Image and Vision Computing 28 (2010) 1413–1426

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

A two-stage scheme for text detection in video images

Marios Anthimopoulos *, Basilis Gatos, Ioannis Pratikakis

Computational Intelligence Laboratory, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", 153 10 Athens, Greece

ARTICLE INFO

Article history:
Received 16 July 2009
Received in revised form 17 February 2010
Accepted 3 March 2010

Keywords:
Text detection
Video OCR
Content-based indexing
SVM

ABSTRACT

This paper proposes a two-stage system for text detection in video images. In the first stage, text lines are detected based on the edge map of the image leading in a high recall rate with low computational time expenses. In the second stage, the result is refined using a sliding window and an SVM classifier trained on features obtained by a new Local Binary Pattern-based operator (eLBP) that describes the local edge distribution. The whole algorithm is used in a multiresolution fashion enabling detection of characters for a broad size range. Experimental results, based on a new evaluation methodology, show the promising overall performance of the system on a challenging corpus, and prove the superior discriminating ability of the proposed feature set against the best features reported in the literature.

1. Introduction

The tremendous increase of multimedia content has raised the need for automatic semantic information indexing and retrieval systems. Many methods have been proposed for the semantics extraction of various granularity levels from audiovisual content. Textual information in videos proves to be an important source of high-level semantics. There exist mainly two kinds of text occurrences in videos, namely artificial and scene text. Artificial text, as the name implies, is artificially added in order to describe the content of the video or give additional information related to it. This makes it highly useful for building keyword indexes. Scene text is textual content that was captured by a camera as part of a scene such as text on T-shirts or road signs and usually brings less related to video information. In Fig. 1, yellow boxes denote artificial text while red boxes delimit the scene text. Text can also be classified into normal or inverse. Normal is denoted any text whose characters have lower intensity values than the background while inverse text is the opposite [1]. In Fig. 2, "EURO" is inverse while "SPORT" is normal text. The procedure of textual information extraction from videos is usually split into four distinct steps: (i) detection, (ii) tracking, (iii) segmentation and (iv) recognition. Among all steps, detection step is the most crucial and although it has been extensively studied in the past decade presenting quite promising results, there are still challenges to meet. Further down we will discuss the different approaches used towards the text detection problem, the corresponding drawbacks and the remaining challenges. In this discussion we will focus on the methods designed for detecting artificial text in video images. For scene text detection in camera-based images the reader should refer to the survey papers [2–4].

Most of proposed text detection methods use as representative text features, color, edge and texture information. To exploit this information, i.e. describe text and discriminate it from the background, some researchers apply heuristic rules derived by empirical constraints while others use machine-learning methods trained on real data. Recently, some hybrid approaches have been proposed.

Many existing heuristic methods, derived from document analysis research area, are based on color or intensity homogeneity of characters. They detect character regions in the image and then group them into words and text lines based on geometrical constraints. These methods, also known as connected component (CC) methods, can perform satisfactorily only on high quality images with simple background and known text color, assumptions that usually do not apply in the case of video images. Moreover, text in video images often suffers from color bleeding due to video compression. Typical CC approaches can be found in [5,6].

Some other heuristic methods detect text based on edge information, i.e. strength, density or distribution. Sato et al. [7] apply a 3×3 horizontal differential filter to the entire image with appropriate binary thresholding followed by size, fill factor and horizontal–vertical aspect ratio constraints. Xi et al. [8] propose an edge-based method based on an edge map created by Sobel operator followed by smoothing filters, morphological operations and geometrical constraints. Cai et al. [9] and Lyu et al. [10] suggest the use of local thresholding on a sobel-based edge strength map. Anthimopoulos et al. [11] use Canny edge map followed by morphological operations and projection analysis. Kim and Kim [12] instead of using an explicit

* Corresponding author. Tel.: +30 210 650 3218; fax: +30 210 653 2175.
E-mail addresses: anthimop@iit.demokritos.gr (M. Anthimopoulos), bgat@iit.demokritos.gr (B. Gatos), ipratika@iit.demokritos.gr (I. Pratikakis).

0262-8856/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2010.03.004

Fig. 1. Example of artificial and scene text. Yellow boxes bound artificial text while red indicate scene text.

edge map as an indicator of overlay text, they suggest the use of a transition map generated by the change of intensity and a modified saturation. A heuristic rule based on the different Local Binary Patterns (LBP) is used for verification. These heuristic techniques proved to be very efficient and satisfactory robust for specific applications with high contrast characters and relatively smooth background. However, the fact that many parameters have to be estimated experimentally condemns them to data dependency and lack of generality.

DCT coefficients of intensity images have been widely used as texture features and have also been used for heuristic text detection methods [13–16]. The DCT coefficients globally map the periodicity of an image and can be a quite efficient solution for jpeg and mpeg encoded images and videos. In this case, the pre-computed coefficients of 8×8 pixel block units are used. However, this block size is not a large enough area to sufficiently depict the periodical features of a text line and the computation of DCT for larger windows even by the fast DCT transform proves quite costly. In addition, these methods still use empirical thresholds on specific DCT-based features and therefore they lack adaptability.

Several machine learning-based approaches have been proposed for the detection of text areas with great success. These approaches are based on sliding windows that scan the image and machine learning techniques which classify each window as text or non-text. Machine learning classifiers have proved to be an appealing solution for many problems that cannot be defined in a strict mathematical manner. Jung [17] and Kim et al. [18] directly use the color and gray values of the pixels as input for a neural network and an SVM, respectively. Wolf and Jolion [19] use an SVM trained on differential and geometrical features. Lienhart and Wernicke in [20] used as features the complex values of the gradient of the RGB input image fed to a complex-valued neural network. Li et al. [21] suggest the use of the mean, second order (variance) and third-order central moments of the LH, HL, and HH component of the first three levels of each window to train a three-layer neural network. Zhang et al. [22] proposed a system for object detection based on Local Binary Patterns (LBP) and Cascade histogram matching. They applied the proposed method to video text and car detection. The main shortcoming of the methods attributed to this category is the high computational complexity since a sliding window is required to scan the entire image with a typical step of 3 or 4 pixels, demanding thousands of calls to the classifier per image.

Fig. 2. Example of inverse and normal text.

Recently, some hybrid methods have also been proposed, that combine the efficiency of heuristic methods with machine-learning accuracy and generalization. These methods usually consist of two stages. The first localizes text with a fast heuristic technique while the second verifies the previous results eliminating some detected area as false alarms using machine learning. In [23], Chen et al. use a localization/verification scheme which claim to be highly efficient and effective. For the verification part, Constant Gradient Variance (CGV) features are fed to an SVM classifier. Ye et al. [24] propose a coarse-to-fine algorithm based on wavelets. The first stage applies thresholding on the wavelet energy in order to coarsely detect text, while the second identifies the coarse results using an SVM and a more sophisticated wavelet-based feature set. Jung et al. [25] apply as a first stage, a stroke filtering and they also verify the result using an SVM with normalized gray intensity and CGV features. Then, a text line refinement module follows, consisting of text boundary shrinking, combination and extension functions. However, the machine learning verification task used by these methods can only take a binary decision, i.e. if an initial result is text or not without having the capability to refine it. For example, if the resulting bounding box of the first stage contains text as well as background pixels, in the second stage it will be either entirely verified as text, or discarded as false alarm.

Concluding the discussion of the previous works, we can say that although the specific research area has shown a great progress, there are still challenges to be considered. Machine-learning methods have shown important capabilities for generalization but proved to be computationally expensive. Hybrid methods benefit regarding efficiency but they are actually based on the initial results of heuristic methods which fail to deal with complex background. Another great challenge is the choice of the feature set used by machine learning techniques to discriminate text from non-text areas. The features used until now, usually inherited from texture segmentation research area, are not capable to adapt to the specific problem and fail to discriminate varying-contrast text from high-contrast background. Finally, another essential factor for the optimization of every text detection algorithm is the choice of an objective evaluation methodology.

In this work we propose a two-stage approach with a novel machine learning refinement which cannot only verify the initial result but refine its boundaries as well. In that way the whole system's performance is based on this refinement, instead of relying to the initial heuristic results. This machine learning stage uses a new, highly discriminative feature set, derived from a proposed operator that captures the edge structure for different levels of contrast, and has shown superior performance against other features reported in the literature. Finally, a novel evaluation methodology is introduced which is based on the estimated number of

characters and proved to be more representative in terms of textual information extraction, contrary to the existing pixel-based or block-based approaches.

The remainder of our paper is structured as follows: In Section 2 the proposed two-stage algorithm is described, Section 3 presents the experimental results and the corresponding discussion, and finally, in Section 4 conclusions are drawn.

