Human activity recognition using a wearable camera

by

Girmaw Abebe Tadesse

Bachelor of Science (Electrical Engineering) (Distinction) 2007

Master of Science (Telecommunications Engineering) 2012

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

March, 2018

Acknowledgments

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 de Catalunya – UPC

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

Acknowledgements

First, I thank my supervisor Professor Andrea Cavallaro whose guidance has been the backbone of this thesis and I learned a lot from his continuous suggestions. I also appreciate the support of Professor Andreu Catala and Professor Xavier Parra. I would also thank my independent assessor Dr. Miles Hansard for his comments during the various stages of my PhD. I am lucky to meet many wonderful colleagues in the span of four years, and our discussions have been truly beneficial. Our social activities have also been very enjoyable.

I am also very grateful for having a family that supports me in every step of my life. I am much indebted to my father, Mr. Abebe Tadesse, for being a great role model in life. I am here today because of your inspiring guidance. Many thanks to my friends whose advices I value dearly.

A very special thanks goes to my wife and best friend, Nafkote, whose affection has always been a source of courage and happiness. Hence, I proudly dedicate this PhD to you! Very lucky to have you. Afekrshalew!!

I, Girmaw Abebe Tadesse, confirm that the research included within this thesis is my own work, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature: Girmaw Abebe Tadesse
Date: 12 March 2018

First supervisor	Second supervisor	Author
Professor Andrea Cavallaro	**Professor Andreu Catala**	**Girmaw Abebe Tadesse**

Human activity recognition using a wearable camera

Abstract

Advances in wearable technologies are facilitating the understanding of human activities using first-person vision (FPV) for a wide range of assistive applications. In this thesis, we propose robust multiple motion features for human activity recognition from first-person videos. The proposed features encode discriminant characteristics from magnitude, direction and dynamics of motion estimated using optical flow. Moreover, we design novel virtual-inertial features from video, without using the actual inertial sensor, from the movement of intensity centroid across frames. Results on multiple datasets demonstrate that centroid-based inertial features improve the recognition performance of grid-based features.

Moreover, we propose a multi-layer modelling framework that encodes hierarchical and temporal relationships among activities. The first layer operates on groups of features that effectively encode motion dynamics and temporal variations of intra-frame appearance descriptors of activities with a hierarchical topology. The second layer exploits the temporal context by weighting the outputs of the hierarchy during modelling. In addition, a post-decoding smoothing technique utilises decisions on past samples based on the confidence of the current sample. We validate the proposed framework with several classifiers, and the temporal modelling is shown to improve recognition performance.

We also investigate the use of deep networks to simplify the feature engineering from first-person videos. We propose a stacking of spectrograms to represent short-term global motions that contains a frequency-time representation of multiple motion components. This enables us to apply 2D convolutions to extract/learn motion features. We employ long short-term memory recurrent network to encode long-term temporal dependency among activities. Furthermore, we apply cross-domain knowledge transfer between inertial-based and vision-based approaches for egocentric activity recognition. We propose sparsity weighted combination of information from different motion modalities and/or streams. Results show that the proposed approach performs competitively with existing deep frameworks, moreover, with reduced complexity.

Contents

Published work

Journal papers

[J1] Girmaw Abebe and Andrea Cavallaro. Hierarchical modeling for first-person vision activity recognition. *Neurocomputing*, 267:362–377, December 2017.

[J2] Girmaw Abebe, Andrea Cavallaro, and Xavier Parra. Robust multi-dimensional motion features for first-person vision activity recognition. *Computer Vision and Image Understanding (CVIU)*, 149:229–248, 2016.

Conference papers

[C1] Girmaw Abebe and Andrea Cavallaro. Inertial-vision: cross-domain knowledge transfer for wearable sensors. In *Proc. of International Conference on Computer Vision Workshops (ICCVW)*, pages 1392–1400, Venice, Italy, October 2017.

[C2] Girmaw Abebe and Andrea Cavallaro. A long short-term memory convolutional neural network for first-person vision activity recognition. In *Proc. of International Conference on Computer Vision Workshops (ICCVW)*, pages 1339–1346, Venice, Italy, October 2017.

[C3] Girmaw Abebe Tadesse and Andrea Cavallaro. Visual features for ego-centric activity recognition: A survey. In *Proc. of ACM Workshop on wearable systems and applications (WearSys)*, Munich, Germany, June 2018. https://doi.org/10.1145/3211960.3211978.

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

Abbreviations and symbols

List of Abbreviations

AP Average pooling, page 12

BAR Basketball activity recognition dataset, page 5

CNN Convolutional neural network, page 15

DogC Dogcentric datasets, page 38

DTCE Decision-level temporal context exploitation, page 20

FFT Fast Fourier transform, page 28

FMDF Fourier transform of motion direction across frames, page 27

FMMF Fourier transform of motion magnitude across frames, page 58

FMPF Fourier transform of grid motion per frame, page 27

FN False negative, page 21

FP False positive, page 21

FPV First-person vision, page 1

GF Grid features, page 58

GOFF Grid optical flow-based features, page 25

HMM Hidden Markov model, page 18

IAR Indoor ambulatory activity recognition dataset, page 5

JPL Jet propulsion laboratory, page 25

KNN K-nearest neighbour, page 18

LR Logistic regression, page 57

LSTM Long short-term memory, page 6

MBH Motion-based histogram, page 40

MDHF Motion direction histogram feature, page 27

MDHSF Motion direction histogram standard deviation feature, page 27

MMHF Motion magnitude histogram feature, page 27

MRGF Multi-resolution good feature, page 37

MTCE Model-level temporal context exploitation, page 20

OVA One-vs.-all, page 17

OVO One-vs.-one, page 17

RANSAC Random sample and consensus, page 13

RMF Robust motion feature, page 24

RNN Recurrent neural networks, page 6

SURF Speeded up robust features, page 40

SVM Support vector machine, page 18

TCE Temporal context exploitation, page 20

TN True negative, page 21

TP True positive, page 21

TPV Third-person vision, page 1

VIF Vision-based inertial features, page 25

List of symbols

$tsum(\cdot)$ temporal sum, page 29

$(\cdot)^t$ time elapsed for computing the argument, page 35

β_d number of direction bins, page 30

β_m number of magnitude bins, page 29

$\boldsymbol{f}_{1n}$ a set of grid features, page 29

$\boldsymbol{f}_{1n}^j$ j^{th} element of $\boldsymbol{f}_{1n}$, page 29

$\boldsymbol{V}$ video sequence, page 4

$\ddot{W}$ virtual acceleration, page 34

$\ddot{W}_k$ acceleration at the k^{th} frame, page 34

$\dot{W}$ virtual velocity, page 34

$\dot{W}_k$ velocity at the k^{th} frame, page 34

Γ_n magnitude of $\dot{W}$ and $\ddot{W}$ components combined with Υ_n, page 34

Λ_n cascade of time domain inertial features of Υ_n, page 35

$\mathcal{A}$ Accuracy, page 40

$\mathcal{C}$ set of egocentric activities, page 4

$\mathcal{F}$ F-score, page 40

$\mathcal{M}_{pq}$ firs-order image moment $p,q \in (0,1)$, page 34

$\mathcal{P}$ Precision, page 21

$\mathcal{R}$ Recall, page 21

$\mathcal{S}$ Specificity, page 40

ν overlapping percentage between V_n and V_{n+1}, page 28

ω_x maximum horizontal projection displacement, page 39

ω_y maximum vertical projection displacement, page 39

Ω_k^x horizontal projection of F_k, page 39

Ω_k^y vertical projection of F_k, page 39

Φ set of hierarchical model parameters, page 57

Φ_k k^{th} element of Φ, page 57

$\phi_{\iota e}$ model parameters corresponding to ι^{th} feature group at M_e, page 57

Ψ_n low frequency component of U_n, page 33

$\mathbf{V}_{train}$ set of training video sequences, page 57

Θ set of high-level activity model parameters, page 57

θ direction component, page 12

Θ_n cascade of frequency domain features from Υ_n, page 35

$\tilde{O}_n$ $fnorm(\cdot)$ of O_n, page 29

Υ_n virtual-inertial equivalent of V_n, page 34

Ξ_n zero-crossing of non-magnitude vectors in Υ_n, page 35

B_k grid representation of E_k, page 27

$B_{k_x}^g$ horizontal component of the g^{th} grid in B_k, page 28

$B_{k_y}^g$ vertical component of the g^{th} grid in B_k, page 28

B_{kn} grid optical flow of the k^{th} frame in V_n , page 28

$binom(\cdot)$ binomial operation, page 58

C width of F_i in pixels, page 27

c_A sampled indices of C, page 27

c_j j^{th} class in $\mathcal{C}$, page 4

$concat(\cdot)$ concatenation of argument vector, page 33

E_k dense optical flow between frames F_k and the F_{k+1}, page 27

F_k the k^{th} frame in $\boldsymbol{V}$, page 4

f_ι ι^{th} low-level feature group, page 57

$fnorm(\cdot)$ per-frame normalisation, page 29

G number of grid in B_k, page 27

I_n magnitude equivalent of B_n , page 29

J_n direction equivalent of B_n, page 30

K_n Fourier response of P_n, page 31

L number of frames in a sample V_n, page 28

M_e e^{th} node in the hierarchy, page 57

N_c number of activities in $\mathcal{C}$, page 4

N_f number of frequency bands in T_n, page 31

N_g length of grid-based feature (GOFF), page 33

N_i dimension of virtual-inertial feature ($\boldsymbol{f}_{2n}$), page 35

N_l number of low frequency coefficients in Θ_n, page 35

N_s number of low frequency coefficients in Ψ_n, page 33

N_v number of frames in $\boldsymbol{V}$, page 4

O_n histogram of I_n using β_m bins, page 29

P_n direction histogram of B_n, page 30

R height of frame F_k in pixels, page 27

r_A sampled indices of R, page 27

T_n K_n grouped into non-overlapping bands, page 31

U_n Fourier response of B_n per frame, page 33

V_n n^{th} window (sample) in $\boldsymbol{V}$, page 4

W intensity centroid, page 34

W_k intensity centroid of the k^{th} frame, page 34

X_k^b horizontal baseline projection velocity, page 39

Y_k^b vertical baseline projection acceleration, page 39

Z_k combines $\dot{W}_k$ and $\ddot{W}_k$, page 34

Rj j^{th} recording of the IAR dataset, page 42

Sj j^{th} subject of the BAR dataset, page 42

Chapter 1

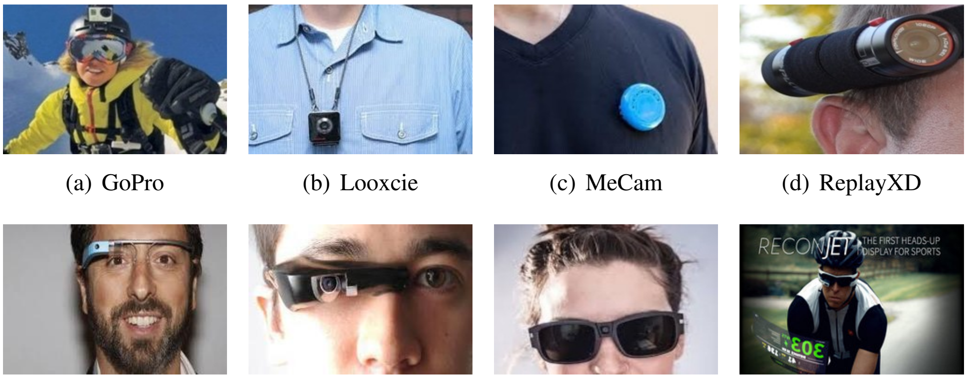
Introduction

1.1 Motivation

Due to the emergence of pervasive computing as well as the development of small, efficient and low power sensing devices, the use of wearable sensors becomes a common practice across different research domains. Among these devices, wearable visual sensors, i.e. wearable cameras, are getting more attention from both the industry and academic communities (Fig. 1.1).

Computer vision research has been traditionally focused on analysing video content recorded from third-person point-of-view, i.e. third-person vision (TPV). However, wearable cameras bring a new computer vision research, i.e. first-person vision (FPV) that provides egocentric information of a subject, i.e. records almost what the subject sees, and results in rich and continuous data while the privacy is partially protected as the subject is not directly seen in the recorded videos.

Wearable camera systems can be categorized according to their applications as lifelogging, activity recording and eye tracking. Lifelogging is a digital capture of life experiences typically through mobile sensors [10, 21, 22, 28, 40, 41]. It can also be considered as an automated biography [20]. Lifelogging oriented wearable cameras, such as SenseCam, Autographer and Memoto, are characterized by their tendency to make automatic snapshots in FPV whenever they are triggered. Although most lifelogging devices are equipped with other built-in sensors, they do not have a video recording feature. Activity recorders refer to wearable vision sensors, such as GoPro, Looxcie, and Google Glass, which have been used to record high-quality egocentric



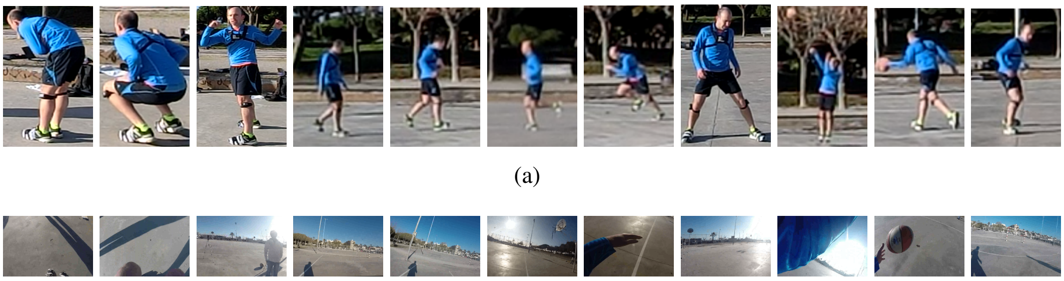
(a) GoPro (b) Looxcie (c) MeCam (d) ReplayXD

(e) Glass (f) Eyetap (g) Pivothead (h) Recon Jet

Figure 1.1: Activity recording cameras (©Google).

videos, e.g. in sports and augmented reality systems [4, 71, 81, 104, 108]. Eye trackers are different from activity recorders with their additional feature to collect user's attention in a scene, which will be valuable for human activity recognition and behaviour understanding [13, 30].

Human activities can be motion-oriented individual proprioceptive activities, e.g. *Walk* [25, 50, 64, 103, 104, 106, 108, 109], or interaction-based activities such as person-to-object interactive daily activities, e.g. *Cook, Read, Write* and *Web browse* [30, 66, 69]; and person-to-person interactions, e.g. *Handshake, Hug* and *Punch* [65, 80]. In particular, proprioceptive activities involve a full-body motion caused by the movement of muscles and joints (Fig. 1.2) and are of interest in a range of tasks from (self-) monitoring of the elderly to performance analysis of athletes. Some of the proprioceptive activities particularly involve upper-body motion, e.g. *Bow* and *Waist-turn* [108]. These activities can also be more specific sport activities, e.g. *Dribble, Defend, Sprint, Pivot* and *Shoot* [4].



(a)

(b)

Figure 1.2: Some of the proprioceptive activities considered in this work; (a): activities viewed from an external camera; (b) frames from the first-person vision acquired by a wearable camera while a user performs the corresponding activity in the top row. The activities in order are *Bow, Sit-Stand, Left-right turn, Walk, Jog, Run, Sprint, Pivot, Shoot, Dribble* and *Defend*.

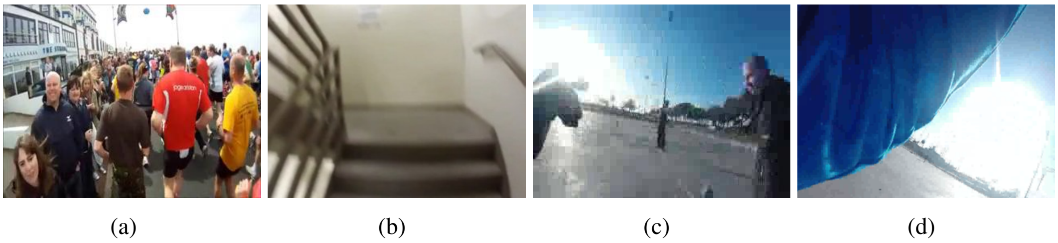


Figure 1.3: Sample frames showing some of the challenges in proprioceptive activity recognition in first-person vision: (a) outlier motions; (b) illumination changes; (c) motion blur; and (d) self-occlusions.

1.2 Challenges

The main challenges for the accurate and robust analysis of activities from FPV include occlusions, motion blur, illumination changes and local motions that do not reflect the activity being performed (e.g. other people captured by the camera) (Fig. 1.3). In addition to this, the specific mounting position of the camera makes the problem more difficult because of self-occlusions [94] (especially for chest-mounted cameras) and spurious motions [71, 104] (especially for head-mounted cameras). Other sensors, such as inertial sensors, could be used to complement the video acquisition in order to address these challenges [64, 104]; however, the signals generated by these additional sensors would require synchronization with the video stream.

Proprioceptive activities are characterized by full-body (global) motion [4], hence, it is necessary to extract features that encode magnitude, direction and dynamics of the global motion while smoothing the noisy local motions [70, 81, 104, 108]. In addition, effective integration of multiple features is required, which exploits the discriminative characteristics of each feature group.

Besides the motion dynamics inside a short video segment, it is crucial to encode the long term motion dynamics that reflect the temporal dependency among activities, which is often implemented with complex modelling frameworks [50, 104]. In addition to temporal relationships, the exploitation of their hierarchical relationship also provides plausible classification of the activities.

The lack of validation datasets is another challenge in the proprioceptive activity recognition in FPV. This is partly because FPV-based research is relatively at its early stage compared to traditional (third-person) vision research. Particularly, the lack of data makes it difficult to explore the recent advances of deep neural architectures in the field.

1.3 Problem formulation

Let $\boldsymbol{V}$=$(F_k)_{k=1}^{N_v}$ be a video sequence captured from FPV and it contains N_v frames. Let $\mathcal{C} = \{c_j\}_{j=1}^{N_c}$ denote a set of N_c ego-centric activities (classes). $\boldsymbol{V}$ might contain only one activity or more activities that occur one after the other. Hence, a windowed segment, V_n, often corresponds to a single activity $c_j \in \mathcal{C}$. V_n segmented on the transition between two activities takes the one with the longest duration as its label. The main aim of the thesis is to exploit the short- and long-term motion dynamics in order to recognise the activity label corresponding to V_n. In addition to the temporal encoding, hierarchical relationships among activities, environment (appearance) and multi-modal information can be utilised. While allowing local motions due to occlusions, we assume that a global motion is dominant over the majority of the frames in V_n.

1.4 Contributions

The main contributions of the thesis are the following:

- To design multiple motion features in FPV that encode the dynamics, direction and magnitude of optical flow data both in time and frequency domains. In addition, we propose novel virtual-inertial features from video without using the actual inertial sensor, which complement the grid-based inertial features [1, 4]. Extensive experiments that include sensitivity to parameter values and robustness to noise demonstrate the goodness of the method.
- To propose a framework that encodes the hierarchical relationships among activities and exploits the temporal continuity during modelling and decision [1]. The outputs of the hierarchy from previous samples are temporally weighted and confidence-based decision smoothing is applied during classification. In addition to grid-based and centroid-based low-level features, we employ pooled frame-level appearance features in the pipeline that has been demonstrated to improve the performance. To address the class imbalance problem, we propose a balancing strategy that undersamples data-rich activities and oversamples data-scarce activities.
- We propose a novel global motion representation that contains a stack of spectrograms of different axial motion components [3]. The stacking helps us to employ 2D convolutions rather than 3D to extract/learn high-level short-term temporal information, which also exploits the intrinsic relationships between motion components and reduces the amount of

training data necessary and therefore the complexity. The LSTM is shown to improve the recognition performance by encoding the long-temporal dependency.

- We apply a cross-domain knowledge transfer between virtual and inertial data [2]. Particularly, by stacking the spectrograms of different components of inertial data, we employ existing image-based deep convolutional networks to encode motion features rather than designing a dedicated deep framework on the inertial data and train it from scratch. In addition, we propose a sparsity weighted combination of multiple features from different modalities, e.g. inertial and visual streams. To the best of our knowledge, this is the first work that takes into account deep features extracted from egocentric inertial and FPV data.
- Two datasets are collected and made publicly available to facilitate research in FPV. The datasets include indoor activity recognition (IAR) and basketball activity recognition (BAR) .

1.5 Organization of the thesis

The thesis is organized as follows. **Chapter 2** reviews the existing methods in FPV-based proprioceptive activity recognition related with data capture, extraction of multiple features, classifiers, temporal context exploitation, feature fusion and performance evaluation.

Chapter 3 presents the extraction of multiple motion features that encode the direction, magnitude and dynamics of optical flow motion data, both in time and frequency domains. The extraction of novel virtual-inertial features from the movement of intensity centroid is also presented. Existing hand-crafted features are compared against the concatenation of the proposed motion features and validated on multiple datasets.

Chapter 4 introduces a multi-layered modelling framework that contains both hierarchical and temporal modelling of activities. The hierarchy is designed manually in order to encode the natural hierarchical relationships among activities, whereas the temporal relationship is encoded both during modelling and decision stages. Features from multiple motion sources are weighted by their temporal distance from a current sample and a confidence-based decision is applied later. New temporal pooling operations are presented on frame-level appearance features. The framework is validated on the largest FPV datasets of proprioceptive activities and compared against the state-of-the-art methods.

Chapter 5 focuses on the use of high-level motion features extracted using existing deep neural networks. This contains an intra-sample encoding, i.e. a novel stacked spectrogram representation of short-term global motion information that enables us to employ 2D convolutions rather than the more complex 3D convolutions. This also helps to exploit existing convolutional models trained on large image datasets. Hence, high-level motion features are extracted from the spectrograms of optical flow and centroid movement without training a dedicated deep convolutional neural networks. Inter-sample temporal encoding is performed using long short-term memory (LSTM) recurrent neural network (RNN) , which encodes the long-term temporal dependency among activities. The proposed CNN features in FPV are compared with existing video representations. We also present cross-domain knowledge transfer between vision-and inertial-based recognition of egocentric human activities. Sparsity-weighted combination of multi-modal information is presented that weight the information based on its discriminative characteristics. The sparsity weighting is validated on multiple inertial and visual datasets.

Chapter 6 concludes the thesis and outlines future research directions on FPV-based proprioceptive activity recognition.

Chapter 2

State of the art

2.1 Introduction

In this chapter, we describe the state-of-the-art proprioceptive activity recognition in FPV. We provide an in-depth description of the main stages, namely data capture, feature extraction and classification (see Fig. 2.1). *Data capture* represents the collection of first-person videos using wearable cameras that may have different specifications and can be mounted at different parts of the human body. *Feature extraction* refers to the encoding of robust features that help to distinguish activities. The features can be manually designed using task-specific knowledge (handcrafted) or learned from data. Efficient fusion of features is necessary to integrate information from multiple streams or modalities. *Classification* includes the recognition of activities by exploiting the discriminative characteristics of the features using activity modelling procedures. Moreover, we discuss key components of the recognition pipeline such as temporal context exploitation, which encodes the temporal relationships among activities and improves the recognition performance. Existing solutions that address FPV challenges, such as self-occlusion and motion blur, are also discussed across the stages in the pipeline.

This chapter is organized as follows. Section 2.2 describes data capture modes using wearable cameras and publicly available datasets for validation. Section 2.3 presents state-of-the-art motion encoding and filtering techniques followed by feature extraction, learning and fusion. Section 2.4 discusses classifiers, temporal context exploitation techniques and decision-level fusion of different feature groups. Multiple recognition performance metrics and decomposition

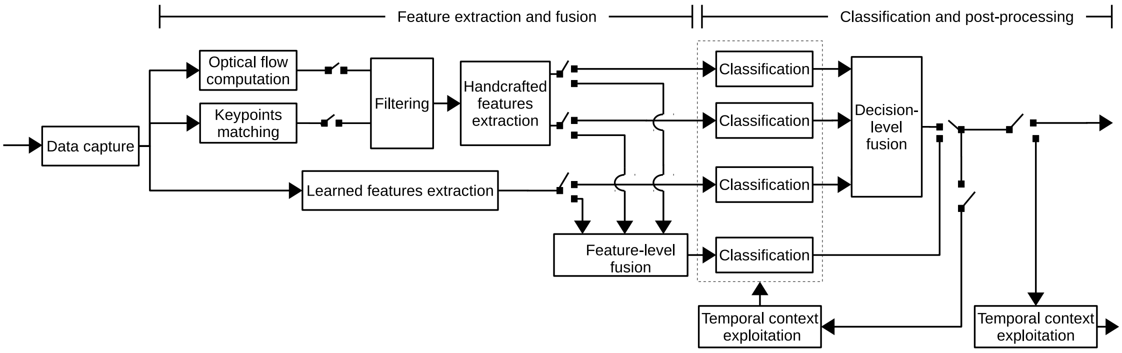

Feature extraction and fusion
Classification and post-processing
Data capture
Optical flow computation
Keypoints matching
Filtering
Handcrafted features extraction
Learned features extraction
Feature-level fusion
Classification
Classification
Classification
Classification
Decision-level fusion
Temporal context exploitation
Temporal context exploitation

Figure 2.1: A unified pipeline for a generic proprioceptive activity recognition system that uses data from a wearable camera. The switches account for alternative state-of-the-art methods.

strategies of available data to train and test sets are also described in-detail. Finally, Section 2.5 concludes the review and outlines future directions.

2.2 Capturing proprioceptive activities

Commonly studied proprioceptive activities (see Fig. 2.2) involve full-body motion, e.g. *Walk, Run, Go-upstairs, Go-downstairs, Sit-down* and *Stand-up* or upper-body motion, e.g. *Bow* and *Waist-turn* [71, 106, 108]. Examples of key frames for some of the activities are shown in Fig. 2.3. Stationary states, such as *Sit, Stand* and *Lie*, might contain motion while the user is stationary [71].

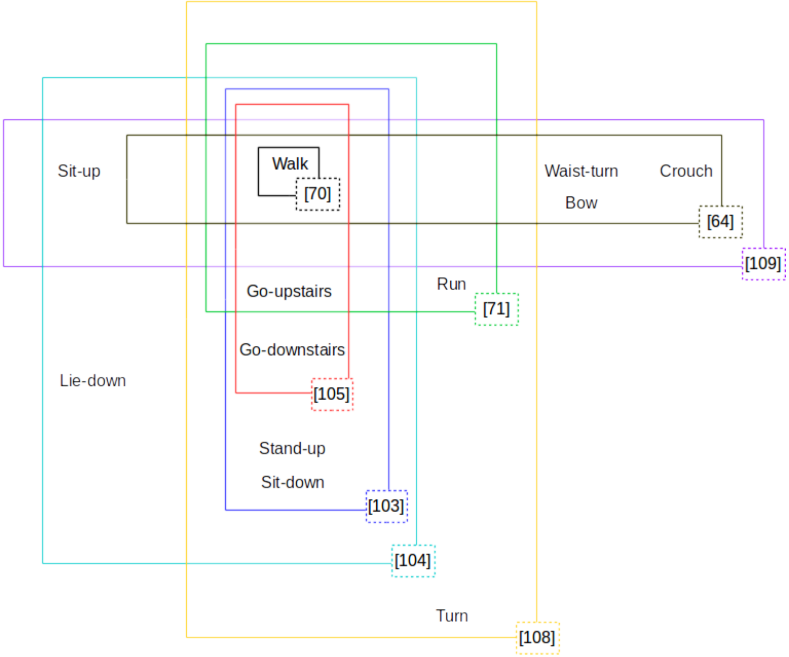


Figure 2.2: Venn diagram of different related works that shows commonly studied proprioceptive activities. Each box represents an existing work referred as shown in its bottom right corner. Inside each box are shown the corresponding proprioceptive activities studied.

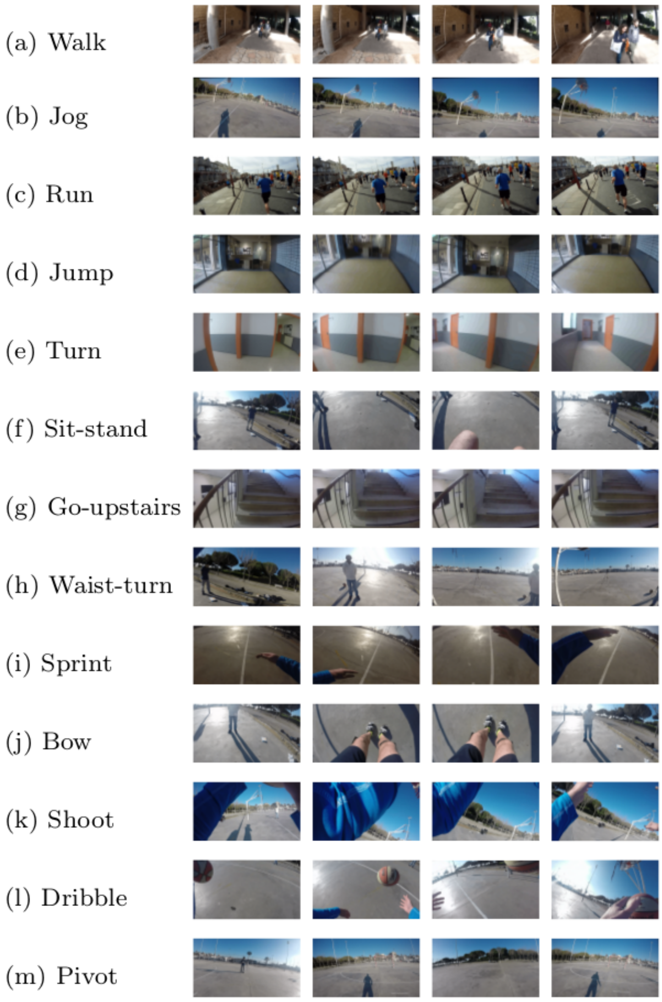


Figure 2.3: Sample key frames from first-person videos of different activities.

Data capture for proprioceptive activities often employs either chest- or head-mounting of wearable cameras. *Chest-mount* capture provides stable videos, but it might contain self-occlusions, particularly due to user's hands (see Fig. 2.3 (k)) [64, 108, 109]. *Head-mount* is subject to large-scale and noisy head-motion though it provides more FPV characteristics, i.e. both the subject and the camera tend to share similar field of view [50, 71]. A camera can also be embedded on eyeglasses that achieve a more natural positioning to record what the user sees [103, 104, 105, 106]. Beyond the proprioceptive activity framework, other mounting positions such as *wrist* [60] can be employed to record objects that are manipulated by a user's hands. Recent trends show that a *head-mount* is often preferred over a *chest-mount* for data capture (see Table 2.1) as the former possesses better FPV characteristics.

Proprioceptive activity recognition from first-person videos is an emerging field that does not have a standard validation dataset yet. The few existing datasets are often affected by the class imbalance problem, which is the disproportional representation of activities in a dataset [70, 71]. To facilitate the improvement of the state of the art, we list publicly available datasets that can be used for validation in Table 2.1.

Table 2.1: Summary of public datasets that can be used for the validation of proprioceptive activity recognition frameworks. #: number; I: indoor; O: outdoor; IO: indoor/outdoor; C: chest; H: head; E: eyeglasses; Res/Fr: resolution and frame rate; D: contains different resolutions and frame rates; IMU: inertial measurement unit; Mic.: microphone; ET: eye tracker; TPV: third-person vision; NS: not specified; S and F are segment- and frame-level annotations.

Dataset	Reference	Environment	Mount	Activities	Res/Fr	Duration (hrs)	Videos (#)	Subjects (#)	Other source	Annotation
IAR	[4]	I	C	Walk, Run, Sit down, Stand up, Jump, Go-upstairs/downstairs, Turn	1080*p*60	0.67	394	1		S
BAR	[4]	O	C	Bow, Defend, Dribble, Jog, Left-right turn, Shoot, Sit-stand, Sprint, Walk	720*p*30	1.20	331	4		S
HUJI	[70, 71]	IO	H	Walk, Sit, Ride bus, Drive, Static, Stand, Cycle, Run, Go-upstairs, Ski, Horseback ride, Sail, Box, Cook	D	82	122	NS		S
LENA	[89]	IO	H	Watch videos, Read, Browse internet, Walk, Run, Eat, Go-upstairs/downstairs, Phone talk, Talk people, Write, Drink, Housework	960*p*30	2	260	10		S
MMD	[90]	IO	H	Walk, Ride elevator/escalator up/down, Sit, PC activities, Eat, Drink, Phone text, Make phone calls, Run, Push-ups, Sit-ups, Cycle	720*p*30	1.50	200	NS	IMU	S
UEC Park	[50]	O	H	Different ego actions (29) in a park, e.g. Jog and Walk	480*p*60	0.50	2	1		S
FPSI	[31]	IO	H	Individual activities (e.g. Walk) and social interactions (e.g. Discussion)	720*p*30	42	113	8		S
ADL	[69]	I	C	Daily indoor activities (18), e.g. Brush teeth	960*p*30	10	20	20		F
UTE	[53]	IO	H	Lifelog of activities such as Eat, Attend lecture, Drive and Cook	320*p*15	17	10	4		F
CMU-MMAC	[91]	I	H	Cook five different recipes: brownies, pizza, sandwich, salad and scrambled eggs	600*p*30	17	185	39	Mic., IMU	F
GTEA	[33]	I	H	Make hotdog sandwich, instant coffee, peanut butter sandwich, jam sandwich, sweet tea, cheese sandwich and coffee with honey	720*p*30	0.50	28	4		F
GTEA Gaze	[32]	I	E	Make American breakfast, Turkey sandwich, cheese burger, Greek salad, pizza, pasta, salad and afternoon snack	480*p*30	1	17	14	ET	F
GTEA Gaze+	[32]	I	E	Similar to GTEA Gaze activities above but more complex	960*p*24	5	30	10	ET	F
NUSFPID	[65]	I	H	Human-human (e.g. Wave), human-object (e.g. Typing), human-object-human interactions (e.g. Pass object)	720*p*30	0.25	260	NS	TPV	S
UTokyo	[66]	I	E	Read book, Watch video, Copy text on screen, Write and Browse internet	960*p*30	2	60	5	ET	S
EGO-SENSORS	[7]	IO	E	Bike, Jog, Run, Stand up, Walk and Wander	*NSp*30	2.78	NS	NS	IMU	S

Note that while some of these datasets might be originally designed for the validation of interactive activities (e.g. social interactions [31] and cooking activities [91]); they can be utilised for proprioceptive activity validation by applying appropriate reannotation, e.g. relabelling the majority of *Cook* activities to *Stand*. This provides diverse representation of proprioceptive activities in different settings. We also provide important characteristics of the datasets, such as environment type, frame rate, resolution, number of subjects and duration. After the first-person videos are collected, the video data are often resized and resampled into smaller values in order to improve the processing speed [71, 105].

2.3 Feature extraction and fusion

This section discusses how apparent motion is estimated in first-person videos of proprioceptive activities. The motion is often dominated by a global motion due to the user's full- or upper-body motion, and it can be encoded as optical flow or displacement of keypoints. Handcrafted features that encode salient characteristics of the motion are extracted. Handcrafting refers to the manual designing of the features tailored to solve a specific problem. We also describe the learning of features from data using deep neural networks and the feature-level fusion of multiple feature groups. The summary is given in Table 2.2.

2.3.1 Optical flow

Optical flow is the main source of motion features (see Fig. 2.4) for proprioceptive activity recognition [70, 71, 104]. It can be derived using a direct motion estimation technique [45] that achieves subpixel accuracy. A grid representation of the optical flow is often preferred to a dense representation in order to avoid redundancy in the assumption of global motion domination [70, 104, 108]. According to the level of complexity employed, we can categorize optical flow-based features into three groups: raw grid, direction and magnitude histogram, and frequency-domain features.

Raw grid features are obtained from the optical flow data with minimum processing. This includes average pooling (AP) [103, 104, 105, 106], which concatenates horizontal and vertical grid components across frames. Poleg et al. [70] used the *radial projection response* of grid optical flow vectors to discriminate *moving* from *stationary*. Similarly, hard-coded rules on grid vector direction (θ) were employed to classify activities, e.g. *Left-turn* satisfies $0^\circ < \theta < 90^\circ$ or

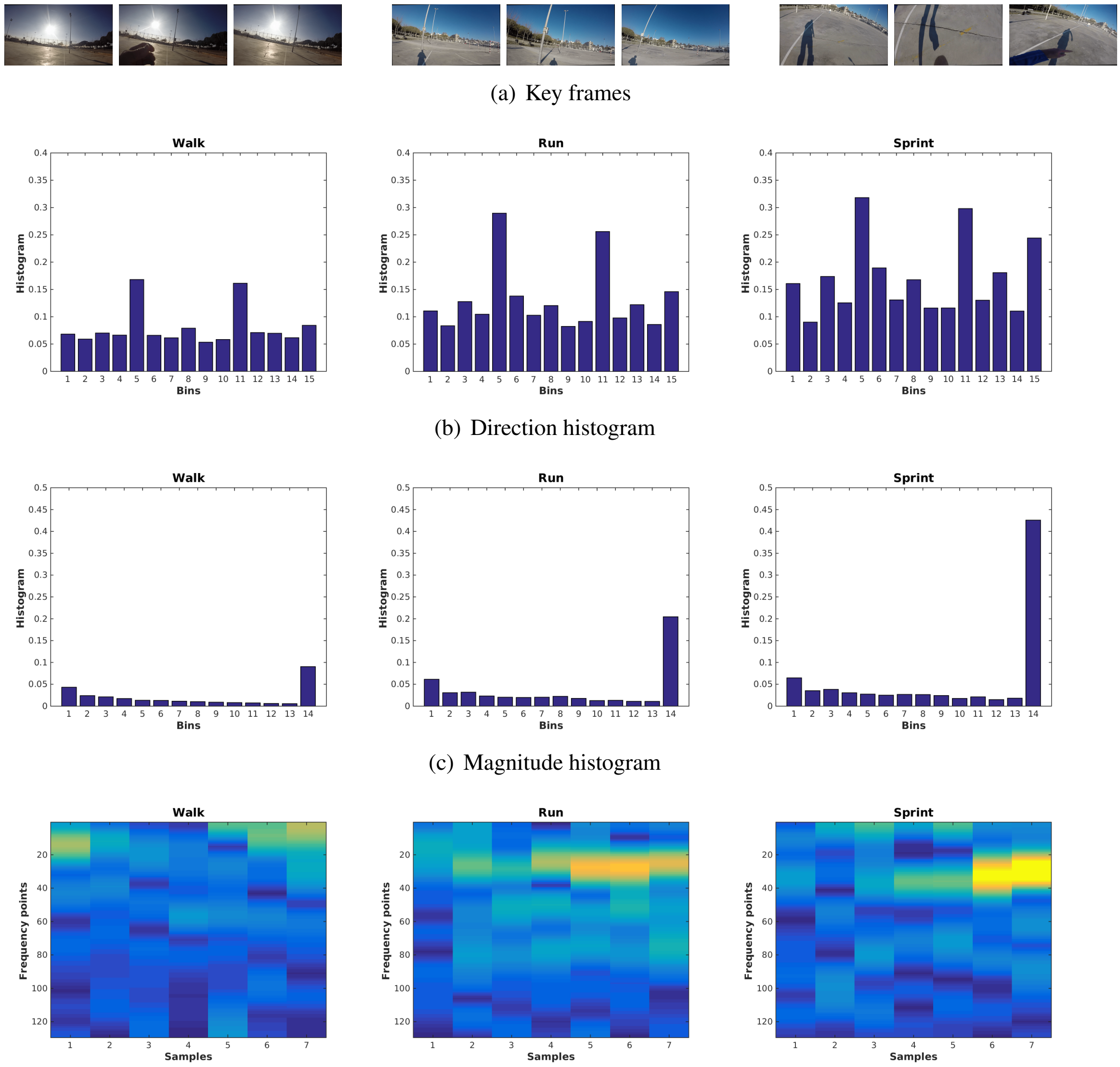
Table 2.2: Summary of the state-of-the-art feature extraction, motion filtering and feature-level fusion of multiple feature groups. RANSAC: random sample consensus.

