# A Hybrid System for Text Detection in Video Frames

Marios Anthimopoulos, Basilis Gatos, Ioannis Pratikakis
*Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,*
*National Center for Scientific Research "Demokritos",*
*GR-153 10 Agia Paraskevi, Athens, Greece.*
*{anthimop, bgat, ipratika}@iit.demokritos.gr*

## Abstract

*This paper proposes a hybrid system for text detection in video frames. The system consists of two main stages. In the first stage text regions are detected based on the edge map of the image leading in a high recall rate with minimum computation requirements. In the sequel, a refinement stage uses an SVM classifier trained on features obtained by a new Local Binary Pattern based operator which results in diminishing false alarms. Experimental results show the overall performance of the system that proves the discriminating ability of the proposed feature set.*

## 1. Introduction

The proliferation of multimedia content has raised the need for automatic content-based indexing and information retrieval systems. Many methods have been proposed for the extraction of various level semantics from video and audio. Textual information in video and images proves to be a source of high-level semantics closely related to the concept of the video.

There exist mainly two kinds of text occurrences in videos, namely artificial and scene text. Artificial text 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 camera as part of scene such as text on T-shirts or road signs. Scene text can appear in any kind of surfaces, in any orientation and perspective and often under occlusion, making its extraction particularly difficult. Moreover scene text usually brings less related to video information. In Figure 1, green boxes denote artificial text while red boxes bound the scene text. Text can also be classified into normal or inverse. Normal is called any text whose characters have lower intensity values than the background while inverse text is the opposite. In Figure 2 "EURO" is inverse while "SPORT" is normal text.

The procedure of textual information extraction from video and images is usually split into three steps: detection, segmentation and recognition. The step of detection is the most crucial step and although it has been extensively studied in the past decade, presenting quite promising results there are still challenges to meet.


**Figure 1** Example of artificial and scene text


**Figure 2** Example of inverse and normal text

## 2. Related work

Text detection methods can be classified into two categories: bottom-up and top-down methods.

Bottom-up methods detect character regions in the image and then group them into words and text lines. These are the first methods proposed, derived from document analysis research area. They can perform

978-0-7695-3337-7/08 $25.00 © 2008 IEEE
DOI 10.1109/DAS.2008.72

satisfactory in high quality images with simple background and known text color. Typical Bottom-up approaches can be found in [1] and [2].

Top-down methods firstly detect text regions in images and then split them in text lines. They treat text areas as a distinct texture and try to segment it from any other texture. These methods are also divided into three sub-categories: Heuristic, Machine learning and hybrid methods.

Heuristic methods use empirical rules and thresholds in order to distinguish text from non text areas. They are usually based on the gradient density and some heuristic spatial and geometrical constraints derived from text characteristics. 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.Xi et al. [3] propose an edge based method based on an edge map created by Sobel operator followed by smoothing filters, morphological operations and geometrical constraints. Sato et al. [5] apply a 3x3 horizontal differential filter to the entire image with appropriate binary thresholding followed by size, fill factor and horizontal-vertical aspect ratio constraints. Du et al. [6] propose a methodology that uses MPCM (Multistage Pulse Code Modulation) to locate potential text regions in colour video images and then applies a sequence of spatial filters to remove noisy regions, merges text regions, produces boxes and finally eliminates the text boxes that produce no OCR output.DCT coefficients of intensity images have been widely used as texture features and have also used for text detection ([4], [7] [20], [21]). 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 that case, the pre-computed coefficients of 8x8 pixel block units are used. However, an 8x8 block 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.