2. Proposed methodology

The proposed text detection methodology constitutes a two-stage, coarse-to-fine approach. In the first stage, namely heuristic coarse detection stage, we adopted a low threshold for the generation of the edge map which provides us with a high recall rate for videos of different contrast values. The second stage, namely machine

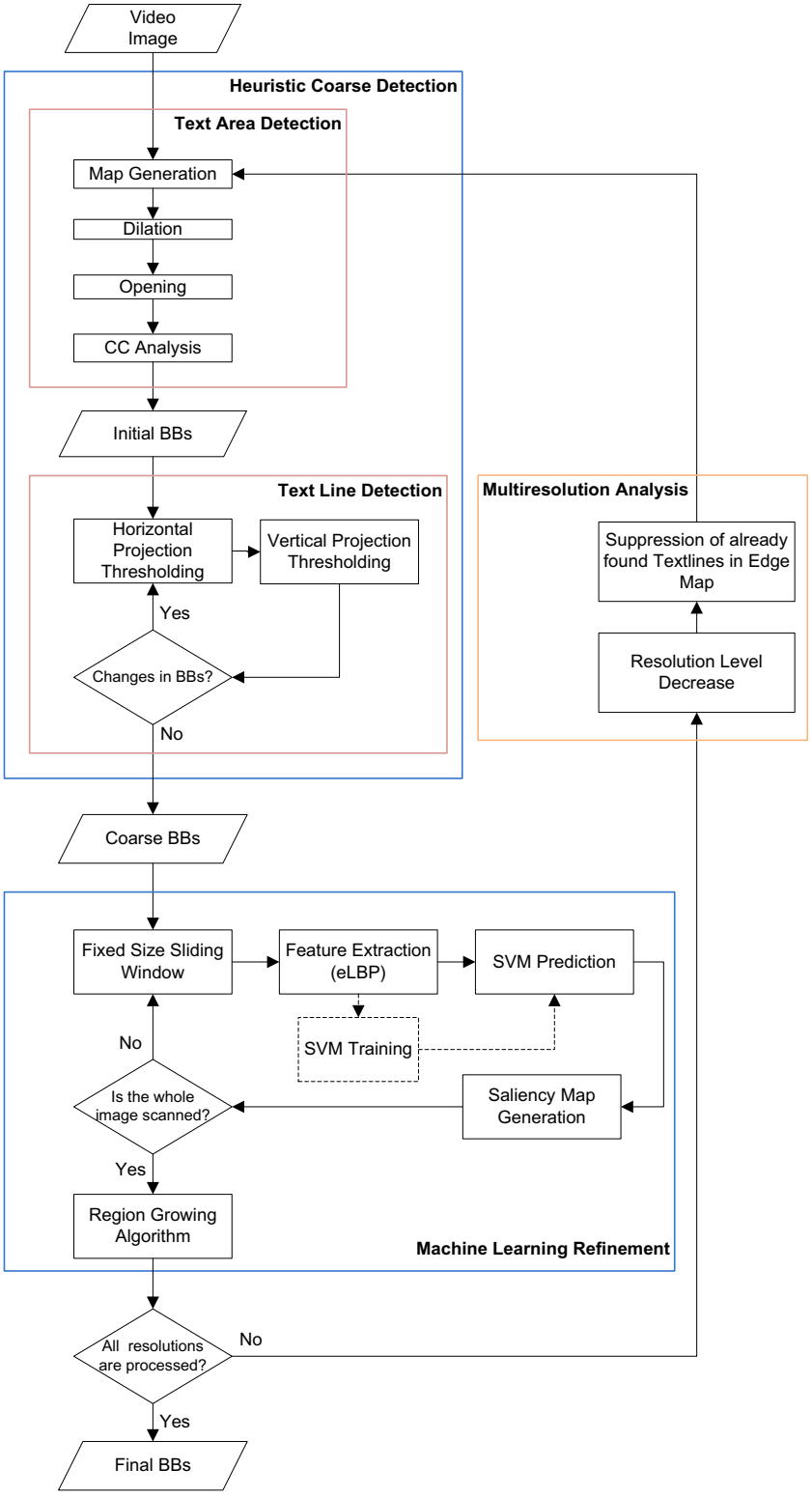

Fig. 3. Flowchart of the proposed algorithm.

learning refinement stage is used to refine the result in every Bounding Box (BB) of the initial result leading to an effective minimization of false alarms (Fig. 3). This stage uses an edge histogram-based feature set, derived from a proposed operator and will be described in detail in Section 2.2.1. Finally the multiresolutional manner of the method aids in detecting characters of various sizes in different resolutions, although the basic method is trained in a fixed scale.

Since the basic two-stage algorithm is trained on a fixed scale, before the further description of the algorithm we have to make some assumptions about the size range of the text to be recognized. We assume that text height varies between MinH and MaxH pixels and every text line consists of at least three characters leading to a minimum width of approximately MinW = 3 ∗ MinH pixels. The choice of the size range is crucial for the system's performance. Generally, smaller range leads to better detection on a fixed scale level since characters of different sizes present different texture characteristics. However, a narrow range would also lead to many levels for the multiresolution approach and thus more difficult combination of the different scales' results. Eventually, the choice of MinH and MaxH values has to satisfy the tradeoff between the performance of the fixed-scale detector and the multiresolution analysis. Further down we will demonstrate the results of the different stages of the algorithm on the video images shown in Fig. 1. Fig. 1a shows an image with a relatively low edge density background while Fig. 1b shows a scene with very strong background edges, thus, it constitutes a great challenge for every text detection system.

2.1. Heuristic coarse text detection

For the first, coarse stage of text detection, we use an approach based on a previous work by Anthimopoulos et al. [11]. This approach exploits the fact that text lines produce strong vertical edges horizontally aligned with a high density. The use of edge information for text detection is justified by the fact that every kind of text presents strong edges, in order to be readable. Moreover, using edges as the prominent feature of our system gives us the opportunity to detect normal or inverse characters of any color.

2.1.1. Text area detection

As a first step of the heuristic coarse detection stage, we produce the edge map of the video image. Several methodologies are used in the literature for computing the edge map of an image [26]. For our algorithm we use Canny edge detector. Canny [27] uses Sobel masks in order to find the edge magnitude of the image intensity and then uses non-maxima suppression and hysteresis thresholding. Ideally, the created edge map is a binarized image with the contour pixels set to one (white) and the remainder pixels equal to zero (black) (Figs. 4a and 5a). After computing the Canny edge map, a dilation is performed to link the character edges of every text line (Figs. 4b and 5b). The structuring element in the dilation is horizontal and its size depends on the estimated maximum distance between the characters. A $3 \times (\text{MaxH}/2)$ cross-shaped element found to be satisfactory for characters up to MaxH pixels height. Then a morphological opening is used, removing the noise and smoothing the candidate text areas (Figs. 4c and 5c). The element used here is also cross-shaped with size equal to MinH × MinW which is actually the size of the smallest text line to be detected. Every component created by the previous dilation with height less than MinH or width less than MinW is suppressed.

Unfortunately, this operation may suppress the edges of text lines with height less than MinH pixels. However, this is not so devastating since very small characters are either way not recognized in the stage of recognition. At this phase, every connected component represents a candidate text area for which we have to compute the bounding box (Figs. 4d and 5d).

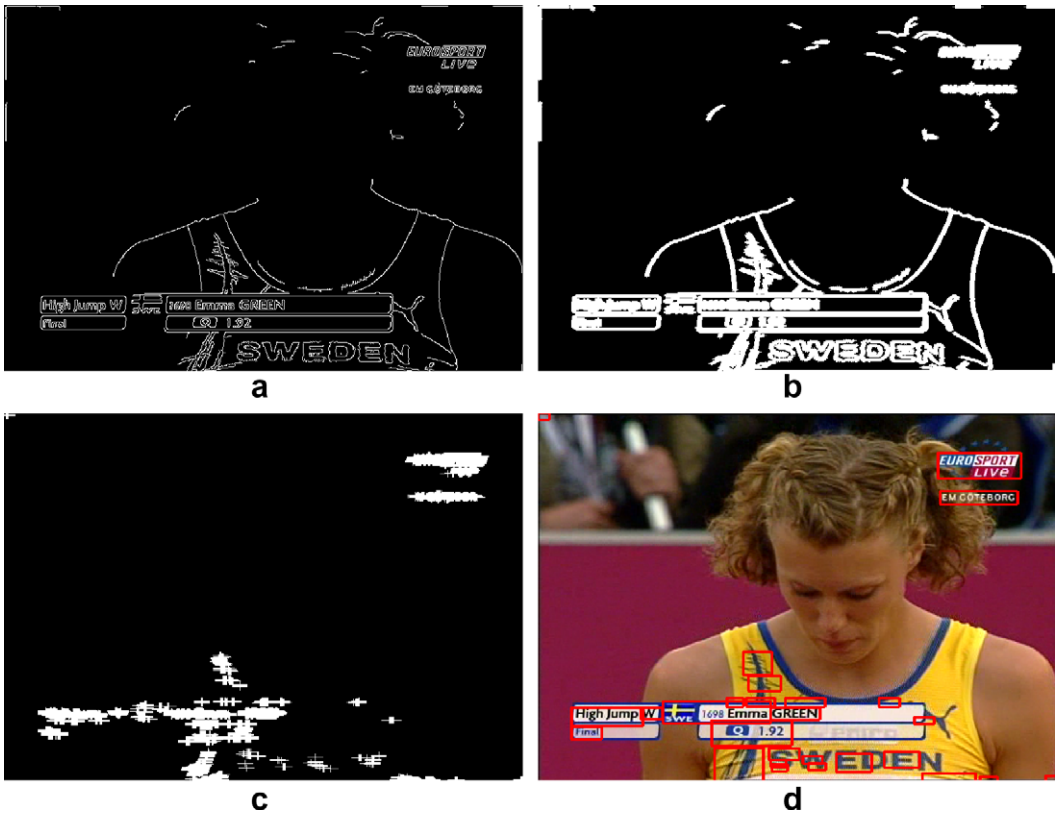

Fig. 4. Example of text area detection in a low edge density image. (a) Edge map, (b) dilated edge map, (c) opening on the edge map, (d) initial bounding boxes after CC analysis.

