

Multiple Action Recognition for Video Games (MARViG)



Author: Victoria Bloom

Director of Studies: Dimitrios Makris

Digital Imaging Research Centre
Faculty of Science, Engineering and Computing
Kingston University
Penrhyn Road, Kingston-upon-Thames
KT1 2EE, London, U.K.

This Thesis is being submitted in partial fulfilment of the requirements of
Kingston University for the Degree of
Doctor of Philosophy (Ph.D.)
August 2015

1. External Examiner: Prof. Robert Fisher

School of Informatics
University of Edinburgh
1.26 Informatics Forum
10 Crichton St
Edinburgh EH8 9AB

2. Internal Examiner: Dr Jaroslav Francik

PRSB3017
Faculty of Science, Engineering and Computing
Penrhyn Road
Kingston upon Thames
Surrey KT1 2EE

Day of the defence: 9th December, 2015

Signature from Chair of Ph.D. committee:

Digital Imaging Research Centre (DIRC)
Faculty of Science, Engineering and Computing (SEC)
School of Computing and Information Systems (CIS)
Kingston University London
Penrhyn Road, Kingston-upon-Thames
London, KT1 2EE
United Kingdom

DECLARATION

This report is submitted as requirement for a Ph.D. Degree in the School of Computing and Information Systems (Faculty of Science, Engineering and Computing) at Kingston University. It is substantially the result of my own work except where explicitly indicated in the text.

No portion of the work referred to in this report has been submitted in support of an application for another degree or qualification of this or any other UK or foreign examination board, university or other institute of learning.

The thesis work was conducted from October 2011 to August 2015 under the supervision of Dimitrios Makris and Vasileios Argyriou, in the Digital Imaging Research Centre (DIRC) of Kingston University in London.

Kingston-upon-Thames, London, United Kingdom.

COPYRIGHT STATEMENT

1. The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright and rights in it (the “Copyright”) and he has given to Kingston University certain rights to use such Copyright for any administrative, promotional, educational and/or teaching purposes.
2. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
3. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
4. The report may be freely copied and distributed provided the source is explicitly acknowledged and copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.
5. Further information on the conditions under which disclosure, publication, exploitation and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy, in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations and in The University's policy on presentation of Theses.

DEDICATION

Στην αγάπη μου, Ματθαίο

“If I have seen further it is by standing on the shoulders of giants.”

- Isaac Newton

ACKNOWLEDGEMENTS

First and foremost I offer my sincerest gratitude to my director of studies, Dr Dimitrios Makris, whose support began before my PhD started and remained strong throughout. I could not wish for a better or friendlier supervisor and I thank you for keeping your sense of humour even when I had lost mine. Also, I would like to thank Dr Vasileios Argyriou, my second supervisor, for his support, stimulating ideas and offering invaluable advice.

I would like to thank members and honouree members both past and present of the Digital Imaging Research Centre group. They each helped make my time during the PhD more fun and interesting. The atmosphere has always been very supportive which was evident from my interview day. I want to specially thank Prof Sergio Velastin, Prof Graeme Jones, Dr Jean-Christophe Nebel, Dr Francisco Florez Revuelta and Dr Gordon Hunter for their guidance and critical insight, Dr Michal Lewandowski and Dr Alexandros Moutzouris for disseminating their expertise, Dr Jesus Martinez-del-Ricon, Dr Maria Valera, Dr Spyros Bakos, Pau Climent Pérez and Dr Reyhaneh Esmailbeiki for their support and friendship.

I would like to extend my gratitude to Kingston University for funding my research. I am also indebted to Coventry University for permitting the completion of this PhD alongside my lecturing duties. I would also like to thank Dr Jaroslaw Francik, Dr James Orwell, Sanjay Patel and the Kingston University students for inspiring me to teach.

I would like to thank the staff, students and interns of Kingston University for participating in the dataset recordings, especially Kevin Bottero and Nicolas Ferrand for their assistance in collecting and annotating the datasets.

Finally, I would like to thank my mum, dad and husband as without their love and support none of this would be possible.

ABSTRACT

Action recognition research historically has focused on increasing accuracy on datasets in highly controlled environments. Perfect or near perfect offline action recognition accuracy on scripted datasets has been achieved. The aim of this thesis is to deal with the more complex problem of online action recognition with low latency in real world scenarios. To fulfil this aim two new multi-modal gaming datasets were captured and three novel algorithms for online action recognition were proposed.

Two new gaming datasets, G3D and G3Di for real-time action recognition with multiple actions and multi-modal data were captured and publicly released. Furthermore, G3Di was captured using a novel game-sourcing method so the actions are realistic. Three novel algorithms for online action recognition with low latency were proposed. Firstly, Dynamic Feature Selection, which combines the discriminative power of Random Forests for feature selection with an ensemble of AdaBoost classifiers for dynamic classification. Secondly, Clustered Spatio-Temporal Manifolds, which modelled the dynamics of human actions with style invariant action templates that were combined with Dynamic Time Warping for execution rate invariance. Finally, a Hierarchical Transfer Learning framework, comprised of a novel transfer learning algorithm to detect compound actions in addition to hierarchical interaction detection to recognise the actions and interactions of multiple subjects.

The proposed algorithms run in real-time with low latency ensuring they are suitable for a wide range of natural user interface applications including gaming. State-of-the art results were achieved for online action recognition. Experimental results indicate higher complexity of the G3Di dataset in comparison to the existing gaming datasets, highlighting the importance of this dataset for designing algorithms suitable for realistic interactive applications. This thesis has advanced the study of realistic action recognition and is expected to serve as a basis for further study within the research community.

TABLE OF CONTENTS

MULTIPLE ACTION RECOGNITION FOR VIDEO GAMES (MARViG)	I
DECLARATION	III
COPYRIGHT STATEMENT	IV
DEDICATION	V
ACKNOWLEDGEMENTS	1
ABSTRACT	2
TABLE OF CONTENTS	3
1.1 List of Tables.....	12
1.2 List of Figures	13
1.3 Acronyms / Abbreviations	22
1.4 Publications	24
CHAPTER 1	25
INTRODUCTION	25
1.1 Problem Summary.....	30
1.2 Aims and objectives	30
1.3 Contributions.....	31

1.4	Structure of the thesis.....	32
CHAPTER 2.....		34
BACKGROUND AND RELATED WORK.....		34
2.1	Depth Sensors	34
2.2	Overview of action recognition.....	37
2.3	Feature Extraction	40
2.3.1	features from Video.....	40
2.3.2	Features from Depth Maps.....	43
2.3.2.1	3D Silhouettes	43
2.3.2.2	Local Depth Spatio-Temporal Interest Points.....	44
2.3.2.3	Local Occupancy.....	45
2.3.2.4	3D Scene Flow	45
2.3.3	Skeleton Features	46
2.3.3.1	Skeleton Data from Motion Capture.....	46
2.3.3.2	Skeleton Data from Depth Maps.....	48
2.3.3.3	Extracting Skeleton Features	49
2.3.4	Skeleton Features used in this Thesis	50
2.4	Classification.....	53
2.4.1	Offline Action Recognition.....	53
2.4.1.1	Nearest Neighbour	53

2.4.1.2	Kernel Machines	54
2.4.1.3	Exemplar Matching	54
2.4.1.4	State Models	55
2.4.2	Online Action Recognition	56
2.4.3	Online Action Recognition in this Thesis	59
2.4.3.1	Online Action Recognition Pipeline	59
2.4.3.1.1	Binary Decision Trees	60
2.4.3.1.2	Random Forest	62
2.4.3.1.3	Boosting	63
2.5	Evaluation	64
2.5.1	Action Recognition datasets	64
2.5.1.1	Scripted Scenarios	65
2.5.1.2	Real-life scenarios	66
2.5.1.3	Datasets used in this thesis	67
2.5.2	Performance Metrics	69
2.5.2.1	Annotation	69
2.5.2.2	Performance metrics	70
2.5.2.2.1	Classification accuracy (sequence level)	71
2.5.2.2.2	Frame F1 -score (frame level)	72
2.5.2.2.3	Action Point F1 -score (instance level)	72
2.5.3	Cross Validation	74

2.5.4 Experimental Setup	75
2.6 Conclusion	75
CHAPTER 3	78
ACTION RECOGNITION USING DYNAMIC FEATURE SELECTION	78
3.1 Introduction	78
3.2 Related Work	79
3.3 Methodology	81
3.3.1 Training Phase.....	82
3.3.1.1 Feature Selection.....	83
3.3.1.2 Dynamic Classification	85
3.3.2 Testing Phase	85
3.4 G3D Dataset.....	88
3.5 Results	91
3.5.1 Datasets	91
3.5.2 Performance Metrics	92
3.5.3 Comparative Study.....	92
3.5.4 Performance Evaluation	96
3.5.4.1 Insights on Feature Selection	100

3.6	Summary	101
CHAPTER 4.....		103
ACTION RECOGNITION USING CLUSTERED SPATIO-TEMPORAL MANIFOLDS		103
4.1	Introduction	103
4.2	Related Work	105
4.2.1	Feature transformation	105
4.2.2	Early action recognition	106
4.2.3	Action prediction.....	109
4.3	Methodology	110
4.3.1	Training Phase.....	111
4.3.1.1	Dimensionality reduction	113
4.3.1.2	Clustering	114
4.3.1.3	Ordering	115
4.3.1.4	Projection	116
4.3.1.5	Peak key pose selection.....	117
4.3.1.5.1	Dynamic Time Warping Algorithm	119
4.3.2	Testing Phase	121
4.3.2.1	Early Action Recognition.....	121
4.3.2.2	Online Action Recognition	123

4.3.2.3	Action Prediction	125
4.4	Results	128
4.4.1	Datasets	128
4.4.2	Online Action Recognition	129
4.4.2.1	Performance Metrics	129
4.4.2.2	Comparative Study.....	129
4.4.2.3	Online Recognition Results.....	130
4.4.3	Early action recognition	133
4.4.3.1	Performance Metrics	133
4.4.3.2	Comparative Study.....	134
4.4.3.3	Early Action Recognition Results.....	134
4.4.4	Action Prediction	138
4.4.4.1	Performance Metrics	138
4.4.4.2	Comparative Study.....	139
4.4.4.3	Action Prediction Results.....	139

4.5	Summary	142
CHAPTER 5.....		144
COMPOUND ACTION RECOGNITION USING HIERARCHICAL TRANSFER LEARNING.....		144
5.1	Introduction	144
5.2	Related Work	146
5.2.1	Datasets	147
5.2.2	Transfer Learning.....	148
5.2.3	Interaction Recognition.....	149
5.2.4	Evaluation Metrics	151
5.3	Methodology	153
5.3.1	Training Phase (source dataset)	154
5.3.2	Model Adaptation Phase (target dataset)	155
5.3.2.1	Body Part Combinations	157
5.3.2.2	Peak Key Pose Selection.....	160
5.3.2.3	Peak Segment Detection	162
5.3.3	Testing Phase (Target Dataset)	165
5.3.3.1	Online Action Recognition	165
5.3.3.2	Hierarchical Interaction Detection Framework.....	167

5.4	G3Di dataset.....	170
5.4.1	Dataset Annotation.....	175
5.5	Results.....	175
5.5.1	Datasets.....	175
5.5.2	Skeleton Data.....	176
5.5.3	Performance Metrics.....	177
5.5.4	Comparative Study.....	178
5.5.5	Online Action and Interaction Results.....	180
5.6	Summary.....	185
CHAPTER 6.....		187
CONCLUSIONS AND FUTURE WORK.....		187
6.1	Contributions.....	187
6.1.1	Realistic Gaming Action Datasets.....	187
6.1.1.1	Issues.....	187
6.1.1.2	Proposed Solutions.....	188
6.1.1.3	Future Work.....	188
6.1.2	Dynamic Feature Selection.....	189
6.1.2.1	Issues.....	189
6.1.2.2	Proposed Solutions.....	189
6.1.2.3	Future Work.....	190

