

# Machine Printed Text and Handwriting Identification in Noisy Document Images

Yefeng Zheng, *Student Member, IEEE*, Huiping Li, *Member, IEEE*, and David Doermann, *Member, IEEE*

**Abstract**—In this paper, we address the problem of the identification of text in noisy document images. We are especially focused on segmenting and identifying between handwriting and machine printed text because: 1) Handwriting in a document often indicates corrections, additions, or other supplemental information that should be treated differently from the main content and 2) the segmentation and recognition techniques requested for machine printed and handwritten text are significantly different. A novel aspect of our approach is that we treat noise as a separate class and model noise based on selected features. Trained Fisher classifiers are used to identify machine printed text and handwriting from noise and we further exploit context to refine the classification. A Markov Random Field-based (MRF) approach is used to model the geometrical structure of the printed text, handwriting, and noise to rectify misclassifications. Experimental results show that our approach is robust and can significantly improve page segmentation in noisy document collections.

**Index Terms**—Text identification, handwriting identification, Markov random field, postprocessing, noisy document image enhancement, document analysis.

## 1 INTRODUCTION

DOCUMENTS are the results of a set of physical processes and conditions and the resulting document can be viewed as consisting of layers (letterhead, content, signatures, annotations, noise, etc., in the case of business correspondence, for example). Document analysis reverses these processes to segment a document into layers with different physical and semantic properties. After decades of research, automatic document analysis has advanced to a point where text segmentation and recognition can be viewed as a solved problem in clean, well-constrained documents. However, the performance degrades quickly when a small amount of noise is introduced. For example, a typical bottom-up page segmentation method starts from the extraction of connected components [1], [2]. Based on spatial proximity and size, connected components are then merged into text lines and zones. A classification process is then used to identify zone types (text, tables, images, etc.). These algorithms work well on clean documents where zones with different properties can be easily separated. However, they often fail on noisy documents where noise mixes with and/or is spatially close to content regions. For example, Figs. 1a and 1b show segmentation results for an extremely noisy document when we use the Docstrum algorithm [2] and ScanSoft SDK [3]. Text and noise are erroneously segmented into the same zones by both algorithms.

In this paper, we present a novel approach to identifying text in extremely noisy documents. Instead of simple noise filtering, as used in other work [1], [2], we treat noise as a

distinguished class and model it based on selected features. We further identify handwriting from machine printed text since: 1) Handwriting in a document often indicates corrections, additions, or other supplemental information that should be treated differently from the main content and 2) segmentation and recognition techniques for machine printed text and handwriting are significantly different. Based on these considerations, we treat the problem as a three-class (machine printed text, handwriting, and noise) identification problem.

In practice, misclassification often happens in an overlapping feature space. This is especially true for handwriting and noise. To deal with this problem, we exploit contextual information in postprocessing and refine the classification. Contextual information is very useful for improving classification accuracy. It is widely used in many OCR systems and its effectiveness has been demonstrated in previous work [4], [5]. The key is to model the statistical dependency among neighboring components. The output of an OCR system is a text stream which is one-dimensional. Therefore, an N-gram language model, based on an  $N$ th order 1D Markov chain, is effective for modeling the context. With assistance from a dictionary, the N-gram approach can correct most recognition errors. Images, however, are two-dimensional. Generally, 2D signals are not causal and it is much harder to model the dependency among neighboring components in an image. Among the image models studied so far, Markov Random Fields (MRF) have been widely studied and successfully used in many applications. MRFs are suitable for image analysis because the local statistical dependency of an image can be well-modeled by Markov properties. MRFs can incorporate a priori contextual information or constraints in a quantitative way. The MRF model has been extensively used in various image analysis applications such as texture synthesis and segmentation, edge detection, and image restoration [6], [7]. In this paper, we use MRFs to model the dependency of segmented neighboring blocks. As postprocessing, MRFs can further improve classification accuracy.

• The authors are with the Language and Media Processing Laboratory, Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742-3275.

E-mail: {zhengyf, huiping, doermann}@cfar.umd.edu.

Manuscript received 12 Dec. 2002; revised 28 Aug. 2003; accepted 7 Oct. 2003.

Recommended for acceptance by T.Tan.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117937.

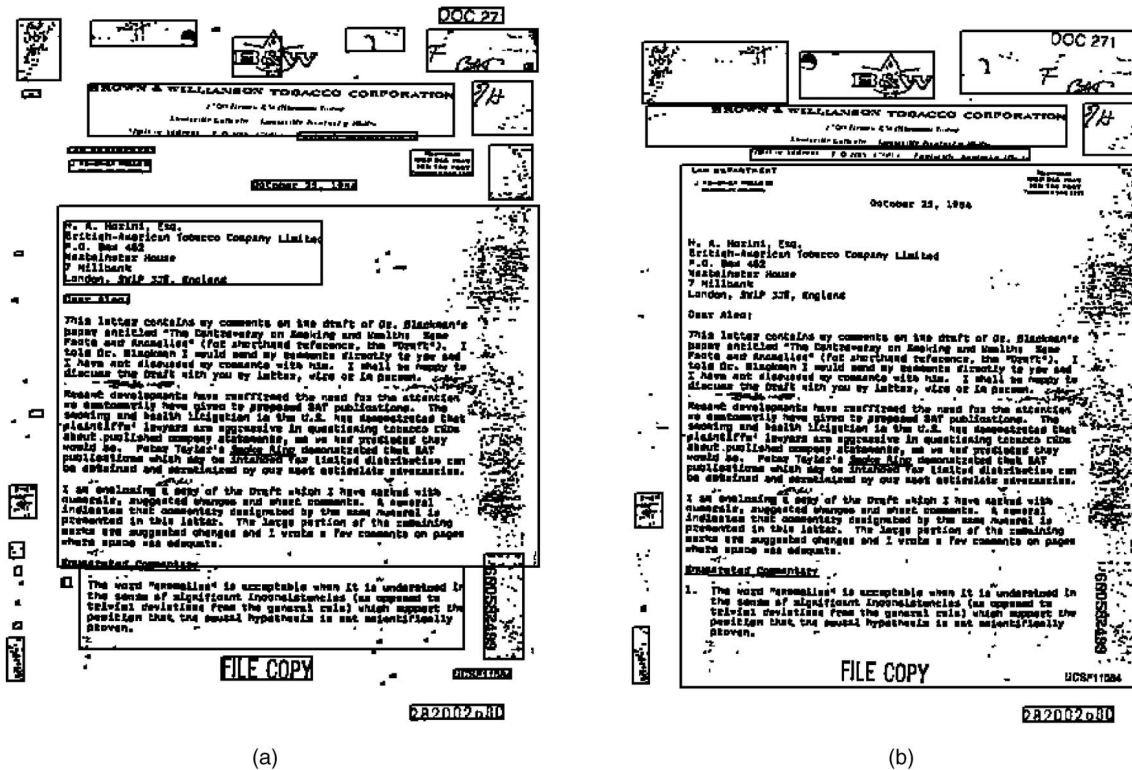


Fig. 1. Page segmentation results for an extremely noisy document using the Docstrum algorithm and ScanSoft SDK. Noise is segmented into text zones erroneously in both cases. (a) Docstrum and (b) ScanSoft.

The documents we are processing are extremely noisy with machine printed text, handwriting, and noise mixed together. We first extract the connected components and merge them at the word level based on spatial proximity. We then extract several categories of features and use trained Fisher classifiers to classify each word into machine printed text, handwriting, or noise. Finally, contextual information is incorporated into MRF models to refine the classification results further.

The rest of the paper is organized as follows: Section 2 is a literature survey of related work, followed by a detailed description of our classification method in Section 3. MRF-based postprocessing is presented in Section 4 and experimental results are presented in Section 5. The paper concludes with a brief summary and a discussion of future work.

## 2 RELATED WORK

The research presented in this paper is related to previous work on page segmentation, zone classification, handwriting identification, and document enhancement.

### 2.1 Page Segmentation

Previous work on page segmentation can be broadly divided into three categories: bottom-up [1], [2], top-down [8], and hybrid [9]. In a typical bottom-up approach such as the Docstrum algorithm proposed by O'Gorman [2], connected components are extracted first and then merged into words, lines, zones, and columns hierarchically based on size and spatial proximity. Bottom-up methods can handle documents with complex layouts. However, they are time consuming and sensitive to noise.

A typical top-down method, such as the X-Y cuts proposed by Nagy et al. [8], starts from the whole document and splits it recursively into columns, zones, lines, words, and characters. Top-down methods are effective for documents with regular layouts, but fail when the documents have a non-Manhattan structure.

Another problem with X-Y cuts is that the global parameters for optimal segmentation are often difficult to find if prior knowledge is not available. Sylwester and Seth proposed a hybrid method which starts from the top [9]. First, they oversegment a document into small zones using the X-Y cut algorithm. Then, they use the bottom-up method which groups oversegmented small zones with the same properties into a single zone.

All of the above methods are based on the analysis of foreground (black pixels). As an alternative, white stream methods based on the analysis of background (white pixels) are presented in [10], [11]. In these methods, rectangles covering white gaps (white pixels) between foreground are extracted. Foreground regions surrounded by these white rectangles are extracted as zones. A more comprehensive survey is presented in [12].

### 2.2 Zone Classification

Zone classification labels the content of each segmented zone as one of a set of predefined types [1], [11], [13], such as text, images, graphics, and tables. Pavlidis and Zhou used correlations of horizontal scan lines as features to distinguish text and diagrams from half-tone images. The black pixel density is used to further distinguish diagrams from text [11]. Wang et al. used 69 features, such as run length mean and variance, spatial mean and variance, fraction of the total number of black pixels in the zone,

width ratio of the zone, and number of text glyphs in the zone, to classify each zone into nine classes. They did experiments on ground-truthed zones of the UW III database and achieved an accuracy as high as 98.52 percent [13]. Jain and Yu directly performed classification on the generalized lines (GTLs) extracted using a bottom-up approach [1]. If the height of a GTL is less than a threshold and the connected components in it are horizontally aligned, it is classified as a text line. Text lines and nontext lines are merged into text regions and nontext regions, respectively. They further classify nontext regions into images, tables, and drawings. This works well for long text lines, but may fail when the text lines are short.

Some other approaches treat text, images, and figures as different textures, and use trained classifiers to segment and identify them [14], [15], [16]. They often work directly on gray-scale images and need classification of each pixel. To reduce the computation complexity, multiresolution techniques are often used.

### 2.3 Handwriting Identification

Some work has been done on handwriting/machine printed text identification. The classification is typically performed at the text line [17], [18], [19], [20], word [21], or character level [22], [23]. At the line level, machine printed text lines are typically arranged regularly with a straight baseline, while handwritten text lines are irregular with a varying baseline. Srihari et al. implemented a text line-based approach using this characteristic and achieved a classification accuracy of 95 percent [20]. One advantage of this approach is that it can be used in different scripts (Chinese, English, etc.) with little or no modification. Guo and Ma proposed an approach based on the vertical projection profile of the segmented words [21]. They used a Hidden Markov Model (HMM) as the classifier and achieved a classification accuracy of 97.2 percent. Although, at the character level, less information is available, humans can still identify the handwritten and machine printed characters easily, inspiring researchers to pursue classification at the character level. Kuhnke et al. proposed a neural network-based approach with straightness and symmetry as features [22]. Zheng et al. used run-length histogram features to identify handwritten and printed Chinese characters and achieved promising results [23]. In previous work, we implemented a handwriting identification method based on several categories of features and a trained Fisher classifier [24]. However, the problems introduced by noise are not addressed.

