Automatic logo transition detection in digital video contents

Chang-Yu Lu, Myung-Cheol Roh, Seung-Yeon Kang & Seong-Whan Lee

Pattern Analysis and Applications

ISSN 1433-7541
Volume 15
Number 2

Pattern Anal Applic (2012) 15:175-187
DOI 10.1007/s10044-012-0267-9

Springer

Pattern Anal Applic (2012) 15:175–187
DOI 10.1007/s10044-012-0267-9

INDUSTRIAL AND COMMERCIAL APPLICATION

Automatic logo transition detection in digital video contents

Chang-Yu Lu · Myung-Cheol Roh · Seung-Yeon Kang · Seong-Whan Lee

Received: 3 August 2009 / Accepted: 8 March 2012 / Published online: 24 March 2012

Abstract The amount of user created contents has been increasing rapidly and is associated with a serious copyright problem. Automatic logo detection and recognition in videos is a natural and efficient way of overcoming the copyright problem. However, logos have varying characteristics, which make logo detection and recognition very difficult. Moreover, logo transitions between two different logos exist in one video comprising several video contents. This disrupts the automatic logo detection and recognition. Therefore, in order to improve logo detection, it is necessary to take into account the logo transitions explicitly. This paper proposes an accurate logo transition detection method for recognizing logos in digital video contents. The proposed method accurately segments a video according to logo and efficiently recognizes various types of logos. The experimental results demonstrate the effectiveness of the proposed method for logo detection and video segmentation according to logo.

C.-Y. Lu · M.-C. Roh · S.-Y. Kang
Department of Computer Science and Engineering,
Korea University, Anam-dong, Seongbuk-ku,
Seoul 136-713, Korea
e-mail: cylu@image.korea.ac.kr

M.-C. Roh
e-mail: mcroh@image.korea.ac.kr

S.-Y. Kang
e-mail: sykang@image.korea.ac.kr

S.-W. Lee (✉)
Department of Brain and Cognitive Engineering,
Korea University, Anam-dong, Seongbuk-ku,
Seoul 136-713, Korea
e-mail: swlee@image.korea.ac.kr

Keywords Logo transition detection · Logo detection and recognition · Sub-video merging · Video segmentation based on logos

1 Introduction

The continuing development of the internet and increasing user created contents (UCCs) is associated with a serious copyright problem for video content. This is depicted in Fig. 1. The user gathers digital video contents via illegal or legal channels and modifies the video content. The modified video contents can be distributed without the permission of the owner and without paying a copyright fee, while the video owner wants to protect copyrighted contents from illegal use of video contents. Automatic logo detection in videos is a natural and efficient way of overcoming the copyright problem. This is because logos contain most important information about copyright claims of owners and no additional information, e.g., watermark, is required. Automatic logo detection is also required in various applications: video cataloging, logo removal, distinguishing between an advertisement and a TV program, and replay shot detection in sports videos [1–4].

Nowadays extra tags are included in many digital media contents, and in that case, the extra tags can be easily utilized to recognize logos. However, there are still more contents which dose not include extra tags. Many such videos can be easily found in Youtube or illegal media sites.

There are several problems in logo detection and recognition systems. One problem is that TV logos have different colors, shape, etc. We can categorize logos into three different types, which are shown in Fig. 2: semi-transparent, opaque, and animated logos. If a logo changes over time,

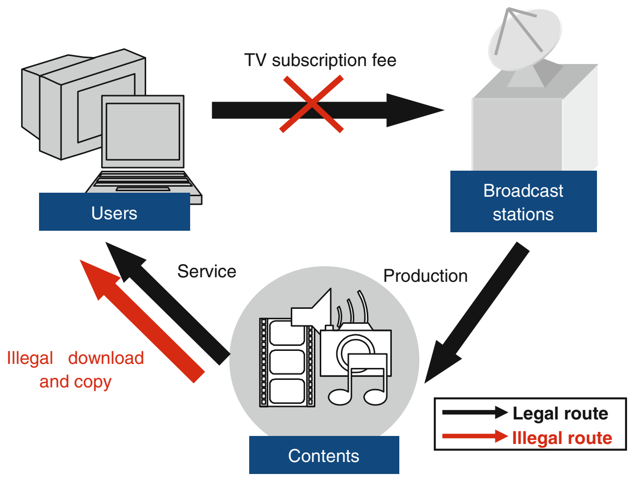

Fig. 1 The flow of video contents by legal and illegal routes. Copyright problem is raised when users redistribute (modified) contents without paying proper fee

then it is defined as an animated logo. If the background of a logo is visible, this is defined as a semi-transparent logo, otherwise, the logo is defined as an opaque logo. An opaque logo is independent of the background. On the other hand, a semi-transparent logo depends on background changes. A semi-transparent logo's characteristics make recognition difficult. Especially, extracting semi-transparent logos from a single frame is very difficult because there is insufficient information. Thus, in general, methods are performed in the temporal domain.

Another problem is that logo transition exists in one video; appearing, disappearing, and changing logos. Some examples are shown in Fig. 3. The transitions are: appearing logo (absence of logo → presence of logo), disappearing logo (presence of logo → absence of logo), changing logo (logo $A \rightarrow$ logo B). In modified videos (e.g., UCCs), there can be frequent logo transitions, and this disrupts logo detection and recognition. Thus, transition detection must be considered in logo detection.

The main novelty idea is to take into account the logo transitions explicitly. Also, we propose an automatic logo determination method. First, video segmentation using a shot boundary detection method is performed. Second, the video is separated into several sub-videos according to the shot boundaries. Third, in each sub-video, a luminance variance image is calculated, and a logo template is extracted from the luminance variance images. Finally, logo recognition results are merged according to logos, which are recognized from templates. By the proposed method, logos can be recognized accurately in digital video contents. Moreover, videos can be efficiently cataloged and segmented by its logos.

The remainder of this paper is structured as follows: In Sect. 2, related work is presented. In Sect. 3, novel frameworks for extracting and recognizing logos are proposed. Experimental environments and results are shown in Sect. 4. Finally, we conclude with the results described in Sect. 5.

