Identification, synchronisation and composition of user-generated videos

by

Sophia Bano

BE in Mechatronics Engineering 2005

MSc in Electrical Engineering 2008

MSc in Vision and Robotics 2011

A dissertation submitted to
The School of Electronic Engineering and Computer Science
in partial fulfilment of the requirements for the Degree of
Doctor of Philosophy
in the subject of
Interactive and Cognitive Environments

Queen Mary University of London
Mile End Road
E1 4NS, London, UK

Month 2015

Acknowledgements

This PhD Thesis has been developed in the framework of, and according to, the rules of the Erasmus Mundus Joint Doctorate on Interactive and Cognitive Environments EMJD ICE [FPA n° 2010-0012] with the cooperation of the following Universities:

Alpen-Adria-Universität Klagenfurt – AAU

Queen Mary, University of London – QMUL

Technische Universiteit Eindhoven – TU/e

Università degli Studi di Genova – UNIGE

Universitat Politècnica Catalunya – UPC

According to ICE regulations, the Italian PhD title has also been awarded by the Università degli Studi di Genova.

Acknowledgements

First and above all, I praise Allah Almighty for the strengths and His countless blessings that have always kept me going. I owe a special thanks to my parents and siblings to whom I dedicate all my achievements for their continuous love and support. I am truly grateful to my primary supervisor Professor Andrea Cavallaro and my secondary supervisor Professor Xavier Parra for their guidance, encouragement and suggestions which made this research possible. Finally, I am thankful to all my colleagues with whom I shared the period of my PhD studies for the open discussions and everlasting friendship.

First supervisor	Second supervisor	Author
Professor Andrea Cavallaro	Professor Xavier Parra	Sophia Bano

Identification, synchronisation and composition of user-generated videos

Abstract

The increasing availability of smartphones is facilitating people to capture videos of their experience when attending events such as concerts, sports competitions and public rallies. The captured User-Generated Videos (UGVs) are made available on media sharing websites. Searching and mining of UGVs of the same event are challenging due to inconsistent tags or incorrect timestamps. A UGV recorded from a fixed location contains monotonic content and unintentional camera motions, which may make it less interesting to playback. Smartphones are equipped with inertial sensors which could be beneficial for event understanding. In this thesis, we propose the following identification, synchronisation and video composition frameworks for UGVs.

We propose a framework for the automatic identification and synchronisation of unedited multi-camera UGVs within a database. The proposed framework analyses the sound to match and cluster UGVs that capture the same spatio-temporal event, and estimate their relative time-shift to temporally align them. We design a novel descriptor derived from the pairwise matching of audio chroma features of UGVs. The descriptor facilitates the definition of a classification threshold for automatic query-by-example event identification. We contribute a database of 263 multi-camera UGVs of 48 real-world events. We evaluate the proposed framework on this database and compare it with state-of-the-art methods. Experimental results show the effectiveness of the proposed approach in the presence of audio degradations (channel noise, ambient noise, reverberations).

Moreover, we present an automatic audio and visual-based camera selection framework for composing uninterrupted recording from synchronised multi-camera UGVs of the same event. We design an automatic audio-based cut-point selection method that provides a common reference for audio and video segmentation. To filter low quality video segments, spatial and spatio-temporal assessments are computed. The framework combines segments of UGVs using a rank-based camera selection strategy by considering visual quality scores and view diversity. The proposed framework is validated on a dataset of 13 events (93 UGVs) through subjective tests and compared with state-of-the-art methods. Suitable cut-point selection, specific visual qual-

ity assessments and rank-based camera selection contribute to the superiority of the proposed framework over the existing methods.

Finally, we contribute a method for Camera Motion Detection using Gyroscope for UGVs captured from smartphones and design a gyro-based quality score for video composition. The gyroscope measures the angular velocity of the smartphone that can be use for camera motion analysis. We evaluate the proposed camera motion detection method on a dataset of 24 multimodal UGVs captured by us, and compare it with existing visual and inertial sensor-based methods. By designing a gyro-based score to quantify the goodness of the multi-camera UGVs, we develop a gyro-based video composition framework. A gyro-based score substitutes the spatial and spatio-temporal scores and reduces the computational complexity. We contribute a multimodal dataset of 3 events (12 UGVs), which is used to validate the proposed gyro-based video composition framework.

Contents

Published work

Journal papers

[J1] S. Bano and A. Cavallaro. Discovery and organization of user-generated videos of the same event. Elsevier Information Sciences, Vol. 302, pp. 108-121, May 2015.

[J2] S. Bano and A. Cavallaro. ViComp: Composition of User-Generated Videos. Multimedia tools and applications, in press, 2015.

Conference papers

[C1] S. Bano, A. Cavallaro and X. Parra. Gyro-based camera motion detection in user-generated videos. In ACM Multimedia (MM'15), Brisbane, Australia, 26 - 30 October, 2015.

Electronic preprints are available at http://www.eecs.qmul.ac.uk/~andrea/publications.html.

Glossary of abbreviations

AC	Audio Chroma	42
AF	Audio Fingerprint	42
AFC	Audio Feature Classification	47
AO	Audio Onset	47
AS	Amplified Sound recordings	42
BER	Bit Error Rate	33
BRISQUE	Blind/Referenceless Image Spatial Quality Evaluator	71
CD	Clustering-based Diversity	98
CE	Conditional Entropy	40
CMDG	Gyro-based Camera Motion Detection	60
CSLBP	Center-Symmetric Local Binary Pattern	40
DBN	Dynamic Bayesian Networks	52
DFT	Discrete Fourier Transform	56
DMOS	Differential-Mean Opinion Score	72
DoG	Difference of Gaussian	41
ERB	Equivalent Rectangular Bandwidth	31
EVVE	EVent VidEo	40
FAST	Features from Accelerated Segment Test	40
GS	Global Signature	41
GUI	Graphical User Interface	114
HB	High Brightness	113
HOOP	Histograms of Orientations of Optical Flow	40
HSV	Hue Satuation Value	41

IGD	Insensitive to Global Degradations	42
ILD	Insensitive to Local Degradations	42
LB	Low Brightness	113
LBP	Local Binary Pattern	40
LPF	Low-Pass Filter	61
LS	Local Signature	41
MFCC	Mel-Frequency Cepstral Coefficient	32
MSCN	Mean Subtracted Contrast Normalised (MSCN) coefficients	72
NAS	Non-Amplified Sound recordings	42
NDV	Near Duplicate Video	29
NDVR	Near Duplicate Video retrieval	39
NRC	Non-Rigid Camera	45
PCA-SIFT	Principal Component Analysis-based Scale-Invariant Feature Transform	40
PI	Pixel Intensity	47
PP	Professional Production recordings	42
PTZ	Pan-Tilt-Zoom	52
RANSAC	RANdom SAmple Consensus	43
RMSE	Root Mean Square Error	61
ROI	Region of Interest	22
SSBelt	Self-Similarity Belt	40
STFT	Short-Time Fourier Transform	33
STIP	Space-Time Interest Point	44
SVM	Support Vector Machine	39
UE	Unconstrained Environment	45
UGV	User-Generated Video	21
ViComp	Video Composition	88
ViCompCD	Video Composition using Clustering-based view Diversity	94
ViRand	Video composition using Random camera selection	94

Glossary of symbols

A^{RMS}	root mean square of A^{st}	65
A^{SC}	spectral centroid of A^{st}	65
A^{SE}	spectral entropy of A^{st}	65
A^{st}	stitched audio of an event	26
A^{st}_{U}	binary vector for the suitable cut-points	65
A_n	audio signal of the nth video recording	25
$A_n^{SR}(g_R)$	spectral rolloff at the g_Rth frame of A_n	66
B^{RMS}	binary vector from D^{RMS}	70
B^{SC}	binary vector from D^{SC}	70
B^{SE}	binary vector from D^{SE}	70
D^{RMS}	derivative of A^{RMS}	68
D^{SC}	derivative of A^{SC}	68
D^{SE}	derivative of A^{SE}	68
D_c	coverage duration of the event	26
D_{P_x}	horizontal projection distance between two consecutive video frames	35
D_{P_y}	vertical projection distance between two consecutive video frames	35
E_k	k^{th} cluster containing spatio-temporal overlapping video recordings	24
E_n	relative ending time of A_n	66
$F_n^{SF}(i)$	SIFT features for the i^{th} frame in C_n	99
F_n	chroma feature vector for the audio A_n	56
$G_q^H(t)$	angle from $G_x^H(t)$ and $G_y^H(t)$	63
$G_q^L(t)$	angle from $G_x^L(t)$ and $G_y^L(t)$	62
G_n	gyroscope data of the n^{th} video recording	25
$G_r^H(t)$	magnitude from $G_x^H(t)$ and $G_y^H(t)$	63
$G_r^L(t)$	magnitude from $G_x^L(t)$ and $G_y^L(t)$	62

$G_x^H(t)$	high-pass filtered signal of $G_x(t)$	63
$G_x^L(t)$	low-pass filtered signal of $G_x(t)$	62
$G_y^H(t)$	high-pass filtered signal of $G_y(t)$	63
$G_y^L(t)$	low-pass filtered signal of $G_y(t)$	62
G_{nx}	gyroscope signal of the n^{th} recording for the rotation around the x-axis	25
G_{ny}	gyroscope signal of the n^{th} recording for the rotation around the y-axis	25
G_{nz}	gyroscope signal of the n^{th} recording for the rotation around the z-axis	25
$H_{ij}(Dt)$	matching histogram for video recordings C_i and C_j	57
I_a	audio sample corresponding to the last recorded audio sample in C	26
I_g	Inertial sensor sample corresponding to the last recorded sample in C	26
I_v	video frame corresponding to the last recorded video frame in C	25
$I_{a^{00}}$	audio sample corresponding to the end of D_o	65
I_{a^0}	audio sample corresponding to the start of D_o	65
K_n^a	total number of audio samples in A_n	25
K_n^g	total number of gyroscope data samples in G_n	25
K_n^v	total number of visual frames in V_n	25
K_n^{SF}	number of extracted features from the i^{th} frame in C_n	99
L_n	relative starting time of A_n	66
L_x	horizontal displacements of pixels in a video V	35
L_x	vertical displacements of pixels in a video V	35
M_j	j^{th} segment of M	26
N_k	number of UGVs contained in the event E_k	24
N_n	total number of non-match descriptors	80
N_p	total number of match descriptors	80
P_{ij}^0	histogram descriptor that is the derivative of P_{ij}	78
P_x	vertical projection of video frame	34
P_y	horizontal projection of video frame	34
P_{ij}	match count between C_i and C_i with respect to T_r	79
Q_{jk}^S	spatial score-based camera selection index for the k^{th} sample in u^j	91
R^{SR}	combined rank vector for the quality indices of N audio recordings in A	66
R_j^C	combined rank for u_j	91

R_j^S	spatial score-based rank for u_j	91
R_j^T	spatio-temporal score-based rank for u_j	91
S_n	spatial scores for the n^{th} video recording in C.	26
T_n	spatio-temporal scores for the n^{th} video recording in C	26
T_r	scanning threshold	79
U	detected suitable cut-points for an event	26
V_x^{OF}	x-component of the optical flow	36
V_y^{OF}	y-component of the optical flow	36
V_n	visual signal of the n^{th} video recording	25
W_a	analysis window	69
Y_n	gyro-based scores for the n^{th} video recording in C.	26
Dd_{ij}	distance difference between C_i and C_j from the sound source	77
Dt_{ij}	time-shift between the i^{th} and j^{th} video recordings	25
$Dt_{q;1:N_k}$	time-shift between C_q and set of UGVs f $C_{k;n}g_{n=1}^{N_k}$	76
G_C	total number of audio frames in D_c of A for cut-point detection	68
G_R	total number of audio frames in D_o of A for ranking	65
G_n	total number of audio frames in A_n	56
L (i)	vector for the feature match count at the i^{th} frame	99
S_a	vector of standard deviation values within W_a	69
°	classification threshold	78
a	tolerance angle	62
$\bar{m}_a$	vector of mean values within W_a	69
$\bar{t}_a$	vector of threshold values within W_a	69
b	tolerance magnitude	63
c	points of minimum distance across each row of $\hat{}_{ij}$	57
d_i	distance of C_i from the sound source	77
d p	panning parameter	35
$\dot{A}^{st}$	intermediate stitched audio	67
h	incremental step	79
g_C	g^{th} audio frame in G_C	68
g_R	g^{th} audio frame in G_R	65

$\hat{e}_x$	value of the correlation peak along x-axis for gyro-visual data synchronisation .	61
$\hat{f}_x$	time-shift along x-axis for gyro-visual data synchronisation	61
$\hat{f}_y$	time-shift along y-axis for gyro-visual data synchronisation	61
k_{N_n}	total number of clusters in N_n non-match descriptors	80
k_{N_p}	total number of clusters in N_p match descriptors	80
$l_{n^0 n}(i)$	feature match count between C_n and C_{n^0} at the i^{th} frame	99
C_m	m^{th} video recording from C	24
C_q	query video recording	25
$C_{k;n_k}$	n^{th} recording belonging to the event E_k	24
D	delay matrix	78
I	identification matrix	80
P^0	row-wise rearrangement of all P^0_{ij}	80
T	set of training vectors	80
A	set of audio signals contained in C	25
A^{SR}	spectral rolloff for the set of audio signals A	65
$A^{SR}(g_R)$	spectral rolloff at the $g_R{}^{th}$ frame of A	66
G	gyroscope data for the set of recordingsC	25
H	matching histograms for all pairs of video recordings in C	78
M	time continuous composed video	26
O^S_{nj}	spatial score-based normalised occurrence of C_n in u_j	91
O^T_{nj}	spatio-temporal score-based normalised occurrence of C_n in u_j	91
Q^S	spatial score-based camera selection indices	91
Q^T	spatio-temporal score-based camera selection indices for u^j	91
Q^T	spatio-temporal score-based camera selection indices	91
Q^S_j	spatial score-based camera selection indices for u^j	91
R^C	combined rank matrix	91
R^S	spatial score-based rank matrix	91
R^T	spatio-temporal score-based rank matrix	91
R^{SR}	rank matrix from A^{SR}	66
S	set of spatial scores for all visual recordings in C	26

T	set of spatio-temporal scores for all visual recordings in C	26
V	set of visual recordings contained in C	25
Y	set of gyro-based scores for all visual recordings in C	26
v	12-dimensional chroma feature vector	34
v_n^p	chroma feature vector for the pth audio frame of the nth video recording	56
I_{ij}	identification value for a pair of video recordings C_i and C_j	80
$P(t)$	detected pan signal in a UGV	62
P_{nd}	detected pan signal for nth recording after morphological filtering	25
$R_x(t)$	cross-correlation of L_x and G_x	61
$R_y(t)$	cross-correlation of L_y and G_y	61
$S(t)$	detected shake signal in a UGV	63
S_{nd}	detected shake signal for nth recording after morphological filtering	25
$T(t)$	detected tilt signal in a UGV	63
T_{nd}	detected tilt signal for nth recording after morphological filtering	25
V_d	12 dimensions (semitones) of the chroma vector v	34
C	set of synchronised multi-camera UGVs of an event	25
f_r	audio frame size	56
s_{ni}	spatial score for the ith frame in C_n	71
t_{ni}	spatio-temporal score for the ith frame in C_n	71
m_a^{RMS}	mean of root mean square within W_a	69
m_a^{SC}	mean of spectral centroid within W_a	69
m_a^{SE}	mean of spectral entropy within W_a	69
$\overline{P}_k^0$	histogram descriptor corresponding to the kth cluster centre	80
s_a^{RMS}	standard deviation of root mean square within W_a	69
s_a^{SC}	standard deviation of spectral centroid within W_a	69
s_a^{SE}	standard deviation of spectral entropy within W_a	69
t_a^{RMS}	threshold for root mean square within W_a	69
t_a^{SC}	threshold for spectral centroid within W_a	69
t_a^{SE}	threshold for spectral entropy within W_a	69
C	database of unorganised and unsynchronised UGVs	24
E	set of events represented in C	24

u_s	speed of sound, i.e. 340 m/s	77
$\hat{}_{ij}$	distance matrix between chroma features F_i and F_j	56
bC_m	m^{th} video recording from bC	78
bC	subset of C	78
bE	set of events represented in bC	78
eC_k	representative video for the event cluster E_k	80
eC	set of representative videos for the set of event clusters E	81
$a^{RMS}(g_C)$	root mean square value at the $g_C{}^{th}$ frame	68
$a^{SC}(g_C)$	spectral centroid value at the $g_C{}^{th}$ frame	68
$a^{SE}(g_C)$	spectral entropy value at the $g_C{}^{th}$ frame	68
a_i^{st}	i^{th} audio sample in A^{st}	26
a_{nk}	k^{th} audio sample in A_n	25
$b^{RMS}(g_C)$	binary value from d^{RMS} at the $g_C{}^{th}$ frame	70
d_{ij}^{pq}	distance between chroma vectors v_i^p and v_j^q	56
$f(l)$	centre frequency at pitch class index l	34
f_c	cut-off frequency	61
f_i	maximum frequency of involuntary human body movement	60
f_s	standard frequency that corresponds to the pitch notation A4	34
h_p	overlap shift between two consecutive audio frames	56
h_{ii^0}	histogram for the count of the consistent time-shifts detected	81
l_s	pitch sub-band that corresponds to the standard frequency	34
l_{max}	maximum limit for the video-shot duration	70
l_{min}	minimum limit for the video-shot duration	70
$r(n)$	n^{th} index pointing to audio rank number	66
s^a	audio sampling rate	25
s^g	gyroscope data sampling rate	25
s^v	video frame rate	25
t	universal time	25
t_i^p	time at the p^{th} sample of the i^{th} video recording	25
u_j	time-stamp of j^{th} cut-point	26
v_r	r^{th} chroma number in v	34

v_{nk}	k^{th} visual frame in V_n	25
y_{ni}	gyro-based scores for the i^{th} inertial sample in C_n	91

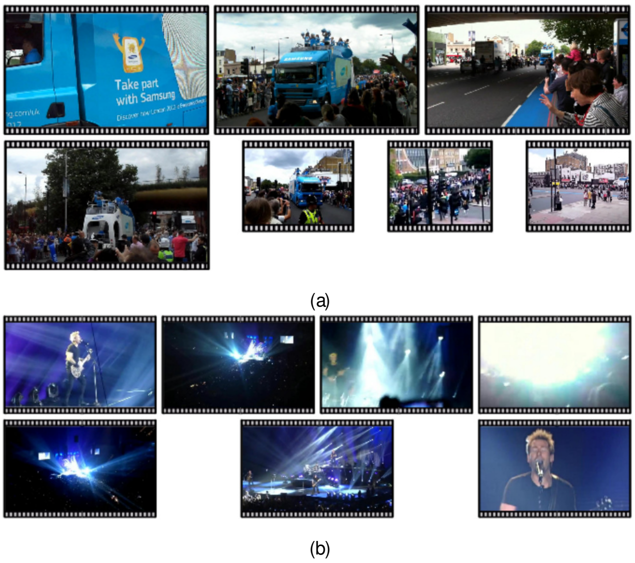
Chapter 1

Introduction

1.1 Motivation

Worldwide smartphone users are reported to be 1.63 billion at the end of year 2014 [100]. With the proliferation of smartphones, more people capture videos of their experience of attending events such as concerts, festivals, sporting competitions and public rallies, from different viewpoints. Social media sites then act as a distribution channel for the users to share their experiences by giving access to these User-Generated Videos (UGVs). 300 hours of video content is uploaded to YouTube every minute that is impossible to be watched by a person in a life span [5]. This has invoked a new research direction involving search and organisation of multimedia data of the same event [14, 111]. We define an event as a continuous action captured simultaneously by multiple user-devices from different positions located in proximity with each other. Multi-camera UGVs of the same event are unorganised due to different starting and ending times. Moreover, it is non-trivial to automatically retrieve UGVs of the same event from a database. The traditional metadata-based methods for event retrieval [62, 147], may not always be effective because meta-data associated with uploaded videos may lack consistent and objective tagging, or correct timestamps [39, 71]. By performing content-based event search, powerful event browsing can be enabled, which in turn can improve web search tools. The existing audio-based methods do not perform event retrieval from a database of multiple real-world events, and consider only the organisation of multi-camera UGVs recorded at the same concert or public event [20, 32, 76].

Multi-camera UGVs organised on a common timeline can be beneficial for the Region of In-



(a)

(b)

Figure 1.1: Synchronised frames from two different events, namely (a) Olympic torch rally and (b) Nickelback concert, recorded from 7 handheld cameras. Variation in the field of views, lighting conditions and video resolution can be observed.

terest (ROI) extraction, video composition and video summarisation [6, J2, 122, 133]. Moreover, smartphones nowadays are equipped with inertial sensors (accelerometer, gyroscope, magnetometer) whose data can be logged along with the video [1]. The inertial sensor data can substitute the visual data in developing camera motion detection and event understanding methods with an added advantage of reduced computational complexity [35, 36]. The existing inertial sensor-based method for camera motion detection utilises accelerometer and magnetometer data [35]. The performance of such method can be improved by using gyroscope data instead, as gyroscope directly gives a measure of the angular velocity of the smartphone [C1].

Professional recordings (e.g. film production) are staged and planned beforehand. On the other hand, UGV recordings are unplanned (i.e. not staged) and are dependent on the interest of the user holding the capturing device. They are often relatively short as the motivation is to record a surprising and interesting event. The visual quality of UGVs is influenced by the presence of visual degradations due to varying lighting conditions, changing field of views, unintentional camera motions and different video resolutions. The visual clues may not be similar across cameras recording in close proximity (Fig. 1.1), however, similarity exists in their audio signals.

Audio quality in UGVs is affected by the presence of audio degradations such as ambient noise (background noise), channel noise (low-level sound due to varying quality of microphones), reverberations (echo in the environment) and varying distance from the sound source [102, 116]. The recorded audio in UGVs can be generated from an amplified source (e.g. concerts) or from a non-amplified source (e.g. local gatherings and protests). In recordings that captured the non-amplified source of sound, usually the ambient noise is dominating while the audio clues of the sound source are weak.

Synchronisation involves spatio-temporal alignment of a set of UGVs of a particular event. Manual synchronisation is cumbersome and may not result in accurate alignment. Automatic synchronisation is hindered due to the presence of various audio and visual degradations. Synchronisation of UGVs using audio features is generally based on onsets (starting point of an audio instant) [132] or fingerprints (compact content-based audio signatures) [132, 76, 20]. However, onsets are sensitive to audio degradations and fingerprints may not be robust in the presence of reverberations [132]. The existing methods do not consider events containing the non-amplified source of sound, which may influence their performance [J1].

The multi-camera UGVs of the same event have limited fields of view, incomplete temporal coverage of the event, and may contain audio and visual degradations. Instead of recording several videos, the user tends to perform camera panning to cover its surroundings. These factors may influence their perceived quality making the content boring when playback individually. To enhance the viewing experience, video composition can be performed, that aims at generating a coherent and time continuous video from the synchronised multi-camera UGVs of the same event. The perceived audio-visual quality is a key factor which makes the content enjoyable and interesting to playback [15, 104]. The existing methods exploited visual content analysis for video composition from multi-camera UGVs [122, 133]. Global feature analysis is performed for understanding the content by attention detection [65], and for filtering the low-quality content by camera motion analysis [21, 65]. Although audio content plays an important part in the judgement of the overall perceived quality [15], it has not been utilised in the existing methods [122, 133].

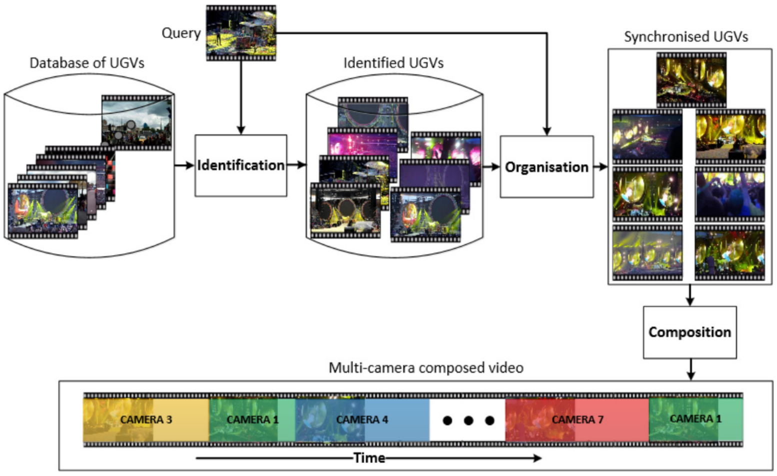


Figure 1.2: Multi-camera UGVs identification, synchronisation and composition. For a query, all UGVs belonging to the same event are identified from the database, that are organised (synchronised) on a common timeline. Video composition is then performed to produce a single continuous video.

1.2 Problem formulation

Let $C = f\, C_m g_{m=1}^{M}$ be a database of M unorganised and unsynchronised UGVs. We are interested in solving the following problems: clustering recordings corresponding to the same event, synchronising the clustered recordings on a common timeline, associating a new camera recording to an existing cluster, detecting camera motions in a recording and composing a single video from synchronised multi-camera UGVs of an event (Fig. 1.2).

1.2.1 Video event clustering

Let $E = f\, E_k g_{k=1}^{K}$ be the set of events represented in C, where K □ M. Each event $E_k = f\, C_{k;n_k} g_{n_k=1}^{N_k}$ contains N_k UGVs recorded from hand-held user-devices located in proximity and have at least partial temporal overlap with each other. Video event clustering aims to organise the database C into K clusters, such that each cluster k represents an event E_k.

1.2.2 Multi-camera synchronisation

Multi-camera synchronisation aims to temporally align the set of UGVs of an event E_k. Without loss of generality, let us consider two videos $C_{k;i}$ and $C_{k;j}$ of the same event E_k, and having the same frame rate. $C_{k,i}$ and $C_{k,j}$ are considered to be synchronised when the recording time t_i^p at the p^{th} frame of $C_{k,i}$ and t_j^q at the q^{th} frame of $C_{k,j}$ correspond to the same moment in the universal time t, an instant referring to the continuous physical time. Let the synchronisation time-shift Δt_{ij} be given by

$$\Delta t_{ij} = t_i^p - t_j^q. \tag{1.1}$$

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

1.2.3 Association of a new camera recording

The problem of associating a new video C_q to a cluster k involves identifying the set $E_k = \{C_{k,n_k}\}_{n_k=1}^{N_k}$ of UGVs that matches C_q.

1.2.4 Camera motion detection

Without loss of generality, let us now consider $C = \{C_n\}_{n=1}^{N}$ be the set of N synchronised and continuous multi-camera UGVs of an event. Let $V = \{V_n\}_{n=1}^{N}$, $A = \{A_n\}_{n=1}^{N}$ and $G = \{G_n\}_{n=1}^{N}$ denote N visual, audio and gyroscope data contained in C, respectively. Each V_n is given by

$$V_n = (v_{n1}, \ldots, v_{nk}, \ldots, v_{nK_n^v}), \tag{1.2}$$

where v_{nk} is the k^{th} visual frame, and is re-sampled to a common frame rate[1] s^v and contains K_n^v number of visual frames. Likewise, each A_n is given by

$$A_n = (a_{n1}, \ldots, a_{nk}, \ldots, a_{nK_n^a}), \tag{1.3}$$

where a_{nk} is the k^{th} audio sample, and is re-sampled to a common sampling rate s^a and contains K_n^a audio samples. Each $G_n = \{G_{nx}, G_{ny}, G_{nz}\}$ is sampled at s^g and contains K_n^g gyroscope data samples. Camera motion detection aims at detecting the unwanted pan P_{nd}, tilt T_{nd} and shake S_{nd} motions in C_n.

1.2.5 Video composition

Each C_n is temporally ordered on a common timeline, such that the first video frame corresponds to the first recorded frame in C and the last video frame, I_v, corresponds to the last video frame

[1] All UGVs are converted to the same frame rate using VirtualDub [86].

in C. Likewise for the audio A which goes from 1 to l_a and gyroscope data which goes from 1 to l_g. Thus, the coverage duration D_c (in seconds (s)) of the event is then given by

$$D_c = \frac{l_v}{s^v} = \frac{l_a}{s^a} = \frac{l_g}{s^g}. \tag{1.4}$$

Let the stitched audio A^{st} for the coverage duration D_c of the event be

$$A^{st} = (a_1^{st}, \cdots, a_i^{st}, \cdots a_{l_a}^{st}), \tag{1.5}$$

where a_i^{st} be the i^{th} audio sample. Let the suitable cut-points U be

$$U = (u_1, \cdots, u_j, \cdots, u_J), \tag{1.6}$$

where u_j is the time-stamp of the j^{th} cut-point and J is the number of segments. Let $S = \{S_n\}_{n=1}^{N}$ denote the spatial score, $T = \{T_n\}_{n=1}^{N}$ denote the spatio-temporal score and $Y = \{Y_n\}_{n=1}^{N}$ denote the gyro-based score for C, respectively. The problem of automatic video composition can be described as to select J segments from C to generate a single coherent video M, given by

$$M = (M_1, \ldots M_j, \ldots, M_J), \tag{1.7}$$

where each segment M_j belongs to one of the video recording C_n.

1.3 Contributions

The variations in the audio and visual qualities of UGVs make their identification, synchronisation and composition challenging. Composing a single multi-view video from multi-camera UGVs of the same event provides scene understanding, which can improve the viewing experience of the user. Camera motion is a key element of UGVs that effects the visual quality. The main contributions of this thesis are as follows:

1. We propose a framework for the automatic identification and alignment of unedited multi-camera UGVs within a database [J1]. We design a descriptor derived from the pairwise matching of audio chroma features of UGVs. The descriptor facilitates the definition of a classification threshold for automatic query-by-example event identification. The framework analyses the sound to match and cluster UGVs that capture the same spatio-temporal event and estimate their relative time-shift for synchronisation.
2. We propose a gyro-based camera motion detection method for UGVs captured from smartphones [C1]. The proposed method is independent of visual degradations due to the use of

gyroscope data. Video and gyroscope data are correlated as they are captured concurrently from the same device. To detect pan and tilt motions, we extract the dominant motions from the gyroscope, whereas shake is detected by analysing high frequencies in the gyroscope data.

3. We propose an automatic audio-visual camera selection framework for composing uninterrupted recordings from multiple UGVs of the same event [J2]. We develop an automatic audio-based cut-point selection method to segment the UGV. The proposed framework combines segments of UGVs using a rank-based camera selection strategy by considering audio-visual quality and view diversity. To filter video segments which contain visual degradations, we perform spatial and spatio-temporal assessment. Furthermore, we design a gyro-based score for quantifying the goodness of the UGVs, and use it to develop a gyro-based video composition framework.

4. We contribute a database of 263 multi-camera UGVs of 43 different concert events collected from the YouTube and 5 different self-captured events. These events are used for the validation of the proposed identification, synchronisation and video composition frameworks. For analysing the proposed gyro-based camera motion detection method, we captured 24 multi-modal (audio, visual and inertial data) recordings at various real-world scenarios using different smartphones. For validating the proposed gyro-based video composition framework, we captured multi-modal data of 4 events (12 UGVs) at a musical performance. To the best of our knowledge, similar multi-modal datasets are not available to the research community.

1.4 Organisation of the thesis

This thesis is organised as follows:

Chapter 1: The introduction and motivation for the thesis are described in Sec. 1.1, followed by the problem formulation in Sec. 1.2. The contributions of the thesis are discussed in Sec. 1.3.

Chapter 2: The introduction to the chapter is provided in Sec. 2.1. Related audio and visual content retrieval, and video composition applications are presented in Sec. 2.2, followed by an introduction to the features used for the content analysis of UGVs (Sec. 2.3). Inertial sensors are introduced in Sec. 2.4. The review of existing identification and synchronisation methods is presented in Sec. 2.5 and Sec. 2.6, respectively. This is followed by the state-of-the-art review of camera motion analysis (Sec. 2.7), and multi-camera video composition (Sec. 2.8). Finally, Sec. 2.9 summaries the chapter.

Chapter 3: Sec. 3.1 presents the introduction to the chapter. Details of the chroma feature that we utilise to design the event identification and synchronisation framework are presented in Sec. 3.2. The audio and visual analyses that are performed for the proposed video composition framework are presented in Sec. 3.4 and Sec. 3.5, respectively. Gyro-based analysis of UGVs for the proposed camera motion detection method is detailed in Sec. 3.3. The chapter is summarised in Sec. 3.6.

Chapter 4: The introduction to the chapter is provided in Sec. 4.1. An overview of the proposed framework is presented in Sec. 4.2. The proposed event identification framework is described in Sec. 4.3, followed by the details of time-shift estimation and cluster membership validation in Sec. 4.4. Sec. 4.6 provides the experimental analysis and comparison with the existing methods. The chapter is summarised in Sec. 4.7.

Chapter 5: The chapter is introduced in Sec. 5.1. The proposed audio and visual-based video composition framework is described in Sec. 5.2, and audio and gyro-based video composition framework is detailed in Sec. 5.3. Subjective test designed for the evaluation of the proposed frameworks is detailed in Sec. 5.4. Experimental evaluation of the gyro-based camera motion detection is presented in Sec. 5.5. This is followed by subjective evaluation and analysis of the proposed frameworks in Sec. 5.6. The chapter is summarised in Sec. 5.7.