			[81]	[71]	[70]	[104]	[103]	[106]	[50]	[108]	[64]	[105]	[109]
Features	Optical flow	Raw grid feature			✓	✓	✓	✓			✓	✓	
		Grid direction histogram	✓						✓				
		Grid magnitude histogram							✓				
		Grid gradient histogram	✓										
		Grid frequency feature							✓				
	Keypoint displacement	Direction histogram								✓			✓
	Learned features	Pooled deep-appearance feature	✓										
		Deep motion feature		✓									
	Filtering	Thresholding			✓					✓			✓
		RANSAC-based filtering							✓	✓		✓	✓
		Gaussian smoothing			✓								
		Average pooling				✓	✓	✓				✓	
	Feature-level fusion		✓		✓				✓			✓	

$270° < \theta < 360°$ in [64]. The discriminating capability of *raw grid features* is limited since specific motion characteristics such as magnitude and direction are not exploited enough in order to achieve a robust and compact motion feature representation.

Direction (see Fig. 2.4 (b)) and *magnitude* (see Fig. 2.4 (c)) histograms of the flow vector provide more discriminant features [50]. Motion magnitude and direction components are generally exploited separately to increase the discrimination. For example, *Sit-down* and *Stand-up* can be distinguished by exploiting their motion directions, whereas magnitude information helps differentiate *Walk* and *Sprint*. The histogram provides a compact representation of the direction and/or magnitude components of the grid flow data [50, 81]. The histogram might be applied using joint spatial and direction bins [81], or joint magnitude, direction and magnitude variance bins [50]. The inclusion of spatial bins [81] is comparatively less effective since multiple proprioceptive activities can be performed in a similar environment. In addition, Ryoo et al. [81] employed motion boundary histogram as one of the multiple motion features from optical flow data that compensates the camera motion, and it is obtained by applying a spatial derivative on the horizontal and vertical optical flow components separately, followed by a magnitude-weighted histogram of motion direction [99].

Frequency-domain features encode temporal characteristics in the frequency-domain, and they improve discrimination using frequency-domain analysis. The standard deviation of magnitude and direction components in time-domain can only encode the high-level motion dynamics. However, frequency-domain analysis exploits the low-level motion dynamics and helps distinguish similar activities, e.g. *Sprint* and *Run* (see Fig. 2.4 (c)) as *Run* involves less frequent changes in motion dynamics [4]. Kitani et al. [50] extracted frequency-domain features from the horizontal and vertical grid components independently. The frequency-domain features can be represented by selecting the low-frequency coefficients that are robust to noise, but the represen-



(a) Key frames

(b) Direction histogram

(c) Magnitude histogram

(d) Direction spectrogram

Figure 2.4: Examples of features derived from the optical flow to encode different motion characteristics of *Walk*, *Run* and *Sprint*. (a) Key frames of the activity in the corresponding column; (b) direction histogram representations; (c), (d) the activities can be easily discriminated using the magnitude histogram and the frequency-domain analysis of the direction information, respectively.

tation does not include the full spectrum characteristics. Similarly to the number of magnitude and direction bins for the histogram representations, the numbers of frequency bands and low frequency coefficients need to be carefully set to avoid under/over-quantization.

2.3.2 Keypoint displacement

Spatial change of keypoints across frames can also be used to infer apparent motion, which involves detection, description and matching of interest points. The detection of the interest points can be either *blob-based* or *corner-based* based on their spatial structure. Examples of blob-based detectors include scale-invariant feature transform [56], speeded-up robust features [12] and center surround extremas [5]. Examples of corner-based detectors include features from accelerated segment test [78], adaptive and generic corner detection based on the accelerated segment test [61] and binary robust invariant scalable keypoints [54]. After a keypoint is detected, its neighbourhood is described fulfilling robust characteristics, such as invariant to rotation, using a *binary* or *non-binary descriptor* [36, 82, 95]. Using a binary descriptor makes matching computationally easier since Euclidean distance can be replaced by a Hamming distance that can be calculated using a bitwise XOR operation [6, 54, 79].

Zhang et al. [109] proposed a keypoint-based feature, inspired by the earlier work of Shi and Tomasi [85]. The matching output of the descriptors was refined by uniqueness (one-to-one correspondence) and epipolar constraints [39]. The frame motion was estimated as a set of displacement vectors between matched descriptor pairs. The direction of each displacement vector that satisfied a magnitude threshold was quantized using a histogram representation. The work was later upgraded to achieve multi-resolution detection of interest points in [108]. Average standard deviation [109] and combined standard deviation [108] of the histogram representation were employed to encode the temporal characteristics that improved the classification accuracy.

Since Zhang et al. [108, 109] did not exploit the magnitude information and encode the dynamics in-detail, their recognition performance is inferior to more advanced features that exploit those characteristics [50]. Generally, keypoint-based methods can handle large displacements. However, they are less effective in poorly textured first-person videos, which are often blurred due to high egomotion.

2.3.3 Learned features

Convolutional neural networks (CNNs) have been successfully applied to learn high-level features from data [47, 71, 83]. Similarly, FPV features can also be learned from egocentric videos using deep neural networks. Motion features can be directly learned [58, 71, 88, 107] or derived from the temporal pooling of deep appearance features [81].

Poleg et al. [71] proposed a compact CNN taking a sparse grid volume as input and learned motion features that are demonstrated to outperform the handcrafted features in their previous work [70]. The network was derived from the temporal component of an existing two-stream network [86], and it was designed with a 3D convolutional layer followed by a 3D pooling to handle the 3D input data [96]. 2D convolution layers were applied afterwards, which suppress the long-term temporal dependency too early in the network. Though the deep motion features are shown to be transferable across datasets of similar nature [71], the interpretation of the knowledge learned at different layers requires further study.

On the other hand, motion can be inferred from the variation of per-frame appearance descriptors, which are learned from data, using temporal pooling operations. Ryoo et al. [81] proposed different pooling strategies that treat each descriptor element across frames as time-series data. The summation and maximum pooling of the appearance features are not effective to encode multi-resolution temporal variations. But *time-series gradient* pooling achieves encoding of short and long temporal variations by applying first-order temporal derivative on each descriptor element [81]. The summation and histogram of positive and negative gradients provide two-stream variation encoding. Comparatively, the histogram representation describes the short-term variation more effectively since its score depends only on the sign of the gradients. This technique could also be applied to handcrafted appearance descriptors such as the histogram of oriented gradients [81].

The discriminative capacity of the feature space can be further improved by encoding more detailed temporal characteristics in the frequency domain. In addition to the temporal variation of the appearance, static appearance information can also be useful when activities are correlated with certain environmental settings, e.g. *Going upstairs* involves staircases [81, 99].

2.3.4 Filtering and feature-level fusion

A variety of filtering approaches are applied to discard noise in the apparent motion or falsely matched descriptors [104, 106, 109, 108]. Common filtering techniques include *thresholding* [70, 108, 109], *random sample consensus (RANSAC) based filtering* [35, 50, 105, 108, 109], *Gaussian smoothing* [70] and *temporal averaging [103, 104, 105, 106].* In thresholding, motion vectors with magnitude values less than a threshold are removed [70, 108, 109]. RANSAC [35] can be employed to discard outliers among optical flow vectors [50, 105] or falsely matched descriptors [108, 109]. Gaussian smoothing can also be applied where there is a high variance of motion data [70]. Average pooling of temporally adjacent grid vectors is also reported to improve the recognition performance [103, 104, 105, 106]. Thresholding and average pooling are simple filtering techniques; however, thresholding might completely remove useful motion information whereas average pooling is less efficient to overcome local motion. Gaussian smoothing and RANSAC-based filtering impose the egomotion domination for proprioceptive activity recognition.

Egocentric proprioceptive activity recognition often involves multiple discriminative features since it is difficult to discriminate the activities using a single feature type [50, 70, 108, 109]. Moreover, some methods employ additional modalities, e.g. inertial sensors, to complement visual features [64, 103, 104].

Feature-level fusion can be applied to integrate feature groups into a single feature vector prior to the classification [50, 70, 81, 105]. Feature-level fusion needs to be carefully applied on multiple feature groups with equivalent scales and dimensions. Otherwise the discriminative characteristics of lower-dimensional and small-scale feature groups could be suppressed as the result of the feature-level fusion, which often employs concatenation.

2.4 Classification and post-processing

In this section, we review different types of classifiers and post-processing techniques that include the exploitation of temporal context and the decision-level fusion of multi-modal/stream information. We also present performance metrics and train-test decomposition strategies used for validation (see Table 2.3). The modelling of an activity can be performed in either a *one-vs.-one* (OVO) or *one-vs.-all* (OVA) approach in a multi-class classification problem. The OVO approach employs a model for each possible class pairs, and during testing the winning class is assumed to be dominant across the majority of the binary classifications. The OVA, also referred as one-vs.-remaining, approach uses a single model for each class that differentiates it against all the remaining classes. The OVA training strategy is often preferred due to the high number of models required in the OVO modelling, particularly when the classification involves more than three classes.

2.4.1 Classifiers

The classifiers employed for proprioceptive activity recognition can be discriminative or generative. Support vector machine (SVM) is the most commonly employed discriminative classifier in the state of the art due to its high margin decision boundary. The majority of existing works in Table 2.3 employed SVM in their pipelines. Polynomial and Gaussian kernels are often preferred to map the original feature space into a high-dimensional space [103, 104]. In addition, multi-channel Chi-square kernels are also used to integrate multiple visual features [81]. K-nearest

Table 2.3: Summary of the state-of-the-art classifiers and post-processing techniques for proprioceptive activity recognition. OVO: one-vs.-one; OVA: one-vs.-all; -: not applicable.

		[81]	[71]	[70]	[104]	[103]	[106]	[50]	[108]	[64]	[105]	[109]
Modelling	OVO			✓	✓	✓		-				
	OVA		✓					-				
Classifiers	Support vector machine	✓		✓	✓	✓	✓		✓	✓	✓	✓
	K-nearest neighbor						✓	✓	✓			
	Logitboost				✓	✓	✓				✓	
	Naive Bayes							✓	✓			
	Hidden Markov model				✓	✓	✓				✓	
	Dirichlet mixture model							✓				
	Conditional random field				✓	✓						
	Convolutional neural network		✓									
Supervision	Supervised	✓	✓	✓	✓	✓	✓			✓	✓	✓
	Unsupervised							✓				
Temporal encoding	Model-level				✓	✓	✓	✓			✓	
	Decision-level		✓									
Decision-level fusion					✓	✓				✓		

neighbour (KNN) is often used as a reference to evaluate the performance of a main classifier (e.g. SVM) [4, 50, 106, 108]. Logitboost is an advanced version of adaboost [37] that exploits weak classifiers via cascading [103, 104, 105, 106]. Naive Bayes classifiers are simple probabilistic classifiers that can be employed for proprioceptive activity classification [50, 92, 108].

Hidden Markov models (HMM) are basic sequential generative models [15, 72, 92] that are often applied to smooth the decision outputs of main classifiers [104, 105, 106]. Another generative model, the Dirichlet mixture model [16], is used for unsupervised segmentation of sport actions [50]. Conditional random field is a discriminative classifier, and it also offers graphical modelling characteristics [92] that enable structure learning across multiple temporal scales [103, 104].

Generative models are generally outperformed by discriminative ones due to the bigger data requirements of generative models to encode the relationships among feature elements. However, they might be preferred in the following cases: i) when the observation values are missing, ii) when the observation model has a useful smoothing effect on the label prediction or iii) when there is a need to predict both the observation and the label [15, 48, 92, 98].

In summary, while SVM-based classification has been successful to date, the use of deep networks is growing in proprioceptive activity recognition [58, 71, 81, 88, 107] and it is facilitated by the transferability of deeply learned features across different activity sets [71] or motion sources [107]. In addition to spatial and temporal streams [58, 107], deep frameworks can be extended to have an egocentric stream that encodes FPV characteristics [88]. Poleg et al. [71] proposed a CNN that is trained end-to-end to learn motion features from a volume of grid optical flow data using 3D convolutions. However, the complexity increases significantly with respect to image-based (2D) networks [47, 83]. Apart from [50], existing classifiers employ the ground

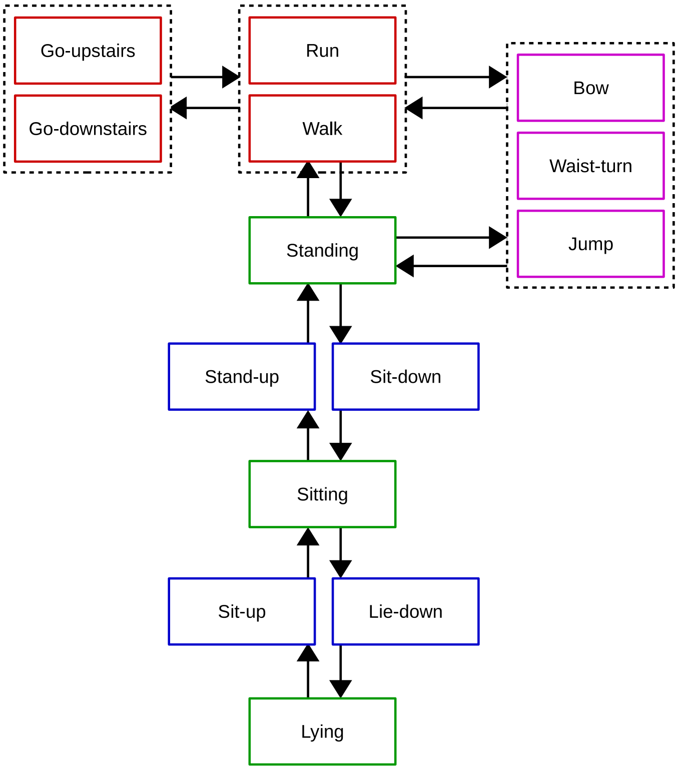


Figure 2.5: Temporal relationships among basic proprioceptive activities. The arrows indicate higher subsequent-occurrence likelihood between the corresponding activities. Activities inside the dashed boxes also possess high subsequent-occurrence likelihood. Color codes – Red and Magenta: activities involving the full- and upper-body motions, respectively; Green: the user may be stationary; Blue: transition activities.

truth data for supervision.

2.4.2 Temporal context exploitation

The proprioceptive activity classification is generally performed on a clip that lasts for two [50, 81], three [103, 104] or four [71] seconds. The duration of the clip can increase (e.g. up to 17 seconds) for long-term activities [70]. The clips are commonly 50% overlapped [71, 103, 104]. Zhang et al. [108] adopted different temporal durations for different activities, e.g. five seconds for *Sit-up* and *Sit*, six seconds for *Bow*, eight seconds for *Crouch* and twelve seconds for *Waist-turn*.

Temporal context exploitation (TCE) refers to the encoding of long-term temporal relationships (see Fig. 2.5) among subsequent clips [50, 71, 103, 104, 106]. TCE aims at utilising continuous occurrence of an activity and subsequent occurrences of activities. *Continuous-occurrence* exploitation assumes that the activity is more likely to continue if it has been performed for some time. *Subsequent-occurrence* encoding involves the exploitation of different transition likelihood of activities from one to the other, e.g. *Sit* is more likely to transit from *Stand* than from *Run*. TCE can be performed at model-level or at decision-level.

Model-level temporal context encoding (MTCE) involves iteratively modelling of the temporal information encoded between previous and current samples. Examples include: multi-scale conditional random fields, where each node represents a different time index [103, 104]; Dirichlet mixture models, designed to maximize the posterior distribution of a current sample given the labelling of the previous samples [50]; and modelling of the SVM outputs using HMM [105, 106]. Model-level encoding has a high degree of flexibility to exploit complex temporal relationships among activities.

Decision-level temporal context encoding (DTCE) refines the final classification output of the current sample towards the outputs of the previous samples without an iterative modelling. This can be implemented using a simple smoothing such as accumulative weighting [71]. Decision-level encoding might result in a rough smoothing of current information with limited capability to exploit subsequent-occurrence likelihoods.

Generally, temporal relationships need to be encoded at different temporal scales to effectively recognize proprioceptive activities. Recurrent neural networks (RNNs) are specific classes of deep neural networks designed to achieve MTCE [29, 59, 87]. Particularly, long short-term memory (LSTM) networks are specific types of RNNs that are effective to learn long-term temporal dependency using *input*, *output* and *forget* gates that act as switches to solve the vanishing and exploding gradient problems in the vanilla RNN.

2.4.3 Decision-level fusion

Decision-level fusion utilises different feature groups by post-processing their classification outputs, e.g. averaging. Decision-level fusion resembles DTCE discussed above. The difference is that DTCE works across samples at different temporal indices whereas the decision-level fusion does across samples from different feature groups at a given temporal index [64, 103, 104]. Though feature-level fusion is easier to combine multiple visual features, decision-level fusion is preferred for multi-modal features since independent local classification is often performed primarily for each modality [64, 104]. The local classification outputs are later exploited for the final decision (e.g. majority vote) [64]. In addition, decision-level fusion is preferred when the feature groups involve different dimensions and scales.

A hybrid of feature-level and decision-level fusions could be more effective in some cases,

Table 2.4: Summary of train-test decomposition strategies used for the validation of proprioceptive activity recognition.

	[81]	[71]	[70]	[104]	[103]	[106]	[108]	[64]	[105]	[109]
Decomposition per subject data				✓	✓					
Decomposition per activity data	✓					✓	✓	✓		✓
Leave-one-subject-out				✓	✓				✓	
Leave-one-data-group-out				✓	✓				✓	✓
Fixed train samples per activity		✓	✓							

e.g. when multiple features are extracted from each modality in a multi-modal sensing. As a result, a feature-level fusion is applied on the same modality features, and a decision-level fusion is applied on the separate classification outputs of different modalities. Another situation is when multiple features are extracted from different feature sources in a single-modal sensing. Examples of these feature sources include optical flow, keypoint displacement and variation of intra-frame appearance in FPV. Therefore, features from similar source are first combined at feature-level, and then classification outputs of different sources are integrated at decision-level.

2.4.4 Performance measures and decomposition strategies

In addition to computational complexity [108], precision ($\mathcal{P}$) and recall ($\mathcal{R}$) are the commonly employed performance metrics, which quantify *true positive (TP)* with respect to *false positive (FP)* and to *false negative (FN)*, respectively [50, 71, 103, 104]. F-score is the harmonic mean of precision and recall as $2*\mathcal{P}*\mathcal{R}/(\mathcal{P}+\mathcal{R})$, and it performs well in highly-biased datasets since it is independent of the class distribution. However, specificity, which evaluates *true negative (TN)* to *false positive (FP)*, is not very informative of the recognition performance in the commonly employed OVA modelling strategy. This is because the OVA results in a highly disproportional *TN* compared to *TP*. As a result, all activities tend to achieve high specificity, which does not reflect the *TP*.

A supervised proprioceptive activity recognition can employ different strategies for decomposing a dataset into train and test sets (see Table 2.4). The strategies may vary according to the *type* and *ratio* of the decomposition. The type of decomposition is often either *per-subject* [103, 104] or *per-activity* [108, 81, 106, 64, 109]. Decomposition per-subject data involves grouping the data collected by each subject into train and test sets [103, 104]. On the other hand, per-activity decomposition uses a subset of videos collected by all subjects for each activity as the train set, and the remaining videos are included in the test set [64, 81, 106, 108, 109]. Different ratios of train to test data are reported in the state of the art [71, 104].

Decomposition per-subject and per-activity in a multi-subject dataset might result in having data from a single subject both in train and test sets, which increases the resemblance of the two sets. *Leave-one-subject-out* validation avoids the resemblance by using the data from a single subject for testing, while the data from all the remaining subjects is used for training [104]. Data from more subjects can be left out rather than just one subject. In addition, the available data can be categorized into plausible groups, and *leave-one-data-group-out* validation can be applied to test the transferability of knowledge across different groups. An example can be the grouping of patients' data according to their disabilities [104].

Generally, train-test decomposition strategy should aim at making the test on future observations that are unseen during the training stage. As a result, when there are multiple subjects in a dataset, one-subject-out and subject-based leave-one-data-group-out validations reduce the resemblance between the train and test sets. On the other hand, similar videos from a single subject might appear in both sets during per-activity train-test decomposition. The majority of the train-test decomposition schemes mentioned above can also be affected by the class imbalance problem existing in many datasets, which represents unbalanced amount of data among activities and/or subjects. Using a fixed amount of training data for all activities can address this problem [70, 71].

2.5 Summary

We reviewed proprioceptive activity recognition in FPV and critically discussed its key components. In order to discriminate different proprioceptive activities, the features are designed to exploit available motion peculiarities, such as magnitude, direction and dynamics. Comparatively, optical flow-based techniques are robust as they are able to estimate motion in the presence of challenges, such as weak texture and motion blur. We propose *virtual-inertial* features extracted from first-person videos in order to complement optical flow features without using the actual inertial sensor (see Chapter 3) [4]. In addition to multiple motion feature groups, we propose to exploit hierarchical relationships among activities and temporal contexts at modelling and decision stages of the recognition pipeline (see Chapter 4).

With the growing size of publicly available datasets and the success of deep networks across different application domains, high-level learned features are expected to outperform handcrafted features. The learning can be implemented to provide: i) frame-level appearance and motion descriptors using 2D convolutions followed by temporal pooling operations or ii) spatio-temporal features using 3D convolutions. As a result, the direct learning of motion features involves high computational complexity. We propose a stacked spectrogram representation of the global motion in FPV (see Chapter 5), which enables learning of motion features using 2D convolutions and hence reduces the complexity [3]. It also provides transfer learning capability from large image datasets. Existing methods do not encode temporal information beyond a few seconds. Hence, we employ long short-term memory (LSTM) recurrent neural networks (RNNs) to exploit the long-term temporal dependency among activities [2, 3].

Existing FPV-based proprioceptive activity datasets are separately collected for different research problems, with limited cross-dataset validations. To facilitate the progression in the field, it is necessary to have an all-inclusive and challenging dataset similarly to ActivityNet [19] - a large-scale benchmark dataset for human activity understanding in third-person vision.

Wearable camera systems may contain additional built-in sensors (e.g. microphones and accelerometers). As a result, it is necessary to have a seamless integration of the multi-modal data without posing acquisition complexity and obtrusiveness. We also propose a cross-domain knowledge transfer that can enhance performance while weighting different modalities or streams of data accordingly to their merits [2]. This is particularly important when there is a scarcity of data to train a dedicated deep network from scratch.

Chapter 3

Multiple motion features from first-person videos

3.1 Introduction

In this chapter, we propose a robust motion-feature (RMF) that combines grid optical flow-based features and video-based inertial features (Fig. 3.1). The types of extracted features are motivated by the nature of variations among activities (Fig. 3.2 and 3.3). Activities such as *Sit-down* and *Stand-up* vary in their direction components (Fig. 3.2(c)) while they possess similar magnitude values (Fig. 3.2(a)). Activities such as *Sprint* and *Walk* have similar direction information (Fig. 3.2(d)) but significantly different in their magnitude patterns (Fig. 3.2(b)). In addition to this, spectrograms of motion direction in Fig. 3.3 show that discriminative features can also be extracted in the frequency domain.

We develop RMF from the optical flow and virtual inertial data by exploiting direction, magnitude and dynamics of motion in first-person videos (Fig. 3.4). We extract virtual inertial data from the movement of intensity centroid across frames in a video without physically using inertial sensors. Intensity centroid [77] is analogue to a center of mass in physics where a rigid body experiences a zero-sum of weighted relative location of its distributed mass. The centroid is computed from weighted averages of intensity values, i.e. image moments [26, 79].

The proposed RMF is generic and can be employed with any classifier. In particular, for validation we use support vector machines (SVM) and k-nearest neighbours (KNN) to test the proposed RMF and compare its performance with state-of-the-art motion features, experimented across different datasets. In order to facilitate the research, we collect two datasets that are made publicly available. The first dataset is used to experiment indoor ambulatory recognition (IAR) of eight activities and the second is related to basketball activity recognition (BAR) of eleven activities recorded in an outdoor court. To the best of our knowledge, BAR dataset is the first dataset that contains basketball activities in FPV [1]. In addition to IAR and BAR, we also validate the experiments on two publicly available datasets: JPL-interaction dataset [80] of seven activities and DogCentric [46] dataset of ten activities.

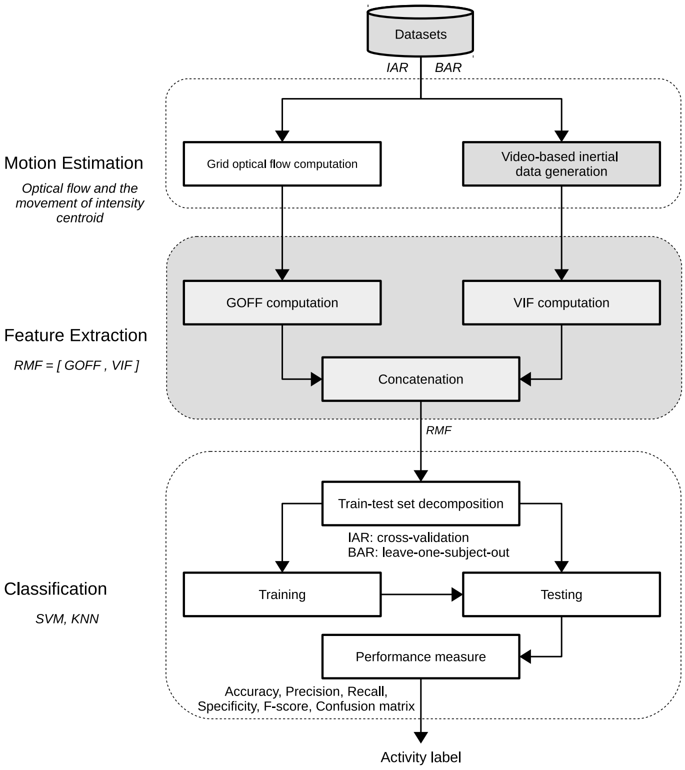
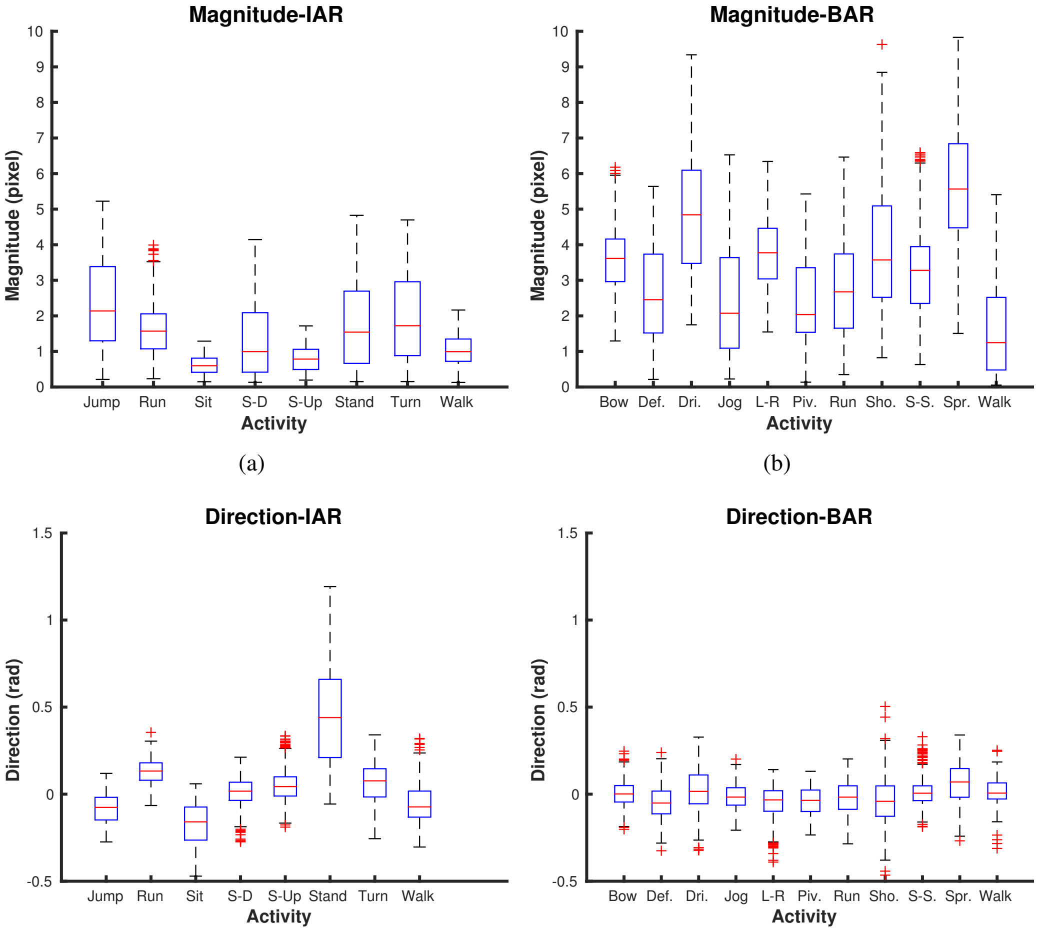

Datasets
IAR
BAR
Motion Estimation
Optical flow and the movement of intensity centroid
Grid optical flow computation
Video-based inertial data generation
Feature Extraction
RMF = [GOFF , VIF]
GOFF computation
VIF computation
Concatenation
RMF
Classification
SVM, KNN
Train-test set decomposition
IAR: cross-validation
BAR: leave-one-subject-out
Training
Testing
Performance measure
Accuracy, Precision, Recall, Specificity, F-score, Confusion matrix
Activity label

Figure 3.1: The overview of the proposed proprioceptive activity recognition system in which highlighted blocks show our contributions. IAR: indoor ambulatory recognition dataset; BAR: basketball activity recognition dataset; GOFF: grid optical flow-based features; VIF: vision-based inertial features; RMF: robust motion feature; SVM: support vector machine; KNN: k-nearest neighbourhood.

The organisation of the Chapter is as follows. Section 3.2 presents features extracted from optical flow. Section 3.3 describes features extracted from the movement of intensity centroid in a video. Complexity analysis of the proposed framework is presented in Section 3.4. Section 3.5 presents the parameter setup and datasets used for validation. Section 3.6 describes the result and discussion. Section 3.7 concludes the chapter.

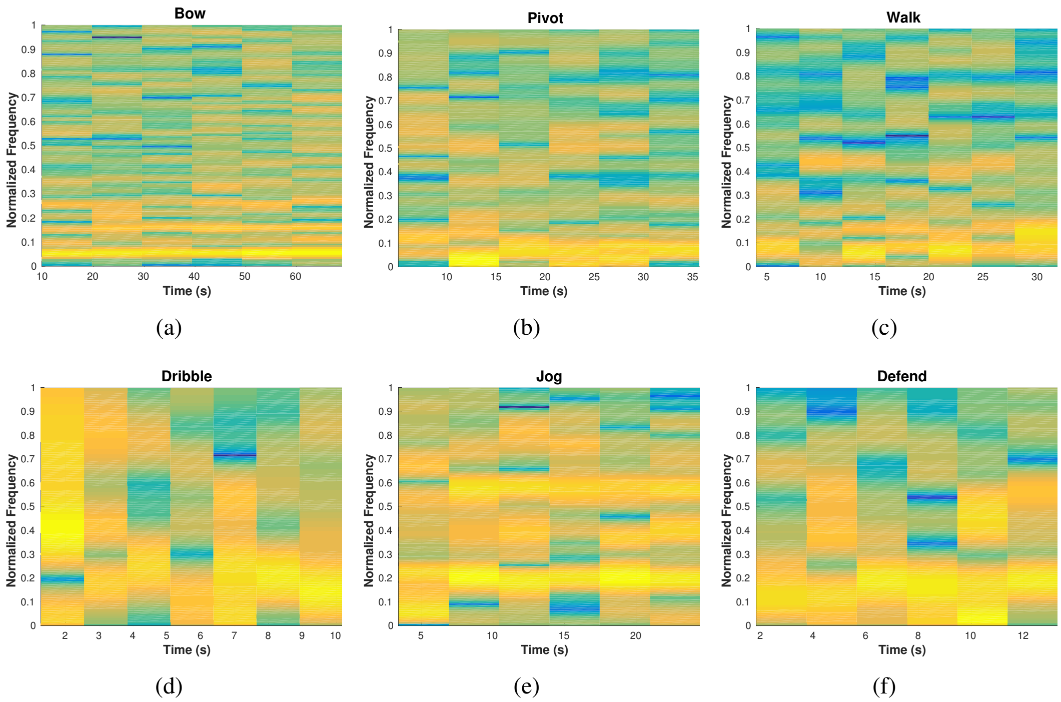
[1] The datasets and the annotations are available at `http://www.eecs.qmul.ac.uk/~andrea/FPV.html`



(a) (b)

(c) (d)

Figure 3.2: Box plots for average magnitude and direction values for activities in (a,c) IAR and (b, d) BAR datasets. For each box, the bottom and top edges reflect the 25^{th} and 75^{th} percentiles, respectively. The central line ('–') shows the median. The whiskers at the top and bottom indicate farthest inliers in both sides. Outliers are represented with '+'. Comparatively, the box plots show that average magnitude of BAR activities is higher and more variant (3.35 ± 1.16 pixels) than that of IAR (1.42 ± 0.58 pixels). From direction point of view, IAR activities show higher average variation (0.05 ± 0.19 rad) while BAR directions are restricted in -0.01 ± 0.03 rad. S-D: *Stair-down*; S-Up: *Stair-up*; Def.: *Defend*; Dri.: *Dribble*; L-R: *Left-right turn*; Piv.: *Pivot*; Sho.: *Shoot*; S-S: *Sit-stand* and Spr.: *Sprint*.



(a) (b) (c)

(d) (e) (f)

Figure 3.3: Motion-direction spectrograms that reveal the discriminating characteristics of frequency-based features. (a): low frequency activities (*Bow*, *Pivot* and *Walk*); (b): high frequency components (*Dribble*, *Jog* and *Defend*).

3.2 Grid optical flow-based features

The proposed motion-features (Fig. 3.4) exploit optical flow data more effectively than existing optical flow-based features [50, 64, 104, 106]. In order to encode the variation in motion magnitude, direction and dynamics among activities, we extract a set of feature subgroups, namely Motion magnitude histogram feature (MMHF), Motion direction histogram feature (MDHF), Motion direction histogram standard-deviation feature (MDHSF), Fourier transform of motion direction across frame (FMDF) and Fourier transform of grid motion per frame (FMPF).

Given a video sequence $\boldsymbol{V} = \{F_k\}_{k=1}^{N_v}$ where each frame $[F_k]_{R\times C}$ has a height of R pixels and a width of C pixels, we compute the Horn-Schunk optical flow [42] for each pair of successive frames. We select the Horn-Schunk method, rather than the Lucas-Kanade approach [57], because of its global smoothness assumption which is preferred in our scenario where a global motion is assumed to be dominant and reflects the ego-motion of a user wearing the camera. Because a dense optical flow representation of a frame $[E_k]_{R\times C}$ contains redundancy of motion information under the assumption of a dominant global motion, we apply a grid representation $[B_k]_{G\times G}$, where G refers to the number of grids in each dimension (Sec. 3.5.1 for the analysis part). We build the grid representation as $B_k = E_k(r_A, c_A)$, where r_A and c_A are G-dimensional row and column vec-

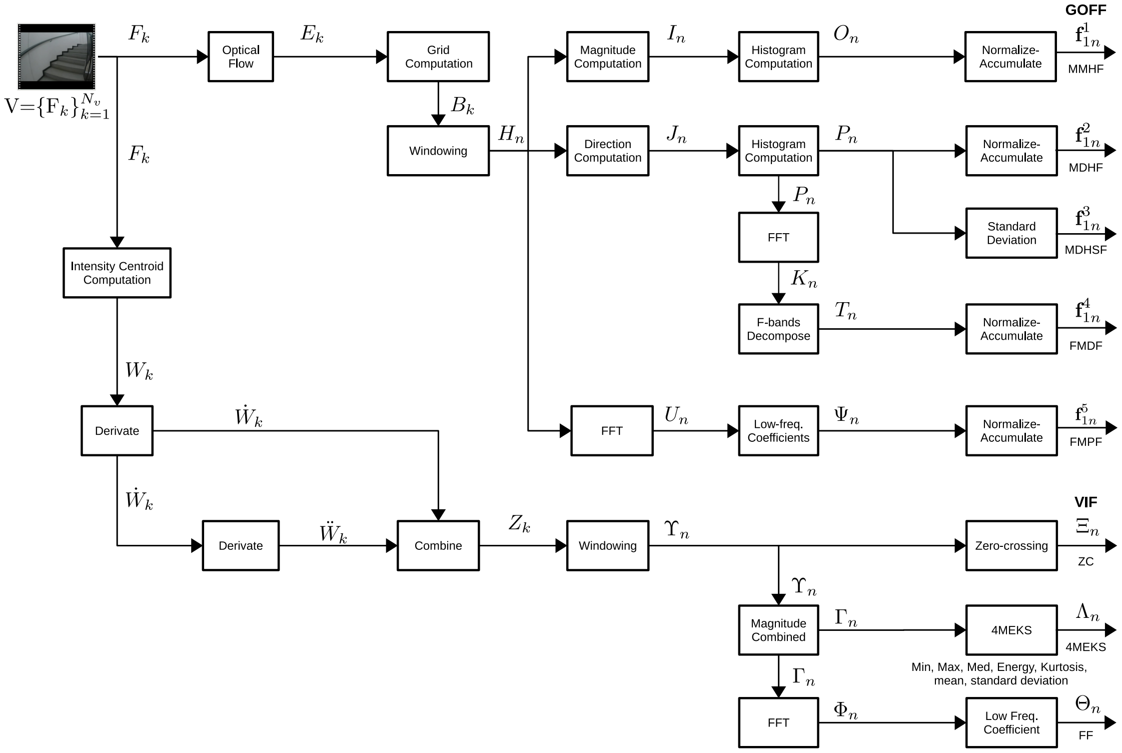


Figure 3.4: Detailed block diagram of the extraction of the proposed multi-dimension motion features. GOFF: grid optical flow-based features; VIF: vision-based inertial features; FFT: fast Fourier transform; MMHF: motion magnitude histogram feature; MDHF: motion direction histogram feature; MDHSF: motion direction histogram standard deviation feature; FMDF: Fourier transform of motion direction across frames; FMPF: Fourier transform of grid motion per frame; ZC: zero-crossing; 4MEKS: minimum, maximum, median, mean, energy, kurtosis and standard deviation; FF: Frequency-based feature.

tors sampled as $r_A = (1, 1+R/G, 1+2R/G, ..., R)$ and $c_A = (1, 1+C/G, 1+2C/G, ..., C)$. The sampling in r_A and c_A is conducted periodically after every R/G and C/G pixels, respectively, so that B_k contains sample motions from all regions in a frame. A vectorised representation of grid-optical flow of a frame includes horizontal and vertical components, i.e. $B_k = (B^g_{k_x} + jB^g_{k_y})^{G^2}_{g=1}$. We reduce the dimension of motion-vector from $R \times C$ in E_k to G^2 in B_k by applying a grid representation.

