Multimed Tools Appl (2016) 75:7187–7210
DOI 10.1007/s11042-015-2641-2

ViComp: composition of user-generated videos

Sophia Bano[1] **· Andrea Cavallaro**[1]

Received: 30 September 2014 / Revised: 28 February 2015 / Accepted: 17 April 2015 /
Published online: 9 June 2015

Abstract We propose *ViComp*, an automatic audio-visual camera selection framework for composing uninterrupted recordings from multiple user-generated videos (UGVs) of the same event. We design an automatic audio-based cut-point selection method to segment the UGV. *ViComp* combines segments of UGVs using a rank-based camera selection strategy by considering audio-visual quality and camera selection history. We analyze the audio to maintain audio continuity. To filter video segments which contain visual degradations, we perform spatial and spatio-temporal quality assessment. We validate the proposed framework with subjective tests and compare it with state-of-the-art methods.

Keywords Video composition · User-generated videos · Audio-visual analysis · Camera selection · Subjective evaluation

1 Introduction

With the increasing availability of multimedia portable devices, more and more people record events (such as a concert, sports game, public speech and rallies) from different angles that are then shared on the Internet. These User-Generated Videos (UGVs) have limited fields of view, incomplete temporal coverage of the event and may contain visual degradations (e.g. unwanted camera movements). Moreover, the audio in each UGV is

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). The authors acknowledge the support of the UK Engineering and Physical Science Research Council (EPSRC), under grant EP/K007491/1.

✉ Sophia Bano
s.bano@qmul.ac.uk

Andrea Cavallaro
a.cavallaro@qmul.ac.uk

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

of varying quality due to different recording devices, and may contain reverberations and sounds from the surrounding environment.

Video editing is performed by discarding low-quality and less interesting visual segments. The edited video is composed of shots, where a shot is an uninterrupted recording. Each visual segment is a sub-shot which can be characterized by its content. The perceived audio-visual quality is a key factor which makes the content enjoyable and interesting to playback [23]. Therefore, audio continuity and uniformity are also desired along with the appropriate view selection. In UGVs, global feature analysis is performed for understanding the content by attention detection [15], and for filtering the low-quality content by camera motion analysis [7, 15]. Existing video composition methods [30, 35] for UGVs perform visual quality analysis and manual cut-point selection. Although audio content plays an important part in the judgment of the overall perceived quality [5], it has not been analyzed in the existing methods [30, 35].

In this paper, we propose a framework (*ViComp*) for the automatic multi-camera composition from UGVs recorded from different viewpoints of an event. To maintain audio uniformity, we propose a method for audio stitching by ranking a set of audio signals from an event based on their quality. We design an automatic cut-point selection method by analyzing the change in the dynamics of three audio features, namely root mean square, spectral centroid and spectral entropy. The selected cut-points are used for video segmentation. To suppress low-quality video segments and extract the ones with high quality, we perform spatial and spatio-temporal analyses. To enhance the viewing experience [43], we rank and select segments using visual quality and view diversity by considering the selection history of the past two video segments. The block diagram of the proposed framework is shown in Fig. 1.

This paper is organized as follows. In Section 2, we present the related work. In Section 3, we define and formulate the problem. The audio and video analyses are described in Sections 4 and 5, respectively, followed by the camera-selection description in Section 6. The comparison of the proposed method with the state-of-the-art methods is presented in Section 7. The experimental results and subjective comparison with the existing methods are detailed in Section 8. Finally, Section 9 concludes the paper.

2 Related work

The state of the art for multi-camera recording editing and camera selection can be grouped based on the scenario for which they are designed. This includes camera selection for lecture webcast [9, 41] and meetings [27, 42], sports video broadcast [8, 11, 38], home-video summarization [7, 15], and multi-camera mashup generation [30, 35]. In video editing for summarization, the continuity of the event is disregarded and only key frames are included in the output video. In video composition, a time continuous video is generated by selecting video segments from multiple cameras.

Camera selection in lectures [9, 41] focuses on the lecturer, slides, or audience. Frame differencing in fixed cameras [9], or online detection and tracking in PTZ cameras [41] is performed for lecturer localization. Dickson et al. [9] developed a lecture recording system that involved two fixed cameras; one capturing the lecturer and the whiteboard, and the other recording the projector screen. The lecturer is localized, cropped (by applying frame differencing), and used in the final GUI presentation along with whiteboard and slides. Winkler

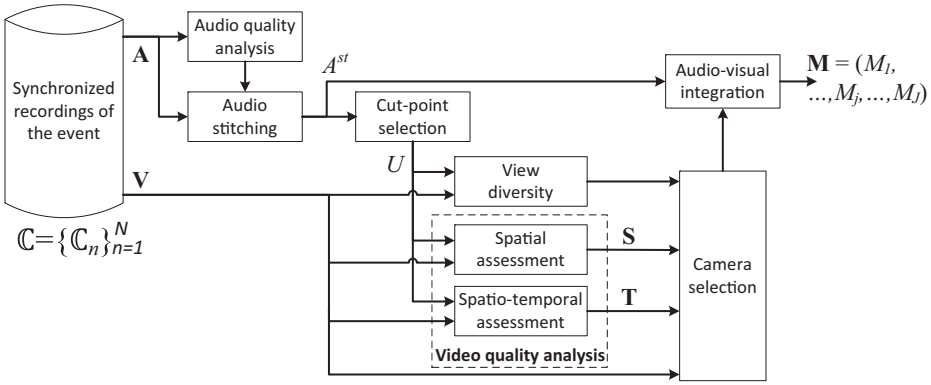

Fig. 1 Block diagram of the proposed multi-camera UGV composition (*ViComp*) framework. The audio signals are analyzed for audio stitching, followed by suitable cut-points selection. The videos are analyzed for quality and view diversity to contribute to the rank-based camera-selection method

et al. [41] presented an automatic control strategy for PTZ cameras for online tracking of the lecturer. The method used depth stream from a KINECT sensor for head detection and translated this position into the PTZ-camera coordinate system for its positioning. Similarly, in meeting room video editing [27, 42] mainly speaker identification, localization, recognition and tracking are performed to select different camera views. Yu and Nakamura [42] presented a detailed survey on smart meeting systems and their architecture along with the involved components for the recognition of audio and visual events in a meeting. For improving video conferencing, Ranjan et al. [27] presented an automatic camera control strategy, that performed speaker localization, head detection and tracking, and close up or pair of speakers spot detection. In these systems, video recordings are generally captured from high-quality cameras having a constraint environment in a lecture or meeting room with a presetting of fixed or stable moving cameras, and adequate lighting conditions, thus providing favorable conditions for speaker localization and recognition. Though linked with camera selection, these methods are not directly applicable for multi-camera selection in UGVs in which the visual quality varies from one camera to another.

For the automatic boardcast of sports videos, Wang et al. [38] computed features like field line, goalmouth, centre circle, ball trajectory, camera motion and audio keywords, and used them for event detection by training a Support Vector Machine (SVM) for three event classes, namely attack, foul and miscellaneous. The method [38] selected the main camera most of the time and the sub-cameras were selected by maximizing the likelihood score of suitable/unsuitable sub-camera segments, classified by applying a Hidden Markov Model (HMM) on camera motion features. Daniyal et al. [8] used the amount of activity (scored using background subtraction), objects' trajectory, size and location as features for content and task-based camera selection, and demonstrated their results on a basketball game, airport surveillance and outdoor videos. To avoid too frequent camera switching, this method [8] modeled the prior selection state using Dynamic Bayesian Networks. D'Orazio and Leo [11] presented a detailed review of vision-based systems for soccer game video analysis, in which they detailed that the type of features extracted for vision-based sports game analysis vary based on the application under study and may include dominant color,

color histogram, camera motion, corner points, ball and player detection, field characteristics detection, texture, and player recognition [11]. Multi-camera sports videos are recorded using professional cameras capable of performing stable pan, tilt and zoom motions [11]. Extracting particular features representing object specific properties might not be applicable on UGVs of concerts in case of poor visual-quality and lighting conditions.

For home video-editing, Hua et al. [15, 16] performed sub-shot, attention and sentence detection. The method [15, 16] filtered out low quality sub-shots, and aligned the sub-shot boundary with the selected music tempo, while preserving the detected sentence portions of the video. Campanella et al. [7] used brightness, contrast, shake, and face detection for obtaining a suitability score for home video sub-shots, where sub-shots are obtained by filtering frames containing unwanted camera motion. The method [7] performed editing by selecting the video segments with the highest score and by allowing the user to perform the editing while watching. Mei et al. [23] analyzed spatio-temporal factors (unstable, jerky, low fidelity, brightness, blur and orientation) and proposed three quality metrics for home-video summarization, namely user study-based (weighted average of all spatio-temporal factors), rule-based (nonlinear fusion [16]), and learning-based (offline two-class quality training). A skim ratio (for the length of the final video) is defined and the sub-shots with maximized quality metric are selected to compose the video. Home video editing methods are closely linked with the generation process of multi-camera recordings because of the similar content type and analysis. Their input information differs, as in the case of multi-camera recording generation, multiple views of an event are available for the composition of a continuous video.

Areu et al. [3] reconstructed the 3D structure of the scene [36] from multiple UGVs, and estimated cameras' positions and orientations to compute their 3D joint attention. The 3D motion of a camera is used to estimate its stabilization cost. The stabilization, camera roll and joint attention cost are then used as features for camera-view selection.

Mashup generation systems from UGVs have been proposed by Shrestha et al. [35] (FirstFit) and Saini et al. [30] (MoViMash). FirstFit [35] analyzed video quality features such as blockiness, blur, brightness and shake to perform camera selection, while MoViMash [30] additionally used occlusion and tilt, and introduced an offline learning stage which incorporated video editing rules, such as shooting angle, shooting distance and shot length. Although these methods are claimed to be automatic [30, 35], they rely on manual segmentation of video clips. Further, MoViMash manually categorized the videos into right, left, center, near and far for learning the shot-transition distributions. Beerends and Caluwe [5] conducted subjective experiments by varying audio and video quality of videos to test their influence on the perceived video/audio quality. Their findings showed that low-quality audio decreases the perceived quality of the video as well. In FirstFit [35], the audio is selected from the same media segment which contributed to camera selection, thus resulting in audio with varying quality when playing back the generated video. For MoViMash [30], the audio is not aligned with the video within the resulting mashups. Wilk and Effelsberg [40] studied the influence of visual degradations on the perceived quality of UGVs. In particular, they studied the effect of camera shake, harmful occlusion and camera misalignment, and rated video clips of 9-12s duration on a 5-point scale corresponding to different levels of degradation. Their results showed that these degradations, in particular camera shake, highly affect the perceived quality of UGVs.

