IJDAR (2001) 4: 84–96

International Journal on Document Analysis and Recognition IJDAR

A generic method of cleaning and enhancing handwritten data from business forms

Xiangyun Ye[1,2], **Mohamed Cheriet**[2], **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] Laboratory for Imagery, Vision and Artificial Intelligence, École de Technologie Supérieure, University of Quebec, 1100, rue Notre-Dame Ouest, Montréal, Québec H3C 1K3, Canada

Received January 19, 2000 / Revised March 20, 2001

Abstract. The automation of business form processing is attracting intensive research interests due to its wide application and its reduction of the heavy workload due to manual processing. Preparing clean and clear images for the recognition engines is often taken for granted as a trivial task that requires little attention. In reality, handwritten data usually touch or cross the preprinted form frames and texts, creating tremendous problems for the recognition engines. In this paper, we contribute answers to two questions: "Why do we need cleaning and enhancement procedures in form processing systems?" and "How can we clean and enhance the hand-filled items with easy implementation and high processing speed?" Here, we propose a generic system including only cleaning and enhancing phases. In the cleaning phase, the system registers a template to the input form by aligning corresponding landmarks. A unified morphological scheme is proposed to remove the form frames and restore the broken handwriting from gray or binary images. When the handwriting is found touching or crossing preprinted texts, morphological operations based on statistical features are used to clean it. In applications where a black-and-white scanning mode is adopted, handwriting may contain broken or hollow strokes due to improper thresholding parameters. Therefore, we have designed a module to enhance the image quality based on morphological operations. Subjective and objective evaluations have been studied to show the effectiveness of the proposed procedures.

Keywords: Form processing – Item extraction – Handwriting recognition – Goal-directed evaluation – Mathematical morphology

1 Introduction

Most business and government organizations use forms as the major means of collecting information. The huge quantities of forms make manual processing a labor-intensive task, and the automation of this procedure has therefore attracted intensive research interest. A typical automatic form processing procedure includes two indispensable 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-in areas. In the character recognition part, the extracted items are sent to optical character recognition (OCR) engines, and the recognition results are stored for subsequent processing.

The form image analysis can be conducted in two principal ways. 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 structures [1–3]. The second way is the extraction of filled-in items or form dropout [4,5]. Using color dropout ink is regarded as a promising approach in separating the preprinted entities from the filled-in items [5]. Yet the high cost in printing and scanning, expensive computation and storage, along with difficulties in changing existing designs prohibit it from being widely adopted. In real-life applications, 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 main types [6,7]. One is the *rigid form*, in which the physical information such as positions and sizes of the fields remain stable. Typical examples include income tax forms, census forms, various application forms, and vehicle violation tickets, etc. These forms are usually described by such features as intersection points [8], rectangles [6], or the images [9]. The other type is the *flexible form*, where the item fields can appear in different locations while preserving certain important logical structures. Typical examples include bankchecks, payment slips, and inventory lists, in which the items-of-interest are usually directed by preprinted baselines or machine-printed characters such as '$', etc. While hierarchical methods [10] are powerful in understanding the structures of document images, a form description language (FDL) and staff lines (also called baselines) are more suitable in analyzing forms,

Correspondence to: M. Cheriet; e-mail: cheriet@gpa.etsmtl.ca

regardless of the rigidity of their structures [1,2]. Once the baselines are located correctly, the filled-in items can be extracted based on FDL or bounding rectangles [11].

Although the form structure analysis and model registration methods have attracted much attention [1–9], and many form-processing systems have been found to be successful when the filled-in items are machine-printed characters, the cleaning and enhancing of filled-in data is often obliterated. In fact, as discussed in [4] and [6], some of these approaches perform the extraction of filled-in items without any attention to the *field overlap* problem that 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 [4,12], preprinted texts remain an unsolved problem. When the filled-in items are unconstrained handwriting, this problem is more pronounced. Our statistics on an image set provided by a form processing company show that roughly 45% of hand filled-in data touch or cross the preprinted entities. Moreover, since the writing utilities are not constrained either, the often used black-and-white scanning mode may introduce hollow and broken strokes. These problems can happen frequently and hamper the recognition process, and thus prevent the whole processing system from functioning properly.

In this paper, we focus on the problems of cleaning and enhancing handwritten form items, and leave out the character segmentation and recognition problems that are beyond the scope of this paper. Our work is based on an existing form registration system that can roughly locate the item-of-interest, and the goal is to provide clean and clear items for the character recognizers. After the problems to be solved are described in Sect. 2, we describe in Sects. 3 and 4 the necessity for and methods of cleaning and enhancing *rigid forms* that are widely used in real-life. These methods are also generalized to *flexible forms* by incorporating pertinent knowledge. To demonstrate the effects of this cleaning and enhancing procedure on the recognition engines, we evaluate the system performance in Sect. 5 both subjectively and objectively. A conclusion is given in Sect. 6.

2 Problem description

As mentioned in the last section, the intensive studies in form structure analysis and model registration enable many existing systems to successfully register form structures, and roughly locate the items of interest. Therefore, we base our work on an existing system, and focus on the techniques of cleaning and enhancing the filled-in data which cross or touch the form frames or preprinted texts. 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 that are usually bounded by baselines and preprinted texts.

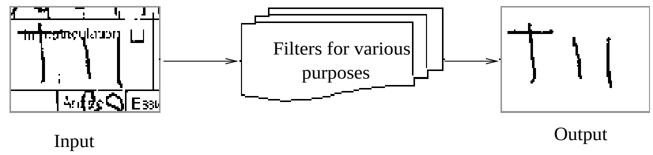


Fig. 1. Objective of the “cleaning” procedures: separating handwritten strokes from the frame lines and the preprinted texts

These three components actually carry two types of information: preprinted entities, which give instructions to the users of the forms; and the machine-typed or handwritten filled-in data. In most applications, *rigid forms* are used, and the preprinted entities appear only at the same expected positions. In the ideal case, the filled-in items can be extracted by a simple subtraction of a registered form model from the input image [9]. However, due to the distortion, skewing, scaling, and noise produced by the scanning and printing procedure, it is almost impossible to find an exact match between the input image and the model form.

In the following discussions, we assume the value of each pixel is $1 = foreground$ and $0 = background$, and denote an image as a set of foreground pixels:

$$I = I(x, y), (x, y) \in [M \times N] \text{ and } I(x, y) = 1$$

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 ($I_b(x, y)$ OR $I_f(x, y) = 1$) having 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 process of item cleaning and enhancing 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.
– Enhancing characters when necessary.

These procedures can be realized in two phases: learning and working phases. In the learning phase, the system takes a blank form as input, analyzes the form structure, and stores information such as the positions of preprinted entities, the scanning resolution, and typical stroke width of handwriting, in a template for future reference. In the working phase, the input form image is matched to the form model by aligning obvious features such as baselines and intersections. Form fields therefore can be cleaned by removing baselines, and restoring the intersecting handwriting. Once touching preprinted entities have been removed, the filled-in data are enhanced when necessary, and finally are provided to recognizers. The following sections will discuss these procedures in detail.

