J. Vis. Commun. Image R. 21 (2010) 838–849

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Video copy detection using multiple visual cues and MPEG-7 descriptors

Onur Küçüktunç [a,*,1], Muhammet Baştan [b], Uğur Güdükbay [b], Özgür Ulusoy [b]

[a] Department of Computer Science and Engineering, The Ohio State University, 43210 OH, United States
[b] Department of Computer Engineering, Bilkent University, 06800 Ankara, Turkey

ARTICLE INFO

Article history:
Received 1 December 2009
Accepted 29 June 2010
Available online 13 July 2010

Keywords:
Content-based copy detection
Video copy detection
Visual ques
MPEG-7
Activity matching
Face detection
Time series analysis
Subsequence matching

ABSTRACT

We propose a video copy detection framework that detects copy segments by fusing the results of three different techniques: facial shot matching, activity subsequence matching, and non-facial shot matching using low-level features. In facial shot matching part, a high-level face detector identifies facial frames/shots in a video clip. Matching faces with extended body regions gives the flexibility to discriminate the same person (e.g., an anchor man or a political leader) in different events or scenes. In activity subsequence matching part, a spatio-temporal sequence matching technique is employed to match video clips/segments that are similar in terms of activity. Lastly, the non-facial shots are matched using low-level MPEG-7 descriptors and dynamic-weighted feature similarity calculation. The proposed framework is tested on the query and reference dataset of CBCD task of TRECVID 2008. Our results are compared with the results of top-8 most successful techniques submitted to this task. Promising results are obtained in terms of both effectiveness and efficiency.

Published by Elsevier Inc.

1. Introduction

With the rapid development of multimedia technologies and media streaming, copyrighted materials become easily copied, stored, and distributed over the Internet. This situation, aside from enabling users to access information easily, causes huge piracy issues.

One possible solution to identify copyrighted media is watermarking. Digital watermarking [1] was proposed for copyright protection and fingerprinting. The basic idea is to embed an information into the signal of the media (audio, video, or photo). Some watermarks are visible (e.g., text or logo of the producer or broadcaster), while others are hidden in the signal, which cannot be perceived by human eye. Today all DVD movies, video games, audio CDs, etc. have fingerprints that prove the ownership of the material.

As a disadvantage, watermarks are generally fragile to visual transformations (e.g., re-encoding, change of the resolution/bit rate). For example, hidden data embedded on a movie will probably be lost when the clip is compressed and uploaded to a video sharing web site Fig. 1. Besides, temporal information of the video segments (e.g., frame number, time-code) are also important in some applications. Watermarking technique is not designed to be used for video retrieval by querying with a sample video clip.

Content-based copy detection (CBCD) is introduced as an alternative, or in fact, a complementary research field to watermarking approach. The main idea of CBCD is that the media visually contains enough information for detecting copies [2]. Therefore, the problem of content-based copy detection is considered as video similarity detection by using the visual similarities of video clips.

In addition to copyright protection issues, there are other applications of video copy detection. For instance, it allows the tracking of news stories across different sources [3,4], measuring the novelty [5], tracking of known or repeated sequences [6], and identification of commercials [7]. Video copy detection techniques also enhance the indexing, searching, and retrieval capabilities of a multimedia database.

1.1. Challenges

Video copy detection is a challenging problem in computer vision due to the following reasons. First of all, the problem domain is exceptionally wide. Depending on the purpose of a video copy detection system, different solutions can be applied. For example, a simple frame-based color histogram similarity approach could be enough for detecting exact duplicates of video segments or identifying commercial breaks. On the other hand, matching news stories across different channels (camera viewpoints) is a totally different problem, and will probably require interest point matching techniques. Therefore, no general solution can be proposed to video copy detection problem. Secondly, the problem space is extremely large, which often requires real-time solutions. For the

* Corresponding author.
E-mail addresses: kucuktunc.1@osu.edu (O. Küçüktunç), bastan@cs.bilkent.edu.tr (M. Baştan), gudukbay@cs.bilkent.edu.tr (U. Güdükbay), oulusoy@cs.bilkent.edu.tr (Ö. Ulusoy).
[1] This work was done while the author was at Bilkent University.

1047-3203/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jvcir.2010.07.001

Fig. 1. Original (first row) and transformed frames (second row). The applied transformations include letter-box, strong re-encoding, mirror, noise addition, picture-in-picture, and text insertion.

case of YouTube, the system needs to process 20 h of uploaded video content per second to find an exact or near-duplicate segment of a copyrighted material [8].

Beginning in 2008, TREC Video Retrieval Evaluation (TRECVID) [9] introduced CBCD as a new task to evaluate. The aim of the task is to determine the location of each query video in the test collection accompanied with a decision score. A copy is defined as a segment of video derived from another video, usually by means of various transformations, such as addition, deletion, modification (of aspect ratio, color, contrast, encoding), camcording, so that the queries are constructed according to this definition.

Each query video in TRECVID CBCD task is constructed by taking a segment from the test collection, transforming and/or embedding it into some other video segment, and finally applying one or more transformations to the entire query segment [10]. The transformations used in CBCD task [11] cover most of the video modifications in daily life (see Table 1).

1.2. Related work

There are notable works on video similarity detection in the literature. An early method based on color histogram intersection is proposed by Satoh [12]. Yeh and Cheng [13] use a method that partitions the image into 4 regions, and extracts a Markov stationary feature (MSF)-extended HSV color histogram. Basharat et al. [14] present a video-matching framework using spatio-temporal segmentation. A set of features (color, texture, motion, and SIFT [15] descriptors) is extracted from each segment, and the similarity between two videos is computed with a bipartite graph and Earth Mover's Distance (EMD). Wu et al. [16] propose that specific types of visual features (i.e., texture, intensity, motion, gradient, frequency, interest point) should be used for different types of transformations by a video near-duplicate video matching system.

Table 1
List of transformations used in the CBCD task.

#	Transformation details
T1	Camcording
T2	Picture-in-picture Type 1
T3	Insertion of patterns (15 different patterns)
T4	Strong re-encoding (change of resolution, bitrate)
T5	Change of gamma
T6	Combination of 3 transformations amongst: blur, gamma, frame dropping, contrast, compression, ratio, noise (A)
T7	Combination of 5 transformations amongst (A)
T8	Combination of 3 transformations amongst: crop, shift, contrast, caption, flip, insertion of pattern, picture-in-picture Type 2 (original video is behind) (B)
T9	Combination of 5 transformations amongst (B)
T10	Combination of 5 transformations amongst all the transformations from 1 to 9

The methods based on points of interest and their trajectories are popular in this field. Joly et al. present a technique for content-based video identification based on local fingerprints [17]. Local fingerprints are extracted around interest points detected with Harris detector, and matched with an approximate nearest neighbors search. In [18,19], the same authors focus on the retrieval process of the proposed CBCD scheme by proposing *statistical similarity search* (S^3) as a new approximate search paradigm. In [20], Joly et al. present distortion-based probabilistic approximate similarity search technique (DPS^2) to speed-up conventional techniques like range queries and sequential scan method in a content-based copy retrieval framework. Zhao et al. [21] extract PCA-SIFT descriptors for matching with approximate nearest neighbor search, and train SVMs to learn matching patterns. Law-To et al. present a video indexing approach using the trajectories of points of interest along the video sequence [22,23]. They compute temporal contextual information from local descriptors of interest points, and use this information in a voting function for matching video segments. Ren et al. [24] employ a similar technique by taking into account spatial and temporal changes of visual words constructed by SIFT descriptors and bag-of-words approach. Willems et al. [25] propose a video copy detection method based on efficiently matching local spatio-temporal feature points with a disk-based indexing scheme. In general, extracting and matching points of interest are costly operations in terms of computation time.

There are also promising copy detection techniques based on the similarity of temporal activities of video clips. Mohan [26] presents a video sequence matching technique that partitions each frame into 3×3 image and computes its ordinal measure to form a fingerprint. The sequences of fingerprints are compared for video similarity matching. Kim and Vasudev [27] use ordinal measures of 2×2 partitioned image and consider the results of various display format conversions, e.g., letter-box, pillar-box.

Some video similarity detection methods take the advantage of visual features that can be directly extracted from compressed videos. Ardizzone et al. [28] use MPEG motion vectors as an alternative to optical flows, and show that the motion-based video indexing method they propose does not require a full

decomposition of the video, and thus, it is computationally efficient. Bertini et al. [29] present a clip-matching algorithm that use video fingerprint based on standard MPEG-7 descriptors. An effective combination of color layout descriptor (CLD), scalable color descriptor (SCD), and edge histogram descriptor (EHD) forms the fingerprint. Fingerprints are extracted from each clip, and they are compared using an edit distance. Sarkar et al. [30] use CLD as video fingerprints and propose a non-metric distance measure to efficiently search for matching videos in high-dimensional space.