Several approaches exist in the literature for audio classification via segmentation [22], e.g. to classify silence, music, environment sound and speech. Some approaches used tempo analysis [31] while others use Mel Frequency Cepstral Coefficients (MFCCs) to perform self-similarity decomposition in order to obtain audio segmentation. Most of these methods

required training to identify the different classes of audio and to perform the segmentation based on structural information. The tempo detection approach [31] is designed using onsets for finding suitable cut-points. This gave unsatisfactory results due to the sensitivity of onsets in the presence of audio degradations [35].

Table 1 summarizes the state-of-the-art methods for multi-camera editing and composition with respect to the scenario for which they are designed.

3 Problem formulation

Let $\mathbb{C} = \{\mathbb{C}_n\}_{n=1}^{N}$ denote N synchronized and continuous multi-camera UGVs of an event. Each $\mathbb{C}_n$ is at least partially overlapping in time with some other UGVs in $\mathbb{C}$. Let $\mathbf{V} = \{V_n\}_{n=1}^{N}$ and $\mathbf{A} = \{A_n\}_{n=1}^{N}$ denote N visual and audio signals in $\mathbb{C}$, respectively. Each $V_n = (v_{n1}, ..., v_{nk}, ..., v_{nK_n})$ is re-sampled to a common frame rate[1], K, and contains K_n frames. Likewise, each $A_n = (a_{n1}, ..., a_{np}, ..., a_{nP_n})$ is re-sampled to a common sampling rate, P, and contains P_n audio samples. Each $\mathbb{C}_n$ is temporally ordered with respect to others on a common timeline, such that the first video frame corresponds to the first recorded frame in $\mathbb{C}$ and the last video frame, I_v, corresponds to the last video frame in $\mathbb{C}$. Likewise for the audio $\mathbf{A}$, which goes from 1 to I_a.

Let $A^{st} = (a_1^{st}, \cdots, a_i^{st}, \cdots a_{I_a}^{st})$ denote the stitched audio of the event and $U = (u_1, \cdots, u_j, \cdots, u_J)$ denote the suitable cut-points, where J is the number of segments. Let $\mathbf{S} = \{S_n\}_{n=1}^{N}$ and $\mathbf{T} = \{T_n\}_{n=1}^{N}$ denote the spatial and spatio-temporal scores for each $\mathbb{C}_n$, respectively. The problem of automatic video composition can be described as selecting J segments from the set of N UGVs to generate a single coherent video $\mathbf{M} = (M_1, ...M_j, ..., M_J)$, where M_j represents the j^{th} video segment.

4 Audio analysis

We propose an audio stitching method that produces consistent and uniform audio ,A^{st}, for the complete duration of the event. We also propose a cut-point selection method, which aims at finding the binary signal, A_U^{st}, for the suitable cut-points, where U is in s (*seconds*) to be used as a common reference point for both audio and video segmentation. We obtain $U = (u_1, \cdots, u_j, \cdots, u_J)$ by analyzing three audio features, namely root mean square, $\mathbf{A}^{RMS}$, spectral entropy, $\mathbf{A}^{SE}$, and spectral centroid, $\mathbf{A}^{SC}$.

4.1 Audio-quality analysis for audio ranking

The overlapping audio signals, $\mathbf{A}$, for an event contain audio captured from different devices and locations. Hence, the quality of audio varies from one video to another. For audio stitching, we need to know which audio is better in $\mathbf{A}$. In order to achieve this, we analyze the spectral rolloff [21] of the set of audio signals, $\mathbf{A}$, to rank the individual A_n based on their quality.

Spectral rolloff estimates the amount of high frequency [21] in the signal by calculating the frequency which contains 85 % of the signal energy. Real-world degradations present in UGVs introduce high frequencies in the audio signal, thus resulting in 85 % of the signal

[1] All UGVs are converted to the same frame rate using VirtualDub [20].

Table 1 State of the art for multi-camera editing and composition

Ref.	Type	AQ	AC	VQ	CM	VD	Features extracted	Camera selection method	Data type	No. E	Comments
[15, 16]	ED			✓	MVF		Entropy, motion intensity, attention sentence detection	Sub-shot boundary & alignment with onset & sentence	Home videos		Beat detection in incidental music
[7]	ED	–	–	✓	LPC		Brightness, contrast, shake, face detection	Highest suitability score and edit while watching	Home videos		Video segmentation by removing shaky frames
[23]	ED	–	–	✓	AMM		Stable, jerk, infidelity, brightness, blur, orientation	Maximisation of quality metric	Home videos		User study, rule & learning-based quality metrics
[38]	CP				MVF		Field line, goalmouth, centre circle, ball trajectory	Likelihood score maximisation for sub-shots	Soccer videos		Event detection using extracted features
[8]	CP	–	–				Object detection, tracking, size estimation	DBN for camera selection	Basketball, Surveillance		Event detection using extracted features
[3]	CP				3D CM		Stability, camera roll, 3D joint attention	Optimisation of feature cost in Trellis graph	UGVs	10	3D reconstruction of the scene
[35]	CP			✓	LPC	✓	Blockiness, blur, brightness, shake	Optimisation of the weighted sum of scores	UGVs	3	Manual segmentation for cut-point selection
[30]	CP			✓	LPC	✓	Blockiness, blur, contrast, brightness, occlusion, tilt	Optimisation of the weighted sum of scores	UGVs	3	Manual cut-point & camera view selection
ViComp	CP	✓	✓	✓	LPC	✓	BRISQUE, shake	Rank-based camera-selection	UGVs	13	Automatic cut-point selection

Key: AQ - Audio Quality; AC - Audio Continuity; VQ - Visual Quality; CM - Camera Motion Analysis; VD - View Diversity; No. E - Number of events tested; ED - Editing; CP - Composition; MVF - Motion Vector Field; LPC - Luminance Projection Correlation; AMM - Affine Motion Model; DBN - Dynamic Bayesian Network; BRISQUE - Blind/Referenceless Image Spatial Quality Evaluator; '-' - not used

energy shifting to a higher frequency value. Therefore, for designing the ranking strategy, we assume that the overlapping audio signals with low-spectral rolloff contain less noise than the spectrum in the audio signals with high-spectral rolloff values. This is illustrated with the help of an example in Fig. 2. The spectrum in Fig. 2b is more concentrated towards low frequency bins and contains less noise as compared to Fig. 2a.

For ranking the audio signals, we decompose each A_n for the overlap duration $[I_{a'}, I_{a''}]$ into non-overlapping frames $1, \cdots, \gamma, \cdots, \Gamma$ using frame size $f_{r1} = 1s$ (selected empirically). We varied the frame size f_{r1} from $0.5s$ to $3.0s$ with a step size of $0.5s$ to calculate the ranks (using the below mentioned method), and found $1s$ to be the most appropriate as the ranks become consistent at and beyond this frame size. We calculate the Fourier transform within each frame γ

$$[X_{n1}(l), \cdots, X_{n\gamma}(l), \cdots, X_{n\Gamma}(l)], \tag{1}$$

where l is the frequency bin. We then compute the spectral rolloff, $\mathbf{A}_n^{SR}$, for the set of audio signals, $\mathbf{A}$, which is given by

$$\mathbf{A}^{SR} = \left[\mathbf{A}^{SR}(1), \cdots, \mathbf{A}^{SR}(\gamma), \cdots, \mathbf{A}^{SR}(\Gamma)\right], \tag{2}$$

where,

$$\mathbf{A}^{SR}(\gamma) = \left[A_1^{SR}(\gamma), \cdots A_n^{SR}(\gamma), \cdots, A_N^{SR}(\gamma)\right]^T. \tag{3}$$

This is followed by computing the rank matrix, $\mathbf{R}^{SR}$, within each frame by sorting each $\mathbf{A}^{SR}(\gamma)$ in ascending order and obtaining its argument. The audio signal which appears the most in each row of $\mathbf{R}^{SR}$ is selected as the one with the best quality, followed by the others. This gives the rank vector $R^{SR} = [r(1), \cdots, r(n), \cdots, r(N)]^T$, where R^{SR} contains the indices of A_n in descending order of quality.

