IJDAR (2013) 16:1–16
DOI 10.1007/s10032-011-0179-z

ORIGINAL PAPER

Handwritten text separation from annotated machine printed documents using Markov Random Fields

Xujun Peng · Srirangaraj Setlur ·
Venu Govindaraju · Ramachandrula Sitaram

Received: 1 March 2010 / Revised: 9 October 2011 / Accepted: 11 October 2011 / Published online: 2 November 2011

Abstract The convenience of search, both on the personal computer hard disk as well as on the web, is still limited mainly to machine printed text documents and images because of the poor accuracy of handwriting recognizers. The focus of research in this paper is the segmentation of handwritten text and machine printed text from annotated documents sometimes referred to as the task of "ink separation" to advance the state-of-art in realizing search of hand-annotated documents. We propose a method which contains two main steps—patch level separation and pixel level separation. In the patch level separation step, the entire document is modeled as a Markov Random Field (MRF). Three different classes (machine printed text, handwritten text and overlapped text) are initially identified using G-means based classification followed by a MRF based relabeling procedure. A MRF based classification approach is then used to separate overlapped text into machine printed text and handwritten text using pixel level features forming the second step of the method. Experimental results on a set of machine-printed documents which have been annotated by multiple writers in an office/collaborative environment show that our method is robust and provides good text separation performance.

X. Peng (✉) · S. Setlur · V. Govindaraju
Department of Computer Science and Engineering, UB Commons, Center for Unified Biometrics and Sensors, 520 Lee Entrance, Suite 202, SUNY at Buffalo, Amherst, NY, 14228 USA
e-mail: xpeng@buffalo.edu

S. Setlur
e-mail: setlur@buffalo.edu

V. Govindaraju
e-mail: govind@buffalo.edu

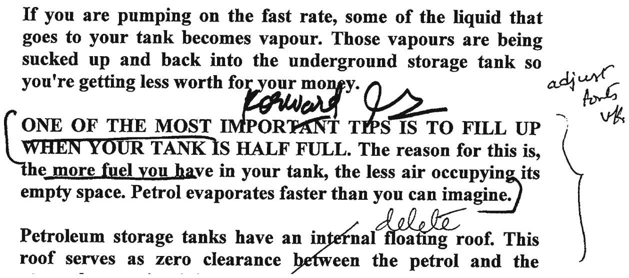
R. Sitaram
HP Labs India, Hosur Main Road, Adugodi,
Bangalore 560030, India
e-mail: sitaram@hp.com

Keywords Text identification · Markov Random Field · Documents retrieval · Ink separation · Segmentation

1 Introduction

After decades of research and development, automatic document processing systems have attained considerable success in that large volumes of paper documents can be digitized and processed to recognize textual content and facilitate information retrieval. Unfortunately, the convenience of retrieval and search is limited to clean machine-printed text document images and recognition of free-form handwriting still remains a considerable challenge. In many scenarios, a mixture of machine printed text and handwriting occur within a single document, such as hand-filled medical forms, tax forms, annotated correspondence as shown in Fig. 1. Sometimes, it may be of interest to know who signed or edited a document or what was written on a document and retrieve memos or a specific keyword by a particular author. The preprocessing of mixed documents to isolate handwritten text from machine printed text is a necessary step in the design of such recognition and retrieval systems.

Although research into the location of text blocks and discrimination of machine printed and handwritten text can be traced back to the early work on extraction and recognition of handwritten ZIP codes from mail pieces [36], much of it was focused on identification of clearly separated and clean text. A significant amount of prior research for handwriting identification is based on document layout analysis and zone classification wherein a document is segmented into words, lines, and zones in a bottom-up approach or in a top–down manner [15,27] and contextual information based on the location of text zones points to the location of handwritten data.



If you are pumping on the fast rate, some of the liquid that goes to your tank becomes vapour. Those vapours are being sucked up and back into the underground storage tank so you're getting less worth for your money.

ONE OF THE MOST IMPORTANT TIPS IS TO FILL UP WHEN YOUR TANK IS HALF FULL. The reason for this is, the more fuel you have in your tank, the less air occupying its empty space. Petrol evaporates faster than you can imagine.

Petroleum storage tanks have an internal floating roof. This roof serves as zero clearance between the petrol and the

Fig. 1 Example of an annotated machine printed document

Another thrust has been on using textual elements alone without considering context or layout. Eduardo et al. [9] suggested two types of features (content related features and shape related features) to characterize handwritten text on bank check images. This approach uses a fixed size frame to extract features, but locating and labeling overlapped text is not part of the model in their work. Jang et al. [20] proposed an approach using geometric features to classify machine printed and handwritten addresses on mail-pieces. Guo and Ma [17] separated handwritten material from documents using a Hidden Markov Model (HMM) by using the vertical projection of each word on the horizontal axis and considering a projected word as a sequential signal. This model is reliable for recognizing machine printed isolated characters but it is difficult to extend the work to recognize annotations and handwritten text because of the large variations of personal fonts and styles. Farooq et al. [12] used an EM based Bayesian neural network model in order to identify Arabic handwritten text in mixed documents. They used multiple classifiers which were weighted by their posterior probabilities and introduced a new neurons layer which used an error function to penalize solutions that led to mis-classification. Fan et al. [11] isolated the machine printed text from handwritten text based on their differences between the geometric arrangement. In order to extract names in mixed documents, Likforman-Sulem et al. [26] identified handwritten and printed pseudo words and classified them with a neural network using both image-based and textual-based features.

One potential problem of previous research on text separation is that although the basic unit (zone, line or word) of the system works well to classify those patches whose sizes are typically larger than words, it cannot separate overlapped patches which contain both handwritten and machine printed text because they are considered as a single unit in the system. Zhao and Davis [38] used color information of each pixel to segment picture foreground from background which can also be used for document segmentation. However, color information is rarely available in the document analysis field where the input is usually a scanned binary document image. Some work in computer vision also provides us with interesting clues to address the problem of overlapped text patches. Eran and Shimon [10] presented a top–down/bottom–up segmentation method using small fragments that contain common object parts. Corso [8] suggested a multilevel aggregation procedure which groups each pixel to a set of aggregate regions in a multilevel coarsening of the image. Alpert et al. [2] used a Bayes rule to determine whether or not to merge two neighbor regions and segment images using these regions. All these approaches show that we can use a unit which is smaller than words as the basic element to separate overlapped text.

In recent years, inspired by the success of Markov Random Field (MRF) in the area of image processing and restoration [13, 14], considerable research has been made to extend MRF to document restoration and preprocessing. Cao and Govindaraju [6, 7] proposed a method using small fixed size patches to represent handwriting and restore broken handwritten text based on a MRF framework. Similarly, Banerjee [3] used a flexible MRF-based optimization framework to remove noise and restore machine-printed text. These methods can be looked at as an extension of Freeman's algorithm [13]. Zheng et al. [39] proposed a two step approach to identify three different types of text in mixed documents. They initially separated text using a Fisher classifier followed by a Gibbs network based relabeling which was further optimized using a Highest Confidence First algorithm. A similar approach but using Conditional Random Field has been proposed by Shetty et al. [35].

2 System structure

In this paper, we propose a two-step composite method to separate handwritten text, machine printed text and overlapped text from annotated documents.

As shown in Fig. 2, the first step (step I) of the proposed method takes the entire document as its input and separates the text into three classes: machine printed text, handwritten text and overlapped text on patch level. This step has three sub-procedures. The first sub-procedure is a preprocessing procedure to extract patches of the document and extract features for each patch which is covered in Sect. 4.1. Section 4.2 describes the second sub-procedure of step I which is a G-means based initial classification procedure. The last sub-procedure of patch level text separation is to relabel the result of initial classification using a MRF based model. Section 4.3 describes the details of the relabeling. Section 4.4 presents the experimental results of MRF based patch level text separation.

The second step (step II) takes the overlapped text patch as its input which is then further separated into machine printed text and handwritten text on pixel level. The first sub-procedure in step II is to extract shape-context based features for each pixel. The next sub-procedure is an aggregation

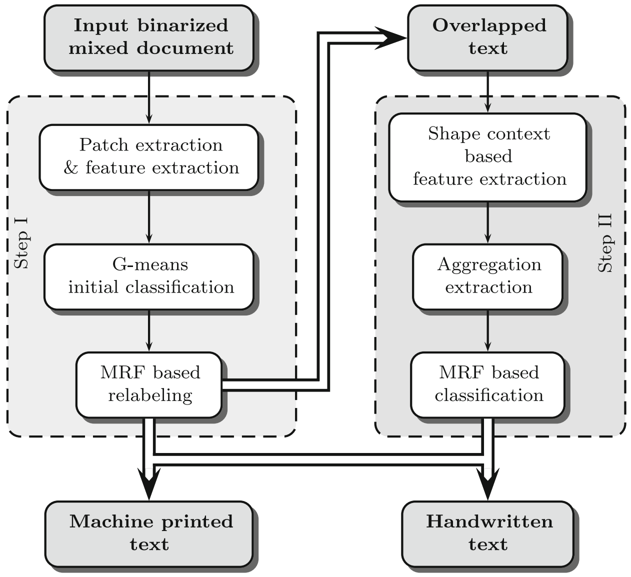

Input binarized mixed document
Patch extraction & feature extraction
G-means initial classification
MRF based relabeling
Step I
Overlapped text
Shape context based feature extraction
Aggregation extraction
MRF based classification
Step II
Machine printed text
Handwritten text

Fig. 2 Overall structure of proposed method

coarsening algorithm which extracts the basic element for the overlapped text separation. Sections 5.1 and 5.2 describe these two sub-procedures. The last sub-procedure is the separation of machine printed text and handwritten text using a MRF model and is described in Sect. 5.3. Section 5.4 shows the experimental setup and results of overlapped text separation.