We consider the grid motion-vectors of a set of L-frames as an activity sample that is assumed to contain adequate motion data to be classified as one of the activities in $\mathcal{C}$. L represents the window length or temporal duration (in number of frames) to be found experimentally (Sec. 3.5.1). The n^{th} activity sample of a video sequence $\boldsymbol{V}$ is formulated as $V_n = \{B_{kn}\}^L_{k=1}$. The number of activity samples in $\mathbf{V}$ depends on its temporal duration, N_v, the window length, L, and overlapping percentage, ν, between a pair of consecutive windows. For example, a video segment with $N_v = 160$ frames, $L = 100$ frames and $\nu = 50\%$ has approximately three activity samples. Indices

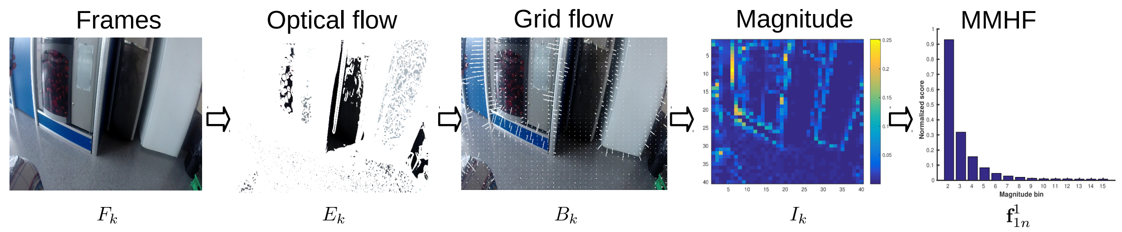


Figure 3.5: A generic example to demonstrate the step-by-step computation of MMHF using $G = 40$ grids and $\beta_m = 15$ magnitude bins for a *Stand-up* activity.

of start and end frames become $[1,100]$, $[51,150]$ and $[101,160]$, for the first, second and third samples, respectively. In order to balance the window length for the third sample, frames from the preceding sample are replicated and its modified indices become $[61,160]$. The order of frames in V_n should be the same with B_k in order to keep the temporal relation across frames, which is later used to extract frequency-based features. We describe below each element of GOFF for an activity sample V_n. The discussion on the analysis of parameter values is given in Sec. 3.5.1.

MMHF is derived from the histogram representation of grid optical flow magnitude $[I_n]_{G^2 \times L}$. A generic example of MMHF computation is shown in Fig. 3.5. The magnitude of each grid motion vector B_k^g is $\sqrt{(B_{k_x}^g)^2 + (B_{k_y}^g)^2}$, and we apply histogram computation on I_n using β_m magnitude bins to obtain the histogram representation $[O_n]_{\beta_m \times L}$. We apply non-uniform quantization since the majority in I_n are less than a single-pixel motion for most of the activities considered. We apply a Gaussian filter to smooth to I_n prior to the histogram computation. The histogram motion representation reduces the motion dimension from $G^2 \times L$ of V_n to $\beta_m \times L$ of O_n since $\beta_m < G^2$. Finally, the MMHF vector $[\boldsymbol{f}_{1n}^1]_{\beta_m \times 1}$ of an activity sample is computed from a normalization per frame, $fnorm(\cdot)$, in Eq. (3.1), followed by a temporal accumulation along each bin, $tsum(\cdot)$, in Eq. (3.2) (similarly to [108]), $\forall j \in \{1,2,...,\beta_m\}$:

$$\tilde{O}_n(j,i) = fnorm(O_n) = O_n(j,i) / \sum_{b=1}^{\beta_m} O_n(b,i), \tag{3.1}$$

$$\boldsymbol{f}_{1n}^1(b) = tsum(\tilde{O}_n) = \sum_{i=1}^{L} \tilde{O}_n(b,i). \tag{3.2}$$

The summation, $tsum(\cdot)$, accumulates the histogram representation of the motion magnitude I_n. In case some of the L frames contain noise or experience false ego-motion (e.g. due to a passer-by), their effect on the final feature vector is minimized by the accumulation with other noise-free frames in $tsum(\cdot)$. $fnorm(\cdot)$ helps to scale down the frame-level feature into $[0,1]$ prior to the accumulation. MMHF is particularly advantageous to discriminate activities which involve

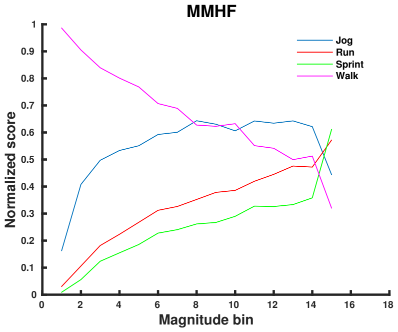


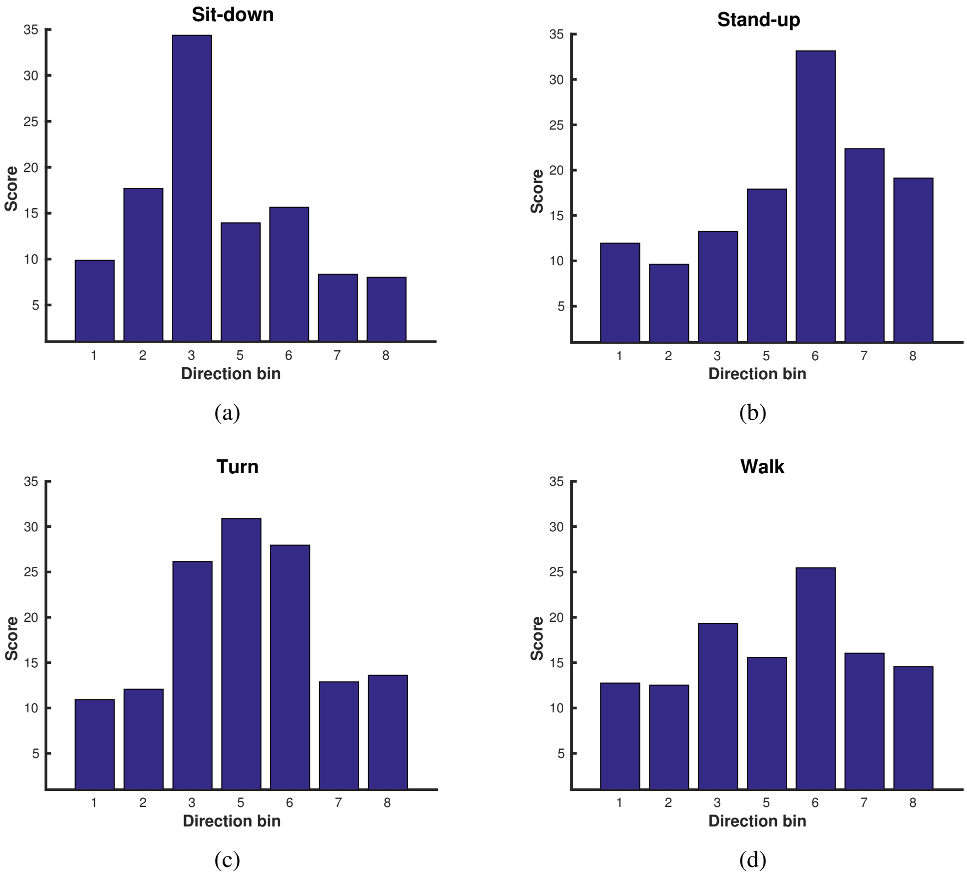
Figure 3.6: MMHF vectors built from using $\beta_m = 15$ magnitude bins in the range of motion magnitudes [0, 1] for *Walk*, *Jog*, *Run* and *Sprint* activities. The figure demonstrates that *Sprint* and *Run* contain higher magnitude values in the 15^{th} bin and few motion grids of lower magnitude values, opposite to *Walk*, whereas *Jog* endures intermediate values as expected.

similar direction patterns but different motion magnitudes. Examples include *Walk*, *Jog*, *Run* and *Sprint* for which the MMHF vectors are plotted in Fig. 3.6. MMHF values after normalization confirm the actual variation of motion magnitudes for these activities. Numerically, *Sprint*, *Run* and *Jog* video segments in BAR dataset (720 × 1280 resolution and 30 *fps*) are found to contain 87%, 81% and 62% of the frames with average magnitude greater than one pixel, respectively; while only 45% of the frames have such magnitude value in a *Walk* segment.

MDHF represents the motion direction that is determined as $\text{arctan2}(B_k^{g,y}, B_k^{g,x})$ for a grid B_k^g as a histogram. MDHF is computed similarly to the MMHF shown in Fig. 3.5, but using motion direction instead of magnitude; hence, it is vital to distinguish activities that might have similar motion magnitudes (Fig. 3.7). We develop the histogram representation $[P_n]_{\beta_d \times L}$ from motion-direction of an activity sample $[J_n]_{G^2 \times L}$ using β_d direction bins, where each bin covers a range of $(2\pi/\beta_d)$ degrees. The histogram representation reduces the motion dimension from $(G^2 \times L)$ of V_n to $(\beta_d \times L)$ of P_n since $(\beta_d < G^2)$. Then the normalization in Eq. (3.1) and the summation in Eq. (3.2) are applied on P_n in order to obtain the MDHF vector, $\boldsymbol{f}_{1n}^2$.

MDHSF represents the standard deviation of each direction bin in MDHF across P_n, formally, $[\boldsymbol{f}_{1n}^3]_{\beta_d \times 1} = \sigma([Pn]^T)$, where $\sigma(\cdot)$ represents the standard deviation function. Activities that involve high ego-motion (e.g. *Sprint* and *Run*) tend to possess higher variations, whereas slower activities (e.g. *Walk*) have minimal variations (Fig. 3.8). Different values of normalized score deviations, *Sprint* (0.11), *Run* (0.09), *Jog* (0.08) and *Walk* (0.06), reflect the level of dynamics in these activities. It is observed that *Sprint* and *Walk* relatively experience the highest and lowest dynamics, respectively.

FMDF is a frequency-domain feature that contemplates the variation of direction bins in P_n;



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

Figure 3.7: The MDHF representations of (a) *Sit-down*, (b) *Stand-up*, (c) *Turn* and (d) *Walk* using $\beta_d = 8$ direction bins. Note that the 4^{th} bin, which contains 0 degree, is not shown to achieve better visualization. It is clearly seen that MDHF vectors of *Sit-down* and *Stand-up* are mirror images to each other, reflecting the opposite motion directions they possess. *Sit-down* contains dominant motion direction of -1.35 ± 0.39 rad while *Stand-up* mainly lie in 1.35 ± 0.39 rad. On the other hand, the high score of the 5^{th} direction bin centred at 80.79 degrees in (c) shows the *Turning* direction in this particular video segment.

and differently to MDHSF, it quantifies the detailed dynamics of motion direction. We compute the fast Fourier transform (FFT) of each bin in P_n to obtain $[K_n]_{\beta_d \times L}$, which is later decomposed into N_f frequency bands, $[T_n]_{N_f \times \beta_d}$. To do so, we consider only the half width ($L/2$) of K_n due to the symmetry property of the Fourier transform. The n_f^{th} band of the b^{th} bin in $[T_n]_{N_f \times \beta_d}$ is obtained as

$$T_n(n_f, b) = \sum_{l=\gamma_i}^{\gamma_f} K_n(b, l), \tag{3.3}$$

where each row of K_n is the FFT of the corresponding row in the direction histogram P_n, $\gamma_i = 1 + \frac{(n_f - 1)L}{2N_f}$ and $\gamma_f = \frac{n_f L}{2N_f}$. The FMDF vector $\boldsymbol{f}_{1n}^4$ is derived from T_n using the normalization and

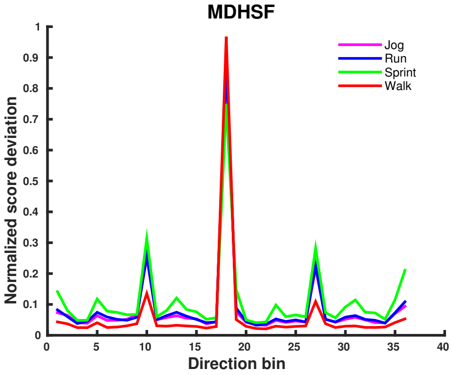


Figure 3.8: MDHSF examples for *Jog*, *Run*, *Sprint* and *Walk* activities which are characterized by similar average direction in Fig. 3.2; but here, they are shown to have different variation of direction information (MDHSF) which reflects the different level of dynamics in the activities.

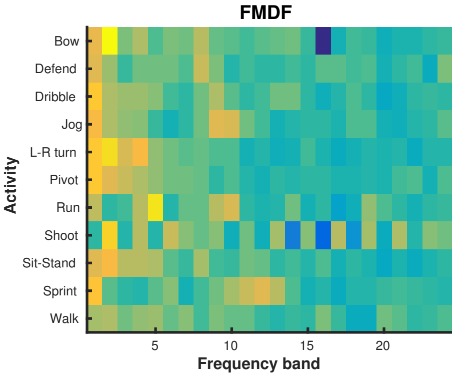


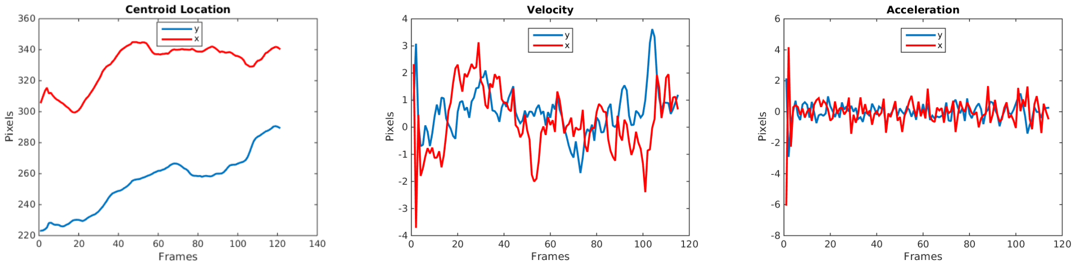
Figure 3.9: FMDF shows the distribution of the frequency response of direction histogram P_n across $N_f = 25$ frequency bands. All activities in the BAR dataset are seen to store much of their energy in the lowest frequency bands. The first band is not shown to visualize the distribution across the 24 frequency bands in-detail. *Jog*, *Run* and *Sprint* have higher frequency characteristics while *Bow*, *Left-right turn* and *Sit-stand* exhibit low frequency characteristics.

summation operations in Eq. (3.1) and Eq. (3.2), respectively. The majority of proprioceptive activities show much of their energy in the low frequency bands though significant variations can be depicted in Fig. 3.9. On the one hand, *Jog* and *Run* are found to have high values in the 10^{th} and 11^{th} frequency bands while *Sprint* possesses even higher frequency components ($12^{th} - 14^{th}$ bands). On the other hand, activities that have simple motion patterns (e.g. *Bow*, *Left-right turn* and *Sit-stand*) are shown to contain significant energy in the 3^{rd} frequency band. The range of each band in the frequency response is defined in Eq. (4.2).

FMPF is another frequency-based feature and measures the variation of grid optical flow in a frame. It is different from FMDF since the FFT is performed on each frame in V_n smoothed by a Gaussian filter. FMPF helps to discriminate complex activities with high dynamics of ego-motion (e.g. *Dribble*) from simple activities (e.g. *Walk*). The higher the ego-motion, the less

Table 3.1: Summary of subgroups in GOFF and the motion characteristics each feature type describes. Variation refers to a difference among classes that a feature subgroup exploits; β_m: magnitude bins; β_d: direction bins; N_f: number of frequency bands in FMDF; N_s: number of low frequency coefficients in FMPF.

Subgroup	Measures	Variation	Symbol	Dimension
MMHF	Motion magnitudes using histogram bins	Average magnitude	$\boldsymbol{f}^1_{1n}$	β_m
MDHF	Motion direction using histogram bins	Average direction	$\boldsymbol{f}^2_{1n}$	β_d
MDHSF	Standard deviation of direction bins	Direction deviation	$\boldsymbol{f}^3_{1n}$	β_d
FMDF	Variation of each direction bin in-detail	Periodicity (frequency)	$\boldsymbol{f}^4_{1n}$	N_f
FMPF	Variation of grid optical flow in a frame	Ego-motion complexity	$\boldsymbol{f}^5_{1n}$	N_s



(a) (b) (c)

Figure 3.10: A step-by-step visualization of virtual-inertial data extraction from an exemplar *Walk* video segment ($\approx 4s$) with 640×480 resolution at 30 *fps*. (a) The intensity centroid tracked across frames; (b), (c) the velocity and acceleration vectors derived using consecutive temporal derivatives, respectively.

likely the grid motion is to remain uniform. Since highly variant optical flow is not expected in a frame with the assumption of a uniform global motion, we select only the first N_s coefficients of the frequency response. The FMPF vector $\boldsymbol{f}^5_{1n}$ is then calculated from $[\Psi_n]_{N_s \times L}$, which is the low frequency part of $[U_n]$, using Eq. (3.1) and (3.2).

Finally, we concatenate the grid-based feature subgroups to obtain the GOFF descriptor for V_n as

$$[GOFF_n]_{N_g \times 1} = concat(\boldsymbol{f}_{1n}), \tag{3.4}$$

where $concat(\cdot)$ concatenates different feature subgroups in $\boldsymbol{f}_{1n}$ to single vector, i.e. $N_g = \beta_m + 2 * \beta_d + N_f + N_s$. The summary of GOFF is given in Table 3.1.

3.3 Vision-based inertial features

In addition to optical flow, features can be extracted from the apparent motion encoded as virtual-inertial data, which contain velocity and acceleration vectors generated from the video (Fig. 3.10). Virtual-inertial features are extracted from the virtual data similarly to the extrac-

```
Data: video ($\boldsymbol{V}$)
Result: intensity centroid ($W$), velocity ($\dot{W}$), and acceleration ($\ddot{W}$)
% initialization
$N_v \leftarrow$ number of frames in $V$, $R \leftarrow$ row size, $C \leftarrow$ column size,
$\mathcal{M}_{pq} \leftarrow$ image moment,
$p, q \in \{0, 1\} \leftarrow$ moment orders,
for $k \leftarrow 1$ to $N_v$ do
    $\mathcal{M}^k_{pq} \leftarrow \sum_{r=1}^{R} \sum_{c=1}^{C} r^p c^q f_k(r, c)$    % image moments;
    $W_k \leftarrow (\frac{\mathcal{M}^k_{O1}}{\mathcal{M}^k_{OO}}, \frac{\mathcal{M}^k_{1O}}{\mathcal{M}^k_{OO}})$    % intensity centroid
end
for $k \leftarrow 1$ to $N_v - 1$ do
    $\dot{W}_k \leftarrow W_{k+1} - W_k$    % velocity
end
for $k \leftarrow 1$ to $N_v - 2$ do
    $\ddot{W}_k \leftarrow \dot{W}_{k+1}$ - $\dot{W}_k$    % acceleration
end
```

Algorithm 1: Algorithm used to derive inertial data (velocity and acceleration) from a video in FPV.

tion of the state-of-the-art inertial features from accelerometer data [4]. Virtual-inertial features provide inertial characteristics without using the actual inertial sensors, and thus avoid synchronization issues.

The virtual inertial data generated from a video contain centroid velocity and acceleration values, both are derived from varying intensity centroid across frames in the video. In order to determine the centroid (Fig. 3.10(a)), we employ the procedure in Rublee et al. [79] that uses the first four image moments, $\mathcal{M}_{pq}$, where $p, q \in \{0, 1\}$. Each image moment of order $p + q$, $\mathcal{M}_{pq}$, is calculated as the weighted average of all intensity values in a frame (Algorithm 1). The velocity (Fig. 3.10(b)) and acceleration (Fig. 3.10(c)) values are computed by applying the first and second derivative, respectively, on the sequence of Gaussian-smoothed intensity centroids in a video. A centroid location, W, is indexed by its horizontal, x, and vertical, y, components. The velocity, $\dot{W}$, and then acceleration, $\ddot{W}$, vectors along with their magnitude make the complete set of the virtual-inertial data [4].

The velocity $[\dot{W}_k]_{2\times1}$ and acceleration $[\ddot{W}_k]_{2\times1}$ vectors for each fame are concatenated as $Z_k = [\dot{W}_k, \ddot{W}_k]^T$ before we apply L-frames long window, similarly to GOFF, to build an activity sample $[\Upsilon_n]_{4\times L} = \{Z_{kn}\}_{k=1}^{L}$. Later, velocity and acceleration magnitudes of each frame are included $[\Gamma_n]_{6\times L} = [\Upsilon_n^T, |\dot{W}_k|^T, |\ddot{W}_k|^T]^T$ in order to extract the inertial features–VIF, which contain time- and frequency-domain features adopted from the state of the art [18, 52, 68, 76, 103, 104].

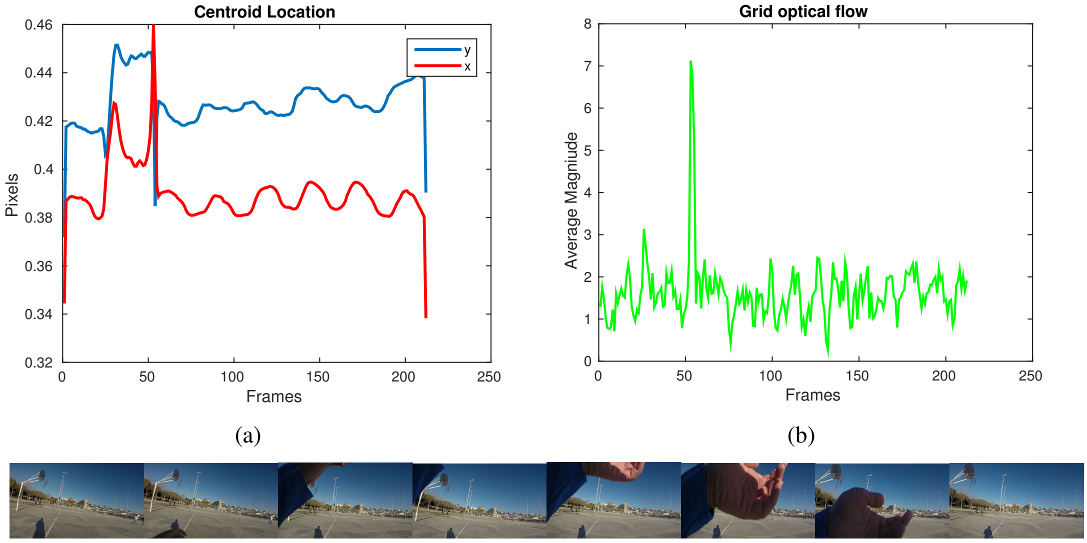
Time-domain features are *minimum*, *maximum*, *median*, *mean*, *energy*, *kurtosis*, *zero-crossing* and *standard deviation* (4MEKS) of each inertial signal. Zero-crossing measures the oscillatory behaviour of a vector in reference to zero magnitude value. Note that zero-crossing is not applied on magnitude vectors; however, the same intuition can be extracted in a reference to a non-zero threshold value. Kurtosis quantifies whether the distribution of an inertial vector is heavy-tailed or light-tailed with respect to a Gaussian distribution. A high kurtosis represents a heavy tail in the distribution, which signals a high probability of outliers [4]. Due to its high order definition, kurtosis is sensitive to noise. However, the ensemble of multiple weak features improves the discriminating potential [4, 104].

A frequency-domain feature $[\Theta_n]_{6N_l \times 1}$ is generated from the FFT response $[\Phi_n]_{6\times L}$ by selecting the first N_l low-frequency coefficients similarly to FMPF in GOFF. VIF for an activity sample (VIF_n) is then obtained from the combination of the three feature subgroups as $[VIF_n]_{N_i \times 1} = \boldsymbol{f}_{2n} = [\Xi_n, \Lambda_n, \Theta_n]$ where $N_i = 4 + 42 + 6N_l$.

Using virtual-inertial data in addition to optical flow improves performance, e.g. when there is self-occlusion or local motion due to a random appearance of a hand as shown in Fig. 3.11. In general, if the duration of a clutter is long, it is considered as a part of the background and false motion is less likely to be detected. However, if the duration is short enough, the remaining clutter-free frames in an activity sample help to reduce the error. Each pixel's intensity value is weighted by its location as shown in Algorithm 1 for intensity-centroid computation. Hence, compared to optical flow, centroid-based virtual-inertial features could be affected more severely by new objects appearing and disappearing at image boundaries. As a result, we apply a Gaussian filtering across the virtual-inertial data extraction in order to reduce noise effects.

3.4 Complexity analysis

The wall-clock computation time elapsed to accomplish each sub-task in the proposed method (Fig. 3.4) is given in Table 3.2, for an averaged video segment of approximately $\approx$ 150 frames. The grid-optical-flow and intensity centroid computations from raw video data took $B_n^t = 2.13$ s and $W_n^t = 3.38$ s, respectively. Among the GOFF subgroups, the frequency-based features FMDF and FMPF needed longer time, $\boldsymbol{f}_{1n}^{4,t} = 6.80$ ms and $\boldsymbol{f}_{1n}^{5,t} = 18.37$ ms, respectively. Overall, GOFF extraction required 2.46 s in relative to 3.39 s of VIF; and hence, RMF is able to be computed in less than six seconds. All experiments were conducted using Matlab2014b, i7-3770 CPU @ 3.40GHz, Ubuntu 14.04 OS with 16GB RAM.



(a)

(b)

Figure 3.11: The effect of a randomly appeared user's hand (last row) during *Walking* on (a) the intensity centroid and (b) the optical flow data, respectively. Note that the pixel values of x and y in the intensity centroid are normalized by C and R, respectively.

Table 3.2: Summary of wall-clock computation time to process a video segment that contains 150 frames at 60 *fps*. Elapsed time is measured for each sub-process in Fig. 3.4 and latter summarized for GOFF and VIF.

		$I_n^t = 0.32$ ms	$O_n^t = 1.66$ ms	$f_{1n}^{1t} = 2.11$ ms	
		$J_n^t = 0.89$ ms	$P_n^t = 1.37$ ms	$f_{1n}^{2t} = 2.41$ ms	
GOFF	$B_n^t = 2.13$ s			$f_{1n}^{3t} = 2.55$ ms	$GOFF_n^t = 2.16$ s
			$T_n^t = 5.69$ ms	$f_{1n}^{4t} = 6.80$ ms	
		$U_n^t = 18.19$ ms		$f_{1n}^{5t} = 18.37$ ms	
			$\Xi_n^t = 0.08$ ms		
VIF	$W_n^t = 3.38$ s	$Z_k^t = 0.21$ ms	$\Lambda_n^t = 1.08$ ms	$f_{2n}^t = 2.12$ ms	$VIF_n^t = 3.39$ s
			$\Theta_n^t = 0.96$ ms		

3.5 Experimental set-up

3.5.1 Parameter setting

GOFF and VIF extractions involve the appropriate setting of parameters, such as G, L, ν, β_d, β_m, N_f, N_s and N_l. The settings of the parameters is performed separately in a sequential optimisation approach as discussed below.

An appropriate number of grids along each dimension of a motion frame is $G = 20$ as further increments of G do not tend to include new discriminative motion characteristic as motion is assumed to be dominantly global over the majority of pixels; so more grids are more likely to cause redundancy of the motion data. We set $L = 3$ s similarly to the window length employed for human activity recognition using inertial data [68] and FPV [103, 104, 106]. Higher values of L do not cause significant improvements in the system performance whereas the motion data become redundant and the number of activity samples decreases. Similarly, the window overlapping (ν) experimented between 10% and 90% does not significantly affect the performance but reduces the number of activity samples, therefore We employed a $\nu = 50\%$ overlapping between a pair of successive samples following our experiment in the range of 10% and 90% overlapping. Very high overlapping, e.g. one-frame shift, results in a higher number of total samples for the classification, however, consecutive samples become highly identical, which does not improve the recognition performance but consumes more computational resources.

We determine the number of bins for direction and magnitude histograms by experimentally optimizing the following trade-off. Very small values of β_d and β_m might not adequately quantize the direction and magnitude information of the optical flow data, whereas very high number of bins results in over-quantization and unnecessarily long feature dimension. Experiments reveal that $\beta_d = 36$ and $\beta_m = 15$ perform better. Similarly, we set $N_f = 25$, $N_s = 25$ and $N_l = 10$. We select fewer coefficients in VIF to minimize the length of the overall feature vector since the Fourier transform is applied on each virtual inertial vector in Γ_n.

We validate the proposed motion-feature (RMF) using two geometrical classifiers [62] SVM and KNN which are, respectively, the most frequently employed parametric and non-parametric modeling techniques in the state of the art (Sec. 2). We select one-versus-all approach for the SVM due to its smaller number of classifications with respect to one-versus-one approach. We assume that a test video segment V_n belongs to only one of the activity classes $A_j \in \mathcal{C}$, and we do not consider undefined class that represents none of the activities in $\mathcal{C}$. Experimental results reveal that polynomial kernel performs better than linear and Gaussian kernels in the SVM (Table 3.3). We set the number of KNN neighbours to be one since the performance does not significantly change for higher number of neighbours.

We set $L = 180$ frames for IAR and $L = 95$ frames for BAR with $\nu = 50\%$ overlapping (for the analysis see Sec. 3.5.1). The difference in window length comes from the different frame rate used in the two datasets. We also set optimal values for the parameters in the state-of-the-art methods. Zhang et al. [108, 109] proposed a multi-resolution good feature (MRGF) with a magnitude threshold of three pixels and eight direction bins; however, higher performance is achieved with seven pixels threshold and thirty-six bins in our datasets, particularly in IAR.

We apply a random decomposition in the IAR dataset to build train and test sets as 80% train and 20% test. The final accuracy is computed from the mean of results obtained from

100 iterations. In the BAR dataset, we employ a leave-one-subject-out approach and the final accuracy is derived from the mean of results obtained after each subject is left-out iteratively. Leave-one-out approach is not applied in the IAR dataset since the recordings (R1, R2 and R3) do not consist of equivalent number of video segments per activity (Table 3.4).

Table 3.3: F-score measure (%) of different kernel types for the SVM classifier. IAR: indoor activity recognition; BAR: basketball activity recognition; JPL: JPL-interaction dataset; DogC: DogCentric dataset. Polynomial kernel achieves the highest accuracy for the majority of the methods in all datasets.

Dataset	Kernels	Baseline	AP	MRGF	MBH	RMF
IAR	Linear	53	51	72	66	83
	Gaussian	57	55	**89**	**67**	87
	Polynomial	**58**	**63**	84	65	**88**
BAR	Linear	**24**	14	25	60	75
	Gaussian	2	**22**	35	64	77
	Polynomial	19	17	**37**	**65**	**80**
JPL	Linear	3	1	42	63	83
	Gaussian	**4**	2	44	**65**	85
	Polynomial	2	**12**	**62**	63	**86**
DogC	Linear	**34**	41	30	40	55
	Gaussian	21	**45**	**40**	48	**61**
	Polynomial	25	41	39	**59**	**61**

However, for the JPL and DogCentric datasets, we adopt the corresponding approaches employed in [80] and [46], respectively. Ryoo et al. [80] used a repeated random sub-sampling validation to measure the classification accuracy. The video sets were split into train and test randomly, each contained six video sets (42 segments). Experiments were repeated iteratively 100 times and the final accuracy was computed from the mean of the iterations. For DogCentric dataset, video sequences of an activity were randomly decomposed into train and test sets, each containing half the number of total video sequences of the activity [46]. The mean final result was obtained from repeating the train-test splits 100 times and computing the mean as in the IAR and JPL datasets.

We develop a baseline method that estimates motion in a video by adopting the approach in Nagasaka et al. [63] (cited in Uehara et al. [97]) that utilises the correlation of intensity projection to approximate pixel-wise displacement between a pair of successive frames. The aim of developing the baseline method is to make a comparison against the state-of-the-art and proposed methods using a simple motion-feature extraction approach.

The overview of the baseline method is shown in Fig. 3.12, which is similar to the VIF part of the proposed method in Fig. 3.4, but the centroid location in VIF is replaced by the projections of

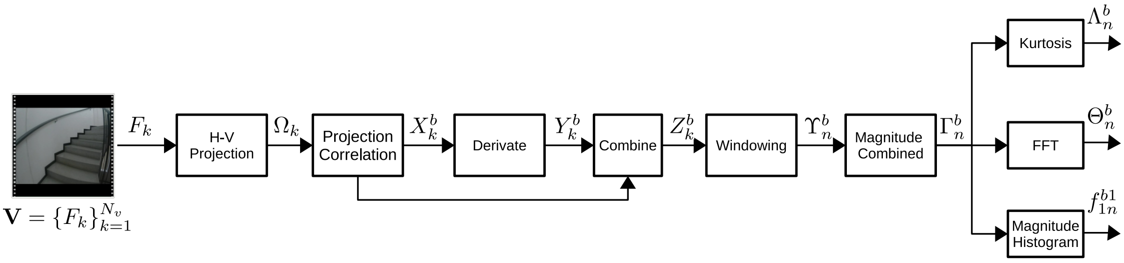


Figure 3.12: Overview of the baseline method. H-V Projection: horizontal and vertical projections $(\Omega_k^j)_{j\in\{x,y\}}$ of intensity values. Projection displacement is computed from the correlation of intensity projections in each pair of successive frames. The final baseline feature vector consists of kurtosis, Λ_n, magnitude histogram, f_{1n}^{b1}, and frequency-domain features, Θ_n.

intensity values $(\Omega_k^j)_{j\in\{x,y\}}$ in the horizontal (x) and vertical (y) directions [63]. Given a current frame $[F_k]_{R\times C}$, the horizontal $[\Omega_k^x]_{C\times 1}$ and vertical $[\Omega_k^y]_{R\times 1}$ projections were computed as

$$\begin{aligned}\Omega_k^x(c) &= (1/R)\sum_{r=1}^{R} F_k(r,c), \\ \Omega_k^y(r) &= (1/C)\sum_{c=1}^{C} F_k(r,c),\end{aligned} \tag{3.5}$$

$\forall c = 1,\cdots,C$ and $\forall r = 1,\cdots,R$. We derive the projection velocity $X_k^b = [X_k^{bx}, X_k^{by}]$ of the current frame F_k from the previous frame F_{k-1} using the following equation:

$$\begin{aligned}X_k^{bx} &= \underset{-\omega_x<\delta<\omega_x}{\operatorname{argmin}} \left(|\Omega_k^x - (\Omega_{k-1}^x < \delta >)|\right), \\ X_k^{by} &= \underset{-\omega_y<\delta<\omega_y}{\operatorname{argmin}} \left(|\Omega_k^y - (\Omega_{k-1}^y < \delta >)|\right),\end{aligned} \tag{3.6}$$

where ω_x and ω_y are, respectively, the maximum projection displacements that are assumed to exist between a pair of frames along the horizontal and vertical directions, and $(\Omega_{k-1}^j < \delta >)$ is the circular shift of the projection Ω_{k-1}^j by δ pixels. Similarly to VIF, by applying a derivation on projection velocity X_k^b across frames, we obtain the corresponding acceleration vector Y_k^b. We extract kurtosis and frequency-domain features as in VIF, and magnitude histogram as in GOFF. We apply the same parameter settings as of the proposed method for the implementation, and set $\omega_x = \omega_y = 40$ pixels assuming that a true global-motion of higher displacement is less likely.

3.5.2 Datasets

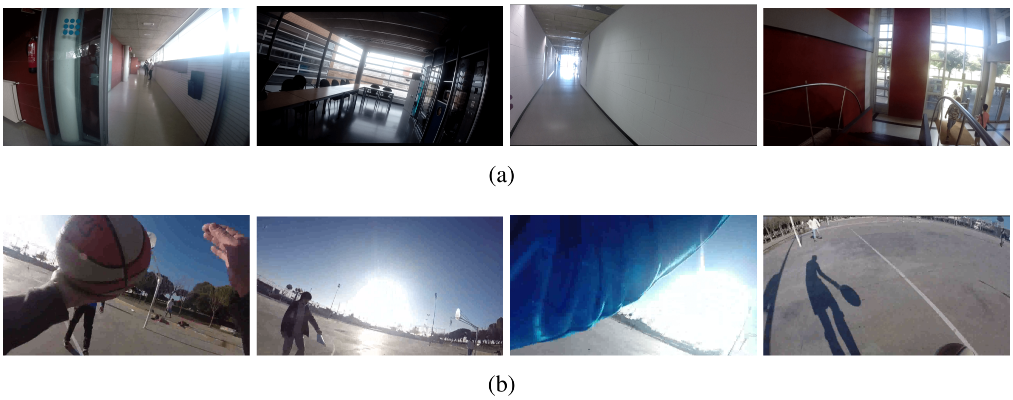
We evaluate the performance of the proposed method and compare it against three state-of-the-art methods ([50],[104] and [108]) using four datasets. The first state-of-the-art method is an interest point-based motion feature extraction approach presented in Zhang et al. [108, 109], and referred to as multi-resolution good-feature (MRGF) implemented with SURF , which outperformed Shi and Tomasi features [85]. The other two existing methods are based on optical flow, which are average pooling (AP) [103, 104, 106] and motion-based histograms (MBH) [50]. AP employs pooling procedure to smooth the grid flow and MBH applies a concatenation of histograms to encode direction, magnitude and frequency components.

