Multimedia Tools and Applications

Audio-Visual Events for Multi-Camera Synchronization

Anna Llagostera Casanovas · Andrea Cavallaro

Abstract We present a multimodal method for the automatic synchronization of audio-visual recordings captured with a set of independent cameras. The proposed method jointly processes data from audio and video channels to estimate inter-camera delays that are used to temporally align the recordings. Our approach is composed of three main steps. First we extract from each recording temporally *sharp* audio-visual events. These audio-visual events are short and characterized by an audio onset happening jointly to a well-localized spatio-temporal change in the video data. Then, we estimate the inter-camera delays by assessing the co-occurrence of the events in the various recordings. Finally, we use a cross-validation procedure that combines the results for all camera pairs and aligns the recordings in a global timeline. An important feature of the proposed method is the estimation of the confidence level on the results that allows us to automatically reject recordings that are not reliable for the alignment. Results show that our method outperforms state-of-the-art approaches based on audio-only or video-only analysis with both fixed and hand-held moving cameras.

Keywords Audio-visual processing · multiple cameras · synchronization · event detection

A. Llagostera Casanovas contributed to this work while at Queen Mary University of London, UK. She was supported by the Swiss National Science Foundation under the prospective researcher fellowship PBELP2-137724. A. Cavallaro acknowledges the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/K007491/1

Anna Llagostera Casanovas
SwissQual AG, Switzerland
E-mail: anna.llagostera@swissqual.com

Andrea Cavallaro
Centre for Intelligent Sensing, Queen Mary University of London, UK
E-mail: andrea.cavallaro@eecs.qmul.ac.uk

1 Introduction

The widespread availability of video cameras and the growing distribution of user-generated videos through large public repositories, such as YouTube, is favoring many applications including entertainment, citizen journalism and forensic search. Multi-camera recordings provide complementary information on the same scene from different viewpoints. These recordings can be combined to provide new views, to generate simultaneous visualizations of multiple streams or simply to select the best view over time [4]. Typically, cameras are controlled from different locations by different people who might start and stop recording at different time instants. For this reason, an important problem that one encounters prior to combining multi-camera recordings is their synchronization.

Multi-camera synchronization in ad-hoc professional settings is performed using a genlock or a clapperboard. The *genlock* is a reference signal shared among all the cameras in a network that provides them the same recording time code. The *clapperboard* is a reference object that generates a temporally sharp movement and sound that are captured by all cameras in the network. This information is then used for synchronizing the recordings of all cameras and also for synchronizing the audio and video channels within each camera. However, such settings are generally not available for non-professional recordings. For this reason, video editing tools [1, 6] require user interaction to locate in the recordings events that can be used for manual multi-camera synchronization.

In this paper, we propose a method for the automated synchronization of multiple camera recordings that emulates the usefulness of a clapperboard, when such an object is not available. The proposed method identifies audiovisual events in a scene and uses them as anchors to synchronize cameras. Unlike previous methods, the proposed method (i) jointly processes audio and video features; (ii) produces and verifies a global synchronization for three or more cameras; and (iii) generates a reliability estimate of the accuracy of the result for each camera. This estimate allows the method to discard inaccurate recordings and to adapt the features used for synchronization in order to use the most reliable ones. We also introduce a cross validation procedure to combine partial results obtained from the analysis of each camera pair in order to position all recordings on a common timeline, thus enabling synchronization when some of the recordings are not overlapping with others. The proposed method is applicable to moving and handheld cameras that are filming the same scene, when similar sounds and related views are recorded.

This paper is organized as follows. Section 2 reviews the state-of-the-art approaches in multi-camera alignment. In Sec. 3 we define the synchronization problem. Section 4 discusses the concept of audio-visual event, whereas in Sec. 5 we describe the proposed procedure for its extraction. In Sec. 6 we present the matching strategy that combines the detected audio-visual events and estimates the shifts among recordings. Sec. 7 presents results and comparisons. Finally, Sec. 8 concludes the paper.

2 Prior work

Multi-camera synchronization approaches generally extract audio or video features from each camera and then match the features to determine temporal shifts (i.e. the offset) among recordings. Based on the analyzed modality, multi-camera synchronization approaches can be divided into two classes, namely video-based approaches and audio-based approaches.

Approaches that analyze the *video* information use global changes or multi-view geometry (MVG). An approach using *global changes* happening simultaneously in all cameras is presented in [18], which is based on the detection of still-camera flashes. Time instants at which a flash is detected are matched using dynamic programming in order to estimate the offset among cameras. Although this approach does not constrain camera geometry or motion, as it relies on the detection of global events, its applicability is limited as it requires the presence of flashes. Most methods that use video information are based on MVG, and align sequences both in space and time by estimating the fundamental matrix and the time shift among cameras [2, 12, 13, 15, 20, 23, 26]. MVG-based methods can be divided into two main groups, namely direct alignment methods and feature-based alignment methods. *Direct alignment methods* compare pixel intensities in video sequences from different cameras and estimate the transformation that minimizes the differences among recordings [2, 23]. As a result, direct methods obtain more accurate alignment results when the pixels intensities are similar. In contrast, *feature-based alignment methods* extract object trajectories [2, 13, 15, 20, 26] or space-time interest points [12, 25, 27] and use robust algorithms such as RANSAC-like approaches [2, 12, 15, 25] or Least Median Squares (LMS) [20]. In this case the goal is to estimate the correspondence between features from different recordings by discarding outliers in order to extract the spatial and temporal alignment parameters of the cameras. Feature-based methods perform better when the scene presents important changes among cameras. In general, MVG approaches assume the camera network geometry to be approximately stationary, i.e. either the cameras are do not change their position or they move jointly. This characteristic makes these methods unsuitable for non-professional videos captured by people recording the scene independently.

Approaches based on *audio* information generally use two types of features for synchronization, namely audio fingerprints and energy onsets. *Audio fingerprints* describe concisely the frequency characteristics of a recorded sound. Their main strength lies in their robustness to noise [3, 10, 18]. *Audio onsets* represent the time instants when sounds start and can be computed over the whole signal energy or for partial energy on frequency bands. The use of frequency bands increases the robustness of onset detections and may help reducing the effect of noise. Similar synchronization results can be achieved with audio fingerprints and onsets extracted from multiple frequency bands [18].

Audio and video information have also been used in parallel [18]: two audio features (fingerprints and onsets) and one video feature (global changes due to flashes) are considered separately, with no exchange of information among modalities. Our proposed approach is the first jointly multi-modal approach that combines audio and video processing for multi-camera synchronization. The main characteristics of state-of-the-art methods and the datasets used in the literature are summarized in Table 1.

Table 1 Comparative summary of state-of-the-art methods for multi-camera synchronization. Key: STIP - Space-Time Interest Points; RANSAC - RANdom SAmple Consensus. '*' indicates the special case that requires three cameras, of which two need to be fixed. The letters a, b and c in the Ref. column denote different approaches proposed in the same paper.