Chapter 6: The chapter presents a summary of the achievements of the thesis (Sec. 6.1) and future directions of work (Sec. 6.2).

Appendix A: Details of the collected dataset for the analysis of the identification and synchronisation framework (Sec. A.1), gyro-based camera motion detection method (Sec. A.2) and video composition frameworks (Sec. A.3) are presented in this appendix.

Chapter 2

Related work

2.1 Introduction

Content identification within a database of multimedia recordings involves identifying all recordings that match in space and time with the query recording provided by the user [91]. Multi-camera synchronisation involves spatio-temporal alignment of a set of recordings of a particular event. Methods for content identification and synchronisation can be categorised into visual-based and audio-based. Visual-based methods identify Near-Duplicate Videos (NDVs) that contain the same visual content as that of the query from a database [28, 67, 91, 120, 136, 137]. Video-based synchronisation is performed on recordings captured in constraint environments [26, 45, 88, 93, 118, 149]. Audio-based methods utilise audio features to analyse UGVs for event identification [20, 32, 76] and synchronisation [20, 23, 32, 74, 76, 132].

Editing of synchronised UGVs of the same event can be performed to generate a single coherent multi-camera video. Camera motion analysis is a key component in designing video editing methods [21, 35, 60, 81]. The video editing process can be split into two main blocks, namely, audio and visual content analysis, and camera view selection. The selection of audio and visual features, and camera view are dependent on the application (editing of lecture, meeting room, home videos, sports videos or UGVs) [9, 42, 104, 117, 122, 133, 152, 156].

This chapter presents a review of the related work on content identification, synchronisation and video editing. First we present the content identification and video editing overview (Sec. 2.2), followed by an introduction of the features for content analysis (Sec. 2.3). An introduction to inertial sensors is then presented in Sec. 2.4. We review the existing visual-based (Sec. 2.5, Sec. 2.6) and audio-based (Sec. ??) methods for the content identification and synchronisation. We detail the related work for camera motion analysis (Sec. 2.7), and multi-camera video editing and composition (Sec. 2.8). Finally, summary and discussion are presented in Sec. 2.9.

2.2 Content identification and video composition overview

Content-based retrieval in multimedia recordings can be grouped into two main categories, i.e. to identify similar [64] or same [91] content as that of the query. Identification of similar content involves retrieving events which are similar but not necessarily occur at the same place and time (e.g. different parties, different sports games). Identification of same content involves retrieving events that occur at the same place and time (e.g. the same party, the same sport game). The focus of this thesis is on the latter category.

Videos that contain the same semantic information but differ in appearance (change in viewpoints, illumination, background, foreground) are termed as NDVs [91]. Identification of NDVs forms the basis for developing several applications such as copyright protection, usage monitoring, re-ranking and recommendation [91]. UGVs, as the name suggests, are the videos captured by people using their hand-held devices (e.g. smartphones). Identification and organisation of multi-camera videos is necessary for video summarisation [62], composition [122, 133, J2], shot detection [145], region of interest detection [35] and content analysis [6]. Identification in UGVs and NDVs is similar as both aim at retrieving the videos containing the same spatio-temporal information. However, they significantly differ due to the nature of the visual content under analysis. NDVs are transformed copies of an original professionally recorded and edited video (e.g. movies, music videos, television news). On the other hand, multi-camera UGVs recorded at the same event differ significantly due to varying fields of view, lighting conditions, camera motions, device settings and location of the users. These variations introduce visual degradations in UGVs making their content identification non-trivial.

Content identification in music involves matching professionally recorded music (e.g. album songs) against their database. It is used for copyright protection, usage monitoring, tagging, play-listing and taste profiling [18, 24, 27, 33]. Methods include those used for Shazam [143] and TrackID [2], which are based on the fingerprinting method by Wang [144] for audio identification. Some patents for audio identification and classification also came in recent years [78, 47, 95]. Audio from UGVs differ from professionally produced content (such as music albums, films) as it contains degradations due to device settings, user-handling and surrounding noise.

Video composition finds applications in lecture and meeting rooms recordings, sports games broadcast and highlights, home video summarisation and multi-camera UGVs composition [9, 42, 104, 122, 133, 152, 156]. In video summarisation, the continuity of the event is not considered 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. Home video refers to the single camera recording of a home event (wedding, birthday party). Multi-camera UGVs composition is closely linked with home video editing due to the similarity of the content, but differ in terms of the input information and target application, as in the case of video composition, multiple UGVs of an event are available for the generation of a continuous video. Unlike professional recordings, which are scripted and recorded from stable cameras, home videos and UGVs are recorded from hand-held devices and are dependent on the interest of the user.

2.3 Features

Audio features such as onsets [16, 126] and fingerprints [57, 144]) are utilised for the identification and synchronisation in UGVs. For UGVs' editing, global features are extracted from the visual data for the analysis of camera motion [6, 122, 133]. In this section, we present an introduction to the audio and visual features, which are commonly used for the content analysis of UGVs.

2.3.1 Audio onset

Onset is defined as the start of a transient region in an audio signal, during which spectral changes occur due to an increase in signal energy [16]. In the onset detection method [126], multiple frequency bands based on equivalent rectangular bandwidth (ERB) scale are first computed. ERB provides an approximation to the frequency bands in human hearing [126] An audio signal is divided into 8, 16 or 24 bands based on the ERB. A fixed or an adaptive threshold is applied on the ERB for onsets detection [23, 132]. In a fixed threshold-based approach [132], ERB is computed in each audio frame and a threshold is applied on the difference of energy between two consecutive frames to detect an onset. The threshold is selected heuristically based on a

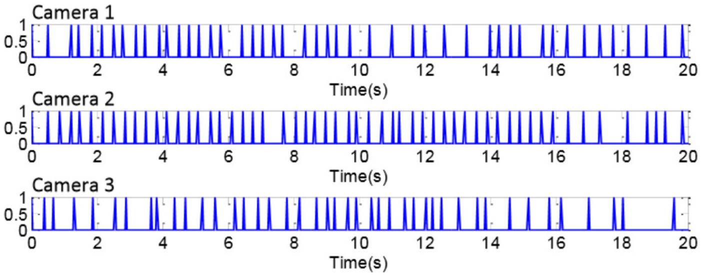


Figure 2.1: Onset [126] visualisation for the first band of three synchronised recordings. Onsets for C_3 appear less correlated with C_1 and C_2 due to the presence of audio degradations.

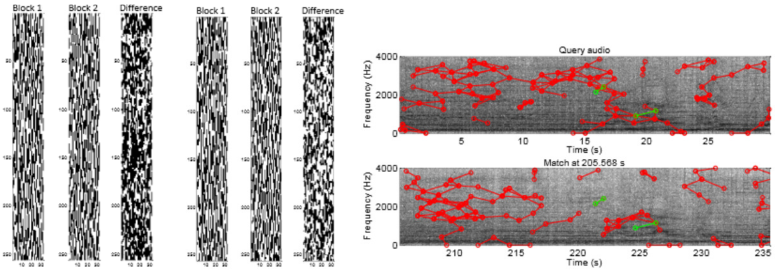
perceptual test [126]. In an adaptive threshold-based approach [23], an audio signal is divided into 8 bands, and a peak detector is applied at each band. The threshold for peak detection is set adaptively by relating it to the average audio energy at each band. At a particular time instant, an onset is detected if peaks are obtained in multiple bands. For visualisation, a band of extracted onsets for three synchronised recordings is shown in Fig. 2.1. It can be observed that C_1 and C_2 show high correlation, however, C_3 does not as it contains audio degradations.

For multi-camera UGVs synchronisation [132], cross-correlation of the multiple frequency bands of a pair of recordings is computed to estimate the time-shift for alignment. Detected onsets can also be integrated with the visual data for audio dependent visual event detection and synchronisation [23].

2.3.2 Audio fingerprint

Audio fingerprint [22] provides a condensed digital representation of an audio segment. It is used to identify audio signals from an audio database that are similar to the query and for audio-based synchronisation of UGVs [76, 132]. There are two key methods used for the extraction of audio fingerprints, namely frame-based [57] and landmark-based [144] methods.

In the frame-based method [57], the audio signal is first segmented into frames. A set of features, such as Fourier coefficients [52], Mel-Frequency Cepstral Coefficient (MFCC) [94], spectral flatness [34], and sharpness [34] are then computed for each frame. These features are mapped on a compact representation by performing quantisation [58], and are referred as sub-fingerprints. A collection of consecutive sub-fingerprints sufficient for audio identification is called fingerprint-block. Fingerprint-blocks from each pair of recordings are matched by com-



(a) (b) (c)

Figure 2.2: Visualisation of frame-based [57] and landmark-based [144] fingerprints. (a) Fingerprint-block of two synchronised recordings and their difference. (b) Fingerprint-block of two unsynchronised recordings and their difference. (c) Landmark-based fingerprints visualisation [46] in which a query is matched with the database recordings to identify and synchronise the same content.

puting Bit Error Rate (BER) for multi-camera UGVs synchronisation [132]. Figure 2.2(a) shows the fingerprint-block from two synchronised recordings and their difference. Fingerprint-blocks for unsynchronised recordings are shown in Fig. 2.2(b) for visualisation.

Landmark-based method [144] is proposed by Wang and is used widely for content identification [78, 95, 143]. In this method, the Short-Time Fourier Transform (STFT) of the audio segment is computed, and landmarks are identified as the spectrogram peaks. Landmarks are in areas of high energy. To extract the fingerprints, each landmark is associated with nine closet landmarks present in its target zone by using the time and frequency difference among them. This gives the hash value for each landmark. Figure 2.2(c) shows the visualisation of fingerprints matched between a query (of 30s duration) and database recording [46].

2.3.3 Audio chroma

Audio chroma feature is advantageous in distinguishing different types of sound, such as voice and musical instruments [12, 109]. This feature is mainly use in professional music recordings for the identification, chord recognition, genre classification, audio thumbnailing, matching and synchronisation [12, 48, 49, 72, 109].

Audio chroma gives a 12-dimensional representation of the tonal content of an audio signal derived by combining bands belonging to twelve pitch classes (C, C$^{\#}$, D, D$^{\#}$, E, F, F$^{\#}$, G, G$^{\#}$, A, A$^{\#}$, B) corresponding to the same distinct semitones (or chromas). The chroma feature vector is represented as $v = (v_0; ; \cdots; v_r; \cdots; v_{11}) \in \mathbb{R}^{12 \times 1}$, where v_0 corresponds to the energy of chroma C, v_1 corresponds to the energy of chroma C$^{\#}$, and so on. Each chroma v_r is computed as [110]

$$v_r = \sum_{s.t.\ l \pmod{12} = r} f(l); \tag{2.1}$$

where $r \in [0; 11]$ indicates the chroma number and l denotes the pitch class index corresponding to a particular spectrum bin index. The pitch class index l depends on their centre frequency $f(l)$ in a logarithmic way, and is given by [110]

$$l = V_d \log_2 \left(\frac{f(l)}{f_s} \right) + l_s; \tag{2.2}$$

where $f_s = 440$Hz is the standard frequency for pitch tuning [110] that corresponds to the concert pitch $l_s = 69$ (A4) and $V_d = 12$ which represents the 12 dimensions (semitones) of the chroma vector. Concert pitch is the reference pitch to which musical devices are tuned. A pitch class is the set of all pitches which share the same chroma. For instance, the pitch class corresponding to chroma C is $(C0; C1; C2; \ldots; C8)$ and relates to the pitch sub-bands $(12; 24; 36; \ldots; 108)$. This is represented using a chromagram. Figure 2.3 illustrates the process of extraction of the chroma feature for a particular audio frame f_r.

2.3.4 Luminance projection correlation

Luminance Projection Correlation (LPC) [112] is a visual method for computing the horizontal and vertical displacements of a camera. This method was introduced by Nagasaka and Miyatake [112, 141]), and is widely used for video content analysis and camera motion detection [21, 122, 133]. Further, it can be extended to detect shake motion [21], and is useful for analysing the spatio-temporal quality of the visual data [J2, 122, 133].

Given the video frame intensity $v(x; y; t)$ at time t, its horizontal $P_y(t; x)$ and vertical $P_x(t; y)$ projections are computed as [141]

$$P_y(t; x) = \frac{1}{h} \sum_{y=1}^{h} v(x; y; t); \tag{2.3}$$

$$P_x(t; y) = \frac{1}{w} \sum_{x=1}^{w} v(x; y; t); \tag{2.4}$$

where h is the height and w is the width of $v(x; y; t)$. The horizontal $L_x(t)$ and vertical $L_y(t)$ displacements at the time t are then calculated as

$$L_x(t) = \arg\min_{dp} \sum_{\substack{x = 1 + dp (dp \geq 0) \\ x = 1 (dp < 0)}}^{\substack{w (dp \geq 0) \\ w - dp (dp < 0)}} D_{P_y}(t; x; dp); \tag{2.5}$$

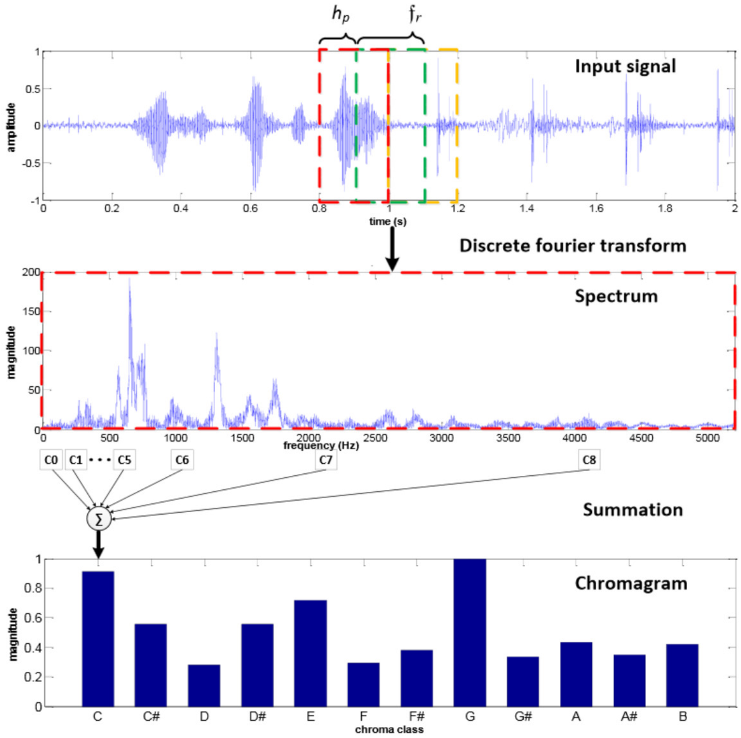


Figure 2.3: An example illustrating chroma feature extraction. The spectrum of a particular audio frame f_r (highlighted in red) is divided into sub-bands and a chromagram is formed by summing all pitch bands corresponding to a particular chroma.

$$L_y(t) = \arg\min_{dp} \sum_{\substack{y=1+dp\,(dp\geq 0)\\ y=1\,(dp<0)}}^{\substack{h\,(dp\geq 0)\\ h-dp\,(dp<0)}} D_{P_x}(t; y; dp); \tag{2.6}$$

where D_{P_x} and D_{P_y} are the projection distances computed as

$$D_{P_y}(t; x; dp) = \{P_y(t; x) - P_y(t+1; x-dp)\}^2;$$
$$D_{P_x}(t; y; dp) = \{P_x(t; y) - P_x(t+1; y-dp)\}^2;$$

and dp is a panning parameter ranging from -20 to 20 pixels displacement. The horizontal $L_x(t)$ and vertical $L_x(t)$ displacements are also referred as camera pan (left-right) and tilt (up-down) motions, respectively.

2.3.5 Optical flow

Optical flow is the apparent motion in an image caused due to the movement of a camera or object in the scene. It finds applications in motion estimation, action recognition, video indexing and retrieval, crowd motion and pedestrian behaviour analysis, image sequence compression (MPEG), robotics (obstacle detection, time to contact) [51].

Optical flow assumes that the brightness of a physical point in the image does not change over the time. If an image $v(x;y;t)$ is displaced by dx and dy between two frames dt, than the brightness constancy constraint can be given by

$$v(x;y;t) = v(x+dx;y+dy;t+dt): \tag{2.7}$$

Assuming the displacement to be very small, we get

$$v(x+dx;y+dy;t+dt) = v(x;y;t) + \frac{dv}{dx}dx + \frac{dv}{dy}dy + \frac{dv}{dt}dt; \tag{2.8}$$

$$\frac{dv}{dx}\frac{dx}{dt} + \frac{dv}{dy}\frac{dy}{dt} + \frac{dv}{dt}\frac{dt}{dt} = 0; \tag{2.9}$$

that gives

$$\frac{dv}{dx}V_x^{OF} + \frac{dv}{dy}V_y^{OF} + \frac{dv}{dt} = 0; \tag{2.10}$$

where V_x^{OF} and V_y^{OF} are the optical flow x and y components. Eq. 2.10 has two unknowns and cannot be solved without additional constraints. Several optical flow estimation methods have been introduced that impose additional constraints for computing the flow [13], among which differential methods are the most common ones [63, 97]. Horn and Schunck [63] proposed a global method that assumed the optical flow to be smooth over the entire image. While Lucade and Kanade [97] proposed a local method by assuming the optical flow to be constant on the current feature point neighborhood.

2.4 Inertial sensors

Smartphones are equipped with sensors, such as accelerometer, compass, gyroscope, GPS, proximity detector, microphone, and camera, which are providing new directions toward the development of sensing applications [82]. These sensors when logged while capturing an image or recording a video are of significant importance for geo-tagging, localisation and video annotation [130, 90]. Additionally, inertial sensors (i.e. accelerometers, gyroscopes, magnetometers) are useful for camera motion analysis and can provide better understanding of the environment and event of interest in UGVs [35].

2.4.1 Accelerometer

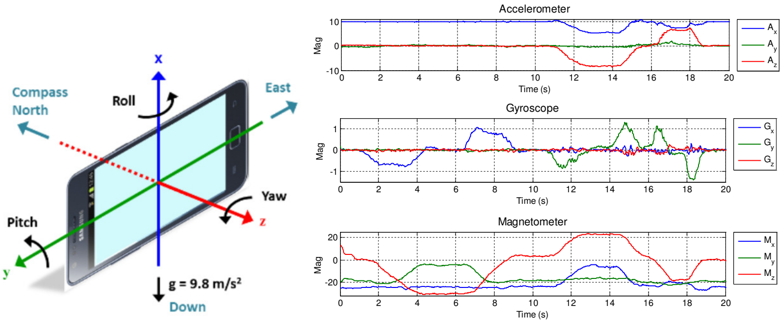
Tri-axial accelerometer measures the proper acceleration (in x, y and z axes) experienced relative to the free fall by a device. The gravitational component of the acceleration ($g = 9.8m/s^2$) is always present in the proper acceleration, such that an accelerometer at rest on the surface of the earth measures 1g in the upward direction [92]. Integral of measured acceleration gives the velocity and double integral gives the displacement of the device. However, the double integral introduces an accumulated position drift [17].

Accelerometer was initially included in smartphones for detecting its rotation by observing the switching of the gravitational component from one axis to another [92]. The motivation was to enhance the viewing experience of the user by rotating the display according to the orientation of the device [82]. Accelerometer data in smartphone is found to be beneficial for developing real-time activity recognition applications for fitness, sports and health monitoring by inferring different activities (e.g. walk, jog, run, sit, stand) [68, 107, 114]. It is also used for detecting driver's behaviour while driving by understanding vehicle's motion [92], and for tracking phone gestures for virtual hand-writing experience [92]. Furthermore, accelerometer data is also used for computing the tilt angle of the device [35]. However, the same tilt angle can be obtain directly from the orientation sensor (i.e. an internal software-type sensor).

2.4.2 Gyroscope

Tri-axial gyroscope is an angular speed sensor which measures the rate of rotation (angular velocity) around their own x, y and z axes. The rate of rotation is given in rad/s units. The rotations around the x, y and z axes are termed as roll, pitch and yaw, respectively (as shown in Fig. 2.4(a)). Unlike accelerometer and magnetometer, gyroscope is neither effected by gravity nor magnetic field. Gyroscope is not influenced by environmental conditions, which makes it useful for navigation in space where magnetometer does not works (e.g. hubble space telescope). This is useful for maintaining the orientation of a device, and is therefore utilised for the stability in navigation of unmanned aerial vehicle, aircraft, helicopter and large boat [10].

Gyroscope is more sensitive, precise and robust compared to an accelerometer. In smartphones, it is used for the development of 3D dynamic games [125]. Gyroscope is also sensitive to acoustic signals in the close proximity of a smartphone, that is exploited for speech recognition [105]. Gyroscope combined with accelerometer provides more accurate orientation and



(a) (b)

Figure 2.4: Smartphone inertial sensors. (a) Visualisation of device's and earth's coordinate systems, and rotation around x, y and z axes. (b) The response of accelerometer, gyroscope and magnetometer when roll and pitch motions are performed between time 1s to 9s and 11s to 19s, respectively. Roll motion is observed by gyroscope and magnetometer but not by accelerometer as the gravitational component of the device remains unaffected.

motion-sensing information, as their fusion facilitates in compensating for the angle and displacement drifts. It is therefore used in conjugation with an accelerometer for applications such as activity recognition, indoor navigation and tracking, mobile security, etc [68, 92, 160]. It can also be exploited for camera motion analysis (i.e. pan and tilt detection) [C1].

2.4.3 Magnetometer

Tri-axial magnetometer measures the orientation of a device with respect to the Earth's magnetic field. An accelerometer and gyroscope measure the relative displacement and rotation with respect to the device's coordinate system. A magnetometer is used to obtain the absolute orientation of the device with respect to the earth coordinate system. A visualisation of device and earth coordinate systems are shown in Fig. 2.4(a). Magnetometer is sensitive to drift and magnetic field induced by the presence of nearby magnetic objects.

In smartphones, magnetometer is mainly used to complement accelerometer and gyroscope information [11] to compensate for the drift error. It is sometimes used in conjugation with an accelerometer and/or gyroscope for activity recognition, localisation and navigation [30, 68, 85]. It can also be used for estimating camera pan movement in UGVs [35]; but the obtained estimation may not be reliable due to electromagnetic noise.

2.4.4 Inertial sensor-based features

Data acquired from the inertial sensors can be further processed for extracting different time and frequency domain features [84]. Most commonly utilised time domain features include mean, standard deviation, mean absolute deviation, minimum and maximum, energy, entropy. Frequency domain features include frequency spectrum skewness, kurtosis and spectral energy, and energy of different frequency bands. In human activity recognition applications, these features are extracted from the accelerometer and gyroscope data, and are then used for training the classifiers (e.g. Support Vector Machine (SVM), neural network, decision tree) for recognising different activities [84, 114, 123].

The response of accelerometer, gyroscope and magnetometer when roll and pitch motions are performed is shown in Fig. 2.4(b). Gyroscope data gives a direct estimate of pan (roll) and tilt (pitch) motions as it measures the angular velocity of the smartphone.

2.5 Identification

Content analysis of UGVs is mainly performed using audio fingerprinting [20, 32, 76, 132]. Near-Duplicate Video Retrieval (NDVR) can be related to the identification of UGVs as both aim at determining whether the database videos contain the same content as that of the query video. NDVR utilises spatial and temporal features (such as appearance, texture, temporal dynamics) for video matching [91, 137, 136, 43, 120, 67, 28]. A general framework for NDVR first represents a video as a set of descriptors extracted from each frame or keyframes [43, 136, 137]. A video signature is then formed that represents a video at local or global level [91]. At local level, each keyframe forms a signature, while at global level, each video forms a single signature. The query video signature is matched with the signatures of the database videos to compute the similarity. Temporal constraints are applied on the matched signatures by weak alignment [155] or Hough voting [43] for NDVR. A review of existing video signature methods is presented by Paschalakis et al. [115]. Audio-based identification mainly involves feature extraction and feature matching. Feature matching is performed by computing pairwise cross-correlation or hash-value similarity of the extracted features [20, 32, 76]. Detailed below is the state-of-the-art for NDVR and audio-based identification.

2.5.1 Local signature-based identification

Local signature-based methods [43, 28, 159, 153] are computationally expensive as compared to global signature-based methods because all keyframes' signatures of the query are compared with all keyframes' signatures of the database followed by temporal verification. Douze et al. [43] used hessian detector and Center-Symmetric Local Binary Pattern (CSLBP) as descriptor to design a compact signature for each frame, and applied modified Hough voting for the retrieval. Chou et al. [28] used Features from Accelerated Segment Test (FAST) detector and Histograms of Orientations of Optical Flow (HOOP) descriptor for spatio-temporal feature extraction from keyframes, followed by encoding them into symbols. A pattern-based prefix tree is constructed offline from the symbols, which facilitated the query search in a constant time. Zhou et al. [159] used Principal Component Analysis-based Scale-Invariant Feature Transform (PCA-SIFT) for feature extraction and constructed an adaptive structure video tensor series. A dimensionality reduction method is designed, and an efficient distance function is proposed to measure the similarity between the query and database tensor series. Wu and Aizawa [153] used Conditional Entropy (CE) and Local Binary Pattern (LBP) to construct the Self-Similarity Belt (SSBelt), which gave the local signature of a video.

2.5.2 Global signature-based identification

Global signature-based methods [136, 137, 67, 120] can perform video identification in real-time, however, they may become less effective in representing long duration videos. Song et al. [136, 137] used multiple features hashing to learn the hash codes and hash functions of the training data. Hash functions facilitated inferring the hash codes of videos that were not included in the training data. Hamming distance was then computed to obtain the similarity between each pair of videos. Huang et al. [67] performed NDVs clustering and used histogram intersection of all pairs of training videos for the adaptive classification of videos. Revaud et al. [120] proposed an event retrieval framework from large video collections and contributed the EVent VidEo (EVVE) database. This method [120] jointly encoded the spatial and temporal information of a video in frequency domain to get the signature for the query video. Match score was computed between pair of video by component-wise matching of their signatures in the frequency domain.

Table 2.1 summarises the state of the art for NDVR. These methods are mainly designed for copy detection in professional videos that have no or narrow view-point change, and are captured

Table 2.1: State of the art for Near Duplicate Video Retrieval (NDVR). Key: LS - local signature; GS - global signature; LBP - local binary pattern; SIFT - scale-invariant feature transform; CSLBP - centre-symmetric LBP; FAST - features from accelerated segment test; HOOF - histograms of orientations of optical flow; DoG - difference of Gaussian; PCA-SIFT - principal component analysis-based SIFT; HSV - hue satuation value.

Ref.	Signature type		Features	NDV database used	Comments
	LS	GS			
[43]	x		hessian, CSLBP	TRECVID2008[1], self-collected data	compact representation and modified hough voting
[28]	x		FAST, HOOF	MUSCLE_VCD[2], CC_WEB_VIDEO, UQ_VIDEO	symbols and pattern-based prefix tree from spatio-temporal feature
[159]	x		DoG, PCA-SIFT	TV boardcast, TREVID2008, CC_WEB_VIDEO	adaptive structure tensor series for spatio-temporal features encoding
[153]	x		CE, LBP	CC_WEB_VIDEO, MUSCLE_VCD, TREVID2008	self-similarity belt as signature and intensity mark for alignment
[136, 137]		x	HSV, LBP	CC_WEB_VIDEO[3], UQ_VIDEO[4]	multiple feature hashing and real-time implementation
[120]		x	dense SIFT	TREVID2008, CC_WEB_VIDEO, EVVE[5]	joint representation of appearance and temporal information
[67]		x	histogram intersection	CC_WEB_VIDEO	adaptive classification and integrated voting for clustering

mainly using fixed cameras.

2.5.3 Audio-based identification

Event identification using audio features has been addressed in [20, 32, 76], which used landmark-based audio fingerprinting [144], where the landmarks are the onsets of local frequency peaks and are identified from the STFT of the audio (see Sec. 2.3.2). Kennedy and Naaman [76] presented an approach for the synchronisation and organisation of a collection of recordings from three concerts, in which the classification threshold was computed based on the mean and standard deviation of the matches. For each set of concert recordings, synchronisation time-shift was obtained to align the recordings on a common timeline. Cotton and Ellis [32] used matching pursuit to obtain a prominent representation of audio events and tested their identification approach on a public speech dataset that contained multiple recordings of the same event. Both

[1] http://www-nlpir.nist.gov/projects/tv2008/tv2008.html
[2] https://www.rocq.inria.fr/imedia/civr-bench/data.html
[3] http://vireo.cs.cityu.edu.hk/webvideo/
[4] http://itee.uq.edu.au/shenht/UQ_VIDEO/ (link broken)
[5] http://pascal.inrialpes.fr/data/evve/

Table 2.2: State of the art methods for identification of multi-camera UGVs. Key: AF: audio fingerprint; AC : audio chroma; ILD: insensitive to local degradations; IGD: insensitive to global degradations; K: total number of events; M: total number of recordings; PP: professional production recordings; AS: amplified sound recordings; NAS: non-amplified sound recordings.

Ref.	Feature		Properties		Matching approach	Dataset Properties				
	AF	AC	ILD	IGD		K	M	PP	AS	NAS
[76]	x			x	Hash-value similarity maximisation	3	608		x	
[20]	x			x	Cross-correlation maximisation	9	203	x	x	
[32]	x			x	Hash-value similarity maximisation	1	733		x	
[J1]		x	x	x	Feature similarity maximisation	48	263		x	x

approaches [76, 32] used hash value similarity maximisation for matching pairs of recordings. A similar approach was presented by Bryan et al. [20] for event identification and synchronisation. This method used landmark cross-correlation for matching and a fixed classification threshold to cluster a speech dataset of 180 professional recordings and 23 UGVs of concerts.

Table 2.2 summarises the state of the art for identification of multi-camera UGVs using audio features. We categorise audio degradations into two groups, namely, local and global degradations. Local degradations are caused by recording device settings, channel and surrounding noise, and reverberations. Global degradations are common to some or all recording devices (e.g. a crowd cheering, a whistle blowing during a specific event) and may help during the synchronisation process. The existing methods mainly considered amplified sound recordings and showed robustness only to local degradations.

2.6 Synchronisation

Several methods for visual-based multi-camera synchronisation are proposed in the literature, which exploit local or global features to achieve synchronisation [25, 88, 138, 149]. Most of the existing methods made use of the multi-view geometry between the stationary cameras and the object been recorded [25, 26, 138]. Synchronisation of UGVs is performed using audio fingerprinting, audio onset, audio feature-based classification and audio-visual event-based methods [23, 132, 134]. Presented below is the related work for visual and audio-based synchronisation methods.

2.6.1 Local feature-based synchronisation

Feature-based synchronisation methods exploit tracking of objects or feature points between each pair of static cameras [26, 45, 93, 118, 138, 148, 161] or across three cameras [88, 150], or interest points detection in space and time [149, 154] for estimating the synchronisation time-shift.

Trajectory-based methods extract the trajectory of objects or interest points by using background subtraction or feature trackers [29, 97]. These methods assume that the moving objects are captured from a pair of stationary cameras [45, 93, 138, 148], with the exception of few which also consider jointly moving cameras [25, 26]. Once the trajectories are computed, these methods solve the homography (2D projective transformation) [25, 26, 138] or epipolar geometry [45, 148] between the two cameras for estimating the synchronisation time-shifts. In multi-view geometry, fundamental matrix [59] relates the 3D scene points with their projections in the 2D camera images. Homography [59] is a special case of fundamental matrix which assumes that the distance between two cameras (C_1 and C_2) is negligible compared to their distances from the scene. The transformation required to map the 2D image from C_2 to the 2D image in C_1 is termed as homography. Homography from C_2 to C_1 is computed using the trajectories of single/multiple interest points or objects. The time-shift is estimated by minimising the sum of squared differences [25, 26] or by optimising the RANdom SAmple Consensus (RANSAC) algorithm between the original and transformed trajectories. Other methods exploit the kinetic changes of moving objects captured against stationary background to estimate the synchronisation time-shift [148, 93]. Zini et al. [161] computed the frame-level correspondences between two camera recordings using the actions of articulated objects. The method assumed objects' association to be known a priori and computed Histogram of Oriented Gradients (HOG) to identify and synchronise the repeated pattern that exists in actions of articulated objects.

Tri-focal tensor [59] is used for the synchronisation of multi-camera recordings of a scene captured using three stationary cameras [88, 150]. Tri-focal tensor is a generalisation of the fundamental matrix that relates the features across three views instead of two [59]. Lei and Yang [88] computed the correspondence of trajectories across three camera views by maximising the feature geometric alignment measure. Instead of solving the trajectory correspondence, Whitehead et al. [150] identified the inflection points (change in a trajectory's direction) across three cameras for computing the synchronisation time-shift. The main constraint of [88] and [150] was that the three cameras should remain stationary throughout the recording. Additionally, Whitehead et al. [150] assumed that the motion of objects in the scene to be non-periodic.