### 2.4 Document Enhancement

There are two types of degradation in document images: 1) physical degradation of the hardcopy documents during creation and/or storage and 2) degradation introduced by digitization. If severe enough, either of them can reduce the performance of a document analysis system significantly. Several document degradation models [25], [26], [27], methods for document quality assessment [28], [29], and document enhancement algorithms [30], [31], [32] have been presented in previous work. One common enhancement approach is window-based morphological filtering [30], [31], [32]. Morphological filtering performs a look up table procedure to determinate an output of ON (black pixel) or OFF (white pixel) for each entry of the table, based on a windowed observation of its neighbors. These algorithms can

be further categorized as manually designed, semimanually designed, or automatically trained approaches. The kFill algorithm, proposed by O’Gorman [32], is a manually designed approach and has been used by several other researchers [28], [33]. Experiments show it is effective for removing salt-and-pepper noise. Liang and Haralick proposed a semimanually designed approach with a  $3 \times 3$  window size [34]. They manually determine some entries to output ON or OFF based on a priori observations. The remaining entries are trained to select the optimal output. It is difficult to manually design a filter with a large window size and success depends on experience. If both ideal and degraded images are available, optimal filters can be designed by training [31]. After registering the ideal and degraded images at the pixel level, an optimal look-up table, based on observation of the outputs of each specific windowed context, can be designed. However, it is difficult to train, store, and retrieve the look-up table when the window size is large. This approach requires both the original and the corresponding degraded images for training. Loce and Dougherty used artificially degraded images generated by models for training [31], while Kanungo et al. proposed methods for validation and parameter estimation of degradation models [35], [36], [37]. Though the uniformity and sensitivity of his approach has been tested by other researchers [27], [38], no degradation model has been declared to pass the validation. Another problem with morphological approaches is the small window sizes. The most commonly used window size is no larger than  $5 \times 5$ , which is too small to contain enough information for enhancement.

Ideally, image quality should be estimated first so the appropriate enhancement algorithms can be applied automatically. Cannon et al. proposed a document quality assessment algorithm based on five factors: small speckle, white speckle, touching characters, broken characters, and font size [28]. They used a linear classifier to select the best one out of four enhancement algorithms and reduced the OCR error rate from 20.27 percent to 12.60 percent on their database. Li and Doermann proposed an approach for quality estimation of color video text which classifies the video text quality into six levels [29].

A majority of the above approaches are focused on improving OCR accuracy in noisy documents. As shown in Fig. 1, degradation will not only deteriorate OCR performance, but other document processing tasks, such as page segmentation, as well. Little work has been done in this area. The difference between our approach and previous work is that we perform classification to identify noise and exploit contextual information of neighboring blocks as a postprocessing to refine the identification. Experiments show that our noise removal algorithm can increase page segmentation accuracy significantly.

## 3 TEXT IDENTIFICATION

In this section, we present our text (machine printed or handwritten) extraction and classification method.

### 3.1 Pattern Unit

Special consideration must be given to the size of the region being segmented before we can perform any classification. We call the smallest unit for classification a *pattern unit*. If the unit is too small, the information contained in it may not

TABLE 1  
Features Used for Machine Printed Text/Handwriting/Noise Classification

Feature set	Feature description	# of features	# of features selected
Structural	Region size, connected components	18	9
Gabor filter	Stroke orientation	16	4
Run-length histogram	Stroke length	20	5
Crossing count histogram	Stroke complexity	10	6
Bi-level co-occurrence	Texture	16	2
2×2 gram	Texture	60	5
Total		140	31

be sufficient for classification; if it is too large, however, different types of components may be mixed in the same region. In previous work, we conducted a performance evaluation for the classification accuracy of machine printed text and handwriting at the character, word, and zone levels and showed that a reliable classification can be achieved at the word level [24]. We therefore segment images at the word level and then perform classification. Since noise has no concept of *word*, we use the terminology *block* and *word* interchangeably in the following presentation.

We first extract connected components and then merge them into words based on geometric proximity and size. Those extremely large word blocks or blocks with very large or small aspect ratios are filtered out. However, noise with size similar to text cannot be filtered out. Our focus is to distinguish text from this type of noise.

### 3.2 Feature Extraction

Several sets of features are extracted for classification. The descriptions and sizes of the feature sets are listed in Table 1. Machine printed text, handwriting, and noise have different visual appearances and physical structures. Structural features are extracted to reflect these differences. Gabor filter features and run-length histogram features can capture the difference in stroke orientation and stroke length between handwriting and printed text. Compared with text, noise blocks often have simple stroke complexity. Therefore, crossing count histogram features are exploited to model such differences. We further take regions of machine printed text, handwriting, and noise blocks as different textures. Two sets of bilevel texture features (bilevel co-occurrence features and bilevel 2 × 2 gram features) are used for classification. In the following sections, we present these features in detail.

#### 3.2.1 Structural Features

We extract two sets of structural features. The first set includes features related to the physical sizes of the blocks such as density of black pixels, width, height, aspect ratio, and area. Suppose the image of the block is  $I(x, y)$ ,  $0 \leq x < w$ ,  $0 \leq y < h$ , and  $w, h$  are its width and height, respectively. Each pixel in the block has two values: 0 representing background (a white pixel) and 1 representing content (a black pixel). Then, the density of the black pixels  $d$  is

$$d = \frac{\sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I(x, y)}{w \times h}. \quad (1)$$

The sizes of machine printed words are more consistent than those of handwriting and noise on the same page. However, machine printed words on different pages may vary significantly. Therefore, we use a histogram technique to estimate the dominant font size [2] and then use the dominant font size to normalize the width ( $w$ ), height ( $h$ ), aspect ratio ( $r$ ), and area ( $a$ ) of the block.

The second set of structural features is based on the connected components inside the block, such as the mean and variance of the width ( $m_w$  and  $\sigma_w$ ), height ( $m_h$  and  $\sigma_h$ ), aspect ratio ( $m_r$  and  $\sigma_r$ ), and area ( $m_a$  and  $\sigma_a$ ) of connected components. The sizes of connected components inside a machine printed word are more consistent, leading to smaller  $\sigma_w$  and  $\sigma_h$ . For a handwritten word or noise block, the bounding boxes of the connected components tend to overlap with each other, as shown in Fig. 2a. For machine printed English words, however, each character forms a connected component not overlapping with others. The overlapping area (the sum of the areas of the gray rectangles in Fig. 2a) normalized by the total area of the block is calculated as a feature. Another feature we use is the variance of the vertical projection. In a machine printed text block, the vertical projection profile has obvious valleys and peaks since neighboring characters do not touch each other. However, for a handwritten word or noise block, the vertical projections are much smoother, resulting in smaller variance.

#### 3.2.2 Gabor Filter Features

Gabor filters can represent signals in both the frequency and time domains with minimum uncertainty [39] and have been widely used for texture analysis and segmentation [15]. Researchers found that they match the mammalian visual system very well, which provides further evidence that we can use it in our classification tasks. In the spatial and frequency domains, the two-dimensional Gabor filter is defined as

$$g(x, y) = \exp \left\{ -\pi \left[ \frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2} \right] \right\} \times \cos \{ 2\pi(u_0x + v_0y) \} \quad (2)$$

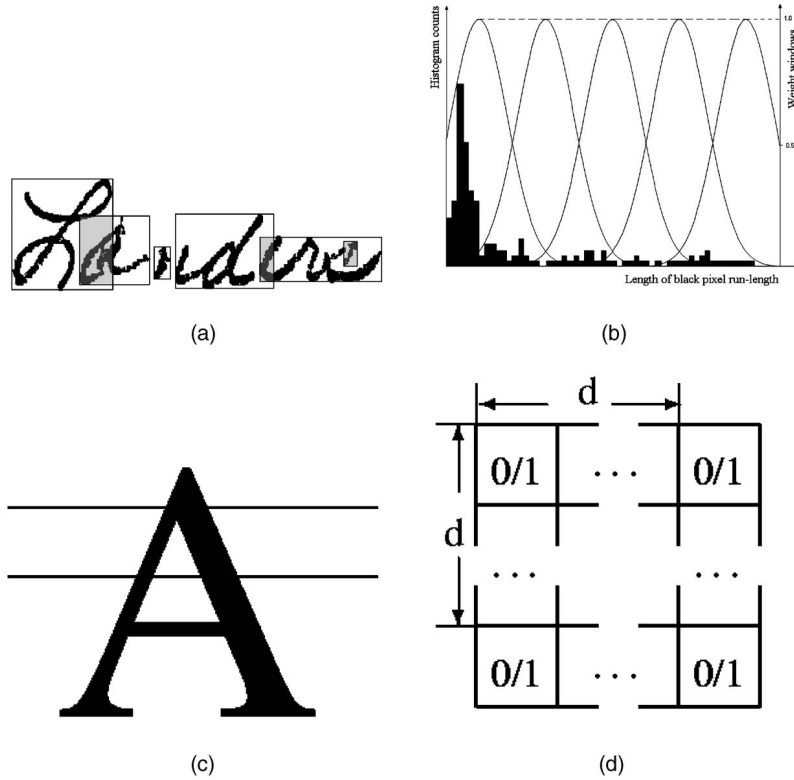


Fig. 2. Illustration of feature extraction. (a) The overlap area of the connected components inside a pattern unit is extracted as a structural feature. (b) Run-length histogram features. (c) Crossing count features. The crossing counts of the top and bottom horizontal scan lines are 1 and 2, respectively. (d) Bilevel  $2 \times 2$  gram features.

$$G(u, v) = 2\pi\sigma_x\sigma_y(\exp\{-\pi[(u' - u'_0)^2\sigma_x^2 + (v' - v'_0)^2\sigma_y^2]\} + \exp\{-\pi[(u' + u'_0)^2\sigma_x^2 + (v' + v'_0)^2\sigma_y^2]\}), \quad (3)$$

where  $x' = -x \sin \theta + y \cos \theta$ ,  $y' = -x \cos \theta - y \sin \theta$ ,  $u' = u \sin \theta - v \cos \theta$ ,  $v' = -u \cos \theta - v \sin \theta$ ,  $u'_0 = -u_0 \sin \theta + v_0 \cos \theta$ ,  $v'_0 = -u_0 \cos \theta - v_0 \sin \theta$ ,  $u_0 = f \cos \theta$ , and  $v_0 = f \sin \theta$ . Here,  $f$  and  $\theta$  are two parameters, representing the central frequency and orientation of the Gabor filter.

The variances of the filtered images are taken as features. In our experiments, 16 Gabor filters with different orientations  $\theta_k = k \times 180/N$ ,  $k = 1, 2, \dots, 16$ , are used, which generate 16 features.

### 3.2.3 Run-length Histogram Features

Run-length histogram features are proposed in [23] for machine printed/ handwritten Chinese character classification. These features are used in our case to capture the difference between the stroke lengths of machine printed text, handwriting, and noise blocks. First, black pixel run-lengths in four directions, including horizontal, vertical, major diagonal, and minor diagonal, are extracted. We then calculate four histograms of run-lengths for these four directions, as shown in Fig. 2b. To get scale-invariant features, we normalize the histograms. Suppose  $C_k$ ,  $k = 1, 2, \dots, N$ , is the number of runs with length  $k$ , and  $N$  is the maximal length of all possible runs, then the normalized histogram  $C'_k$  is

$$C'_k = \frac{C_k}{\sum_{i=1}^N C_i}. \quad (4)$$

We then divide the histogram into five bins with equal width and use five Gaussian-shaped weight windows to get the final features (Fig. 2b). Taking the horizontal run-length histogram as an example, the run-length histogram feature  $Rh_i$  is calculated as

$$Rh_i = \sum_{k=1}^w G(k; u_i, \sigma) C'_k, \quad i = 1, 2, 3, 4, 5, \quad (5)$$

where  $w$  is the width of the block (the maximal length of all possible horizontal run-lengths) and  $G(k; u_i, \sigma)$  is a Gaussian-shaped function:

$$G(k; u_i, \sigma) = \exp\left\{-\frac{(k - u_i)^2}{2\sigma^2}\right\}. \quad (6)$$

As shown in Fig. 2b,  $\sigma$  is chosen so the weight on each bin border is 0.5. Another alternative is to use rectangular windows without overlap between neighboring bins. Experiments show that the extracted features with Gaussian weighted windows are more robust. Five features are extracted in each direction, leading to 20 features.