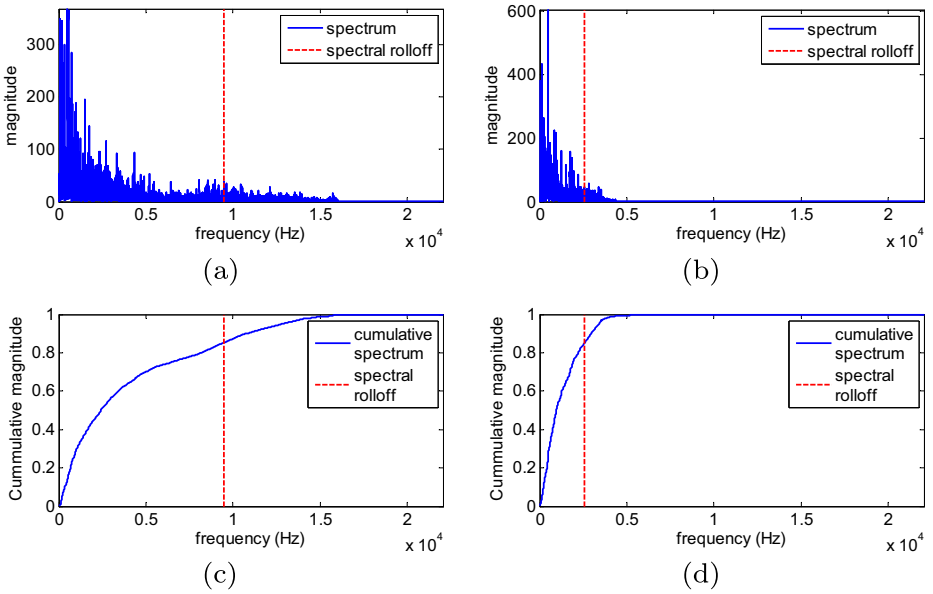

Fig. 2 Spectral rolloff analysis for audio ranking. (a) and (b) show the spectrum of a synchronized audio frame from two audio signals, and (c) and (d) show their respective cumulative spectra. The spectral rolloff is shown in red. (b) contains less noise as compared to (a) since (b) is more concentrated towards low frequency bins

4.2 Audio-stitching using the rank vector

To obtain a continuous audio track starting from the earliest starting video till the last ending one, we perform audio stitching A^{st} using the rank vector R^{SR}. From the synchronization, we have the relative start, G_n, and end, E_n, timings of each A_n. Our audio-stitching method works as follows. At level 1 of the stitching, $A_{r(1)}$ is selected to span for the duration $G_{r(1)}$ to $E_{r(1)}$, thus resulting in intermediate stitched audio $\dot{A}^{st} = (a_{G_{r(1)}}, \cdots, a_{E_{r(1)}})$. At level 2, in order to reduce the number of stitched points, we compromise between the quality and the number of stitch points. Therefore, we update $\dot{A}^{st}$ by checking if $A_{r(1)}$ is completely, before or after contained within $A_{r(2)}$ (see Algo. 1). In a situation where $A_{r(2)}$ is completely contained within $A_{r(1)}$, we do not update $\dot{A}^{st}$. The process continues until we obtain the stitched audio $A^{st} = \left(a_1^{st}, \cdots, a_i^{st}, \cdots a_{I_a}^{st}\right)$ for the complete duration. This process of audio ranking and stitching is illustrated in Fig. 3.

Algorithm 1 Audio stitching algorithm at level 2.

```
if G_{r(2)} < G_{r(1)} & E_{r(2)} > E_{r(1)} then
    Ȧ^st = (a_{G_{r(2)}}, ··· , a_{E_{r(2)}})
else if G_{r(2)} < G_{r(1)} & E_{r(2)} < E_{r(1)} then
    Ȧ^st = (a_{G_{r(2)}}, ··· , a_{G_{r(1)}}, ··· , a_{E_{r(1)}})
else if G_{r(2)} > G_{r(1)} & E_{r(2)} > E_{r(1)} then
    Ȧ^st = (a_{G_{r(1)}}, ··· , a_{E_{r(1)}}, ··· , a_{E_{r(2)}})
else
    Ȧ^st = (a_{G_{r(1)}}, ··· , a_{E_{r(1)}})
end
```

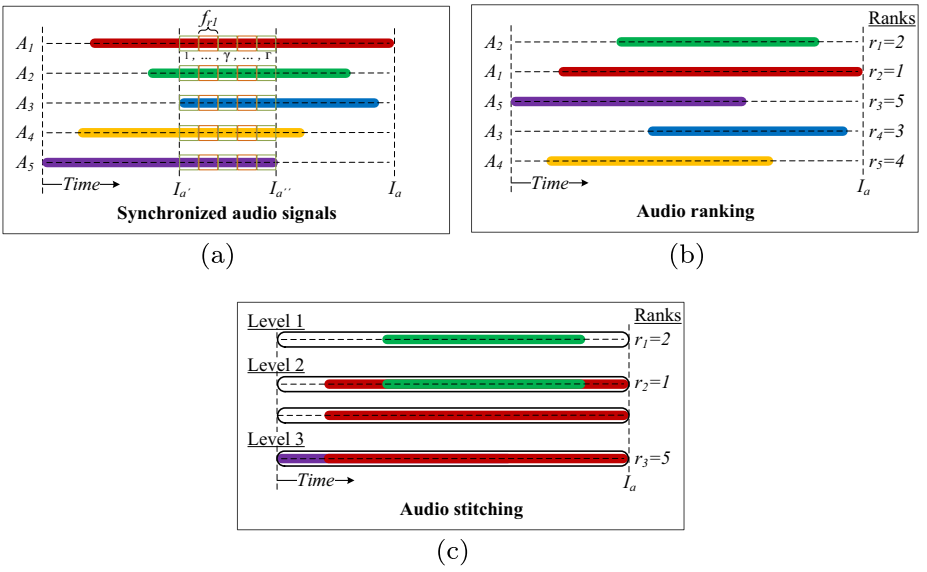

Fig. 3 Audio-stitching illustration (audio signals represented by colored bars). (a) Synchronized A_n are decomposed into non-overlapping frames, Γ, using f_{r1} for the $[I_{a'}, I_{a''}]$ duration. (b) Rank vector, R, is then obtained by analyzing audio quality within each frame. (c) Finally, audio stitching is performed to obtain a continuous audio signal for the complete duration of the event

4.3 Cut-point selection using audio features

According to professional film-editing rules, every cut should have a motivation such as camera motion, occlusion or silence-to-voice transition [10]. In our proposed method, cut-points are selected by analyzing the dynamics of the stitched audio as we assume that it is meaningful to change camera view when a change in audio occurs (e.g. silence to audio/music, change or addition of an instrument, low to high volume, music to vocal).

We propose a cut-point selection method by analyzing low level audio features of A^{st} to detect those audio samples where the change occurs. The three features used are *root mean square*, A^{RMS}, [21] *spectral centroid*, A^{SC}, [21] and *spectral entropy*, A^{SE} [21]. *Root mean square*, A^{RMS}, is useful for detecting silence periods in audio signals and for discriminating between different audio classes. *Spectral centroid*, A^{SC}, is effective in describing the spectral shape of the audio as it measures the center of mass of the audio spectrum, and it is useful for predicting the brightness of the sound. A sudden change in A^{SC} is interpreted as an instrumental change in music [21, 32]. *Spectral entropy*, A^{SE}, is used to detect silence and voice segments of speech [28]. It is also useful for discriminating between speech and music. We compute the change in these features and use their agreement for the cut-point selection.

In our method, we first decompose the input audio signal, A^{st}, into non-overlapping frames $1, \cdots, f, \cdots, F$ with frame size, f_{r2}, (Section 4.4) and compute the low level features A^{RMS}, A^{SC}, A^{SE} within each frame f, as

$$A^{RMS} = [a^{RMS}(1), \cdots, a^{RMS}(f), \cdots, a^{RMS}(F)], \tag{4}$$

$$A^{SC} = [a^{SC}(1), \cdots, a^{SC}(f), \cdots, a^{SC}(F)], \tag{5}$$

$$A^{SE} = [a^{SE}(1), \cdots, a^{SE}(f), \cdots, a^{SE}(F)]. \tag{6}$$

The total number of frames is computed as $F = \frac{I_a}{P f_{r2}}$, where I_a is the total number of samples in A^{st} and P is the sampling rate. We then compute the derivative D^{RMS}, D^{SC}, D^{SE} of the features A^{RMS}, A^{SC}, A^{SE}, as

$$D^{RMS} = [d^{RMS}(1), \cdots, d^{RMS}(f), \cdots, d^{RMS}(F)]. \tag{7}$$

Likewise, we obtain D^{SC} and D^{SE}. The response of the three features computed for the input audio signal along with their derivatives is shown in Fig. 4.

For statistical analysis, we analyze the dynamics of feature derivatives D^{RMS}, D^{SC} and D^{SE} within an analysis window W_a by computing the mean $\bar{\mu} = \left(\mu_a^{RMS}, \mu_a^{SC}, \mu_a^{SE}\right)$ and standard deviation $\Sigma = \left(\sigma_a^{RMS}, \sigma_a^{SC}, \sigma_a^{SE}\right)$ within each W_a. The threshold $\bar{\tau} = \left(\tau_a^{RMS}, \tau_a^{SC}, \tau_a^{SE}\right)$ to be applied within each W_a is computed as

$$\bar{\tau} = \bar{\mu} + \eta\Sigma, \tag{8}$$

where η defines the weight for the standard deviation, Σ, to be applied for computing the outliers within each W_a. For initialization, we set $\eta = 2.5$ by considering that the data under W_a is normally distributed (giving a confidence interval of 0.985 [19]). The threshold, $\bar{\tau}$, is computed within W_a for each feature vector derivative and is locally applied to it. The values of feature vector derivatives above $\bar{\tau}$ correspond to outliers, where there is a significant

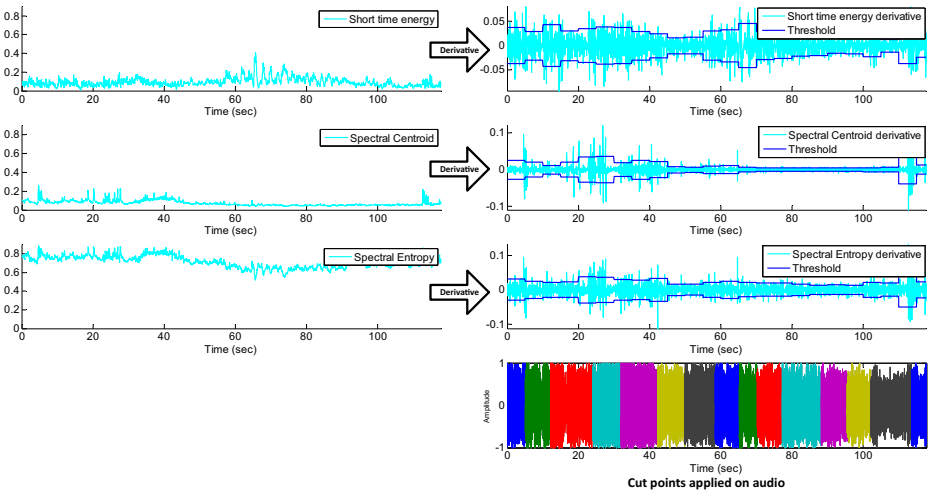

Fig. 4 Audio features extraction and cut-point selection. Root mean square (top left), spectral centroid (middle left) and spectral entropy (bottom left) of the input audio signal. The respective derivatives are shown (on the right). A dynamic threshold is applied within an analysis window W_a and the cut points are computed while staying within the minimum, $l_{\min}$, and maximum, $l_{\max}$, video-shot duration limits

change in the dynamics of the input audio signal. These values are marked as one while the values below $\bar{\tau}$ are marked as 0. This gives the binary value

$$b^{RMS}(f) = \begin{cases} 0 & d^{RMS}(f) < \tau_a^{RMS}, \\ 1 & otherwise, \end{cases} \tag{9}$$

for the binary vector

$$B^{RMS} = [b^{RMS}(1), \cdots, b^{RMS}(f), \cdots, b^{RMS}(F)]. \tag{10}$$

Likewise, B^{SC} and B^{SE} are computed. The three binary vectors are then fused together with a logic AND operator

$$A_U^{st} = B^{RMS} \cdot B^{SC} \cdot B^{SE}. \tag{11}$$

Finally, we overlay the binary vector A_U^{st} on the audio signal to get its suitable cut-points, U. Figure 4 (right) shows the D^{RMS}, D^{SC} and D^{SE} along with the applied threshold $\bar{\tau}$ and the resulting segmented audio signal.