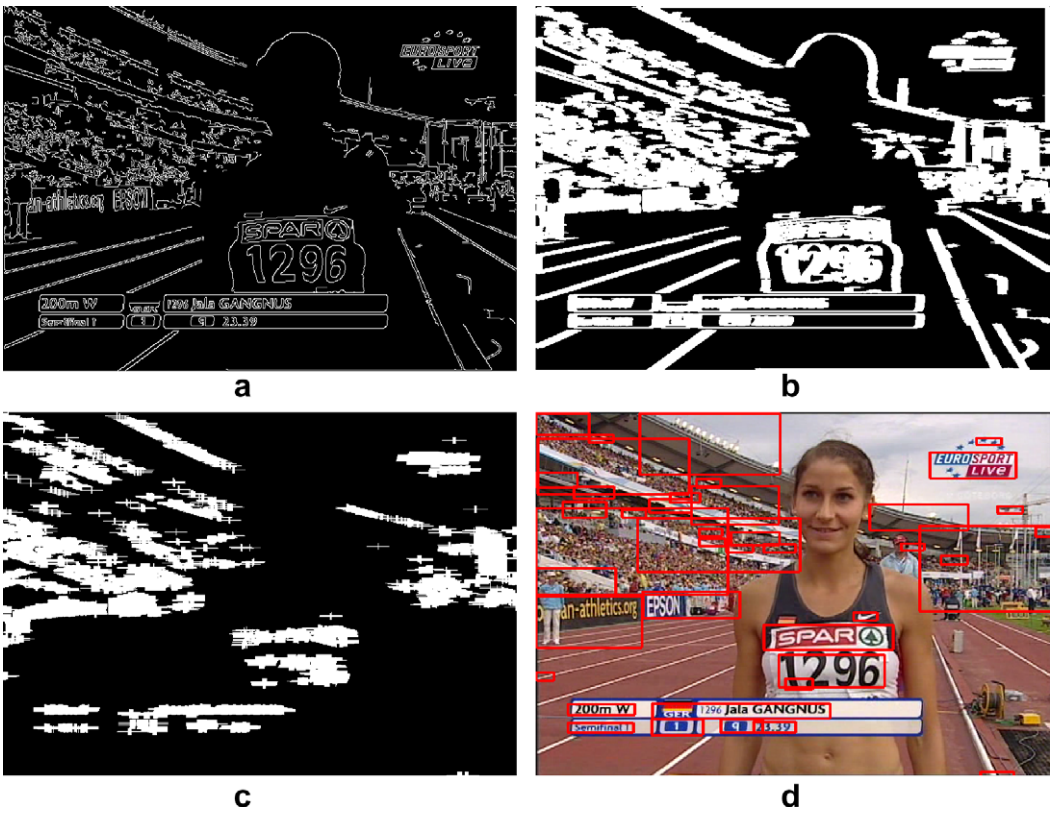

Fig. 5. Example of text area detection in a high edge density image. (a) Edge map, (b) dilated edge map, (c) opening on the edge map, (d) initial bounding boxes after CC analysis.

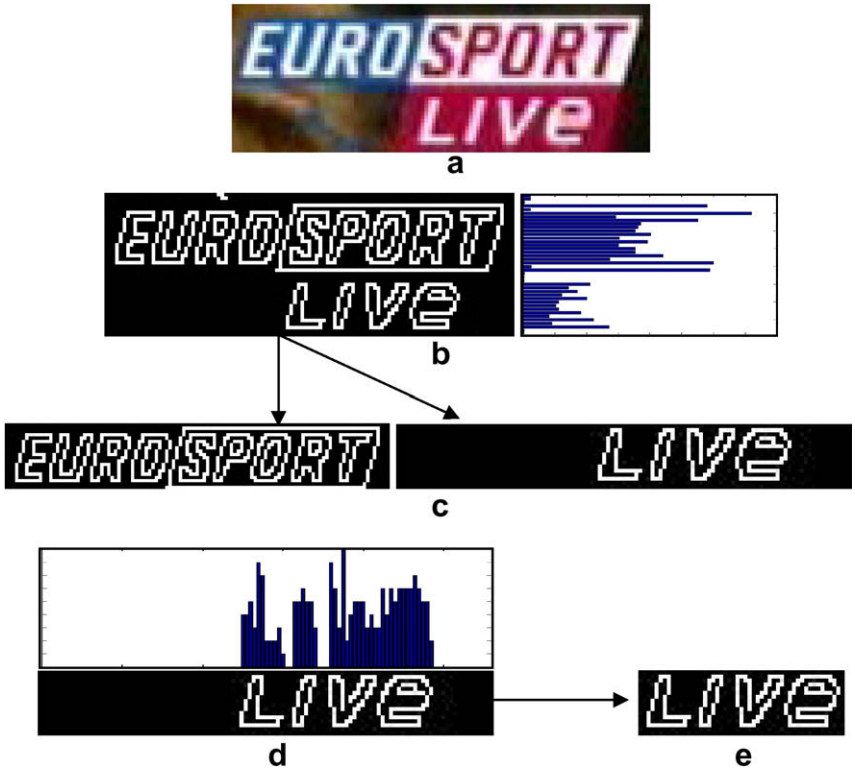

Fig. 6. (a) Initial detection result (b) horizontal projection, (c) results of horizontal projection, (d) vertical projection, (e) result of vertical projection.

2.1.2. Text line detection

The previous stage has a high detection rate but relatively low precision due to many false positives. This means that most of the text lines are included in the text area boxes while at the same time some bounding boxes may include more than one text line as well as noise. The noise usually originates from objects with high intensity edges that connect to the text lines during the dilation process. The low precision also originates from detected bounding boxes which do not contain text but objects with high vertical edge density. To improve precision by rejecting the false alarms we use a method based on horizontal and vertical projections.

Firstly, the horizontal edge projection of every box is computed and lines with projection values below a threshold are discarded. In this way, boxes with more than one text line are split and some lines with noise are also discarded (Fig. 6b). Nevertheless, boxes which do not contain text are usually split in a number of boxes with very small height and discarded by the next stage due to size constraints.

A box is discarded if:

- Height is lower than MinH.
- Width is lower than a MinW.
- Ratio width/ height is lower than a threshold (set to 1.5).

Then, a similar procedure with vertical projection follows (Fig. 6d). This method would actually break every text line in words or even in characters. However, this is not an intention of the algorithm so finally the vertically divided parts are reconnected if the distance between them is less than a threshold which depends on the height of the candidate text line (set to 1.5 $*$ height). In this way, a bounding box will split only if the distance between two words is larger than the threshold which means that actually belong to different text lines or if a part of the candidate text line contain only noise. The whole procedure with horizontal and vertical projections is repeated until no changes occur. Examples of projection analysis results are presented in Figs. 7 and 8.

The final results of the coarse stage for the two examples lead us to some observations. The video image with the smoothest background presents a quite satisfactory result (Fig. 7) while the high edge density image produces many false alarms (Fig. 8).

2.2. Machine learning refinement

Edge-based heuristic methods detect text based mainly on the edge density. However, in many cases, non-text regions present

Fig. 7. Bounding boxes after projection analysis of the low edge density image.

Fig. 8. Bounding boxes after projection analysis of the high edge density image.

edge density values adequate to produce false alarms that human optical perception system would have avoided. This fact motivated the research of larger feature sets which capture not only the abrupt intensity changes of the image, but their spatial distribution, as well. The large number of features and the great generalization capability of Support Vector Machines (SVM's) [28] led us to use an SVM and a sliding window model followed by a region growing algorithm to refine the result. This approach was initially proposed in our previous work [29]. However the features used in [29] proved to be much weaker than the new proposed feature set, which will be described in the next section.

2.2.1. Feature extraction

The majority of the features used for text detection originate from texture segmentation research area since a text area as a periodic repetition of similar objects with specific alignment presents some of the fundamental characteristics of texture. Local Binary Pattern (LBP) has proven to be highly discriminative for texture segmentation and its advantages, namely, its invariance to monotonic gray-level changes and computational efficiency, make it suitable for demanding image analysis tasks. This fact motivated us to use the concept of LBP for text detection and adjust it to the specific problem.

LBP was originally introduced by Ojala et al. [30] as a non parametric operator measuring the local contrast for efficient texture classification. The LBP operator consists of a 3×3 kernel where the center pixel is used as a threshold. Then the eight binarized neighbours are multiplied by the respective binomial weight producing an integer in the range $[0 \ldots 255]$ (Fig. 9). Each of the 256 different 8-bit words is considered to represent a unique texture pattern.

