PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 37 (2004) 595–608

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Text detection and recognition in images and video frames

Datong Chen*, Jean-Marc Odobez, Hervé Bourlard

Dalle molle Institute for Perceptual Artificial Intelligence (IDIAP), Rue du Simplon 4, Case postale 592, CH 1920 Martigny, Switzerland

Received 30 December 2002; accepted 20 June 2003

Abstract

This paper presents a new method for detecting and recognizing text in complex images and video frames. Text detection is performed in a two-step approach that combines the speed of a text localization step, enabling text size normalization, with the strength of a machine learning text verification step applied on background independent features. Text recognition, applied on the detected text lines, is addressed by a text segmentation step followed by an traditional OCR algorithm within a multi-hypotheses framework relying on multiple segments, language modeling and OCR statistics. Experiments conducted on large databases of real broadcast documents demonstrate the validity of our approach.

Keywords: Text localization; Text segmentation; Text recognition; SVM; MRF; Video OCR

1. Introduction

Content-based multimedia database indexing and retrieval tasks require automatic extraction of descriptive features that are relevant to the subject materials (images, video, etc.). The typical low-level features that are extracted in images and video include measures of color [1], texture [2], or shape [3]. Although these features can easily be obtained, they do not give a precise idea of the image content. Extracting more descriptive features and higher level entities, such as text [4] and human faces [5], has recently attracted significant research interest. Text embedded in images and video, especially captions, provide brief and important content information, such as the name of players or speakers, the title, location, date of an event, etc. This text can be a keyword resource as powerful as the information provided by speech recognizers. Besides, text-based search has been successfully applied in many applications, while the robustness and computation cost of feature matching algorithms based on other high-level features is not efficient enough to be applied to large databases.

Text detection and recognition in images and video frames, which aims at integrating advanced optical character recognition (OCR) and text-based searching technologies, is now recognized as a key component in the development of advanced image and video annotation and retrieval systems. Unfortunately, text characters contained in images and videos can be any gray-scale value (not always white), low-resolution, variable size and embedded in complex backgrounds. Experiments show that applying conventional OCR technology directly leads to poor recognition rates. Therefore, efficient detection and segmentation of text characters from the background is necessary to fill the gap between image and video documents and the input of a standard OCR system. Previously, proposed methods can be classified into bottom-up methods [6–8] and top-down methods [8–11]. Bottom-up methods segment images into regions and then group "character" regions into words. The recognition performance therefore relies on the segmentation algorithm and the complexity of the image content. Top-down algorithms first detect text regions in images and then segment each of them into text and background. They are able to process more complex images than bottom-up approaches but difficulties are still encountered at both the detection and segmentation/recognition stages.

* Corresponding author.
E-mail address: chen@idiap.ch (D. Chen).

0031-3203/$30.00
doi:10.1016/j.patcog.2003.06.001

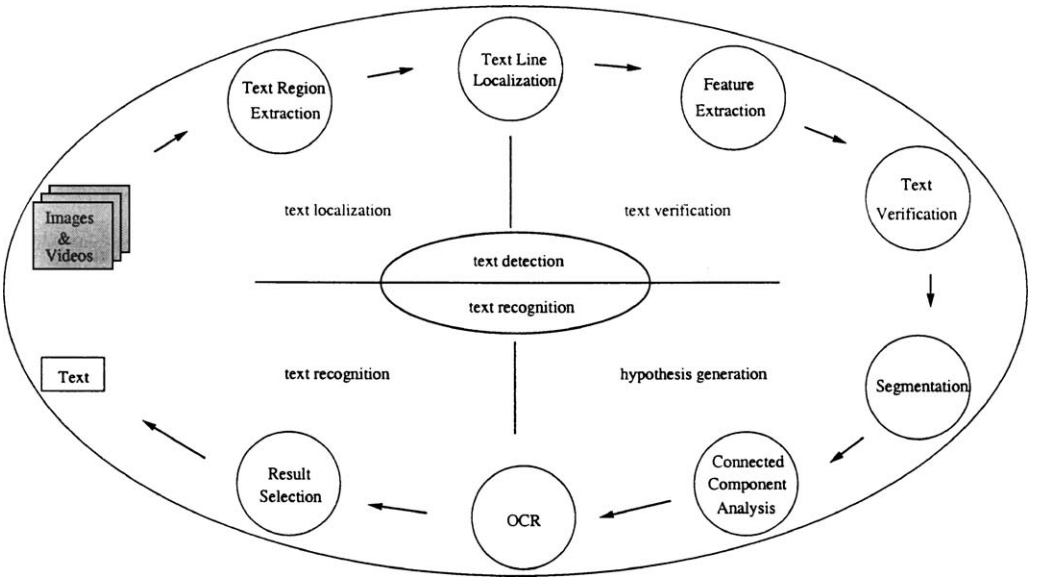

Fig. 1. Algorithm proposed for text detection and recognition.

The method we propose belongs to the top-down category, and consists of two main tasks as illustrated by Fig. 1 : a text detection task, and a text recognition task applied to the detected text regions. Following the cascade filtering idea, which consists of the sequential processing of data with more and more selective filters, the text detection task is decomposed into two subtasks. These are a text localization step, whose goal is to quickly extract potential text blocks in images with a low rejection rate and a reasonable precision, and a text verification step based on machine learning. Such an approach allows us to obtain high performance with a lower computation cost than other methods.

To address the recognition task we propose a multi-hypotheses approach. More precisely in this approach, the text image is segmented two or three times, assuming a different number of classes in the image each time. The different classes, all considered as text candidates, are processed by a commercial optical character recognition (OCR) software, and the final result is selected from the generated text string hypotheses using a confidence level evaluation based on language modeling. Additionally, we propose a segmentation method based on Markov random field [12,13] to extract more accurate text characters. This methodology allows us to handle background gray-scale multi-modality and unknown text gray-scale values, which are the problems that are often not taken into account in the existing literature. When applied to a database of several hours of sports video, it reduces by more than 50% the word recognition error rate with respect to a standard Otsu binarization step followed by the OCR.

The rest of the paper is organized as follows. Section 2 presents a more detailed review of text detection and segmentation/recognition. Section 3 describes the detection step, whereas Section 4 is devoted to the text recognition task. Section 5 describes our databases, which come from two European projects, together with the performance measures and experimental results of our approach. Section 6 provides some discussions and concluding remarks.

2. Related work

In this section, we review existing methods for text detection and text recognition. These two problems are often addressed separately in the literature.

2.1. Text detection

Text can be detected by exploiting the discriminate properties of text characters such as the vertical edge density, the texture or the edge orientation variance. One early approach for localizing text in covers of Journals or CDs [8] assumed that text characters were contained in regions of high horizontal variance satisfying certain spatial properties that could be exploited in a connected component analysis process. Smith et al. [14] localized text by first detecting vertical edges with a predefined template, then grouping vertical edges into text regions using a smoothing process. These two methods are fast but also produce many false alarms because many background regions may also have strong horizontal contrast. The method of Wu et al. [9] for text localization is based on texture segmentation. Texture features are computed at each pixel from the derivatives of the image at different scales. Using a K-means algorithm, pixels

are classified into three classes in the feature space. The class with highest energy in this space indicates text while the two others indicate non-text and uncertainty. However, the segmentation quality is very sensitive to background noise and image content and the feature extraction is computationally expensive. More recently, Garcia et al. [11] proposed a new feature referred to as variance of edge orientation. This relies on the fact that text strings contain edges in many orientations. Variation of edge orientation was computed in local area from image gradient and combined with edge features for locating text blocks. The method, however, may exhibit some problems for characters with strong parallel edges characteristics such as "i" or "1".