Ref.	Features						Matching Strategy	Multi-camera cross-validation	Confidence estimation	Dataset Characteristics			
	Video				Audio							Video	
	Trajectories	Pixel intensities	STIP	Global brightness variation	Onsets	Fingerprints				Number of scenes	Total number of recordings	Independent camera motion	Different distances
[20]	✓						Least Median Squares (LMS)			1	3		
[13]	✓						Tri-view geometry constraints			4	12	*	
[15]	✓						Timeline constraint + RANSAC			4	9		
[2]a	✓						Heuristics on the trajectories + RANSAC			3	6		✓
[2]b		✓					Sum of Squared Differences (SSD) minimization			4	8		
[23]		✓					Normalized Correlation (NC) maximization			2	4		
[27]			✓				Cross-correlation maximization			4	8		
[25]			✓				Local jets similarity + RANSAC			2	4		
[18]a				✓			Dynamic programing			7	30	✓	✓
[18]b					✓		Cross-correlation maximization			7	30	-	-
[18]c						✓	Bit error rate minimization			7	30	-	-
[10]						✓	Hash values similarity maximization			3	608	-	-
This work		✓			✓		Cross-correlation maximization	✓	✓	8	40	✓	✓

3 Problem formulation

Let a scene be recorded by a set $\mathbf{C} = \{C^1, \ldots, C^M\}$ of M cameras. Each camera C^i is composed of an image sensor and audio sensors (microphones) whose sampling rates are s_V^i (in frames per second) and s_A^i (in Hz), respectively. For simplicity, we consider here that each C^i has only one microphone. Let $v(\mathbf{x}^i)$ be the video signal corresponding to C^i with spatio-temporal coordinates $\mathbf{x}^i = (x^i, y^i, t_V^i)$ and let $a(t_A^i)$ be the audio signal from C^i with temporal coordinate t_A^i [1]. The conversion from the video and audio time indexes (t_V^i and t_A^i) to the universal time t^i (in seconds) for each camera C^i is performed as

$$t^i = t_V^i / s_V^i \,, \quad t^i = t_A^i / s_A^i \,, \tag{1}$$

where t_V^i and t_A^i are integers denoting the frame and sample indexes, respectively.

The time shift (offset) between *video* recordings from cameras C^i and C^j (in seconds) can be computed as

$$\Delta t_{(V)}^{ij} = \frac{t_V^i}{s_V^i} - \frac{t_V^j}{s_V^j} \,. \tag{2}$$

Analogously, the time shift (in seconds) among *audio* recordings from cameras C^i and C^j is defined by

$$\Delta t_{(A)}^{ij} = \frac{t_A^i}{s_A^i} - \frac{t_A^j}{s_A^j} \,. \tag{3}$$

[1] t_V denotes the discrete temporal coordinate of the video signal (in frames) and t_A corresponds to the discrete temporal coordinate of the audio signal (in samples).

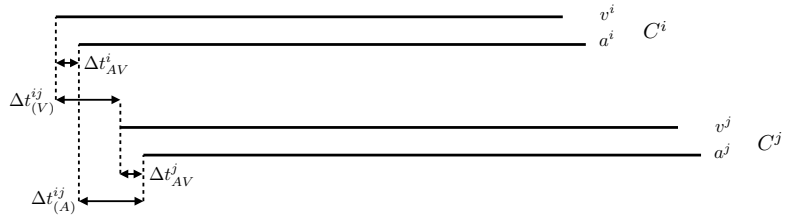

Fig. 1 Time shifts among cameras and modalities within the same camera. The relative positioning of audio and video recordings (solid lines) corresponding to each camera indicates the time shift.

Even if audio and video recordings corresponding to the same camera start approximately at the same time and are synchronized through the recording process, there is usually a small delay among channels. Thus Δt^i_{AV}, the time shift (in seconds) across the two modalities of C^i, can be computed as

$$\Delta t^i_{AV} = \frac{t^i_A}{s^i_A} - \frac{t^i_V}{s^i_V} \,. \tag{4}$$

$\Delta t^{ij}_{(V)} \approx \Delta t^{ij}_{(A)}$ if the time shift between audio and video modalities in cameras C^i and C^j is sufficiently small. This is the case when there is no perceived drift between audio and video modalities: the thresholds for the detectability of the time shift between modalities are about $+45$ ms to -125 ms [17], where the positive value denotes an advance of the sound with respect to the corresponding image.

A schematic visualization of the time shifts between cameras and sensors (modalities) within each camera is shown in Fig. 1. The multi-camera synchronization problem consists in estimating the time shifts Δt^{ij} , $\forall i, j = 1 \dots M$ among recordings generated by the set of M cameras in order to align them on a common timeline.

4 What is an Audio-Visual Event?

We define as audio-visual event a simultaneous change in the audio and video channels. An audio-visual event is well-localized in time and can be exploited for synchronizing recordings by emulating the sound and motion generated with a clapperboard in professional settings.

Audio-visual events can happen by generation, by reaction or by co-occurrence. In audio-visual events *by generation* (Fig. 2(a)) the audio and video signals are physically related: the movement of an object generates a sound that is captured by the microphones. Examples of this type of audio-visual events are the movement of the lips of a speaker and the movement of a drummer over a percussion instrument. Audio-visual events *by reaction* (Fig. 2(b)) happen when an event in one modality generates a reaction that is observable in the other one, synchronously or almost synchronously. Examples are the movements of a dancer correlated with the music (sound causing motion) and the scream of a person when an object is thrown over him (motion causing sound). Finally, audio-visual events *by co-occurrence* (Fig. 2(c)) are composed of audio events and video events that are not directly

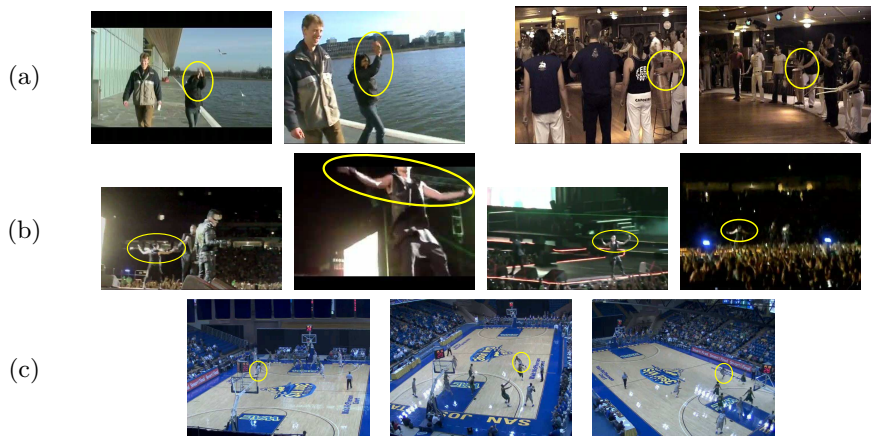

Fig. 2 Different types of audio-visual events (highlighted with ovals) observed simultaneously by different cameras. (a) Audio-visual events *by generation* that are physically related when movement produces sounds, e.g. when clapping or hitting a drum. (b) An audio-visual event *by reaction*, when motion is a consequence of sound, e.g. during a dance movement. (c) An audio-visual event *by co-occurrence*, when audio and video are not directly related, e.g. when the movement of a basketball player and the audience cheering are synchronous.

related, but happen in synchrony. These events are difficult to model and to exploit, but they can be useful for the synchronization task. An example is the absence or presence of motion and sounds in a basketball match, that might indicate that the game is paused or going on. As long as all C^i capture the same sounds and motion patterns, the coherence among observations will help synchronizing the recordings. In fact, when events take place simultaneously in the audio and video domains, they become more robust indicators than audio-only or video-only events.

Synchrony of related events in the audio and video modalities has already been exploited for joint audio-visual analysis in several other applications. Examples include speech recognition [16], sound source localization [11], audio [19] and audio-visual [14] source separation, video content classification [9], and robotics [7]. These approaches are based on psychophysical studies that show the relationship between sounds and motion [21, 22, 24]. To the best of our knowledge, our proposed method is the first that exploits the joint analysis of these two modalities for multi-camera synchronization.

5 Audio-Visual Event Detection

The proposed process for the extraction of audio-visual events is divided into three main steps. First, we build an activation vector for the audio modality that indicates the presence of a sound. Then, we search in the frames for local motion peaks that match the presence of the sound. Finally, we construct the audio-visual activation vector for each recording to mark the simultaneous presence of events in the audio and video modalities. This process is summarized in Fig. 3 and discussed in the following sections.

