A FAST METHOD FOR ANIMATED TV LOGO DETECTION

Ersin Esen[1, 2], Medeni Soysal[1, 2], Tuğrul K. Ateş[1, 2], Ahmet Saracoğlu[1, 2] and A. Aydın Alatan[2]

[1]TÜBİTAK Space Technologies Research Institute
[2]Department of Electrical and Electronics Engineering, M.E.T.U.
Balgat 06531 Ankara, Turkey

ABSTRACT

As a recent trend some TV stations prefer to use animated logos, therefore the detection of the presence of an animated TV logo emerges as a new requirement for certain applications. In this paper we present a novel method for the detection of animated television logos in real-time. The main idea is to handle all frames of the animated logo in a unified manner. For this purpose a unified logo boundary representation is utilized. In the training stage, the boundaries of the animated logo from each frame are placed in a single set. During detection, a voting-based decision scheme is performed in order to determine the presence of the trained logo. Furthermore robustness of the method is improved by incorporating negative clues regarding the existence of the animated logo obtained from the region of interest. Aforementioned clues are unified in order to reach a final decision by using effective combination rules. Finally, time windowing is used for eliminating false positives with short durations. The proposed method is examined through typical broadcast data and promising results are obtained.

1. INTRODUCTION

Television (TV) broadcast stations use logos mainly as a visual aid to express their ownership of the aired media. These ownership indicating logos often differ in accordance with the type of the broadcasted material (program, commercial etc.). In addition, due to legal regulations in many countries, additional logos indicating the audience suitability rating of programs, shows and films are becoming widespread. These logos are displayed for a significant amount of time at the beginning of each program to inform the audience about its rating decided by experts according to some standards. Another usage of logos is for indicating the sponsor firms; sponsored programs, such as live games, involve the intermittent display of an overlaid logo of the sponsor.

Logo detection has its roots in document analysis and data mining areas [1],[2]. In the document domain, the logos are used to infer some important properties, such as source of the document. However, utilization of the logos in TV broadcast resulted in a variety of reasons for their detection, such as video content understanding. Automatic classification of broadcast type can be easily performed utilizing logos [3]. However, in some situations, the user wants to remove the traces of the station's ownership from the video recorded from the TV. Such an aim is also possible with the techniques given in literature [4] by the prerequisite that the logo is automatically detectable. Another client-side application of logo detection is within personal video recorders (PVR). Commercial breaks that the most of common audience does not like can be removed from scheduled recordings by the exploitation of the fact that most TV stations change their logos, while switching between programs and commercials. In addition to this, for online viewers a zap-back alert can be set for a station which gave a commercial break. In this way, a viewer might zap through the other stations during the commercials and then be precisely alerted to return to the original station as soon as the commercial break ends.

Another category of usage for logo detection in TV broadcast is publicity companies. These companies conduct the analysis of multiple TV stations' broadcasts in order to prepare reports for their clients about the durations and frequency of their advertisements and/or logos. Broadcast companies themselves also make use of logo detection in their own broadcast. Utilization of logo detection enables them to easily assess the quality of the broadcast at the receiver side [5]. Panel monitor producers also depend on logo detection in a fundamental problem [6]. Due to the long-time static scenes in the display, a ghostly appearance of these static pixels occurs even after the display is turned off [6]. This situation is called as the 'burn-in' problem and might cause permanent damage in the display. By the help of the detection of logo pixels, this damage can be prevented by regular alteration of color values. Finally, government institutions that are responsible for surveillance of activities of TV broadcast stations, take advantage of logo detection for the enforcement of legal regulations [7]. These regulations include ones that are based on the length and frequency of commercial breaks. These parameters can easily be determined automatically, provided that a robust, efficient and generic logo detection algorithm is at disposal.

As already mentioned, logo detection has a great variety of applications in TV domain. The same variety can be observed in types of logo instances. Logos in TV domain have two dimensions of variance, namely *transparency* and *motion*. These two dimensions lead to a natural classification of TV logos into four categories, as *opaque-static*, *translucent-static*, *opaque-animated* and *translucent-animated*. A typical logo may consist of any combination of these four types.