3 Cleaning of handwritten items

3.1 Form model learning

A precise analysis of the input data is a prerequisite to obtain the necessary information about the form frames (or baselines) and the items to be cleaned. In this stage, the system is exposed to an image of a blank form field. From this image, we are able to collect statistical features from the preprinted objects to be removed. These features include the existence, relative positions, average thickness and so on of the baselines, etc. In order to process form fields that cannot be described by regular shapes, we propose storing a dilated form template for reference purpose[1]:

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

For the sake of simplicity, B is chosen as an $n \times n$ square-shaped 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, $n = 5$ for 300 dpi (dots per inch) scanning modes.

3.2 Frame line removal

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 image. Similarly, a vertical baseline is composed of a group of long vertical line segments. The skew angle tolerance of this assumption is $\arctan(t/l)$, in which t and l are, respectively, the thickness and the minimal length of a segment that is considered long.

3.2.1 Frame line extraction and removal. Baselines are composed of line segments that are longer than a predefined threshold. The positions of possible baselines can be determined by analyzing the horizontal and vertical projection histograms [12], and they can be removed by simple local structural analysis [2,4,13], or by following a "good continuity criterion" [14,15], or by applying a set of morphological operations [16–19]. To implement the system in a generalized framework for both gray-level and binary images and facilitate the information restoration procedure, we adopt the framework derived in [19] and remove the baselines by applying morphological operations in the surrounding region until all line segments longer than a fixed threshold are eliminated. Based on shape, the mathematical morphology formalized by Serra [20] provides an efficient approach to processing digital images that were difficult to solve by linear filters [21]. Appropriately used, mathematical morphological operations tend to simplify image data by preserving their essential shape characteristics and eliminating irrelevant noise. A comprehensive introduction to practical applied mathematical morphology is given in [22]. The basic mathematical morphological operations are erosion and dilation. Based on the composition of erosion and dilation, opening and closing are defined. Considering the case of a binary image, let A be the set of points representing the binary 'on' pixels of the original binary image, and B be the set of points representing binary 'on' pixels of the structuring element. The basic morphological operations are defined below:

- Dilation $A \oplus B = \{b + a | \text{ for some } b \in B \text{ and } a \in A\}$.
- Erosion $A \ominus B = \{p | b + p \in A \text{ for every } b \in B\}$.
- Opening $A \circ B = (A \ominus B) \oplus B$.
- Closing $A \bullet B = (A \oplus B) \ominus B$.

Given the blank form model, the exact shapes of the horizontal and vertical baselines can be extracted by morphological reconstruction [39]. In this paper, the extraction of exact shapes of the form lines is not necessary as long as the form items are clearly identified. Therefore, we adopted the following baseline removal method that requires less computational efforts. Considering the definition of foreground and background pixels in Sect. 2, we may use the opening operation with linear shape structuring elements on foreground pixels to remove any line segment shorter than the predefined size. With SE_H and SE_V denoting the structuring elements that are a set of horizontally or vertically aligned pixels, and their lengths equal to the thresholds of the shortest line segments to be removed, the frame line removal procedure can be described as:

$$I_f^1 = I_f \otimes (I_f \circ SE_H \cup I_f \circ SE_V) \tag{2}$$

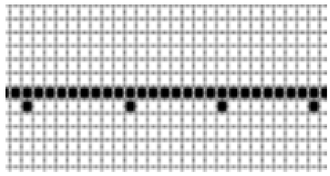
where $\otimes$ means **XOR**. Meanwhile, we are able to obtain the horizontal and vertical lines:

$$\begin{aligned} L_H &= I_f \circ SE_H \\ L_V &= I_f \circ SE_V \end{aligned} \tag{3}$$

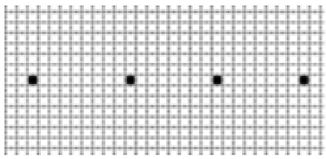
This morphological line removal method is very effective for gray-level images, but may leave some residue around the line region when the input image is binary (Fig. 2). In Sect. 3.2.4, we will present a solution to this problem.

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

[1] The numbers in the superscripts indicate different steps of the operation; f and b stand for filled and blank forms, respectively.

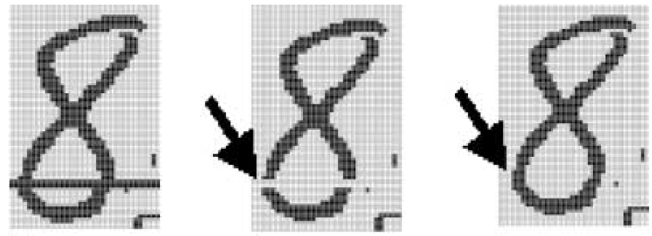


a



b

Fig. 2a,b. Problems encountered by the morphological line removal method on binary images: residue found in line regions. **a** Input image with small noise caused by scanning under the line; **b** after line removal, some residues remain on the image



a b c

Fig. 3a–c. Baseline removal and handwriting restoration. **a** Input binary image (partial image of a filled form); **b** baseline removed by Eq. (2); and **c** handwriting restored by Eq. (4). The *b*lack arrow indicates an example of how a handwritten stroke is broken during line removal and is restored by morphology with a dynamical kernel

module. While the line removal procedure has acquired wide consensus, the information restoration procedure remains an attractive and active topic. The gaps left by line removal procedures can be filled by linking nodes in a "Block Adjacency Graph" [4], simply by joining close gaps [14], analyzing the type of handwriting overlapping the lines [13], interpolating by neural networks shifted along the baselines [12], or applying morphological closing operations [16–19]. Some of these methods require time-consuming thinning processes [14], and encounter problems when crucial handwriting strokes touch or overlap with baselines [13,14]; some require expensive floating point computation [12]. Moreover, all of these methods except for [19] deal with only binary images. To unify the line removal and stroke restoration methods, we use the morphological scheme of [19] with little computation demand.