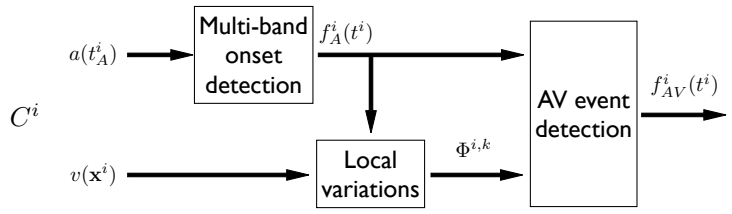

Fig. 3 Block diagram of the proposed approach for the estimation of audio-visual activation vectors for a single camera C^i. First, we compute an audio activation vector $f_A^i(t^i)$ that captures the presence of onsets (audio events) in frequency bands. Then, we extract a set of video blocks $\Phi^{i,k}$ presenting an activation (motion peak) synchronous with an audio event. Finally, the audio-visual activation vector $f_{AV}^i(t^i)$ is computed based on the presence of audio events and the number of associated video events.

5.1 Audio event detection

To detect the presence of an audio event, we first divide the audio signal of each C^i into 8 non-overlapping frequency bands. These bands are defined according to the equivalent rectangular bandwidth (ERB) scale and cover the frequency range between 20 Hz and 6200 Hz [18]. In our approach, the energy measurement in each band is computed in 0.5-second window at the maximum frame rate among cameras. Then, we define that an audio event is occurring when an onset is detected at time $t^{i,k}$ in multiple bands, where $t^{i,k}$ denotes the time index in which onset k takes place in the audio recording of C^i. To detect onsets we use a peak detector whose threshold E_b is related to the average audio energy $\bar{A}_b$ in the corresponding frequency band b as:

$$E_b = 0.005 \cdot \bar{A}_b \,. \tag{5}$$

An adaptive thresholding is necessary because different bands present different energy levels. Next, we extract a set o_A^i of K audio events that are consistent across frequency bands:

$$o_A^i = \{t^{i,k}\}_{k=1}^{K} \,. \tag{6}$$

Finally, we build a binary audio activation vector $f_A^i(t^i)$ to encode the presence of these audio events (energy onsets):

$$f_A^i(t^i) = \begin{cases} 1 & \text{if} \quad t^i = t^{i,k}, \;\; \forall k = 1, \ldots, K \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

5.2 Detection of synchronous local video events

We define a synchronous video event as a region in the camera's field of view that presents a strong local variation (motion) at approximately the same time as the occurrence of a sound onset, detected as described in the previous section. We choose local variations so that we can determine *if* an event takes place without the need to define *what* the event is. This enables the applicability of the proposed method to a broad range of scenarios.

To extract local video events that take place simultaneously to audio events, for each audio onset $k = 1, \ldots, K$ in C^i, we extract a set of video regions that are

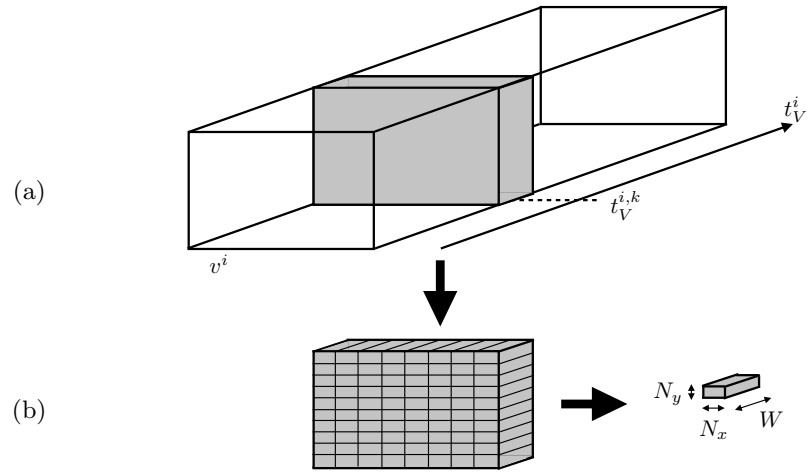

Fig. 4 Example of space-time block around an audio onset. (a) A slice of the video signal is extracted in a time window of size W around the equivalent in frames of the audio onset time $t_V^{i,k}$. (b) The spatio-temporal slice is divided into blocks of spatial dimensions N_x and N_y.

active at *approximately* the same time $\Phi^{i,k}$ (Fig. 4). The video signal around audio onset k is divided into a set of L blocks to allow the study of the variations in sub-regions of the field of view of the camera. N_x, N_y and W are the dimensions of a block in x, y and t_V, respectively. The video time index $t_V^{i,k}$ (in frames) corresponding to the audio onset time $t^{i,k}$ is approximated as

$$t_V^{i,k} = \mathcal{N}(s_V^i \cdot t^{i,k}) \,, \tag{8}$$

where $\mathcal{N}(\cdot)$ denotes the nearest integer function and s_V^i is the sampling rate of the video signal.

The total amount of variation $m^{i,k,l}$ for each block l and time index $t_V^{i,k}$ is computed as the sum of absolute differences (SAD) among consecutive frames in this spatio-temporal block as

$$m^{i,k,l}(w^{i,k}) = \sum_{(x,y)\in\mathcal{L}} |v^i(x,y,w^{i,k}) - v^i(x,y,w^{i,k}-1)| \,, \tag{9}$$

where $\mathcal{L}$ denotes the part of the image domain that composes block l, and

$$w^{i,k} \in [t_V^{i,k} - W/2, t_V^{i,k} + W/2] \tag{10}$$

is the temporal window around onset k in which the absolute variation is computed.

Then, we estimate that a video event takes place in block l in synchrony with the audio event if the total amount of variation in the block has its global maximum at the same time as the onset, *within a certain temporal tolerance* T_{AV} between modalities. This temporal tolerance allows us to capture related events that are not perfectly synchronous, e.g. the lips motion and the resulting speech sounds or the hand of a guitarist and the corresponding sound generated by the strings.

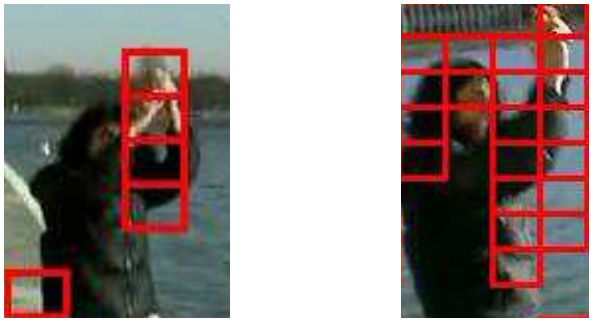

Fig. 5 Example of active video blocks (rectangles) for an audio-visual event observed by two cameras. The relative position and angle between the camera and the moving object determines the number of blocks that become active.

Thus, the set of video blocks $\Phi^{i,k}$ that are active in the recording corresponding to C^i and approximately synchronous with audio onset k is defined as:

$$\Phi^{i,k} = \left\{ l \in \{1 \dots L\} : |t_V^{i,k} - \underset{w^{i,k}}{\operatorname{argmax}}\, m^{i,k,l}| < T_{AV} \right\} . \tag{11}$$

T_{AV} is the maximum time shift allowed between audio and video channels in the same camera (see Fig. 1) and thus we should ensure $T_{AV} > \Delta t_{AV}^i, \ \forall i$. Figure 5 shows an example of active video blocks corresponding to an audio-visual event captured by two cameras in different locations. In each camera the peak in the motion is captured with a different number of blocks.

To avoid the detection of *single-camera* audio-visual events caused for example by camera motion, the presence of an audio-visual event is determined by the number of regions that present an activation (motion peak) approximately simultaneous with the audio energy peak. Let $L_{act}^{i,k}$ be the number of video blocks that are active during audio onset k in C^i:

$$L_{act}^{i,k} = \|\Phi^{i,k}\|_0 \tag{12}$$

and γ be the parameter that determines the maximum proportion of active video blocks to estimate that an audio-visual event occurs in the observed scene. Then we consider that an audio-visual event takes place at time t^k (in seconds) if

$$1 \leq L_{act}^{i,k} \leq \gamma L \, , \tag{13}$$

The upper bound in the number of active video blocks is introduced to discard the effect of camera motion.