Many machine learning approaches have been proposed the last years for the detection of text areas with great success. These algorithms are based on classification techniques trained on text and non text patterns which scan the image in order to localize the text occurrences. Machine learning classifiers have proved to be the best solution for many problems having stochastic characteristics without a rigid mathematical definition. The main problem of the methods belonging to this category is the high computational complexity. A sliding window scans the entire image with a typical step of 3 or 4 pixels, demanding many thousands of prediction calls to the classifier.Wolf et al. [9] use an SVM trained on derivative and geometrical features. Yan et al. [10] use a Back Propagation Artificial Neural Network trained on Gabor edge features. Ye et al. [11] use SVM and wavelets. Wu et al. [12] propose a system of two co-trained SVM's on edge and color features. Clark et al. [14] presents five statistical measures for training a Neural Network.Lienhart et al. in [13] used as features the complex values of the gradient of the RGB input image. The channels of the image are split and for each channel the horizontal and vertical derivatives are computed. The final gradient image is the sum of R, G and B derivatives for each direction. For the estimation of the derivatives, Sobel masks are the most common and robust solution.Li et al. [8] suggest using 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. Wavelet decomposition naturally captures directional frequency content at different scales. Li et al. use a three-layer neural network and a 16x16 sliding window.Zhang, et al. [19] proposed a system for object detection based on Local Binary Patterns (LBP) and Cascade histogram matching. The LBP operator consists of a 3x3 kernel where the center pixel is used as a threshold. Then the eight binarized neighbours are multiplied by the binomial weight producing an integer that represents a unique texture pattern. Zhang applied the proposed method to videotext and car detection.

Some hybrid methods have also been proposed. 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 boxes as false alarms. In [15] Chen et al. use a localization/verification scheme which claim to be highly efficient and effective. For the verification part features like Greyscale spatial derivatives, distance maps, constant gradient variance and DCT coefficients are fed to an SVM classifier. However the verification task can only decide if an initial result is real text or not without having the capability to refine it. This means that if a resulted 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.

In this work we propose a new hybrid approach that combines accuracy with efficiency. As a first stage the algorithm applies a very fast heuristic method with a great recall rate and then a more sophisticated machine learning technique is used to refine the result in every bounding box of the initial result and minimize false

alarms. This technique is based on features derived from a new operator that captures the edge structure. The structure of the remaining of our paper is as follows: Section 2 describes the first, heuristic stage of the system, section 3 presents the second, machine learning refinement stage, section 4 provides the results and relative discussion and finally section 5 concludes.

## 2. Heuristic coarse text detection

For the first, coarse stage of text detection, we use an algorithm proposed by Anthimopoulos et al. [16]. The algorithm (Figure 3) exploits the fact that text lines produce strong vertical edges horizontally aligned and follow specific shape restrictions. Using edges as the prominent feature of our system gives us the opportunity to detect characters with different fonts and colours since every character present strong edges, despite its font or color, in order to be readable.

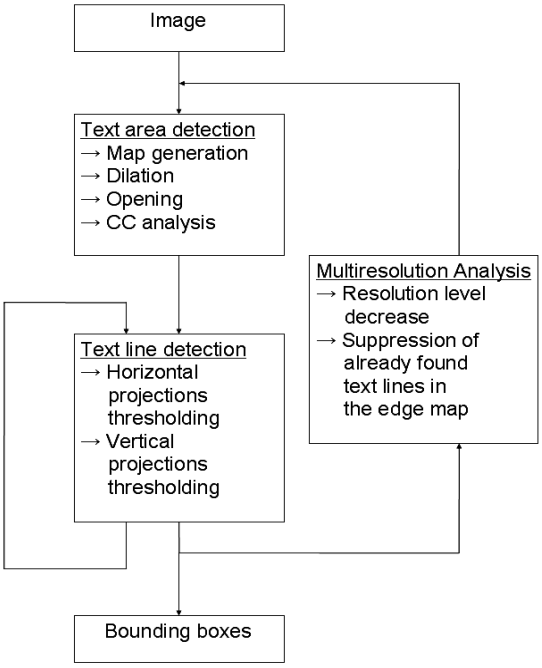
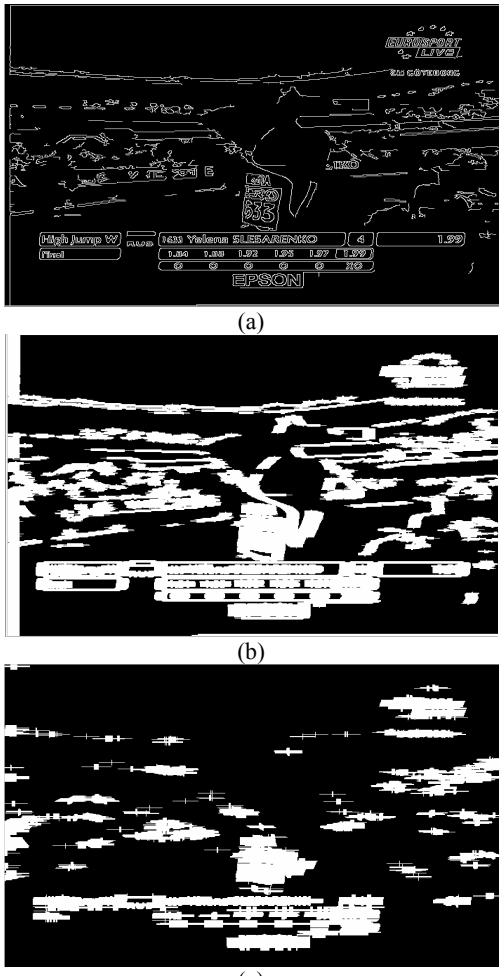