6.1.3 Clustered Spatio-Temporal Manifolds.....	190
6.1.3.1 Issues.....	190
6.1.3.2 Proposed Solutions.....	191
6.1.3.3 Future Work.....	191
6.1.4 Hierarchical Transfer Learning.....	191
6.1.4.1 Issues.....	192
6.1.4.2 Proposed Solutions.....	192
6.1.4.3 Future Work.....	192
6.2 Epilogue.....	193
BIBLIOGRAPHY.....	195
APPENDIX.....	206

1.1 List of Tables

Table 2-1 Depth sensor specifications	36
Table 2-2 Confusion matrix for two-class problems	71
Table 3-1 Dynamic Feature Selection Algorithm (Training).....	83
Table 3-2 Dynamic Feature Selection Algorithm (Testing)	86
Table 3-3 The total number of training and testing instances for gaming action datasets	92
Table 3-4 Action Point F1-scores at $\Delta=333$ ms and computation times, the average and standard deviations over ten leave-persons-out runs are shown. The results shown in italics were published by the method authors, all other results were re-created.....	97
Table 4-1 Action Point F1-scores at $\Delta=333$ ms, the average and standard deviations over ten leave-persons-out runs are shown. The results shown in italics were published by the method authors, all other results were generated by my own implementations.....	130
Table 4-2 G3D Temporal Frame Based Results: Correct classifications are shown in green and failure cases in red. The majority of failure cases were in the neutral or very early stage of the action.	136
Table 5-1 Gaming interactions for the boxing and table tennis scenarios in G3Di.	169
Table 5-2 Comparison of gaming datasets.....	174
Table 5-3 The total number of action and interaction instances used from each dataset	176
Table 5-4 A comparison of the average action latency.....	184

1.2 List of Figures

Figure 1-1 Commercial full body bowling game [1]	25
Figure 1-2 Professional rehabilitation system with biofeedback [2]	25
Figure 1-3 A humanoid robot designed to live with humans [3]	26
Figure 1-4 Educational game used in the classroom to reinforce mathematics skills [4]	26
Figure 1-5 Microsoft Kinect (front) with depth (back left), skeleton (back middle) and RGB (back right) images [8].....	27
Figure 1-6 Simple boxing sequence with a single person performing a punch (KTH) [10].....	28
Figure 1-7 A continuous stream of different actions from the G3D dataset.	28
Figure 1-8 Viewpoint differences: changes in camera position relative to the subject	29
Figure 1-9 Anthropometric variations: differences in size and proportions of the human body	29
Figure 2-1 Inside the Kinect: Depth and Colour Sensors [16].....	35
Figure 2-2 Levels of human activities.....	37
Figure 2-3 Peak poses for different actions, from left to right: right punch, left punch, right kick, left kick and defend	38
Figure 2-4 Action recognition generic pipeline	39
Figure 2-5 Actions along with their Motion History Images, from left to right: sit- down, sit-down MHI, arms-wave and arms-wave MHI [25].....	41

Figure 2-6 Appearance-based features, from left to right: colour, dense optical flow, spatial gradients and temporal gradients [7]	41
Figure 2-7 Left: depth map and right: sampled 3D points [19]	44
Figure 2-10 Overview of real-time pose recognition. From a single input depth image (left), a per-pixel body part distribution is inferred (middle), local modes are estimated to give proposals for the 3D locations of body joints (right) [6]	48
Figure 2-11 Qualitative features describing geometric relations between the body points of a pose that are indicated by red and black markers (image adapted from [56]).....	50
Figure 2-12 Pose-based features used in this thesis	51
Figure 2-13 Exemplar matching between two kicking sequences with different non-linear execution rates. Each number represents a particular pose of the subject. [22].....	54
Figure 2-14 An example hidden Markov model for the action stretching an arm [22].	56
Figure 2-15 Online action recognition pipeline, the key differences with the offline approach are the streamed testing data, an additional post processing step to temporally detect the action and the action point F1 latency measure.	60
Figure 2-16 Decision Tree Example for classifying the species of flower (Setosa, Versicolor, Virginica) by petal measurements [76]	61

Figure 2-17 Random Forest: consisting of multiple decision trees learnt on random subsets of the training data. At each node a small subset of variables are selected at random and the variable that optimises the split is found [77]	62
Figure 2-18 KTH [10] intensity data	65
Figure 2-19 HDM05 [82] mocap data	65
Figure 2-20 Action3D [40] depth data	65
Figure 2-21 UCF Kinect dataset [12]	66
Figure 2-22 PETS [84]	67
Figure 2-23 Hollywood 2 [86]	67
Figure 2-24 UCF sports action dataset [27]	67
Figure 2-25 MSRC-12 Gaming Actions instructions provided to subjects (image modality) [71]	68
Figure 2-26 Annotation [13]	69
Figure 2-27 Action point F1 metric for a single action: a fixed time window of size 2Δ is centered around the ground truth action point annotation (marked ●) and used to split the three detected action points into correct (marked ○) and incorrect detections (marked ×). If there is more than one detected action point within the ground truth window only one prediction is counted. All incorrect detections are counted.	73
Figure 3-1 Dynamic Feature Selection (Training)	82
Figure 3-2 Dynamic Classification (testing)	86
Figure 3-3 Frame based certainties for a fighting sequence from the G3D dataset	87

Figure 3-4 Smoothed results for a fighting sequence from the G3D dataset and detected action points for $nw = 10$	87
Figure 3-5 Colour image	88
Figure 3-6 Depth map	88
Figure 3-7 Skeleton data	88
Figure 3-8 Correctly inferred joints (yellow).....	89
Figure 3-9 Incorrectly inferred joints (yellow).	89
Figure 3-10 A fighting sequence from the G3D dataset with "action point" ground truth	90
Figure 3-16 Change Weapon Action Point frame.....	98
Figure 3-17 Night Goggles frame near the end of the action.....	98
Figure 3-18 G3D Fighting results by action, all experiments conducted with a single-frame feature vector	99
Figure 3-19 MSRC-12 Fighting results by action, all experiments conducted with a single-frame feature vector	99
Figure 3-20 Feature Importance (G3D)	101
Figure 3-21 Feature Importance (MSRC-12).....	101
Figure 4-1 Observations required for offline, early, online action recognition and prediction.....	105
Figure 4-2 Action templates with four key stages: dimensionality reduction, clustering, ordering and projection.	112
Figure 4-3 TLE: temporal neighbours (green dots) of a given data of a given data point, x_{it} , (red dots) in a) adjacent and b) repetition graphs. [106]113	

Figure 4-4 Clustered Spatio-Temporal manifold with the low dimensional points Y shown as points, coloured according to their cluster and the cluster centers C as black circles.....	115
Figure 4-5 Cyclic clustered action manifold and highest probability transitions	116
Figure 4-6 Right punch action template.....	117
Figure 4-7 Template fragment matching: peak pose fragment (left), matched key pose fragment (right)	118
Figure 4-8 Comparison of the Euclidean and DTW matching. (a) The Euclidean matching compares the samples at the same time instants, whereas (b) the DTW measure compares samples with similar shapes to minimise the distance [122].	119
Figure 4-9 Cross-distance matrix Γ between sequences Q and R , showing the optimum warping path L , that minimises the distance between Q and R [123]	120
Figure 4-10 Template fragment matching: observed test poses and matched action template	121
Figure 4-11 (Top) Normalised DTW distance for each frame (Bottom) Action classification label for each frame. At this stage all frames are classified as an action, even the neutral frames. To overcome this limitation action points are detected at the next stage to only classify the peak frame of each action.....	123
Figure 4-12 Clustered Action Manifold cluster indices for each frame with ground truth action points (*) and detected action points (o).....	125

Figure 4-13 Right Punch Clustered Action Manifold with peak key pose index <i>ip</i> with matched key pose index <i>im</i> and last matched key pose index <i>il</i> ..	125
Figure 4-14 Linear regression at time <i>td</i> to predict the time <i>tp</i> at which the partially observed action will reach its peak.	127
Figure 4-15 Clustered Action Manifold cluster indices for each frame with ground truth action points (*) and detected action points (o).....	132
Figure 4-16 G3D Fighting Online Action Recognition Results by Action.....	132
Figure 4-17 MSRC-12 Fighting Online Action Recognition Results by Action	133
Figure 4-18 G3D Frame F1-scores, the average over ten leave-persons-out runs are shown	135
Figure 4-19 MSRC-12 Frame F1-scores, the average over ten leave-persons-out runs are shown	135
Figure 4-20 G3D Action Point F1-score curves, the average over ten leave-persons-out runs are shown	140
Figure 4-21 MSRC-12 Action Point F1-score curves, the average over ten leave-persons-out runs are shown.....	140
Figure 4-22 Two subjects performing a (right kick), at different speeds (classified right kick poses •, classified left kick poses •, ground truth peak pose *, predicted peak pose ◦)	141

Figure 5-1 Boxing interactions: A real attack (left) occurs when one person punches the other person and makes physical contact, in contrast a virtual attack (middle) and a virtual block (right) occur when both players face the screen and perform actions toward the computer screen so there is no physical contact.....	145
Figure 5-2 Localisation results from the Multi-KTH dataset, red - handclapping, blue - boxing, yellow - running, pink - walking, green - hand waving [139]	152
Figure 5-3 Methodology Overview.....	153
Figure 5-4 Training overview which is performed on the source dataset for each action	154
Figure 5-5 Right punch action template, consisting of key poses k_1 to k_m where m is the number of clusters	155
Figure 5-6 Model Adaptation overview which is performed on the target dataset for each action.....	155
Figure 5-7 Body parts: the skeleton is divided into body parts, right arm (red), left arm (blue), right leg (green), left leg (pink) and torso (black).	156
Figure 5-8 Body Part Combinations: The selection factors (W) are optimised for each action based on their ability to discriminate compound actions in the target dataset. The bottom skeletons show potential body parts configurations for the defence (left) and right punch (right) actions.	159
Figure 5-9 Peak key pose selection: each action is considered independently at this stage.....	161

Figure 5-10 (Top) Interaction detection based on action segments which correctly detects actions with long duration. (Bottom) Interaction detection based on action points, which only works if both actions occur at the same time and incorrectly detects interactions if an action has a long duration. 163

Figure 5-11 Normalised DTW distances: the lowest value represents the most similar action, where this value is lower than the threshold T it represents the detected action. The right punch is displayed in yellow, left punch displayed in green and the defence in magenta..... 164

Figure 5-12 Hierarchical view of interaction recognition performed on the target dataset..... 165

Figure 5-15 Recording environment with 2 depth cameras for simultaneous gameplay and recording..... 171

Figure 5-16 Complex fighting sequences between multiple players, performing multiple actions in quick succession so that the movements temporally overlap (G3Di) [143]. Each row represents a different sequence with visual examples taken every 3 frames..... 172

Figure 5-17 Synchronised colour, depth and skeleton data from a boxing game 173

Figure 5-18 A timeline for a boxing game, showing the true positives (TP), false positives (FP) and false negatives (FN). A TP, is a correct interaction identified within Δ frames of the ground truth. A FN, is an undetected interaction on the ground truth..... 178

Figure 5-21 Example of a typical failure case caused by noisy skeleton data. The colour image (right) shows that this is a block interaction but the algorithm detects an attack interaction as the defence action is not correctly detected due to incorrect skeleton data for the player on the left. This instance will be penalised twice by the action point metric, firstly a FP for the attack and secondly a FN for the block. 184

1.3 Acronyms / Abbreviations

2D	Two dimensional
3D	Three dimensional
4D	Four dimensional
AFD	Average Frame Distance
ARMA	Autoregressive Moving-Average Model
BoW	Bag-of-Words
BPM	Body Part Matching
CCTV	Closed-circuit television
CPU	Central Processing Unit
CSTM	Clustered Spatio-Temporal Manifolds
DFS	Dynamic Feature Selection
DM	Diffusion Map
DTW	Dynamic Time Warping
fn	False negative
FOV	Field of View
FPS	First Person Shooter
fp	False positive
fps	Frames Per Second
G3D	Multimodal Gaming Action Dataset
G3Di	Multimodal Gaming Action and Interaction Dataset
GPU	Graphical Processing Unit
HMM	Hidden Markov Models
HTL	Hierarchical Transfer Learning framework
IR	Infrared
LE	Laplacian Eigenmaps

LOSOVCV	Leave-one-subject out cross validation
MEI	Motion Energy Image
MHI	Motion History Image
MSRC-12	Microsoft Research Cambridge-12 Kinect gesture data set
NUI	Natural User Interface
OOB	Out of Bag
OS	Operating System
OVA	One-vs-All
PNG	Portable Network Graphics
PSM	Peak Segment Matching
RBFN	Radial Basis Function Network
RGB	Red Green Blue
STIP	Spatio-temporal Interest Points
SVM	Support Vector Machine
TLE	Temporal Laplacian Eigenmaps
TLM	Transfer Learning Matching
tn	True negative
tp	True positive
VLMM	Variable-Length Markov models
VR	Virtual Reality
XML	Extensible Markup Language

1.4 Publications

V. Bloom, V. Argyriou, and D. Makris, “Hierarchical Transfer Learning for Online Recognition of Compound Actions” Computer Vision and Image Understanding, 2015.