2 Related work

Many text detection and recognition methods have been proposed. Recently, Kleban et al. [5] proposed a logo detection in natural scene, which is based on mining frequent spatial configurations of discretized local features at multiple resolution. Bagdanov et al. [6] proposed a trademark matching and retrieval system for sports videos, which are based on SIFT feature descriptor. The localization and

Fig. 2 The various types of logos. **a** Example of semi-transparent logos. **b** Example of opaque logos. **c** Example of an animated logo

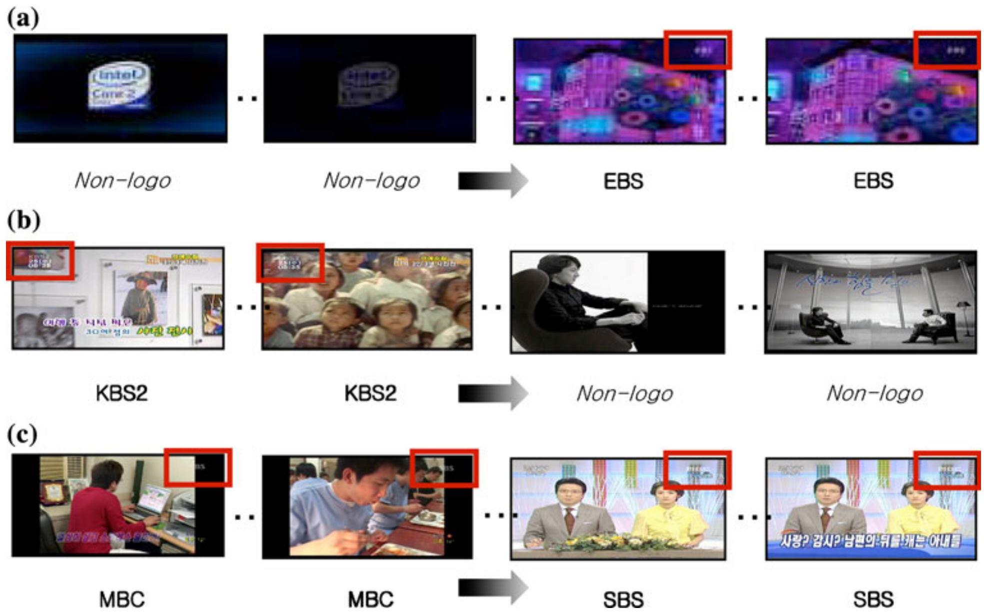

Fig. 3 Examples of logo transitions. **a** Appearing logo. **b** Disappearing logo. **c** Changing logo

classification of the trademarks were performed by the features. Gao et al. [7] proposed a logo detection method based on spatial-spectral saliency and partial spatial context. They conducted experiments on 10 logos.

Text detection and recognition methods showed very good results in various environments [8, 9]. However, the scene texts are tend to be discriminated from their background and are opaque. Therefore, it is not suitable to use the approaches to detect and recognize TV logos since many of the TV logos are rather not to be discriminated clearly in each video frame but are still distinguishable in frame sequence. Moreover, there are many semi-transparent TV logos.

Many works on logo detection in TV have been applied in various types of applications: commercial detection [10–12], sports replay detection [4, 13], logo removal [2, 14–16], video cataloging [17], and video segmentation. Logos are detected from a single frame or consecutive frames, according to the applications.

For semi-transparent logo detection, Seeber et al. [18] proposed a TV logo detection method based on template matching using inner and external edge masks in a single frame. This algorithm only efficiently detects semi-transparent logos in high-resolution videos. Duffner et al. [19] proposed a multi-resolution scheme based on convoluted neural networks.

Generally, logo detection algorithms used temporal information. Yan et al. [2] demonstrated a logo removal application. The logo is detected by exploiting frame differences in video sequences. The proposed method involves locating fragments of logos using a Bayesian approach. The accuracy of this method is low for logos having color information.

In many sports videos, a replay logo is animated in uniform frames. Bai et al. [4] proposed a logo detection method based on the motion analysis. A logo template is learned using dynamic programming and unsupervised clustering. Characteristics of logo transitions are extracted using optical flow features. The detection performance for different types of sports videos shows that this method is effective and robust.

Cózar et al. [1] presented a framework for cataloging videos. The authors performed temporal and spatial segmentation by calculating the minimum luminance variance region (MVLR). If the logo detection method fails to recognize a logo, a new logo template is registered in the database. This method may suffer from detection errors due to logo transitions.

There are a few methods that use complex methods such as HMMs, NNs. In terms of logo recognition, in order to utilize the logo recognition in various video contents, the complex methods require a large set of training data. It is possible to collect enough data for a few logos. However, collecting large enough data for many logos would be valuable, but hard work.

This paper proposes a logo transition detection method for logo detection using shot boundary detection. The logo detection/recognition and logo transition detection are very close issues. If we have a perfect logo detection/recognition, the logo transition detection is not required. However, if we do not have, the logo detection fails due to logo transitions in a video. The proposed method takes explicitly into account logo transition on the logo detection method. Thus, the logo detection and the logo transition detection are tied up together in this paper. As well as the logo transition detection, an opaque and semi-transparent logo detection method based on automatic determination of logo type is also proposed.

3 Proposed method of logo detection and recognition

The overview of the proposed method is shown in Fig. 4. The proposed method consists of five components. An input video is segmented into several sub-videos, using a shot boundary detection method. Candidate logo templates are extracted based on luminance variances in image sequences of sub-videos. Prior to recognition, it is determined whether the extracted candidate logos are opaque or semi-transparent. Then, each logo in a sub-video is recognized by using appropriate methods, and sub-videos are merged to eliminate unnecessary boundaries and find logo transition boundaries. Figure 5 shows the overview of the proposed logo recognition method with a sample video that consists of two different logo videos.

3.1 Video segmentation

A video edited by the user may comprise a set of video contents. We assume that if there is a logo transition there is a shot change. In order to extract and recognize logos in the video, a detection method of logo transition boundaries is needed. In previous work, logo templates are extracted from frames of fixed length. However, as shown in Fig. 6a, if a video is separated into sub-videos based on fixed length frames, some sub-videos may include several logos. This causes incorrect result of logo recognition. To solve this problem, in previous work, a logo-tracking technique was suggested. However, tracking logos are insufficient for cases in which there are similar logos in consecutive sub-videos.

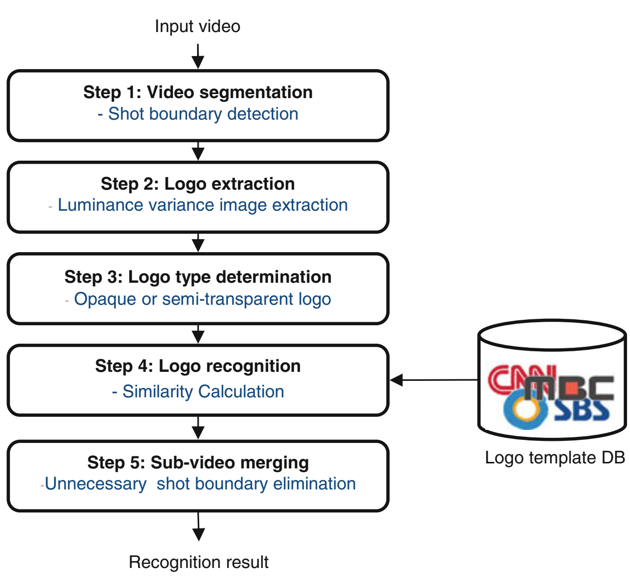

Fig. 4 The proposed logo recognition system in video

We proposed a video segmentation method based on shot boundary detection, as shown in Fig. 6b. A video is segmented into several sub-videos based on shot boundary detection. A video that have one logo can be possibly separated into several sub-videos. However, no sub-video includes several logos. There are logo changes in some of these shot boundaries.