**Figure 3** Flowchart of heuristic text detection method

Two are the main steps here, text area detection and text line detection, applied in a multiresolution manner. In the first step, an edge map is created using the Canny edge detector [17]. Then, morphological dilation and opening are used in order to connect the vertical edges and eliminate false alarms. Bounding boxes are determined for every non-zero valued connected component, consisting the initial candidate text areas. Finally, edge projection analysis is applied, refining the result and splitting text areas in text lines. The whole algorithm is applied in different resolutions to ensure text detection with size variability.



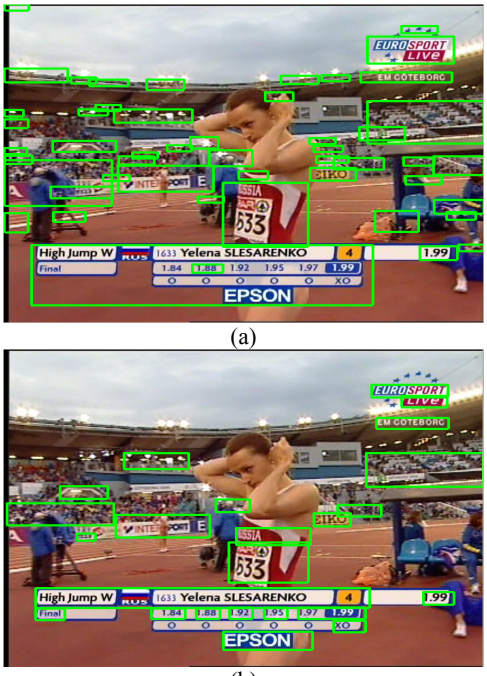

(a) (b) (c)

**Figure 4** Text area detection. (a) Edge map; (b) Dilation; (c) Opening.

This approach performs successfully when the needed parameters are estimated for restricted corpuses with relatively low-contrast background. However it lacks generality since there is not an optimal way of choosing the threshold value of the edge detector. This, crucial for the method, threshold has to be large enough to eliminate the background edges but not so large to eliminate text edges also. In the proposed methodology we set this threshold to a low value that assures that almost every text occurrence will be detected. However, this causes a large amount of false alarms which are intended to be discarded by the following machine learning step.

Figures 4 and 5 show an example of the heuristic algorithm applied in image of figure 1. The effects of

using a low threshold for the edge detector are obvious. The edge density is high even in areas without text. The result is greatly improved after projection analysis (Figure 5(b)) although some false alarms still remain.




(a)

(b)

**Figure 5** (a) Initial bounding boxes; (b) Bounding boxes after projection analysis.

## 3. Machine learning refinement

Edge-based heuristic methods detect text based mainly on the edge density. However in many cases, non text regions present edge density values adequate to produce false alarms that human optical perception system would have avoid. This fact provided motivation to the researchers to explore larger sets of features which capture not only the abrupt intensity changes of the image, but their two-dimensional distribution as well. The large number of features and the great generalization capability of Support Vector Machines (SVM's) [23] led us to use an SVM and a sliding window model to refine the result. The most important aspect, though, in designing the machine learning technique is the choice of the features.