Besides the properties of individual characters, Sobottka et al. [7] suggested that baseline detection could be used for text string localization. More precisely, printed text strings are characterized by specific top and bottom baselines, which can be detected in order to assess the presence of a text string in an image block.

The above manually designed heuristic features usually perform fast detection but are not very robust when the background texture is very complex. As an alternative, a few systems considered machine learning tools to perform the text detection [10,15]. These systems extracted wavelet [10] or derivative features [15] from fixed-size blocks of pixels and classified the feature vectors into text or non-text using artificial neural networks. However, since the neural network based classification was applied to all the possible positions of the whole image, the detection system was not efficient in terms of computation cost and produced unsatisfactory false alarm and rejection rates.

2.2. Text recognition review

Since commercial OCR engines achieve high recognition performance when processing black and white images at high resolution, almost all the methods in the literature that addressed the issue of text recognition in complex images and videos employed an OCR system. However, these OCR systems cannot be applied directly on regions extracted by a text localization procedure. Experience shows that OCR performance in this context is quite unstable [6], and significantly depends on the segmentation quality, in the sense that errors made in the segmentation are directly forwarded to the OCR.

Some bottom-up based-techniques addressed the segmentation problem for text recognition. For instance, Lienhart [6] and Bunke [7] clustered text pixels from images using standard image segmentation or color clustering algorithm. Although these methods can avoid explicit text localization, they are very sensitive to character size, noise and background patterns. On the other hand, most top-down text segmentation methods are performed after text string localization. These methods assume that the gray-scale distribution is bimodal and that characters a priori correspond to either the white part or the black part, but without providing a way of choosing which of the two possibilities applies. Great efforts are thus devoted to performing better binarization, combining global and local thresholding [16], M-estimation [17], or simple smoothing [9]. However, these methods are unable to filter out background regions with similar gray-scale values to the characters. If the character gray-scale value is known, text enhancement methods can help the binarization process [18]. However, without proper estimation of the character scale, the designed filters cannot enhance character strokes with different thickness [19]. In videos, multi-frame enhancement [10] can also reduce the influence of background regions, but only when text and background have different movements.

These methods mostly considered segmentation as the main way to improve the text recognition results. In Section 4, we propose a multiple hypotheses framework to achieve the same goal.

3. Text detection

There are two problems in obtaining efficient and robust text detection using machine learning tools. One is how to avoid performing computational intensive classification on the whole image, the other is how to reduce the variance of character size and gray scale in the feature space before training. In this paper, we address these problems by proposing a localization/verification scheme that quickly extracts text blocks in images with a low rejection rate. This localization process allows us to further extract individual text lines and normalize the size of the text. We then perform precise verification in a set of feature spaces that are invariant to gray-scale changes.

3.1. Text localization

The first part of the text localization procedure consists of detecting text blocks characterized by short horizontal and vertical edges connected to each other. The second part aims at extracting individual text lines from these blocks.

3.1.1. Candidate text region extraction

Let S denote the set of sites (pixels) in an input image. The task of extracting text-like regions, without recognizing individual characters, can be addressed by estimating at each site s ($s \in S$) in an image I the probability $P(T|s, I)$ that this site belongs to a text block and then grouping the pixels with high probabilities into regions. To this end, vertical and horizontal edge maps C_v and C_h are first computed from the directional second derivative zeros produced by a Canny filter [20]. Then, according to the type of edge, different dilation operators are used so that vertical edges extend in horizontal direction while horizontal edges extend in vertical direction:

$$D_v(s) = C_v(s) \oplus Rect_v \quad \text{and} \quad D_h(s) = C_h(s) \oplus Rect_h. \tag{1}$$

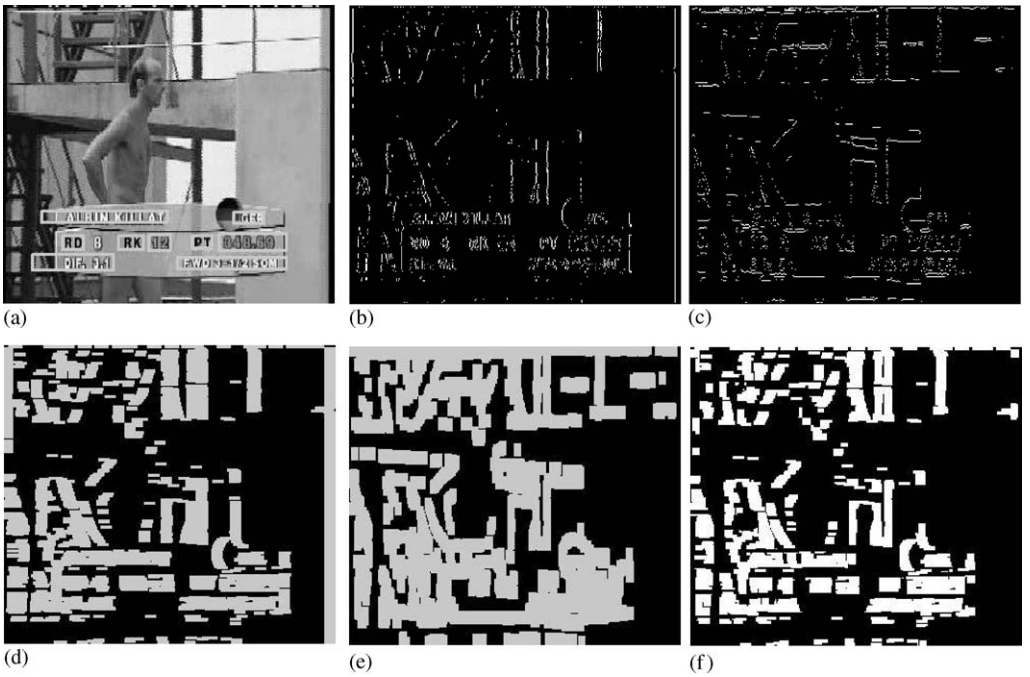

Fig. 2. Candidate text region extraction. (a) original image, (b) vertical edges detected in image (a), (c) horizontal edges detected in image (a), (d) dilation result of vertical edges using 5×1 vertical operator, (e) dilation result of horizontal edges using 3×6 horizontal operator, (f) candidate text regions.

The dilation operators $Rect_v$ and $Rect_h$ are defined to have the rectangle shapes 1×5 and 6×3. Fig. 2 (b) and (c) displays the vertical and horizontal edges resulting of this process for the video frame showed in Fig. 2(a). The vertical and horizontal edge dilation results are shown in Fig. 2(d) and (e). Due to the connections between character strokes, vertical edges contained in text-like regions should be connected with some horizontal edges, and vice versa, we consider only the regions that are covered by both the vertical and horizontal edge dilation results as candidate text regions. Thus, the probability $P(T|s,I)$ can be estimated as

$$P(T|s,I) = D_v(s)D_h(s). \quad (2)$$

Fig. 2 (f) illustrates the result of this step.