Formally, the decimal form of the resulting 8-bit word (LBP code) can be expressed as follows:

$$\mathrm{LBP}(x_c,\ y_c) = \sum_{n=0}^{7} S(i_n - i_c)2^n \tag{1}$$

where i_c corresponds to the grey value of the center pixel (x_c, y_c), i_n $(n \in [0,7])$ to the grey values of the eight surrounding pixels, and function $S(x)$ is defined as:

$$S(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

When local binary pattern is applied in a greyscale image, another 8-bit greyscale image is created in which each pixel value represents the texture pattern of the respective pixel in the original im-

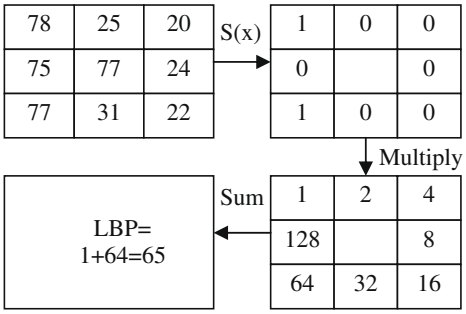

Fig. 9. Example of LBP computation.

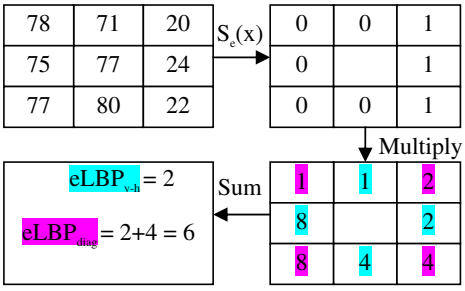

Fig. 11. Example of $eLBP_{v-h}$ and $eLBP_{diag}$ computation.

age. Thus, the 256 histogram values of an image region depict its texture structure.

Although the original LBP operator has shown satisfactory performance for many kinds of texture classification it faces two important problems in capturing the characteristics of textual texture. The first is that in text detection normal and inverse text is considered as one class although LBP produce quite different histograms for the two cases. The second problem is related to the fact that LBP cannot capture the pattern of equal neighbours since it treats them with the same manner with higher valued neighbours. If we also consider the noise, we come to the conclusion that an equal neighbour could arbitrary produce 0 or 1 to the binary pattern. To solve these problems we propose the edge Local Binary Pattern (eLBP) that is a modified LBP operator which actually describes the local edge patterns appeared in an image.

In eLBP, a neighbouring pixel is represented by 0 if it is close to the center pixel or 1 if not. In that way, we solve the first problem mentioned above by capturing only the fact that there is an edge between the center pixel and a neighbouring pixel. Since the operator does not consider if it is a positive or negative edge, it recognises normal and inverse text as the same texture. In order to solve the second problem, we require a minimum absolute distance e from the center to give to the pixel the binary value one (Fig. 10).

Formally, the new eLBP operator is defined as:

$$\mathrm{eLBP}(x_c,\ y_c) = \sum_{n=0}^{7} S_e(i_n - i_c)2^n \tag{3}$$

where function $S_e(x)$ is defined by:

$$S_e(x) = \begin{cases} 1, & |x| \geqslant e \\ 0, & |x| < e \end{cases} \tag{4}$$

The value of e has to be large enough in order to avoid the arbitrary intensity variations caused by noise and small enough to detect all the deterministic intensity changes of texture. In [29] a value near 20 was proved to be satisfactory, after relative experimentation. Although this value was optimal for the discrimination of most text and non-text patterns there were still problems. Some

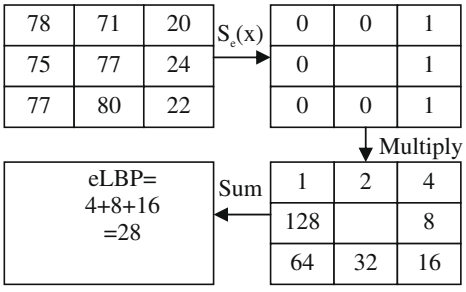

Fig. 10. Example of eLBP computation.

text patterns of low contrast images presented edges that did not exceed the specified threshold and thus, were classified as non-text while non-text patterns of high contrast images presented strong enough edges to be classified as text. In order to solve this problem we propose the generation of multilevel eLBP edge histograms with different values for e, which will describe the edge distribution in different detail levels. However, this will increase dramatically the dimension of the feature set.

In order to reduce the number of histogram bins, namely the number of features, we split the neighbours of every pixel in two groups, the vertical–horizontal and the diagonal neighbours generating two different operators (5) and (6), and thus two different maps with depth equal to $2^4 = 16$ (Fig. 11). In that way, the feature set is reduced from 256 to 32 features, with a slight decrease of the set's discrimination ability. However, the new reduced eLBP descriptor (Eqs. (5) and (6)) can now be used with different values for e, achieving multilevel edge description with impressive performance. By using eight different levels, as will be described in the results section, the final dimension will be 256 instead of 2048 of the full eLBP set. In order to choose the number of different levels and the relative threshold values which will give the best performance with regard to dimensionality, we have to examine the distribution of the gradient values in the vertical, horizontal and diagonal direction. The gradient values are considered as the absolute distance in gray levels, between adjacent pixels in an image.

$$\mathrm{eLBP}_{v-h}(x_c,\ y_c) = \sum_{n=0}^{3} S_e(i_{2*n+1} - i_c)2^n \tag{5}$$

$$\mathrm{eLBP}_{\mathrm{diag}}(x_c,\ y_c) = \sum_{n=0}^{3} S_e(i_{2*n} - i_c)2^n \tag{6}$$

It is well-known [31], and easily verified, that the probability density function (PDF) of an image's wavelet-transform sub-bands, has a Laplacian distribution. The gradient values used here actually resemble to the absolute value of the Haar wavelet coefficients and since the distribution of wavelet coefficients is Laplacian, the PDF of these gradient values will be exponential. Fig. 12 shows a typical PDF of an image gradient. As it can be seen, the probability falls for high distance values. The threshold e at Eq. (4) would actually binarize this distribution in order to distinguish edge from non-edge pixels and then operators (5) and (6) will consider the local spatial distribution of these edge pixels to generate the edge histograms. The optimal set of thresholds for the multilevel edge description will have to cluster the PDF in clusters with equal probability. To this end, we fit the distribution of the image gradient values to the exponential distribution:

$$\mathrm{PDF}_{\exp}(x) = \lambda * e^{-\lambda * x} \tag{7}$$

To achieve that, we calculate the histogram of the gradient values and set its mean value equal to λ^{-1}. An example of PDF fitting can be seen in Fig. 12.

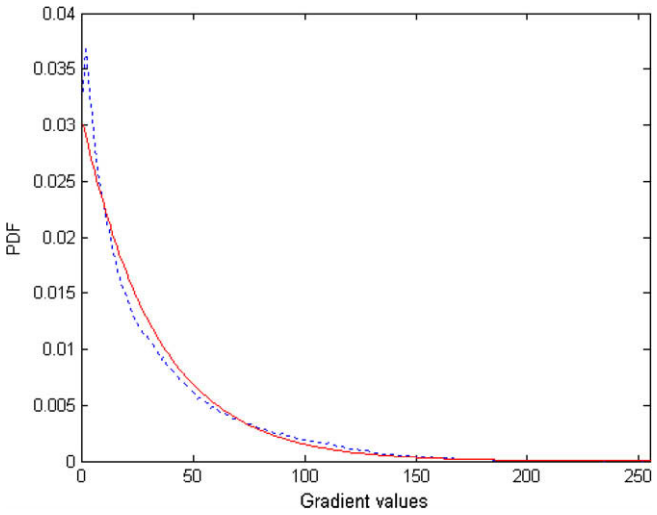

Fig. 12. Typical distribution of image gradient values (blue-dotted), fitted exponential distribution (red-solid).

The quantile function (inverse cumulative distribution function) of PDF_{exp} is denoted as:

$$F_{\exp}(p) = -\lambda^{-1} * \ln(1-p) \tag{8}$$

where $0 \leqslant p < 1$. The quantile function returns the value below which random draws from the given distribution would fall, $p \times 100\%$ of the time. Therefore, in order to cluster our distribution in equal probability clusters we use as thresholds the values of $F_{\exp}$ for equally spaced values of p in [0,1). That is,

$$p = i/(L+1) \tag{9}$$

where $i = 1 \ldots L$ and L is the number of different levels. In that way, the selected threshold values will be denser close to zero where the PDF shows higher values. This PDF fitting will be done for every image portion detected by the heuristic stage, re-establishing the threshold set and giving to the feature set the ability to adapt to cases with different contrast profile.

2.2.2. Saliency map generation

Every sub-image that is detected heuristically as a text line is scanned by a 20×10 sliding window and the responses of the classifier (text = 1, non-text = 0) are accumulated in a saliency map from which the final bounding boxes will be extracted (Fig. 13). The step of the moving window was set to 6 pixels horizontally and 3 pixels vertically, since this values showed a good tradeoff between accuracy and efficiency. Meanwhile another map is generated, in which every value represents the number of visits by the sliding window to the corresponding pixel. This map is used for the normalization of the saliency map from 0 to 1. After normalization, every value of the map represents the estimated by the system probability of the respective pixel, to belong to text area.

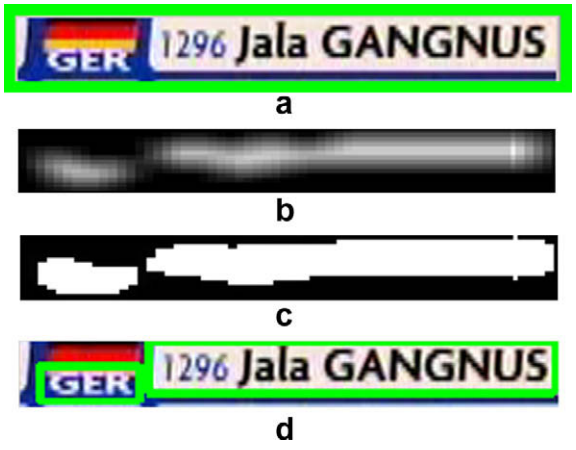

Fig. 14. Example of machine learning refinement. (a) Text block detected heuristically, (b) saliency map, (c) region growing result, (d) refined result.