Two types of cuts are used for shot boundary detection: abrupt cuts and gradual cuts. However, in general, logo transitions involve abrupt cuts, because a logo change implies changes of contents or background in video. Therefore, in this paper, a shot boundary detection method based on differences between consecutive two frames is used for video segmentation. This is among the predominant methods for shot boundary detection.

Fang et al. [3] and Boussaid et al. [20] described general features for shot boundary detection which work regardless of logos. We used two features for shot boundary detection: the color histogram difference and the motion information difference. Features are calculated between consecutive frames. The first feature is the color histogram difference (CHD), which calculates the difference between two color histogram distributions in two images. CHD is usually considered a global feature. CHD is defined as follows:

$$\mathrm{CHD}_i = 1 - \frac{1}{3p}\left[\sum_{j=1}^{256} \min\left(R_i^j, R_{i-1}^j\right) + \sum_{j=1}^{256} \min\left(G_i^j, G_{i-1}^j\right) + \sum_{j=1}^{256} \min\left(B_i^j, B_{i-1}^j\right)\right] \quad (1)$$

where p represents the total number of pixels in a frame and R_i^j, G_i^j, and B_i^j represent the values of the jth bins of the red, green, and blue color histograms of the ith frame, I_i. The value of CHD_i is in the range [0, 1]. The more closely the value approaches 1, the larger the difference between the two frames.

The second feature is the motion compensation. Motion information enables us to measure the visual discontinuity between two frames. Two adjacent frames are divided into blocks of $N \times N$ pixels and the blocks are not overlapped. Here, we set $N = 16$. In order to measure the level of match between the blocks in $i-1$th and ith frames, sum of absolute differences are calculated as follows:

$$S_\mathrm{d}(u, v) = \sum_{x=1}^{N} \sum_{y=1}^{N} \left| B_i^u(x, y) - B_{i-1}^v(x, y) \right| \quad (2)$$

where $B_i^u(x, y)$ and $B_{i-1}^v(x, y)$ represent the pixel intensities at (x,y) of the uth block in I_i and vth block in I_{i-1}, and N is the size of the block. The block having the minimum S_d value in I_{i-1} is selected as the best matching one with the

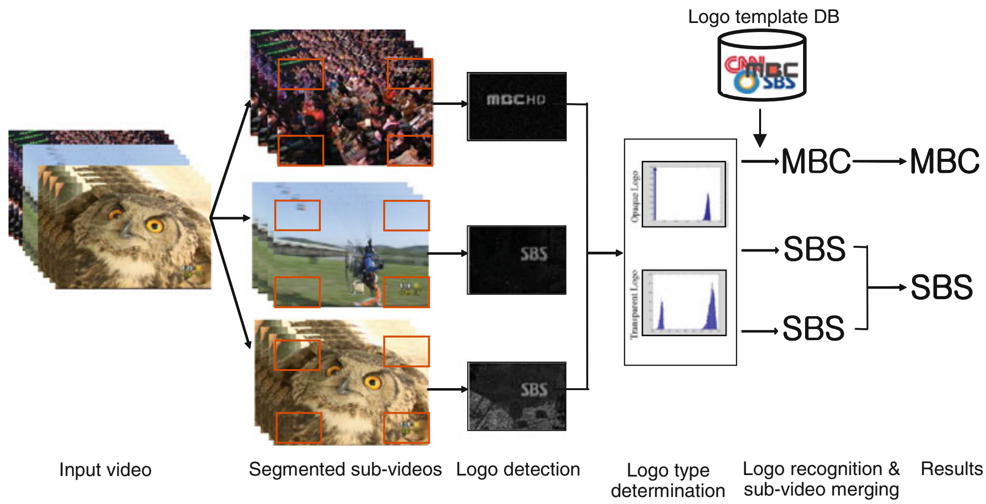

Fig. 5 The overview of the proposed logo recognition method

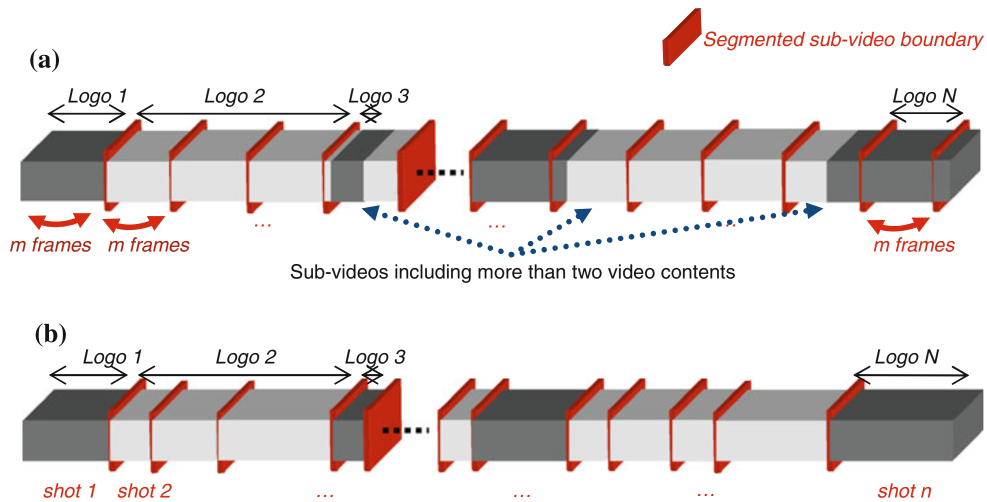

Fig. 6 Video segmentation **a** based on fixed frame length, **b** based on shot boundary detection

reference one in I_i. Based on the best matching blocks, the motion compensation feature is defined as follows:

$$\mathrm{Mc}_i = \frac{1}{N_\mathrm{b}} \sum_{n=1}^{N_\mathrm{b}} \left(\left| \overline{Y_n} - \overline{Y_n}' \right| + \left| \overline{U_n} - \overline{U_n}' \right| + \left| \overline{V_n} - \overline{V_n}' \right| \right) \quad (3)$$

where N_b is the number of blocks in the frame; $\overline{Y}, \overline{U}$, and $\overline{V}$ are the average Y, U, and V components of the blocks in I_i; and $\overline{Y}', \overline{U}'$, and $\overline{V}'$ are the corresponding components of the best matching blocks in I_{i-1}. YUV color space is used in most methods for motion estimation and those features are used to detect shot boundary in [3, 20].

