Text detection in video frames

M. Anthimopoulos, B. Gatos, I. Pratikakis
Computational Intelligence Laboratory,
Institute of Informatics and Telecommunications,
National Center for Scientific Research “Demokritos”,
GR-153 10 Agia Paraskevi, Athens, Greece. http://iit.demokritos.gr/cil/ ,
{anthimop, bgat, ipratika}@iit.demokritos.gr

Abstract

In this paper we present the state of the art for detecting text in images and video frames and propose an edge-based algorithm for artificial text detection in video frames. First, an edge map is created using the Canny edge detector. Then, morphological filtering is used, based on geometrical constraints, in order to connect the vertical edges and discard false alarms. A connected component analysis is performed to the filtered edge map in order to determine a bounding box for every candidate text area. Finally, horizontal and vertical projections are calculated on the edge map of every box and a threshold is applied, refining the result and splitting text areas in text lines. The whole algorithm is applied in multiresolution fashion to ensure text detection with size variability. Experimental results prove that the method is highly effective and efficient for artificial text detection.

Keywords: Text detection, video frames, artificial text, edge-based, state of the art

1. Introduction

Nowadays the size of the available digital video content is increasing rapidly. This fact leads to an urgent need for fast and effective algorithms for information retrieval from multimedia content for applications in video indexing, editing or even video compression. Text in video and images proves to be a source of high-level semantics closely related to the concept of the video. Moreover, artificial text can provide us with even more powerful information for television captured video indexing since this kind of text is added in order to describe the content of the video or give additional information related to it.

The procedure of retrieving text from video is usually called “Video OCR” and consists of 3 basic stages: text detection, text segmentation and recognition. Text detection is a crucial step towards the completion of the recognition process. The aim of this paper is to give an effective and computationally efficient algorithm for the spatial detection of artificial text in video frames. The algorithm intends to produce one bounding box for every text line of the frame. Artificial text presents some features and follows some characteristics in order to be readable from humans, like high intensity vertical edge strokes, colour homogeneity, contrast between text and background, horizontal alignment, various geometrical constraints etc. The above features and constraints are usually used by the text detection systems for distinguishing text areas from non-text areas. On the other hand, there are many challenges that have to be faced like, text embedded in complex backgrounds, with unknown color, size, font or low resolution text.

The structure of the remaining of our paper is as follows: Section 2 presents a short state of the art of text detection, section 3 describes the proposed algorithm and its different stages, section 4 presents the evaluation method and the experimental results and section 5 provides the conclusion.

2. State of the art

Many researchers have proposed methods based on different architectures, feature sets, and classifiers in order to deal with text detection problem. These methods generally can be classified into two categories: Bottom-up methods and Top-down methods.

2.1 Bottom-up methods

Bottom-up methods segment images into “character” regions and group them into words. Due to the difficulty of developing efficient segmentation algorithms for text in complex background, the methods are not robust for detecting text in many camera-based images and videos. Due to their relatively simple implementation, Connected Component (CC) based methods are widely used. Nearly all CC-based methods have four processing stages:

(i) Preprocessing, such as color clustering and noise reduction
(ii) CC generation (split-merge algorithm, region-grow algorithm)
(iii) Filtering out non-text components
(iv) Component grouping.

Lienhart et al. [Lienhart (1995)] regard text regions as connected components with the same or similar color and size, and apply motion analysis to enhance the text extraction results for a video sequence. The input image is segmented using a split-and-merge algorithm. Finally, a geometric analysis, including the width, height, and aspect ratio, is used to filter out any non-text components. Sobottka et al. [Sobottka (1999)] use a region growing method in order to detect homogeneous regions. Beginning with a start region, pixels are merged if they belong to the same cluster. Then the regions are grouped to form text lines assuming that text lines consist of more than three regions having a small horizontal distance and a large vertical overlap to each other.

2.2 Top-down methods

Top-down methods firstly detect text regions in images and then split them in text lines. These methods are able to process more complex images than bottom-up approaches and they are also divided into two sub-categories: Heuristic methods and Machine learning methods.