Hampapur and Bolle [2] made a comparative analysis of color histogram-based and edge-based methods for detecting video copies. Another study by Hampapur et al. [31] compares motion direction, ordinal intensity signature, and color histogram signature matching techniques. As a result of this study, they conclude that the techniques using ordinal features outperform the others. State-of-the-art copy detection techniques are evaluated in the comparative study by Law-To et al. [32]. Compared descriptors are categorized into 2 groups: global and local. Global descriptors use techniques based on the temporal activity, spatial distribution and spatio-temporal distribution. Local descriptors compared in their study are based on extracting Harris interest points for keyframes with high global intensity of motion (AJ), for every frame (*ViCopT*), and interest points where image values have significant local variations in both space and time. It is stated that no single technique is optimal for all applications; but ordinal temporal measure is very efficient for small transformations.

1.3. Motivation and contributions

Best results for a complete CBCD framework have been achieved by interest point-based methods so far; yet it is known that the detection and matching of interest points is computationally inefficient (for both off-line and online stages). There are other works focusing on video similarity detection based on (1) global and local visual features and their combinations, (2) temporal activities, and (3) some novel techniques that use face detection for news clip similarity detection in the literature (see Section 1.2); however, as discussed in [16], a single method is not sufficient to detect copies/near-duplicates of video clips with different types of transformations.

We propose a CBCD framework that uses the results of three techniques using multiple visual cues and MPEG-7 descriptors: facial shot detection, spatio-temporal activity matching, and low-level visual feature similarity. The proposed framework combines the advantages of each component, and is still efficient in terms of computation time compared to the interest point-based methods.

2. Proposed CBCD framework

The proposed CBCD framework comprises of two main components: an off-line (preprocessing) and an online stage. The overview of the overall CBCD framework is shown in Fig. 2. MPEG-7 visual descriptors (SCD, CLD, CSD, HTD, EHD) are defined in Section 2.3.

2.1. Off-line (Preprocessing) stage

As in many content-based retrieval system, reference content is indexed with a preprocessing (off-line) stage in our CBCD framework. Activity sequences, shot-boundaries, facial shots, and fingerprints of the keyframes are extracted for each reference video and this information is stored in a database.

In activity sequence extraction part, intensity averages of each frame in a reference video are extracted and stored. In contrast with the prior works [27,26], we preferred using the numerical intensity averages of partitions instead of their ordinal measures. The reason is that when the length of the query video is small (query videos used in TRECVID are between 3 s and 3 min [33]), there might be more than one subsequence that have very similar ordinal values.

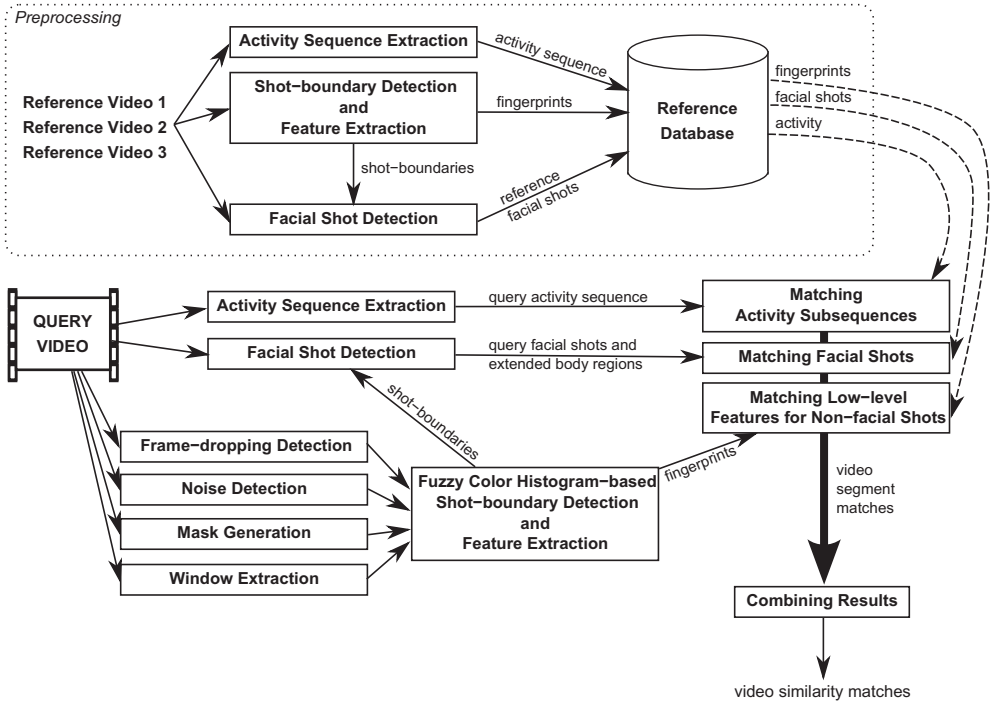

Fig. 2. The overview of our CBCD framework.

Then, shot-boundaries of reference videos are detected by comparing the histograms of successive frames in RGB color space. The mentioned five low-level MPEG-7 visual features are extracted from the median frame of each shot. In order to accelerate the similarity search, we use k–d trees to store the descriptors (SCD, CSD, and EHD) that use l_1-norm as the similarity measure, and ANN [34] library to search.

The last part of the preprocessing stage is the detection of facial shots. The details of face detection and extraction will be given in Section 3. Similar to low-level feature extraction, color descriptors (i.e., CSD, SCD, and CLD) of each face are also extracted and stored in the reference database.

2.2. On-line stage

At the beginning of the online stage, we try to detect some specific transformations in a query video, such as frame-droppings, noise, text/pattern additions, and picture-in-picture transformation windows. These modifications seriously affect the detection process. Briefly, frame-droppings are detected by thresholding the average intensity value of a frame, noise level is identified by comparing a frame with its Median filter-applied variant, and the picture-in-picture transformation windows are extracted by matching vertical lines in the image of standard deviation of pixel intensities (see Fig. 3). The details of these detectors are given in [35].

Shot-boundaries of query videos are detected using a fuzzy color histogram-based method proposed in [35]. After this point, the system tries to find the matching video segment in the reference database by comparing the facial shots, activities, and non-facial shots. Detection and matching with these methods are explained in the following sections.

Each method (facial shot matching, activity subsequence matching, and low-level feature matching) returns the best matches for all the queries. When combining the results, some of them point to the same reference video and similar temporal locations. These candidate results are merged and reported with the rest of the matching candidates with a decision score.

2.3. Used MPEG-7 visual descriptors

The following color and texture descriptors are used in our framework:

Scalable Color Descriptor (SCD) is a color histogram in the HSV color space, encoded by a Haar transform. In our method, we used 128 coefficients (histogram bins). l_1-norm based matching is used for comparing SCDs.

Color Layout Descriptor (CLD) is a compact and resolution-invariant color feature that efficiently represents spatial distribution of colors. Input image is divided into 8×8 blocks, transformed by discrete cosine transformation (DCT), and DCT coefficients for the luminance and the chrominance are extracted. For matching two CLDs, $\{DY, DCr, DCb\}$ and $\{DY', DCr', DCb'\}$, the following distance measure is used [36]:

$$D = \sqrt{\sum_i w_{yi}\left(DY_i - DY_i'\right)^2} + \sqrt{\sum_i w_{bi}\left(DCb_i - DCb_i'\right)^2} + \sqrt{\sum_i w_{ri}\left(DCr_i - DCr_i'\right)^2}. \quad (1)$$

Color Structure Descriptor (CSD) is a color feature descriptor that represents an image by both color distribution (similar to color histogram) and the local spatial structure of the color. An 8×8 element is used to embed color structure information into the descriptor. CSD uses the l_1-norm for matching as the similarity measure.

Homogeneous Texture Descriptor (HTD) represents region textures using the mean energy and energy deviation in 30 frequency channels. The similarity between a query image (TD_q) and a reference image (TD_{ref}) is measured by summing the weighted absolute difference between two sets of vectors:

$$D(TD_q, TD_{ref}) = \sum_k \left| \frac{TD_q(k) - TD_{ref}(k)}{\alpha(k)} \right|, \quad (2)$$

where the recommended normalization value $\alpha(k)$ is the standard deviation of all $TD_{ref}(k)$ values. For intensity invariant matching, the first component is not used.