The principal idea of restoring handwriting strokes is to observe the morphological closing operation as a detector and preserver of the information that intersects with the baselines. A dynamic procedure of selecting the proper structuring element is applied to restore the lost information. In theory, the local orientation of strokes that intersect with baselines can be obtained by openings and closings at the corresponding images points [23]. In practice, by approximating the orientations where the handwriting intersects a baseline in three directions (45°, 135°, and 90° or 0° for restoring strokes that intersect with horizontal or vertical baselines, respectively), the dynamic kernel is able to merge the broken strokes with minimal distortion and little computational cost (Fig. 3). The restoration procedure is formalized as follows:

$$I_f^2 = \left\{ I_f^2(x,y), (x,y) \in [M_f \times N_f] \right\}$$

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

in which $L = L_H \cup L_V$ is the region of baselines; $\{D_k,\ k = 1,2,3\}$ are the line-shape structuring elements in three directions (45°, and 135°, and 90° or 0° for restoring strokes that intersect with horizontal or vertical baselines, respectively), with sizes chosen as the average thickness of the baselines. For each foreground pixel in a binary image, we define the stroke width 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. Defining the sizes of the line-shape structuring elements as the number of 'on' pixels, these are calculated as the cardinality of the sets $\{D_k, k = 1,2,3\}$:

$$Card(D_k) = \arg\max_{i=1}^{T} \{hist[i = SW(x,y) | (x,y) \in L]\} \quad (5)$$

where $k = 1,2,3$. T represents the predefined limit of largest thickness of the baselines, and $hist[S]$ is a histogram array gathered from set S that contains only natural integers.

3.2.3 Form model matching. As discussed in Sect. 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 involved in scanning, I_b and I_f can differ in size, position, and orientation. Theoretically speaking, the generalized Hough transform [24] 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 computational 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 $P_{RB}(x_{RB}, y_{RB})$ (LT stands for left-top, and RB stands for right-bottom) in the model and the filled form, respectively, and the model form can be linearly deformed to $I_b^2 = \{I_b^2(x, y)\}$ according to the following expression:

$$I_b^2(x,y) = I_b^1\left((x - x_{LT}^b)\frac{x_{RB}^f - x_{LT}^f}{x_{RB}^b - x_{LT}^b} + x_{LT}^f,\right.$$
$$\left.(y - y_{LT}^b)\frac{y_{RB}^f - y_{LT}^f}{y_{RB}^b - y_{LT}^b} + y_{LT}^f\right) \quad (6)$$
$$\forall (x,y) \in I_b^1$$

For the rotation problem, Eq. (6) can be replaced by an affine transform incorporating more than two landmark points. In Sect. 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 is equal to 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 following sections.

3.2.4 Seeded region growing based on area-of-interest. 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 the AOI, and not protrude beyond 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

$$S = \{s(x, y), (x, y) \in rect(P_{LT}, P_{RB}) \cap I_f^2 \cap \bar{I}_b^2\} \quad (7)$$

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 Sect. 3.2.1, and extract the filled-in data as:

$$H^0 = \{h^0(x, y), (x, y) \in I_f^2 \cap N_8(S)\} \quad (8)$$

Here, $N_8(S)$ is the set of all 8-connected components originated from seed set S . When the filled-in data are isolated from any preprinted texts, i.e., $H^0 \cap I_b^2 = \Phi$, the item extraction task is fulfilled. As discussed in Sect. 3.2.3, the blank form model is composed of two types of preprinted entities: form frames (including horizontal and vertical baselines $L_H \cup L_V$) and texts T, i.e., $I_b^2 = (L_H \cup L_V) \cup T$. In Sects. 3.2.1 and 3.2.2, 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 the solutions when $H^0 \cap T \neq \Phi$.

3.3 Preprinted text removal

When the filled-in areas are small and surrounded by preprinted texts or instructions, people are prone to write partially on top of them. In this case, the filled-in data touch the preprinted texts (Fig. 1), making it difficult to extract the proper data for recognition purposes. In the literature, little research has been found on removing these preprinted texts. Only [4] and [14] mentioned that this problem is difficult to handle, and high-level or knowledge-based post-processing needs to be employed in order to remove the ambiguities. In the assessment of an image set of practical business forms, we have found that the preprinted texts interfere with roughly 20% of the handwritten characters and become one of the major sources of noise that needs to be removed before the characters can be recognized.

3.3.1 Graph theory-based solution. The first solution is inspired by apparently irrelevant research: removing hand-drawn interference marks from handwritten or machine printed texts. Early research can be traced back to a precise definition of 'line' images by graph [25]. By dividing the image into *regular* and *singular* regions, the interfering lines and the handwriting can be represented by the edges and vertices of a graph. In [14] and [26], the definition of 'line' is extended to all long 'natural lines' including both straight lines and smooth curves. Thinning processes are utilized to vectorize the topological structure of the input image. The skeleton image obtained from thinning processes can be represented by nodes connected by paths, and the interference marks are extracted from the spine and long paths. Since the extraction of spine and long paths follows the "good continuity rules" [15], these graph theory-based methods are effective for removing curves in any direction as long as they have a smooth curvature.

Considering these methods in a reverse way, i.e., keeping the interference marks as the desired data, and removing the remaining texts of higher curvature and frequency, we believe these methods are helpful in eliminating the preprinted texts in some cases. However, the computational cost of the entire procedure of thinning, graph labeling, path traversing, and construction of multiple directional stroke planes is quite high. In addition, the parts of the handwritten data interfering with the preprinted texts are not always in a smooth curvature; sometimes the preprinted texts connect with the handwriting in continuous ligatures; sometimes the handwriting is a loop that has no spine or long path as defined in [14] and [26]. In these cases, the handwritten data can be extracted with either missing or extra parts. The major reason is that the graph theory-based methods

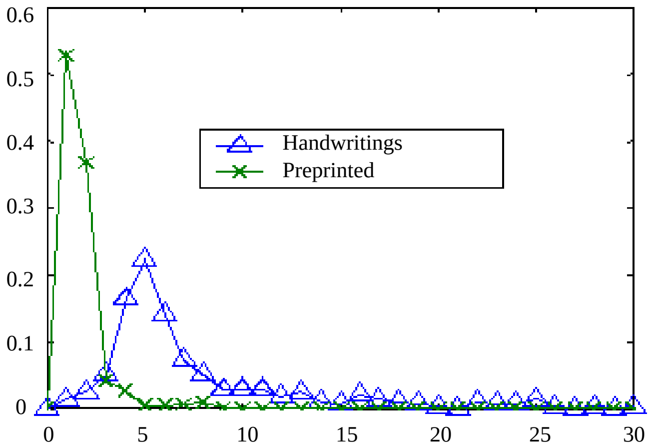


Fig. 4. Histograms of the stroke widths of handwriting and preprinted entities obtained from ten sample form images scanned at 300 dpi

rely on skeletons, which have lost all information about stroke-width properties that may help to distinguish the handwriting from the preprinted texts. In the next section, we present a more efficient and inexpensive solution based on statistical features.

3.3.2 Stroke model-based solution. One of the most important characteristics of character objects is the stroke width. In [27], we modeled character strokes as *double-edges*, and used this model to extract handwriting from gray-level images with complex backgrounds. As for binary images, the stroke model is simplified because the foreground and background pixels are all well defined. As defined in Sect. 3.2.2, for each foreground pixel in a binary image, the stroke width is calculated 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 many real-life applications, the form frames and the instructions are printed in relatively small fonts. When the users fill in the forms with ball-point or ink pens, the handwriting is usually thicker than the preprinted entities. This assumption is application-dependent, and is validated in our experiments with samples collected from different sources such as custom declaration forms, time sheets, parking tickets, etc. The histograms of the handwriting and the preprinted entities in ten form samples scanned at 300 dpi are shown in Fig. 4, which clearly shows the different distributions.

