Information Sciences 302 (2015) 108–121

Contents lists available at ScienceDirect

Information Sciences

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

Discovery and organization of multi-camera user-generated videos of the same event

Sophia Bano *,[1], Andrea Cavallaro

Centre for Intelligent Sensing, Queen Mary University of London, E1 4NS London, UK

ARTICLE INFO

Article history:
Received 2 February 2013
Received in revised form 7 July 2014
Accepted 7 August 2014
Available online 13 August 2014

Keywords:
Multi-camera synchronization
User-generated video
Audio chroma feature
Video identification
Event clustering
Audio degradation

ABSTRACT

We propose a framework for the automatic grouping and alignment of unedited multi-camera User-Generated Videos (UGVs) within a database. The proposed framework analyzes the sound in order to match and cluster UGVs that capture the same spatio-temporal event and estimate their relative time-shift to temporally align them. We design a descriptor derived from the pairwise matching of audio chroma features of UGVs. The descriptor facilitates the definition of a classification threshold for automatic query-by-example event identification. We evaluate the proposed identification and synchronization framework on a database of 263 multi-camera recordings of 48 real-world events and compare it with state-of-the-art methods. Experimental results show the effectiveness of the proposed approach in the presence of various audio degradations.

1. Introduction

With the increasing availability of smartphones, more people capture videos of their experience of attending events such as concerts, sporting competitions and public rallies. Social media sites then act as a distribution channel to share these experiences by giving access to these *unorganized* and *unsynchronized* User-Generated Videos (UGVs). This trend has invoked a new research direction involving search and organization of multimedia data of the same event [2,40]. We define an *event* as a continuous action captured simultaneously by multiple user-devices from different positions located in proximity with each other.

By identifying videos belonging to a specific event, powerful event browsing can be enabled, which in turn can improve web search tools. However, it is non-trivial to automatically identify UGVs of the same event. In fact traditional metadata-based methods for event retrieval [21,33] may not always be effective because metadata associated with uploaded videos may lack consistent and objective tagging, or correct timestamps [15,23]. Moreover, UGVs are not synchronized, and automatic synchronization is difficult due to the presence of various audio and visual degradations. We are interested in using the audio signal for identifying and synchronizing UGVs. Synchronization of UGVs using audio features is generally based on onsets (starting point of an audio instant) or fingerprints (compact content-based audio signatures) [43,28]. In order for a method to be successful, audio degradations and noise have to be taken into account.

* Corresponding author.
E-mail addresses: sophia.bano@eecs.qmul.ac.uk (S. Bano), andrea.cavallaro@eecs.qmul.ac.uk (A. Cavallaro).

[1] S. Bano was supported by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the Education, Audiovisual and Culture Executive Agency (FPA n. 2010-0012). This work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/K007491/1.

http://dx.doi.org/10.1016/j.ins.2014.08.026
0020-0255/© 2014 Elsevier Inc. All rights reserved.

We categorize audio degradations into two groups, namely, local and global degradations. *Local degradations* are caused by recording device settings, channel noise, surrounding noise and reverberations. *Global degradations* are common to some or all recording devices (e.g. a crowd cheering, a whistle blowing during a specific event or a public rally) and may help during the synchronization process.

In this paper, we propose an automatic query-by-example event identification and synchronization framework using audio chroma features. Although the recording of a specific event captured by multiple devices might differ in loudness or sound intensity due to the varying quality of recording devices, the distance of the device from the sound source and surrounding noise, the pitch of the recorded sound will remain constant [10]. For this reason, we use chroma as an audio feature [18], as it gives the distribution of energy along different pitch classes. The novelty of this work also lies in the design of a descriptor from match and non-match histograms that facilitates the definition of an automatic classification threshold for event identification and clustering. We show the robustness of the proposed synchronization approach compared to alternative methods over various audio degradations.

The paper is organized as follows. In Section 2, we present the related work. In Section 3, we define and formulate the video identification and synchronization problem. In Section 4, we present an overview of the proposed framework. In Section 5, we describe our proposed event identification framework, which is followed by time-shift estimation and cluster membership validation in Section 6. In Section 7, we describe our dataset of UGVs, assess our method and compare the method with the existing state of the art. Finally, Section 8 concludes the paper.

2. Related work

In this section, we discuss the state of the art for content identification for videos, music and generic sounds, and synchronization for multi-camera videos.

Video identification aims at identifying videos that match with a query, for example, to filter unauthorized distribution of copyrighted videos [22,31,32,35,41,44]. Extending these approaches to UGVs is not trivial because there might not exist the same visual evidence between pairs of UGVs due to variations in the field of view, changing and poor lighting conditions, and visual quality. A related topic is content identification in music for tagging, play-listing and taste profiling [4,8,9,12]. Methods include those used for Shazam and TrackID [29,34,46], which are based on the fingerprinting method by Wang et al. [47] for audio identification.

Event identification using audio features has been addressed in [28,11,5], which use landmark-based audio fingerprinting [47], where the landmarks are the onsets of local frequency peaks and are identified from the short-time Fourier transform. Kennedy and Naaman [28] presented an approach for the synchronization and organization of a collection of concert recordings, in which the classification threshold is computed based on the mean and standard deviation of the matches. Cotton and Ellis [11] used matching pursuit to obtain a prominent representation of audio events and tested their event identification approach on a public speech dataset. Both approaches [28,11] use hash value similarity maximization for matching pairs of recordings. A similar approach is presented by Bryan et al. [5] for event identification and synchronization. This method uses landmark cross-correlation for matching and a fixed classification threshold to cluster a speech dataset of 180 professional recordings and 23 user-generated recordings of concerts. However, a fixed classification threshold [5,28] can be applied only if the dataset under analysis is small.

Existing methods for multi-camera UGV synchronization involve extraction and matching of features such as audio fingerprints [43,28,5,6], audio onsets [43,3], audio feature-based classification [42] and audio-visual events [7], where an audio-visual event was defined to be a simultaneous change in the audio and video streams which are well localized in time. Also, Kammerl et al. [27] proposed graph-based methods for temporal synchronization built by analyzing consistency in pairwise cross-correlation of three audio features, namely, spectral flatness, zero crossing and signal energy. The audio fingerprinting method of Haitsma and Kalker [20] is exploited by Shrestha et al. [42,43]: a 32-bit sub-fingerprint (binary) is generated based on spectrum-temporal analysis of the audio in an overlapping window. Two fingerprint-blocks of 256 consecutive sub-fingerprints are considered to be matching if the number of bit errors (BER) is smaller than a threshold [20]. The landmark-based fingerprinting approach by Wang [47] is used by Kennedy and Naaman [28] and Bryan et al. [5] for the synchronization of collections of concert recordings. However, fingerprinting might become sensitive to reverberations [43] and strong local degradations [36] (see Section 5.3).

In comparison to audio fingerprints, onset-based methods [43] are more sensitive to *audio degradations* as they reflect only positive changes in energy and false positive onsets can be generated by channel and background noise. Casanovas and Cavallaro [7] presented an audio-visual method for multi-camera synchronization in which an audio event is detected using audio onsets [43] followed by visual event detection by analyzing the local variation of pixel intensities within a predefined space–time blocks of a detected audio event. A space–time block is considered to be active if its local variation is greater than a threshold, and an audio-visual event is detected when several active blocks are in close proximity. This method is sensitive to *audio degradations*, in the same way as the onset based method [43] is and is dependent on camera motion and near or far fields of view.

An audio feature classification method for multi-camera synchronization is presented in [42], which is based on low-level signal properties, mel-frequency cepstral coefficients (MFCC), psychoacoustic features (roughness, loudness, sharpness), and temporal envelope fluctuations model. Quadratic discriminant analysis [37] is performed to estimate the probabilities of

Table 1
State of the art methods for identification and synchronization of multi-camera UGVs. Key: ID: identification; SYN: synchronization; AFC: audio feature classification; AF: audio fingerprint; AO: audio onset; PI: pixel intensity; AC: audio chroma; ILD: insensitive to local degradations; IGD: insensitive to global degradations; K: total number of events; M: total number of recordings; PP: professional production recordings; AS: amplified sound recordings; NAS: non-amplified sound recordings. The letters a and b in Ref. indicate different methods proposed in the same paper.

Ref.	Method		Feature					Properties		Matching approach	Dataset properties				
	ID	SYN	AFC	AF	AO	PI	AC	ILD	IGD		K	M	PP	AS	NAS
[42]		✓	✓							Cross-correlation maximization	5	11		✓	
[43][a]		✓			✓					Cross-correlation maximization	7	30		✓	
[43][b]		✓		✓					✓	Bit Error Rate (BER) minimization	7	30		✓	
[28]	✓	✓		✓					✓	Hash value similarity maximization	3	608		✓	
[5]	✓	✓		✓					✓	Cross-correlation maximization	9	203	✓	✓	
[11]	✓			✓					✓	Hash value similarity maximization	1	733		✓	
[7]		✓			✓	✓				Cross-correlation maximization	8	40		✓	✓
This work	✓	✓					✓	✓	✓	Feature similarity maximization	48	263		✓	✓

silence, music, speech, noise and crowd classes for every frame size of 11.6 ms. Cross-correlation is then used to match recordings to estimate the time-shift.