An abrupt shot cut is detected if one of the peak values of the two features is detected. The peak is found using a sliding window which contains frames $i-1$, i and $i+1$. The peak is detected if the conditions $\Delta_1^k > 0$ and $\Delta_2^k < 0$ are satisfied. The Δs are defined as follows:

$$\Delta_1^k = F_i^k - F_{i-1}^k, \quad \Delta_2^k = F_{i+1}^k - F_i^k \quad (4)$$

where $F^k, (k \in \{1, 2\})$ is one of the two feature values extracted from Eqs. (1) and (3). For more details about shot detection we used in this paper, please refer [3, 20].

As a result, a single video is divided into several sub-videos, and each sub-video has at most one logo, as we assumed that if there is a logo transition, there is a shot change.

3.2 Logo extraction

Subsequent to video segmentation, a video is separated into several sub-videos. In this section, the method of extracting a logo template from a sub-video is described.

Logos can be divided into two types: animated and static logos. Static logos are classified into two types: opaque and semi-transparent logos. This paper deals with static logos. Animated logo detection needs a different approach, since the appearance of these kinds of logos are changed continually. Static logos have several characteristics. Firstly, TV logos with the same owner have the same shape and size.

Secondly, the variations in TV logos are smaller than variation in the background in the temporal domain. Thirdly, logos are detected at the specified positions. Logos are generally located in one of four areas: top-left, top-right, bottom-left and bottom-right in the video. Fourthly, logos are displayed in monochrome or a small number of colors. Finally, logos generally contrast with the background so that they are distinguished from others in the video. Based on these characteristics, a robust logo extraction method is presented.

Let a video V consist of consecutive sub-videos, $V = \{S_1, S_2, \ldots, S_n\}$. Each sub-video is a set of consecutive frames, $S_i = \{I_1, I_2, \ldots, I_{k_i}\}$ where $1 \leq i \leq n$ and k_i is the number of frames in the sub-video S_i. Sub-videos are shots which are extracted by the shot boundary detection method we mentioned in previous section.

To extract the logo template from each sub-video, the luminance variance image (LVI) of the sub-video is calculated [1]. LVI represents the maximum luminance variance in the sub-video. Since luminance variance values in a logo region are less than those in a non-logo region, pixels with low luminance variance values are logo candidate pixels. Let $S_i(k)$ be the set of kth pixels in all frames in the sub-video S_i. LVI of ith sub-video is calculated as follows:

$$\mathrm{LVI}_i(k) = \max(S_i(k)) - \min(S_i(k)), \quad 1 \leq k \leq p \tag{5}$$

where k is the index of a pixel and p is the number of pixels in a frame. The logo candidate template in the sub-video, CL_i, is extracted as follows:

$$\mathrm{CL}_i(k) = \begin{cases} \mathrm{LVI}_i(k) & \text{if } \mathrm{LVI}_i(k) \leq \tau \mathrm{LVI}_{\mathrm{mean}} \\ 0 & \text{if } \mathrm{LVI}_i(k) > \tau \mathrm{LVI}_{\mathrm{mean}} \end{cases}, \quad 1 \leq i \leq p \tag{6}$$

where $\mathrm{LVI}_{\mathrm{mean}}$ is a mean value of the LVI and, for a given τ, $\tau\ \mathrm{LVI}_{\mathrm{mean}}$ is a threshold value that depends on the $\mathrm{LVI}_{\mathrm{mean}}$.

3.3 Determination of the logo type

Prior to performing logo recognition, we determine the types of candidate logos since opaque and semi-transparent logos have different characteristics and, therefore, different methods of feature extraction for recognition must be applied according to their logo types. The proposed logo type determination method is described in this section.

The luminance variance value of a semi-transparent logo is larger than that of an opaque logo. The luminance variance value of an opaque logo approaches 0 in the LVI intensity histogram. However, the luminance variance value of a semi-transparent logo is somewhat right skewed in the LVI intensity histogram. Using such a difference, we can determine whether it is an opaque logo or a semi-transparent logo.

Based on the Bayes rule, $P(Logo\ Type|Feature)$ is represented as follows :

$$P(\mathrm{LT}|F) = kP(F|\mathrm{LT})P(\mathrm{LT}) \tag{7}$$

where LT, F, and k represent logo types, features of a logo, and normalization factors, respectively.

Huge number of video data are being generated every second in the global. It can be thought that the number of data is almost infinite. Among the video data in the global, the number of sample data we can collect would be very small. Moreover, the number of different types of the collected video logos is likely to be very different depending on your location, time, date, favor, available TV channels, and so on. Therefore, we cannot expect that the ratio between semi-transparent and opaque logos of collected data is nearly consistent every time we collect a data set. It is seemed that the uniform prior assumption in Eq. 7 is intuitively natural in this case. Thus, we assumed

$$P(\mathrm{LT} = \text{``semi-transparent''}) = P(\mathrm{LT} = \text{``opaque''}). \tag{8}$$

Using training data, the probability distributions of $P(F|\mathrm{LT})$ are modeled by Gaussians. Then, the posterior probabilities, $P(\mathrm{LT} = \text{“semi-transparent”} |F)$ and $P(\mathrm{LT} = \text{“opaque”}|F)$, are easily estimated from the training data.

3.4 Feature extraction and logo recognition in sub-videos

For accurate logo recognition, feature extraction has to be done. One of the useful features is edge. However, the edge image which is directly extracted from the LVI is not always adequate as a good feature in logo recognition. Figure 7 shows two examples of the cases that the edge within the logo or the edges of the logo itself are disappeared in LVI. In Fig. 7, images in the first column show the input video frames (a-1 and a-2). Images in the second (b-1 and b-2) and third column (c-1 and c-2) show the extracted feature images using the method of Cózar et al. and the proposed method, respectively. Fig. 7a-1 shows an example of a logo in which the text is wrapped within a static circle. Figure 7b-1 shows an example of a logo in which the text is surrounded by a static background (a black border which frequently appears in movies). For these videos, the traditional method fails to extract features in the logo regions (Fig. 7b-1, b-2). Therefore, we proposed a feature extraction method based on logo type.