The above text detection procedure is fast and invariant to text intensity changes. Also, ideally, the threshold of the edge detection step can be set in such a way so that no true text regions will be rejected. The false alarms resulting from this procedure are often slant stripes, corners, and groups of small patterns, for example human faces. Their number can be greatly reduced using the techniques introduced in the next sections.

3.1.2. Text line localization in candidate text regions

In order to normalize text sizes, we need to extract individual text lines from paragraphs in candidate text regions. This task can be performed by detecting the top and bottom baselines of horizontally aligned text strings. Baseline detection also has two additional purposes. Firstly, it will eliminate false alarms, such as slant stripes, which do not contain any well-defined baselines. Secondly, it will refine the location of text strings in candidate regions that contain text connected with some background objects. This baseline detection is performed by three algorithms described in Ref. [4].

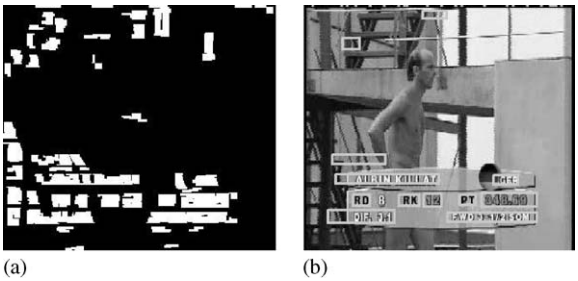

Fig. 3. Text line localization: (a) candidate text region with located baselines (top and bottom boundaries), (b) the rectangle boundaries of candidate text lines.

Fig. 3(a) illustrates the result of applying this text line localization step in Fig. 2(f). Typical characteristics of text strings are then employed to select the resulting regions

and the final candidate text line should satisfy the following constraints: it contains between 75 and 9000 pixels; the horizontal–vertical aspect ratio is more than 1.2; the height of the region is between 8 and 35. Fig. 3(b) shows the rectangle boundaries of the candidate text lines. In general, the size of the text can vary greatly (more than 35 pixels high). Larger characters can be detected by using the same algorithm on a scaled image pyramid [9].

3.2. Text verification

As in other work, the text localization procedure described in the previous subsection is rather empirical and may therefore produce false alarms (i.e. non text regions). To remove these false alarms, we used verifiers trained on both positive (text) and negative (false-alarms) examples resulting from the localization step. There are two kinds of machine learning methods based on either empirical risk minimization or structural risk minimization. The empirical risk minimization based methods, e.g. multi-layer perceptrons (MLP), minimize the error over the data set, whereas structural risk minimization methods, e.g. support vector machines (SVM) [21], aim at minimizing a bound on the generalization error of a model in high-dimensional space. The training examples that lie far from the decision hyperplanes will not change the support vectors, which may indicate better generalization on unseen backgrounds. In this section, both MLP and SVM are tested for the text verification task.

3.2.1. Feature extraction

After the text localization step, each candidate text line is normalized using bilinear interpolation into an image I having a 16 pixels height. A feature image I_f is then computed from I. The fixed size input feature vectors z_i for the MLP or SVM are directly extracted from I_f on 16×16 sliding windows. Since the gray-scale values of text and background are unknown, we tested four alternative features invariant to gray-scale changes.

3.2.1.1. Gray-scale spatial derivatives features. To measure the contribution of contrast in the text verification process, the spatial derivatives of the image brightness function in both the X and Y directions are computed at each site s.

3.2.1.2. Distance map features. Since the contrast of text characters is background dependent, the brightness spatial derivatives may not be a stable feature for text verification. Thus, we considered as a second feature image the distance map DM, which only relies on the position of strong edges in the image. It is defined by [22]

$$\forall s \in S, \quad DM(s) = \min_{s_i \in E} d(s, s_i), \tag{3}$$

where $E \subseteq S$ is a set of edge points, and d is a distance function, in our case the Euclidean distance. Though the distance map is independent of the gray-scale value of characters, the edge set E still relies on the threshold employed in edge detection.

3.2.1.3. Constant gradient variance features. To avoid the need for a threshold, we propose a new feature, called constant gradient variance (CGV), to normalize the contrast at a given point using the local contrast variance computed in a neighborhood of this point. More formally, let $g(s)$ denote the gradient magnitude at site s, and let $LM(s)$ (resp. $LV(s)$) denote the local mean (resp. the local variance) of the gradient defined by

$$LM(s) = \frac{1}{|\mathcal{G}_s|} \sum_{s_i \in \mathcal{G}_s} g(s_i) \quad \text{and}$$

$$LV(s) = \frac{1}{|\mathcal{G}_s|} \sum_{s_i \in \mathcal{G}_s} (g(s_i) - LM(s))^2, \tag{4}$$

where $\mathcal{G}_s$ is a 9×9 neighborhood around s. Then, the CGV value at site s is defined as

$$CGV(s) = (g(s) - LM(s)) \sqrt{\frac{GV}{LV(s)}}, \tag{5}$$

where GV denotes the global gradient variance computed over the whole image grid S. It can be shown that statistically, each local region in the CGV image has a zero mean and the same contrast variance equal to the global gradient variance GV.

3.2.1.4. DCT coefficients. The last feature vector we tested is composed of discrete cosine transform (DCT) coefficients computed over 16×16 blocks using a fast DCT algorithm presented by Feig [23]. These frequency domain features are commonly used in texture analysis.

3.2.2. Multi-layer perceptrons (MLPs)

MLPs are a widely used neural network, usually consisting of multiple layers of neurons: one input layer, hidden layers and one output layer. Each neuron in the hidden or output layers computes a weighted sum of its inputs (each output of the neurons in the previous layer) and then passes this sum through a non-linear transfer function to produce its output. In the binary classification case, the output layer usually consists of one neuron whose output encodes the class membership. In theory, MLPs can approximate any continuous function, and the goal in practice consists of estimating the parameters of the best approximation from a set of training samples. This is usually done by optimizing a given criterion using a gradient descent algorithm.

3.2.3. Support vector machine (SVMs)

SVMs are a technique motivated by statistical learning theory which have shown their ability to generalize well in high-dimensional spaces [24,25], such as those spanned by the texture patterns of characters. The key idea of SVMs is to implicitly project the input space into a higher dimensional space (called feature space) where the two classes

are more linearly separable. This projection, denoted ϕ, is implicit since the learning and decision process only involve an inner dot product in the feature space, which can be directly computed using a kernel K defined on the input space. An extensive discussion of SVMs can be found in Ref. [21]. In short, given m labeled training samples: $(x_1, y_1), \ldots, (x_m, y_m)$, where $y_i = \pm 1$ indicates the positive and negative classes, and assuming there exists a hyperplane defined by $w\phi(x) + b = 0$ in the feature space separating the two classes, it can be shown that w can be expressed as a linear combination of the training samples, i.e. $w = \sum_j \lambda_j y_j \phi(x_j)$ with $\lambda_j \geqslant 0$. The classification of an unknown sample z is thus based on the sign of the SVM function:

$$G(z) = \sum_{j=1}^{m} \lambda_j y_j \phi(x_j)\phi(z) + b$$

$$\doteq \sum_{j=1}^{m} \lambda_j y_j K(x_j, z) + b, \tag{6}$$

where $K(x_j, z) = \phi(x_j)\phi(z)$ is called the kernel function. The training of an SVM consists of estimating the λ_j (and b) to find the hyperplane that maximizes the margin, which is defined as the sum of the shortest distance from the hyperplane to the closest positive and negative samples.