In this work, we exploit chroma features to identify clusters of UGV and then perform synchronization. Chroma features are mainly used in professional music recordings for identification [16], chord recognition [25], genre classification, audio thumbnailing [1], matching [39] and synchronization [17]. Müller et al. [39] presented an audio matching approach using Chroma Energy distribution Normalized Statistics (CENS), which is a variant of chroma features. In this method either the number of matches to be retrieved or the threshold value for the distance of a retrieved match need to be pre-defined. Ewert et al. [17] proposed a method of score-to-audio alignment in music that combines chroma with onset features and performs matching using dynamic time warping (DTW). The testing is performed on noiseless synthesized music files. To the best of our knowledge chroma feature has not been used for analyzing audio content of UGV which contains several *audio degradations* and an *amplified* or *non-amplified sound source*.

Table 1 summarizes the state of the art for identification and synchronization of multi-camera UGV. Unlike [43,28,5,11], which are sensitive to reverberations and local degradations, our proposed framework shows robustness to both local and global degradations (see Section 7).

3. Problem formulation

Let $\mathbb{C} = \{\mathbb{C}_m\}_{m=1}^{M}$ be a database of unorganized and unsynchronized User-Generated Videos (UGVs) containing M recordings $\mathbb{C}_m$. Let $E = \{E_k\}_{k=1}^{K}$ be the set of events represented in $\mathbb{C}$, where $K \leqslant M$. We are interested in solving three problems: clustering videos corresponding to the same event, synchronizing the clustered videos on a common timeline and associating a new video ($\mathbb{C}_q$) to an existing cluster.

Video event clustering aims to organize the database $\mathbb{C}$ into K clusters, such that each cluster k represents an event $E_k = \{\mathbb{C}_{k,n}\}_{n=1}^{N_k}$ containing N_k UGVs having partial or complete temporal overlap with each other. *Multi-camera synchronization* aims to temporally align the set of UGVs belonging to an event E_k. Without loss of generality, let us consider two videos $\mathbb{C}_{k,1}$ and $\mathbb{C}_{k,2}$ of the same event E_k, and having the same frame rate. $\mathbb{C}_{k,1}$ and $\mathbb{C}_{k,2}$ are considered to be synchronized when the recording time t_1^i at the ith frame of $\mathbb{C}_{k,1}$ and t_2^j at the jth frame of $\mathbb{C}_{k,2}$ correspond to the same moment in the universal time t, an instant referring to the continuous physical time. Let the time-shift be $\triangle t_{12}$. Finally, the problem of *associating a new video* $\mathbb{C}_q$ to a cluster k involves identifying the set $E_k = \{\mathbb{C}_{k,n}\}_{n=1}^{N_k}$ of UGVs matching $\mathbb{C}_q$.

4. Proposed framework

Our proposed discovery and organization framework can be split into two main stages, as depicted in Fig. 1. For a query video $\mathbb{C}_q$, the set of UGVs $\{\mathbb{C}_{k,n}\}_{n=1}^{N_k}$ belonging to event E_k is identified, then the synchronization time-shifts $\Delta t_{q,1:N_k}$ are estimated. In order to eliminate false identifications, a validation of the synchronization time-shifts is performed. Our multi-camera visualizer is then used to playback the set of synchronized UGVs belonging to E_k. In this section, we present the proposed framework and the main assumptions.

We extract the chroma feature vector F_m using an audio frame size f_r. A feature matching strategy is proposed that maximizes the similarity of pairs of overlapping feature vectors F_i and F_j and provides a histogram representation for the match and non-match recording pairs. The histogram depicts the occurrence of the value of similarity between F_i and F_j. The frame size f_r in feature extraction is important for the design of the video identification and synchronization framework. By refining f_r for the identified cluster of videos for an event E_k, we can estimate the synchronization time-shift. For video identification, a small value of f_{r1} would make the identification process extremely slow, while a large f_{r1} might not give accurate results.

For video event identification, we assume that the histograms for match and non-match recording pairs are separable. For example, when audio signals A_i and A_j belong to the same E_k event, matching of their feature vectors F_i and F_j shows strong

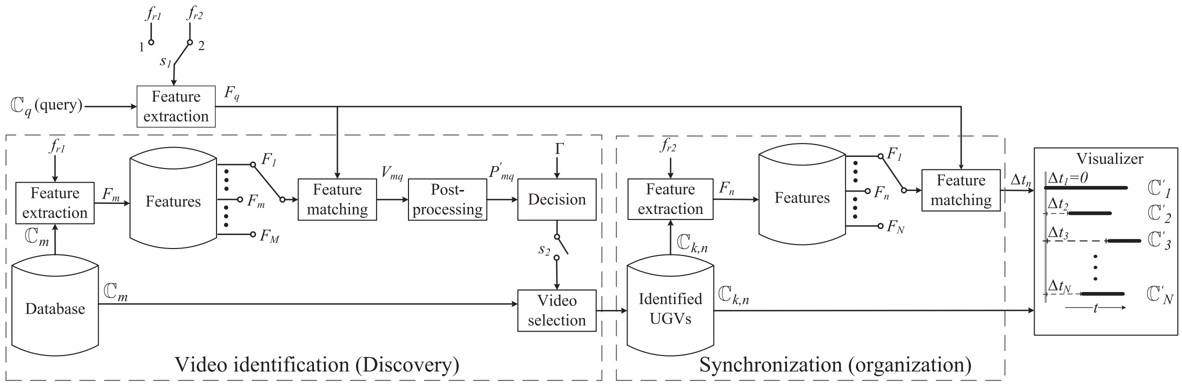


Fig. 1. Block diagram of the proposed framework, which is composed of two stages, video identification (discovery) and synchronization (organization). For a given query video $\mathbb{C}_q$, feature extraction is performed with f_{r1} ($s_1 = 1, s_2 =$ OFF), and its feature matching is done with the feature database of M UGVs to generate the feature matching vector V_{mq}. Post-processing is then performed and a classification threshold Γ is applied to identify the set of N overlapping recordings. The time-shift estimation Δt_n is then performed with f_{r2} ($s_1 = 2, s_2 =$ ON) for these N recordings in order to synchronize them. A multi-camera visualizer is used for playback of the N synchronized UGVs.

correlation represented by a high peak in the matching histogram V_{ij}, otherwise, there is no dominant peak. Unlike existing methods for video identification [5,28], which use a fixed classification threshold to detect the matching recording pairs, we propose an automatic classification threshold strategy in which we learn the threshold (as detailed in Section 5.4).

For synchronization, we assume that the time difference of arrival of a sound is negligible. Two recording devices observing the same event might have a time difference of arrival of sound ε_{12} due to their different distances from the sound source. Let the audio signal of the nth video recording be $A_n(t_n^i), t_n^i = \frac{i}{s_n}, 0 \leqslant i < K_n$, where i is the index of the audio sample, t_n^i is the time at the ith sample for the nth recording, A_n is the amplitude of the audio sample at time t_n^i, s_n is the audio sampling rate and K_n is the total number of audio samples. The estimated time-shift obtained between $\mathbb{C}_{k,1}$ and $\mathbb{C}_{k,2}$ is

$$\triangle t_{12} = t_1^i - t_2^j + \varepsilon_{12}, \tag{1}$$

where $\varepsilon_{12} = \frac{\triangle\delta_{12}}{\upsilon_s}$ is the time difference of arrival, in which $\triangle\delta_{12} = \delta_1 - \delta_2$ reflects the distance of the cameras from the sound source and $\upsilon_s = 340$ m/s is the speed of sound. Let us consider that the videos are recorded at a frame rate of $f_{rate} = 25$ fps. The separation allowed between two cameras while staying in a video frame tolerance of ±1 *frame* ($\varepsilon_{12} = 0.04$ s) is $\triangle\delta_{12} = 14$ m. In the case of UGVs, $\triangle\delta_{12}$ is unknown as, when sharing these videos on the internet, the information about the geographical location of the cameras and their distance from the sound source is generally not available. We assume that the cameras recording a particular event lie in the vicinity of each other so that $\triangle\delta_{12} < 14$ m, thus making ε_{12} negligible.

Some recording devices might yield the problem of audio drifting out of sync with the video when the recording time is long. Audio drift is generally caused by audio sample rates that do not match the audio settings in the recording device. In this work, we assume that no UGV is affected by the audio drifting out of sync with the video issue.

5. Event identification

In this section, we present our video event clustering approach which aims to identify multi-camera UGVs of the same event E_k. The two main blocks of this approach are feature extraction and feature matching. We propose an approach for learning the classification threshold Γ from the match/non-match histograms of the training video events. Event clustering is then performed followed by the association of a new video $\mathbb{C}_q$ to the database.

