Video OCR for Sport Video Annotation and Retrieval

Datong Chen, Kim Shearer and Hervé Bourlard, *Fellow, IEEE*
Dalle Molle Institute for Perceptual Artificial Intelligence
Rue du Simplon 4
CH-1920 Martigny
Switzerland
Email: {chen, bourlard}@idiap.ch
http://www.idiap.ch/~chen

***Abstract*--This paper presents a video OCR system that automatically extracts closed captions from video frames as keywords (or as we called "cues") for building annotations of sport videos. In this system, text regions that contain closed captions are first identified using support vector machines (SVMs). We then enhance the identified text regions by using two groups of asymmetric filters and recognize them using commercial OCR software package. The resulting captions are recorded as cues in XML format for video annotation and retrieval task.**

***Index terms*—video OCR, video retrieval, video annotation, text identification, text recognition**

I. INTRODUCTION

Text embedded in sport video usually provides brief and important information about the content, such as the name of a game, the name of a player or a team, the score, location and date (or time) etc. This kind of embedded text, referred to as closed caption, is a powerful keyword resource in building video annotation and retrieval system. Due to the huge amount of data carried by video, closed caption extraction by hand is a very expensive work in terms of time and human power. Therefore, automatic extraction of closed captions has gained research importance recently.

However, text extraction is a difficult task because background, color, size of text strings may vary in a same image. Many papers [2][7] show that available binarization methods, including global and adaptive thresholding (which has been well used in extracting characters printed on clean papers) do not work well for typical video frames. Furthermore, video digitalization and compression may seriously blur closed captions in chrome space. In comparing with the closed captions in other types of video, for example news video, the closed captions in sports video are usually displayed in table form with multiple rows and columns and last a shorter period (less than half second), which makes them even harder to be extracted.

In this paper, we present an automatic text extraction system for sport video, which consists of three stages: text identification, text recognition, and cue producing for annotation. In the first stage, we identify single text line in video frames using SVMs. The detail of this text identification algorithm is described in Section 2. In Section 3, we will introduce text enhancement using two groups of asymmetric filters and text recognition. The extracted captions are then used as cues in XML format for video annotation and retrieval task, which will be discussed in Section 4.

II. TEXT IDENTIFICATION

Previous work on text identification in image or video can be briefly classified into region-based, texture-based and edge-based methods. Region-based methods detect character as the monochrome regions satisfying certain heuristic constraints. Since the grayscale or color of the text pixels in input image are often not uniform, image segmentation [1][4][5] or color clustering [10] preprocess has to be performed to reduce the total number of colors or grayscales in image. The performance of region-based methods greatly rely on the monochrome assumption of text characters and are therefore not robust to complex background and compressed video. Texture-based methods employ texture features to decide whether a pixel or block of pixels belongs to text or not. Wu et al. [2][3] proposed a K-means based algorithm to identify text pixels on the basis of nine second order derivatives of Gaussians at three scales. Li et al. [8] used a neural network to extract text blocks in Haar wavelet decomposition feature space. Zhong and Jain [4][9] presented a text region identification approach to combine spatial variance (texture feature) and connected component (regions) analysis together. Texture-based methods are able to detect text in complex background but are very time consuming [8] and cannot always perform accurate text location [4]. Edge-based method detects text in video by finding vertical edges. In [11], vertical edges are detected by a 3×3 filter and are connected into text clusters by using a smoothing filter. The same as the region and texture-based methods, the resulting text clusters are then selected by using geometric heuristic constraints. This algorithm performs fast text detection but reports many false alarms.

In this section, we propose a text identification algorithm using SVMs. Two characteristics of closed captions in video have been explored. First, a visible character always forms some edges against its background. Second, a text string has a special kind of texture pattern, a rectangle shape and horizontal alignment. On the basis of these two characteristics, our text identification algorithm consists of three stages: candidate text region detection, text line location and SVM-based identification. We first quickly detect candidate text regions with high location rate and reasonable false alarms. These candidate text regions are then segmented into text lines on the basis of baseline location and heuristic constraints. Further, we identify the resulting text lines using SVMs to achieve a lower false alarm rate.