### 3.2.4 Crossing Count Histogram Features

A crossing count is the number of times the pixel value changes from 0 (white pixel) to 1 (black pixel) along a horizontal or vertical raster scan line. As shown in Fig. 2c, the crossing counts of the top and bottom horizontal scan lines are 1 and 2, respectively. Crossing counts can be used to measure stroke complexity [24], [40]. In our approach, first the crossing count for each horizontal and vertical scan

line is calculated. Similarly, we get two histograms for the horizontal and vertical crossing counts, respectively. The same technique (as in extracting the run-length histogram features) is exploited to get the final features from the histograms. A total of 10 features are extracted.

### 3.2.5 Bilevel Co-Occurrence Features

A co-occurrence count is the number of times a given pair of pixels occurs at a fixed distance and orientation [41]. In the case of binary images, the possible co-occurrence pairs are white-white, black-white, white-black, and black-black. In our case, we are concerned primarily with the foreground. Since the white background region often accounts for up to 80 percent of a document page, the occurrence frequency of white-white or white-black pixel pairs will always be much higher than that of black-black pairs. The black-black pairs carry most of the information. To eliminate the redundancy and reduce the effects of overemphasizing the background, we consider only black-black pairs. Four different orientations (horizontal, vertical, major diagonal, and minor diagonal) and four distance levels (1, 2, 4, and 8 pixels) are used for classification (16 features total). The horizontal co-occurrence count  $C_h(d)$ , for example, is defined as

$$C_h(d) = \sum_x \sum_y I(x, y)I(x + d, y), d = 1, 2, 4, 8. \quad (7)$$

$I(x, y) = 0$  for white pixels; therefore, only black-black pixel pairs contribute. For a fixed distance  $d$ , we normalize the occurrence by dividing by the sum of the occurrences in all four directions.

### 3.2.6 Bilevel $2 \times 2$ Gram Features

The  $N \times M$  grams were first introduced in the context of image classification and retrieval [42]. An  $N \times M$  gram extends the one-dimensional co-occurrence feature to the two-dimensional case. We only consider  $2 \times 2$  grams, which count the numbers of occurrences of the patterns shown in Fig. 2d. The cells labeled 0/1 should take specific values and the values of other cells are irrelevant. Therefore, there are  $2^4 = 16$  patterns for each distance  $d$ . Like the co-occurrence features, the all white patterns are removed to reduce overemphasis on the background. For a fixed distance, the occurrences are normalized by dividing by the sum of all occurrences. Four distances (1, 2, 4, and 8 pixels) are chosen, generating  $4 \times 15 = 60$  features.

## 3.3 Feature Selection

There are two purposes for feature selection. First, reducing the computation needed for feature extraction and classification. As shown in Table 1, we extract a total of 140 features from the segmented blocks. Though these features are designed to distinguish between different types of blocks, some features may contain more information than others. Using only a small set of the most powerful features reduces the time for feature extraction and classification. The second purpose is to alleviate the curse of dimensionality. When the number of training samples is limited, using a large feature set may decrease the generality of a classifier [43]. The larger the feature set, the more training samples are needed. Therefore, we perform feature selection before feeding the features to the classifier.

We use a forward search algorithm to perform feature selection [44]. We first divide the whole feature set  $\mathcal{F}$  into a

currently selected feature set  $\mathcal{F}_s$  and an unselected feature set  $\mathcal{F}_n$  which satisfy

$$\mathcal{F}_s \cap \mathcal{F}_n = \Phi \quad (8)$$

$$\mathcal{F}_s \cup \mathcal{F}_n = \mathcal{F}. \quad (9)$$

The selection procedure can then be described as

1. Set  $\mathcal{F}_s = \Phi$ , and  $\mathcal{F}_n = \mathcal{F}$ .
2. Label all features in  $\mathcal{F}_n$  as untested.
3. Select one untested feature  $f \in \mathcal{F}_n$  and label it as tested.
4. Put  $f$  and  $\mathcal{F}_s$  together and generate a temporary selected feature set  $\mathcal{F}_s^f$ .
5. Estimate the classification accuracy with feature set  $\mathcal{F}_s^f$  using a 1-NN classifier and leave-one-out cross validation technique. The basic idea is that at each iteration only one sample is used for testing, while the others have been used for training. We repeat this process until all samples have been used as testing samples once. The average accuracy for all iterations is taken as the estimated accuracy for the current feature set. The leave-one-out cross validation technique can estimate the accuracy of a classifier with small variation [43].
6. If there are untested features in  $\mathcal{F}_n$ , go to Step 3.
7. Find a feature  $\hat{f} \in \mathcal{F}_n$ , such that the corresponding temporary feature set  $\mathcal{F}_s^{\hat{f}}$  has the highest classification accuracy:

$$\hat{f} = \arg \max_{f \in \mathcal{F}_n} Accuracy(\mathcal{F}_s^f), \quad (10)$$

then move  $\hat{f}$  from  $\mathcal{F}_n$  to  $\mathcal{F}_s$ .

8. If  $\mathcal{F}_n \neq \Phi$ , go to Step 2; otherwise exit.

We use LNKnet pattern classification software to conduct our feature selection experiments [45]. LNKnet provides several classifiers, such as likelihood classifiers, k-NN classifiers, and Neural Network classifiers, and several feature selection algorithms such as forward search, backward search, and forward and backward search. Feature selection can be an extremely expensive task. Considering the large number of feature sets to evaluate and the number of classifiers to train, the lightweight forward feature selection algorithm and 1-NN classifier, which does not need training, are used in our feature selection experiment.

We collected about 1,500 blocks for each class. As shown in Fig. 3a, when the number of selected features increases, the error rate decreases sharply at first. The trend reverses at some point. The best classification is achieved when only 31 features are selected, with an error rate of 5.7 percent. When all features are used, the error rate increases to 9.2 percent due to the limited number of training samples and large feature set. The last column in Table 1 lists the number of features selected in each set. It shows that texture features, such as bilevel co-occurrence and  $2 \times 2$  grams, are less discriminating than other feature sets, mainly due to the small region size. Only 1/8 of the bilevel co-occurrence features and 1/12 of the  $2 \times 2$  gram features are selected. Crossing count histogram features and structural features are very effective, with more than half of the original features in both sets selected in the final feature set.

Principal Component Analysis (PCA) is another technique for reducing feature dimension [43]. To extract the first

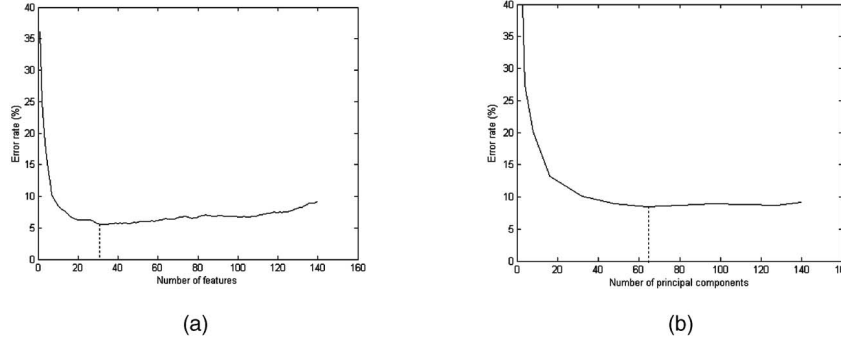


Fig. 3. Feature analysis. (a) Feature selection: The best classification result is achieved when 31 features are selected. (b) PCA: The best classification result is achieved when 64 principal components are used.

$n$  principal components, we need to search a subspace of dimension  $n$  with basis  $\underline{w}$ . Suppose the mean is already removed from the feature vector  $\underline{X}$  and let the projection of  $\underline{X}$  onto this subspace be  $\hat{\underline{X}}$

$$\hat{\underline{X}} = (w_1^T \underline{X})w_1 + (w_2^T \underline{X})w_2 + \cdots + (w_n^T \underline{X})w_n. \quad (11)$$

PCA finds the optimal subspace  $\hat{\underline{w}}$  such that the energy contained in  $\hat{\underline{X}}$  is maximized:

$$\begin{aligned} \hat{\underline{w}} = \arg \max_{w_1, \dots, w_n} \sum_{i=1}^n \text{Var}[\hat{\underline{X}}_i] \\ \text{s.t. } w_i^T w_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases} \end{aligned} \quad (12)$$

The optimal basis is the first  $n$  eigenvectors of the covariance matrix of  $\underline{X}$ , corresponding to the first  $n$  largest eigenvalues [43]. The first  $n$  principal components are  $P_i = w_i^T \underline{X}$ ,  $i = 1, \dots, n$ . The idea of PCA is to concentrate the energy into the first several principal components. Assuming the classification information is contained in the energy, the first several principal components are more powerful than the remaining components. Furthermore, PCA analysis can remove the correlation among features. As in the feature selection experiment, the 1-NN classifier and the leave-one-out technique are used to estimate the classification accuracy. Fig. 3b shows the classification error rate versus the number of principal components used. As in feature selection, the error rate drops quickly at first until 16 principal components are added. The minimal error rate, 8.5 percent, is achieved when 64 principal components are used. Compared with the minimum error rate of 5.7 percent achieved by the feature selection technique, PCA is not as powerful as feature selection in this problem. Furthermore, to perform PCA, all features must be extracted first. However, for feature selection, we only need to extract the desired features, which would increase the feature extraction speed. Therefore, in the following, we do classification on the 31 selected features.

### 3.4 Classification

Compared with the Neural Network (NN) and the Support Vector Machine (SVM), the Fisher classifier is easier to train, faster for classification, needs fewer training samples, and does not suffer from overtraining problems. According to the comparison experiment in Section 5.2, the SVM classifier

performs slightly better than the Fisher classifier, but the latter is much faster; we therefore use it for classification.