Space-Time Interest Points (STIP) are also used for the synchronisation of a pair of video recordings captured from stationary cameras [154, 149]. STIPs [83] are derived from spatial interest points (Harris corner detector) to detect an interest point both in space and time. Yan and Pollefeys [154] correlated the histograms of STIPs' distribution of a pair of video recordings to obtain the synchronisation time-shift. Wedge et al. [149] applied RANSAC-based temporal model and homography or fundamental matrix-based spatial model on STIPs of a pair of videos to estimate the time-shift.

2.6.2 Global feature-based synchronisation

Direct alignment methods are based on image intensity [25, 142] or luminance changes [132, 134]. These methods do not require feature detection and object or interest point tracking. Instead, they rely on spatio-temporal variations in pixel intensities (e.g. fireworks, camera flashes) for the alignment.

Caspi and Irani [25] proposed a direct alignment method to synchronise a pair of videos captured using stationary calibrated cameras. A Gaussian spatio-temporal pyramid is computed, and an iterative algorithm is applied to minimise the sum of squared differences in pixel intensities at each level of the pyramid using the estimate of the spatio-temporal model. Likewise, Ukrainitz and Irani [142] maximised the space-time correlation of local pixel variations of a pair of videos to estimate the synchronisation offset.

Shrestha et al. [132, 134] proposed a global brightness variation (flashes) based method for multi-camera synchronisation. Flash enabled cameras produce instantaneous flashes of light which illuminate the scene [132]. The luminance histogram of a frame containing flash shows concentration of pixels in the higher bins of the histogram. To detect flashes, luminance difference curve was computed by taking the difference of accumulated high brightness pixels (range 171 to 255) across consecutive frames. A locally adaptive threshold was applied on the luminance difference curve to detect flashes. The flash patterns of a pair of videos were matched to determine the synchronisation time-shift. Flashes are used only in indoor or night events and are not always captured in videos due to camera shutter closure or field of view variation.

Table 2.3 summarises the state of the art for visual-based multi-camera synchronisation. Most of these methods have constraints on the number of cameras, rigidity of cameras and field of view.

Table 2.3: State-of-the-art of visual-based multi-camera synchronisation. Key: NRC: non-rigid camera; UE: unconstrained environment; N: Number of cameras to be synchronised; '*' indicates that atleast three cameras are fixed; The letters a, b in Ref. indicates different methods proposed in the same paper.

Ref.	Feature type		Feature used	Camera constraints		
	Local	Global		NRC	UE	N
[138]	X		trajectory			2
[25]a	X		trajectory			2
[148]	X		trajectory			2
[88]	X		trajectory	X □		3
[150]	X		trajectory	X □		3
[118]	X		trajectory			2
[45]	X		trajectory			2
[93]	X		trajectory			2
[154]	X		space-time interest point			2
[149]	X		space-time interest point			2
[161]	X		histogram of oriented gradient			2
[132]		X	flashes	X	X	> 2
[25]b		X	pixel intensity			2
[142]		X	pixel intensity			2

2.6.3 Audio-based synchronisation

Existing methods for multi-camera UGV synchronisation involve extraction and matching of features (audio fingerprints [132, 76, 20] and audio onsets [132]), audio feature-based classification [131], and audio-visual events [23]. Also, Kammerl et al. [74] proposed graph-based methods for temporal synchronisation inferred by analysing the consistency in pairwise cross-correlation of three audio features, namely, spectral flatness, zero-crossing and signal energy.

The audio fingerprinting method of Haitsma and Kalker [57] was exploited by Shrestha et al. [131, 132]: a 32-bit sub-fingerprint (binary) was generated based on spectrum-temporal analysis of the audio in an overlapping window (see Sec. 2.3.2). Two fingerprint-blocks of 256 consecutive sub-fingerprints were considered to be matching if the number of error bits (BER) was smaller than a threshold [57]. The landmark-based fingerprinting approach by Wang [144] was used by Kennedy and Naaman [76] and Bryan et al. [20] for the synchronisation of collections of concert recordings. The same approach [144] was used by Duong and Thudor [44] for the synchronisation and identification of removed and re-ordered segments of movies. Audio fingerprinting-based methods are commonly used due to their robustness to audio degradations. However, they might become sensitive to reverberations [132] and strong local degradations [102].

An onset-based method was presented by Shrestha et al. [132], which performed cross-correlation of multiple frequency bands of two recordings to compute their synchronisation time-shift (see Sec. 2.3.1). In comparison to audio fingerprints, onset-based methods [132] are more sensitive to audio degradations as false positive onsets may get detected due to channel and background noise.

An audio feature classification method for multi-camera synchronisation was presented in [131], which was based on low-level signal properties, i.e. MFCC, roughness, loudness, sharpness and temporal envelope fluctuations model. Quadratic discriminant analysis [103] was performed to estimate the probabilities of silence, music, speech, noise and crowd classes for each audio frame of size 11:6ms. Cross-correlation was then computed to estimate the time-shift.

2.6.4 Audio-visual synchronisation

Casanovas and Cavallaro [23] presented an audio-visual events-based method for multi-camera synchronisation in which an audio-visual event was defined to be a simultaneous change in the audio and video streams. The method first detected an audio event using audio onsets (see Sec. 2.3.1). A space-time visual block was then defined around each detected audio event, and the local variation of pixel intensities were analysed in each block for visual event detection. A space-time block was considered to be active if its local variation was greater than a threshold, and an audio-visual event was detected when several active blocks were in close proximity. This method is sensitive to audio degradations, in the same way as the onset-based method [132] is. Additionally, it is dependent on camera motion, and near or far fields of view. Table 2.4 summarises the state of the art for synchronisation of multi-camera UGVs using audio and audio-visual features.

2.6.5 Audio chroma for music alignment

Chroma features are mainly use in professional music recordings for the identification, chord recognition, genre classification, audio thumbnailing, matching and synchronisation [12, 48, 49, 72, 109, 98]. Muller et al. [109] presented an audio matching approach using Chroma Energy

Table 2.4: State of the art methods for synchronisation of multi-camera UGVs. Key: AFC: audio feature classification; AF: audio fingerprint; AO: audio onset; PI: pixel intensity; AC : audio chroma; ILD: insensitive to local degradations; IGD: insensitive to global degradations; BER: Bit Error Rate; K: total number of events; M: total number of recordings; PP: professional production recordings; AS: amplified sound recordings; NAS: non-amplified sound recordings. The letters a and b in Ref. indicate different methods proposed in the same paper.

Ref.	Feature					Properties		Matching approach	Dataset Properties				
	AFC	AF	AO	PI	AC	ILD	IGD		K	M	PP	AS	NAS
[131]	X							Cross-correlation maximisation	5	11		X	
[132]a			X					Cross-correlation maximisation	7	30		X	
[132]b		X					X	BER minimisation	7	30		X	
[76]		X					X	Hash-value similarity maximisation	3	608		X	
[44]		X						Hash-value similarity maximisation	11	264	X	X	
[20]		X					X	Cross-correlation maximisation	9	203	X	X	
[23]			X	X				Cross-correlation maximisation	8	40		X	X
[J1]					X	X	X	Feature similarity maximisation	48	263		X	X

distribution Normalised Statistics (CENS), which is a variant of chroma features. In this method, either the number of matches to be retrieved or the threshold value for the distance of a retrieved match need to be pre-defined. Ewert et al. [49] proposed a method of score-to-audio alignment in music that combines chroma with onset features and performs matching using Dynamic Time Warping (DTW). The testing is performed on noise-free synthesised music files. Macrae et al. [98] extracted chroma features from an input music and the corresponding streaming music video, and used DTW for their real-time synchronisation.

2.7 Camera motion analysis

Camera motion analysis is performed for detecting shot boundaries and unwanted camera movements in home videos and UGVs [6, 21, 104, 141]. The majority of methods for camera motion analysis utilise visual content by template matching, optical flow and Luminance Projection Correlation (LPC) [21, 60, 81]. To the best of our knowledge, there exists only one method that used inertial sensors to detect camera motion [35]. Table 2.5 lists the state of the art methods for camera motion detection.

2.7.1 Visual-based methods

Template matching-based methods [60, 81] are used for the video-shot classification in cinematographic, home and sports videos. Template matching involves dividing a frame into smaller

Table 2.5: State of the art for camera motion detection. Key: VM - visual method; IS - inertial sensor; TM - template matching; OF - optical flow; LPC - luminance projection correlation; A - accelerometer; M - compass; G - gyroscope; CMDG - proposed method.

Ref.		[60]	[81]	[101]	[7]	[99]	[112]	[141]	[80]	[35]	[C1]
VM	TM	X	X			X					
	OF			X	X	X					
	LPC						X	X	X		
IS	A									X	
	M									X	
	G										X

blocks and matching each block across consecutive frames to estimate object or camera motion. Hassan et al. [60] applied block matching between two consecutive frames to obtain motion vector field. Each block was then processed to give a camera motion histogram descriptor. Lan et al. [81] used template matching with full search to estimate horizontal, vertical and radial background camera motions. This method was used by [6] for UGV analysis. Lan et al. [81] labelled camera motion into zoom, fast motion, shake and stable based on a decision tree structure. Features , namely average velocity, average acceleration, variance of acceleration, and average number of direction changes in vertical and horizontal directions were extracted for training the SVM classifiers for camera motion detection.

Optical flow-based methods are commonly used for camera motion analysis [7, 87, 101, 113]. Optical flow measures the relative velocity of objects in the scene with respect to the camera. Most of the existing methods used magnitude and orientation of the optical flow for detecting and classifying the motion [87, 101, 113]. In [101], the matched feature points were obtained from the optical flow of two consecutive frames, that were then used to detect change points by applying a threshold. A frame was considered to contain significant motion if the percentage of change points was greater than a threshold. To detect pan, orientations of the optical flow vectors were calculated between two consecutive frames. An eight bin orientation histogram was then constructed, that detected the dominant motion with respect to a threshold. In [7], linear combination of optical flow models for pan, tilt, zoom and roll was utilised for camera motion estimation. In [87, 113], optical flow combined with template matching was also used for camera motion analysis. A frame was divided into four sub-regions, each occupying one of the four edges of the frame. Templates were build based on these sub-regions such that each template represented one of the six camera motions, namely, pan left, pan right, tilt up, tilt down, zoom in and zoom out. The magnitude of optical flow vectors within each sub-region was used to detect motion with respect to a pre-defined threshold. The optical flow vectors of the motion detected frame were matched against each template to classify it into one of the six motions.

LPC is exploited for the camera motion estimation in home videos and UGVs [21, 122, 133]. LPC computes the pan and tilt motions by projecting a frame on the horizontal and vertical axes, and correlating the projections of two consecutive frames (see Sec. 2.3.4). Campanella et al. [21] used LPC for shake detection, by computing the normalised differences of the detected pan and tilt with the filtered pan and tilt signals. It is also used for camera-work judgement for the video shooting navigation (stable, pan, tilt, zoom) [141, 80].

Visual-based method are computationally expensive as compared to inertial sensor-based methods. These methods may get influenced by moving objects present in the frame and low brightness (e.g. recording fireworks at night). Either a threshold is applied to extract dominant motion in a frame or a template is used to suppress the effect of objects' motion.

2.7.2 Inertial sensor-based methods

Cricri et al. [35] proposed the only inertial sensor-based method, with application to event understanding in UGVs. This method detected pan by calculating angular speed of the camera from the low-pass filtered compass data (sampled at 10Hz). Tilt angle acquired from the unfiltered accelerometer data (sampled at 40Hz) was differentiated for tilt detection. Shake was computed from the high-pass filtered accelerometer data. An inertial sensor-based method reduces the computational cost due to the reduction in the amount of data to be processed. However, compass is sensitive to drifts and errors induced by nearby magnetic objects. Furthermore, the unfiltered accelerometer data contains noise which may reduce the accuracy of [35].

2.8 Video composition

Video composition is used for camera selection in lecture webcast [40, 152] and meetings [117, 156], sports video broadcast [38, 42, 145], home-video summarisation [21, 65] and multi-camera mashup generation [122, 133]. Video editing and composition frameworks can be split into two main blocks, namely, content analysis and camera view selection. Content analysis involves the extraction of audio and visual features mainly for scoring the recording content. These scores are then utilised for camera view selection. In this section, we present the state-of-the-art with respect to these two blocks. Table 2.6 summaries the current state-of-the-art methods for multi-camera editing and composition along with the scenario for which they are designed.

2.8.1 Audio and visual content analysis

The type of features extracted for vision-based sports game analysis vary based on the application and may include dominant colour, colour histogram, camera motion, corner points, ball and player detection, field characteristics detection, texture and player recognition [42]. Multi-camera sports videos are recorded using fixed professional cameras capable of performing pan, tilt and zoom [42]. For the automatic boardcast of sports video, Wang et al. [145] computed features like field line, goalmouth, centre circle, ball trajectory, camera motion and audio keyword, and used them for event moment detection by training a SVM for three event classes, namely, attack, foul and miscellaneous. In [38], features like amount of activity, objects' trajectory, size and location are used for the designing of a video composition framework, which was tested on multi-camera basketball game, airport surveillance and outdoor videos datasets.

Table 2.6: State of the art for multi-camera editing and composition. Key: AQ: audio quality; AC: audio continuity; VQ: visual quality; CM: camera motion analysis; VD: view diversity; K: Number of events tested (mentioned only for UGVs); 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.

Ref.	Type	AQ	AC	VQ	CM	VD	Features used	Camera selection method	Data type	K	Comments
[65, 66]	ED			X	MVF		Entropy, motion intensity, attention & sentence detection	Sub-shot boundary alignment with onset & sentence	Home videos		Beat detection in incidental music
[21]	ED	-	-	X	LPC		Brightness, contrast, shake, face detection	Highest suitability score and edit while watching	Home videos		Video segmentation by removing shaky frames
[104]	ED	-	-	X	AMM		Stable, jerk, infidelity, brightness, blur, orientation	Maximisation of quality metric	Home videos		User study, rule & learning-based quality metrics
[145]	CP				MVF		Field line, goalmouth, centre circle, ball trajectory	Likelihood score maximisation for sub-shots	Soccer videos		Event detection using extracted features
[38]	CP	-	-				Object detection, tracking, size estimation	DBN for camera selection	Basketball, Surveillance		Event detection using extracted features
[9]	CP				3D CM		Stability, camera roll, 3D joint attention	Optimisation of feature cost in Trellis graph	UGVs	10	3D reconstruction of the scene
[133]	CP			X	LPC	X	Blockiness, blur, brightness, shake	Optimisation of the weighted sum of scores	UGVs	3	Manual segmentation for cut-point selection
[122]	CP			X	LPC	X	Blockiness, blur, contrast, brightness, occlusion, tilt, shake	Optimisation of the weighted sum of scores	UGVs	3	Manual cut-point & camera view selection
[J2]	CP	X	X	X	LPC	X	BRISQUE, shake	Rank-based camera-selection	UGVs	14	Automatic cut-point selection

For home video editing, Hua et al. [65, 66] performed sub-shot detection (obtained using predefined video length and frame difference curve maximisation), attention detection (obtained by analysing camera and object motion), sentence detection [96], low quality video filtering (obtained using entropy and motion intensity) and analysis of user-supplied music (by computing onset and tempo). In [21], brightness, contract, shakiness, and face detection were used for obtaining suitability score for home video sub-shots. Each sub-shot was extracted by filtering frames containing unwanted camera motion. Mei et al. [104] analysed spatio-temporal factors for home video summarisation, where unstableness and jerkiness were considered as temporal factors, and low fidelity (image with low contrast), brightness, blur and orientation were considered as spatial factors.

Mashup generation systems from UGVs have been proposed by Shrestha et al. [133] (FirstFit) and Saini et al. [122] (MoViMash). FirstFit [133] analysed video quality features such as blockiness, blur, brightness, shake, while MoViMash [122] additionally used occlusion and tilt for assigning scores to each frame. MoViMash [122] also introduced an offline learning stage which incorporated video editing rules, such as shooting angle, shooting distance and shot length. These methods performed manual segmentation of video clips. Further, MoViMash manually categorised the videos into right, left, centre, near and far for learning the shot-transition distributions. Low-quality audio decreases the perceived quality of the video as well [15], but audio signals were not analysed in FirstFit and MoViMash. In FirstFit [133], the audio was selected from the same media segment which contributed to camera selection, thus resulting in audio with varying quality. This sounded unpleasant when playing back the generated video. For MoViMash [122], the audio was not aligned with the video within the resulting mashups. Wilk and Effelsberg [151] 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 9-12 s duration on a 5-point scale corresponding to different levels of degradations. Their results showed that these degradations, in particular camera shake highly affected the perceived quality of UGVs. In [9], 3D structure of the scene [135] was reconstructed from multi-camera UGVs, and cameras' positions and orientations were estimated to compute their 3D joint attention. The 3D motion of a camera was used to estimate its stabilisation cost. The stabilisation, camera roll and joint attention cost were then used as features for camera-view selection. 3D reconstruction from hand-held cameras may fail if the number of cameras are not sufficient, scene is not well textured, or visual degradations (low luminance, poor contrast, motion blur) are present [9].

2.8.2 Camera view selection

Camera view selection is required to give the best viewing experience to the user. Most of the existing methods [38, 21, 133, 122, 104] select camera views by optimising the combined feature scores along with the introduction of sub-shot length constraint. Camera view selection strategies are detailed below for different video editing and composition frameworks for which the content analysis details were presented in the previous section (Sec. 2.8.1).

Camera selection in lectures [40, 152] focuses on the lecturer, slides or audience. Frame differencing in fixed cameras [40], or online detection and tracking in Pan-Tilt-Zoom (PTZ) cameras [152] is performed for the localisation of the lecturer. Similarly for meeting room video editing [117, 156], mainly person identification, speaker localisation, recognition and tracking are performed to select different camera views. The videos are generally captured from high quality fixed or stable moving cameras having constraint environments in a lecture and meeting rooms, and adequate lighting conditions, hence providing favourable conditions for speaker localisation 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 [145], the main camera is selected for most of the duration and sub-cameras are selected by maximising the likelihood score of suitable sub-camera segments. The sub-camera segments are classified by exploiting camera motion. For content and task-based best camera selection, Daniyal et al. [38] computed frame score by using number of objects, amount of activity, cumulative object score and event score, and applied Dynamic Bayesian Network (DBN) model for avoiding too frequent camera switching and for camera view selection.

For home video-editing, Hua et al. [65, 66] filtered the low quality video sub-shots by exploiting camera motion, and aligned the boundaries of better quality sub-shots with the selected music tempo, while preserving the detected sentence portions of the video. In [21], home video editing was performed by selecting sub-shots with the highest suitability score and by allowing the user to perform the editing while watching. Mei et al. [104] proposed three quality metric for home video summarisation, namely, user study-based (weighted average of all spatio-temporal factors), rule-based (nonlinear fusion [66]) and learning-based (offline two-class quality training) methods. A skim ratio (corresponding to the length of the final video summary) was defined and the sub-shots with maximised quality metric were selected to compose the summarised video.

In UGVs, camera selection is generally performed by selecting the shot with maximum quality score [133, 122]. Firstfit [133] applied an optimisation approach for camera view selection using image quality, cut-point, view diversity and user-preference score. In MoViMash [122], all videos were ranked based on the linear combination of visual quality, diversity and shake score. The camera switching instant was determined using the offline learned relationship between shot category (centre, left, right, near, far) and shot length. Center shots were generally selected for longer duration. At every second, occlusion and shake were also checked against a threshold to trigger camera switching. due to which frequent camera switching occurred. This introduced an unpleasant effect during video playback. In [9], the cost for stabilisation, camera roll and joint attention were optimised to compose the video, while adding a constraint on the minimum and maximum length of the sub-shot.

2.9 Summary

We reviewed the state-of-the-art video-based and audio-based identification and synchronisation methods. Video-based methods are designed for professional or controlled environment recordings, captured mostly using static and stable professional cameras (Sec. 2.5 and 2.6). Due to the sharpness of these recordings, the extraction and analysis of visual features are possible. Extending these approaches to UGVs is not trivial because there might not exist the same visual evidence between pairs of UGVs due to variations in the field of view, changing and poor lighting conditions, and visual quality. On the other hand, audio-based are used to organise multi-camera recordings of the same event (Sec. ??). A fixed classification threshold is used for matching a pair of recordings [20], that limits the generalisation of these methods for the identification and synchronisation of multiple events. Synchronisation of UGVs using audio features is generally based on onsets, fingerprints or audio-visual events. The performance of the existing audio-based methods is limited due to their sensitivity towards audio degradations. There is a need for an event identification and synchronisation framework for the automatic organisation of UGVs of several real-world events.

We discussed the existing video-based and inertial sensor-based camera motion detection methods (Sec. 2.7). Video-based methods are computationally expensive (e.g. 1s of 720×480 pixels resolution video at 30 fps contains 10 million pixels) and less accurate in the presence of moving objects and low luminance. An inertial sensor-based method reduces the computational cost due to the reduction in the amount of data to be processed (e.g. 1s of an inertial sensor contains 50 samples). Accelerometer and compass are used for camera motion detection [35] but the performance is limited by error introduced by the compass, and noise that exists due to the use of unfiltered accelerometer data. The use of gyroscope data can facilitate in camera motion detection as it provides a direct estimate of the angular velocity of the recording device.

We detailed the existing video composition methods designed for lecture webcast and meetings, sports video broadcast, home-video summarisation and multi-camera mashup generation (Sec. 2.8). The selection of features for audio/visual content analysis is dependent on the type of application under consideration. In UGVs, the perceived audio and visual quality is a key factor which makes the content enjoyable and interesting to playback [15, 151]. Therefore, audio continuity and uniformity is also required along with the appropriate view selection. Existing video composition methods [133, 122] for UGVs performed visual quality analysis only and manually selected the cut-points. Analysis of audio content for audio quality and continuity may facilitate in improving the overall perceived quality of the composed video.

In the next chapter, we present the multi-modal feature analysis for the designing of our proposed identification, synchronisation and video composition framework.

Chapter 3

Multi-modal feature analysis

3.1 Introduction

In this chapter, we introduce and analyse the audio, visual and inertial features used to design our proposed event identification and synchronisation [J1], and video composition [J2] frameworks. Identification or synchronisation of UGVs has been mainly performed by utilising audio features [20, 32, 76, 132], however the existing methods consider organisation of a single event only, and are sensitive to audio degradations (see Table ??). We exploit audio chroma feature [53] to develop an automatic query-by-example event identification and synchronisation framework [J1]. Details of the feature extraction, matching and analysis are presented in Sec. 3.2. Camera motion analysis of the synchronised UGVs can contribute in the designing of a video composition framework. The introduction of inertial sensors in smartphones is easing UGVs content analysis for camera motion detection and semantic information extraction [36, 37]. Therefore, we utilise gyroscope data for the camera motion analysis of UGVs [C1], and present the details in Sec. 3.3. Existing methods of video composition from overlapping multi-camera UGVs considered the visual content analysis only (see Table 2.6). Audio quality also plays a key role in enhancing the viewing experience [15]. Therefore, we analyse both audio and visual features to design a multi-camera video composition framework [J2]. Analyses of audio and visual features are presented in Sec. 3.4 and Sec. 3.5, respectively. The chapter is summarised in Sec. 3.6.

3.2 Audio-based identification and synchronisation

Automatic identification and synchronisation of UGVs involve feature extraction and matching to identify time overlapping recordings of the same event, and to estimate their synchronisation time-shifts. To achieve this goal, we use audio chroma feature as it gives the distribution of energy along different pitch classes (as presented in Sec. 2.3.3). Below we give details of the extraction, matching and analysis of the audio chroma feature.

3.2.1 Feature extraction

We first decompose a given audio signal A_n of a UGV into overlapping audio frames and then compute chroma features for each audio frame. Each audio frame is composed of an audio segment of frame size f_r and overlap shift h_p between two consecutive frames (as shown in Fig. 2.3). The number of audio frames G_n in A_n is a function of the number of audio samples K_n^a and is computed as

$$G_n = \frac{K_n^a}{s_n^a f_r h_p}. \tag{3.1}$$

The frequency spectrum $f(l)$ of each audio frame is then computed by applying the Discrete Fourier Transform (DFT), and is mapped into the pitch class using Eq. 2.2. The chroma vector for a particular audio frame is thus represented as $v^p \in \mathbb{R}^{12\times 1}$, such that p defines the time-stamp of a particular frame position. Chroma features F_n for the n^{th} audio signal A_n, segmented into G_n audio frames are given by

$$F_n = \{v_n^p\}_{g=1}^{G_n}, \tag{3.2}$$

where $v_n^p \in \mathbb{R}^{12\times 1}$ is the chroma feature vector for the p^{th} frame of the n^{th} camera's audio signal.

3.2.2 Feature matching

After feature extraction, feature matching is performed for computing the similarity and time-shifts between pairs of video recordings. Our proposed matching method operates by maximising the feature similarity between two video recordings. For a pair of recordings C_i and C_j, the distance between their chroma features F_i and F_j is given by $d_{ij}^{pq} = E(v_i^p, v_j^q)$, where $E(\cdot)$ is the Euclidean distance [140] between the p^{th} and q^{th} feature vector, and $p \in [1, G_i]$ and $q \in [1, G_j]$ give the range of frame numbers for C_i and C_j, respectively. The distance matrix Λ_{ij} between F_i and F_j is then given by

$$\Lambda_{ij} = [d_{ij}^{pq}]_{\mathbb{R}^{G_i \times G_j}}. \tag{3.3}$$

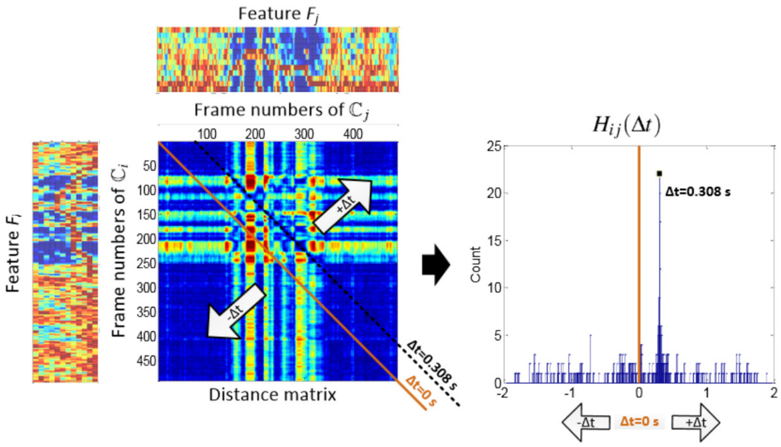


Figure 3.1: Feature matching using the distance matrix for two test audio signals of duration 2s is shown. The main diagonal of the distance matrix corresponds to zero, the lower diagonal corresponds to negative and the upper diagonal corresponds to positive time-shifts. The minimum across each row is calculated and the count of minimum distances is accumulated across each diagonal to give the histogram $H_{ij}(Dt)$. Peak in the histogram corresponds to the time-shift.

Figure 3.1 shows the distance matrix ^$_{ij}$ for two feature vectors obtained from two overlapping video recordings, each of 2s duration. The distance matrix ^$_{ij}$ contains information about the feature matching of two recordings. In order to interpret this information, the points of minimum distance across each row of the distance matrix ^$_{ij}$ are calculated. These correspond to the points c where the likely matches occur:

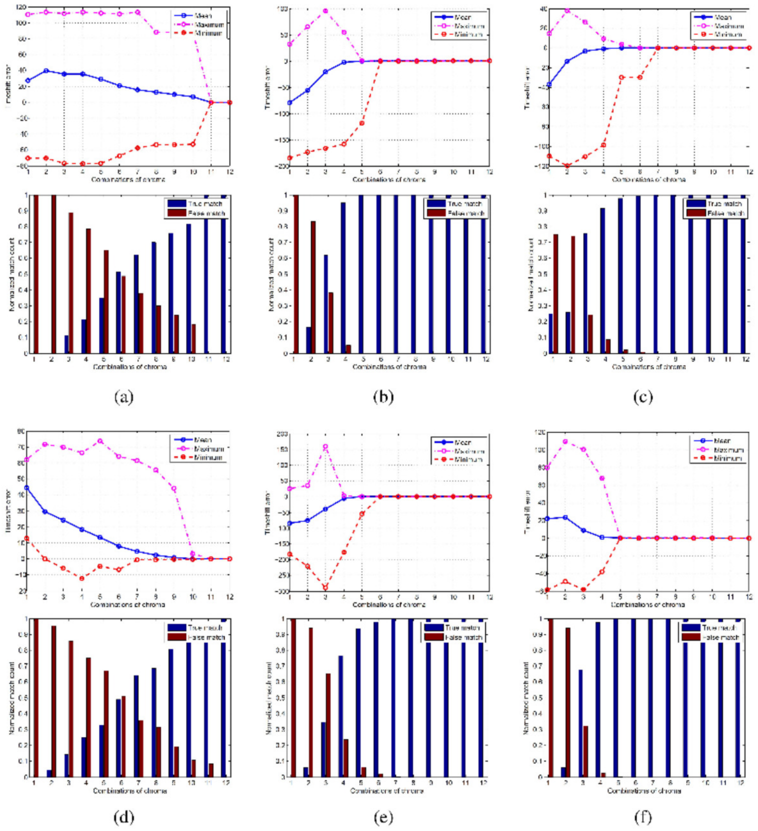
$$c = \underset{s}{\operatorname{argmin}}\,[d^{st}];\ 8t\ 2\ [1;G_j]: \tag{3.4}$$

The distance matrix ^$_{ij}$ is a rectangular matrix in which the main diagonal corresponds to zero time-shift. The upper and lower diagonals correspond to positive and negative time-shifts, respectively. We calculate the matching histogram $H_{ij}(Dt)$ for video recordings C_i and C_j from the distance matrix ^$_{ij}$, that gives the count of the number of minimum distances along each diagonal. This is illustrated in Fig. 3.1. The x and y-axes in $H_{ij}(Dt)$ correspond to the time-shifts and counts, respectively. If a pair of recordings is overlapping, we get a dominant peak in the matching histogram which represents the synchronisation time-shift, otherwise, it is unlikely to have a dominant peak.

3.2.3 Feature analysis

In order to find the lowest dimension of chroma feature which can provide the correct synchronisation time-shift, we conducted an experiment by analysing pairs of audio signals from different events. For F_i and F_j, we computed the synchronisation time-shifts for all combinations of 1 to 12 dimensions of chroma features, which are 12, 66, 220, 495, 792, 924, 792, 495, 220, 66, 12 and 1, respectively. Figure 3.2 shows the effect of varying the dimension of the chroma feature on six pairs of video recordings, where the first and third rows depict the maximum, mean and minimum time-shift error when testing with all possible combinations. The second and fourth rows show the occurrence of true and false matches which correspond to correct and incorrect synchronisation time-shifts with a □0:05s tolerance, normalised over all the combinations of varying dimensions of the chroma feature.

From this analysis, we observe that when the overlap between two signals is greater than 20% (18s) (Fig. 3.2 (b), (c), (e) and (f)), any combination of chroma beyond 6-dimensions is sufficient for achieving synchronisation. Otherwise, if the two audio signals are only partially overlapping and the length of one signal with respect to the other is short (7s with minimum 8% overlap), the synchronisation time-shift is not achieved until the 11th and 12th dimensions of the chroma feature as shown in Fig. 3.2 (a) and (d), respectively. In the case of Fig. 3.2 (a), a concert event pair containing amplified sound, the minimum overlap is 8% (7s) with respect to the longer recording. In the case of Fig. 3.2 (d), a public event pair containing strong audio degradations along with non-amplified sounds, the minimum overlap is 14% (12s) with respect to the longer recording. This overlap is required to get the correct synchronisation time-shift. Note that audio fingerprinting [132] is unable to give the correct synchronisation time-shift for these cases. It is observed that the minimum value of 8% overlap between a pair of recordings is required when performing feature matching. In the proposed feature matching, we use the minimum distance across each row c for the estimation of time-shift. This results in outliers in the matching histogram that may dominate when one of the recording is shorter than the other. This effect can be overcome by setting an empirical threshold on c for outlier removal or by using all 12 dimensions of chroma.



(a) (b) (c)

(d) (e) (f)

Figure 3.2: Effect of varying the dimensions of the chroma feature. The first and third rows show the maximum, mean and minimum time-shift errors for pairs of camera recordings. The second and fourth rows show the normalised true and false matches as counted for all combinations of varying dimensions of the chroma feature. (a) Nickelback_Event3 recording pair of duration 3:18 and 0:18s, (b) Nickelback_Event14 recording pair of duration 3:29 and 4:45s, (c) Madonna_Event recording pair of duration 2:59 and 1:20s, (d) Olympic torch Sheffield Event recording pair of duration 1:28 and 0:39s, (e) Olympic torch Mile end Event recording pair of duration 6:22 and 6:27s, and (f) Xmas dinner event recording pair of duration 3:19 and 2:17s.