V. Bloom, V. Argyriou, and D. Makris, “Clustered Spatio-Temporal Manifolds for Online Action Recognition” in IEEE International Conference on Pattern Recognition, 2014.

V. Bloom, V. Argyriou, and D. Makris, “G3Di: A Gaming Interaction Dataset with a Real Time Detection and Evaluation Framework”, in European Conference on Computer Vision Workshop, 2014

V. Bloom, V. Argyriou, and D. Makris, “Dynamic Feature Selection for Online Action Recognition,” in ACM Multimedia Workshop, 2013.

V. Bloom, D. Makris, and V. Argyriou, “G3D: A gaming action dataset and real time action recognition evaluation framework,” in Computer Vision and Pattern Recognition Workshops, 2012.

This thesis is substantially the result of my own work under the guidance and supervision of Dr Dimitrios Makris and Dr Vasileios Argyriou. The literature reviews, programming, experiments, results and evaluation were all my undertaking. My progress was discussed in regular meetings with my supervisors and their feedback was incorporated into my work.

CHAPTER 1

INTRODUCTION

The research field of human action recognition has rapidly expanded in recent years with many innovative applications in a range of sectors including healthcare, education, robotics and entertainment (as illustrated in Figure 1-1 to Figure 1-4). In healthcare, action recognition enables touch-free browsing of medical images in operating rooms, physical therapy at home and in clinics and patient monitoring. In education, action recognition can increase the engagement of users by providing realistic and immersive training simulations. In robotics, action recognition facilitates natural interaction between humans and robots. In entertainment, action recognition enables touch-free interaction with Smart TVs and games consoles for more intuitive and natural interaction. A key requirement of these interactive applications is the ability to robustly detect actions in real-time so the system can provide an appropriate response to the user with no apparent delay.



Figure 1-1 Commercial full body bowling game [1]



Figure 1-2 Professional rehabilitation system with biofeedback [2]

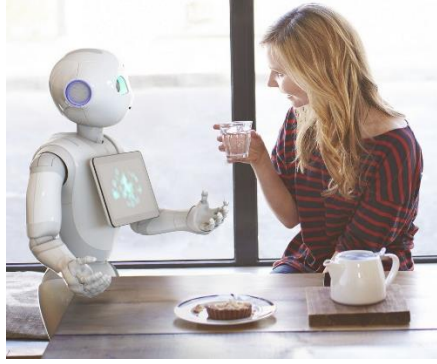


Figure 1-3 A humanoid robot designed to live with humans [3]



Figure 1-4 Educational game used in the classroom to reinforce mathematics skills [4]

Human action recognition is an active area of research in computer vision. In the past, research focused on recognising actions from video cameras. Low level appearance features were extracted from the colour images and pose-based approaches were previously disregarded due to the complexity of estimating the human pose. However, the release of the first low cost depth sensor [5] combined with a real-time state-of-the-art pose estimation algorithm [6] has facilitated the rapid growth of research on depth and skeleton data. Yao et al. [7] experiments showed that pose-based features outperform low-level appearance features in a home monitoring scenario.

Each modality: depth, skeleton and colour as exemplified in Figure 1-5 has advantages and disadvantages. Colour and depth data contain contextual information but are both dependent on the camera view and the person's appearance. Depth and skeleton data are more robust than colour data when there are occlusions or a lot of illumination changes and can even work in total darkness. Skeleton data is both invariant to the camera location and subject appearance, but lacks contextual information and does not work well when the player is not standing or sitting upright.



Figure 1-5 Microsoft Kinect (front) with depth (back left), skeleton (back middle) and RGB (back right) images [8]

Until recently, action recognition research has focused on increasing accuracy on datasets in highly controlled environments. These datasets normally contained a single person that was instructed to perform a single action clearly performed (see Figure 1-6). Recognition was performed offline using pre-segmented action sequences containing a single action and information from all the frames to classify the action. The action was recognised after its completion and the computation time was unrestricted. These simplifications resulted in over inflated accuracy and action recognition algorithms not suitable for real-world applications. A recent survey [9] showed perfect or near perfect offline action recognition accuracy on simple datasets with a small number of actions.

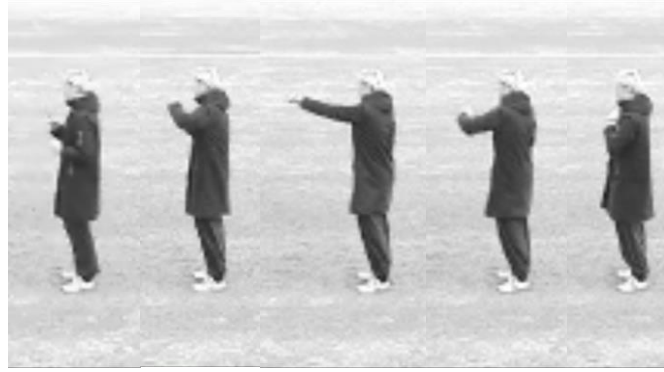


Figure 1-6 Simple boxing sequence with a single person performing a punch (KTH) [10]

Recent research has pursued the more complex challenge of online action recognition that processes a continuous stream of actions in real-time (as shown in Figure 1-7). Online recognition systems need to run in real-time however the latency of the recognition can vary depending on the application. For example, a sign language recognition system may delay recognition until a sequence of words has been parsed [11]. Such systems can benefit from increased accuracy by delaying the recognition. However, many applications with a human-machine interface in a range of domains including home entertainment, healthcare, sports, and robotics do not have this option, as they require low latency since the action should be detected before it is completed. Consider a volleyball game where a player is about to return the ball to the opposing team, it is important to detect the point when the player would hit the ball and update the trajectory of the ball before the action finishes.

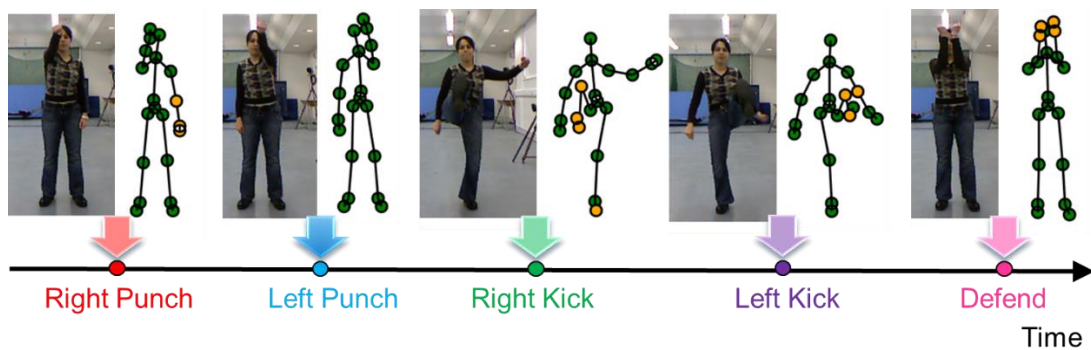


Figure 1-7 A continuous stream of different actions from the G3D dataset.

Online recognition systems for different applications may have very different requirements in terms of latency and research has highlighted a trade-off between accuracy and latency [12], [13]. High accuracy and low latency is critical for interactive games to be responsive to the users' actions. Accuracy of action recognition may be affected by four main sources of intra-class variations: viewpoint, anthropometry, execution rate and personal style (as shown in Figure 1-8 and Figure 1-9). The introduction of skeleton data has reduced the viewpoint and anthropometric variations as variances arising from gender, clothing and hair styles have been removed. Therefore, in this thesis the focus is on addressing execution rates and personal style. Execution rate variation is due to temporal differences arising from the range of speeds that the same human movements can be performed. Personal style can also affect the performance as different people may perform the same action differently [14].



Figure 1-8 Viewpoint differences: changes in camera position relative to the subject

Figure 1-9 Anthropometric variations: differences in size and proportions of the human body

1.1 Problem Summary

Perfect or near perfect offline action recognition accuracy on existing datasets has been achieved. These datasets normally contain a single person that was instructed to perform a single action clearly which over-simplifies the task of action recognition.

Many types of machine learning algorithms have been applied to action recognition but the majority of approaches have been applied offline and even the online approaches have high latency. The latest online action recognition algorithms are less accurate than the offline approaches due to the increased difficulty of the task.

Evaluation of action recognition algorithms is typically done in isolation, focusing historically on high accuracy and more recently also on low latency. However, in reality most actions form part of an interaction where the duration of the action is important.

1.2 Aims and objectives

The aim of this thesis is to deal with the problem of complex action recognition in real world scenarios with multiple subjects. To reach the final goal the problem is decomposed into a series of simpler tasks which culminate with the most complex task in the last technical chapter. A simple action is defined as an action that is performed by a single subject in a controlled environment according to a set of instructions. Whereas, complex actions form interactions with multiple subjects in real-world scenarios. The first task is online action recognition for a single player performing simple actions. The second task is early action detection, online action recognition and action prediction for a single player performing simple actions. The final task is online action recognition for multiple players performing compound actions.

The objectives needed to fulfil this aim are:

- Produce public datasets of a range of simple and complex gaming actions.
- Design and develop novel algorithms for online action recognition for single and interaction for multiple subjects.
- Develop an evaluation framework for complex action recognition.

1.3 Contributions

The first contribution is the capture and release of two new multi-modal gaming datasets, G3D and G3Di for real-time action recognition. G3D is the first public gaming action dataset to contain multiple actions and multi-modal data (introduced in section 3.4). In contrast to existing datasets where the interactions are scripted, G3Di was captured using a novel game-sourcing method so the actions are more realistic and more complex (introduced in section 5.4).

The second contribution is a novel online action recognition algorithm, Dynamic Feature Selection (DFS), proposed in chapter 3. DFS combines the discriminative power of Random Forests for feature selection with an ensemble of AdaBoost classifiers for dynamic classification.

The third contribution is a novel online action recognition algorithm developed by modelling the dynamics of human actions with Clustered Spatio-Temporal Manifolds (CSTM), proposed in chapter 4. The core of the algorithm creates style invariant action templates that when combined with Dynamic Time Warping (DTW) provides execution rate invariance to achieve state-of-the-art results for online action recognition and enables early recognition and prediction from a continuous stream.

The fourth contribution is a Hierarchical Transfer Learning framework (HTL), proposed in chapter 5. The HTL framework is comprised of a novel transfer learning algorithm for

compound actions in addition to hierarchical interaction detection. Specifically, transfer learning is employed to allow the tasks of action segmentation and modelling to be performed on a related but simpler dataset, combined with adaptation of body part models to improve online action recognition performance on a more complex dataset.