Prior to presenting the details of each step of the proposed method, we briefly provide a background of MRF in Sect. 3 and our conclusions are presented in Sect. 6

3 Background—Markov Random Field

The Markov Random Field (MRF) is a graphical model in which a set of random variables have a Markov property described by an undirected graph and "*provides a convenient and consistent way of modeling context-dependent entities such as image pixels and correlated features*" [25]. MRFs are widely used in the field of image processing and document analysis, including image binarization [21,22], document restoration and image super-resolution [18,33], image and document segmentation [5], noise removal and filtering [3], etc. These applications can be categorized as the low-level processing. The high-level processing in image processing and vision field includes object detection and recognition. Li proposed a general framework of using MRFs for a computer vision problem by converting the high-level vision problem to a low-level labeling problem [24]. He used this idea to match objects in [23]. To detect objects in the image, Sheikh designed a two-step algorithm which used the probability density to estimate the object, and refined the final result by using the MAP-MRF decision framework [34].

From the view of mathematics, the MRF can be considered as a labeling procedure which is a proper model for the purpose of the document text separation (labeling).

We can use MRF to model the entire document in step I or overlapped text in step II and to separate patches (defined in Sect. 4.1) in step I or aggregations (defined in Sect. 5.2) in step II. We assume that the the foreground of a document/overlapped text image I is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v : 1 \leq v \leq N\}$ is the set of vertices which correspond to patches or aggregations in the image. $\mathcal{E}$ is the set of edges which connect vertices based on a neighbor system. The neighbor system can be based on four-neighbors lattice connectivity or any other metric defined by the user.

Given the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a random field $\mathcal{X} = x_1, \ldots, x_N$ which is indexed by $\mathcal{V}$ can be formed where each vertex takes a random variable x_v from a set of possible configurations Ω as its underlying label. In this paper, the configurations are taken from a set of cluster centers which are described in the following sections in details. According to these configurations (cluster centers), we can easily map a hidden random variable x_v to the corresponding index as its label. Let $\mathcal{Y} = y_1, \ldots, y_N$ be the observations which are the corresponding features for each vertex. An optimal configuration of the random field $\mathcal{X}$ is computed by maximizing the posterior:

$$\hat{\mathcal{X}} = \arg\max_{\mathcal{X}} P(\mathcal{X} \mid \mathcal{Y}) = \arg\max_{\mathcal{X}} \prod_{v=1}^{N} P(x_v \mid y_v, \mathcal{X}_{\mathcal{V}-v}) \tag{1}$$

where y_v and x_v are the observed feature and hidden configuration respectively for a given vertex v, and $\mathcal{V} - v$ is the set of all vertices in graph $\mathcal{G}$ except vertex v. This equation shows that the configuration of site v is dependent on its observation and all other vertices in the graph.

By taking the Bayes rule and the Markov property, Eq. 1 can be rewritten as:

$$\hat{\mathcal{X}} = \arg\max_{\mathcal{X}} \prod_{v=1}^{N} \frac{P(y_v \mid x_v, \mathcal{X}_{\mathcal{V}-v}) P(x_v \mid \mathcal{X}_{\mathcal{V}-v})}{P(y_v \mid \mathcal{X}_{\mathcal{V}-v})} \tag{2}$$

Since the denominator $P(y_v \mid \mathcal{X}_{\mathcal{V}-v})$ is just a normalizing constant and remains unchanged, we can disregard it and have the following equation:

$$\begin{aligned}\hat{\mathcal{X}} &= \arg\max_{\mathcal{X}} \prod_{v=1}^{N} P(y_v \mid x_v, \mathcal{X}_{\mathcal{V}-v}) P(x_v \mid \mathcal{X}_{\mathcal{V}-v}) \\ &= \arg\max_{\mathcal{X}} \prod_{v=1}^{N} P(y_v \mid x_v) P(x_v \mid \mathcal{X}_{\mathcal{N}(v)})\end{aligned} \tag{3}$$

where prior $P(x_v \mid \mathcal{X}_{\mathcal{N}(v)})$ means the configuration of site v is conditioned by its immediate neighbors $\mathcal{N}(v)$. Likelihood

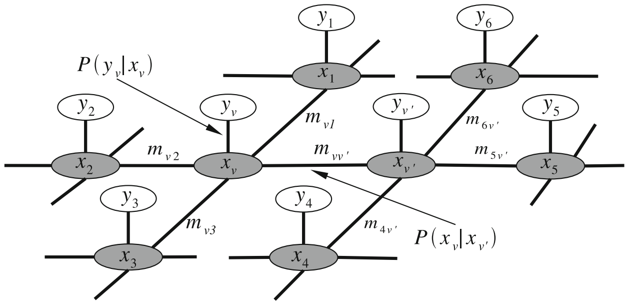


Fig. 3 Message update and belief propagation in MRF. Each *grey* node ν which is a configuration for a text patch/aggregation connects to its four nearest neighbors $\mathcal{N}(\nu)$. It has associated a configuration, x_ν, for a text patch/aggregation, and an observation, y_ν, which is a feature vector corresponding to its hidden lable x_ν [29]

$P(y_\nu \mid x_\nu)$ describes the relation of a configuration and its corresponding observation for a given site ν. Equation 3 shows that the configuration of site ν is dependent on its observation and immediate neighbors only.

Inference of the MRF model can be achieved using belief propagation (BP) method. We use the topology as shown in Fig. 3 to illustrate the belief propagation and message update procedure for our MRF model. The messages in a network propagate in two opposite directions [13,28,37] which lead to two update rules for vertex ν.

1. The procedure for calculating *maximum a posteriori probability* (MAP) of belief can be expressed as:

$$x'_\nu = \arg\max_{x_\nu} P(y_\nu \mid x_\nu) \prod_{\nu' \in \mathcal{N}(\nu)} m_{\nu,\nu'}(x_\nu),\ 1 \le \nu \le N \tag{4}$$

   where $m_{\nu,\nu'}(x_\nu)$ is the incoming message to vertex ν from its neighbor ν' and ν' runs over all neighbors of vertex ν.
2. The method to update the message from vertex ν' to vertex ν is given by:

$$m_{\nu,\nu'}(x'_\nu) = \max_{x'_{\nu'}} P(x'_\nu \mid x'_{\nu'}) P(y_{\nu'} \mid x'_{\nu'}) \prod_{\nu'' \in \mathcal{N}(\nu') - \nu} m_{\nu',\nu''}(x'_{\nu'}) \tag{5}$$

   where ν'' runs over all neighbors of vertex ν' except vertex ν.

The prior $P(x_\nu \mid x_{\nu'})$ and likelihood $P(y_\nu \mid x_\nu)$ in Eqs. 4 and 5 are approximated as a similarity function and dependency function respectively in our MRF model and are described in detail in Sect. 4.3.2 for step I and Sect. 5.3.2 for step II.

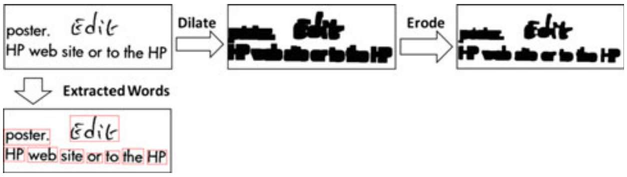


Fig. 4 The procedure to extract patches from a binarized document

4 Patch level text separation (step I)

The first step of our proposed method consists of three components: (i) preprocessing which segments entire document into patches and extracts features for each patch, (ii) G-means based initial classification and (iii) MRF based relabeling.

4.1 Preprocessing

Our pre-processing consists of two steps: patch extraction and feature extraction.

4.1.1 Patch extraction

Prior to classification, each binarized document is segmented into patches which are small snippets of the image as described in [29]. The patches are the basic units in our MRF based classification system for patch level text separation which models each document as a random field. The extraction of patches is done by using a $m \times n$ (5×5 in our experiments) sized window based morphology closing operation on the original binarized image and the original content within the bounding box of each connected component is defined as a single patch. The size of the window is empirically chosen such that the resultant patch typically represents a handwritten or machine-printed word. Patches are eliminated as noise if their size is smaller than a threshold t_l or larger than a threshold t_h.

The procedure of patch extraction for a binarized document is shown in Fig 4.

4.1.2 Feature extraction

Three different categories of features are considered for classification of a given patch into one of three classes viz., handwritten text, machine-printed text and overlapped text. These features were introduced briefly in [29] and are described in greater detail in the following sections.

- *Patch level features*
  Considering each patch as a single unit in our system, we extract several sets of features at the patch level. For a given patch, the first feature is its relative location i and j with respect to the entire document. The relative width

w and height h of the patch with respect to its nearest neighbor measures the size of this patch. The foreground density of the patch is measured by equation:

$$d = \frac{\sum_{i,j} I(i,j)}{w \cdot h} \tag{6}$$

where $I(i,j)$ is the density of pixel (i,j) which represents black pixel as 0 and white pixel as 1 in our binary image.

Generally, machine printed text and handwritten text individually have constant stroke width. Therefore, we calculate the average stroke width using:

$$s = \frac{\sum_{i,j} I(i,j)}{l} \tag{7}$$

where l is the length of contour for a given patch. The crossing number within a patch measures the complexity of the stroke [29,39] which counts the amount of pixels whose density differs from its direct neighbors in horizontal and vertical direction and is defined as:

$$\begin{aligned} f_h &= \frac{\sum_{i,j} I(i,j) \oplus I(i+1,j)}{h} \\ f_v &= \frac{\sum_{i,j} I(i,j) \oplus I(i,j+1)}{w} \end{aligned} \tag{8}$$

where $\oplus$ is the *exclusive or* operator.

