Extraction and Recognition of Artificial Text in Multimedia Documents

Christian Wolf and Jean-Michel Jolion

RFV RR-2002.01

Laboratoire Reconnaissance de Formes et Vision
INSA de Lyon
Bât. J.Verne
20, Av. Albert Einstein
69621 Villeurbanne cedex
FRANCE

Tel.: (0033).4.72.43.60.89
Fax.: (0033).4.72.43.80.97
Email: {wolf,jolion}@rfv.insa-lyon.fr

Abstract

The systems currently available for content based image and video retrieval work without semantic knowledge, i.e. they use image processing methods to extract low level features of the data. The similarity obtained by these approaches does not always correspond to the similarity a human user would expect. A way to include more semantic knowledge into the indexing process is to use the text included in the images and video sequences. It is rich in information but easy to use, e.g. by key word based queries. In this paper we present an algorithm to localize artificial text in images and videos using a measure of accumulated gradients and morphological processing. The quality of the localized text is improved by robust multiple frame integration. A new technique for the binarization of the text boxes based on a criterion maximizing local contrast is proposed. Finally, detection and OCR results for a commercial OCR are presented, justifying the choice of the binarization technique.

Keywords:
Text Extraction, image enhancement, binarization, OCR, video Indexing

Originality and Contribution:
Only few of the existing works for text extraction represent complete methods beginning from the detection of the text until the last binarization step before passing the results to an OCR. In most cases the model behind the detection algorithms is very simple: Detection of edges, regrouping into rectangles. We present an integral approach to text extraction and recognition. As many properties as possible of the video signal are used by combining several constraints and several processing steps in order to exploit more than only the gray value properties of the signal. We present robust algorithms which are adapted to the data found in video signals. Regarding practical applications, our approach is directly usable in indexing applications, as intended by our industrial partner France Télécom, who patented the core of the detection system.

1 Introduction

Content Based Image Retrieval and its extension to videos is a research area which gained a lot of attention in the recent years. Various methods and techniques have been presented, which allow to query big databases with multimedia contents (images, videos etc.) using features extracted by low level image processing methods and distance functions which have been designed to resemble human visual perception as closely as possible. Nevertheless, query results returned by these systems do not always match the results desired by a human user. This is largely due to the lack of semantic information in these systems. Systems trying to extract semantic information from low level features have already been presented [17], but they are error prone and very much depend on large databases of pre-defined semantic concepts and their low level representation. Another method to add more semantics to the query process is relevance feedback, which uses interactive user feedback to steer the query process. See [25] and [12] for surveying papers on this subject.

Systems mixing features from different domains (image and text) are an interesting alternative to mono-domain based features [4]. However, the keywords are not available for all images and are very dependent on the indexer's point of view on a given image (the so-called polysemy of images), even if they are closely related to the semantic information of certain video sequences (see figure 1).

In this paper we focus on text extraction and recognition in videos. The text is automatically extracted from the videos in the database and stored together with a link to the video sequence and the frame number. This is a complementary approach to basic keywords. The user submits a request by providing some keywords, which are robustly matched against the previously extracted text in the database. Videos containing the keywords (or a part of them) are presented to the user. This can also be merged with image features like color or texture.

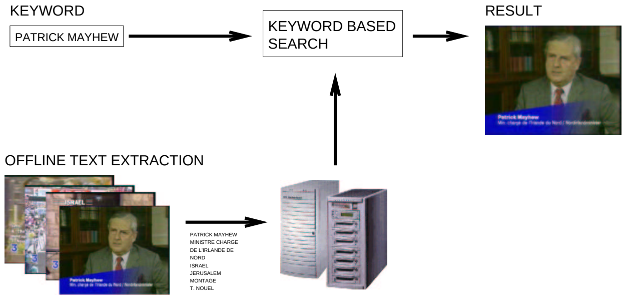


Figure 1: Keyword based indexing using automatically extracted text.

The algorithm as it is presented in this paper has been used for indexing purposes in the framework of the TREC 2002 video track, a video indexing competition organized by the National Institute of Standards and Technology. In collaboration with the University of Maryland, a video indexing algorithm based on text recognition, speech recognition, color and binary features has been developed [30].

Extraction of text from images and videos is a very young research subject, which nevertheless attracts a large number of researchers. The first algorithms, introduced by the document processing community for the extraction of text from colored journal images and web pages, segmented characters before grouping them to words and lines. Jain et al. [6] perform a color space reduction followed by color segmentation and spatial regrouping to detect text. Although processing of touching characters is considered by the authors, the segmentation phase presents major problems in the case of low quality documents, especially video sequences. A similar approach, which gives impressive results on text with large fonts, has been presented by Lienhart [14]. A segmentation algorithm and regrouping algorithm is combined with a filter detecting high local contrast, which results in a method which is more adapted to text of low quality, but still the author cannot demonstrate the applicability of his algorithm to small text. False alarms are removed by texture analysis, and tracking is performed on character level. Similar methods working on color clustering and regrouping of components have been presented by Zhou and Lopresti [34] and by Sobottka, Bunke et al. [26].

The methods based on segmentation worked fine for high resolution images as journals but failed in the case of low resolution video, where characters are touching and the font size is very small. New methods based on edge detection or texture analysis soon followed.

The video indexing system introduced by Sato, Kanade et al. [23] combines closed caption extraction with super-imposed caption (artificial text) extraction. The text extraction algorithm is based on the fact that text consists of strokes with high contrast. It searches for vertical edges which are grouped into rectangles. The authors recognized the necessity to improve the quality of the text before passing an OCR step. Consequently, they perform an interpolation of the detected text rectangles before integrating multiple frames into a single enhanced image by taking the minimum/maximum value for each pixel. They also introduced an OCR step based on a correlation measure. A similar method using edge detection and edge clustering has been proposed by Agnihotri and Dimitrova [1]. Wu, Manmatha and Riseman [31] combine the search for vertical edges with a texture filter to detect text. Unfortunately, these binary edge clustering techniques are sensitive to the binarization step of the edge detectors. A similar approach has been developed by Myers et al.[18]. However, the authors concentrate on the correction of perspective distortions of scene text after the detection. Therefore, the text must be large so that baselines and vanishing points can be found. Since the detection of these features is not always possible, assumptions on the imaging geometry need to be made.

LeBourgeois [11] moves the binarization step after the clustering by calculating a measure of accumulated gradients instead of edges. Our work is based on a slightly modified variant of this filter with additional constraints and additional processing in order to exploit more than only the gray value properties of the signal (see section 2). In his work, LeBourgeois also proposes an OCR algorithm which uses statistics on the projections of the gray values to recognize the characters.

Various methods based on learning have been presented. Li and Doermann use the Haar wavelet for feature extraction [13]. By gliding a fixed size window across the image, they feed the wavelet coefficients to a MLP type neural network in order to classify each pixel as “text” or “non-text”. Clark and Mirmehdi also leave the classification to a neural network fed with various features, as the histogram variance, edge density etc. [2]. Similarly, Wernike and Lienhart extract overall and directional edge strength as features and use them as input for a neural network classifier [29]. In a more recent paper, the same authors change the features to a 20×10 map of color edges, which is fed directly to the neural network[15]. Jung directly uses the gray values of the pixels as input for a neural network to classify regions whether they contain text or not [7]. Kim et al. also use the gray values only as features, but feed them to a support vector machine for classification [9]. Tang et al. use unsupervised learning to detect text [27]. However, their features are calculated from the differences between adjacent frames of the same shot, resulting in a detection of appearing and disappearing text. Text which is present from the beginning of a shot to the end (a type of text frequently encountered for locations in news casts) is missed. As usual, learning methods depend on the quality of the training data used to train the systems. However, in video, text appears in various sizes, fonts, styles etc., which makes it very difficult to train a generalizing system.

Methods working directly in the compressed domain have been introduced. Zhong, Zhang and Jain extract features from the DCT coefficients of MPEG compressed video streams [33]. Detection is based on the search of horizontal intensity variations followed by a morphological clean up step. Randall and Kasturi [3] compute horizontal and vertical texture energy from the DCT coefficients. Unfortunately they only obtain good results for large text. Gu [5] also uses the DCT coefficients to detect static non-moving text and removes false alarms by checking the motion vector information. One of the problems of these methods is the large number of existing video formats (MPEG 1&2, MPEG 4, Real Video, wavelet based methods etc.) and the high rate of innovation in the video coding domain, which makes methods working on compressed data quickly obsolete.

Although the research domain is very young, there exist already a high number of contributions. Only few of them present complete methods beginning from the detection of the text until the last binarization step before passing the results to an OCR. In most cases the model behind the detection algorithms is very simple: Detection of edges, regrouping into rectangles.

Text appears in videos in a wide range of writings, fonts, styles, colors, sizes, orientations etc., which makes an exact modeling of all types of text almost impossible. Since the motivation behind our method is semantic indexing, we concentrated on horizontally aligned, artificial text, as it appears e.g. in news captions. Artificial text, as opposed to scene text, has been superimposed on the signal artificially *after* capturing the signal with a camera. It has been designed to be readable. On the other hand, scene text consists of the text found *in* the scene, i.e. text on T-shirts, street signs etc. The significance of the two types of text for text extraction systems is controversial and depends on the purpose. Generally, artificial text is considered as very valuable for indexing video broadcasts (as intended by our system), whereas scene text is of higher value for source selection, i.e. processing of video sources by e.g. government agencies.

The paper is outlined as follows: In section 2 we describe our system[1] and the intermediate steps detection, tracking, image enhancement and binarization. Section 3 describes the experiments we performed to evaluate the system and gives detection results and the results of the commercial OCR system we used. Section 4 gives a conclusion and an outlook on our future research.

2 The proposed system

A global scheme of our proposed system is presented in figure 2. The detection algorithm is applied to each frame of the sequence separately. The detected text rectangles are passed to a tracking step, which finds corresponding rectangles of the same text appearance in different frames. From several frames of an appearance, a single enhanced image is generated and binarized before passing it to a standard commercial OCR software. The respective steps are given in the next sub sections.