As a result, we have now a set of audio-visual events defined as

$$o_{AV}^i = \{t^{i,h}\}_{h=1}^H \, , \tag{14}$$

with $H \leq K$, since the audio-visual events are the subset of audio-only events with an associated video event.

The audio-visual activation vector $f_{AV}^i(t^i)$ is non-zero only for time indexes (in seconds) when an audio-visual event occurs:

$$f_{AV}^i(t^i) = \begin{cases} 1 \text{ if } \ t^i = t^{i,h}, \ \ \forall h = 1, \dots, H \\ 0 \text{ otherwise} \, . \end{cases} \tag{15}$$

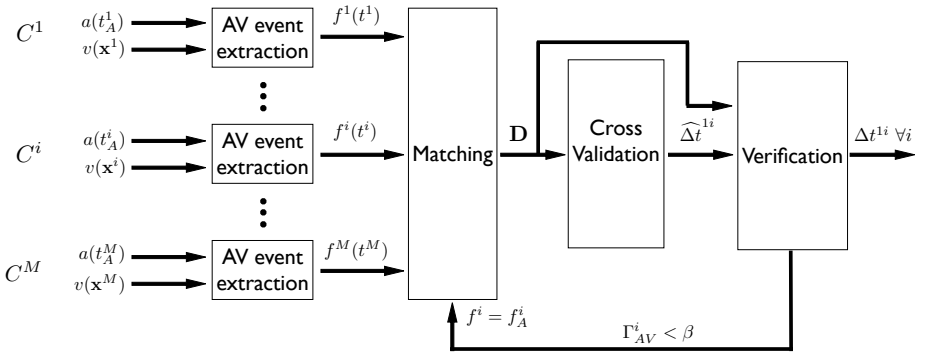

Fig. 6 Block diagram of the proposed approach for the synchronization of multi-camera audio-visual recordings. First, audio-visual activation vectors f_{AV}^i indicating the presence of audio-visual events are extracted from each camera recording. Then, a delay matrix $\mathbf{D}$ containing the time shifts between each pair of recordings is computed from the audio-visual activation vectors ($f^i = f_{AV}^i$) obtained in the previous step using cross-correlation. Next, this information is used in the cross-validation step to obtain time-shift estimates that are consistent among all cameras and sort the recordings on a global timeline. A final verification step automatically detects unreliable audio-visual results and opts for an audio-only analysis ($f^i = f_A^i$) if preferable.

6 Multi-camera synchronization

Given the set of activation vectors capturing the presence of audio-visual events in each C^i, our goal is to combine them to estimate the time shifts among recordings and achieve synchronization. In this section we first measure the temporal co-occurrence of audio-visual events between each camera pair and then we combine this information in a global step that verifies the consistency of the obtained values and the reliability of the recordings. Figure 6 shows the block diagram summarizing the main steps of the proposed approach.

6.1 Matching

To combine audio-visual activation vectors extracted from each camera and to estimate the time shifts across camera pairs C^i-C^j we use cross-correlation. The cross-correlation $\chi^{ij}(t')$ at time t' between the activation vectors of C^i and C^j is computed as

$$\chi^{ij}(t') = \sum_t f_{AV}^i(t) \cdot f_{AV}^j(t+t')\,, \tag{16}$$

where $f_{AV}^i(t)$ is the activation vector for camera C^i resampled to the highest sampling rate among all cameras.

Let $\mathbf{D} = \left[D^{ij}\right]$ be an $M \times M$ delay matrix composed of the estimated time shifts between each camera pair, where M is the number of cameras. We compute the inter-camera time shift as the argument that maximizes the cross-correlation through time:

$$D^{ij} = \underset{t'}{\operatorname{argmax}}\, \chi^{ij}(t')\,. \tag{17}$$

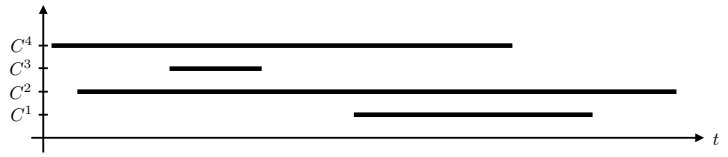

Fig. 7 Typical situation in which the proposed cross-validation procedure, unlike current state-of-the-art methods, helps synchronizing a set of recordings. Since recordings from cameras 1 and 3 do not overlap on the global timeline, the time shift among them cannot be computed directly and recordings 2 and 4 are required for their synchronization.

Ideally, $\mathbf{D}$ should be antisymmetric ($D^{ij} = -D^{ji}$) since the time shift between cameras C^i and C^j and the time shift between C^j and C^i are opposite numbers, i.e. if the recording of C^i is in advance with respect to that of C^j then the recording of C^j starts later. However, in practice this is not always the case and if $\chi^{ij}(t')$ presents two global maxima (with the same magnitude) the value chosen for D^{ij} might differ from that chosen for D^{ji}. Examples of this issue are shown in matrix $\mathbf{D}$ in Fig. 8, where $D^{12} = -15.60$ and $D^{21} = 4.27$. In this case, we capture at this stage the two different possible time shifts in $\mathbf{D}$ and leave the final decision to the next stage of our approach.

6.2 Global synchronization via cross-validation

Time shifts have been so far estimated using audio and video signals of each camera pair independently. However, not all results lead to the same alignment of the recordings on a global timeline because of errors in time-shift estimates between camera pairs. An illustrative example of the importance of using a global procedure to validate the partial results is depicted in Fig. 7. In this case, it is not possible to directly synchronize recordings corresponding to C^1 and C^3, because there is no temporal overlap between these recordings. Our goal is to validate the results globally to obtain *consistent* time shifts across cameras. To this end, we propose a cross-validation approach that locates all recordings on a common timeline by computing time shifts not only between the two cameras but also for intermediate cameras (e.g. C^2 and C^4 in Fig. 7). The time shift among a pair of recordings that do not overlap in time (and consequently cannot be computed directly) are obtained through the relative time shifts of each of these recordings with other recordings. Most previous state-of-the-art methods (Sec. II) were focused on a one-to-one synchronization (i.e. between camera pairs only) and did not provide a solution for the global synchronization problem of the entire set of M recordings.

Without loss of generality, we use the time when the recording from the first camera starts as reference time and we define the estimate time shifts as Δt^{1i}, $\forall i$. Negative time stamps represent recordings starting before that of C^1, while positive values indicate recordings that start after.

In a first stage, we generate the histograms of the obtained time shifts between camera C^1 and the remaining cameras C^i by taking into account the estimated offsets with intermediate cameras C^j. Then, we choose the consistent time shift Δt^{1i} between cameras C^1 and C^i to be the most frequent value in the histogram. The proposed cross-validation procedure is explained schematically in Fig. 8.