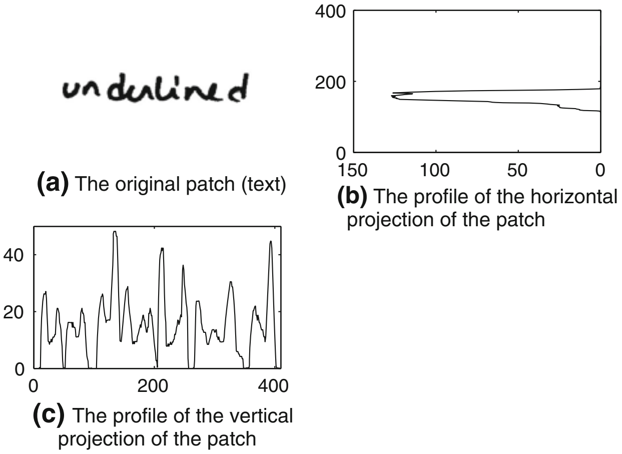
The standard deviation of horizontal and vertical projection of a patch and maximum runlength within the patch in these two directions are also used to measure the difference between the three different classes. Figures 5 and 6 illustrate the process of computing these two sets of features.

Total number of 12 patch level features are extracted for each patch.

– *Connected component (CC) features*

A set of features is based on the statistical properties of connected components within a patch which is extracted from the original (non-dilated) binarized image.

Firstly, we compute the mean and standard deviation of width and height of CCs within a patch. Normally, machine printed character has a constant width and height in a certain range which results in a small value of variance. On the contrary, the width and height of the connected component within handwritten text are more likely to be scattered in the feature space and have larger variance value. The maximum width and height of CCs are also used as features.



(a) The original patch (text)

(b) The profile of the horizontal projection of the patch

(c) The profile of the vertical projection of the patch

Fig. 5 The projection of a patch

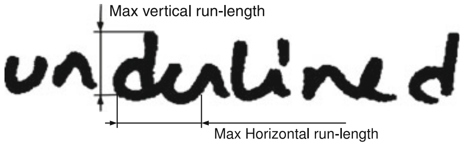


Fig. 6 The vertical and horizontal maximum run-length of a patch

For a handwritten text patch, the area of holes within characters is typically larger than the area of holes for characters in a machine printed patch. So the hole area normalized by the size of the patch is computed as a feature to distinguish handwritten text from the other two kinds of patches. Another CC feature is the overlap ratio which is the area of overlap between the bounding box of CCs within the patch divided by the total area of the patch [29,39].

To each patch, nine connected component based features are extracted.

– *Gabor features*

Gabor filters can serve as directional band-pass filters which are modulations of a complex sinusoidal and Gaussian function. The 2-D Gabor filter is defined as Eq. 9 in the space domain (details of parameters can be found in [16,31,32]):

$$\begin{aligned} & g_{\lambda,\theta,\varphi,\delta,\gamma}(i,j) \\ & \quad = K \exp\left(-\frac{i'^2 + \gamma^2 j'^2}{2\delta^2}\right) \cos\left(2\pi \frac{i'}{\lambda} + \varphi\right) \end{aligned} \tag{9}$$

where $i' = i\cos\theta + j\sin\theta$, $j' = -i\sin\theta + j\cos\theta$ and λ is the wavelength of the cosine factor of the Gabor filter kernel, θ is the orientation, φ is the phase offset, γ is the aspect ratio and δ is the squared deviation of the Gaussian function. A set of different θ and λ lead to the eight gabor filters used in our experiments which are shown in Fig. 7. The eight gabor features are obtained by calculating the standard deviation of each gabor filtered image (patch).

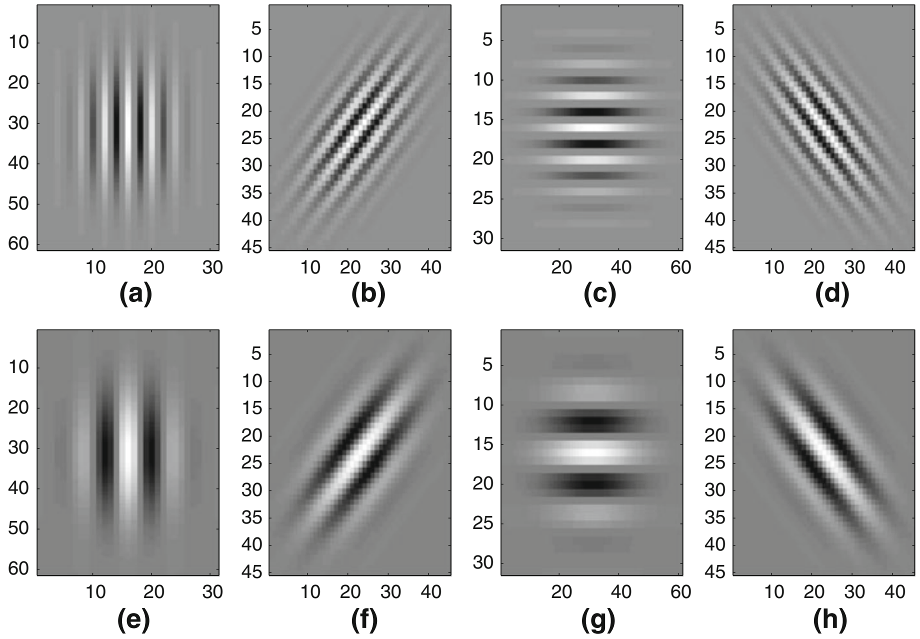


Fig. 7 A set of Gabor filters. **a–d** Gabor filters with $\lambda = 4$ and $\theta = \{0°, 45°, 90°, 134°\}$, **e–h** Gabor filters with $\lambda = 8$ and $\theta = \{0°, 45°, 90°, 134°\}$

4.2 G-means based training and classification

Training on the three different kinds of patches is carried out using a modified K-means clustering algorithm known as G-means [19].

Unlike normal K-means which is widely used in clustering methods but where k has to be determined in advance, G-means estimates the number of clusters based on the distribution of the training data. The underlying principle of G-Means is to split the training set using K-means with $k = 2$, and if any sub-cluster does not have a Gaussian distribution, then K-means with $k = 2$ is applied again for this sub-cluster until each cluster has a Gaussian distribution. Further details of G-means can be found in [19].

In our training phase, we run G-means clustering algorithm for machine printed patches, handwriting patches and overlapped patches individually to get three sets of clusters and corresponding centers which construct the configuration set Ω of the MRF. The total number of centers is M.

For each cluster m, we calculate the co-variance:

$$\Sigma_m = \frac{1}{K_m}\sum_{k=1}^{K_m}(y_k - c_m)(y_k - c_m)^T, \ 0 \le m < M \qquad (10)$$

where K_m is the number of feature vectors within the cluster, y_k is a feature vector belonging to this cluster and c_m is the center of the cluster.

During the patch classification phase (i.e., the labeling initialization process), for each test feature vector y which is an observation in the MRF framework, we find the nearest cluster to this feature point using Mahalanobis distance D_y which is defined in Eq. 11 and maps the test feature point to the center of this cluster. The index of this cluster is initially assigned as the label of the corresponding vertex ν of the graph $\mathcal{G}$:

$$\begin{aligned}\hat{m} &= \arg\min_{0\le m<M} D_y(y_\nu, c_m)\\ &= \arg\min_{0\le m<M} \sqrt{(y_\nu - c_m)^T \sum\nolimits_m^{-1} (y_\nu - c_m)}\end{aligned} \qquad (11)$$

This G-means based classification can be looked at as a nearest neighbor search (NNS) with a reduced search space. The label of the test feature point can be further mapped to one of the three classes because the class to which each center belongs (handwritten text, machine printed text or overlapped text) is already known during the training phase.

From the viewpoint of the MRF, our G-means based initial classification is a vector quantization (VQ) procedure which constructs the configuration set Ω for the MRF as described in Sect. 3.

4.3 MRF based relabeling

In practice, misclassification cannot be avoided using a single classifier due to overlaps in the feature space. Therefore, postprocessing or relabeling is needed. The intuition for relabeling is that a patch surrounded by patches from a single different class has a high probability of belonging to that class. We use a MRF which describes the statistical dependency between observed patch features and their hidden states (label) to model different patches in an annotated machine printed document and then relabel the patches in our scenario.

In step I of the proposed method where we do word (patch) level text separation, we model the annotated machine printed document I as the graph $\mathcal{G}$ defined in Sect. 3. Each vertex in set $\mathcal{V}$ represents a patch within the document and the set of edges $\mathcal{E}$ connect vertices based on a neighbor system described in Sect. 4.3.1. The random field $\mathcal{X}$ defined in Sect. 3 is consistent with the graph $\mathcal{G}$ and we let the set of centers Ω which is extracted using G-means from Sect. 4.2 to be the set of all possible configurations of $\mathcal{X}$, and the patch features be the observations $\mathcal{Y}$.

As described in the previous Sect. 3, the inference of the MRF model is implemented by using belief propagation algorithm which is shown in Fig. 3. Each grey node x_ν in the hidden layer exclusively corresponds to a hidden configuration for a document patch and is assigned to a label $m(0 \leq m < M)$ after initial classification. Each white node y_ν is a observed feature point for that patch. In real documents, the neighbor system of patches is determined by a convex-hull based distance metric as defined in Sect. 4.3.1 but may not necessarily be located on a grid as shown in Fig. 3.

4.3.1 Definition of neighbor system

As shown in Eqs. 4 and 5, we need to update the message and belief in the network based on a neighbor system to compute the optimal configuration for MRF $\mathcal{X}$. Normally, the neighbor system of a MRF is based on a four-neighbors lattice connectivity. However, if we consider each patch in the document as a single vertex in the graph $\mathcal{G}$ which may not be rigidly located on a grid, we need to define a flexible neighbor system to represent the spatial relationship between patches.