Edge Histogram Descriptor (EHD) calculates spatial distribution of five types of edges. The image is divided into 4×4 subimages, and then the edges in 16 subimages are categorized into five types: vertical, horizontal, 45° diagonal, 135° diagonal, and non-directional edges. For matching edge histograms, global (h^g – 5 bins) and semi-global (h^s – 65 bins) edge distributions are calculated from the local histogram bins (h – 80 bins). We use l_1-norm for similarity matching.

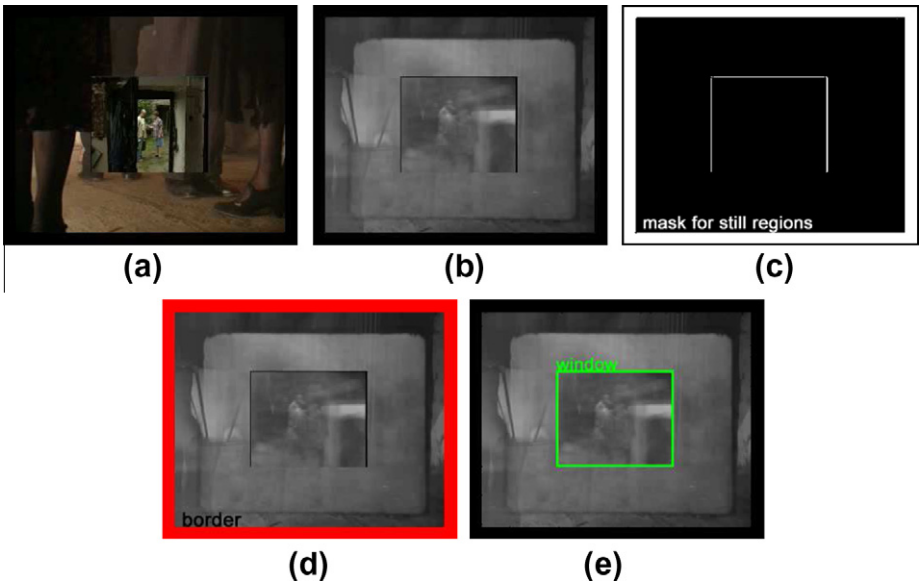

Fig. 3. Example query frame (a) and its detected still regions (c), borders (d), windows (e) using the standard deviation of pixel intensities throughout the video (b).

3. Detecting and matching facial shots

The first part of video similarity detection of our CBCD framework is to detect facial shots in the reference and query videos. Although the detection process for reference videos does not require any specific adjustment, we need to consider the video manipulations applied to query videos while detecting faces in query videos. For example, a Median filter is applied on noisy query frames, a smaller scale is selected for minimum face size within the window of picture-in-picture transformation, and so on. After obtaining the faces from detected facial shots, we extract visual features and match them to find the matching video segments.

We use an object detector, proposed by Viola and Jones [37], improved by Lienhart and Maydt [38], for detecting faces in video frames/shots. The face classifier (named as *cascade of boosted classifiers working with Haar-like features*) is trained with positive and negative instances. The responses of Haar-like features are extracted, and a decision tree-based classifier is trained for face samples. These features are specified by their shapes, positions within the region of interest, and scales.

In order to search for the face in the frame, the algorithm moves the search window across the frame while checking each location using the classifier. The scan procedure is performed several times at different window scales for finding the faces with different sizes. For this purpose, the classifier is designed in such a way that it can be easily resized. A binary decision is generated as a result of the classifier.

In our implementation, we use Canny edge detector to reject some image regions that contain very few or too much edges, and thus, cannot contain faces. The particular thresholds are tuned for face detection and the pruning speeds up the processing. The detector finds faces with at least 20×20 pixels and returns only the largest object (if any) in the image. For the windows of picture-in-picture transformation, we use a smaller scale (10×10 pixels), since the faces are generally half-size of the ones in the full frame.

The frontal face detector trained by Lienhart in OpenCV [39] tends to generate many false alarms. From our observations, these false alarms can survive very few frames. We modified the face detection algorithm by taking spatial and temporal information into account to eliminate false detections, so that face detection would work more stable than the original method.

The modified face detection algorithm works as follows. Each detected face is considered as a candidate. The candidate faces with

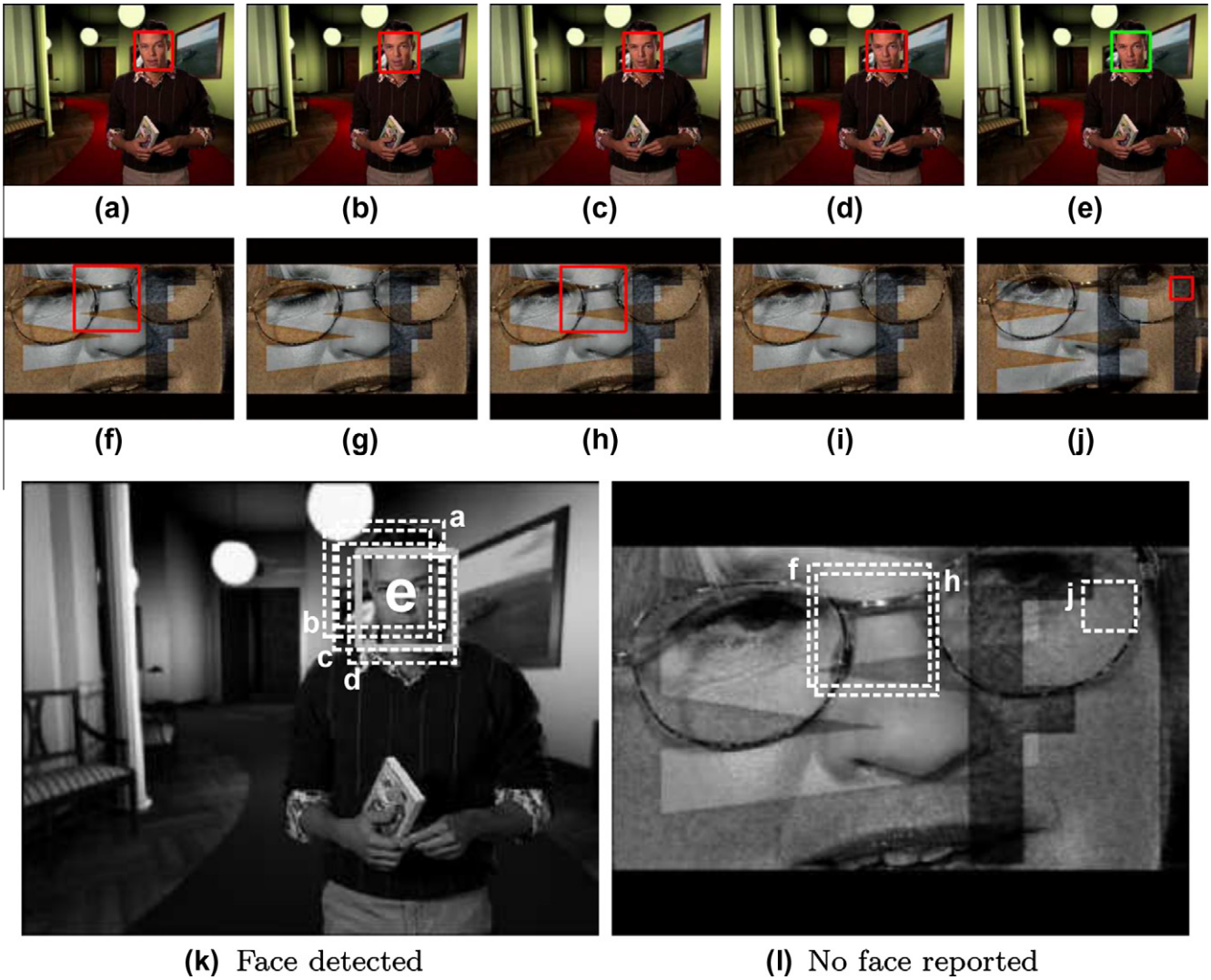

Fig. 4. Examples of face detection with false alarm elimination. Red rectangles are candidate (unstable) detections, while green ones are stable faces. Successive frames from different video clips (a–e and f–j) are shown in this figure. (k) and (l) are the spatial locations of detected faces over $f_s = 5$ frames for each clip. Since candidate faces are spatio-temporally stable during (k), we extract the last candidate face (e). However, in the second example, candidate faces are not consecutive and spatially stable. Therefore, no face is extracted from this sequence of video. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

a stable behavior both in time and location (space) are assumed to be the real faces. To accomplish the aforementioned stability, we track candidate faces in successive frames. If a candidate face appears at least $f_s = 5$ successive frames, the algorithm fuses multiple face detections and marks as a face of the related shot. Fig. 4 shows two face detections and how false alarms are eliminated.