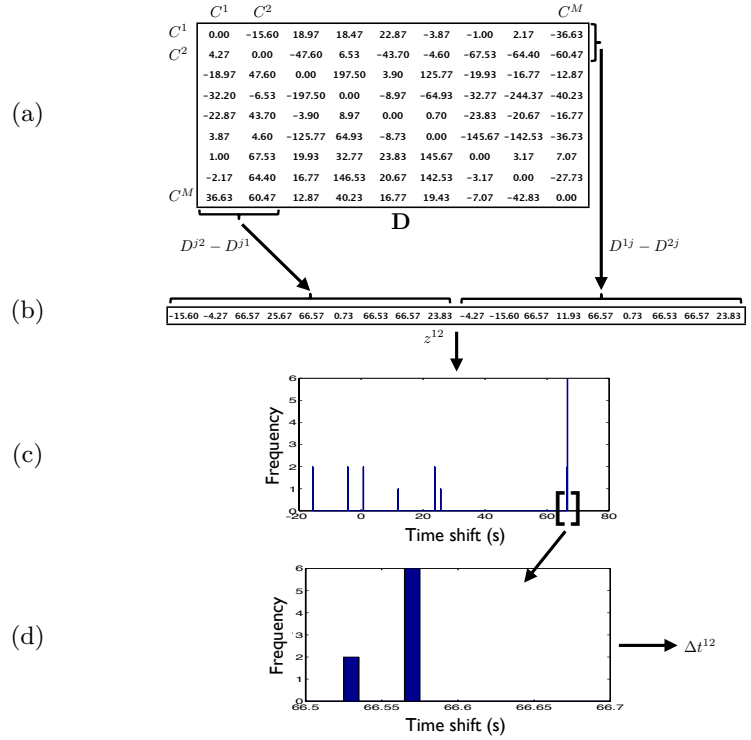

Fig. 8 Scheme illustrating the cross-validation procedure. First, a vector z^{12} (b) containing all possible time shifts between the recordings of cameras C^1 and C^2 is extracted from the delay matrix $\mathbf{D}$ in (a). Then, coarse synchronization (c) and fine synchronization (d), are performed sequentially to obtain a consistent estimate of the time shift between this camera pair Δt^{12}. The same procedure is applied for the estimation of the remaining time shifts Δt^{1i}, $\forall i$.

The vector z^{1i} that contains a list of possible time shifts between C^1 and C^i is built by taking into account time shifts with intermediate cameras C^j as

$$z^{1i} = \{(D^{ji} - D^{j1}) \cup (D^{1j} - D^{ij}) : j = 1, \ldots, M\}. \tag{18}$$

Then, the most frequent value in the vector z^{1i} is chosen as the globally consistent time shift $\widehat{\Delta t}^{1i}$. A coarse-to-fine strategy is employed to group similar time-shift values and to avoid the choice of less reliable values. Two consecutive steps based on histograms perform respectively a broad synchronization (with a 100 ms resolution) and a fine synchronization (10 ms resolution), and provide the final value for the time shift $\widehat{\Delta t}^{1i}$. The second step takes the maximum of the first histogram and computes another histogram at a finer resolution in a time window of 200 ms around this maximum.

Input $f^i_{AV}, f^i_A, \forall i : 1 \dots M$ (activation vectors)
Output $\Delta t^{1i}, \forall i : 1 \dots M$ (time shift between C^1 and C^i)

0. $m = AV$ (use Audio and Video jointly)
1. Compute matching and cross-validation with $f^i = f^i_m$. Obtain $\mathbf{D}_m$ (delay matrix) and $\widehat{\Delta t}^{1i}_m$ (time shifts).
2. Estimate Γ^i_m (confidence)
if Γ^i_{AV} is sufficiently large **then**
  $\Delta t^{1i} = \widehat{\Delta t}^{1i}_{(AV)}$
else
  Repeat steps 1 and 2 using Audio only (i.e. $m = A$)
  $\Delta t^{1i} = \widehat{\Delta t}^{1i}_{(A)}$ if $\Gamma^i_A > \Gamma^i_{AV}$
end

Algorithm 1: Verification of the alignment

6.3 Verification

Once the recordings are consistently sorted on a common timeline, we analyze the delay matrix $\mathbf{D}$ to estimate which cameras might be misleading. This procedure allows us to isolate cameras whose video/audio modalities are difficult to synchronize, and to automatically choose between the joint audio-visual method or an audio-only processing when the video modality is unreliable. This latter case happens with low-quality video or when a camera points at a different location compared to the other cameras. Algorithm 1 summarizes the overall multi-camera synchronization approach from the verification point of view. When the confidence on the audio-visual result Γ^i_{AV} is low, an audio-only processing is started in order to obtain more reliable time-shift estimations.

First, we compute the absolute difference $\mathbf{E} = |\mathbf{D} - \hat{\mathbf{D}}|$ between the delay matrix $\mathbf{D}$ that we obtain and a consistent delay matrix $\hat{\mathbf{D}}$ built according to the time shifts $\widehat{\Delta t}^{1i}$ as

$$\hat{D}^{ij} = \widehat{\Delta t}^{1j} - \widehat{\Delta t}^{1i}. \tag{19}$$

$\mathbf{E}$ captures inconsistencies between the estimates and the global result. When the inconsistency E^{ij} is large, the estimated time shift between C^i and C^j does not match the time shift obtained after cross-validation.

When a camera leads to errors in the estimates of several other cameras, we conclude that the recording obtained with this camera is misleading and therefore not helpful for the synchronization task. This results in a a measure of the reliability of the corresponding recording and associated time shift. The number of wrong estimates, $\xi^i \in [0, M-1]$, associated to camera C^i is computed as

$$\xi^i = \frac{1}{2}|\psi^i|_0 \,, \tag{20}$$

where ψ^i is the set of elements in the i-th row and i-th column of the inconsistency matrix $\mathbf{E}$ presenting a large difference with the global result:

$$\psi^i = \{j \in 1 \dots M : (E^{ij} > \tau) \cup (E^{ji} > \tau)\} \,. \tag{21}$$

Here τ defines the threshold of acceptability for a time-shift estimate to be considered consistent with the global synchronization result. As the values on the diagonal of $\mathbf{D}$, $\hat{\mathbf{D}}$ and $\mathbf{E}$ are zero, the maximum number of wrong estimates ξ^i is $M-1$.

Using this information, we define the *confidence on the audio-visual result*, $\Gamma^i_{AV} \in [0,1]$, for camera C^i as

$$\Gamma^i_{AV} = 1 - \frac{\xi^i}{M-1}. \tag{22}$$

This value indicates the confidence on the recording and corresponding result for camera C^i. When Γ^i_{AV} is high, the time shifts estimated directly between this camera and the other cameras in the set are coherent with the global result. In contrast, when Γ^i_{AV} is small, the recording is expected to be a low-quality one and therefore be misleading for the other cameras.

Small values of Γ^i_{AV} might be caused by low qualities of audio and/or video recordings, or by a mismatch between modalities. We still have an opportunity to improve the results when the problem is due to the video signal, by exploiting the audio-only events detected in Sec. 5.1. When

$$\Gamma^i_{AV} < \beta\,, \tag{23}$$

where β defines the minimum confidence allowed, we repeat the matching and cross-validation procedures (see Fig. 6) with the audio activation vectors $f^i_A(t^i)$, $\forall i = 1 \dots M$, instead of the audio-visual activation vectors. If the confidence on the audio-only result Γ^i_A is higher ($\Gamma^i_{AV} < \Gamma^i_A$) then the resulting time shift Δt^{1i} for C^i is the one computed by means of the audio-only analysis.

When most recordings are unreliable, most time-shift values estimated at the end of the cross-validation will be wrong. However, these recordings will have low-confidence values because the time-shift estimates for each camera pair, D^{ij}, will not be consistent across the delay matrix (typically, in such cases most values in the inconsistency matrix are high and therefore the confidence in the audio-visual estimate becomes small). For this reason, errors in the time-shift estimates after cross-validation do not affect the verification process and the estimated confidence is still valid.

6.4 Automatic detection of unrelated recordings

Until now, we have implicitly assumed that all the recordings are related, i.e. they overlap and can thus be synchronized. However in practice this is not always the case, since a certain amount of false positives (unrelated recordings) can be present. In this section we propose a simple criterion for the automatic detection of recordings that are not related to the considered scene, for sets with $M > 3$ cameras [2]. We classify the recording from C^i as *unrelated* if the following condition is fulfilled:

$$\Gamma^i_m \leq \frac{1}{M-1} \quad \forall m = A, AV\,. \tag{24}$$

[2] Note that with $M = 3$ cameras the proposed method can detect that there is an unrelated recording but not *which* recording is actually unrelated.

Table 2 Main characteristics of the experimental dataset.