Firstly, we define a distance metric to measure the spatial distance between each pair of patches in a document:

$$D_x(\nu, \nu') = \frac{(d_v(\nu, \nu') - \hat{v})^2}{2\hat{v}^2} + \frac{(d_h(\nu, \nu') - \hat{h})^2}{2\hat{h}^2} \qquad (12)$$

where $[d_h(\nu, \nu'), d_v(\nu, \nu')]$ represent the convex-hull distance between patches ν and ν' in the horizontal and vertical directions, $\hat{v}$ is the dominant gap between words and $\hat{h}$ is the dominant gap between text lines over the entire document. Dominant gaps $\hat{v}$ and $\hat{h}$ can be estimated using histograms. Based on spatial distance, the four closest neighbors are considered for each patch. The bottom part of Fig. 8 shows the four nearest neighbors (which are represented by the four black rectangles) of the patch contained in the red rectangle.

By using the convex-hull distance metric based neighbor system, we can measure the similarity between patches in spatial space by taking the distance as the variable of an exponential function. In other words, the similarity to a given patch decreases exponentially as the distance increases and only patches which have the greatest similarity are considered as the neighbors of the center patch. The plot at the top of Fig. 8 shows the decrease of similarity with distance from the center patch.

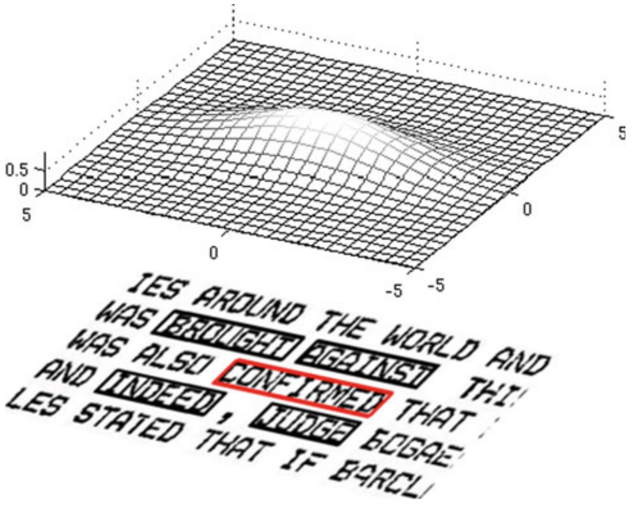


Fig. 8 A patch and its nearest neighbors

4.3.2 Prior and likelihood

In order to compute an optimal configuration which maximizes the posterior as described in Eq. 1, we use belief propagation which calculates the local maximum messages and beliefs for each vertex to achieve global maximum. To use belief propagation for a given vertex ν, the prior $P(x_\nu \mid x_{\nu'})$ and likelihood $P(y_\nu \mid x_\nu)$ in Eqs. 4 and 5 are modeled by a similarity function which measures the similarity of configuration between hidden node x_ν and $x_{\nu'}$ and a dependency function which describes the influence of the observation y_ν on its hidden node x_ν.

The prior $P(x_\nu \mid x_{\nu'})$ is approximated by:

$$P(x_\nu \mid x_{\nu'}) \approx \mathcal{L}(\alpha \mathrm{e}^{-D_x(\nu,\nu')} + \beta \mathrm{e}^{-D_\mathrm{e}(x_\nu, x_{\nu'})}) \qquad (13)$$

where $D_x(\nu, \nu')$ is the distance between two neighboring patches (which corresponds to vertex ν and ν' in graph $\mathcal{G}$) calculated from Eq. 12. $D_\mathrm{e}(x_\nu, x_{\nu'})$ is the Euclidean distance between the configurations of two patches which represent the assigned centers in the feature space as described in Sect. 4.2. α and β are two parameters that control the influence from neighbors in the spatial space and the feature space respectively (We set $\alpha = 0.1$ and $\beta \in [1.0, 1.5]$ in our experiment). The final prior is normalized with a linear transformation $\mathcal{L}$ so that $P(x_\nu \mid x_{\nu'}) \in [0, 1]$.

Equation 13 illustrates that the influence from the neighbors of a patch depends not only on their spatial distance but also on the distance of the two configurations in feature space. So, in our system, belief propagation is encouraged for those patches which are close to each other in both the spatial as well as the feature space.

Similarly, the likelihood $P(y_\nu \mid x_\nu)$ is approximated by:

$$P(y_\nu \mid x_\nu) \approx \mathcal{L}'(\mathrm{e}^{1/(\lambda D_y(x_\nu, y_\nu))}) \tag{14}$$

where $D_y(x_\nu, y_\nu)$ is the Mahalanobis distance calculated from Eq. 11, which measures the distance from the observation y_ν of vertex ν to its assigned center x_ν in feature space. Parameter λ controls the influence of observations and their hidden states (The final parameter of λ we chose in our experiment is $\lambda \in [0.1, 1.0]$). The final likelihood is normalized to the range between 0 and 1 using a linear mapping $\mathcal{L}'$.

Equation 14 shows the relationship between the configuration of a given patch and its corresponding observed features.

Optimal configuration for MRF is achieved by applying BP algorithm as shown in Eqs. 4 and 5. All messages $m_{\nu,\nu'}$ in these two equations are initially set to 1. During the inference, the prior $P(x_\nu \mid x_{\nu'})$ and the likelihood $P(y_\nu \mid x_\nu)$ are calculated according to Eqs. 13 and 14. The updated message $m_{\nu,\nu'}$ is passed from node ν' to node ν and updating is terminated until there is no flipping occurring during belief propagation or when maximum iteration is achieved. More details of belief propagation can be found in [13].

4.4 Experimental results for patch level text separation

Experiments were conducted on data sets generated at HP Labs to train and test for patch level text separation (step I). *Precision* (P) and *recall* (R) metrics were used to estimate the performance of the MRF based relabeling system and compare the proposed method with other classifiers. To any one of three classes, the precision and recall are defined by:

$$P(k) = \frac{T_{\mathrm{P}}(k)}{T_{\mathrm{P}}(k) + F_{\mathrm{P}}(k)} \tag{15}$$

$$R(k) = \frac{T_{\mathrm{P}}(k)}{T_{\mathrm{P}}(k) + F_{\mathrm{N}}(k)} \tag{16}$$

where $T_{\mathrm{P}}(k)$ is the true positive which counts the number of correctly classified patches of class k by the system, $F_{\mathrm{P}}(k)$ is the false positive which counts the number of patches misclassified as class k by the system, and $F_{\mathrm{N}}(k)$ is the false negative which measures the number of patches classified as other classes than k but belonging to class k.

The HP Labs data set consists of binarized images of annotated office documents scanned at a resolution of 300 dpi. 82 documents from the HP Labs data set were used for these experiments where the fonts of machine printed text in documents were mainly *Calibri* and *Times New Roman* with the size of 10–12 points. The handwritten text (annotations) which included arrows, brackets, circles, cross-lines, signatures and comments were annotated by 10 different writers. The data in the documents can be broadly classified into three classes—machine printed text, handwritten text, and overlapping text. This extremely imbalanced data set contains over 25,000 machine printed text patches, around 3,200 handwritten text patches and less than 400 overlapped text patches. We used N-fold cross-validation technique to estimate the classification accuracy for patch level text separation where the data set was exclusively divided into N subsets ($N = 3$ in our experiment) and one subset was used for testing and the remaining subsets were used for training iteratively.

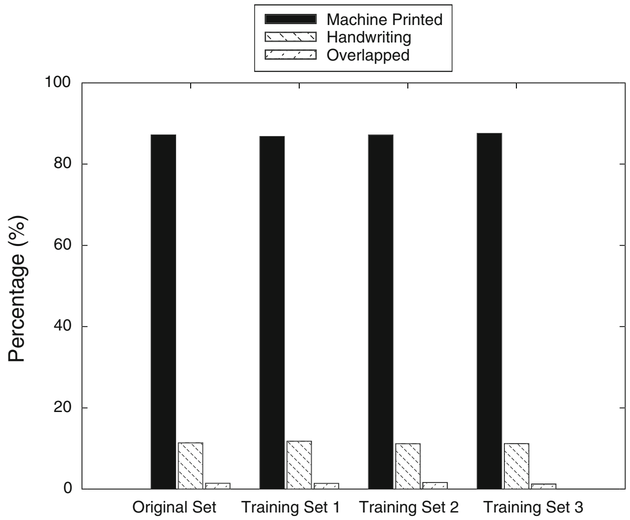


Fig. 9 Distributions of different type of text for training

In Fig. 9, we showed distributions of different type of text for the entire data set and training sets of three splits. We can see that three training sets have the similar distribution to the entire data set. Thus, the evaluation on each fold predicts the classification capability on the entire data set.

Prior to classification, a few preprocessing steps were performed. A morphological closing operation with a 5 × 5 window was applied and patches were extracted. This was followed by feature extraction. The G-means based clustering algorithm described in Sect. 4.2 was used to extract centers for the machine printed patches, handwritten patches and overlapped patches on the training data set individually and to construct our configuration set $\mathcal{X}$. The initial classification was carried out by assigning each test patch to the nearest center in the feature space eventually mapping them to one of the three classes. The MRF based relabeling procedure was used to correct the misclassification for all patches as described in Sect. 4.3.