3.2.4. Training

The database consists of samples extracted from the text and non-text examples resulting from the localization step. It was divided into a training set and a test set of equal size. Training and testing were performed using either an MLP or a SVM classifier.

The MLP network consists of one input layer, one hidden layer and one output layer with one neuron. We used the sigmoid as transfer function, and the network was trained using the backpropagation algorithm and the standard techniques for input normalization, initialization, learning rate decay. The number of hidden neurons, which is related to the capacity of the MLP, is chosen by performing a M-fold cross validation on the training set.

The SVM classifier is trained using standard quadratic programming technique. As the kernel, we choose the Radial basis function (RBF) defined by

$$K(x, x_j) = \mathrm{e}^{-\|x - x_j\|^2 / 2\sigma^2}, \tag{7}$$

where the kernel bandwidth σ is determined by M-fold cross-validation.

3.2.5. Text-line verification

In the text verification step, the feature vectors discussed in Section 3.2.1 and provided to the classifier are extracted from the normalized candidate text line on 16×16 sliding windows with a slide step of 4 pixels. Thus, for each candidate text line $\mathbf{r}$, we obtained a set of feature vectors $Z_r = (z_1^r, \ldots, z_l^r)$. The confidence of the whole candidate text line $\mathbf{r}$ is defined as

$$Conf(r) = \sum_{i=1}^{l} G(z_i^r) \times \frac{1}{\sqrt{2\pi}\sigma_0} \mathrm{e}^{-d_i^2 / 2\sigma_0^2}, \tag{8}$$

where d_i is the distance from the geometric center of the ith sliding window to the geometric center of the text line $\mathbf{r}$, σ_0 is a scale factor depending on the text line length, and $G(z_i^r)$ denotes the output of the MLP or the magnitude of the SVM (cf. Eq. (6)), which indicates the confidence that the vector z_i^r belongs to a text line. Finally, the candidate text line $\mathbf{r}$ is classified as a real text region if $Conf(r) \geqslant 0$.

4. Text recognition

In this section, we first describe the overall text recognition scheme. We then describe more thoroughly the different elements of the algorithm.

4.1. Overall description

Most of the previous methods that addressed text recognition in complex images or video worked on improving the binarization method before applying an OCR module. However, an optimal binarization might be difficult to achieve when the background is complex and the gray-scale distribution exhibits several modes. Moreover, the gray-scale value of text may not be known in advance. These problems are illustrated by the image of Fig. 2(a) and examples of detected text lines in Fig. 4.

Figs. 5 and 6 outline the multi-hypotheses approach we propose to handle these problems. A segmentation algorithm that classifies the pixels into K classes is applied on the text image. Then, for each class label, a binary text image hypothesis is generated by assuming that this label corresponds to text and all other labels correspond to background. This binary image is then passed through a connected component analysis and gray-scale consistency constraint module and forwarded to the OCR system, producing a string hypothesis (see Fig. 6). Rather than trying to estimate the right number of classes K, e.g. using a minimum description length criterion, we use a more conservative approach that varies K

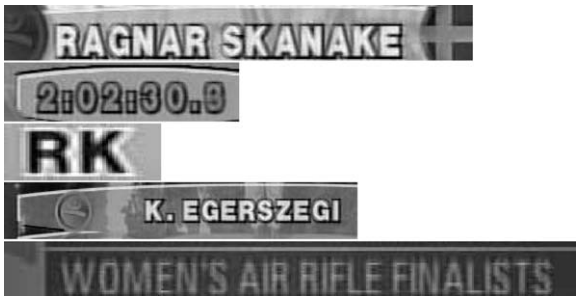

Fig. 4. Examples of detected textlines.

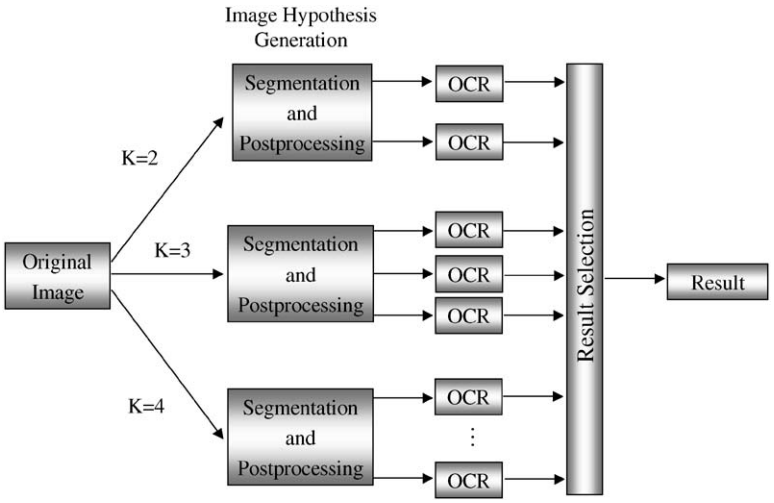

Fig. 5. Text recognition scheme.

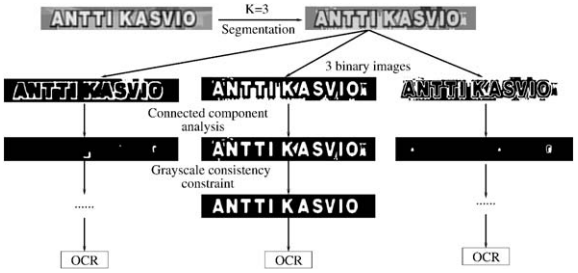

Fig. 6. Segmentation and postprocessing steps.

from 2 to 3 (resp. 4), generating in this way five (resp. nine) string hypotheses from which the text result is selected.

4.2. Segmentation methods

Let o denote the observation field $o = \{os, s \in S\}$, where o_s corresponds to the gray-level value at site (pixel) s. We assume that the image intensity distribution is composed of K classes, also referred to as layers. Each class is expected to represent regions of the image having similar gray levels, one of them being text. The segmentation is thus stated as a statistical labeling problem, where the goal is to find the label field $e = \{e_s, 1 \leqslant e_s \leqslant K, s \in S\}$ that best accounts for the observations, according to a given criterion. To perform the segmentation, we tested 3 algorithms. In the first two cases, the probability that a gray value o_s arises at a given site s within a particular layer i is modeled by a Gaussian, i.e. $p_i(o_s) = \mathcal{N}(\mu_i, \sigma_i)$.