1.4 Structure of the thesis

In chapter 2, an overview of related work on action recognition and background information that is important in the context of this thesis is presented. First, the sensors for capturing human motion using computer vision techniques are introduced and the modalities compared. Then, the relevant state-of-the-art real-time pose estimation algorithms and pose-based features are investigated. Subsequently, popular machine learning algorithms that have been used in academia and commercial game titles for action recognition are presented. Finally, the datasets and evaluation metrics that are used to validate the contributions to the thesis are reviewed against their applicability to real world scenarios.

Chapter 3 introduces dimensionality reduction techniques that can improve computation time and accuracy of action recognition algorithms focusing on feature selection approaches. A novel online action recognition algorithm, Dynamic Feature Selection (DFS) is proposed and a new gaming action dataset, G3D is presented. Two online action recognition algorithms with low latency, AdaBoost and Random Forest, are used as a baseline for the new gaming action dataset, G3D. Additionally, the MSRC-12 dataset is used to show that the proposed method achieves results comparable to state-of-the-art algorithms.

Different dimensionality techniques are introduced in chapter 4, focusing on feature transformation approaches that maintain the temporal dynamics of human actions. Four distinctive approaches for action recognition are discussed: offline, online, early and

prediction and related works in each of these areas contrasted. A novel algorithm, Clustered Spatio-Temporal Manifolds (CSTM), is proposed which achieves state-of-the-art results for online action recognition and enables early recognition and prediction in a continuous stream.

Chapter 5 presents the challenges of multi-player gaming which include compound actions and describes how the action duration becomes important to detect virtual interactions. A novel Hierarchical Transfer Learning (HTL) framework is proposed for online action recognition of compound actions and interactions. To test the proposed framework in a realistic context a new complex multi-player gaming dataset G3Di is presented using a novel game-sourcing approach so the actions captured are more realistic and challenging in comparison to scripted actions. Experimental results indicate higher complexity of the new dataset in comparison to the existing gaming datasets, highlighting the importance of this dataset for designing algorithms suitable for real-world applications.

Finally, in chapter 6, conclusions and future work are presented.

CHAPTER 2

BACKGROUND AND RELATED WORK

An overview of related work on action recognition and background information that is important in the context of this thesis is presented in this chapter. First, the depth sensors for capturing human motion using computer vision techniques are introduced and the different modalities compared. Then the state-of-the-art real-time pose estimation algorithms and pose-based features that have made this research possible are investigated. After popular machine learning algorithms that have been used in academia and commercial game titles as well as other areas for action recognition are presented. Finally, the datasets and evaluation metrics that are used to validate the contributions to the thesis are reviewed against their applicability to real world scenarios.

2.1 Depth Sensors

The Kinect [5] originally developed for the Xbox 360 games console initiated a new generation of games where the human body was the controller. Due to its low cost and innovative depth sensing technology, the Xbox Kinect become the fastest selling gaming peripheral [15]. Subsequently, different hardware versions of this depth sensor have been developed for the PC (Kinect v1 [16] and v2 for Windows [17] and the Xtion PRO Live [18]) and tablets (Structure Sensor [19]) enabling the rapid development of a wide range of applications for both industry and academia.

The technical specifications for the most popular depth sensors, most of which are accompanied by a colour camera are shown in Table 2-1. The trend has been for an increase in camera resolution over time and miniaturisation of the sensor. The other key

difference between the sensors is the software support available in terms of operating system support, drivers and development libraries. The Kinect has the most software support for the Windows Operating System (OS) whereas the Structure Sensor has more support for the Mac OS and the Xtion PRO Live has the same support for all the major operating systems. At the time of recording the datasets in this thesis the Kinect for Windows v1 and Xtion (Live) were the only available sensors. The devices have a similar specification (see Table 2-1) but as the former has better support for developers in terms of libraries and documentation it was selected. The Kinect for Windows v1, contains both infrared (IR) and colour sensors (as shown in Figure 2-1) to provide depth and colour data at 30 frames per second (fps).

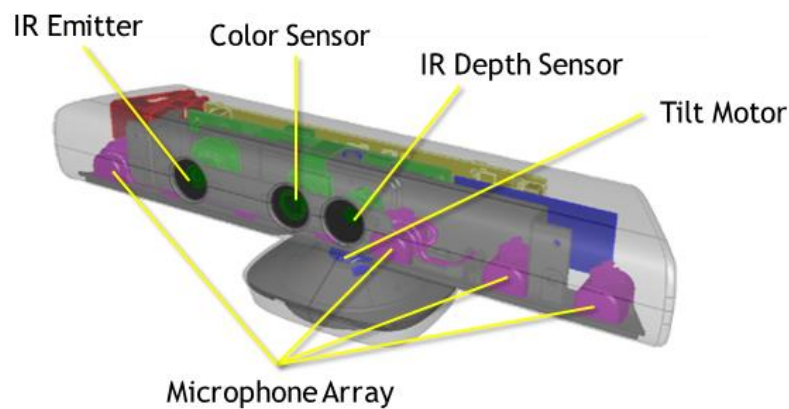


Figure 2-1 Inside the Kinect: Depth and Colour Sensors [16]

Depth sensors are adversely affected by sunlight as it uses IR technology so it is most suited to indoor applications, although it should work well outside at night. They have a limited field of view (FOV) which may cause problems for surveillance or robotics applications which can be addressed by using multiple sensors. If the FOV from the sensors overlap then interference in the IR patterns occurs introducing noise into the depth images. This interference can be reduced by additional hardware [20] or eliminated by placing the sensors in a pattern where their FOVs do not overlap. However, the latter case requires additional techniques to calibrate the sensors [21].

Table 2-1 Depth sensor specifications

	Kinect for Windows v1 / Kinect for Xbox 360 [16]	Kinect for Windows v2 / Kinect for Xbox One [17]	Xtion (Live) [18]	Structure Sensor [19]
Date of release	2011	2014	2011	2013
Dimensions	30.5cm x 7.6cm x 6.4cm	24.9cm x 6.6cm x 6.7cm	17.8cm x 5.1cm x 3.8cm	11.9cm x 2.8cm x 2.9cm
Framerate	30fps	30fps	30 / 60fps	30 / 60fps
Colour Camera	640 x 480	1920 x 1080	(1280 x 1024) ¹	NA
Depth Camera	320 x 240	512 x 424	640 x 480 / 320 x 240	640 x 480 / 320 x 240
Max Depth Distance	4m	8m	3.5m	3.5m
Min Depth Distance	40cm / 80cm	50cm	80cm	40cm
Horizontal FOV	57 °	70 °	58 °	58 °
Vertical FOV	43 °	60 °	45 °	45 °
Number of tracked skeletons	2	6	4	Up to 15

¹ The Xtion Live has a colour camera whereas the Xtion does not have a colour camera, both are available as a PRO version for developers which are supplied with additional software.

2.2 Overview of action recognition

There is a vast wealth of research on human activity recognition in computer vision. Human activities can be conceptually subdivided depending on the level of complexity of the activity. The four levels defined by Aggarwal and Ryoo [22] are gestures, actions, interactions and group activities (see Figure 2-2).

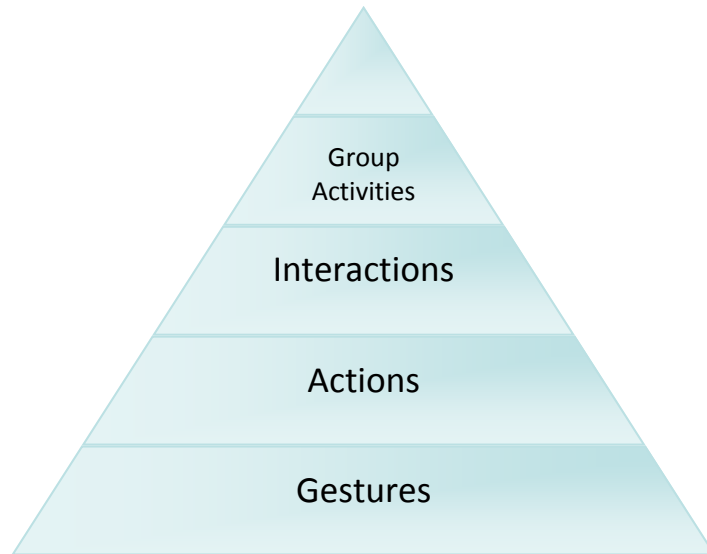


Figure 2-2 Levels of human activities

Gestures are elementary movements of a person's body part e.g. "stretching an arm". Actions are single person activities, such as "punching" and "kicking" that may be composed of multiple gestures. For example, a punch is an action that comprises of the gestures "raising a hand" and "stretching an arm" [22]. Interactions involve two or more persons and / or objects. For example "two people fighting" is a human interaction and "a person picking up a gun" is a human-object interaction. Finally, group activities are groups of persons performing an activity such as "two groups fighting".

The focus in this chapter is on action recognition as both chapter 3 and 4 cover single player games and interaction recognition is introduced in chapter 5 when multiple players are introduced. In this thesis, the peak of an action is a key concept, which is defined as the moment when the goal of the action is satisfied. For example, in a boxing game the aim of punching is to hit the opponent which is fulfilled when the arm is maximally extended. The poses in the dataset that fulfil the action goal are manually labelled as peak poses with one peak pose labelled for each action instance. Examples of peak poses for different actions are illustrated in Figure 2-3.

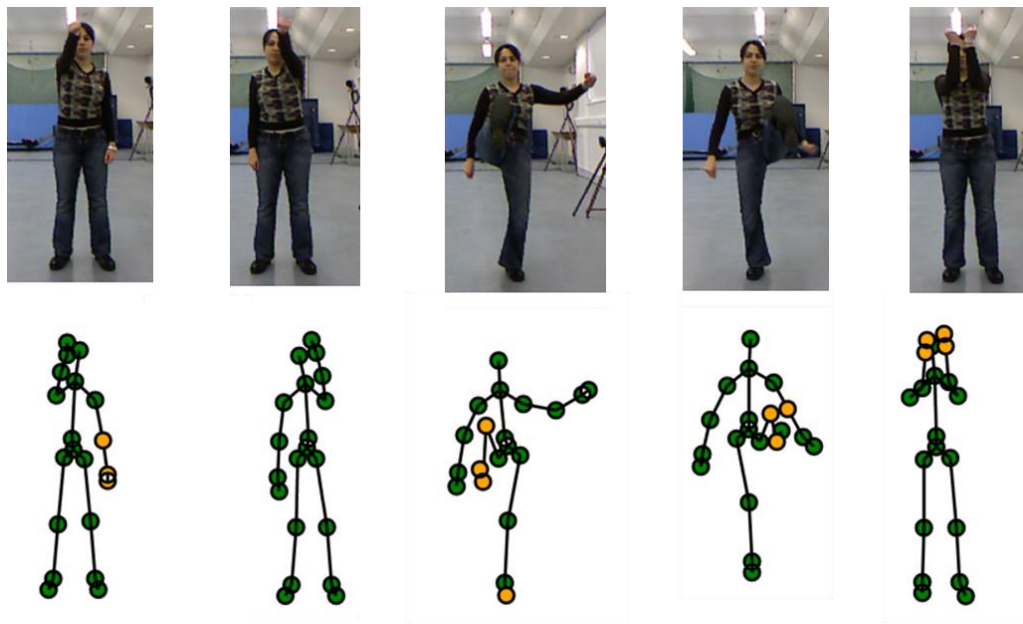


Figure 2-3 Peak poses for different actions, from left to right: right punch, left punch, right kick, left kick and defend

Action recognition in commercial games ranges from heuristic based techniques to machine learning algorithms. The approach taken depends on the number and complexity of gestures to be performed in the game. For example, a bowling game only requires a few simple gestures and an algorithm can be hardcoded for each gesture. However, this approach may not work well for a greater number of complex gestures where machine

learning algorithms are more suitable. Various machine learning techniques can be applied to more complex games, for example AdaBoost with a boxing game and exemplar matching with a tennis game [23]. The benefit of machine learning algorithms is that they can be trained to recognise a wide range of actions including sporting, driving and action-adventure actions such as walking, running, jumping, dropping, firing, changing weapon, throwing and defending. This approach can increase the complexity and appeal of games that can be developed to include popular genres like action-adventure games.