In Table 1, we compared the proposed method to a G-means based classifier and a backpropagation neural network classifier which was implemented using a modified public ANN tool, FANN [1]. The BP neural network with two hidden layers used the sigmoid function for each neuron in the hidden layer and the sigmoid symmetric function for output the layer. The overall accuracy in the table is defined as:

Table 1 Performance of patch level text separation (classification into handwritten, machine print and overlapped text) on the HP data set

	BP neural network		G-means		Proposed method	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Machine-printed	99.36	92.16	99.43	94.63	99.49	95.73
Handwritten	67.06	89.72	74.16	92.16	79.11	93.85
Overlapped	32.97	80.97	40.44	81.55	46.24	83.10
	Accuracy (%)		Accuracy (%)		Accuracy (%)	
Overall	91.72		94.16		95.52	

Table 2 Error reduction rates of patch level text separation

	Overall number of misclassified patches			Error reduction rates	
	BP neural network	G-means	Proposed method (MRF)	ϵ_r (%)	ϵ_g (%)
Machine-printed	572	396	316	44.6	20.1
Handwritten	99	76	67	32.3	11.8
Overlapped	23	23	21	8.7	8.7

the number of all patches correctly classified by the system divided by the total number of patches in the test set. We see that the MRF based relabeling method had an overall accuracy of 95.52% which outperformed the backpropagation neural network (91.72%) as well as the G-means based classifier (94.16%). The precision and recall, especially for the two minor classes (handwritten text and overlapped text), were also increased by using the MRF based labeling method.

In Table 2, we showed the "Error Reduction Rates" which were defined as:

$$\epsilon_r(k) = \frac{F_{Nr}(k) - F_{Nt}(k)}{F_{Nr}(k)} \tag{17}$$

$$\epsilon_g(k) = \frac{F_{Ng}(k) - F_{Nt}(k)}{F_{Nr}(k)} \tag{18}$$

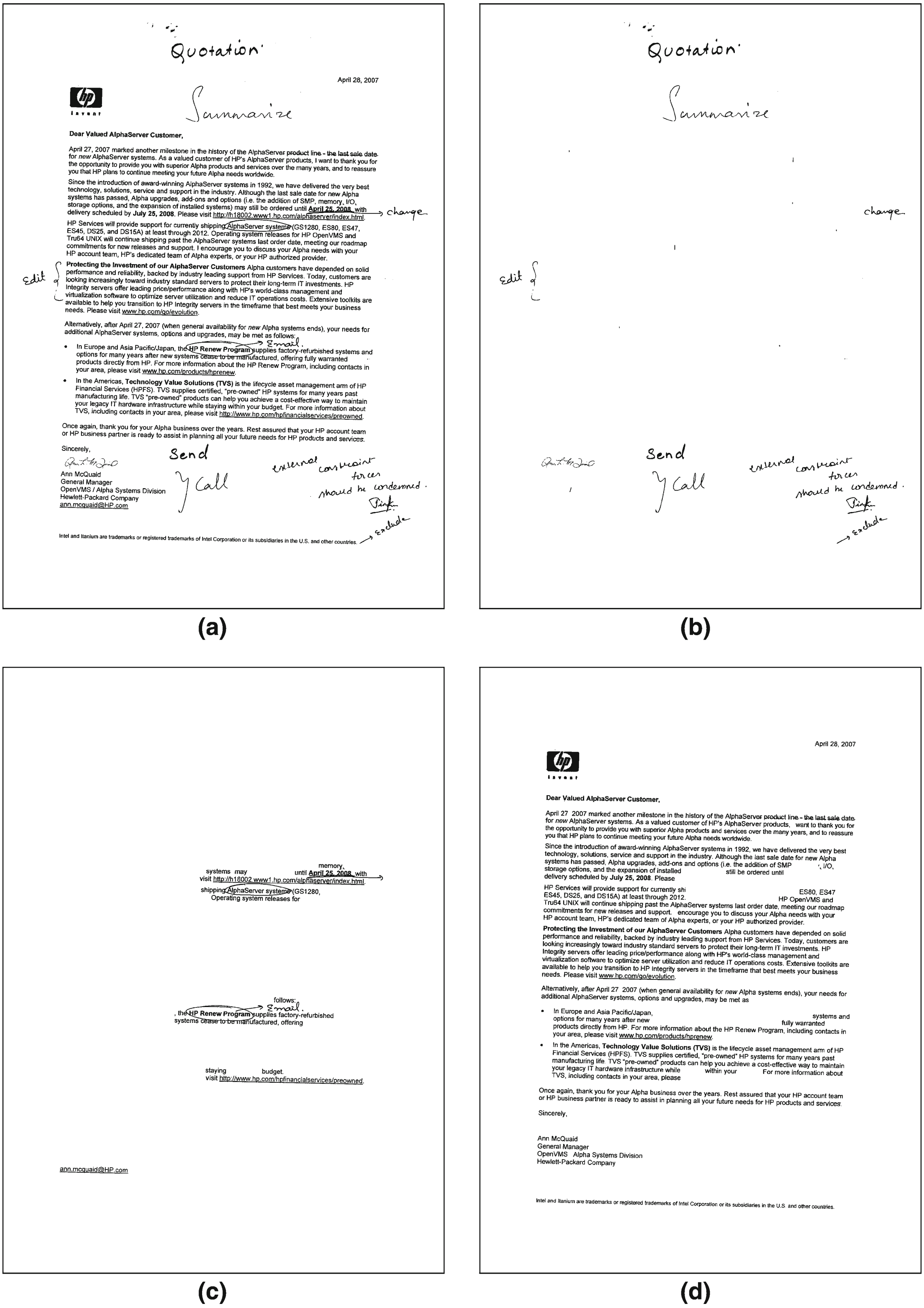
where ϵ_r denotes the reduction rate of misclassified patches between Neural network and the MRF based method for a given type of text. $F_{Nr}(k)$ is the false negative of class k by using Neural network and $F_{Nt}(k)$ counts the false negative by using MRF based method. Similarly, ϵ_g calculates the error reduction rate between G-means based method and the proposed method. As shown in the table, by using the MRF based method, the error rate reduced to around 45 and 20% for machine printed text compared with Neural network and G-means based method. The error reduction rates for handwritten text and overlapped text were also decreased effectively in the table.

Figure 10 shows results of the MRF based text identification on a sample image from the HP Labs data set where the original binarized document (Fig. 10a) is decomposed to handwritten text document (Fig. 10b), overlapped text document (Fig. 10c) and machine printed text document (Fig. 10d).

To illustrate the effectiveness of the MRF based labeling, we showed three example results of machine printed text extracted by different methods. As shown in Fig. 11, BP Neural network based classification algorithm only considered the features of every single patch. Thus some machine printed text were misclassified as other type of text, such as word "Screen Size", "Board" and "Laptop", even these words were among the context of machine printed text. Due to the smoothness effect, MRF based method obtained better classification performance than Neural network and G-Means based method, especially in the case of that a large amount neighboring text were from the same class.

In our experiment, the patches which contained several touched words by underline, rule-line or noises were labeled as machine printed but were typically classified by the system as overlapped patches as shown in the Fig. 10c. The machine printed text were easily classified as overlapped text in our system either if the size of text was big or the font of text was bold. Most misclassifications for handwritten patches and overlapped patches happened in the case of isolated handwritten patch or overlapped patch surrounded by machine printed patches which was smoothed by MRF.

The relatively low precision for handwritten text and overlapped text is mainly due to the imbalanced nature of the HP Labs data set. The amount of machine printed text is much more than the handwritten annotations and overlapped text and dominates in the data set. If even a small proportion of the machine printed text is misclassified as one of the other two classes, it is still a significant number when compared to the handwritten or overlapped text samples and



Quotation

April 28, 2007

invent

Summarize

Dear Valued AlphaServer Customer,

April 27, 2007 marked another milestone in the history of the AlphaServer product line - the last sale date for *new* AlphaServer systems. As a valued customer of HP's AlphaServer products, I want to thank you for the opportunity to provide you with superior Alpha products and services over the many years, and to reassure you that HP plans to continue meeting your future Alpha needs worldwide.

Since the introduction of award-winning AlphaServer systems in 1992, we have delivered the very best technology, solutions, service and support in the industry. Although the last sale date for new Alpha systems has passed, Alpha upgrades, add-ons and options (i.e. the addition of SMP, memory, I/O, storage options, and the expansion of installed systems) may still be ordered until **April 25, 2008**, with delivery scheduled by **July 25, 2008**. Please visit http://h18002.www1.hp.com/alphaserver/index.html. → change

HP Services will provide support for currently shipping AlphaServer systems (GS1280, ES80, ES47, ES45, DS25, and DS15A) at least through 2012. Operating system releases for HP OpenVMS and Tru64 UNIX will continue shipping past the AlphaServer systems last order date, meeting our roadmap commitments for new releases and support. I encourage you to discuss your Alpha needs with your HP account team, HP's dedicated team of Alpha experts, or your HP authorized provider.

Edit

Protecting the Investment of our AlphaServer Customers Alpha customers have depended on solid performance and reliability, backed by industry leading support from HP Services. Today, customers are looking increasingly toward industry standard servers to protect their long-term IT investments. HP Integrity servers offer leading price/performance along with HP's world-class management and virtualization software to optimize server utilization and reduce IT operations costs. Extensive toolkits are available to help you transition to HP Integrity servers in the timeframe that best meets your business needs. Please visit www.hp.com/go/evolution.

Alternatively, after April 27, 2007 (when general availability for *new* Alpha systems ends), your needs for additional AlphaServer systems, options and upgrades, may be met as follows:

→ Email.

- In Europe and Asia Pacific/Japan, the **HP Renew Program** supplies factory-refurbished systems and options for many years after new systems ~~cease to be~~ manufactured, offering fully warranted products directly from HP. For more information about the HP Renew Program, including contacts in your area, please visit www.hp.com/products/hprenew.
- In the Americas, **Technology Value Solutions (TVS)** is the lifecycle asset management arm of HP Financial Services (HPFS). TVS supplies certified, "pre-owned" HP systems for many years past manufacturing life. TVS "pre-owned" products can help you achieve a cost-effective way to maintain your legacy IT hardware infrastructure while staying within your budget. For more information about TVS, including contacts in your area, please visit http://www.hp.com/hpfinancialservices/preowned.

Once again, thank you for your Alpha business over the years. Rest assured that your HP account team or HP business partner is ready to assist in planning all your future needs for HP products and services.