There are logo detection algorithms in the literature that successfully handle opaque-static logos [5],[8],[9]. These logos have constant color regions overlaid on the broadcast video and always retain their color information. Therefore their locations can be detected by simple variance calculations and template matching. Opaque logos are becoming rare due to the aesthetic concerns, and being replaced with mildly translucent logos.

The appearance of the translucent logos varies depending on the background content of the video frame. Therefore, detection and tracking methods for the opaque logos which rely on color consistency becomes inapplicable for the translucent logos. Although there are only a few recent attempts to solve this problem, a recent work on detection of translucent logos mainly focuses on determination of existence or absence of predetermined contours of the logo. Wang *et al.* [5] proposed template matching over generalized gradients which is the spatio-temporal gradient over multiple video frames. The training part of this method relies heavily on variation of the background content and needs long durations for synchronization to achieve acceptable detection rates. In [5], three seconds of temporal accumulation is proposed for synchronization. On the other hand, Santos *et al.* [9] proposed template matching over spatial gradients of temporally accumulated frames. This method achieves detection five seconds after the appearance of translucent logos. Apart from these two methods, there are also some stochastic methods that detect translucent logos without discriminating opaque and translucent logos, as in Ekin *et al.* [6]. In their method, the logo pixels are modeled as outliers and detected via stochastic measures. Furthermore, in [10], a multi-resolution approach based on convolutional neural network is used for the detection of transparent logos. As a consequence of the intricacies in the training step, the method does not reach high detection rates.

As a result of the increasing graphic hardware processing capabilities, a new differentiation opportunity, namely, *animated logos* are appreciated and utilized by TV stations. Animated logos are also becoming popular to attract attention to special moments, such as replays or flash news. In the literature, however, animated logos do not seem to attract enough attention. Other than the stochastic method in [6] that treats all kind of logos similarly, no significant effort has been put on the subject. Animated logos are not normative in length and their type of motion, which obviously increases the level of challenge for this problem. Even in a single broadcast station, the same logo sequence might appear at random times and with random speeds.

The prominent mathematical model for composition of video and logo [11] can be slightly modified in the presence of animated logos, as follows:

$$\hat{I} = \alpha(t,x,y)L(t,x,y) + \left(1-\alpha(t,x,y)\right)I(t,x,y) \qquad (1)$$

In addition to I representing the pixels of the frame, L component representing the logo also changes w.r.t. time. In opaque-animated logos, change in L is independent of the background video since α component remains constant and is equal to one. Generally, this logo is a frequently repeating pattern overlaid on the video. On the other hand, in translucent-animated logos embedded into broadcast, the repeating multi-frame pattern also changes depending on the background video.

In this paper, we present a method for real-time detection of animated TV logos in general. The proposed method is simple and fast in comparison with treating each frame of the animated logo as a singular logo. It also utilizes a robust multi-frame contour representation. The contours extracted from the multi-frame logo are matched by the contours of a test video in a scheme, which is mainly based on Generalized Hough Transform [12]. Utilizing Hough-based voting system makes the algorithm much faster than template matching based methods. This result is due to the fact that the matching scheme complexity is unaffected by increasing/decreasing period of logo sequence. Finally, the results of the main algorithm are refined by majority voting that helps eliminating unstable decisions.

2. PROPOSED ANIMATED LOGO ALGORITHM

The proposed method is designed to operate on video streams and the detection is performed on each frame by using the unified boundary information of a multi-frame logo sequence. This information is obtained from frame based boundaries extracted beforehand. The boundaries of all frames are searched inside the video stream in a Hough Transform based voting scheme using this unified information.

Detection of both opaque and translucent logos requires the use of boundaries, since they are the most robust features for translucent logos; hence, reliable algorithms are required for this crucial step. In these algorithms, logo contours should remain stable, even if the logo is affected from the background video.