Scene	Number of cameras	Clips duration	Resolution (pixels)	Video frame rate	Audio sampling rate	Moving cameras	Outdoors	Different distances	Different views
BasketballA	3	4-6 min	640 × 480	30 fps	44.1 kHz				
BasketballB	3	16 min	360 × 288	25 fps	44.1 kHz				
Dance	2	10 min	450 × 360	25 fps	44.1 kHz	✓			
Office	3	30-80 s	450 × 360	25 fps	44.1 kHz	✓			
ConcertA	9	20-368 s	many	13-30 fps	44.1 kHz	✓		✓	
ConcertB	10	1-7 min	many	25-30 fps	22, 44.1 kHz	✓	✓	✓	
Walk	2	30-38 s	492 × 360	25 fps	44.1 kHz	✓	✓		
UCD	8	1-15 min	many	18-30 fps	44.1 kHz	✓	✓	✓	✓
Dataset	40	4 hours	many	13-30fps	22, 44.1 kHz	✓	✓	✓	✓

Thus, a recording is classified as unrelated to the rest when its estimated time shift is not coherent with the ones of at least two other cameras, neither with the audio-visual nor with the audio-only analysis. By combining equations (22) and (24) we obtain that the recording of C^i is classified as unrelated if the number of wrong estimates is $\xi^i \geq M - 2$ for $m = A$ and $m = AV$. In practice, when a recording has a low confidence on the audio-visual result, the verification steps leads to an audio-only analysis, and then the recording can be detected as unrelated by checking both Γ^i_{AV} and Γ^i_A.

7 Results

7.1 Experimental setup

We consider a dataset composed of eight different scenes: *BasketballA*, *BasketballB*, *Dance*, *Office*, *ConcertA*, *ConcertB*, *Walk* and *UCD*. These scenes include recordings made with hand-held cameras and fixed cameras, are shot indoors and outdoors, and the views of the cameras overlap most of the time in some situations and just for a short period in others. The set of scenes and their characteristics are summarized in Table 2. The first two scenes correspond to basketball games: *BasketballA* depicts a college game and *BasketballB* shows a female basketball game [5]. *Dance*, *Office*, *ConcertA* and *Walk* are scenes taken from [18]. *ConcertB* and *UCD* are two additional scenes that depict a concert and an event in a university campus. The videos resolutions range from 202×360 to 640×360 pixels. In some cases the cameras are situated at very different distances from the target location (e.g. in a concert less than 10m, around 50m and more than 200m), which is defined as *different distances* in Table 2. Other scenes contain *different views*, i.e. the cameras point to different locations for most of the time (e.g. *UCD*). Fig. 9 shows the temporal distribution of the recordings on a common timeline. In some situations the temporal overlap among recordings is large (e.g. *BasketballB*), while

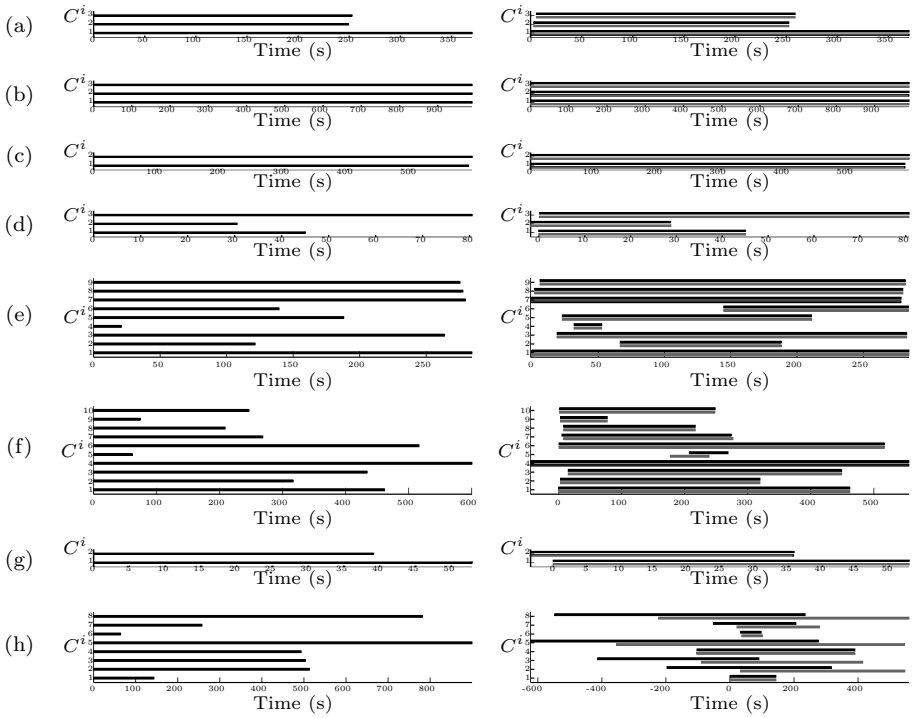

Fig. 9 Temporal distribution of the recordings before [left] and after [right] synchronization for BasketballA (a), BasketballB (b), Dance (c), Office (d), ConcertA (e), ConcertB (f), Walk (g), UCD (h). The plots on the right show the results obtained with our method *AV-V* (in black) compared to the groundtruth (in gray).

for other scenes the overlap is very short (e.g. *UCD*). The additional datasets used in the experiments are available online[3].

7.2 Methods under analysis

We compare seven methods (i) from the state of the art that can deal with independently moving cameras[4] as well as (ii) various combinations of elements of our proposed method. These methods are termed *AV-V*, *AV-X*, *A-X*, *A-F*, *A-O*, *A-R* and *V-F*. *AV-V* is the proposed method: audio-visual events matched by cross-correlation, *plus the verification* described in Sec. 6.3. *AV-X* corresponds to audio-visual events matched by cross-correlation (i.e. the results obtained with the method described up to the end of Sec. 6.2). *A-X* are the audio onsets obtained as described at the end of Sec. 5.1, matched by cross-correlation. *A-F* is the audio fingerprint approach presented in [18]. *A-O* is the audio-only method based on onset detection in 8 frequency bands [18]. *V-F* is a video-only approach that uses the flash detection [18] and matches the video activation vector using cross-correlation

[3] `http://www.eecs.qmul.ac.uk/ andrea/synchro.html`

[4] The constraints of the method in [13] make it not applicable to our dataset since it requires a minimum of 3 cameras, of which two need to be static.

Table 3 Comparative summary of the results. The cells show the number of recordings that are synchronized with an error that is smaller than T1=50 ms and T2=100 ms; the average error (in ms) for the synchronized recordings and the computation time (in seconds) for the whole dataset. Since the values of *A-R* are averaged over 100 realizations, its computation time is not depicted. M: number of cameras.

Scene	M	**Number of synchronized recordings**													
		AV-V		*AV-X*		*A-X*		*A-F*		*A-O*		*V-F*		*A-R*	
		T1	T2	T1	T2	T1	T2	T1	T2	T1	T2	T1	T2	T1	T2
BasketballA	3	3	3	3	3	3	3	3	3	3	3	0	0	2.9	2.9
BasketballB	3	2	2	2	2	2	2	2	2	2	2	0	0	2	2
Dance	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Office	3	3	3	3	3	2	3	0	3	2	3	0	0	1.9	3
ConcertA	9	9	9	9	9	8	8	6	8	9	9	0	0	7.9	7.9
ConcertB	10	7	7	6	6	7	7	7	7	7	7	2	2	6.2	6.3
Walk	2	0	2	0	2	0	2	0	0	0	2	0	0	0	1.9
UCD	8	2	2	2	2	2	2	3	6	3	3	0	0	0.3	0.3
Total synchronized		28	30	27	29	26	29	23	31	28	31	4	4	23.5	26.6
Average error (ms)		10	12	10	12	4	9	15	28	14	18	8	8	7	12
Computation Time (s)		26,372		26,370		978		198,718		467		8,368		-	

(instead of dynamical programming). A final method, *A-R*, keeps a random subset of audio onsets for each camera to build an activation vector that is then matched by cross-correlation. The amount of audio onsets that is kept is the same than with *AV-X*. Thus, *A-R* allows the comparison of the results when using video to select robust onsets with a random approach, and to determine the reliability of video selection. Please notice that for a fair comparison the consistency of the estimated time shifts across cameras is ensured for all methods by using the proposed cross validation strategy described in Sec. 6.2. Thus, results might be improved compared to estimating the time shift among each camera pair independently, as done in some of the prior work.