Since our aim is not recognizing faces, label persons, etc., extracting only the faces does not seem to be an efficient way to match faces in video clips. Yet, face matching has some well-known drawbacks, such as sensitivity to pose and illumination changes. To overcome this problem, we employ the method proposed by Zhai and Shah [3]. Instead of extracting visual features from the face, we extend the detected region to cover the upper part of the body. Therefore, we can match the shots with the same person (e.g., an anchor man or a political leader) by considering the clothes or some background as well. Fig. 5 displays detected faces and their extended body regions.

After finding the facial shots from both the query and reference videos, some visual features are extracted from the extended body region image of each face. We preferred using color-based MPEG-7 descriptors (CSD, CLD, and SCD). Edge-based methods are excluded because of the distortions applied on query videos. We give the result with an edge-based descriptor (i.e., HTD) for comparison (see Table 2).

Table 2
Evaluation of facial shot matching method. Total number of copies to be detected for each transformation is 134.

	CSD	SCD	CLD	HTD	Proposed Method CSD + SCD + CLD
T1	0	0	18	1	18
T2	25	25	28	1	36
T3	38	40	42	25	48
T4	22	28	41	2	49
T5	24	21	49	23	54
T6	22	21	37	3	49
T7	12	14	47	5	51
T8	28	34	21	13	40
T9	26	25	10	5	32
T10	12	12	22	4	30

4. Subsequence matching of activity time-series

Spatio-temporal sequence matching is a technique that is robust to many distortions caused by digitization and encoding. In addition, it provides the precise temporal location of the matching video parts. These two features are crucial for a video copy detection system.

The notation used in this section is as follows: $V = \{V[0], \ldots, V[n-1]\}$ represents a video with n frames. $V[i] = \{V^1[i], V^2[i], V^3[i], V^4[i]\}$ denotes ith frame with 4 features, which are the average intensity values of 4 partitions (for *top-left*, *top-right*, *bottom-left*, and *bottom-right* regions). Then V^j represents a sequence of the jth partition. A video segment V with N frames is defined as $V[p: p+N-1]$, where the first frame is $V[p]$. The problem of subsequence matching of time-series can be defined as follows:

Problem. Given a query video V_Q with N frames, find the matching subset of the reference video V_R with M frames, if the dissimilarity between two video clips $D(V_Q, V_R)$ is less than a threshold ϵ.

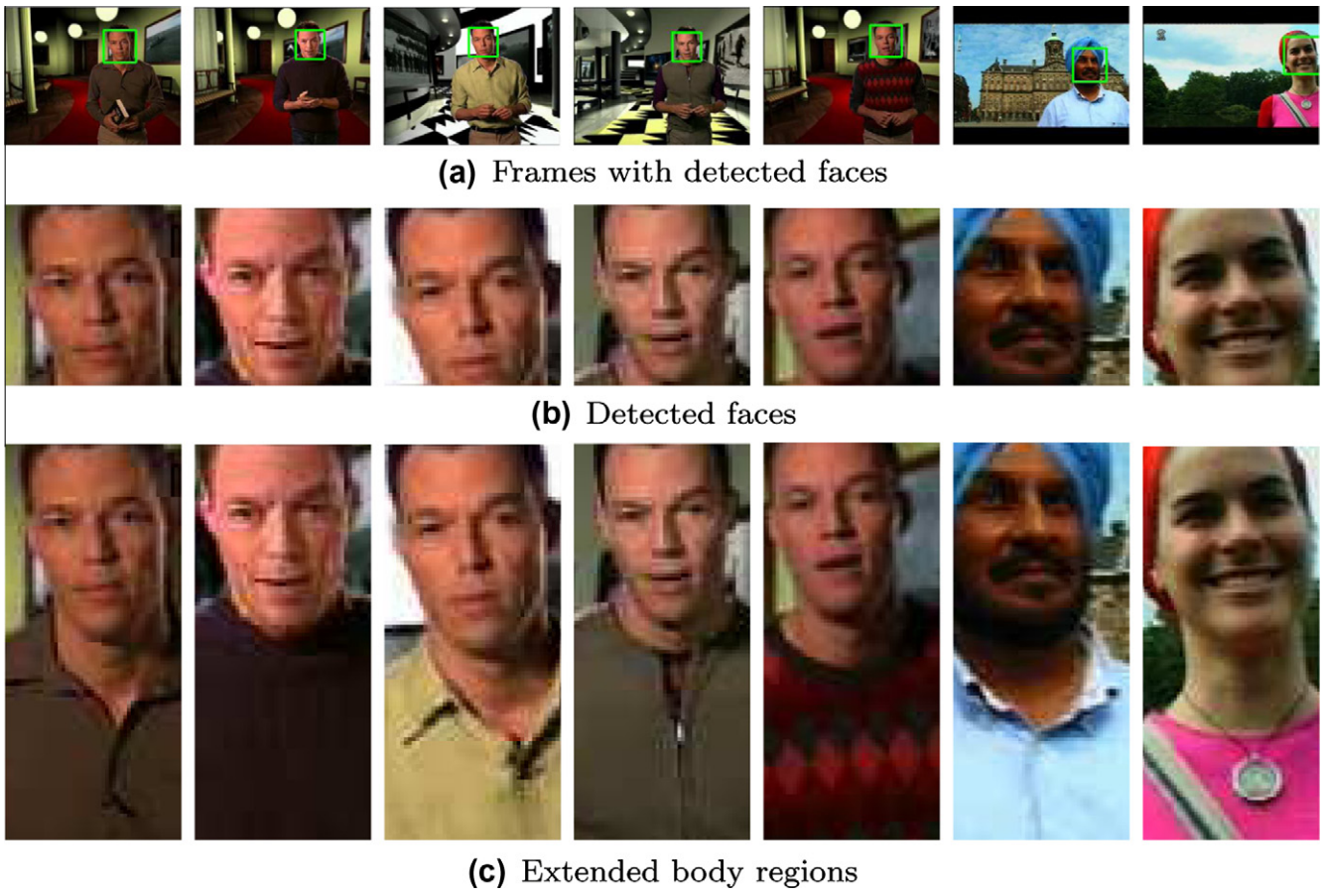

Fig. 5. Examples of extended body regions: first five examples are faces of the same person in different events/scenes. Because our goal is to match shots instead of faces, we use extended body regions (c). Solely the facial regions do not give discriminative visual features; the differences of clothing help us identify the same person in different scenes.

Fig. 6 represents a reference video with 200 frames, and a query video with 40 frames. Although they look quite different, we know that the query video V_Q is originated from the video segment $V_R[81:140]$.

Due to the manipulations in the query generation process (e.g., changing quality, gamma value, contrast), average intensity values of the query frames may be higher or lower than the original video. Therefore, we need to normalize average intensity values for both the reference and query videos (see Fig. 6). This is done using histogram equalization [27].

After this point, the problem becomes *matching time-series (signals) with different amplitudes*. To overcome the differences in the amplitudes of time-series, we define $\alpha_X[i]$ as the maximum distance of a partition intensity value to the center-point ($c = 128$) for the ith frame of video V_X, and $\beta_X[p: p+N-1]$ as the maximum value of $\alpha_X[i]$ for all frames of $V_X[p: p+N-1]$. Then β_X is calculated for all frames of V_X.

$$\alpha_X[i] = \max_j \left| V_X^j[i] - c \right|, \quad \text{where } j \in \{1,2,3,4\}, \tag{3}$$

$$\beta_X[p : p+N-1] = \max_i \alpha_X[i], \quad \text{where } i \in [p, p+N-1], \tag{4}$$

$$\beta_X = \beta_X[1 : M]. \tag{5}$$

By using α and β functions, we calculate the dissimilarity between a query video and a reference video segment as:

$$D(V_Q, V_R[p : p+N-1]) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{4}\left|\overline{V_Q^j[i]} - V_R^j[p+i]\right|}{N}, \tag{6}$$

$$\overline{V_Q^j[i]} = \frac{V_Q^j[i] - c}{\beta_Q / \beta_R[p : p+N-1]} + c. \tag{7}$$

Therefore, the dissimilarity between a query and a reference video can be defined as the minimum of video segment dissimilarities for $p \in [1, M-N]$:

$$D(V_Q, V_R) = \min_p \frac{D(V_Q, V_R[p : p+N-1])}{N}. \tag{8}$$

If the dissimilarity $D(V_Q, V_R)$ is less than a pre-specified threshold (ϵ), we report that the query video V_Q can be a copy of V_R starting from the frame number p with a decision score of $1 - D(V_Q, V_R)$. The threshold depends on the detected transformations applied to the video. If the query video has noise, or picture-in-picture transformation, the dissimilarity values would be higher. Therefore, we increase the decision threshold ϵ in such cases. Fig. 7 shows the activity series of four query videos originated from the same reference video segment.