3.3 Gyro-based camera motion detection

Camera motion can be classified into four types, namely, pan/tilt, shake, stable and zoom [6]. We aim to detect the first three types using gyroscope, while zoom is not considered as it is independent of the inertial sensors. Gyroscope is more accurate for rotation estimation than other inertial sensors (i.e. accelerometer and magnetometer) as it measures the angular velocity around

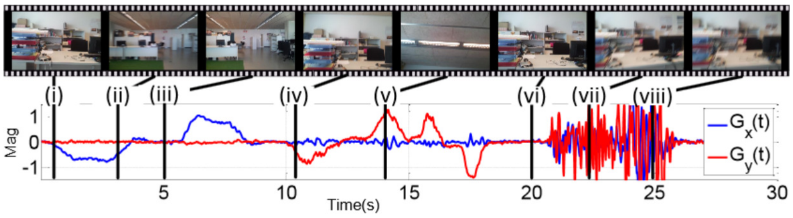


Figure 3.3: A sample multi-modal recording containing pan, tilt and shake motions. (i)(ii) Pan, (iv)(v) tilt, (vii)(viii) shake, and (iii)(vi) stable motions are labelled for visualisation.

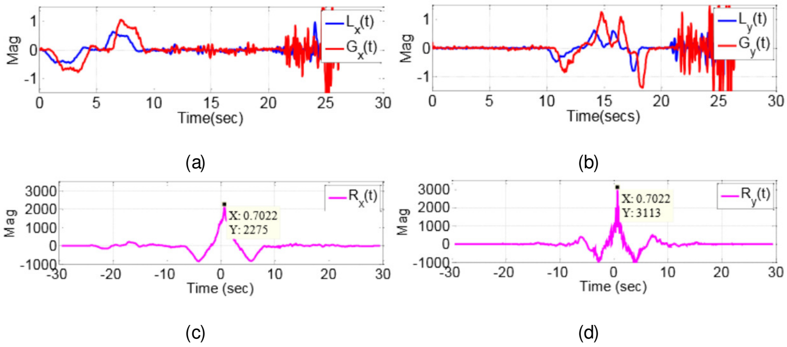
the device's x, y and z axes. These angular velocities correspond to the camera pan, tilt and roll motions, respectively (Fig. 2.4(a)). High correlation exists between visual and gyroscope data when captured from a single device. Figure 3.3 shows a sample multi-modal recording containing dominant camera motions.

The frequency of involuntary human body movement lies within $f_i = 20$Hz [75]. Inertial sensors are logged and analysed at $s^l = 50\text{Hz} > 2f_i$, thus satisfying the Nyquist theorem [55]. The magnitude of $G(t)$ is never zero for a smartphone video captured without tripod because of the involuntary human body movement. In the absence of intentional camera motion, the involuntary body movement results in low magnitude camera motion. This information is sufficient for the video and gyroscope data synchronisation (as detailed in Sec. 3.3.1). We propose CMDG, a gyroscope-based method for camera motion detection in UGVs [C1]. We assume that there is no translational motion of the camera. We utilise the dominant motions in the polar representation of the low-pass filtered gyroscope data for pan and tilt detections, and consider shake as dominant high frequency movements. We further apply morphology to remove outliers and identify time continuous motions. The proposed method is presented below in detail.

3.3.1 Gyro-visual synchronisation

In sensor-based activity recognition [8, 124], inertial and visual data are recorded independently from two devices. These modalities are synchronised either by time-stamp or manual observation of an intentional event in both devices. The process of synchronisation can be simplified and automated if both modalities are logged from a single device giving an ego-centric view.

Visual data when logged with sensors has an unknown delay due to the time taken by the camera to start the recording. We correlate the gyroscope and visual data to correct this delay.



(a) (b) (c) (d)

Figure 3.4: Gyro-visual synchronisation. (a) LPC pan $L_x(t)$ and gyroscope $G_x(t)$, (b) LPC tilt $L_y(t)$ and gyroscope $G_y(t)$. Correlation (c) $R_x(t)$ and (d) $R_y(t)$.

We use LPC [112, 141]) for computing the horizontal $L_x(t)$ and vertical $L_y(t)$ displacements from $V(t)$ (as detailed in Sec. 2.3.4). $L_x(t)$ and $L_y(t)$ are referred as pan (left-right) and tilt (up-down) motions, and correspond with $G_x(t)$ and $G_y(t)$, respectively (Fig. 3.4(a) and (b)). We down-sample $G(t)$ by re-sampling it at the same rate as of $V(t)$, and compute the cross-correlation $R_x(t)$ as

$$\begin{aligned} R_x(t) &= \sum_{k=-\infty}^{\infty} G_x(k)L_x(k+t); \\ \tau_x &= \operatorname*{argmax}_t R_x(t); \\ \rho_x &= \max_t R_x(t); \end{aligned} \tag{3.5}$$

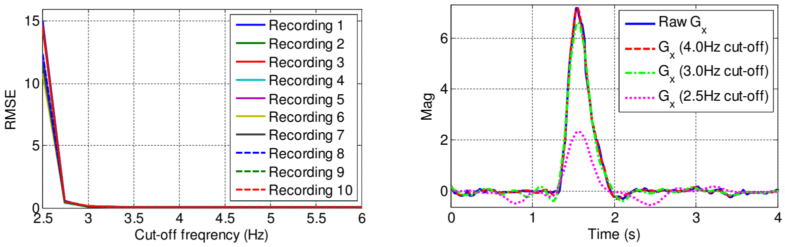
where τ_x is the estimated delay and ρ_x is the correlation peak from the pan signals. Likewise, $R_y(t)$, τ_y and ρ_x are computed from the tilt signals. τ_x and τ_y are equal for recordings containing pan and tilt, and are only approximately equal if only one of the motion exists. The overall estimated delay τ is selected as

$$\tau = \begin{cases} \tau_x & \text{if } (\rho_x > \rho_y); \\ \tau_y & \text{if } (\rho_y > \rho_x): \end{cases} \tag{3.6}$$

Figure 3.4(c) and (d) show the cross-correlations $R_x(t)$ and $R_y(t)$, and their peaks.

3.3.2 Camera motion detection using gyroscope data

In order to find the cut-off frequency f_c for the Low-Pass Filter (LPF) to detect pan and tilt, we captured a set of recordings by performing as fast panning as possible. LPF is then applied by varying f_c from 2.5Hz to 6Hz, and the Root Mean Square Error (RMSE) between the raw $G_x(t)$



(a) (b)

Figure 3.5: Analysis of the cut-off frequency for the LPF. (a) The RMSE between the $G_x(t)$ and LPF $G_x^L(t)$ signals for 10 sample recordings containing fast pan. The cut-off frequency is varied from 2.5Hz to 6.0Hz. (b) The raw $G_x(t)$ (for the fast pan) for one of the recordings and its low-passed signal at different cut-off frequencies.

and filtered $G_x^L(t)$ signal is computed (shown in Fig. 3.5(a)). RMSE is high when $f_c \leq 3$Hz, as the signals lose substantial information. The same affect is observed from Fig. 3.5(b), where $G_x(t)$ (for fast pan) and $G_x^L(t)$ at different f_c are shown for one of the recordings. We select $f_c = 4$Hz as it gives a very small value of the average RMSE for the set of fast pan recordings, showing the signals' information is retained.

A panning motion produces high magnitude of $G_x(t)$, and ideally zero magnitude of $G_y(t)$ which is also true for tilt motion. An independent threshold on LPF $G_x^L(t)$ and $G_y^L(t)$ might be applied for camera motion detection but this would require to verify if for the detected pan in $G_x^L(t)$, there is no detected tilt in $G_y^L(t)$, and vice versa (Fig. 3.6(a)). To overcome this, we jointly analyse $G_x^L(t)$ and $G_y^L(t)$ by transforming them into polar coordinates to get the magnitude $G_\rho^L(t) = \sqrt{G_x^L(t)^2 + G_y^L(t)^2}$ and angle $G_\theta^L(t) = \arctan\left(\frac{G_y^L(t)}{G_x^L(t)}\right)$ (Fig. 3.7). Pan $P(t)$ is the displacement along the horizontal (Fig. 3.7(b)) and is detected as

$$P(t) = \begin{cases} +1 & \text{if } (0 - \alpha \leq G_\theta^L(t) \leq 0 + \alpha); \\ -1 & \text{if } (180 - \alpha \leq G_\theta^L(t) \leq 180 + \alpha); \\ 0 & \text{otherwise}; \end{cases} \tag{3.7}$$

where α (in degrees) is the tolerance angle, and $+1$ and -1 denote pan left and right, respectively. Fig. 3.6(d) shows the absolute detected pan $|P(t)|$.

Likewise, tilt $T(t)$ is the displacement along the vertical (Fig. 3.7(c)) and is detected as

$$T(t) = \begin{cases} +1 & \text{if } (90 \square a \square G^L_q(t) \square 90 + a); \\ \square 1 & \text{if } (270 \square a \square G^L_q(t) \square 270 + a); \\ 0 & \text{otherwise;} \end{cases} \qquad (3.8)$$

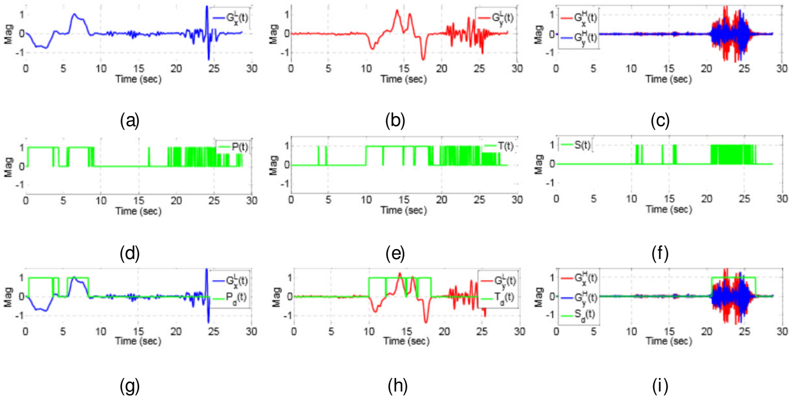
where $+1$ and $\square 1$ denote tilt down and up, respectively. Figure 3.6(e) shows the absolute detected tilt $jT(t)j$. A typical value of a should be the one which facilitates the detection of horizontal and vertical motions. In real-world scenarios, when a smartphone user performs freehand pan during recording, the magnitude of $G_x(t)$ is high but $G_y(t)$ also has some low magnitude value. This is because freehand pan is not strictly along x-axis which is also true for the tilt. Hence, a can not be close to zero. Through extensive experimentation, we selected $a = 30^o$, and showed the effect of varying a in Fig. 5.5(a) (to be discussed in Sec. 5.5).

For shake detection, we obtain $G^H_r(t)$ and $G^H_q(t)$ from $G^H_x(t)$ and $G^H_y(t)$ (Fig. 3.7(d)). $G^H_q(t)$ can take any direction but $G^H_r(t)$ defines the amount of data to be classified as shake. Therefore, shake $S(t)$ is detected as

$$S(t) = \begin{cases} 1 & \text{if } G^H_r(t) > b; \\ 0 & \text{otherwise;} \end{cases} \qquad (3.9)$$

where b is the tolerance magnitude. $G^H_r(t)$ ranges from 0 to 0.5. The involuntary body movement in freehand recordings lies in the high frequency, and needs to be thresholded in order to effectively detect the shake. A value of b close to zero makes the detection extremely sensitive and detects the involuntary movement as well. A value higher than 0.1 thresholds significant amount of $G^H_r(t)$, making the detection ineffective. We selected $b = 0{:}06$ through experimentation, and presented the effect of varying b in Fig. 5.5(b) (to be discussed in Sec. 5.5).

$jP(t)j$, $jT(t)j$ and $S(t)$ give binary signals for samples with detected pan, tilt and shake motions (Fig. 3.6(d-f)). It is possible to detect false motion in few samples as till now we have not considered time continuity. In order to remove outliers and to detect time continuous segments, we apply morphological operations of Opening and Closing [55]. We apply Opening to $jP(t)j$ and $jT(t)j$ that performs erosion to remove false detection followed by dilation to detect the continuous segments, and obtain the final detection $P_d(t)$ and $T_d(t)$ (Fig. 3.6(g-h)). To detect continuous segments of shake, we apply Closing that performs dilation to connect discontinuous segments followed by erosion to maintain the original length of the shake detected segments. Segments smaller than $0{:}25\,s$ are considered as outliers, and are removed to obtain the final shake

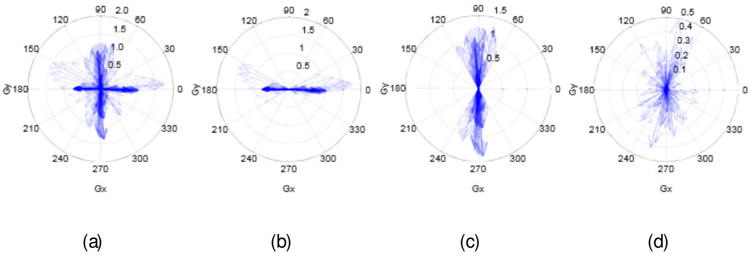


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.6: Pan, tilt and shake detection. For pan, (a) $G^L_x(t)$, (d) $jP(t)j$, (g) $P_d(t)$ along with $G^L_x(t)$; for tilt, (b) $G^L_y(t)$, (e) $jT(t)j$, (h) $T_d(t)$ along with $G^L_y(t)$; for shake, (c) $G^H_x(t)$ and $G^H_y(t)$, (f) $S(t)$, (i) $S_d(t)$ along with $G^H_x(t)$ and $G^H_y(t)$, are shown.



(a) (b) (c) (d)

Figure 3.7: Analysis of $G_x(t)$ and $G_y(t)$ in polar coordinate. (a) $G^L_r(t)$ and $G^L_q(t)$ for pan and tilt detection, (b) detected pan vectors, (c) detected tilt vectors, (d) $G^H_r(t)$ and $G^H_q(t)$ for shake detection, are shown.

detection $S_d(t)$ (Fig. 3.6(i)). We perform binary classification to independently detect pan, tilt and shake. Therefore, it is possible to detect samples containing pan and shake, or tilt and shake motions as well. We can also detect stable samples by combining $P_d(t)$, $T_d(t)$ and $S_d(t)$ using logic OR and inverting the resulting binary signal. The stable samples are useful for shot selection applications.

The experimental validation of the proposed CMDG is presented in Sec. 5.6.

3.4 Audio analysis for video composition

We analyse the audio quality of multiple overlapping video recordings for the multi-camera video composition, and propose an audio-stitching method that involves obtaining consistent and uniform audio, A^{st}, for the coverage duration, D_c, of the event. We also propose a cut-point selection method, which aims at finding the binary signal, A^{st}_U, for the suitable cut-points, $U = (u_1; \ldots; u_j; \ldots; u_J)$, where U is in second (s) to be used as a common reference point for both audio and video segmentation. We obtain U by analysing three audio features, namely root mean square, A^{RMS}, spectral entropy, A^{SE}, and spectral centroid, A^{SC}, of the stitched audio, A^{st}. The proposed audio-stitching and cut-point selection methods are presented below in detail.

3.4.1 Audio-quality analysis for audio ranking

The set of synchronised and overlapping audio signals, A, of an event contains sound recorded by different devices at different locations. Hence, the quality of audio varies from one video to another. For audio-stitching, we need to know which audio is better in A. In order to achieve this, we analyse the spectral rolloff [89] of the set of audio signals, A, to rank their quality.

Spectral rolloff estimates the amount of the right-skewedness of the frequency spectrum by calculating the frequency (rolloff point) below which 85% of the signal energy is contained [89]. Real-world degradations present in UGVs introduce high frequencies in the audio signal and shifts the resulting spectral rolloff point to a higher frequency bin of the spectrum. Therefore, for designing the ranking strategy, we assume that the overlapping audio signal with low spectral rolloff point contains less noise than the audio signals with high spectral rolloff point. This is illustrated with the help of an example in Fig. 3.8. The spectrum in Fig. 3.8(b) is more concentrated towards low frequency bins and contains less noise as compared to the spectrum in Fig. 3.8(a).

For ranking the audio signals, we decompose each A_n for the overlap duration, $D_o = [l_{a^0}; l_{a^{00}}]$, into non-overlapping frames $1; \ldots; g_R; \ldots; G_R$ using frame size $f_{r3} = 1$s (selected empirically). G_R is the total number of audio frames in D_o of each A_n. We varied the frame size, f_{r3}, from 0:5s to 3:0s 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. To compute the spectral rolloff point, we first obtain the Fourier transform of the audio signals, A. Cumulative frequency is then computed within each frame, g_R, of A to estimate the spectral rolloff point. The spectral

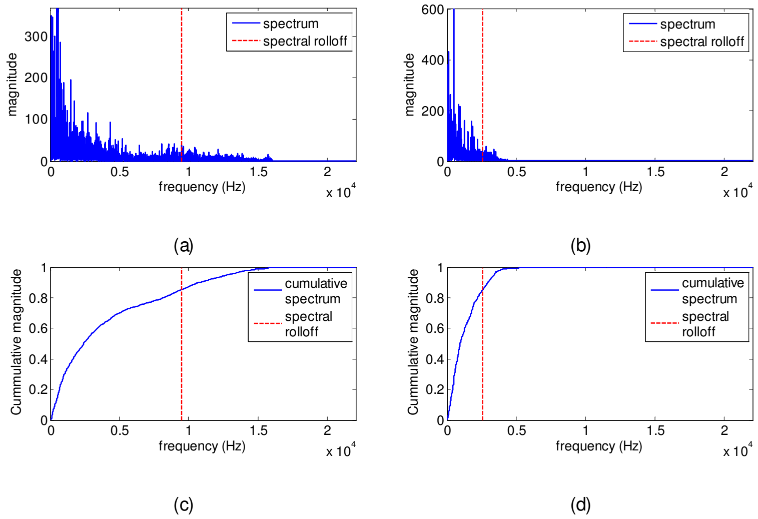


Figure 3.8: Spectral rolloff analysis for audio ranking. (a) and (b) show the spectra of a synchronised audio frame from two audio signals, and (c) and (d) show their respective cumulative spectra. The spectral rolloff is shown in red. Spectrum in (b) contains less noise as compared to the spectrum in (a) since the spectrum in (b) is more concentrated towards low frequency bins.

rolloff point, A^{SR}, for A is given by

$$A^{SR} = [A^{SR}(1); \cdots; A^{SR}(g_R); \cdots; A^{SR}(G_R)]; \tag{3.10}$$

where,

$$A^{SR}(g_R) = [A_1^{SR}(g_R); \cdots A_n^{SR}(g_R); \cdots; A_N^{SR}(g_R)]^T : \tag{3.11}$$

$A^{SR}(g_R)$ is the spectral rolloff at the g_Rth frame for all N recordings in A and $A_n^{SR}(g_R)$ is the vector listing spectral rolloff at the g_Rth frame of A_n. This is followed by computing the rank matrix R^{SR} within each frame by sorting each $A^{SR}(g_R)$ in ascending order and obtaining its argument. The most frequently occurring audio signal in each row of R^{SR} is selected to be the one with the best quality, followed by others. This gives the rank vector $R^{SR} = [r(1); \cdots; r(n); \cdots; r(N)]^T$ that contains the indices of the N audio recordings in descending order of their quality.

3.4.2 Audio-stitching using the rank vector

To obtain a continuous audio track from the earliest starting video till the last ending one, we perform audio-stitching using the rank vector, R^{SR}. Synchronisation provides the relative starting,

```
if $L_{r(2)} < L_{r(1)}$ & $E_{r(2)} > E_{r(1)}$ then
    $\dot{A}^{st} = (a_{L_{r(2)}}; \cdots; a_{E_{r(2)}})$
else if $L_{r(2)} < L_{r(1)}$ & $E_{r(2)} < E_{r(1)}$ then
    $\dot{A}^{st} = (a_{L_{r(2)}}; \cdots; a_{L_{r(1)}}; \cdots; a_{E_{r(1)}})$
else if $L_{r(2)} > L_{r(1)}$ & $E_{r(2)} > E_{r(1)}$ then
    $\dot{A}^{st} = (a_{L_{r(1)}}; \cdots; a_{E_{r(1)}}; \cdots; a_{E_{r(2)}})$
else
    $\dot{A}^{st} = (a_{G_{r(1)}}; \cdots; a_{E_{r(1)}})$
end
```

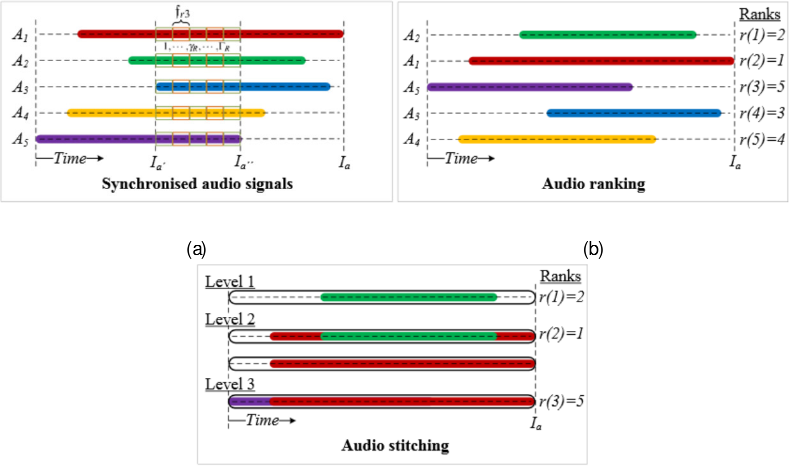
Algorithm 1: Audio stitching algorithm at level 2.

L_n, and ending, E_n, times of each A_n. Our audio-stitching algorithm contains N levels, but it terminates as soon as stitched audio, $A^{st} = (a_1^{st}; \cdots; a_i^{st}; \cdots a_{l_a}^{st})$, for the coverage duration, D_c, is obtained. At level 1 of the stitching, $A_{r(1)}$ is selected to span for the duration $L_{r(1)}$ to $E_{r(1)}$, thus resulting in intermediate stitched audio, $\dot{A}^{st} = (a_{L_{r(1)}}; \cdots; a_{E_{r(1)}})$. At level 2, in order to reduce the number of stitched points, we compromise between the quality and the number of stitched points. Therefore, we update $\dot{A}^{st}$ by checking if $A_{r(2)}$ is completely, before (earlier starting time) or after (later ending time) contained within $A_{r(1)}$ (see Algo. 1). In a situation where $A_{r(2)}$ is completely contained within $A_{r(1)}$, we do not update $\dot{A}^{st}$ and move to the next level. The process continues until we obtain A^{st} for the coverage duration, D_c. This process of audio ranking and stitching is illustrated in Fig. 3.9.

3.4.3 Cut-point selection using audio features

According to professional film-editing rules, every cut-point should have a motivation such as camera motion, occlusion or silence to voice transition [41]. We select cut-points by analysing the dynamics of A^{st}. This is supported by our two assumptions: (i) change in camera view is meaningful when a transition in audio occurs (e.g. silence to audio/music, change or addition of an instrument, low to high volume, music to vocal), and (ii) transition within an audio/music signal causes a significant change in the dynamics of its features.

We propose a cut-point selection method by analysing three low-level audio features of A^{st} to detect those audio samples where the change occurs. These features are root mean square, A^{RMS}, spectral centroid, A^{SC}, and spectral entropy, A^{SE}, [89]. Root mean square, A^{RMS}, is useful for detecting silence periods in audio signals and for discriminating between different audio classes.



(a) (b)

(c)

Figure 3.9: Audio-stitching illustration (audio signals represented by coloured bars). (a) Synchronised A_n are decomposed into non-overlapping frames (G_R) using f_{r3} for the $D_o = [I_{a'}; I_{a''}]$ duration. (b) Rank vector (R^{SR}) is then obtained by analysing audio quality within each frame. (c) Finally, audio-stitching is performed to obtain a continuous audio signal for the coverage duration D_c of the event.

Spectral centroid, A^{SC}, is effective in describing the spectral shape and predicting the brightness of the audio as it measures the centre of mass of the audio spectrum. A sudden change in A^{SC} is interpreted as an instrumental change in music [89, 127]. Spectral entropy, A^{SE}, is used to detect silence and voice segments of the speech [119]. It is also used to discriminate 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 A^{st} into non-overlapping frames $1; \cdots; g_C; \cdots; G_C$ with frame size, f_{r4}, (see Sec. 3.4.4), and compute the low level features A^{RMS}, A^{SC}, A^{SE} within each frame g_C as

$$A^{RMS} = [a^{RMS}(1); \cdots; a^{RMS}(g_C); \cdots; a^{RMS}(G_C)]; \tag{3.12}$$

$$A^{SC} = [a^{SC}(1); \cdots; a^{SC}(g_C); \cdots; a^{SC}(G_C)]; \tag{3.13}$$

$$A^{SE} = [a^{SE}(1); \cdots; a^{SE}(g_C); \cdots; a^{SE}(G_C)]; \tag{3.14}$$

where $a^{RMS}(g_C)$, $a^{SC}(g_C)$ and $a^{SE}(g_C)$ are the root mean square, spectral centroid and spectral entropy values at the $g_C{}^{th}$ frame, respectively. The total number of frames are computed as

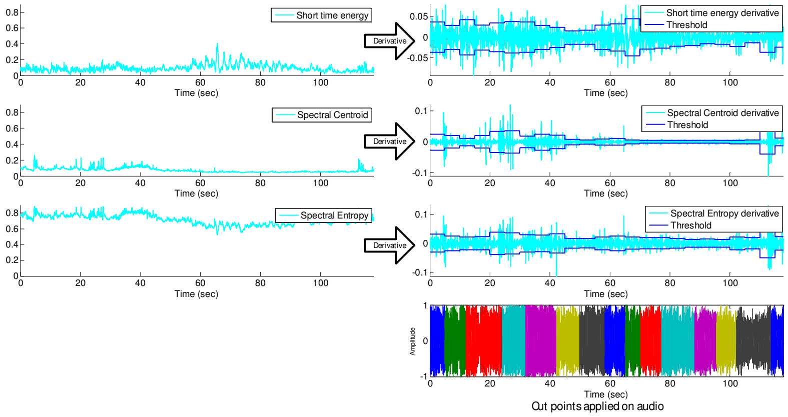


Figure 3.10: Audio features extraction and cut-point selection. Root mean square (top left), spectral centroid (middle left) and spectral entropy (bottom left) of the input audio signal. The respective derivatives are shown (on the right). A dynamic threshold is applied within an analysis window (W_a) and the cut points are computed while staying within the minimum l_{min} and maximum l_{max} video-shot duration limits.

$G_C = \frac{D_c}{f_{r4}}$. We then compute the derivative D^{RMS}, D^{SC}, D^{SE} of the features A^{RMS}, A^{SC}, A^{SE} as

$$D^{RMS} = [d^{RMS}(1); \cdots; d^{RMS}(g_c); \cdots; d^{RMS}(G_c)]: \tag{3.15}$$

Likewise, we obtain D^{SC} and D^{SE}. The response of the three features computed for the stitched audio A^{st} along with their derivatives is shown in Fig. 3.10.

For statistical analysis, we inspect the dynamics of the feature derivatives D^{RMS}, D^{SC}, D^{SE} within an analysis window, W_a, by computing the mean, $\bar{m}_a = [m_a^{RMS}, m_a^{SC}, m_a^{SE}]^T$, and standard deviation, $S_a = [s_a^{RMS}, s_a^{SE}, s_a^{SC}]^T$. The threshold, $\bar{t}_a = [t_a^{RMS}, t_a^{SC}, t_a^{SE}]^T$, is computed as

$$\bar{t}_a = \bar{m}_a + h S_a; \tag{3.16}$$

$$\begin{bmatrix} t_a^{RMS} \\ t_a^{SC} \\ t_a^{SE} \end{bmatrix} = \begin{bmatrix} m_a^{RMS} \\ m_a^{SC} \\ m_a^{SE} \end{bmatrix} + h \begin{bmatrix} s_a^{RMS} \\ s_a^{SC} \\ s_a^{SE} \end{bmatrix}; \tag{3.17}$$

where h defines the weight for the standard deviation S_a to be applied for computing the outliers within each W_a. For initialisation, we set $h = 2:5$ by considering that the data under W_a is normally distributed which gives a confidence interval of 0.985 [77]. The threshold $\bar{t}_a$ is applied locally to each feature vector. The values of feature vector derivatives above $\bar{t}_a$ correspond to outliers representing significant changes in the dynamics of A^{st}. These values are marked as one while the values below $\bar{t}_a$ are marked as 0. This results in the binary value

$$b^{RMS}(g_c) = \begin{cases} 0 & d^{RMS}(g_c) < t_a^{RMS}; \\ 1 & \text{otherwise}; \end{cases} \tag{3.18}$$

for the binary vector

$$B^{RMS} = [b^{RMS}(1); \cdots; b^{RMS}(f); \cdots; b^{RMS}(F)]: \tag{3.19}$$

Likewise, B^{SC} and B^{SE} are computed. The three binary vectors are then fused together with a logic AND ($\wedge$) operator to get the binary cut-points

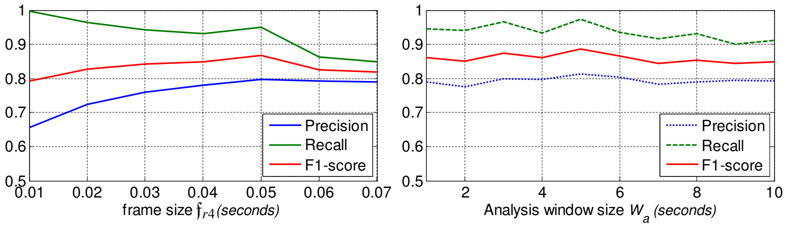
$$A_U^{st} = B^{RMS} \wedge B^{SC} \wedge B^{SE}: \tag{3.20}$$

Finally, we overlay the binary cut-points vector, A_U^{st}, on the audio signal, A^{st}, to get the time-stamps for its suitable cut-points, U. Figure 3.10 (right) shows the D^{RMS}, D^{SC} and D^{SE} along with the applied threshold, $\bar{t}_a$, and the resulting segmented audio signal.

3.4.4 Parameters for cut-point selection

To decompose an audio signal into frames for the feature extraction, we select the frame size $f_{r4} = 0:05$s. Typical value for the frame size is between 0.01s to 0.05s [54, 128]. The frame size should be large enough to have sufficient data for the feature extraction and short enough to make the signal (approximately) stationary [54]. In order to validate the frame size selection, we manually labelled an audio signal (of 8 minutes duration) to obtain the ground-truth cut-points. We evaluate our proposed cut-point detection method by varying f_{r4} from 0.01s to 0.07s (Fig. 3.11(a)). It is observe 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 between 1s to 10s [54]. We select $W_a = 5$s for our proposed method. We demonstrate the effect of varying W_a in Fig. 3.11(b). It is observe that the F_1-score does not vary significantly between the typical value range and the mean F_1-score is 86% with standard deviation of $1:4$%.

We select the minimum, l_{min}, and maximum, l_{max}, limits for the video-shot duration, and adjust 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



(a) (b)

Figure 3.11: Analysis of frame f_{r4} and analysis window W_a size. (a) The effect of varying f_{r4} while fixing $W_a = 5$s. (b) The effect of varying W_a while fixing $f_{r4} = 0{:}05$s.

shorter than l_{min} may not be understandable [9, 157]. 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 $\bar{f}_a$ (Eq. 3.17) to enforce shot duration limits on the cut-point selection method. When h is high, $\bar{f}_a$ 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 $\bar{f}_a$ is lowered iteratively by decreasing h 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.

3.5 Visual analysis for video composition

For multi-camera video composition, we analyse the visual content V of C for computing certain visual assessment scores to account for the visual quality, camera motion and view diversity. The video quality assessment aims at obtaining spatial $S = \text{f } S_n g^N_{n=1}$ and spatio-temporal $T = \text{f } T_n g^N_{n=1}$ quality scores from V, where $S_n = (s_{n1}; □□□; s_{ni}; □□□; s_{nl_v})$ and $T_n = (t_{n1}; □□□; t_{ni}; □□□; t_{nl_v})$, respectively. Detailed below are the spatial and spatio-temporal quality assessments, and view diversity strategy that we employed for designing the proposed video composition framework.

3.5.1 Spatial quality assessment

In order to filter low-quality video frames, we perform spatial quality analysis of UGVs. We use BRISQUE (Blind/Referenceless Image Spatial Quality Evaluator) [108] for the image spatial quality-assessment. BRISQUE quantifies several degradations caused by video compression, image blur and additive white Gaussian noise, as compared to other approaches that are degradation-specific [50, 129, 139, 158]. It is a non-reference based image quality measure, which is designed using Mean Subtracted Contrast Neutralized (MSCN) coefficients [121]. MSCN coefficients refer to a property of natural scene statistics, which states that the subtraction of local means from image luminances and normalisation by local variances produces decorrelated coefficients [121]. BRISQUE computes features by fitting a generalised Gaussian distribution to the MSCN coefficients, and by fitting asymmetric generalised Gaussian distribution to pairwise products of neighbouring MSCN coefficients. 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 SVM regressor.

For each V_n in V, the spatial quality score, S_n, is computed using BRISQUE. Each S_n in $S = \{S_n\}_{n=1}^{N}$ is synchronised such that an assessment score s_{1i} for C_1 at i^{th} frame corresponds to the same time instant for the score s_{2i} for C_2. Value of s_{1i} lies between 0 and 1, where a higher value indicates better visual quality. Figure 3.12 shows the computed BRISQUE score for three synchronised UGVs of an event. The spatial quality scores S are normalised using the z-score.