The parameters are the same for all the experiments. We use $L = 400$ blocks in which the video signal around an audio event is divided, corresponding to 20 divisions in the horizontal axis and 20 divisions in the vertical axis. The temporal tolerance between events in audio and video channels is $T_{AV} = 100$ ms according to the thresholds of detectability in [17]. To assess the video blocks activation we use a window W of 1 second duration around the audio onset. $\gamma = 0.3$ and thus an activation in more than 30% of active blocks is considered to be caused by a global camera motion. Finally, we use $\tau = 0.1$ seconds and $\beta = 0.3$ for the verification.

7.3 Discussion

A visualization of the final alignment result is shown in Fig. 9. Results obtained using the proposed approach *AV-V* are close to the groundtruth in all scenes except for *UCD*. In this scene our assumption does not hold because cameras record different views most of the time. Figure 10 shows sample results of the audio-visual event detection part of the algorithm, with the visualization of the blocks that have been considered synchronous with an audio event.

Quantitative results comparing the accuracy of the different synchronization approaches are shown in Table 3. The cells show the number of synchronized recordings when allowing 50 and 100 ms as temporal tolerances between the estimated time shifts and the ground-truth values. The number of synchronized

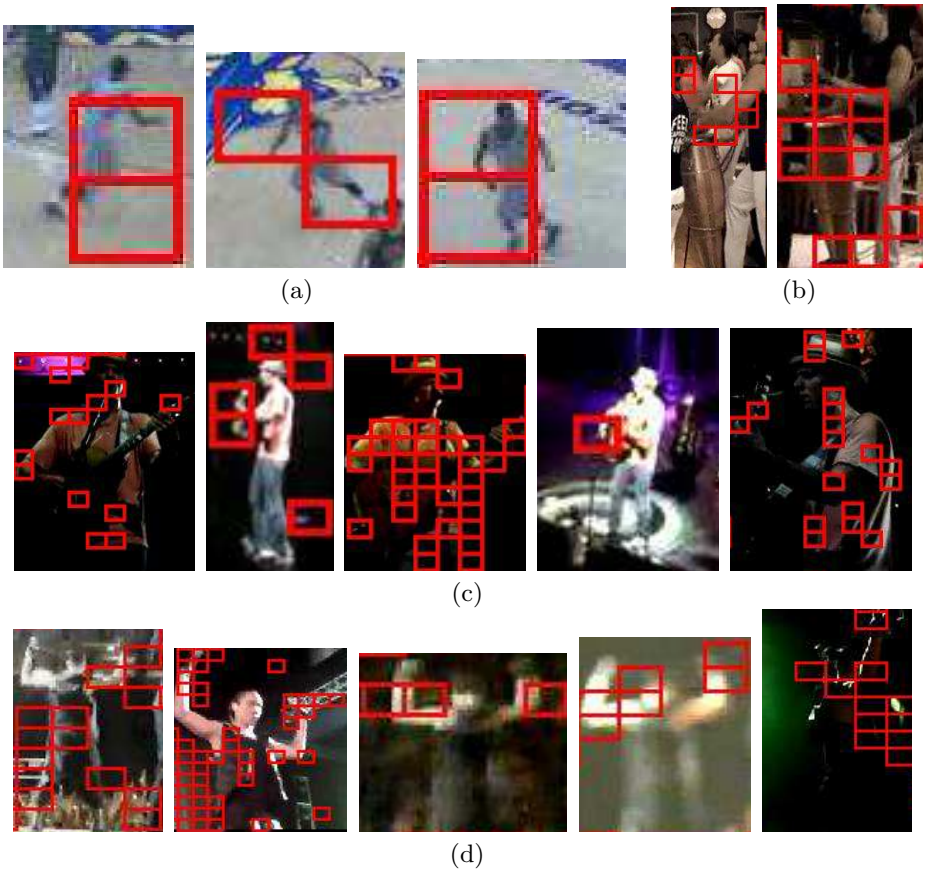

Fig. 10 Examples of active video blocks for audio-visual events detected with our approach in multiple cameras. The depicted scenes correspond to (a) a player movement in *BasketballA*, (b) a hit in the drum in *Dance*, (c) a guitarist hand in *ConcertA* and (d) a dance move in *ConcertB*.

recordings using the audio-visual analysis (*AV-X*) is larger than that of *A-R*. *A-X* fails in synchronizing C^4 in *ConcertA*, which is the shortest recording in this scene. In contrast, *AV-X* can cope with this recording but fails in synchronizing C^2 in *ConcertB* (apart from C^5, C^7 and C^{10} in which all approaches fail). In fact, in C^2 of *ConcertB* audio and video are not synchronized and the recording starts with a still image for 2.5s showing the band name while the soundtrack is already playing. This recording is a mix between unrelated audio and video signals that our verification method is able to detect (the confidence on the audio-visual result is small). In all situations *AV-V* can synchronize more recordings than *AV-X* and *A-X*, because it can get the best features from each modality. For each scene the value for *AV-V* is the maximum between the synchronized recordings with *AV-X* and those synchronized with *A-X*. This demonstrates the efficiency of the verification step in our algorithm. From all scenes, *V-F* is only able to synchronize the two recordings in *Dance*, as well as C^1 and C^3 in *ConcertB*. In the first case flashes are very visible and in the second case the cameras are close to the scene and variations in their global brightness are comparable. The average overall error

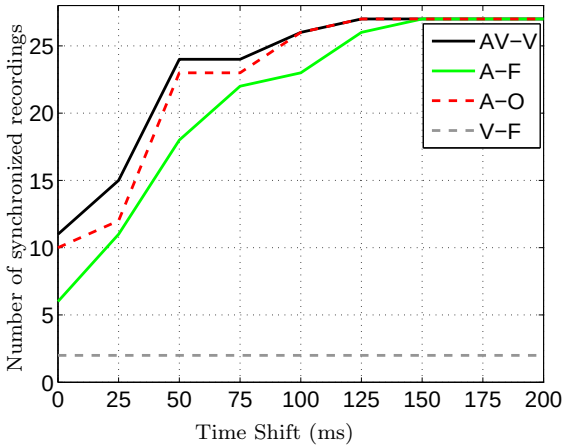

Fig. 11 Number of successfully synchronized recordings with an error in the estimated time shift smaller or equal to the value in the x coordinate (in ms) for the recordings with a high confidence on the result ($\Gamma^i_{AV} \geq \beta$). The faster the curves reach a high value in the plot, the better the performance.

in the synchronized recordings is lower when using our method (AV-V) than with previous approaches (A-F, A-O, V-F). Finally, AV-V synchronizes more recordings than the other methods with an error in the time-shift estimation lower than 100 ms (Fig. 11) when the confidence on the audio-visual result is high.

The computation time that is required by MATLAB implementations of the different methods to analyze the totality of our dataset is shown in Table 3. The results are obtained in a cluster composed of 32 processors Intel Xeon CPU E7- 8837 @ 2.67 GHz and 530GB memory. Please, notice that the methods have not been optimized or parallelized. The proposed method AV-V requires considerably less time than the fingerprints approach A-F but longer than the audio-only method A-O. The time complexity of the current implementation of our method depends linearly (i) on the number of cameras M, (ii) on the number of audio onsets and (iii) on the image resolution. As expected, the video analysis is the most demanding part in our approach (AV-X takes much longer than A-X), and more than 50% of the time required by AV-V is used to read the parts of the video signal around the audio onsets. The performance of AV-V could be easily improved by parallelizing its execution (the audio-visual onsets extraction can be performed independently for each camera and the matching step is computed pairwise) and optimizing the video processing pipeline.