We use the following measures to assess performance of the recognition system: precision, $\mathcal{P}$, sensitivity or recall, $\mathcal{R}$, specificity, $\mathcal{S}$, accuracy, $\mathcal{A}$, and F-score, $\mathcal{F}$. GOFF and VIF are studied independently and the contributions of their feature elements to the overall performance are analysed. In order to test the robustness of the methods, we introduce artificial Gaussian noise of different signal-to-noise ratio (SNR) values on the motion data. In addition to this, we test the proposed features for noisy data collected in previously unseen environment during training. We also analyse the sensitivity of our method by varying parameter settings of the feature subgroups, particularly β_d, β_m, N_f, N_s and N_l.

Our datasets

We collected two datasets with the aim of providing different environmental conditions and various activities: indoor ambulatory activity recognition (IAR) dataset and basketball activity recognition (BAR) dataset (Fig. 3.13). The IAR dataset contains the most frequently studied activities in the state of the art [64, 103, 104, 108, 109], namely *Walk*, *Run*, *Sit-down*, *Stand-up*, *Going upstairs*, *Going downstairs* and *Turn* in addition to *Jump*. Recording was conducted in three buildings with different light conditions and indoor architectures such as staircases, corridors and wall textures. We assumed a separate occurrence of each activity, meaning that, activities like *Run* while *Going upstairs* were not considered. However, we included scenarios such as *Sit* or *Jump* on staircases. Note that even if the recordings were done in indoor locations, outdoor scenes and lighting were sometimes present (Fig. 3.13(a)).

The BAR dataset is composed of three warming-up exercises (adopted from [103, 104, 108, 109]) and eight activities in a basketball game. To the best of our knowledge, BAR is the first dataset that includes basketball activities in FPV. The activities are *Bow, Sit-Stand, Left-right turn, Walk, Jog, Run, Sprint, Pivot, Shoot, Dribble* and *Defend*. Basketball activities were primarily defined by experts interviewed before the data collection, which was performed in an outdoor basketball court with four male subjects of different ages and playing experiences. Even if only a camera wearer was engaged in playing basketball during the recording, the scenes often



(a)

(b)

Figure 3.13: Key-frames from (a) IAR and (b) BAR datasets depict some of the challenges in FPV. The challenges in the IAR dataset include the effect of outdoor lighting, the lack of adequate indoor lighting and texture, and the mixing of outdoor scenes where the walls are made of glass. The challenges in the BAR dataset include self-occlusion, local motions and shadows.

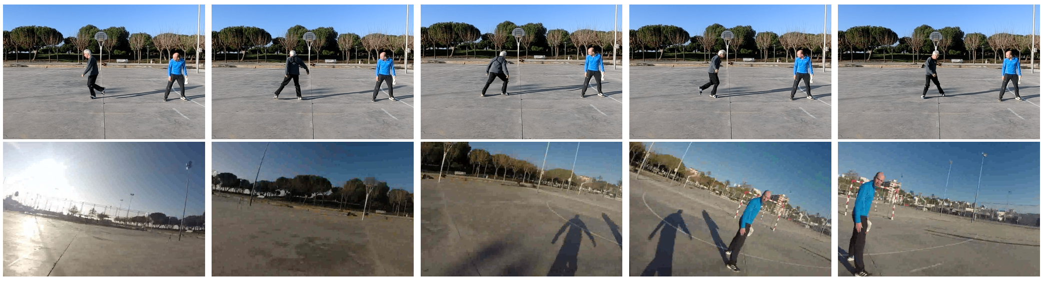


Figure 3.14: Temporal sequence of frames from a *Pivot* video. The top frames are from the ground-truth video (external camera) while the bottom frames represent the corresponding instances in the first-person video. Depending on the relative position of the wearable camera to the external camera, similar or different external scene contents might appear in the two videos.

contained the other subjects and/or shadows (Fig. 3.14).

In general, activities in the BAR dataset are more challenging as compared to the IAR dataset due to the following reasons. First, motion capture of sport activities usually involve motion parallax, blur and shutter effect along with high ego-motion [50]. Second, there is less inter-activity variation among few activities. Examples include *Left-right turn* and *Pivot*, *Bow* and *Sit-Stand*, as well as *Jog* and *Run*. Third, the BAR dataset does also contain high intra-class variations in some activities. Examples include *Shoot*, which can be a *jump-shoot* or *layout-shoot*; *Pivot*, which can be performed in clockwise or counter clockwise directions; *Defend*, which can be *slide-defend* or *backward-defend*. Other challenges result from different age and playing experience of the subjects. An example is the similarity between *Sprint* and *Run* of older and younger subjects, respectively. A chest mounted GoPro Hero3+ camera is used to record all the activities. Chest mounting is selected in order to maximize the quality of the data with respect to acquiring a full-body motion [69, 108, 109]. IAR was collected with a resolution of

Table 3.4: Summary of our IAR and BAR datasets; the top sub-table describes the IAR dataset and the number of video segments per activity in the three recordings (R1, R2 and R3). The bottom sub-table presents the contribution of the four subjects (S1, S2, S3 and S4) in the BAR dataset. Note that activities with shorter durations (e.g. *Shoot*) tend to have more video segments in order to achieve data balance. Reco.: Recording; Sub.: Subject; L-R: Left-right turn; S-S: Sit-Stand; Dur: Duration in minutes.

					IAR					Dur.
Reco.	Walk	Turn	Stand	Up-stair	Down-stair	Sit	Run	Jump	**Total**	(min)
R1	14	16	13	15	13	13	13	14	**111**	12
R2	21	21	24	18	17	23	22	20	**166**	11
R3	21	23	21	3	3	23	9	14	**117**	17
Total	**56**	**60**	**58**	**36**	**33**	**59**	**44**	**48**	**394**	**40**

						BAR							Dur.
Sub.	Bow	Defend	Dribble	Jog	L-R	Pivot	Run	Shoot	S-S	Sprint	Walk	**Total**	(min)
S1	4	3	8	4	8	14	4	30	4	2	4	**85**	15
S2	4	6	8	4	4	6	4	30	4	4	4	**78**	15
S3	4	9	8	4	4	14	4	29	4	4	4	**88**	22
S4	4	6	6	4	5	12	4	26	5	4	4	**80**	20
Total	**16**	**24**	**30**	**16**	**21**	**46**	**16**	**115**	**17**	**14**	**16**	**331**	**72**

1080 × 1920 and 60 *fps*, while 720 × 1280 with 30 *fps* was set for the BAR dataset. A summary of our datasets is shown in Table 3.4.

Public datasets

JPL-interaction [80] and DogCentric [46] are the other datasets used in this work. JPL-interaction dataset was collected in five indoor locations of varied background conditions. A toy that emulated a robot was placed on a chair, on which a GoPro camera with a resolution of 320 × 240 and 30 *fps* was mounted. The set of activities (Fig. 3.15) include four friendly, one neutral and two hostile interactions between a participant and the toy. The friendly activities are *Hug*, *Pet*, *Shake* and *Wave*. The neutral interaction is *Point*, where two persons often point towards the toy while they are having a conversation. *Punch* and *Throw* are the hostile interactions. Eight participants were involved and a total of twelve video sets were produced (two subjects did more than one experiments). Most of the sets contain seven video segments, one per activity. Key-frames from the dataset are shown in Fig. 3.15 and the summary of segment durations in the video sets is presented in Table 3.5.

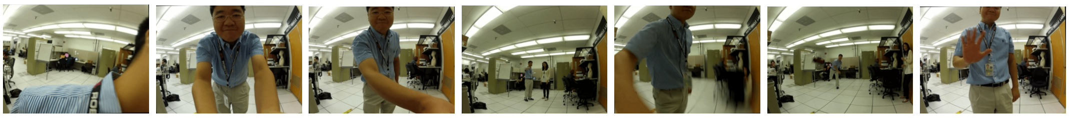


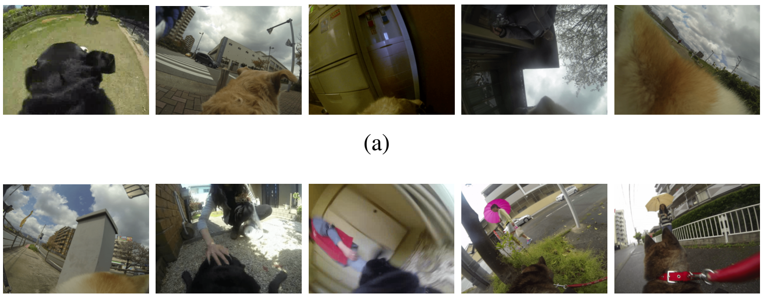
Figure 3.15: Key-frames from JPL-interaction dataset [80]. Activities from left to right are *Hug*, *Pet*, *Shake*, *Point*, *Punch*, *Throw* and *Wave*. All videos were recorded indoors with the participation of eight subjects.

Due to the lack of public motion-oriented human-centric datasets, we also experimented on

Table 3.5: Summary of JPL-interaction dataset, where duration of video segments for each activity is measured in seconds. The whole dataset is $\approx$ 10 min long in which activities *Point* and *Hug* account more than half of the overall dataset duration whereas *Wave* is the shortest activity.

	JPL dataset - video sets												**Total**
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	(Sec.)
Hug	9	11	10	9	17	10	11	6	11	15	13	10	**132**
Pet	8	8	6	11	16	6	6	5	7	10	7	7	**97**
Point	14	6	10	13	18	22	35	13	17	34	25	30	**237**
Punch	2	3	1	2	1	2	2	3	1	2	2	1	**22**
Shake	6	7	5	5	5	4	3	3	8	7	3	5	**61**
Throw	4	5	3	4	4	4	5	4	3	3	4	2	**45**
Wave	1	1	1	1	2	2	1	1	2	2	2	2	**18**
Total	**44**	**41**	**36**	**45**	**63**	**50**	**63**	**35**	**49**	**73**	**56**	**57**	**612**

recently released DogCentric dataset [46]. Though the motion patterns of dogs are completely different from human motions; IAR, BAR and DogCentric datasets share similar guideline as the motion in an egocentric video infer the type of activity being performed by a human subject in IAR and BAR datasets or by a dog in the DogCentric dataset. Four dogs were used while a GoPro camera (320×240 and 30 *fps*) was mounted on the back of each of the four dogs. The dog-centric activities considered are *Play with a ball*, *Car passing-by*, *Drink*, *Look-left*, *Look-right*, *Pet*, *Shake*, *Sniff* and *Walk*. Key-frames from the dataset are shown in Fig. 3.16 and the number and duration of video segments collected from each dog and per activity type are shown in Table 3.6.



(a)

(b)

Figure 3.16: Key-frames of the DogCentric dataset [46]. Activities from left to right are: (a) *Ball*, *Car*, *Drink*, *Feed*, *Look-left*; (b) *Look-right*, *Pet*, *Shake*, *Sniff* and *Walk*. The dataset was recorded in both indoor and outdoor environments that may contaion *people*.

3.6 Results and discussions

The results show that the proposed feature representation (RMF) performs consistently higher than the state-of-the-art methods in the four datasets considered. This highlights RMF's flexibility to work on a variety of activities and environmental conditions (Table 3.7). In IAR, MRGF

Table 3.6: Details of DogCentric dataset. The number of video segments per activity is shown for each dog participated in the experiment. The overall duration of video segments recorded from each dog and for each activity is also presented. Dur.: duration in seconds (s).

	DogCentric- video segments per activity										Total	Dur.
	Ball	Car	Drink	Feed	Look-left	Look-right	Pet	Shake	Sniff	Walk	(#)	(Sec.)
DogA	6	7	5	7	8	7	8	8	8	7	**71**	248
DogB	5	1	2	3	4	2	4	2	7	4	**34**	139
DogC	3	14	2	8	3	4	8	3	7	7	**59**	313
DogD	0	4	1	7	6	5	5	5	5	7	**45**	142
Total	**14**	**26**	**10**	**25**	**21**	**18**	**25**	**18**	**27**	**25**	**209**	842

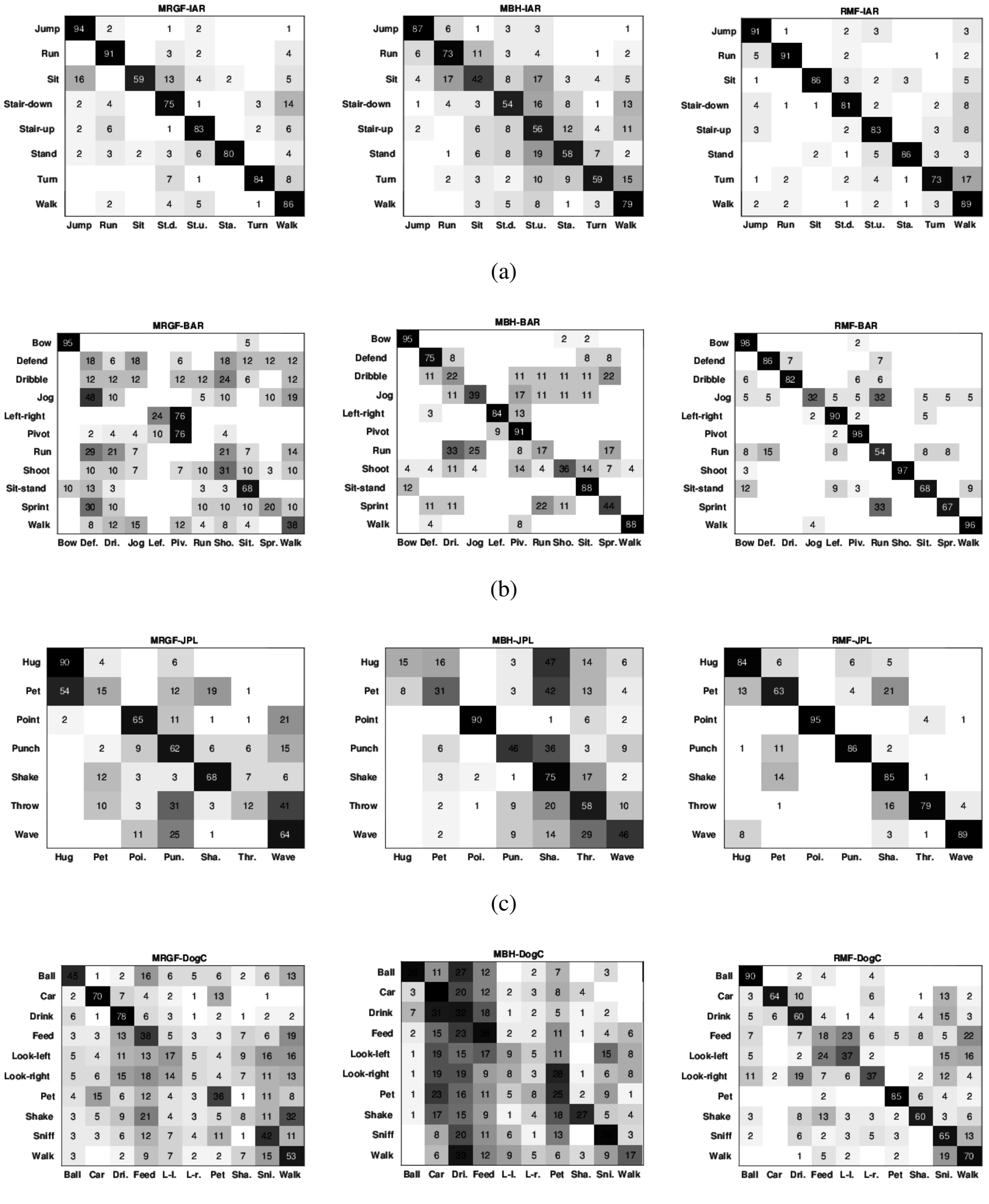
and RMF achieve equivalent performance ($\mathcal{F} = 88\%$) followed by MBH ($\mathcal{F} = 65\%$), whereas AP and Baseline were found to be the least performing motion features ($\mathcal{F} = 56\%$). MRGF performed similarly to RMF (in IAR) and MBH (in JPL) because the scene is relatively closer to the camera in the two datasets since they were recorded indoors; and hence it is less challenging for MRGF to detect interest points. However, in the BAR and DogCentric datasets, which contain complex ego-motions and activities with different motion magnitude patterns, both MRGF and MBH are found to have restricted discriminating potential due to their lack of magnitude-based features and less effective encoding of direction information, respectively.

Table 3.7: Comparative performance (%) of the proposed (RMF) and state-of-the-art methods with respect to the baseline. IAR: indoor activity recognition; BAR: basketball activity recognition; JPL: JPL-interaction dataset [80]; DogC: dogcentric dataset [46]; SVM performance measures include $\mathcal{A}$: accuracy, $\mathcal{P}$: precision, $\mathcal{R}$: recall and $\mathcal{F}$: F-score. KNN classification is validated and its F-score is also given. In IAR dataset, MRGF and RMF achieve similar performance and significantly higher than the other methods. However, all the methods, except RMF, find it difficult to achieve robust performance across the datasets. DogCentric dataset is proved to be more challenging for all the methods.

		SVM					KNN
Datasets	Methods	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{F}$	$\mathcal{F}$
	Baseline	91	69	46	98	55	53
IAR	AP	92	85	42	**99**	56	49
	MRGF	97	90	**87**	98	**88**	**79**
	MBH	91	62	68	94	65	67
	RMF	**97**	**91**	85	**99**	**88**	78
	Baseline	83	18	20	90	19	17
BAR	AP	90	24	14	97	18	31
	MRGF	89	35	39	93	37	48
	MBH	95	63	67	97	64	71
	RMF	**98**	**81**	**79**	**99**	**80**	**78**
	Baseline	84	5	1	**98**	2	13
JPL	AP	76	5	16	86	7	34
	MRGF	85	55	72	87	62	55
	MBH	87	66	53	92	59	61
	RMF	**96**	**87**	**85**	97	**86**	**82**
	Baseline	83	49	17	91	25	28
DogC	AP	87	39	30	92	34	47
	MRGF	88	39	39	94	39	42
	MBH	86	38	27	92	32	51
	RMF	**92**	**62**	**59**	**96**	**61**	**58**

In Fig. 3.17(b) and Table 3.8, we can see that MRGF has failed to discriminate *Jog*, *Run* and *Sprint* activities, which have similar direction patterns but different motion magnitude and frequency characteristics. The confusion between *Left-right turn* and *Pivot* (Fig. 3.17(b)) also happened due to the same reason. MBH is also achieving lower accuracy in BAR dataset where activities like *Dribble*, *Run* and *Sprint* are misclassified to each other due to the lack of effective encoding of direction alternation (periodicity), which MDHSF and FMDF are proposed to exploit in the RMF.

The performance of MBH to discriminate *Hug* is less than MRGF in the JPL dataset (Fig. 3.17(c)) and Table 3.8) due to their differences in the exploitation of the direction information. In the JPL dataset, *Pet* is misclassified to *Shake* by MRGF, MBH and RMF because participants involve shake-type actions while petting the toy (e.g. by holding the two hands of the toy and moving up and down). The DogCentric dataset is found to be more challenging for all the methods since the type of activities in the dataset (e.g. *Drink*, *Feed* and *Sniff*) contain more discriminant local information than the global motion. *Look-left* and *Look-right* are also hardly recognized (Table 3.8) as they are misclassified to *Walk* (Fig. 3.17(d)) because the dogs were walking most of the time while they were performing these activities. In addition, the camera was mounted on the back of the dogs (not on the head), hence, crucial activity information was not recorded.



(a)

(b)

(c)

(d)

Figure 3.17: Confusion matrices of MRGF, MBH and RMF in: (a) IAR, (b) BAR, (c) JPL and (d) DogCentric datasets. Baseline and AP are found to be ineffective compared to other methods (Table 3.7 and 3.8). Though RMF achieves the best performance in the majority of the datasets, it is possible to notice the difficulty posed by inter-class similarity between *Jog*, *Run* and *Sprint* in BAR dataset. In addition, weak recognition performances of RMF for *Feed*, *Look-left* and *Look-right* activities in the DogCentric dataset signal the need of local descriptors, beside the limitation imposed by the mounting position of the camera.

Table 3.8: Per-class recognition performance (%) of RMF and the state-of-the-art methods. $\mathcal{P}$: precision; $\mathcal{R}$: recall; $\mathcal{F}$: F-score of the SVM output. Apart from IAR dataset, RMF is shown to significantly outperform the state-of-the-art methods in the BAR, JPL and DogCentric datasets. Baseline and AP are shown to be less discriminant motion features since they do not encode magnitude and direction information effectively.

		Baseline			**AP**			**MRGF**			**MBH**			**RMF**		
Dataset	Activity	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
IAR	Jump	94	75	83	72	25	37	**96**	**95**	**95**	87	89	88	89	92	90
	Run	96	86	91	82	22	35	92	**89**	90	72	75	73	**97**	86	**91**
	Sit-down	38	25	30	**94**	41	57	91	**91**	**91**	46	52	49	91	85	88
	Stair-down	81	39	53	82	31	45	83	68	75	42	60	49	**88**	**82**	**85**
	Stair-up	49	12	19	87	61	72	89	83	86	45	58	51	**92**	**85**	**88**
	Stand-up	39	29	33	90	57	70	**96**	88	92	57	62	59	94	**90**	**92**
	Turn	88	52	65	**97**	34	50	93	**90**	**91**	72	68	70	90	77	83
	Walk	64	54	59	78	66	72	83	**87**	**85**	72	81	76	**84**	83	83
BAR	Bow	37	23	28	38	4	7	90	95	92	92	99	95	**91**	**96**	**93**
	Defend	7	49	12	0	0	0	3	7	4	59	66	62	**82**	**88**	**85**
	Dribble	10	40	16	0	0	0	11	12	11	8	16	11	**87**	**85**	**86**
	Jog	0	0	0	0	0	0	1	2	1	51	44	47	**68**	**26**	**38**
	Left-Right	6	10	8	52	12	20	67	30	41	89	91	90	**90**	**86**	**88**
	Pivot	25	6	10	23	34	27	49	83	62	68	92	78	**89**	**97**	**93**
	Run	3	6	4	0	0	0	2	2	2	23	24	23	**40**	**42**	**41**
	Shoot	97	16	27	75	3	6	37	45	41	75	51	61	**99**	**97**	**98**
	Sit-stand	8	23	12	19	10	13	56	64	60	75	**92**	**83**	**80**	75	77
	Sprint	2	25	4	0	0	0	13	31	18	56	62	59	**76**	**76**	**76**
	Walk	4	18	7	51	91	65	59	54	56	**91**	95	**93**	**91**	**96**	**93**
JPL	Hug	0	0	0	0	0	0	75	96	84	76	16	26	**77**	**90**	**83**
	Pet	0	0	0	7	1	2	57	68	62	64	32	43	**80**	**68**	**74**
	Point	0	0	0	10	46	16	83	54	65	98	89	93	**100**	**92**	**96**
	Punch	0	0	0	0	0	0	36	62	46	70	59	64	**98**	**98**	**98**
	Shake	0	0	0	2	1	1	50	84	63	41	77	54	**72**	**92**	**81**
	Throw	34	8	13	0	0	0	26	65	37	43	51	47	**93**	**70**	**80**
	Wave	0	0	0	15	62	24	54	77	63	68	50	58	**92**	**88**	**90**
DogC	Play-Ball	24	14	18	28	87	42	55	48	51	72	36	48	**79**	**91**	**85**
	Car	75	22	34	16	64	26	75	**71**	73	32	47	38	**88**	66	**75**
	Drink	39	16	23	**76**	19	30	43	**76**	55	11	32	16	58	56	**57**
	Feed	29	33	31	1	0	0	33	**40**	**36**	**34**	38	**36**	21	21	21
	Look-Left	33	28	30	0	0	0	23	18	20	19	11	14	**43**	**34**	**38**
	Look-Right	**78**	12	21	61	18	28	20	5	8	9	1	2	63	**39**	**48**
	Pet	31	14	19	**97**	56	71	44	35	39	25	27	26	90	**85**	**87**
	Shake	65	9	16	2	0	0	19	9	12	**74**	27	40	68	**58**	**63**
	Sniff	**60**	6	11	38	33	35	43	40	41	54	39	45	53	**66**	**59**
	Walk	52	13	21	**68**	21	32	39	51	44	53	17	26	60	**72**	**65**

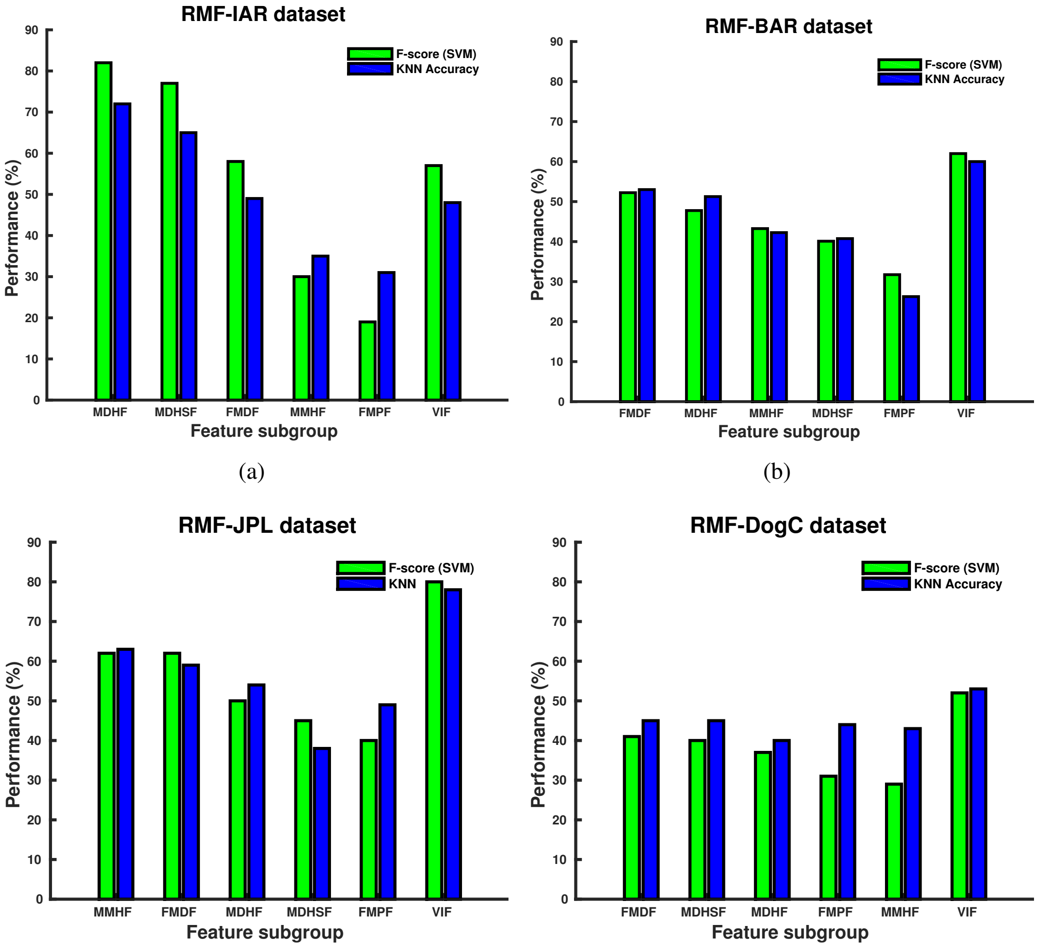
Generally, MBH and MRGF complement each other, particularly in JPL and DogCentric datasets, since MRGF is based on motion direction while MBH mainly exploits motion magnitude. Table 3.7 demonstrates that the comparative performance of the methods when validated using KNN and SVM. Due to the one-versus-all approach of the SVM classifier, the normal accuracy and specificity of all the methods are expectedly very high.

RMF is also shown to have the potential to discriminate interaction-based activities by achieving competitive results in Table 3.8 and Fig. 3.17 compared to Ryoo et al. [80] (86% vs. 90%) and DogCentric [46] (61% vs. 60%) that utilised structural matching and combination of multiple local features in addition to global motion features.

Table 3.8 shows that RMF is superior to other methods in all datasets except IAR, which is less challenging compared to other datasets (Sec. 3.5.2). The train-test splitting scheme also plays a role since the cross-validation approach in IAR introduces correlation between train and test set activities; in comparison to the leave-one-subject-out approach in BAR and equal decomposition of train and test sets in JPL and DogCentric datasets. In BAR, RMF better recognises simple activities that are characterized by dominant motion along a single dimension while the player remains in a fixed position. Examples include *Bow ($\mathcal{F} = 93\%$), Left-Right Turn ($\mathcal{F} = 88\%$), Pivot ($\mathcal{F} = 93\%$), Shoot ($\mathcal{F} = 97\%$)* and *Sit-Stand* ($\mathcal{F} = 77\%$). MBH follows RMF closely in BAR more than any of the other state-of-the-art methods except for *Dribble* where frequent changes of motion direction were not encoded effectively in the MBH.

Feature subgroups in GOFF are independently validated in all the datasets and compared against VIF as shown in Fig. 3.18. The results verify that the feature subgroups are ranked differently across the datasets, which reflects the existence of different nature of variations among activities in the datasets. For example, due to directional variation of activities in the IAR dataset (Fig. 3.2), direction-based feature subgroups (MDHF, MDHSF and FMDF) show superiority to magnitude-based feature MMHF (Fig. 3.18(a)). On the other hand, in the BAR and JPL datasets, which contain activities with different ego-motions and/or dynamics (e.g. *Sprint*, *Dribble* and *Defend*), MMHF and FMDF become significantly more important. In the DogCentric dataset, none of the feature groups is found to dominantly surpass the others. The novel intensity centroid-based virtual inertial feature (VIF) is shown to excel more than any of the GOFF subgroups in the BAR, JPL and DogCentric datasets. As expected, FMPF is the least performing subgroup of GOFF in the IAR and BAR datasets, where global motion is assumed to be dominant. Contrarily, FMPF becomes more discriminative in JPL and DogCentric datasets, where local motion contains salient information as shown Fig. 3.18(c) and (d).

Table 3.9 presents how the combination of feature subgroups improves system performance in the proposed framework. The concatenation of GOFF subgroups (S.No. 1-5), in accordance with their ranking in Fig. 3.18, and later with VIF (S.No. 6) realizes the full implementation of RMF where the highest recognition performance is achieved in each dataset. VIF is highly discriminant as it improves performance from both SVM and KNN outputs in all the datasets. In order to re-evaluate the significance of each subgroup, we remove, one-by-one, the previ-



(a) (b)

(c) (d)

Figure 3.18: Independent performance of GOFF subgroups (sorted by the F-score of SVM outputs) and VIF of the proposed RMF in the (a) IAR and (b) BAR (c) JPL and (d) DogCentric datasets. MDHF: motion direction histogram feature; MDHSF: motion direction histogram standard deviation feature; FMDF: Fourier transform of motion across frames; MMHF: motion magnitude histogram feature; FMPF: Fourier transform of grid motion per frame; VIF: vision-based inertial feature. According to the variation among activities (Fig. 3.2), direction-based features top the ranking in the IAR dataset whereas magnitude and frequency-based features become more discriminant in the BAR and JPL datasets. The different in the ranking of feature subgroups in different datasets reveals the importance of all subgroups for efficient encoding of motion information.

ously added GOFF subgroups (S.No. 7-11), where a gradual performance reduction is experienced. The different ranking of the subgroups in different datasets (Fig. 3.18), the improvement of recognition performance when we concatenate them (S.No. 1-6) and the gradual decline when we remove features (S.No. 7-11) disclose the importance of all feature subgroups in order to achieve the highest performance. Independent performance evaluation of VIF subgroups in another experiment (Table 3.10) shows that the frequency-domain feature (FF) is expectedly the best performing feature subgroup.

Table 3.9: Evaluation of the combination of features in RMF. **+**: a feature subgroup is concatenated to the above set; **-**: a feature subgroup is removed from the above set; MDHF: motion direction histogram feature; MDHSF: motion direction histogram standard deviation feature; FMDF: Fourier transform of motion direction across frames; MMHF: motion magnitude histogram feature; FMPF: Fourier transform of grid motion per frame; VIF: vision-based inertial feature; F-scores of SVM and KNN classifiers are given for comparison - $\mathcal{F}(\%)$; Generally, improved performance is achieved when we combine GOFF subgroups (S.No. 1-5) and VIF (S.No. 6), and the performance starts to decline slowly when we remove features (S.No. 7-11). This elucidates that all feature subgroups, which have different discriminative levels, are required to achieve the best performance.

	IAR			**BAR**			**JPL**			**DogC**		
S.No.	Feature	SVM	KNN	Feature	SVM	KNN	Feature	SVM	KNN	Feature	SVM	KNN
1	MDHF	82	72	FMDF	52	53	MMHF	62	63	FMDF	42	46
2	+ MDHSF	85	74	+ MDHF	66	66	+FMDF	67	65	+MDHSF	45	47
3	+ FMDF	87	75	+ MMHF	71	69	+MDHF	72	67	+MDHF	46	47
4	+ MMHF	88	78	+ MDHSF	71	72	+MDHSF	78	68	+FMPF	48	48
5	+ FMPF	88	79	+ FMPF	72	73	+FMPF	79	68	+MMHF	51	50
6	+ VIF	**88**	**79**	+ VIF	**80**	**78**	+VIF	**86**	**82**	+VIF	**61**	**59**
7	- FMPF	88	77	- FMPF	79	77	-FMPF	85	82	-MMHF	60	58
8	- MMHF	87	76	- MDHSF	79	74	-MDHSF	85	82	-FMPF	58	58
9	- FMDF	86	76	- MMHF	76	72	-MDHF	84	81	-MDHF	58	59
10	- MDHSF	84	72	- MDHF	72	66	-FMDF	81	80	-MDHSF	57	57
11	- MDHF	57	48	- FMDF	62	60	-MMHF	80	78	-FMDF	48	47

In order to measure the robustness of the methods, we artificially introduce white Gaussian noise with different signal-to-noise ratio (SNR) values in the motion data. The motion implies the grid optical flow in both RMF and AP, whereas it refers to the pixel-wise displacement of matched interest points in MRGF. We apply the noise on MBH once the motion-based histograms were computed. Figure 3.19 illustrates that RMF uniquely achieves consistent stability for a range of SNR values in the four datasets.

We experiment further the robustness of RMF by testing on a new noisy dataset (Sitges) collected in busy streets (Fig. 3.20). A subject performs all the BAR activities except for *Dribble* and the replacement of *Shoot* with *Jump*. Some of the challenges introduced in this new dataset include highly dynamic occlusions by pedestrians, which might be both in similar and opposite directions to the user motion and a lack of illumination since the recording was performed around sunset opposite to the BAR dataset, which was collected in the morning just after a sunrise. We train activity models using the BAR dataset and test them on the Sitges dataset. RMF achieved a performance of $\mathcal{F} = 56\%$ validated on SVM, higher than any of the other methods considered (Table 3.11). The SVM-based confusion matrix of the proposed method is shown in Fig. 3.21. Similarly to Fig. 3.17(b), *Run* and *Jog* are misclassified to each other. However, misclassification of erratic samples, mainly, to *Left-right turn* and *Pivot* happens because of rotation-like motions introduced due to a large field-of-view of the camera and closer appearance of buildings in this

Table 3.10: Report on the independent performance (%) of VIF subgroups. IAR: indoor activity recognition; BAR: basketball activity recognition; JPL: JPL-interaction dataset; DogC: DogCentric dataset; Min.: minimum; Max.: maximum; Med.: median; En.: energy; Kur.: kurtosis; Z-c.: zero-crossing; Std.: standard deviation; FF: frequency-domain feature; All: concatenation of all feature subgroups in VIF. F-scores of SVM and KNN classifiers are given for comparison - $\mathcal{F}(\%)$;

Dataset	Measure	Min.	Max.	Med.	En.	Kur.	Z-c.	Mean	Std.	FF	All
IAR	SVM	32	31	40	24	19	16	29	38	53	**57**
	KNN	34	31	42	30	23	27	32	40	44	**48**
BAR	SVM	26	22	36	20	20	23	34	35	**65**	62
	KNN	32	30	43	27	20	23	31	40	56	**60**
JPL	SVM	38	34	34	31	29	32	36	36	77	**80**
	KNN	40	33	39	22	41	41	39	44	66	**78**
DogC	SVM	17	17	22	16	18	23	26	21	45	**48**
	KNN	27	23	36	18	29	30	32	23	45	**47**

Table 3.11: Comparative performance (%) of the methods when they are trained on the BAR dataset and tested on the Sitges data recorded in streets with pedestrians. $\mathcal{F}$: F-scores of SVM and KNN classification outputs are given. The proposed method surpasses the state-of-the-art motion features, and MBH follows closely.

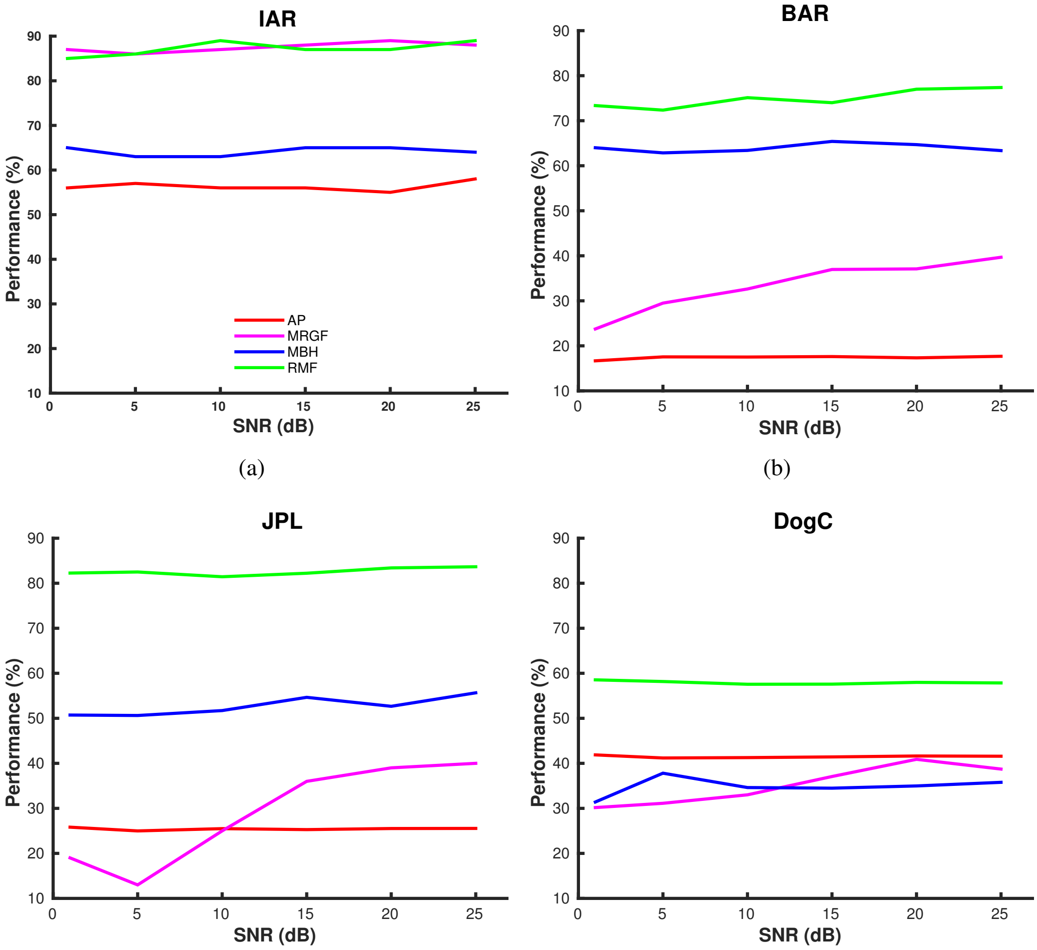
	Methods				
	Baseline	AP	MRGF	MBH	RMF
SVM	15	11	34	51	**56**
KNN	14	22	36	53	**51**

dataset. In general, the results show that RMF has a strong potential to discriminate activities even in crowded environments in FPV.