3.5.2 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 [112] in which they estimate the camera pan and tilt using Luminance Projection Correlation (LPC) (detailed in Sec. 2.3.4). We use this approach [112] as opposed to other optical flow-based [7] and template matching-based [6] approaches which are computationally expensive. Furthermore, LPC has been previously tested for hand-held camera's video analysis [21]. 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, we obtain the tilt signal. A threshold [112] is applied to these signals for detecting the pan and tilt. Pan left is labelled as positive and right as negative. Tilt up is labelled as positive and down as negative.

In order to estimate spatio-temporal quality score which is given by camera shake, we use the method proposed by Campanella et. al [21] in which they apply low pass filtering to the pan and tilt signals [112], and compute the camera shake by taking the differences of original and filtered pan and tilt signals. The higher the value of t_{ni} (score for the i^{th} frame of the n^{th} camera) the more stable the video. Figure 3.12 shows the computed spatio-temporal score for three synchronised UGVs of an event. The computed $T = \{T_n\}_{n=1}^{N}$ is normalised using the z-score normalisation.

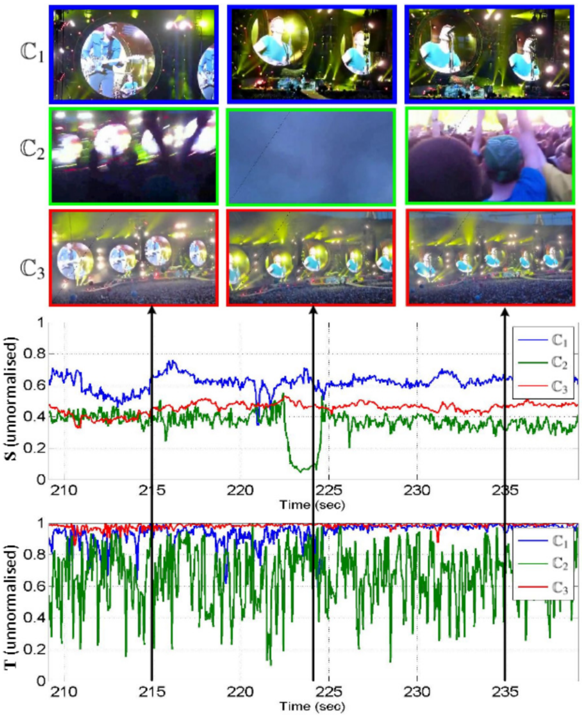
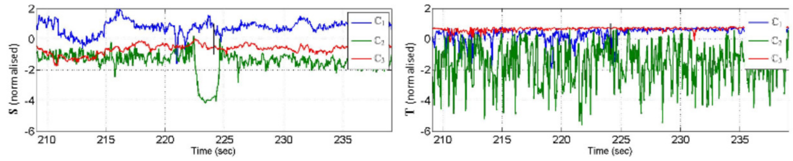


Figure 3.12: The spatial and spatio-temporal score without z-score normalisation for three synchronised camera recordings. Representative frames at three time instants are shown for visualisation. The two scores are not comparable because of been independent from each other.

The magnitudes of spatial and spatio-temporal scores are not comparable without normalisation because they have different scales. The z-score normalises each score to have mean equal to zero and standard deviation equal to one, thus allowing their comparison. S and T are shown after z-score normalisation in Fig. 3.13.

3.6 Summary

In this chapter, we presented the extraction, matching and analysis of the audio chroma feature that we exploit for designing the proposed identification and synchronisation [J1] framework. We selected audio chroma feature as it gives the distribution of audio energy along different pitch classes, making it discriminant even in the presence of audio degradations. Since perceived



(a) (b)

Figure 3.13: Spatial S and spatio-temporal T scores after z-score normalisation for the time duration shown in Fig. 3.12. The two scores are comparable after normalisation.

visual quality is influenced by camera motion [151], we proposed a method for camera motion detection using gyroscope data for UGVs [C1]. The proposed method used tri-axial gyroscope data captured simultaneously with the video to time synchronise sensor-visual data, and to detect pan, tilt and shake motions. We analysed both audio and visual content for designing the proposed multi-camera video composition framework [J2]. In order to rank multiple overlapping audio signals of an event in descending order of their quality, we analysed their spectral rolloff. We used this ranking to obtain consistent and uniform audio for the coverage duration of the event. We proposed a suitable cut-points detection method by analysing three audio features, namely root mean square, spectral entropy and spectral centroid. We analysed the visual content of multiple overlapping UGVs to obtain spatial and spatio-temporal quality assessments. Further, we presented a view diversity strategy to be employed for camera selection during video composition.

In the subsequent chapters, we use the above mentioned features and present the proposed identification and synchronisation framework, and video composition framework in detail. We also exploit gyroscope data for designing a variant of our proposed video composition framework.

Chapter 4

Identification and synchronisation of multi-camera user-generated videos

4.1 Introduction

Multi-camera event identification in UGVs requires a discriminant descriptor for obtaining the event representation followed by learning for the automatic retrieval of all UGVs that are overlapping with the query. Synchronisation involves spatio-temporal matching of features from two or more UGVs of the same event to estimate the perfect alignment between them. Existing methods use audio fingerprinting for organising UGVs of the same event [76, 32, 20] by applying a fixed classification threshold. Audio onsets [132], audio fingerprintings [132, 76] or audio-visual events [23] are used for multi-camera UGVs synchronisation. However, most of these methods are sensitive to reverberations and local degradations. We propose an automatic query-by-example event identification and synchronisation framework using audio chroma feature [J1]. Although the recording of a specific event captured by multiple devices might differ in loudness due to the varying quality of recording devices, the distance of the device from the sound source and surrounding noise, the pitch of the recorded remains constant [31]. For this reason, we use chroma as an audio feature [53], as it gives the distribution of energy along different pitch classes. The novelty of this work lies in the design of a descriptor from match and non-match histograms that facilitates the definition of an automatic classification threshold for event identification and clustering. We show the robustness of the proposed synchronisation method compared to alternative methods over various audio degradations.

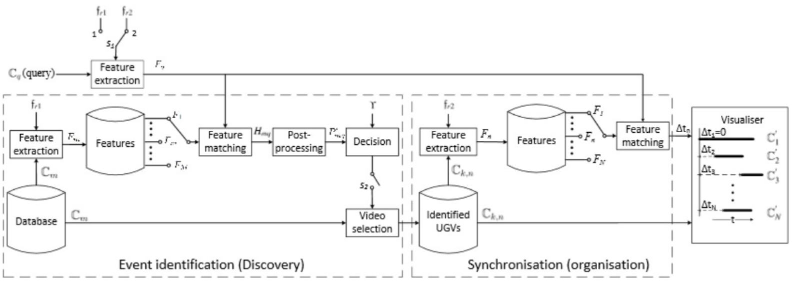


Figure 4.1: Block diagram of the proposed framework, which is composed of two stages, event identification (discovery) and synchronisation (organisation). For a given query video C_q, feature extraction is performed with f_{r1} (s_1 = 1; s_2 = OFF), and its feature matching is done with the feature database of M UGVs to generate the feature matching histogram H_{mq}. Post-processing is then performed and a classification threshold ° is applied to identify the set of N overlapping recordings. The time-shift estimation Dt_n is then performed with f_{r2} (s_1 = 2; s_2 = ON) for these N recordings in order to synchronise them. A multi-camera visualiser is used for playback of the N synchronised UGVs.

This chapter is organised as follows. In Sec. 4.2, we present an overview of the proposed framework. In Sec. 4.3, we describe our proposed event identification framework, which is followed by time-shift estimation and cluster membership validation in Sec. 4.4. In Sec. 4.6, we describe our dataset of UGVs, assess our method and compare the method with the existing state of the art. Finally, the chapter is summarised in Sec. 4.7.

4.2 Proposed framework

Our proposed framework can be split into two main stages, namely, event identification and synchronisation, as depicted in Fig. 4.1. For a query video C_q, the set of UGVs, f $C_{k;n}g^{N_k}_{n=1}$, belonging to event, E_k, is identified, then the synchronisation time-shifts, $Dt_{q;1:N_k}$, with reference to the query are estimated. Feature extraction and matching (detailed in Sec. 3.2.1 and Sec. 3.2.2) are the key components of the proposed framework. In order to eliminate false identifications, a validation of the synchronisation time-shifts is performed. A multi-camera visualiser is developed to playback the set of synchronised UGVs belonging to E_k. In this section, we present the proposed framework and the main assumptions.

We extract the chroma feature vector, F_m, using an audio frame size, f_r. Our proposed feature matching strategy maximises the similarity of pairs of overlapping feature vectors F_i and F_j, and provides a histogram representation for the match and non-match recording pairs (see Sec. 3.2.2).

The histogram depicts the occurrence of the value of similarity between F_i and F_j. The frame size, f_r, in feature extraction is important for the design of the event identification and synchronisation framework. By making f_r coarser, we can build an efficient event identification framework to identify the cluster of videos belonging to an event E_k. By refining f_r for the identified cluster E_k, we can estimate the synchronisation time-shift. For event identification, a small value of f_r would make the identification process extremely slow, while a large f_r might not give accurate results.

For video event identification, we assume that the histograms for match and non-match video pairs are separable. For example, when audio signals A_i and A_j from C_i and C_j belong to the same E_k event, matching of their feature vectors F_i and F_j shows strong correlation represented by a high peak in the matching histogram $H_{ij}(Dt)$, otherwise, there is no dominant peak. Unlike existing methods for event identification [20, 76], which used a fixed or mean and standard deviation-based classification threshold to detect the matching recording pairs, we propose an automatic classification threshold strategy. We design a novel descriptor from the histograms for match and non-match video pairs (detailed in Sec. 4.3.1). We learn the classification threshold from the histogram descriptors by training using SVM (detailed in Sec. 4.3.2).

For synchronisation, we assume that the time difference of arrival of a sound is negligible. Two recording devices, $C_{k;i}$ and $C_{k;j}$, observing the same event, E_k, might have a time difference of arrival of sound, e_{ij}, due to their different distances from the sound source [116]. Let the audio signal of the i^{th} video recording be $A_i(t_i^p)$, $t_i^p = \frac{p}{s_n^A}$, $0 \square p < K_n^a$, where p is the index of the audio sample, t_i^p is the time at the p^{th} sample for the i^{th} video recording, A_i is the amplitude of the audio sample at time t_i^p, s_n^a is the audio sampling rate and K_n^a is the total number of audio samples. The estimated time-shift obtained between $C_{k;i}$ and $C_{k;j}$ is

$$Dt_{ij} = t_i^p \square t_j^q + e_{ij}; \tag{4.1}$$

where $e_{ij} = \frac{Dd_{ij}}{u_s}$ is the time difference of arrival, in which $Dd_{ij} = d_i \square d_j$ depicts the distance difference between $C_{k;i}$ and $C_{k;j}$ from the sound source and $u_s = 340$ m/s is the speed of sound. Let us consider that the videos are recorded at a frame rate of $s^v = 25$ fps. The separation allowed between two cameras while staying in a video frame tolerance of $\square 1$ frame ($e_{ij} = 0{:}04$ sec) is $Dd_{ij} = 14$m (metres). In the case of UGVs, Dd_{ij} is unknown, as when sharing these videos on the Internet, the information about the geographical location of the cameras and their distance from the sound source is generally not available. We assume that the cameras recording a particular event lie in the vicinity of each other such that $\Delta d_{ij} < 14$m, thus making e_{ij} negligible.

4.3 Event identification

In this section, we present our video event clustering approach which aims to identify multi-camera UGVs of the same event E_k. The two main blocks of this approach are feature extraction and feature matching (Fig. 4.1). We present the novel histogram descriptor, and propose an approach for learning the classification threshold γ from the match/non-match histograms descriptors of the training video events. Event clustering is then performed followed by the association of a new video C_q to the database.

4.3.1 Histogram descriptor extraction

Let us take $\hat{C} \subset C$ of UGVs such that $\hat{C} = \{\hat{C}_m\}_{m=1}^{\hat{M}}$, where $\hat{M} \leq M$ for training the classification threshold such that these recordings are not included in the test data. The database $\hat{C}$ contains $\hat{M}$ videos for $\hat{E} = \{E_{\hat{k}}\}_{\hat{k}=1}^{\hat{K}}$ events, where $\hat{K} \leq K$, such that we have at least two overlapping videos for each $\hat{E}_k$. For these $\hat{M}$ videos, we extract the features $\{F_m\}_{m=1}^{\hat{M}}$ using frame size f_{r1}. The selection of f_{r1} is done empirically and will be discussed in Sec. 4.6.1. We compute the matching histograms H for all $\hat{M} \times \hat{M}$ video recording pairs (as discussed in Sec. 3.2.2). The matching histograms are given by

$$H = \{H_{ij}(\Delta t)\}, \forall i, j \in [1, \hat{M}]. \quad (4.2)$$

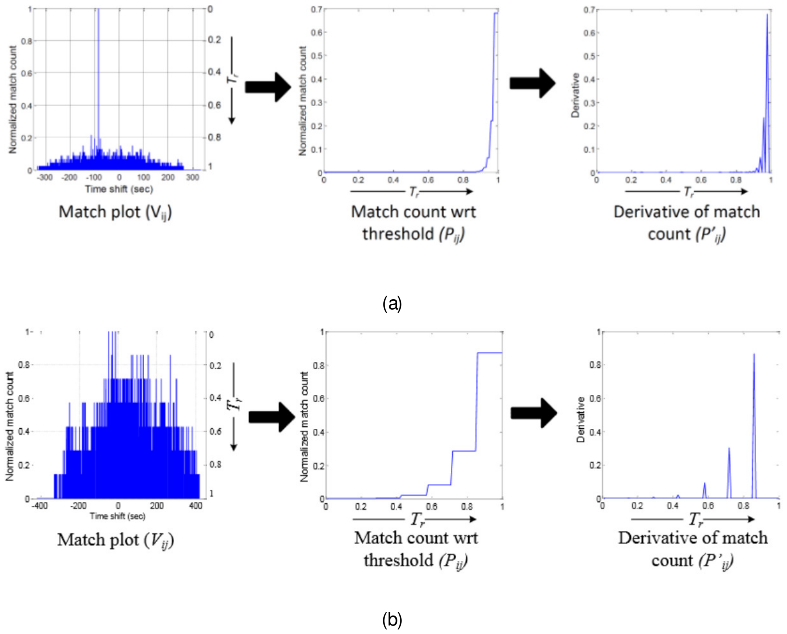
We then compute the delay matrix D for all video recording pairs, such that $D = [\Delta t_{ij}]^{\hat{M} \times \hat{M}}$, where each element of D is given by

$$\Delta t_{ij} = \operatorname*{argmax}_{\Delta t} V_{ij}(\Delta t). \quad (4.3)$$

We propose a method for the extraction of descriptors from histograms H, which are invariant within the match and non-match classes. Using these descriptors, we train a SVM classifier for $\hat{C}$ to obtain the classification threshold γ (Sec. 4.3.2).

From the histogram $H_{ij}(\Delta t)$, we compute the descriptor P'_{ij} by performing a post-processing step to remove the dependency of match count on the time axis (Fig. 4.1). Each histogram $H_{ij}(\Delta t)$ is first normalised with respect to its maximum value at Δt:

$$\hat{H}_{ij}(\Delta t) = \frac{H_{ij}(\Delta t)}{\max_{\Delta t} H_{ij}(\Delta t)}. \quad (4.4)$$



(a)

(b)

Figure 4.2: Post-processing of matching histogram $H_{ij}(Dt)$: (a) example histogram obtained for the match class, and (b) example histogram obtained for the non-match class. Histogram descriptors P^0_{ij} are computed for all $H_{ij}(Dt)$ by scanning from top to bottom using 0:0 □ T_r □ 1:0 and taking their derivative.

A scanning threshold parameter 0 □ T_r □ 1 is then defined, which scans $H_{ij}(Dt)$ from top to bottom counting the number of matches on each incremental step h (where h = 0:01 of T_r). This gives the match count P_{ij} with respect to the scanning threshold parameter T_r making it independent of the time-shifts (Fig. 4.2). The derivative P^0_{ij} which reflects the change in P_{ij} is then computed thus giving a 100□point descriptor of the histogram $H_{ij}(Dt)$. P_{ij} is a step representation which shows the accumulation of the number of matches. By taking its derivative P^0_{ij}, we get a unique representation in which the descriptor only shows high value at the instances of change and remains zero elsewhere. Therefore, the descriptor P^0_{ij} 2 R^{100} is distinguishable for match and non-match classes in the same way as their histogram $H_{ij}(Dt)$ is, but it gives a common representation for all variations of match and non-match classes. Figure 4.2 illustrates the process of the histogram descriptor P^0_{ij} extraction from match and non-match histograms.

4.3.2 Classification threshold

The obtained histogram descriptors P^0_{ij} are rearranged and labelled as belonging to the match and non-match classes for training the classifier. P^0_{ij} are rearranged row-wise to give the set

$$P^0 = f\, P^0_{11}; P^0_{12}; :::; P^0_{\hat{M}\hat{M}}\, g: \tag{4.5}$$

P^0 contains N_p match descriptors and N_n non-match descriptors, where $N_p + N_n = \hat{M} □ \hat{M}$. In order to obtain a compact representation of the data, we use a bag-of-words [73] like approach. We perform k-means clustering for match and non-match class vectors by selecting k_{N_p} and k_{N_n} as the number of clusters which are determined using the elbow method [79]. The returned cluster centres represent the possible variations within a class in the training data which are then considered as the training set. The clustered set of training vectors T belonging to match and non-match classes are given by

$$T = f\,(\bar{P}^0_1; 1); □□□; (\bar{P}^0_k; 1); □□□; (\bar{P}^0_{k_{N_p}}; 1); (\bar{P}^0_1; □1); □□□; \bar{P}^0_k; □1); □□□; (\bar{P}^0_{k_{N_n}}; □1)\,g; \tag{4.6}$$

where $\bar{P}^0_k$ represents the histogram descriptor corresponding to the k^th cluster centre. We use a linearly separable SVM [146] for separating the two classes and computing the classification threshold °. SVM learns ° using the training data T, such that it maximises the distance between the support vectors of the two classes. The learned classification threshold ° is then used to classify and cluster the testing database (Sec. 4.3.3). For each identified cluster E_k, the time-shift estimation and validation is then performed for synchronisation (Sec. 4.4).

4.3.3 Event clustering

In order to identify the group of UGVs that belongs to the same event E_k, we extract the descriptors $P^0_{ij}; 8i; j\ 2\ [1; M]$, such that $\hat{M}$ video recordings used for training are not included. The classification threshold ° is then used to identify overlapping UGVs belonging to the same events. As a result, we get an identification matrix $I = f\, I_{ij}; j\, I_{ij}\ 2\ Z_{[□1;1]}\, g;\ I\ 2\ Z^{M□M}$, which is symmetric. I_{ij} takes the value 1 if an overlapping video is identified, otherwise its value is □1. Our proposed method does not require initialisation by the number of clusters to be identified. The group of identical rows in I corresponds to the videos identified as belonging to the same event E_k. The set of videos are grouped to form an event cluster $E_k = f\, C_{k;n}\, g^{N_k}_{n=1}$. Once the clusters are identified, the longest UGV within each cluster, $\hat{C}_k$, is taken as the representative for each event cluster E_k in order to facilitate overlaps with the rest of the recordings belonging to that cluster.

As a result of event clustering, we obtain the set of representative videos $\hat{C}$ for the set of event clusters E,

$$\hat{C} = \{\hat{C}_k : \forall k \in [1,K]\}. \tag{4.7}$$

4.3.4 Association of new videos to the database

Let C_q be a query video to be assigned to an event. Since we already performed event clustering, instead of matching C_q with C, we perform its matching with $\hat{C}$. The feature vector F_k for all representative video recordings $\hat{C}$ are precomputed using frame size f_{r1}. We compute chroma features F_q for the query video using f_{r1}. The matching histograms H_{kq} and descriptors $P'_{qk} : \forall k \in [1,K]$ are obtained as discussed in Sec. 4.3.1. The descriptors are then mapped on to the classification threshold γ, which identifies the event cluster E_k containing the set of UGVs having the same overlapping event as C_q.

4.4 Time-shift estimation and cluster membership validation

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

Without loss of generality, let us consider $C_{k,1} = \hat{C}_k$ as the reference video with the longest duration in E_k. To achieve high precision for the synchronisation, the feature vectors $\{F_{k,n}\}_{n=1}^{N_k}$ for $\{C_{k,n}\}_{n=1}^{N_k}$ are computed using a frame size of $f_{r2} < f_{r1}$ (as discussed in Sec. 4.2). Feature matching is then performed between all recording pairs ($N_k \times N_k$) to estimate the synchronisation time-shifts, which results in the delay matrix $D = [\Delta t_{ij}]^{N_k \times N_k}$ (Eq. 4.3). The delay matrix D is anti-symmetric ($\Delta t_{ij} = -\Delta t_{ji}$) if all UGVs are partially or completely overlapping. However, if false positive identification occurs the delay matrix D might not be anti-symmetric. The analysis of D is thus required for the validation of the identification results, elimination of any false identifications and for the calculation of consistent time-shifts.

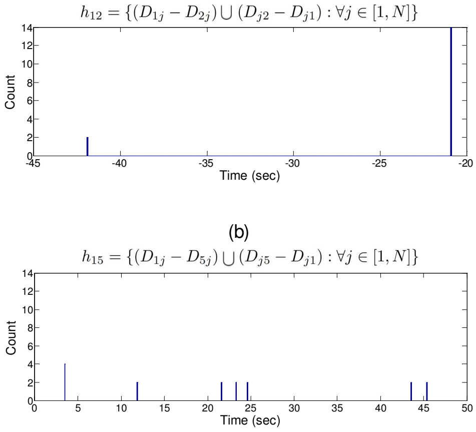
We analyse the delay matrix D using the time-shift validation method of Casanovas and Cavallaro [23] for validating the cluster membership. We generate the histogram $h_{ii'}$ where $i \neq i', \forall i, i' \in [1,N_k]$. The histogram $h_{ii'}$ contains the count for the consistent time-shifts detected between i and i' columns, and i' and i rows of the delay matrix D. This is given by

$$h_{ii'} = \{(D_{ij} - D_{i'j}) \cup (D_{ji'} - D_{ji})\}, \tag{4.8}$$

	$\mathbb{C}_1$	$\mathbb{C}_2$	•	•	•			$\mathbb{C}_N$
$\mathbb{C}_1$	0.00	-20.86	0.22	-20.21	3.54	-6.50	60.05	-4.29
$\mathbb{C}_2$	20.86	0.00	21.15	0.67	45.46	14.36	80.92	16.58
	-0.22	-21.15	0.00	-20.44	43.36	-6.74	59.75	-4.60
•	20.21	-0.67	20.44	0.00	43.52	13.71	80.26	15.91
•	-3.54	-45.46	-43.36	-43.52	0.00	-51.87	48.15	-25.93
•	6.50	-14.36	6.74	-13.71	51.87	0.00	66.57	2.20
	-60.05	-80.92	-59.75	-80.26	-48.15	-66.57	0.00	-64.35
$\mathbb{C}_N$	4.29	-16.58	4.60	-15.91	25.93	-2.20	64.35	0.00

$\mathbb{D}$

(a)



(b)

(c)

Figure 4.3: Cluster membership validation using [23]: (a) example delay matrix for N = 8 belonging to the same event, (b) histogram h_{12} for the time-shift between C_1 and C_2, showing its consistency, (c) histogram h_{15} for the time-shift between C_1 and C_5, showing its inconsistency.

where j 2 [1;N_k]. The returned h_{ij^0} is quantised to the first decimal place for consistency. The most frequently occurring value on this histogram is selected as the consistent time-shift Dt_{ij^0}. A video that does not belong to the same event as that of the other videos contained in the cluster will have no consistency, and this information is used to remove false identifications. Figure 4.3 illustrates this validation process with the help of a delay matrix in which video $C_{k;5}$ is intentionally selected to be different from all other UGVs for the purpose of demonstration.

4.5 Multi-camera visualiser

We developed a multi-camera visualiser using VLC multimedia player library [3], to further validate the obtained results and to coherently playback the identified UGVs. The visualiser

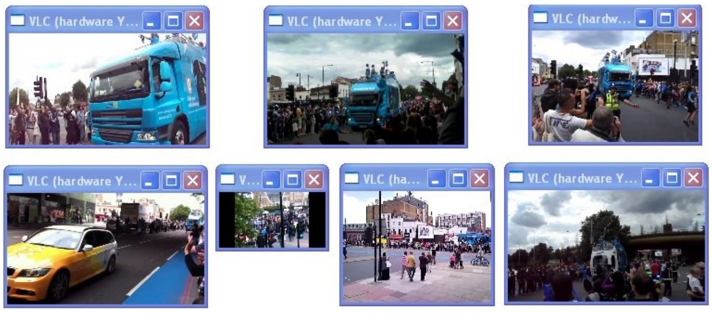


Figure 4.4: Multi-camera visualiser. A snapshot of the synchronised videos of Olympic torch event.

loads multiple video players (equal to the number of identified UGVs), align the videos using the estimated time-shifts, and simultaneously playback them for visualisation. Figure 4.4 shows a snap-shot of the visualiser.

4.6 Results

In this section we present the experimental setup, the validation of the proposed method for video identification and synchronisation, and a comparison with state-of-the-art methods. The dataset used in this experimentation is detailed in Sec. ??.

4.6.1 Experimental setup

For the computation of audio features, the audio signal from a UGV C_i is segmented into overlapping audio frames G_n with hop $h_p = 25\%$ of f_r and frame size $f_{r2} = 0{:}04$ sec for time-shift computation, which gives an accuracy of 0.01 sec for synchronisation. For video identification, a value of $f_{r1} = 3{:}0$ sec was found to be an appropriate compromise between efficiency and accuracy. The energy spectrum of the audio frames is computed on the logarithmic scale, where the minimum and maximum are set to 100Hz and 5000Hz [56]. The computed spectrum energy is then redistributed along the 12 pitch classes (chroma) and matching is performed using the proposed method detailed in Sec. 3.2.2. To compute the classification threshold ° we used a training dataset of 7 events containing 42 UGVs. This dataset gave 1764 matching pairs, out of which 288 belonged to the match class. We trained the classifier by selecting $k_{N_p} = 15$ and $k_{N_n} = 28$ determined using the elbow method for selecting the number of clusters. As a result we obtained

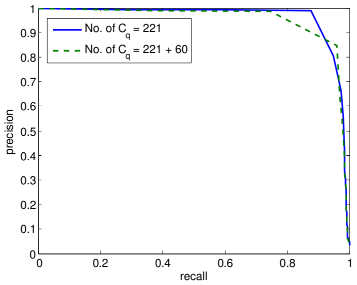


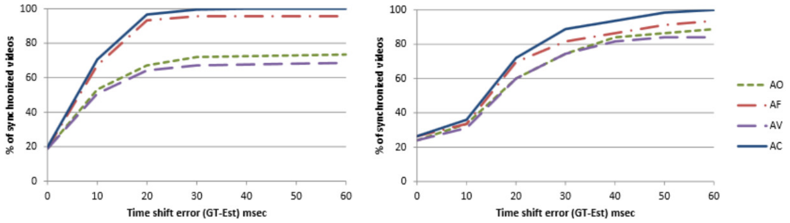
Figure 4.5: Video identification framework result showing the performance for two sets of query (C_q): 41 events containing 221 UGVs (which are contained in the database), and 60 additional videos along with 221 UGVs (where the additional UGVs are not contained in the database).

the classification threshold °.

4.6.2 Discussion and comparisons

For video identification and event clustering, testing is performed on two sets of UGVs: (a) 41 events containing 221 UGVs which forms our database, (b) 60 additional events along with 221 UGVs (of 41 events) where the additional 60 UGVs are not contained in our database. All (a) 221 □ 221 = 48;841 and (b) (221 + 60) □ 221 = 62;101 possible match pairs are computed and the ground-truth for video identification is generated. Figure 4.5 shows the precision-recall curve for the two test sets. High precision is achieved in both test cases with the area under the precision-recall curve to be 0.97 and 0.96, respectively. This shows the robustness of the proposed framework even with the additional UGVs. Video identification is followed by automatic event clustering using which we identified 41 clusters.

To perform synchronisation, we use the complete dataset of 48 events (263 UGVs) for the evaluation, as we are interested in synchronising all the events. The synchronisation results are shown in Fig. 4.6(a). Despite several audio degradations, all videos are synchronised with errors between estimated and ground-truth time-shifts smaller than 0:03 sec. The proposed synchronisation approach is even effective for videos of a short duration (as analysed in Sec. 3.2.3) and fails to correctly show the time-shifts for only one UGV (belonging to Olympic Torch Mile End dataset) out of the 263 in the test. The error is due to the recording device malfunctioning and not capturing the audio signal for most of the time during recording. Time-shift validation is also performed in order to verify that the obtained cluster of recordings belongs to the same event.



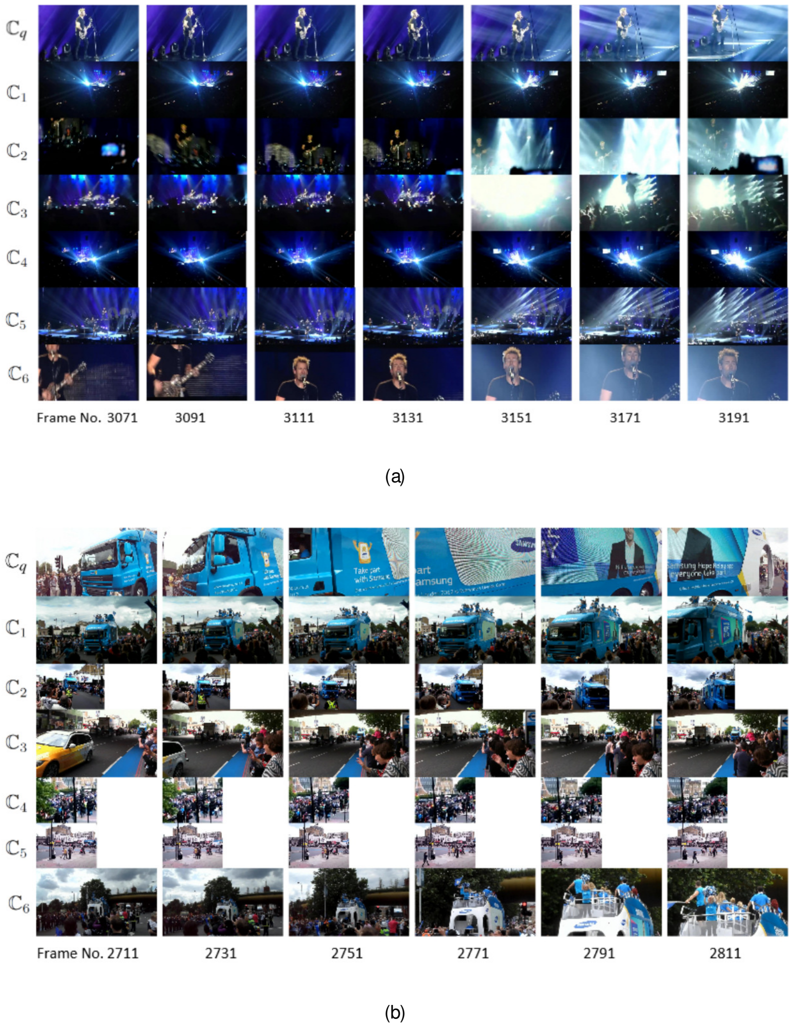
(a) (b)

Figure 4.6: Comparison results showing the percentage of synchronised videos versus time-shift error with respect to the ground truth. (a) Synchronisation results on the whole dataset (Tab. A.2). (b) Synchronisation results for the dataset used in [23, 132]. Key: AO indicates the audio onset based method [132]; AF indicates the audio fingerprinting method [132]; AV indicates the audio-visual event method [23]; AC indicates the proposed method.

In order to further validate our proposed Audio Chroma (AC) based synchronisation method, we compare it with state-of-the-art methods based on Audio Onset (AO) [132], Audio Fingerprinting (AF) [132] and Audio-Visual Event (AV) [23] using our dataset (Fig. 4.6(b)). AO and AV are comparable, while at times AV gave slightly worse results than AO. Since these two methods are highly sensitive to audio degradations, they failed to synchronise a large number of UGVs. Likewise, Audio Fingerprinting (AF) [132] is robust to ambient noise but failed to give the correct result for some recordings containing reverberations and channel noise. Furthermore, AF failed to synchronise UGVs of a short duration (< 30 sec). AC outperformed the other three methods as it was able to synchronise 262 out of 263 UGVs, followed by AF, giving an overall accuracy of 99:62% and 94:79%, respectively.

To have a fair comparison with the state of the art, we also perform testing with the dataset used in [23, 132] (Fig. 4.6(b)). The same trend can be observed as for our dataset: the results obtained with AC and AF are comparable, but AC outperforms the other methods. The best overall performance is achieved by AC, followed by AF, AO and AV.