If a logo template is determined as opaque logo, the feature is extracted by combining the LVI edges and a randomly selected frame in the sub-video. Since the edges of an opaque logo are always clear in every frame in a sub-video, we can select any frame with a logo as a reference frame. Features of an opaque logo template are defined as follows:

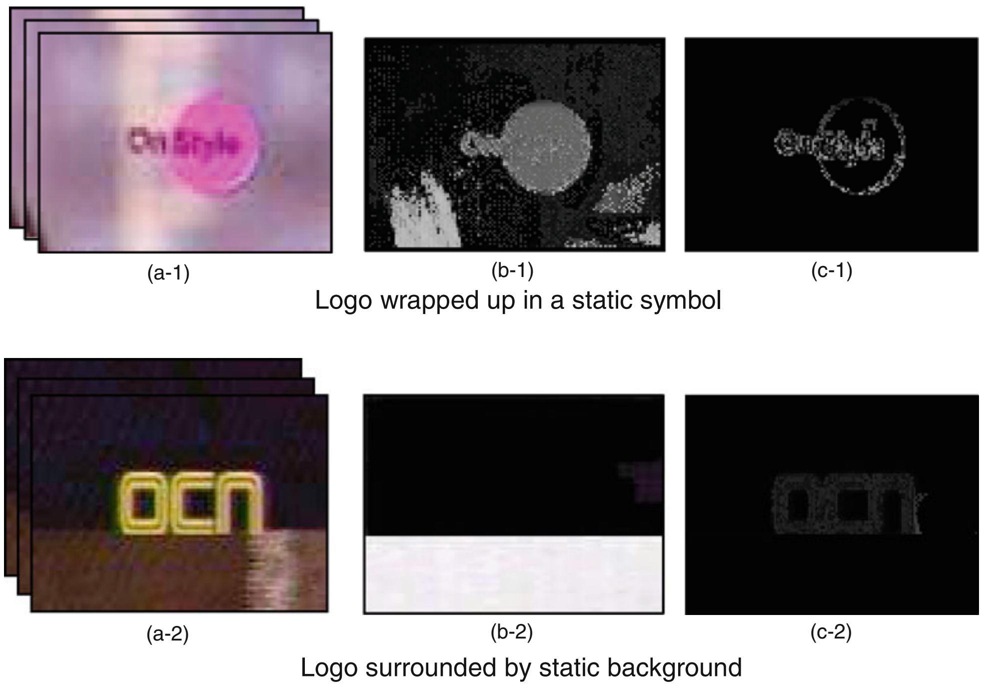

Fig. 7 Examples of feature extraction of logo wrapped up in a static symbol (*first row*) and logo surrounded by static background (*second row*). **a-1**, **a-2** Input video frames. **b-1 b-2** Extracted features by Cózar et al. **c-1**, **c-2** Extracted features by the proposed method

$$L(i) = \begin{cases} \max(E_R(i), E_{\mathrm{CL}}(i)) & \text{if } \mathrm{CL}(i) = 0 \\ E_{\mathrm{CL}}(i) & \text{otherwise} \end{cases} \tag{9}$$

where $E_R(i)$ and $E_{\mathrm{CL}}(i)$ are the ith pixel values of the edge images of the reference frame and CL (a logo candidate template), respectively. The edge image E of an image I, is extracted using the magnitude of the gradient of the image, as defined below:

$$E(I) = \|\nabla I\| = \sqrt{\left[\frac{\partial I(x,y)}{\partial x}\right]^2 + \left[\frac{\partial I(x,y)}{\partial y}\right]^2}. \tag{10}$$

If a logo template is determined as semi-transparent, it is difficult to extract the logo edges from a random frame in the manner used with an opaque logo. As mentioned before, semi-transparent logos depend on the background and frequently change. However, in general, the inside of a semi-transparent logo does not contain much text information and is rather simple, so it is possible to immediately use the edge information extracted from the LVI as its features.

For logo recognition, the edge-based template matching method is employed. To measure the similarity, the normalized cross-correlation $N(m, n)$ is employed and is scaled to ensure that it is in the range [0, 1]. Remark that for two different logo types, the applied feature extraction methods are different according to its logo types, although the recognition methods are identical.

3.5 Merging sub-videos for logo transition detection

In this section, we describe a sub-video merging method. In order to detect logo transitions, a shot boundary detection method is performed in previous step. However, segmentation is achieved by shot boundary detection and unnecessary boundary shots can be detected during video segmentation. It may be the case that a video is divided into several consecutive shots, although the shots have the same logos. This can be eliminated by merging sub-videos. Moreover, if logo detection in a sub-video fails, it can be compensated for by considering prior and subsequent sub-videos.

If a logo in a sub-video is detected and recognized, then the sub-video is tagged by the logo. If the logo is same as the prior sub-video, they are merged in one new sub-video. However, some sub-videos with an insufficient number of frames may fail to detect and recognize the logo. If a logo in a sub-video is not detected or recognized due to little information of a logo, the sub-video is merged with prior tagged sub-video or next sub-video, and retries to recognize a logo from the merged sub-video. If a logo in the merged sub-video is not recognized, the sub-video is tagged by non-logo, but if the merged sub-videos is recognized with some confidence (correlation value), the merged sub-video is tagged by the logo. It is repeated for all remaining sub-videos. Algorithm 1 shows the detailed algorithm for merging sub-videos.

Algorithm 1: Sub-video merging

Let S be a stack having sub-videos as elements, and $Recog(v)$ be a recognition function that returns an index (I) of best matched logo in a logo template DB, and normalized-correlation value (c) between I and the logo candidate template in the input sub-video, v. Th and ϵ are threshold values. Let $\mathcal{V}$ be an input video which consists of sub-videos, $\{v_1, v_2, \cdots, v_n\}$.

```
for i ← 1 to n do
    v ← v_i;
    (c,I) ← Recog(v);
    if c < Th then
        if S.isEmpty() then
            S.push(v);
        else
            v_new ← v;
            while !S.isEmpty() do
                v_new ← Merge{ S.pop(), v_new };
                (c_new, I_new) ← Recog(v_new);
                if c_new < c − ε or c_new < Th then
                    S.removeAll();   S.push(v);   break;
                else
                    v ← v_new;   c ← c_new;
                end
            end
        end
    else if !S.isEmpty() then
        v_new ← v;
        while !S.isEmpty() do
            v_new ← Merge{ S.pop() , v_new };
            (c_new,I_new) ← Recog(v_new);
            if I = I_new and c_new ≥ c − ε then
                v ← v_new;
            else
                break;
            end
        end
        S.removeAll();   S.push(v);
    end
end
```

4 Experiments

4.1 Experimental environment

In order to make experiments, various 300 videos containing logos were collected from 5 Korean broadcast station channels, 7 cable channels, and various other TV channels; MBC, KBS1, KBS2, SBS, EBS, MTV, Tooniverse, MBN, OCN, Onstlye, Olive, MBC ESPN, Storyon, ABC, CTV, CW, and CNN. The frame size of the video is normalized to 320 × 240 pixels. The time lengths of the videos are various, from 15 to 43 s, and the average length is 26.4 s. Figure 8 shows some examples of the collected video data. Figure 8a–c shows example frames with semi-transparent logos. Figure 8d shows frames with opaque logos.