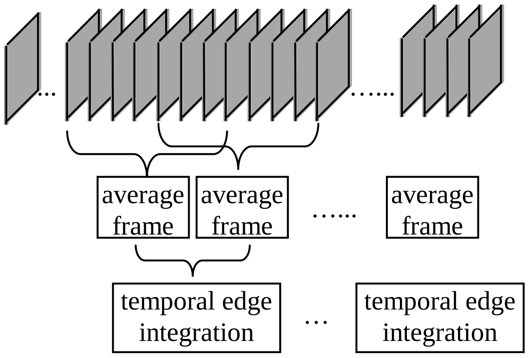

Figure 1 Temporal information processing

A. Candidate text region detection

In this algorithm, texture pattern of text string is simply regarded as a group of short vertical and horizontal edges mixed together. We integrate this texture information and temporal information for text detection in the follow process:

1. Multiple intensity frame integration is performed by computing average image of consecutive frames; (see Figure 1)

2. Detect edges in vertical and horizontal orientation respectively with Canny operators [12];

3. Integrate temporal edge information by keeping only edge points that appear in consecutive two average images; (see Figure 1)

4. Dilate vertical and horizontal edges respectively into clusters. Different dilation operators are used so that the vertical edges are connected in horizontal direction while horizontal edges are connected in vertical direction. The dilation operators are designed to have rectangle shapes: vertical operator 5×1, horizontal operator 3×6, which are shown in Figure 2.

5. Integrate vertical and horizontal edge clusters by keeping the pixels that appear in both vertical and horizontal dilated edge images.

Figure 3(a-d), illustrates the clusters resulting from this detection process.

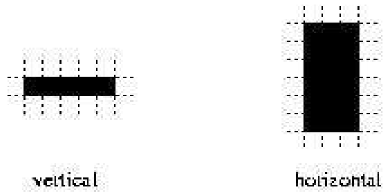

Figure 2 Vertical and horizontal edge dilation operators

B. Text line location

Text line location will extract single text lines from detected clusters by locating horizontal baselines. We first roughly segment a candidate region at those lines that the value of derivation in horizontal projection is bigger than a fixed threshold. Because the cluster may consist of no text line or more than one text line, one threshold is not optimal for all the cases. We therefore further refine each segmented region by an iterative procedure consisting of the following heuristic process:

1. Fill-factor check. If the fill-factor (the density of the region in its smallest rectangle boundary) of the given region is less than 70%, this region is going to be refined in step 2.

2. Segmentation refinement by first finding a line x using Otsu's thresholding method [14] on Y-axis projection of the region to be refined. If the length of line x is less than 65% of the longest line in this region, we segment this region at line x and go back to step 1.

3. Baseline refinement for locating accurately the top and bottom baselines of a text string. We move the top and bottom baselines to the center until the fill-factor is equal or greater than 70%.

The typical heuristic character of a text string is then employed to select the candidate text lines. In our experiments the candidate text line should satisfy the constraints that: the size of region is between 75 to 9000; the horizontal-vertical aspect ratio is more than 1.2; the height of the region is between 8 to 35. Figure 3(e,f) shows the baseline location and the candidate text lines.