### 3.1. Feature extraction

The majority of the features used for text detection originate from texture segmentation or object detection research areas. Some researchers refer to text as an object while others consider it as a texture. We can say that every character is considered as an object and thus a text line as a periodic repetition of similar objects with specific alignment. The question is: "Does really text have the characteristics of a texture?" The fact is that there is not a formal definition for texture although there have been some attempts. Cross & Jain [22] argued: "We consider a texture to be a stochastic, possibly periodic, two-dimensional image field." In that sense we can refer to text as texture having though some special characteristics.

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 LBP for text detection and adjust it to the specific problem.

LBP was originally introduced by Ojala et al.[18] as a non parametric operator measuring the local contrast for efficient texture classification. The LBP operator consists of a 3x3 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 between 0 and $2^8$-1=255 (Figure 6). Each of these 256 different 8-bit words represents a unique texture pattern.

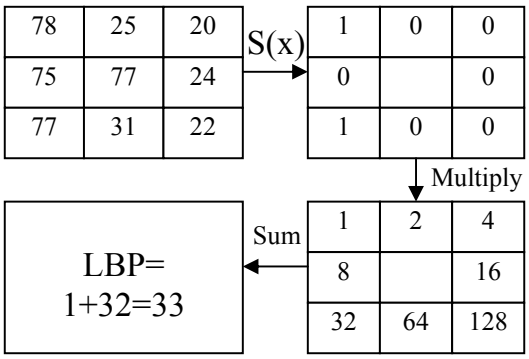



**Figure 6** Example of LBP computation

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

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

where $i_c$ corresponds to the grey value of the center pixel ($x_c$, $y_c$), $i_n$ to the grey values of the 8 surrounding pixels, and function s(x) is defined as:

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

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 image. Thus, the 256 histogram values of an image region depict its texture structure and can be used as features.

Although the original LBP operator has showed 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 Patern (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 equal to the center pixel or 1 if not. In that way we solve the first problem mentioned above since we capture only the fact that a pixel is equal or different to the center, recognising normal and inverse text as the same texture. In order to solve the second problem, we require a minimum absolute distance *d* from the center to give to the pixel the binary value 1 (Figure 7).

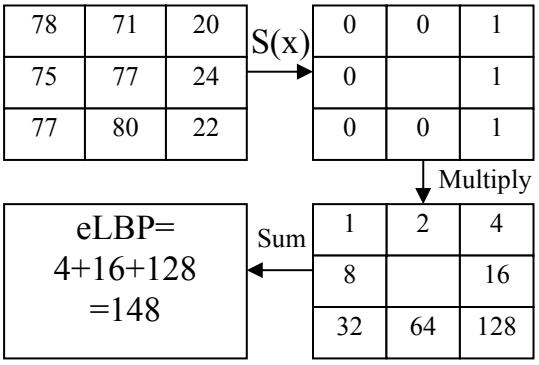



**Figure 7** Example of eLBP computation

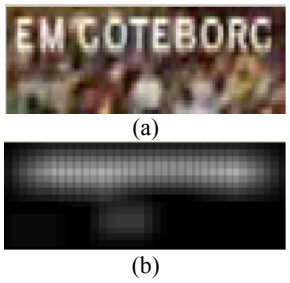
Formally, the new eLBP operator differs from the original in the definition of function S(x):

$$S(x) = \begin{cases} 1 , |x| \geq d \\ 0 , |x| < d \end{cases} \quad (3)$$

The value of d 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. Although inserting a heuristic parameter is not usually intended for creating a generic method the actual value of *d* does not affect severely the result. Experimentally, a value near 20 proved to be satisfactory.

### 3.2. Saliency map generation

Every sub-image that is detected heuristically as a text line is scanned by a 20x10 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 (Figure 8).The step of the moving window was set to 4 pixels since this value showed a good trade off between accuracy and efficiency. The same procedure is applied in different scales in order to detect text with different sizes since the designed detector is fixed-size.