4.2 Experimental results

To measure the performance of the proposed method, two experiments were performed. The first experiment measures the logo detection accuracy and the second experiment measures the logo transition accuracy.

4.2.1 Logo recognition

For logo recognition experiment, 300 videos with 17 different logos are used. Among 300 videos, 150 videos have opaque logos and 150 videos have semi-transparent logos.

The performance of the logo recognition system is evaluated by

$$\text{Recognition rate } (\%) = \frac{N_{r_c}}{N_{r_c} + N_{r_f} + N_{r_m}} \times 100 \tag{11}$$

where N_{r_c} is the number of correctly recognized logos, N_{r_f} is the number of incorrectly recognized logos and N_{r_m} is the number of missed logos. The recognition result is shown in Table 1 with various values of τ given in Eq. 6. Setting τ low ($\tau = 0.50$) means that we set the threshold value for extracting logo tightly. Therefore, false negative (missing in the table) would be increased. On the other hand, if we set τ high ($\tau = 1.00$) the threshold value is loose. Therefore, false positives (false in the table) would be increased while false negative would be decreased.

It is clear that the proposed method based on determination of logo type yields good performance for opaque

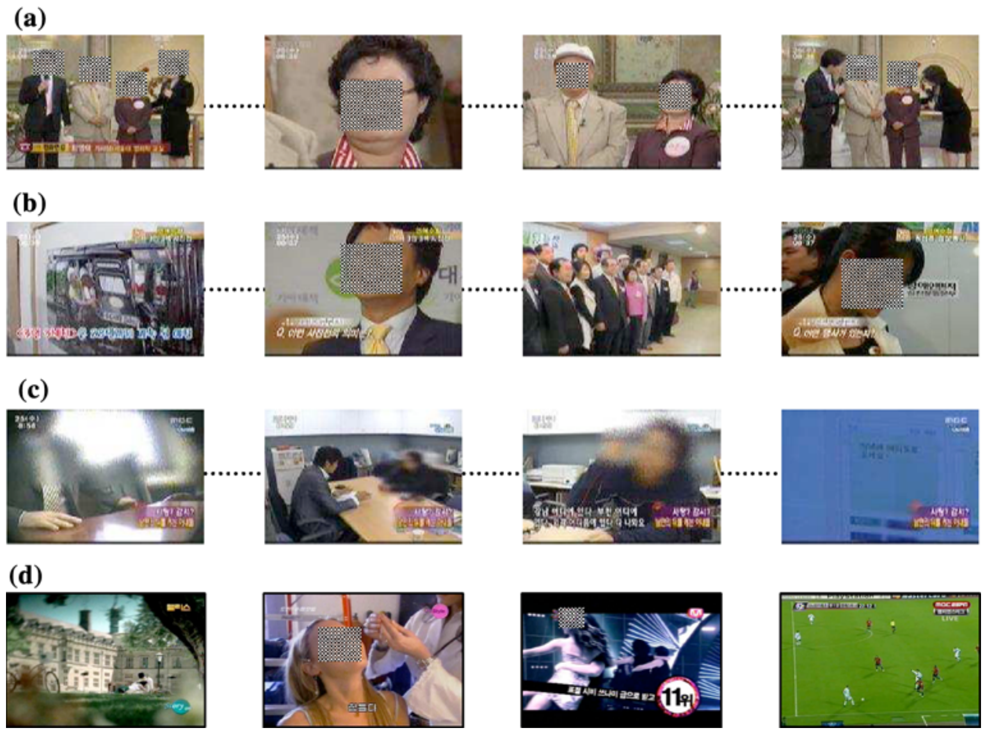

Fig. 8 Examples of video data. **a–c** Show some frames of various videos that contain semi-transparent logos. **d** shows some frames of various videos that contain opaque logos

Table 1 The logo recognition result

Method	Logo type	Total	Correct	False	Missing	Rate (%)
The proposed method						
$\tau = 0.50$	OP	150	142	0	8	82.33
	ST	150	105	0	45	
$\tau = 0.75$	OP	150	142	0	8	91.67
	ST	150	133	5	12	
$\tau = 1.00$	OP	150	140	2	8	87.33
	ST	150	122	17	11	
Cózar's method [1]	OP	150	128	8	14	82.33
	ST	150	119	12	19	

The OP and ST represent opaque logo and semi-transparent logo, respectively

and semi-transparent logos. The proposed method achieved 91.67 % accuracy at maximum, while Cózar's method achieved 82.33 % accuracy. Figure 9 shows more examples of logo detection for various videos.

4.2.2 Logo transition detection

We made experiments with 2 videos, which can be segmented into 19 and 25 sub-videos according to logos. The video lengths are 453 s (7 min and 33 s) and 947 s (15 min and 47 s), respectively. To evaluate the logo transition rate, the precision and recall are defined as follows:

$$\text{Precision }(\%) = \frac{N_{t_c}}{N_{t_c} + N_{t_f}} \times 100 \tag{12}$$

$$\text{Recall }(\%) = \frac{N_{t_c}}{N_t} \times 100 \tag{13}$$

where N_t, N_{t_c}, and N_{t_f} are the number of all logos, the number of correctly detected logo transitions, and the number of incorrectly detected logo transitions. We set τ as 0.75.

The detection result for logo transition is shown in Table 2. The proposed method has achieved good performance. Figure 10 shows the comparison results for logo

Fig. 9 Examples of logo detection results by the proposed method

Table 2 The logo transition detection result

Method	Video	Total	Correct	False	Miss	Precision (%)	Recall (%)
The proposed method	Video1	19	19	0	0	100	100
	Video2	25	24	2	1	92.3	96
Cózar's method [1]	Video1	19	14	0	5	100	73.7
	Video2	25	18	0	6	100	72.0

detection using Cózar's method (upper) and the proposed method (lower). Figure 10a shows that Cózar's method failed to extract logo regions accurately because of the black or white border line. However, the proposed method was successful, due to the feature extraction method based on determination of logo type. Figure 10b shows that Cózar's method failed to extract the logo, because the background of the sub-video was almost static. However, the proposed method could extract the logo via the proposed sub-video merging technique. Although logo detection failed in a sub-video, this could be recovered by the sub-video merging technique.