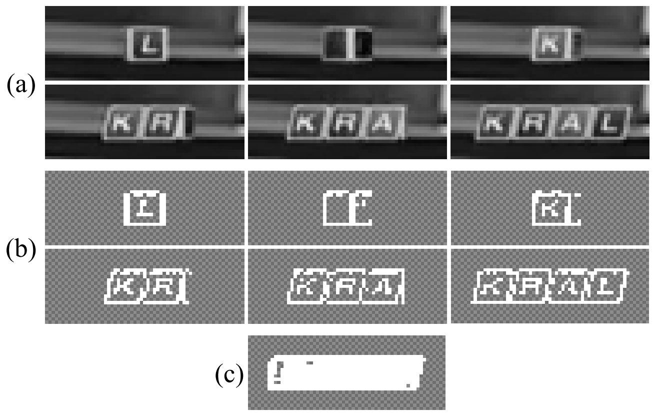

Fig. 1. Training steps for some frames from Kral TV logo. (a) Original logo sequence, (b) extracted boundaries and (c) unification of masks, i.e. B in (2).

2.1. Training of Animated Logos

Training step starts with the frame level boundary extraction from a selected video segment. For an animated logo sequence of length N frames, N binary boundary masks (B_1 ... B_N) are obtained. Following the extraction of these N masks, a unification process is applied during which fusion via set-union operation given in (2) is performed.

$$B = \bigcup_{k=1}^{N} B_k \tag{2}$$

Fig. 1 illustrates this operation. In addition to this unified boundary, a reverse map from the pixel positions to the list of logo sequence frames, whose boundary intersects with these positions, is also created. This map can be represented mathematically as in (3).

$$S(x, y) = \{ n \mid (x, y) \in B_n \}, n = 1 \ldots N \tag{3}$$

This map is utilized during detection for fast comparison of video frame logos against the multi-frame logo sequence.

2.2. Animated Logo Detection

The outputs of training, B and $S(x, y)$, are used during the detection process. Logo detection operations that are performed for each frame in the test video starts by computation of Laplacian operator and the resulting edge image is denoted as $L_t(x, y)$. In this notation, t subscript denotes the frame index of the input test sequence. The resulting image is then thresholded to obtain a binary map. This empirically decided threshold (τ_1) is used to select the

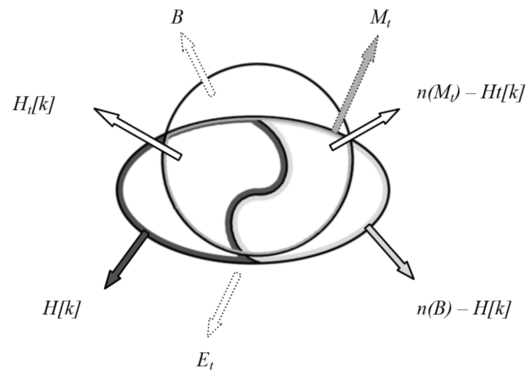

Fig. 2. Decision parameters used in detection. Arrows with dotted boundaries show sets and arrows with solid boundaries show values defined.

pixels for which $L_t(x, y) \geq \tau_1$ or $L_t(x, y) \leq -\tau_1$ as candidates for logo contour. These candidate points are denoted by E_t and are further filtered by applying the requirement of existence in the unified contour map. This new boundary set, which is obtained by the intersection of B and E_t, is called M_t.

$$M_t = B \cap E_t \tag{4}$$

It should be noted that during the boundary matching/intersection phase in (4), the region of interest (ROI) from which candidate boundary pixels are extracted is decided iteratively, in order to account for insignificant logo position variations. The limits of position change and therefore ROI sliding width and height is preset and constant throughout the experiments. After the filtered test frame boundary candidates, M_t, are obtained, these results should be efficiently compared to all boundaries of all frames in a multi-frame logo. However, this would result in a computation time proportional to the length of logo sequence. On the other hand, exploiting the analogy between the animated logo and the parametric curve detection problems [12], this template matching case can be handled in a parallel voting scheme.