Fig. 15. The refined result of the coarse outcome shown in Fig. 8.

2.2.3. Region growing algorithm – refined bounding boxes generation

After the saliency map generation a region growing algorithm is applied in order to produce the final result. Two thresholds *th1* and *th2* (with *th1* > *th2*) are used to define whether an area of the map belongs to text. All the pixels of the map with value over *th1* are considered to belong to text and therefore they are used as seeds. Also, if the value of a pixel is below *th1* but over *th2* and has a neighbouring pixel already classified as text it is also considered as a text pixel. The values of *th1* and *th2* are experimentally estimated to 0.8 and 0.4, respectively which means that a text region must contain at least one pixel classified as text by the 80% of the sliding windows (usually in the center of text) but all pixels have to be classified as text by at least 40% of sliding windows. A connected component analysis follows to output one bounding box for every text region. Fig. 14 provides an example of the refinement method while Fig. 15 presents the final result of the refinement step.

The contribution of this stage is the capability of refining instead of just verifying the initial results like the previous hybrid ap-

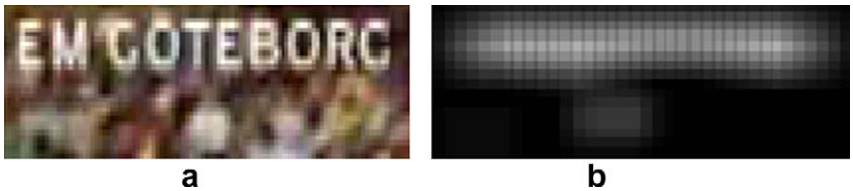

Fig. 13. Example of saliency map generation. (a) Text block detected heuristically, (b) saliency map.

proaches proposed [23–25]. This means that while an image is refined, the machine learning algorithm can:

- Discard a part of the text image as false alarm.
- Discard the whole image.
- Split the image into different text lines.

2.3. Multiresolution analysis

Using differential features in order to detect text gives to the method independence from text color. However, this method clearly depends on the size of the characters. The size of the elements for the morphological operations and the geometrical constraints give to the algorithm the ability to detect text in a specific range of character sizes. Moreover the classifier of the machine learning stage is trained on the same narrow range of font sizes since characters of different sizes present different texture characteristics.

To overcome this problem, we adopt a multiresolution approach. Both coarse and refinement stages of the algorithm are applied to the images in different resolutions and finally the results are combined to the original resolution. This combination might be quite difficult if we consider that the same text might be detected in different resolutions so bounding boxes will overlap. To avoid that, the algorithm suppresses the edges of the already detected characters in a resolution before the edge map is passed to the next resolution. Figs. 16 and 17 give examples of the algorithm performance using multiresolution framework for our two examples.

3. Experimental results and discussion

For our experiments we used a corpus consisting of two video image sets. The first one contains 214 frames from athletic events with 2963 text occurrences while the second one includes 172 frames from news and advertisements with a total of 672 text occurrences. These sets have been generated by selecting video images from 10 different videos, containing artificial and scene text, and constitute a much more general and thus difficult corpus than the one used in [11]. For our experiments, we chose MinH = 9 and MaxH = 30 since the height of the smaller character found in the corpus was equal to nine and the range of sizes considered reasonable for the fixed-scale detector. For the multiresolution analysis, we used two resolutions: the initial, and the one with a scale factor of 0.5. In this way the system can detect characters with height up to 60 pixels which was considered to be satisfying. Taking into account

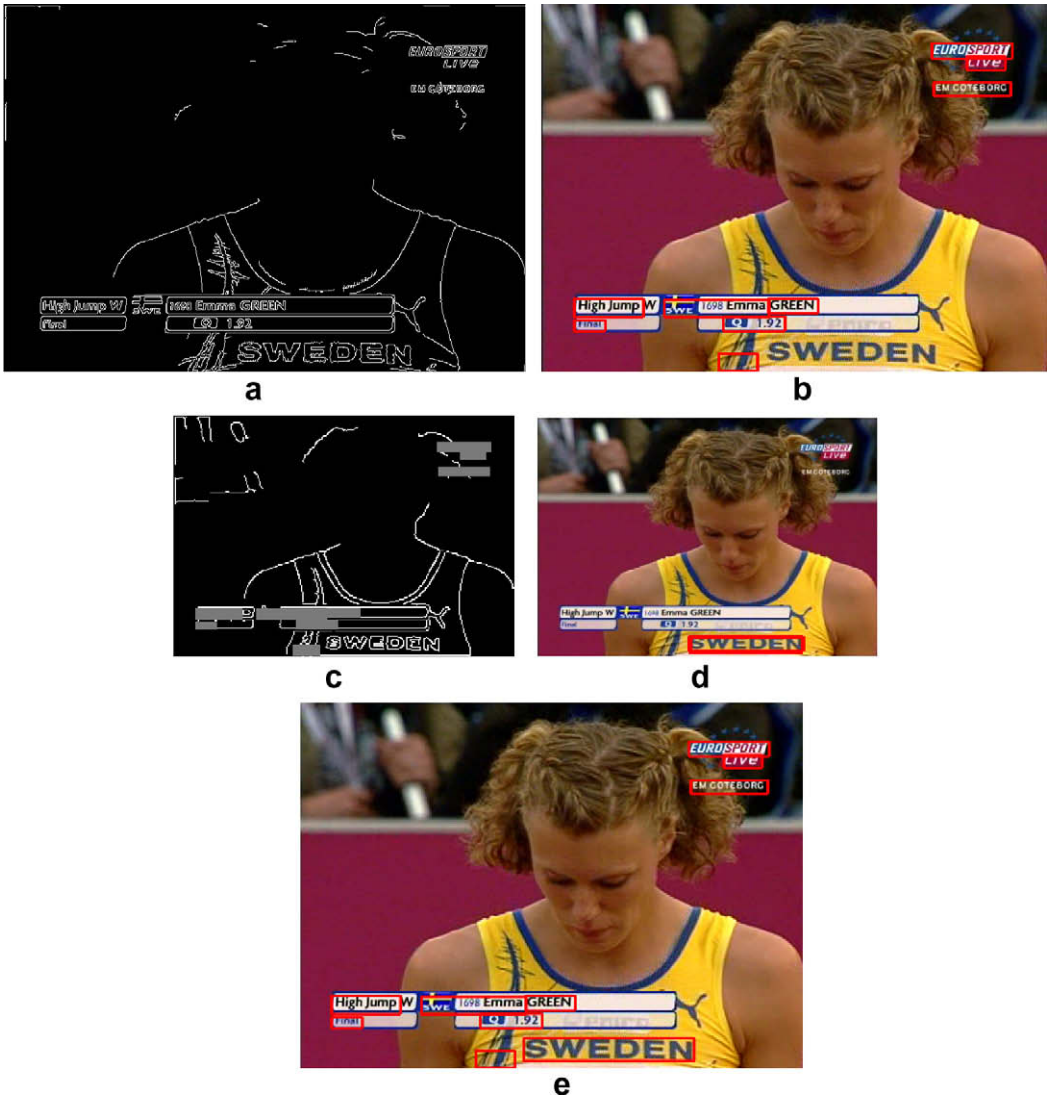

Fig. 16. Multiresolution analysis. (a) Original resolution edge map, (b) result at original resolution, (c) low resolution edge map (gray pixels denote the area of the already found text), (d) result at low resolution, (e) final result.