5.1. Feature extraction

We use chroma feature [18] as a descriptor for the audio content of UGVs. Audio chroma gives a 12-dimensional representation of the tonal content of an audio signal derived by combining bands belonging to twelve pitch classes $(C, C^\#, D, D^\#, E, F, F^\#, G, G^\#, A, A^\#, B)$ corresponding to the same distinct semitones (or chroma). The chroma feature is useful in distinguishing different types of sound, such as voice and musical instruments [1,39]. The chroma feature vector is represented as $\mathbf{v} = (v_0, v_1, \ldots v_{11}) \in \mathbb{R}^{12\times 1}$, where v_0 corresponds to the energy of chroma C, v_1 corresponds to the energy of chroma $C^\#$, and so on. Each chroma is computed as [38]

$$v_\rho = \sum_{s.t\ \ l\ \ (mod\ \ 12)=\rho} f(l), \tag{2}$$

where $\rho \in [0, 11]$ indicates the chroma number and l denotes the pitch class index corresponding to a particular spectrum bin index. The pitch class index l depends on their center frequency $f(l)$ in a logarithmic way and is given by [38]

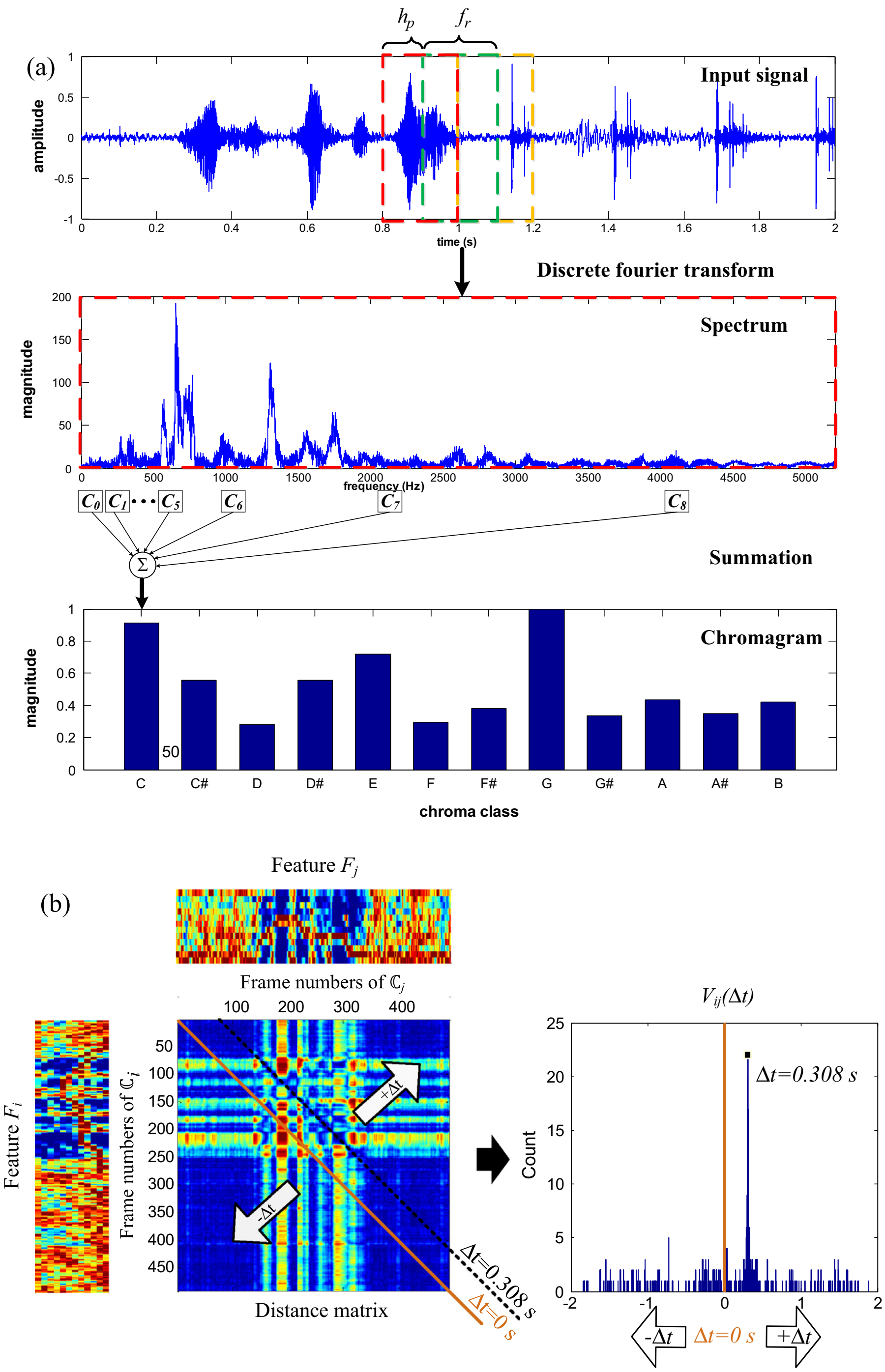


Fig. 2. An example illustrating chroma feature extraction and matching. (a) For a particular audio frame f_r (highlighted in red), the spectrum is divided into subbands and a chromagram is formed by summing all pitch bands belonging to a particular chroma. (b) Feature matching using the distance matrix for two test audio signals of duration 2 s is shown. The main diagonal of the distance matrix corresponds to zero, the lower diagonal corresponds to negative and the upper diagonal corresponds to positive time-shifts. The minimum across each row is calculated and the count of minimum distances is accumulated across each diagonal to give the histogram V_{ij}, and its peak corresponds to the time-shift. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$l = v_d log_2\left(\frac{f(l)}{f_s}\right) + l_c, \tag{3}$$

where $f_s = 440$ Hz, which is the standard frequency for pitch tuning [38] that corresponds to the concert pitch (reference pitch to which musical devices are tuned) $l_c = 69$ (A4) and $\upsilon_d = 12$ which represents the 12-dimensions (semitones) of the chroma vector. A pitch class is the set of all pitches which share the same chroma. For instance, the pitch class corresponding to chroma C is $(C0, C1, C2, \ldots, C8)$ and relates to the pitch subbands $(12, 24, 36, \ldots, 108)$. This is represented using a chromagram as illustrated in Fig. 2(a).

We first decompose a given audio signal of a UGV into overlapping audio frames and then compute chroma features for each audio frame. Each audio frame is composed of an audio segment of frame size f_r and overlap shift h_p between two consecutive frames (as shown in Fig. 2(a)). The number of audio frames Υ_n in $A_n(t_n)$ is a function of the number of audio samples K_n in $A_n(t_n)$ and is computed as

$$\Upsilon_n = \frac{K_n}{s_n f_r h_p}. \tag{4}$$

The frequency spectrum $f(l)$ of each audio frame is then computed by applying the discrete Fourier transform (DFT), and is mapped into the pitch class using Eq. (3). The chroma vector for a particular audio frame is thus represented as $\mathbf{v}^q \in \mathbb{R}^{12\times 1}$, such that q defines the time stamp of a particular frame position. Fig. 2(a) illustrates the process of extraction of the chroma feature for a particular audio frame. Chroma features for the nth audio signal $A_n(t_n)$, segmented into Υ_n audio frames are given by

$$F_n = \{\mathbf{v}_n^q\}_{q=1}^{\Upsilon_n}, \tag{5}$$

where $\mathbf{v}_n^q \in \mathbb{R}^{12\times 1}$ is the chroma feature vector for the qth frame of the nth camera's audio signal.

5.2. Feature matching

Once the features are extracted, the next step is to perform feature matching for computing the similarity and time-shifts between pairs of recordings. Our proposed matching method operates by maximizing the feature similarity between a pair of camera recordings. For a pair of recordings $\mathbb{C}_i$ and $\mathbb{C}_j$, the distance between their chroma features F_i and F_j is given by $d_{ij}^{st} = E(\mathbf{v}_i^s, \mathbf{v}_j^t)$, where $E(\cdot)$ is the Euclidean distance [45] between the sth and tth feature vector, and $s \in [1, \Upsilon_i]$ and $t \in [1, \Upsilon_j]$ give the range of frame numbers for $\mathbb{C}_i$ and $\mathbb{C}_j$, respectively. The distance matrix $\wedge_{ij}$ between F_i and F_j is then given by

$$\wedge_{ij} = [d_{ij}^{st}]_{\mathbb{R}^{\Upsilon_i \times \Upsilon_j}}. \tag{6}$$

Fig. 2(b) shows the distance matrix $\wedge_{ij}$ for two feature vectors obtained from two overlapping camera recordings, each of 2 s duration. The distance matrix $\wedge_{ij}$ contains information about the feature matching of two recordings. In order to interpret this information, the point of minimum distance across each row of the distance matrix $\wedge_{ij}$ is calculated. This corresponds to the point where a likely match occurs:

$$\chi = \operatorname*{argmin}_s \, [d_{st}], \quad \forall\, t \ \in [1, \Upsilon_j]. \tag{7}$$

The distance matrix $\wedge_{ij}$ is a rectangular matrix in which the main diagonal corresponds to *zero* time-shift. The diagonal above and below the main diagonal correspond to positive and negative time-shifts, respectively. We calculate the matching histogram $V_{ij}(\Delta t)$ for camera recordings $\mathbb{C}_i$ and $\mathbb{C}_j$ from the distance matrix $\wedge_{ij}$ for the counts of the number of minimum distances along each diagonal. This is illustrated in Fig. 2(b). The x and y-axes in $V_{ij}(\Delta t)$ correspond to the time-shifts and counts, respectively. If the overlapping between a pair of recordings is greater than 8%, we get a dominant peak in the matching histogram which represents the synchronization time-shift, otherwise, it is unlikely to have a dominant peak.

5.3. Feature analysis

In order to find the lowest dimension of chroma feature which can provide the correct synchronization time-shift, we conducted an experiment by analyzing pairs of audio signals from different events. For F_i and F_j, we computed the synchronization time-shifts for all combinations of 1–12 dimensions of chroma features, which are 12, 66, 220, 495, 792, 924, 792, 495, 220, 66, 12 and 1 respectively. Fig. 3 shows the effect of varying the dimension of the chroma feature on five pairs of recordings, where the first row depicts the maximum, mean and minimum time-shift error when testing with all possible combinations. The second row shows the occurrence of true and false matches which correspond to correct and incorrect synchronization time-shifts with a ±0.05 s tolerance, normalized over all the combinations of varying dimensions of the chroma feature.