(a)

(b)

**Figure 8** Example of Saliency map generation. (a) Text block detected heuristically, (b) Saliency map

### 3.3. Refined bounding boxes generation

After the saliency map generation a region growing algorithm is applied in order to produce the final result. All the pixels of the map with value over th1 are considered to belong to text. Also if the value of a pixel is below th1 but over th2 and has a neighboring pixel already classified as text it is also considered as text pixel. The threshold values th1 and th2, with th1>th2 depend on the size of the sliding window and the sliding step. A connected component analysis follows to output one bounding box for every text region. Figure 9 provides an example of the refinement method while figure 10 presents the final result of the refinement step.

The contribution of this stage is that the image which has been previously detected as text is scanned for textual occurrences without making any assumptions about the success of the heuristic stage. This means that while a sub-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.

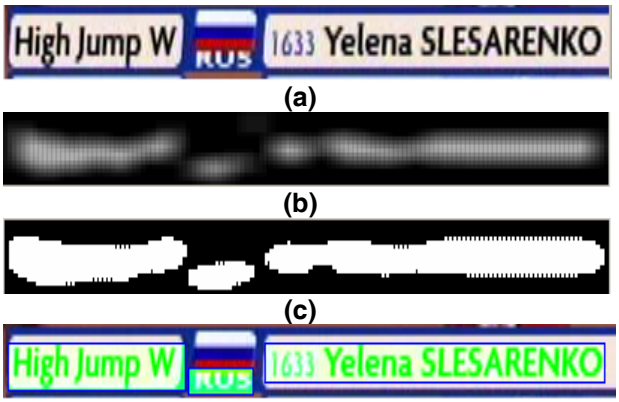



**Figure 9** Example of machine learning refinement. (a) Text block detected heuristically; (b) Saliency map; (c) Region growing result; (d) refined result.


**Figure 10** Final result of the method

## 4. Results and discussion

For the experiments we used as classifier a Support Vector Machine trained on 3500 text and 6500 non-text patterns (figure 11). Each pattern is a 20x10 image that is either entirely contained in a ground truth bounding box (text) or not at all (non-text). This database was created by 150 captured frames from 5 different videos concerning news broadcasts, commercials and athletic events. Text patterns were taken from textlines with height in the range of 10 to 20 pixels. 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 showed better performance. Also the first coefficient of DCT transform is omitted since it is proportional to the intensity mean and does not contain frequency information.

(a) (b)

**Figure 11** (a) Text samples; (b) Non-text samples

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

| Features | Correctly classified | Text recall | Text precision | #of features |
|---|---|---|---|---|
| DCT | 96.7 | 95.3 | 95.2 | 199 |
| Color Gradient | 94.2 | 93.8 | 89 | 400 |
| Haar | 95.3 | 92.9 | 93.2 | 180 |
| LBP | 93 | 89.5 | 89.6 | 256 |
| eLBP | 96.7 | 94.8 | 95.2 | 256 |

From the table we can see that the best performance is achieved by DCT coefficients and eLBP histogram features. However calculating DCT for every 20x10 block will be computational prohibitive. For this reason many researchers proposed methods based on feature maps like gradients. Except the fact that these kinds of maps are generated really fast they also benefit in another way. Each feature, namely each pixel of the map is calculated once but it is used as feature from many different overlapping sliding windows. The proposed feature set provides equivalent results with DCT but better than the commonly used gradient with the additional advantage of a relative small feature set which is actually independent of the size of the window (if normalized).

For the evaluation of the entire system we used a set of 110 frames containing 1640 text occurrences. As evaluation measures we adopted recall and precision rates in a pixel basis as well as their F-measure. This

set has been generated by selecting frames from 5 different videos, containing artificial and scene text, and consists a much more general and thus difficult corpus than the one used in [16].

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 rise from 66.17% to 76.42%. However, evaluating the result of text detection is not as trivial as it might seem. The fall of recall rate after refinement in many cases is the result of the tighter bounding boxes which are in fact totally correct. On the other hand in some cases precision falls because the system detects barely readable scene text that was not considered as text in the ground-truthing procedure.

## 5. Conclusion

In this paper we presented a hybrid system for text detection in video frames. The system consists of a very efficient first stage with high recall and a second machine learning refinement stage which reduces the false alarms. The main contributions of this work are the highly discriminating feature set based on a new texture operator, and the architecture of the refinement stage which is based on a sliding window and an SVM classifier.