5. Non-facial shot matching with low-level visual features

Selecting representative frames, namely *keyframes*, is a common approach for reducing the amount of video data to store and index

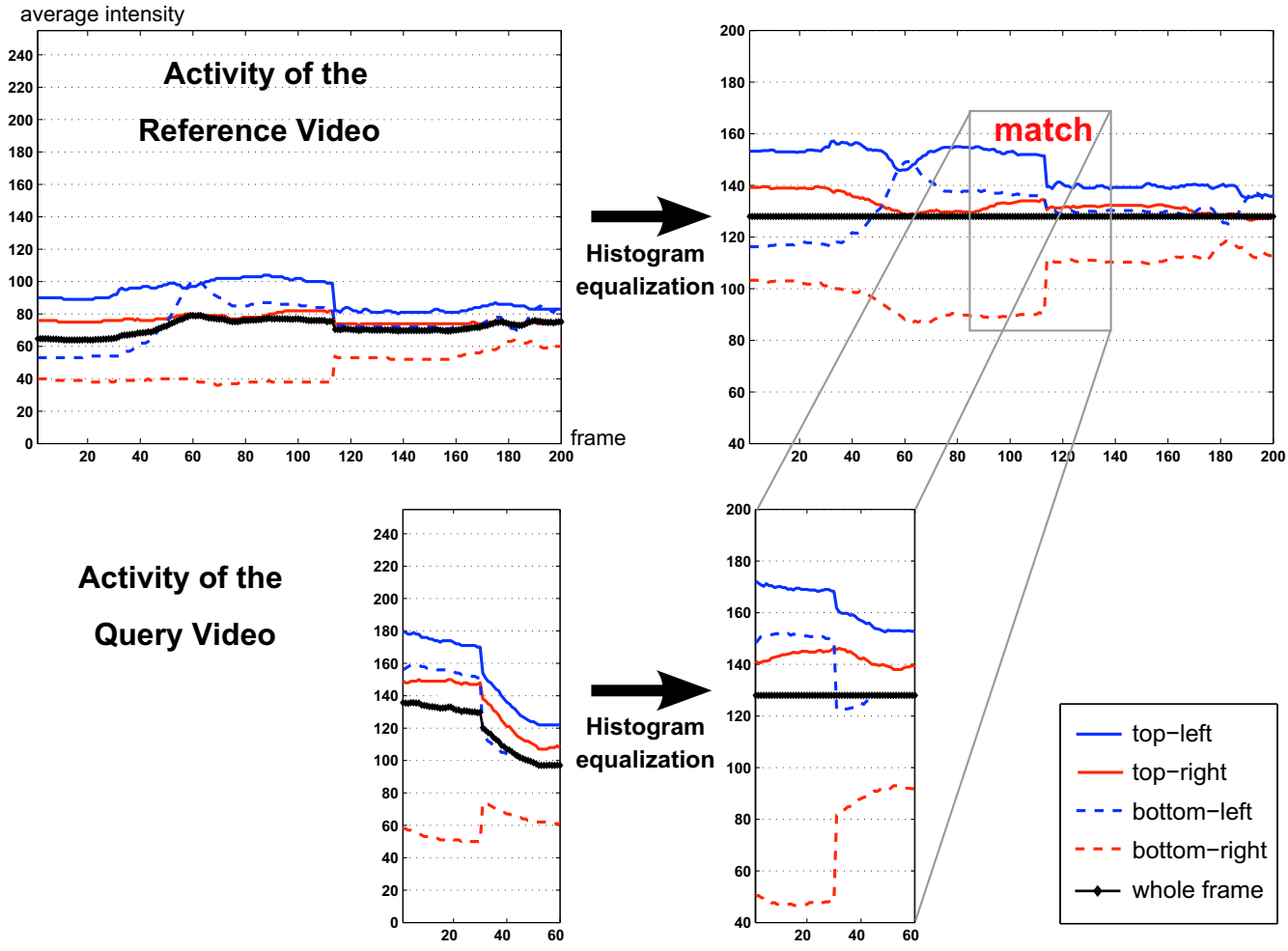

Fig. 6. Average intensity values before and after histogram equalization for the reference and query videos.

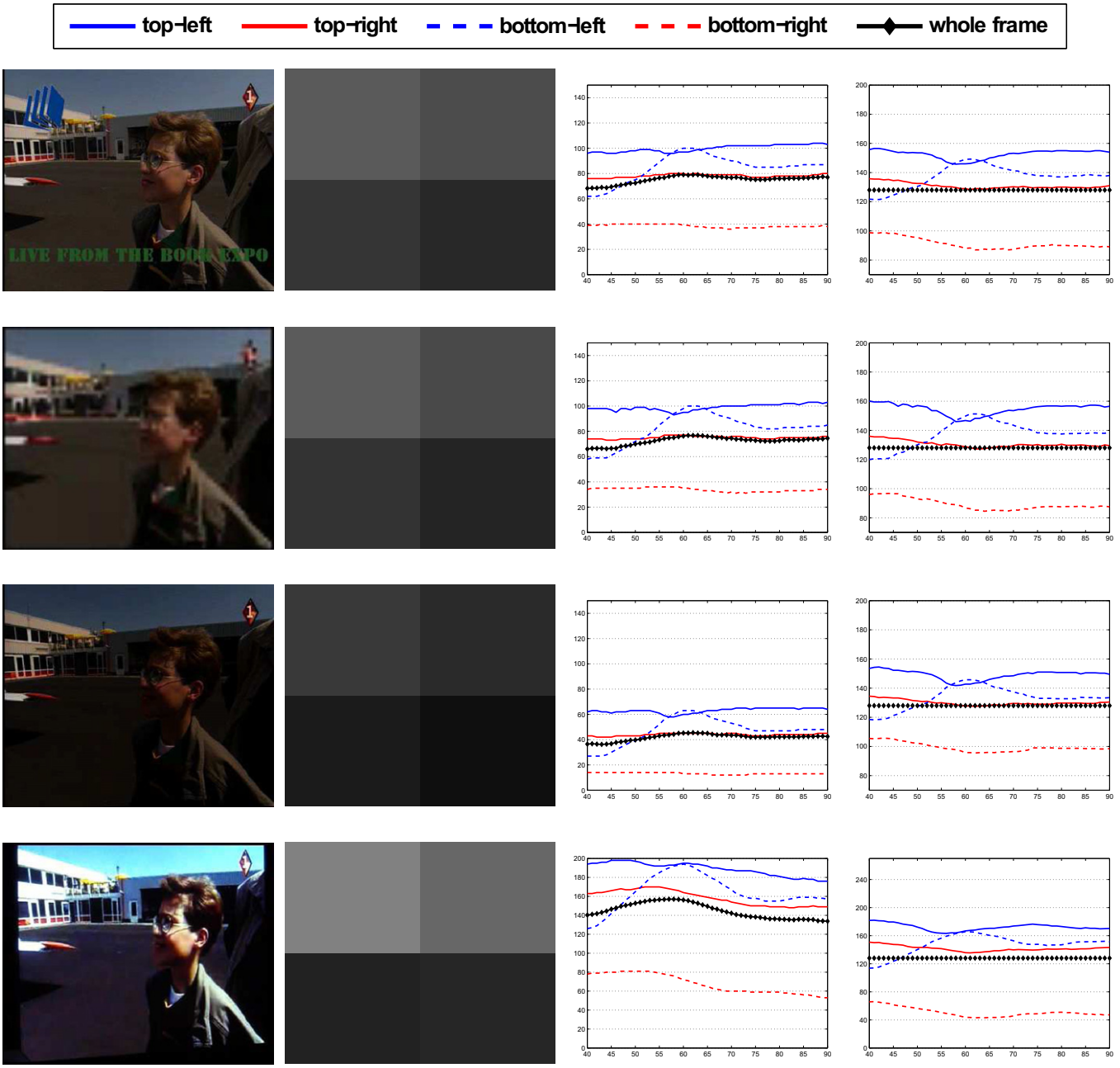

Fig. 7. Generation of the activity sequence: frames from query videos that correspond to the same reference video (first column), average intensity values for 2×2 partitions (second column), spatio-temporal activities of frames (third column), and normalized spatio-temporal activities of frames (fourth column). The first query video is very similar to the original video, except for the logo insertion. The second and the third videos have gamma change and strong re-encoding transformations. The fourth video is recorded with a camcorder. Although the spatio-temporal activities and average intensity values are very different, their normalized intensity sequences are close to each other.

for efficient content-based search. The third part of the proposed framework consists of extracting low-level visual features from the reference and query videos, and keyframe-based matching of video segments. If a shot is marked as facial (i.e., a face was detected in one of the frames of this shot), facial shot matching technique handles detecting the copies. In contrast with facial shot matching part, low-level feature matching uses low-level color and texture information extracted from the whole frame.