For a feature vector  $\underline{X}$ , the Fisher classifier projects  $\underline{X}$  onto one dimension  $Y$  in direction  $\underline{W}$

$$Y = \underline{W}^T \underline{X}. \quad (13)$$

The Fisher criterion finds the optimal projection direction  $\underline{W}_o$  by maximizing the ratio of the between-class scatter to the within-class scatter, which benefits the classification. Let  $\underline{S}_w$  and  $\underline{S}_b$  be the within and between-class scatter matrices respectively,

$$\underline{S}_w = \sum_{k=1}^K \sum_{x \in \text{class } k} (x - \underline{u}_k)(x - \underline{u}_k)^T \quad (14)$$

$$\underline{S}_b = \sum_{k=1}^K (\underline{u}_k - \underline{u}_0)(\underline{u}_k - \underline{u}_0)^T \quad (15)$$

$$\underline{u}_0 = \frac{1}{K} \sum_{k=1}^K \underline{u}_k, \quad (16)$$

where  $\underline{u}_k$  is the mean vector of the  $k$ th class,  $\underline{u}_0$  is the global mean vector, and  $K$  is the number of classes. The optimal projection direction is the eigenvector of  $\underline{S}_w^{-1} \underline{S}_b$ , corresponding to its largest eigenvalue [43]. For a two-class classification problem, we do not need to calculate the eigenvectors of  $\underline{S}_w^{-1} \underline{S}_b$ . It is shown that the optimal projection direction is

$$\underline{W}_o = \underline{S}_w^{-1} (\underline{u}_1 - \underline{u}_2). \quad (17)$$

Let  $Y_1$  and  $Y_2$  be the projections of two classes and let  $E[Y_1]$  and  $E[Y_2]$  be the means of  $Y_1$  and  $Y_2$ , respectively. Suppose  $E[Y_1] > E[Y_2]$ , then the decision can be made as

$$C(\underline{X}) = \begin{cases} \text{class 1} & \text{If } Y > (E[Y_1] + E[Y_2])/2 \\ \text{class 2} & \text{Otherwise.} \end{cases} \quad (18)$$

It is known that, if the feature vector  $\underline{X}$  is jointly Gaussian distributed and the two classes have the same covariance matrices, then the Fisher classifier is optimal in a minimum classification error sense [43].

The Fisher classifier is often used for two-class classification problems. Although it can be extended to multiclass classification (three classes in our case), the classification accuracy decreases due to the overlap between neighboring classes. Therefore, we use three Fisher classifiers, each

optimized for a two-class classification problem (machine printed text/handwriting, machine printed text/noise, and handwriting/noise). Each classifier outputs a confidence in the classification and the final decision is made by fusing the outputs of all three classifiers.

### 3.5 Classification Confidence

In a Fisher classifier, the feature vector is projected onto an axis on which the ratio of between-class scatter to within-class scatter is maximized. According to the central limit theorem [46], the distribution of the projection can be approximated by a Gaussian distribution, if no feature has dominant variance over the others, as follows:

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{y-m}{\sigma}\right)^2\right], \quad (19)$$

where  $f_Y(y)$  is the probability density function of the projection. The parameters  $m$  and  $\sigma$  can be estimated from training samples. The classification confidence  $C_{i,j}$  of class  $i$  using classifier  $j$  is defined as

$$C_{i,j} = \begin{cases} \frac{f_Y(y|X \in \text{class } i)}{f_Y(y|X \in \text{class } i) + f_Y(y|X \in \text{another class})} & \text{If } i \text{ is applicable for classifier } j. \\ 0 & \text{Otherwise,} \end{cases} \quad (20)$$

where  $i$  is the class label and  $j$  represents the trained classifiers. If a classifier is trained to classes 1 and 2, its output is not applicable to estimating the classification confidence of class 3. Therefore,  $C_{3,j} = 0$ . The final classification confidence is defined as

$$C_i = \frac{1}{2} \sum_{j=1}^3 C_{i,j}. \quad (21)$$

$C_{i,j} \in [0, 1]$  for the two applicable classifiers and  $C_{i,j} = 0$  for the third classifier,  $C_i \in [0, 1]$ . However,  $C_i$  is not a good estimate of the a posteriori probability since  $\sum_{i=1}^3 C_i = 1.5$  instead of 1. We can take  $C_i$  as an estimate of a nondecreasing function of the a posteriori probability, which is a kind of generalized classification confidence [47].

Fig. 4 shows the word segmentation and classification results (with the Fisher classifier) for the whole and parts of a document image, with solid, dot, and dashed rectangles representing noise, handwriting, and printed text, respectively. We can see that most of the blocks are correctly classified. However, some blocks are misclassified due to overlap in the feature space. For example, some noise blocks are classified as handwriting in Fig. 4b and some small printed words are classified as noise in Fig. 4c. Since very little information is available in such small areas, it is very hard to get good results. In the next section, we present a method of Markov Random Field-based (MRF) postprocessing to refine the classification by incorporating contextual information.

## 4 MRF-BASED POSTPROCESSING

### 4.1 Background

Let  $X$  denote the random field defined on  $\Omega$  and let  $\Gamma$  denote the set of all possible configurations of  $X$  on  $\Omega$ .  $X$  is an MRF with respect to the neighborhood  $\eta$  if it has the following Markov property

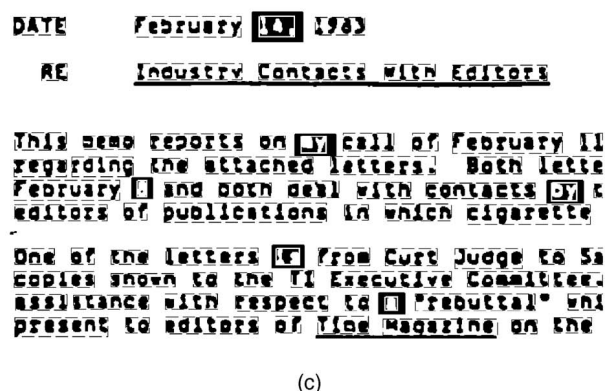
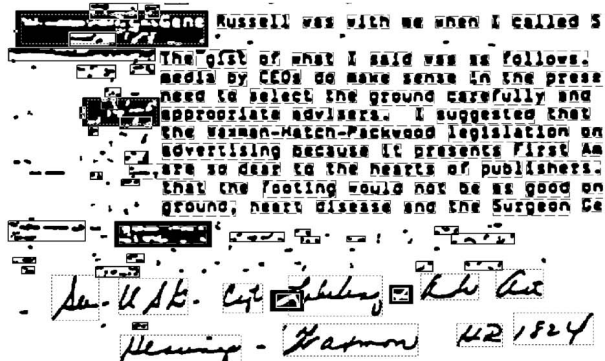
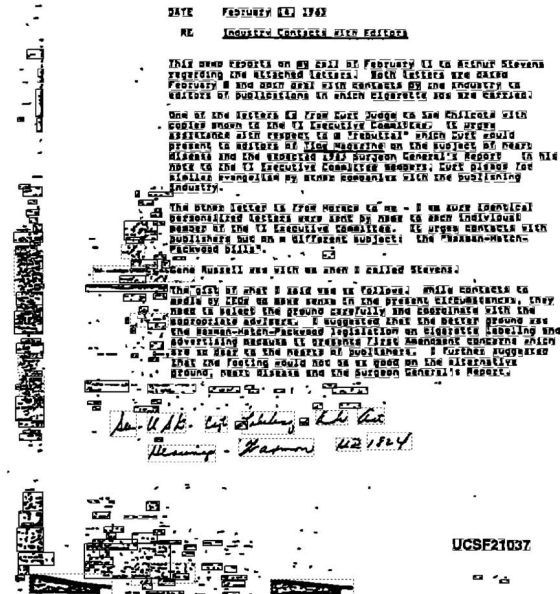


Fig. 4. Word block segmentation and classification results, with solid, dot, and dashed rectangles representing noise, handwriting, and printed text, respectively. (a) A whole document image. (b) and (c) Two parts of the image in (a). The classification errors are highlighted with reverse video in (b) and (c).

$$\Pr(X = x) > 0 \quad \text{for all } x \in \Gamma \quad (22)$$

$$P(x_s/x_r, r \in \Omega, r \neq s) = P(x_s/x_r, r \in \eta). \quad (23)$$

Compared with Markov chains, one difficulty with MRFs is that there is no chain rule for MRFs. The joint



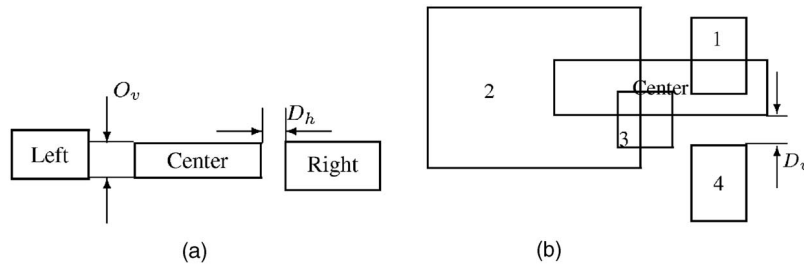


Fig. 5. Clique definition. (a)  $C_p$  for horizontally arranged machine printed words. (b)  $C_n$  for noise blocks.

probability  $P(\underline{X} = \underline{x})$  cannot be recursively written in terms of local conditional probabilities  $P(x_s/x_r, r \in \eta)$ . Therefore, it is difficult to get an optimal estimate of the MRF  $\hat{\underline{X}}$  which maximizes the posteriori probability

$$\hat{\underline{X}} = \arg \max_{\underline{X}} P(\underline{X}/\underline{Y}). \quad (24)$$

The establishment of the connection between the MRF and Gibbs distribution provides a way to optimize of the MRF. To maximize the posteriori probability of the MRF, we need to minimize the total energy of the corresponding Gibbs distribution

$$\hat{\underline{X}} = \arg \min_{\underline{X}} \sum_{c \in \mathcal{C}} V_c(\underline{X}). \quad (25)$$

Here, a *clique*  $c$  is defined as a subset of sites in which every pair of distinct sites are neighbors. The clique potential  $V_c(\underline{X})$  is the energy associated with a clique and depends on the local configuration on clique  $c$ . Therefore, the optimization problem (24) is converted to another optimization problem (25). The information about the observation  $\underline{Y}$  is contained in the clique system.

In the study of MRFs, the problems are often posed as labeling problems in which a set of labels are assigned to sites of an MRF [7]. In our problem, each block constitutes a site of an MRF. A label (as one of machine-printed text, handwriting, and noise) is assigned to each block and context information (encoded by the MRF model) is used to flip the labels so that the total energy of the corresponding Gibbs distribution is minimized. Relaxation algorithms are often used for MRF optimization [7].

## 4.2 Clique Definition

As shown in (25), the MRF is totally determined by clique  $c$  and clique potential  $V_c(\underline{X})$ . The design of the clique and its potential is crucial, but a systematic method is not yet available. In our case, machine printed text, handwriting, and noise exhibit different patterns of geometric relationships. Our definition of cliques reflects these differences.

Printed words often form horizontal (or vertical) text lines. Clique  $C_p$  is defined in Fig. 5a, which models contextual constraints on neighboring machine printed words. We first define the *connection* between word blocks  $i$  and  $j$ . As shown in Fig. 5a,  $O_v$  is the vertical overlap between two blocks and  $D_h$  is the horizontal distance between two blocks. The distance between block  $i$  and  $j$  is

$$D(i, j) = |D_h(i, j) - G_w| + |H_i - H_j| + |Ch_i - Ch_j|, \quad (26)$$

where  $D_h(i, j)$  is the horizontal distances between words  $i$  and  $j$ ,  $G_w$  is the estimated average word gap in the whole

document,  $H_i$  and  $H_j$  are the heights of blocks  $i$  and  $j$ , respectively, and  $Ch_i$  and  $Ch_j$  are the vertical centers of the two blocks. Two blocks are connected if they satisfy

1.  $O_v \geq \min(H_i, H_j)/2$ .
2.  $0 \leq D_h \leq 2G_w$ .
3.  $D(i, j) < T_p$ , where  $T_p$  is a threshold, which is not sensitive to postprocessing.

After defining the connection between two blocks, we can construct a graph in which nodes represent blocks and edges link two connected nodes. The property of an edge can be measured by the distance  $D(i, j)$  between two blocks. If a node is connected with more than one node on one side (left or right), we only keep the edge with the smallest distance. Clique  $C_p$  can be represented by nodes together with their left and right neighbors. If we cannot find neighbors on the left or/and right sides, the corresponding neighbor is set to NULL.