Heuristic methods usually use heuristic filters in order to detect text. Malobabic et al.[Malobabic (2004)] and Xi et al. [Xi (2001)] et al. propose edge based methods for detecting text regions. An edge map is calculated followed by smoothing filters, morphological operations and geometrical constraints. However the use of Sobel operator cannot preserve successfully the contours of the characters. Zhong et al. [Zhong (2000)] use the DCT coefficients of compressed JPEG or MPEG files in order to distinguish the texture of textual regions from non-textual regions. Sato et al. [Sato (1998)] apply a 3x3 horizontal differential filter to the entire image with appropriate binary thresholding. If a bounding region which is detected by the horizontal differential filtering technique satisfies size, fill factor and horizontal-vertical aspect ratio constraints, it is selected for recognition as a text region. Du et al. [Du (2003)] 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. Crandall et al. [Crandall (2003)] use the DCT coefficients to detect text areas. Then connected component analysis is performed in them followed by an iterative greedy algorithm which refines the skew, position and size of the initial bounding boxes. The algorithm is designed to detect artificial text with special effects as well as scene text.

Machine learning methods use trained, machine learning techniques in order to detect text. Li et al. [Li (2000)] propose a method based on neural networks (NN) trained on wavelet features. The NN classifies the pixels of a sliding window of 16x16 pixels. In the detected area a connected component analysis is applied so the textbox’s center, width and height are computed.Skew is also estimated and corrected. This method works for artificial and scene text. Wolf et al. [Wolf (2004)] use an SVM trained on derivative and geometrical features. Yan et al. [Yan (2003)] use a Back Propagation Artificial Neural Network trained on Gabor edge features. Ye et al. [Ye (2005)] use SVM and wavelets. Wu et al. [Wu (2005)] propose a system of two co-trained SVM’s on edge and color features. Lienhart et al. [Lienhart (2002)] propose a method based on neural network classification using gradient features. Chen et al. [Chen (2001)] use several heuristics based on edges to detect text and then refine the results using a Bayesian Classifier trained on features based on geometry and projection analysis. Clark et al. [Clark (2000)] presents five statistical measures for training a NN. Chen et al. [Chen (2003)] use features like Greyscale spatial derivatives, distance maps, constant gradient variance and DCT coefficients fed to an SVM classifier.

3. Text detection algorithm

The proposed algorithm (Figure 1) 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 colors since every character presents strong edges, despite its font or color, in order to be readable. An example of artificial text in a video frame is given in Figure 2.

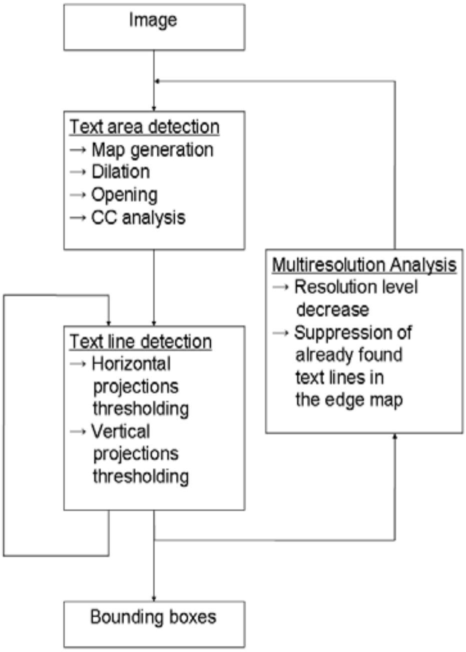


Image
Text area detection
→ Map generation
→ Dilation
→ Opening
→ CC analysis
Multiresolution Analysis
→ Resolution level decrease
→ Suppression of already found text lines in the edge map
Text line detection
→ Horizontal projections thresholding
→ Vertical projections thresholding
Bounding boxes

***Figure 1**. Flowchart of the proposed text detection algorithm.*

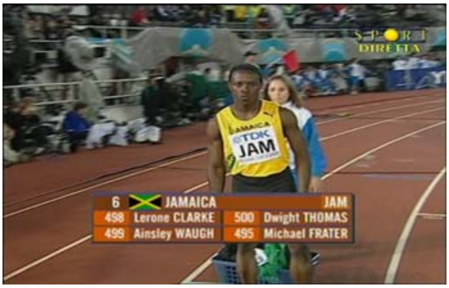
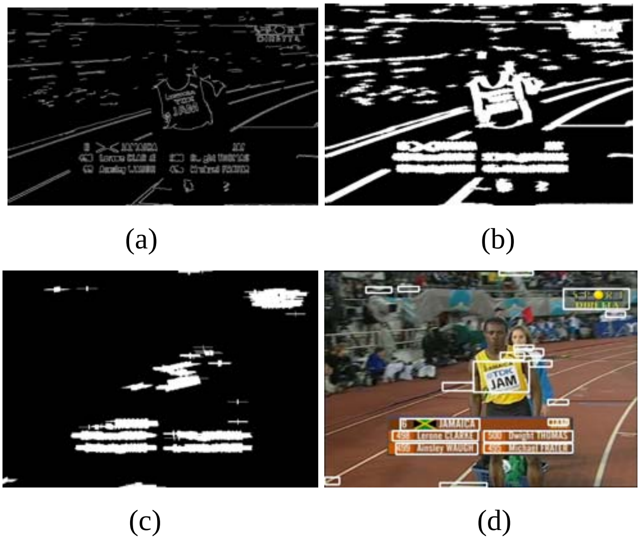