4.2.1. The basic EM algorithm

Here, individual processes are combined into a probabilistic mixture model according to

$$p(o_s) = \sum_{k=1}^{K} p(o_s|e_s = k)\, p(e_s = k) = \sum_{k=1}^{K} \pi_k\, p_k(o_s). \qquad (9)$$

Given an image, the goal is thus to find the set of parameters $(\varphi, \pi) = (\mu_i, \sigma_i, \pi_i)_{i=1,\ldots,K}$ maximizing the likelihood of the data set o defined as $L^{\varphi}(o) = \ln p(o) = \sum_{s \in S} \ln p(o_s)$. Using the standard "expectation–maximization" (EM) algorithm [26], the expected log-likelihood of the complete data (i.e., observations and labels) can be iteratively maximized with respect to the unknown data (the labels). After maximization, the labels can be assigned to the most probable layer according to the following rule:

$$\forall s \quad e_s = \arg\max_i\, p_i(o_s). \qquad (10)$$

4.2.2. The Gibbsian EM (GBEM) algorithm

Although the EM algorithm is able to capture most of the gray-level distribution properties, it does not model the spatial correlation between assignment of pixels to layers, resulting in noisy label images. To overcome this, we introduce some prior by modeling the label field as a Markov random field (MRF). Then, instead of using the simple rule (10) to estimate e, we perform a MAP optimization, i.e., we maximize the *a-posteriori* distribution of the labels given the observations. Due to the equivalence between MRF and Gibbs distribution [12], we have

$$p(e) = \frac{1}{Z(V)}\, \mathrm{e}^{-U_1^V(e)},$$

where $Z(V)$ is a normalizing constant that cannot be computed in practice due to the high dimension of the configuration space. Thus, the MAP optimization is equivalent to the minimization of an energy function $U(e, o)$

given by

$$U(e,o) = U_1^V(e) + U_2^\varphi(e,o), \tag{11}$$

with

$$U_2^\varphi(e,o) = \sum_{s\in S}(-\ln\, p_{e_s}(o_s)), \tag{12}$$

expressing the adequacy between observations and labels, as in the EM algorithm. For the definition of U_1, we only considered second-order neighbors and set U_1 to

$$U_1^V(e) = \sum_{s\in S} V_{11}(e_s) + \sum_{\langle s,t\rangle\in\mathscr{C}_{hv}} V_{12}^{hv}(e_s,e_t) + \sum_{\langle s,t\rangle\in\mathscr{C}_{diag}} V_{12}^{d}(e_s,e_t), \tag{13}$$

where $\mathscr{C}_{hv}$ (resp. $\mathscr{C}_{diag}$) denotes the set of two elements cliques (i.e. two neighbor pixels) in the horizontal/vertical (resp. diagonal) direction. The V are the (local) interaction potentials which express the prior on the label field. One may wish to learn these parameters off-line, from examples. However, the use of learned parameters in the optimization process would require knowing the correspondence between learned labels/layers and current ones. [1] Moreover, the scale of the characters plays an important role in the optimum value of these parameters.

The second algorithm we propose consists of estimating all the parameters $\Theta = (\varphi, V)$ using an EM procedure [13]. Recall that the expectation step involves the computation of

$$\mathbb{E}[\ln\, p_{eo}^{\Theta}|o,\Theta^n] = \sum_e \ln(p_{o|e}^{\Theta}(e,o)\,p_e(e))\,p_{e|o}^{\Theta^n}(e,o) \tag{14}$$

which is then maximized over Θ. Two problems arise here. First, this expectation on $p_{e|o}^{\Theta^n}$ cannot be computed explicitly neither directly. Instead, this law will be sampled using Monte Carlo methods, and the expectation will be approximated along the obtained Markov chain. We used a Gibbs-sampler for the simulation.

Second, the joint log-likelihood probability p_{eo}^{Θ} is not completely known, because of the presence of the uncomputable normalizing constant $Z(V)$ in the expression of $p(e)$. To avoid this difficulty, we use the pseudo-likelihhod function [13] as a new criterion, that is, in Eq. (14), we replace $p(e)$ by its pseudo-likelihood $p_S(e)$ defined from the local conditional probabilities:

$$p_S^V(e) \doteq \prod_{s\in S} p(e_s|e_{\mathscr{G}_s}), \tag{15}$$

where $e_{\mathscr{G}_s}$ represents the label in neighborhood of s. Using this new criterion, the maximization of expectation (14) can be performed, providing new estimates of (μ_i,σ_i) and V. The complexity of the GBEM algorithm is approximately 4 times greater than the complexity of the EM algorithm.

[1] Remember that text may be black, white or gray.

4.2.3. The Kmeans algorithm

In this method, the goal is to find the K means of the disjoint subsets S_i (the layers) which minimizes the intra-class variance [27], that is

$$IV = \sum_{i=1}^{K}\sum_{s\in S_i}\|o_s-\mu_i\|^2 = \sum_{s\in S}\|o_s-\mu_{e_s}\|^2.$$

The minimization can be achieved using standard algorithms, which iteratively assign each data to the class of the nearest center and then recompute the means.

4.3. Postprocessing: connected component analysis (CCA) and grayscale consistency constraint (GCC)

To help the OCR system, a simple connected component analysis is used to eliminate non-character regions in each hypothesis based on their geometric properties. We only keep connected components that satisfy the following constraints: size is bigger than 120 pixels; width/height ratio is between 4.5 and 0.1; the width of the connected component is less than 2.1 times the height of the whole text region.

Since we only consider 2–4 layers, regions from the background with a gray value slightly different from that of characters may still belong to the text layer/class. We thus developed another module to enforce a more stringent gray consistency among the connected components (see Fig. 6): after the CCA step, we estimate with a robust estimator [28] the gray-level mean m^* and standard deviation st^* of the set S_r of sites belonging to the remaining regions. More precisely, a least-median squares estimator is employed to identify the gray-level value that fits the majority of pixel gray-level values and then eliminate the pixel with outlier gray-level values. m^* and st^* are estimated on the remaining valid pixels using standard formula [28]. Finally, a connected component is eliminated from the layer if more than 50% of its pixels have a gray level value different than the majority of pixels, that is, verify

$$\frac{|o_s - m^*|}{st^*} > k. \tag{16}$$

An illustration of the result of this step is displayed in Fig. 7.

4.4. OCR and result selection

The selection of the result from the set of strings generated by the segmentation processes (see Fig. 5) is based on a confidence value evaluation relying on language modeling and OCR statistics. From a qualitative point of view, when given text-like background or inaccurate segmentation, the OCR system produces mainly garbage characters like ., ,!, &, etc. and simple characters like i,l, and r.

Let $T = (T_i)_{i=1\ldots l_T}$ denote a string where l_T denotes the length of the string and each character T_i is an element of the character set

$$\mathscr{T} = (0,\ldots,9,a,\ldots,z,A,\ldots,Z,G_b),$$

Fig. 7. Applying gray-scale consistency: (a) original image, (b) text layer (3 layers, GBEM algorithm), (c) same as (b), after the connected component analysis, (d) same as (c), after the gray-level consistency step.

in which G_b corresponds to any other garbage character. Furthermore, let H_a (resp. H_n) denote the hypothesis that the string T or the characters T_i are generated from an accurate (resp. a noisy) segmentation. The confidence value is estimated using a variant of the log-likelihood ratio

$$C_v(T) = \log\left(\frac{p(H_a|T)}{p(H_n|T)}\right) + l_T * b \tag{17}$$

$$= \log(p(T|H_a)) - \log(p(T|H_n)) + l_T * b \tag{18}$$

when assuming an equal prior on the two hypotheses and b is a bias that is discussed below. We estimated the noise free language model $p(.|H_a)$ by applying the CMU-Cambridge Statistical Language Modeling (SLM) toolkit on Gutenberg collections.[2] A bigram model was selected. Cutoff and back-off techniques [29] were employed to address the problems associated with sparse training data for special characters (e.g. numbers and garbage characters). The noisy language model $p(.|H_n)$ was obtained by applying the same toolkit on a database of strings collected from the OCR system output when providing as input to the OCR either badly segmented texts or text-like false alarms coming from the text detection process. Only a unigram model was used because the size of the background data set was insufficient to obtain a good bigram model. The bias b is necessary to account for the string length. It was observed that without this bias, the likelihood ratio would quite often select strings with only a few quite reliable letters instead of the true string. By incorporating this bias in the confidence value, the selection module was encouraged to compare string results whose length was in accordance with the underlying text image width. Setting $b = 0.7$, the confidence value is defined as

$$C_v(T) = \log p(T_1|H_a) + \sum_{i=2}^{l_T} \log p(T_i|T_{i-1}, H_a) - \sum_{i=1}^{l_T} \log p(T_i|H_n) + 0.7 * l_T.$$

[2] www.gutenberg.net

5. Experiments and results

In this section, we report results on text localization, text verification, and text recognition.