Figure 11 shows experimental results for logo recognition and sub-video merging steps. This example shows the effect of the sub-video merging technique. Figure 11a–c shows segmented video, logo recognition within each sub-video, and sub-videos merging. Figure 11d shows ground-truth.

Figure 11a shows the result for segmentation of the video based on shot boundary detection. According to the ground-truth (d), we can see some videos are divided into several sub-videos. For an example, the first video of the 'MBC' in Fig. 11d is divided into 4 sub-videos in Fig. 11a (sub-video index 1 ,2, 3, and 4).

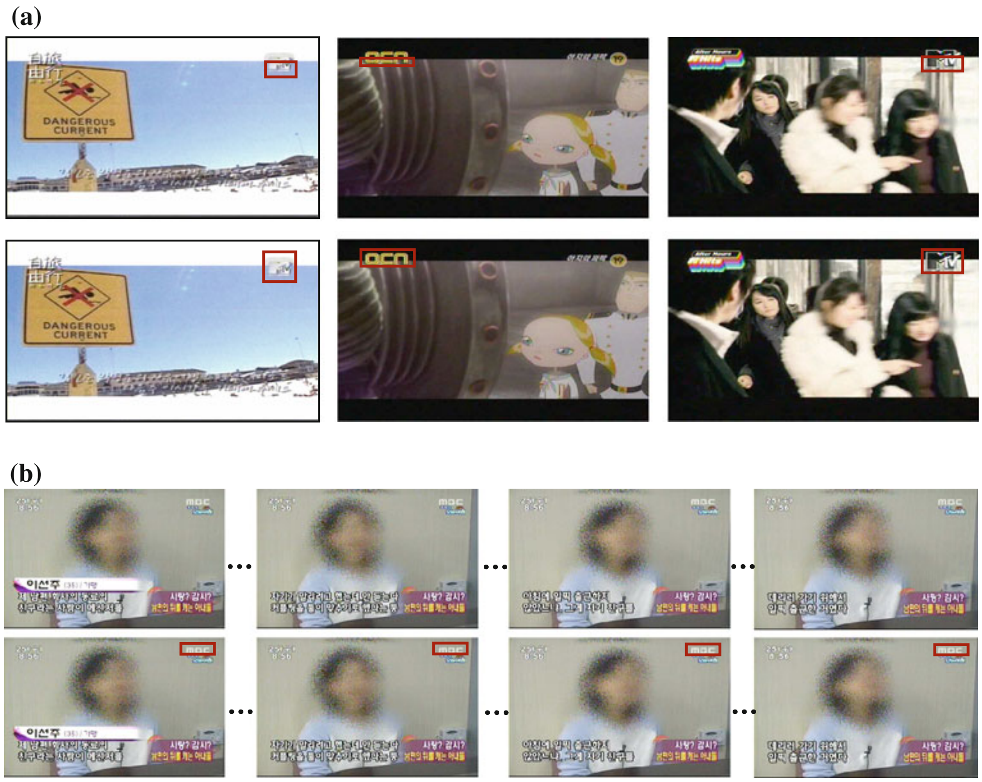

Fig. 10 The comparison of logo detection results by the Cózar's method (*upper*) and the proposed method (*lower*). **a** Logo region is almost static. **b** Whole region is static

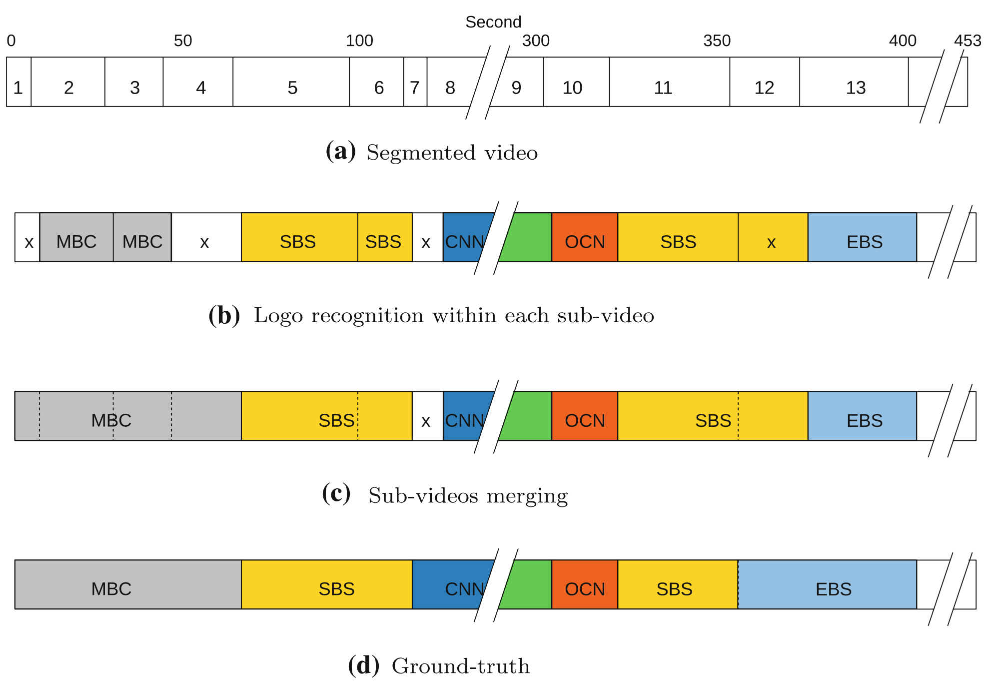

Fig. 11 Logo recognition result in each step: **a** video segmentation, **b** logo recognition within sub-video, **c** sub-videos merging. **d** Shows ground-truth. $1, 2, \ldots, 13$ in **a** represent sub-video index and *x* indicates that no logo is recognized in the sub-video

Figure 11b shows the logo recognition result of each sub-videos. In some sub-videos (index 1, 4, and 7), no logos were recognized because enough information was not available to recognize the logos. These occurred where the length of the sub-video is too short to recognize logo or the frames in the sub-videos are almost static.

Figure 11c shows the result of sub-video merging using the proposed merging method. The sub-videos 1 and 4 are merged with the sub-videos 2 and 3. It shows that the failure of the logo recognition in sub-videos was recovered by the merging method. However, there is a unrecognized sub-video (index 7). The reason was that the logo was not clearly seen due to the background colors which were almost same as the logos, and thus, during the proposed merging processes the sub-video merging did not give confident value to be merged.