Machine learning algorithms consist of two key phases: the training phase and the testing phase as summarised in Figure 2-4. There are different approaches to the training phase but they begin with the training data which is processed to obtain features and then used with the ground truth action labels to train a learning algorithm. The same pre-processing step is used in the testing phase and then the testing data is classified using the trained models.

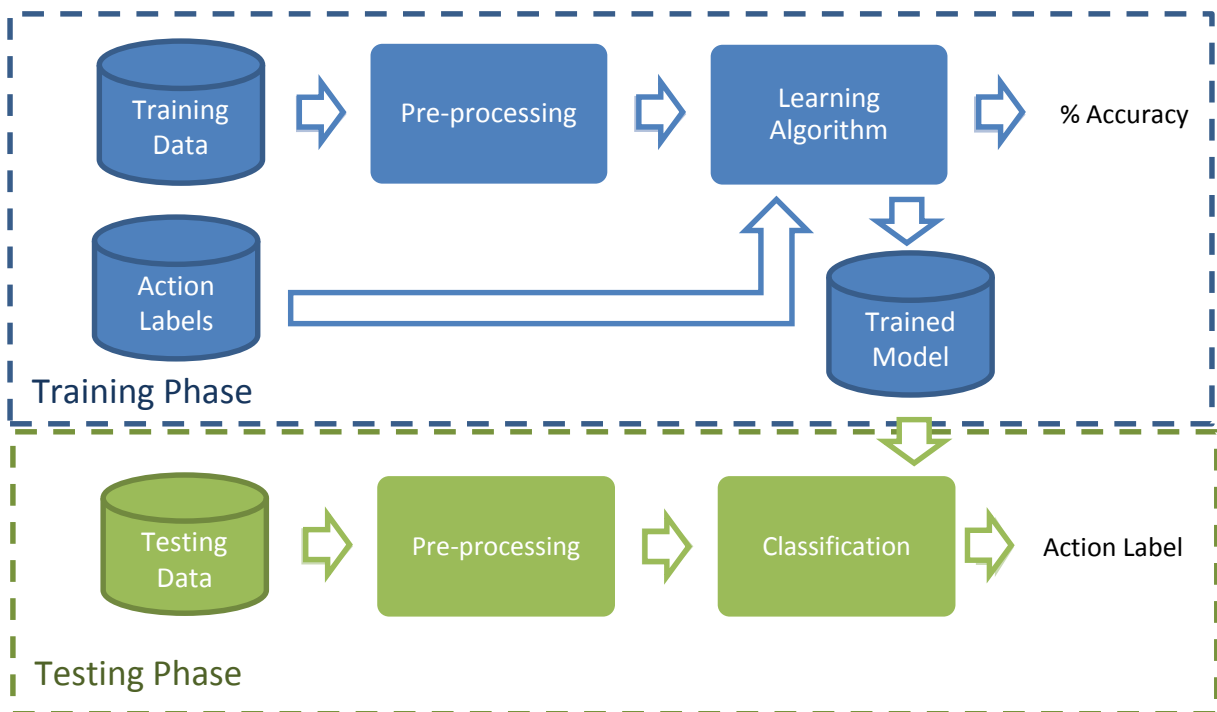


Figure 2-4 Action recognition generic pipeline

2.3 Feature Extraction

In general, features can be defined as abstractions of the sensor data. The purpose of pre-processing is to find the main characteristics of the data that accurately represent the original data and discriminate between different classes. There is a vast wealth of research on features for human action recognition from video (for a comprehensive review see Aggarwal and Ryoo [22]) and 3D data (for a comprehensive review see Aggarwal and Xia [24]). Historically, the majority of the algorithms in action recognition were appearance-based as low level features can easily be extracted from video sequences. Due to recent technological developments in depth camera technology it is now possible and economical to capture real-time sequences of depth images. This has resulted in many depth and pose-based approaches being developed.

2.3.1 FEATURES FROM VIDEO

Appearance based features have several advantages as they require little high-level processing and can avoid the difficulties of pose estimation. They are also not restricted to the human body so can encode contextual information such as background. The context of the environment can be used to further improve accuracy as intuitively certain actions will only happen in specific scenes. For example, performing a golf swing in a real golf game would require a golf club and will occur outdoors, probably on a green field. However, performing a golf swing in a Kinect game the user has no golf club and is performing the action indoors. The restricted environment associated with gaming, typically the user's lounge, poses the challenge of missing context. The normal scene and objects usually associated with a given action are missing. This lack of contextual information in a gaming scenario may mean that appearance-based action recognition approaches may under-perform.

Bobick and Davis [25] represented each action as a space-time template consisting of two images: motion-energy image (MEI) and motion history image (MHI). The two images were constructed from a sequence of foreground images which are weighted 2D (x, y) projections of the original 3D (x, y, t) data, as illustrated in Figure 2-5. Due to its compact 2D representation action recognition can be performed in real-time. This approach has been extended to modelling actions as 3D space-time volumes [26], [27] but the major disadvantage of both the 2D and 3D approaches is the difficulty in recognising actions when multiple people are in the scene, due to occlusions if the actions overlap spatially.



Figure 2-5 Actions along with their Motion History Images, from left to right: sit-down, sit-down MHI, arms-wave and arms-wave MHI [25]

Yao et al. [7], [28] used person detection to track and segment the person of interest from a video sequence and then extracted low level features from the action track including colour, optical flow, spatial and temporal gradients, as illustrated in Figure 2-6 [7]. Segmentation and tracking of people in videos can be difficult due to poor lighting and occlusions. To avoid this low level appearance features such as Gabor filter responses [29] and optical flow [30] have been applied globally.



Figure 2-6 Appearance-based features, from left to right: colour, dense optical flow, spatial gradients and temporal gradients [7]

Similarly, spatio-temporal interest points have become popular as they do not depend on segmentation and tracking. Spatio-temporal interest points are widely used in object and scene recognition and have been extended for action recognition to incorporate the temporal information present in videos. Many different feature detectors (cuboids [31], 3D Harris Corners [32], 3D Hessians [33] and 3D salient points [34]) and descriptors (HOG/HOF [35], HOG3D [36], extended SURF [37]) have been proposed. The feature detectors are used to find distinctive key points in the video and the surrounding region of each key point is used to compute a local descriptor.

The bag-of-words approach is commonly used for natural language processing and was adapted for computer vision tasks by introducing the concept of visual words. The bag of visual words approach is very popular for object detection and action recognition [31], [35], [59]. The typical bag-of-words approach for action recognition starts by selecting spatio-temporal interest points and extracting low level descriptors around these points. These feature descriptors are then sampled and clustered to make the video words which form the codebook. The features in the training data are assigned to histograms of video word occurrences for the entire video sequence. The limitation of the bag-of-words approach is that it does not model the spatio-temporal distribution of features so it would not be able to differentiate between actions with similar motions but occurred in different order.

Spatio-temporal interest points are generally scale and translation invariant and work well with background clutter and multiple people in the scene. However, they are computationally intensive especially if using dense sampling which gives the best accuracy [38]. Moreover, these appearance-based features may be unreliable in a gaming environment due to background clutter and possible lack of illumination.

2.3.2 FEATURES FROM DEPTH MAPS

Various features have been proposed that are based directly on depth images and they can be split into four categories: 3D silhouettes, local spatio-temporal, local occupancy and 3D scene flow.

2.3.2.1 3D Silhouettes

Early attempts on action recognition showed that silhouettes, or extremities of silhouettes (e.g. head, hands and feet) carry important body shape information [25], [39]. In a depth image it is easier to extract the silhouette of a person compared to colour images, especially when there is background clutter and bad lighting conditions. In addition, the depth image contains additional body shape information across the camera plane which enables them to model more than just parallel motions. Several features have been proposed to recognise actions based on 3D silhouettes which either project the 3D data to 2D planes [40], [41] or temporally stack the 3D data.

Li et al. [40] sample a small set of 3D representative points from the contours of the planar projection of the silhouette (see Figure 2-7). Their results show recognition errors were halved when using 3D depth data in comparison with 2D silhouettes from colour images. Similarly, Jalal et al. [41] use a Radon transform to project 3D silhouettes along specified view angles before a further projection to 1D. A significant increase in accuracy over conventional binary silhouettes was achieved.

An alternative set of approaches stack the 3D silhouettes or energy along the temporal domain [42], [43] extending the original MHI/MEI (as described in section 2.3.1) to achieve superior accuracy.

The loss of information when computing the projections [40], [41] or temporal stacking [42], [43] limits 3D silhouette features to recognising simple atomic actions of single

people. Additionally, as shape information is only present on the side of the body facing the camera 3D silhouette features are inherently view dependent.

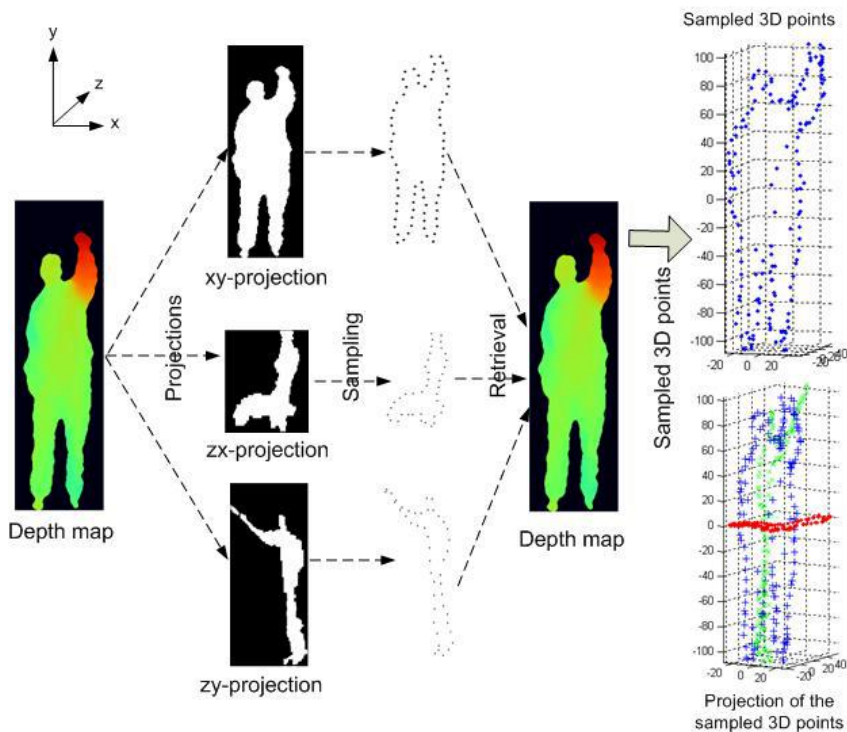


Figure 2-7 Left: depth map and right: sampled 3D points [19]

2.3.2.2 Local Depth Spatio-Temporal Interest Points

Local spatio-temporal interest points (STIP) are very popular for action recognition from video (as discussed in section 2.3.1) and their success has encouraged the exploration of spatio-temporal features from depth data. An early attempt by Ni et al. [43] used the depth as an auxiliary channel to partition the colour space into layers and the Harris3D [32] detector and HOG/HOF [35] descriptors were applied in the traditional manner to the colour channel. This was shortly followed by several applications of the Harris3D detector to extract the STIPs directly from the depth videos [44], [45]. Cheng et al. [44] extracted

local features from both colour and depth and showed that depth features increased performance by more than 10% and a fusion of both modalities showed the best performance. Harris3D was originally designed for RGB data which is less noisy than depth data and does not contain missing values. To overcome this Xia and Aggarwal [46] proposed STIPs designed specifically for depth with noise suppression functions which outperformed the Harris3D detector.