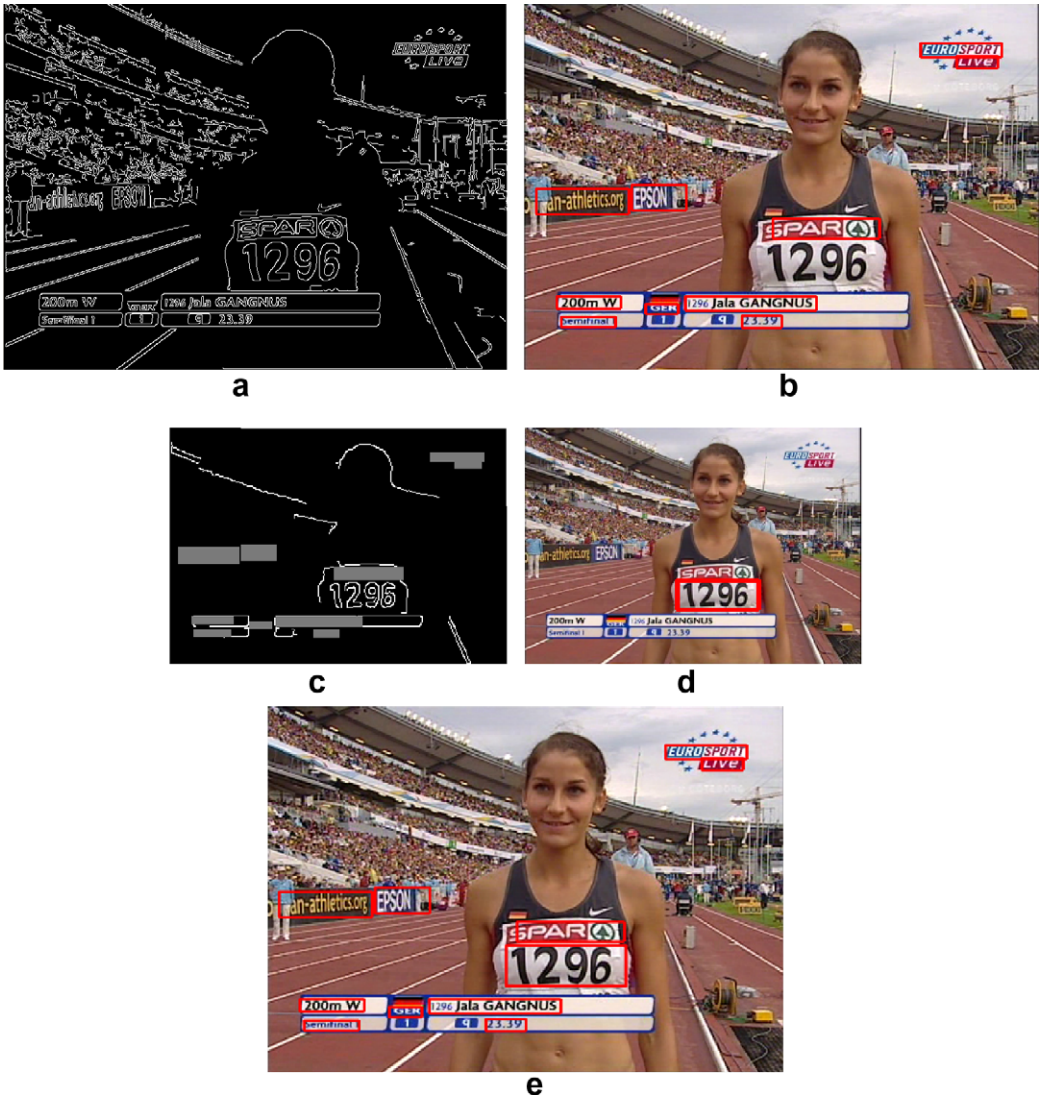

Fig. 17. Multiresolution analysis. (a) Original resolution edge map, (b) result at original resolution, (c) low resolution edge map (gray pixels denote the area of the already found text), (d) result at low resolution, (e) final result.

that text in videos usually does not contain very large characters and from the experience of related experiments, we can assume that these values are typical and adequately generic.

Before discussing the evaluation of the whole system, we present the results of the experiments that compare the proposed feature set with some widely used features and prove its superior discrimination ability. For the experiments, we used as classifier a Support Vector Machine with a Radial Basis Function (RBF) kernel, trained on 3500 text and 6500 non-text patterns selected from 150 representative video images from the videos mentioned above. Examples of text and non-text patterns can be seen in Fig. 18. Each pattern is a 20×10 image that is either entirely contained in a ground truth bounding box (text) or not at all (non-text). Text patterns were taken from textlines with height in the range of 9–30 pixels. All training patterns are taken from the results of the first-heuristic stage so even the non-text patterns belong to areas that were erroneously labelled as text. The comparative results of the feature experimentation are presented in Table 1. For the evaluation of classification we use cross-validation with 10 folds. For our tests, we used the raw values of LH, HL and HH components of the first two levels of Haar decomposition since they have shown better performance than any other wavelet-based feature set. Also, the first coefficient of DCT transform is omitted since it is proportional to the intensity mean and does not contain any frequency information. For the proposed multilevel reduced eLBP descriptor we chose eight levels generating 256 features.

From Table 1 we can see that the proposed feature set achieved much better performance against the best feature sets found in literature. Moreover, we can notice the remarkable performance of the reduced eLBP feature set without sub-pixel information which achieved over 95% accuracy with only 32 features.

The evaluation of the whole text detection system is an aspect not as trivial as it might seem. Most researchers use for their experimentation simple methods such as pixel-based or box-based recall, precision and accuracy measures [10,20,23,25]. Very few works have focused on the problem of evaluation. Moreover these works propose evaluation strategies with several drawbacks and require great effort for the generation of the ground truth ([32–34]).

In [34,33] Kasturi and coworkers propose as overall measure of text detection in a frame, a box-based measure called Frame Detection Accuracy (FDA):

$$\mathrm{FDA} = \frac{\mathrm{Overlap\_Ratio}}{\frac{N_G + N_D}{2}} \tag{10}$$

Fig. 18. (a) Text samples, (b) non-text samples.

Table 1
Results of text/non-text classification using different feature sets.

Features	Feature dimension	Classification accuracy	Text recall	Text precision	Text *F*-measure
LBP	256	93	89.5	89.6	89.6
CGV	200	93.1	89.8	90.2	90
Grayscale raw values	200	93.6	91	90	90.5
Grayscale gradient	400	94.1	90.3	91.9	91.1
Color gradient	400	94.8	94.3	90.6	92.4
Reduced eLBP ($e = 20$)	32	95.1	91.3	93.9	92.6
Haar	180	95.3	92.9	93.2	93.2
eLBP ($e = 20$)	256	96.7	94.8	95.2	95
DCT	199	96.7	95.3	95.2	95.3
Multilevel reduced eLBP ($L = 8$)	256	98	97	97	97

where N_G are the ground truth objects, N_D the detected objects and

$$\text{Overlap\_Ratio} = \sum_{i=1}^{N_{\text{mapped}}} \frac{|G_i \cap D_i|}{|G_i \cup D_i|} \tag{11}$$

here N_{mapped} is the number of mapped object pairs, where the correspondence is established between objects, which have the best spatial overlap. Like many other previous approaches, the authors also propose a thresholding of the overlap ratio in order to forgive minor inconsistencies between the boundaries of the system's output and the ground truth boxes. Thus, the thresholded overlap ratio is defined by:

$$\text{Thresholded\_Overlap\_Ratio} = \sum_{i=1}^{N_{\text{mapped}}} \frac{\text{FDA\_}T(i)}{|G_i \cup D_i|} \tag{12}$$

where

$$\text{FDA\_}T(i) = \begin{cases} |G_i \cup D_i|, & \text{if } \frac{|G_i \cap D_i|}{|G_i \cup D_i|} \geqslant \text{th} \\ |G_i \cap D_i|, & \text{if } \frac{|G_i \cap D_i|}{|G_i \cup D_i|} < \text{th, and non\_binary\_thresholding} \\ 0, & \text{if } \frac{|G_i \cap D_i|}{|G_i \cup D_i|} < \text{th, and binary\_thresholding} \end{cases} \tag{13}$$

The evaluation methods of this kind are based on the mapping between ground truth and detected objects. Especially for the text detection problem, text lines are considered to be the objects where a text line is usually defined as an aligned series of characters with a small intermediate distance relative to their height. However, this subjectively small distance can result arbitrarily in bounding box splits or merges among annotators and detectors making the object mapping inappropriate. In addition, the number of correctly retrieved boxes is not generally a measure of the retrieved textual information since the number of characters in different boxes may vary considerably. Driven by the problem of the text object inexplicit definition, instead of using one-to-one mapping, we based our evaluation method on the overall overlap between the resulted and the ground truth areas.

Wolf and Jilion [35] used the main idea of Liang et al. [36] which was oriented to document page segmentation evaluation and adjusted it to video text detection evaluation. Liang et al. proposed the creation of match score matrices with the overlap between every possible pair of blocks in order to evaluate document structure extraction algorithms. The benefit of this kind of algorithms is their ability to consider the possible splits or merges of the bounding boxes besides one-to-one matching. However, in order to match two ground truth boxes with one resulting box, the total overlap threshold (as described in Wolf et al. paper) has to be very low (~40%). This will have as a result accepting as correct, a box with size even higher than the double size of the ground truth box. Wolf et al. also mentioned the low threshold problem due to the growth of a rectangle area to the square of its side lengths. The same problem is not so intense in the document page segmentation field, since for classic documents the text areas are much bigger compared to the interval created by splits and merges. Moreover Yanikoglu et al. [37] proposed the use of only the text (black) pixels for the computation of overlaps overcoming the low threshold problem for printed documents. However, this approach could not be applied to video text detection since a binarization result is lacking and text pixels cannot be distinguished from background pixels.

Many researchers have used the overall overlap to compute pixel-based recall and precision measures [12,20,23].

$$\text{Recall}_{\text{pixel}} = \frac{|G \cap D|}{|G|} \tag{14}$$

$$\text{Precision}_{\text{pixel}} = \frac{|\text{G} \cap \text{D}|}{|\text{D}|} \tag{15}$$

where G is the ground truth area and D the detected area.

However the main drawback here similarly to the box-based approaches is the fact that the number of retrieved pixels does not correspond to proportional textual information since different textlines may contain characters of various sizes.

Ideally a text detection method as a part of a text extraction system should not be evaluated on the size of detected areas nor the number of detected boxes but on the number of the detected characters. Unfortunately, the number of characters in a bounding box cannot be defined by the algorithm but it can be approximated by the ratio width/height of the box, if we assume that this ratio is invariable for every character, the spaces between different words in a text line are proportional to its height and each textline contains characters of the same size.

Table 2
Evaluation values for images of Fig. 19 without thresholding.