The association and synchronisation for a concert (Nickelback_Event1) and the Olympic torch (OlympicTorchMileEnd) event are shown in Fig. 4.7, where row one shows C_q and the identified cluster videos are shown in the subsequent rows. Each column represents the synchronised frame for these video recordings. Note the different visual quality (C_4 and C_5 in Fig. 4.7(b)), variations in the field of views (C_1 and C_4 shows far field of views as compared to C_q and C_6 in Fig. 4.7(a)), lighting (C_2 and C_3 in Fig. 4.7(a)) and camera motion (C_6 in Fig. 4.7(a)



(a)

(b)

Figure 4.7: The association and synchronisation result for (a) a concert (Nickelback_Event1 as named in Tab. A.2) and (b) the Olympic torch (OlympicTorchMileEnd as named in Tab. A.2) event. Row 1 represents a snapshot frame from the query video. Each row represents a different video from the identified cluster event. Each column corresponds to temporally aligned frames from the videos.

and (b) showing zooming in motion) in the snapshot frames.

To test the robustness of the proposed framework, association is also performed using similar UGVs (using an additional dataset of 60 UGVs, which is detailed in Sec. ??). Though depicting similar events but with no time overlap, no event cluster is identified when performing association with these additional UGVs. This is also shown in Fig. 4.5, which further validates the robustness of our framework.

4.7 Summary

In this chapter, we presented an automatic identification and synchronisation framework for multi-camera UGVs and query-by-example video event search. The proposed framework used audio chroma feature to cluster UGVs belonging to the same event and to estimate their relative time-shifts. Coarser frame size for audio feature extraction facilitated in efficient video identification, while refining it for the identified cluster gave precise time-shift estimation. We designed a novel descriptor from the histograms for match and non-match video pairs that gave a discriminant representation for match and non-match classes. Unlike existing identification methods [20, 76], we proposed an automatically determined classification threshold using the novel descriptor for clustering and association of new incoming videos. The classification threshold is trained using a relatively smaller dataset, and testing for the video identification is performed on unseen event dataset. We demonstrated the robustness of the proposed method to audio degradations including high ambient and channel noise, and discussed a comparative analysis with existing state-of-the-art methods.

The goal of this chapter was to identify all UGVs belonging to the same event and to organise the identified videos on a common timeline. Once organised, applications can be developed for better understanding the event, localising the region of interest, multi-camera video composition and summarisation [122, 133, 35]. In the next chapter, we propose a framework for composing a time continuous video from multi-camera synchronised UGVs by considering audio and visual qualities of UGVs (as detailed in Sec. 3.4 and 3.5). Gyroscope data is useful for the estimation of camera motions with reduced computational cost as compared to the visual data (Sec 3.3). Therefore, we design a gyro-based video composition framework as well, by considering audio and gyro-based qualities of UGVs.

Chapter 5

Video composition from multi-camera user-generated videos

5.1 Introduction

Video composition from multi-camera UGVs of the same event aims at generating a coherent and time continuous video providing a multi-view experience [122, 133, J2]. Video composition may involve discarding low-quality and less interesting visual segments. Among a pair of time overlapping visual segments, the segment recorded from a stable hand-held device with better visual quality can be considered as interesting. The composed video contains non-overlapping segments of multi-camera UGVs selected to improve the view diversity. We define view diversity as the introduction of variety of views in the camera selection process in order to enrich the content of the composed video. The perceived audio-visual quality is a key factor which makes the content enjoyable and interesting to playback [104]. Existing video composition methods [122, 133] for UGVs perform visual quality analysis and manual cut-point selection. We propose an automatic video composition framework (ViComp) for UGVs recorded from different viewpoints in an event [J2]. ViComp exploits visual quality and view diversity to select segments using a rank-based camera selection. ViComp maintains audio uniformity and exploits audio-content analysis to automatically select cut-points for visual segments generation. We design a subjective test for the comparative evaluation of the proposed framework. Gyroscope data when captured coherently with the video facilitates in camera motion analysis [C1]. Therefore, we design a gyro-based assessment score for qualifying the visual quality and used it to develop a gyro-based

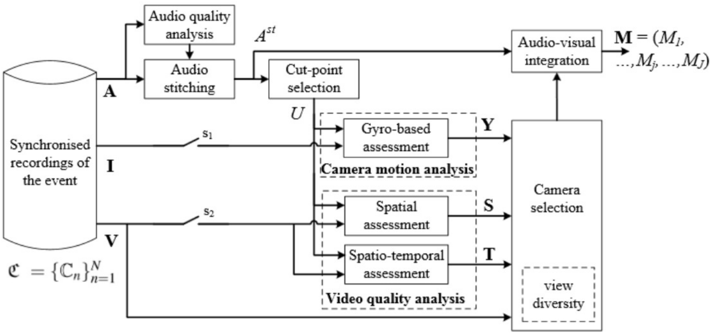


Figure 5.1: Block diagram of the proposed multi-camera UGV composition framework. The audio signals are analysed for audio stitching, followed by suitable cut-points selection. For ViComp (s_1 = OFF, s_2 = ON), spatial and spatio-temporal assessments are computed by analysing video quality. For ViCompG (s_1 = ON; s_2 = OFF), gyro-based assessment is computed by considering camera motion analysis. ViComp and ViCompG integrate their respective assessments with the view diversity for the designing of the rank-based camera-selection method.

video composition framework (ViCompG).

In this chapter, we first present the proposed video composition frameworks, namely ViComp (Sec. 5.2) and ViCompG (Sec. 5.3). Subjective test designed for the evaluation of the proposed frameworks is detailed in Sec. 5.4. Experimental evaluation of the gyro-based camera motion detection is presented in Sec. 5.5. This is followed by experimental validation and analysis of ViComp and ViCompG in Sec. 5.6. The chapter is summarised in Sec. 5.7.

5.2 ViComp: Audio and visual-based video composition

We develop ViComp, a video composition framework, by considering audio and visual feature analyses (as described in Sec. 3.4 and Sec. 3.5). The block diagram of the proposed framework is shown in Fig. 5.1. To maintain audio uniformity, we propose a method for audio stitching by ranking the set of audio signals, A, from an event based on their quality. This results in coherent audio, A^{st}, for the coverage duration, D_C, of the event. An automatic cut-point selection method is then designed by analysing the change in the dynamics of the audio features (as detailed in Sec. 3.4.3). The selected cut-points, U, are used for the segmentation of the set of video recordings, V. We use spatial, S, and spatio-temporal, T, scores as the quality assessment measures for each segment. To compose the video, we rank the segments using visual quality and impose the view diversity condition. Detailed below is the proposed view diversity condition, followed by

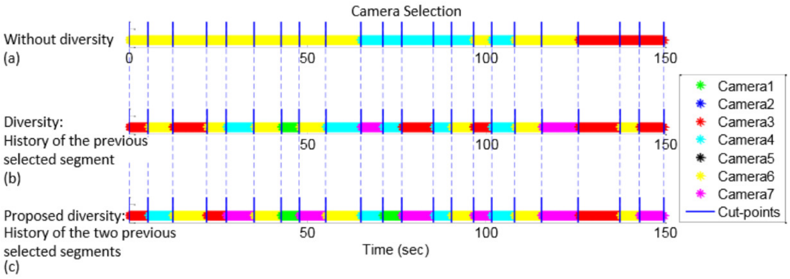


Figure 5.2: View diversity illustration. Camera selection is shown for three cases: (a) No diversity condition is applied. (b) History of the previous selected segment is considered for the diversity. (c) Proposed view diversity condition in which history of the previous two selected segments is considered.

the rank-based camera selection method.

5.2.1 View diversity

View diversity enhances the viewing experience and is a component of professionally edited videos [19, 157]. UGVs of the same event differ in viewing angles and distances from the object of interest. Therefore, we assume that if at least the previous two consecutive selected cameras are different from the current selection, sufficient view diversity is achieved. A video selected for the segment M_j is not the one selected for the previous two segments M_{j-1} and M_{j-2} provided that we at least have three video recordings of the event at that time instant. This is given by

$$M_j \in C_n \mid M_{j-1} \notin C_n \ \& \ M_{j-2} \notin C_n. \tag{5.1}$$

Figure 5.2 shows an illustration of the proposed view diversity condition (Fig. 5.2(c)) in comparison to when no diversity (Fig. 5.2(a)), or history of the previous selected segment (Fig. 5.2(b)) is applied for the camera selection. Without view diversity, camera selection is merely a selection of the top ranked cameras, and does not introduce variety of views. In the proposed view diversity condition, switching between three or more cameras take place by considering their ranks. The rank-based camera selection strategy is presented in the following section.

5.2.2 Rank-based camera selection

In order to construct a camera selection strategy, we analyse the spatial, S, and spatio-temporal, T, assessments within each cut-point segment, u_j, while considering the proposed view diversity condition. We analyse the segment v_{nj} for all N cameras by using both spatial, S_n, and spatio-temporal, T_n, quality scores. We first perform the best camera selection independently with respect to the S_n and T_n scores, and store the selected camera indices in $Q^S = \{Q^S_j\}_{j=1}^J$ and $Q^T = \{Q^T\}_{j=1}^J$, respectively. For the cut-point segment u_j, the selected camera indices for S are given by $Q^S_j = (Q^S_{j1}, \cdots, Q^S_{jk}, \cdots, Q^S_{jK^v_j}) \in u_j$, where K^v_j is the total number of samples in u_j. The same is applicable for Q^T_j. We then compute the spatial score-based normalised occurrence of camera C_n in each u_j as

$$O^S_{nj} = \frac{\sum_{k=1}^{K^v_j} Q^S_{jk}}{K^v_j} \tag{5.2}$$

where $Q^S_{jk} \in C_n$. By varying $1 \leq n \leq N$, we get the normalised occurrence for all the cameras in u_j. Similarly, we compute O^T_{nj} for Q^T_j, and arrange O^S_{nj} and O^T_{nj} in descending order to get the rank vectors R^S_j and R^T_j, respectively, for all C_n. The spatial and spatio-temporal rank matrices are given by $R^S = [R^S_1, \cdots, R^S_j, \cdots, R^S_J]$ and $R^T = [R^T_1, \cdots, R^T_j, \cdots, R^T_J]$, respectively. We compute the combined rank matrix R^C using R^S and R^T that ensures that the segments with better visual quality always get higher ranks. The combined rank vector R^C_j for u_j is computed by combining the unique stable values from R^S_j and R^T_j. At cut-point segment u_j, we assign the top combined rank $R^C_j(1)$ to M_j followed by imposing the proposed view diversity condition. The complete algorithm for ViComp is detailed in Algorithm 2.

5.3 ViCompG: Audio and gyro-based video composition

ViCompG replaces the visual quality assessments (S, T) with a gyro-based quality assessment, $Y = \{Y_n\}_{n=1}^N$, that aims at obtaining the quality scores from the gyroscope data, G, for the set of synchronised UGVs, C (Fig. 5.1). Each Y_n is given by $Y_n = (y_{n1}, \cdots, y_{ni}, \cdots, y_{nl_g})$, such that the first sample corresponds to the first recorded gyroscope sample in C and the last sample, l_g, corresponds to the last gyroscope sample in C.

Unintentional camera motions influence the perceived quality of UGVs [151]. For example, fast pan, tilt and shake results in blurred frames. In Sec. 3.3, we utilised gyroscope data recorded simultaneously with the video for camera motion analysis. We exploit these findings to obtain the gyro-based quality assessment, Y, for UGVs. The magnitude of the gyroscope data, G_n, for C_n video recording is given by

$$|G_n| = \sqrt{(G^2_{nx}) + (G^2_{ny}) + (G^2_{nz})}; \tag{5.3}$$

```
Input: $l_{min}$, $l_{max}$, A, V, N                % N is the number of UGVs
Output: (M)
$R^{SR} \leftarrow$ Audio ranking (A)
$A^{st} \leftarrow$ Audio stitching (A, $R^{SR}$)
(U, J) $\leftarrow$ Cut-point selection (A, $R^{SR}$)    % J is the number of segments
S $\leftarrow$ Spatial assessment (V)
T $\leftarrow$ Spatio-temporal assessment (V)
$R^S \leftarrow$ Spatial rank matrix (S, U, J)
$R^T \leftarrow$ Spatio-temporal rank matrix (T, U, J)
$R^C \leftarrow$ Unique rank ($R^S$, $R^T$)
for j $\leftarrow$ 1 to J do
    if j = 1 then
        $M_j = R^C_j(1)$                    % first segment selection
    else
        $M_j = R^C_j(1)$
        if $M_j = M_{j-1}$ & $R^C_j(2) \neq 0$ then
            $M_j = R^C_j(2)$                % diversity check provided $N \geq 2$
        end
    end
    if $(j - 2) > 0$ then
        if $M_j = M_{j-1}$ & $R^C_j(3) \neq 0$ then
            $M_j = R^C_j(3)$                % diversity check provided $N \geq 3$
        end
    end
end
```

Algorithm 2: The algorithm for ViComp. The rank-based camera selection method is described in detail.

where G_{nx}, G_{ny} and G_{nz} are the gyroscope signals with respect to the x, y and z axes, respectively for the C_n recording. In th presence of camera motions, the magnitude of gyroscope is not zero. The higher the magnitude, the lower is the perceived visual quality. To normalise the magnitude for all C_n in C, and to compute the gyro-based assessment score, Y_n, we perform min-max normalisation followed by computing the inverse. Let the minimum, G_{min}, and maximum,

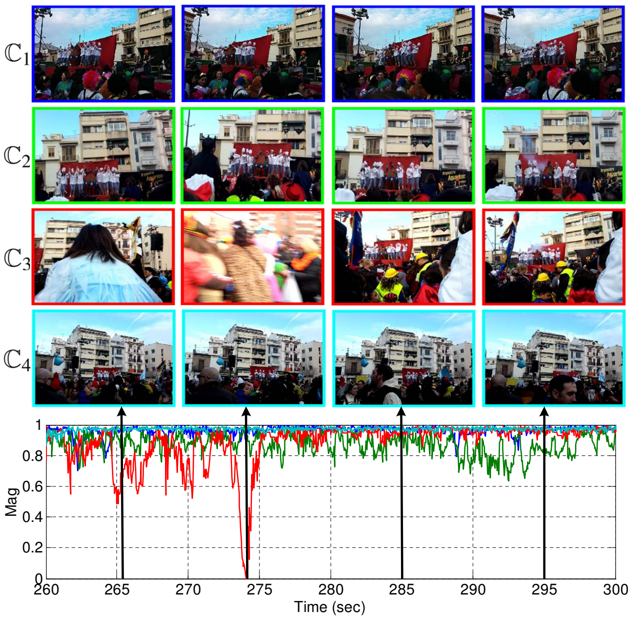


Figure 5.3: Visualisation for gyro-based assessment score, Y, for the event titled Caramel_Event2. The snap-shots of each recording at four time instances (marked by the black upward arrows) are shown.

G_{min}, magnitude values be

$$G_{min} = \min(|G_1|, \cdots, |G_n|, \cdots, |G_N|), \tag{5.4}$$

$$G_{max} = \max(|G_1|, \cdots, |G_n|, \cdots, |G_N|), \tag{5.5}$$

respectively. Then the gyro-based assessment score is computed as

$$Y_n = \frac{G_{max} - |G_n|}{G_{max} - G_{min}}. \tag{5.6}$$

The visualisation of Y is shown in Fig. 5.3 for the event titled Caramel_Event2 that comprises four UGV recordings. At time 274s, motion blur occurred in C_3 that lowered the score. The intentional dance motion while recording in C_2 lowered the score due to shake. Recordings from C_1 and C_4 are comparatively stable with respect to C_2 and C_3.

5.4 Subjective test design

A subjective test is designed to analyse the overall quality of the proposed ViComp and ViCompG methods in comparison with Firstfit [133], MoViMash [122], ViCompCD (Sec. 5.6.1) and ViRand (Sec. 5.6.1). As there are many ways of showing videos to subjects in order to record their assessment, the ITU-R recommendation [70] presented four standardised methods for the subjective video-quality assessment. We selected Pair Comparison (PC) [70]-like method for analysing 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 different methods one after another and asked the subject to provide a comparative rank from the best to the worst quality video. The subjects 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 60s. 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[1] 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.

We conducted a survey on the quality of the generated videos obtained in pair of three methods (Table 5.2). The null and alternate hypothesis are formulated as

- H_o = There is no significant difference among the videos generated by the three different methods.
- H_a = There is a significant difference among the videos generated by the three different 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. In order to test the consistency in ranking patterns, we used the Friedman Two-Way ANOVA by ranks [69]. In the Friedman Two-Way ANOVA test, the data is 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

[1] http://www.eecs.qmul.ac.uk/~andrea/vicomp

statistic (X^2) and p-value are then computed for the recorded data for the analysis (Sec. 5.6.2).

5.5 Experimental evaluation of gyro-based camera motion detection

In this section, we present the experimental results of the proposed gyro-based camera motion detection (CMDG) method (detailed in Sec. 3.3) and compare it with an existing visual [141, 21] and inertial sensor [35] based methods. The multi-modal dataset used for the validation of the proposed method is detailed in Sec. A.2.

For gyro-visual synchronisation, we down-sample the gyroscope data to the same frame rate as that of the video. The obtained delay is in seconds (s) and is applied to the gyroscope data (at the original sampling rate) to align it with the video. The proposed CMDG is then applied for pan, tilt and shake detection.

The results for gyro-visual synchronisation are shown in Fig. 5.4. Acceptable delays are obtained for all UGVs with an absolute error of 0.7s between the GT and the estimate. This error is mainly due to the imprecise GT labels as it was difficult to manually observe a coherent motion both in the video and gyroscope data for labelling. By jointly visualising the synchronised data, we cross-validated the correctness of the obtained results. LPC used for synchronisation (see Sec. 3.3.1) is dependent on the change of illumination. Although the illumination is extremely low in some LB recordings (e.g. fireworks) that resulted is low magnitude of $L_x(t)$ and $L_y(t)$ and inaccurate camera motion detection, correlation existed between the gyroscope and visual data. A slight clue of brightness (e.g. exploding fireworks) is sufficient for establishing the correlation. Thus, acceptable delay is achieved even in the presence of slight camera motion. To investigate the robustness of gyro-visual synchronisation, we varied the overlap duration between the gyroscope and visual data for all UGVs. OverlapN denotes that the complete visual data and only N% of the duration of the gyroscope data are used. For Overlap80, Overlap60, Overlap40 and Overlap20, the percentage of synchronised recordings are 91%, 87%, 78% and 48%, respectively (see Fig. 5.4). Note that the visual quality and frame rate are low in some of the night-time recordings, which affect $L_x(t)$ and $L_y(t)$, and decrease the performance when the overlap is decreased.

For the evaluation of the proposed CMDG, we analyse its performance with respect to the GT. To select a and b for pan, tilt and shake detection, we analysed the effect of varying these parameters on the detection results (Fig. 5.5). At a = 30^{o}, the best F_1-score of 0:94 for pan and tilt, and at b = 0:06, the best F_1-score of 0:85 for shake were achieved and selected for the

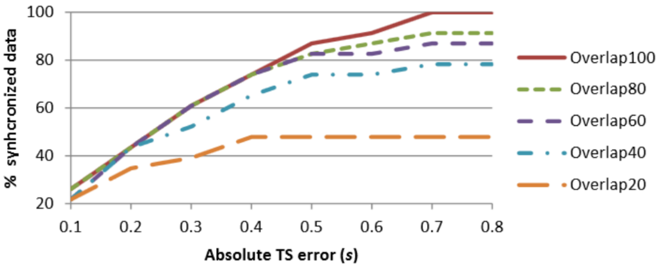
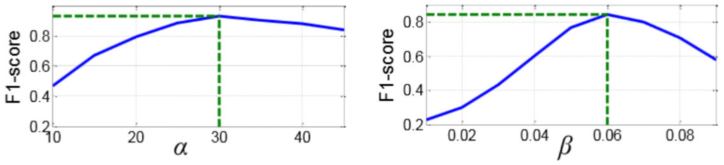


Figure 5.4: Gyroscope-visual synchronization. % of synchronized multi-modal data w.r.t the absolute time-shift error. OverlapN means N% of the duration of the gyroscope is used.

experimentation. We compare the proposed CMDG with an existing visual [141, 21] (referred as VISUAL) and inertial sensor-based [35] (referred as ISENSOR) methods (shown in Table 5.1). In order to investigate their performance, we divided the UGVs into HB and LB recordings having total durations of 30 mins and 40 mins, respectively. To have a fair comparison, the parameters within the VISUAL and ISENSOR are adjusted to give the best possible results. In our dataset, most events of interest existed in the latitudinal plane (e.g. singer, crowd, parade), with the exception of few that existed in the longitudinal plane (e.g. fireworks, flying balloons), resulting in fewer tilt samples (see Table 5.1). CMDG outperformed the existing methods giving the F_1-score of 94%, 82% and 83% for $P_d(t)$, $T_d(t)$ and $S_d(t)$, respectively, for the HB recordings, and 93%, 85% and 86% for the LB recordings. VISUAL and ISENSOR are the second best for the HB and LB recordings, respectively.

VISUAL is effected by the motion of objects and light conditions, thus reducing its performance in LB recordings as compared to CMDG and ISENSOR, which are independent of these factors. ISENSOR is designed using compass and accelerometer, and is effected by magnetic noise, low sampling rate and unfiltered processing, resulting in false detections. CMDG gives a better solution for camera motion detection because of the use of more accurate sensor (gyroscope), and inclusion of the post-processing stage that suppresses the outliers. Pan signals from CMDG and VISUAL are comparable in HB recordings. However, ISENSOR is less accurate due to low sampling rate (of 10Hz) of the compass [35]. Increasing the sampling rate to 50Hz increases the effect of noise, and makes the derivative of compass signal ineffective.



(a) (b)

Figure 5.5: F_1-score of the proposed CMDG method with respect to the varying values of (a) a and (b) b .

Table 5.1: Results for CMDG and its comparison with a VISUAL [141, 21] and ISENSOR [35] methods. Key: HB: high brightness recordings; LB: low brightness recordings; TP: true positive; FP: false positive; P - precision; R - recall; F_1 - F_1 score.

		Pan						Tilt					
Method	Type	GT	TP	FP	P	R	F_1	GT	TP	FP	P	R	F_1
CMDG	HB	294	272	11	0.96	0.93	0.94	36	29	6	0.83	0.81	0.82
VISUAL [141, 21]			217	64	0.77	0.74	0.75		19	42	0.31	0.53	0.39
ISENSOR [35]			175	52	0.77	0.60	0.67		14	46	0.23	0.39	0.29
CMDG	LB	123	117	12	0.91	0.95	0.93	49	41	7	0.85	0.84	0.85
VISUAL [141, 21]			31	44	0.41	0.25	0.31		10	48	0.17	0.20	0.19
ISENSOR [35]			44	40	0.52	0.36	0.43		23	24	0.49	0.47	0.48

(a) Pan and tilt detections in HB and LB recordings

		Shake					
Method	Type	GT	TP	FP	P	R	F_1
CMDG	HB	389	365	129	0.74	0.94	0.83
VISUAL [141, 21]			260	118	0.69	0.67	0.68
ISENSOR [35]			188	93	0.67	0.48	0.56
CMDG	LB	272	235	37	0.86	0.86	0.86
VISUAL [141, 21]			200	606	0.25	0.74	0.37
ISENSOR [35]			213	129	0.62	0.78	0.69

(b) Shake detection in HB and LB recordings

5.6 Experimental validation of video composition

We compare the proposed ViComp with Firstfit [133], MoViMash [122], ViRand, and ViCompCD (see Sec. 5.6.1) using the designed subjective test (Sec. 5.4). The performance of ViCompG is tested against ViComp and ViRand. A dataset of 16 events (105 UGVs) is used to for the

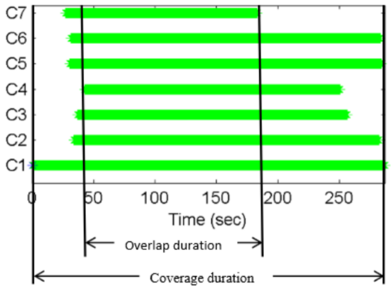


Figure 5.6: Example timeline for a dataset that illustrates the definition of the coverage (D_c) and overlap (D_o) durations.

evaluation of the proposed frameworks. The dataset is detailed in Sec. A.3.

5.6.1 Experimental setup

The UGVs are pre-processed before feeding into the ViComp and ViCompG frameworks as the video frame rate (s^v) and frame size are varying among the UGVs of the same event. All UGVs have been re-sampled to 25 fps using VirtualDub [86]. All frames have been re-scaled to the same size for all the videos before camera selection. Also, all UGVs belonging to an event have been synchronised to a common timeline using [J1]. For the selection of suitable cut-points, we fixed the value of l_{min} and l_{max} to 3 and 10s, respectively (Sec. 3.4.4). For the evaluation test, we used the overlap duration (as shown in Table A.4) that is the duration for which all UGVs in an event are available (as shown in Fig. 5.6). The overlap duration has been used as opposed to the coverage duration to avoid monotonic camera views that might occur when the recording is available from one camera only.

For comparison, we implemented two more strategies, ViRand, and ViCompCD. In ViRand, the visual 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 cluster 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 organised into cluster-1 and cluster-2, where cluster-1 contains the indices of all views similar to the last frame $(i - 1)$ of the previously selected segments M_{j-1}, and cluster-2 contains the indices of all the dissimilar views. At a time instant i, we apply the

Harris affine detector [106] to extract affine invariant regions followed by applying the Scale Invariant Feature Transform (SIFT) descriptor to extract features $F_n^{SF}(i) \in \mathbb{R}^{K_n^{SF} \times 128}$, where K_n^{SF} is the number of extracted features from the i^{th} frame in the n^{th} camera. We used this detector as it is capable of identifying similar regions in pairs of video frames captured from different viewpoints. For a camera $C_{n'}$, we calculate its feature matching with the features $F_n^{SF}(i)$ of all other cameras. The match count between current $C_{n'}$ and all C_n at the i^{th} frame is given by $L(i) = [l_{n'1}(i), \cdots, l_{n'n}(i), \cdots, l_{n'N}(i)]^T$. The highest number of matches is obtained when $n' = n$. We make this value $l_{n'n'}(i)$ 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 another camera recordings. Next, we apply k-means clustering by initialising two clusters such that cluster-1 is with the highest mean value. Ideally, this ensures that cluster-1 always contains frames with similar camera views 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 cluster-1 for which the combined rank R_j^C (in the j^{th} cut-point segment) is high and satisfies the proposed diversity condition. Figure 5.7 shows an example of CD strategy. Matching is performed between last frame of previously selected camera C_7 and all C_n, as a result frames similar to C_7 form the cluster-1 while dissimilar frames form the cluster-2.

The validation is performed by conducting five experiments as detailed in Table 5.2. In the first and the second experiments, we selected Event1-4 and Event5-8, 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 mashup obtained using Firstfit [133] and MoViMash [122] were made available by their authors for Event9-11 and Event12-13, respectively. In the third experiment, we used Event9-11 and tested ViComp, ViCompCD and FirstFit [133]. In the fourth experiment, we used Event12-13 and tested ViComp, ViCompCD and MoViMash [122]. Finally, in the fifth experiment, we used Event14-16 for which the inertial sensor data is available, and tested ViComp, ViCompG and ViRand. The audio in Firstfit [133] is varying and discontinuous which may negatively influence the subject's decision while ranking [15]. In order to remove this bias, we used the same audio track that we obtained from audio stitching for all methods.

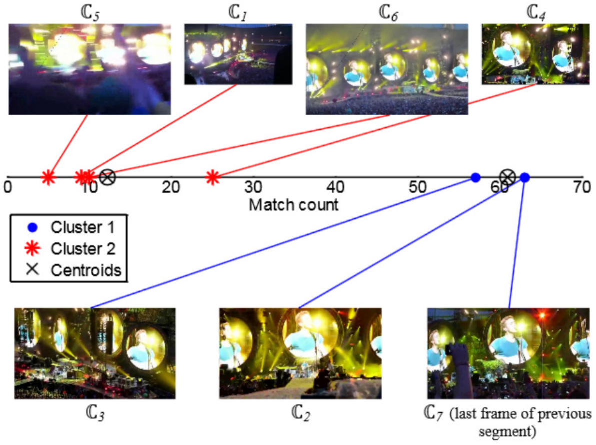


Figure 5.7: Clustering-based diversity example. C_7 is the last frame of the previously selected segment which is matched with all C_n. This process divides the cameras into similar (C_1) and dissimilar (C_2) clusters.

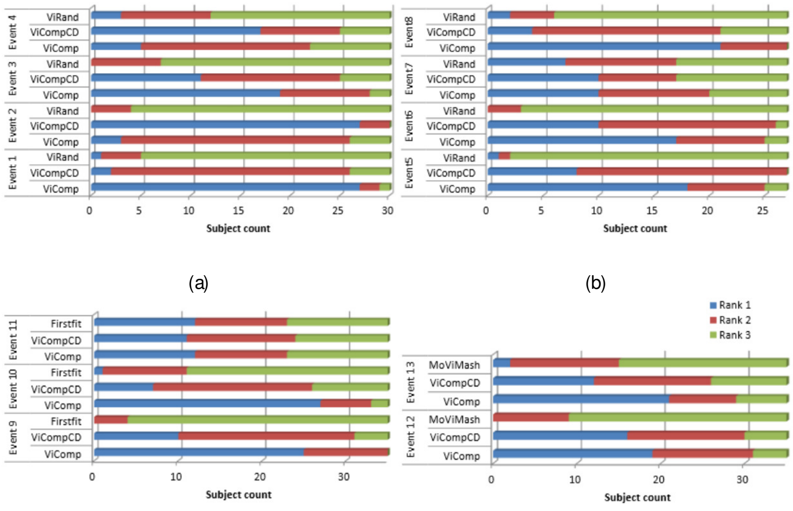
Table 5.2: Details of the conducted subjective experiments and their evaluation. Median age of subjects in all the experiments came out to be approx. 30 years. Key: Exp. - Experiment number; M - Male subjects; F - Female subjects; X^2 - Chi-square statistic.

Exp.	Events	Methods under test						Gender		X^2	p-value
		ViComp (proposed)	ViCompG (proposed)	ViCompCD	ViRand	Firstfit	MoViMash	M	F		
1	1-4	x		x	x			21	9	120:46	6:9e^{-27}
2	5-8	x		x	x			18	9	113:56	2:2e^{-25}
3	9-11	x		x		x		26	9	56:11	6:6e^{-13}
4	12-13	x		x			x	26	9	51:54	6:4e^{-12}
5	14-16	x	x		x			13	6	59:75	1:1e^{-13}

5.6.2 Subjective test results

In total 146 subjects took part in the five experiments. The age of the subjects who took part in the first and second experiments ranged from 19-50 years (median 29.5 years), for the third and fourth experiments ranged from 23 to 53 years (median 30 years) and for the fifth experiment ranged from 16 to 39 years (median 28 years). The recorded ranks for the experiments are presented in Fig. 5.8 and Fig. 5.10(a).

The Friedman Chi-square statistic (X^2) and p-value are computed for all five experiments and



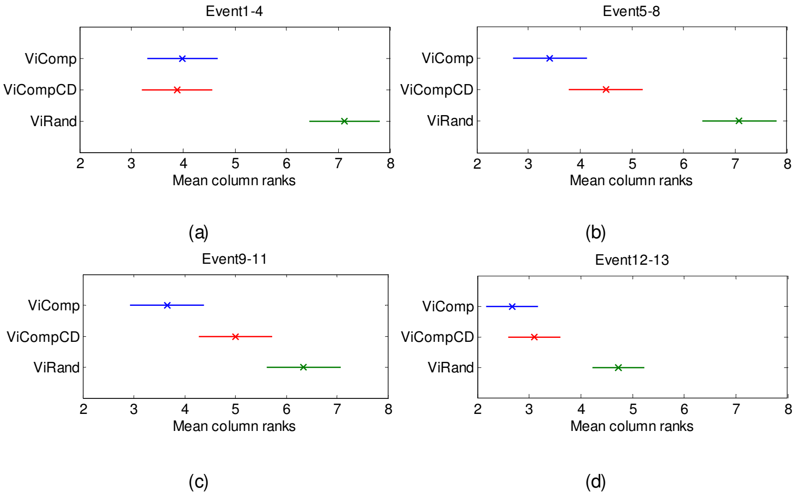
(a) (b)

(c) (d)

Figure 5.8: 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 [133] for the Events from Firstfit, (d) Experiment 4: Ranks assigned by subjects for the videos composed by ViComp (proposed), ViCompCD and MoViMash [122] for the Events from MoViMash.

are detailed in Table 5.2. All results are statistically significant as the p-values are close to zero, hence we can reject the null hypothesis (See. 5.4). 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 [61] for the five experiments. For the first two experiments (Fig. 5.9(a)-(b)), the proposed ViComp and ViCompCD appeared to be significantly different from the ViRand. For the third experiment (Fig. 5.9(c)), the mean column rank of the ViComp was significantly different from the Firstfit [133]. 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 (Sec. 5.6.3). For the fourth experiment (Fig. 5.9(d)), the proposed ViComp and ViCompCD performed better than MoViMash [122]. For the fifth experiment (Fig. 5.10(b)), the mean column



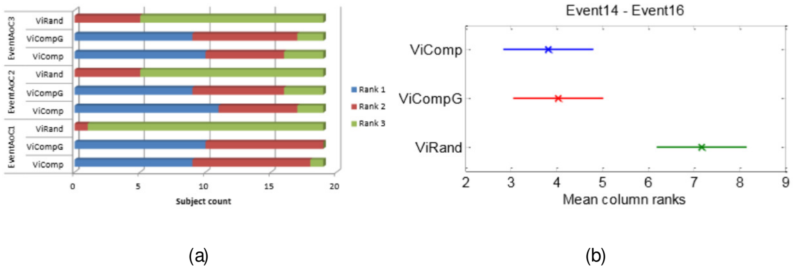
(a)

(b)

(c)

(d)

Figure 5.9: The corresponding multiple comparison of mean column ranks for subjective test shown in Fig. 5.8: Comparison is shown for the (a) Experiment 1 (Event1-4), (b) Experiment 2 (Event5-8), (c) Experiment 3 (Event9-11), and (d) Experiment 4 (Event12-13).