4.4 Parameters for cut-point selection

To decompose an audio signal into frames for the feature extraction, we selected the frame size $f_{r2} = 0.05s$. Typical value for the frame size is between $0.01s$ and $0.05s$ [13, 33]. The frame size should be large enough to have sufficient data for the feature extraction. At the same time, it should be short enough to make the signal (approximately) stationary [13]. In order to validate the frame size selection, we manually labeled an audio signal (of 8 *minutes* duration) to obtain the ground-truth cut-points. We evaluated our proposed cut-point detection method by varying f_{r2} from $0.01s$ to $0.07s$ (Fig. 5a). It is observed that the F_1-score is comparatively high for the typical value range. The performance decreases when the frame size is increased beyond $0.05s$, which suggests that frames are not (approximately) stationary beyond this value. Likewise, the typical value for the analysis window size, W_a, is

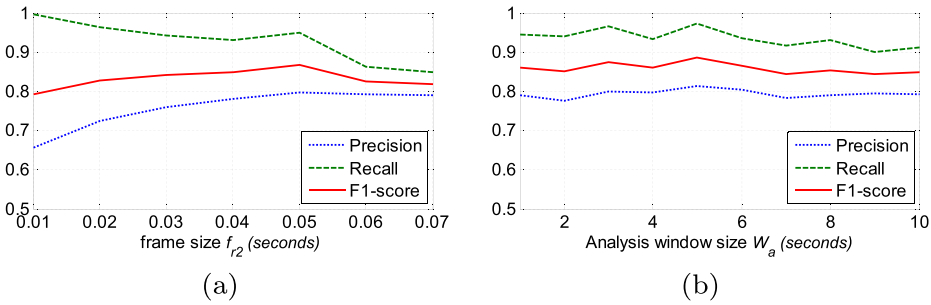

Fig. 5 Analysis of frame, f_{r2}, and analysis window, W_a, size. (a) The effect of varying f_{r2} while fixing $W_a = 5s$. (b) The effect of varying W_a while fixing $f_{r2} = 0.05s$

between $1s$ and $10s$ [13]. We selected $W_a = 5s$ for our proposed method. We demonstrated the effect of varying W_a in Fig. 5b. It is observed that the F_1-score does not vary much between the typical value range and the mean F_1-score is 86 % with standard deviation of 1.4 %.

We selected the minimum, $l_{\min}$, and maximum, $l_{\max}$, limits for the video-shot duration and adjusted the cut-point selection method to satisfy this condition. The $l_{\min}$ and $l_{\max}$ are dependent on the audio genre under study. A segment longer than $l_{\max}$ is perceived as boring and a segment shorter than $I_{\min}$ may not be understandable [3, 43]. In this work, we set the $l_{\min}$ and $l_{\max}$ to $3s$ and $10s$, respectively, and use them to define a meaningful transition from one field of view of a camera to another. We adjust the threshold $\bar{\tau}$ (8) to enforce shot duration limits on the cut-point selection method. When η is high, $\bar{\tau}$ within W_a is high and less frames are detected as outliers, resulting in few cut-points with possible length longer than $l_{\max}$. The threshold $\bar{\tau}$ is lowered iteratively by decreasing η until the $l_{\max}$ condition is satisfied. In order to satisfy the $l_{\min}$ condition, two adjacent segments which are less than $l_{\min}$ apart are merged to obtain one segment.

5 Video analysis

Given the set $\mathbb{C} = \{\mathbb{C}_n\}_{n=1}^{N}$ of multi-camera UGVs, we analyze $\mathbf{V}$ by computing certain visual assessment scores to account for the visual quality, camera motion and view diversity. The video quality assessment aims at obtaining spatial $\mathbf{S} = \{S_n\}_{n=1}^{N}$ and spatio-temporal $\mathbf{T} = \{T_n\}_{n=1}^{N}$ quality scores, where $S_n = (s_{n1}, \cdots, s_{ni}, \cdots, s_{nI_v})$ and $T_n = (t_{n1}, \cdots, t_{ni}, \cdots, t_{nI_v})$, respectively.

5.1 Spatial quality assessment

In order to filter low-quality video frames, we perform spatial quality analysis of UGVs. We use *BRISQUE* [25] (Blind/Referenceless Image Spatial Quality Evaluator) for the image spatial quality-assessment as it quantifies several degradations caused by video compression, image blur and additive white Gaussian noise, as compared to other approaches that are degradation-specific [12, 34, 37, 44]. *BRISQUE* is a non-reference based image quality measure which is designed based on the natural scene statistics [29]. *BRISQUE* is designed using Mean Subtracted Contrast Normalized (MSCN) coefficients [29]. MSCN coefficients refer to a property of natural scene statistics, which states that the subtraction of local means

from image luminances and normalization by local variances produces decorrelated coefficients [29]. *BRISQUE* computes features by fitting a generalized Gaussian distribution to the MSCN coefficients and by fitting asymmetric generalized Gaussian distribution to pairwise products of neighboring MSCN coefficients. Finally, in order to obtain a measure of image quality, BRISQUE learns a mapping between features and human Differential-Mean Opinion Score (DMOS) by using a support vector machine regressor.

Each S_n in $\mathbf{S} = \{S_n\}_{n=1}^{N}$ is synchronized such that an assessment score s_{1i} for $\mathbb{C}_1$ at i^{th} frame corresponds to the same time instant for the score s_{2i} for $\mathbb{C}_2$. $\mathbf{S}$ is normalized using the z-score to have mean equal to zero and standard deviation equal to one.

5.2 Spatio-temporal quality assessment

In order to filter video frames containing unwanted camera movements, we perform spatio-temporal quality analysis of UGVs. We use the approach of Nagasaka and Miyatake [26] in which they estimate the camera pan and tilt using Luminance Projection Correlation (LPC). We use this approach [26] as opposed to other optical flow-based [2] and template matching-based [1] approaches which are computationally expensive. Furthermore, LPC has been previously tested for hand-held camera's video analysis [7]. We obtain the pan signal by projecting the image on the horizontal axis and by correlating it with the projection of the previous image. Likewise, the tilt signal is computed. A threshold [26] is applied to these signals for detecting the pan and tilt (see Fig. 6). Pan left is labeled as positive and right as negative. Tilt up is labeled as positive and down as negative.

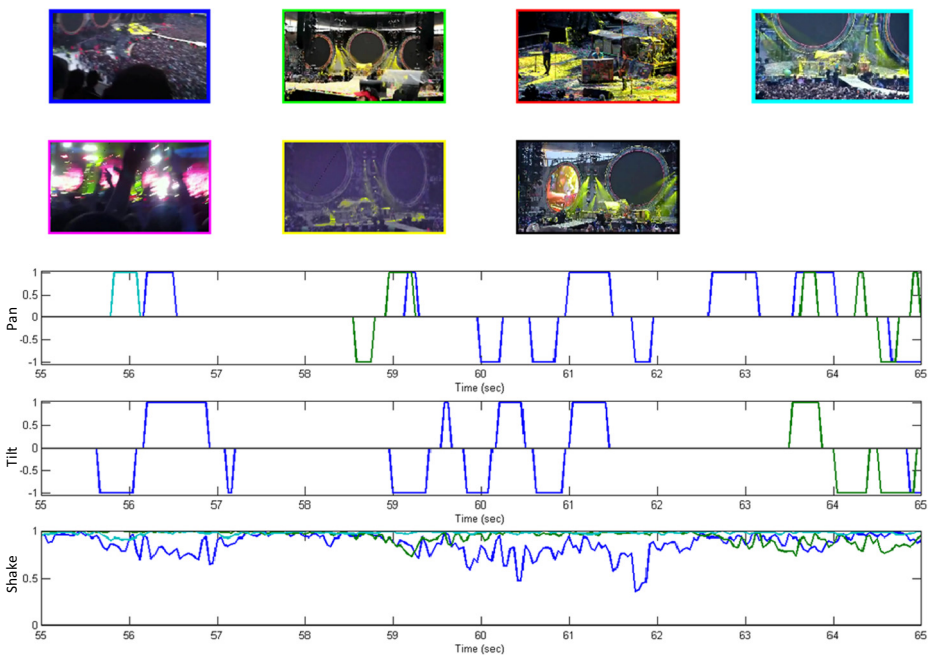

Fig. 6 Camera motion analysis [26]. Pan and tilt are shown along with camera shake score [7] for the three active cameras (labeled with green, blue and cyan colors) from the same event

In order to estimate spatio-temporal quality score which is given by camera shake, we use the method proposed by Campanella et al. [7] in which they apply low pass filtering to the pan and tilt signals [26], and compute the camera shake by taking the difference of original and filtered pan and tilt signals. The computed spatio-temporal quality score $\mathbf{T} = \{T_n\}_{n=1}^{N}$ is normalized using the z-score normalization. The higher the value of t_{ni} (score for the i^{th} frame of the n^{th} camera) the more stable the video. Figure 6 shows the results for camera pan, tilt and shake analysis for two cameras belonging to the same event.