To evaluate the resilience of the proposed method to unrelated recordings (Sec. 6.4), we used the scenes in Table 2. For each scene a total of $M + 1$ recordings have been considered, where the additional recording corresponds to the *first recording in the following scene*, that is an unrelated recording. The composition of this modified dataset together with the results obtained with our method are shown in Table 4. A true positive TP is a recording that is correctly classified as related to the rest; a true negative TN is a recording that is correctly classified as unrelated to the rest; false positives FP and false negatives FN are misclassified as related or unrelated, respectively. The criterion used to classify a recording as unrelated (or non-overlapping) is described in Sec. 6.4. According to the proposed

Table 4 Resilience to unrelated recordings on the modified experimental dataset. TP, TN, FP and FN denote, respectively, the number of true positives, true negatives, false positives and false negatives obtained with our method. M: number of cameras.

Scene	M	**Recordings**	**TP**	**TN**	**FP**	**FN**
BasketballA	4	3 BasketballA + 1 BasketballB	3	1	0	0
BasketballB	4	3 BasketballA + 1 Dance	3	1	0	0
Dance	3	2 Dance + 1 Office	2	0	1	0
Office	4	3 Office + 1 ConcertA	3	1	0	0
ConcertA	10	9 ConcertA + 1 ConcertB	9	1	0	0
ConcertB	11	10 ConcertB + 1 Walk	8	1	0	2
Walk	3	2 Walk + 1 UCD	2	0	1	0
UCD	9	8 UCD + 1 BasketballA	3	1	0	5
Dataset	48	40 related + 8 unrelated	33	6	2	7

classification criterion, we obtain a 94% precision and a 83% recall. The FN result is mainly driven by the difficulty of the *UCD* scene: a total of 7 recordings are misclassified as unrelated to their corresponding scenes since they cannot be reliably synchronized. There are only 2 false positives, that happen when the number of cameras $M = 3$ and our criterion cannot be applied since the unrelated recording cannot be distinguished.

8 Conclusions

We proposed a method for the synchronization of multi-camera recordings which is based on the joint analysis of audio and video signals. Audio-visual events are extracted from each recording and then matched using cross correlation. Two novel steps based on cross validation and verification are used to ensure the consistency of the results globally and to test the reliability of the estimated time shifts. When the confidence in the joint audio-visual analysis is low due to poor video quality or very different views, an audio-only strategy is automatically adopted. Our method is generally applicable and can deal with sets of recordings that do not necessarily share the same time interval. The proposed method was compared with alternative approaches over a heterogeneous dataset containing recordings from handheld cameras at a wide range of distances from the target location being filmed. As the current formulation is aimed at compensating for time shifts, in our future work we will incorporate time drifts in order to address its effect on the alignment of longer recordings. An approach in this direction is presented in [8], but it requires human interaction.

References

1. Adobe Premiere Pro: `http://www.adobe.com/products/premiere.html`, Last accessed: 26 August 2013
2. Caspi, Y., Irani, M.: Spatio-temporal alignment of sequences. IEEE Trans. Pattern Analysis and Machine Intelligence **24**, 1409–1424 (2002)
3. Cremer, M., Cook, R.: Machine-assisted editing of user-generated content. In: Proc. SPIE-IS&T Electronic Imaging, vol. 7254 (2009)

4. Daniyal, F., Taj, M., Cavallaro, A.: Content and task-based view selection from multiple video streams. Multimedia Tools Appl. **46**, 235–258 (2010)
5. EU, FP7 project APIDIS (ICT-216023): `http://www.apidis.org/Dataset/`, Last accessed: 26 August 2013
6. Final Cut Pro: `http://www.apple.com/finalcutpro/`, Last accessed: 26 August 2013
7. Fritsch, J., Kleinehagenbrock, M., Lang, S., Fink, G.A., Sagerer, G.: Audiovisual person tracking with a mobile robot. In: Int. Conf. Intelligent Autonomous Systems (2004)
8. Guggenberger, M., Lux, M., Boszormenyi, L.: Audioalign - Synchronization of A/V-streams based on audio data. International Symposium on Multimedia **0**, 382–383 (2012)
9. Jiang, W., Cotton, C., Chang, S.F., Ellis, D., Loui, A.C.: Audio-visual atoms for generic video concept classification. ACM Trans. on Multimedia Computing, Communications, and Applications **6**(3), 1–19 (2010)
10. Kennedy, L.S., Naaman, M.: Less talk, more rock: automated organization of community-contributed collections of concert videos. In: Proc. ACM WWW (2009)
11. Kidron, E., Schechner, Y.Y., Elad, M.: Cross-modal localization via sparsity. IEEE Trans. Signal Processing **55**(4), 1390–1404 (2007)
12. Laptev, I., Belongie, S.J., Prez, P., Wills, J.: Periodic motion detection and segmentation via approximate sequence alignment. In: ICCV (2005)
13. Lei, C., Yang, Y.H.: Tri-focal tensor-based multiple video synchronization with subframe optimization. IEEE Trans. Image Processing **15**(9), 2473–2480 (2006)
14. Llagostera Casanovas, A., Monaci, G., Vandergheynst, P., Gribonval, R.: Blind Audio-Visual Source Separation based on Sparse Redundant Representations. IEEE Trans. Multimedia **12**(5), 358–371 (2010)
15. Padua, F.L., Carceroni, R.L., Santos, G.A., Kutulakos, K.N.: Linear sequence-to-sequence alignment. IEEE Trans. Pattern Analysis and Machine Intelligence **32**, 304–320 (2010)
16. Potamianos, G., Neti, C., Gravier, G., Garg, A., Senior, A.W.: Recent advances in the automatic recognition of audiovisual speech. Proc. IEEE **91**, 1306–1326 (2003)
17. RECOMMENDATION ITU-R BT.1359-1: Relative Timing of Sound and Vision for Broadcasting (1998)
18. Shrestha, P., Barbieri, M., Weda, H., Sekulovski, D.: Synchronization of multiple camera videos using audio-visual features. IEEE Trans. Multimedia **12**, 79–92 (2010)
19. Sodoyer, D., Girin, L., Jutten, C., Schwartz, J.L.: Developing an audio-visual speech source separation algorithm. Speech Commununication **44**(1-4), 113–125 (2004)
20. Stein, G.: Tracking from multiple view points: Self-calibration of space and time. In: CVPR (1999). DOI 10.1109/CVPR.1999.786987
21. Sumby, W.H., Pollack, I.: Visual contribution to speech intelligibility in noise. Journal of the Acoustical Society of America **26**(2), 212–215 (1954)
22. Summerfield, Q.: Some preliminaries to a comprehensive account of audio-visual speech perception. In: Hearing by Eye: The Psychology of Lipreading, pp. 3–51. Lawrence Erlbaum Associates (1987)

23. Ukrainitz, Y., Irani, M.: Aligning sequences and actions by maximizing space-time correlations. In: ECCV (2006)
24. Vroomen, J., Keetels, M.: Perception of intersensory synchrony: A tutorial review. Attention, Perception & Psychophysics **72**(4), 871–884 (2010)
25. Wedge, D., Huynh, D., Kovesi, P.: Using space-time interest points for video sequence synchronization. In: Proc. IAPR Conf. Machine Vision Applications (2007)
26. Whitehead, A., Laganiere, R., Bose, P.: Temporal synchronization of video sequences in theory and in practice. In: Proc. IEEE Workshop Motion and Video Computing (2005)
27. Yan, J., Pollefeys, M.: Video synchronization via space-time interest point distribution. In: Proc. Advanced Concepts for Intelligent Vision Systems (2004)