When the overlap between two signals is greater than 20% (18 s) (Fig. 3(b), (d), and (e)), any combination of chroma beyond 6-dimensions is sufficient for achieving synchronization. Otherwise, if the two audio signals are only partially overlapping and the length of one signal with respect to the other is short (7 s) with minimum 8% overlap), the synchronization time-shift is not achieved until the 11th and 12th dimensions of the chroma feature as shown in Fig. 3(a) and (c), respectively. In the case of Fig. 3(a), a concert event pair containing amplified sound, the minimum overlap is 8% (7 s) with respect to the longer recording. In the case of Fig. 3(c), a public event pair containing strong audio degradations along with

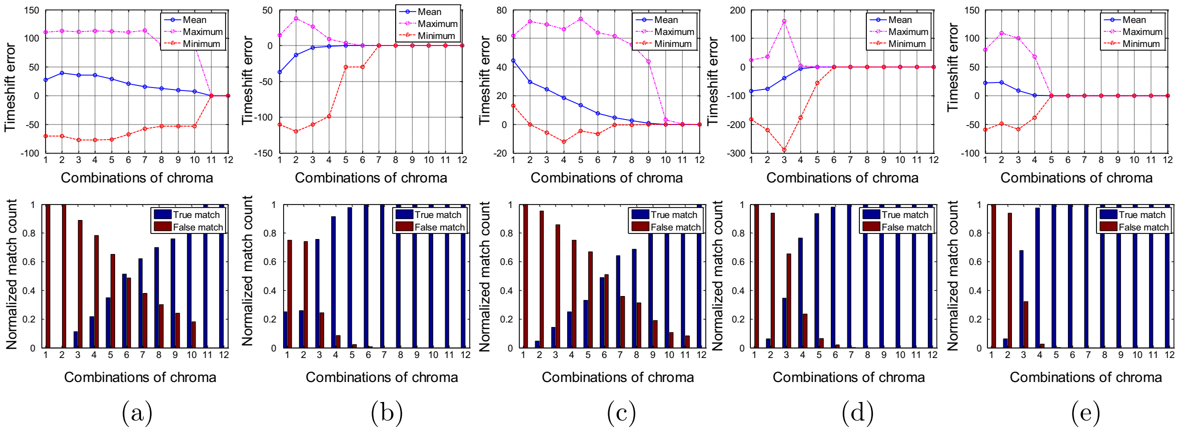


Fig. 3. Effect of varying the dimensions of the chroma feature. The first row shows the maximum, mean and minimum time-shift errors for pairs of recordings. The second row shows the normalized true and false match as counted for all combinations of varying dimensions of the chroma feature. (a) Nickelback_Event3 recording pair of duration 3:18 and 0:18 s, (b) Madonna_Event recording pair of duration 2:59 and 1:20 s, (c) Olympic torch Sheffield Event recording pair of duration 1:28 and 0:39 s, (d) Olympic torch Mile end Event recording pair of duration 6:22 and 6:27 s, and (e) Xmas dinner event recording pair of duration 3:19 and 2:17 s.

non-amplified sounds, the minimum overlap is 14% (12 s) with respect to the longer recording. This overlap is required to get the correct synchronization time-shift. Note that audio fingerprinting [43] is unable to give the correct synchronization time-shift for these cases.

The minimum value of 8% overlap between a pair of recordings is required when performing feature matching in which we use the minimum distance across each row χ. This results in outliers in the matching histogram with shorter signals. This effect can be overcome by setting an empirical threshold on χ for outlier removal or by using all 12 dimensions of chroma.

5.4. Classification threshold

Let us take $\widehat{\mathbb{C}} \subseteq \mathbb{C}$ of UGVs such that $\widehat{\mathbb{C}} = \{\mathbb{C}_{\widehat{m}}\}_{\widehat{m}=1}^{\widehat{M}}$, where $\widehat{M} \ll M$ for training the classification threshold such that these recordings are not included in the test data. The database $\widehat{\mathbb{C}}$ contains $\widehat{M}$ videos for $\widehat{E} = \{E_{\widehat{k}}\}_{\widehat{k}=1}^{\widehat{K}}$ events, where $\widehat{K} \ll K$, such that we have at least two overlapping videos for each $E_{\widehat{k}}$. For these $\widehat{M}$ videos, we extract the features $\{F_m\}_{m=1}^{\widehat{M}}$ using frame size f_{r1}. The selection of f_{r1} is done empirically and will be discussed in Section 7.2. We compute the matching histograms $\mathbf{V}$ for all $\widehat{M} \times \widehat{M}$ video recording pairs (as discussed in Section 5.2). The matching histograms are given by

$$\mathbf{V} = \{V_{ij}(\Delta t)\}, \quad \forall i,j \in [1, \widehat{M}]. \tag{8}$$

We then compute the delay matrix $\mathbb{D}$ for all video recording pairs, such that $\mathbb{D} = [D_{ij}]^{\widehat{M}\times\widehat{M}}$, where each element of $\mathbb{D}$ is given by

$$D_{ij} = \underset{\Delta t}{\operatorname{argmax}} \; V_{ij}(\Delta t). \tag{9}$$

We propose a method for the extraction of a descriptor from histograms $\mathbf{V}$, which is invariant within the match and non-match classes (Section 5.4.1). Using these descriptors we train a support vector classifier for $\widehat{\mathbb{C}}$ to obtain the classification threshold Γ (Section 5.4.2).

5.4.1. Histogram descriptor extraction

For the histogram $V_{ij}(\Delta t)$, we compute the descriptor P'_{ij} by performing a post-processing step (Fig. 1). Each histogram $V_{ij}(\Delta t)$ is first normalized with respect to its maximum value at Δt:

$$\widehat{V}_{ij}(\Delta t) = \frac{V_{ij}(\Delta t)}{\max_{\Delta t} V_{ij}(\Delta t)}. \tag{10}$$

A scanning threshold parameter $0 \leqslant T_r \leqslant 1$ is then defined, which scans $\widehat{V}_{ij}(\Delta t)$ from top to bottom counting the number of matches on each incremental step η (where $\eta = 0.01$ of T_r). This gives the match count P_{ij} with respect to the scanning threshold parameter T_r making it independent of the timeshifts (Fig. 4). The derivative P'_{ij} which reflects the change in P_{ij} is then computed thus giving a 100-point descriptor of the histogram $V_{ij}(\Delta t)$. P_{ij} is a step representation which shows the accumulation of the number of matches. By taking its derivative P'_{ij}, we get a unique representation in which the descriptor shows high value at the instances of change and reminds zero elsewhere. Therefore, the descriptor $P'_{ij} \in \mathbb{R}^{100}$ is distinguishable for match and non-match classes as their histogram $V_{ij}(\Delta t)$ is, but it gives a common representation for all variations of

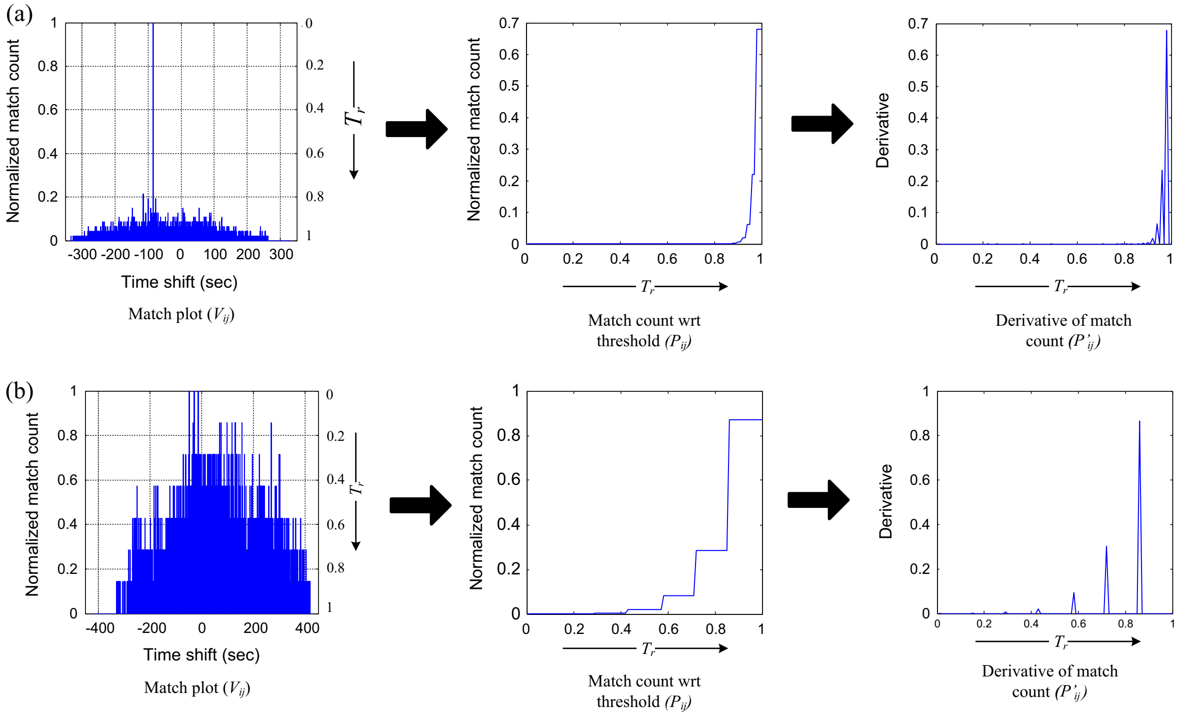


Fig. 4. Post-processing of matching histogram $V_{ij}(\Delta t)$: (a) example histogram obtained for the match class, and (b) example histogram obtained for the non-match class. Histogram descriptors P'_{ij} are computed for all $V_{ij}(\Delta t)$ by scanning from top to bottom using $0.0 \leqslant T_r \leqslant 1.0$ and taking their derivative.

match and non-match classes. Fig. 4 illustrates the process of the histogram descriptor P'_{ij} extraction from match and non-match histograms.