Each pixel in the filtered candidate boundary (M_t) votes for the logo frames in whose boundary it is included, analogous to the points in original Hough Transform voting for all the lines passing through them. For this step to be performed, the map from the pixel positions to the list of logo sequence frame numbers is used.

$$S_t(x, y) = \{ S(x, y) \mid (x, y) \in M_t \} \tag{5}$$

The votes are accumulated in a histogram, H_t, whose resolution is equal to N, which is the number of frames in the logo sequence. Next, this histogram is normalized non-

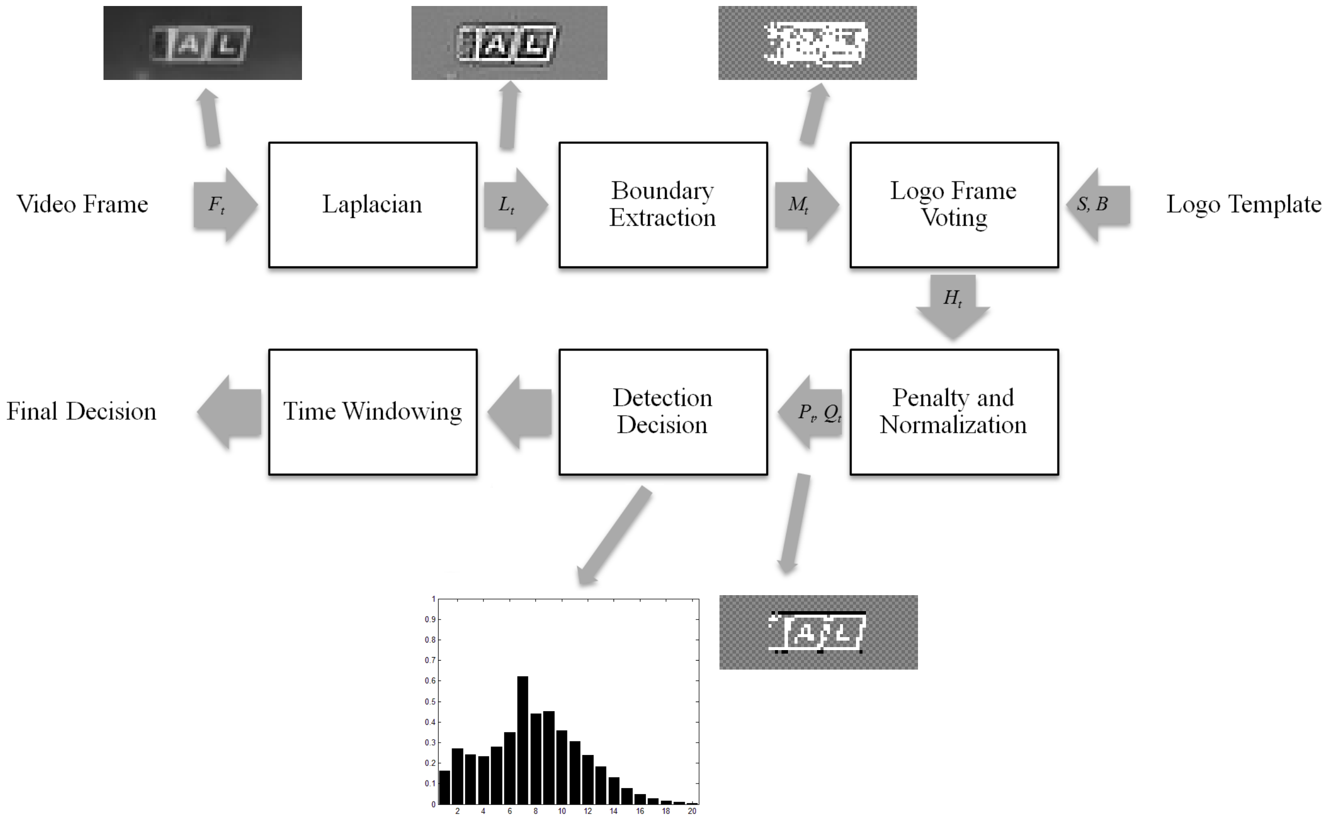

Fig. 3. Overview of the overall detection system.

uniformly using the support size of each logo frame as the normalizing factor. The result is a measure (T_t) to compare the existence probability of each logo frame with each other and a threshold.

$$T_t[k] = \frac{H_t[k]}{H[k]}, \forall k \tag{6}$$

In (6), $H_t[k]$ denotes the number of votes to k^{th} logo frame, and $H[k]$ is the number of pixels on its contour, B_k. The set of edges which vote for $H_t[k]$ are depicted in Fig. 2. The maximum value of this histogram is the most probable frame of the logo which the current test frame contains. This proposal is evaluated using a predetermined threshold τ_2 that is constant for all experiments similar to τ_1. The decision function is given in (7).