5.3 View diversity

Diversity is defined as the use of a variety of views in the camera selection process in order to increase the information content in the generated video. This enhances the viewing experience and is a component of professionally edited videos [6, 43]. We assume that if at least the past two consecutive selected cameras are different from the current selection, sufficient view diversity is achieved.

In order to impose view diversity, we make use of the past segments. We implement a simple condition that a video selected for the segment M_j differs in view point from the previous segment M_{j-1} and it is not the one selected for the previous two segments M_{j-1} and M_{j-2} provided that we at least have 3 UGVs for an event at that time instant. Figure 7 shows an illustration of the proposed view diversity condition (Fig. 7c) in comparison to when no diversity (Fig. 7a), or history of the previous selected segment (Fig. 7b) is applied for the camera selection. By considering the history of the previous selected segment, switching between two top ranked cameras takes place. In the proposed view diversity condition, switching between three or more cameras takes place by considering their ranks. The rank-based camera selection strategy is presented in the following section.

6 Rank-based camera selection

In order to construct a camera selection strategy, we analyze the spatial, **S**, and spatio-temporal, **T**, assessment within each segment j while considering the view diversity within the last two selected visual segments. We analyze the video segment v_{nj} for all N cameras by using both spatial S_n and spatio-temporal T_n quality scores. We first perform the

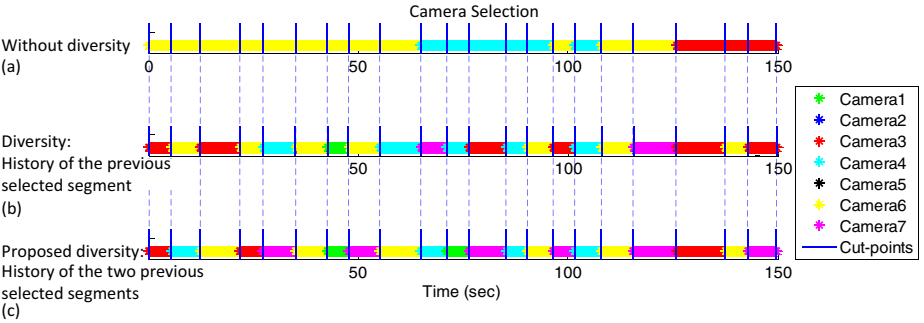

Fig. 7 View diversity illustration. Camera selection is shown for three cases: (a) No diversity condition is applied. (b) History of the previous selected segment is considered for the diversity. (c) Proposed view diversity condition in which history of the two previous selected segments is considered

best camera selection independently with respect to the S_n and T_n scores, and store the selected camera indices in $\mathbf{I}^S$ and $\mathbf{I}^T$, respectively. The selected camera indices are given by $\mathbf{I}_j^S = \left(I_{j1}^S, \cdots, I_{jk}^S, \cdots, I_{jK_j}^S\right) \in u_j$, where K_j is the total number of samples in the segment u_j when analyzing $\mathbf{S}$. The same is applicable for $\mathbf{I}_j^{\mathbf{T}}$. We then compute the normalized occurrence of a camera $\mathbb{C}_n$ in a segment u_j when analyzing S_n, which is given by

$$\widehat{\mathbf{O}}_{nj}^S = \frac{\sum_{k=1}^{K_j} I_{jk}^S}{K_j}, \tag{12}$$

where $I_{jk}^S \in \mathbb{C}_n$. By varying n from $1, \cdots, N$, we get the normalized occurrence for all the cameras in the segment u_j. Similarly, we compute $\widehat{\mathbf{O}}_{nj}^T$ for $\mathbf{I}_j^T$, and arrange $\widehat{\mathbf{O}}_{nj}^S$ and $\widehat{\mathbf{O}}_{nj}^T$ in descending order to get the rank vectors R_j^S and R_j^T, respectively, for all $\mathbb{C}_n$. We then compute the rank vector R_j by combining the unique stable values from R_j^S and R_j^T. By considering the camera ranking in current and past two segments, we develop the camera selection method such that the video selected for M_j should not be the same as in M_{j-1} and M_{j-2}. At level $l = 1$ in a segment u_j, we assign the top combined rank $R_j(1)$ to M_j followed by checking the different camera selected at $l = 2$ and $l = 3$ for imposing view diversity from the past two M_{j-1} and M_{j-2} segments. The complete algorithm for this method is detailed in Algo. 2.

Algorithm 2 The algorithm for rank-based camera selection.

```
Input: R_j, V_n, ∀n = 1 ··· N, j = 1 ··· J
Output: (M)
for j ← 1 to J do
    if j = 1 then
        M_j = R_j(1) % selection for the first segment
    else
        M_j = R_j(1)
        if M_j = M_{j−1} & R_j(2) ≠ 0 then
            M_j = R_j(2) % M_{j−1} check provided at least 2 cameras are active
        end
    end
    if (j − 2) > 0 then
        if M_j = M_{j−1} & R_j(3) ≠ 0 then
            M_j = R_j(3) % M_{j−2} check provided at least 3 cameras are active
        end
    end
end
```

7 ViComp compared with Firstfit and MoViMash

Our method is similar to Firstfit [35] and MoViMash [30] as we also perform visual quality and camera motion analysis for video composition but differ significantly from them as we perform audio stitching and automatic cut-point selection. We used a single spatial quality measure, and applied a rank-based strategy for camera selection. This comparison is presented below in detail.

Firstfit and MoViMash do not consider audio quality. These methods are not fully automatic, and perform manual cut-point selection of the UGVs. We use audio as opposed to manual video cut-point selection as it gives a single cut-point at a time instant to be used for all the UGVs.

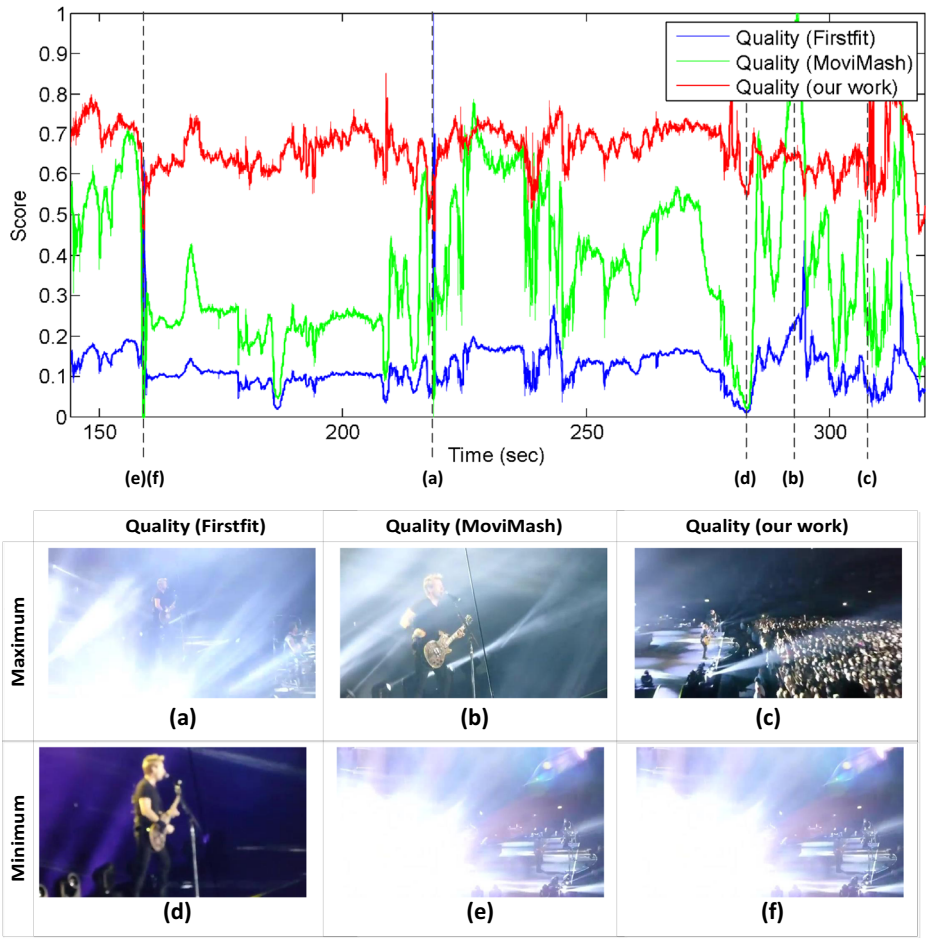

Fig. 8 Comparison of the quality measure from FirstFit [35], MoViMash [30] and proposed *ViComp* (that uses BRISQUE [25]) is shown for a UGV. The frames corresponding to the maximum and minimum scores for each measure are shown for visualization

Firstfit and MoViMash used individual measures for quality assessment. We used instead BRISQUE [25] as it incorporates different visual degradations in a single score. The quality measure of the FirstFit [35] is the multiplication of normalized blockiness, blur, brightness and shake score, and the quality measure of MoViMash [30] is the multiplication of the normalized blockiness, blur, illumination, contrast and burned pixels. Multiplication-based combination might suppress the effect of one individual score over the other. In Fig. 8, a comparison of the quality measure BRISQUE [25] (used in our framework) with respect to the quality measures used in FirstFit and MoViMash is shown. It can be observed that in this example the brightness score is dominating in FirstFit and MoViMash, and frames with low brightness do not get a high score even if their visual quality is good.

MoViMash [30] used a structural similarity based approach (SSIM [39]) for measuring the view similarity between a current video frame and a frame stored in history. SSIM might

not be an appropriate measure for view similarity analysis as UGVs contain varying quality recordings from different viewing angles and distances with narrow and wide baselines.

Firstfit and MoViMash used optimization strategies which rely on weighted addition of quality, cut-point and diversity scores in FirstFit, and quality, diversity and shake scores in MoViMash. These methods require tuning of weights and a shot that gets the highest total score is selected for the current view. We designed a rank-based camera selection strategy to combine the effect of the quality scores (**S**, **T**) along with the view diversity condition. In our proposed strategy, a preferable camera in terms of quality and another preferable camera in terms of shake in a segment are both likely candidate for the current view. View diversity is then imposed to decide which camera to select (Section 6).