SPORT DIRETTA
JAMAICA
TDK
JAM
6 JAMAICA JAM
498 Lerone CLARKE 500 Dwight THOMAS
499 Ainsley WAUGH 495 Michael FRATER

***Figure 2**. Example of artificial text.*

3.1 Text area detection

As a first step of our methodology we produce the edge map of the video frame image using Canny [Canny (1986)] edge detector applied in greyscale images. Canny uses Sobel masks in order to find the edge magnitude of the image, in grayscale, and then uses non-Maxima suppression and hysteresis thresholding. With these two post-processing operations Canny edge detector manage to remove non-maxima pixels, preserving the connectivity of the contours. Ideally the created edge map is a binarized image with the pixels of contours set to one (white) and the background equal to zero (black) (Figure 3a).

After computing the Canny edge map, a dilation by an element 5x21 is performed to connect the character contours of every text line (Figure 3b). Experiments showed that a cross-shaped element has better results. Then a morphological opening is used, removing the noise and smoothing the shape of the candidate text areas (Figure 3c). The element used here is also cross-shaped with size 11x45. Every component created by the previous dilation with height less than 11 or width less than 45 is suppressed. This means that every edge which could not connect to a component larger than the element of the dilation will be lost. Unfortunately this operation may suppress the edges of text lines with height less than 12 pixels. However, this is not so devastating since characters of this size are either way not recognized in the final stage of the video OCR system. Finally a connected component analysis helps us to compute the initial bounding boxes of the candidate text areas. (Figure 3d).



(a) (b)

(c) (d)

Figure 3. *Text area detection. (a) Edge map, (b) Dilation, (c) Opening, (d) CC analysis, Initial bounding boxes.*

3.2 Text line detection

The previous stage has a high detection rate but relatively low precision. This means that most of the text lines are included in the initial text boxes although some text boxes may include more than one text line as well as noise. This noise usually comes from objects with high intensity edges that connect to the text lines during the dilation process. The low precision also comes from detected bounding boxes which do not contain text but objects with high vertical edge density. To increase the precision and reject 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 divided and some lines with noise are also discarded (Figure 4(a)). Besides, boxes which do not contain text are usually split in a number of boxes with very small height and discarded by a next stage due to geometrical constraints. A box is discarded if:

- Height is lower than a threshold (set to 12),
- Height is greater than a threshold (set to 48),
- Ratio width/ height is lower than a threshold (set to 1.5).

Then, a similar procedure with vertical projection follows (Figure 4(b)). This method breaks every text line in parts only if the distance between them is greater 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.

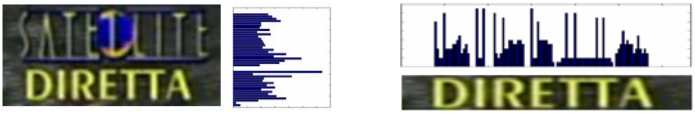


***Figure 4**. (a) Example of horizontal projection, (b) Example of vertical projection*

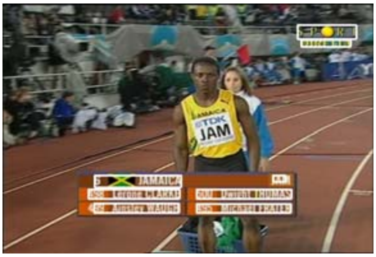
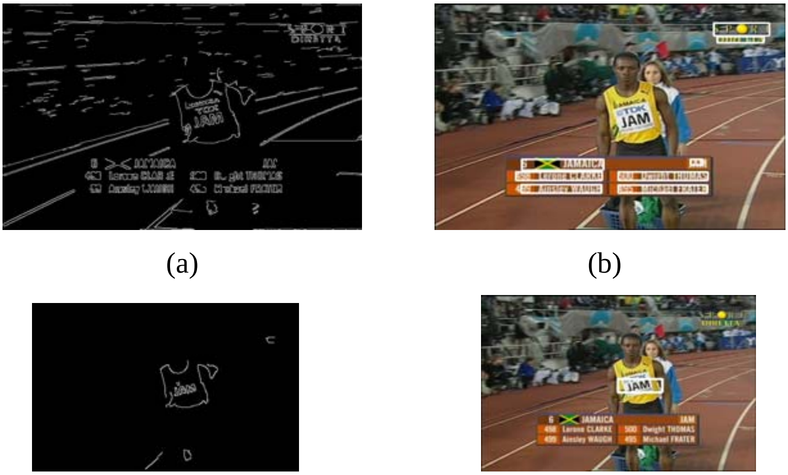