## 5. References


[1] Lienhart Rainer and Frank Stuber, 1995. Automatic text recognition in digital videos, Technical Report / Department for Mathematics and Computer Science, University of Mannheim.

[2] Sobottka K. and Bunke H., 1999. Identification of Text on Colored Book and Journal Covers, International Conference on Document Analysis and Recognition, Bangalore, India, pp. 57-62.

[3] Xi Jie, Xian-Sheng Hua, Xiang-Rong Chen, 2001. Liu Wenyin, HongJiang Zhang, A Video Text Detection And Recognition System, IEEE International

[4] Y. Zhong, H. Zhang, and A.K. Jain. Automatic caption localization in compressed video.
*IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(4):385–392, 2000.

[5] Sato T. , Kanade T. , E. Hughes, and M. Smith , 1998. Video OCR for Digital News Archives, IEEE Workshop on Content-Based Access of Image and Video Databases(CAIVD'98), pp. 52 – 60

[6] Du, Yingzi, Chang, Chein-I Thouin, Paul D., 2003. Automated system for text detection in individual video Images, Journal of Electronic Imaging, 12(3), 410 - 422.

[7] Crandall David, Sameer Antani, Rangachar Kasturi, 2003. Extraction of special effects caption text events from digital video IJDAR(5), No. 2-3, pp. 138-157

[8] Li, D. Doermann and O. Kia. Automatic Text Detection and Tracking in Digital Video. *IEEE Transactions on Image Processing*. Vol. 9, No. 1, pp. 147-156, Jan. 2000.

[9] Wolf Christian and Jean-Michel Jolion, 2004. 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. March 19th

[10] Yan Hao, Yi Zhang, Zengguang Hou, Min Tan, 2003. Automatic Text Detection In Video Frames Based on Bootstrap Artificial Neural Network and CED. The 11-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2003.

[11] Ye Qixiang, Qingming Huang, Wen Gao, Debin Zhao, 2005. Fast and robust text detection in images and video frames. Image Vision Computing 23(6): 565-576.

[12] Wu W., D. Chen and J. Yang, Integrating, 2005. Co-Training and Recognition for Text Detection, Proceedings of IEEE International Conference on Multimedia & Expo 2005 (ICME 2005), pp. 1166 - 1169.

[13] R. Lienhart and A.Wernicke. Localizing and segmenting text in images and videos. *IEEE*
*Trans. on Circuits and Systems for Video Technology*, 12(4):256–268, 2002.

[14] Clark P. and M. Mirmehdi, 2000. Finding Text Regions Using Localised Measures, Proceedings of the 11th British Machine Vision Conference.

[15] D. Chen, J-M. Odobez and J-P. Thiran, "A Localization/Verification Scheme for Finding
Text in Images and Videos Based on Contrast Independent Features and Machine Learning
Methods", to be published in Signal processing: Image Communication (SPIC).

[16] Marios Anthimopoulos, Basilios Gatos, Ioannis Pratikakis: Multiresolution text detection in video frames. VISAPP (2) 2007: 161-166

[17] Canny J., 1986. A computational approach to edge detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8, 679-698.

[18] Ojala, T., Pietik¨ainen, M., Harwood, D.: A comparative study of texture measures
with classification based on feature distributions. Pattern Recognition 29 (1996)

[19] Hongming Zhang Wen Gao Xilin Chen Debin Zhao Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on, 1806- 1811 vol. 3(2005)

[20] Y. K. Lim, S. H. Choi, and S.W. Lee, Text Extraction in MPEG Compressed Video for Content-based Indexing, Proc. of International Conference on Pattern Recognition, 2000 pp. 409-412.

[21] Ullas Gargi, David J. Crandall, Sameer Antani, Tarak Gandhi, Ryan Keener, Rangachar Kasturi: A System for Automatic Text Detection in Video. ICDAR 1999: 29-32

[22] Cross G & Jain A (1983) Markov random field texture models. IEEE Transactions on
Pattern Analysis and Machine Intelligence 5: 25–39.

[23] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.