8 Experimental validation

In order to validate the proposed *ViComp*, we design a subjective evaluation test and compare it with Firstfit [35] and MoViMash [30]. We test our framework on a dataset of 13 events (93 UGVs).

8.1 Experimental setup

Table 2 details the dataset for the subjective evaluation. Each event was captured by 4 to 12 hand-held cameras which were overlapping in time. Event 1-4 comprise multiple recordings of four different songs from a Nickelback concert. Event 5-8 comprise the multiple recordings of four different songs from an Evanescence concert. Event 9-11 are the same recordings as used by the FirstFit [35] that are pop and rock concerts, and Event 12-13 are the same recordings as used in MoViMash [30] that are dance sequences at a local show.

The UGVs are pre-processed before feeding into the *ViComp* framework as the video frame rate and frame size are varying among the UGVs of the same event. The frame rate

Table 2 Details of the dataset used for testing. All recordings have audio sampled at 44.1 *kHz*. Key: *N* - Number of cameras

Event	*N*	Video frame rate min – max (fps)	Duration min – max(min : *s*)	Coverage duration	Overlap duration	Frame size (pixels)
1	7	16-30	04:01 - 05:20	05:23	04:05	(640, 360)
2	9	16-30	04:00 - 04:42	04:44	03:56	(480, 360), (640, 360)
3	7	16-30	02:26 - 04:46	04:46	03:14	(640, 360)
4	5	24-30	03:20 - 04:56	04:56	03:20	(640, 360)
5	6	25-30	03:17 - 03:57	03:57	03:09	(640, 360), (568, 360)
6	6	29-30	03:02 - 04:03	04:05	02:42	(640, 360),(480, 360)
7	6	25-30	02:57 - 04:08	04:08	02:57	(480, 360), (640, 360)
8	7	24-30	03:35 - 04:04	04:02	03:58	(640, 360), (480, 360)
9 [35]	5	25	04:24 - 04:45	04:44	04:17	(320, 240)
10 [35]	5	25-30	05:01 - 06:58	07:01	04:32	(320, 240)
11 [35]	4	15-30	02:24 - 05:17	05:15	02:47	(320, 240)
12 [30]	12	30	04:01 - 04:57	05:00	04:05	(720, 480)
13 [30]	12	30	03:45 - 04:13	04:13	03:49	(720, 480)

for all UGVs are re-sampled to 25 *fps a-priori* by using VirtualDub [20]. All frames are re-scaled to the same size for all the videos before camera selection. Also, all UGVs belonging to an event are synchronized *a-priori* to a common timeline [4]. For the selection of suitable cut-points, we fixed the value of $l_{\min}$ and $l_{\max}$ to 3 and 10*s*, respectively (Section 4.4). For the evaluation test, we used the overlap duration (as shown in Table 2) that is the duration for which all UGVs in an event are available.

For comparison, we implemented two more strategies, *ViRand*, and *ViCompCD*. In *ViRand*, the video segments are selected randomly at each cut-point while the segment length $l_{\min}$ and $l_{\max}$ are fixed. We also design the *Clustering-based Diversity* (CD) condition and included it in *ViCompCD* for comparison. For implementing the *CD* condition, we develop a strategy for clustering the video frames from N cameras at i^{th} time instant into similar and dissimilar views by matching view points. At a time instant i, the views are organized into clusters C_1 and C_2, where C_1 contains the indices of all views similar to the last frame $(i-1)$ of the previously selected segment M_{j-1}, and C_2 contains the indices of all the dissimilar views. At a time instant i, we apply the Harris affine detector [24] to extract affine invariant regions followed by applying the Scale Invariant Feature Transform (SIFT) descriptor to extract features, $E_n(i)$, in a frame. We used this detector as it is capable of identifying similar regions in pairs of video frames captured from different viewpoints. We compute visual features for all $\mathbb{C}_n$ at the i^{th} time instant, where $E_n(i) \in \mathbb{R}^{Y_n \times 128}$ and Y_n is the number of features extracted in the n^{th} camera. For a camera $\mathbb{C}_{n'}$, we calculate its feature matching with the features $E_n(i)$ of all other cameras. The match count between current $\mathbb{C}_{n'}$ and all $\mathbb{C}_n$ at i^{th} time instant is given by $\Lambda(i) = [\lambda_{n'1}(i), \cdots, \lambda_{n'n}(i), \cdots, \lambda_{n'N}(i)]^T$. The highest number of matches is obtained when $n' = n$. We make this value $\lambda_{n'n'}(i)$ equal to the second highest match value in order to avoid bias in the clustering stage; as when a frame is matched with itself a sufficiently large number of matches occurs as compared to when it is matched with video frames from other camera recordings. Next, we apply k-means clustering by setting $k = 2$ such that C_1 is the cluster with the highest mean value. Ideally, this ensures that C_1 always contains frames from the N cameras at time instant i that contains a similar camera view as of n'. However, this is not always true as visual degradations reduce the sharpness of the video frame; thus making the feature matching insignificant. In order to implement the *CD* condition in the camera selection process, we select a camera index from C_2 for which the combined rank R_j (in the j^{th} segment) is high and satisfies the proposed view diversity condition. Figure 9 shows an example of *CD* strategy. Matching is performed between $\mathbb{C}_7$ (last frame of previously selected camera) and all $\mathbb{C}_n$, as a result frames similar to $\mathbb{C}_7$ form the cluster C_1 while dissimilar frames form the cluster C_2.

8.2 Subjective evaluation

A subjective test is designed to analyze the overall quality of the proposed method (*ViComp*) in comparison with *ViCompCD*, *ViRand*, Firstfit [35] and MoViMash [30]. As there are many ways of showing videos to subjects in order to record their assessment, the ITU-R recommendation [18] presented four standardized methods for the subjective video-quality assessment. We selected Pair Comparison (PC) [18]-like method for analyzing the composed multi-camera video based on a subject's level of interest. Our choice is motivated by the fact that in order to have a fair comparison, a subject must watch all three composed videos of an event before ranking them. For example, if the subject is asked to assess one video at a time, he/she will not be sure what is the reference that defines a good quality. In each test set, we presented the test videos from three methods one after another and asked the subject to provide a comparative rank from the best to the worst video. The subjects

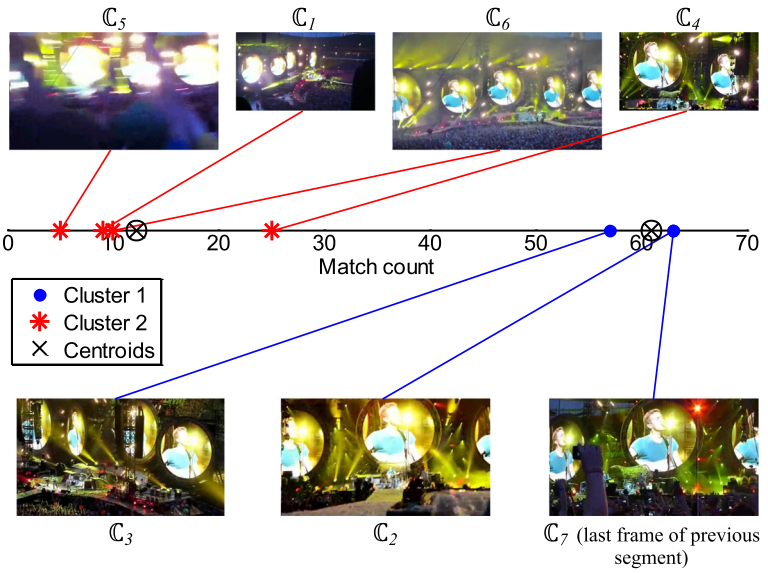

Fig. 9 Clustering-based diversity example. $\mathbb{C}_7$ is the last frame of the previously selected segment which is matched with all $\mathbb{C}_n$. This process divides the videos into two clusters: similar, Cluster 1, and dissimilar, Cluster 2

were not disclosed about the method used to compose these videos. In order for the subject to stay involved in the test and to remember the properties of the videos, the length of each test video is selected to be approximately of 60*s*. Therefore, the videos in a particular test set took 3-4 minutes to be watched and ranked by the subjects. We designed a web-page[2] for the distribution of the test, in which guidelines for taking the test are given to the subjects. The subject's information (name, age, gender) is recorded before the test begins.

The validation is performed by conducting four experiments as detailed in Table 3. In the first and the second experiments, we selected Event 1-4 and Event 5-8, respectively, that contain UGVs of the same artist for the same concert, and tested three methods, namely *ViComp*, *ViCompCD* and *ViRand*. This selection is done in order to avoid a subject's bias towards a particular artist. The output mashups obtained using Firstfit [35] and MoViMash [30] were made available by their authors for Event 9-11 and Event 12-13, respectively. In the third experiment, we used Event 9-11 and tested *ViComp*, *ViCompCD* and FirstFit [35]. In the fourth experiment, we used Event12-13 and tested *ViComp*, *ViCompCD* and MoViMash [30]. The audio in Firstfit [35] is varying and discontinuous which may negatively influence the subject's decision while ranking [5]. In order to remove the bias induced due to varying audio quality, we used the same audio track for all methods.

8.3 The method

We conducted a survey on the quality of videos generated by three methods (Table 3). The null and alternate hypothesis are formulated as H_o = 'There is no significant difference among the videos generated by the three methods', and H_a = 'There is a significant

[2] http://webprojects.eecs.qmul.ac.uk/sb303/evalvid/

Table 3 Details of the subjective experiments and their evaluation. Median age of subjects in all the experiments came out to be approx. 30 years old. Key: Exp. - Experiment number; **M** - Male subjects; **F** - Female subjects; $\mathfrak{X}^2$ - Chi-square statistic

Exp.	Events	Methods under test					Subject		$\mathfrak{X}^2$	p-value
		ViComp (proposed)	ViComp-CD	ViRand	First-fit	MoVi-Mash	M	F		
1	1-4	✓	✓	✓			21	9	120.46	$6.9e^{-27}$
2	5-8	✓	✓	✓			18	9	113.56	$2.2e^{-25}$
3	9-11	✓	✓		✓		26	9	56.11	$6.6e^{-13}$
4	12-13	✓	✓			✓	26	9	51.54	$6.4e^{-12}$

difference among the videos generated by the three methods'. The test is designed as a k-related sample test in which the subjects are told to assign rank 1 to the method which appears to them as the best in terms of visual quality, rank 2 to the second best and rank 3 to the worst. The recorded ranks for the four experiments are presented in Fig. 10. The age of the subjects who took part in the first and second experiments ranged from 19-50 years (median 29.5 years). And the age of the subjects who took part in the third and fourth experiments ranged from 23 to 53 years (median 30 years).