***Figure 5**. Bounding boxes after projection analysis*

The whole procedure is repeated three times in order to segment even the most complicated text areas and results to the final bounding boxes (Figure 5).

3.3 Multiresolution analysis

Using edge features in order to detect text gives to the method independence from text color and different fonts. However, this method clearly depends on the size of the characters. The size of the elements for the morphological operations and the geometrical constraints described above, give to the algorithm the ability to detect characters with height from 12 to 48 pixels. To overcome this problem we adopt a multiresolution approach. The algorithm described above is applied to the image in different resolutions and finally the results are fused to the initial resolution. This fusion 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 recognised characters in a resolution before the edge map is passed to the next resolution. For every resolution, except for the initial a blurring filter is applied so the edges of the background become weaker compared to the edges of the text which still remain strong. This filter is not applied to the first resolution because it would destroy the contrast of the small characters that already suffer the blurring caused by video compression. Taking into account that artificial text in videos usually does not contain very large characters and from the experience of related experiments we chose to use two resolutions for this approach: the initial, and the one with a scale factor of 0.6. In this way the system can detect characters with height up to 80 pixels which was considered to be satisfying. (Figure 6)



(a) (b)

(c) (d)

***Figure 6**. Multiresolution analysis. (a) Fine resolution, (b) Result at fine resolution, (c) Coarse resolution, (d) Result at coarse resolution.*

4. Evaluation method and experimental results

Designing evaluation methods for text detection is an aspect that has not be studied extensively. Very few related works have been published, moreover these works propose evaluation strategies with very complicated implementations or demand great effort for the generation of the ground truth [Hua (2003)]. Many of the researchers use their own evaluation tool to test the success of their algorithm. This fact leads to the inability to compare the performance of the different algorithms which indubitably consists a barrier to the evolution of the area. In this work we used as evaluation indicators the recall and precision rates on a pixel base. For the computation of these rates we need to calculate the number of the pixels for the ground truth bounding boxes, for the bounding boxes of the detection method and for their intersection. However this method proved to have several drawbacks which have to be faced.

The first is that there is not an optimal way to draw the ground truth bounding boxes. This means that two boxes may be accurate enough for bounding a text line although they may not include exactly the same pixels. In other words, the result of the detection method may be correct although the evaluation method gives a percentage less than 100%. To overcome this problem one can segment the text pixels from the background pixels and then demand the presence of text pixels in the output bounding box. However, this would make the detection evaluation depend on the performance of text segmentation which is something surely not desirable. In this work, we follow a more simple strategy to solve this problem. The ground truth bounding boxes are drawn in a way that the margins between the text pixels and the edge of the box are equal for all text lines. Another drawback is the fact that this method actually measures the percentage of detected pixels. However the goal of the detection algorithm is not to detect maximal amount of pixels but the maximal number of characters. Unfortunately, the number of characters in a box cannot be defined by the algorithm but it can be approximated by the ratio width/height of the bounding box, if we assume that this ratio is invariable for every character and the spaces between different words in a text line is proportional to its height. In this way, every pixel counts for $\frac{1}{h^2}$ when calculating the recall and precision rates, where h is the height of the bounding box in which the pixel belongs (1), (2).

$$\text{Precision} = \frac{\sum_{i=1}^{M} \frac{EDG_i}{hd_i^{\,2}}}{\sum_{i=1}^{M} \frac{ED_i}{hd_i^{\,2}}} \quad \textbf{(1)} \qquad \text{Recall} = \frac{\sum_{i=1}^{N} \frac{EGD_i}{hg_i^{\,2}}}{\sum_{i=1}^{N} \frac{EG_i}{hg_i^{\,2}}} \quad \textbf{(2)}$$