5.4.2. Classifier

The obtained histogram descriptors P'_{ij} are rearranged and labeled as belonging to the match and non-match classes for training the classifier. P'_{ij} are rearranged row-wise to give the set

$$\mathbb{P}' = \{P'_{11}, P'_{12}, \ldots, P'_{\widehat{M}\widehat{M}}\}. \tag{11}$$

$\mathbb{P}'$ contains N_p match descriptors and N_n non-match descriptors, where $N_p + N_n = \widehat{M} \times \widehat{M}$. In order to avoid over-fitting the data, we use a bag-of-words approach [26]. We perform k-means clustering for match and non-match class vectors by selecting κ_{N_p} and κ_{N_n} as the number of clusters which are determined using the elbow method [30]. The returned cluster center represents the possible variations within a class which are then considered as the training set. The clustered set of training vectors belonging to match and non-match classes are given by

$$\mathbb{T} = \{(\overline{P}'_1, 1), \ldots, (\overline{P}'_\kappa, 1), \ldots, (\overline{P}'_{k_{N_p}}, 1), (\overline{P}'_1, -1), \ldots, (\overline{P}'_\kappa, -1), \ldots, (\overline{P}'_{k_{N_n}}, -1)\}, \tag{12}$$

where $\overline{P}'_\kappa \in \mathbb{R}^{100}$ represents the cluster center. We use a linearly separable support vector classifier (SVC) [48] for separating the two classes and computing the classification threshold Γ. SVC learns Γ using the training data $\mathbb{T}$, such that it maximizes the distance between the support vectors of the two classes. The learned classification threshold Γ is then used to classify and cluster the testing database (Section 5.5) for which the time-shift estimation and validation is then performed for synchronization (Section 6).

5.5. Event clustering

In order to identify the group of UGVs that belongs to the same event E_k, we extract the descriptors $P'_{ij}, \ \forall i,j \in [1, M]$, where $\widehat{M}$ recordings are not included. The classification threshold Γ is then used to identify overlapping UGVs belonging to the same events. As a result, we get an identification matrix $\mathbb{I} = \{I_{ij}, \ |I_{ij} \in \mathbb{Z}_{[-1,1]}\}, \ \mathbb{I} \in \mathbb{Z}^{M \times M}$, which is symmetric. I_{ij} takes the value 1 if an overlapping video is identified, otherwise its value is -1. Our proposed method does not require initialization by the number of clusters to be identified. The group of identical rows in $\mathbb{I}$ corresponds to the videos identified as belonging to the same event E_k. The set of videos are grouped to form an event cluster $E_k = \{\mathbb{C}_{k,n}\}_{n=1}^{N_k}$. Once the clusters are identified, the longest UGV within each cluster, $\widetilde{\mathbb{C}}_k$, is taken as the representative for each event cluster E_k in order to facilitate overlaps with the rest of the recordings belonging to that cluster. As a result of event clustering, we obtain the set of representative videos

$$\widetilde{\mathbb{C}} = \{\widetilde{\mathbb{C}}_k : \forall k \in [1, K]\}. \tag{13}$$

5.6. Association of new videos to the database

Let $\mathbb{C}_q$ be a query video to be assigned to an event. Since we already performed event clustering, instead of matching $\mathbb{C}_q$ with $\mathbb{C}$, we perform its matching with $\widetilde{\mathbb{C}}$. The feature vector F_k for all representative video recordings $\widetilde{\mathbb{C}}$ are precomputed using frame size f_{r1}. We compute chroma features F_q for the query video using f_{r1}. The matching histograms V_{kq} and descriptors $P'_{qk} : \forall\ k\ \in [1, K]$ are obtained as discussed in Section 5.4. The descriptors are then mapped on to the classification threshold Γ, which identifies the event cluster E_k containing the set of UGVs having the same overlapping event as $\mathbb{C}_q$.

6. Time-shift estimation and cluster membership validation

Once each event cluster E_k containing the set of overlapping videos is identified, the next step is to synchronize these UGVs on a common timeline. In this section, we present our time-shift estimation and validation approach.

Without loss of generality, let us consider $\mathbb{C}_{k,1} = \widetilde{\mathbb{C}}_k$ as the reference video with the longest duration in E_k. To achieve high precision for the synchronization, the feature vectors $\{F_{k,n}\}_{n=1}^{N_k}$ for $\{\mathbb{C}_{k,n}\}_{n=1}^{N_k}$ are computed using a frame size of $f_{r2} < f_{r1}$ (as discussed in Section 4). Feature matching is then performed between all recording pairs ($N_k \times N_k$) to estimate the synchronization time-shifts, which results in the delay matrix $\mathbb{D} = [D_{ij}]^{N_k \times N_k}$ (Eq. (9)). The delay matrix $\mathbb{D}$ is anti-symmetric ($D_{ij} = -D_{ji}$) if all UGVs are partially or completely overlapping. However, if false positive identification occurs the delay matrix $\mathbb{D}$ might not be anti-symmetric. The analysis of $\mathbb{D}$ is thus required for the validation of the identification results, elimination of any false identifications and for the calculation of consistent time-shifts.

We analyze the delay matrix using the time-shift validation method of Casanovas and Cavallaro [7] for validating the cluster membership. We generate the histogram $h_{ii'}$ where $i \neq i',\ \forall\ i, i' \in [1, N_k]$. The histogram $h_{ii'}$ contains the time-shifts between camera recording i and i' and is given by

$$h_{ii'} = \{(D_{ij} - D_{i'j}) \cup (D_{ji'} - D_{ji})\}, \tag{14}$$

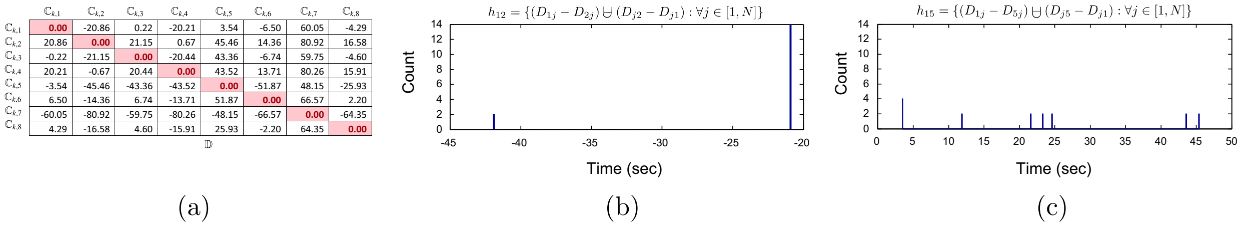
where $j \in [1, N_k]$. The returned $h_{ii'}$ is quantized to the first decimal place for consistency. The most frequently occurring value on this histogram is selected as the consistent time-shift $\Delta t_{ii'}$. A video that does not belong to the same event as the other videos contained in the cluster will have no consistency and this information is used to remove false identifications. Fig. 5 illustrates this validation process with the help of a delay matrix in which video $\mathbb{C}_{k,5}$ is intentionally selected to be different from all other UGVs for the purpose of demonstration.

7. Results

In this section we present the datasets, the experimental setup, the validation of the proposed method for video identification and synchronization, and a comparison with state-of-the-art methods.

7.1. Datasets

We collected 263 multi-camera UGVs of 43 different concert events and 5 different public events such as the Changing of the Guard, the Olympic torch relay, the New Year fireworks and a private dinner (Table 2). In total we collected multi-camera UGVs for 48 events, with a total duration of 1200 min (minutes). The concerts contain multiple recordings: a Nickelback concert, an Evanescence concert and an Alice Cooper concert. The concert recordings contain audio degradations such as noise, background music and crowd cheering. Moreover, these UGVs contain moving cameras with pan and tilt, shake, varying



	$\mathbb{C}_{k,1}$	$\mathbb{C}_{k,2}$	$\mathbb{C}_{k,3}$	$\mathbb{C}_{k,4}$	$\mathbb{C}_{k,5}$	$\mathbb{C}_{k,6}$	$\mathbb{C}_{k,7}$	$\mathbb{C}_{k,8}$
$\mathbb{C}_{k,1}$	**0.00**	-20.86	0.22	-20.21	3.54	-6.50	60.05	-4.29
$\mathbb{C}_{k,2}$	20.86	**0.00**	21.15	0.67	45.46	14.36	80.92	16.58
$\mathbb{C}_{k,3}$	-0.22	-21.15	**0.00**	-20.44	43.36	-6.74	59.75	-4.60
$\mathbb{C}_{k,4}$	20.21	-0.67	20.44	**0.00**	43.52	13.71	80.26	15.91
$\mathbb{C}_{k,5}$	-3.54	-45.46	-43.36	-43.52	**0.00**	-51.87	48.15	-25.93
$\mathbb{C}_{k,6}$	6.50	-14.36	6.74	-13.71	51.87	**0.00**	66.57	2.20
$\mathbb{C}_{k,7}$	-60.05	-80.92	-59.75	-80.26	-48.15	-66.57	**0.00**	-64.35
$\mathbb{C}_{k,8}$	4.29	-16.58	4.60	-15.91	25.93	-2.20	64.35	**0.00**

$\mathbb{D}$

(a)
$h_{12} = \{(D_{1j} - D_{2j}) \uplus (D_{j2} - D_{j1}) : \forall j \in [1, N]\}$
Count
0 2 4 6 8 10 12 14
-45 -40 -35 -30 -25 -20
Time (sec)
(b)
$h_{15} = \{(D_{1j} - D_{5j}) \uplus (D_{j5} - D_{j1}) : \forall j \in [1, N]\}$
Count
0 2 4 6 8 10 12 14
0 5 10 15 20 25 30 35 40 45 50
Time (sec)
(c)

Fig. 5. Cluster membership validation using [7]: (a) example delay matrix for $N = 8$ belonging to the same event, (b) histogram h_{12} for the time-shift between $\mathbb{C}_1$ and $\mathbb{C}_2$, showing its consistency, and (c) histogram h_{15} for the time-shift between $\mathbb{C}_1$ and $\mathbb{C}_5$, showing its inconsistency.