Local depth spatio-temporal features are invariant to shifts and scales and naturally deal with occlusions and multiple people. However, as the cuboids are extracted from the (x, y, t) volume these features are view dependent. Secondly, the existing implementations require the whole video and the feature computation algorithm is computationally expensive limiting its real-time application.

2.3.2.3 Local Occupancy

Local occupancy features are the representation of the points that the sensor captured from the real world. They were proposed in 3D (x, y, z) [47] and 4D space (x, y, z, t) [48], [49] and the latter are similar to local spatio-temporal features in that they describe local appearance in the space and time domain. In a local occupancy pattern, an occupied location will have a value of 1 and others 0. Therefore, in contrast to spatio-temporal features the local occupancy is quite sparse as the majority of its elements are zero. Another difference is that spatio-temporal features contain background information while local occupancy features only contain information around a specific point, which could be beneficial in a gaming scenario.

2.3.2.4 3D Scene Flow

Optical flow is a feature for action recognition from video and there are promising works on 3D scene flow from stereoscopic data [50], [51] but these algorithms have a high

computation cost. Depth cameras enable simpler and faster methods to get optical flow in 3D (x, y, z) [52]–[55]. However, the research on 3D optical flow is still in its preliminary stage and computing the 3D scene flow in real time is still a challenging task.

2.3.3 SKELETON FEATURES

Skeletal data obtained by motion capture (mo-cap) systems has been widely used for film making and video game creation. However, these systems are often deployed in commercial studios or research labs as they require expensive specialist equipment. Recent progression in estimating the human skeleton from low cost depth cameras in real time has enabled skeletal data to be captured in peoples' homes Both systems provide the spatial coordinates of joint positions in three dimensions. Spatial coordinates should not be used directly for action recognition as there are significant spatio-temporal variants between logically related motions [56]. However, various skeleton features can be obtained from the joint positions.

2.3.3.1 Skeleton Data from Motion Capture

Motion capture systems require special optical markers to be placed on the human body, multiple RGB cameras to be positioned around the subject as well as specialist capture and tracking software, which must be calibrated before each session as illustrated in Figure 2-8. This enables high quality robust capture of joint position and rotation information of complex actions. Mo-cap systems have been widely used in commercial studios for film making and video game creation. Due to the specialist equipment and setup procedure traditional mo-cap systems are not suited to home use. However, body suits comprised of inertial sensors, as shown in Figure 2-9 are currently under development to enable virtual reality (VR) experiences and mo-cap at home.



Figure 2-8 Overview of a commercial motion capture system. Multiple cameras are suspended from a rig in the ceiling, the subject has many markers on her body and the specialist software used for capturing and tracking the markers is shown.



Figure 2-9 Overview of a body suit, currently under development designed for virtual reality or mo-cap at home. The body suit comprises of 17 inertial sensors and 2 hand controllers, to control the game character in conjunction with the Oculus rift to display the game to the player in 3D.

2.3.3.2 Skeleton Data from Depth Maps

The real-time pose recognition algorithm proposed by Shotton et al. [6] accurately determines 3D positions of body images from a single depth image (as shown in Figure 2-10 left). Their approach is based on an object recognition approach and uses no temporal information. They proposed an intermediate body part representation (as shown in Figure 2-10 middle) that simplifies the problem to a per-pixel classification problem. They employed a simple depth comparison feature. At a given pixel m , the features compute:

$$f^\theta(I, m, \mathbf{u}, \mathbf{v}) = d_I \left(m + \frac{\mathbf{u}}{d_I(m)} \right) - d_I \left(m + \frac{\mathbf{v}}{d_I(m)} \right) \quad (2-1)$$

where $d_I(m)$ is the depth of pixel m in image I , and \mathbf{u}, \mathbf{v} are offsets. The normalisation of the offsets ensures the features are depth invariant. Finally, local modes are used to generate confidence scores of 3D proposals of body joints (as shown in Figure 2-10 right). A large and highly varied training set, hundreds of thousands of training images, allowed their random forest classifier to estimate body parts invariant to anthropometric differences. The algorithm is fast and runs at 200fps enabling pose-based features to be obtained in real-time.



Figure 2-10 Overview of real-time pose recognition. From a single input depth image (left), a per-pixel body part distribution is inferred (middle), local modes are estimated to give proposals for the 3D locations of body joints (right) [6]

2.3.3.3 Extracting Skeleton Features

Given joint information either from mo-cap systems or depth cameras a range of pose based features can be extracted. The simplest feature to extract from the joint positions are the position difference features, defined as the difference between pairs of joints [7], [23], [57], [58]. This formula has been applied to different joints in a single frame to obtain a distance feature and to a specific joint between two different frames to determine the joint velocity.

Joint angle features are more robust than joint distances as they are invariant to scale and anthropometric differences. If the joint orientation is computed relative to the world coordinates or the torso then the joint angle feature is also rotation invariant [33], [58]. In both cases joint angles were represented by quaternions as this overcomes the difficulties of gimbal lock suffered by other representations such as Euler angles. Joint angles can be measured in a single frame to determine orientation [23], [33] and also between two different frames to measure the angular velocity. The angular velocity has been applied to sequential frames [23], [57], [58] and it has also been used to create an offset feature by applying it to the first frame and the current frame [57]. The latter assumes that the first frame is the neutral pose which may not be the case in a gaming scenario.

Müller et al. [56] introduced a set of qualitative geometric features to express geometric relations between certain body points of a pose. Examples include if hand is above neck height or not (see Figure 2-11 left), if two hands are touching (see Figure 2-11 middle) and whether a leg is bent or straight (see Figure 2-11 right). Müller et al. [56] also defined geometric non-Boolean features such as the absolute speed of certain joints and the relative speed of certain joints with respect to other joints. These Boolean features are very robust to spatial variations and although they were initially designed for the indexing and retrieval of motion capture data, they showed promising results for action recognition

as features for a Hough Forest [7] but a large number of random tests was needed to optimise the binary test for each tree node.

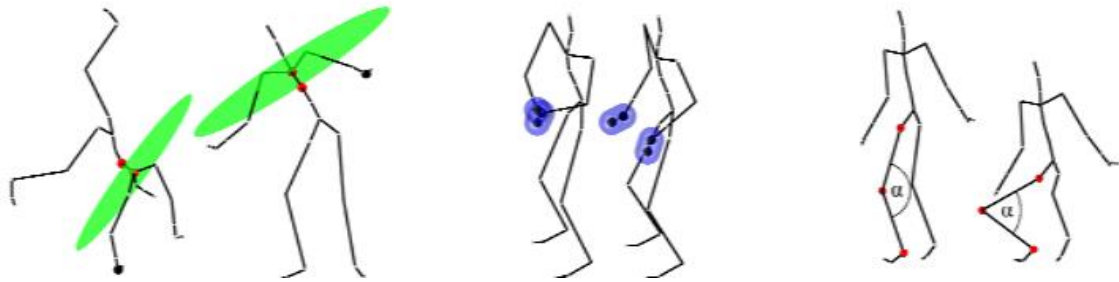


Figure 2-11 Qualitative features describing geometric relations between the body points of a pose that are indicated by red and black markers (image adapted from [56])

Yao et al. [7] posed the question “Does Human Action Recognition Benefit from Pose Estimation?” Their experiments compared appearance based, pose-based and a combined approach in a home monitoring scenario using the same classifier and same dataset. The appearance based features used were colour, dense optical flow and spatio-temporal gradients. The pose-based features were qualitative geometric features [56]. Yao et al. [7] results showed that the optimum approach was pose-based. This significantly outperformed the appearance based approach and was even slightly better than the combined approach.

2.3.4 SKELETON FEATURES USED IN THIS THESIS

Pose-based features are invariant to subject appearance and have outperformed appearance based features so will be used in this thesis. The specific pose-based features used in this thesis are: position difference, position velocity, position velocity magnitude, angle velocity and joint angles as illustrated in Figure 2-12. These pose-based features were selected as they are invariant to the camera location [23].

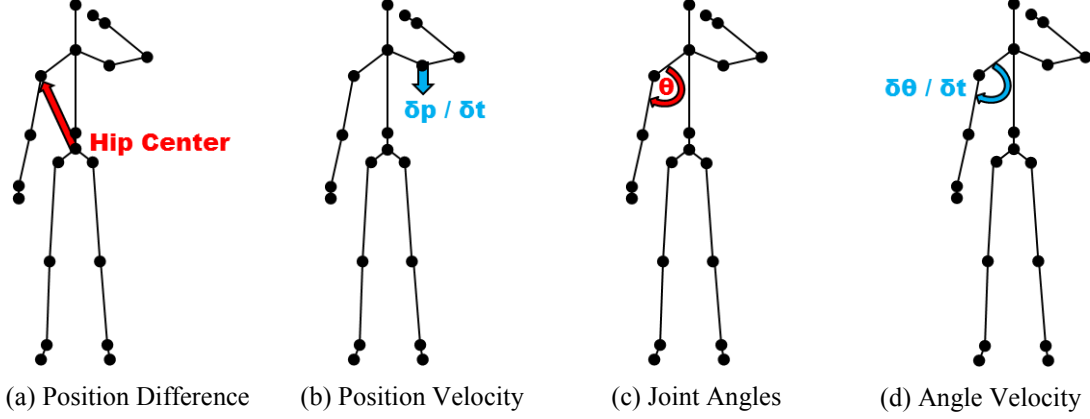


Figure 2-12 Pose-based features used in this thesis

Let $p_{j_i,t} \in \mathbb{R}^3$ be the 3D location $(x_{j_i,t}^c, y_{j_i,t}^c, z_{j_i,t}^c)$ of joint j_i at time t . The position difference features are defined as the difference between each joint $(j_1 - j_{n_j})$ and the hip centre j_0 in a single pose, where n_j is the number of joints in a pose.

$$f^{pdx}(j_i, j_0; t_1) = x_{j_i,t_1}^c - x_{j_0,t_1}^c \quad (2-2)$$

$$f^{pdy}(j_i, j_0; t_1) = y_{j_i,t_1}^c - y_{j_0,t_1}^c \quad (2-3)$$

$$f^{pdz}(j_i, j_0; t_1) = z_{j_i,t_1}^c - z_{j_0,t_1}^c \quad (2-4)$$

The position velocity features encode the difference over time of a single joint, where $t_1 \neq t_2$.

$$f^{pvx}(j_i; t_1, t_2) = \frac{x_{j_i,t_1}^c - x_{j_i,t_2}^c}{t_1 - t_2} \quad (2-5)$$

$$f^{pvy}(j_i; t_1, t_2) = \frac{y_{j_i,t_1}^c - y_{j_i,t_2}^c}{t_1 - t_2} \quad (2-6)$$

$$f^{pvz}(j_i; t_1, t_2) = \frac{z_{j_i,t_1}^c - z_{j_i,t_2}^c}{t_1 - t_2} \quad (2-7)$$

The position velocity magnitude feature is defined as the Euclidean distance between a single joint separated by time, where $t_1 \neq t_2$.

$$f^{pvd}(j_i; t_1, t_2) = \frac{\|p_{j_i, t_1} - p_{j_i, t_2}\|}{t_2 - t_1} \quad (2-8)$$

The joint angle features are defined as the quaternions of the angle between three connected joints in a single pose e.g. right wrist, wright elbow and right shoulder. The quaternions $f^q \in \mathbb{C}^4$ were built in the standard polar (axis-angle) form:

$$f^q = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)(in_x + jn_y + kn_z) \quad (2-9)$$

where n is the (unit length) axis of rotation, θ is the angle, and i, j and k are the imaginary basis vectors.