This observation helps us to distinguish the preprinted frames and texts from handwriting 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}) are estimated at run-time by collecting histograms of stroke widths (T represents a predefined limit for stroke width):

$$t_{hw} = \arg \max_{i=1}^{T} \{hist[i = SW(x, y)|(x, y) \in I_f^2 \cap \bar{I}_b^2]\} \quad (9)$$

$$t_{pp} = \arg \max_{i=1}^{T} \{hist[i = SW(x, y)|(x, y) \in I_f^2 \cap I_b^2]\} \quad (10)$$

Following a unified scheme of baseline removal and information restoration described in the previous paragraphs, we designed a set of binary morphological operators [27] 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:

$$\begin{aligned} &H^1 = H^0 \otimes PP \\ &PP = \{pp(x, y), (x, y) \in H^0 \cap I_b^2 \text{ AND} \\ &\qquad SW(x, y) < (t_{hw} + t_{pp})/2\} \end{aligned} \quad (11)$$

When t_{hw} and t_{pp} are equal, another possible solution can be found in Sect. 3.3.1 by reversing the noise removal procedure, with a high computational complexity cost. An example of the stroke model-based cleaning procedures and the corresponding intermediate results are illustrated in Fig. 5.

4 Item enhancing

If written with proper writing tools, and scanned with optimal parameters, the handwritten items are now ready to be processed by the recognizers. However, the degradation of images induced by scanning can be destructive to the recognizers. The recognizers based on structural features derived from skeletons are subject to errors when spikes and notches corrupt the character contours, or when hollow and broken strokes are present. The degradation can be more severe when the scanning parameters are adjusted for drop-out ink. Figure 6 illustrates some examples taken from the NIST SD3 database provided by the American *National Institute of Standards and Technology*. Without proper enhancement, these images will raise serious problems in the character segmentation procedure, let alone in the recognition.

4.1 Quality assessment of handwritten strings

The degradation models for machine printed texts have attracted much interest in the last decade [28–30]. For the global and local distortion of printed text images introduced in printing, photocopying, and scanning processes, the Morphological Document Degradation Model [29] and the Bell Labs Image Defect Model [28] provide comprehensive parameters that can be estimated by a set of degraded instances. However, due to the unconstrained nature of handwriting, the above-mentioned degradation models are not directly applicable to handwriting. To evaluate the quality of typewritten document images and automatically select an optimal restoration method, [31] used five quality measures to assess the extent of background speckle noise, and touching and broken characters. For unconstrained handwriting, the 'font size' information is not available. Therefore, we investigated the characteristics of the degraded handwriting images, and derived the following heuristic quality measurements to select enhancing methods according to the various types of degradation involved:

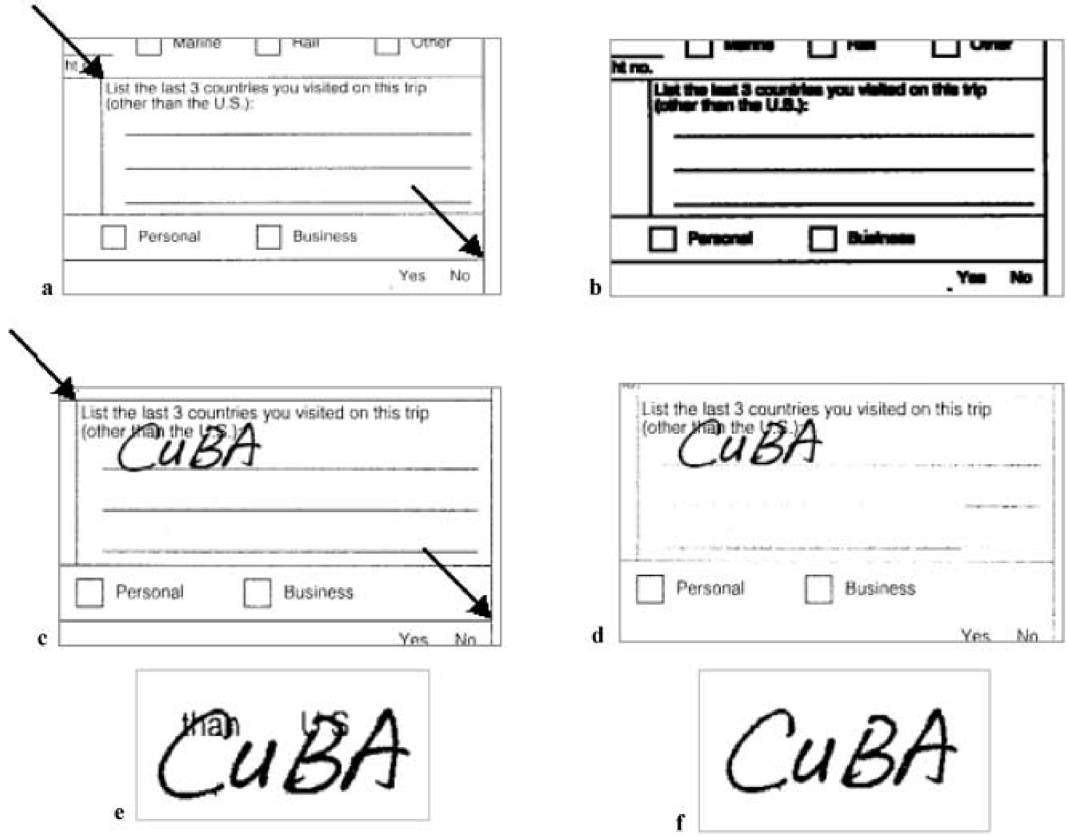


Fig. 5a–f. An example of cleaning procedures. **a** Blank model form; **b** dilated blank form, stored as template; **c** input filled form; **d** baseline removal and stroke restoration; **e** seeded region growing based on AOI (the *black arrows* indicate the crossing points used as the landmarks to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

a b

Fig. 6a,b. Some examples of degraded handwritten items taken from NIST hsf_3. **a** f1839_31_10; **b** f1839_31_21

- Hollow stroke measurement: the number of 8-connected black and 4-connected white components whose sizes are smaller than a predefined threshold.
- Broken stroke measurement: the standard deviation of the heights of all 8-

Although these quality measurements cannot model the degradation of handwriting completely, they are useful in detecting the hollow and broken strokes. When these quality measurements are high (compared to an empirically chosen threshold), the corresponding enhancement procedures are carried out as described in the following sections.