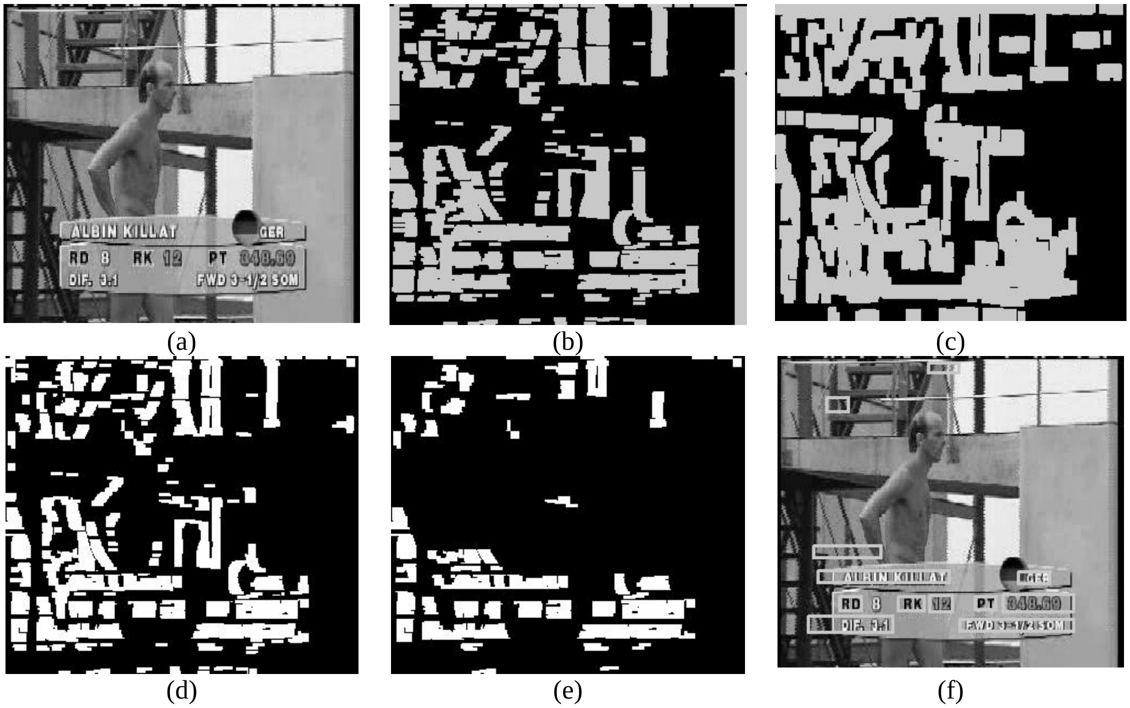

Figure 3 Candidate text region detection and text line location: (a) original image, (b) vertical edge dilation, (c) horizontal edge dilation, (d) integration of vertical and horizontal edge dilation, (e) updated regions after baseline location, (f) candidate text lines

C. SVM-Based Text Identification

We further identify the resulting candidate text lines by using SVMs to achieve a lower false alarm rate. SVMs have been successfully applied to the problem of binary and multi-class classification because of its strong generalization capability. The detailed theoretical presentation of the SVMs can be found in [13][16]. The basic idea to use SVMs for pattern recognition consists of two steps:

1. Map the input feature into a high dimensional feature space via a non-linear mapping.

2. Construct an optimal hyper-plane for separating input samples in the high dimensional feature space.

Figure 4 Normalized candidate text lines

The candidate text lines, which may have varying resolutions, are first normalized to rectangles with 16 pixels in height by using bilinear interpolation (8 pixels between the baselines, 8 pixels for top and bottom boundary). The width of text line keeps the same proportional as the height. Some examples of normalized text lines are shown in Figure 4. Since the text may have varying gray-scales in video frames, we therefore choose gray-scale independent feature: distance map (as explained below) of 16×16 slide window as input feature of SVMs.

The distance map [13] *DM(X)* of window *X* is denote as the set of all the associated distance values $v(x, y)$ in the window *X* with respect to a distance function *dis* according to

$$\forall (x, y) \in X : v(x, y) \overset{def}{=} \min_{(x_i, y_i) \in B} dis[(x, y), (x_i, y_i)],$$

where, *B* is a set of points, $B \subseteq X$. We compute the point set *B* by extracting strong edges in window *X*. The distance function used here is Euclidean.

We use Radial basis function (RBF) kernel, where the kernel bandwidth σ determined through cross-validation. The details of the requirement of the kernel and construction of the hyper-plane via a dual optimization process can be found in [16].

The SVMs are trained with a database consists of 6000 samples labeled as text or non-text (false alarms resulting from text line location) with the software package called SVMTorch [17].

The support vectors resulting from SVM are used to estimate the confidence $C_w(X)$ of the block of pixels in the 16×16 test window X. In the identification process, we slide the test window every four pixels from left to right in each normalized text line and compute the confidence of each window. The confidence $Conf(R)$ of a text line R was defined as:

$$Conf(R) = \sum_{X \subseteq R} C_w(X) \cdot \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(\frac{d_X{}^2}{2\sigma_0{}^2}\right),$$

where d_X is the distance between the center of window X and the center of the text line R. Here, we use $\sigma_0 = 10$. A candidate text line R is identified as a text string if $Conf(R) \geq 0$.

Figure 5 Valid baseline ranges: the shading parts indicate the valid baseline range

Experiments were carried out on a database consisting of totally 18,000 frames from different sports videos, (with resolution of 352×288). We employ identification rate (IR) and false alarms to evaluate the performance of the text identification. A text string is considered to be correctly identified if and only if the located baselines are in the valid ranges, which is labeled by human visual inspection as shown in Figure 5. The false alarms are reported by both false region alarm and false pixel alarm. The false region alarm rate (FRR) is measured by the percentage of the number of false alarm regions in all the identified regions. The false pixel alarm rate (FPR) is defined as the total area of false alarm regions as a percentage of the whole area of the video frames.

Table 1 compares the performances and running time costs of the proposed algorithm with typical region-based [5], texture-based [2] and edge-based [11] methods. The proposed algorithm is a good tradeoff between high identification rate and low false alarm rates. The mounts of CPU required by this algorithm is higher than region and edge based method but much lower than texture-based method. Most of the time is spent in SVM-based identification phase. Figure 6 shows some text identification results including correct identifications and false alarms.