Noise blocks exhibit rather random patterns in geometric relationships and tend to overlap or be very close to each other. As shown in Fig. 5b, the noise block labeled "Center" is overlapped with blocks 1, 2, 3, and is very close to block 4. Clique  $C_n$  is defined primarily for noise blocks. Similarly, the distance between two blocks is defined as

$$D(i, j) = \max(D_h(i, j), D_v(i, j)), \quad (27)$$

where  $D_h(i, j) = \max(L_i, L_j) - \min(R_i, R_j)$ ,  $D_v(i, j) = \max(T_i, T_j) - \min(B_i, B_j)$ , and  $L, R, T, B$  are the left, right, top, and bottom coordinates of the corresponding blocks. If two blocks overlap in the horizontal or vertical direction, then  $D_h(i, j) < 0$  or  $D_v(i, j) < 0$ . Blocks  $i$  and  $j$  are connected if and only if  $D(i, j) < T_n$ , where  $T_n$  is a threshold. If  $T_n$  is too big, incorrect label flips of noise and handwriting between two printed text lines may happen. If  $T_n$  is too small, the contextual constraints on the noise blocks cannot be fully used. We set  $T_n$  as half of the dominant character height (about 10 pixels in our experiments). Each node, together with all nodes connected to it, defines clique  $C_n$ . The number of connected nodes may vary from 0 to about 10, depending on the size of the block. As an approximation, we consider only the first four nearest connected neighbors. If the number of neighbors is less than four, we set the corresponding neighbors to NULL.

The geometric constraint on handwriting has weaker horizontal or vertical structure than machine printed words and, thus, is partially reflected in both cliques  $C_p$  and  $C_n$ . Therefore, we do not define a specific clique for handwriting.

## 4.3 Clique Potential

Clique potential is the energy associated with a clique. Generally, we assign high energy to an undesirable

configuration of the clique and low energy to a preferred configuration. For example, an undesired configuration of clique  $C_p$  (as shown in Fig. 5a) is that the left and right blocks are labeled as printed text and the center block as noise. Flipping the label of the center block from noise to printed text would achieve a more preferred configuration and reduce the total energy. Another undesirable configuration is that all blocks are labeled as printed text for the clique  $C_n$  in Fig. 5b. It should have higher energy than the configuration in which all blocks are labeled as noise. In many applications, the clique potentials are defined in ad hoc ways. One systematic way is to define clique potential as the occurrence frequency of each clique in the training set, which can be expressed as a function of local conditional probabilities. Based on this idea, we define two clique potentials  $V_p(c)$  and  $V_n(c)$  for cliques  $C_p$  and  $C_n$  as

$$V_p(c) = -\frac{P(X_l, X_c, X_r)}{(P(X_l)P(X_c)P(X_r))^w}, \quad (28)$$

$$V_n(c) = -\frac{P(X_c, X_1, X_2, X_3, X_4)}{(P(X_c)P(X_1)P(X_2)P(X_3)P(X_4))^w}, \quad (29)$$

where  $X_l$ ,  $X_c$ , and  $X_r$  are labels for the left, center, and right blocks of clique  $c$ ,  $w$  is a constant, and  $X_i$ ,  $i = 1, 2, 3, 4$ , is the label of the  $i$ th nearest block. The energy of the corresponding Gibbs distribution is

$$U(\underline{X}/\underline{Y}) = w_s \sum_{s \in \Omega} [-P(x_s/y_s)] + w_p \sum_{c \in C_p} V_p(c) + w_n \sum_{c \in C_n} V_n(c), \quad (30)$$

where  $w_s$ ,  $w_p$ , and  $w_n$  are weights which adjust the relative importance of classification confidence and contextual information for cliques  $C_p$  and  $C_n$ . If  $w_s = 1$ ,  $w_p = 0$ , and  $w_n = 0$ , no contextual information is used; with increase in  $w_p$  and  $w_n$ , more contextual information is emphasized. If we set  $w_p = w_n = \infty$  or, equivalently,  $w_s = 0$ , no classification confidence is used.

In the following experiments, we want to use MRFs for word block labeling. The number of handwritten words is much smaller than that of the other two types, leading to a lower estimated frequency of cliques with handwriting. As a result, the optimization tends to label handwritten words as machine printed text or noise. Therefore, we regularize the estimated clique frequency  $P(X_l, X_c, X_r)$  and  $P(X_c, X_1, X_2, X_3, X_4)$  by dividing by the product of the probabilities of the word block labels which compose the clique. The above regularization is very similar to the previous approach [48], where  $w$  is set to 1. In our case,  $w$  is changeable; increasing  $w$  will emphasize handwritten words. Our clique potential definition is very systematic and can be optimized for different applications.

After defining the cliques and the corresponding clique potential, we can search the optimal configuration of the labels of all blocks so that the total energy of the corresponding Gibbs distribution is minimized. Relaxation algorithms are often used for MRF optimization. There are two types of relaxation algorithms: stochastic and deterministic [7]. Stochastic algorithms can always converge to the global optimal solution if some constraints are satisfied. They are, however, computationally demanding. Deterministic algorithms are simpler, but only converge to local optimal

solutions depending on the initial value. In our experiments, Highest Confidence First (HCF), a deterministic approach, is used for MRF optimization due to its fast speed and good performance [49]. The HCF algorithm finds a block such that the flipping of its label to another label would reduce the total energy the most and then flips its label to the desired one. It repeats this procedure until no single flipping can further reduce the total energy. Since each flipping would reduce the energy and the energy is bounded below, the HCF algorithm converges in a finite number of steps. Fig. 6 is an example of the refined classification results after postprocessing. Compared with Fig. 4, we can see in Fig. 6 that most misclassified noise blocks are corrected, with a few exceptions due to their having fewer constraints. The misclassified small machine printed words are all corrected in Fig. 6c.

## 5 EXPERIMENTS

### 5.1 Data Set

We collected a total of 318 business letters from the tobacco industry litigation archives. These document images are noisy with a lot of handwritten annotations and signatures, few logos, and no figures or tables. Currently, we identify three classes: machine printed text, handwriting, and noise. Since the groundtruthing of each word block in the images of the entire database would be time consuming, we only did it for 94 extremely noisy document images. These 94 images are used for testing, and the other 224 images for training. All handwritten words (about 1,500) in the training set are groundtruthed. Since there is much more machine printed text and noise, we randomly selected and groundtruthed about the same number of samples of each type in the training set. We use *accuracy* and *precision* as metrics to evaluate the result:

$$\text{Accuracy of type } i = \frac{\# \text{ of correctly classified blocks of type } i}{\# \text{ of blocks of type } i} \quad (31)$$

$$\text{Precision of type } i = \frac{\# \text{ of correctly classified blocks of type } i}{\# \text{ of blocks classified as type } i}. \quad (32)$$

### 5.2 Classifier Comparison

In this section, we compare the performance of three different classifiers: the k-NN classifier, the Fisher classifier, and the SVM classifier. The SVM classifier is based on VC dimension theory and structural risk minimization theory of statistical learning [50]. A public domain SVM tool, LibSVM, is used in the following experiment [51]. The N-fold verification technique, a variation of the leave-one-out technique, is used to estimate the classification accuracy. Instead of holding one sample for testing at each iteration, it first divides the data set into  $N$  groups ( $N = 10$  in our experiment) and then holds one group of samples for testing and the remaining groups for training. The classification accuracies of all the classifiers are shown in Table 2. We can see that the SVM classifier achieved the highest accuracy. Considering the large variance, the improvement is not significant. The variance of the classification accuracy of all classifiers is the smallest for printed text and the largest for handwriting, indicating that the printed text is more compact in the feature space. Among all three

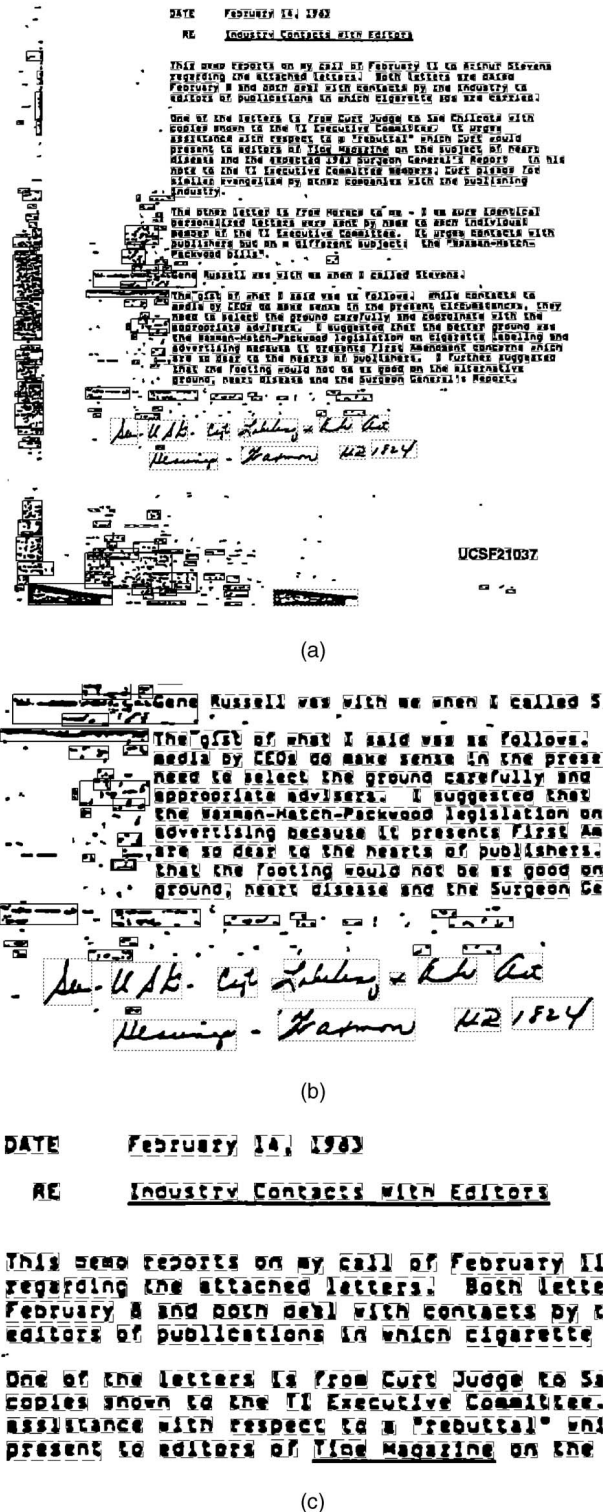


Fig. 6. Word block classification results after postprocessing. The result before postprocessing is shown in Fig. 4. (a) The whole document image. (b) and (c) Two parts of the image in (a). There is only one classification error in (b), which is highlighted with reverse video. All errors in (c) are corrected after postprocessing.

classifiers, the Fisher classifier is the fastest since only one vector multiplication is needed to perform a classification. Therefore, we use the Fisher classifier for the rest of experiments.

The classification result on the test set of 94 images, using the Fisher classifier, is shown in Table 3. The accuracies on all three classes range from 93.2 percent to 96.8 percent, with the overall accuracy 96.1 percent. While this overall accuracy is very high, we notice that the precision for handwriting is very low (62.9 percent). This is mainly because of the small number of handwritten words in the testing set. Even small percentages of misclassification of machine printed text and noise as handwriting will significantly decrease the precision of handwriting.