5.1. Text localization results

The text localization step is evaluated on a half hour video containing a Belgian news program[3] in French provided by Canal+ in the context of the CIMWOS[4] European project. The performance of the text localization step is measured in terms of rejection rate and precision.

We counted the text strings that were correctly located. A ground-truth text region is considered to be correctly located if it has an 80% overlap with one of the detected string regions. With the proposed method, we extracted 9369 text lines and 7537 false alarms in the CIMWOS database. There were no rejected regions. The precision of this localization step on this database is $\frac{9369}{9369+7537} = 55.4\%$. A detailed comparison between the algorithm proposed in subsection 3.1, the derivative texture algorithm [9], and the vertical edge based algorithm [14] can be found in Ref. [30].

5.2. Text verification results

The text verification algorithms were designed and trained on a database consisting of still images and half an hour of video recorded from TV. The videos contain excerpts from advertisements, sports, interviews, news and movies. The still images include covers of journals, maps and flyers. The video frames have a size of 720×576 and are compressed in MPEG, while the still images have a size of 352×288 and are compressed in JPEG. Only the gray-scale information was used in the experiments.

The feature extraction, training and testing procedures described in Section 3.2 were applied on this database. More precisely, 2400 candidate text regions containing both true text lines and false alarms were randomly selected from the output of the text localization step applied on this database. From these regions the feature extraction step produced 76,470 vectors for each of the four kinds of features. It was ensured that the test set and the training set contained vectors extracted from the same windows (i.e. same image and location) in all the experiments, where one experiment is characterized by a couple (classifier, feature).

Table 1 lists the error rate measured on the test set for each feature and for each classifier. First of all, we can see that these results are very good and better than those reported when running the classifier on the whole image without applying size normalization (13–30% [15]). Additionally, the proposed scheme runs approximately five times faster. Second, whatever the considered feature, the SVM

[3] From the Radio-Télévision Belge d'expression Française (RTBF).

[4] Combined Image and Word Spotting.

Table 1
Error rates of the SVM and MLP classifiers for the text verification task

Training tools	DIS (%)	DERI (%)	CGV (%)	DCT (%)
MLP	5.28	4.88	4.40	4.72
SVM	2.56	3.99	1.07	2.0

DIS denotes the distance map feature. DERI denotes the grayscale spatial derivative feature. CGV denotes the constant gradient variance feature. DCT denotes the DCT coefficients.

classifier gives better results than the MLP classifier, showing its ability to generalize better. Finally, we can see that the proposed constant gradient variance feature provides the best result. This can be explained by its better invariance to text/background contrast.

The SVM classifier together with the CGV feature was employed to verify the extracted text regions of the CIMWOS database, based on the confidence value given by Eq. (8). This verification scheme removed 7255 regions of the 7537 false alarms while only rejecting 23 true text lines, giving a 0.24% rejection rate and a 97% precision rate. Fig. 8 shows examples of detected text on some images in our databases.

5.3. Text recognition results

The multiple hypotheses recognition scheme was tested on a sports database of the Barcelona 1992 Olympic games provided by the BBC in the context of the ASSAVID[5] European project. From about five hours of video, we only kept approximately 1 h of video that contained text. The text localization and verification algorithms were applied. As the text regions located in consecutive video frames usually contain similar text and background, the video data results where sub-sampled in time, leading to a database of 1208 images containing 9579 characters and 2084 words. Text characters are embedded in complex background with JPEG compression noise, and the grayscale value of characters is not always the highest, as shown in Fig. 8.

To assess the performance of the different algorithms, we use character recognition rate (CRR) and character precision rate (CPR) that are computed on a ground truth basis as

$$CRR = \frac{N_r}{N} \quad \text{and} \quad CPR = \frac{N_r}{N_e},$$

N is the true total number of characters, N_r is the number of correctly recognized characters and N_e is the total number of extracted characters. The number of correctly recognized characters is computed using an edit distance[6] between the recognized string and the ground truth.[7] More precisely, let l_T, del, ins and sub, respectively, denote the length of the recognized text string, the number of deletions, insertions, and substitutions obtained when computing the edit distance. The number N_r of correctly recognized characters in this string is then defined as

$$N_r = l_T - (del + sub).$$

Intuitively, if in order to match the ground truth, we need to delete a character or substitute a character, it means that this character is not in the ground truth.

Additionally, we compute the word recognition rate (WRR) to get an idea of the coherency of character recognition within one solution. For each text image, we count the words from the ground truth of that image that appear in the string result. Thus, WRR is defined as the percentage of words from the ground truth that are recognized in the string results.

We first report results where the string result is selected from the hypotheses generated by applying the segmentation algorithm one time with a fixed K value, and when applying only the connected component analysis as a post-processing step. This will serve as a baseline for the rest of the experiments. Table 2 lists the results obtained with the three described segmentation methods.

It can be seen that the usual bi-modality ($K = 2$) hypothesis yields the best character and word recognition rate with the GBEM and Kmeans algorithms. However, in the case of $K = 3$, the Kmeans algorithm also gives quite similar results. Indeed, some text images are composed of grayscale characters, contrast contours around characters, and background (see Fig. 4). In this case, the gray-scale values are better modeled with 3 or more clusters. Under the bimodality assumption ($K = 2$), the GBEM algorithm yields better results than the typical Otsu's binarization method (Kmeans with $K = 2$) in terms of both CRR and WRR. This is probably due to the regularization power of the GBEM algorithm, which learns the spatial properties of the text and background layers. It helps in avoiding over segmentation, as can be seen from the example of Fig. 9. However, the improvement is not very important and is deceptive. It can be explained by the fact that the MRF approach mainly improves the character shape, a kind of noise the OCR has been trained on and to which it is probably not very sensitive.

The gray-scale consistency constraint (GCC) technique described in Section 4 was added to the baseline system, and the corresponding results are listed in Table 2. When $K = 2$, they show an increase in absolute value of about 4% of the WRR together with an increase of 1.2% of both the CRR and the CPR. This is due to the ability of the GCC to remove burst-like noise (i.e. significant background regions with a slightly different gray-level value than

[5] ASSAVID : Automatic Segmentation and Semantic Annotation of Sports Videos.

[6] The edit distance of two strings, s_1 and s_2, is defined as the minimum number of character operations needed to change s_1 into s_2, where an operation is one of the following: deletion, insertion, substitution.

[7] Only numeric, upper and lower case letters are kept.

Fig. 8. Detected text regions in images or video frames.