Text in videos has gray level properties (e.g. high contrast in given directions), morphological properties (spatial distribution, shape), geometrical properties (length, ratio height/length etc.) and temporal properties (stability). Our method makes use of these properties in this order (see subsections 2.1 to 2.6), starting from the signal and going sequentially to the more domain dependent properties. The final step (a binarization presented in subsection 2.6) results in a set of binary boxes containing text which can be recognized by a classical commercial OCR system.

[1] This work is supported by France Télécom R&D in the context of project ECAV. The first part resulted in patent ref. # FR 01 06776, submitted in May 25th 2001 by France Télécom. Interactive online demonstrations of the presented detection algorithm for still images and for the binarization algorithm can be accessed at http://telesun.insa-lyon.fr/˜wolf/demos

In the following, thresholds and constants are set up to a mean image size of 388×284 pixels, which is basically the size we have to work with in MPEG 1 video sequences.

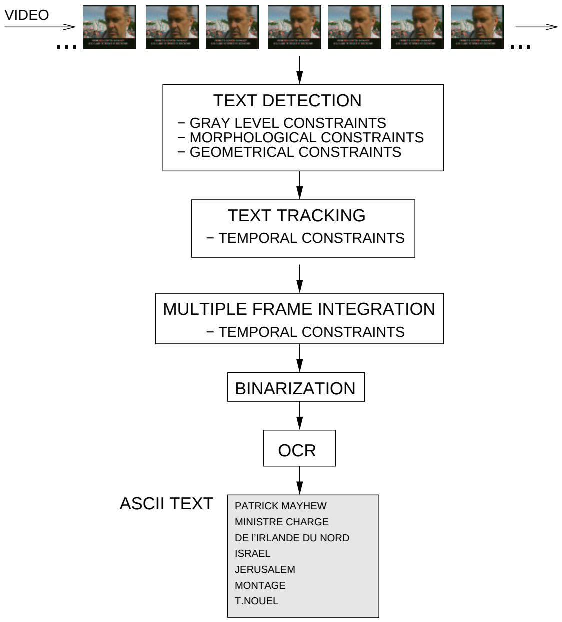


Figure 2: The scheme of our system.

2.1 Gray level constraints

As already stated before, we concentrate on horizontal artificial text. This text has been designed to be read easily. Therefore, the contrast of the characters against the background can be assumed to be high. On the other hand, we cannot assume uniform background, arbitrary complex background needs to be supported. The second assumption concerns the color properties of the text. Since the text is designed to be read easily, it can be extracted using the luminance information of the video signal only (else news cast would not be readable on black and white TVs). We therefore convert all frames into gray scale images before we start processing.

Our detection method is based on the fact that text characters form a regular texture containing vertical strokes which are aligned horizontally. We slightly modified the algorithm of LeBourgeois [11], which detects the text with a measure of accumulated gradients:

$$A(x,y) = \left[\sum_{i=-\lfloor S/2 \rfloor}^{\lfloor S/2 \rfloor} \left(\frac{\partial I}{\partial x}(x+i,y) \right)^2 \right]^{\frac{1}{2}} \qquad (1)$$

The parameters of this filter are the implementation of the partial derivative and the size S of the accumulation window. We chose the horizontal version of the Sobel operator as gradient measure, which obtained the best results in our experiments. The size of the accumulation window depends on the size of the characters and the minimum length of words to detect. Since the results are not very sensitive to this parameter, we set it to a fixed value. The filter response is an image containing for each pixel a measure of the probability to be part of text.

Binarization of the accumulated gradients is done with a two-threshold version of Otsu's global thresholding algorithm [20]. Otsu calculates an optimal threshold from the gray value histogram by assuming two Gaussian distributions in the image (class 0 for "non-text" and class 1 for "text" in our case) and maximizing a criterion used in discriminant analysis, the inter-class variance:

$$t = \arg\max_t (\omega_0 \ \omega_1 (\mu_1 - \mu_0)^2) \qquad (2)$$

where $\omega_0 = \sum_{i=1}^{t} \frac{H(i)}{N}$ is the normalized mass of the first class, $\omega_1 = \sum_{i=t+1}^{L} \frac{H(i)}{N}$ is the normalized mass of the second class, μ_0 and μ_1 are the mean gray levels of the respective classes, H denotes the histogram, N the number of pixels and L the number of bins of the histogram.

In order to make the binarization decision more robust, we added a second threshold and changed the decision for each pixel as follows:

$$\begin{array}{lll} I_{x,y} < k_l & \Rightarrow & B_{x,y} = 0 \\ I_{x,y} > k_h & \Rightarrow & B_{x,y} = 255 \\ k_l \leq I_{x,y} \leq k_h & \Rightarrow & B_{x,y} = \begin{cases} 255 & \text{if there is a path}^2 \text{ to} \\ & \text{a pixel } I_{u,v} > k_h \\ 0 & \text{else} \end{cases} \end{array} \qquad (3)$$

where k_h is the optimal threshold calculated by Otsu's method and k_l is calculated from k_h and the first mode m_0 of the histogram: $k_l = m_0 + \alpha(k_h - m_0)$, where α is a parameter. The result of this step is a binary image containing text pixels and background pixels.

2.2 Morphological constraints

A phase of mathematical morphology follows the binarization step for several reasons:

- To reduce the noise.
- To correct classification errors using information from the neighborhood of each pixel.
- To distinguish text from regions with texture similar to text based on some assumptions (e.g. minimal length).
- To connect loose characters in order to form complete words.

[2] only made of pixels (x,y) such that $I_{x,y} > k_l$.

- To make the detection results less sensitive to the size of the accumulation window and to cases where the distances between characters are large.

The morphological operations consist of the following steps:

Step 1: Close (1 iteration). Structuring element:

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \textcircled{1} & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4}$$

This first preliminary step closes small holes in the components and connects components separated by small gaps.

Step 2: Suppression of small horizontal bridges between connected components.

This step of the morphological phase has been designed to deconnect connected components corresponding to text regions from connected components corresponding to non-text regions. It removes small bridges between connected components by suppressing pixels of columns whose local connected component's height is under a certain threshold. To do this, we first build a matrix A of the same size as the input image. Each element of the matrix corresponds to a pixel of the image and holds the local height of the connected component it belongs to. Finally, a thresholding step removes pixels (x, y) such that $A_{x,y} < t_1$, where t_1 is a fixed parameter.

A side effect of this operation may be the separation of a noisy text region into two or more connected components. Also, in cases where the distances between characters are high, text regions may be decomposed in several connected components. Steps 3 and 4 of the morphological phase address this problem:

Step 3: Conditional dilation (16 iterations). Structuring element:

$$B_C = \begin{bmatrix} \textcircled{1} & 1 \end{bmatrix} \tag{5}$$

The conditional dilation and erosion algorithms have been developed to connect loose characters in order to form words, we thus want to merge all connected components which are horizontally aligned and whose heights are similar. The dilation algorithm consists of a standard dilation algorithm using additional conditions on the shape of the connected components of the image. First, a connected components analysis for 4-connected neighborhoods is performed on the binary input image. We assume, that each component C_i corresponds to a character or a word. The component height is used in the following dilation step. To ease the implementation, we build a height matrix H of the same size as the binary input image. Every element of the matrix corresponds to a pixel of the input image and contains the height of the respective connected component. A second matrix P is constructed, which holds for each pixel the y coordinate of the corresponding connected component. Next, the non-zero values of the matrices are "smeared" to the left over the zero values, i.e. each zero value is replaced with its closest non-zero neighbor to the right of the same row:

$$H_{x,y} = \begin{cases} H_{x,y} & \text{if } H_{x,y} \neq 0 \\ H_{v,y} : \, v = \min\limits_{u>x}(H_{u,y} > 0) & \text{else} \end{cases} \tag{6}$$

The same treatment is also applied to the matrix P.

The matrices H and P contain the information necessary to perform the conditional dilation. The algorithm traverses each line of the image from the left to right and each non-text pixel (x,y) preceded by a text pixel is set to text (i.e. dilation) under certain conditions. These conditions are based on the relative differences of height (respectively the y coordinate) of the component corresponding to the set pixel to the left of the current pixel and the height of the neighboring component to the right. These heights can be taken directly from the matrix H. Since the test for dilation is performed on a non-set pixel, the matrix H contains the height of the next neighboring component to the right (see equation 6). This situation is illustrated in figure 3.

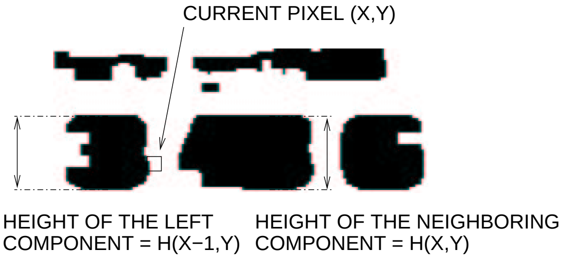

CURRENT PIXEL (X,Y)
HEIGHT OF THE LEFT COMPONENT = H(X−1,Y)
HEIGHT OF THE NEIGHBORING COMPONENT = H(X,Y)

Figure 3: Determining the heights of neighboring connected components.

The difference function used for the conditions is defined as follows:

$$\Delta(a, b) = \frac{\mid a - b \mid}{\min(a, b)} \tag{7}$$

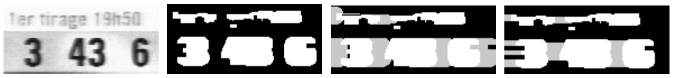
Dilation is performed, if:

- The relative difference of the heights of the connected component including the given pixel and the neighboring component to the right does not exceed a threshold t_2, i.e. $\Delta(H(x - 1, y), H(x, y)) < t_2$
- The relative difference of the y positions of these two connected components does not exceed a threshold t_3, i.e. $\Delta(P(x - 1, y), P(x, y)) < t_3$

If the test is valid and the maximum number of dilations (i.e. the number of iterations) since the last dilated text pixel has not been passed, then the pixel is set to a pseudo color which is neither 0 (non-set) nor 255 (set).

Step 4: Conditional erosion (16 iterations). Structuring element: B_C.