Table 2
Summary of the main characteristics of the dataset along with its challenges. (Key. k: event number; N: number of UGVs; f_v: video frame rate; s: audio sampling rate; –: indicates that only some of the UGVs contain that property).

k	Event title	General characteristics				Challenges				
		N	f_v (fps)	s (kHz)	*Duration* (min:s)	Moving cameras	Varying distance	Channel noise	Ambient noise	Non-amplified sound
1	Nickelback_Event1	7	16–30	44.1	4:01–5:20	✓	✓	–	–	
2	Nickelback_Event2	9	16–30	44.1	4:00–4:42	✓	✓	–	–	
3	Nickelback_Event3	6	24–30	44.1	0:18–4:29	✓	✓	–	–	
4	Nickelback_Event4	7	16–30	44.1	2:26–4:47	✓	✓	–	–	
5	Nickelback_Event5	5	25–30	44.1	3:20–4:56	✓	✓	–	–	
6	Nickelback_Event6	4	25–30	44.1	3:43–4:16	✓	✓	–	–	
7	Nickelback_Event7	6	17–30	44.1	2:01–5:25	✓	✓	–	–	
8	Nickelback_Event8	5	24–30	44.1	1:39–4:06	✓	✓	–	–	
9	Nickelback_Event9	4	24–30	44.1	2:59–8:16	✓	✓	–	–	
10	Nickelback_Event10	4	24–25	44.1	3:37–5:22	✓	✓	–	–	
11	Nickelback_Event11	3	25–30	44.1	1:41–3:35	✓	✓	–	–	
12	Nickelback_Event12	3	24–25	44.1	2:51–4:42	✓	✓	–	–	
13	Nickelback_Event13	3	25	44.1	3:35–4:16	✓	✓	–	–	
14	Nickelback_Event14	3	25–30	44.1	3:29–4:45	✓	✓	–	–	
15	Nickelback_Event15	3	25–30	44.1	4:12–4:42	✓	✓	–	–	
16	Nickelback_Event16	3	25–30	44.1	2:58–3:55	✓	✓	–	–	
17	Nickelback_Event17	3	30	44.1	3:23–3:52	✓	✓	–	–	
18	Nickelback_Event18	2	24–30	44.1	3:09–8:46	✓	✓	–	–	
19	Nickelback_Event19	2	25	44.1	3:48–4:18	✓	✓	–	–	
20	Nickelback_Event20	2	25–30	44.1	4:22–5:04	✓	✓	–	–	
21	Evanescence_Event1	16	25–30	44.1	0:45–5:56	✓	✓	–	–	
22	Evanescence_Event2	7	25–30	44.1	0:59–3:57	✓	✓	–	–	
23	Evanescence_Event3	10	25–30	44.1	2:00–4:47	✓	✓	–	–	
24	Evanescence_Event4	9	24–30	44.1	0:20–4:03	✓	✓	–	–	
25	Evanescence_Event5	6	25–30	44.1	2:57–4:08	✓	✓	–	–	
26	Evanescence_Event6	9	30	44.1	0:55–4:54	✓	✓	–	–	
27	Evanescence_Event7	8	24–30	44.1	2:02–4:04	✓	✓	–	–	
28	Evanescence_Event8	9	24–30	44.1	1:08–5:08	✓	✓	–	–	
29	Evanescence_Event9	4	24–30	44.1	2:27–4:21	✓	✓	–	–	
30	Evanescence_Event10	6	24–25	44.1	1:37–3:32	✓	✓	–	–	
31	AliceCooper_Event1	8	30	44.1	3:12–5:00	✓	✓	–	–	
32	AliceCooper_Event2	11	24–30	44.1	3:07–6:03	✓	✓	–	–	
33	AliceCooper_Event3	2	29–30	44.1	2:38–2:57	✓	✓	–	–	
34	AliceCooper_Event4	3	30	44.1	3:56–4:10	✓	✓	–	–	
35	AliceCooper_Event5	3	25	44.1	3:36–4:28	✓	✓	–	–	
36	AliceCooper_Event6	3	25–30	44.1	3:36–6:41	✓	✓	–	–	
37	AliceCooper_Event7	3	17–30	44.1	3:15–4:0	✓	✓	–	–	
38	AliceCooper_Event8	4	24–30	44.1	1:24–3:04	✓	✓	–	–	
39	AliceCooper_Event9	2	30	44.1	3:26–3:27	✓	✓	–	–	
40	Madonna_Event	11	24–30	44.1	0:28–5:37	✓	✓	–	–	
41	Coldplay_Event	7	24–30	44.1	4:16–7:50	✓	✓	–	–	
42	LesMesirable_Event	7	24–30	44.1	2:33–6:44	✓	✓	–	–	
43	Springsteen_Event	6	24–30	44.1	3:24–6:35	✓	✓	–	–	
44	ChangeofGuard	2	25–30	32–44.1	0:34–2:02	✓	✓		✓	✓
45	OlympicTorchSheffield	2	30	44.1	0:39–1:28	✓	✓		✓	✓
46	OlympicTorchMileEnd	7	16–30	16–48	5:54–7:01	✓	✓	✓	✓	✓
47	XmasDinner	3	30	16	2:35–3:19	✓	✓		✓	✓
48	FireworksLondon	11	25–30	16–44.1	0:29–14:16	✓	✓		✓	✓

visual quality, fields of view and lighting conditions. The public events which we recorded ourselves introduce additional challenges for audio synchronization as they contain considerable ambient noise, moving cameras widely separated from each other and moving audio sources with non-amplified sound. Table 2 summarizes the main characteristics of our datasets along with their challenges. We also collected 60 additional UGVs to be used as the query $\mathbb{C}_q$, which are not overlapping with any of the 48 events but belonged to similar events such as the same concert of Nickelback, Evanescence, and Alice Cooper, the Changing of the Guard in different parts of the world and the Olympic torch relay in different places in the UK.

The ground-truth for video identification and synchronization was generated manually. When observing and matching two UGVs, an annotation error of ±1 video frame (±0.04 s) can occur.

7.2. Experimental setup

For the computation of audio features, the audio signal from a UGV $\mathbb{C}_i$ is segmented into overlapping audio frames Υ_n with hop $h_p = 25\%$ of f_r and frame size $f_{r2} = 0.04\,\text{s}$ for time-shift computation, which gives an accuracy of 0.01 s for

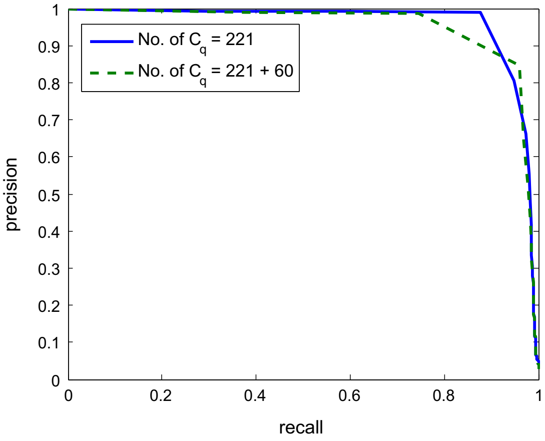


Fig. 6. Video identification framework result showing the performance for two sets of query ($\mathbb{C}_q$): 41 events containing 221 UGVs (which are contained in the database), and 60 additional videos along with 221 UGVs (where the additional UGVs are not contained in the database).

synchronization. For video identification, a value of $f_{r1} = 3.0$ s was found to be an appropriate compromise between efficiency and accuracy. The energy spectrum of the audio frames is computed on the logarithmic scale, where the minimum and maximum are set to 100 Hz and 5000 Hz as proposed by [19]. The computed spectrum energy is then redistributed along the 12 pitch classes (chroma) and matching is performed using the proposed method detailed in Section 5.2. To compute the classification threshold Γ we used a training dataset of 7 events containing 42 UGVs. This dataset gave 1764 matching pairs, out of which 288 belonged to the match class. We trained the classifier by selecting $\kappa_{N_p} = 15$ and $\kappa_{N_n} = 28$ determined using the elbow method for selecting the number of clusters. As a result we obtained the classification threshold Γ.