Extracting features from query keyframes requires a mask for picture-in-picture transformation window (if any), and the still regions. After discarding patterns, texts, and other inserted videos from the query keyframes, the rest of the frame represents the original content better. The visual features of the keyframes of reference videos are already computed and stored in a structure in the preprocessing (off-line) stage. The overview of low-level feature matching part is shown in Fig. 8.

A common approach for visual feature weighting is to assign fixed weights to each visual feature, as used in [29]. However, in video copy detection, some copies can only be identified by edge-based feature similarity, while others may better respond to color layout based similarity. This is simply a result of various transformations applied on query videos. If noise is added to video, we can use color structure information. If there is a change in the color content (e.g., by camcording, change of gamma), edge-based comparisons are likely to give better matches. As a result, an automatic copy detection system cannot decide which visual feature is

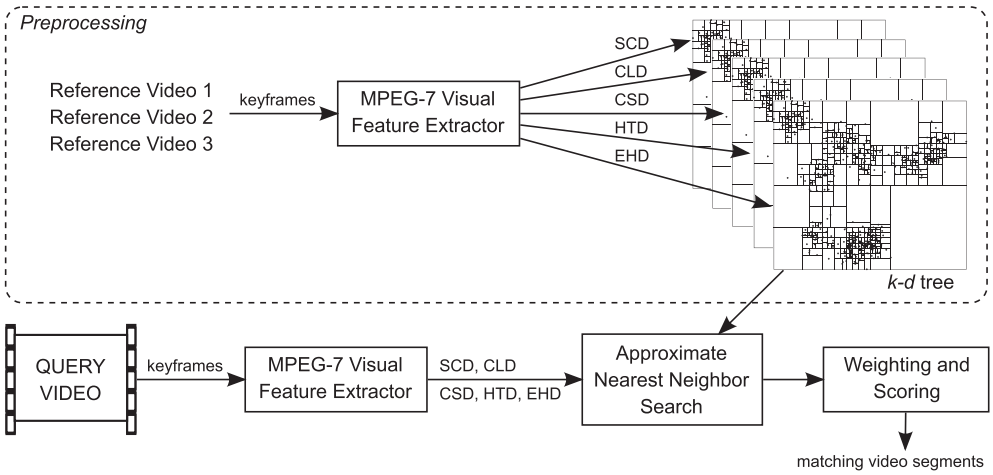

Fig. 8. The overview of the low-level feature matching algorithm.

appropriate for matching a query video. Most of the content-based image/video retrieval systems prefer using fixed weights for visual features, or simply take the average of the similarities of different features.

Problem. Given a query keyframe q with visual features f_q, find the matching reference keyframe r_m from the feature database f_R.

Our solution is to use a *dynamic-weighted feature similarity calculation* based on the success rate of the visual similarities of different features. Inspired from interest point matching techniques, where two closest matches are compared to each other, we define the success rate (weight) of a descriptor as the ratio of similarity values of the most similar match to the 5th one. So the weights for each visual feature are calculated for each query keyframe separately.

The motivation behind the dynamic-weighted feature similarity calculation is that some images are better matched with color information, while texture or edge information may be more appropriate for the others. For example, finding a match for an outdoor scene with a lake and sunset could be easier with CLD descriptor because the spatial distribution of the colors is important. On the other hand, edge-based EHD descriptor for a similar (or transformed) image can generate a fairly dissimilar feature vector to the original one. If we can find a good match, there will be a gap between the dissimilarity of this match and the matches that comes after. Otherwise, the dissimilarity values will be higher and the gaps will be smaller.

The jth most similar visual feature to f_q^i is found as $f_{r,j}^i$ in the reference database of features f_R^i by k-nearest neighbor search. We calculate the dynamic-weights of each visual feature i (i.e., CSD, CLD, SCD, EHD, and HTD) for the query keyframe q with:

$$\omega_i(f_q) = 1 - \frac{D_i\left(f_q^i, f_{r,1}^i\right)}{D_i\left(f_q^i, f_{r,5}^i\right)}. \tag{9}$$

By using the dynamic-weights, we find the most similar reference keyframe r_m by minimizing the combined dissimilarity:

$$r_m = \arg\min_r \frac{\sum_i \omega_i(f_q) \times (1 - D_i(f_q, f_r))}{\sum_i \omega_i(f_q)}. \tag{10}$$

Recall that SCD, CSD, EHD use l_1-norm, and D_{CLD} and D_{HTD} are given in Eqs. (1) and (2). Keyframe-based similarities are calculated with dynamic-weighted MPEG-7 visual features. The most similar and most voted matching reference videos are reported as copy candidates.

6. Experiments

6.1. TRECVID CBCD task dataset

The reference dataset consists of approximately 100 h of Sound & Vision data used as training and test videos for TRECVID 2007 search and HLF tasks, plus another 100 h of Sound & Vision data prepared for TRECVID 2008 search and HLF tasks. In total, there are 438 reference video files.

The query dataset prepared for TRECVID 2008 CBCD task is constructed using ~200 h of reference videos and videos not in the reference database (to test false positive rate). The 2007 BBC rushes video was used as non-reference data. Some of the queries are composed of a segment of reference videos, while some may contain no reference video segments. 67 video segments are prepared for each type. By applying 10 different transformations (see Table 1) to all generated videos, final query videos are generated. As a result, there are total of 2010 MPEG-1 videos (34.17 GB), which is about 80 h of video segments with various transformations applied.

6.2. Evaluation of facial shot matching

The color-based MPEG-7 descriptors used in facial shot matching method are color structure (CSD), scalable color (SCD), and color layout (CLD) descriptors. The number of correct detections obtained using each descriptor is listed in Table 2. Although the color descriptors are generally illumination-dependent feature vectors; we implement the similarity function of CLD illumination-invariant by ignoring DC components of the descriptor. Out of 9612 query faces and 32,597 reference faces, our method successfully retrieved a total of 407 copies, which corresponds to ~30% of the copies.

6.3. Evaluation of activity subsequence matching

Subsequence matching of activity time-series method is evaluated for the CBCD task of TRECVID 2008. Among 2010 query videos, 1340 of them are originated from a reference video. CBCD evaluation software generates the analysis reports for each transformation type. The objective of the task is to detect all 134 copies with the correct reference video and temporal locations.

Our activity matching method consists of three parts: detecting full frame copies, detecting copies of foreground videos generated

Table 3
Evaluation of activity subsequence matching method. Total number of copies to be detected for each transformation is 134.

	Baseline Ordinal	Normal	Window	Flip	Proposed Method Normal + Window + Flip
T1	52	55	2	–	55
T2	2	2	36	–	37
T3	42	46	–	1	47
T4	74	76	–	–	76
T5	67	70	–	–	70
T6	73	74	–	–	74
T7	65	62	–	2	63
T8	11	17	1	22	40
T9	2	2	–	20	22
T10	14	16	4	7	27

Table 4
Evaluation of each proposed method and their combinations. Total number of copies that can be detected for each transformation type is 134. FSM: facial shot, ASM: activity subsequence, LFM: low-level feature matching.

	FSM	ASM	LFM	FSM ASM	FSM LFM	ASM LFM	Combined	
							All	Hit (%)
T1	18	55	51	61	59	79	82	61.20
T2	36	37	1	60	37	37	60	44.78
T3	48	47	97	68	104	109	113	84.33
T4	49	76	113	95	116	125	126	94.03
T5	54	70	114	98	119	123	127	94.78
T6	49	74	102	89	109	118	122	91.05
T7	51	63	64	85	85	92	103	76.87
T8	40	40	75	62	92	91	103	76.87
T9	32	22	37	47	60	52	70	52.24
T10	30	27	27	47	47	45	58	43.29
All	407	511	681	712	828	871	964	71.94

with picture-in-picture transformation (T2), and flip transformation (T8–T9). We present the experimental results of these parts separately. Subsequence matching based on ordinal measure is taken as the baseline for our comparison. The results in terms of the number of correct detections for each transformation type are given in Table 3.