The conditional erosion step performs a “standard” morphological erosion using the same structuring element B_C, but contrary to the conditional dilation algorithm, there are additional conditions on the gray levels of the pixels instead of the shape. The image is eroded using the structuring element B_C with the condition that only pixels marked with the pseudo color are eroded. Finally, all pixels marked with the pseudo color are marked as text. The effect of this step is the connection of the all connected components which are horizontally aligned and whose heights are similar.



(a) (b) (c) (d)

Figure 4: Original image (a) binarized accumulated gradients (b) after conditional dilation (c) and after conditional erosion (d).

An example for the two conditional operations (Steps 3 and 4) is shown in figure 4. Figure 4a shows the original input image. Figure 4b shows the binarized accumulated gradients. The words are separated and there also gaps between the digits. Figure 4c shows the pseudo color image after the conditional dilation step. Figure 4d shows the image after the conditional erosion. Since the neighboring components are similar in height and they are aligned horizontally, the pixels between them are dilated, which connects these components.

Step 5: Horizontal erosion (12 iterations). Structuring element:

$$B_H = \left[\begin{array}{ccc} 1 & \textcircled{1} & 1 \end{array}\right] \tag{8}$$

This very strong horizontal erosion removes components which do not respect the hypothesis, that text has a certain minimum length. This hypothesis, which has already been imposed on the gray levels as well (text consists of vertical strokes arranged horizontally with a minimum length), distinguishes between small components containing high contrast strokes and text with a minimum length.

Step 6: Horizontal dilation (6 iterations). Structuring element: B_H.
This dilation step resizes the remaining components to almost their original size (i.e. the estimated size of the text rectangles). Instead of performing a horizontal dilation with the same number of iterations, which bears the danger of reconnecting text components with non-text components, only half of the iterations are performed. A connected components analysis extracts the remaining components, whose bounding boxes are used for further processing. Finally, the bounding boxes are grown horizontally 3 pixels to the left and 3 pixels to the right in order to compensate for the difference in the horizontal erosion and dilation step.

Figure 5 shows the intermediate results after imposing the gray level constraints and the morphological constraints. Figures 5a and 5b display the original image and the gradient calculated by the horizontal Sobel filter. Figures 5c shows the accumulated gradients, where text areas are visible as emphasized white rectangles. Figure 5d shows the binarized image which contains white text components and black non-text background, but also white non-text components due to the background texture in the original image. Almost all these are removed after imposing the morphological constraints, shown in figure 5e. Figure 5f shows the original image with superimposed bounding boxes of the connected components.

2.3 Geometrical constraints

After the morphological processing, the following geometrical constraints are imposed on the rectangles in order to further decrease their number:

$$\frac{width}{height} > t_4 \tag{9}$$

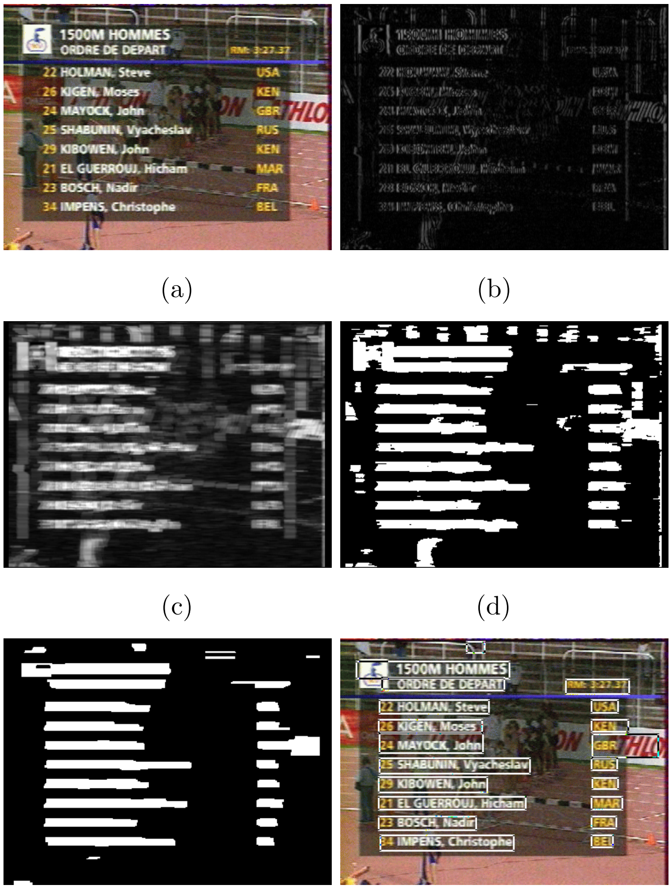
$$\frac{number\ of\ text\ pixels\ of\ the\ component}{area\ of\ the\ bounding\ box} > t_5 \tag{10}$$

where t_4 and t_5 are fixed thresholds.

Some special cases are considered, which increase the size of the bounding boxes of the rectangles to include the heads and tails of the characters, which have been removed by the morphological step. Finally, rectangles are combined in order to decrease their number. Two (partly) overlapping rectangles are combined into a single one of bigger dimension, if *one* of the following two conditions holds:

$$\begin{aligned} C_1 : &\quad \frac{A(R_s) - A(R_b \cap R_s)}{A(R_s)} < t_6 \\ C_2 : &\quad \frac{A(R_s) - A(R_b \cap R_s)}{A(R_s)} < t_8 \quad \wedge \quad \frac{A(R_s)}{A(R_b)} < t_7 \end{aligned} \tag{11}$$

where R_b is the bigger rectangle, R_s is the smaller rectangle, $A(R)$ is the area of rectangle R, and $t_6 - t_8$ are fixed thresholds which respect the condition $t_6 < t_8$. In plain words, if the non-overlapping part of the small rectangle (i.e. the part which does not overlap with the bigger rectangle) is smaller than threshold t_6, then the two rectangles are joined. Condition 2 states, that the two rectangles are even joined if the non-overlapping part is bigger than t_6 but still smaller than the second threshold t_8 (since $t_8 > t_6$), if the difference in size of the two rectangles is big enough. In other words, if a very small rectangle is glued to



(a) (b)

(c) (d)

(e) (f)

Figure 5: The intermediate results during the detection process: The input image (a), the gradient (b), the accumulated gradient (c), the binarized image (d), The image after morphological post processing (e), the final result (f).

a bigger rectangle, then the two are joined even if the overlap area is not as big as required by condition 1.

2.4 Temporal constraints - Tracking

It is obvious that text has to remain during a certain amount of successive frames in order to be readable. We take advantage of this property in order to create appearances of text and filter out some false alarms. To achieve this, we use the overlap information between the list of rectangles detected in the current frame and the list of currently active rectangles (i.e. the text detected in the previous frames which is still visible in the last frame). During the tracking process we keep a list $\bar{L}$ of currently active text appearances. For each frame i, we compare the list L^i of rectangles detected in this frame with list $\bar{L}$ in order to check whether the detected rectangles are part of the already existing appearances.

The comparison is done by calculating the overlap area between all rectangles $\bar{L}_k$ of the list of active rectangles and all rectangles L^i_l of the current frame i. For all pairs (k, l) with non-zero overlap $A(\bar{L}_k \cap L^i_l)$, the following conditions are verified:

- The difference in size is below a certain threshold.
- The difference in position is below a certain threshold (We assume non moving text).
- The size of the overlap area is higher than a certain threshold.

If for a new rectangle L^i_l one or more active appearances respecting these conditions are found, then the one having the maximum overlap area is chosen and the rectangle L^i_l is associated to this active appearance $\bar{L}_k$. If no appearance is found, then a new one is created (i.e. new text begins to appear on the screen) and added to the List $\bar{L}$. Active appearances in list $\bar{L}$, which are not associated to any rectangles of the current frame, are not removed immediately. Instead, they are kept another 5 frames in the list of active frames. This compensates for possible instabilities in the detection algorithm. If during these 5 frames no new rectangle is found and associated to it, then the appearance is considered as finished, removed from the list $\bar{L}$ and processing of this appearance begins.

Two additional temporal constraints are imposed on each appearance. First we use the length of the appearance as a stability measure, which allows us to distinguish between temporarily unstable appearances and stable text. Only frames which stay on the screen for a certain minimum time are considered as text. Since we concentrate on artificial text, which has been designed to be readable, this hypothesis is justified[3].

The second temporal constraint relates to the detection stability. Like stated above, an active appearance is not finished immediately if new corresponding text is found in the current frame, but kept for another 5 frames. If text is found within these 5 frames then it is associated to the appearance. Therefore, “holes” are possible in each appearance, i.e. frames which do not contain any detected text. If the number of these holes is too large, then we consider the appearance as too unstable as to be text. The two conditions, which result in rejection, can be summarized as:

$$\textit{number of frames} < t_9 \tag{12}$$

$$\frac{\textit{number of frames containing text}}{\textit{number of frames}} < t_{10} \tag{13}$$

[3] There is a type of text which stays on the screen for only a very small amount of time (1 or 2 frames), the so-called subliminal messages, which are most times intended to convince people to buy products without their conscious knowledge. We ignore this type of text.

2.5 Temporal constraints - multiple frame integration

Before passing the images of a text appearance to the OCR software, we enhance their contents by creating a single image of better quality from all images of the sequence. We also increase their resolution, which does not add any additional information, but is necessary because the commercial OCR programs have been developed for scanned document pages and are tuned to high resolutions. Both steps, interpolation and multiple frame enhancement, are integrated into a single, robust algorithm.