(%)	$\text{Recall}_{\text{ecn}}$	$\text{Precision}_{\text{ecn}}$	F_{ecn}	$\text{Recall}_{\text{pixel}}$	$\text{Precision}_{\text{pixel}}$	F_{pixel}	FDA
Fig. 19a	70	79.5	74.4	32.5	80.3	46.3	54.7
Fig. 19b	81.9	100	90	81.9	100	90	36.3
Fig. 19c	7.3	100	13.6	10.6	100	19.2	50
Fig. 19d	85.7	44	58.1	85.7	44	58.1	29

Table 3
Evaluation values of Wolf and Jolion [35] for images of Fig. 19 (with default parameters).

(%)	$\text{Recall}_{\text{ecn}}$	$\text{Precision}_{\text{ecn}}$	*F*-measure
Fig. 19a	33.3	50	40
Fig. 19b	80	100	88.89
Fig. 19c	33.3	100	50
Fig. 19d	100	100	100

Let w_i, h_i, r_i be the width, the height, and the ratio width to height respectively, with $i = c, s, t$ denoting characters, spaces and textlines. Obviously we can assume that $h_c = h_s = h_t$. A textline containing n_c characters will have a width $w_t = n_c w_c + n_s w_s$, where n_s is the number of spaces. The number of spaces between n_c characters is $n_s = n_c - 1$. Assuming that there are also spaces at the ends of the text line, approximately equal to $w_s/2$, the number of spaces becomes equal to the number of characters: $n_s = n_c$.

Thus, $w_t = n_c w_c + n_c w_s = n_c(w_c + w_s) = n_c(h_c r_c + h_s r_s) = n_c h_t (r_c + r_s) \Rightarrow n_c = \frac{r_t}{(r_c + r_s)}$ where r_c and r_s are constants. In other words, the number of characters in a text line is proportional to the ratio r_t.

Based on that, we define that the contribution of every box to the overall evaluation will be $r_t = w_t/h_t$ and therefore the contribution of every pixel that belongs to the box will be $\frac{w_t h_t}{w_t * h_t} = \frac{1}{h_t^2}$. In that way, the evaluation will be based on the recall and precision of the area coverage, normalized by the approximation of the number of characters for every box (see Eqs. (16) and (17)). The overall metric will be the weighted harmonic mean of precision and recall also referred as the *F*-measure (18). The normalization of a ground truth bounding box obviously estimates the actual number of characters contained in the box, while the same technique for the detected boxes estimates the number of characters of the recognition system's output, since before recognition text images are normalized according to their height.

$$\text{Recall}_{\text{ecn}} = \frac{\sum_{i=1}^{N} \frac{|\text{GDI}_i|}{hg_i^2}}{\sum_{i=1}^{N} \frac{|\text{GB}_i|}{hg_i^2}} \tag{16}$$

$$\text{Precision}_{\text{ecn}} = \frac{\sum_{i=1}^{M} \frac{|\text{DGI}_i|}{hd_i^2}}{\sum_{i=1}^{M} \frac{|\text{DB}_i|}{hd_i^2}} \tag{17}$$

$$F_{\text{ecn}} = \frac{2 * \text{Precision}_{\text{ecn}} * \text{Recall}_{\text{ecn}}}{\text{Precision}_{\text{ecn}} + \text{Recall}_{\text{ecn}}} \tag{18}$$

where GB_i is the ground truth bounding box number i and hg_i is its height, while DB_i is the detected bounding box number i and hd_i is its height. N is the number of ground truth bounding boxes and M is the number of detected bounding boxes and GDI, DGI are the corresponding intersections:

$$\text{GDI}_i = \text{GB}_i \cap \left(\bigcup_{i=1}^{M} \text{DB}_i \right) \tag{19}$$

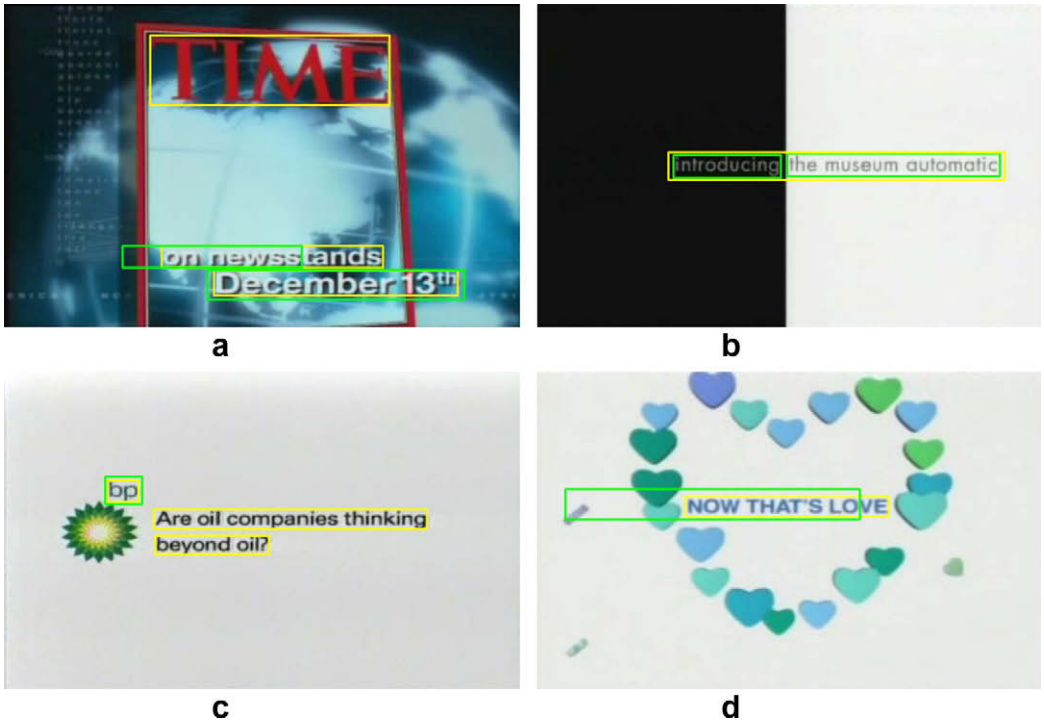

Fig. 19. Examples of text detection (yellow = ground truth, green = supposed output).

Table 4
Evaluation values for images of Fig. 19 with non-binary thresholding (th = 80%).

(%)	Recall_{ecn}	Precision_{ecn}	F_{ecn}	Recall_{pixel}	Precision_{pixel}	F_{pixel}	FDA
Fig. 19a	70	88.4	78.1	32.5	91.7	48	61.7
Fig. 19b	100	100	100	100	100	100	36.3
Fig. 19c	7.3	100	13.6	10.6	100	19.2	50
Fig. 19d	100	44	61.1	100	44	61.1	29

$$\text{DGI}_i = \text{DB}_i \cap \left(\bigcup_{i=1}^{N} \text{GB}_i \right) \tag{20}$$

In order to avoid penalizing minor inconsistencies by non-binary thresholding similarly to (13), the definitions (19) and (20) become:

$$\text{GDI}_i = \begin{cases} \text{GB}_i, & \text{if } \dfrac{\text{GB}_i \cap \left(\bigcup_{i=1}^{M} \text{DB}_i \right)}{\text{GB}_i} \geqslant \text{th} \\ \text{GB}_i \cap \left(\bigcup_{i=1}^{M} \text{DB}_i \right), & \text{if } \dfrac{\text{GB}_i \cap \left(\bigcup_{i=1}^{M} \text{DB}_i \right)}{\text{GB}_i} < \text{th} \end{cases} \tag{21}$$

$$\text{DGI}_i = \begin{cases} \text{DB}_i, & \text{if } \dfrac{\text{DB}_i \cap \left(\bigcup_{i=1}^{N} \text{GB}_i \right)}{\text{DB}_i} \geqslant \text{th} \\ \text{DB}_i \cap \left(\bigcup_{i=1}^{N} \text{GB}_i \right), & \text{if } \dfrac{\text{DB}_i \cap \left(\bigcup_{i=1}^{N} \text{GB}_i \right)}{DB_i} < \text{th} \end{cases} \tag{22}$$

In Tables 2 and 3 we can see the results of the different evaluation measures presented above for the four examples of Fig. 19. Looking at the images and the corresponding values we can notice that the proposed estimated character-based measures were the most representative. In Fig. 19a the true character recall was 21/30 = 70% (with the spaces between words counted) exactly equal to the proposed recall while the pixel-based recall was 32.5% and Wolf's recall was 33.3%. Fig. 19b presents an obviously good detection result, however FDA was only 36.3% due to compulsory one to one matching. On the other hand, although in Fig. 19c the result was quite poor, FDA and Wolf's *F*-measure scored 50% deceived by the box-based strategy. Moreover, the evaluation method of Wolf will score 100% for the unsatisfactory result of Fig. 19d due to the low threshold values enforced by the multi-to-multi matching policy. Table 4 presents the corresponding results to Table 2 with non-binary thresholding and th = 80. Using thresholding we can see that the system forgive the non perfect detection of the third text line in Fig. 19a, increasing the precision. FDA is also increased. However, for the Fig. 19b although the pixel based and the proposed evaluation strategy give 100%, FDA still fails due to one to one mapping.