$$\begin{array}{l} T_t[k_0] \geq \tau_2 \rightarrow Detected \\ T_t[k_0] < \tau_2 \rightarrow Not\ \ Detected \end{array}, k_0 = \arg\max_k \left(T_t[k]\right) \tag{7}$$

Although this measure mostly performs acceptable, it has a vulnerability that may degrade the performance of the overall algorithm. Some logo sequences may contain small logo frames with a very few boundary pixels. In such cases, edges arising from the background video can easily delude this simple measure to detect a small logo boundary with high confidence. In order to handle these deceptive cases that affect the precision of the overall algorithm, it is proposed to incorporate negative clues as well as the positives in the decision measure. Positive ($P_t[k]$) and negative ($Q_t[k]$) clues regarding the existence of the kth logo frame at the video frame denoted by index t are defined as in (8).

$$P_t[k] = \frac{H_t[k]}{H[k]} = T_t[k] \qquad Q_t[k] = \frac{n(M_t) - H_t[k]}{n(B) - H[k]} \tag{8}$$

where $n(M_t)$ and $n(B)$ are number of pixels in M_t and B respectively. These two measures are normalized to the interval $[0,1]$ and therefore can be treated like probability measurements. This also renders them suitable for combination in a generic scheme developed for classifiers [13]. Two of the decision combination rules are used for the experiments in this paper, namely Sum Rule and Product Rule. The resulting measures are given in (9). Separation of edges based on logo templates are depicted in Fig. 2.

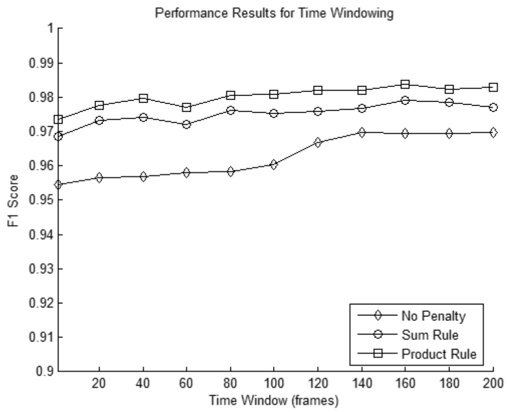

Fig. 4. Performance results.

$$\begin{array}{l} \textit{Sum Rule: } P_t[k] + (1 - Q_t[k]) \xrightarrow{\textit{simplified}} P_t[k] - Q_t[k] \\ \textit{Product Rule: } P_t[k] \cdot (1 - Q_t[k]) \end{array} \quad (9)$$

Simulation results are obtained for each of these three measures, and the effects of using combination rules are analyzed. Detection results obtained in frame level using this algorithm are further refined in order to prevent spurious detections. This is achieved as shown in Fig. 3 by a time-windowed majority vote filtering, in which uncommon detection patterns yield to preceding stable decision blocks.

3. EXPERIMENTS

The performance of the proposed method is investigated utilizing three different real life animated TV logos. First, the experiment setup is described. Then the detection performance is given. Finally the computational measurements are summarized.

3.1. Experiment Setup

Experiments are carried out on a test set of approximately 27 hours that contains the three animated broadcast logos of three Turkish TV channels, namely *Airbox TV*, *Kral TV* and *Akilli TV*. Each channel has a share of approximately nine hours.

Test videos are captured from digital video broadcasts and transcoded in MPEG-4 H.264 with CIF resolution, 25fps and have average bitrate of 300 kb/s. The incoming broadcast video is transcoded to a lower resolution and bitrate in order to observe the performance of the algorithm in this setup, since modern archive systems generally use the most recent encoding standard, MPEG-4 H.264, due to its compression superiority.

Since the algorithm operates on spatial domain, videos are decoded into pixel domain for processing. Therefore computation time also includes the decoding process.

3.2. Detection Performance

The detection performance is measured using the F1 score metric [14], which is given in (10).

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \quad (10)$$

Precision and recall values are computed by comparing the algorithm results and the ground truth of the test set.