(a)

(b)

Figure 5.10: Subjective evaluation results are shown for the fifth experiment. (a) Ranks assigned by subjects for the videos composed by ViComp (proposed), ViCompG (proposed) and ViRand for the Caramel events (Event14-16). (b) Multiple comparison test.

ranks of ViCompG and ViComp were significantly different from the ViRand one.

5.6.3 Discussion and comparisons

The subjective evaluation shows that the quality of ViComp is comparable to ViCompCD in some events but overall ViComp outperformed all the other methods (Fig. 5.8 and Fig. 5.9). Moreover, the quality of ViComp and ViCompG in all three tested events is comparable.

The ranks for ViComp and ViCompCD were comparable in the first experiment, while Vi-

Rand was ranked low (Fig. 5.8(a)). Only for Event4, ViRand received a sufficiently high rank but not higher than ViComp and ViCompCD. This is because Event4 contained 5 UGVs, all of them having comparable visual quality, which made difficult for a subject to take a decision.

For the second experiment (Fig. 5.8(b)), ViComp and ViCompCD outperformed ViRand for Event5, 6 and 8. An interesting case is the one of Event7, 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 were either from far field of view (with less shake) or near field of view (with high shake). The composed videos were not interesting as far fields of view give less information about the event and near fields of view seemed unpleasant because of high camera-shake.

For the third experiment (Fig. 5.8(c)), ViComp outperformed the other two methods. All three events used in this experiment contained 4-5 overlapping UGVs, having low resolution (320×240 pixels). Subject's agreement was not achieved for Event11 because of the poor visual-quality (jerky and shake,compression artifacts) of all 4 UGVs contained in this event.

For the fourth experiment (see Fig. 5.8(d)), the two events under analysis contained 12 UGVs of comparable quality that were recorded from near field, which resulted in comparable ranks for both ViComp and ViCompCD. MoViMash was ranked low because UGVs containing high brightness (and poor visual quality) were selected as a consequence of learning the field-of-view distributions. Also, sometimes the length of a selected visual segment in MoViMash was 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.

For the fifth experiment (Fig. 5.10), ViComp and ViCompG both gave comparable results for all three events under consideration. ViComp suppressed the low quality segments by considering spatial and spatio-temporal scores, while ViCompG solely considered spatio-temporal score from the gyro-based assessment. The composed videos from both ViComp and ViCompG mainly contained stable segments. The subjects, therefore found difficulty in assigning 1st and 2nd ranks to these composed videos. ViCompG can be a preferred choice when the inertial sensor data is available due to the less amount of data that requires processing.

In some cases ViComp outperformed ViCompCD and vice versa (e.g. Event1 and Event2 in Fig. 5.8(a)) because of the CD condition. Since dissimilar and similar clusters were formed in the CD condition, visual segments which received lower total rank (based on quality) got selected if they belonged to the dissimilar cluster. Without the CD condition, visual segments with better quality were selected while considering the view diversity. 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, it was observed that CD and SSIM-based diversity (MoViMash [122]) lowered the overall quality of the generated videos.

5.7 Summary

In this chapter, we proposed ViComp, a framework for the 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. Our method is similar to Firstfit [133] and MoViMash [122] as we also perform visual quality and camera motion analysis for video composition; but differ significantly as we further proposed an audio stitching, automatic cut-point detection and rank-based camera selection methods. Audio stitching is used to avoid audio variation that occurs when switching camera views. As opposed to manual cut-point selection [133, 122], we proposed an automatic cut-point selection method for UGV segmentation. We used a single holistic spatial quality measure (BRISQUE [108]) instead of multiplication-based combination of individual quality measures [133, 122]. Multiplication-based combination might suppress the effect of one individual score over the other. We designed a rank-based camera selection strategy to combine the effect of the spatial and spatio-temporal quality scores along with the view diversity condition. Furthermore, we contributed ViCompG, a variant of ViComp, that solely considered gyro-based camera motion assessment for ranking low quality visual segments. Our frameworks were tested on a dataset of 16 events (105 UGVs). In order to analyse the user satisfaction, we designed a subjective test by considering the ITU-R recommendations. The subjective evaluation showed better or comparable results of ViComp with ViCompCD, and ViComp outperformed ViRand, FirstFit [133] and MoViMash [122]. ViCompG was also found to be comparable with ViComp.

We also presented the results for camera motion detection using gyroscope. The method aligned the multi-modal data and used the tri-axial gyroscope data captured simultaneously with the video to detect pan, tilt and shake motions. Our proposed method outperformed existing inertial sensor-based and visual methods by giving the collective F_1-score of 89% for pan, tilt and shake detection. The method showed potential towards designing real-time applications for camera motion analysis and video composition.

Chapter 6

Conclusions

6.1 Summary of achievements

This thesis focused on designing an end-to-end framework for the automatic identification of multi-camera UGVs of the same event from a database, synchronisation of the identified UGVs, camera motion detection and composition of a continuous video. We exploited multi-modal (audio, visual and gyroscope) data for the development of the proposed framework. Detailed below are the specific achievements of this thesis.

Existing audio-based event clustering methods only organised UGVs recorded at the same concert or public address [32, 76], and used a fixed classification threshold to identify matched recording pairs [20]. The performance of the existing audio-based synchronisation methods decreases in the presence of audio degradations (reverberations, ambient noise) [132, 76, 131]. We proposed an automatic identification and synchronisation framework for unedited multi-camera UGVs that considered query-by-example video event search [J1]. We contributed a novel descriptor derived from the pairwise matching of audio features of UGVs. The designed descriptor gave a discriminant representation that facilitated the definition of a classification threshold for automatic query-by-example event identification. Audio chroma feature was used to cluster UGVs of the same event and to estimate their relative time-shifts. Coarser frame size for audio feature extraction facilitated the efficient query-by-example video event identification while refining it for the identified videos gave precise time-shift estimation. We contributed a database of 263 multi-camera UGVs of 48 real-world events and used it for the evaluation of the proposed framework. The classification threshold was trained using a relatively smaller dataset (7 events, 42 UGVs) that contained amplified and non-amplified sound sources, and audio degradations. Testing for the event identification was performed on an unseen event database (41 events, 221 UGVs) and additional 60 UGVs used as a query. The high value of the area under the precision-recall curve (0.97 and 0.96) for both test cases suggested the effectiveness of our framework. The proposed synchronisation method outperformed the existing methods [132, 132, 23] giving an overall accuracy of 99.62%. While designing the framework, we assumed that the time difference of arrival of a sound to the recording device is negligible. When matching recording pairs, we computed the minimum across each row of the distance matrix for the estimation of time-shifts. This resulted in dominant outliers when one recording is shorter than the other.

The synchronised UGVs are useful for developing event understanding and video composition applications [122, 133, J2, 35]. Camera motion analysis is an important component of video composition applications as unintentional motion like fast pan, fast tilt and shake influence the perceived visual quality. Generally, visual content-based methods [6, 7, 99, 141] for camera motion detection are computationally expensive and get influenced by moving objects and brightness changes. The performance of the existing inertial sensor-based method [35] is limited due to noisy compass and accelerometer data. A gyroscope is more accurate than compass and accelerometer for the rotation estimation. Therefore, we developed a gyro-based camera motion detection method for UGVs captured from smartphones [C1]. Pan and tilt were detected by extracting the dominant motions from the gyroscope, whereas shake was detected by analysing high frequencies in the gyroscope data. For the experimental evaluation, we collected multi-modal data (24 single camera UGVs, 70 mins duration) at several real-world scenarios that contained varying brightness (day and night-time recordings). The proposed method outperformed the existing visual [141, 21] and inertial sensor-based [35] methods giving the accuracy of 0:94; 0:84 and 0:85 for pan, tilt and shake detection, respectively.

Audio quality influences the perceived quality of the composed video [15]. The existing methods for video composition [133, 122] from multi-camera UGVs did not consider audio content analysis, and performed manual video segmentation [133] and manual classification of camera views [122]. We proposed an automatic audio-visual camera selection framework for composing a continuous multi-view video from multiple UGVs of the same event [J2]. We developed a stitching method to solve the audio variation issue, which occurs when switching between camera views, followed by an automatic audio-based cut-point selection method to segment the videos. The proposed framework combined time continuous video segments from multiple UGVs using a rank-based camera selection strategy by considering audio-visual quality and view diversity. We also designed a gyro-based assessment score for ranking the quality of video segments and used this score to contribute a gyro-based video composition framework. In order to analyse the user satisfaction, we designed a subjective test by considering the ITU-R recommendations [70]. We evaluated the proposed frameworks through subjective tests on a dataset of 16 real-world events (105 UGVs) and compared them with state-of-the-art methods [133, 122]. The proposed frameworks performed better than the existing methods [133, 122] due to the designed suitable cut-point selection, specific visual quality assessments and rank-based camera selection methods. The proposed gyro-based video composition framework could be preferred when the gyroscope data is available as this significantly reduced the computational complexity and simplified the problem. The proposed frameworks do not consider event understanding and ROI localisation, which may further improve the perceived quality of the composed videos.

6.2 Future work

Below are discussed the future directions of this thesis work:

1. The proposed identification and synchronisation framework [J1] automatically clustered the database UGVs into events provided at least two recordings existed for an event. The proposed framework generated fixed number of event clusters that cannot be updated later on. Considering the increasing availability of UGVs, online update and dynamic growth of the database would be an important aspect to analyse. Therefore, future work could focus on generating a new cluster for a query video for which a matching UGV does not exist in the database.

2. In the proposed identification and synchronisation framework, we computed the minimum across each row of the distance matrix for time-shift estimation. This resulted in dominant outliers when the duration of one recording was short (less than 30s with 10% overlap) than the other. Therefore, the future work could involve decomposing the audio signals into blocks and performing block-wise matching of the recording pairs to suppress the effect of outliers and to obtain the time-shift.

3. We utilised empirically selected thresholds for pan, tilt and shake classification in the pro-

posed camera motion detection method [C1]. Future work could focus on designing a probabilistic model for learning the classification boundaries.

4. We substituted the visual data with the gyroscope data in the proposed camera motion detection method [C1] and achieved better performance with reduced computational complexity. Since, the use of gyroscope data significantly reduces the amount of data to be processed, future work could involve the designing of a real-time application for camera motion analysis on smart devices.

5. The proposed audio and visual-based framework for video composition (ViComp) [J2] made use of global visual feature analysis. Generally, the UGVs contain visual degradations that limit their analysis using local features. Provided the visual data is recorded from high-resolution multiple devices and in a sufficiently textured environment with high brightness, future work could focus on 3D reconstruction of the scene [9]. This could provide semantic details of the scenario for an in-depth scene analysis.

Appendix A

Collection of user-generated video datasets

Multi-camera UGVs dataset of multiple events is required for the validation of our proposed frameworks. Therefore, we collected time overlapping UGVs of multiple events from YouTube [4] and by ourselves. This dataset is used for the validation of our proposed identification, synchronisation and video composition frameworks (Sec. 4.6 and Sec. 5.6). A multi-modal (audio, video and inertial sensors) dataset of different events, captured from single or multiple cameras, is also collected by ourselves for the validation of our proposed gyro-based methods (Sec. 5.5 and Sec. 5.6). Mentioned below are the details of the collected datasets.

A.1 Dataset for identification and synchronisation

We collected 263 multi-camera UGVs of 43 different concert events and 5 different self-captured events (Tab. A.2). The concert recordings are collected from YouTube [4], while the other events are captured by ourselves. In total we collected multi-camera UGVs for 48 events, with a total duration of 1200 minutes (mins). The concerts include 20 events from a Nickelback concert, 10 events from an Evansence concert, 9 events from Alice Cooper concert, an event from Madonna, Coldplay and Bruce Springsteen concerts, and an event from Les Miserables musical performance (detailed in Tab. A.1). The ROI in the concert events was mainly the singer and the musicians on the stage. Sometimes the users performed pan and tilt motion to capture the cheering audience, off-stage performer or sky. Some users seemed to perform activities like dancing while recording that introduced motion blur and degraded the visual quality. Moreover, recording duration, lighting conditions, fields of view and distance from the ROI varied from one recording

Table A.1: Description of collected datasets' event. Key: k: number of events

Event title	k	Location	Date	Collection source
Nickelback concert	20	O2 Arena, London	01/10/2012	Youtube
Evansence concert	10	Wembley Arena, London	09/11/2012	Youtube
Alice Cooper concert	9	Wembley Arena, London	28/10/2012	Youtube
Madonna concert	1	MDNA tour, Abu Dhabi	03/06/2012	Youtube
Coldplay concert	1	Emirates Stadium, London	01/06/2012	Youtube
Les Miserables musical	1	O2 arena, London	03/10/2012	Youtube
Bruce Springsteen concert	1	Wrecking ball tour, Barcelona	17/05/2012	Youtube
Change of guard	1	Buckingham Palace, London	01/07/2012	self
Olympic torch relay	1	Sheffield	06/07/2012	self
Olympic torch relay	1	Mile end, London	23/07/2012	self
Xmas dinner	1	London	06/12/2011	self
NYE fireworks	1	Embankment, London	31/12/2012	self+Youtube

to another. These scenarios contained amplified sound source, but the recorded audio signals were degraded due to channel noise, background music, reverberations and crowd cheering.

The self-captured events that we recorded ourselves include the Changing of the Guard, the Olympic torch relay, the New Year fireworks and a dinner (detailed in Tab. A.1). These events introduced additional challenges for audio synchronisation as they contained considerable ambient noise, moving cameras widely separated from each other and moving audio sources with non-amplified sound. The ROI in Changing of the Guard and Olympic torch relay was moving, and the cameras were well separated apart. The field of view of some cameras were not overlapping with the other. The scenario contained high local ambient noise due to crowded environment The audio and visual quality varied significantly because to the recording device specifications and varying distance from the ROI. The audio signal for one of the recording in Olympic torch event was not captured properly due to malfunctioning of the device. The dinner event also contained high ambient noise. The NYE fireworks contained low illumination recordings, ambient noise and varying distance from the sound source and non-overlapping fields of view. Table A.2 summarises the main characteristics of our datasets along with their challenges. Key frames for each collected UGV are displayed in Sec. A.4 for the visualisation of their visual quality, fields of view and distance from the ROI.

Table A.2: Summary of the main characteristics of the dataset along with its challenges. (Key: N: number of UGVs; s^v: video frame rate; s^a: audio sampling rate; -: indicates that only some of the UGVs contain that property: MC: Moving cameras; VD: varying distance; CN: channel noise; AN: ambient noise; NAS: non-amplified sound recordings).

No.	Event title	General characteristics				Challenges				
		N	s^v (fps)	s^a (KHz)	Duration (min:s)	MC	VD	CN	AN	NAS
1	Nickelback_Event1	7	16□ 30	44:1	4:01 - 5:20	x	x	-	-	
2	Nickelback_Event2	9	16□ 30	44:1	4:00 - 4:42	x	x	-	-	
3	Nickelback_Event3	6	24□ 30	44:1	0:18 - 4:29	x	x	-	-	
4	Nickelback_Event4	7	16□ 30	44:1	2:26 - 4:47	x	x	-	-	
5	Nickelback_Event5	5	25□ 30	44:1	3:20 - 4:56	x	x	-	-	
6	Nickelback_Event6	4	25□ 30	44:1	3:43 - 4:16	x	x	-	-	
7	Nickelback_Event7	6	17□ 30	44:1	2:01 - 5:25	x	x	-	-	
8	Nickelback_Event8	5	24□ 30	44:1	1:39 - 4:06	x	x	-	-	
9	Nickelback_Event9	4	24□ 30	44:1	2:59 - 8:16	x	x	-	-	
10	Nickelback_Event10	4	24□ 25	44:1	3:37 - 5:22	x	x	-	-	
11	Nickelback_Event11	3	25□ 30	44:1	1:41 - 3:35	x	x	-	-	
12	Nickelback_Event12	3	24□ 25	44:1	2:51 - 4:42	x	x	-	-	
13	Nickelback_Event13	3	25	44:1	3:35 - 4:16	x	x	-	-	
14	Nickelback_Event14	3	25□ 30	44:1	3:29 - 4:45	x	x	-	-	
15	Nickelback_Event15	3	25□ 30	44:1	4:12 - 4:42	x	x	-	-	
16	Nickelback_Event16	3	25□ 30	44:1	2:58 - 3:55	x	x	-	-	
17	Nickelback_Event17	3	30	44:1	3:23 - 3:52	x	x	-	-	
18	Nickelback_Event18	2	24□ 30	44:1	3:09 - 8:46	x	x	-	-	
19	Nickelback_Event19	2	25	44:1	3:48 - 4:18	x	x	-	-	
20	Nickelback_Event20	2	25□ 30	44:1	4:22 - 5:04	x	x	-	-	
21	Evanescence_Event1	16	25□ 30	44:1	0:45 - 5:56	x	x	-	-	
22	Evanescence_Event2	7	25□ 30	44:1	0:59 - 3:57	x	x	-	-	
23	Evanescence_Event3	10	25□ 30	44:1	2:00 - 4:47	x	x	-	-	
24	Evanescence_Event4	9	24□ 30	44:1	0:20 - 4:03	x	x	-	-	
25	Evanescence_Event5	6	25□ 30	44:1	2:57 - 4:08	x	x	-	-	
26	Evanescence_Event6	9	30	44:1	0:55 - 4:54	x	x	-	-	
27	Evanescence_Event7	8	24□ 30	44:1	2:02 - 4:04	x	x	-	-	
28	Evanescence_Event8	9	24□ 30	44:1	1:08 - 5:08	x	x	-	-	

Continued on next page

Table A.2 – continued from previous page

No.	Event title	General characteristics				Challenges				
		N	s^v (fps)	s^a (KHz)	Duration (min:s)	MC	VD	CN	AN	NAS
29	Evanescence_Event9	4	24□ 30	44:1	2:27 - 4:21	x	x	-	-	
30	Evanescence_Event10	6	24□ 25	44:1	1:37 - 3:32	x	x	-	-	
31	AliceCooper_Event1	8	30	44:1	3:12 - 5:00	x	x	-	-	
32	AliceCooper_Event2	11	24□ 30	44:1	3:07 - 6:03	x	x	-	-	
33	AliceCooper_Event3	2	29□ 30	44:1	2:38 - 2:57	x	x	-	-	
34	AliceCooper_Event4	3	30	44:1	3:56 - 4:10	x	x	-	-	
35	AliceCooper_Event5	3	25	44:1	3:36 - 4:28	x	x	-	-	
36	AliceCooper_Event6	3	25□ 30	44:1	3:36 - 6:41	x	x	-	-	
37	AliceCooper_Event7	3	17□ 30	44:1	3:15 - 4:0	x	x	-	-	
38	AliceCooper_Event8	4	24□ 30	44:1	1:24 - 3:04	x	x	-	-	
39	AliceCooper_Event9	2	30	44:1	3:26 - 3:27	x	x	-	-	
40	Madonna_Event	11	24□ 30	44:1	0:28 - 5:37	x	x	-	-	
41	Coldplay_Event	7	24□ 30	44:1	4:16 - 7:50	x	x	-	-	
42	LesMesirable_Event	7	24□ 30	44:1	2:33 - 6:44	x	x	-	-	
43	Springsteen_Event	6	24□ 30	44:1	3:24 - 6:35	x	x	-	-	
44	ChangeofGuard	2	25□ 30	32□ 44:1	0:34-2:02	x	x		x	x
45	OlympicTorchSheffield	2	30	44:1	0:39-1:28	x	x		x	x
46	OlympicTorchMileEnd	7	16□ 30	16□ 48	5:54-7:01	x	x	x	x	x
47	XmasDinner	3	30	16	2:35-3:19	x	x		x	x
48	FireworksLondon	11	25□ 30	16□ 44:1	0:29-14:16	x	x		x	x

We also collected 60 additional UGVs to be used as the query C_q, which are not overlapping with any of the 48 events but belonged to similar events such as the same concert of Nickelback, Evanescence, and Alice Cooper, the Changing of the Guard in different parts of the world and the Olympic torch relay in different places in the UK.

The ground-truth for video identification and synchronisation was generated for all the UGVs by manually observing one or more audio, visual or audio-visual instances in them for each event. Two observers logged the local time of the instance that appeared in some or all UGVs of an event. This information was then used for aligning the UGVs within each event on a common

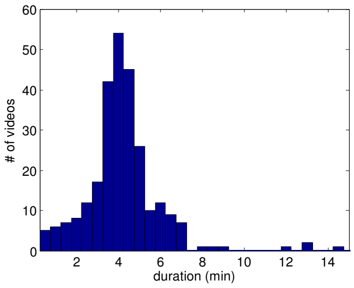


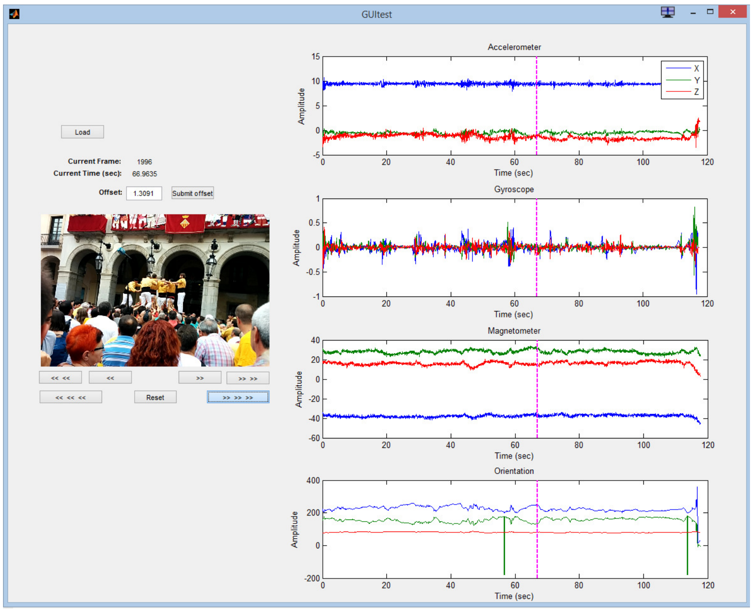
Figure A.1: Distribution of UGVs duration (mins) in the collected dataset.

timeline. When observing and matching two UGVs, an systematic error of □1 video frame (□0:04s) because of annotation can occur. This error can increase if careful observations are not taken.

The collected UGVs are of varying duration, the distribution of which is shown in Fig. A.1. The video lengths are mainly clustered between 3-6 mins primarily because most of the video recordings are from concerts, and the usual length of a song played is around 4 mins, which is the event of interest for most of the audience. There are also some videos of short length (below 2 mins), which reflects the fact that user interest varies from person to person and one might just want to record a particular instance within an event. There are few videos of length greater than 8 mins mainly because it might be tiring for the user to hold a hand-held camera for a longer duration and usually an event of interest is of a short duration.

A.2 Multi-modal dataset for camera motion detection

For analysing camera motion using gyroscope data, we captured multi-modal data (audio, video, inertial sensors) at several real-world scenarios using Sensor Data Logger App [1]. Different smartphones (Samsung Galaxy SII and SIII, LG Nexus 5) with embedded inertial sensors were used for capturing the data. 24 multi-modal UGVs were captured at events such as concerts, parade, festivals and fireworks that took place in Vilanova, Spain in 2014 (listed in Table A.3). The data was captured in different High Brightness (HB) and Low Brightness (LB) scenarios (e.g. day and night-time) for a total duration of 70 mins. The video frame rate and frame size varied depending on the brightness of the recorded scene due to the programmed settings of the



Current Frame: 1996
Current Time (sec): 66.9635
Offset: 1.3091

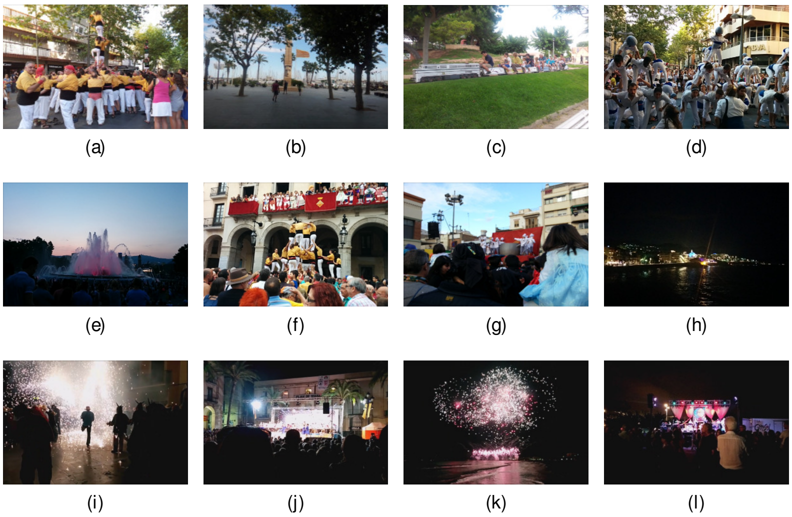
Figure A.2: Multi-modal data visualizer. The inertial sensor and video data are loaded in the visualiser. An offset is entered to add delay in the inertial sensor data. The video frames are incremented or decremented and accordingly the pointer in the inertial sensor data is displaced to correspond to the same time as that of the current video frame.

App. The collected dataset contains single camera recordings at distinct timings and locations, changing lights and varying camera motions. The representative frames for each event are shown in Fig. A.3 from which the variation in brightness can be observed.

In order to visualise the captured multi-modal data, we developed a Graphical User Interface (GUI) using matlab. The GUI was designed to show the captured video and inertial sensor data coherently (see Fig. A.2). An offset variable was introduced which shifted the inertial sensor data that facilitated in the annotation of the ground-truth delay for synchronisation. This delay was obtained by observing a pan/tilt/shake motion both in visual and gyroscope data. Each captured video was manually annotated to obtain labels for pan, tilt and shake at every second.

Table A.3: Multi-modal data collected at f_r = 50 Hz. Key: N_o: number of non-overlapping multi-modal recordings; s^v: video frame rate; VFS: video frame size; TD: total duration; HB: high brightness; LB: low brightness.

Event title	N_o	s^v	VFS	TD(min:s)	Time	Device
HumanTower1	3	30	(720,480)	06:45	Day (HB)	Samsung SII
VilanovaRambla	1	30	(720,480)	01:24	Day (HB)	Samsung SII
MiniTrain	3	30	(720,480)	02:25	Day (HB)	Samsung SII
Falcons	1	30	(720,480)	01:45	Day (HB)	Nexus 5
MagicFountain	1	30	(720,480)	00:35	Day (HB)	Nexus 5
HumanTower2	3	30	(720,480)	08:33	Day (HB)	Nexus 5
Caramel	1	30	(720,480)	08:03	Day (HB)	Samsung SIII
SantJordi	1	12	(854,480)	01:17	Night (LB)	Nexus 5
Correfoc	3	27	(854,480)	06:27	Night (LB)	Nexus 5
Orchestra	3	23	(854,480)	11:17	Night (LB)	Nexus 5
Fireworks	1	12	(854,480)	13:43	Night (LB)	Nexus 5
Concert	2	13	(854,480)	06:44	Night (LB)	Nexus 5



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure A.3: Respresentative frames for High Brightness (HB) and Low Brightness (LB) multi-modal recordings. HB recordings include (a) HumanTower1, (b) VilanovaRambla, (c) MiniTrain, (d) Falcons, (e) MagicFountain, (f) HumanTower2, (g) Caramel. LB recordings include (h) SantJordi, (i) Correfoc, (j) Orchestra, (k) Fireworks, (l) Concert.

Table A.4: Details of the dataset used for testing. All recordings have audio sampled at 44.1 kHz. Key: k: event number; N: number of UGVs; s^v: video frame rate; D_C: coverage duration; D_O: overlap duration; VFS: video frame size; ISD: inertial sensor data.

k	Event title	N	s^v	Duration	D_c	D_o	VFS	ISD
			min-max (fps)	min-max (min:s)	(min:s)	(min:s)	(pixels)	
1	Nickelback_Event1	7	16-30	04:01 - 05:20	05:23	04:05	(640, 360)	
2	Nickelback_Event2	9	16-30	04:00 - 04:42	04:44	03:56	(480, 360), (640, 360)	
3	Nickelback_Event3	7	16-30	02:26 - 04:46	04:46	03:14	(640, 360)	
4	Nickelback_Event4	5	24-30	03:20 - 04:56	04:56	03:20	(640, 360)	
5	Evanescence_Event1	6	25-30	03:17 - 03:57	03:57	03:09	(640, 360), (568, 360)	
6	Evanescence_Event2	6	29-30	03:02 - 04:03	04:05	02:42	(640, 360), (480, 360)	
7	Evanescence_Event3	6	25-30	02:57 - 04:08	04:08	02:57	(480, 360), (640, 360)	
8	Evanescence_Event4	7	24-30	03:35 - 04:04	04:02	03:58	(640, 360), (480, 360)	
9	Concert_Event1 [133]	5	25	04:24 - 04:45	04:44	04:17	(320, 240)	
10	Concert_Event2 [133]	5	25-30	05:01 - 06:58	07:01	04:32	(320, 240)	
11	Concert_Event3 [133]	4	15-30	02:24 - 05:17	05:15	02:47	(320, 240)	
12	Dance_Event1 [122]	12	30	04:01 - 04:57	05:00	04:05	(720, 480)	
13	Dance_Event2 [122]	12	30	03:45 - 04:13	04:13	03:49	(720, 480)	
14	Caramel_Event1	4	30	06:49 - 09:35	11:44	04:53	(720, 480)	x
15	Caramel_Event2	4	30	08:01 - 10:05	11:31	06:43	(720, 480)	x
16	Caramel_Event3	4	30	03:08 - 10:00	14:31	07:43	(720, 480)	x

A.3 Dataset for video composition

For the subjective evaluation of the proposed video composition frameworks, we used the dataset detailed in Table A.4. We used 8 concert events from our previously collected dataset, 3 concert events from [133], 2 dance events from [122] and 3 carnival events that we collected ourselves.

Each event was captured by 4 to 12 hand-held cameras which were overlapping in time. Event1-4 comprise multiple recordings of four different songs from a Nickelback concert, and Event5-8 comprise the multiple recordings of four different songs from an Evanescence concert, that we collected from YouTube (details of Event1-8 are presented in Sec. A.1). Event9-11 are the same recordings as used by the FirstFit [133] that are pop and rock concerts, and Event12-13 are the same recordings as used in MoViMash [122] that are dance sequences at a local show. The recordings were captured in dynamic environments, and contained varying field of views, changing lights and moving cameras, that directly influenced the visual quality of each recording.

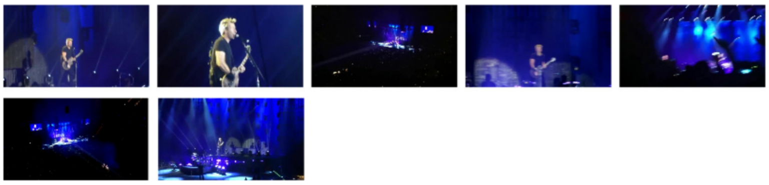
We recorded Event 14-16 by ourselves, such that inertial sensor data was also captured. Event 14-16 are three different carnival performances that took place during the 2015 caramel festival in Vilanova, Spain. Four volunteers who helped in capturing the events were instructed to record from any location of their interest in the vicinity of the performances. All recordings contained high brightness as these events took place on a clear day in an outdoor square. Four android devices (Samsung Galaxy SII, SIII and S5 mini and LG Nexus 5) were used to capture these events. The scenario comprises two stages, one with a dance team, other with the singing band. The object of interest varied depending on the field of view of each volunteer.

A.4 Key-frames of multi-camera user-generated videos

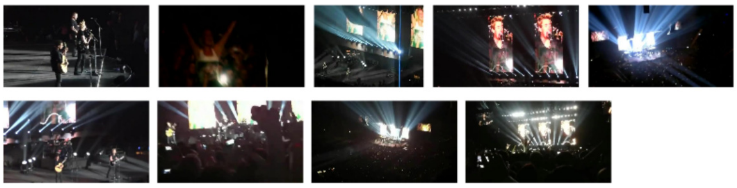
We collected multi-camera UGVs of 48 events as detailed in Sec. A.1. The representative frames for all UGVs listed in Table A.2 are shown below for visualisation. For each event, the synchronised frames are shown from which the variation in visual quality, frame size, field of view and distance from the ROI can be observed.