The angle velocity features $f^{qd} \in \mathbb{C}^4$ are defined as the change in the quaternions of the angle over time, where $t_1 \neq t_2$.

$$f^{qd}(t_1, t_2) = \frac{f^q(t_1) - f^q(t_2)}{t_1 - t_2} \quad (2-10)$$

Human actions are a high dimensional and complex phenomenon, which are extremely difficult to model by a machine due to variations in viewpoint, anthropometry, execution rate and personal style. The introduction of pose-based features has reduced the viewpoint and anthropometric variations, as variances arising from gender, clothing and hair styles. Therefore, in this thesis the focus of the learning algorithm to address execution rates and personal style.

2.4 Classification

There are many types of machine learning algorithms that have been applied to action recognition, including nearest neighbour, kernel machines, exemplar matching, state models, random forests and boosting approaches. The majority of approaches have been applied to offline action recognition so these are reviewed first, evaluating their ability for invariance to execution rate and personal style (see section 2.4.1). Then the more recent online action recognition methods are analysed focusing on latency (see section 2.4.2). Finally, the online action recognition approaches with low latency are described in more detail as they will be used in the comparative experiments in this thesis (see section 2.4.3).

2.4.1 OFFLINE ACTION RECOGNITION

2.4.1.1 Nearest Neighbour

Nearest Neighbour is a simple approach which classifies objects based on the closest training examples in the feature space. The nearest neighbour approach assigns a sample based on a majority vote among the classes of the nearest training samples. The Euclidean distance is a common distance metric but suffers the curse of dimensionality for high dimensional data. Dimensionality reduction can be used on the feature set prior to classification to overcome this problem. An alternative approach is to use a bag-of-words (as described in section 2.3.1.) to represent videos as sets of video words and classify the histograms using nearest neighbour. More complex classifiers such as Support Vector Machines (SVMs) have shown better accuracy than the simple nearest neighbour algorithm [10].

2.4.1.2 Kernel Machines

Support Vector Machines are a state-of-the-art classifier widely used for pattern recognition in many domains especially natural language processing and bioinformatics. A basic SVM performs linear classification but when combined with a kernel can solve non-linear problems [60]. Schuldt et al [10] used a non-linear SVM to classify simple cyclical actions such as jogging and hand waving by extracting spatio-temporal interest points in video. Similarly, Laptev et al. [35] used a non-linear SVM for recognition of natural human actions such as answer phone, get out of car, sit down and stand up. The benefits of SVMs are they are robust and accurate and only require a small amount of data for training.

2.4.1.3 Exemplar Matching

Exemplar matching approaches use training examples directly to create a representative template sequence or set of sample sequences of each action. The sequence of feature vectors from a new sequence can be compared with template sequences for the best match. Dynamic time warping (DTW) originally developed for speech processing can be used to allow for variations in the speed the actions are performed (see Figure 2-13) and achieve execution rate invariance. The problem is that using the training examples directly is computationally and memory intensive especially if using the DTW algorithm [22], [33].

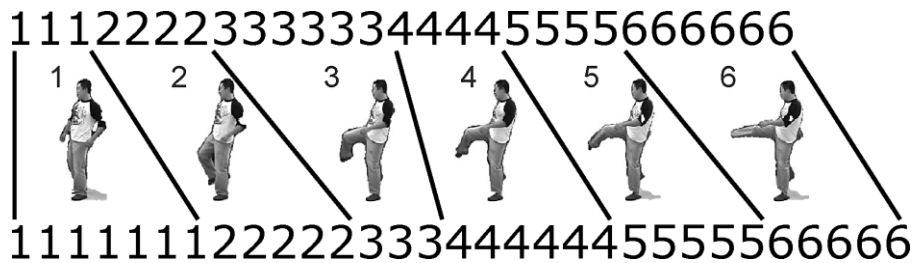


Figure 2-13 Exemplar matching between two kicking sequences with different non-linear execution rates. Each number represents a particular pose of the subject. [22]

Different optimisation techniques have been proposed that involve the removal of redundant poses. Clustering can be applied at the pose or sequence level to reduce the size or number of templates. Key poses [61] are a representative subset of the template poses selected by clustering techniques. Chaaoui et al. [61] clustered poses in the high dimensional space and matched pose sequences with DTW. Gavrila and Davis [62] applied the clustering at the sequence level to maintain the temporal history within the action templates. Similarly, Veeraraghavan et al. [63] learnt an average sequence from the samples of each class of action and a function space capturing the permissible action specific time warping transformations. The removal of redundant poses reduces the computational cost of template matching and can also improve classification accuracy. Combining template matching with DTW achieves execution rate invariance but the existing approaches [61]–[63] match the entire action template with pre-segmented sequences so observational latency is high and recognition is offline.

2.4.1.4 State Models

Hidden Markov Models (HMM) [64] are generative state models with success in speech recognition and broad applicability to time series tasks. Yamato et al's [65] were the first to use HMMs for action recognition to reliably recognise various types of tennis play. Yu and Aggrawal [39] used an HMM for the recognition of a person climbing a fence. The benefit of state-based approaches is their ability to quantify the probability of an action. The limitation is that most HMM-based recognition approaches [39], [65]–[67] require temporal segmentation of the action instances and the entire test sequence must be observed before the labels of any time step can be generated which restricts recognition to offline settings.

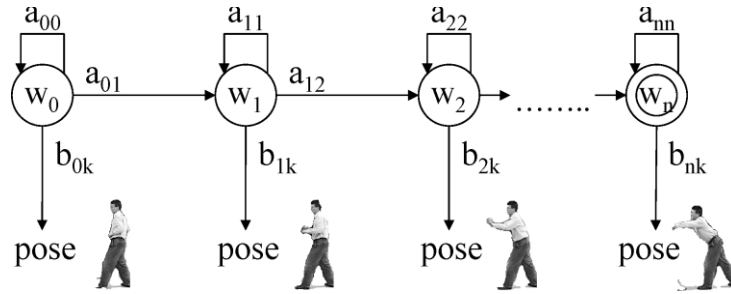


Figure 2-14 An example hidden Markov model for the action stretching an arm [22].

2.4.2 ONLINE ACTION RECOGNITION

Most of the existing action recognition algorithms are far from operating online and with low latency. Low latency is required to make action recognition methods applicable for a range of real-time applications including gaming, surveillance systems, human-computer interfaces, intelligent robots and autonomous vehicles. Latency is dependent on two separate factors which have been identified as observational latency and computational latency [12]. Observational latency is the time it takes the system to observe enough frames to make a decision, whereas computational latency is the actual time to perform the computation on a frame. Ellis et al. [12] measured observational latency from a rest state which is not possible with multiple actions as the subjects may not return to the rest state between actions. Therefore, in this thesis observation latency is defined as the time after the peak of the action at which the action is detected which at any rate is a more suitable measurement for evaluating latency for natural user interface (NUI) applications.

Both observational and computational latency should be considered to ensure that the developed algorithms are suitable for real-time applications. Computational latency can be reduced by simplifying the algorithm in order to increase efficiency or in the case of algorithms that are suitable for parallelisation utilising the processing power of many cores of the central processing unit (CPU) or graphical processing unit (GPU) to decrease computational time. There are two distinct approaches to address observational latency:

the first is automatic segmentation of the sequence followed by classification of the individual actions and the second is to perform continuous classification.

Automatic segmentation is a natural progression to enable existing offline recognition approaches to be used online. De Torre et al. [34] use a clustering algorithm to cut sequences into action instances. However, their segmentation algorithm is processed offline so subsequent action recognition would also be offline. To overcome this limitation Gong et al. [45] fused the segmentation with matching. However, as the segmentation is based on capturing transitions between actions, the recognition can only occur after the action is complete incurring high observational latency, because of the potential difference between peak time and completion time.

An alternative approach for online action recognition with very low latency is to reduce template matching to single pose matching. Ellis et al. [24] automatically reduce the number of key poses to a single canonical pose for each action. The disadvantage of such an approach is that no temporal history of an action is used, and as a consequence matching of just a single pose may lead to false detections especially when different actions contain similar poses.

Eickeler et al. [68] proposed two methods based on HMM for continuous recognition of gestures: smoothing and filtering. The former approach achieved high accuracy but with high observational latency (12 seconds) which may be acceptable in some applications e.g. sign language recognition but not suitable for human-computer interaction. The latter approach reduced the time delay of recognition but only if the gestures were temporally isolated which limits its suitability for gaming scenarios. Natarajan and Nevatia [69] proposed a hierarchical HMM with variable size sliding temporal window to achieve high accuracy at low observational latency (average 3.2 frames) and real-time computation (28.6fps) for online action recognition. Although, this method allows continuous action

recognition the method requires prior knowledge of the structure of the actions, like the limbs involved.

To precisely measure latency Nowozin and Shotton [13] introduced action points, a temporal anchor for action instances within a sequence. For example, an action point for a punch could be defined as the moment at when the arm is maximally extended. They also proposed two recognition models that can detect action points in real time. Their first approach, Firing Hidden Markov Model [13] is a variation of HMM with an explicit firing state which detects action points when the probability of the action exceeds a threshold. In their experiments they compared offline smoothing with online filtering. As expected the accuracy of the online variant is significantly lower than of the offline method, as the latter incorporates the whole action sequence.

Nowozin and Shotton second approach, online Random Forests [70] was adapted for continuous action recognition using a sliding window approach. Experiments showed that Random Forest was simpler, faster and more reliable than the HMM approach [13], [71]. However, the fixed size of the sliding window in these approaches is a source of error due to execution rate variations. To address this Zhao et al. [72] optimised the size of the segment during their pre-processing using a DTW variant for subsequence matching. However, as the average length of their templates is 35 frames observational latency is high. Sharaf et al. [73] achieved state-of-the-art results for online action recognition with a feature selection approach combined with a SVM. Sharaf et al. used features at multi-scales to improve execution rate invariance but their approach is computationally limited to a couple of levels which limits the execution rate invariance.

Similarly, a sliding window approach enabled online AdaBoost [23], for action recognition in commercially released games but due to commercial sensitivity relatively little information was available regarding the technical details. A comparison of Random Forests and AdaBoost showed that AdaBoost can provide higher classification accuracy

at the cost of less efficient computation [74]. Due to their success for online action detection with low latency AdaBoost and Random Forests will be used as baselines in this thesis and their implementation details are discussed in section 2.4.3.

2.4.3 ONLINE ACTION RECOGNITION IN THIS THESIS

The online action recognition pipeline used in this thesis is introduced and contrasted with the offline action recognition pipeline. A more detailed examination of the classifiers Random Forest and AdaBoost used for the comparative experiments in this thesis are provided in addition to the introduction of Decision Trees which are the foundation of both of these classifiers.

2.4.3.1 Online Action Recognition Pipeline

The online action recognition pipeline used in this thesis is shown in Figure 2-15. The key differences with the offline approach illustrated in Figure 2-4 are the streamed testing data, the ground truth labels, evaluation metrics and an additional post-processing step to temporally localise the action which depends on the classifier. If the classifier outputs the probability of the action label at each frame this can be compared with a threshold to determine if an action point has been detected. The testing data is streamed to simulate a real-world application where at any point in time only past occurrences are available. For repeatability and comparison with other approaches public action recognition datasets where available are used. Action point ground truth labels and the action point F_1 -score performance metric are used to evaluate both latency and accuracy (these are discussed in section 2.5.2).