### 5.3 Postprocessing Using MRFs

In the following experiments, we investigate how MRFs can improve classification accuracy. In the first run, we set  $w_s = 0$ ,  $w_n = 0$ , and  $w_p = 1$  to show the effectiveness of clique  $C_p$ . Fig. 7a shows the number of corrected blocks which were previously misclassified, with change in  $w$ . As expected,  $C_p$  is very effective for machine printed words, but not so effective for handwriting and noise. When  $w = 0.3$  (under this condition, the classification accuracy of all three classes increases), 355 (46 percent) of the previously misclassified machine printed words are corrected. When  $w$  increases, handwriting is more emphasized, leading to higher classification accuracy of handwriting and lower accuracy of machine printed words and noise. In practice,  $w$  can be adjusted to optimize the overall accuracy.

In the second run, we test the effectiveness of clique  $C_n$  by setting  $w_s = 0$ ,  $w_p = 0$ , and  $w_n = 1$ . As shown in Fig. 7b, clique  $C_n$  is very effective in correcting classification errors of noise blocks. The classification error of noise blocks is greatly reduced when  $w$  is small. For  $w = 0.6$  (under this condition, the classification accuracy of all classes increases), the number of misclassified noise blocks is reduced by 99 (35 percent).  $C_n$  can also correct some classification errors of machine printed words, but is less effective than  $C_p$ , as shown in Fig. 7a.

The third run tests the effectiveness of classification confidence for postprocessing. Fig. 7c shows postprocessing results by adjusting  $w_p$  when  $w = 0.3$ ,  $w_n = 0$ , and  $w_s = 1$ . Adjusting  $w_p$  will change the total flip number greatly. When  $w_p = 0$ , the energy reaches the minimum with the initial labels and the total flip number is 0. When  $w_p$  increases, more emphasis is put on the contextual information and the flip number increases. When  $w_p \rightarrow +\infty$ , it converges to the case of  $w_p = 1$  and  $w_s = 0$ , the setting of the first run. The maximal overall classification accuracy is achieved when  $w_p = 6$ . Compared with the first run, the total number of corrected blocks increases from 389 to 424 by incorporating classification confidence. Similar results are achieved by combining classification confidence with clique  $C_n$ .

In the last run, we fix  $w_s = 1$  and manually adjust  $w$ ,  $w_p$ , and  $w_n$  to optimize the overall classification accuracy. The final parameters we chose are  $w = 0.39$ ,  $w_p = 5$ , and  $w_n = 4$ . Table 4 shows the results after postprocessing. The "Error Reduction Rate" in Table 4 is defined as follows:

$$\text{Error Reduction Rate} = \frac{\# \text{ of Errors Before Postprocessing} - \# \text{ of Errors After Postprocessing}}{\# \text{ of Error Before Postprocessing}} \quad (33)$$

The error rate reduces to about half of the original for both machine printed text and noise, but increases slightly for handwriting. However, compared with Table 3, the precision of handwriting increases from 62.9 percent to 83.3 percent due to fewer machine printed text and noise misclassifications as handwriting. The overall accuracy increases from 96.1 percent to 98.1 percent.

TABLE 2  
Performance Comparison of Three Different Classifiers: The k-NN Classifier, the Fisher Classifier, and the SVM Classifier

	# of blocks	the k-NN classifier			the Fisher classifier			the SVM classifier		
		Correct	Accuracy	Variance	Correct	Accuracy	Variance	Correct	Accuracy	Variance
Printed text	1,519	1,489	98.0%	1.4%	1,473	97.0%	1.1%	1,480	97.4%	1.2%
Handwriting	1,518	1,390	91.6%	2.3%	1,410	92.9%	2.2%	1,435	94.5%	2.1%
Noise	1,512	1,406	93.0%	2.0%	1,451	96.0%	1.5%	1,453	96.1%	1.2%
Overall	4,549	4,285	94.2%	1.3%	4,344	95.5%	0.9%	4,368	96.0%	0.9%

TABLE 3  
Single Word Block Classification

	# of blocks	Percentage	# of correctly classified blocks	# of misclassified blocks	Accuracy	Precision
Printed text	19,227	66.9%	18,446	781	95.9%	99.5%
Handwriting	701	2.4%	653	48	93.2%	62.9%
Noise	8,802	30.7%	8,522	280	96.8%	93.0%
Overall	28,730	100.0%	27,621	1,109	96.1%	N/A

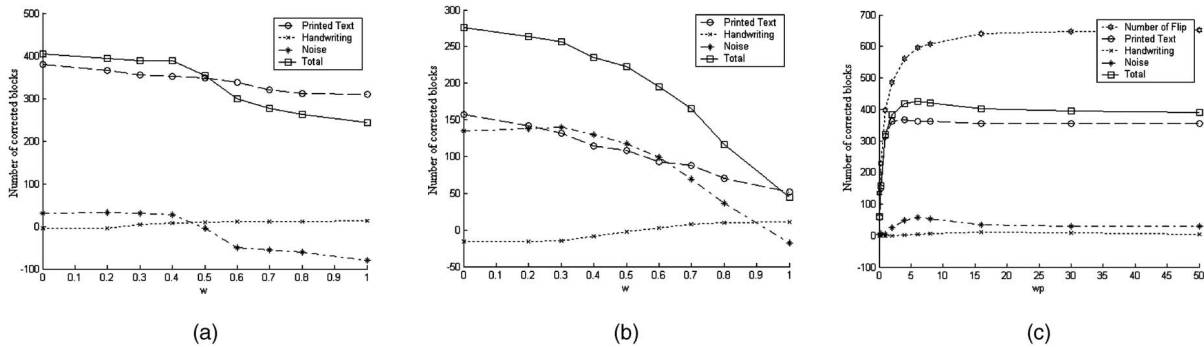


Fig. 7. MRF-based postprocessing. (a) Number of corrected blocks using clique  $C_p$ . (b) Number of corrected blocks using clique  $C_n$ . (c) Number of corrected blocks using clique  $C_p$  and classification confidence.

TABLE 4  
Word Block Classification after MRF-Based Postprocessing

	# of blocks	# of correctly classified blocks	# of misclassified blocks	Reduction of misclassified blocks	Error reduction rate	Accuracy	Precision
Printed text	19,227	18,835	392	389	49.8%	98.0%	99.7%
Handwriting	701	652	49	-1	-2.1%	93.0%	83.3%
Noise	8,802	8,682	120	160	57.1%	98.6%	96.0%
Total	28,730	28,169	561	548	49.4%	98.1%	N/A

Fig. 8 shows another example of machine printed text and handwriting identification from noisy documents. To display the classification results clearly, we decompose the classified image into three layers, representing machine printed text (Fig. 8b), handwriting (Fig. 8c), and noise (Fig. 8d), respectively. The result is good with very few misclassifications.

Our approach is very general and can be extended to other languages with minor modification. Fig. 9 shows

identification results for a Chinese document. In Chinese, there is no clear definition of words and no spaces between neighboring words. Therefore, the parameters of our word segmentation module are adjusted to get characters. We only need to retrain the classifiers; the postprocessing module is intact. We can see that most handwriting and noise blocks are classified correctly, but several machine printed digits are misclassified as handwriting. On the right margin of the

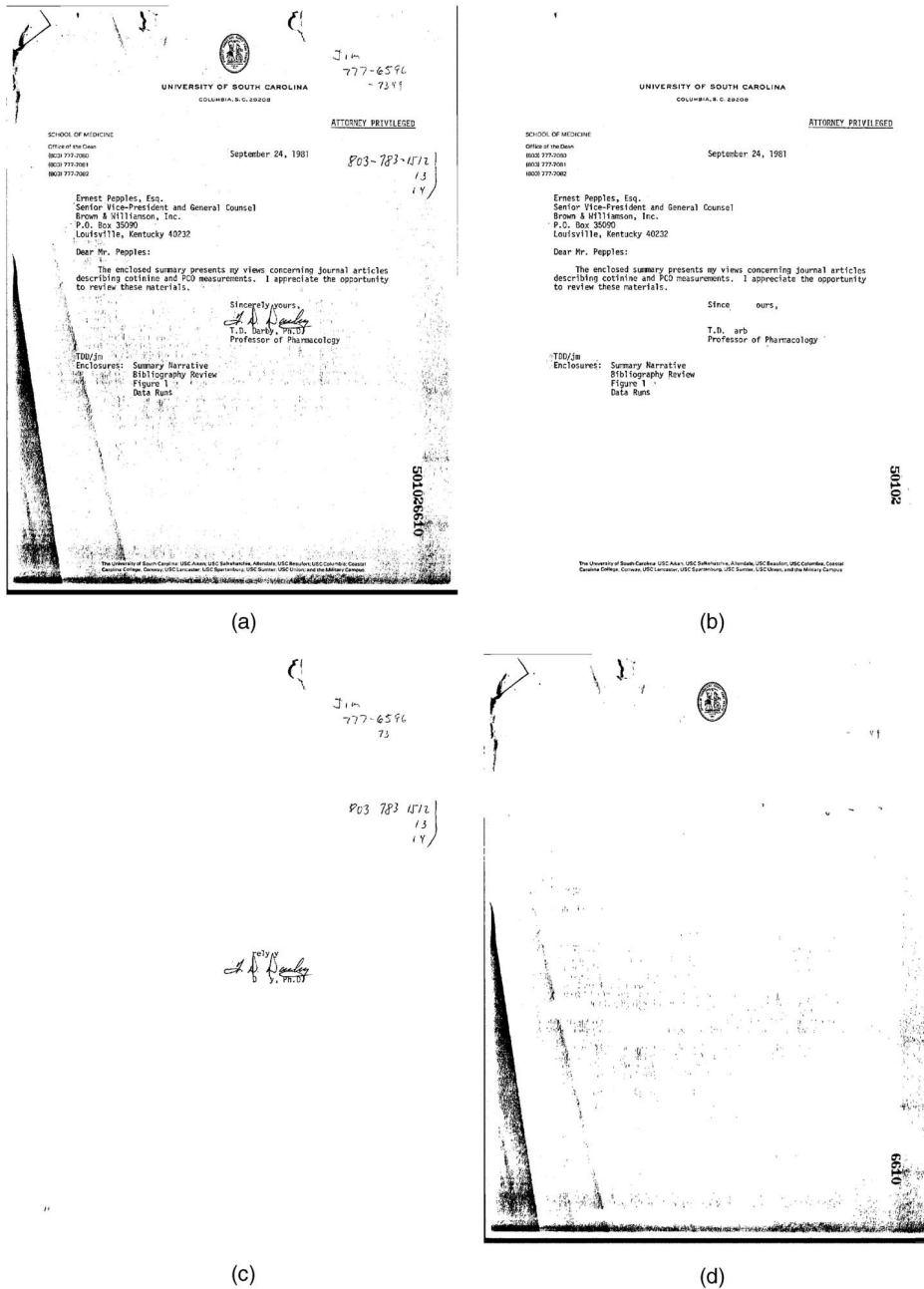


Fig. 8. An example of machine printed text and handwriting identification from noisy documents. (a) The original document image, (b) machine printed text, (c) handwriting, and (d) noise. The logo is classified as noise since currently we only consider three classes.