Table 2
Recognition results: number of extracted characters (Ext.), character recognition rate (CRR), precision (CPR) and word recognition rate (WRR)

	K	Algorithm	Ext.	CRR (%)	CPR (%)	WRR (%)
		EM	7715	74.9	93.1	61.0
	2	GBEM	9300	92.8	95.6	83.5
		Kmeans	9260	92.5	95.7	82.8
Without		EM	9239	89.9	93.2	80.9
GCC	3	GBEM	9302	89.9	92.6	80.7
		Kmeans	9394	91.3	93.1	82.2
		EM	9094	87.4	92.1	77.7
	4	GBEM	9123	88.3	92.8	79.9
		Kmeans	9156	88.0	92.1	79.7
		EM	7914	77.9	94.2	66.2
	2	GBEM	9307	94.0	96.8	87.1
		Kmeans	9291	93.8	96.7	86.8
With		EM	9245	90.3	93.6	81.7
GCC	3	GBEM	9268	89.5	92.5	81.1
		Kmeans	9395	91.2	93.0	83.4
		EM	9136	88.0	92.3	78.9
	4	GBEM	9123	87.7	92.1	79.1
		Kmeans	9195	88.9	92.6	80.4

characters) which greatly impairs the recognition of the OCR. For higher values of K, the increase is less important. Indeed, in these cases, the gray-scale consistency constraint is inherently better respected.

Table 3 lists the results obtained by generating 5 or 9 hypotheses (using $K = 2$ to 4) in the multi-hypotheses framework. Without employing the GCC postprocessing, the method achieves a 96.6% CRR and a 93.1% WRR, which constitutes a reduction of more than 50% for both rates with respect to the best baseline result (GBEM with $K = 2$). These results demonstrate first the complementary of the solutions provided when assuming different K values, and second the ability of our selection algorithm to choose the right solution. Interestingly, the results obtained with 9 hypotheses are not better than the results obtained using only 5 hypotheses. It probably means that the segmentation with $K = 4$ does not generate additional interesting results with respect to the $K = 2$ and 3 cases.

When integrating the GCC algorithm in the multiple hypotheses framework, we can notice that the GCC postprocessing improves the result when using the Kmeans or EM segmentation algorithms and remain similar for the GBEM algorithm. This smaller improvement, when compared to the improvement obtained when adding the GCC to the baseline, can be explained by the fact that the multiple hypotheses framework has less need for burst-noise elimination, since it can select between alternative modelization of the gray-level distribution.

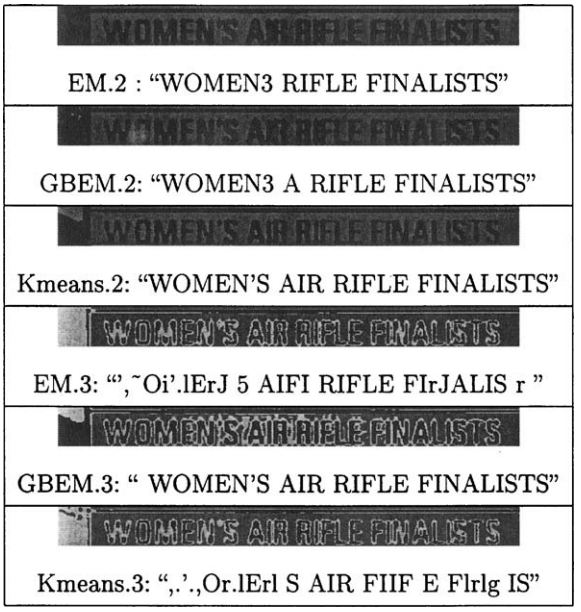

Fig. 9. Segmentation output of the three studied algorithms and associated recognition results using 2 or 3 layers/classes.

6. Discussion and conclusions

This paper presents a general scheme for extracting and recognizing embedded text of any gray-scale value in images and videos. The method is split into two main parts: the detection of text lines, followed by the recognition of text in these lines.

Applying machine learning methods for text detection encounters difficulties due to character size and gray-scale variations and heavy computation cost. To overcome these problem, we proposed a two-step localization/verification scheme. The first step aims at quickly locating candidate text lines, enabling the normalization of characters into a unique size. In the verification step, a trained SVM or MLP is applied on background independent features to remove the false alarms. Experiments showed that the proposed scheme improves the detection result at a lower cost in comparison with the same machine learning tools applied without size normalization, and that an SVM was more appropriate than an MLP to address the text texture verification problem.

The text recognition method we propose embeds the traditional character segmentation step followed by an OCR algorithm within a multiple hypotheses framework. A new gray-scale consistency constraint (GCC) algorithm was proposed to improve segmentation results. The experiments that were conducted on approximately 1 h of sports video demonstrate the validity of our approach. More specifically, when compared to a baseline system consisting of the standard Otsu binarization algorithm, the GCC postprocessing step was able to reduce the character and word error rates by more than 20%, showing its ability to remove burst-like noise that greatly disturbs the OCR software. Moreover, added to the multiple hypotheses framework, the whole system yielded approximately 97% character recognition rate and a more than 93% word recognition rate on our database, which constitutes a reduction of more than 50% w.r.t.

Table 3
Recognition results from 5, 9 hypotheses, with or without GCC: number of extracted characters (Ext.), character recognition rate (CRR), character precision rate (CPR) and word recognition rate (WRR)

	K	Algorithm	Ext.	CRR	CPR	WRR
Without GCC	2, 3	EM	9480	93.3	94.3	86.7
		GBEM	9606	96.6	96.3	93.1
		Kmeans	9565	96.6	96.8	92.8
	2, 3, 4	EM	9417	93.2	94.8	86.8
		GBEM	9604	96.6	96.2	93.0
		Kmeans	9547	96.6	97.0	92.9
With GCC	2, 3	EM	9449	94.0	95.3	88.1
		GBEM	9579	96.5	96.5	92.8
		Kmeans	9587	97.1	97.0	93.7
	2, 3, 4	EM	9411	93.9	95.6	88.1
		GBEM	9557	96.6	96.8	93.0
		Kmeans	9560	97.0	97.2	93.8

the baseline system. This clearly shows that (i) several text images may be better modeled with 3 or 4 classes rather than using the usual 2 class assumption (ii) multiple segmentation maps provide alternative solutions and (iii) the proposed selection algorithm based on language modeling and OCR statistics is often able to pick up the right solution.

We proposed to use a maximum a posteriori criterion with a MRF modeling to perform the segmentation. Used as a traditional binarization algorithm, it performed better than Otsu's method. However, embedded in the multi-hypotheses system with GCC, it yielded similar results to the Kmeans. Thus, the latter is preferred for real applications since it runs faster.

The performance of the proposed methodology is good enough to be used in a video annotation and indexing system. In the context of the ASSAVID European project, it was integrated with other components (shot detector, speech recognizer, sports and event recognizers, etc.) in a user interface designed to produce and access sports video annotation. A simple complementary module combining the results from consecutive frames containing the same text was added. User experiments with librarians at the BBC showed that the text detection and recognition technology produced robust and useful results, i.e. did not produce many false alarms and the recognized text was accurate. The same proposed scheme is currently used in the CIMWOS project to index French news programs.

Acknowledgements

The authors would like to thank the Swiss National Fund for Scientific Research that supported this work through the Video OCR project. This work has been performed partially within the frameworks of the "Automatic Segmentation and Semantic Annotation of Sports Videos (ASSAVID)" project and the "Combined Image and Word Spotting (CIMWOS)" project both granted by the European IST Programme.