4.2 Hollow stroke filling

With improper writing tools (e.g., crayon) and scanning parameters, hollow strokes may appear in faint thick handwriting (Fig. 7a). The skeletons obtained from this type of image (Fig. 7b) will cause serious problems for recognizers based on structural information. To avoid this problem, we applied a morphological closing operation with a 3×3 structuring element (Fig. 7c).

a b c

Fig. 7a–c. Filling hollow strokes by morphological closing. **a** A handwritten item with hollow strokes caused by improper scanning parameters; **b** skeleton of the input image, showing problems raised to structural feature-based recognition methods; **c** filling the hollow strokes by morphological closing operations

4.3 Smoothing

Edge smoothing presented in [32] is performed to reduce the spikes and notches originated from the scanning noise. When a 3×3 window matches the pattern of Fig. 8a, the central pixel is filled. Filling is also carried out when the window matches the pattern of Fig. 8b

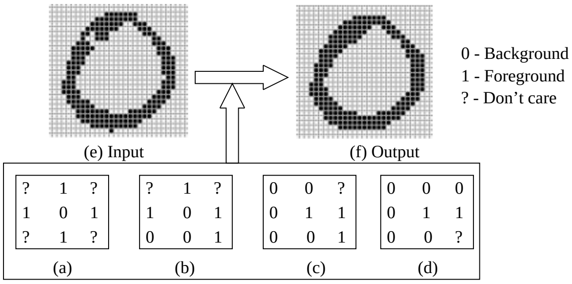


Fig. 8a–d. Smoothing of binary images. **a–d** Smoothing masks; **e** a typical input image; and **f** the output image

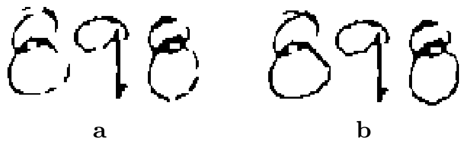


Fig. 9a,b. Mending broken strokes in a string image. **a** Input image; **b** output image

or its equivalents with 90°, 180°, and 270° of rotations. Similarly, deletion of the central pixel is carried out when the 3×3 window matches either of the two patterns of Fig. 8c or 8d, or any of the six equivalent patterns obtained with rotations of 90°, 180°, and 270°.

4.4 Broken stroke mending

Due to noise, distortions, and other factors during scanning, the gray-scale images produced from a scanner may lead to a loss of information when a thresholding technique is used to obtain binary images. When the handwriting needs to be recognized, the broken strokes along with touching characters accentuate the character segmentation problem. Simple connected component detection can no longer be used as the major approach to extracting characters for recognition purposes. A macrostructure analysis (MSA) stroke-mending method on single characters was proposed in [33]. It searches for all end-points that have only one neighbor, and defines the direction of an end-point by its relationship with the neighbors. The two strokes that end with reverse directions are most likely to be connected and those ending in parallel directions are least likely. Having assumed that an input image contains a single character for which only one outer contour is expected, the MSA stroke-mending algorithm repeats until only one outer contour is detected in the output image. Although effective on single character images, direct application of the MSA on real data is difficult, since the input images usually contain strings of characters. The following two questions arise:

– When should the stroke-mending algorithm be applied (Fig. 9a)?
– How can we avoid mending two nearby characters as shown in Fig. 10a?

To partially answer the questions (further solution involves character segmentation and recognition modules

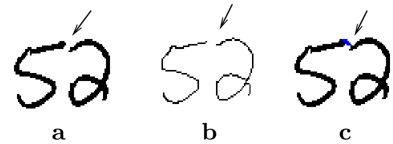


Fig. 10a–c. Problem of single character stroke-mending algorithms on a string image. **a** The two characters in a string are close to each other with only a small gap; **b** two end-points are found in the skeletonized image, satisfying the mending criteria; **c** the gap between the characters is falsely mended. The *arrows* indicate the gap causing problems

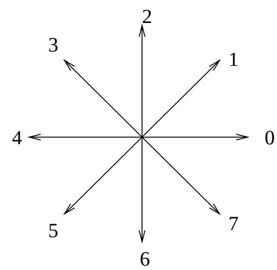


Fig. 11. Directions of end points

that are beyond the scope of this paper), we derived in practice the following heuristic rules:

– When the standard deviation of heights of all connected components is higher than a threshold, the input string image needs to be mended.
– The mending procedure starts with small connected-components.
– Instead of mending strokes far from each other as in [33], the mending is limited to a certain distance.
– Two components are not connected if both are larger than a predefined size.
– If only one of the two components satisfying the mending criteria of being larger than a predefined size, a general-purpose recognizer is used to recognize the larger component. If it is recognized as a character with high confidence, the two components are not connected.

In the mending procedure, we use a skeletonization algorithm proposed by [34], and keep the Freeman chain code directions [35] as shown in Fig. 11 for simplicity. Assuming the error tolerance of angle change in a continuous stroke is ±45° , the following directions of two stroke ends are considered to be compatible:

$$\mathit{Compatibility(Dir\_In,\ Dir\_Out)} = 1$$
$$\text{iff } \mathit{Dir\_In} = (\mathit{Dir\_Out} + 4 \pm 1)\ \%\ 8$$

If the distance between two compatible stroke ends is shorter than a predefined range, the two strokes are considered as one, and the gap between them is filled by line segments (Fig. 9b).

Now let us take a look at Fig. 6 again; the original and the enhanced images are shown in Fig. 12 along with their respective skeleton images. The effectiveness of the enhancement procedure is apparent from the smooth and continuous skeletons obtained.

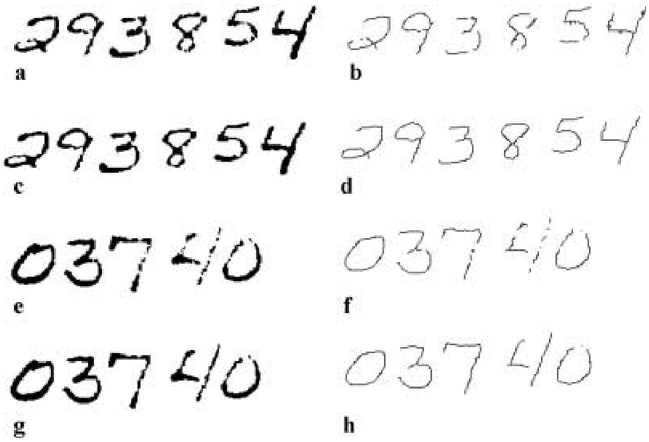


Fig. 12a–h. Enhancement of Fig. 6. **a**, **e** Original image samples from NIST SD3 database; **b**, **f** skeletons obtained from the original image; **c**, **g** enhanced image; **d**, **h** skeletons obtained from the enhanced image

5 Evaluation of the cleaning methods

The evaluation of the performance and effectiveness of the proposed cleaning and enhancing is conducted in both subjective and objective manners. The experiments used a set of sub-images obtained from the 'license' field of vehicle violation tickets, a type of *rigid forms*; 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.