The area of the enhanced image $\hat{F}$ consists of the bounding box of all text images F_i taken from the T frames i of the sequence. Each image F_i is interpolated robustly in order to increase its size:

$$F_i' = \uparrow (F_i, M, S) \qquad i = 1..T$$

where the image M contains for each pixel the temporal mean of all images and S the temporal standard deviation. Hence, the interpolation function $\uparrow$ takes into account not only the original image but also information on the temporal stability of each pixel. Since the size of the interpolated image is larger than the size of the original image, from now on we will denote coordinates in the interpolated images with a prime (e.g. $\hat{F}(p', q'), F_i'(p', q')$) as opposed to the coordinates in the uninterpolated text images (e.g. $F_i(p,q), M(p,q)$). The interpolation is applied to all frame images F_i, the final enhanced image $\hat{F}$ being the mean of the interpolated images:

$$\hat{F}(p', q') = \frac{1}{T} \sum_{i=1}^{T} F_i'(p', q')$$

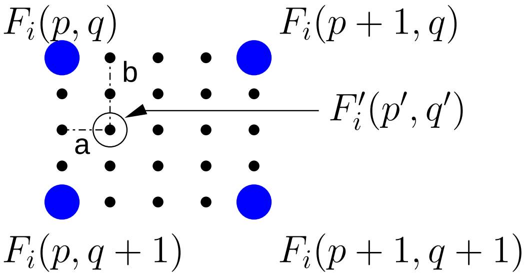


Figure 6: Bi-linear interpolation (factor 4x). The interpolated pixel with coordinates (p´,q´) is calculated from the its 4 neighbors with coordinates (p,q), (p+1,q), (p,q+1) and (p+1,q+1).

We used two different interpolation functions, the bi-linear and the bi-cubic method, and adapted them to be more robust to temporal outliers. In the bi-linear algorithm, the gray value of each pixel $F_i'(p', q')$ is a linear combination of the gray values of its 4 neighbors ($F_i(p,q)$, $F_i(p+1,q)$, $F_i(p,q+1)$ and $F_i(p+1,q+1)$), where

$$p = \left\lfloor \frac{p'}{u} \right\rfloor \tag{14}$$

$$q = \left\lfloor \frac{q'}{u} \right\rfloor$$

and u is the interpolation factor (see figure 6).

The weights $w_{m,n}$ for each neighbor $F_i(p+m, q+n)$ ($m,n \in [0,1]$) depend on the distance between the pixel and the respective neighbor (calculated through the horizontal and vertical distances a and b respectively, between the pixel and the reference neighbor $F_i(p,q)$):

$$a = \frac{p'}{u} - \left\lfloor \frac{p'}{u} \right\rfloor \tag{15}$$

$$b = \frac{q'}{u} - \left\lfloor \frac{q'}{u} \right\rfloor$$

From this distance, the weights are calculated as follows:

$$\begin{array}{rcl} w_{0,0} & = & (1-a)\cdot(1-b) \\ w_{1,0} & = & a \cdot (1-b) \\ w_{0,1} & = & (1-a)\cdot b \\ w_{1,1} & = & a \cdot b \end{array} \tag{16}$$

To increase the robustness of the integration process, we added an additional weight $g^i_{m,n}$ for each neighbor $F_i(p+m, q+n)$ to the interpolation scheme which decreases the weights of outlier neighbor pixels :

$$g^i_{m,n} = \left(1 + \frac{|F_i(p+m,q+n) - M(p+m,q+n)|}{1 + S(p+m,q+n)}\right)^{-1} \tag{17}$$

The final weight for neighbor $F_i(p+m, q+n)$ is therefore $w_{m,n} g^i_{m,n}$, resulting in the following equation for the interpolated pixel $F_i'(p', q')$ of a given frame F_i:

$$F_i'(p', q') = \frac{\sum_{m=0}^{1} \sum_{n=0}^{1} w_{m,n} g^i_{m,n} F_i(p+m, q+n)}{\sum_{m=0}^{1} \sum_{n=0}^{1} w_{m,n} g^i_{m,n}} \tag{18}$$

In computer vision, bi-linear interpolation is not considered as one of the best interpolation techniques, because of the low visual quality of the produced images. Bi-cubic interpolation produces images which are more pleasing to the human eye, so we adapted it in the same manner as the bi-linear technique to be robust to temporal outlier pixels.

The bi-cubic interpolation method passes a cubic polynomial through the neighbor points instead of a linear function. Consequently there are 16 neighbor points needed instead of 4 to be able to calculate a new image pixel $F_i'(p', q')$ (see figure 7).

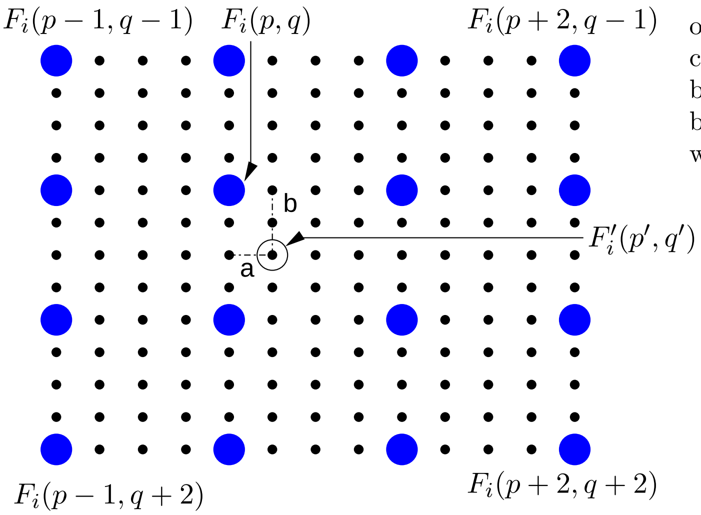


Figure 7: Bi-cubic interpolation (factor 4x).

As in the bi-linear interpolation scheme, the weights for the different neighbors $F_i(p+m,q+n)$ depend on the distances to the interpolated pixel and are calculated using the horizontal distances a and b to the reference point $F_i(p,q)$ (see figure 7). However, instead of setting the weights proportional to the distance, the weight $w_{m,n}$ for each neighbor is calculated as

$$w_{m,n} = R_c(m-a)R_c(-(n-b)) \tag{19}$$

where R_c is the cubic polynomial

$$R_c(x) = \frac{1}{6}(x^3 - 3x^2 - 12x + 9) \tag{20}$$

and

$$(x)^m = \begin{cases} x^m & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{21}$$

The interpolated pixel $F_i'(p',q')$ can therefore be calculated by the equation

$$F_i'(p',q') = \frac{\sum_{m=-1}^{2}\sum_{n=-1}^{2} w_{m,n} g_{m,n}^i F_i(p+m,q+n)}{\sum_{m=-1}^{2}\sum_{n=-1}^{2} w_{m,n} g_{m,n}^i} \tag{22}$$

Figure 8 shows an example for a sequence of multiple frames integrated into a single enhanced image by three different methods. Each result image is also thresholded by a manually selected threshold (right column). Figure 8a shows the result using a standard bi-linear interpolation on each frame and averaging for the integration into a single image. The result of the robust version of the bi-linear interpolation presented in Figure 8b is less noisy at the edges, which is clearly visible at the edges of the thresholded image. The image created by the robust bi-cubic method presented in Figure 8c is visually more attractive but also much smoother. Which interpolation technique is the best will be decided by the OCR results given in section 3, i.e. we employ a goal directed selection of techniques.

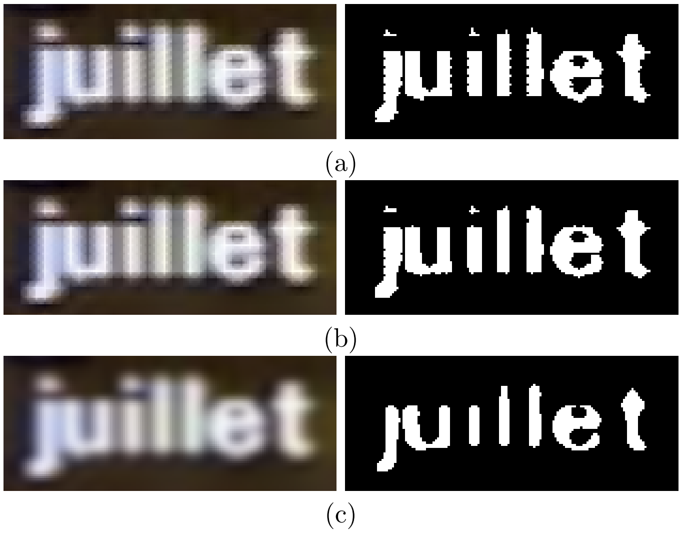


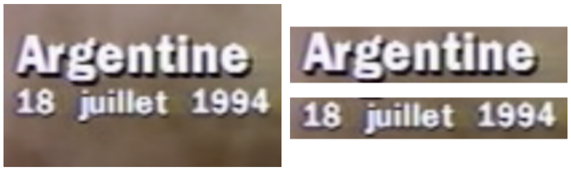
Figure 8: Interpolated images (left), after a manual thresholding (right): Bi-linear (a) robust bi-linear (b) robust bi-cubic (c).

After the multiple frame integration step, the enhanced image is of a better quality than the original frame images. This allows us to perform some additional segmentation algorithms at this stage of the process. Due to noise and background texture in the original signal, the detected text boxes may contain several lines of text in a single box. The smoothed background in the enhanced image makes it easier to separate the boxes into several boxes containing only one text line each. To do this, we redo the steps already performed to detect the text in a single frame, i.e. we recalculated the accumulated gradients on the enhanced image and binarize them. By searching peaks in the horizontal projection profiles of the text pixels of the binarized image, we are able to extract single lines. Figure 9 shows an example rectangle before and after the separation into single lines.

Additionally, since the obtained text boxes now contain only one line of text each, we again impose geometrical constraints on these rectangles, i.e. we remove all rectangles having a ratio of width/height smaller than a threshold t_{11}, where $t_{11} > t_4$.

2.6 Final binarization

As a final step, the enhanced image needs to be binarized. Several binarization algorithms have been developed by the computer vision and the document analysis community. Good



(a) (b)

Figure 9: The enhanced image before separation (a) and after separation into two images with one text line each (b).

surveys of existing techniques are [22] for the general thresholding case and the well known paper of Trier and Jain on binarization techniques for document images [21]. Therein, the methods have been evaluated by comparing the results of a following OCR phase. The approaches considered as the best are Niblack's method [19] and Yanowitz-Bruckstein's method [32], where Niblack's is one of the simplest and fastest algorithms and Yanowitz-Bruckstein's one of the most complex and slowest. Trier et al. performed their experiments on "traditional" documents produced by scanners.

As already stated, the images extracted from videos of low quality do not have the same characteristics as scanned document images. In our experiments we found out, that for our purposes, the simpler algorithms are more robust to the noise present in video images. We therefore chose Niblack's method for our system, and finally derived a similar method based on a criterion maximizing local contrast.