References

[1] M. Swain, H. Ballard, Color indexing, Int. J. Comput. Vision 7 (1991) 11–32.

[2] B. Manjunath, W. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 837–842.

[3] F. Mokhtarian, S. Abbasi, J. Kittler, Robust and efficient shape indexing through curvature scale space, in: British Machine Vision Conference, 1996, pp. 9–12.

[4] D. Chen, H. Bourlard, J.-P. Thiran, Text identification in complex background using SVM, in: International Conference on Computer Vision and Pattern Recognition, 2001, pp. 621–626.

[5] R.K. Srihari, Z. Zhang, A. Rao, Intelligent indexing and semantic retrieval of multimodal documents, Inf. Retri. 2 (2/3) (2000) 245–275.

[6] R. Lienhart, Automatic text recognition in digital videos, in: Proceedings SPIE, Image and Video Processing IV, 1996, pp. 2666–2675.

[7] K. Sobottka, H. Bunke, H. Kronenberg, Identification of text on colored book and journal covers, in: International Conference on Document Analysis and Recognition, 1999, pp. 57–63.

[8] Y. Zhong, K. Karu, A.K. Jain, Locating text in complex color images, Pattern Recognition 10 (28) (1995) 1523–1536.

[9] V. Wu, R. Manmatha, E.M. Riseman, Finding text in images, in: Proceedings of ACM International Conference on Digital Libraries, 1997, pp. 23–26.

[10] H. Li, D. Doermann, Text enhancement in digital video using multiple frame integration, in: ACM Multimedia, 1999, pp. 385–395.

[11] C. Garcia, X. Apostolidis, Text detection and segmentation in complex color images, in: International Conference on Acoustics, Speech and Signal Processing, 2000, pp. 2326–2329.

[12] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, PAMI 6 (6) (1984) 721–741.

[13] B. Chalmond, Image restoration using an estimated Markov model, Signal Process. 15 (2) (1988) 115–129.

[14] M.A. Smith, T. Kanade, Video skimming for quick browsing based on audio and image characterization, Technical Report CMU-CS-95-186, Carnegie Mellon University, July 1995.

[15] R. Lienhart, A. Wernicke, Localizing and segmenting text in images and videos, IEEE Trans. Circuits Syst. Video Technol. 12 (4) (2002) 256–268.

[16] H. Kamada, K. Fujimoto, High-speed, high-accuracy binarization method for recognizing text in images of low spatial resolutions, in: International Conference on Document Analysis and Recognition, 1999, pp. 139–142.

[17] O. Hori, A video text extraction method for character recognition, in: International Conference on Document Analysis and Recognition, 1999, pp. 25–28.

[18] T. Sato, T. Kanade, E.K. Hughes, M.A. Smith, Video OCR for digital news archives, in: IEEE Workshop on Content Based Access of Image and Video Databases, Bombay, 1998, pp. 52–60.

[19] D. Chen, K. Shearer, H. Bourlard, Text enhancement with asymmetric filter for video OCR, in: Proceedings of the 11th International Conference on Image Analysis and Processing, 2001, pp. 192–198.

[20] J.F. Canny, A computational approach to edge detection, IEEE Trans. on Pattern Anal. Mach. Intell. 8 (1) (1986) 679–698.

[21] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[22] J. Toriwaki, S. Yokoi, Distance transformations and skeletons of digitized pictures with applications, Pattern Recognition (1981) 187–264.

[23] E. Feig, S. Winograd, Fast algorithms for the discrete cosine transform, IEEE Trans. Signal Process. 40 (28) (1992) 2174–2193.

[24] B. Schölkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, Comparing support vector machines with gaussian kernels to radial basis functions classifiers, IEEE Trans. Signal Process. 45 (11) (1997) 2758–2765.

[25] R. Collobert, S. Bengio, Y. Bengio, A parallel mixture of svms for very large scale problems, Neural Comput. 14 (5) (2002) 1105–1114.

[26] A. Dempster, N. Laird, D. Rubin, Maximum-likelihood from incomplete data via the em algorithm, R. Stat. Soc. B-39 (1977) 1–38.
[27] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 1 (9) (1979) 62–66.
[28] P. Rousseeuw, Least median of squares regression, Am. Stat. Assoc. 79 (388) (1984) 871–880.
[29] S. Katz, Estimation of probabilities from sparse data for the language model component of a speech recognizer, IEEE Trans. Acoust. Speech Signal Process. 35 (1987) 400–401.
[30] D. Chen, J.-M. Odobez, A new method of contrast normalization for text verification in complex backgrounds, IDIAP-RR-02 16, IDIAP, April 08, 2002.

About the Author—DATONG CHEN received his M.S and B.S. in computer science and engineering from Harbin Institute of Technology, China. He joined IDIAP (Dalle Molle Institute for perceptual artificial intelligence) as an research assistant from 1999 and became a Ph.D. candidate in Swiss Federal Institute of Technology at Lausanne (EPFL), Switzerland from 2001. Before joined IDIAP, he worked in Motorola-NCIC joint R& D laboratory in China and the University of Karlsruhe in Germany. His research interests are in the areas of pattern recognition, machine learning, stochastic modeling, computer vision, multimedia indexing and retrieval, mobile computing and video coding.

About the Author—JEAN-MARC ODOBEZ was born in France in 1968. He graduated from the Ecole Nationale Supérieure de Télécommunications de Bretagne (ENSTBr) in 1990, and received his Ph.D degree in Signal Processing and Télécommunication from Rennes University, France in 1994. He performed his dissertation research at IRISA/INRIA Rennes on dynamic scene analysis (image stabilization, object detection and tracking, image sequence coding) using statistical models (robust estimation, 2D statistical labeling with Markov Random Field). He then spent one year as a post-doctoral fellow at the GRASP laboratory, University of Pennsylvania, USA, working on visually guided navigation problems. From 1996 until september 2001, he was associate professor at the Université du Maine, France. He is now a senior researcher at IDIAP, working mainly on the development of statistical methods for multimodal dynamic scene analysis and video structuring and indexing.

About the Author—HERVE BOURLARD received the Electrical and Computer Science Engineering degree and the Ph.D. degree in Applied Sciences both from Facult'e Polytechnique de Mons, Mons, Belgium. After having been a member of the Scientific Staff at the Philips Research Laboratory of Brussels (PRLB, Belgium), and an R& D Manager at L& H SpeechProducts (BE), he is now Director of IDIAP (Dalle Molle Institute for Perceptual Artificial Intelligence), Professor at the Swiss Federal Institute of Technology at Lausanne (EPFL), and External Fellow of the International Computer Science Institute (ICSI) at Berkeley (CA). His current interests mainly include statistical pattern classification, artificial neural networks, and applied mathematics, with applications to signal processing, speech and speaker recognition, and language modeling. H. Bourlard is the author/coauthor of over 140 reviewed papers (including one IEEE paper award) and book chapters, as well as two books. H. Bourlard is a member of the programme and/or scientific committee of numerous international conferences, and Editor-in-Chief of the "Speech Communication" journal (Elsevier). He is also a Member of the Advisory Council of ISCA (International Speech Communication Association), Member of the IEEE Technical Committee on Neural Network Signal Processing, and an appointed expert for the European Commission. He is a Fellow of the IEEE "for contributions in the fields of statistical speech recognition and neural networks", as well as a member of the Board of Trustees of the Intl. Computer Science Institute (Berkeley, CA).