In order to test the consistency in ranking patterns, we used the Friedman Two-Way ANOVA by ranks [17]. In the Friedman Two-Way ANOVA test, the data are arranged in a tabular form in which the rows correspond to blocks (subject's rank for each event) and columns correspond to treatments (the three methods under test). The Friedman Chi-square statistic $\mathfrak{X}^2$ and p-value are computed for all four experiments and are detailed in Table 3. All four results are statistically significant as the p-values are close to zero, hence we can reject the null hypothesis. These sufficiently small p-values suggest that there is at least one column median in each experiment that is significantly different from others. Generally, if the p-value is less than 0.05 or 0.01, it casts doubt on the null hypothesis.

In order to determine which pairs of column effects are significantly different, we perform multiple comparison tests [14] for the four experiments. Figure 11 shows the result for the multiple comparisons of mean column ranks for all four experiments. For the first and the second experiments (Fig. 11a and b, respectively), the mean column ranks of the proposed *ViComp* and *ViCompCD* are significantly different from the *ViRand* one. For the third experiment (Fig. 11c), the mean column rank of the *ViComp* is significantly different from the Firstfit [35] one. Since the events used in this experiment are of poor visual quality and with limited number of UGVs, the subjects found difficulty to judge the overall quality (Section 8.4). For the fourth experiment (Fig. 11d), the mean column ranks of the proposed *ViComp* and *ViCompCD* are significantly different from the MoViMash [30] one.

8.4 Discussion

The subjective evaluation suggests that the quality of *ViComp* and *ViCompCD* is comparable in some events but overall *ViComp* outperformed all the other methods (see Figs. 10 and 11). In general, *ViRand*, Firstfit [35] and MoViMash [30] received lower ranks.

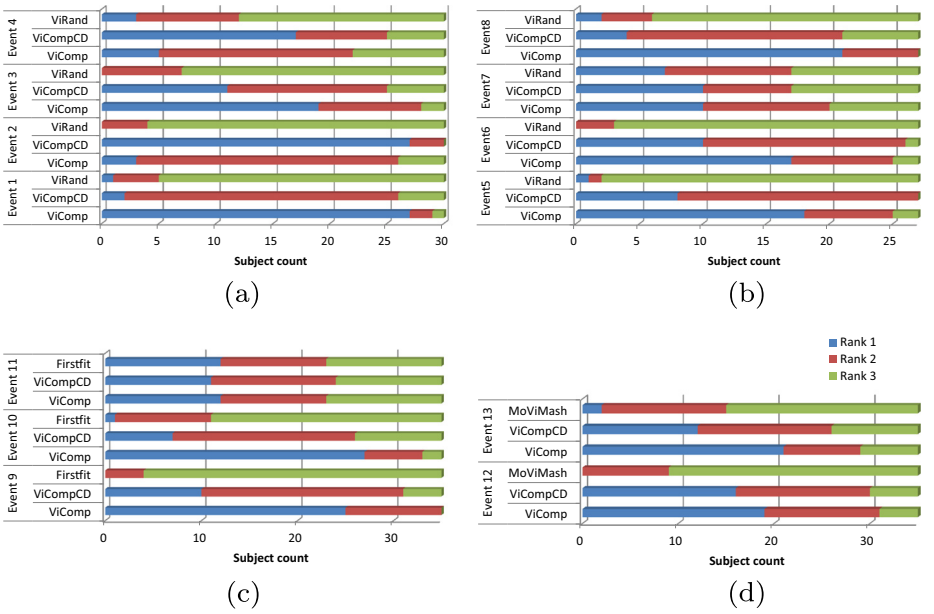

Fig. 10 Subjective evaluation test: (a) Experiment 1: Ranks assigned by subjects for the videos composed by *ViComp* (proposed), *ViCompCD* and *ViRand* for the Nickelback concert, (b) Experiment 2: Ranks assigned by subjects for the videos composed by *ViComp* (proposed), *ViCompCD* and *ViRand* for the Evanescence concert, (c) Experiment 3: Ranks assigned by subjects for the videos composed by *ViComp* (proposed), *ViCompCD* and Firstfit [35] for the Events from Firstfit, (d) Experiment 4: Ranks assigned by subjects for the videos composed by *ViComp* (proposed), *ViCompCD* and MoViMash [30] for the Events from MoViMash

For the first experiment (Fig. 10a), a general observation is that *ViRand* was ranked low while the ranks for *ViComp* and *ViCompCD* were comparable. This is verified by the multiple comparison test (Fig. 11a). For Event1, *ViComp* received a higher rank than *ViCompCD*, while for Event 2, this order was reversed. Similarly for Event 3 and Event 4, *ViComp* and *ViCompCD* received comparable ranks. Note that for Event 4, *ViRand* received a sufficiently high rank but not higher than *ViComp* and *ViCompCD*. This is because Event 4 consists of 5 UGVs, all of them having comparable visual quality, which makes difficult for a subject to take a decision.

For the second experiment (Fig. 10b), *ViComp* and *ViCompCD* outperformed *ViRand* for Event 5, 6 and 8. An interesting case is the one of Event 7, in which the subjects seemed confused about the quality of the videos and found difficult to take a decision. This is because all 6 UGVs in this event are either from far field of view (with less shake) or near field of view (with a lot of shake). The composed videos are therefore not interesting to playback as far fields of view do not give much information of the event and near fields of view seemed unpleasant because of high camera-shake.

For the third experiment (Fig. 10c), *ViComp* outperformed the other two methods. All three events used in this experiment contained 4-5 overlapping UGVs, having low resolution (320×240 pixels). An interesting case is the one of Event11, from which it can be observed that subject's agreement is not achieved. This is because this event contained 4 UGVs, all having poor visual-quality (jerky and shake) and compression artifacts. The resulting composed videos from all the three methods are therefore indistinguishable.

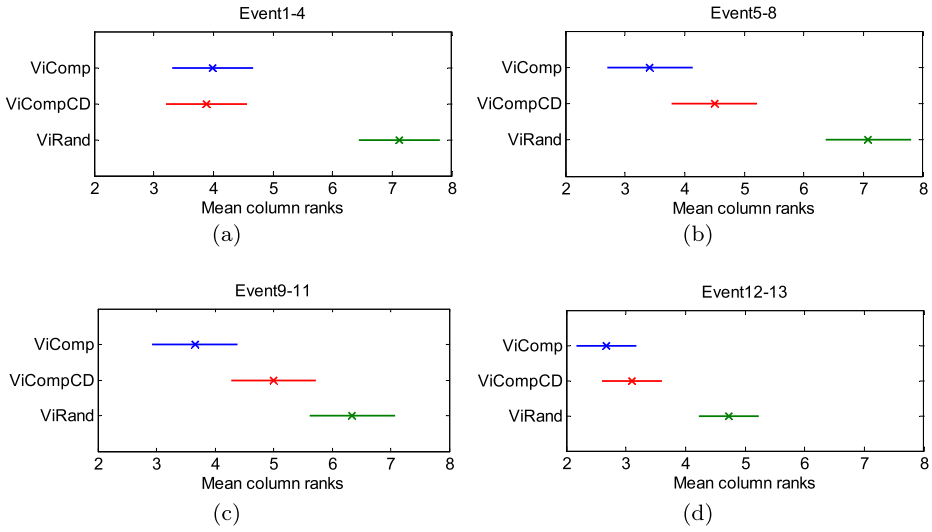

Fig. 11 The corresponding multiple comparison of mean column ranks for subjective test shown in Fig. 10: Comparison is shown for the (a) Experiment 1 (Event 1-4), (b) Experiment 2 (Event 5-8), (c) Experiment 3 (Event 9-11), and (d) Experiment 4 (Event 12-13)

For the fourth experiment (Fig. 10d), *ViComp* and *ViCompCD* both gave comparable results. An interesting case is the one of Event12, in which *ViComp* and *ViCompCD* received equal ranks from the subjects. This is mainly because both Event12 and Event13 contained 12 UGVs of comparable quality that were recorded from near field. The mean column ranks for *ViComp* and *ViCompCD* are significantly different from MoViMash [30]. This is because in MoViMash, UGVs containing high brightness (and poor visual quality) are selected as a consequence of learning the field-of-view distributions. Also, sometimes the length of a selected video segment in MoViMash is as small as 1s. This is because at every second, MoViMash checked for occlusions and shake against a threshold to trigger camera switching, which created an unpleasant effect.

In some cases *ViComp* outperformed *ViCompCD* and vice versa (example Event 1 and Event 2 in Fig. 10a). Since dissimilar and similar clusters are formed in the *CD* condition, video segments which receive a lower total rank (based on quality) might also get selected if they belong to the dissimilar cluster. Without the *CD* condition, video segments with better quality are selected while considering the past two selected segments. As these two methods are sometimes comparable, a better choice would be to select *ViComp* as it is computationally less expensive. In general, *ViComp* outperformed *ViCompCD*. Furthermore, clustering-based diversity (*ViCompCD*) and SSIM-based diversity (MoViMash [30]) lowered the overall quality of the generated videos.