Niblack's algorithm calculates a threshold surface by shifting a rectangular window across the image. The threshold T for the center pixel of the window is computed using the mean m and the variance s of the gray values in the window:

$$T = m + k \cdot s \tag{23}$$

where k is a constant set to -0.2. The results are not very sensitive to the window size as long as the window covers at least 1-2 characters. However, the drawback is noise created in areas which do not contain any text (due to the fact that a threshold is created in these cases as well). Sauvola et al. [24] solve this problem by adding a hypothesis on the gray values of text and background pixels (text pixels have gray values near 0 and background pixels have gray values near 255), which results in the following formula for the threshold:

$$T = m \cdot (1 - k \cdot (1 - \frac{s}{R})) \tag{24}$$

where R is the dynamics of the standard deviation fixed to 128 and k is fixed to 0.5. This method gives better results for document images, but creates additional problems for video frames whose contents do not always correspond to the hypothesis, even if images containing bright text on dark background are reversed to resemble better the desired configuration (see section 3 on the experimental results for examples).

We propose to normalize the different elements used in the equation which calculates the threshold T, i.e. formulate the binarization decision in terms of contrast instead of in terms of gray values, which is a natural way considering the motivation behind Niblack's technique[4]. How can we define contrast? If we refer to Niblack [19], p. 45, then the contrast of the center pixel of a window of gray levels is defined as

$$C_L = \frac{|m - I|}{s} \tag{25}$$

Since we suppose dark text on bright background, we do not consider points having a gray value which is higher than the local mean, therefore the absolute value can be eliminated. Denoting M the minimum value of the gray levels of the whole image, the maximum value of this local contrast is given by

$$C_{max} = \frac{m - M}{s} \tag{26}$$

It is difficult to formulate a thresholding strategy defined only on the local contrast and its maximum value, since this does not take into account the variation of the window with respect to the rest of the image. This is the reason why we also propose the definition of a more global contrast, the contrast of the window centered on the given pixel:

$$C_W = \frac{m - M}{R} \tag{27}$$

where $R = max(s)$ is the maximum value of the standard deviations of all windows of the image. This contrast tells us, if the window is rather dark or bright with respect to the rest of the image (a high value characterizes the absence of text). Now we can express our binarization strategy in terms of these contrasts. A simple thresholding criterion is to keep only pixels which have a high local contrast compared to it's maximum value corrected by the contrast of the window centered on this pixel:

$$I : C_L > a(C_{max} - C_W) \tag{28}$$

where a is a gain parameter. Developing this equation we obtain the threshold value:

$$T = (1 - a)m + aM + a\frac{s}{R}(m - M) \tag{29}$$

In the case where the given pixel is the center of a window with maximum contrast, i.e. $s = R$, we get $T = m$. Thus, the algorithm is forced to keep the maximum number of points of the window. On the other hand, if the variation is low ($s << R$), then the probability that the window contains text is very

[4] Niblack derived his method from algorithms to transform gray level histograms in order to increase the contrast in images, e.g. to display them.

low. We therefore keep a pixel only if it's local contrast is very high. The threshold is therefore $T \approx (1-a)m + aM$. The gain parameter a allows to control the incertitude around the mean value. A simple solution is to fix it at 0.5, which situates the threshold between m and M.

The computational complexity of this new technique is slightly higher than Niblack's version, since one of the thresholding decision parameters is the maximum standard deviation of all windows of the image. Therefore two passes are required. On the other hand, it is not necessary to shift the window twice across the image, if the mean value and standard deviations for each pixel (i.e. each window) are stored in two-dimensional tables.

Furthermore, for the calculation of the mean and standard deviation, only the first window for the first pixel of each line needs to be traversed entirely. The mean value and the standard deviation of the other windows can be calculated incrementally from the preceding windows, if for each window the sum of the gray values and the sum of the squares of the gray values are held.

3 Experimental results

To estimate the performance of our system we carried out exhaustive evaluations using a video database containing 60.000 frames in 4 different MPEG 1 videos with a resolution of 384×288 pixels and 113 still images with resolutions between 320×240 and 384×288. The videos provided by INA[5] contain 322 appearances of artificial text (see figure 10) from the French television channels TF1, France 3, Arte, M6 and Canal+. They mainly contain news casts and commercials. We manually created ground truth for each video sequence to be able to evaluate the results[6]. The ground truth also allowed us to optimize the parameters of the system. Table 1 shows the parameters we determined from our test database.

3.1 Detection performance in still images

The images of our still image database contain artificial horizontal text of various sizes. In order to automatically determine the parameters of the system, the detected text boxes are compared to the ground truth and precision (P) and recall (R) values are calculated, taking into account the overlap information between the ground truth rectangle GR and the detected rectangle DR:

$$R = \frac{\text{area of } (GR \cap DR)}{\text{area of } GR} \qquad P = \frac{\text{area of } (GR \cap DR)}{\text{area of } DR} \qquad (30)$$

Figure 11 shows these values combined for the still image database for the four parameters S (accumulation length), α (binarization), and t_4 and t_5 (geometrical constraints). The single precision and recall values are weighted by the sizes of the rectangles before averaging them. As can be seen, the detection performance is robust to small changes in the parameters.

The measures described above are not sufficient for a reliable evaluation of the detection performance, as can be noticed in the example image in figure 12. The text box containing "sécurité" has been detected with a recall value of R=89% but nevertheless a character is cut off. The word "routière" has been detected with R=100% but the detection has been scattered across three different rectangles. Therefore, instead of choosing the parameters on the basis of the precision and recall values only, we chose them according to a performance measure which also takes into account the amount of text rectangles matched against a ground truth rectangle as well as the spatial distribution of the overlap area, i.e. we verify whether the non-detected text pixels are located around the rectangle or concentrated in one area, which tends to cut off characters.

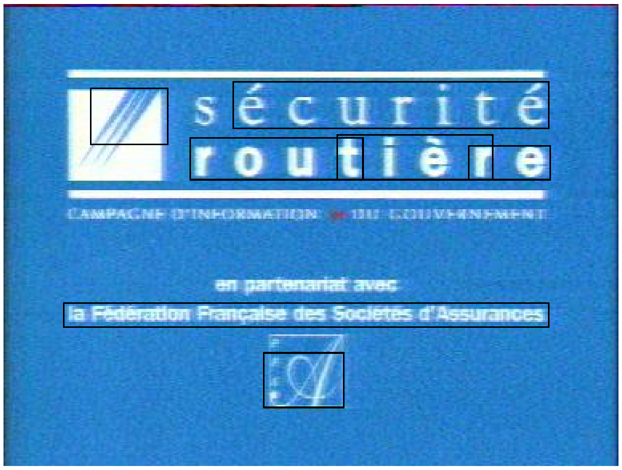

sécurité
routière
CAMPAGNE D'INFORMATION DU GOUVERNEMENT
en partenariat avec
la Fédération Française des Sociétés d'Assurances

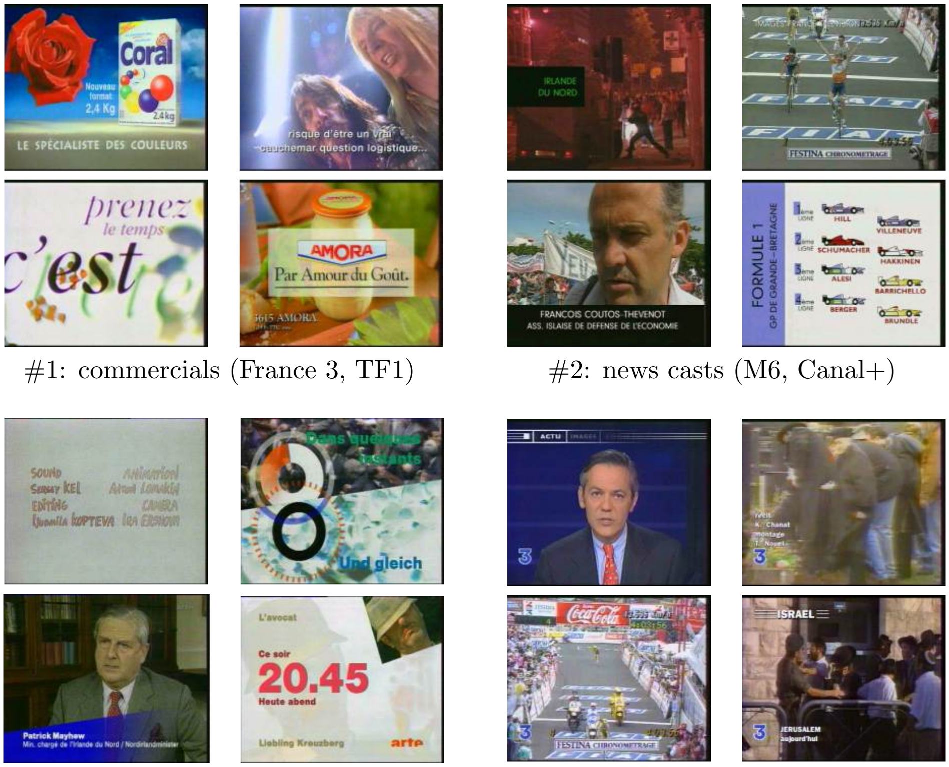
Figure 12: Detection result for an example image.

3.2 Detection performance in video sequences

The automatic scheme described above for the evaluation of the detection in still images is very difficult to extend to the detection in video sequences and bears the risk to be error prone. We therefore decided to perform a manual evaluation of the detection in video sequences by deciding manually for each detected text box whether is has been detected correctly or not. Hence, the precision and recall values we give in this section are not based on surfaces but on counts only (i.e. the number of detected rectangles, number of existing rectangles etc.).

[5] The *Institut National de l'Audiovisuel* (INA) is the French national institute in charge of the archive if the public television broadcasts. See http://www.ina.fr

[6] The ground truth files can be downloaded from http://rfv.insa-lyon.fr/˜wolf/groundtruth



#1: commercials (France 3, TF1)

#2: news casts (M6, Canal+)

#3: cartoons and news casts (Arte)

#4: news casts (France 3)

Figure 10: Examples of our test database.