Table 1 Performances and running costs

X-based method	IR	FRR	FPR	Sec/(average image)
Region	89.6	59.2%	5.8%	1.15
Texture	99.1%	11.5	3.3%	11.27
Edge	92.6%	24.1%	12.3%	0.52
Proposed	98.7%	1.7%	0.38%	2.76

III. TEXT RECOGNITION VIA ENHANCEMENT

There are both the pixels of text and the pixels of background inside the identified text line. An enhancement procedure is necessary to enhance the contrast between text and background so that the text pixels can be segmented easily from the background by using binarization algorithm. In [2], Wu simply smoothes the detected text regions in order to lead to better binarization results. Smoothing eliminates noise but can not filter out background. This method, therefore, can not reliably extract text in complex background. In [5], the authors use multi-frame integration to enhance captions in video. The influence of the background is reduced on the basis of motion clues. The multi-frame method can efficiently enhance the text in video frames with rather different background movements, for example static text with fast moving background, but is not able to clean the background with same or slightly different movements. Sato [6] enhances the text on the basis of its sub-structure: line element, by using filters with four orientations: vertical, horizontal, left diagonal and right diagonal in the located text block. However, because real scales (thickness of the line elements) are unknown, it is not possible to design a filter that can enhance the line elements with widely varying widths.

Figure 6 Identified text lines and false alarms in images or video frames

We have integrated multiple frames text enhancement method proposed in [5] in text identification process (see section 2.a). In this section, we presents a method for enhancing the text in video via locating the orientation and scale of character strokes. We first detect strong edges in text lines as candidate edges of character strokes, and then search for the orientations and scales of these edge points using two groups of asymmetric filters as explained below. We enhance the contrast of these candidates character strokes at each edge point using filter with corresponding orientation and scale.

We introduce two groups of Gabor-based asymmetric filters: edge-form filters $E_{\lambda,\theta}(x,y)$ and stripe-form filters $S_{\lambda,\theta}(x,y)$ to obtain the precise scale information of the located edges in an image, which are defined as:

$$\begin{aligned} E_{\lambda,\theta}(x,y) &= \exp\left(-\frac{\hat{x}^2+\gamma^2\hat{y}^2}{2\sigma^2}\right)\cos\left(2\pi\frac{\hat{x}}{\lambda}+\frac{\pi}{2}\right) \\ \hat{x} &= x\cos\theta + y\sin\theta \\ \hat{y} &= -x\sin\theta + y\cos\theta \end{aligned}$$

and

$$\begin{aligned} S_{\lambda,\theta}(x,y) &= \exp\left(-\frac{\hat{x}^2+\gamma^2\hat{y}^2}{2\sigma^2}\right)\cos\left(2\pi\frac{\hat{x}}{\lambda}\right) \\ \hat{x} &= x\cos\theta + y\sin\theta - \frac{\lambda}{4} \\ \hat{y} &= -x\sin\theta + y\cos\theta \end{aligned}$$

where the arguments x and y represent the pixel coordinators, parameter θ specifies the orientation of the filter, parameter γ determines the spatial aspect ratio, and $\frac{1}{\lambda}$ is called the spatial frequency.

The rational behind this is that those asymmetric filters can give strong response on candidate edge points in optimal orientation and scale. The edge-form and stripe-form filters keep most of the properties of the Gabor filters except the specified translation on the position and the phase offset. Figure 7 shows the pattern of the edge-form filters (Fig. 7a) and stripe-form filters (Fig. 7bc) in 8 orientations with $\gamma = 0.92$.

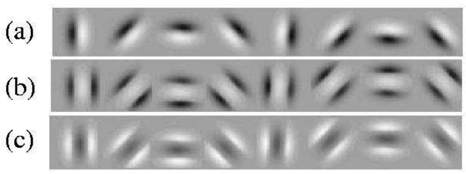

Figure 7 Asymmetric filters in 8 orietations: (a) edge-form filters, (b,c) stripe-form filters

These two groups of filters are particularly useful in determining the orientation and scale of character strokes. We then enhance the image with the resulting optimal orientation and scale at each candidate character stroke edge point. The detail description of this algorithm is presented in [18].

Figure 8 (a) original image, (b) binarization result of the image, (c) enhanced image using proposed method, (d) binarization result of the enhanced image

Otsu's thresholding method [14] is then employed to binarize the enhanced image so that it can be used as an input of conventional OCR system. Figure 8 shows the results of binarization of a text region in original image and enhanced image.