9 Conclusions

We proposed a framework for automatic multi-camera composition from user-generated videos (UGVs) of the same event. The framework combined audio-visual quality and view diversity to generate a coherent recording of the complete event to enhance the viewing experience. Unlike FirstFit [35] and MoViMash [30], we performed the analysis of audio

signals of UGVs and proposed a stitching method to solve the audio variation issue, that occurs when switching between cameras. We also proposed an automatic cut-point selection method by analyzing the change in audio. We imposed a video-shot length condition on the cut-points, and low quality and shaky video segments that received low score were automatically filtered during the camera selection process. We applied a rank-based strategy for camera selection. Our framework was tested on a dataset of 13 events (93 UGVs). In order to analyze the user satisfaction, we designed a subjective test by considering the ITU-R recommendations [18]. The subjective evaluation showed better or comparable results of *ViComp* with its variant *ViCompCD*, and *ViComp* outperformed *ViRand*, FirstFit [35] and MoViMash [30].

As future work, we are interested in analyzing the semantic details of audio and visual data, which may further enhance the quality of the composed videos. Additionally, since smartphones are equipped with inertial sensors (i.e. accelerometer, gyroscope, magnetometer), we are interested in obtaining the video quality score from the motion analysis of these sensors.

References

1. Abdollahian G, Taskiran CM, Pizlo Z, Delp EJ (2010) Camera motion-based analysis of user generated video. IEEE Trans Multimedia 12(1):28–41
2. Almeida J, Minetto R, Almeida TA, Torres RS, Leite NJ (2009) Robust estimation of camera motion using optical flow models. In: Advances in Visual Computing, pp. 435–446. Springer
3. Arev I, Park HS, Sheikh Y, Hodgins J, Shamir A (2014) Automatic editing of footage from multiple social cameras. ACM Trans Graphics 33(4):81
4. Bano S, Cavallaro A (2014) Discovery and organization of multi-camera user-generated videos of the same event. Elsevier Information Sciences 302:108–121
5. Beerends JG, De Caluwe FE (1999) The influence of video quality on perceived audio quality and vice versa. J Audio Eng Soc 47(5):355–362
6. Bowen CJ, Thompson R (2013) Grammar of the Edit. CRC Press
7. Campanella M, Weda H, Barbieri M (2007) Edit while watching: home video editing made easy. In: Electronic Imaging, vol. 6506, p. 65060L. International Society for Optics and Photonics
8. Daniyal F, Taj M, Cavallaro A (2010) Content and task-based view selection from multiple video streams. Multimedia Tools and Applications 46:235–258
9. Dickson PE, Adrion WR, Hanson AR, Arbour DT (2009) First experiences with a classroom recording system. In: Proceedings of the ACM SIGCSE Conference on Innovation and Technology in Computer Science Education, Paris, France, vol. 41, pp. 298–302. ACM
10. Dmytryk E (1984) On Film Editing. Focal Press
11. D'Orazio T, Leo M (2010) A review of vision-based systems for soccer video analysis. Pattern Recognit 43(8):2911–2926
12. Ferzli R, Karam LJ (2009) A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb). IEEE Trans Image Process 18(4):717–728
13. Giannakopoulos T (2009) Study and application of acoustic information for the detection of harmful content, and fusion with visual information. Department of Informatics and Telecommunications, vol. PhD. University of Athens, Greece
14. Hochberg Y, Tamhane AC (1987) Multiple comparison procedures. Wiley
15. Hua XS, Lu L, Zhang HJ (2003) Ave: automated home video editing. In: Proceedings of the ACM International Conference on Multimedia, California, USA, pp. 490–497. ACM
16. Hua XS, Lu L, Zhang HJ (2004) Optimization-based automated home video editing system. IEEE Trans Circuits Syst Video Technol 14(5):572–583

17. Israel D (2009) Data analysis in business research: A step-by-step nonparametric approach. SAGE Publications
18. ITU-T RECOMMENDATION P. (1999) Subjective video quality assessment methods for multimedia applications
19. Kenney JF (1962) Mathematics of Statistics part I. Princeton, NJ: Van Nostrand
20. Lee A (2001) Virtualdub home page. http://www.virtualdub.org/index
21. Lerch A (2012) An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics. Wiley-Blackwell
22. Lu L, Jiang H, Zhang H (2001) A robust audio classification and segmentation method. In: Proceedings of the ACM International conference on Multimedia, Ottawa, Canada, pp. 203–211. ACM
23. Mei T, Hua XS, Zhu CZ, Zhou HQ, Li S (2007) Home video visual quality assessment with spatiotemporal factors. IEEE Trans Circuits Syst Video Technol 17(6):699–706
24. Mikolajczyk K, Schmid C (2004) Scale & affine invariant interest point detectors. Int J Comput Vision 60(1):63–86
25. Mittal A, Moorthy A, Bovik A (2012) No-reference image quality assessment in the spatial domain. IEEE Trans Image Process 21(12)
26. Nagasaka A, Miyatake T (1999) Real-time video mosaics using luminance-projection correlation. Trans. IEICE:1572–1580
27. Ranjan A, Henrikson R, Birnholtz J, Balakrishnan R, Lee D (2010) Automatic camera control using unobtrusive vision and audio tracking. In: Proceedings of Graphics Interface, pp. 47–54. Canadian Information Processing Society
28. Renevey P, Drygajlo A (2001) Entropy based voice activity detection in very noisy conditions. In: Proceedings of the 7th European Conference on Speech Communication and Technology, 2nd INTERSPEECH Event, Aalborg, Denmark, pp. 1887–1890
29. Ruderman DL (1994) The statistics of natural images. Netw Comput Neural Syst 5(4):517–548
30. Saini MK, Gadde R, Yan S, Ooi WT (2012) Movimash: online mobile video mashup. In: Proceedings of the ACM International Conference on Multimedia, pp. 139–48. ACM
31. Scheirer ED (1998) Tempo and beat analysis of acoustic musical signals. J Acoust Soc Am 103:588
32. Schubert E, Wolfe J, Tarnopolsky A (2004) Spectral centroid and timbre in complex, multiple instrumental textures. In: Proceedings of the International conference on Music Perception and Cognition, North Western University, Illinois, pp. 112–116
33. Schuller BW (2013) Intelligent audio analysis. Springer
34. Sheikh HR, Bovik AC, Cormack L (2005) No-reference quality assessment using natural scene statistics: Jpeg2000. IEEE Trans Image Process 14(11):1918–1927
35. Shrestha P, Weda H, Barbieri M, Aarts EHL et al. (2010) Automatic mashup generation from multiple-camera concert recordings. In: Proceedings of the ACM International Conference on Multimedia, pp. 541–550. ACM
36. Snavely N, Seitz SM, Szeliski R (2006) Photo tourism: exploring photo collections in 3d. ACM Transactions on Graphics 25(3):835–846
37. Suthaharan S (2009) No-reference visually significant blocking artifact metric for natural scene images. Signal Process 89(8):1647–1652
38. Wang J, Xu C, Chng E, Lu H, Tian Q (2008) Automatic composition of broadcast sports video. Multimedia Systems 14(4):179–193
39. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: From error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
40. Wilk S, Effelsberg W (2014) The influence of camera shakes, harmful occlusions and camera misalignment on the perceived quality in user generated video. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Chengdu, China, pp. 1–6. IEEE
41. Winkler MB, Hover KM, Hadjakos A, Muhlhauser M (2012) Automatic camera control for tracking a presenter during a talk. In: Proceedings of the IEEE International Symposium on Multimedia (ISM), California, USA, pp. 471–476. IEEE
42. Yu Z, Nakamura Y (2010) Smart meeting systems: A survey of state-of-the-art and open issues. ACM Computing Surveys (CSUR) 42(2):8
43. Zettl H (2011) Sight, sound, motion: Applied media aesthetics. Wadsworth Publishing
44. Zhang J, Ong SH, Le TM (2011) Kurtosis-based no-reference quality assessment of jpeg2000 images. Signal Process Image Commun 26(1):13–23

Sophia Bano received the Bachelor of Mechatronics Engineering degree from the National University of Sciences and Technology (NUST) in 2005, the M.Sc. degree in Electrical Engineering from the National University of Sciences and Technology (NUST) in 2007, the M.Sc. degree in vision and robotics (VIBOT), a joint Masters program in three European universities: Heriot-Watt University, Edinburgh, U.K., University of Girona, Girona, Spain, and the University of Burgundy, Dijon, France, in 2011. Since January 2012, she has been with Queen Mary University of London, UK, and Universitat Politècnica de Catalunya, Barcelona, Spain, as a PhD researcher under the supervision of Prof. A. Cavallaro and Prof. X. Parra. She was awarded Erasmus Mundus scholarship for her M.Sc. and Erasmus Mundus fellowship for her Double Doctorate in Interactive and Cognitive Environments. Her research interests include multimodal analysis of user-generated videos, event discovery and multi-camera synchronization.

Andrea Cavallaro is Professor of Multimedia Signal Processing and Director of the Centre for Intelligent Sensing at Queen Mary University of London, UK. He received his Ph.D. in Electrical Engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, in 2002. He was a Research Fellow with British Telecommunications (BT) in 2004/2005 and was awarded the Royal Academy of Engineering teaching Prize in 2007; three student paper awards on target tracking and perceptually sensitive coding at IEEE ICASSP in 2005, 2007 and 2009; and the best paper award at IEEE AVSS 2009. Prof. Cavallaro is Area Editor for the IEEE Signal Processing Magazine and Associate Editor for the IEEE Transactions on Image Processing. He is an elected member of the IEEE Signal Processing Society, Image, Video, and Multidimensional Signal Processing Technical Committee, and chair of its Awards committee. He served as an elected member of the IEEE Signal Processing Society, Multimedia Signal Processing Technical Committee, as Associate Editor for the IEEE Transactions on Multimedia and the IEEE Transactions on Signal Processing, and as Guest Editor for seven international journals. He was General Chair for IEEE/ACM ICDSC 2009, BMVC 2009, M2SFA2 2008, SSPE 2007, and IEEE AVSS 2007. Prof. Cavallaro was Technical Program chair of IEEE AVSS 2011, the European Signal Processing Conference (EUSIPCO 2008) and of WIAMIS 2010. He has published more than 130 journal and conference papers, one monograph on Video tracking (2011,Wiley) and three edited books: Multi-camera networks (2009, Elsevier); Analysis, retrieval and delivery of multimedia content (2012, Springer); and Intelligent multimedia surveillance (2013, Springer).