, rather than residual noise in the cleaned image. This result proves the effectiveness of the ‘cleaning’ system, whose output is ready for character segmentation and recognition.

Table 2 Goal-directed evaluation of the cleaning system (test on isolated alpha-numerics). On a Pentium® II 200MHz, 64MB RAM PC, a 350x100 form field takes around 0.3s to extract a clean item.

(Alpha-Num)	Total	Rec. Rate	Err. Rate	Rej. Rate	Reliability
Number	651	622	29	0	
Rate		95.5%	4.5%	0.00%	95.5%

5 Conclusions

In this paper we have described a generic ‘cleaning’ system, which takes a blank form model as template, and registers the template to filled forms by aligning landmark points. The filled-in items are extracted by a seeded region growing method based on the Area-Of-Interest obtained from the landmarks. The form frames and the preprinted texts are removed using a set of morphological operations. Subjective and objective evaluations of the cleaning method show encouraging results for rectangular and underlined types of form fields. However, the system is not able to eliminate the noise from the filled-in data in the following cases: (i) the filled-in data cross or touch handwritings in the neighboring fields; (ii) the data area contains isolated characters that do not belong to current data fields; (iii) the filled-in data are expected to overlap with the preprinted text. The solution to these problems requires more intelligent analyses, such as feedback from recognizers and segmentation modules. This will be our goal in the future.

Acknowledgements

This research was partially supported by the grants from the Natural Sciences and Engineering Research Council of Canada (NSERC), and Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR). The authors would like to thank Mr. Claude Rheault from DocImage Inc. for providing training and testing form images, and Ms. Christine P. Nadal for her assistance in data collection.

Reference:

1. Tang, Y. Y., Suen, C. Y., Yan, C. D. and Cheriet, M., Financial document processing based on staff line and description language, *IEEE Trans. SMC*, **25**(5), 738-754, 1995.
2. Suen, C.Y., Lam, L., Guillevic, D., Strathy, N.W., Cheriet, M., Said, J.N. and Fan, R. Bank check processing system, *Int'l J. of Imaging Systems and Technology*, **7**:392-403, 1996.
3. Cesarini, F., Gori, M., Marinai, S. and Soda, G., INFORMys: A flexible invoice-like form-reader system, *IEEE Trans. PAMI*, **20**(7):730-745, 1998.
4. Yu, B., Jain A.K. A generic system for form dropout, *IEEE Trans. PAMI*, **18**(11):1127-1132, 1996.
5. Cracknell, C. and Downton, A. C., A colour classification approach to form dropout, *Proc. IWFHR 6*, 485-494, Taejon, Korea, 1998.
6. Arai, H and Odaka, K., Form reading based on background region analysis, *Proc. ICDAR 97*, 164-169, Ulm, Germany, 1997.
7. Wang, D. and Srihari, S. N., Analysis of form images, *Proc. ICDAR 91*, 181-191, Saint Malo, France, 1991.
8. Yuan, J., Xu. L. and Suen, C. Y., Form items extraction by model matching, *Proc. ICDAR 91*, 210-218, Saint Malo, France, 1991.
9. Belaid, Y., Belaid, A. and Turolla, E., Item searching in forms: application to French tax form, *Proc. ICDAR 95*, 744-747, Montreal, Canada, 1995.
10. Yoshikawa, K., Adachi, Y. and Yoshimura, M., Extracting the signature from traveler's checks, *Proc. IWFHR 6*, 657-666, Taejon, Korea, 1998.
11. Govindaraju, V. and Srihari, S. N., Separating handwritten text from overlapping non-textual contours, *Proc. IWFHR2*, 111-119, France, 1992.
12. Liang, S., Ahmadi, M., and Shridhar, M., Segmentation of handwritten interference marks using multiple directional stroke planes and reformalized morphological approach, *IEEE Trans. Image Processing*, **6**(8):1195-1202, 1997.
13. Suen, C.Y., Nadal, C., Legault, R., Mai, T.A. and Lam, L. Computer recognition of unconstrained handwritten numerals, *Proc. IEEE*, **80**(7):1162-1180, 1992.
14. Guillevic, D. and Suen, C.Y. Cursive script recognition: A fast reader scheme, *Proc. ICDAR 93*, 311-314, Tsukuba Science City, Japan, 1993.
15. Ye, X., Cheriet, M., Suen, C. Y. and Liu, K., Extraction of Bank-check Items by Mathematical Morphology, *Int'l J. Document Analysis and Recognition*, 2:53-66, 1999.
16. Cheriet, M., Said, J. N. and Suen, C.Y., A formal model for document processing of business forms, *Proc. ICDAR 95*, 210-213, 1995.
17. Ye, X., Cheriet, M. and Suen, C. Y., Model-Based Character Extraction from Complex Backgrounds, *Proc. ICDAR99*, 511-514, 1999.