Where hg_i is the height of the ith ground truth bounding box, EG_i is its number of pixels, EGD_i is the number of pixels of the intersection that belong to ith ground truth bounding box, hd_i is the height of the ith detection bounding box, ED_i is its number of pixels, EDG_i is the number of pixels of the intersection that belong to ith detection bounding box, N is the number of ground truth bounding boxes and M is the number of detected bounding boxes. As an overall measure we use the weighted harmonic mean of precision and recall also referred to as the F-measure (3).

$$F = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad \textbf{(3)}$$

By using this evaluation method we try to approximate the rates of character detection through pixel detection rates. For testing the algorithm’s performance, 3 sets of video frames (720x480) have been used, captured from TRECVID 2005 and 2006 (http://www-nlpir.nist.gov/projects/trecvid/). For the results of Table 1, a Pentium 4, 3.2Ghz processor has been used.

***Table 1**. Results of the algorithm.*

	Set1	Set2	Set3
Number of images	95	83	61
Number of ground truth boxes	384	272	241
Recall	90.41%	82%	90.58%
Precision	83.57%	89.02%	91.66%
F-measure	87.1%	85.36%	91.17%
Time (secs)	40	33	22

The results of set2 proved to be worse than the others. This is probably because this set contains images with very large fonts and also some scene text. Experimental results showed that very large fonts cannot be detected using only the edge information of the image because the edge texture of a large font has many similarities with the texture of background objects. Set3 contains artificial text with small fonts and Set1 contains text in many different sizes as well as some scene text.

5. Conclusion

In this paper we present the state of the art for detecting text in images and video frames and propose an edge-based algorithm for artificial text detection in video frames. The proposed methodology exploits the fact that text lines produce strong vertical edges horizontally aligned and follow specific shape restrictions. Although the algorithm is designed to detect horizontal artificial text, scene text can also be detected in some cases. Experimental results advocate very good performance in a variety of different video frames. The method is vulnerable in very complex backgrounds. In our future work, we plan to exploit the color homogeneity of text.

References

Canny J., 1986. *A computational approach to edge detection*, PAMI, 8, 679-698.

Chen D. , H. Bourlard, 2001. and J. -P. Thiran, *Text Identification in Complex Background using SVM*, IEEE CVPR Vol. 2, pp. 621-626.

Chen Datong, Kim Shearer and Herve Bourlard, 2003. *Extraction of special effects caption text events from digital video*, IJDAR(5), No. 2-3, pp. 138-157

Clark P. and M. Mirmehdi, 2000. *Finding Text Regions Using Localised Measures*, 11th British Machine Vision Conference.

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

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

Hua Xian-Sheng, Liu Wenyin, HongJiang Zhang, 2004. *An automatic performance evaluation protocol for video text detection algorithms*. IEEE Trans. Circuits Syst. Video Techn. 14(4), 498-507.

Li Huiping , David Doermann, 2000. *A Closed-Loop Training System for Video Text Detection*, Cognitive and Neural Models for Word Recognition and Document Processing, World Scientific Press.

Lienhart Rainer and Frank Stuber, 1995. *Automatic text recognition in digital videos*, Technical Report, University of Mannheim.

Malobabic J, O'Connor N, Murphy N, and Marlow S., 2004. *Automatic Detection and Extraction of Artificial Text in Video*, WIAMIS 2004, Lisbon, Portugal.

Rainer Lienhart and Axel Wernicke, 2002.*Localizing and Segmenting Text in Images and Videos*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, NO. 4

Sato T. , Kanade T. , E. Hughes, and M. Smith , 1998. *Video OCR for Digital News Archives*, IEEE CAIVD'98, pp. 52 - 60.

Sobottka K. and Bunke H., 1999. *Identification of Text on Colored Book and Journal Covers*, ICDAR, Bangalore, India, pp. 57-62.

Wolf Christian and Jean-Michel Jolion, 2004. *Model based text detection in images and videos: a learning approach*. Technical Report LIRIS-RR-2004-13 INSA de Lyon, France. March 19th.

Wu W., D. Chen and J. Yang, Integrating, 2005. *Co-Training and Recognition for Text Detection*, IEEE ICME 2005, pp. 1166 - 1169.

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

Yan Hao, Yi Zhang, Zengguang Hou, Min Tan, 2003. *Automatic Text Detection In Video Frames Based on Bootstrap Artificial Neural Network and CED*. WSCG 2003

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.

Zhong Yu, HongJiang Zhang, Anil K. Jain, 2000. *Automatic Caption Localization in Compressed Video*, IEEE Trans. PAMI, 22(4): 385-392.