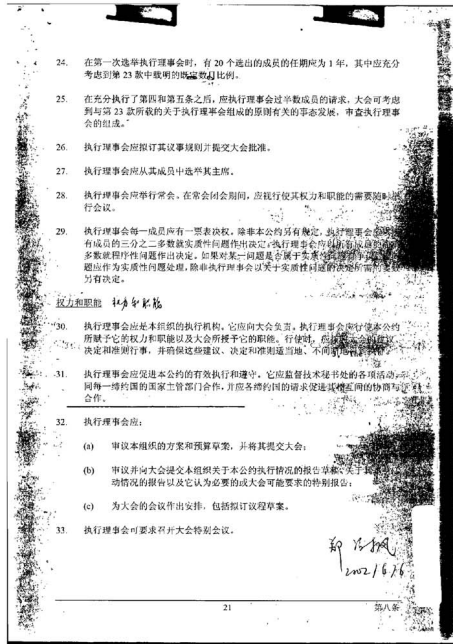
document, some machine printed text is identified as noise due to touching.

Our approach is fast; the averaging processing time for a business letter scanned at 300 DPI is about 2-3 seconds on a PC with 1.7 GHZ CPU and 1.0 GB memory.

**5.4 Page Segmentation in Noisy Images**

In this experiment, we show that our method can improve general page segmentation results after removing identified noise. We evaluated two widely used zone segmentation algorithms: the Docstrum algorithm [2] and ScanSoft SDK, a commercial OCR software package [3]. Many different zone segmentation evaluation metrics have been proposed in previous work. Kanai et al. [52] evaluated zone segmentation

accuracy from the OCR aspect. Any zone splitting and merging, if it does not affect the reading order of the text, is not penalized. The approach of Mao and Kanungo is based on text lines, which penalizes only horizontal text line splitting and merging since it will change the reading order of text [53]. Randriamasy et al. [54] proposed an evaluation method based on multiple ground truth, which is very expensive. Liang et al.'s approach is performed at the zone level [30]. After finding the correspondence between the segmented and groundtruthed zones, any large enough difference is penalized. We use Liang et al.'s scheme in our experiment since we focus more on zone segmentation. From the OCR perspective, vertical splitting or merging of different zones should not be penalized even when these zones have



(a)

24. 在第一次选举执行理事会时, 有 0 个选出的成员的任期应为 1 年, 其中应充分考虑第 23 款中载明的赔款数目比例。

25. 在充分执行了第四和第五条之后, 应执行理事会过半数成员的请求, 大会可考虑对第 23 款所载的关于执行理事会组成的原则有关的事态发展, 审查执行理事会的组成。

26. 执行理事会应拟订其议事规则并提交大会批准。

27. 执行理事会应从其成员中选举其主席。

28. 执行理事会应举行年会。在常会闭会期间, 应行使使其权力和职能的必要临时性会议。

29. 执行理事会每一成员应有一票表决权, 除非本公约另有规定。执行理事会应有成员的三分之二多数就实质性事项作出决定。执行理事会应向其成员的多数成员中间性问题作出决定。如果对某一问题是属于于先决条件的, 则应作为实质性事项处理, 除非执行理事会以关于实质性问题的决定而另有决定。

**权力和职能**

30. 执行理事会应是本组织的执行机构, 它应向大会负责, 执行理事会应行使本公约所赋予它的权力和职能以及大会所授予它的职能。行使时, 应遵守大会的决议和决定, 并将这些建议、决定和决定通知地、决定和决定通知地。

31. 执行理事会应促进本公约的有效执行和遵守。它应监督技术秘书处各项活动, 并同每一缔约国的国家主管部门合作, 并同各缔约国的请求使这些机构间的协调与合作。

执行理事会应:

(a) 审议本组织的方案和预算草案, 并将其提交大会;

(b) 审议并向大会提交本组织关于本公约执行情况的报告草案, 关于其活动情况的报告以及它认为必要的或大会可能要求的特别报告;

(c) 为大会的会议作出安排, 包括拟订议程草案。

33. 执行理事会可要求召开大会特别会议。

(b)

23. 2

23. 4

26. 力

29. 秘书职能

30. 会

32. 会

33. 2

郑在枫  
2002/6/16

(c)

5.

8.

郑在枫  
2002/6/16

(d)

Fig. 9. An example of machine printed text and handwriting identification from Chinese documents. (a) Original Chinese document image, (b) machine printed text, (c) handwriting, and (d) noise.

different physical and semantic properties, but, from the point view of zone segmentation, it should be penalized.

There are 1,374 machine printed text zones in 94 noisy document images. The experimental results are shown in Table 5. All merging and splitting errors are counted as partially correct in the table. Before noise removal, ScanSoft gets very poor results, with an accuracy of 15.9 percent, on noisy documents under this metric. After analyzing the segmentation results, we found that ScanSoft tends to merge horizontally arrayed zones into one zone, which is suitable for

documents with simple layouts such as technical articles, but not suitable for other document types such as business letters. The Docstrum algorithm outputs many more zones than ScanSoft, resulting in a higher accuracy (53.0 percent), but also a higher false alarm rate (114.1 percent). After noise removal, the accuracy of both algorithms increases significantly, from 15.9 percent to 48.4 percent for ScanSoft and from 53.0 percent to 78.0 percent for the Docstrum algorithm. The false alarm rate is reduced from 32.5 percent to 1.3 percent for ScanSoft and from 114.1 percent to 7.9 percent for Docstrum.

TABLE 5  
Machine Printed Zone Segmentation Experimental Results  
on 94 Noisy Document Images (Total 1,374 Zones), before and after Noise Removal

	Before noise removal				After noise removal			
	Correctly segmented zones	False alarm zones	Partially correctly segmented zones	Missed zones	Correctly segmented zones	False alarm zones	Partially correctly segmented zones	Missed zones
ScanSoft	219 (15.9%)	446 (32.5%)	1148 (83.7%)	7 (0.5%)	665 (48.4%)	18 (1.3%)	671 (48.8%)	38 (2.8%)
Docstrum	728 (53.0%)	1568 (114.1%)	646 (47.0%)	0 (0.0%)	1071 (78.0%)	109 (7.9%)	270 (19.7%)	33 (2.4%)

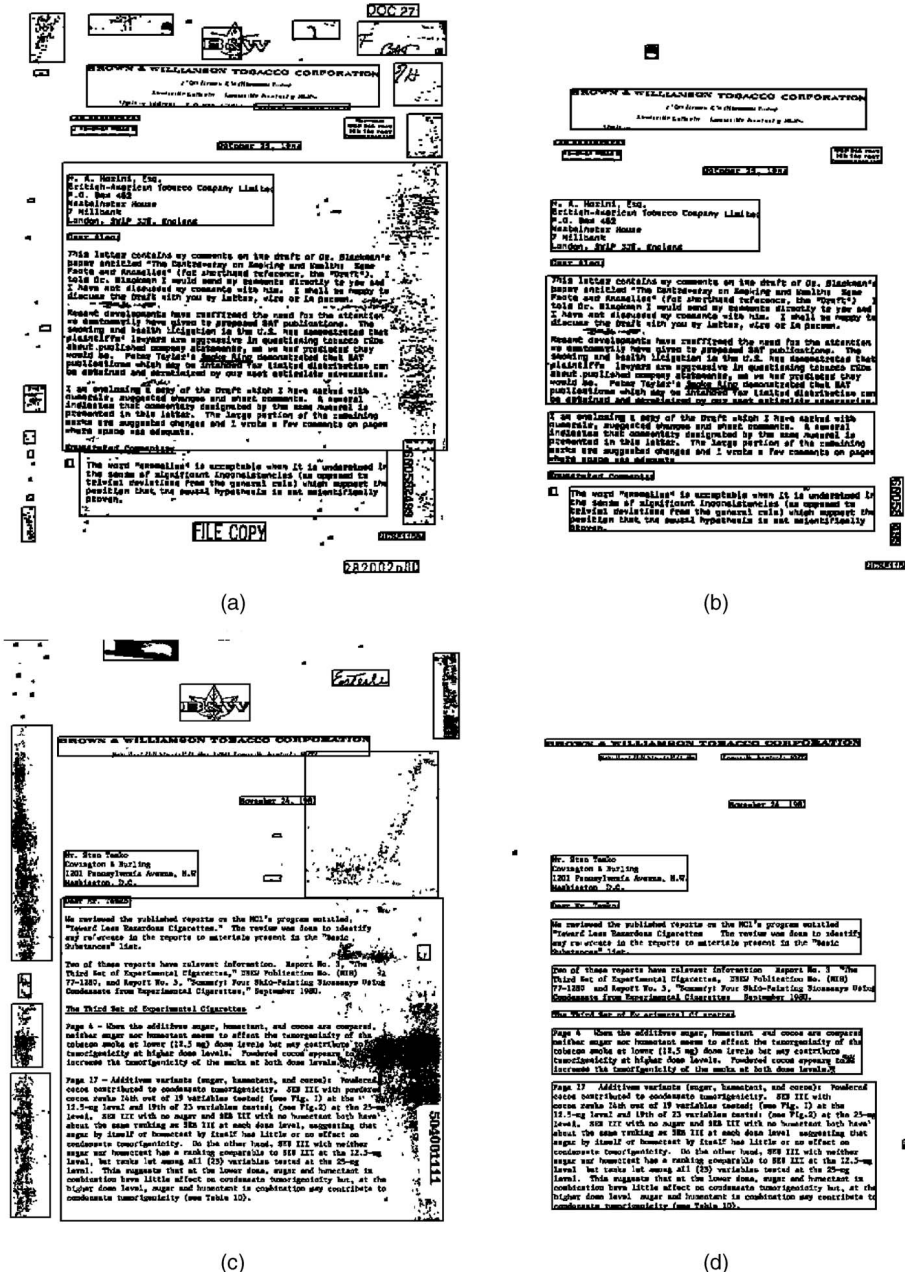


Fig. 10. Zone segmentation before and after noise removal using the Docstrum algorithm. (a) and (c) Show the results before noise removal. (b) and (d) Are the results after noise removal.

Fig. 10 shows the zone segmentation results for two noisy documents with the Docstrum algorithm before and after noise removal. The handwriting is output to another layer which is not shown here. We can see that, after noise removal, there are many fewer splitting and merging errors and overall the segmentation results are significantly improved.

## 6 SUMMARY

In this paper, we have presented an approach to segmenting and identifying text from extremely noisy document images. Instead of using simple filtering rules, we treat noise as a distinct class and use statistical classification techniques to classify each block into machine printed text, handwriting, and noise. We then use Markov Random Fields to incorporate contextual information for postprocessing. Experiments show that MRFs are a very effective tool for modeling local dependency among neighboring image components. After postprocessing, the classification error rate is reduced by approximately 50 percent. Our method is general enough to be extended to documents in other languages. The technique presented in this paper can be used for image enhancement to improve page segmentation accuracy of noisy documents. After noise identification and removal, the zone segmentation accuracy increase from 53 percent to 78 percent using the Docstrum algorithm.

Currently, our clique potential definition considers only the labels of each block inside the clique, which may lose useful information. For example, for clique  $C_p$ , a clique of three printed words with roughly the same height is quite different from one with different heights. In the latter case, it is possible that one of the blocks is erroneously identified. Another potential improvement is to integrate high-level contextual information in addition to the local contextual information that we used. For example, the text line and zone segmentation results can be fed back to our classification module to refine the classification. Effective use of contextual information is one of our future research directions.

## ACKNOWLEDGMENTS

The authors would like to thank Dr. Azriel Rosenfeld for carefully editing this manuscript and the anonymous reviewers for their constructive comments. This research is supported by the US Department of Defense under contract MDA-9040-2C-0406.