5.1 Subjective evaluation of the "cleaning" procedures

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 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 (methods described in Sect. 3), out of 343 touching the preprinted entities, four are left with minor residual noise which do not distort the geometric features for recognition purpose; four are connected to handwriting intruding from neighboring item fields; and in one case, a stroke that overlaps entirely with the baseline is removed. Figure 13 illustrates some examples of both successful and unsuccessful cleaning. Figures 13a and 13b demonstrate the effectiveness of the proposed cleaning method in removing touching preprinted entities from the filled-in handwriting, while in Figs. 13c and 13d, touching handwriting intruding from neighboring item fields remains unsolved.

We hereby define the *cleaning rate* as $\frac{N_0^T - N_1^T}{N_0^T} \times 100\%$, where N_0^T and N_1^T are the numbers of characters that touch or cross the preprinted entities before and after the cleaning procedures, respectively. On average, the cleaning rate of our system reaches 97.4% (Table 1).

Table 1. Subjective evaluation of the cleaning system

		Touching with preprinted entities	
Char Type	Char Num	Before cleaning	After cleaning
Digits	421	200(47.5%)	5(1.2%)
Letters	314	143(45.5%)	4(1.3%)
Total	735	343(46.7%)	9(1.2%)

5.2 Objective evaluation

The proposed approach is also evaluated objectively in a goal-directed manner, which means an image understanding module based on the results of the low-level image processing routine in question is used for quantitative evaluation [36]. In our experiments, three general-purpose alphanumeric string recognizers are used. These recognizers are obtained from difference sources including commercial products and research projects, therefore the training set and recognition methods are inaccessible to us. Rather than comparing the performance of these individual recognizers, the purpose of using more than one recognizer in the evaluation is to prove that the improvement of performance brought about by the cleaning and enhancement procedure is independent of the features or methods that are used in the recognizers. Hereafter, we refer to them simply as *Recognizer* 1, *Recognizer* 2, and *Recognizer* 3.

5.2.1 Evaluation of the cleaning procedures at character level. To isolate the evaluation of the cleaning procedure from the segmentation problems of touching character strings, we conduct the first test by excluding characters that touch their neighbors. The number of characters whose true identities appear as the top choice of the recognizers' output is counted. Out of the manually selected 670 isolated characters in the same testing set as in Sect. 5.1, an average of 85.1% characters is correctly recognized. If intrinsically confusing pairs such as <'S', '5'> and <'Z', '2'> are ignored, the average recognition rate reaches 91.8%, with the highest recognition rate obtained from *Recognizer* 2, being 96.1% (shown in Table 2[2],[3]). Visual inspections of the mis-recognized characters show that abnormal writing styles, rather than residual noise in the cleaned image are the major cause of errors. This result confirms the effectiveness of the cleaning system as observed in the subjective evaluation.

[2] Due to the variablility of writing styles and the nature of the application, character pairs <'O','0'>, <'I', 'l'> are not distinguished.

[3] In many cases, character pairs <'5','S'>, <'Z','2'> are also hard to distinguish (Fig. 14). They are considered as compatible pairs when recognition rate is calculated.

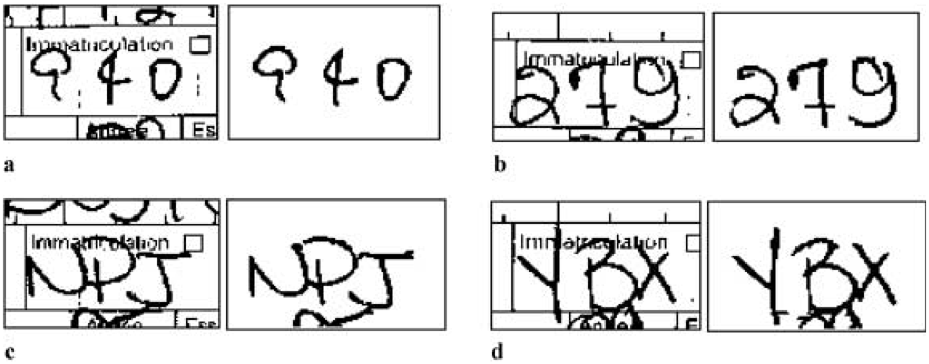


Fig. 13a,b. Examples of successful cleaning: touching frames and preprinted texts are removed. **c**, **d** Examples of remaining problems in cleaning: touching handwriting cannot be removed

Table 2. Goal-directed evaluation of the cleaning system on isolated alphanumeric characters

Total 670 isolated alphanumeric characters	Recognizer 1	Recognizer 2	Recognizer 3	Average
Number of correct ID on top choice	563	603	544	570
Non-rejection Rec. Rate	84.0%	90.0%	81.2%	85.1%
Number of compatible ID on top choice	609	644	593	615
Non-rejection Rec. Rate	90.9%	96.1%	88.5%	91.8%

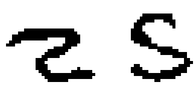


Fig. 14. An ambiguous string image: “25”, “ZS”, “2S”, or “Z5”?

5.2.2 Evaluation of the enhancing procedures at string level. The high recognition rate of isolated characters does not directly lead to high performance of a complete form reading system. When the expected output is a string instead of a single character, the segmentation problem becomes the major concern. In order to evaluate the effect of enhancing procedures on the system performance, we conduct the second test on a larger set of string images (239 strings, 1,460 characters in total) with various types of degradation as mentioned in Sect. 4.

Two aspects of the cleaning and enhancing procedures have been investigated: (a) the string segmentation rate representing the percentage of strings that are segmented into the right number of characters (denoted as ‘Str.Seg.’); and (b) the character recognition rate (denoted as ‘Char.Rec.’) *versus* reliability (limited to the strings that are segmented into right number of characters). As demonstrated in Table 3, while the cleaning procedure provides clean images for string recognizers and enables the whole system to function, the enhancing procedure boosts the performance of the entire system by more than 10% on both string segmentation and the character recognition rate. This improvement is independent of the internal structures and methods of individual string recognizers. A more comprehensive comparison on the character recognition rates and reliabilities obtained by rejecting characters recognized with confidence lower than different thresholds is illustrated in Fig. 15.