7.3. Discussion and comparisons

For video identification and event clustering, testing is performed on two sets of UGVs: (a) 41 events containing 221 UGVs which forms our database, (b) 60 additional events along with 221 UGVs (of 41 events) where the additional 60 UGVs are not contained in our database. All (a) 221 × 221 = 48,841 and (b) (221 + 60) × 221 = 62,101 possible match pairs are computed and the ground-truth for video identification is generated. Fig. 6 shows the precision-recall curve for the two test sets. High precision is achieved in both test cases with the area under the precision-recall curve to be 0.97 and 0.96, respectively. This shows the robustness of the proposed framework even with the additional UGVs. Video identification is followed by automatic event clustering using which we identified 41 clusters.

To perform synchronization, we use the complete dataset of 48 events (263 UGVs) for the evaluation, as we are interested in synchronizing all the events. The synchronization results are shown in Fig. 7(a). Despite several challenges, all videos are synchronized with errors between estimated and ground-truth time-shifts smaller than 0.03 s. The proposed synchronization approach is even effective for videos of a short duration (as analyzed in Section 5.3) and fails to correctly show the time-shifts for only one UGV (belonging to Olympic Torch Mile End dataset) out of the 263 in the test. The error is due to

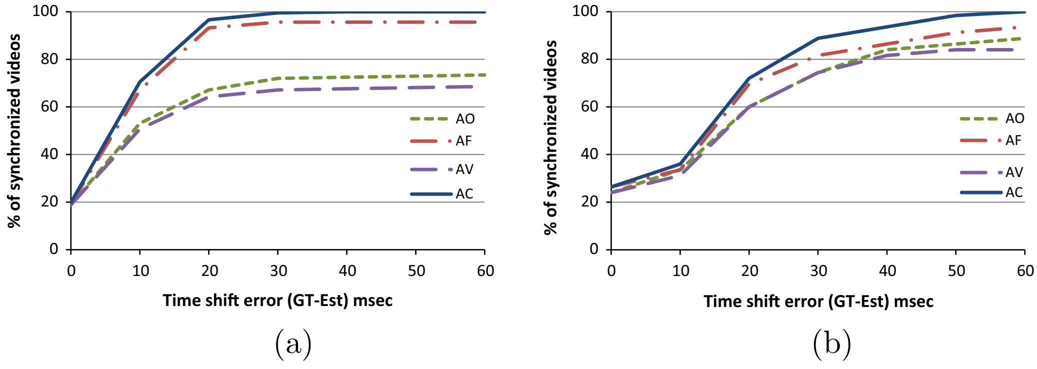


Fig. 7. Comparison results showing the percentage of synchronized videos versus time-shift error with respect to the ground truth. (a) Synchronization results on the whole dataset (Table 2). (b) Synchronization results for the dataset used in [7,43]. Key: AO indicates the audio onset based method [43]; AF indicates the audio fingerprinting method [43]; AV indicates the audio-visual event method [7]; AC indicates the proposed method.

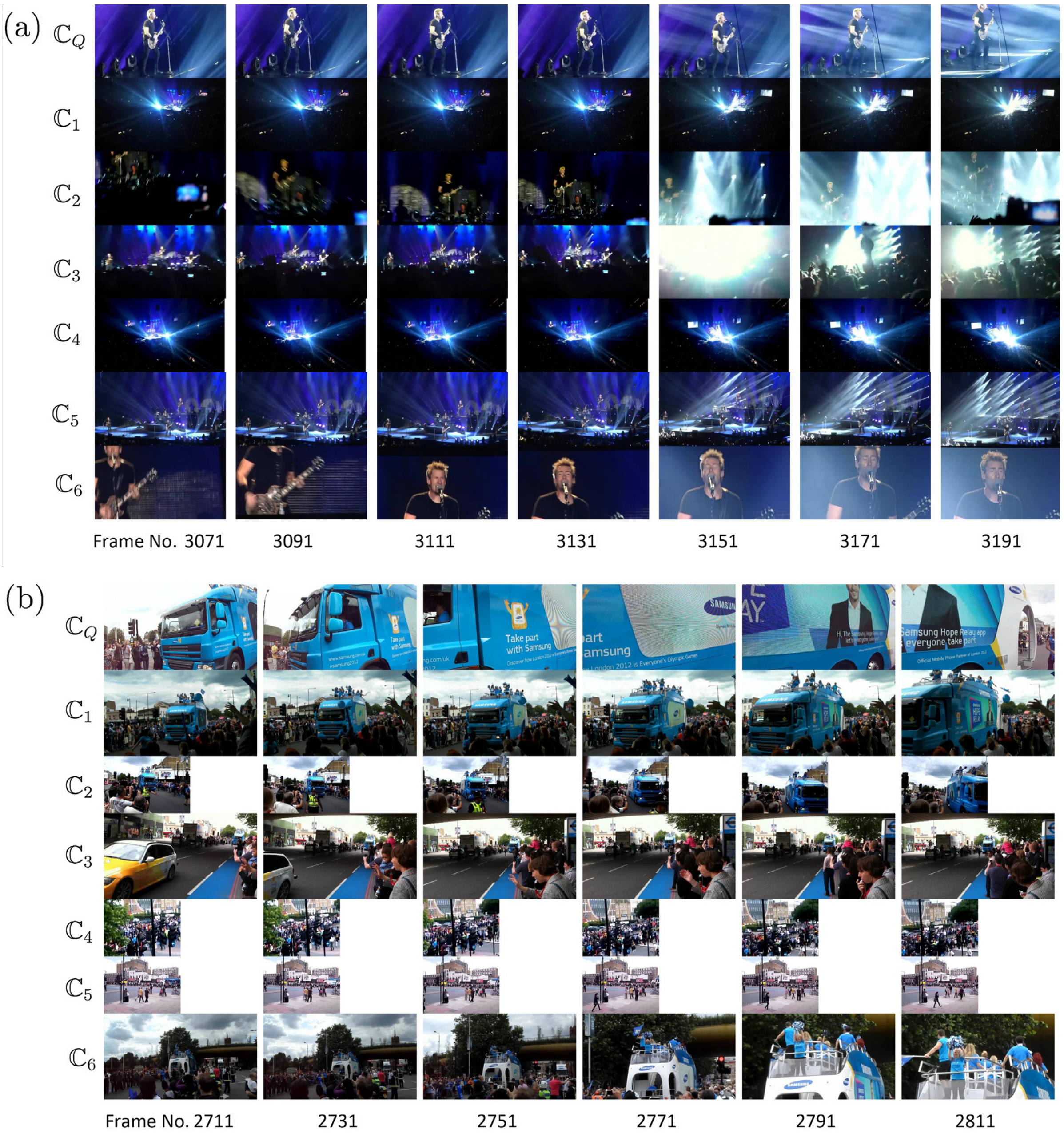


Fig. 8. The association and synchronization result for (a) a concert (Nickelback_Event1 as named in Table 2) and (b) the Olympic torch (OlympicTorch-MileEnd as named in Table 2) event. Row 1 represents a snapshot frame from the query video. Each row represents a different video from the identified cluster event. Each column corresponds to temporally aligned frames from the videos.

the recording device malfunctioning and not capturing the audio signal for most of the time during recording. Time-shift validation is also performed in order to verify that the obtained cluster of recordings belongs to the same event.

In order to further validate our proposed *Audio Chroma (AC)* based synchronization method, we compare it with state-of-the-art methods based on *Audio Onset (AO)* [43], *Audio Fingerprinting (AF)* [43] and *Audio-Visual Event (AV)* [7] using our dataset (Fig. 7(b)). *AO* and *AV* are comparable, while at times *AV* gave slightly worse results than *AO*. Since these two methods are highly sensitive to audio degradations, they failed to synchronize a large number of UGVs. Likewise, Audio Fingerprinting (*AF*) [43] is robust to ambient noise but failed to give the correct result for some recordings containing reverberations and channel noise. Furthermore, *AF* failed to synchronize UGVs of a short duration (<30 s). *AC* outperformed the other three methods as it was able to synchronize 262 out of 263 UGVs, followed by AF, giving an overall accuracy of 99.62% and 94.79%, respectively.

To have a fair comparison with the state of the art, we also perform testing with the dataset used in [7,43] (Fig. 7(b)). The same trend can be observed as for our dataset: the results obtained with *AC* and *AF* are comparable, but *AC* outperforms the other methods. The best overall performance is achieved by *AC*, followed by *AF*, *AO* and *AV*.

The association and synchronization for a concert (Nickelback_Event1) and the Olympic torch (OlympicTorchMileEnd) event are shown in Fig. 8, where row one shows $\mathbb{C}_q$ and the identified cluster videos are shown in the subsequent rows. Each column represents the synchronized frame for these video recordings. Note the different visual quality ($\mathbb{C}_4$ and $\mathbb{C}_5$ in Fig. 8(b)), variations in the field of views ($\mathbb{C}_1$ and $\mathbb{C}_4$ shows far field of views as compared to $\mathbb{C}_q$ and $\mathbb{C}_6$ in Fig. 8(a)), lighting ($\mathbb{C}_2$ and $\mathbb{C}_3$ in Fig. 8)) and camera motion ($\mathbb{C}_6$ in Fig. 8(a) and (b) showing zooming in motion) in the snapshot frames.

To test the robustness of the proposed framework, association is also performed using similar UGVs (using an additional dataset of 60 UGVs, which is detailed in Section 7.1). Though depicting similar events but with no time overlap, no event cluster is identified when performing association with these additional UGVs. This is also shown in Fig. 6, which further validates the robustness of our framework.