In addition, we also test the sensitivity of our proposed method for manual variation of parameter settings described in Sec. 3.5.1. In particular, we vary the parameters in GOFF and VIF: β_d, β_m, N_f, N_s and N_l. We also measured mean and standard deviation for the variation of parameter settings and recognition performances. Table 3.12 depicts the stability of RMF for the manual variations of the parameter settings in the four datasets. SVM classifier results more stable outputs with F-score variation ranging from $\sigma = 0.9$ in DogCentric to $\sigma = 2.2$ in JPL, in comparison with KNN that varies from $\sigma = 0.8$ in DogCentric to $\sigma = 3.9$ in the BAR dataset.

3.7 Summary

We designed a set of multiple motion features from optical flow and virtual inertial data generated from video in first-person vision (FPV). Discriminant features were extracted that exploited motion magnitude, direction and periodic characteristics. The proposed RMF was validated for the classification of proprioceptive activities using our two datasets, which are publicly available to the research community, and further two public interaction-based datasets. Results showed that



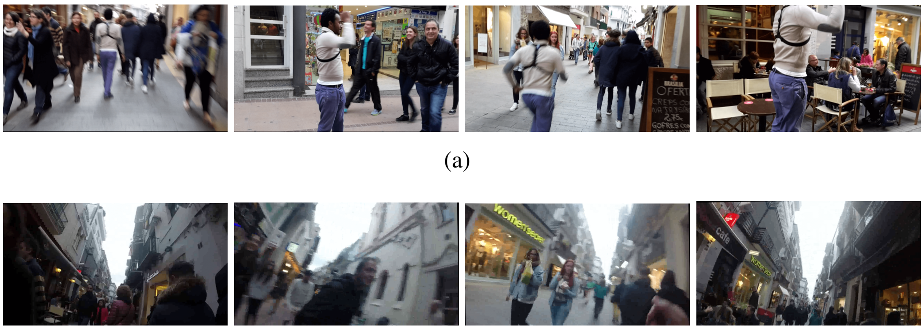
(a) (b)

(c) (d)

Figure 3.19: Robustness analysis of the methods when a Gaussian noise with SNR values ranging from 1dB to 25dB, is introduced in the motion data. Our proposed RMF demonstrates consistent robustness across the datasets.

RMF outperformed state-of-the-art features, especially in classifying activities that contain complex ego-motions. The robustness to noise and stability under different parameter settings were also demonstrated by the proposed RMF, which also outscored existing methods in more challenging environments. Motion analysis techniques, proposed for proprioceptive activity recognition in this chapter, can be applied to different applications, e.g. virtual-inertial data can be generated from the tracking of a hand or an object to recognize hand- or object-driven activities.

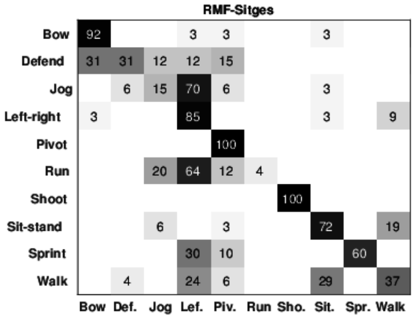
In the next chapter, we extend the low-level features, e.g. by including the pooling of frame-level appearance features and optical flow-based virtual-inertial features. We also apply efficient encoding of temporal and hierarchical relationships among activities.



(a)

(b)

Figure 3.20: Key-frames from the Sitges dataset collected to validate the flexibility of our method; (a) activities viewed from an external camera, (b) frames from first-person videos acquired by a chest-mounted wearable camera while a user performs the corresponding activity in (a). The activities from left to right are *Walk, Left-right turn, Jog* and *Jump*.



RMF-Sitges

	Bow	Def.	Jog	Lef.	Piv.	Run	Sho.	Sit.	Spr.	Walk
Bow	92			3	3			3		
Defend	31	31	12	12	15					
Jog		6	15	70	6			3		
Left-right	3			85				3		9
Pivot					100					
Run			20	64	12	4				
Shoot							100			
Sit-stand			6		3			72		19
Sprint				30	10				60	
Walk		4		24	6			29		37

Figure 3.21: Confusion matrix of the SVM output using RMF in the Sitges dataset recorded in streets with pedestrians. Erratic samples are mainly classified as either *Left-right turn* or *Pivot*, because the high field-of-view of the wearable camera and closer appearance of the buildings introduce a sense of rotational motion in FPV.

Table 3.12: Sensitivity analysis of the proposed method for variations of parameter settings in the IAR, BAR, JPL and DogCentric datasets; β_d: direction histogram bins in MDHF and MDHSF; β_m: magnitude histogram bins in MMHF; N_f: frequency bands in FMDF; N_s in FMPF and N_l in VIF: low frequency coefficients; $\mathcal{F}$: F-scores of SVM and KNN classification outputs are given (%); μ: mean; σ: standard deviation.

	Parameters					**IAR**		**BAR**		**JPL**		**DogC**	
	β_d	β_m	N_f	N_s	N_l	SVM	KNN	SVM	KNN	SVM	KNN	SVM	KNN
	36	15	25	25	10	88	78	80	78	86	82	61	58
	24	10	20	20	15	84	77	76	75	85	81	61	58
	16	5	15	15	20	83	75	75	70	86	83	62	57
	10	20	10	20	25	83	72	75	69	85	81	61	56
	18	25	30	10	5	85	76	77	78	82	79	60	58
	48	30	5	30	18	86	78	78	70	80	78	61	58
	30	5	35	5	8	85	76	77	76	84	80	59	58
μ	26.0	15.7	20.0	17.8	14.4	84.8	76.0	76.8	73.7	84.0	80.5	60.7	57.5
σ	13.1	9.7	10.8	8.5	7.1	1.7	2.1	1.7	3.9	2.2	1.7	0.9	0.8

Chapter 4

Hierarchical modelling and temporal continutiy exploitation

4.1 Introduction

In this chapter, we propose a hierarchical proprioceptive activity recognition framework (Fig. 4.1) using low-level features from optical flow, virtual inertial data and variations of intra-frame appearance descriptors and applying multi-level temporal context exploitation. Our main contributions are: (i) the exploitation of temporal continuity both during modelling and decision by applying temporal weighting on previous information; (ii) a high-level feature that encodes hierarchical and temporal relationships among activities; (iii) a confidence-based output smoothing approach that exploits the decisions of previous samples only when the current decision does not achieve a minimum confidence threshold; and (iv) low-level features from optical flow and appearance descriptors that improve the discrimination capability of motion features in Chapter 3. We also employ frequency-domain pooling operations to encode the variation of intra-frame appearance descriptors but with shorter dimension of the feature space compared with time-series gradient pooling [81].

The chapter is organized as follows. Section 4.2 presents the overview of the proposed framework that contains the hierarchical modelling of activities. Section 4.3 describes the extraction of discriminative low-level features, and Section 4.4 presents the exploitation of temporal continuity during modelling and decision. The complexity analysis and the experimental set-up are described in Sections 4.5 and 4.6. Section 4.7 presents the results in comparison with the state of the art. Finally, Section 4.8 summarises the chapter.

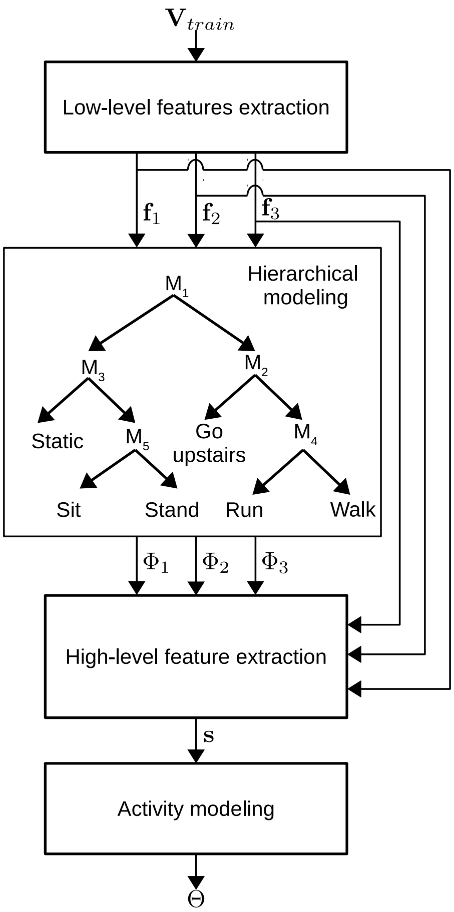
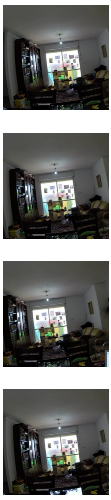
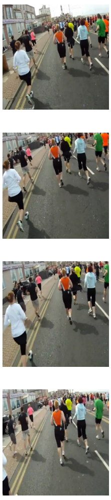
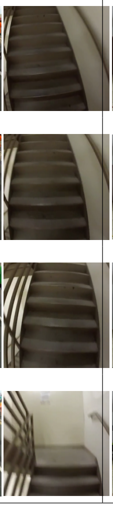
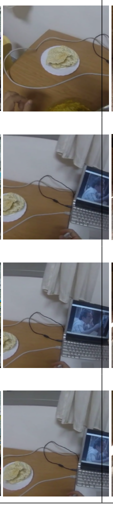
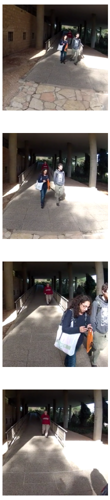
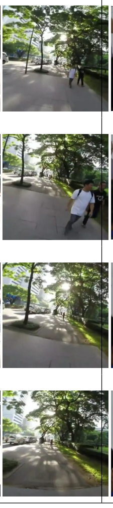
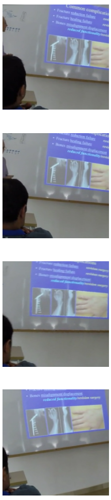


Figure 4.1: Details of the proposed multi-layer modelling framework used for learning a set of hierarchical model parameters, $\mathbf{\Phi}$, and high-level activity model parameters, Θ. Given a set of training videos, $\mathbf{V}_{train}$, the low-level feature groups, $\mathbf{f}_\iota$, $\iota \in \mathbb{Z}_{[1,3]}$, are extracted and used to find the corresponding hierarchical model parameters, Φ_ι. These parameters are then used to extract a high-level activity feature, $\mathbf{s}$, that utilises hierarchical outputs and their temporal relationships. One-vs-all activity modelling is performed on $\mathbf{s}$ to obtain Θ.

4.2 Hierarchical modelling

The proposed hand-designed hierarchical modelling is a modification of [70]. Each node in the hierarchy, M_e, $e \in \mathbb{Z}_{[1,5]}$, represents a binary classification (Fig. 4.1): M_1: *Stationary* vs. *Locomotive*; M_2: *Go upstairs* vs. *Move along flat-space*; M_3: *Static* vs. *Semi-static*; M_4: *Run* vs. *Walk*; M_5: *Sit* vs. *Stand*. Semi-static states involve moderate head and leg movements, e.g. *Sit* and *Stand*. The activities at each M_e are defined in Table 4.1. We model the hierarchy by employing a binary SVM classifier at each node, M_e. Let $\mathbf{f}_\iota$, $\iota \in \mathbb{Z}_{[1,3]}$, be a low-level feature group (Section 4.3). We use each $\mathbf{f}_\iota$ separately to find the corresponding hierarchical model parameters, $\Phi_\iota = \{\phi_{\iota e}\}_{e=1}^{5}$. Then we employ a logistic regression (LR) on a high-level feature, $\mathbf{s}$, that encodes the

Table 4.1: Definitions of activities per node, M_e, in the proposed hierarchy. Corresponding exemplar frames per activity set are shown vertically in order of increasing temporal indices.

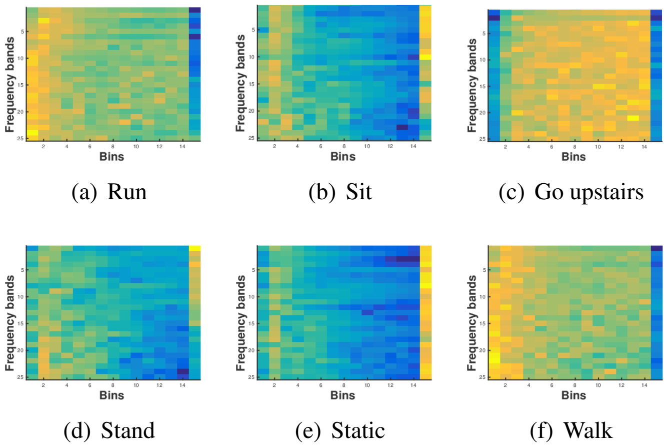
Node	M_1		M_2		M_3		M_4		M_5	
Activity	Stationary	Locomotive	Move along flat-space	Go upstairs	Static	Semi-static	Walk	Run	Sit	Stand
Definition	User stays in similar place. Includes *Sit*, *Stand* and *Static*.	User changes location. Includes *Run*, *Walk* and *Go upstairs*.	Includes *Run* and *Walk* in a flat-path (no staircases).	Usual meaning. May contain *Stationary* rest or *Run* segments.	Head fixated while legs are stationary. E.g. *Watch TV*.	*Static* but there is moderate (head and leg) motion.	Change location on foot. Slow forward motion.	User moves forward faster than Walk. E.g. *morning Jog*.	Usual sitting with natural head motion. E.g. *Sitting on a chair*.	Usual standing. May contain a few walking steps.
Exemplar frames										

hierarchical and temporal relationships among activities (Section 4.4). One-vs-all strategy is applied for the high-level activity modelling since it requires fewer classifiers compared to one-vs-one, i.e. $N_c < binom(N_c, 2)$ for $N_c > 3$, where $binom(\cdot)$ computes the binomial coefficient of the first argument to the second.

4.3 Low-level features

We extend the sets of motion features to improve the discrimination of activities and group them as *grid*, *virtual-inertial* and *pooled-appearance* features.

Grid features (GF) for the n^{th} activity sample, $\mathbf{f}_{1n}$, mainly include the features that encode magnitude, direction and dynamics (frequency) of optical flow data as presented in Chapter 3. Motivated by the effectiveness of the direction-based frequency-domain feature, FMDF, in Chapter 3, we propose a new subgroup, the Fourier transform of motion magnitude feature (FMMF), $\mathbf{f}_{1n}^5$, exploiting the variation of motion magnitude across frames (Fig. 4.2) that replaces the least effective FMPF, which encodes the frame-level variation. The remaining feature subgroups of



(a) Run (b) Sit (c) Go upstairs

(d) Stand (e) Static (f) Walk

Figure 4.2: The proposed FMMF exploits the frequency response of motion magnitude and groups it into different bands. The figures demonstrate that FMMF can easily classify *Stationary* and *Locomotive* activities. *Stationary* states do not have significant motion patterns except the high-frequency noise due to head-motion. *Going upstairs* involves high motion dynamics due to the closer scene appearance in indoor environments.

$\mathbf{f}_1$ (the subscript n is removed for clarity) are MMHF ($\mathbf{f}_1^1$), MDHF ($\mathbf{f}_1^2$), MDHSF ($\mathbf{f}_1^3$) and FMDF ($\mathbf{f}_1^4$), which are defined in Chapter 3. MDHF, MDHSF and FMDF exploit the motion direction. MDHF represents the average direction information, whereas MDHSF and FMDF evaluate the variation of direction in time and frequency domains, respectively. MMHF and FMMF describe the average and the frequency-response of motion magnitude, respectively. The importance of each feature subgroup depends on the type of variation among activities. For example, MMHF and FMMF are more useful to distinguish *Walk* and *Run*, whereas MDHF, MDHSF and FMDF are more useful to discriminate activities containing different direction patterns (Fig. 2.4).

Similarly to the remaining subgroups of $\mathbf{f}_1$, the FMMF for the n^{th} activity sample of L consecutive frames is derived from the grid optical flow data, $H_n = \{B_k\}_{k=1}^{L}$. B_k is the set that contains $G \times G$ grid vectors of a frame, $B_k = \{B_k^g\}_{g=1}^{G^2}$, where each B_k^g has horizontal, B_k^{gx}, and vertical, B_k^{gy}, components. We obtain the histogram representation for motion magnitude, O_n, as

$$O_n = hist(\{|B_k^g| : \forall B_k^g \in B_k\}; \beta_m), \tag{4.1}$$

where $hist(\cdot)$ is the operator that computes the histogram of its first argument, and β_m is the numbers of magnitude bins. We apply frame-level normalization (Eq. 3.1) and then temporal accumulation (Eq. 3.2) to O_n to obtain MMHF. The normalization and accumulation help to minimize the effect of short-term occlusion, illumination change and local motion in the segment.

FMMF represents the frequency domain analysis on O_n, which involves grouping of its Fourier response into N_m bands. Let $\bar{O}_n$ be the frequency response of O_n, the grouping of $\bar{O}_n$ into N_m bands, $\hat{O}_n$, is performed as

$$\hat{O}_n(n_m, b) = \sum_{j=\gamma_i}^{\gamma_f} \log|\bar{O}_n(b, j)|, \tag{4.2}$$

where $n_m \in \mathbb{Z}_{[1,N_m]}$ and its elements for the summation are $\gamma_i = 1 + \frac{(n_m-1)L}{2N_m}$ and $\gamma_f = \frac{n_m L}{2N_m}$ rounded to the nearest integer. We apply Eq. 3.1 and then Eq. 3.2 operations on $\hat{O}$ to obtain FMMF, $\mathbf{f}^5_{1n}$. The final set of grid-based features is $\mathbf{f}_{1n} = concat(\mathbf{f}^j_{1n})$, $\forall J = 1, \cdots, 5$.

Virtual-inertial features (VF), $\mathbf{f}_{2n}$, are extracted from the virtual-inertial data generated from video without employing the actual inertial sensor (Fig. 4.3). We extend the centroid-based virtual-inertial feature extraction in Chapter 3 by extracting additional inertial features from the grid optical flow that improves the discrimination capacity of VF. We employ the average of the grid flow per frame, $\dot{B}_k = \frac{1}{G^2}\sum_{g=1}^{G^2} B^g_k$ in addition to the instantaneous velocity of the intensity centroid, $\dot{W}_k = W_k - W_{k-1}$.

Once we obtain the two virtual velocities from grid optical flow and intensity centroid, each with horizontal and vertical components, we cascade them as $\chi_k = \{\dot{W}_k, \dot{B}_k\}$, and apply a *pre-extraction processing* that derives acceleration, $\dot{\chi}_k$, and magnitude components for both χ_k and $\dot{\chi}_k$. The acceleration component is derived from the temporal derivation of the corresponding velocity component. The complete virtual inertial data of the n^{th} activity sample, $\bar{Z}_n$, contains twelve vectors, i.e. six velocity and six acceleration vectors. Finally, $\mathbf{f}_{2n}$ is obtained from a cascade combination of the state-of-the-art time- and frequency-domain inertial features that are extracted for each vector of Z_n in time and frequency domains as described in Chapter 3. The majority of the features in $\mathbf{f}_{2n}$ are low-dimensional and susceptible to noise, however, they become significantly discriminative and robust when they are combined.

Pooled appearance features (AF), $\mathbf{f}_{3n}$, exploit intra-frame descriptors to obtain additional discriminative motion information besides the grid features and virtual inertial features. *Pooling* operations are applied to extract the temporal variation of intra-frame descriptors. We employ two intra-frame descriptors of different abstractions utilised in [81], i.e. HOG [81] and Overfeat [83]. HOG is selected due to its simplicity, and Overfeat is a deep appearance feature extracted from the last hidden layer of the corresponding CNN framework [83]. Overfeat has been applied successfully across different vision-based recognition tasks [81, 84].

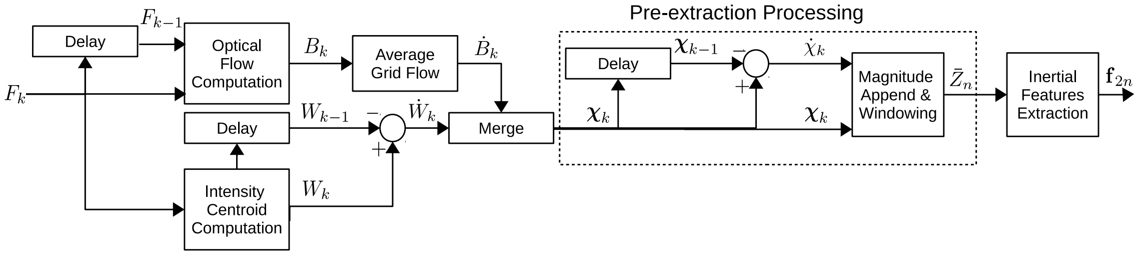


Figure 4.3: The proposed pipeline for virtual-inertial features extraction from video. Given a sequence of video frames, average grid flow and instantaneous centroid velocity are computed separately for each pair of consecutive frames (F_{k-1} and F_k) and cascaded later. Then we generate the corresponding acceleration values using a simple difference operation. We derive and append magnitude components for the velocity and acceleration vectors. Finally, we extract a set of the state-of-the-art inertial features, $\mathbf{f}_{2n}$, in time and frequency domains for a windowed sample of L frames.

Our proposed intra-frame appearance pooling consisting of one time-domain, $\upsilon_1(\cdot)$, and two frequency-domain, $\upsilon_2(\cdot)$ and $\upsilon_3(\cdot)$, pooling operations to encode short and long temporal characteristics of the intra-frame appearance descriptors. The time-series gradient (TSG) [81] only considers time-domain *summation* and *histogram* of the gradient. $\upsilon_1(\cdot)$ encodes the standard deviation of intra-frame descriptors across frames in a video, similarly to MDHS of grid-based features. $\upsilon_2(\cdot)$ groups the frequency response of each time series data into bands as FMMF and FMDF. $\upsilon_3(\cdot)$ encodes the power of each feature element in the frequency domain. Finally, $\mathbf{f}_{3n}$ is obtained from the concatenation of $\upsilon_1(\cdot)$, $\upsilon_2(\cdot)$ and $\upsilon_3(\cdot)$ outputs of the HOG and Overfeat descriptors.

4.4 Temporal continuity exploitation

We exploit the temporal relationships among activities during both modelling and decision stages to improve recognition performance in case of short-term occlusions, blurred motion or large head-motion (Fig. 4.4).

4.4.1 Model-level temporal continuity exploitation

Model-level temporal continuity exploitation (MTCE) encodes the temporal context from the hierarchical outputs of all the feature groups during activity modelling. MTCE provides the temporal component of the high-level feature using a temporally weighted accumulation of past outputs of the hierarchy.

Given the feature-based hierarchical model parameters, Φ_1, Φ_2 and Φ_3, the hierarchical de-

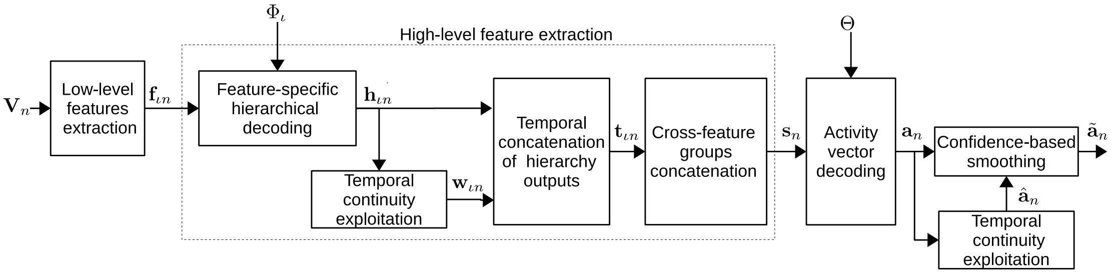

Φ_ι
High-level feature extraction
Θ
$\mathbf{V}_n$
Low-level features extraction
$\mathbf{f}_{\iota n}$
Feature-specific hierarchical decoding
$\mathbf{h}_{\iota n}$
Temporal continuity exploitation
$\mathbf{w}_{\iota n}$
Temporal concatenation of hierarchy outputs
$\mathbf{t}_{\iota n}$
Cross-feature groups concatenation
$\mathbf{s}_n$
Activity vector decoding
$\mathbf{a}_n$
Confidence-based smoothing
$\tilde{\mathbf{a}}_n$
$\hat{\mathbf{a}}_n$
Temporal continuity exploitation

Figure 4.4: Overview of the proposed FPV activity recognition for a video sample, V_n, that uses three low-level feature groups, $\{\mathbf{f}_{\iota n}\}_{\iota=1}^{3}$. A high-level feature vector that encodes hierarchical and temporal relationships among activities is then extracted from the hierarchical outputs of the low-level features, followed by consecutive temporal and cross-feature groups concatenations. The activity decision vector, $\mathbf{a}_n$, is filtered using the proposed confidence-based smoothing approach. The hierarchical model parameters, Φ_ι, and high-level model parameters, Θ, are obtained during modelling (Fig. 4.1)

.

coding of the n^{th} activity sample results in a ten unit long output per feature group, $\mathbf{h}_{\iota n} = \{\mathbf{h}_{\iota n}^{e}\}$, $\forall e \in \mathbb{Z}_{[1,5]}$, where $\mathbf{h}_{\iota n}^{e}$ contains the classification scores for both classes (c_1 and c_2) at each binary node M_e, i.e. $\mathbf{h}_{\iota n}^{e} = [\mathbf{h}_{e\iota n}^{c_1}, \mathbf{h}_{e\iota n}^{c_2}]$. The outputs of both classes, rather than the winner only, are used to exploit the level of confidence in the binary classification from their relative scores. It also reduces the likelihood of bias in the activity modelling by increasing the high-level feature dimension. Since the $\mathbf{h}_{\iota n}$ values from the SVM are not bounded, we apply a sigmoid (logistic) function, $S(\cdot)$, that maps any real value λ to a value inside the bounded region $(0,1)$ as

$$S(\lambda) = \frac{1}{1+exp(-\lambda)}. \tag{4.3}$$

MTCE provides $\mathbf{w}_{\iota n}$ that represents the accumulation of hierarchical outputs of D previous samples, weighted according to their temporal distance to the current index n as

$$\mathbf{w}_{\iota n} = \sum_{d=1}^{D} W(d)\mathbf{h}_{\iota n-d}, \tag{4.4}$$

where $W(\cdot)$ is the weighting function applied to give more importance to recent samples and less importance to earlier samples as

$$W(d) = \frac{exp(-d/D)}{\sum_{d=1}^{D} exp(-d/D)}. \tag{4.5}$$

The current, $\mathbf{h}_{\iota n}$, and weighted, $\mathbf{w}_{\iota n}$, hierarchical outputs are concatenated to extract feature-specific temporal vectors, $\mathbf{t}_{\iota n} = [\mathbf{h}_{\iota n}, \mathbf{w}_{\iota n}]$. The high-level feature vector for the activity modelling is obtained from the cross-feature groups concatenation as $\mathbf{s}_n = concat(\mathbf{t}_{1n}, \mathbf{t}_{2n}, \mathbf{t}_{3n})$. The high discrimination characteristic of $\mathbf{s}_n$ is derived from the temporal and hierarchical information extracted from the three low-level feature groups.

4.4.2 Decision-level temporal continuity exploitation

In addition to the activity modelling, we exploit previous temporal information during the activity vector decoding (Fig. 4.4). Decision-level temporal continuity exploitation (DTCE) is applied to smooth the decision when the confidence of the classification fails to achieve a minimum threshold, which is performed by exploiting the temporal continuity, similarly to MTCE, using the decisions of the previous samples. We define *confidence* as the relative weight of the winning class probabilistic score (maximum of the activity vector) with the second maximum score. Rather than using a 'blind' accumulation with previous samples' outputs as in [71], we propose a confidence-based smoothing strategy (Algorithm 2).

We argue that smoothing may not improve the recognition performance (if it does not degrade) when the confidence level is high. On the other hand, if the confidence of a decision does not satisfy the threshold value, additional decision knowledge from previous samples is more likely to improve performance. DTCE gives more weight to the recent decisions; whereas [71] applied equal weights to all previous decisions, which undermines the significance of the current output and its closely related temporal samples.

Let the decoding of the n^{th} sample, with a feature vector, $\mathbf{s}_n$, using a set of model parameters, Θ, be $[\mathbf{a}_n]_{N_c \times 1}$. We measure the confidence level, r_n, from the ratio of the maximum probabilistic value (winning class score), $\mathbf{a}_n^1$, to the second maximum value, $\mathbf{a}_n^2$. We compare r_n to an experimentally found threshold value, r_t. If the threshold is satisfied, the final prediction vector $\tilde{\mathbf{a}}_n$ becomes $\mathbf{a}_n$. Otherwise we update $\mathbf{a}_n$ to $\mathbf{a}_n^{'}$ by including temporal information obtained using a weighted accumulation of the previous D activity prediction vectors, $\hat{\mathbf{a}}_n$, similarly to Eq. 4.4 and Eq. 4.5, and the confidence is then re-evaluated, $r_n^{'}$. If $r_n^{'}$ satisfies the threshold, $\tilde{\mathbf{a}}_n$ becomes $\mathbf{a}_n^{'}$. Otherwise we check the confidence, $\hat{r}_n$, of $\hat{\mathbf{a}}_n$ and if $\hat{r}_n > r_t$, then $\tilde{\mathbf{a}}_n = \hat{\mathbf{a}}_n$. If none of r_n, $r_n^{'}$ and $\hat{r}_n$ satisfies the threshold, $\tilde{\mathbf{a}}_n$ becomes one of $\mathbf{a}_n, \mathbf{a}_n^{'}$ and $\hat{\mathbf{a}}_n$ that corresponds to the maximum confidence score.

Both MTCE and DTCE exploit the previous knowledge in order to improve the recognition of the current sample. However, they might undermine the recognition of a short activity segment (e.g. *Stand*) that appears abruptly in the middle of an other activity (e.g. *Walk*). Comparatively,

```
Require: Decision vectors, $\{\mathbf{a}_n, \mathbf{a}_{n-1}, \cdots \mathbf{a}_{n-d}, \cdots \mathbf{a}_{n-D}\}$,
         Weighting function, $W(\cdot)$
         Vector selection with the highest confidence, $I(\cdot)$
         Confidence threshold, $r_t$
Ensure: Final prediction vector, $\tilde{\mathbf{a}}_n$
  $\mathbf{a}_n^1 \leftarrow \max(\mathbf{a}_n)$,
  $\mathbf{a}_n^2 \leftarrow \max(\mathbf{a}_n \setminus \mathbf{a}_n^1)$
  $r_n \leftarrow \frac{\mathbf{a}_n^1}{\mathbf{a}_n^2}$
  if $r_n > r_t$ then
    $\tilde{\mathbf{a}}_n \leftarrow \mathbf{a}_n$
  else
    $\hat{\mathbf{a}}_n \leftarrow \sum_{d=1}^{D} W(d)\mathbf{a}_{n-d}$
    $\mathbf{a}_n^{'} \leftarrow \mathbf{a}_n \cdot \hat{\mathbf{a}}_n$
    $\mathbf{a}_n^{'1} \leftarrow \max(\mathbf{a}_n^{'})$
    $\mathbf{a}_n^{'2} \leftarrow \max(\mathbf{a}_n^{'} \setminus \mathbf{a}_n^{'1})$
    $r_n^{'} \leftarrow \frac{\mathbf{a}_n^{'1}}{\mathbf{a}_n^{'2}}$
    if $r_n^{'} > r_t$ then
      $\tilde{\mathbf{a}}_n \leftarrow \mathbf{a}_n^{'}$
    else
      $\hat{\mathbf{a}}_n^1 \leftarrow \max(\hat{\mathbf{a}}_n)$
      $\hat{\mathbf{a}}_n^2 \leftarrow \max(\hat{\mathbf{a}}_n \setminus \hat{\mathbf{a}}_n^1)$
      $\hat{r}_n \leftarrow \frac{\hat{\mathbf{a}}_n^1}{\hat{\mathbf{a}}_n^2}$
      if $\hat{r}_n > r_t$ then
        $\tilde{\mathbf{a}}_n \leftarrow \hat{\mathbf{a}}_n$
      else
        $\tilde{\mathbf{a}}_n \leftarrow I(\mathbf{a}_n, \mathbf{a}_n^{'}, \hat{\mathbf{a}}_n)$
      end if
    end if
  end if
```

Algorithm 2: Algorithm for confidence-based smoothing

MTCE provides a framework to learn from temporal relationships since the previous knowledge is incorporated in the modelling stage, whereas DTCE adopts a slightly rough smoothing, limited to exploit additional discriminative characteristics from the current and previous decisions.

4.5 Complexity Analysis

Let R be the height and C be the width of each frame in pixels. The complexity of the optical flow computation is $\mathcal{O}(n_w^2 RC)$ per frame pair, with n_w being the number of warp parameters [11]. The computation of the intensity centroid requires $\mathcal{O}(RC)$ and the computation of the average grid flow with $G \times G$ grids requires $\mathcal{O}(G^2)$ per frame. The cost of generating Overfeat descriptor is approximately $\mathcal{O}((RC(\varphi+1))^{RC+\kappa})$ per frame, where φ is the number of layers and κ is the number of hidden neurons per layer [14]. As for *grid-based features*, most of the intermediate steps in extracting feature subgroups exhibit linearly growing complexity. For example, MDHF and MMHF cost $\mathcal{O}(G^2+\beta_d)$ and $\mathcal{O}(G^2+\beta_m)$, respectively, for β_d direction bins and β_m magnitude bins, after the corresponding grid direction and magnitude are computed.

FMDF and FMMF cost $\mathcal{O}(\beta_d L \log L)$ and $\mathcal{O}(\beta_m L \log L)$, respectively, for a video segment of L frames. Furthermore, each Fourier transform cost is increased by $\mathcal{O}(L^3 N_b)$ due to the magnitude computation of the frequency response, logarithmic scale change and the grouping into $N_b \in \{N_d, N_m\}$ frequency bands. N_d and N_m represent the number of direction and magnitude bins for FMDF and FMMF, respectively. Similarly to $\mathbf{f}_1$, it is only the frequency feature that has a significant complexity among subgroups in $\mathbf{f}_2$, which is equivalent to $\mathcal{O}(L^3 \log L)$ for each virtual inertial vector. The proposed pooling operations, $\upsilon_1(\cdot)$, $\upsilon_2(\cdot)$ and $\upsilon_3(\cdot)$, applied on β_q dimensional intra-frame descriptor cost $\mathcal{O}(\beta_q)$, $\mathcal{O}(\beta_q L^4 N_q \log L)$ and $\mathcal{O}(\beta_q L^2 \log L)$, respectively. An SVM training costs $\mathcal{O}(\max(N_t,N_\iota),\min(N_t,N_\iota)^2)$ on a data of N_t train samples, where each sample is represented with N_ι-dimensional $\mathbf{f}_\iota$ [24]. The logistic regression cost increases linearly with the data size as $\mathcal{O}(N_t)$. The temporal continuity constraints introduce a complexity of $\mathcal{O}(D)$ per feature group, $\mathbf{f}_\iota$.

Table 4.2 shows the summary of the wall-clock computation time elapsed for the extraction of the proposed features for a randomly selected ≈ 3 s long segment at 30 *fps*. The computation bottleneck lays on the initial motion estimation (grid optical flow and intensity centroid) or appearance description (HOG and Overfeat) than the proposed features extraction. Particularly, it takes about 140 s to derive Overfeat [83], which is partly because we use the pre-compiled binaries. The experiments were conducted using Matlab2014b, i7-4770 CPU @ 3.40GHZ, Ubuntu 14.04 OS and 16GB RAM.

4.6 Experimental set-up

4.6.1 Parameter setting

To extract GF and VF, we adopt the parameter values in Chapter 3. We employed the settings in [81] for HOG and Overfeat extraction, but we change the grid dimension for HOG from 5×5 to 7×7 as the frame resolution changes from 320×240 to 640×480, respectively. We use the same number of bands for $\upsilon_2(\cdot)$ similarly to the FMDF and FMMF. The number of previous samples used for extracting the temporal knowledge is found experimentally by iteratively testing different temporal duration (previous samples) on each feature group and their combination for

Table 4.2: Summary of wall-clock time elapsed for the computation of proposed features experimented on a randomly selected $\approx 3s$ long video segment at 30 *fps*. MDHF: motion direction histogram; MDHS: motion direction histogram standard deviation; FMDF: Fourier transform of motion direction feature; MMHF: motion magnitude histogram feature; FMMF: Fourier transform of motion magnitude feature; HOG: histogram of oriented gradient; υ_1: standard deviation pooling; υ_2 and υ_3 are frequency domain pooling operations. υ_2 decomposes the frequency response into bands whereas υ_3 computes the power in frequency domain.