It is seen from the results that considering the activities of picture-in-picture transformation windows (for T2), and matching with the mirror of each query video (for T8 and T9) increase the accuracy of the proposed method.

6.4. Evaluation of low-level feature matching

Non-facial shot matching with low-level visual features is the most successful matching part of the system for the transformations of text/logo insertion (T3), strong re-encoding (T4), and gamma change (T5) (see Table 4, LFM column).

Although the matching resulted in low correct detection rates for camcording (T1), picture-in-picture transformation (T2), and complex transformations (T9–T10); shot matching with low-level features has a huge efficiency for detecting video copies. We were able to represent ~200 h of reference videos with only 87,598 keyframes.

6.5. Combined results

The number of correct detections for each method and the combined results are compared in Table 4. It should be noted that there are some copies detected by more than one method. The overall correct detections are not the sum, but the union of the correctly detected query videos.

The results given in Table 4 show that our video copy detection framework achieves high correct detection rates for the transformations T3, T4, T5, and T6, mostly because of the frame-dropping detection, mask generation, noise detection, and border detection parts of the method. Similarly, we obtained promising results for moderately complex transformations T7 and T8.

The contributions of each method (FSM, ASM, and LFM) to the final results are shown in Fig. 9. We can conclude that the method of low-level feature matching has the most contribution for the transformations with high overall accuracy (over 75%: T3, T4, T5, T6, T7, T8). However, we cannot ignore the benefits of FSM and ASM, especially for the complex transformations that include picture-in-picture (T2, T9, T10).

Detecting copies of query videos with camcording (T1) and picture-in-picture transformation (T2) can be improved as a future work. Currently, low-level feature matching method only works for the whole query frame; however, it can be modified in a way that the visual features of the picture-in-picture transformation window can be extracted and compared with the reference features.

6.6. Comparisons with other groups in TRECVID 2008

We compare our results with the best 8 (out of 46) runs of the groups participated in CBCD task of TRECVID 2008. Three of the best results are submitted by INRIA-IMEDIA team [40]. *Joly* is a combination of dissociated dipoles features extraction [41] in sampled keyframes, features indexing and retrieval with distortion-based similarity search structure [20], and spatio-temporal registration of retrieved features. *ViCopT* performs tracking of visual local features and indexes them differently according to some labels of behavior [22], applies distortion-based similarity search structure directly on the local features extracted in keyframes [20], and uses a robust voting algorithm based on labels of behavior [23]. The run named *Joly + ViCopT* is the combination of two approaches, which is invariant to the flip, resize, strong noise, and picture-in-picture transformations. INRIA-IMEDIA group was

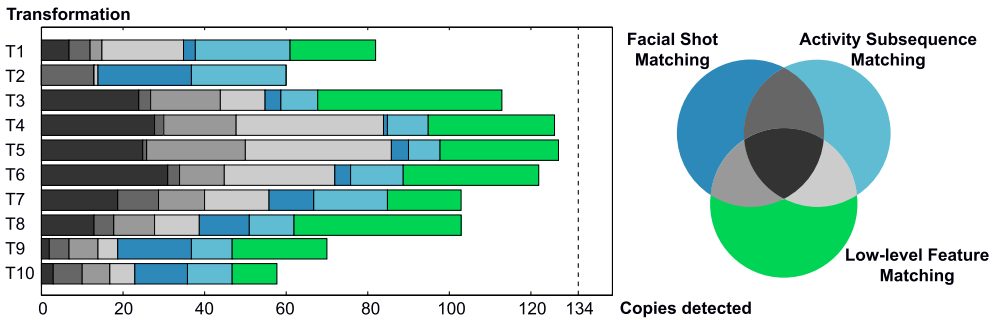

Fig. 9. Contribution of each component for each transformation.

Table 5
Comparison of the proposed method with the best 8 groups in TRECVID'08 CBCD task. Total number of copies that can be detected for each transformation type is 134. CDR: correct detection rate %, QPT: total query processing time in minutes.

	Joly	ViCopT	J +V	LEAR	IBM	Col.U	O.Lab	T & I	Proposed
T1	86	85	95	106	44	67	99	78	82
T2	17	3	121	109	10	16	129	–	60
T3	125	110	129	113	79	98	129	90	113
T4	126	31	126	106	116	29	97	119	126
T5	129	128	134	114	125	111	134	132	127
T6	115	66	117	105	98	46	95	93	122
T7	77	18	76	92	74	17	66	42	103
T8	66	103	115	110	75	56	128	54	103
T9	21	96	102	112	51	19	129	15	70
T10	48	29	62	97	49	23	71	19	58
Total	810	669	1077	1064	721	482	1077	642	964
CDR	60.44	49.92	80.37	79.40	53.80	35.97	80.37	47.91	71.94
QPT	4037	571	9657	4112	696	14851	693	759	648

also responsible for the query video generation and CBCD evaluation software preparation in TRECVID 2008.

The method used by INRIA-LEAR [42] extracts SIFT features from keyframes, and they generate image descriptors using bag-of-features approach and Hamming Embedding. The similarity scores between video clips are geometrically verified and the scores are aggregated to generate video segment matches. Orange Labs [43] uses visual features calculated around regions of interest, and an adaptive and parameter-free method for scoring the matches. Tsinghua University and Intel China Research Center [44] propose a CBCD system that uses SURF descriptors [45] and ANN-based matching.

Our comparisons with other groups are based on the correct detection rate (CDR) and the total query processing time (QPT) for all of the 2010 query videos (see Table 5). Correct detection values of each transformation are calculated with the CBCD evaluation software. Total QPTs are computed from the run files for other groups.

Because we implemented each part of the method separately (for evaluation and comparison purposes), our query processing time is estimated. To speed-up the matching process, various techniques are employed in different stages. In facial shot detection, the method skips to the end of the shot when a face is detected and extracted. In activity subsequence matching, we employ a pruning step in order to discard the reference video segments with very low similarities. However, shot-boundary detection part should process each frame one-by-one. Since other video segmentation techniques cannot handle video files with heavy transformations (see the experiments in [35]), we use the fuzzy color histogram-based shot-boundary detector (see Section 2.2), which is the bottleneck of the online stage. We estimate the total QPT based on the processing time of shot-boundary detector for 3,891,542 query frames, which can be completed in 648 min.

7. Conclusions

We propose a framework for content-based copy detection and video similarity detection. The proposed framework consists of three parts for video segment matching: facial shot matching, activity subsequence matching, and low-level feature matching. We were able to make a fair comparison by testing the method on the query and reference dataset of CBCD task of TRECVID 2008. Our results were compared with the results of top-8 most successful techniques submitted to this task. Experimental results show that the proposed method performs better than most of the state-of-the-art techniques, in terms of both effectiveness and efficiency. It is clear that the system already achieves high correct detection rates for the transformations of text/logo insertion, strong re-encoding, gamma change, and noise addition; however, there is still some potential for improvement to detect copies with camcording and picture-in-picture transformations.

Our future extensions will focus on how to improve the effectiveness in transformations like camcording, picture-in-picture, and very complex ones. The results for camcorded query videos can be improved by translating the frames with the camera parameters calculated automatically from the video, if the video includes camcording transformation. For picture-in-picture transformations, we may need to improve the window extraction method, and consider these windows in low-level feature matching part.

Acknowledgment

Sound and Vision video is copyrighted. The Sound and Vision video used in this work is provided solely for research purposes through the TREC Video Information Retrieval Evaluation Project Collection.