Sincerely,

Send

Call

external constraint forces should be condemned.

Ann McQuaid
General Manager
OpenVMS / Alpha Systems Division
Hewlett-Packard Company
ann.mcquaid@HP.com

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the U.S. and other countries. → exclude

(a)

(b)

(c)

(d)

Fig. 10 Experiment results on an example document from the HP data set: **a** the original binarized document, **b** handwritten text, **c** overlapped text, and **d** machine printed text

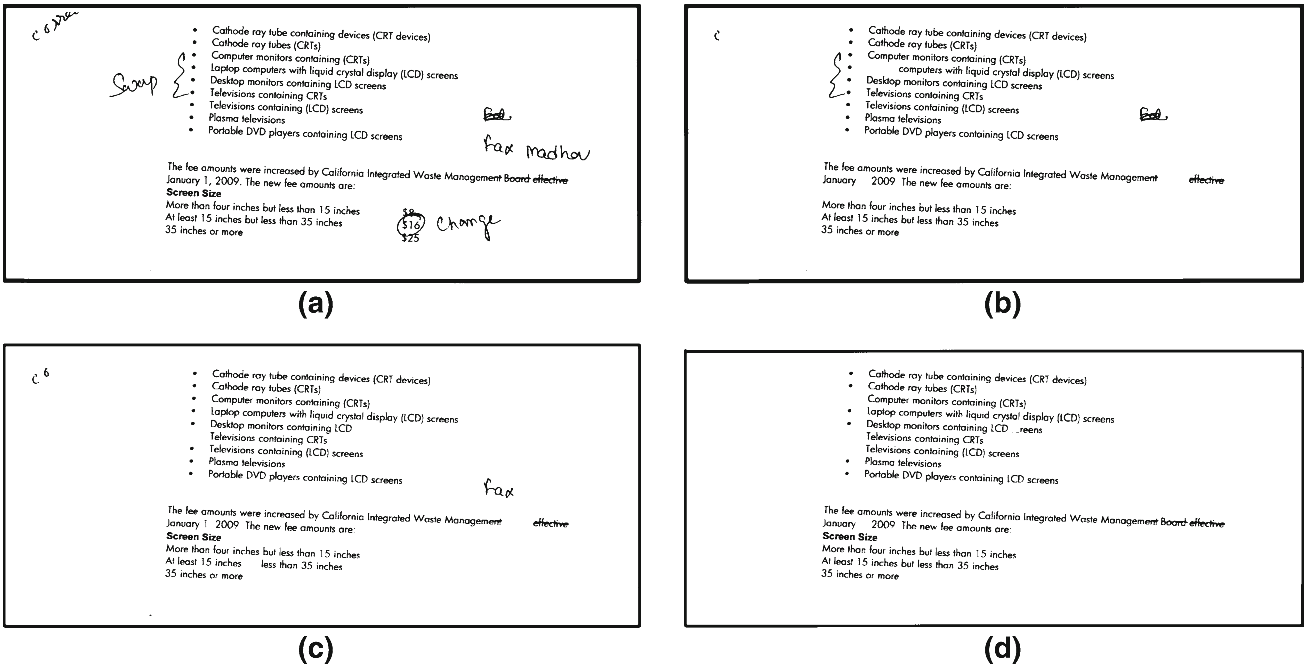


Fig. 11 Machine printed text extracted using different methods: **a** the original binarized document, **b** machine printed text extracted by FANN, **c** machine printed text extracted by G-means, and **d** machine printed text extracted by the proposed method

hence leads to low precision. A tree-structured initial classifier could perhaps be used to overcome the imbalanced data set problem and we propose to test this in future work.

5 Pixel level text separation (step II)

As shown in Fig. 2, the second step of our proposed method takes as its input the overlapped text from step I. Since the overlapped text is considered as a single unit in step I, its basic element (patch) and the features for patches are not suitable to further segment the overlapped text into machine printed and handwritten text. In this section, we propose a method to decompose the overlapped text image into smaller units using a coarsening procedure, which aggregates foreground pixels of the overlapped text image according to their coherence properties [30]. Shape context, a pixel level feature, is used to measure the coherence or variance between pixels or aggregations. The MRF based segmentation of overlapped text is based on these small aggregations which are the basic element in step II.

5.1 Shape context features

Prior to coarsening pixels to aggregations, we first extract features at each pixel. We use shape context features to characterize each pixel or aggregation. Shape context features were first used by Belongie et al. [4] to compare the similarity between two shapes.

Given a shape which contains a set of points, we can draw vectors from a given point on this shape to each of the other points and the shape can be exactly represented by these vectors. Shape context features were inspired by this notion but instead of using vectors to characterize each point on the shape, shape context features tend to capture the coarse distribution of the rest of the points on a shape with respect to a given point.

To compute the shape context features $v_{i,j}$ for a foreground pixel whose coordinate is (i, j), a polar system is centered on this pixel which divides the entire image into several small bins. The feature $v_{i,j}$ is obtained by calculating the number of foreground pixels within each bin. More details of computing shape context features can be found in [4].

Figure 12 illustrates two examples of computing histograms for two different points on the same shape by centering the polar coordinate system on the points. The bottom part of each polar system shows the extracted shape context features where each colored rectangle corresponds to a bin in the polar system and darker color represents a larger value. In our experiments, we use three circles and eight quadrants to divide the overlapped text patch into 24 bins as shown in Fig. 12.

5.2 Aggregation coarsening

In order to separate overlapped text, we propose a coarsening procedure by which foreground pixels in the overlapped text

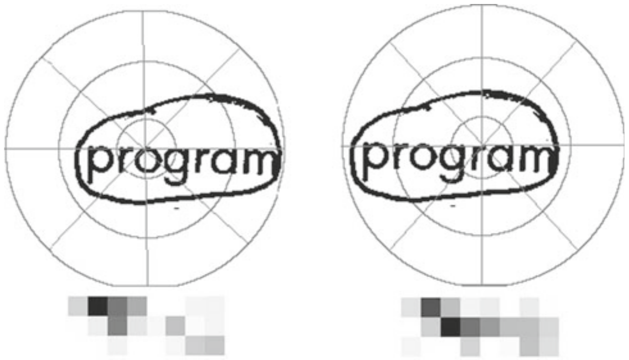


Fig. 12 The polar system and shape context corresponding to two different points

are grouped to aggregations, which have coherent attributes (similar feature for each foreground pixel within aggregation) and can be used as the basic element for training and classification.

We assume an overlapped text image I is modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex corresponds to an aggregation which contains a set of neighboring foreground pixels $\mathcal{P}_\nu = \{p_{i_1,j_1}, p_{i_2,j_2}, \ldots, p_{i_m,j_m}\}$ from the image, and edges $\mathcal{E}$ connect vertices based on four-neighbors lattice connectivity.

For each vertex ν, we measure its coherence by using the mean μ_ν and variance δ_ν of shape context features within the corresponding aggregation.

Then we define the diversity between two adjacent vertices ν and ν' as the difference of their mean values:

$$\eta_{\nu,\nu'} = \mu_\nu - \mu_{\nu'} \tag{19}$$

By using variance δ_ν within a vertex ν and difference $\eta_{\nu,\nu'}$ between two adjacent vertices ν and ν', we define a criterion as presented in Eq. 20 to determine which neighboring vertex needs to be merged with a given vertex ν to create a new single vertex (aggregation).

$$\hat{\nu'} = \arg \max_{\nu' \in N(\nu)} \sum_{\nu'' \in N(\nu)-\nu'} \eta_{\nu,\nu''} \mathrm{e}^{-\delta_{\nu'''}} \tag{20}$$

where $\nu' \in N(\nu)$ represents the neighbors of vertex ν, $\nu'' \in N(\nu) - \nu'$ represents all neighbors of the vertex ν but excludes vertex ν', ν''' means a test vertex which is merged by vertex ν and ν', and $\delta_{\nu'''}$ is the variance of the features within this test new merged vertex (ν merged with ν').

The underlying principle of Eq. 20 is that a vertex ν is temporarily merged with its immediate neighbors ν' individually to create a set of candidate vertices $\{\nu'''\}$ whose variances are $\{\delta_{\nu'''}\}$. Then, the vertex $\hat{\nu'}$ whose average feature value is close to vertex ν's average feature value and minimizes the variance of the new merged vertex ν''' is the optimal vertex which should be merged with ν.