Feature source	Feature subgroups	Feature groups

Grid optical flow = 3.83 s	MDHF = 3.92 ms
	MDHS = 3.97 ms
	FMDF = 4.34 ms
	MMHF = 2.80 ms
	FMMF = 1.95 ms
		GF = 3.84 s

Intensity centroid = 6.69 s
Average grid flow =3.84 s
	time-domain = 1.58 ms
	frequency-domain = 3.28 ms
		VF = 10.54 s

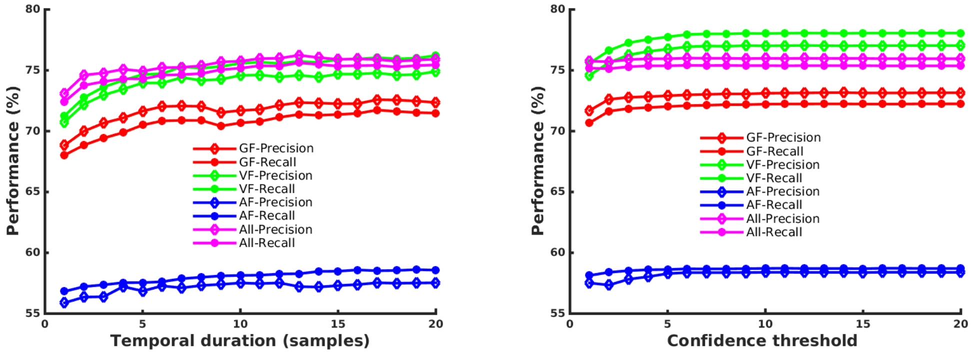
HOG [81] = 13.16 s	HOG-υ_1 = 0.15 ms HOG-υ_2 = 1.36 ms HOG-υ_3 = 0.21 ms	AF = 153.39 s
Overfeat [83] = 140.07 s	Overfeat-υ_1 = 2.73 ms Overfeat-υ_2 = 48.13 ms Overfeat-υ_3 = 4.30 ms	

a fixed set of train and test data (Fig. 4.5(a)). Finally, we set $D = 13$ samples (≈ 20 s) and it is shown that more previous knowledge does not significantly improve the performance. We set the confidence threshold, r_t, for the DTCE after similar experiment is performed iteratively as shown in Fig. 4.5(b). It is observed that all the separate feature groups (GF, VF and AF) achieve performance improvement up to $r_t = 6$, which we set for our experiments. The performance becomes stable for higher values of r_t because the DTCE follows hard-coded rules (Algorithm 2). Hence, as the threshold becomes too large to satisfy (for a fixed D), the Algorithm follows the last option and the final prediction vector becomes one of $\mathbf{a}_n$, $\mathbf{a}_n^{'}$ and $\tilde{\mathbf{a}}_n$ that has the highest confidence score. It is also observed that the DTCE is more effective on the separate feature groups than their combination, which can satisfy the threshold easily as the combination provides higher discriminative capacity.

4.6.2 Datasets

We compare state-of-the-art approaches on multiple datasets. We use a proprioceptive subset (15 hours) of the largest public FPV dataset (HUJI [1]). The video sequences were collected in unconstrained settings (Table 4.3). All video segments are preprocessed to have a 640×480

[1] `http://www.vision.huji.ac.il/egoseg/videos/dataset.html`



(a)

(b)

Figure 4.5: Experimental setting of parameters using fixed set of train and test sets; (a) different numbers of previous samples, D, are experimented; (b) different threshold values are experimented for confidence-based temporal encoding. Results show that the temporal context encoding improves the performance of separate feature groups more significantly than that of their combination.

Table 4.3: Number of video segments and their total duration per activity in the considered dataset [71]. The percentage that each activity covers of the whole dataset is also given. The class imbalance problem can be easily depicted as *Run* activity alone amounts for 47% of the whole dataset whereas *Stand* covers only 5%.

	Classes						
	Run	Sit	Go upstairs	Stand	Static	Walk	Total
Number of segments	13	11	13	15	14	19	**85**
Duration (mins)	409	96	151	47	104	62	**869**
Percentage (%)	47	11	17	5	12	7	**100**

Table 4.4: Summary of the number of video segments collected by each of the four subjects (S_1, S_2, S_3 and S_4) in the BAR dataset. Sub.: Subject; L-R: Left-right turn; S-S: Sit-Stand.

Sub.	Bow	Defend	Dribble	Jog	L-R	Pivot	Run	Shoot	S-S	Sprint	Walk	**Total**
S_1	4	3	8	4	8	14	4	30	4	2	4	**85**
S_2	4	6	8	4	4	6	4	30	4	4	4	**78**
S_3	4	9	8	4	4	14	4	29	4	4	4	**88**
S_4	4	6	6	4	5	12	4	26	5	4	4	**80**
Total	**16**	**24**	**30**	**16**	**21**	**46**	**16**	**115**	**17**	**14**	**16**	**331**

resolution and a 30 *fps* frame rate.

We also validate the proposed framework on another public dataset of basketball activity recognition, BAR[2], which is smaller (1.2 hrs) than the HUJI subset, but contains more dynamic basketball activities such as *Sprint, Dribble,* and *Shoot* (Table 4.4).

We employ equal decomposition of the available per-class video sequences into train and test sets (50% each) on the HUJI dataset [71]; whereas we employ a *one-subject-out* cross validation on the BAR dataset as the four subjects contribute equivalent amount of data. Different train

[2] `http://www.eecs.qmul.ac.uk/~andrea/FPV.html`

and test set splits enable us to experiment the proposed framework under different validation strategies. Each experiment is repeated 100 times and the average performance is reported.

4.7 Results and discussion

We evaluate the proposed hierarchical and temporal encoding approach using ten main experiments. First, we compare its recognition accuracy with the state-of-the-art methods. We compare our framework with cumulative displacement curves (CDC) [70], robust motion features (RMF) [4], average pooling (AP) [104, 108] and multi-resolution good features (MRGF) [108, 109] in the state of the art. CDC [70] is selected due to its hierarchy-based decomposition of activities similar to the proposed framework, whereas RMF [4] is chosen as it involves similar magnitude, direction and dynamics encoding strategies. AP [104, 108] is a baseline as it contains *raw grid features* with no explicit extraction of specific motion characteristics. We also evaluate MRGF [108, 109], which is a keypoint-based approach that exploits the direction of the displacement vector between matched descriptors.

Second, we evaluate the performance of each feature group on the hierarchical classification. Third, we evaluate the subgroups of each feature group separately. Fourth, we show how the proposed temporal context exploitation (TCE) strategy improves the recognition performance. The fifth experiment validates the proposed TCE when it is applied on the state-of-the-art features. Sixth, following the misclassification analysis, we show how the TCE becomes more effective when the activities are distinctively defined, first, by merging *Sit* and *Stand* to *Sit/Stand*, followed by a merging of *Static* and *Sit/Stand* to *Stationary*. Seventh, we compare our proposed pooling of the intra-frame descriptors with time-series gradient pooling [81]. Eighth, we validate the discriminative characteristics of the proposed feature groups across different classifiers, in comparison with the state of the art. The ninth experiment provides the results of three weighting strategies applied to solve the class imbalance problem. Finally, we also validate the proposed TCE on another public dataset and compare it with the state-of-the-art-methods.

4.7.1 Comparison with alternative methods

Table 4.5 shows that CDC [70], MRGF [108, 109] and AP [104, 106] achieve at least 22% lower in $\mathcal{P}$ and $\mathcal{R}$ with respect to the *Proposed*. The superiority of the proposed method is due to the higher discriminative capability of its feature groups and the use of past information via

Table 4.5: Per-class recall performance of the state-of-the-art features validated using SVM and compared with the proposed framework. $\mathcal{P}$: Precision (%); $\mathcal{R}$: Recall (%).

	Classes						Overall	
Feature	Run	Sit	Up-stair	Stand	Static	Walk	$\mathcal{P}$	$\mathcal{R}$
CDC [70]	74	42	63	12	87	48	56	56
RMF [4]	91	53	90	**15**	88	80	69	71
MRGF [108, 109]	61	19	66	14	69	40	45	47
AP [104, 106]	44	48	81	10	95	43	52	57
Proposed	**99**	**87**	**100**	3	**96**	**88**	**78**	**79**

MTCE and DTCE. Compared to RMF [4], our proposed low-level features, FMMF and grid-based virtual-inertial features, improve $\mathcal{P}$ and $\mathcal{R}$ by 13% and 11%, respectively. The results also show keypoint-based methods struggle in such a challenging dataset compared to optical flow-based methods. Since CDC [70] was proposed for the recognition of long-term activity segments ($\approx$ 17 s), it is shown to be less effective for short activity segments ($\approx$ 3 s). Generally, the results demonstrate the higher capability of our method to deal with the FPV challenges.

Among the classes, *Sit* has been improved significantly from 53% using RMF to 87% using *Proposed*. However, due to the following reasons, the same improvement cannot be achieved to *Stand* though both *Sit* and *Stand* are stationary states with head-driven motion in their first-person videos. First, the amount of data available for each of the two states is not equivalent as *Sit* (11%) contains twice the amount of data than *Stand* (5%) as shown in Table 4.3. Hence, the lack of more training information for *Stand* results in the underfitting of its model, i.e. lower performance. Second, the same temporal smoothing process in the proposed framework affects *Sit* and *Stand* differently due to their different frequencies and durations in the dataset. Compared to *Sit*, *Stand* video segments are often observed in between other activities with a shorter duration. Specifically, there are 15 *Stand* and 11 *Sit* video segments in the dataset as shown in Table 4.3. However, the average duration of a *Stand* segment is 3.15 mins ($\sigma = 7.34$ mins) compared to 9.35 mins ($\sigma = 9.15$ mins) of a *Sit* segment, where σ represents the standard deviation of segment durations. As the result, a *Stand* sample is more likely to be smoothed towards its pre-occurring activity in the sequence. Third, per the definitions of the activities in Table 4.1, *Stand* may contain a few walking steps that results in misclassification of *Stand* samples to *Walk* as shown in Fig. 4.7 and Fig. 4.8.

Table 4.6: Performance of existing and proposed feature groups at each node, M_e, of the hierarchy in terms of the binary classification accuracy, $\mathcal{A}$ (%). GF: grid features; VF: virtual-inertial features; AF: pooled appearance features.

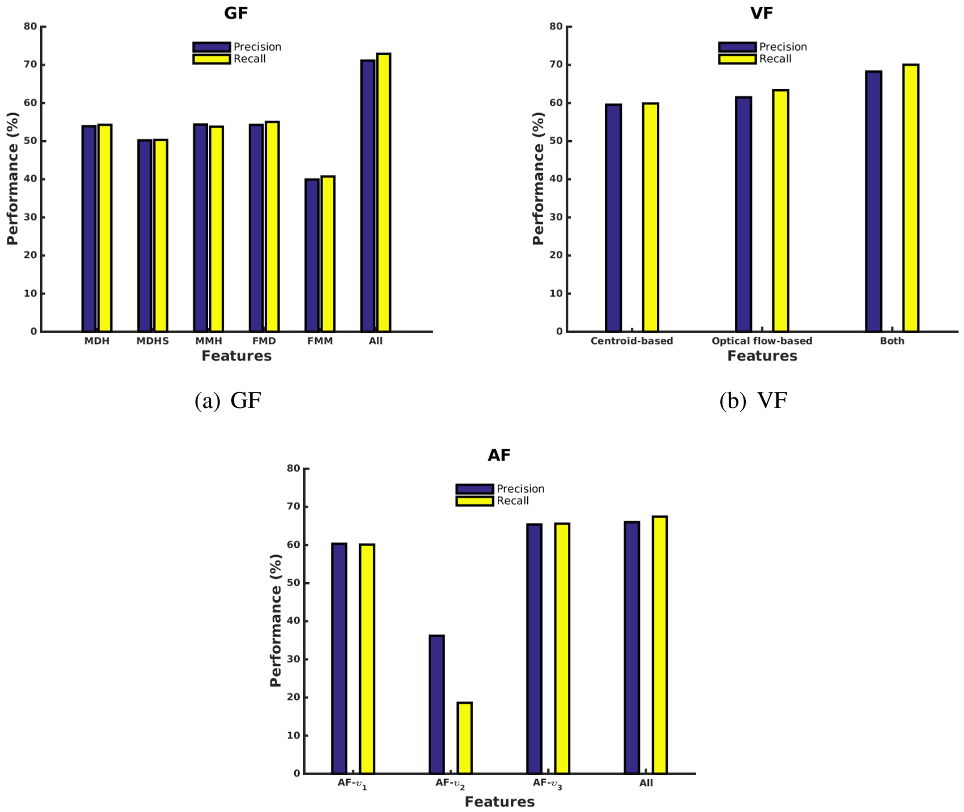
		Nodes				
	Features	M_1	M_2	M_3	M_4	M_5
Existing	CDC [70]	90	85	83	79	54
	RMF [4]	**96**	**98**	83	**90**	**58**
	MRGF [108, 109]	78	79	72	80	56
	AP [104, 106]	86	93	**88**	62	56
Proposed	GF	**96**	99	88	92	**64**
	VF	**96**	95	85	**94**	59
	AF	93	**100**	**93**	74	62
	GF+VF	**96**	98	88	**94**	62
	GF+AF	95	99	**96**	79	58
	VF+AF	94	99	95	76	63
	GF+VF+AF	**96**	**100**	**96**	86	**66**

4.7.2 Evaluation of the feature groups

We evaluate the independent performance of each feature group (and their combinations) at each node, M_e, of the hierarchy in the proposed framework, and we also compare with the state-of-the-art features as shown in Table 4.6. Note that we use the acronyms for the proposed feature groups (GF, VF and AF) for clarity, rather than the variables ($\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{f}_3$), in the following discussion. Almost all features are shown to achieve more than 85% accuracy at M_1 (*Stationary* vs. *Locomotive*). MRGF [108, 109] achieves the lowest accuracy (78%) at M_1 expectedly since it does not utilise magnitude information that could have easily discriminated the two activity sets. Note that $\mathcal{A}$ is affected by the class imbalance problem.

For all nodes, M_1 - M_5, at least one of the proposed feature groups achieves the highest accuracy. *VF* achieves higher accuracy in classifying activities with well-defined motion patterns, e.g. *Run* vs. *Walk* at M_4, whereas *GF* is more effective when the motion patterns are less distinct, e.g. *Sit* vs. *Stand* at M_5. *AF* achieves higher accuracy at M_2 (*Move along flat-space* vs. *Go upstairs*) and M_3 (*Static* vs. *Semi-static*) as there are unique appearance descriptors of *staircases* at M_2, whereas *Static* videos at M_3 contain a typical case of *a person sitting while watching a movie* or *reading on the computer screen* in the dataset.

Generally, superior performances of *GF* at M_1 and M_5, *VF* at M_1 and M_4, and *AF* at M_2 and M_3 validate our proposal of utilising different features groups according to their importance across the nodes in the hierarchy. Though the feature groups are used separately in the hierarchy, the combination of GF, VF and AF achieves the highest performance almost at all the nodes except at M_4. M_4 refers to the binary classification between *Run* and *Walk* activities. Since



(a) GF

(b) VF

(c) AF

Figure 4.6: Performance of feature subgroups in each of the proposed GF, VF and AF. The pooling operations include υ_1: standard deviation, υ_2: grouping into frequency bands and υ_3: power in frequency domain.

the two activities experience different motion dynamics, they can be easily differentiated with motion-driven features (92% with GF and 94% with VF) as shown in Table 4.6. However, these activities do not involve differences in their occurring environments, i.e. both contain similar appearance information. Hence, appearance-driven features (74% with AF) are not as discriminant as motion-driven features. Thus, the concatenation of the high dimensional AF with GF and VF introduces the less discriminative characteristics between *Run* and *Walk*, though it improves the performance at all other nodes (M_1, M_2, M_3 and M_5).

We evaluate the significance of the subgroups within each feature group: MDHF, MDHS, MMHF, FMDF and FMMF in GF; centroid-based and optical flow-based virtual inertial features in VF; and intra-frame appearance descriptors pooled with $\upsilon_1(\cdot)$, $\upsilon_2(\cdot)$ and $\upsilon_3(\cdot)$ operations. Figure 4.6 shows that GF, VF and AF achieve improved performance by including all their corresponding feature subgroups. Figure 4.6(a) illustrates that motion direction contains more dynamic information than the magnitude as depicted from their corresponding Fourier domain analysis. Figure 4.6(b) shows that the proposed optical flow-based virtual inertial feature outperforms the centroid-based inertial feature because optical flow represents more direct estimation of motion than the displacement of intensity centroid. Fig. 4.6(c) shows that $\upsilon_2(\cdot)$ pooling is less effective compared to $\upsilon_1(\cdot)$ and $\upsilon_3(\cdot)$ since $\upsilon_2(\cdot)$ reduces the original dimension of the intra-frame descriptor into few bands resulting under-fitting, whereas $\upsilon_1(\cdot)$ and $\upsilon_3(\cdot)$ keep the original feature dimension.

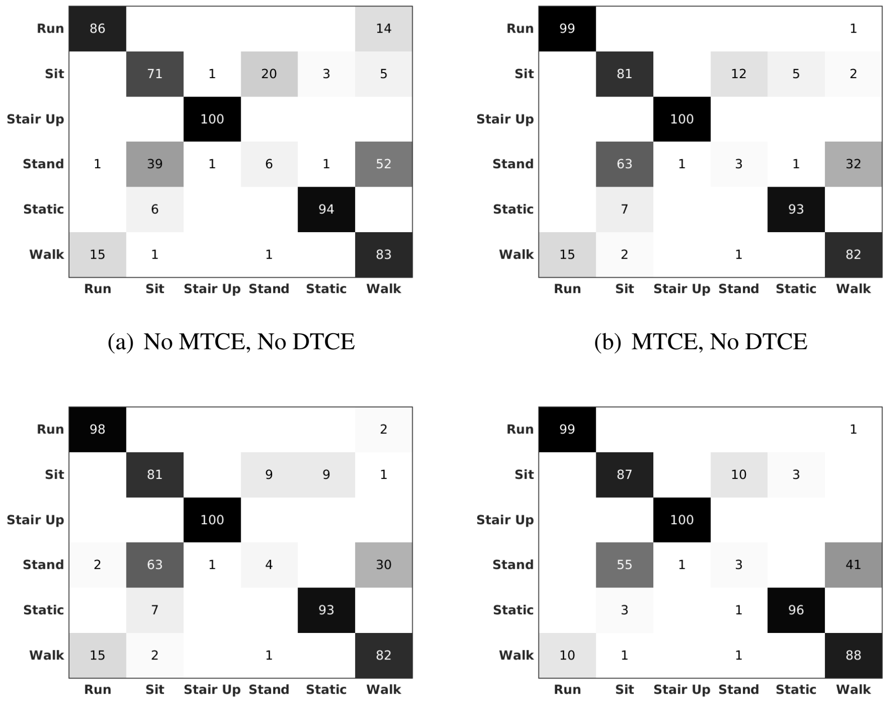
4.7.3 Temporal context

The temporal context exploitation, achieved using MTCE and DTCE, is the main reason for the superior performance of the proposed framework to the state of the art. Figure 4.7 shows the improvement of the recognition performance for almost all classes due to MTCE and DTCE. A significant improvement is observed as the misclassification of *Sit* to *Stand* reduces from 20% in Fig. 4.7(a) to 12% in Fig. 4.7(b) due to MTCE and to 9% in Fig. 4.7(c) due to DTCE. The same analogy can be applied to the 14% misclassification of *Run* to *Walk*. MTCE and DTCE are shown to improve the performance equivalently though MTCE is supposed to be more influential. However, the confidence-based smoothing and the weighted accumulation of previous outputs in DTCE plays a more crucial role than initially anticipated.

The combination of both MTCE and DTCE reduces the misclassification of *Walk* to *Run* from 15% in Fig. 4.7(a), 4.7(b), 4.7(c) to 10% in Fig. 4.7(d). The less effectiveness of the TCE for *Walk* and *Stand*, in comparison with *Run* and *Sit*, is due to the skewness problem in the dataset. Activities occurring for long temporal duration, e.g. *Run* and *Sit*, are more likely to dominate the less represented short duration activities, e.g. *Stand* and *Walk*.

Furthermore, we validate the significance of our proposed temporal continuity exploitation by applying it on the state-of-the-art features during modelling and decision. Figure 4.8 shows the following average per-class recognition improvements: 17% on CDC [70], 8% on RMF [4], 26% on MRGF [108, 109] and 22% on AP [104, 106]. This highlights the potential of our temporal context approach to advance the discriminative characteristics of any feature type.

Across the confusion matrices in Fig. 4.7 and 4.8, two misclassification errors have occurred consistently. First, *Sit* and *Stand* states are often classified with inferior performance with a significant misclassification between them. This can also be understood from the least performance at M_5 of the hierarchy in Table 4.6. The main reasons are, first, neither *Sit* nor *Stand* has distinctive characteristics (motion and/or appearance) that can be utilised during feature extraction.



	Run	Sit	Stair Up	Stand	Static	Walk
Run	86					14
Sit		71	1	20	3	5
Stair Up			100			
Stand	1	39	1	6	1	52
Static		6			94	
Walk	15	1		1		83

(a) No MTCE, No DTCE

	Run	Sit	Stair Up	Stand	Static	Walk
Run	99					1
Sit		81		12	5	2
Stair Up			100			
Stand		63	1	3	1	32
Static		7			93	
Walk	15	2		1		82

(b) MTCE, No DTCE

	Run	Sit	Stair Up	Stand	Static	Walk
Run	98					2
Sit		81		9	9	1
Stair Up			100			
Stand	2	63	1	4		30
Static		7			93	
Walk	15	2		1		82

(c) No MTCE, DTCE

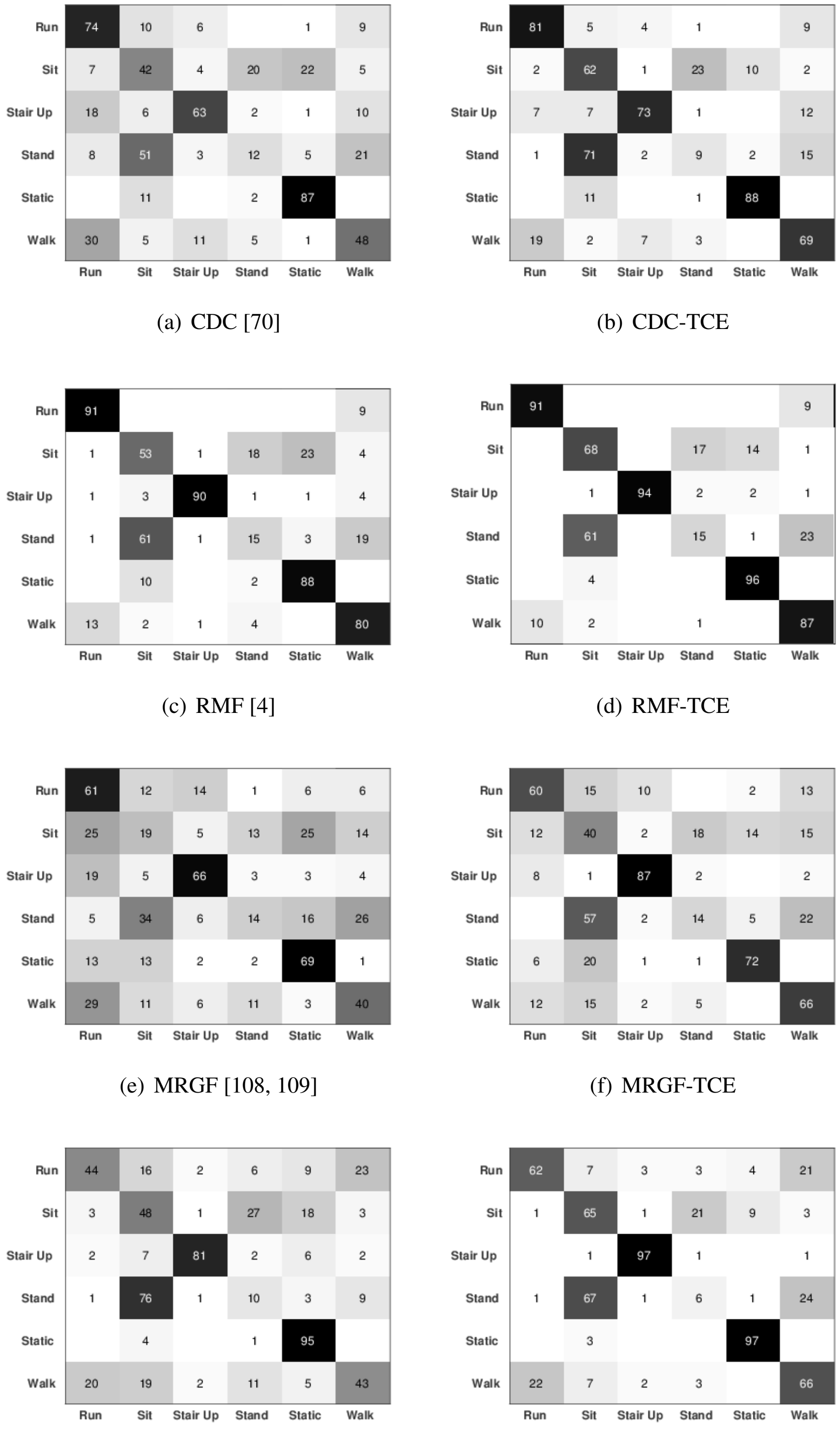
	Run	Sit	Stair Up	Stand	Static	Walk
Run	99					1
Sit		87		10	3	
Stair Up			100			
Stand		55	1	3		41
Static		3		1	96	
Walk	10	1		1		88

(d) MTCE, DTCE

Figure 4.7: Comparative performance of the proposed framework at different stages; 4.7(a): the hierarchical output without the use of any previous knowledge; 4.7(b): only previous samples knowledge is encoded during modelling (MTCE); 4.7(c): only confidence-based smoothing is applied (DTCE); 4.7(d): both MTCE and DTCE are applied.

Second, the lack of enough data for these activities in the dataset (Table 4.3) worsens the problem and results in underfitting. Misclassification of *Walk* segments to *Run* is often evident in the confusion matrices due to the significant resemblance of some *Run* segments to *Walk* segments in the dataset. In addition, the significant percentage of *Stand* activity is also misclassified as *Walk* because considerable *Stand* videos in the dataset include short walking segments as defined in Table 4.1, e.g. *a subject standing and waiting for a bus while making a few walking steps at the bus stop*.

We also validate the proposed framework by merging activities with no distinctive FPV characteristics between them. The merging also eases the class imbalance problem in the dataset. We start by merging *Sit* and *Stand* to *Sit/Stand* as they both involve random head movement while the subject is stationary. The result is shown in Fig. 4.9(a). We further merge *Static* and *Sit/Stand* to *Stationary* and the result is shown in Fig. 4.9(b). In comparison with Fig. 4.7(d), we accomplish higher performance improvement in Fig. 4.9 that confirms the effectiveness of our framework for well defined activities, and further validates the resemblance of *Sit* and *Stand* segments.



(a) CDC [70]

(b) CDC-TCE

(c) RMF [4]

(d) RMF-TCE

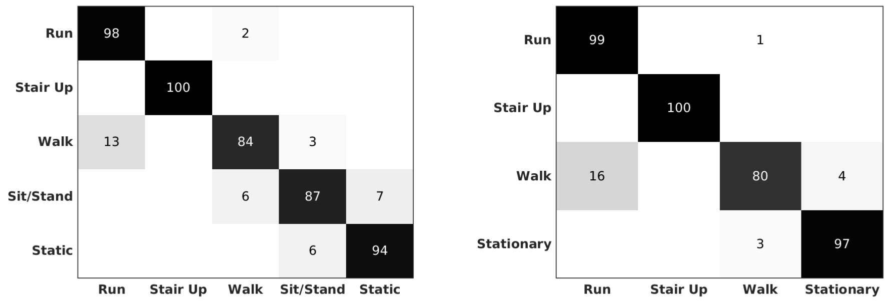
(e) MRGF [108, 109]

(f) MRGF-TCE

(g) AP [104, 106]

(h) AP-TCE

Figure 4.8: The validation of the proposed temporal continuity exploitation (TCE) on the state-of-the-art features. Figures 4.8(a), 4.8(c), 4.8(e) and 4.8(g) represent the original performances without TCE, and Figures 4.8(b), 4.8(d), 4.8(f) and 4.8(h) show their respective improved performances after TCE.



(a) (b)

Figure 4.9: Performance is improved when similar classes are merged to a single activity; (a) *Sit* and *Stand* are merged to be a *Sit/Stand* activity; (b) *Sit/Stand* and *Static* are further merged to be a *Stationary* activity. Results show the improvement of the recognition performance when less clearly distinctive activities are merged.

Table 4.7: Comparison of the proposed pooling of intra-frame appearance descriptors with the time-series gradient (TSG) pooling [81]. Per-class recall ($\mathcal{R}$) values are given followed by the overall averaged precision ($\mathcal{P}$) and recall values (%). Dim.: dimension of the feature vector obtained after the pooling; *raw* refers the summation pooling of the raw feature elements across frames.

	Per-class						Overall		
Feature	Run	Sit	Up-stair	Stand	Static	Walk	$\mathcal{P}$	$\mathcal{R}$	Dim.
HOG-raw [46]	77	21	75	1	71	62	51	51	392
HOG-TSG [81]	**81**	29	**93**	**3**	**85**	**73**	**59**	**61**	**2352**
HOG-proposed	77	**32**	90	**3**	81	65	57	58	809
Overfeat-raw [83]	78	43	99	0	92	74	62	64	4096
Overfeat-TSG [81]	**83**	57	99	**2**	**98**	**77**	**68**	**69**	**24576**
Overfeat-proposed	78	**59**	**100**	0	97	72	64	68	8217

4.7.4 Pooling

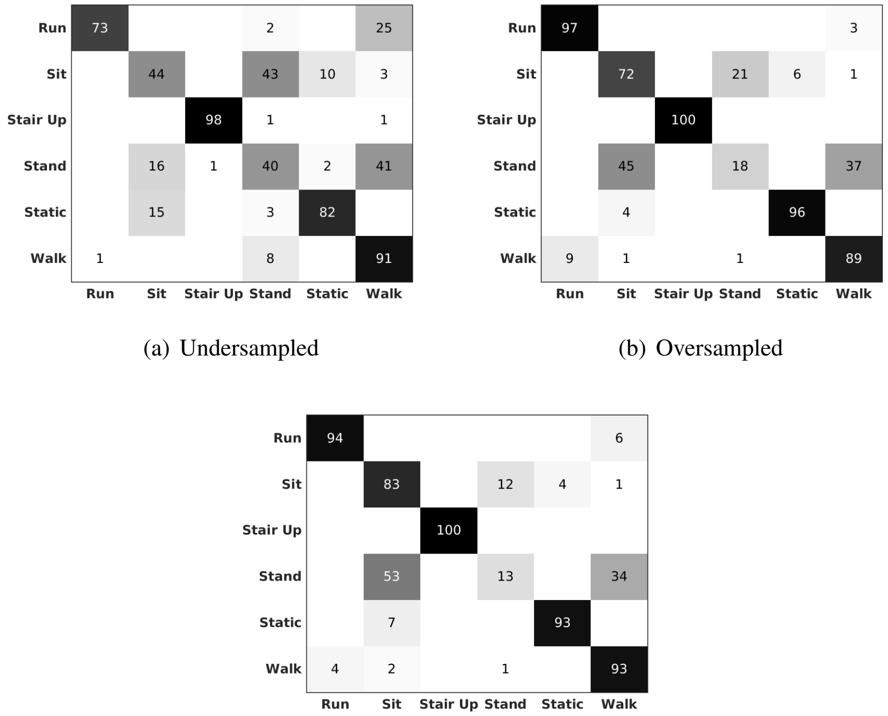
We also experiment the proposed pooling for intra-frame descriptors (HOG and Overfeat [83]) with the time-series gradient (TSG) pooling [81] as shown in Table 4.7. The results show that the proposed and TSG pooling improve the discrimination among activities in comparison with raw appearance features for both HOG and Overfeat. Among the two intra-frame descriptors, Overfeat expectedly outperforms HOG. Our pooling that contains $\upsilon_1(\cdot), \upsilon_2(\cdot)$ and $\upsilon_3(\cdot)$ often performs equivalently to TSG, while we manage to reduce the feature dimension almost three times. A specific reason for slight superiority of [81] is due to its preservation of the raw appearance information through *maximum* and *summation* pooling operations whereas our proposed approach solely focuses on motion information derived from the raw description. Generally, appearance-driven features are shown to discriminate environment-specific activities (*Go upstairs* and *Static*) near-perfectly in comparison with motion-specific activities (*Run* and *Walk*) expectedly.

Table 4.8: Accuracy, $\mathcal{A}$ (%), comparison of proposed features in the proposed framework when they are validated on different classifiers. SVM: support vector machine, KNN: k-nearest neighbours, LR: logistic regression, DT: decision tree and HMM: hidden Markov model. GF: grid features; VF: virtual-inertial features; AF: pooled appearance features.

		Node				
Feature	Classifier	M_1	M_2	M_3	M_4	M_5
GF	HMM	66	64	62	82	46
	DT	94	97	80	86	55
	KNN	95	**99**	81	88	59
	LR	69	98	60	91	52
	SVM	**96**	**99**	**88**	**92**	**64**
VF	HMM	56	81	61	43	57
	DT	93	93	81	91	56
	KNN	94	94	83	93	**60**
	LR	63	79	84	87	54
	SVM	**96**	**95**	**85**	**94**	59
AF	HMM	66	77	56	71	**63**
	DT	88	98	86	68	58
	KNN	92	96	89	**76**	52
	LR	**93**	**100**	91	72	62
	SVM	**93**	**100**	**93**	74	62

4.7.5 Classifiers

In addition to using SVM and LR, we test the proposed feature groups on different classifiers, namely KNN, decision tree (DT) and HMM. DT follows the hierarchical topology of distinguishing activities similar to the proposed framework. Table 4.8 shows the accuracy achieved by each proposed feature group at each node of the hierarchy using different classifiers. Expectedly, SVM achieves superior performance consistently across different feature groups and nodes in the hierarchy due to its discriminative and high-margin classification properties. The results validate our selection of the SVM as the principal classifier in the proposed framework. DT follows SVM closely and performs equivalently to KNN, which reflects the advantage of tree-based activity classification, i.e. the hierarchical structure in the proposed framework. HMM is shown to perform significantly inferior to the other discriminative classifiers due to its dependency on the input data model as of any generative models. LR also lags behind the SVM, DT and KNN but it provides equivalent performance to SVM on high dimensional pooled appearance features. Moreover, it is due to its simplicity that we select LR for the activity modelling using the high-level feature.



(a) Undersampled

(b) Oversampled

(c) Under-oversampled

Figure 4.10: Comparison of different weighting strategies: undersampling, oversampling and under-oversampling, applied on the dataset followed by the proposed framework. These strategies aim to achieve equal amount of data among activities. Under-oversampling provides more accurate recognition performance than the remaining two approaches as it optimizes the bias (underfitting) due to undersampling and the variance (overfitting) due to the oversampling.

4.7.6 Weighted performance

Because data size variations among activities (class imbalance) affect the recognition performance as data-scarce activities (e.g. *Stand*) do not help the model generalize. Moreover, the dominance of data-rich activities (e.g. *Run*) results in their over-smoothing during temporal encoding.

To address the class imbalance problem, we apply three weighting strategies, namely *undersampling, oversampling* and *under-over sampling*. Undersampling reduces all activities to the minimum number of samples per activity in the dataset. *Oversampling* interpolates all activities to the maximum number of samples per activity in the dataset. Figure 4.10(a) shows that undersampling introduces the reduction of recall performance for the majority of the activities except *Stand* (40%) since training is performed on less amount of data per class (i.e. a smaller dataset). Oversampling hardly achieves real data equivalence among activities as the interpolated samples are just replicas that do not introduce new information.

Table 4.9: Per-class recall (%) performance of the state-of-the-art features on the basketball activity recognition (BAR) dataset, which contains highly dynamic basketball activities. Results show that the proposed framework applied on GF and VF features result in the highest performance for the majority of the classes.

	Classes										
Feature	Bow	Defend	Dribble	Jog	L-R	Pivot	Run	Shoot	S-S	Sprint	Walk
CDC [70]	59	40	13	42	28	63	15	50	14	22	86
RMF [4]	**98**	86	82	32	**90**	98	**54**	97	68	**67**	96
MRGF [108, 109]	95	18	12	0	24	76	0	31	68	20	38
AP [104, 106]	4	0	0	0	12	34	0	3	10	0	91
Proposed	90	**89**	**95**	**54**	89	**99**	45	**100**	**71**	52	**100**

As trade-off between the two approaches, we under-oversample the dataset. This approach undersamples data-rich activities and oversamples data-scarce activities to the mean number of samples per class in the dataset. Figure 4.10(c) shows that equivalent overall performance is achieved with the original approach, but under-oversampling reduces the deviation among per class recall values from $\sigma = 37.55$ (Fig. 4.7(d)) to $\sigma = 32.98$ (Fig. 4.10(c)).

4.7.7 Validation on multiple datasets

In addition to HUJI [71], we validate the proposed temporal context exploitation approach on BAR [4]. We also apply different train and test sets split strategy (*one-subject-out*) during validation.

Table 4.9 shows that the proposed multi-layer temporal context encoding helps improve performance. GF and VF are used to classify the basketball activities separately using SVM in one-vs.-all approach. The proposed MTCE is applied on their outputs followed by the confidence-based DTCE. The recognition performance is improved for the majority of the classes. The accuracy for *Bow*, *Run* and *Sprint* is slightly reduced due to temporal smoothing. The misclassifications between *Bow* and *Sit-stand* as well as among *Jog, Run* and *Sprint* (Fig. 4.11) result due to similar motion patterns of the corresponding sequential activities in the dataset. Hence, temporal modelling would further smooth the distinction between similar and sequential activities.

4.8 Summary

We proposed a framework that exploits hierarchical and temporal information using optical flow, virtual inertial data and intra-frame appearance descriptors to classify proprioceptive activities from FPV. We extended the motion features in Chapter 3 that exploit salient characteristics of magnitude, direction and dynamics both in time and frequency domains and utilised frame-level

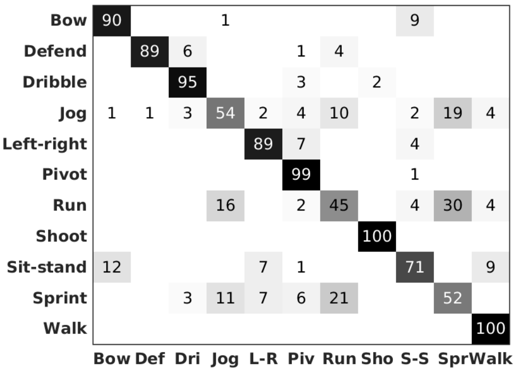


Figure 4.11: Confusion matrix of the proposed temporal context exploitation approach applied on the basketball activity recognition (BAR) dataset [4]. Misclassification among *Jog*, *Run* and *Sprint* samples due to the similar motion patterns of these activities.

appearance information using pooling operations. Each low-level motion and appearance feature group was separately used in the hierarchy in order to exploit its advantages across different nodes. We extracted a high-level feature, which contains both hierarchical and temporal information in order to model each activity. The temporal component was encoded using a temporally weighted accumulation of the hierarchical outputs of the previous samples. The classification output was further refined using a confidence-based smoothing strategy. We validated the proposed framework on multiple datasets. Results demonstrated that the proposed feature groups are more discriminative than the state-of-the-art features. We showed that appearance features can be effectively integrated to enhance performance using well-designed pooling operations. The proposed temporal continuity exploitation strategies improve the recognition performance significantly. However, an activity with shorter duration and random occurrences inside a long temporal activity might be unnecessarily smoothed.

In the next chapter, we employ deep neural frameworks to extract high-dimensional global motion features. Particularly, we aim to exploit convolutional neural networks to extract short-term motion features, and to utilise long short-term memory recurrent neural networks to encode long-term temporal dependency among activities.