There is a mis-merged sub-video (index 12). The SBS and EBS logos consist of three character images of ‘SBS’ and ‘EBS’, respectively, and two of the three characters are same, which are ‘B’ and ‘S’. Moreover, the two logos were transparent. It makes the logo recognition very hard. That is the reason the sub-video 12 was not merged correctly. However, overall, the merging method worked very good and resulted in very good result in logo transition detection as presented in Table 2.

We assumed that there is no false negative and according to our experiments using the shot boundary detection method, there was no false negative. However, if there are false negatives, there is a possibility that some sub-videos contain more than one logos. Therefore, we cannot be sure of the success of the logo recognition.

The proposed system consists of five steps as it is shown in Fig. 4. The time complexities of the video segmentation step and the logo feature extraction (steps 1 and 2) depend on the length of the input video. These steps can be run in on-line system that the algorithms process the input frames piece-by-piece in a serial fashion. The time complexities of the logo type determination and the recognition (steps 3 and 4) depend on the number of sub-videos which are the output of the video segmentation step. These steps also can be run in on-line system. The time complexity of the sub-video merging is dependent on the number of the sub-videos in which no logos were recognized. This step contains searching prior and subsequent sub-videos. Thus, this cannot be run in an on-line system. The steps 1, 2, 3, and 4 can be run in real-time if the CPU has enough computational power. However, the step 5 cannot be run in real-time, but can be run in delayed real-time. As a result, the average execution time of the steps 1–4 was 7 frames per second on a Windows system with a Core 2 Duo processor.

5 Conclusions and future research

We proposed an automatic logo transition detection method in a digital video content which may contain various logos. In video contents edited by users, there are logo transitions, and different types of logos are displayed. The proposed method uses a multi-stage process to determine the type of logos and detect logo transitions. Logo transition detection involves video segmentation via shot boundary detection. For sub-videos, candidate logo templates are extracted and it is determined whether candidate logo templates are opaque or semi-transparent logos. The method of determination is useful for effective feature extraction, prior to recognizing the corresponding logo. Also, in order to combine the segmented sub-videos, we proposed a method for merging them. Although logo detection sometimes fails for some sub-videos, it is compensated for by considering prior and subsequent sub-videos using the merging method. Experimental results showed that the proposed framework outperformed the previous method for various videos with various logos.

In redistributed videos the logo quality is often poor, since logo images on frames can be easily corrupted by video compression techniques. Although the videos we used were not compressed, most of videos in practice are compressed. It will be an interesting issue to deal with different compression methods on logo detection. In the future, we will consider the compression techniques of video. Also, we are going to extend this work to enable detection of animated logos and are researching on more sophisticated shot boundary detection method.

Acknowledgments This work was supported by the World Class University Program through the National Research Foundation of Korea funded by the Ministry of Education, Science, and Technology, under Grant R31-10008.

References

1. Cózar JR, Guil N, Gonzalez-Linares JM, Zapata EL, Izquierdo E (2007) Logotype detection to support semantic-based video annotation. Signal Process Image Commun 22(7):669–679
2. Yan WQ, Wang J, Kankanhalli M (2005) Yan WQ, Wang J, Kankanhalli M (2005) Automatic video logo detection and removal. Multimed Syst 10(5):379–391
3. Fang H, Jiang J, Feng Y (2006) A fuzzy logic approach for detection of video shot boundaries. Pattern Recognit 39(11): 2092–2100
4. Bai H, Hu W, Wang T, Tong X, Liu C, Zhang Y (2006) A novel sports video logo detector based on motion analysis. LNCS Neural Inf Process 4233:448–457
5. Kleban J, Xie X, Ma WY (2008) Spatial pyramid mining for logo detection in natural scenes. In: Proceedings of IEEE international conference on multimedia and expo, Hannover, Germany, pp. 1077–1080
6. Bagdanov A, Ballan L, Bertini M, Bimbo AD (2007) Trademark matching and retrieval in sports video databases. In: Proceedings of international workshop on multimedia information retrieval, Augsburg, Germany, pp. 79–86
7. Gao K, Lin S, Zhang Y, Tang S, Zhang D (2009) Logo detection based on spatial-spectral saliency and partial spatial context. In: Proceedings of IEEE international conference on Multimedia and Expo, New York, USA, pp. 322–329
8. Wolf C, Jolion JM (2003) Extraction and recognition of artificial text in multimedia documents. Pattern Anal Appl 6(4):309–326

9. Byun HR, Roh MC, Kim KC, Choi YW, Lee SW (2002) Scene text extraction in complex images. LNCS Doc Anal Syst 2423:329–340
10. Albiol A, Fulla MJ, Albiol A, Torres L (2004) Commercials detection using HMMs. In: Proceedings of IEEE international workshop on image analysis for multimedia interactive services, Lisboa, Portugal
11. Yeh JH, Chen JC, Kuo JH, Wu JL (2005) TV commercial detection in news program videos. In: Proceedings of IEEE international symposium on circuits and systems, Kobe, Japan, vol 5, pp 4594–4597
12. Albiol A, Fulla MJ, Albiol A, Torres L (2004) Detection of TV commercials. In: Proceedings of IEEE international conference on acoustics, speech, and signal processing, Montreal, Canada, vol 3, pp 541–544
13. Tong X, Lu H, Liu Q, Jin H (2004) Replay detection in broadcasting sports video. In: Proceedings of 3rd international conference on image and graphics, Hong Kong, China, pp 337–340
14. Wang J, Liu Q, Duan L, Lu H, Xu C (2007) Automatic TV logo detection, tracking and removal in broadcast. LNCS Adv Multimed Model 4352:63–72
15. Meisinger K, Troeger T, Zeller M, Kaup A (2005) Automatic TV logo removal using statistical logo detection and frequency selective inpainting. In: Proceedings of 13th European signal processing conference, Antalya, Turkey
16. Yan WQ, Wang J, Kankanhalli MS (2005) Automatic video logo detection and removal. Multimed Syst 10(5):379–391
17. Cózar JR , Guil N, Gonzalez-Linares JM, Zapata EL (2006) Video cataloging based on robust logotype detection. In: Proceedings of 2006 IEEE international conference on image processing, Atlanta, USA, pp 3217–3220
18. Seeber B, Yager N, Amin A (2007) Real-time detection of semi-transparent watermarks in decompressed video. In: Proceedings of IEEE workshop on applications of computer vision, Texas, USA, pp 49–54
19. Duffner S, C. Garcia (2006) A neural scheme for robust detection of transparent logos in TV programs. LNCS Artif Neural Netw 4132:14–23
20. Boussaid L, Mtibaa A, Abid M, Paindavoine M (2007) A real-time shot cut detector: hardware implementation. Comput Stand Interfaces 29(3):335–342