On top of the promising results we obtained from the cleaning and enhancing procedures, the implementation of the proposed procedures is simple. Most operations

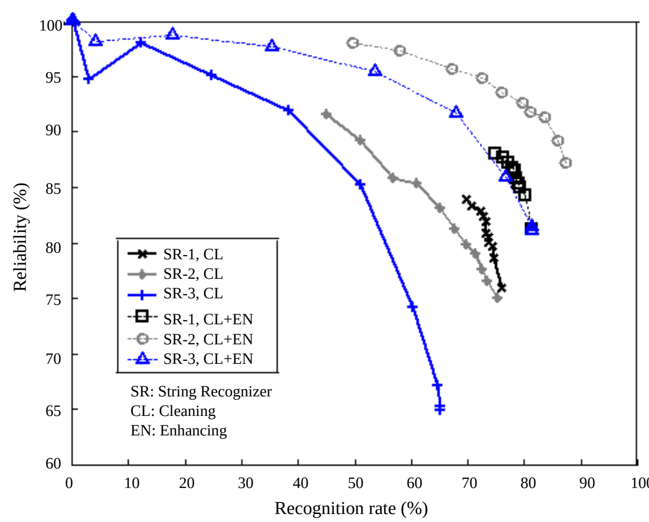


Fig. 15. The effectiveness of cleaning and enhancing procedures evaluated at the string level. The curves are obtained by varying the thresholds of confidence for rejecting a recognition output. While the cleaning procedure enables the system to function, the enhancing procedure can boost the performance to a much higher level

can be realized with a unified morphological scheme that is computationally inexpensive. References [37] and [38] have proven that morphological dilation or erosion operations with line-shaped structuring elements can be implemented with only three comparisons per pixel, regardless of the size of the structuring element. The computational complexity is therefore O(N), in which N is the number of pixels within the input image. Since most operations in this paper can be localized to item fields, whose sizes are normally small, the computation time is usually within an acceptable range. On a PentiumII 200 MHz, 64 MB RAM PC, extracting a clean and enhanced item from a 350×100 form field takes 0.15 s.

Table 3. Goal-directed evaluation of the cleaning and enhancing procedures on strings of alphanumeric characters (total 239 alphanumeric strings composed of 1,460 characters). 'CL' and 'EN' denote the cleaning and enhancing procedures, respectively

	Recognizer 1		Recognizer 2		Recognizer 3		Average	
	CL	CL+EN	CL	CL+EN	CL	CL+EN	CL	CL+EN
Str.Seg.	84.1%	91.8%	44.5%	79.8%	81.1%	88.1%	69.9%	86.6%
Non-rej. Char.Rec	75.9%	81.3%	75.1%	87.2%	65.0%	81.1%	72.0%	83.2%

6 Conclusions

In this paper, we have described a generic cleaning and enhancing system for automatic business form processing. It takes a blank form model as a 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. The system is not able to eliminate noise from the filled-in data in the following cases: (a) the filled-in data cross or touch handwriting in the neighboring fields; (b) the data area contains isolated characters that do not belong to current data fields; and (c) the filled-in data are expected to overlap with the preprinted texts. The solution to these problems requires more intelligent analyses, such as feedback from the recognizers and segmentation modules. In addition, we proposed enhancing methods for various types of image degradation, including filling hollow strokes and mending broken strokes. A goal-directed evaluation demonstrates the improvement achieved by enhancing procedures. Most parts of the proposed methods are based on a unified scheme of morphological operations, enabling easy implementation and fast processing speed, as well as direct applicability to both gray-level and binary input images. The proposed methods have been evaluated on real-life applications including rigid and flexible forms. The results demonstrate the importance and effectiveness of the proposed preprocessing techniques.

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, Ms. Christine P. Nadal for her assistance in data collection, and Mr. Jianxiong Dong for his help in typesetting and proofreading this paper.

References

1. Y.Y. Tang, C.Y. Suen, C.D. Yan, M. Cheriet: Financial document processing based on staff line and description language. IEEE Trans. Syst. Man Cybern. 25(5):738–754, 1995
2. C.Y. Suen, L. Lam, D. Guillevic, N.W. Strathy, M. Cheriet, J.N. Said, R. Fan: Bank check processing system. Int. J. Imaging Syst. Technol. 7:392–403, 1996
3. F. Cesarini, M. Gori, S. Marinai, G. Soda: INFORMys: a flexible invoice-like form-reader system. IEEE Trans. Pattern Anal. Mach. Intell. 20(7):730–745, 1998
4. B. Yu, A.K. Jain: A generic system for form dropout. IEEE Trans. Pattern Anal. Mach. Intell. 18(11):1127–1132, 1996
5. C. Cracknell, A.C. Downton: A colour classification approach to form dropout. Proc. Int. Workshop on Frontiers of Handwriting Recognition 6:485–494, Taejon, Korea, 1998
6. T. Watanabe, Q. Luo, N. Sugie: Layout recognition of multi-kinds of table-form documents. IEEE Trans. Pattern Anal. Mach. Intell. 17(4):432–445, 1995
7. H. Arai, K. Odaka: Form reading based on background region analysis, Proc. 4th Int. Conf. on Document Analysis and Recognition, pp. 164–169, Ulm, Germany, 1997
8. D. Wang, S.N. Srihari: Analysis of form images. Proc. 1st Int. Conf. on Document Analysis and Recognition, pp. 181–191, Saint Malo, France, 1991
9. J. Yuan, L. Xu, C.Y. Suen: Form items extraction by model matching, Proc. 1st Int. Conf. on Document Analysis and Recognition, pp. 210–218, Saint Malo, France, 1991
10. Y.Y. Tang, M. Cheriet, J. Liu, J.N. Said, C.Y. Suen: Document analysis and recognition by computers. In: C.H. Chen, L.F. Lau, P.S. P. Wang (eds.) Handbook of pattern recognition and computer vision (2nd edn.), World Scientific, Singapore, pp. 579–612, 1998
11. Y. Belaid, A. Belaid, E. Turolla: Item searching in forms: application to French tax forms. Proc. 3rd Int. Conf. on Document Analysis and Recognition, pp. 744–747, Montreal, Canada, 1995
12. K. Yoshikawa, Y. Adachi, M. Yoshimura: Extracting the signature from traveler's checks. Proc. 6th Int. Workshop on Frontiers of Handwriting Recognition 6:657–666, Taejon, Korea, 1998
13. J.Y. Yoo, M.K. Kim, S.Y. Han, Y.B. Kwon: Line removal and restoration of handwritten characters on the form documents. Proc. 4th Int. Conf. on Document Analysis and Recognition, pp. 128–131, Ulm, Germany, 1997
14. V. Govindaraju, S.H. Srihari: Separating handwritten text from interfering strokes. IN: S. Impedovo, J.C. Simon (eds.) From pixels to features III - frontiers in handwriting recognition. North-Holland, Amsterdam, pp. 17–28, 1992
15. K. Koffa: Principles of gestalt psychology. Harcourt Brace, New York, 1935
16. G. Dimauro, S. Impedovo, G. Pirlo, A. Salzo: Removing underlines from handwritten text: an experimental investigation. In: A.C. Downton, S. Impedovo (eds.) Progress