Chapter 5

Multi-layer temporal encoding using deep frameworks

5.1 Introduction

The current technology offers a variety of sensors that make data collection easier than ever before. In addition to smart and small sensors, new data collection techniques, e.g. *Amazon mechanical turk*, help build large datasets in a short span of time. In addition to data, the availability of improved computational facility, e.g. GPU, reinstates neural network research, particularly, deep neural networks. Compared to the traditional (shallow) neural networks, deep networks have the capacity to learn features from data that avoids feature engineering. Inspired by the success of deep networks for image-based problems such as object recognition [27] and the availability of large datasets, deep frameworks have also been proposed for video-based activity recognition [49, 96, 101]. However, the recognition performance is not satisfactory yet due to the difficulty associated with the effective learning of spatiotemporal features [3].

Due to the success of deep networks in computer vision [27], convolutional and recursive networks have also been proposed for time-series inertial data [38, 67, 73, 74]. Particularly, recursive networks are shown to be effective in encoding the temporal information [67]. However, deep features learned from inertial data did not achieve significant superiority compared to hand-crafted (shallow) features [67, 73]. In addition to the lack of ImageNet [27]-equivalent large public datasets, existing deep frameworks for activity recognition from inertial data use separate convolutions on each motion axis thus limiting the encoding of intrinsic relationships among axial components [73, 74]. In addition, the research is relatively at an early stage and the conceptual adaptation of deep learning into the inertial domain, i.e. interpretations of learned feature, is missing along with architectural and parametric settings [38]. Furthermore, an IMU may contain multiple sensors, such as accelerometer and gyroscope, and current approaches often apply feature-level fusion using concatenation [75]. It is also desirable to effectively integrate different feature streams and/or modalities. Though IMU and FPV modalities are complementary [104], the integration of their deep features in a multi-modal setting has not been addressed yet [2].

In this chapter, first, we present a long short-term memory (LSTM) convolutional neural network (CNN) for the continuous recognition of proprioceptive activities in FPV (Fig. 5.1). We propose a novel global motion representation for intra-sample encoding, which employs a stacking of frequency-time motion representations, i.e. spectrograms. Intra-sample encoding refers to the exploitation of the motion dynamics inside a sample segment which lasts for a few seconds. The proposed representation enables us to use 2D convolutions to learn global motion of a video segment. The spectrogram representation of global motion benefits 2D convolutional networks to have reduced network parameters and less complexity compared to 3D networks. Moreover, it leads to transfer learning capabilities from models that are trained on images, e.g. ImageNet [27]. We employ an LSTM network for inter-sample temporal encoding that exploits long-term temporal dependency among activities. We validate the proposed framework against the state-of-the-art video-based temporal encoding frameworks on the largest FPV public dataset of proprioceptive activities.

Secondly, we present cross-domain knowledge transfer between inertial data and first-person video in a multi-modal setting (see Fig. 5.4). We employ the proposed CNN-LSTM framework that exploits the discriminative characteristics of multi-modal feature groups. The stacked spectrogram representation is also applied to encode the intrinsic relationships among axial motion components. We propose Hoyer-based sparsity measure [43] to integrate information from different streams and/or modalities based on their discriminative characteristics using a logistic regression (LR). This reduces the LSTM network complexity and the amount of data for training. To the best our knowledge, this work is the first that integrates deep features extracted from inertial and visual data for the recognition of proprioceptive activities in FPV. The framework is validated on multiple inertial and FPV datasets.

The chapter is organized as follows. Section 5.2 presents spectrogram-based intra-sample encoding. Section 5.3 describes the use of existing convolutional neural networks to extract

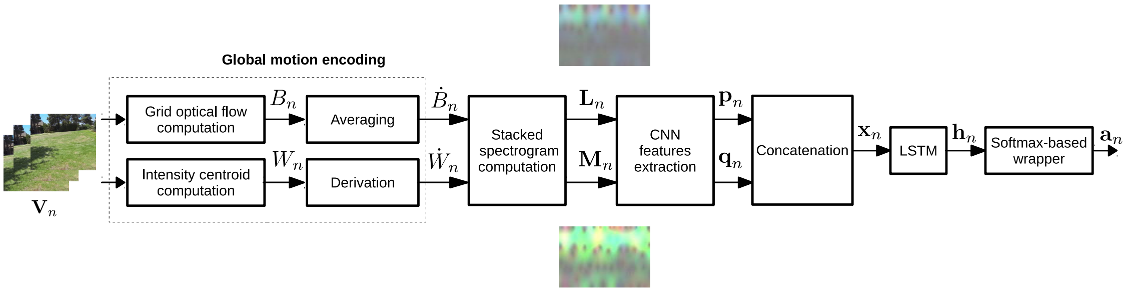


Figure 5.1: The proposed method for the recognition of proprioceptive activities using a CNN-LSTM framework employed for intra-sample and inter-sample temporal encoding in FPV.

high-level features. Section 5.4 presents the use of long short-term memory recurrent neural network to encode long temporal dependency among activities. Section 5.5 describes cross-domain knowledge transfer in a multi-modal setting that contains egocentric inertial and vision data. Section 5.6 presents a sparsity weighted combination of information from multiple streams and/or modalities. Section 5.7 evaluates the complexity of the proposed framework and compares with the state of the art, and the setting of parameters and the datasets used for validation are described in Section 5.8. Section 5.9 discusses the experimental results, and Section 5.10 summarises the chapter.

5.2 Intra-sample temporal encoding

Intra-sample encoding exploits the global motion dynamics in a sample V_n using a CNN with 2D convolutions only. We encode the global motion in V_n using two complementary motion sources: mean grid optical flow, $\dot{B}_n = \{\dot{B}_k\}_{k=1}^{L-1}$, and intensity centroid velocity, $\dot{W}_n = \{\dot{W}_k\}_{k=1}^{L-1}$, similarly as in Chapter 4.

We propose to encode the dynamics of the optical flow-based and centroid-based global motion data using frequency-domain analysis. Particularly, we derive frequency-time (spectrogram) representation of each axial component and later stack them together (Fig. 5.2). A given time-domain signal is segmented into different overlapping windows, and a fast Fourier transform (FFT), $\mathcal{F}(\cdot)$, is applied on each axis, which provides a corresponding set of spectrograms. The spectrogram contains the frequency response magnitude of all the chunks at different frequency bins. Let $\dot{B}_n^x$ and $\dot{B}_n^y$ be the horizontal and vertical axial components of the grid-based global motion, $\dot{B}_n$. The FFT applied on $\dot{B}_n^x$ and $\dot{B}_n^y$ provides the spectrograms $\bar{B}_n^x = \mathcal{F}(\dot{B}_n^x)$ and $\bar{B}_n^y = \mathcal{F}(\dot{B}_n^y)$, respectively. Inspired by the success of direction-based motion features in Chapter 3 and 4, we propose to include the spectrogram of the direction component, $\bar{B}_n^\theta = \mathcal{F}(\dot{B}_n^\theta)$ in addition to $\bar{B}_n^x$

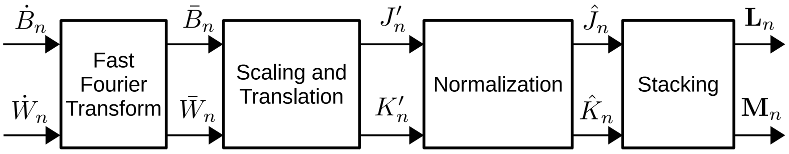


Figure 5.2: Stacking of the spectrograms from global motion representations encoded from the mean grid optical flow, $\dot{B}_n$, and centroid velocity, $\dot{W}_n$. The fast Fourier transform is first applied on each axial component to obtain frequency-time representation followed by scaling, translation and normalization operations that bound the spectrogram values to $[0,255]$. The stacking of the spectrograms provides 3-channel representation that enables us to apply transfer learning from image-based models.

and $\bar{B}^y_n$, where $\dot{B}^\theta_n = \text{arctan2}(\dot{B}^y_n, \dot{B}^x_n)$.

Following the FFT, we scale each spectrogram component of $\bar{B}_n$ by α, translate it by τ and apply normalization operations similarly to [29] in order bound the spectrogram values to $[0,255]$. For example, the scaling, translation and normalization are applied on the horizontal spectrogram component, $\bar{B}^x_n$, as

$$\bar{J}^x_n = \alpha * \bar{B}^x_n + \tau \tag{5.1}$$
$$\tilde{J}^x_n = \max(\bar{J}^x_n, 0) \tag{5.2}$$
$$\hat{J}^x_n = \min(\tilde{J}^x_n, 255). \tag{5.3}$$

Similarly, $\hat{J}^y_n$ and $\hat{J}^\theta_n$ components are derived from $\bar{B}^y_n$ and $\bar{B}^\theta_n$, respectively. In order to encode the high-level CNN features from the spectrograms with 2D convolutions, we stack the spectrograms of x, y and θ components into 3-channel motion representation as $\mathbf{L}_n = (\hat{J}^x_n, \hat{J}^y_n, \hat{J}^z_n)$. The stacking helps to encode the intrinsic relationship among multiple motion components during the convolution. The direction spectrogram is included to exploit its discriminating characteristics, unlike Ng et al. [102] that filled the third channel with zero values. The normalization enables us to visualize the stacked spectrograms as RGB images and we store the stacked spectrograms in JPEG format, which facilitates the effectiveness of applying transfer learning from image datasets, e.g. using ImageNet pre-trained CNN models.

We also employ the same stacking procedure on the centroid velocity, $\dot{W}_n$. Let the horizontal and vertical centroid velocity be $\dot{W}^x_n$ and $\dot{W}^y_n$, respectively, and their corresponding direction component be, $\dot{W}^\theta_n = \text{arctan2}(\dot{W}^y_n, \dot{W}^x_n)$. Similarly to the grid-component of the pipeline in Fig. 5.2, we employ the FFT on each component that provides $\bar{W}_n$, i.e. $\bar{W}^x_n = \mathcal{F}(\dot{W}^x_n)$, $\bar{W}^y_n = \mathcal{F}(\dot{W}^y_n)$ and $\bar{W}^\theta_n = \mathcal{F}(\dot{W}^\theta_n)$. The scaling, translation and normalization operations applied on each of $\bar{W}^x_n$, $\bar{W}^y_n$ and $\bar{W}^\theta_n$ provides $\hat{K}^x_n$, $\hat{K}^y_n$ and $\hat{K}^\theta_n$, respectively. Finally, we obtain the corresponding stacked spectrogram from the centroid velocity as $\mathbf{M}_n = (\hat{K}_n^x, \hat{K}_n^y, \hat{K}_n^\theta)$ and store it in JPEG format.

5.3 CNN-based high-level motion feature extraction

We propose to employ pre-trained CNN models to extract high-level features of the global motion from the low-level stacked spectrogram representation, which improves the generalizing capability of the features. Once the global motion streams of V_n are represented as stacked and normalized spectrograms, $\mathbf{L}_n$ and $\mathbf{M}_n$, it is possible to employ a sequence of 2D convolution filters to extract the high-level intra-sample global motion features. In addition to the benefit of transfer learning, our 2D CNN-based approach reduces the number of network parameters, and hence, the amount of data required for training. This is useful in first-person vision, where the datasets are not as large as the traditional vision datasets, such as Sports-1M [49].

We use GoogleNet [93] to extract high-dimensional inception features, $\mathbf{p}_n$ and $\mathbf{q}_n \in \mathbb{R}^D$, from $\mathbf{L}_n$ and $\mathbf{M}_n$, respectively. The final feature vector is the concatenation of the inception features, $\mathbf{x}_n = (\mathbf{p}_n, \mathbf{q}_n)^T$, where $(\cdot)^T$ represents the transpose operation. The feature $\mathbf{x}_n \in \mathbb{R}^{2D}$ encodes the temporal evolution of motion magnitude and direction inside a segment, which later becomes the input to long-term temporal dependency encoding using recurrent neural networks.

5.4 Inter-sample temporal encoding

We exploit the temporal relationships among consecutive samples or different activities to improve the recognition performance. To this end we employ a recurrent neural network (RNN) that learns temporal dynamics using previous information, $\mathbf{h}_{n-1} \in \mathbb{R}^\kappa$, in order to estimate the current hidden information, $\mathbf{h}_n \in \mathbb{R}^\kappa$, where κ is the number of neurons in the hidden layer.

The limitations of basic RNN models make learning of long temporal dependency impossible: the *vanishing* and *exploding* gradient problems, which occur during training, particularly, during error propagation. Vanishing of the gradient happens when the gradient becomes zero due to consecutive multiplications of small gradients values across the T temporal indices. This phenomenon incorrectly suggests optimal learning of the network parameters. The exploding of the gradient happens due to the consecutive multiplication of the gradient with large numbers. As the result the gradient becomes too large to minimize. This might result in the saturation of the weights gradient at the very high level that would give the incorrect impression of high discriminative capability. Comparatively, exploding gradient is easier to address, e.g. using truncating.

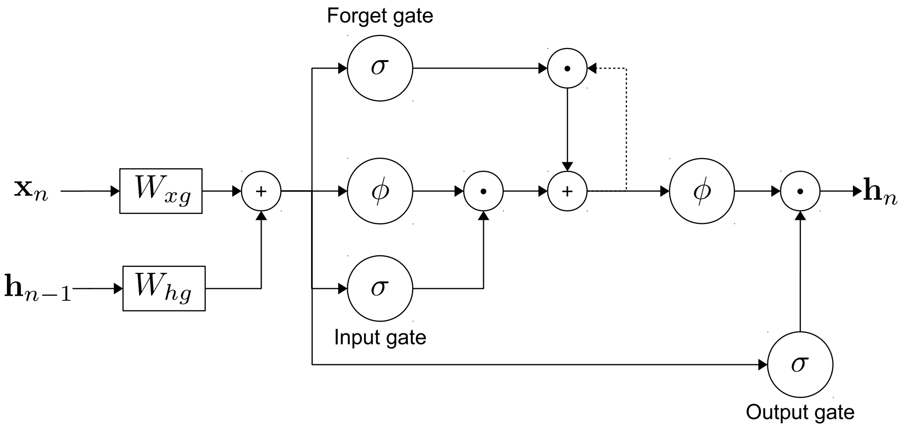


Figure 5.3: The LSTM framework used for long-term temporal dependency (inter-sample) encoding in the framework. $W_{xg} \in \{W_{xf}, W_{xi}, W_{xc}, W_{xo}\}$; $W_{hg} \in \{W_{hf}, W_{hh}, W_{hi}, W_{hc}, W_{ho}\}$; ϕ and σ are *tanh* and *sigmoid* activation functions, respectively; $\odot$ is an element wise multiplication.

As a result we employ a long short-term memory (LSTM) network that overcomes vanishing and exploding gradient problems using three additional gates: forget, input and output, that act as switches for monitoring information flow from the current input, $\mathbf{x}_n$, and previous hidden state, $\mathbf{h}_{n-1}$, to the current hidden state, $\mathbf{h}_n$, via the memory cell state $\mathbf{c}_n$.

The forget gate, $\mathbf{f}_n$, helps to discard less useful information from the previous cell state, $\mathbf{c}_{n-1}$, as

$$\mathbf{f}_n = \sigma(W_{xf}\mathbf{x}_n + W_{hf}\mathbf{h}_{n-1} + \mathbf{b}_f), \tag{5.4}$$

where $\sigma(\cdot)$ represents the *sigmoid* activation function and $\mathbf{b}_f$ is the bias in the forget gate.

The input gate, $\mathbf{i}_n$, weights the candidate cell information, $\bar{\mathbf{c}}_n$, to be the current state of the cell, $\mathbf{c}_n$, as

$$\mathbf{i}_n = \sigma(W_{xi}\mathbf{x}_n + W_{hi}\mathbf{h}_{n-1} + \mathbf{b}_i), \tag{5.5}$$

$$\bar{\mathbf{c}}_n = \phi(W_{xc}\mathbf{x}_n + W_{hc}\mathbf{h}_{n-1} + \mathbf{b}_c), \tag{5.6}$$

$$\mathbf{c}_n = \mathbf{f}_n \odot \mathbf{c}_{n-1} + \mathbf{i}_n \odot \bar{\mathbf{c}}_n, \tag{5.7}$$

where $\phi(\cdot)$ represents the *tanh* activation function, $\odot$ is an element-wise product and $\mathbf{b}_i$ and $\mathbf{b}_c$ represent the input gate and memory cell biases, respectively.

The output gate, $\mathbf{o}_n$, evaluates the cell information, $\mathbf{c}_n$, to predict $\mathbf{h}_n$.

$$\mathbf{o}_n = \sigma(W_{xo}\mathbf{x}_n + W_{ho}\mathbf{h}_{n-1} + \mathbf{b}_o) \tag{5.8}$$

$$\mathbf{h}_n = \mathbf{o}_n \odot \phi(\mathbf{c}_n) \tag{5.9}$$

The weight parameters $\{W_{hf}, W_{hh}, W_{hi}, W_{hc}, W_{ho}\} \in \mathbb{R}^{\kappa \times \kappa}$ correspond to the states $\{\mathbf{f}_n, \mathbf{h}_n, \mathbf{i}_n, \mathbf{c}_n, \mathbf{o}_n\} \in \mathbb{R}^{\kappa}$, respectively. The parameters $\{W_{xf}, W_{xi}, W_{xc}, W_{xo}\} \in \mathbb{R}^{\kappa \times 2D}$ correspond to the input, $\mathbf{x}_n \in \mathbb{R}^{2D}$.

Finally, output projection wrapper is applied using the softmax normalization that provides the activity prediction vector, $\mathbf{a}_n \in \mathbb{R}^{N_c}$, for V_n as

$$\mathbf{a}_n^c = \frac{e^{W_{ha}^c \mathbf{h}_n}}{\sum_{c=1}^{N_c} e^{W_{ha}^c \mathbf{h}_n}}, \tag{5.10}$$

where $\mathbf{a}_n^c$ is the prediction score for the c^{th} class and N_c is the number of activity classes, and $W_{ha} \in \mathbb{R}^{N_c \times \kappa}$ is the wrapping matrix.

5.5 Cross-domain knowledge transfer in a multi-modal setting

We extend the proposed FPV-based CNN-LSTM framework to integrate motion information from multi-modal data, i.e. IMU and FPV, using sparsity weighted combination that encodes the discriminative capacity of each modality as shown in Fig. 5.4.

Deep frameworks for human activity recognition (HAR) applications from time-series sensory data are often built from scratch with limited amount of data [67, 73, 74]. In addition, they do not exploit *cross-domain knowledge transfer*, i.e. the use of knowledge extracted from vision problems of similar applications. This includes the use of successful image models that are pre-trained on large image datasets such as ImageNet [27]. Transferring cross-domain knowledge from successful models, e.g. from vision research, could help reduce the amount of training data required and ease the training stage.

In inertial-vision multi-modal setting (Fig. 5.4), the n^{th} activity sample contains inertial, $\tilde{I}_n$, and first-person video, V_n, data. $\tilde{I}_n$ contains triaxial accelerometer, $\tilde{I}_{an}$ and/or gyroscope, $\tilde{I}_{gn}$, streams. Multi-stream global motion is also extracted from V_n as in Section 5.2 that resembles inertial motion representation. The stacked spectrograms for grid optical flow and centroid velocity of V_n is performed similarly as before using horizontal, vertical and direction components. But the stacked spectrograms from the gyroscope and accelerometer components of $\tilde{I}_n$ is arranged slightly different from the stacking in FPV. As accelerometer or gyroscope sample often has three dimensions, i.e. $\tilde{I}_{an} = (\tilde{I}_{an}^x, \tilde{I}_{an}^y, \tilde{I}_{an}^z)$ and $\tilde{I}_{gn} = (\tilde{I}_{gn}^x, \tilde{I}_{gn}^y, \tilde{I}_{gn}^z)$, respectively, thus, the third channel of the stack contains a normalized spectrogram of the z-component of the inertial data rather than the direction.

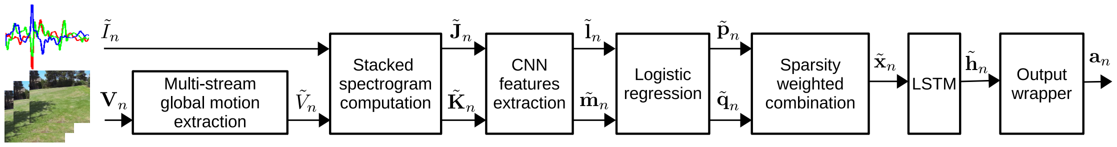


Figure 5.4: The proposed method for proprioceptive activity recognition from multi-modal egocentric data that may contain inertial and/or first-person vision data. Global motion is encoded from the mean of grid optical flow and the derivative of the intensity centroid in the video. A spectrogram representation is derived from each stream in the video and inertial data. The spectrogram values are scaled, translated, normalised and stacked to enable the extraction of CNN features that become input to a logistic classifier. The classification outputs of different streams are weighted by their sparseness and combined as input to the LSTM, which encodes the temporal dependency among activities. Finally, an output wrapper with softmax normalisation produces the activity prediction vector.

High-dimensional short-term motion features are extracted using a pre-trained CNN framework from each motion stream of inertial, $\tilde{\mathbf{J}}_n$, and visual, $\tilde{\mathbf{K}}_n$, stacked spectrograms resulting $\tilde{\mathbf{l}}_n$ and $\tilde{\mathbf{m}}_n$, respectively. For inertial component of the pipeline, the stacked spectrogram representation enables us to achieve cross-domain knowledge transfer using pre-trained image models. This avoids the need of training a dedicated deep network from scratch, i.e. it reduces system complexity.

5.6 Sparsity weighted combination

We employ sparsity measure to evaluate the decision confidence of each motion stream. The logistic regression is proposed to obtain independent classification outputs of different streams. The outputs are then weighted by their corresponding discriminative characteristics and fused as input to the LSTM. The logistic classification also transforms high-dimensional input features, $\tilde{\mathbf{l}}_n$ and $\tilde{\mathbf{m}}_n$, respectively, to $\tilde{\mathbf{p}}_n$ and $\tilde{\mathbf{q}}_n$, i.e. $\tilde{\mathbf{p}}_n, \tilde{\mathbf{q}}_n \in \mathbb{R}^{N_c}$, where N_c is the number of activities. This further reduces the complexity of the LSTM network proposed to encode the long-term temporal dependency among activities and hence, the amount of data required to train it. We apply a sigmoid function, $\sigma(\cdot)$, to transform the logistic outputs, $\tilde{\mathbf{p}}_n$ and $\tilde{\mathbf{q}}_n$, to $\tilde{\mathbf{r}}_n$ and $\tilde{\mathbf{s}}_n$, that are bounded to $(0,1)$ as

$$\sigma(\xi) = \frac{1}{1+exp(-\xi)}, \tag{5.11}$$

where $\xi \in \{\tilde{\mathbf{p}}_n, \tilde{\mathbf{q}}_n\}$. In order to compute the sparseness of the logistic classification output, we apply the Hoyer measure [43], $\psi(\cdot)$, which is an effective approach for N_c dimensional vector [44]. It is defined as

$$\psi(\eta) = \frac{\sqrt{N_c} - \frac{||\eta||_1}{||\eta||_2}}{\sqrt{N_c} - 1}, \tag{5.12}$$

where $\eta \in \{\tilde{\mathbf{r}}_n, \tilde{\mathbf{s}}_n\}$ and $||\cdot||_1$ and $||\cdot||_2$ are ℓ_1 and ℓ_2 norms, respectively. The final feature input to the LSTM network, $\tilde{\mathbf{x}}_n \in \mathbb{R}^{N_c}$, is the accumulation of the logistic classification vectors of the existing streams weighted by their corresponding sparseness measure as

$$\tilde{\mathbf{x}}_n = \sum_{\eta \in \{\tilde{\mathbf{r}}_n, \tilde{\mathbf{s}}_n\}} \eta \, \psi(\eta). \tag{5.13}$$

We employ an LSTM framework on the sparsity-weighted output to encode the long-term temporal relationships among activities. Finally, we apply an output projection wrapper on the estimated multi-modal hidden state, $\tilde{\mathbf{h}}_n$. This provides an activity prediction vector, $\mathbf{a}_n \in \mathbb{R}^{N_c}$ using the softmax normalization as in Eq. (5.10).

5.7 Complexity analysis

The grid optical flow and intensity centroid computation posses similar computational complexity as presented in Chapter 4. The lower computational cost of 2D convolutions compared to 3D convolutions arise from a fewer dimension of operations. Note that 3D convolution practically involves convolutions in four dimensions, x, y, t, m, and 2D convolution is applied in three dimensions, x, y, m, where x: horizontal; y: vertical; t: temporal; and m: filter maps. 3D convolution networks result in complex networks that further require larger datasets to train [96].

The proposed deep framework for inertial data provides less computational complexity compared to the state-of-the-art deep frameworks. Dedicated convolutional networks are often proposed to learn features for each modality in the existing frameworks. This requires exhaustive hyper-parameter setting and training of the network in addition to large data requirement and architectural design choices. The proposed framework avoids the computationally expensive CNN training stages via cross-domain knowledge transfer using existing CNN models that are trained on large image datasets. In addition to its merit-based effective integration, the sparsity weighted fusion of multi-modal features provides significant reduction of the dimension of the feature space input to the LSTM from $\mathbb{R}^{\lambda D}$ to $\mathbb{R}^{N_c}$, where λ is the number of modalities, D is the dimension of the CNN feature extracted from each modality, and N_c is the number of activity classes. This reduces the complexity of the LSTM network.

5.8 Experimental set-up

In this section we describe the state-of-the-art methods used for comparison for both visual and inertial components of the proposed pipeline, setting of parameters used in the intra-sample and inter-sample encoding stages. We also discuss the datasets used for validation.

5.8.1 Parameter setting for the visual component

We compare the vision-based pipeline of the proposed approach against four state-of-the-art video representation methods, namely C3D: 3D convolutional descriptors [96]; TDD: trajectory-pooled deep descriptors [101]; VD: VideoDarwin [34]; and TGP, time-series gradient pooling [81];.

C3D represents methods that employ 3D convolutions to learn spatiotemporal features, which result in complex networks and require large datasets for training. TDD is highly discriminative video representation that contains both spatial and temporal streams encoding using 2D convolutions. TDD also exploits the effectiveness of improved trajectory representation [100] that takes into account the camera motion, compared to the dense representation [99]. Thirdly, VD has a representation of the video data using ranking functions. It is recently extended to handle any of handcrafted or CNN features [34]. TGP is used as a baseline method, which employs histogram and sum pooling of each feature element, which is treated as a time-series data.

We set the length of an activity sample to 3 seconds, i.e, $L = 90$ and number of grids, $G = 100$, similarly to Chapter 4. We use a chunk of 15 frames with an overlap of 14 frames in the FFT to generate smooth spectrograms of the global motion vectors. The scaling factor, $\alpha = 16$, and translation of $\tau = 128$ are used, similarly to [29], which are then normalized to $(0, 255)$.

We use inception-v3, which is pre-trained on the ImageNet [27], to extract the CNN features on the spectrogram images. The inception-v3 reaches the top-5 error rate of 3.46% on ImageNet. We extracted the features from the next-to-last layer of the CNN, i.e. $'pool\_3:0'$, which provides $D = 2,048$ dimensional high-level global motion feature. For the LSTM network, we focus on its simplicity due to the limited dataset size and high dimensional feature input. Thus we employ only a single hidden layer, which contains $\kappa = 128$ neurons trained with a batch size of 100 and with 80 epochs. We set the recursive duration to contain $T_v = 20$ samples and the learning rate to be 0.01. We also resize the videos to a resolution of 320×240.

For VD, we use concatenated histograms of motion magnitude and direction, with 15 and

36 bins, respectively, similarly to [4], as input for the ranking functions, which results $D = 102$ feature vector. We use the C3D models pre-trained on the Sports-1M dataset [49] for feature extraction. For an activity sample of $L = 90$ frames, $D = 4,096$ long C3D feature is extracted from the sixth layer ($'fc6 - 1'$) for each chunk of 16 frames. Average pooling of these C3D features is performed for the final C3D representation of the activity sample.

We use the TDD model pre-trained on the UCF101 dataset. Features from *conv4* and *conv5* layers are extracted from the spatial stream, and from *conv3* and *conv4* layers are extracted from the temporal stream, each provides $D = 512$ long feature vector. We also apply both spatiotemporal and channel normalizations. The final TDD feature for an activity sample is $D = 4,096$ feature vector.

TGP is derived by applying sum and histogram pooling on the gradient of inception features extracted from flow images. TGP provides $D = 12,288$ dimension from input of $D = 2,048$ inception feature. One-vs.-all Support vector machine (SVM) classifier is used to validate the state-of-the-art methods and the intra-sample encoding evaluation of the proposed framework.

Our performance metrics to evaluate the recognition performance are precision, $\mathcal{P}$, recall, $\mathcal{R}$, and f-score, $\mathcal{F}$. We first evaluate the performance metrics per each class and finally report their average as the overall system performance. All experiments are conducted with 100 iterations and the average performance of the iterations is reported as a final recognition result.

5.8.2 Parameter setting for the inertial component

We compare six inertial-based approaches with the proposed inertial-based deep framework: Handcrafted-1 [73], Handcrafted-2 [9], Catal et al. [23], Alsheikh et al. [8], Ravi et al. [74] and Ravi et al. [73].

Handcrafted-1 [73] and Catal et al. [23] are based on low-dimensional shallow features extracted in time-domain, whereas Handcrafted-2 [9] additionally include frequency-domain features. On the other hand, Alsheikh et al. [8] and Ravi et al. [74] employed learned deep features using dedicated networks. Ravi et. al. [73] integrated the deep features in [74] with Handcrafted-1 features.

We set the parameters of the inertial component similarly to the state-of-the-art methods [9, 73, 74]. As the result, we set the window length for the inertial component to be 10 s, with no overlapping. The dimension of the shallow features Alsheikh et al. [8], Handcrafted-1 [73] and Handcrafted-2 [9] become 43, 102 and 394, respectively.

We employ the same setting as the visual component (Section 5.8.2) to extract CNN features using inception-v3. In order to compare the inception features with the state-of-the-art inertial methods in tenfold validation as in [73, 74], we employ a support-vector machine (SVM) classifier with a polynomial kernel implemented on MATLAB 2014b. We use the results of the state-of-the-art-methods reported in [73] for the comparison.

The full pipeline that contains the logistic regression, the sparsity weighted combination and the LSTM is implemented in Python 3.5. Equal amount of data is preserved for both training and test sets (50% each) in ActiveMiles and WISDM-v2.0. We use fixed train and test sets to reduce the number of iterations that linearly increases with the number of epochs in the LSTM. Each experiment is repeated ten times and the average performance is reported.

We use one-vs.-all (OVA) validation for the logistic regression. For the LSTM network, we apply similar setting as the visual component except the recursive duration for the inertial component $T_i = 10$ samples as the window length is bigger in the inertial component. For the BAR multi-modal datasets that contains both inertial and visual data, $T_v = T_i = 5$ since the dataset is small and there is no longer temporal dependency among samples. We use precision, $\mathcal{P}$, recall, $\mathcal{R}$, and accuracy, $\mathcal{A}$, to evaluate the recognition performance. We first evaluate the performance metrics per each class and finally report their average as the overall system performance.

5.8.3 Datasets

We use multiple inertial and visual datasets for validation. ActiveMiles [74] and WISDM-v2.0 [55] are the inertial datasets, whereas HUJI [71] and BAR [4] are the FPV datasets. The summary of the datasets is shown in Table 5.1.

ActiveMiles [74] is one of the largest datasets released to the public with 30 hours (h) labelled raw data ($4,390,726$ samples) collected with smartphones. It contains accelerometer and gyroscope data of seven activities: *Casual Movement, Cycling, No Activity (Idle), Public Transport, Running, Standing* and *Walking*. Ten subjects participated during the unconstrained collection. ActiveMiles includes the different sampling rates of the smartphones (50-200 Hz).

WISDM-v2.0 [55] was collected in an uncontrolled environment with 563 subjects using similar sensing configuration for six activities: *Walking, Jogging, Stairs, Sitting, Standing* and *Lying Down*. The dataset contains $2,980,765$ samples with 20 Hz rate, which covers approximately 41.4 h. WISDM-v2.0 contains only accelerometer data.

HUJI [71] is the largest public dataset for FPV activity recognition, which is collected using

Table 5.1: Summary of the datasets used for validation. Acc. is accelerometer; Gyro. is gyroscope; FPV: first-person vision; ✓: shows the existence of a specific motion data; NS: not specified; #: number; h: hour.

	Modalities					
	Inertial		Visual			
Dataset	Acc.	Gyro.	FPV	Activities (#)	Subjects (#)	Duration (h)
ActiveMiles[73]	✓	✓		7	10	30
WISDM-v2.0[51]	✓			6	530	41.4
HUJI[71]			✓	5	NS	15
BAR[4]	✓		✓	11	3	1.2

a head-mounted camera. We utilise a subset (15 hrs) of the dataset that contains *Go upstairs, Run, Walk, Sit/Stand* and *Static*. Since we are interested in full- or upper-body motion driven activities, we discard videos in the original dataset that involve the subject *travels by car* or *bus* or *rides a bicycle*. We merge *Sit* and *Stand* states as one *Sit/Stand* state since they both involve large head motion though the subject is often stationary. The *Static* state is included as a a reference, which involves neither body nor head motion of significant magnitude. Approximately 50% of the subset dataset or 17 out of 44 video sequences in the dataset are collected from publicly available YouTube videos. We applied equal decomposition of the video sequences to train and test sets as in Chapter 4.

BAR [4] dataset is composed of three warming-up exercises and eight basketball activities. This is the first dataset that includes basketball activities in FPV. The activities are *Bow, Sit-Stand, Left-right turn, Walk, Jog, Run, Sprint, Pivot, Shoot, Dribble* and *Defend*. Four subjects participated and a chest-mounted camera with 30 fps was used. Accelerometer data was also collected for the three subjects using a back-mounted inertial unit with 200 Hz.

5.9 Results and discussion

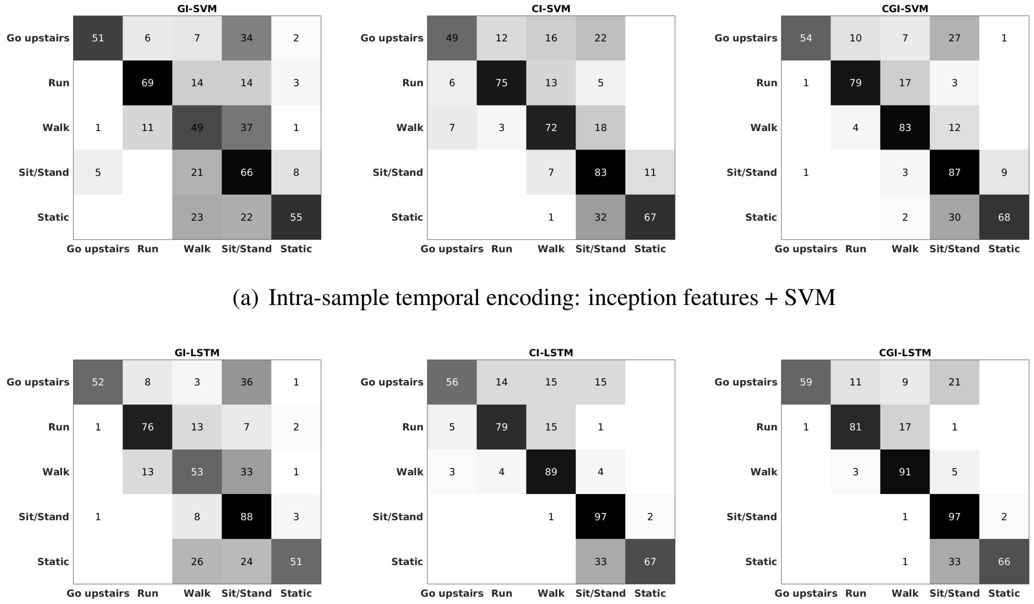
First, we describe the results achieved by the proposed intra-sample and inter-sample temporal encoding approaches and their performance comparison with existing video representations (Section 5.9.1). Second, we present the evaluation of inertial-vision cross domain knowledge transfer and sparsity weighted combination of multiple motion streams in a multi-modal setting (Section 5.9.2).

5.9.1 Intra-sample and inter-sample temporal encoding

Figure 5.5 shows the confusion matrices that validate the combination of motion features from grid optical flow data and the movement of intensity centroid. Grid inception (GI) and centroid

Table 5.2: Average per-class recall, $\mathcal{R}$, precision, $\mathcal{P}$, and f-score, $\mathcal{F}$, of various methods with the proposed intra-sample encoding framework in FPV. An SVM classifier is used for all the methods. Proposed*: the SVM output after intra-sample encoding, i.e. no inter-sample encoding.

Methods	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$	$\mathcal{F}(\%)$
TGP [81]	57	61	59
VD [34]	59	62	61
C3D [96]	64	65	65
TDD [101]	63	73	68
Proposed*	**70**	**74**	**72**



GI-SVM

	Go upstairs	Run	Walk	Sit/Stand	Static
Go upstairs	51	6	7	34	2
Run		69	14	14	3
Walk	1	11	49	37	1
Sit/Stand	5		21	66	8
Static			23	22	55

CI-SVM

	Go upstairs	Run	Walk	Sit/Stand	Static
Go upstairs	49	12	16	22	
Run	6	75	13	5	
Walk	7	3	72	18	
Sit/Stand			7	83	11
Static			1	32	67

CGI-SVM

	Go upstairs	Run	Walk	Sit/Stand	Static
Go upstairs	54	10	7	27	1
Run	1	79	17	3	
Walk		4	83	12	
Sit/Stand	1		3	87	9
Static			2	30	68

(a) Intra-sample temporal encoding: inception features + SVM

GI-LSTM

	Go upstairs	Run	Walk	Sit/Stand	Static
Go upstairs	52	8	3	36	1
Run	1	76	13	7	2
Walk		13	53	33	1
Sit/Stand	1		8	88	3
Static			26	24	51

CI-LSTM

	Go upstairs	Run	Walk	Sit/Stand	Static
Go upstairs	56	14	15	15	
Run	5	79	15	1	
Walk	3	4	89	4	
Sit/Stand			1	97	2
Static				33	67

CGI-LSTM

	Go upstairs	Run	Walk	Sit/Stand	Static
Go upstairs	59	11	9	21	
Run	1	81	17	1	
Walk		3	91	5	
Sit/Stand			1	97	2
Static			1	33	66

(b) Inter-sample temporal encoding: inception features + LSTM

Figure 5.5: Grid-based (GI) and centroid-based (CI) motion features in FPV improve performance in both (a) intra-sample and (b) inter-sample temporal encoding

inception (CI) are complementary and lead to improved performance when these two features are concatenated in both intra-sample and then inter-sample encoding components of the proposed framework. Particularly, an average of 6% f-score performance improvement is achieved in intra-sample encoding. Note that there are misclassification of *Go upstairs* activity to *Sit/Stand*, because people commonly *Stand* to take a rest during *Going upstairs* in the dataset, particularly when the the number of stairs becomes higher. Moreover, due to the similarity of their motion dynamics, *Run* and *Walk* activities are also sometimes misclassified to each other. The stationary nature of the subject involving *Sit/Stand* and *Static* also causes misclassification.