Experiments are based on extracted text lines involving 27054 characters (closed captions). The proposed text enhancement method yields recognition rate of 82.6%, while recognition without text enhancement yields the recognition rate of 36.1%.

IV. Cue Producing for Annotation

Extracted closed caption is recorded as a cue in XML format for producing video annotations. The closed caption cue is generated automatically involving text information (the content of text string) and its confidence (resulting from the character recognition algorithm), time code, video tape number, location (resulting from the text location algorithm). An example of a closed caption cue in XML format is:

```
<VisualCue featuretype="closed_caption" name= "text">
  <start>        01:26:15:12       </start>
  <end>          01:26:15:16       </end>
  <tape>         tape number       </tape>
  <confidence>   0.7               </confidence>
  <location>     left top right bottom      </location>
</VisualCue>
```

"VisualCue" indicates that "closed_caption" is a type of cue extracting from visual information. Video indexing and retrieval system will combine multiple types of visual cues, such as the closed caption, color, texture, motion, and audio cues, for example the speech, in a reasoning engine. The advantage of using XML format is that annotation and retrieval system can easily access the yielded annotations through the internet.

V. Discussion and Conclusion

In this paper, we have presented an automatic text detection and recognition system for sport video annotation and retrieval. In this system, closed captions are first identified using SVMs and are enhanced based on asymmetric filters. The system presented achieves high text identification rate as well as low false alarm rates.

The character recognition performance was also improved using proposed enhancement method.

We do not use color although many systems also make use of color information in detecting text in color image [5][10]. The main reason is that the start point of our system, the edge evidence, is mostly coming from intensity in compressed image.

The proposed system yields character recognition error when one character touches another one in its neighborhood. The recognition performance can be improved in the future by developing character classification algorithm that robust to touched characters.

VI. ACKNOWLEDGEMENTS

This work has been performed within the framework of the "Automatic Segmentation and Semantic Annotation of Sports Videos (ASSAVID)" project granted by the European IST Programme.

VII. REFERENCES

[1] J. Ohya, A. Shio, and S. Aksmatsu, "Recognition characters in scene images. IEEE Trans. Pattern Analysis and Machine Intelligence", 16(2):214--220, 1994.

[2] V. Wu, R. Manmatha, and E. M. Riseman, "Finding text in images", In Proc. ACM Int. Conf. Digital Libraries, 1997.

[3] V. Wu, R. Manmatha, and E. M. Riseman, "Textfinder: An automatic system to detect and recognize text in images", IEEE Trans. on PAMI, 20(11):1224--1229, 1999.

[4] Y. Zhong, K. Karu, and A. K. Jain, "Locating text in complex color images", Pattern Recognition, 28(10):1523--1536, 1995.

[5] R. Lienhart, "Automatic text recognition in digital videos", In Proc. SPIE, Image and Video Processing IV, January 1996.

[6] T. Sato, T. Kanade, E. K. Hughes, M. A. Smith, and S. Satoh, "Video OCR: indexing digital news libraries by recognition of superimposed caption", In ACM Multimedia System Special Issue on Video Libraries, Feb. 1998.

[7] H. Li and D. Doermann, "Text enhancement in digital video using multiple frame integration", ACM Multimedia 1999.

[8] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video", Maryland Univ. LAMP Tech. Report 028, 1998.

[9] K. Jain and B. Yu, "Automatic text localisation in images and video frames", Pattern Recognition, 31(12):2055--2076, 1998.

[10] K. Sobottka, H. Bunke, H. Kronenberg, "Identification of text on colored book and journal covers", ICDAR, pp: 57-63, 1999.

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

[12] J. F. Canny, "A computational approach to edge detection", IEEE Tran. On PAMI-8, pp679-698,1986.

[13] J. Toriwaki and S. Yokoi, "Distance transformations and skeletons of digitized pictures with applications", in L. N. Kanal and A. Rosenfeld, editors, Progress in pattern recognition, North-Holland, Amsterdam, 1981.

[14] N. Ostu, "A thresolding selection method from gray-level histogram", IEEE SMC-8, pp62-66,1978.

[15] C. Burgess, "A tutorial on support vector machines for pattern recognition", Data mining and knowledge discovery, 1998.

[16] V. Vapnik, "Statistical learning theory", John Wiley & Sons, 1998.

[17] R. Collobert, and S. Bengio, SVMTorch: Support Vector Machines for Large-Scale Regression Problems , in Journal of Machine Learning Research, 2001.

[18] D. Chen, K. Shearer, and H. Bourlard, "Text enhancement with asymmetric filter for video OCR", in Proc. of the 11th Int. Conf. Image Analysis and Processing, 2001.