## REFERENCES

- [1] A.K. Jain and B. Yu, "Document Representation and Its Application to Page Decomposition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 294-308, Mar. 1998.
- [2] L. O'Gorman, "The Document Spectrum for Page Layout Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162-1173, Nov. 1993.
- [3] ScanSoft Corp, ScanSoft Developer's Kit 2000, <http://www.scansoft.com>, 2003.
- [4] J.J. Hull, "Incorporating Language Syntax in Visual Text Recognition with a Statistical Model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1251-1256, Dec. 1996.
- [5] R.M.K. Sinha, B. Prasada, G.F. Houles, and M. Sabourin, "Hybrid Contextual Text Recognition with String Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 915-925, Sept. 1993.
- [6] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721-741, 1984.
- [7] S.Z. Li, *Markov Random Field Modeling in Image Analysis*, second ed. New York: Springer-Verlag, 2001.
- [8] G. Nagy, S. Seth, and S. Stoddard, "Document Analysis with an Expert System," *Pattern Recognition in Practice II*, pp. 149-155, Elsevier Science, 1984.
- [9] D. Sylwester and S. Seth, "Adaptive Segmentation of Document Images," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 827-831, 2001.
- [10] H.S. Baird, S.E. Jones, and S.J. Fortune, "Image Segmentation by Shape-Directed Covers," *Proc. Int'l Conf. Pattern Recognition*, pp. 820-825, 1990.
- [11] T. Pavlidis and J. Zhou, "Page Segmentation and Classification," *CVGIP*, vol. 54, no. 6, pp. 484-496, 1992.
- [12] R.M. Haralick, "Document Image Understanding: Geometric and Logical Layout," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 385-390, 1994.
- [13] Y. Wang, R.M. Haralick, and I.T. Phillips, "Zone Content Classification and Its Performance Evaluation," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 540-544, 2001.
- [14] K. Etemad, D. Doermann, and R. Chellappa, "Multiscale Document Page Segmentation Using Soft Decision Integration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 92-96, Jan. 1997.
- [15] A.K. Jain and S. Bhattacharjee, "Text Segmentation Using Gabor Filters for Automatic Document Processing," *Machine Vision and Applications*, vol. 5, pp. 169-184, 1992.
- [16] S.-W. Lee and B.-S. Ryu, "Parameter-Free Geometric Document Layout Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1240-1256, Nov. 2001.
- [17] K.C. Fan, L.S. Wang, and Y.T. Tu, "Classification of Machine-Printed and Handwritten Texts Using Character Block Layout Variance," *Pattern Recognition*, vol. 31, no. 9, pp. 1275-1284, 1998.
- [18] J. Fanke and M. Oberlander, "Writing Style Detection by Statistical Combination of Classifier in Form Reader Applications," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 581-585, 1993.
- [19] V. Pal and B.B. Chaudhuri, "Machine-Printed and Handwritten Text Lines Identification," *Pattern Recognition Letters*, vol. 22, nos. 3-4, pp. 431-441, 2001.
- [20] S.N. Srihari, Y.C. Shim, and V. Ramanprasad, "A System to Read Names and Address on Tax Forms," Technical Report CEDAR-TR-94-2, CEDAR, SUNY, Buffalo, N.Y., 1994.
- [21] J.K. Guo and M.Y. Ma, "Separating Handwritten Material from Machine Printed Text Using Hidden Markov Models," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 439-443, 2001.
- [22] K. Kuhnke, L. Simoncini, and Z.M. Kovacs-V, "A System for Machine-Written and Hand-Written Character Distinction," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 811-814, 1995.
- [23] Y. Zheng, C. Liu, and X. Ding, "Single Character Type Identification," *Proc. SPIE Conf. Document Recognition and Retrieval*, pp. 49-56, 2002.
- [24] Y. Zheng, H. Li, and D. Doermann, "The Segmentation and Identification of Handwriting in Noisy Document Images," *Proc. Int'l Workshop Document Analysis Systems*, pp. 95-105, 2002.
- [25] H.S. Baird, "Calibration of Document Image Defect Models," *Proc. Symp. Document Analysis and Information Retrieval*, pp. 1-16, 1993.
- [26] T. Kanungo, R.M. Haralick, and I. Phillips, "Nonlinear Local and Global Document Degradation Models," *Int'l J. Imaging Systems and Technology*, vol. 5, no. 4, pp. 220-230, 1994.
- [27] S. Sural and P.K. Das, "A Two-State Markov Chain Model of Degraded Document Images," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 463-466, 1999.
- [28] M. Cannon, J. Hochberg, and P. Kelly, "Quality Assessment and Restoration of Typewritten Document Images," *Int'l J. Document Analysis and Recognition*, vol. 2, pp. 80-89, 1999.
- [29] H. Li and D. Doermann, "Text Quality Estimation in Video," *Proc. SPIE Conf. Document Recognition and Retrieval*, pp. 232-243, 2002.
- [30] J. Liang, I.T. Phillips, and R.M. Haralick, "Performance Evaluation of Document Layout Analysis Algorithms on the UW Data Set," *Proc. SPIE Conf. Document Recognition*, pp. 149-160, 1997.
- [31] R.P. Loce and E.R. Dougherty, *Enhancement and Restoration of Digital Documents—Statistical Design of Nonlinear Algorithms*. SPIE Optical Eng. Press, 1997.
- [32] L. O'Gorman, "Image and Document Processing Techniques for the RightPages Electronic Library System," *Proc. Int'l Conf. Pattern Recognition*, pp. 820-825, 1992.



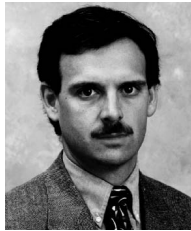
- [33] K. Chinnsarn, Y. Rangsanseri, and P. Thitimajshima, "Removing Salt-and-Pepper Noise in Text/Graphics Images," *Proc. IEEE Asia-Pacific Conf. Circuits and Systems*, pp. 459-462, 1998.
- [34] J. Liang and R.M. Haralick, "Document Image Restoration Using Binary Morphological Filters," *Proc. SPIE Conf. Document Recognition*, pp. 274-285, 1996.
- [35] T. Kanungo, R.M. Haralick, H.S. Baird, W. Stuetzle, and D. Madigan, "Document Degradation Models: Parameter Estimation and Model Validation," *Proc. Int'l Workshop Machine Vision Applications*, pp. 552-557, 1994.
- [36] T. Kanungo, H.S. Baird, and R.M. Haralick, "Validation and Estimation of Document Degradation Models," *Proc. Symp. Document Analysis and Information Retrieval*, pp. 217-228, 1995.
- [37] T. Kanungo and Q. Zheng, "Estimation of Morphological Degradation Model Parameters," *Proc. IEEE Int'l Conf. Speech and Signal Processing*, pp. 1961-1964, 2001.
- [38] H.S. Baird, "Document Image Quality: Making Fine Discriminations," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 459-462, 1999.
- [39] D. Gabor, "Theory of Communication," *J. Instructional Electrical Eng.*, vol. 93, pp. 429-459, 1946.
- [40] T. Akiyama and N. Hagita, "Automated Entry System for Printed Documents," *Pattern Recognition*, vol. 23, no. 11, pp. 1141-1154, 1990.
- [41] R. Haralick, B. Shanmugam, and I. Dinstein, "Texture Features for Image Classification," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610-622, 1973.
- [42] A. Soffer, "Image Categorization Using Texture Features," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 233-237, 1997.
- [43] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second ed. New York: Academic Press, 1990.
- [44] A.K. Jain and D. Zongker, "Feature Selection: Evaluation, Application and Small Sample Performance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153-158, Feb. 1997.
- [45] L. Kukolick and R. Lippmann, *LNKnet User's Guide*, <http://www.ll.mit.edu/IST/lnknet>, 2003.
- [46] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, second ed. Oxford Univ. Press, 2001.
- [47] X. Lin, X. Ding, and M. Chen, "Adaptive Confidence Transform Based Classifier Combination for Chinese Character Recognition," *Pattern Recognition Letters*, vol. 19, no. 10, pp. 975-988, 1998.
- [48] C. Wolf and D. Doermann, "Binarization of Low Quality Text Using a Markov Random Field Model," *Proc. Int'l Conf. Pattern Recognition*, 2002.
- [49] P.B. Chou, P.R. Cooper, and M.J. Swain, "Probabilistic Network Inference for Cooperative High and Low Level Vision," *Markov Random Fields: Theory and Application*, R. Chellapa and A. Jain, eds., San Diego, Calif.: Academic Press, 1993.
- [50] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [51] C.-C. Chang and C.-J. Lin, *Libsvm—A Library for Support Vector Machines*, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, 2003.
- [52] J. Kanai, S.V. Rice, T.A. Nartker, and G. Nagy, "Automated Evaluation of OCR Zoning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 86-90, Jan. 1995.
- [53] S. Mao and T. Kanungo, "Empirical Performance Evaluation Methodology and Its Application to Page Segmentation Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 242-256, Mar. 2001.
- [54] S. Randriamasy, L. Vincent, and B. Wittner, "An Automatic Benchmarking Scheme for Page Segmentation," *Proc. SPIE Conf. Document Recognition*, pp. 217-227, 1994.



**Yefeng Zheng** received the BE and ME degrees from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 1998 and 2001, respectively. Currently, he is pursuing the PhD degree in the Department of Electrical and Computer Engineering at the University of Maryland, College Park. His research interests include document image analysis, pattern recognition, and computer vision. He is a student member of the IEEE.



**Huiping Li** received the BSc degree from the Department of Electrical Engineering at the University of Science and Technology of China in 1992. He received the PhD degree from the Department of Electrical Engineering at the University of Maryland, College Park, in 2000 with a thesis entitled "Automatic Processing and Analysis of Text in Digital Video." Currently, Dr. Li is a research faculty member at the Laboratory for Language and Media Processing in the University of Maryland's Institute for Advanced Computer Studies, leading the document-related research. His research topics include document decomposition, handwritten detection and retrieval in degraded documents, text detection, tracking, enhancement, and recognition in real scene images, sports video classification, and surveillance video compression. Before joining the University of Maryland, he worked for Object Video, Inc. to develop object-based video codecs. His work included developing algorithms for video content analysis, video image enhancement in low bit rates, and coding performance evaluation and analysis. He has consulted on contracts to various corporations and government agencies interested in document and multimedia related projects. He has worked extensively in developing sign recognition capabilities with Perceptek and is the principal investigator of the PDA-based sign translation sponsored by the US Army's SBIR project. Some recent projects include the integration of native video codec into Java Media Framework and video-over-IP. He is a member of the IEEE.



**David Doermann** received the BSc degree in computer science and mathematics from Bloomsburg University in 1987. He received the MSc degree in 1989 from the Department of Computer Science at the University of Maryland. He continued his studies in the Computer Vision Laboratory, where he received the PhD degree in 1993 with a thesis entitled "Document Image Understanding: Integrating Recovery and Interpretation." He is codirector of the Laboratory for Language and Media Processing in the University of Maryland's Institute for Advanced Computer Studies and an adjunct member of the Graduate Faculty. His research centers widely around the topics of document image analysis and multimedia information processing. Recent intelligent document image analysis projects include page decomposition, structural analysis and classification, page segmentation, logo recognition, document image compression, duplicate document image detection, image-based retrieval, character recognition, generation of synthetic OCR data, signature verification. In video processing, projects have centered around the segmentation of compressed domain video sequences, structural representation and classification of video, detection of reformatted video sequences, and the performance evaluation of automated video analysis algorithms. Dr. Doermann has served on numerous board and program committees and is an editor of the newly formed *International Journal on Document Analysis and Recognition*. In addition, he is the principle investigator on a number of contracts to government agencies and corporations. He is a member of the IEEE.