8. Conclusions

We presented an automatic identification and synchronization framework for multi-camera UGVs and query-by-example video event search. The proposed approach uses audio chroma features to cluster UGVs belonging to the same event and to estimate their relative time-shift. Unlike existing identification approaches [5,28], we proposed an automatically determined classification threshold for clustering and association of new incoming videos. We demonstrated the robustness of the proposed method to audio degradations including high ambient and channel noise, and discussed a comparative analysis with existing state-of-the-art methods.

As future work, we are interested in generating a new cluster for a query video for which a matching UGV does not exist in the dataset. Also, the audiovisual content uploaded on media sharing websites will increasingly be accompanied by additional information from other sensors embedded in the recording devices [13,14,24], we will analyze these multimodal signals in order to increase the efficiency of the analysis and event clustering.

References

[1] M.A. Bartsch, G.H. Wakefield, Audio thumbnailing of popular music using chroma-based representations, IEEE Trans. Multimedia 7 (1) (2005) 96–104.

[2] H. Becker, M. Naaman, L. Gravano, Event identification in social media, in: Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB09), Rhode Island, USA, 2009.

[3] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, M. Sandler, A tutorial on onset detection in music signals, IEEE Trans. Speech Audio Process. 13 (5) (2005) 1035–1047.

[4] Marie-Luce Bourguet, J. Wang, A robust audio feature extraction algorithm for music identification, in: Proceedings of 129th Audio Engineering Society Convention, San Francisco, CA, 2010, pp. 8180 – 8189.

[5] N. Bryan, P. Smaragdis, G. Mysore, Clustering and synchronizing multi-camera video via landmark cross-correlation, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 2012, pp. 2389–2392.

[6] P. Cano, E. Batlle, T. Kalker, J. Haitsma, A review of audio fingerprinting, J. VLSI Signal Process. 41 (3) (2005) 271–284.

[7] A.L. Casanovas, A. Cavallaro, Audio-visual events for multi-camera synchronization, Multimedia Tools Appl. (2014) 1–24.

[8] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, M. Slaney, Content-based music information retrieval: current directions and future challenges, Proc. IEEE 96 (4) (2008) 668–696.

[9] C. Chen, R. Cook, M. Cremer, P. DiMaria, Content identification in consumer applications, in: Proceedings of IEEE International Conference on Multimedia and Expo (ICME), New York, USA, 2009, pp. 1536–1539.

[10] P.R. Cook, Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics, The MIT Press, 2001 (Chapter 13).

[11] C. Cotton, D. Ellis, Audio fingerprinting to identify multiple videos of an event, in: Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Texas, USA, 2010, pp. 2386–2389.

[12] E. Coviello, L. Barrington, A. Chan, G. Lanckriet, Automatic music tagging with time series models, in: Proceedings of 11th International Society for Music Information Retrieval (ISMIR) Conference, Utrecht, Netherlands, 2010, pp. 81–86.

[13] F. Cricri, I. Curcio, S. Mate, K. Dabov, M. Gabbouj, Sensor-based analysis of user generated video for multi-camera video remixing, Adv. Multimedia Model. 7131 (2012) 255–265.

[14] F. Cricri, M. Roininen, J. Leppanen, S. Mate, I.D. Curcio, S. Uhlmann, M. Gabbouj, Sport type classification of mobile videos, IEEE Trans. Multimedia 16 (4) (2014) 917–932.

[15] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al., The youtube video recommendation system, in: Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 2010, pp. 293–296.

[16] D. Ellis, G. Poliner, Identifying 'cover songs' with chroma features and dynamic programming beat tracking, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hawaii, USA, vol. 4, 2007, pp. 1429–1432.

[17] S. Ewert, M. Muller, P. Grosche, High resolution audio synchronization using chroma onset features, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Taipei, Taiwan, 2009, 2009, pp. 1869–1872.

[18] T. Fujishima, Realtime chord recognition of musical sound: a system using common lisp music, in: Proceedings of International Computer Music Conference (ICMC), Beijing, China, 1999, 1999, pp. 464–467.

[19] E. Gutiérrez, Tonal Description of Music Audio Signals, Ph.D. Thesis, Universitat Pompeu Fabra, 2006.

[20] J. Haitsma, T. Kalker, A highly robust audio fingerprinting system with an efficient search strategy, J. New Music Res. 32 (2) (2003) 211–221.

[21] R. Hong, J. Tang, H.-K. Tan, C.-W. Ngo, S. Yan, T.-S. Chua, Beyond search: event-driven summarization for web videos, ACM Trans. Multimedia Comput. Commun. Appl. (TOMCCAP) 7 (4) (2011) 35.

[22] T. Hung, C. Zhu, G. Yang, Y. Tan, Video organization: near-duplicate video clustering, in: Proc. of IEEE International Symposium on Circuits and Systems (ISCAS), Seoul, Korea, 2012, pp. 1879–1882.

[23] I. Ivanov, P. Vajda, J. Lee, T. Ebrahimi, In tags we trust: trust modeling in social tagging of multimedia content, IEEE Signal Process. Mag. 29 (2) (2012) 98–107.

[24] P. Jain, J. Manweiler, A. Achary, K. Beaty, Focus: clustering crowdsourced videos by line-of-sight, in: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, Rome, Italy, 2013, pp. 63–76.

[25] N. Jiang, P. Grosche, V. Konz, M. Müller, Analyzing chroma feature types for automated chord recognition, in: Proceedings of AES 42nd Conference on Semantic Audio, Ilmenau, Germany, 2011, pp. 1–10.

[26] T. Joachims, Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms, Kluwer Academic Publishers, 2002.

[27] J. Kammerl, N. Birkbeck, S. Inguva, D. Kelly, A. Crawford, H. Denman, A. Kokaram, C. Pantofaru, Temporal synchronization of multiple audio signals, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 4636–4640.

[28] L. Kennedy, M. Naaman, Less talk, more rock: automated organization of community-contributed collections of concert videos, in: Proceedings of the 18th ACM International Conference on World Wide Web, Madrid, Spain, 2009, pp. 311–320.

[29] S. Kenyon, L. Simkins, et al., Audio Identification System and Method, US Patent 7,783-489, August 24 2010.

[30] D.J. Ketchen, C.L. Shook, The application of cluster analysis in strategic management research: an analysis and critique, Strategic Manage. J. 17 (6) (1996) 441–458.

[31] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, F. Stentiford, Video copy detection: a comparative study, in: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, Amsterdam, Netherlands, 2007, pp. 371–378.

[32] J. Liu, Z. Huang, H. Cai, H.T. Shen, C.W. Ngo, W. Wang, Near-duplicate video retrieval: current research and future trends, ACM Comput. Surv. (CSUR) 45 (4) (2013) 44.

[33] X. Liu, R. Troncy, B. Huet, Finding media illustrating events, in: Proceedings of the 1st ACM International Conference on Multimedia Retrieval, Trento, Italy, 2011, pp. 58–66.

[34] X. Lou, Feature Extraction for Identification and Classification of Audio Signals, US Patent 8,140-331, March 20 2012.

[35] J. Lu, Video fingerprinting for copy identification: from research to industry applications, in: Proceedings of SPIE Media Forensics and Security, San Jose, CA, USA, 2009, pp. 2–15.

[36] M. Mauch, S. Ewert, The audio degradation toolbox and its application to robustness evaluation, in: Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR), Curitiba, Brazil, 2013, pp. 83–88.

[37] M. McKinney, J. Breebaart, Features for audio and music classification, in: Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR), Maryland, USA, vol. 3, 2003, pp. 151–158.

[38] M. Müller, Information Retrieval for Music and Motion, Springer, 2007.

[39] M. Müller, F. Kurth, M. Clausen, Audio matching via chroma-based statistical features, in: Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR), London, UK, 2005, pp. 288–295.

[40] M. Naaman, Social multimedia: highlighting opportunities for search and mining of multimedia data in social media applications, Multimedia Tools Appl. 56 (1) (2012) 9–34.

[41] L. Shang, L. Yang, F. Wang, K. Chan, X. Hua, Real-time large scale near-duplicate web video retrieval, in: Proceedings of the 18th ACM International Conference on Multimedia, Firenze, Italy, 2010, pp. 531–540.

[42] P. Shrestha, M. Barbieri, H. Weda, Synchronization of multi-camera video recordings based on audio, in: Proceedings of the 15th ACM International Conference on Multimedia, Bavaria, Germany, 2007, pp. 545–548.

[43] P. Shrestha, M. Barbieri, H. Weda, D. Sekulovski, Synchronization of multiple camera videos using audio-visual features, IEEE Trans. Multimedia 12 (1) (2010) 79–92.

[44] J. Song, Y. Yang, Z. Huang, H. Shen, R. Hong, Multiple feature hashing for real-time large scale near-duplicate video retrieval, in: Proceedings of the 19th ACM International Conference on Multimedia, Scottsdale, AZ, USA, 2011, pp. 423–432.

[45] A.S. Thakur, N. Sahayam, Speech recognition using euclidean distance, Int. J. Emerging Technol. Adv. Eng. 3 (3) (2013) 587–590.

[46] A. Wang, The Shazam music recognition service, Commun. ACM 49 (8) (2006) 44–48.

[47] A. Wang, et al., An industrial strength audio search algorithm, in: 4th International Conference on Music Information Retrieval, Baltimore, Maryland, USA, 2003, pp. 7–13.

[48] L. Wang, Support Vector Machines: Theory and Applications, vol. 17, Springer, 2005.