Effects of time windowing and decision rules on the detection performance are investigated. The results are depicted in Fig. 4. It can be observed from Fig. 4 that both penalty rules perform better than counting only edge matches. This result indicates that the presence of edges outside the template pixels supplies discriminative information for logo detection. Time windowing slightly improves the performance for all decision rules, which concludes the fact that decisions given frame by frame are almost always stable.

Frame by frame performance of the algorithm reaches an F1 score of 97.36%. Table 1 shows the results obtained for three logos separately. The accuracy increases to 98.10% when a time window, which introduces a 4-second delay, is used.

Table 1. F1 scores for each channel with all three decision rules.

	No Penalty	Sum Rule	Product Rule
Airbox TV	99.83%	99.32%	99.52%
Kral TV	95.83%	94.40%	95.78%
Akilli TV	90.62%	96.82%	96.74%

3.3. Computation Time

A reference implementation written in C++ is tested on a PC with Intel Core 2 2.4GHz central processor and 1GB main memory which is running Linux kernel 2.4.6.

Detection of logos for all three measures has an average running rate of 477.83 frames per second. Decoding of the videos consume 4.97% of actual media time and 0.25% of the real time is required for the detection of animated logos with our method. Time required with the naive method, which compares every video frame with each of the logo frames (i.e. treats each logo frame as a distinct static logo), becomes 5.81% of the real time. Thus the proposed method can be considered about 20 times faster than treating each logo frame as a separate static logo. Average number of frames of the three logos used in the experiments is 134.

4. CONCLUSIONS

In this paper, we have visited animated television logo detection problem. A real-time method for this detection problem has been developed based on a unified logo boundary, which is extracted from the multi-frame logo sequence. A voting-based decision method is utilized for the determination of logo presence. It has been demonstrated that the discussed detection system obtains considerable performance results. Moreover, significant performance gains are achieved by utilizing Sum Rule and Product Rule for the combination of the introduced positive and negative clues that are associated with the existence of the logo. Furthermore, time-windowed majority voting has been shown to boost accuracy.

5. REFERENCES

[1] Doermann, D. S.; Rivlin, E.; and Weiss, I.; “Logo recognition using geometric invariants”, Proc. of Second ICDAR, 1993.

[2] Seiden, S.; Dillencourt, M.; Irani, S.; Borrey, R.; and Murphy, T.; “Logo detection in document images”, Proc. CISST, 1997.

[3] Gunsel, B.; Ferman, A.M.; and Tekalp, A.M.; "Video indexing through integration of syntactic and semantic features", Third IEEE WACV, Sarasota, USA, 1996.

[4] Meisinger, K.; Troeger, T.; Zeller, M.; Kaup, A.; “Automatic TV Logo Removal Using Statistical Based Logo Detection and Frequency Selective Inpainting”, EUSIPCO, 2005.

[5] Wang, J.; Duan, L.; Li, Z.; Liu, J.; Lu, H.; Jin, J. S.; “A robust method for TV logo tracking in video streams”, IEEE ICME 2006.

[6] Ekin, A.; Braspenning, E.; “Spatial detection of TV channel logos as outliers from the content”, in Proc. VCIP. SPIE, 2006.

[7] “Law on the Establishment of Radio and Television Enterprises and Their Broadcasts Law No. 3984 of 20 April 1994”, Turkey.

[8] Albiol, A.; Ch, M. J.; Albiol, F. A.; Torres, L.; “Detection of TV Commercials”, IEEE ICASSP 2004.

[9] Santos, A. R.; Kim, H. Y.; “Real-time opaque and semi-transparent TV logos detection”, I2TS, 2006.

[10] Duffner, S. and Garcia, C. “A neural scheme for robust detection of transparent logos in TV programs”, ENNS International Conference on Artificial Neural Networks, 2006.

[11] Porter, T. and Duff, T., “Compositing digital images”, SIGGRAPH '84. ACM, 1984.

[12] Duda, R.O. and Hart, P.E.; “Use of the Hough Transform to Detech Lines and Curves in Pictures", Comm. ACM, vol. 15, pp. 11-15, 1972.

[13] Kittler, J.; Hatef, M.; Duin, R.; and Matas, J. “On combining classifiers”, IEEE TPAMI, 20(3):226--239, 1998.

[14] Van Rijsbergen, C.J. (1979) Information Retrieval. London: Butterworths.