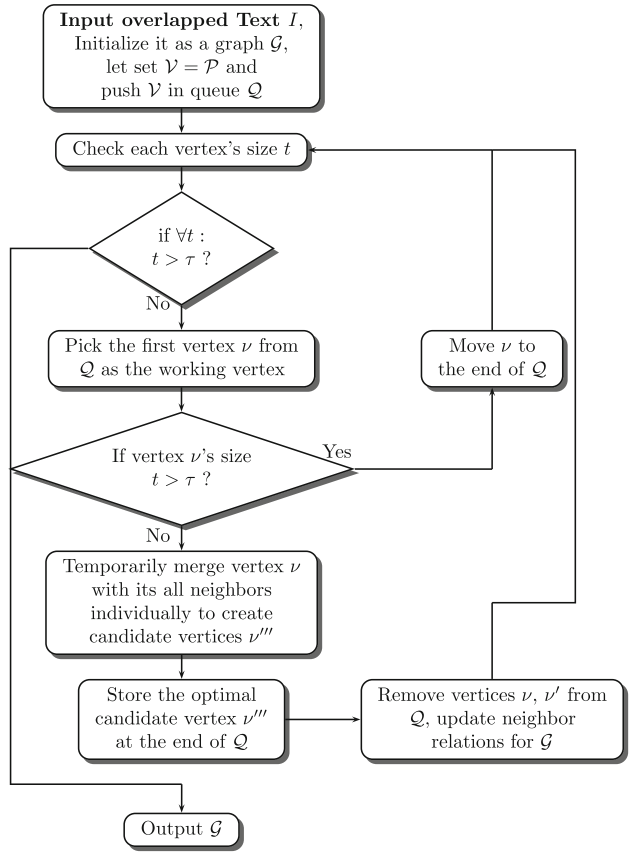

Input overlapped Text I, Initialize it as a graph $\mathcal{G}$, let set $\mathcal{V} = \mathcal{P}$ and push $\mathcal{V}$ in queue $\mathcal{Q}$
Check each vertex's size t
if $\forall t$: $t > \tau$?
No
Pick the first vertex ν from $\mathcal{Q}$ as the working vertex
Move ν to the end of $\mathcal{Q}$
If vertex ν's size $t > \tau$?
Yes
No
Temporarily merge vertex ν with its all neighbors individually to create candidate vertices ν'''
Store the optimal candidate vertex ν''' at the end of $\mathcal{Q}$
Remove vertices ν, ν' from $\mathcal{Q}$, update neighbor relations for $\mathcal{G}$
Output $\mathcal{G}$

Fig. 13 The coarsening procedure

Based on this criterion, we can extract aggregations starting from a single pixel to produce larger coherent coarsened regions. Figure 13 shows the procedure of coarsening. Initially, each aggregation for the vertex ν contains only one pixel and all vertices $\mathcal{V}$ in graph $\mathcal{G}$ are pushed in a queue $\mathcal{Q}$. Prior to executing the main loop of the coarsening procedure, the size t of all vertices is checked and the first vertex whose size is smaller than a pre-defined threshold τ is picked as the working vertex. The working vertex ν is merged with an optimal immediate neighbor according to Eq. 20 to produce a new vertex which is stored at the end of queue $\mathcal{Q}$ and its neighbors are inherited from its two parent vertices. The two merging vertices are removed from the queue $\mathcal{Q}$ and the neighbor system for graph $\mathcal{G}$ is also updated accordingly. The iteration of coarsening is stopped when every vertex's size is bigger than threshold τ which is set to 130 in our experiment.

Figure 14 shows an example of coarsening for an overlapped text image where machine printed text is circled and touched by handwriting. Figure 14a is the original binarized overlapped text image. Figure 14b shows the coarsening result after first iteration when every pixel is coarsened with its neighbor at least once. Figure 14c shows the aggregations after the third round of iteration and Fig. 14d is the final

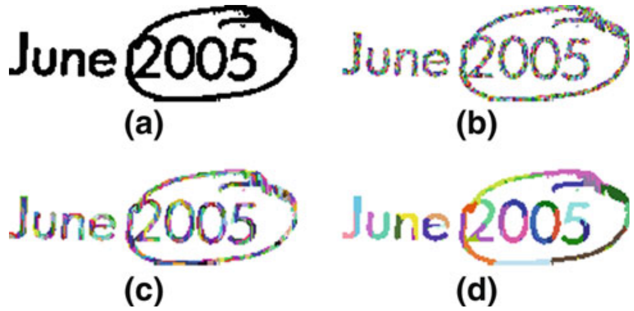


Fig. 14 Example of coarsening procedure and aggregation: **a** the original binarized text, **b** the coarsening result after one iteration, **c** the coarsening result after three iterations, **d** the final coarsening result. All aggregations are it colored randomly

coarsening result. The aggregations are randomly colored in these figures.

5.3 MRF based classification

5.3.1 Modeling overlapped text using MRF

Markov Random Field model is reused in step II to separate handwritten text from an overlapped text image. As shown in Sect. 3, a key step in the usage of MRF is the definition of the random field $\mathcal{X}$, the hidden configuration set Ω and the observations $\mathcal{Y}$. Unlike the definition of an MRF for the entire document in Sect. 4.3 where each vertex in the MRF corresponds to a text patch, the overlapped text image is modeled as a random field whose nodes represent aggregations extracted using the coarsening algorithm from Sect. 5.2. To build the configuration set Ω, G-means clustering algorithm is used as a vector quantization (VQ) procedure to separate the training set into several small clusters and the center of each cluster is used to build a codebook to represent the configuration set Ω. The observations $\mathcal{Y}$ for each vertex is obtained by normalizing each aggregation to the size of 16×16. The neighbor system for the MRF of step II is inherited from the coarsening algorithm where the neighbors of each vertex are its direct connected vertices based on the four-neighbors lattice connectivity relation.

5.3.2 Prior and likelihood

Similar to step I which uses Belief Propagation (BP) algorithm to obtain the optimal configuration $\mathcal{X}$ for MRF in Sect. 4.3, our MRF based classification for overlapped text also achieves the optimal configuration based on Eq. 2 but re-models the prior $P(x_\nu \mid x_{\nu'})$ and likelihood $P(y_\nu \mid x_\nu)$.

The prior $P(x_\nu \mid x_{\nu'})$ which is approximated by a similarity function in our MRF based overlapped text separation step is modeled as:

$$P(x_\nu \mid x_{\nu'}) \approx \mathcal{L}''(t\{\alpha f(x_\nu, x_{\nu'}) + \beta \mathrm{e}^{-D_\mathrm{e}(x_\nu, x_{\nu'})}\}) \qquad (21)$$

where $x_\nu, x_{\nu'} \in \Omega$ are two configurations assigned to vertex ν and ν' respectively and satisfy $\nu' \in N(\nu)$, t is the size of vertex ν which indicates the number of pixels contained in the corresponding aggregation represented by the vertex ν. $f(x_\nu, x_{\nu'})$ is the function that counts the pairwise occurrence frequency of configurations of x_ν and $x_{\nu'}$ for the neighboring vertex ν and ν' in training set, and function $D_\mathrm{e}(x_\nu, x_{\nu'})$ measures the Euclidean distance between two configurations (corresponding cluster centers) in the feature space respectively. Parameters α and β control the strength of the influence from these two terms on vertex ν (We set $\alpha = 0.05$ and $\beta = 0.01$ in our experiment for overlapped text separation).

Equation 21 indicates that for two neighboring vertices ν and ν', two aggregations which co-occur frequently in the training set and whose configurations are close to each other in the feature space are encouraged by our criterion. The size t in this equation shows that a larger vertex has more influence on its neighbors.

The final prior $P(x_\nu \mid x_{\nu'})$ is normalized to the range between 0 and 1 using a linear mapping $\mathcal{L}''$.

The likelihood $P(y_\nu \mid x_\nu)$ in Eq. 2 is modeled by a dependency function which can be approximated as:

$$P(y_\nu \mid x_\nu) \approx \frac{1}{1 + \mathrm{e}^{\lambda D_\mathrm{e}(x_\nu, y_\nu)}} \qquad (22)$$

where x_ν is a configuration assigned to vertex ν, $D_\mathrm{e}(x_\nu, y_\nu)$ is an Euclidean distance function to measure the distance from an observation to a configuration x_ν for vertex ν. This equation represents the probability of a configuration for a vertex given its observation y_ν.

The same belief propagation algorithm as described in Sect. 3 is used again for step II to get the optimal configuration $\mathcal{X}$. The final classification result is obtained by mapping the optimal configuration to two classes (machine printed text and handwritten text) as their relationship is already known during the G-means clustering procedure.

5.4 Experimental results for overlapped text separation

To estimate the performance of MRF based classification algorithm for overlapped text separation, we used the same morphology closing operation as step I to collect a total of 330 overlapped patches (including crosses, edit marks and other annotations within a single patch) from the HP Labs data set. The set of overlapped patches was exclusively split into three subsets and each subset was used for testing and the remaining for training iteratively. Figure 15 shows the similar distributions of machine printed text and handwritten text for different training sets.

In the training phase, the coarsening method described in Sect. 5.2 was used to extract aggregations for training

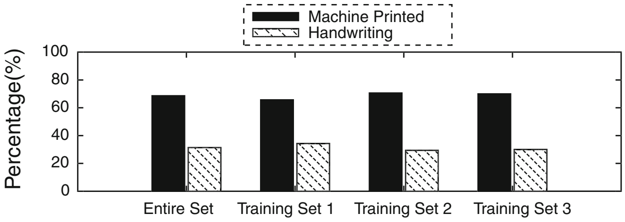


Fig. 15 Distributions of machine printed and handwritten text for different training sets

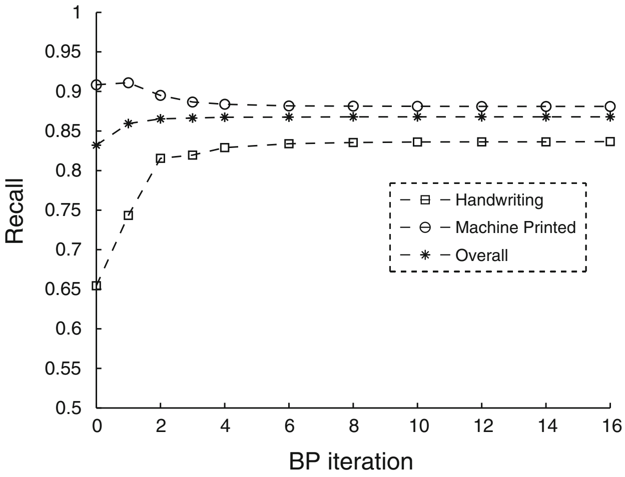


Fig. 16 Recall curves during BP iteration

Table 3 Performance of step II (separation of overlapped text into handwritten and machine printed text) on HP data set

	BP neural network		MRF (proposed method)	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Machine-printed	82.52	94.08	92.63	88.15
Handwritten	81.57	55.94	75.90	83.65
	Accuracy (%)		Accuracy (%)	
Overall	82.44		86.82	

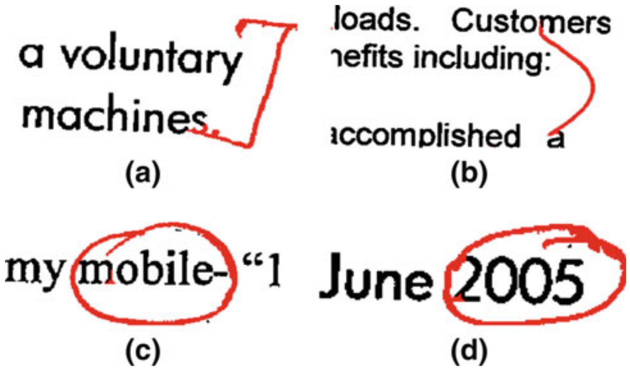


Fig. 17 Examples with the labeled results for overlapped text from the system

data followed by a vector quantization procedure based on G-means clustering to construct the configuration set Ω for machine printed text and handwritten text.