Table 5 provides the results of the whole text detection system before and after the use of the machine learning refinement stage, as well as the performances of three state-of-the-art methods presented in Section 1. The refinement stage of the proposed methodology increases the precision rate and combined with the high recall of the initial result makes the overall system performance to improve from 64.2% and 74.2% to 81.47% and 83%, for the first and the second set respectively. Table 6 presents the corresponding results using non-binary thresholding (see Eqs. (21) and (22)). From Table 6 we can see that the results after the refinement stage have improved much more, compared to Table 5, than the coarse results. This means that the refined bounding boxes tend to resemble to the ground truth bounding boxes. The better performance of the system for *Set2* was expected because of the smoother background in these kinds of videos. Table 7 shows the average processing time of the proposed, and the state-of-the art algorithms for images with resolution 768×576. The experiments were conducted on an Intel single core CPU at 3.2 GHz. Regarding computational complexity, the most time consuming part of text detection methods are the prediction calls to the classifier for the machine learning stage. Although SVM proved to be the most time consuming classifier, it is chosen because of its superior performance.

Table 5
Performance of different text detection algorithms without thresholding.

	Frames #	Text boxes #	Method	Recall (%)	Precision (%)	*F*-measure (%)
Set1	214	2963	Proposed without refinement	93.5	48.9	64.2
			[18]	66.9	66.7	66.8
			[23]	65.4	75.6	70.1
			[21]	81.1	70.5	75.4
			Proposed with refinement	83.9	79	81.4
Set2	172	672	[18]	63.3	69.2	66.1
			[23]	68.2	71.1	69.6
			Proposed without refinement	93.3	61.6	74.2
			[21]	80.6	71.5	75.8
			Proposed with refinement	82.7	83.5	83

Table 6
Performance of different text detection algorithms with thresholding (th = 80%).

	Frames #	Text boxes #	Method	Recall (%)	Precision (%)	*F*-measure (%)
			Proposed without refinement	94	49.9	65.2
Set1	214	2963	[18]	68.3	68.6	68.4
			[23]	68.9	80.5	74.2
			[21]	85.5	73.3	78.9
			Proposed with refinement	86.8	84.5	85.6
Set2	172	672	[18]	65	69.8	67.3
			[23]	69.5	72.8	71.1
			Proposed without refinement	95.4	65.5	77.7
			[21]	84.2	74	78.8
			Proposed with refinement	87	88.2	87.7

4. Conclusion

In this paper we presented a two-stage system for text detection in video images. The system consists of a very efficient, edge-based, first stage with high recall and a second machine learning refinement stage which improves performance by reducing the false alarms. The main contributions of this work are the highly discriminating feature set based on a new texture operator, and the incorporation of the refinement stage which is based on a

Table 7
Average processing time per frame of the different text detection algorithms.

Method	Average processing time per frame (s)
Proposed without refinement	0.33
[18]	8
[23]	3.35
[21]	1.5
Proposed with refinement	2

sliding window, an SVM classifier and a saliency map. The system's performance evaluation is based on the intersection of the ground truth and the resulted bounding boxes, normalized by the estimated number of contained characters. Experimental results showed great robustness on the detection of horizontal textlines in very complex backgrounds even for scene text. The method does not take into consideration any temporal information in order to separate the text detection from a text tracking stage that may be used to track text lines between periodical spatial detections.

Acknowledgement

Part of this research was carried out within the framework of the co-funded by the EU project CASAM (FP7-ICT-217061).

References

[1] R. Lienhart, Video OCR: A Survey and Practitioner's Guide, Video Mining, Kluwer Academic Publisher, 2003, pp. 155–184
[2] K. Jung, K.I. Kim, A.K. Jain, Text information extraction in images and video: a survey, Pattern Recognition 37 (5) (2004) 977–997.
[3] J. Liang, D. Doermann, H. Li, Camera-based analysis of text and documents: a survey, International Journal on Document Analysis and Recognition 7 (2–3) (2005) 84–104.
[4] D. Doermann, J. Liang, H. Li, Progress in camera based document image analysis, in: International Conference on Document Analysis and Recognition, 2003, pp. 606–617.
[5] R. Lienhart, W. Effelsberg, Automatic text segmentation and text recognition for video indexing, ACM/Springer Multimedia Systems 8 (2000) 69–81.
[6] 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.
[7] T. Sato, T. Kanade, E. Hughes, M. Smith, Video OCR for digital news archives, IEEE Workshop on Content-Based Access of Image and Video Databases (1998) 52–60.
[8] J. Xi, X.-S. Hua, X.-R. Chen, L. Wenyin, H. Zhang, A video text detection and recognition system, multimedia and expo, IEEE International Conference (2001) 873–876.
[9] M. Cai, J. Song, M.R. Lyu, A new approach for video text detection, IEEE International Conference on Image Processing (2002) 117–120.
[10] M.R. Lyu, J. Song, M. Cai, A comprehensive method for multilingual video text detection, localization, and extraction, IEEE Transactions. Circuit and Systems for Video Technology 15 (2) (2005) 243–255.
[11] M. Anthimopoulos, B. Gatos, I. Pratikakis, Multiresolution text detection in video frames, in: International Conference on Computer Vision Theory and Applications, 2007, pp. 161–166.
[12] W. Kim, C. Kim, A new approach for overlay text detection and extraction from complex video scene, Image Processing, IEEE Transactions 18 (2) (2009) 401–411.
[13] Y. Zhong, H. Zhang, A.K. Jain, Automatic caption localization in compressed video, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (4) (2000) 385–392.
[14] D. Crandall, S. Antani, R. Kasturi, Extraction of special effects caption text events from digital video, International Journal on Document Analysis and Recognition 5 (2–3) (2003) 138–157.
[15] Y.K. Lim, S.H. Choi, S.W. Lee, Text extraction in MPEG compressed video for content-based indexing, International Conference on Pattern Recognition (2000) 409–412.
[16] U. Gargi, D.J. Crandall, S. Antani, T. Gandhi, R. Keener, R. Kasturi, A system for automatic text detection in video, International Conference on Document Analysis and Recognition (1999) 29–32.
[17] K. Jung, Neural network-based text location in color images, Pattern Recognition Letters 22 (14) (2001) 1503–1515.
[18] K.I. Kim, K. Jung, S.H. Park, H.J. Kim, Support vector machine-based text detection in digital video, Pattern Recognition 34 (2) (2001) 527–529.
[19] C. Wolf, J.-M. Jolion, Model Based Text Detection in Images and Videos: A Learning Approach, Technical Report LIRIS-RR-2004-13 Laboratoire d'Informatique en Images et Systemes d'Information, INSA de Lyon, France, 2004.
[20] R. Lienhart, A. Wernicke, Localizing and segmenting text in images and videos, IEEE Transactions on Circuits and Systems for Video Technology 12 (4) (2002) 256–268.
[21] H. Li, D. Doermann, O. Kia, Automatic text detection and tracking in digital video, IEEE Transactions on Image Processing 9 (1) (2000) 147–156.
[22] H. Zhang, W. Gao, X. Chen, D. Zhao, Learning informative features for spatial histogram-based object detection, International Joint Conference on Neural Networks 3 (2005) 1806–1811.
[23] D. Chen, J.-M. Odobez, J.-P. Thiran, A localization/verification scheme for finding text in images and videos based on contrast independent features and machine learning methods, Image Communication 19 (3) (2004) 205–217.
[24] Q. Ye, Q. Huang, W. Gao, D. Zhao, Fast and robust text detection in images and video frames, Image Vision Computing 23 (6) (2005) 565–576.
[25] C. Jung, Q. Liu, J. Kim, A stroke filter and its application to text localization, Pattern Recognition Letters 30 (2) (2009) 114–122.
[26] R. Gonzalez, R. Woods, Digital Image Processing Addison Wesley (1992).
[27] J. Canny, A computational approach to edge detection, IEEE Transactions, Pattern Analysis and Machine Intelligence 8 (1986) 679–698.
[28] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.
[29] M. Anthimopoulos, B. Gatos, I. Pratikakis, A hybrid system for text detection in video frames, in: International Workshop on Document Analysis Systems, 2008, pp. 286–292.
[30] T. Ojala, M. Pietikainen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognition 29 (1) (1996) 51–59.
[31] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image coding using wavelet transform, IEEE Transactions, Image Processing (1992) 205–220.
[32] X.-S. Hua, L. Wenyin, H.-J. Zhang, An automatic performance evaluation protocol for video text detection algorithms, IEEE Transactions on Circuits and Systems for Video Technology 14 (4) (2004) 498–507.
[33] M. Vasant, P. Soundararajan, M. Boonstra, H. Raju, D. Goldgof, R. Kasturi, J. Garofolo, Performance evaluation of text detection and tracking in video. International Workshop on Document Analysis Systems, pp. 576–587.
[34] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, J. Zhang, Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2008) 319–336.
[35] C. Wolf, J. Jolion, Object count/area graphs for the evaluation of object detection and segmentation algorithms, International Journal on Document Analysis and Recognition 8 (4) (2006) 280–296.
[36] J. Liang, I.T. Phillips, R.M. Haralick, Performance evaluation of document layout analysis algorithms on the UW Data Set, Document Recognition IV, SPIE (1997) 149–160.
[37] B.A. Yanikoglu, L Vincent, Pink panther: a complete environment for ground-truthing and benchmarking document page segmentation, Pattern Recognition 31 (9) (1998) 1191–1204.