For object classification tasks, where a special object property (e.g. *the object is text*) needs to be detected, the evaluation results are traditionally given in a confusion matrix. The matrix consists of 4 different cases:

- the object is text and has been detected as text.
- the object is non-text and has been detected as non-text.
- the object is text and has been detected as non-text.
- the object is non-text and has been detected as text.

Our system for text detection in videos is different to traditional classification systems in various ways. There is no such figure as *the object is non-text and has been detected as non-text*, since text which does not exist and is not detected, cannot be counted. Furthermore, an actual text string present in the ground truth may probably correspond to several detected text rectangles, since the detection process may detect a text occurrence more than once. Therefore, the rows and columns of such a confusion matrix do not sum up correctly.

We therefore present detection results in a different way, as shown in table 2. The table consists of two parts. The upper part is ground truth oriented, i.e. the categories are parts of the set of ground truth rectangles. It shows the number of rectangles in the ground truth which have been correctly detected (*Class. T*) and missed (*Class. NT*). The total number of rectangles in the ground truth is shown as *Total GT*. From these figures the recall of the experiment can be calculated as

$$Recall = \frac{Class.\ T}{Total\ GT} \tag{31}$$

The lower part of the table is detection related, i.e. the categories are parts of the set of detected rectangles. It shows the number of correctly detected artificial text rectangles (*Positives*) [7], the number of false alarms (*FA*), the number of rectangles corresponding to logos (*Logos*) and detected scene text rectangles (*Scene Text*). A total number of detected rectangles is given as *Total det.* The precision of the experiment can be calculated as

$$Precision = \frac{Positives + Logos + Scene\ Text}{Total} \tag{32}$$

This figure is based on the number of artificial text, scene text and logos (*Total-FA* = *Positives* + *Logos* + *Scene Text*), since

[7] Like stated above, the number of Positives may differ from the number of corrected classified ground truth rectangles (*Class. T*) if text has been detected as more than one text appearance.

Par.	Value	Description
S	13	Size of the gradient accumulation filter.
α	0.87	Determination of the second threshold for the binarization of the accumulated gradients.
t_1	2	Threshold on column height.
t_2	1.05	Threshold on height difference.
t_3	0.5	Threshold on position difference.
t_4	1.2	Threshold on ratio width/height.
t_5	0.3	Threshold on ratio pixels/area.
t_6	0.1	Threshold for combining rectangles.
t_7	0.2	Threshold for combining rectangles.
t_8	0.7	Threshold for combining rectangles.
t_9	40	Threshold on appearance length.
t_{10}	0.4	Threshold on appearance stability.
t_{11}	1.5	Threshold on ratio width/height.
a	0.5	Parameter for the binarization of the text boxes.

Table 1: The parameters of the system.

the latter two types of text cannot be considered as false alarms, although the detection system has not been designed with their detection in mind.

As shown in table 2, we achieve an overall detection rate of 93.5% of the text appearing in the video. The remaining 6.5% of missing text are mostly special cases, which are very difficult to treat, and very large text (larger than a third of the screen), which could be detected using a multi resolution method, but at the additional cost of a higher number of false alarms. The precision of the system is situated at 34.4%, which means that there are unfortunately a large number of false alarms. This is due to the fact, that there exist many structures with similar properties as text, which can only be distinguished by techniques using recognition. The number of false alarms can be decreased by changing the parameters of the constraints we introduced in sections 2.1 to 2.4, but this will lower the high detection recall.

	Video files					Precision,
Category[8]	#1	#2	#3	#4	Total	Recall
Class. T	102	80	59	60	301	**93.5%**
Class. NT	7	5	4	5	21	
Total GT	109	85	63	65	322	
Positives	114	78	72	86	350	
FA	138	185	374	250	947	
Logos	12	0	34	29	75	
Scene text	22	5	28	17	72	
Total - FA	148	83	134	132	497	**34.4%**
Total det.	286	268	508	382	1444	

Table 2: The detection results for video sequences. The results are given on text box level.

3.3 Binarization performance

Figure 13 shows 4 images taken from the set of detected and enhanced images and results of the different binarization techniques described in the previous section together with other

[8] T=Text, NT=Non-Text, GT=Ground truth, FA=False alarms

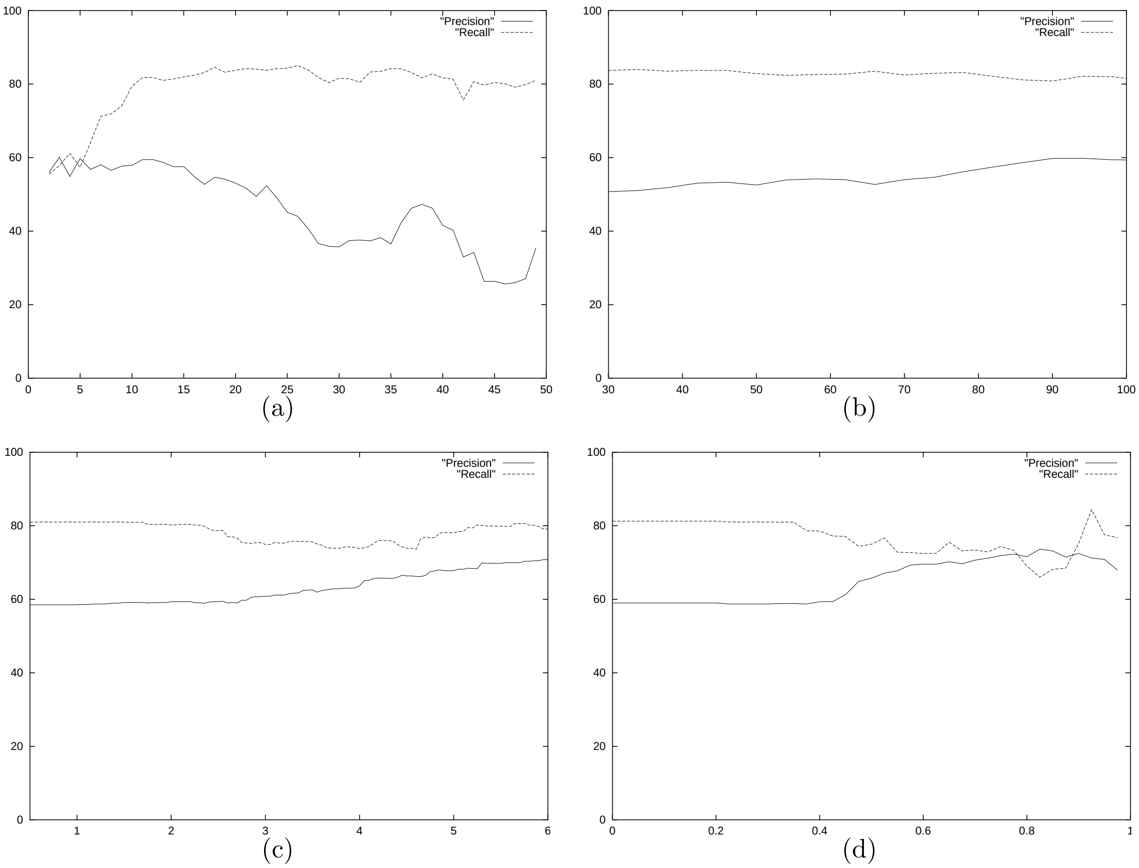


Figure 11: Precision and recall on the still images database for different parameter values of S (a) α (b) t_4 (c) and t_5 (d).

standard methods from image processing:

- Otsu's method derived from discriminant analysis [20](see section 2.1 on the binarization of the accumulated gradients).
- A local version of Otsu's method, where a window is shifted across the window and the threshold is calculated from the histogram of the gray values of the window.
- Yanowitz-Bruckstein's method [32], which detects the edges in image and then passes a threshold surface through the edge pixels.
- Yanowitz-Bruckstein's method including their post-processing step, which removes ghost objects by calculating statistics on the gray values of the edge pixels of the connected components.
- Niblack's method [19] with parameter $k = -0.2$ (see section 2.6).
- Sauvola et al.'s method [24] with parameters $k = 0.5$ and $R = 128$ (see section 2.6).
- Our contrast based method with parameter $a = 0.5$ (see section 2.6).

As can be seen in Figure 13, Yanowitz-Bruckstein's edge based method is very sensitive to noise and very much depends on the quality of edge detection, even if the post-processing step is employed. Otsu's method suffers from the known disadvantages of global thresholding methods, i.e. unprecise separation of text and background in case of high variations of the gray values. The windowed version improves the result (the characters are better segmented), but creates additional structure in areas where no text exists. This is due to the fact, that a threshold separating text and background is calculated even in areas where only background exists.

The same problem can be observed for Niblack's method, which segments the characters very well, but also suffers from noise in the zones without text. Sauvola's algorithm overcomes this problem with the drawback that the additional hypothesis causes thinner characters and holes. Our solution solves this problem by combining Sauvola's robustness vis-a-vis background textures and the segmentation quality of Niblack's method. An evaluation of the last three binarization techniques using OCR are given in the next subsection.

3.4 OCR performance

In this sub section we present a goal directed evaluation of the performance of our algorithms. We passed the final binarized text boxes to a commercial OCR product (Abbyy Finereader 5.0) for recognition. We processed each of the 4 MPEG files seperately. Furthermore, the rectangles for file #3, which contains contents of the French-German channel Arte, are split up into two sets containing French and German text respectively, and treated with different dictionaries during the OCR step.