Table A.5: Key-frames of multi-camera UGVs as listed in Table A.2.

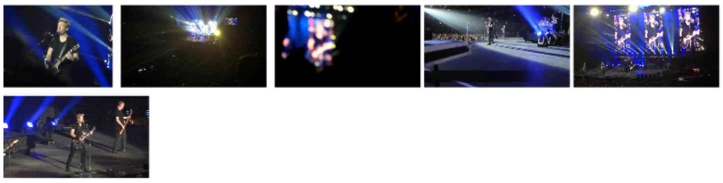
1. Nickelback_Event1



2. Nickelback_Event2



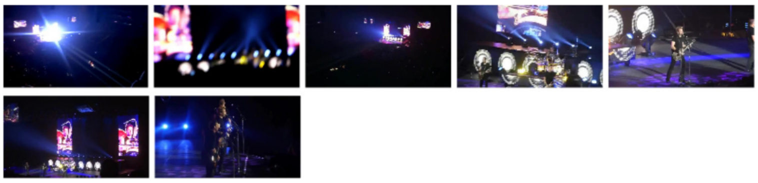
3. Nickelback_Event3



Continued on next page

Table A.5 – continued from previous page

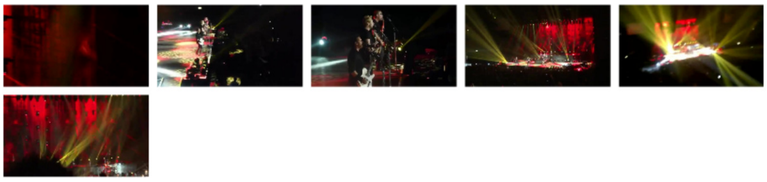
4. Nickelback_Event4



5. Nickelback_Event5

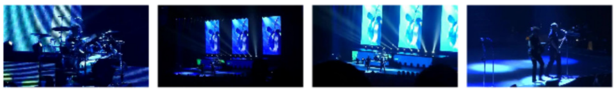
6. Nickelback_Event6

7. Nickelback_Event7



8. Nickelback_Event8

9. Nickelback_Event9



10. Nickelback_Event10

Continued on next page

Table A.5 – continued from previous page

11. Nickelback_Event11

12. Nickelback_Event12

13. Nickelback_Event13

14. Nickelback_Event14

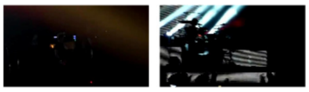
15. Nickelback_Event15

16. Nickelback_Event16

17. Nickelback_Event17

18. Nickelback_Event18

19. Nickelback_Event19

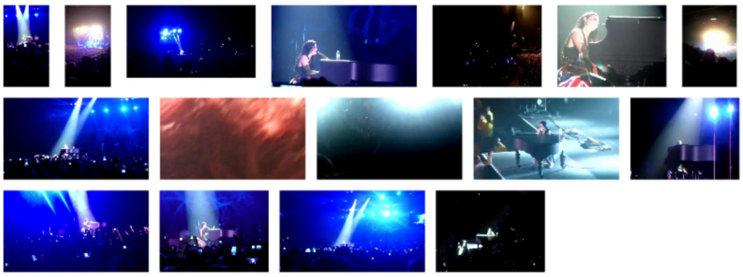


Continued on next page

Table A.5 – continued from previous page

20. Nickelback_Event20

21. Evanescence_Event1



22. Evanescence_Event2

23. Evanescence_Event3

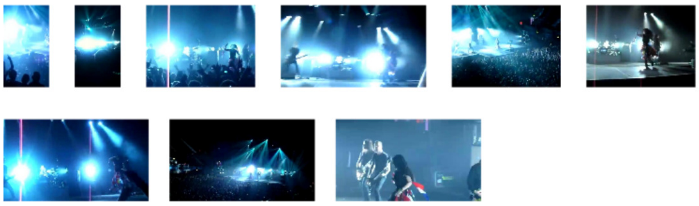
24. Evanescence_Event4

Continued on next page

Table A.5 – continued from previous page

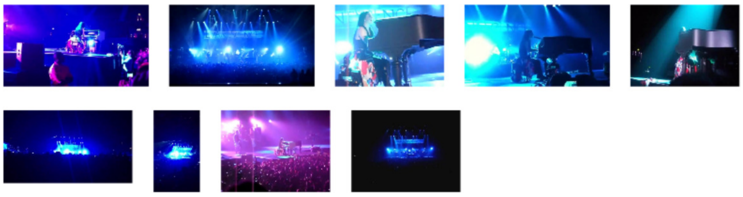
25. Evanescence_Event5

26. Evanescence_Event6



27. Evanescence_Event7

28. Evanescence_Event8



29. Evanescence_Event9

30. Evanescence_Event10

Continued on next page

Table A.5 – continued from previous page

31. AliceCooper_Event1

32. AliceCooper_Event2

33. AliceCooper_Event3

34. AliceCooper_Event4

35. AliceCooper_Event5

36. AliceCooper_Event6

37. AliceCooper_Event7

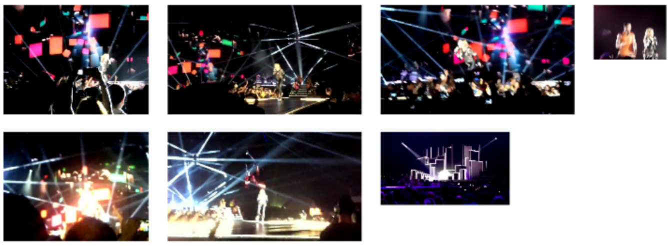
Continued on next page

Table A.5 – continued from previous page

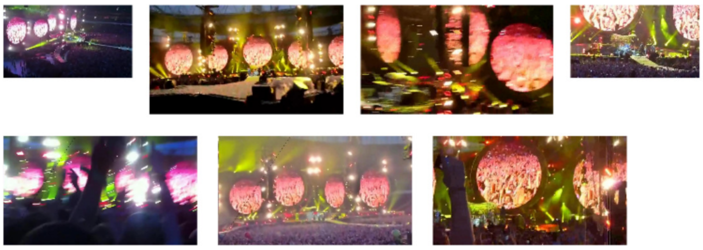
38. AliceCooper_Event8

39. AliceCooper_Event9

40. Madonna_Event



41. Coldplay_Event



42. LesMesirable_Event

Continued on next page

Table A.5 – continued from previous page

43. Springsteen_Event

44. Change of Guard

45. Olympic torch Sheffield

46. Olympic torch Mile End

47. Xmas Dinner

Continued on next page

Table A.5 – continued from previous page

48. Fireworks London

A.5 Summary

This appendix presented the details of the datasets that we collected and used for the validation of our proposed frameworks. We collected a dataset of 263 multi-camera UGVs of 48 real-world events that we used for analysing our proposed identification and synchronisation framework. The proposed ViComp framework was validated on 8 events that were selected from the above mentioned dataset and 5 events that were the same as used by [122, 133]. Moreover, we collected a dataset of 24 multi-modal UGVs and 3 events (12 multi-modal UGVs) that we used for validating the proposed CMDG method and ViCompG framework, respectively.

Bibliography

[1] Data logger. https://code.google.com/p/cellbots/downloads/detail?name=CellbotsDataLogger_v1.1.0_full.apk. Accessed: 2015-06-25.

[2] Trackid. https://play.google.com/store/apps/details?id=com.sonyericsson.trackid&hl=en. Accessed: 2015-06-25.

[3] Videolan. http://www.videolan.org/vlc/index.html. Accessed: 2015-06-25.

[4] Youtube. https://www.youtube.com/. Accessed: 2015-06-25.

[5] Youtube statistics. https://www.youtube.com/yt/press/statistics.html. Accessed: 2015-06-25.

[6] G. Abdollahian, C.M. Taskiran, Z. Pizlo, and E.J. Delp. Camera motion-based analysis of user generated video. IEEE Transactions on Multimedia, 12:28–41, 2010.

[7] J. Almeida, R. Minetto, T.A. Almeida, R.S. Torres, and N.J. Leite. Robust estimation of camera motion using optical flow models. In Advances in Visual Computing, pages 435–446. Springer, 2009.

[8] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J.L. Reyes-Ortiz. Energy efficient smartphone-based activity recognition using fixed-point arithmetic. Journal of Universal Computer Science, 19:1295–1314, 2013.

[9] I. Arev, H.S. Park, Y. Sheikh, J. Hodgins, and A. Shamir. Automatic editing of footage from multiple social cameras. ACM Transactions on Graphics, 33(4):81, 2014.

[10] M. N. Armenise, C. Ciminelli, F. Dell'Olio, and V. M. N. Passaro. Advances in gyroscope technologies. Springer Science & Business Media, 2010.

[11] S. Ayub, A. Bahraminisaab, and B. Honary. A sensor fusion method for smart phone orientation estimation. In Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool, UK, 2012.

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

[13] S.S. Beauchemin and J.L. Barron. The computation of optical flow. ACM Computing Surveys (CSUR), 27(3):433–466, 1995.

[14] H. Becker, M. Naaman, and L. Gravano. Event identification in social media. In Proc. of the ACM SIGMOD Workshop on the Web and Databases, Rhode Island, USA, 2009.

[15] J.G. Beerends and F.E. De Caluwe. The influence of video quality on perceived audio quality and vice versa. Journal of the Audio Engineering Society, 47(5):355–362, 1999.

[16] J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M.B. Sandler. A tutorial on onset detection in music signals. IEEE Transactions on Speech and Audio Processing, 13(5):1035–1047, 2005.

[17] D.M. Boore. Analog-to-digital conversion as a source of drifts in displacements derived from digital recordings of ground acceleration. Bulletin of the Seismological Society of America, 93(5):2017–2024, 2003.

[18] M.-L. Bourguet and J. Wang. A robust audio feature extraction algorithm for music identification. In Proc. of the 129th Audio Engineering Society Convention, San Francisco, CA, pages 8180 – 8189, 11 2010.

[19] C.J. Bowen and R. Thompson. Grammar of the Edit. CRC Press, 2013.

[20] N.J. Bryan, P. Smaragdis, and G.J. Mysore. Clustering and synchronizing multi-camera video via landmark cross-correlation. In Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, pages 2389–2392, 2012.

[21] M. Campanella, H. Weda, and M. Barbieri. Edit while watching: home video editing made easy. In Electronic Imaging 2007, pages 65060L–65060L. International Society for Optics and Photonics, 2007.

[22] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. The Journal of VLSI Signal Processing, 41(3):271–284, 2005.

[23] A.L. Casanovas and A. Cavallaro. Audio-visual events for multi-camera synchronization. Multimedia Tools and Applications, 74(4):1317–1340, 2014.

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

[25] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(11):1409–1424, 2002.

[26] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. International Journal of Computer Vision, 68(1):53–64, 2006.

[27] C.W. Chen, R. Cook, M. Cremer, and P. DiMaria. Content identification in consumer applications. In Proc. of the IEEE International Conference on Multimedia and Expo (ICME), New York, USA, pages 1536–1539, 2009.

[28] C.-L. Chou, H.-T. Chen, C.-C. Hsu, C.-P. Ho, and S.-Y. Lee. Near-duplicate video retrieval by using pattern-based prefix tree and temporal relation forest. In Proc. of the IEEE International Conference on Multimedia and Expo (ICME), Chengdu, China, pages 1–6, 2014.

[29] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), South Carolina, US, volume 2, pages 142–149, 2000.

[30] I. Constandache, R.R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In Proc. of the IEEE Conference on Computer Communications (INFOCOM), San Diego, CA, pages 1–9, 2010.

[31] Perry R. Cook. Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics, Ch: 13. The MIT Press, 2001.

[32] C.V. Cotton and D.P.W. Ellis. Audio fingerprinting to identify multiple videos of an event. In Proc. of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Texas, USA, pages 2386–2389, 2010.

[33] E. Coviello, L. Barrington, A.B. Chan, and G.R.G. Lanckriet. Automatic music tagging with time series models. In Proc. of the 11th International Society for Music Information Retrieval (ISMIR) Conference, Utrecht, Netherlands, pages 81–86, 2010.

[34] M. Cremer, B. Froba, O. Hellmuth, J. Herre, and E. Allamanche. Audioid: Towards content-based identification of audio material. In Audio Engineering Society Convention 110. Audio Engineering Society, 2001.

[35] F. Cricri, K. Dabov, I. Curcio, S. Mate, and M. Gabbouj. Multimodal extraction of events and of information about the recording activity in user generated videos. Multimedia Tools and Applications, 70:119–158, 2012.

[36] F. Cricri, K. Dabov, M.J. Roininen, S. Mate, I.D.D. Curcio, and M. Gabbouj. Multimodal semantics extraction from user-generated videos. Advances in Multimedia, 2012:1–17, 2012.

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

[38] F. Daniyal, M. Taj, and A. Cavallaro. Content and task-based view selection from multiple video streams. Multimedia Tools and Applications, 46:235–258, 2010.

[39] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al. The youtube video recommendation system. In Proc. of the ACM conference on Recommender systems, Barcelona, Spain, pages 293–296, 2010.

[40] P.E. Dickson, W.R. Adrion, A.R. Hanson, and D.T. Arbour. First experiences with a classroom recording system. In Proc. of the ACM SIGCSE Conference on Innovation and Technology in Computer Science Education, Paris, France, volume 41, pages 298–302, 2009.

[41] Edward Dmytryk. On Film Editing. Focal Press, 1984.

[42] T. D'Orazio and M. Leo. A review of vision-based systems for soccer video analysis. Pattern recognition, 43(8):2911–2926, 2010.

[43] M. Douze, H. Jégou, C. Schmid, and P. Pérez. Compact video description for copy detection with precise temporal alignment. In Proc. of the European Conference on Computer Vision (ECCV), Crete, Greece, pages 522–535, 2010.

[44] N.Q. Duong and F. Thudor. Movie synchronization by audio landmark matching. In Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),Vancouver, Canada, pages 3632–3636. IEEE, 2013.

[45] A. Elhayek, C. Stoll, K.I. Kim, H.-P. Seidel, and C. Theobalt. Feature-based multi-video synchronization with subframe accuracy. Springer, 2012.

[46] D. Ellis. Robust landmark-based audio fingerprinting. web resource (http://labrosa.ee.columbia.edu/matlab/fingerprint/), 2009.

[47] D.P.W. Ellis, C.V. Cotton, T. Friedland, and K. Esterson. Methods, systems, and media for mobile audio event recognition, September 21 2012. US Patent App. 13/624,532.

[48] D.P.W. Ellis and G.E. Poliner. Identifying 'cover songs' with chroma features and dynamic

programming beat tracking. In Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hawaii, USA, volume 4, pages 1429 –1432, 2007.

[49] S. Ewert, M. Muller, and P. Grosche. High resolution audio synchronization using chroma onset features. In Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Taipei, Taiwan, 2009, pages 1869–1872, 2009.

[50] R. Ferzli and L. J. Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb). IEEE Transactions on Image Processing, 18(4):717–728, 2009.

[51] D. Fortun, P. Bouthemy, and C. Kervrann. Optical flow modeling and computation: a survey. Computer Vision and Image Understanding, 134:1–21, 2015.

[52] D. Fragoulis, G. Rousopoulos, T. Panagopoulos, C. Alexiou, and C. Papaodysseus. On the automated recognition of seriously distorted musical recordings. IEEE Transactions on Signal Processing, 49(4):898–908, 2001.

[53] T. Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In Proc. of the International Computer Music Conference (ICMC), Beijing, China, pages 464–467, 1999.

[54] T. Giannakopoulos. 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, 2009.

[55] R.C. Gonzalez and R.E. Woods. Digital Image Processing, Chap. 9. Pearson Education, 2008.

[56] E.G. Gutiérrez. Tonal description of music audio signals. PhD thesis, Universitat Pompeu Fabra, 2006.

[57] J. Haitsma and T. Kalker. A highly robust audio fingerprinting system with an efficient search strategy. Journal of New Music Research, 32(2):211–221, 2003.

[58] J. Haitsma, T. Kalker, and J. Oostveen. Robust audio hashing for content identification. In International Workshop on Content-Based Multimedia Indexing, volume 4, pages 117–124, 2001.

[59] R.I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[60] M.A. Hasan, M. Xu, X. He, and C. Xu. Camhid: Camera motion histogram descriptor and its application to cinematographic shot classification. IEEE Transactions on Circuits and Systems for Video Technology, 24(10):1682–1695, 2014.

[61] Y. Hochberg and A. C. Tamhane. Multiple comparison procedures. John Wiley & Sons, Inc., 1987.

[62] R. Hong, J. Tang, H.-K. Tan, C.-W. Ngo, S. Yan, and T.-S. Chua. Beyond search: Event-driven summarization for web videos. ACM Transactions on Multimedia Computing, Communications, and Applications, 7(4):35, 2011.

[63] B.K. Horn and B.G. Schunck. Determining optical flow. In 1981 Technical symposium east, Washington DC, USA, pages 319–331. International Society for Optics and Photonics, 1981.

[64] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank. A survey on visual content-based video indexing and retrieval. Part C: Applications and Reviews, IEEE Transactions on Systems, Man, and Cybernetics, 41(6):797–819, 2011.

[65] X.-S. Hua, L. Lu, and H.-J. Zhang. Ave: automated home video editing. In Proc. of the ACM International Conference on Multimedia, California, USA, pages 490–497, 2003.

[66] X.-S. Hua, L. Lu, and H.-J. Zhang. Optimization-based automated home video editing system. IEEE Transactions on Circuits and Systems for Video Technology, 14(5):572–583, 2004.

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

[68] O.D. Incel, M. Kose, and C. Ersoy. A review and taxonomy of activity recognition on mobile phones. BioNanoScience, 3(2):145–171, 2013.

[69] D Israel. Data analysis in business research: A step-by-step nonparametric approach. SAGE Publications, 2009.

[70] P ITU-T RECOMMENDATION. Subjective video quality assessment methods for multimedia applications. 1999.

[71] I. Ivanov, P. Vajda, J.S. Lee, and T. Ebrahimi. In tags we trust: Trust modeling in social tagging of multimedia content. IEEE Signal Processing Magazine, 29(2):98–107, 2012.

[72] N. Jiang, P. Grosche, V. Konz, and M. Müller. Analyzing chroma feature types for automated chord recognition. In Proc. of the AES 42nd Conference on Semantic Audio, Ilmenau, Germany, pages 1–10, 7 2011.

[73] T. Joachims. Learning to classify text using support vector machines: methods, theory and algorithms. Kluwer Academic Publishers, 2002.

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

[75] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE Transactions on Information Technology in Biomedicine, 10:156–167, 2006.

[76] L. Kennedy and M. Naaman. Less talk, more rock: automated organization of community-contributed collections of concert videos. In Proc. of the ACM International Conference on World Wide Web, Madrid, Spain, pages 311–320, 2009.

[77] J. F. Kenney. Mathematics of Statistics part I. Princeton, NJ: Van Nostrand, 1962.

[78] S.C. Kenyon and L. Simkins. Audio identification system and method, August 24 2010. US Patent 7,783,489.

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

[80] M. Kumano, K. Uehara, and Y. Ariki. Online training-oriented video shooting navigation system based on realtime camerawork evaluation. In Proc. of the IEEE International Conference on Multimedia and Expo (ICME), Toronto, Canada, pages 1281–1284, 2006.

[81] D. Lan, Y. Ma, and H. Zhang. A systemic framework of camera motion analysis for home video. In Proc. of the IEEE International Conference on Image Processing (ICIP), Barcelona, Spain, pages 289–292, 2003.

[82] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A.T. Campbell. A survey of mobile phone sensing. IEEE Communications Magazine, 48(9):140–150, 2010.

[83] I. Laptev. On space-time interest points. International Journal of Computer Vision, 64(2):107–123, 2005.

[84] O.D. Lara and M.A. Labrador. A survey on human activity recognition using wearable sensors. IEEE Communications Surveys and Tutorials,, 15(3):1192–1209, 2013.

[85] E. Le Grand and S. Thrun. 3-axis magnetic field mapping and fusion for indoor localization. In Proc. of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Hamburg, Germany, pages 358–364, 2012.

[86] A Lee. Virtualdub home page. URL: www. virtualdub. org/index, 2001.

[87] S. Lee and M. Hayes. Real-time camera motion classification for content-based indexing and retrieval using templates. In Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), FL, USA, pages 3664–3667, 2002.

[88] C. Lei and Y.H. Yang. Tri-focal tensor-based multiple video synchronization with subframe optimization. IEEE Transactions on Image Processing, 15(9):2473–2480, 2006.

[89] A. Lerch. An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics. Wiley-Blackwell, 2012.

[90] M. Li, B. Kim, and A. I. Mourikis. Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, pages 4712–4719, 2013.

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

[92] M. Liu. A study of mobile sensing using smartphones. International Journal of Distributed Sensor Networks, 2013:1–11, 2013.

[93] Y. Liu, M. Yang, and Z. You. Video synchronization based on events alignment. Pattern Recognition Letters, 33(10):1338–1348, 2012.

[94] B. Logan et al. Mel frequency cepstral coefficients for music modeling. In International Symposium on Music Information Retrieval, volume 28, page 5, 2000.

[95] X. Lou. Feature extraction for identification and classification of audio signals, March 20 2012. US Patent 8,140,331.

[96] L. Lu, H. Jiang, and H. Zhang. A robust audio classification and segmentation method. In Proc. of the ACM International Conference on Multimedia, Ottawa, Canada, pages 203–211, 2001.

[97] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proc. of the International Joint Conference on Artifical Intelligence (IJCAI), Vancouver, Canada, volume 81, pages 674–679, 1981.

[98] R. Macrae, J. Neumann, X. Anguera, N. Oliver, and S. Dixon. Real-time synchronisation of multimedia streams in a mobile device. In Proc. of the IEEE International Conference on Multimedia and Expo (ICME),Barcelona, Spain, pages 1–6. IEEE, 2011.

[99] A. Mahabalagiri, K. Ozcan, and S. Velipasalar. Camera motion detection for mobile smart cameras using segmented edge-based optical flow. In Proc. of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), Seoul, Korea, pages 271–276, 2014.

[100] A.C. Mahajan. Worldwide active smartphone users forecast 2014 2018: More than 2 billion by 2016. `http://dazeinfo.com/2014/12/18/worldwide-smartphone-users-2014-2018-forecast-india-china-usa-report/`, December 2014. Accessed: 2015-06-25.

[101] V. Makkapati. Robust camera pan and zoom change detection using optical flow. In National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, 2008.

[102] M. Mauch and S. Ewert. The audio degradation toolbox and its application to robustness evaluation. In Proc. of the International Society for Music Information Retrieval Conference (ISMIR), Curitiba, Brazil, pages 83–88, 2013.

[103] M.F. McKinney and J. Breebaart. Features for audio and music classification. In Proc. of the International Conference on Music Information Retrieval (ISMIR), Maryland, USA, volume 3, pages 151–158, 2003.

[104] T. Mei, X.-S. Hua, C.-Z. Zhu, H.-Q. Zhou, and S. Li. Home video visual quality assessment with spatiotemporal factors. IEEE Transactions on Circuits and Systems for Video Technology, 17(6):699–706, 2007.

[105] Y. Michalevsky, D. Boneh, and G. Nakibly. Gyrophone: Recognizing speech from gyroscope signals. In Proc. 23rd USENIX Security Symposium (SEC14), San Diego, CA, 2014.

[106] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. International journal of computer vision, 60(1):63–86, 2004.

[107] E. Mitchell, D. Monaghan, and N.E. O'Connor. Classification of sporting activities using smartphone accelerometers. Sensors, 13(4):5317–5337, 2013.

[108] A. Mittal, A.K. Moorthy, and A.C. Bovik. No-reference image quality assessment in the spatial domain. IEEE Transactions on Image Processing, 21(12):4695–4708, 2012.

[109] M. Müller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In Proc. of the International Conference on Music Information Retrieval (ISMIR), London, UK, pages 288–295, 2005.

[110] Meinard Müller. Information retrieval for music and motion. Springer, 2007.

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

[112] A. Nagasaka and T. Miyatake. Real-time video mosaics using luminance-projection correlation. Trans. IEICE, pages 1572–1580, 1999.

[113] N. Nguyen, D. Laurendeau, and A. Branzan-Albu. A robust method for camera motion estimation in movies based on optical flow. International Journal of Intelligent Systems Technology and Applications, 9(3):228–238, 2010.

[114] J.L.R. Ortiz. Smartphone-Based Human Activity Recognition. Springer, 2015.

[115] S. Paschalakis, K. Iwamoto, P. Brasnett, N. Sprljan, R. Oami, T. Nomura, A. Yamada, and M. Bober. The mpeg-7 video signature tools for content identification. IEEE Transactions on Circuits and Systems for Video Technology, 22(7):1050–1063, 2012.

[116] M. Pollefeys and D. Nister. Direct computation of sound and microphone locations from time-difference-of-arrival data. In Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas, USA, pages 2445–2448, 2008.

[117] A. Ranjan, R. Henrikson, J. Birnholtz, R. Balakrishnan, and D. Lee. Automatic camera control using unobtrusive vision and audio tracking. In Proc. of Graphics Interface, pages 47–54. Canadian Information Processing Society, 2010.

[118] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood. View-invariant alignment and match-

ing of video sequences. In Proc. of the IEEE International Conference on Computer Vision (ICCV), Nice, France, pages 939–945, 2003.

[119] P. Renevey and A. Drygajlo. Entropy based voice activity detection in very noisy conditions. In Proc. of European Conference on Speech Communication and Technology (EUROSPEECH), Aalborg, Denmark, pages 1887–1890, 2001.

[120] J. Revaud, M. Douze, S. Cordelia, H. Jégou, et al. Event retrieval in large video collections with circulant temporal encoding. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, Oregon, pages 2459–2466, 2013.

[121] D. L. Ruderman. The statistics of natural images. Network: Computation in Neural Systems, 5(4):517–548, 1994.

[122] M.K. Saini, R. Gadde, S. Yan, and W.T. Ooi. Movimash: online mobile video mashup. In Proc. of the ACM International Conference on Multimedia, Nara, Japan, pages 139–48, 2012.

[123] A. Samà et al. Human movement analysis by means of accelerometers: application to human gait and motor symptoms of parkinson's disease. PhD Thesis, 2013.

[124] A. Sama, C. Pérez-López, D. Rodriguez-Martin, J. Cabestany, J.M. Moreno, and A. Rodriguez-Molinero. A heterogeneous database for movement knowledge extraction in parkinsons disease. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2013.

[125] K. Saunders and J. Novak. Game development essentials: Game interface design. Cengage Learning, 2012.

[126] J.E. Schrader. Detecting and interpreting musical note onsets in polyphonic music. Master's thesis, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2003.

[127] E. Schubert, J. Wolfe, and A. Tarnopolsky. Spectral centroid and timbre in complex, multiple instrumental textures. In Proc. of the International Conference on Music Perception and Cognition, North Western University, Illinois, pages 112–116, 2004.

[128] B. W. Schuller. Intelligent audio analysis. Springer, 2013.

[129] H. R. Sheikh, A. C. Bovik, and L. Cormack. No-reference quality assessment using natural scene statistics: Jpeg2000. IEEE Transactions on Image Processing, 14(11):1918–1927, 2005.

[130] Z. Shen, S. Arslan Ay, S. H. Kim, and R. Zimmermann. Automatic tag generation and ranking for sensor-rich outdoor videos. In Proc. of the ACM International Conference on Multimedia, Arizona, USA, pages 93–102, 2011.

[131] P. Shrestha, M. Barbieri, and H. Weda. Synchronization of multi-camera video recordings based on audio. In Proc. of the 15th ACM international conference on Multimedia, Bavaria, Germany, pages 545–548, 2007.

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

[133] P. Shrestha, H. Weda, M. Barbieri, E. HL Aarts, et al. Automatic mashup generation from multiple-camera concert recordings. In Proc. of the ACM International Conference on Multimedia, Firenze, Italy, pages 541–550, 2010.

[134] P. Shrestha, H. Weda, M. Barbieri, and D. Sekulovski. Synchronization of multiple video recordings based on still camera flashes. In Proc. of the ACM International Conference on Multimedia, CA, USA, pages 137–140, 2006.

[135] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. ACM Transactions on Graphics, 25(3):835–846, 2006.

[136] J. Song, Y. Yang, Z. Huang, H. Shen, and J. Luo. Effective multiple feature hashing for large-scale near-duplicate video retrieval. IEEE Transactions on Multimedia, 15(8):1997–2008, 2013.

[137] J. Song, Y. Yang, Z. Huang, H.T. Shen, and R. Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In Proc. of the ACM international conference on Multimedia, Scottsdale, AZ, USA, pages 423–432, 2011.

[138] G.P. Stein. Tracking from multiple view points: Self-calibration of space and time. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CO, USA, volume 1, 1999.

[139] S. Suthaharan. No-reference visually significant blocking artifact metric for natural scene images. Signal Processing, 89(8):1647–1652, 2009.

[140] A.S. Thakur and N. Sahayam. Speech recognition using euclidean distance. International Journal of Emerging Technology and Advanced Engineering, 3(3):587–590, 2013.

[141] K. Uehara, M. Amano, Y. Ariki, and M. Kumano. Video shooting navigation system by real-time useful shot discrimination based on video grammar. In Proc. of the IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, volume 1, pages 583–586, 2004.

[142] Y. Ukrainitz and M. Irani. Aligning sequences and actions by maximizing space-time correlations. In Proc. of the European Conference on Computer Vision (ECCV), Graz, Austria, pages 538–550, 2006.

[143] A. Wang. The shazam music recognition service. Communications of the ACM, 49(8):44–48, 2006.

[144] A. Wang et al. An industrial strength audio search algorithm. In Proc. of the International Conference on Music Information Retrieval, Maryland, USA, pages 7–13, 2003.

[145] J. Wang, C. Xu, E. Chng, H. Lu, and Q. Tian. Automatic composition of broadcast sports video. Multimedia Systems, 14(4):179–193, 2008.

[146] L. Wang. Support Vector Machines: Theory and Applications, volume 177. Springer, 2005.

[147] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua. Event driven web video summarization by tag localization and key-shot identification. IEEE Transactions on Multimedia, 14(4):975–985, 2012.

[148] D. Wedge, D. Huynh, and P. Kovesi. Motion guided video sequence synchronization. In Proc. of the Asian Conference on Computer Vision (ACCV), India, pages 832–841, 2006.

[149] D. Wedge, D. Huynh, and P. Kovesi. Using space-time interest points for video sequence synchronization. In Proc. of IAPR Conference on Machine Vision Applications, Tokyo, Japan, pages 190–194, 2007.

[150] A. Whitehead, R. Laganiere, and P. Bose. Temporal synchronization of video sequences in theory and in practice. In Proc. of the IEEE Workshop on Motion and Video Computing, CO, USA, volume 2, pages 132–137, 2005.

[151] S. Wilk and W. Effelsberg. The influence of camera shakes, harmful occlusions and camera misalignment on the perceived quality in user generated video. In Proc. of the IEEE International Conference on Multimedia and Expo (ICME), Chengdu, China, pages 1–6, 2014.

[152] M.B. Winkler, K.M. Hover, A. Hadjakos, and M. Muhlhauser. Automatic camera control for tracking a presenter during a talk. In Proc. of the IEEE International Symposium on Multimedia (ISM), California USA, pages 471–476, 2012.

[153] Z. Wu and K. Aizawa. Self-similarity-based partial near-duplicate video retrieval and alignment. International Journal of Multimedia Information Retrieval, 3(1):1–14, 2014.

[154] J. Yan and M. Pollefeys. Video synchronization via space-time interest point distribution. In Proc. of the Advanced Concepts for Intelligent Vision Systems, Brussels, Belgium, pages 501–504, 2004.

[155] M.-C. Yeh and K.-T. Cheng. Video copy detection by fast sequence matching. In Proc. of the ACM International Conference on Image and Video Retrieval (CIVR), Greece, page 45, 2009.

[156] Z. Yu and Y. Nakamura. Smart meeting systems: A survey of state-of-the-art and open issues. ACM Computing Surveys (CSUR), 42(2):8, 2010.

[157] H. Zettl. Sight, sound, motion: Applied media aesthetics. Wadsworth Publishing, 2011.

[158] J. Zhang, S. H. Ong, and T. M. Le. Kurtosis-based no-reference quality assessment of jpeg2000 images. Signal Processing: Image Communication, 26(1):13–23, 2011.

[159] X. Zhou, L. Chen, and X. Zhou. Structure tensor series-based large scale near-duplicate video retrieval. IEEE Transactions on Multimedia, 14(4):1220–1233, 2012.

[160] J. Zhu, P. Wu, X. Wang, and J. Zhang. Sensec: Mobile security through passive sensing. In Proc. of the IEEE International Conference on Computing, Networking and Communications (ICNC), San Diego, CA, pages 1128–1133, 2013.

[161] L. Zini, A. Cavallaro, and F. Odone. Action-based multi-camera synchronization. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 3(2):165–174, 2013.