in handwriting recognition. World Scientific, Singapore, pp. 497–501, 1997
17. D. Guillevic, C.Y. Suen: Cursive script recognition: a fast reader scheme. Proc. 2nd Int. Conf. on Document Analysis and Recognition 93:311–314, Tsukuba Science City, Japan, 1993
18. J.N. Said, M. Cheriet, C.Y. Suen: Dynamical morphological processing: a fast method for base line extraction. Proc. 13th Int. Conf. Pattern Recognition, pp. 8–12, Vienna, Austria, 1996
19. X. Ye, M. Cheriet, C.Y. Suen, K. Liu: Extraction of bankcheck items by mathematical morphology. Int. J. Doc. Anal. Recognition 2:53–66, 1999
20. J. Serra: Image analysis and mathematical morphology. Academic, London, 1982
21. R. Haralick, S. Sternberg, X. Zhuang: Image analysis using mathematical morphology. IEEE Trans. Patt. Anal. Mach. Intell. 9:532–550, 1987
22. P. Soille: Morphological image analysis: principles and applications. Springer, Berlin Heidelberg New York, 1999
23. P. Soille, H. Talbot: Image structure orientation using mathematical morphology. Proc. ICPR '98, Brisbane, Australia, 2:1467–1469, 1998
24. J. Illingworth, J. Kittler: A survey of the Hough transform. Comput. Vision, Graphics, Image Process. 44:87–116, 1988
25. J.C. Simon, K. Zerhoumi: Robust description of a line image. Proc. 1st Int. Conf. on Document Analysis and Recognition, Saint-Malo, France, 1:3–14, 1991
26. S. Liang, M. Ahmadi, M. Shridhar: Segmentation of handwritten interference marks using multiple directional stroke planes and reformalized morphological approach. IEEE Trans. Image Process. 6(8):1195–1202, 1997
27. X. Ye, M. Cheriet, C.Y. Suen: Model-based character extraction from complex backgrounds. Proc. 5th Int. Conf. on Document Analysis and Recognition, Bangalore, India, pp. 511–514, 1999
28. H.S. Baird: Document image defect models. Proc. IAPR Workshop on Syntactic and Structural Pattern Recognition, pp. 38–46, Murray Hill, N.J., 1990
29. T. Kaungo, R.M. Haralick, I. Phillips: Global and local document degradation models. Proc. 2nd Int. Conf. on Document Analysis and Recognition, Tuskuba, Japan, pp. 730–734, 1993
30. T. Kaungo, H.S. Baird, R.M. Haralick: Validation and estimation of document degradation models. Proc. 4th UNLV Symp. on Document Analysis and Information Retrieval, Las Vegas, Nev., pp. 217–228, 1995
31. M. Cannon, J. Hochberg, P. Kelly: Quality assessment and restoration of typewritten document images. Int. J. Doc. Anal. Recognition, 2:80–89, 1999
32. C.Y. Suen, C. Nadal, R. Legault, T.A. Mai, L. Lam: Computer recognition of unconstrained handwritten numerals. Proc. IEEE, 80(7):1162–1180, 1992
33. J. Wang, H. Yan: Mending broken handwriting with a macrostructure analysis method to improve recognition. Pattern Recognition Lett. 20:855–864, 1999
34. T.Y. Zhang, C.Y. Suen: A fast parallel algorithm for thinning digital patterns. Comm. ACM, 27(3):236–239, 1984
35. A. Rosenfeld, A.C. Kak: Digital picture processing. Academic, New York, 1982
36. D. Trier, A.K. Jain: Goal-directed evaluation of binarization methods. IEEE Trans. Pattern Anal. Mach. Intell. 17:1191–1201, 1995
37. P. Soille, E. Breen, R. Jones: Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. IEEE Trans. Pattern Anal. Mach. Intell. 18(5):562–567, 1996
38. S.C. Douglas: Running max/min calculation using a pruned ordered list. IEEE Trans. Signal Process. 44(11):2872–2877, 1996
39. L. Vincent: Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. IEEE Trans. Image Process. 2(2):176–201, 1993

Xiangyun Ye received her Ph.D. degree in Communication and Electronic Systems from Shanghai Jiao Tong University in 1997, and M.E. and B.E. degrees in Biomedical Engineering from Southeast University in 1994 and 1991, respectively. Since October 1997, Dr. Ye has been with CENPARMI at Concordia University and LIVIA at École de Technologie Supérieure de l'Université du Québec, as a post-doctoral fellow. Her research interests include image processing, pattern recognition, character recognition, and document analysis.

Mohamed Cheriet received his B. Eng. degree in computer science from Université des Sciences et de Technologie d'Alger (Algiers) in 1984, and received his M.Sc. and Ph.D. degrees, also in computer science, from University of Pierre et Marie Curie (Paris VI) in 1985 and 1988, respectively. Dr. Cheriet was appointed Assistant Professor in 1992, Associate Professor in 1995, and Full Professor in 1998 in the Department of Automation Engineering, École de technologie supérieure (ETS) of University of Quebec, in Montreal. Currently, he is the Director of LIVIA, the Laboratory for Imagery, Vision and Artificial Intelligence at ETS, and an active member of CENPARMI, the Centre for Pattern Recognition and Machine Intelligence. Prof. Cheriet's research focuses on image processing, pattern recognition, character recognition, text processing, documents analysis and recognition, and perception. He has published more than 90 technical papers in the field. He was a guest Editor of the International Journal of Pattern Recognition and Artificial Intelligence and the Machine, Perception, and Artificial Intelligence series books, published by World Scientific. He was the Co-Chair of the 11th and the 13th Vision Interface Conferences held, respectively, in Vancouver, in 1998 and in Montreal, in 2000. He is currently the General Co-Chair of the 8th International Workshop on Frontiers on Handwriting Recognition, to be held in Niagara-on-the-Lake in 2002. Dr. Cheriet is a senior member of IEEE.

Ching Y. Suen Ching Y. Suen received an M.Sc.(Eng.) degree from the University of Hong Kong and a Ph.D. degree from the University of British Columbia, Canada. In 1972, he joined the Department of Computer Science of Concordia University where he became Professor in 1979 and served as Chairman from 1980 to 1984, and as Associate Dean for Research of the Faculty of Engineering and Computer Science from 1993 to 1997. Currently, he is the distinguished Chair of Artificial Intelligence and Pattern Recognition, and Director of CENPARMI, the Centre for Pattern Recognition and Machine Intelligence. Prof. Suen is the author/editor of 11 books and more than 300 papers on subjects ranging from computer vision and handwriting recognition, to expert systems and computational linguistics. He is the founder and Editor-in-Chief of a journal and an Associate Editor of several journals related to pattern recognition. A Fellow of the IEEE, IAPR, and the Academy of Sciences of the Royal Society of Canada, he has served several professional societies as President, Vice-President, or Governor. He is also the founder and chair of several conference series, including ICDAR, IWFHR, and VI, and the General Chair of the International Conference on Pattern Recognition to be held in Quebec City in 2002. Dr. Suen is the recipient of several awards, including the ITAC/NSERC Award in 1992 and the Concordia "Research Fellow" award in 1998.