To measure the OCR performance, we used the similarity measure introduced by Wagner and Fisher [28]. The resemblance is defined as the minimal cost of transformations $\delta(x,y)$ of the ground truth string x to the corresponding output string y. The transformation is realized as a sequence of the following basic operations:

Substitution of a character:	$a \rightarrow b$	cost $\gamma(a,b)$
Insertion of a character:	$\lambda \rightarrow b$	cost $\gamma(\lambda,b)$
Deletion of a character:	$a \rightarrow \lambda$	cost $\gamma(a,\lambda)$

Under the assumption that $\delta(a,b)=\gamma(a,b)$, i.e. that the minimal distance of two characters is the cost of changing one character into another, this transformation also assigns each correctly recognized character its partner character in the ground truth string. It is therefore easy to compute the number of correctly recognized characters. Additionally to the cost, we give the traditionally used measures of precision and recall to measure the performance of the OCR phase:

$$Recall = \frac{Correctly\ recognized\ characters}{Characters\ in\ the\ ground\ truth} \qquad (33)$$

$$Precision = \frac{Correctly\ recognized\ characters}{Characters\ in\ the\ OCR\ output} \qquad (34)$$

The evaluation measures, cost as well as recall and precision, depend on the character cost function, which we implemented for our experiments as follows (λ specifies the empty character):

$$\gamma(\alpha,\beta) = \begin{cases} 0 & \text{if } \alpha=\beta \quad , \quad \alpha,\beta \in X \cup \{\lambda\} \\ 0.5 & \text{if } \alpha=\beta \quad , \quad \alpha,\beta \in X \cup \{\lambda\} \quad , \\ & \alpha \text{ and } \beta \text{ have different cases} \\ 1 & \text{else} \end{cases} \qquad (35)$$

$$\gamma(\lambda,\beta) = \begin{cases} 0.5 & \text{if } \beta \text{ is a white space} \\ 1 & \text{else} \end{cases} \qquad (36)$$

$$\gamma(\lambda,\beta) = \gamma(\beta,\lambda) \qquad (37)$$

where X is the alphabet of the input characters and λ specifies the empty character.

Table 3 shows the results of the OCR step for the two different enhancement methods presented in section 2.5, robust bi-linear interpolation and robust bi-cubic interpolation. As can be seen, the figures for the two methods are comparable. Thus, although the robust bi-cubic interpolated images are of a better quality for a human viewer, the OCR system does not transform this better quality into better recognition results.

Input	Enh. method	Recall	Precision	Cost
#1	Bilinear robust	81.5%	88.3%	361.1
	Bicubic robust	80.5%	86.5%	355.1
#2	Bilinear robust	94.8%	91.5%	116.9
	Bicubic robust	96.7%	92.8%	95.5
#3-fr	Bilinear robust	88.1%	92.9%	102.9
	Bicubic robust	85.4%	92.0%	114.4
#3-ge	Bilinear robust	98.9%	97.4%	11.2
	Bicubic robust	97.5%	95.4%	14.8
#4	Bilinear robust	76.2%	89.3%	252.7
	Bicubic robust	82.7%	87.1%	239.0
Total	Bilinear robust	85.4%	90.7%	844.8
	Bicubic robust	86.4%	89.6%	818.7

Table 3: OCR Results for different enhancement methods.

Table 4 presents the results of the OCR step for different binarization methods:

- Otsu's method.
- Niblack's method with parameter $k = -0.2$.
- Sauvola et al.'s method with parameters $k = 0.5$ and $R = 128$.
- Our method with parameter $a = 0.5$.

The figures confirm the known disadvantages of global techniques, as Otsu's method. Although the precision is very high at 90%, only 47% of the rectangles are correctly recognized. Niblack obtains a rate of 80% of corrected characters. Sauvola's algorithm, which has been developed to overcome the problems of Niblack, obtains worse results. This comes from the quality of the video frames, where the contrast is not always as high as in document images, and the hypothesis assumed by Sauvola et al. does not hold. Our normalization contains the advantages of Sauvola's method without the sensitivity to contrast

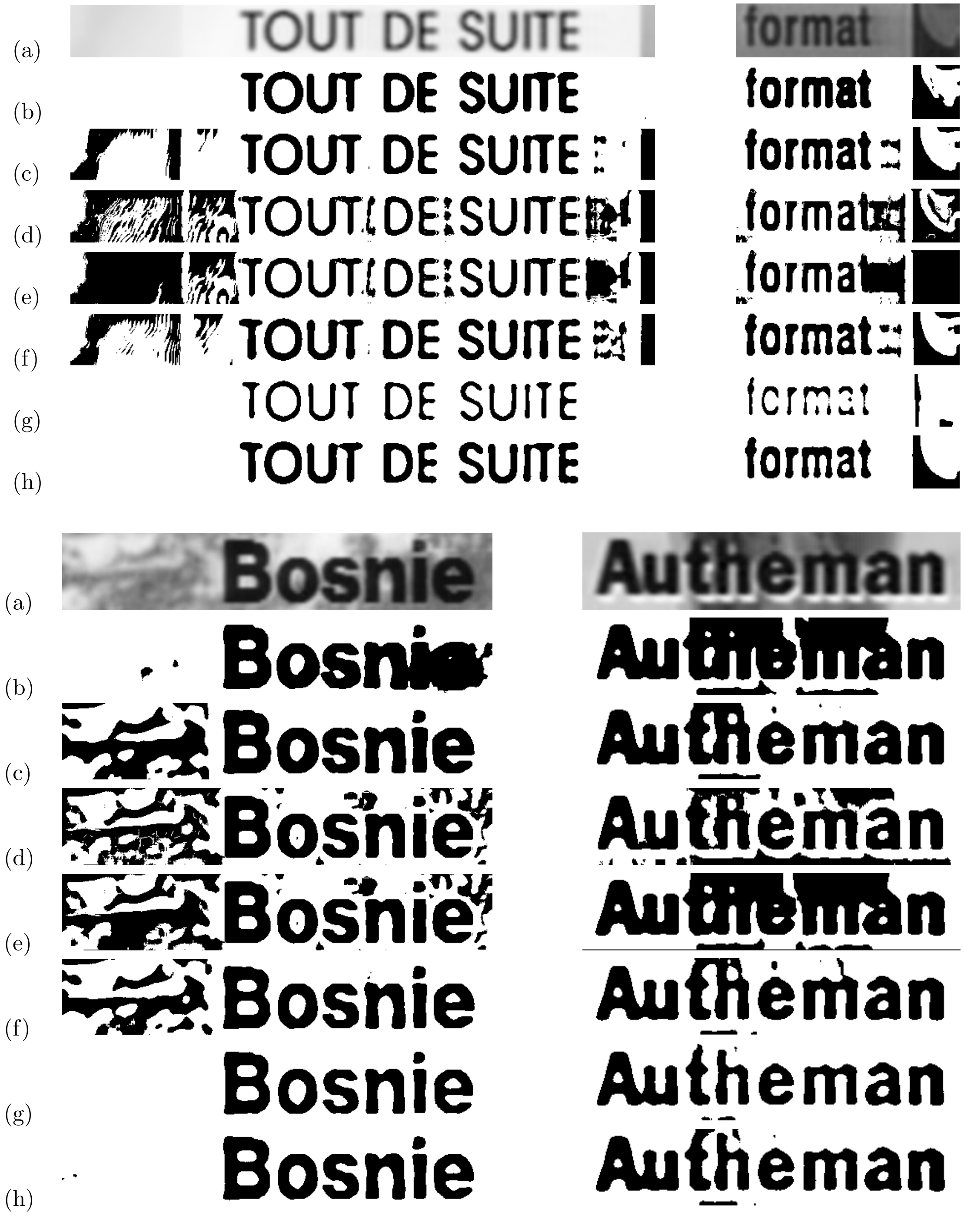


Figure 13: The results of the different binarisation techniques for 4 different example images: The original image (a) Otsu's method (b) a local windowed version of Otsu's method (c) Yanowitz and Bruckstein's method (d) Yanowitz and Bruckstein's method including their post processing step (e) Niblack's method (f) Sauvola et al.'s method (g) Our contrast based method (h).

and gray value range. The results obtained by our method are better than Niblack's and Sauvola's.

Input	Bin. method	Recall	Precision	Cost
#1	Otsu	39.6%	87.5%	791.7
	Niblack	79.0%	78.3%	528.4
	Sauvola	66.5%	75.8%	625.8
	Our method	**81.5%**	**88.3%**	**361.1**
#2	Otsu	54.8%	**93.2%**	371.6
	Niblack	92.4%	79.6%	257.2
	Sauvola	82.8%	88.5%	203.8
	Our method	**94.8%**	91.5%	**116.9**
#3-fr	Otsu	51.1%	**96.1%**	300.1
	Niblack	85.1%	93.4%	129.8
	Sauvola	61.5%	68.2%	367.2
	Our method	**88.1%**	92.9%	**102.9**
#3-ge	Otsu	57.5%	**98.2%**	121.8
	Niblack	**99.0%**	93.4%	25.4
	Sauvola	80.3%	98.1%	73.7
	Our method	98.9%	97.4%	**11.2**
#4	Otsu	45.7%	84.8%	500.9
	Niblack	62.8%	70.5%	527.9
	Sauvola	76.0%	85.5%	281.4
	Our method	**76.2%**	**89.3%**	**252.7**
Total	Otsu	47.3%	90.5%	2086.1
	Niblack	80.5%	80.4%	1468.7
	Sauvola	72.4%	81.2%	1551.9
	Our method	**85.4%**	**90.7%**	**844.8**

Table 4: OCR Results for different binarization methods.

3.5 The parameters of the system

A joint project between our team and INA^5 is currently under way, whose goal is to investigate the role of text in video sequences from a documentalist's point of view and to develop the technology to steer detection techniques with a priori knowledge on the contents of the video [10]. The documentalists at INA manually create semantic descriptions for all archived videos. This time consuming work can be assisted by methods which automatically extract contents, as e.g. text detection. On the other hand, the operator's knowledge on the contents of the material can also be used to enhance the detection quality.

Most broadcasts, as e.g. television journals, newscasts etc., follow strict rules on the presentation of the material, which are enforced by the broadcast station. These rules condense to restrictions on fonts, color, style, position etc. of the placed text according to the type of message transported by it. For instance during news casts, names of locations are always displayed at the same position and in the same style, as well as names of interviewed people.

The parameters of our detection system are related to the properties of the text and can therefore be tuned by an operator to the style of the video to be treated in order to further decrease the number of false alarms. As an example we could imagine special parameters for movies, whose subtitles have a fixed size and position. The constraints have to be more relaxed for TV commercials, which contain text in many styles.

Table 1 shows the parameters of our system together with the values we determined by the evaluation process described in sub section 3.1. They can be grouped into four different types:

Geometric properties Although these parameters (S, t_1-t_8, t_{11}) relate to the shape and size of the characters, most of them are quite stable to small changes, as can be seen for the example of the parameters S, t_4 and t_5 in figure 11.