A similar coarsening procedure was applied on the test data samples to extract aggregations for each vertex ν to build the MRF to model the overlapped text patch.

Prior to MRF based classification, isolated characters were filtered out based on an estimation of their size. The MRF classification described in Sect. 5.3.2 was implemented to segment overlapped text patch into machine printed and handwritten text.

We measured the performance of the MRF classification algorithm for step II using the recall metric and precision metric. The recall for machine printed text is defined as the ratio of the amount of pixels which are correctly classified as machine printed text to all machine printed pixels in the test set. The precision for machine printed text is defined as the ratio of the amount of pixels which are correctly classified as machine printed text to all pixels which are classified as machine printed text. The same metrics were applied to handwritten text.

Figure 16 shows the recall curves of machine printed text and handwritten text during BP iteration of classification. The recall for handwritten text increased 27.72% and the overall accuracy increased 4.38% from the start of BP.

We compared the proposed MRF based segmentation algorithm with an artificial neural network classifier which was implemented using FANN [1]. The neural network had one hidden layer which used sigmoid function for each hidden neuron in our experiment. Table 3 shows that our MRF based method achieved higher overall accuracy as well as higher accuracy on handwritten text than FANN.

Figure 17 shows sample segmentation results from step II. Red strokes represent handwriting and black strokes represent machine printed text.

The main reason for the misclassification of overlapped text in step II of proposed method is that although the coarsened aggregations have coherent attributes, their sizes are determined by a predefined threshold τ which causes them to not be strictly along the edges of areas where the machine printed text and handwriting intersect. Using a heuristic method to calculate an optimal threshold for each overlapped text segment and applying a shape-driven coarsening algorithm to restrict the growth of aggregates along certain directions should improve performance.

6 Conclusions

In this paper, we propose a novel MRF based framework to classify three different kinds of text (machine printed text, handwritten text and overlapped text). The proposed method contains two individual steps. In step I, where a document image is segmented into handwritten, machine printed and overlapped text, a distance function is defined to measure the spatial distance between patches which is used to construct the neighbor system for MRF. Unlike other relabeling systems that only consider the neighboring relationship in spatial space, our model uses distances from both feature space and spatial space to determine the similarity of two neighbors. A similar MRF framework is used in step II to further separate the overlapped text into machine printed text and handwritten text. In step II, we propose a coarsening procedure to extract the basic element for classification. The merit of the proposed MRF framework is that it is easy to integrate other classifiers which can provide a reliable distance measure in feature space into our system as an initial classifier. The experimental results show that the proposed method outperforms other methods upon the accuracy. Our future work includes the use of other classifiers to overcome the issue of imbalanced data sets, and development of better algorithms for shape-driven coarsening.

References

1. FANN. http://leenissen.dk/fann/ (2007)
2. Alpert, S., Galun, M., Basri, R., Brandt A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. IEEE Conf. Comput. Vis. Pattern Recognit. 1–8 (2007)
3. Banerjee, J., Namboodiri, A., Jawahar, C.: Contextual restoration of severely degraded document images. IEEE Conf. Comput. Vis. Pattern Recognit. 517–524 (2009)
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 509–522 (2002)
5. Burns, T.J., Corso, J.J.: Robust unsupervised segmentation of degraded document images with topic models. IEEE Conf. Comput. Vis. Pattern Recognit. 1287–1294 (2009)
6. Cao, H., Govindaraju, V.: Handwritten carbon form preprocessing based on Markov Random Field. IEEE Conf. Comput. Vis. Pattern Recognit. 1–7 (2007)
7. Cao, H., Govindaraju, V.: Preprocessing of low-quality handwritten documents using Markov Random Fields. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 1184–1194 (2009)
8. Corso, J.: Discriminative modeling by boosting on multilevel aggregates. IEEE Conf. Comput. Vis. Pattern Recognit. 1–8 (2008)
9. Eduardo, J.B.D.S., Dubuisson, B., Bortolozzi, F.: Characterizing and distinguishing text in bank cheque images. In: Proceedings of the XV Brazilian Symposium on Computer Graphics and Image Processing, pp. 203–209 (2002)
10. Eran, B., Shimon, U.: Combined top-down/bottom-up segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **30**, 2109–2125 (2008)
11. Fan, K.-C., Wang, L.-S., Tu, Y.-T.: Classification of machine printed and handwritten texts using character block layout variance. Pattern Recognit. **31**, 1275–1284 (1998)
12. Farooq, F., Sridharan, K., Govindaraju, V.: Identifying handwritten text in mixed documents. IEEE Int. Conf. Pattern Recognit. **2**, 1142–1145 (2006)
13. Freeman, W., Carmichael, O., Pasztor, E.: Learning low-level vision. Int. J. Comput. Vis. **40**, 25–47 (2000)
14. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. IEEE Trans. Comput. Graph. Appl. **22**, 56–65 (2002)
15. Gorman, L.O.: The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. **15**, 1162–1173 (1993)
16. Grigorescu, S., Petkov, N., Kruizinga, P.: Comparison of texture features based on Gabor filters. IEEE Trans. Image Process. **11**, 1160–1167 (2002)
17. Guo, J., Ma, M.: Separating handwritten material from machine printed text using hidden Markov models. IEEE Int. Conf. Document Anal. Recognit. 439–443 (2001)
18. Gupta, M., Rajaram, S., Petrovic, N., Huang, T.: Models for patch based image restoration. IEEE Workshop Comput. Vis. Pattern Recognit. 17–22 (2006)
19. Hamerly, G., Elkan, C.: Learning the k in k-means. Adv. Neural Inf. Process. Syst. (2003)
20. Jang, S., Jeong, S., Nam, Y.: Classification of machine-printed and handwritten addresses on Korean mail piece images using geometric features. IEEE Int. Conf. Pattern Recognit. **2**, 383–386 (2004)
21. Lelore, T., Bouchara, F.: Document image binarisation using markov field model. IEEE Int. Conf. Document Anal. Recognit. 551–555 (2009)
22. Lettner, M., Sablatnig, R.: Spatial and spectral based segmentation of text in multispectral images of ancient documents. IEEE Int. Conf. Document Anal. Recognit. 813–817 (2009)
23. Li, S.Z.: A markov Random Field model for object matching under contextual constraints. IEEE Conf. Comput. Vis. Pattern Recognit. 866–869 (1994)
24. Li, S.Z.: Markov Random Field models in computer vision. Third European Conf. Comput. Vis. **2**, 361–370 (1994)
25. Li, S.Z.: Markov Random Field Modeling in Image Analysis. Springer, London (2009)
26. Likforman-Sulem, L., Vaillant, P., de Bodard de la Jacopiere, A.: Automatic name extraction from degraded document images. Pattern Anal. Appl. **9**, 211–227 (2006)
27. Nagy, G., Seth, S., Stoddard, S.: Document analysis with an expert system. Pattern Recognit. Pract. II. **2**, 149–155 (1984)
28. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
29. Peng, X., Setlur, S., Govindaraju, V., Sitaram, R.: Overlapped text segmentation using Markov Random Field and aggregation. In: 9th IAPR International Workshop on Document Analysis Systems (DAS), pp. 129–134 (2010)
30. Peng, X., Setlur, S., Govindaraju, V., Sitaram, R., Bhuvanagiri, K.: Markov Random Field based text identification from annotated machine printed documents. IEEE Int. Conf. Document Anal. Recognit. 431–435 (2009)
31. Petkov, N.: Biologically motivated computationally intensive approaches to image pattern recognition. Futur. Gener. Comput. Syst. **11**, 451–465 (1995)
32. Petkov, N., Kruizinga, P.: Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells. Biol. Cybern. **76**, 83–96 (1997)
33. Schultz, R.R., Stevenson, R.L.: Extraction of high-resolution frames from video sequences. IEEE Trans. Image Process. **5**, 996–1011 (1996)
34. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 1778–1792 (2005)
35. Shetty, S., Srinivasan, H., Beal, M., Srihari, S.: Segmentation and labeling of documents using conditional random fields. SPIE Int. Conf. Document Recognit. Retr. IV. 6500U-1-11 (2007)
36. Srihari, S.N., Govindaraju, V., Shekhawat, A.: Interpretation of handwritten addresses in U. S. mailstream. IEEE Int. Conf. Document Anal. Recognit. 291–294 (1993)

37. Xiong, L., Wang, F., Zhang, C.: Multilevel belief propagation for fast inference on Markov Random Fields. IEEE Int. Conf. Data Min. 371–380 (2007)
38. Zhao, L., Davis, L.S.: Iterative figure-ground discrimination. In: Proceedings of the 17th International, IEEE International Conference Pattern Recognition, vol. 1, pp. 67–70, Aug 2004
39. Zheng, Y., Li, H., Doermann, D.: Machine printed text and handwriting identification in noisy document images. IEEE Trans. Pattern Anal. Mach. Intell. **26**, 337–353 (2004)