References

[1] G. Langelaar, I. Setyawan, R. Langedijk, Watermarking digital image and video data. A state-of-the-art overview, IEEE Signal Processing Magazine 17 (5) (2000) 20–46.
[2] A. Hampapur, R. Bolle, Comparison of distance measures for video copy detection, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'01), 2001, pp. 737–740.
[3] Y. Zhai, M. Shah, Tracking news stories across different sources, in: Proceedings of 13th ACM International Conference on Multimedia (MULTIMEDIA'05), 2005, pp. 2–10.
[4] W. Hsu, S.F. Chang, Topic tracking across broadcast news videos with visual duplicates and semantic concepts, in: Proceedings of the IEEE International Conference on Image Processing (ICIP'06), 2006, pp. 141–144.
[5] X. Wu, A.G. Hauptmann, C.W. Ngo, Measuring novelty and redundancy with multiple modalities in cross-lingual broadcast news, Computer Vision and Image Understanding 110 (3) (2008) 418–431.
[6] T. Can, P. Duygulu, Searching for repeated video sequences, in: Proceedings of the International Workshop on Multimedia Information Retrieval (MIR'07), 2007, pp. 207–216.
[7] L.-Y. Duan, J. Wang, Y. Zheng, J. Jin, H. Lu, C. Xu, Segmentation, categorization, and identification of commercial clips from TV streams using multimodal analysis, in: Proceedings of 14th ACM International Conference on Multimedia (MULTIMEDIA'06), 2006, pp. 201–210.
[8] R. Junee, Zoinks! 20 Hours of Video Uploaded Every Minute!, 2009. <http://www.youtube.com/blog?entry=on4EmafA5MA>.
[9] A.F. Smeaton, P. Over, W. Kraaij, Evaluation campaigns and TRECVID, in: Proceedings of Eigth ACM International Workshop on Multimedia Information Retrieval (MIR'06), 2006, pp. 321–330.
[10] Content-based copy detection, Guidelines for the TRECVID 2008 Evaluation, 2009. <http://www-nlpir.nist.gov/projects/tv2008>.
[11] TRECVID 2008 Final List of Transformations, 2008. <http://www-nlpir.nist.gov/projects/tv2008/active/copy.detection/final.cbcd.video.transformations.pdf>.
[12] S. Satoh, News video analysis based on identical shot detection, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'02), vol. 1, 2002, pp. 69–72.

[13] M.-C. Yeh, K.-T. Cheng, Video copy detection by fast sequence matching, in: Proceedings of ACM International Conference on Image and Video Retrieval (ACM CIVR'09), 2009.
[14] A. Basharat, Y. Zhai, M. Shah, Content based video matching using spatiotemporal volumes, Computer Vision and Image Understanding 110 (3) (2008) 360–377.
[15] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.
[16] Z. Wu, S. Jiang, Q. Huang, Near-duplicate video matching with transformation recognition, in: Proceedings of the 17th ACM international conference on Multimedia, ACM, 2009, pp. 549–552.
[17] A. Joly, C. Frélicot, O. Buisson, Robust content-based video copy identification in a large reference database, in: Proceedings of ACM International Conference on Image and Video Retrieval (CIVR'03), 2003, pp. 414–424.
[18] A. Joly, O. Buisson, C. Frélicot, Statistical similarity search applied to content-based video copy detection, in: Proceedings of the 21st International Conference on Data Engineering Workshops, 2005, p. 1285.
[19] A. Joly, C. Frélicot, O. Buisson, Content-based video copy detection in large databases: A local fingerprints statistical similarity search approach, in: Proceedings of the IEEE International Conference on Image Processing (ICIP'05), vol. 1, 2005, pp. 505–508.
[20] A. Joly, O. Buisson, C. Frélicot, Content-based copy retrieval using distortion-based probabilistic similarity search, IEEE Transactions on Multimedia 9 (2) (2007) 293–306.
[21] W.L. Zhao, C.W. Ngo, H.K. Tan, X. Wu, Near-duplicate keyframe identification with interest point matching and pattern learning, IEEE Transactions on Multimedia 9 (5) (2007) 1037–1048.
[22] J. Law-To, V. Gouet-Brunet, O. Buisson, N. Boujemaa, Local behaviours labelling for content based video copy detection, in: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), vol. 3, 2006, pp. 232–235.
[23] J. Law-To, O. Buisson, V. Gouet-Brunet, N. Boujemaa, Robust voting algorithm based on labels of behavior for video copy detection, in: Proceedings of the 14th annual ACM International Conference on Multimedia (MM'06), 2006, pp. 201–210.
[24] H. Ren, S. Lin, D. Zhang, S. Tang, K. Gao, Visual words based spatiotemporal sequence matching in video copy detection, in: Proceedings of the 2009 IEEE International Conference on Multimedia and Expo, Institute of Electrical and Electronics Engineers Inc., 2009, pp. 1382–1385.
[25] G. Willems, T. Tuytelaars, L. Van Gool, Spatio-temporal features for robust content-based video copy detection, in: MIR'08: Proceeding of the First ACM International Conference on Multimedia Information Retrieval, ACM, New York, NY, USA, 2008, pp. 283–290. http://doi.acm.org/10.1145/1460096.1460143 .
[26] R. Mohan, Video sequence matching, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98), vol. 6, 1998, pp. 3697–3700. doi:doi:10.1109/ICASSP.1998.679686.
[27] C. Kim, B. Vasudev, Spatiotemporal sequence matching for efficient video copy detection, IEEE Transactions on Circuits and Systems for Video Technology 15 (1) (2005) 127–132.
[28] E. Ardizzone, M.L. Cascia, A. Avanzato, A. Bruna, Video indexing using mpeg motion compensation vectors, in: Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS'99), 1999, p. 725.
[29] M. Bertini, A.D. Bimbo, W. Nunziati, Video clip matching using MPEG-7 descriptors and edit distance, Lecture Notes in Computer Science 4071 (2006) 133–142.
[30] A. Sarkar, V. Singh, P. Ghosh, B.S. Manjunath, A. Singh, Efficient and robust detection of duplicate videos in a large database, IEEE Transactions on Circuits and Systems for Video Technology :doi:10.1109/TCSVT.2010.2046056.
[31] A. Hampapur, K. Hyun, R. Bolle, Comparison of sequence matching techniques for video copy detection, in: Proceedings of the International Conference on Storage and Retrieval for Media Databases, vol. 4676, 2002, pp. 194–201.
[32] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, Video copy detection: a comparative study, in: Proceedings of sixth ACM International Conference on Image and Video Retrieval (CIVR'07), 2007, pp. 371–378.
[33] Building video queries for Trecvid 2008 copy detection task, 2009. <http://www-nlpir.nist.gov/projects/tv2008/TrecVid2008CopyQueries.pdf>.
[34] D.M. Mount, S. Arya, ANN - Approximate Nearest Neighbor Searching, 2009. <http://www.cs.umd.edu/~mount/ANN/>.
[35] O. Küçüktunç, U. Güdükbay, O. Ulusoy, Fuzzy color histogram-based video segmentation, Computer Vision and Image Understanding 114 (1) (2010) 125–134.
[36] B.S. Manjunath, P. Salembier, T. Sikora, Introduction to MPEG-7, Multimedia Content Description Interface, John Wiley and Sons, Ltd., 2002.
[37] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01), vol. 1, 2001, pp. 511–518.
[38] R. Lienhart, J. Maydt, An extended set of haar-like features for rapid object detection, in: Proceedings of the International Conference on Image Processing (ICIP'02), vol. 1, 2002, pp. 900–903.
[39] Open Source Computer Vision Library, 2009. <http://opencvlibrary.sourceforge.net>.
[40] A. Joly, J. Law-to, N. Boujemaa, INRIA-IMEDIA TRECVID 2008: Video copy detection, in: Proceedings of the TREC Video Retrieval Evaluation (TRECVID), 2008. <http://www-nlpir.nist.gov/projects/tvpubs/tv8.papers/inria-imedia.pdf>.
[41] A. Joly, New local descriptors based on dissociated dipoles, in: Proceedings of Sixth ACM International Conference on Image and Video Retrieval (CIVR'07), 2007, pp. 573–580.
[42] M. Douze, A. Gaidon, H. Jegou, M. Marszalek, C. Schmid, INRIA-LEARGs video copy detection system, in: Proceedings of the TREC Video Retrieval Evaluation (TRECVID), 2008. <http://www-nlpir.nist.gov/projects/tvpubs/tv8.papers/inria-lear.pdf>.
[43] N. Gengembre, S.-A. Berrani, The orange labs real time video copy detection system - TRECVID 2008 results, in: Proceedings of the TREC Video Retrieval Evaluation (TRECVID), 2008. <http://www-nlpir.nist.gov/projects/tvpubs/tv8.papers/orangelabs.pdf>.
[44] Y. Liang, X. Liu, Z. Wang, J. Li, B. Cao, Z. Cao, Z. Dai, Z. Guo, W. Li, L. Liu, Z. Meng, Y. Qin, Q. Shi, A. Tian, D. Wang, Q. Wang, C. Zhu, X. Hu, J. Yuan, P. Yuan, B. Zhang, THU and ICRC at TRECVID 2008, in: Proceedings of the TREC Video Retrieval Evaluation (TRECVID), 2008. <http://www-nlpir.nist.gov/projects/tvpubs/tv8.papers/thu-icrc.pdf>.
[45] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool, Speeded-up robust features (SURF), Computer Vision and Image Understanding 110 (3) (2008) 346–359.