Temporal stability The parameters t_9 and t_{10} relate to the temporal behavior of text, which largely depends on the type of broadcast. E.g. text is rather stable in news casts and television journals, whereas commercials tend to have text which stays on the screen for very small time spans only.

General properties of the script The parameter a of the binarization routine (character segmentation) largely depends on very general properties of the script and the font and is very stable.

Properties of the frame The parameter α controls the binarization of the text filter response. It has been added to make Otsu's method more robust and depends on the amount of texture in the frame. Deviations of its optimal value produce artifacts which are largely eliminated by the following clean-up phases.

As stated before, for the moment we set the values of these parameters by optimizing them for our test database, but a further reduction of the amount of false alarms will be possible by controlling these parameters according to the a priori knowledge we have on the type of the text in the processed media.

4 Discussion

In this paper we presented an algorithm to detect and process artificial text in images and videos. A very high detection rate is obtained with a simple algorithm where only 6.5% of the text boxes in the test data are missed. Our method contains an integral approach beginning with the localization of the text to the multiple frame enhancement and the binarization of the text boxes before they are passed to a commercial OCR. The main contributions lie in the robustness of the algorithm and the exploitation of the additional constraints of artificial text used for indexing.

The presented detection technique is very robust due to it's small number of parameters. The size S of the accumulation window theoretically depends on the font size of the text. However, in practice the detection algorithm works very well with a fixed parameter for the high range of text sizes found in our test database.

The binarization technique we presented allows, because of the introduction of global statistical parameters, a better adaptation to the variations in video contents. This new method is a priori not in its final form in the sense that new parameters could be included. As an example, we described the case of low local variations ($s << R$), see equation (29). It could be interesting to link this property not only to the maximum value of s, but also to the minimum value. This other extreme value allows to quantify the additive noise of the image and therefore to access the signal/noise ratio. Various methods for the estimation of this minimal variance exist [16]. They are often computational complex and their adaptation to the special properties of video needs to be studied beforehand.

Apart from the sole objective to recognize the text, we would also like to study the usability of these indicators to efficiently filter false alarms, i.e. detected rectangles which do not contain any text, which are still too numerous in our system (in consequence to our efforts not to miss text), and which cause additional processing costs during the OCR phase of the system. Our binarization technique has been formulated in terms of contrast and can therefore be linked to the work in this area, which is numerous in the image processing domain. In particular, it will be necessary to study the relation to Kholer's work [8].

Due to the efforts made in order not to miss much text, the number of false alarms is currently still very high. We count on the fact, that the false alarms will not hurt the performance of the applications, because the OCR will not recognize them as useful text. Moreover, in the framework of the TREC 2002 competition we developed a classifier which classifies the text boxes as text or false alarms according to the OCR output [30]. Furthermore, we are currently working on an image based rejection strategy in order to further decrease the number of false alarms before the OCR phase. This rejection strategy uses a support vector machine based learning algorithm to classify the text boxes after their detection with the information from edge and corner features.

Our future research will be concentrated on text with fewer constraints, i.e. scene text, text with general orientations and moving text. Scene text, which has not been designed intentionally to be read easily, demands a different kind of treatment. Processing of the color signal may be necessary due to reflections or fonts rendered with color contrast against a background with similar luminance values. The perceptual grouping needs to be adapted to fonts with arbitrary orientations, and finally perspective distortions need to be taken into account.

References

[1] L. Agnihotri and N. Dimitrova. Text detection for video analysis. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 109–113, June 1999.

[2] P. Clark and M. Mirmehdi. Combining Statistical Measures to Find Image Text Regions. In IEEE Computer Society, editor, *Proc. of the ICPR*, pages 450–453, 3 September 2000.

[3] D. Crandall and R. Kasturi. Robust Detection of Stylized Text Events in Digital Video. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 865–869, 2001.

[4] G. Giocca, I. Gagliardi, and R. Schettini. Quicklook: An Integrated Multimedia System. *Journal of Visual Languages and Computing*, (12):81–103, 2001.

[5] L. Gu. Text Detection And Extraction in MPEG Video Sequences. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, pages 233–240, 19 September 2001.

[6] A.K. Jain and B. Yu. Automatic Text Location in Images and Video Frames. *Pattern Recognition*, 31(12):2055–2076, 1998.

[7] K. Jung. Neural network-based text location in color images. *Pattern Recognition Letters*, 22:1503–1515, 2001.

[8] R. Kholer. A segmentation system based on thresholding. *Computer, Graphics and Image Processing*, 15:241–245, 1981.

[9] K.I. Kim, K. Jung, S.H. Park, and H.J. Kim. Support vector machine-based text detection in digital video. *Pattern Recognition*, 34(2):527–529, 2001.

[10] R. Landais, J.-M. Jolion, and L. Vinet. L'apport du texte présent dans les vidéos aux pratiques documentaires à

l'institut national de l'audiovisuel (in french). In *Proceedings of ORASIS 2003: Journées Francophones des Jeunes Chercheurs en Vision par Ordinateur (Submitted)*.

[11] F. LeBourgeois. Robust Multifont OCR System from Gray Level Images. In *Proceedings of the 4th Int. Conference on Document Analysis and Recognition*, pages 1–5, August 1997.

[12] M.S. Lew, editor. *Principles of Visual Information Retrieval*, chapter 9. Springer Verlag, 2001.

[13] H. Li and D. Doermann. Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing*, 9(1):147–156, 2000.

[14] R. Lienhart. Automatic Text Recognition for Video Indexing. In *Proceedings of the ACM Multimedia 96, Boston*, pages 11–20, November 1996.

[15] R. Lienhart and A. Wernike. Localizing and Segmenting Text in Images, Videos and Web Pages. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(4):256–268, 2002.

[16] P. Meer, J.M. Jolion, and A. Rosenfeld. A Fast Parallel Algorithm for Blind Estimation of Noise Variance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):216–223, 1990.

[17] M.R.Naphade and T.S.Huang. Semantic Video Indexing using a probabilistic framework. In *Proceedings of the ICPR 2000*, pages 83–88, 3 September 2000.

[18] G. Myers, R. Bolles, Q.-T. Luong, and J. Herson. Recognition of Text in 3D Scenes. In *Fourth Symposium on Document Image Understanding Technology, Maryland*, April 2001.

[19] W. Niblack. *An Introduction to Digital Image Processing*, pages 115–116. Englewood Cliffs, N.J.: Prentice Hall, 1986.

[20] N.Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[21] O.D.Trier and A.K. Jain. Goal-Directed Evaluation of Binarization Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1191–1201, 1995.

[22] P.K. Sahoo, S. Soltani, and A.K.C. Wong. A Survey of Thresholding Techniques. *Computer Vision, Graphics and Image Processing*, 41(2):233–260, 1988.

[23] T. Sato, Takeo Kanade, E.K. Hughes, M.A. Smith, and S. Satoh. Video OCR: Indexing digtal news libraries by recognition of superimposed captions. *ACM Multimedia Systems: Special Issue on Video Libraries*, 7(5):385–395, 1999.

[24] J. Sauvola, T. Seppänen, S. Haapakoski, and M. Pietikäinen. Adaptive Document Binarization. In *International Conference on Document Analysis and Recognition*, volume 1, pages 147–152, 1997.

[25] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[26] K. Sobottka, H. Bunke, and H. Kronenberg. Identification of text on colored book and journal covers. In *Proceedings of the 5. Int. Conference on Document Analysis and Recognition*, pages 57–62, September 1999.

[27] X. Tang, X. Gao, J. Liu, and H. Zhang. A Spatial-Temporal Approach for Video Caption Detection and Recognition. *IEEE Transactions on Neural Networks*, 13(4):961–971, 2002.

[28] R.A. Wagner and M.J.Fisher. The string to string correction problem. *Journal of Assoc. Comp. Mach.*, 21(1):168–173, 1974.

[29] A. Wernike and R. Lienhart. On the segmentation of text in videos. In *Proc. of the IEEE Int. Conference on Multimedia and Expo (ICME) 2000*, pages 1511–1514, July 2000.

[30] C. Wolf, D. Doermann, and M. Rautiainen. Video indexing and retrieval at UMD. In National Institute for Standards and Technology, editor, *Proceedings of the text retrieval conference - TREC (To appear)*, November 2002.

[31] V. Wu, R. Manmatha, and E.M. Riseman. Textfinder: An automatic system to detect and recognize text in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1224–1229, 1999.

[32] S.D. Yanowitz and A.M. Bruckstein. A new method for image segmentation. *Computer Vision, Graphics and Image Processing*, 46(1):82–95, 1989.

[33] Y. Zhong, H. Zhang, and A.K. Jain. Automatic Caption Localization in Compressed Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):385–392, 2000.

[34] J. Zhou and D. Lopresti. Extracting text from www images. In *Proceedings of the 4. Int. Conference on Document Analysis and Recognition*, pages 248–252, August 1997.

Christian WOLF received his Master's degree (the Austrian Dipl. Ing.) in computer science in 2000 at Vienna University of Technology, and is currently working on his PhD in the Lyon Research Center for Images and Intelligent Information Systems, France. His research interests include image and video indexing and contents extraction from multimedia documents. (more details at http://rfv.insa-lyon.fr/~wolf)

Jean-Michel JOLION received the *Diplôme d'ingénieur* in 1984, and the PhD in 1987, both in Computer Science from the National Institute for Applied Science (INSA) of Lyon (France). From 1987 to 1988, he was a staff member of the Computer Vision Laboratory, University of Maryland (USA). From 1988 to 1994 he was appointed as an Associate Professor at the University Lyon 1 (France). From 1994, he has been with INSA and the Lyon Research Center for Images and Intelligent Information Systems where he is currently Professor of computer science. His research interests belong to the computer vision and pattern recognition fields. He has studied robust statistics applied to clustering, multiresolution and pyramid techniques, graphs based representations ... mainly applied to the image retrieval domain. Professor Jolion is a member of IAPR and IEEE (M'90). (more details at http://rfv.insa-lyon.fr/~jolion)