Table 5.2 compares the performance of the state-of-the-art methods with only the CNN-based intra-sample encoding part of the proposed framework. The features from all methods are validated on an SVM classifier. The concatenation of inception-features extracted from the grid-

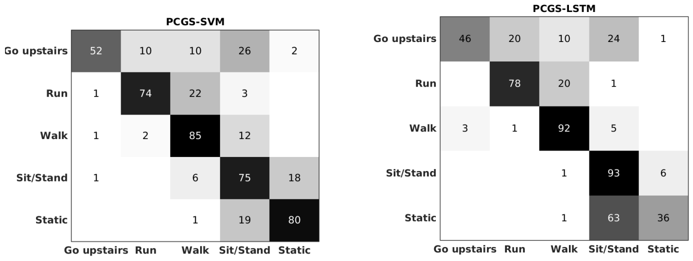
Table 5.3: The proposed LSTM-based inter-sample encoding in FPV is also evaluated for existing methods beyond the proposed inception features. Per-class recall performance (%) with and without the proposed inter-sample encoding.

	Without LSTM					With LSTM				
Methods	Go upstairs	Run	Walk	Sit/Stand	Static	Go upstairs	Run	Walk	Sit/Stand	Static
TGP [81]	52	34	82	57	81	52	34	83	60	84
VD [34]	54	67	46	73	70	55	71	55	89	89
C3D [96]	67	74	73	57	53	63	68	74	47	**94**
TDD [101]	**68**	76	**95**	52	**72**	**70**	71	86	83	39
Proposed	54	**79**	83	**87**	68	59	**81**	**91**	**97**	66

based and centroid-based normalized spectrograms (Proposed*) outperforms existing methods with at least 4% improvement in F-score. VD [34] has the smallest feature dimension ($D = 102$), but achieves slightly higher performance than the baseline method, which employs the histogram and sum pooling of the time-series gradient of the inception features, TGP [81]. TDD [101] achieves second to the proposed framework and outperforms the state-of-the-art methods since it utilises multi-stream handcrafted as well as deep learned features.

Table 5.3 shows the results for the validation of the inter-sample encoding of the proposed framework. The effectiveness of the LSTM-based long-term temporal dependency encoding is tested not only for the proposed video representation, but for state-of-the-art methods. The proposed framework achieves the best performance in the majority of the activities, i.e. *Run (*81%*),* *Walk* (91%) and *Sit/Stand* (97%). On the other hand, the proposed is inferior to TDD [101] and C3D [96] in recognizing *Go upstairs* and *Static* activities, respectively, because our proposed framework is built-on the global motion characteristics only, while TDD and C3D also include spatial (appearance) information. Indeed *Go upstairs* can be better distinguished using appearance features that detects the staircases. Similarly video sequences *Static* in the dataset are mainly collected in similar indoor environments.

We also experiment the discriminative characteristics of the spectrograms derived from the horizontal, vertical and direction components of the grid-based and centroid-based global motion by avoiding the CNN-based feature extraction in the proposed framework. The SVM classification output of the pooled spectrogram in Fig. 5.6 (a) is equivalent to the SVM outputs of the inception features in Fig. 5.5 (a) as the histogram and sum gradient pooling [81] are applied that encode the temporal variation extensively. However, the pooling operations increase the feature dimension significantly as the result the LSTM-based inter-sample temporal encoding performs worse than Fig. 5.5 (b).



(a) Pooled spectrogram + SVM (b) Pooled spectrogram + LSTM

Figure 5.6: The horizontal, vertical and direction spectrograms for each of the grid and centroid streams are independently pooled across time and all concatenated together. (a) SVM and (b) LSTM classification outputs of the concatenated pooled spectrogram features are given.

5.9.2 Cross-domain knowledge transfer in a multi-modal setting

Next, we describe results of the multi-modal framework that employs sparsity weighted combination of multiple motion streams. Table 5.4 and 5.5 show the competitive performance of the inception features with the state-of-the-art methods on the inertial datasets, without even employing the sparsity weighting and LSTM-based temporal encoding. Table 5.6 and 5.7 show significance of different fusion strategies of multi-stream information in inertial and visual datasets, respectively. The importance of the LSTM-based temporal encoding can be seen in Fig. 5.7. Finally, Table 5.8 shows the effect of different weighting strategies.

Table 5.4 shows that the overall accuracy, $\mathcal{A}(\%)$, of the proposed inception features extracted from the inertial datasets outperform the existing deep frameworks [8, 73, 74]. Unlike [73], the inception features provide improved performance without the concatenation of the shallow features. In addition, the results show that Handcrafted-2 [9] outperforms Handcrafted-1 [73] since the former includes frequency-domain features.

Table 5.5 provides a detailed comparison in per-class recall values, $\mathcal{R}(\%)$, between a deep framework baseline [73] and the proposed CNN features. Though the CNN features are extracted from a pre-trained image model, GoogleNet [93], they achieve equivalent performance with the baseline [73] that integrated deep learned and handcrafted features. Particularly, the concatenation of the inception features from accelerometer and gyroscope data in ActiveMiles provides improved performance for all activities. The equivalent performance between the proposed and the baseline features in Table 5.5 suggest that it is possible to avoid the extensive training of dedicated deep networks by using effective cross-domain knowledge transfer from vision research. The significant superiority of the proposed features in their overall accuracy (Table 5.4) than the

Table 5.4: Accuracy, $\mathcal{A}(\%)$, comparison of the state-of-the-art approaches with the proposed inception features in the inertial datasets. SVM is employed with one-vs.-all strategy in ten fold validation, similarly to [73, 74]. Prop. Inception refers to the concatenation of inception features from the accelerometer and gyroscope data in ActiveMiles, where it is only the inception features from the accelerometer in WISDM-v2.0.

	$\mathcal{A}(\%)$	
Approach	ActiveMiles [73]	WISDM-v2.0 [51]
Handcrafted-1 [73]	95.0	92.5
Handcrafted-2 [9]	98.1	97.6
Catal et al. [23]	91.7	89.8
Alsheikh et al. [8]	84.5	82.5
Ravi et al. [74]	95.1	88.5
Ravi et al. [73]	95.7	92.7
Prop. Inception	**98.8**	97.3
Prop. Inception+Handcrafted-2	98.4	**97.9**

Table 5.5: Recall, $\mathcal{R}(\%)$, comparison of the CNN features with a baseline approach [73]. ActiveMiles contains both accelerometer and gyroscope data, whereas WISDM-v2.0 contains only accelerometer data. Prop. Acc. refers to the inception features from accelerometer data, Prop. Gyro. refers to the inception features from gyroscope data, Prop. Acc.+Gyro. refers to the concatenation of the inception features from the accelerometer and the gyroscope data.

	ActiveMiles[73]						
	Casual	Cycling	Idle	Transport	Running	Standing	Walking
Ravi et al. [73]	96.1	**96.6**	96.5	95.2	98.8	**73.0**	**96.5**
Prop. Acc.	88.7	94.4	96.7	94.7	98.8	46.7	94. 8
Prop. Gyro.	92.3	90.7	80.6	89.8	97.5	15.8	91.9
Prop. Acc.+Gyro.	**98.2**	94.5	**97.1**	**96.8**	**99.4**	54.2	95.8
	WISDM-v2.0 [51]						
	Walking	Jogging	Stairs	Sitting	Standing	Lying	
Ravi et al. [73]	**97.2**	**97.7**	**77.0**	**89.3**	**82.1**	85.8	
Prop. Acc.	96.4	97.3	65.7	89.2	78.8	**88.4**	

recall values (Table 5.5) is partly due to the OVA strategy adopted, in which the rate of true negative is expectedly higher.

The first (*Individual*) part of Table 5.6 shows the individual classification outputs of feature groups from the ActiveMiles and the WISDM-v2.0 datasets using a logistic regression (LR). The second (*Fusion*) part of Table 5.6 shows the performance improvements when feature-level and decision-level fusion strategies are applied on information from different modalities and/or streams. C-LR-LSTM and C-LSTM employ feature-level fusion by concatenating the feature groups, which gives equal weight to all the feature groups. As a result, the performance improvements are not significant. LR-C-LSTM and LR-S-LSTM employ decision-level fusion using sparsity weighted concatenation and accumulation of the LR outputs, respectively. As a result significant performance improvements are achieved compared to individual feature groups. LR-S-LSTM has additional advantage as its accumulation reduces the input dimension of the LSTM,

Table 5.6: The performance of the full proposed framework implemented on the inertial datasets. First, individual performance of each feature group is given validated using logistic regression (LR). Then the performance improvement due to the fusion of multi-stream information using LSTM is given. C: concatenation; S: accumulation.

		ActiveMiles[73]		WISDM-v2.0[51]	
	Approach	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$
Individual	Inception-Acc.	41.6	33.0	65.6	58.0
	Inception-Gyro.	40.2	29.9	-	-
	Handcraft-Acc. [9]	42.1	35.9	65.3	56.0
	Handcraft-Gyro. [9]	44.5	37.2	-	-
Fusion	C-LSTM	54.0	43.6	64.3	56.2
	C-LR-LSTM	52.5	33.4	61.5	56.2
	LR-C-LSTM	61.4	53.5	66.2	57.8
	LR-S-LSTM	**61.6**	**55.2**	**72.7**	**58.4**

Table 5.7: The performance of the full proposed framework implemented on the FPV datasets. The different fusion strategies of the features groups and the use of LSTM to encode the temporal relationships improved performance.

		HUJI[71]		BAR[4]	
	Approach	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$
Individual	Inception-Grid	57.4	55.4	45.5	48.6
	Inception-Centroid	62.1	67.0	37.4	39.0
	Inception-Inertial	-	-	79.0	71.1
	Handcrafted-2 [9]	-	-	76.1	**76.3**
Fusion	C-LSTM	72.1	**78.1**	75.6	74.9
	C-LR-LSTM	70.7	74.6	47.2	49.0
	LR-C-LSTM	71.6	73.6	**83.7**	75.0
	LR-S-LSTM	**72.3**	75.4	83.1	**76.3**

and hence, reduces the size of the weight parameters, W_{xo}, W_{xi}, W_{xf} and W_{xc}. Generally, the temporal encoding using the LSTM improved the precision and recall by at least 15% in ActiveMiles. The improvement in WISDM-v2.0 is not significant since it does not contain gyroscope data, i.e. it contains fewer feature groups, compared to ActiveMiles.

The trend is similar in Table 5.7, where fusion of different feature groups showed performance improvement in the FPV datasets. Due to the larger size of HUJI dataset, C-LSTM achieves the highest performance, while proposed LR-S-LSTM provides 10% and 8% precision and recall improvements, respectively, compared to the best individual performance, i.e. Inception-Centroid. Since the BAR dataset is very small, the performance improvement due to the LSTM-based temporal encoding is not significant. But it is shown that the CNN features extracted from the stacked spectrograms of the accelerometer data perform equivalent to the hand-crafted inertial features, and higher than the CNN features from grid optical flow and centroid displacement, which demonstrates the advantage of cross-domain knowledge transfer for human activity recognition when there are multi-modal information sources.

Table 5.8: Results of different sparsity weighting strategies. NSW: only the accumulation of LR outputs without a sparsity weighting; SWNS: sparsity weighted but without a sigmoid smoothing; LR-S-LSTM: sigmoid applied on the LR outputs followed by accumulation (Proposed).

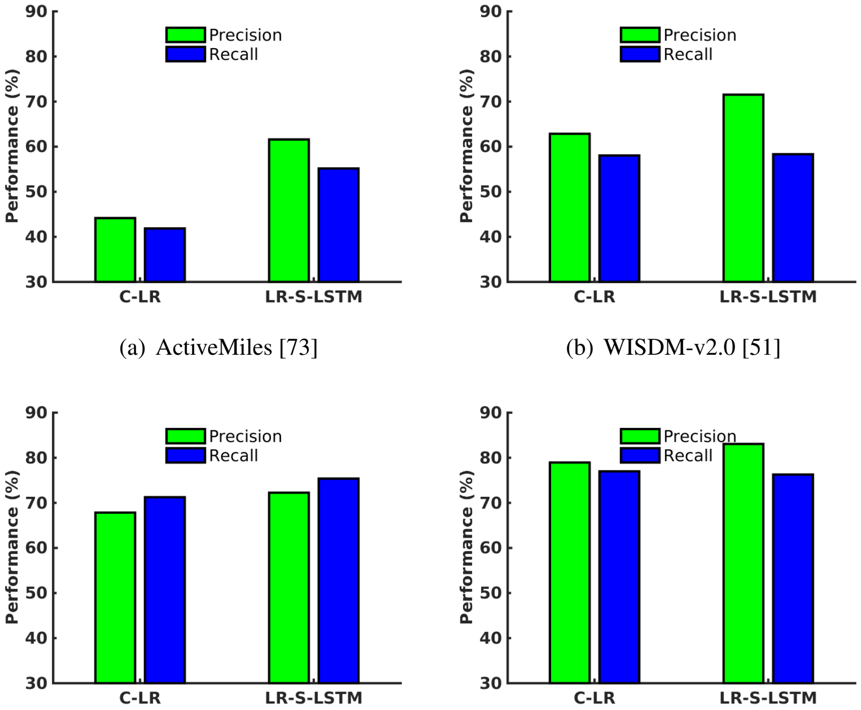
	ActiveMiles[73]		WISDM-v2.0[51]		HUJI[71]		BAR[4]	
Approach	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$
NSW	**62.5**	**65.8**	70.4	58.4	71.4	74.4	77.6	72.5
SWNS	60.5	60.9	68.6	**58.5**	71.9	**75.5**	73.9	70.4
LR-S-LSTM	61.6	55.2	**72.7**	58.4	**72.3**	75.4	**83.1**	**76.3**

Fig. 5.7 shows the significance of the LSTM-based long-term temporal encoding by comparing with C-LR outputs. C-LR employs concatenation of the features followed by the logistic regression. The results show that the LSTM improves the performance across all the datasets consistently. Particularly, the inertial components undergo significant improvements (Fig. 5.7(a) and 5.7(b)), since the inertial pipeline takes advantages of both hand-crafted and CNN-driven features. Generally, the LSTM reduces the false positives, i.e. increases the precision by utilising the long-term temporal dependency.

Table 5.8 shows the results of different weighting strategies and the importance of the sigmoid activation prior to sparsity computation. Generally, the performance improves using the proposed weighting strategy (LR-S-LSTM). Comparatively, LR-S-LSTM becomes significantly useful in the multi-modal dataset, BAR, where the inertial and visual features have different discriminative characteristics, i.e. 5.6% and 3.8% improvements of $\mathcal{P}$ and $\mathcal{R}$, respectively. The weighting however tends to suppress discriminative characteristics in ActiveMiles [73], which contains equivalent discriminative characteristics among its streams. Moreover, the importance of the sigmoid smoothing is shown across all the datasets as the performance of SWNS (sparsity weighted without sigmoid smoothing) is inferior to that of LR-S-LSTM.

5.10 Summary

We proposed long short-term memory convolutional neural network in order to continuously recognize human activities from first-person videos. The activities are characterized by dominant full- or upper-body motion. Hence, we proposed a novel global motion representation that enables us to encode the temporal information using a CNN with only 2D convolutions. In addition to its simplicity, the novel representation provides the benefit of transferring knowledge from large image datasets, and hence reduces the need of large datasets to learn global motion. On top of the CNN-based intra-sample temporal encoding, we proposed LSTM-based encoding of



(a) ActiveMiles [73]

(b) WISDM-v2.0 [51]

(c) HUJI [71]

(d) BAR [4]

Figure 5.7: The performance improvement due to the LSTM-based long-term temporal encoding in the proposed framework (LR-S-LSTM), compared with a concatenation of the features followed by the logistic regression (C-LR).

long-term temporal dependencies among samples (inter-sample) or sequential occurrence likelihood of activities. We validated the proposed framework on the largest first-person activities dataset and compared against the state-of-the-art video-based temporal encoding methods. Results showed that the proposed framework outperformed the existing methods. It is benefited by the combination of complimentary grid-based and centroid-based motion features as well as the intra-sample and the inter-sample temporal encoding strategies.

For our multi-modal framework, we integrated first-person vision with ego-centric inertial data based on their discriminative characteristics evaluated using a logistic regression. The network achieves cross-domain knowledge transfer between the two modalities. The global motion representation in first-person vision is simplified as the inertial data, whereas the stacking of the spectrograms of different inertial motion components enables us to use successful CNN-based image models to extract high-dimensional motion features. We proposed sparsity weighted accumulation of information from different motion streams and/or modalities using logistic regression. This also helps to reduce the input dimension to the LSTM network, and hence, it reduces the network complexity. LSTM network is used to encode long temporal dependency among activities. The proposed framework is validated on multiple inertial and visual datasets. Particularly, state-of-the-art performance is achieved on inertial datasets using only CNN features without explicit training of a dedicated network nor with the fusion of hand-crafted features.

The performance of the proposed framework can be further improved by re-training the last layers of the existing CNN frameworks with spectrograms by applying domain-specific data augmentation techniques.

Chapter 6

Conclusions

6.1 Summary of achievements

In this thesis, we addressed four main problems regarding human activity recognition from first-person videos.

The *first* problem focuses on the designing of multiple robust motion features. We showed that effective encoding of magnitude, direction and dynamics of optical flow-based motion data outperforms the state of the art. We also proposed novel virtual-inertial features from a video, without using the actual inertial sensors. Hence, it avoids the synchronisation issues associated with multi-modal sensing. The virtual-inertial features compliment optical flow features, and common time- and frequency-domain inertial features can be extracted from velocity and acceleration data of the intensity centroid displacement. Though each subgroup of the inertial features are susceptible to noise, their combination provides significant discrimination among activities. The high discriminative nature of the proposed set of motion features was demonstrated with multiple classifiers on multiple datasets. We also collected two novel datasets that are made publicly available.

The *second* problem involves encoding the hierarchical and temporal relationships among activities. To this end, we manually designed the hierarchy of activities where each node in the hierarchy represents a binary classification of activities with similar characteristics, e.g. *Stand* and *Sit* share a similar *Stationary* characteristics. We proposed to encode the long-term temporal dependency at two stages of the recognition pipeline, i.e. activity modelling and decision, which exploits the likelihood of subsequent occurrence of activities, e.g. *Walk* followed by *Run*. Hierarchical outputs were weighted by their temporal distance from the current sample, accumulated and provided as input to the activity modelling. During testing, confidence-based smoothing of the decision output was applied. When the decision could not achieve the minimum threshold set a priori experimentally, the previous decision outputs were exploited. In addition to the optical flow and virtual-inertial features, pooled appearance features were also used to improve the discriminative characteristics of the feature space. We validated the proposed framework on multiple datasets that include the largest FPV dataset (HUJI). Different class-balancing strategies, such as *under-over sampling*, were applied to alleviate the class-imbalance problem in the HUJI dataset.

The *third* problem involves the use of learned features for FPV activity recognition. Rather than handcrafted features that are problem specific, features learned from data using deep neural networks have better generalising capability. We employed convolutional neural networks (CNNs) to encode temporal information in a windowed segment (intra-sample dynamics). We also utilised long short-term memory (LSTM) recurrent neural network (RNN) to encode the long-term temporal dependency, i.e. inter-sample encoding. We proposed novel global motion representation using stacked spectrograms, which contain the frequency-time characteristics of the intra-sample dynamics. The mean of grid optical flow and the velocity of the intensity centroid were used to compute the global motion. The spectrograms of horizontal and vertical motion components along with their corresponding direction component were stacked, which enables the CNN to learn intrinsic relationships among different motion components. In addition, the spectrograms are scaled, translated and normalised into a 3-channel representation and stored in JPEG formats. This approach helps to learn high-dimensional intra-sample motion features using 2D convolutions rather than the 3D convolutions often applied in the state of the art. This also gives the benefit of exploiting existing 2D CNNs that were trained on large image datasets. The results showed that the proposed approach achieves competitive performance with existing methods but with lower computational complexity.

The *fourth* problem involves the cross-domain knowledge transfer between inertial-based and vision-based approaches (Chapter 5). Research to deep learning is awakened by the computer vision community, which is facilitated by the availability of large validation datasets. Motivated by its success in vision research, deep learning has also been applied on other domains, such as inertial-based activity recognition that aims to learn features from time-series inertial data. However, this involves the designing and training of dedicated convolutional neural networks that require large training datasets. In addition the features learned from such networks are not as interpretable as vision features. Hence, we proposed to exploit vision-based models as feature extractors from the inertial data. For the inertial data, we employed a 3-channel stacked spectrogram corresponding to the three axes in accelerometer or gyroscope data, followed by the scaling, translation and normalisation operations. We proposed merit-based fusion of features derived from multiple motion modalities and/or streams. The merit of a particular feature stream was measured by the sparsity of its classification outputs using a logistic regression, i.e. a highly sparse output reflects higher discriminative characteristic. The sparsity weighted and accumulated output from multiple modalities/streams was used as input to the LSTM framework that encodes long-term temporal dependency. Results showed that the proposed framework achieved equivalent to the state of the art, and it is validated on multiple inertial and visual datasets.

In addition to using existing deep frameworks, pre-trained with ImageNet, to extract features from the stacked spectrogram representations, we also experimented automatic learning of features using an autoencoder. The spectrogram representation enables us to increase the amount of training data by using both vision- and inertial-based spectrograms together. However, the recognition performance of autoencoded features is found to be still inferior to the inception features. This suggests highly discriminant characteristics of features extracted using existing deep frameworks, and more data and deeper autoencoder are necessary to improve the discrimination capability of autoencoded features.

6.2 Future directions

The multi-dimensional robust features are designed to encode the global motion in FPV, as the activities of interest are characterised by full- or upper-body motion. Future work can expand to include activities, such as person-to-person and person-to-object interactions, which require an effective integration of local and global motion features in addition to appearance descriptors [101].

In addition to the inference of the activity performed by the camera wearer, indirect inference of the other people in the scene can be performed, e.g. group activity recognition. Furthermore, the integration of FPV and the traditional (third-person vision) can be exploited for improved understanding of the scene.

In Chapter 5, we showed that cross-domain knowledge transfer can be achieved between vision and inertial pipelines, and competitive performance is obtained across multiple datasets [2]. Another future direction can include more modalities, e.g. audio, and apply cross-modal adaptation [17]. The weight of each modality can also be learned during training in the end-to-end framework with domain-adaptation functionality. *Data augmentation* techniques are shown to be effective to increase the size of image datasets required for training deep networks. However, common augmentation techniques, such as flipping, might not be effective on FPV data for ego-centric activities. Hence, domain-specific new augmentation techniques can be proposed for FPV.

The state-of-the-art recognition of egocentric activities can be facilitated by a collection of a large activity dataset, which is equivalent with ImageNet [27] of object classification and Sports1M [49] of video-based activity recognition in third-person vision. Challenges similar to ActivityNet [19] can be organized for ego-centric activities. This further facilitates the improvement of the state of the art.

With the emergence of wearable sensors, *privacy* issues are also drawing attention to the research community. Further research could be performed to ensure multi-level privacy protection. Sensitive ego-centric content can be filtered during acquisition, or modelling can be performed on encrypted data [110]. Finally, online implementation of the proposed framework can be achieved in the near future using better designed wearable technologies with improved functionalities and computation power.

Bibliography

[1] Girmaw Abebe and Andrea Cavallaro. Hierarchical modeling for first-person vision activity recognition. *Neurocomputing*, 267:362–377, December 2017.

[2] Girmaw Abebe and Andrea Cavallaro. Inertial-vision: cross-domain knowledge transfer for wearable sensors. In *Proc. of International Conference on Computer Vision (ICCV)*, pages 1392–1400, Venice, Italy, October 2017.

[3] Girmaw Abebe and Andrea Cavallaro. A long short-term memory convolutional neural network for first-person vision activity recognition. In *Proc. of International Conference on Computer Vision (ICCV)*, pages 1339–1346, Venice, Italy, October 2017.

[4] Girmaw Abebe, Andrea Cavallaro, and Xavier Parra. Robust multi-dimensional motion features for first-person vision activity recognition. *Computer Vision and Image Understanding (CVIU)*, 149:229 – 248, 2016.

[5] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. CenSurE: Center surround extremas for realtime feature detection and matching. In *Proc. of European Conference on Computer Vision (ECCV)*, 2008.

[6] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: Fast retina keypoint. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] Stefano Alletto, Giuseppe Serra, and Rita Cucchiara. Motion segmentation using visual and bio-mechanical features. In *Proc. of the ACM Multimedia*, pages 476–480, Amsterdam, The Netherlands, October 2016.

[8] Mohammad Abu Alsheikh, Ahmed Selim, Dusit Niyato, Linda Doyle, Shaowei Lin, and Hwee-Pink Tan. Deep activity recognition models with triaxial accelerometers. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, Arizona, USA, February 2016.

[9] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes. A public domain dataset for human activity recognition using smartphones. In *Euro-*

pean Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 1971–1980, Bruges, Belgium, April 2013.

[10] Yicheng Bai, Chengliu Li, Yaofeng Yue, Wenyan Jia, Jie Li, Zhi-Hong Mao, and Mingui Sun. Designing a wearable computer for lifestyle evaluation. In *Proc. of Northeast Bioengineering Conference (NEBEC)*, pages 93–94, Philadelphia, USA, March 2012.

[11] Simon Baker and Iain Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[12] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, December 2008.

[13] A. Betancourt, P. Morerio, C. S. Regazzoni, and M. Rauterberg. The evolution of first person vision methods: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(5):744–760, 2015.

[14] Monica Bianchini and Franco Scarselli. On the complexity of shallow and deep neural network classifiers. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 23–25, Bruges, Belgium, April 2014.

[15] Guillaume Bouchard and Bill Triggs. The tradeoff between generative and discriminative classifiers. In *Proc. of International Symposium on Computational Statistics (COMPSTAT)*, pages 721–728, Prague, Czech Republic, Aug 2004.

[16] Nizar Bouguila, Djemel Ziou, and Jean Vaillancourt. Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application. *Proc. of IEEE Transactions on Image Processing*, 13(11):1533–1543, 2004.

[17] Alessio Brutti and Andrea Cavallaro. Online cross-modal adaptation for audio-visual person identification with wearable cameras. *IEEE Transactions on Human-Machine Systems*, 47(1):40–51, 2017.

[18] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):1–33, January 2014.

[19] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *Proc. of*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 961–970, Boston, USA, June 2015.

[20] Niamh Caprani, Noel E O'Connor, and Cathal Gurrin. Experiencing SenseCam: a case study interview exploring seven years living with a wearable camera. In *Proceedings of the International SenseCam & Pervasive Imaging Conference*, 2013.

[21] Niamh Caprani, Noel E O'Connor, and Cathal Gurrin. Investigating older and younger peoples' motivations for lifelogging with wearable cameras. In *Proceedings of IEEE International Symposium on Technology and Society (ISTAS)*, 2013.

[22] Niamh Caprani, Paulina Piasek, Noel E O'Connor, Cathal Gurrin, Kate Irving, and Alan F Smeaton. Identifying motivations for life-long collections and their implications for lifelogging. In *Proceedings of Irish HCI Conference*, 2013.

[23] Cagatay Catal, Selin Tufekci, Elif Pirmit, and Guner Kocabag. On the use of ensemble of classifiers for accelerometer-based activity recognition. *Applied Soft Computing*, 37:1018–1022, 2015.

[24] Olivier Chapelle. Training a support vector machine in the primal. *Neural computation*, 19(5):1155–1178, 2007.

[25] Yongwon Cho, Yunyoung Nam, Yoo-Joo Choi, and We-Duke Cho. SmartBuckle: human activity recognition using a 3-axis accelerometer and a wearable camera. In *Proc. of International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, pages 1–3, Colorado, USA, June 2008.

[26] Brian P Clarkson, Kenji Mase, and Alex Pentland. Recognizing user context via wearable sensors. In *Proc. of International Symposium on Wearable Computers (ISWC)*, page 69, Atlanta, USA, October 2000.

[27] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, USA, June 2009.

[28] Aiden R Doherty and Alan F Smeaton. Automatically segmenting lifelog data into events. In *Proc. International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 20 – 23, Klagenfurt, Austria, May 2008.

[29] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Sub-

hashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, Boston, USA, June 2015.

[30] Alireza Fathi. *Learning Descriptive Models of Objects and Activities from Egocentric Video*. PhD thesis, Georgia Institute of Technology, 2013.

[31] Alireza Fathi, Jessica K Hodgins, and James M Rehg. Social interactions: A first-person perspective. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1226–1233, 2012.

[32] Alireza Fathi, Yin Li, and James M Rehg. Learning to recognize daily actions using gaze. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 314–327, 2012.

[33] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *Proc. of IEEE Conference On Computer Vision and Pattern Recognition (CVPR)*, pages 3281–3288, 2011.

[34] Basura Fernando, Efstratios Gavves, José Oramas, Amir Ghodrati, and Tinne Tuytelaars. Rank pooling for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(4):773–787, 2017.

[35] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[36] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, Hong Kong, China, May 2014.

[37] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proc. of International Conference on Machine Learning (ICML)*, volume 96, pages 148–156, Bari, Italy, July 1996.

[38] Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*, 2016.

[39] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[40] Steve Hodges, Emma Berry, and Ken Wood. SenseCam: A wearable camera that stimulates and rehabilitates autobiographical memory. *Memory*, 19(7):685–696, October 2011.

[41] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. SenseCam: A retrospective memory aid. In *Proc. of International Conference on Ubiquitous Computing (UbiComp)*, pages 177–193, California, USA, September 2006.

[42] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185 – 203, 1981.

[43] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5:1457–1469, 2004.

[44] Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

[45] Michal Irani and P Anandan. About direct methods. In *Proc. of International Workshop on Vision Algorithms: Theory and Practice*, pages 267–277, Corfu, Greece, September 1999.

[46] Yumi Iwashita, Asamichi Takamine, Ryo Kurazume, and MS Ryoo. First-person animal activity recognition from egocentric videos. In *Proc. of International Conference on Pattern Recognition (ICPR)*, pages 4310–4315, Stockholm, Sweden, August 2014.

[47] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. of ACM International Conference on Multimedia*, pages 675–678, Florida, USA, November 2014.

[48] A Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *Advances in neural information processing systems*, 14:841, 2002.

[49] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. of IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, Ohio, USA, June 2014.

[50] Kris Makoto Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3241–3248, Colorado, USA, June 2011.

[51] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[52] Oscar D. Lara and Miguel A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, November 2013.

[53] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1346–1353, Providence, USA, June 2012.

[54] Stefan Leutenegger, Margarita Chli, and Roland Yves Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 2548–2555, Barcelona, Spain, November 2011.

[55] Jeffrey W Lockhart, Gary M Weiss, Jack C Xue, Shaun T Gallagher, Andrew B Grosner, and Tony T Pulickal. Design considerations for the wisdm smart phone-based sensor mining architecture. In *Proc. of ACM International Workshop on Knowledge Discovery from Sensor Data*, pages 25–33, San Diego, USA, August 2011.

[56] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(12):91–110, 2004.

[57] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, Vancouver, Canada, 1981.

[58] Minghuang Ma, Haoqi Fan, and Kris M. Kitani. Going deeper into first-person activity recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1894–1903, Las Vegas, USA, June 2016.

[59] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in LSTMs for activity detection and early detection. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1942–1950, Las Vegas, USA, June 2016.

[60] Takuya Maekawa, Yasue Kishino, Yutaka Yanagisawa, and Yasushi Sakurai. WristSense:

Wrist-worn sensor device with camera for daily activity recognition. In *Proc. of IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 510–512, Lugano, Switzerland, March 2012.

[61] Elmar Mair, Gregory D Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 183–196, Crete, Greece, September 2010.

[62] Andrea Mannini and Angelo Maria Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, February 2010.

[63] Akio Nagasaka and Takafumi Miyatake. Real-time video mosaics using luminance-projection correlation. *Trans. IEICE*, pages 1572–1580, 1999.

[64] Yunyoung Nam, Seungmin Rho, and Chulung Lee. Physical activity recognition using multiple sensors embedded in a wearable device. *ACM Transactions on Embedded Computing Systems*, 12(2):26:1–26:14, February 2013.

[65] Sanath Narayan, Mohan S Kankanhalli, and Kalpathi R Ramakrishnan. Action and interaction recognition in first-person videos. In *Proc. of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 526 – 532, Columbus, USA, June 2014.

[66] K. Ogaki, K. M. Kitani, Y. Sugano, and Y. Sato. Coupling eye-motion and ego-motion features for first-person activity recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1 – 7, Providence, USA, June 2012.

[67] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

[68] Jorge Luis Reyes Ortiz. *Smartphone-Based Human Activity Recognition*. Springer, 2015.

[69] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2847 – 2854, Providence, USA, June 2012.

[70] Yair Poleg, Chetan Arora, and Shmuel Peleg. Temporal segmentation of egocentric videos. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2537–2544, Ohio, USA, June 2014.

[71] Yair Poleg, Ariel Ephrat, Shmuel Peleg, and Chetan Arora. Compact CNN for indexing egocentric videos. In *Proc. of IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, New York, USA, March 2016.

[72] Lawrence R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.

[73] D. Ravi, C. Wong, B. Lo, and G. Z. Yang. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2016.

[74] Daniele Ravi, Charence Wong, Benny Lo, and Guang-Zhong Yang. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *Proc. of IEEE International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 71–76, San Francisco, USA, July 2016.

[75] Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171:754–767, 2016.

[76] Daniel Rodriguez-Martin, Albert Sama, Carlos Perez-Lopez, Andreu Catala, Joan Cabestany, and Alejandro Rodriguez-Molinero. SVM-based posture identification with a single waist-located triaxial accelerometer. *Expert Systems with Applications*, 40(18):7203–7211, December 2013.

[77] Paul L Rosin. Measuring corner properties. *Computer Vision and Image Understanding (CVIU)*, 73(2):291–307, Februrary 1999.

[78] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proc. of European Conference on Computer Vision (ECCV)*, 2006.

[79] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 2564 – 2571, Barcelona, Spain, November 2011.

[80] Michael S Ryoo and Larry Matthies. First-person activity recognition: what are they doing to me? In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2730 – 2737, Portland, USA, June 2013.

[81] Michael S Ryoo, Brandon Rothrock, and Larry Matthies. Pooled motion features for first-

person videos. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 896–904, Boston, USA, March 2015.

[82] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry. *IEEE Robotics & Automation Magazine*, 18(4):80–92, December 2011.

[83] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proc. of International Conference on Learning Representations (ICLR)*, Banff, Canada, April 2014.

[84] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 806–813, Ohio, USA, June 2014.

[85] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 593 – 600, Seattle, USA, June 1994.

[86] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, Montreal, Canada, December 2014.

[87] Bharat Singh, Tim K. Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1961 – 1970, Las Vegas, USA, June 2016.

[88] Suriya Singh, Chetan Arora, and C. V. Jawahar. First person action recognition using deep learned descriptors. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2620–2628, Las Vegas, USA, June 2016.

[89] Sibo Song, Vijay Chandrasekhar, Ngai-Man Cheung, Sanath Narayan, Liyuan Li, and Joo-Hwee Lim. Activity recognition in egocentric life-logging videos. In *Proc. of Asian Conference on Computer Vision (ACCV)*, pages 445–458, Singapore, November 2014.

[90] Sibo Song, Ngai-Man Cheung, Vijay Chandrasekhar, Bappaditya Mandal, and Jie Liri. Egocentric activity recognition with multimodal Fisher vector. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2717–2721, Shanghai, China, March 2016.

[91] Ekaterina H Spriggs, Fernando De La Torre, and Martial Hebert. Temporal segmentation and activity classification from first-person sensing. In *Proc. of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 17–24, Miami, USA, June 2009.

[92] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *arXiv preprint arXiv:1011.4088*, 2010.

[93] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Boston, USA, June 2015.

[94] Cheston Tan, Hanlin Goh, Vijay Chandrasekhar, Liyuan Li, and Joo-Hwee Lim. Understanding the nature of first-person videos: Characterization and classification using low-level features. In *Proc. of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 535–542, Ohio, USA, June 2014.

[95] Philip HS Torr and Andrew Zisserman. Feature based methods for structure and motion estimation. In *Proc. of International Workshop on Vision Algorithms: Theory and Practice*, pages 278–294, Corfu, Greece, September 1999.

[96] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, Santiago, Chile, December 2015.

[97] 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 IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, pages 583–586, Taipei, Taiwan, June 2004.

[98] Ilkay Ulusoy and Christopher M Bishop. Generative versus discriminative methods for object recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 258–265, San Diego, USA, June 2005.

[99] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision (IJCV)*, 103(1):60–79, 2013.

[100] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 3551–3558, Sydney, Australia, December 2013.

[101] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4305–4314, Boston, USA, June 2015.

[102] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, Boston, USA, June 2015.

[103] Kai Zhan, Steven Faux, and Fabio Ramos. Multi-scale conditional random fields for first-person activity recognition. In *Proc. of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 51–59, Budapest, Hungary, March 2014.

[104] Kai Zhan, Steven Faux, and Fabio Ramos. Multi-scale conditional random fields for first-person activity recognition on elders and disabled patients. *Pervasive and Mobile Computing*, 16, Part B:251–267, January 2015.

[105] Kai Zhan, Vitor Guizilini, and Fabio Ramos. Dense motion segmentation for first-person activity recognition. In *Proc. of IEEE International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 123–128, Marina Bay Sands, Singapore, December 2014.

[106] Kai Zhan, Fabio Ramos, and Steven Faux. Activity recognition from a wearable camera. In *Proc. of IEEE International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 365 – 370, Guangzhou, China, December 2012.

[107] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector CNNs. *arXiv preprint arXiv:1604.07669*, 2016.

[108] Hong Zhang, Lu Li, Wenyan Jia, John D Fernstrom, Robert J Sclabassi, Zhi-Hong Mao, and Mingui Sun. Physical activity recognition based on motion in images acquired by a wearable camera. *Neurocomputing*, 74(12):2184–2192, June 2011.

[109] Hong Zhang, Lu Li, Wenyan Jia, John D Fernstrom, Robert J Sclabassi, and Mingui Sun. Recognizing physical activity from ego-motion of a camera. In *Proc. of IEEE Interna-*

tional Conference on Engineering in Medicine and Biology Society (EMBC), pages 5569–5572, Buenos Aires, Argentina, August 2010.

[110] Qingchen Zhang, Laurence T Yang, and Zhikui Chen. Privacy preserving deep computation model on cloud for big data feature learning. *IEEE Transactions on Computers*, 65(5):1351–1362, 2016.