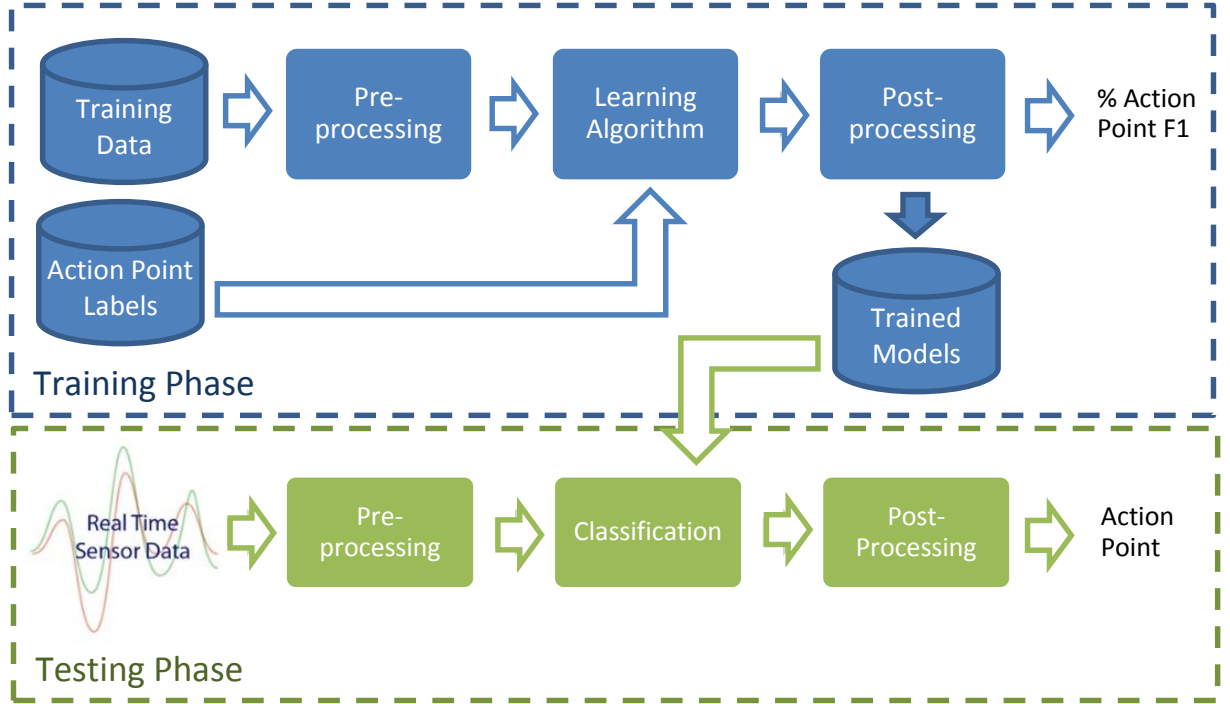


Figure 2-15 Online action recognition pipeline, the key differences with the offline approach are the streamed testing data, an additional post processing step to temporally detect the action and the action point F1 latency measure.

2.4.3.1.1 Binary Decision Trees

A Decision Tree [75] is a discriminative classifier. The tree finds one data feature and a threshold at the current node that best divides the data into separate classes, as shown in Figure 2-16. For classification, an impurity metric is employed. Three common impurity measures are entropy, Gini index and misclassification. All the algorithms attempt to minimise the impurity at a node but Gini impurity Eq. (2-11), is the most commonly used, where $P(\omega_{j_\omega})$ denotes the fraction of patterns at node N that are in class ω_{j_ω} .

$$\gamma(N) = \sum_{i_\omega \neq j_\omega} P(\omega_{i_\omega})P(\omega_{j_\omega}) \tag{2-11}$$

The Decision Tree searches through the feature vector to find which feature combined with which threshold most purified the data. The data is split by branching features below the threshold to the left and the remaining features right. This procedure is repeated recursively down the left and right branches of the tree. Decision Trees are not affected by variance differences in feature variables as each variable is searched only for its effectiveness to split the data. Therefore, features do not need to be normalised unlike other classifiers.

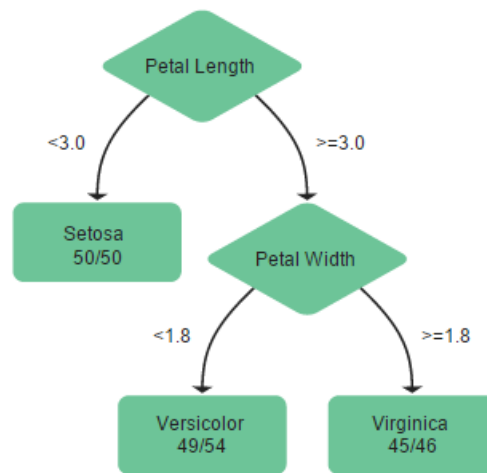


Figure 2-16 Decision Tree Example for classifying the species of flower (Setosa, Versicolor, Virginica) by petal measurements [76]

Decision trees are extremely useful due to their simplicity, ease of interpretation and natural way of assigning importance to the data features but they are often not the best-performing classifiers as they can be prone to overfitting. Nevertheless, they form the basis of state-of-the-art machine learning algorithms such as AdaBoost and Random Forests which inherit many of their useful properties.

2.4.3.1.2 Random Forest

A Random Forest [70] is a collection of many Decision Trees, each built randomly, as shown in Figure 2-17. During learning a random subset of the original features are used to build each tree so that they become statistically independent. Random Forests is a multi-class classifier as at test time votes are collected at the leaves of each of the many trees and the maximum vote is the winner. Averaging many trees counterbalances the overfitting problems encountered with individual trees.

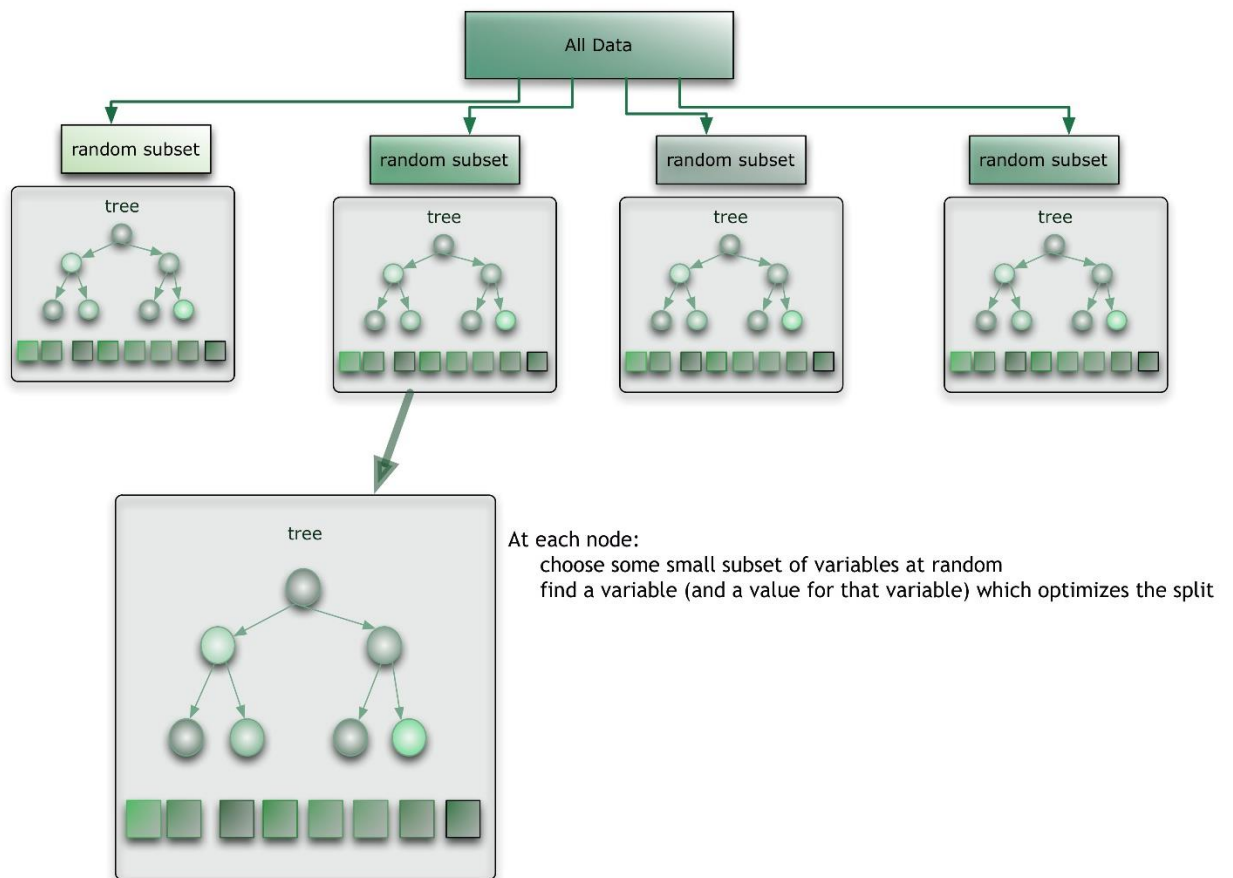


Figure 2-17 Random Forest: consisting of multiple decision trees learnt on random subsets of the training data. At each node a small subset of variables are selected at random and the variable that optimises the split is found [77]

To ensure each tree is different, a random feature subset is chosen to best split the data and the feature subset is different for each subsequent node in the tree. The size of these subsets is often the square root of the number of features. To increase robustness Random Forests use an “out of bag” (OOB) measure to verify splits. At a given node, training occurs on a new subset of the data that is randomly selected with replacement, and performance is estimated using the rest of the data (OOB data). The OOB data usually contains one third of all the data points and can be used to estimate how well the Random Forest will perform on unseen data. If the training data has a similar distribution to the test data, the OOB performance prediction can be quite accurate.

2.4.3.1.3 Boosting

The aim of boosting is to combine a group of weak classifiers to produce a strong classifier. A weak classifier has a slightly better chance of obtaining the correct classification than random guessing and can be implemented as decision trees with only one split (decision stumps [78]) or at most a few levels of splits. Each classifier has a weighted vote ξ_w in the final decision making process. A data point weighted distribution informs the algorithm how much misclassifying a data point will “cost”. The key feature of boosting is that, this cost will evolve so that weak classifiers trained later will focus on the data points that were misclassified earlier.

When the training is complete the final strong classifier $\Psi(\mathbf{x})$ takes a new input vector \mathbf{x} and classifies it using a weighted sum over the learned weak classifiers ψ_w calculated as:

$$\Psi(\mathbf{x}) = \text{sign} \left(\sum_{i_w=1}^W \xi_w \psi_w(\mathbf{x}) \right) \quad (2-12)$$

where W is the number of weak classifiers and each classifier has a weighted vote ξ_w .

It should be noted that AdaBoost is a binary classifier whereas action recognition is a multiclass problem. There are different strategies for converting binary classifiers into multiclass classifiers. One-vs-all (OVA) is computationally simple and as accurate as any other approach [79]. OVA trains Ω binary classifiers one for each class to distinguish the examples from one class from all other classes. To output a label ω for unseen example \mathbf{z} , the Ω classifiers are run and the classifier that outputs the highest certainty score is chosen:

$$\omega = \arg \max_{i_{\Omega}=1 \dots \Omega} f^{\Omega}(\mathbf{z}) \quad (2-13)$$

2.5 Evaluation

An overview of the wide range of public action recognition datasets is provided which are assessed with regard to the modality of the data and the type of actions. The limitations of the existing datasets are presented and the public datasets used in thesis are introduced. The ground truth and performance metrics used for action recognition are appraised in respect to approaches that can evaluate both latency and accuracy. Finally, cross validation approaches are investigated that can provide an unbiased estimate of the generalisation error ensuring the proposed algorithms will perform as expected on unseen subjects.

2.5.1 ACTION RECOGNITION DATASETS

Traditionally, human action datasets were recorded with visible light cameras and consist of colour or intensity data (for a comprehensive review of these also see Aggarwal and Ryoo [22]). The major problem with these cameras is that there is a considerable loss of information related to human motion when the real-world data (3D) is projected to 2D. After the recent release of low cost depth sensors there has been a rapid growth of 3D datasets that provide depth data and/or skeleton data (for a summary of these datasets see

Aggarwal and Xia [24]). The datasets can be categorised based on the scenarios where the actions are performed, the general trend has been to move away from scripted scenarios in controlled environments to real-life scenarios such as surveillance, daily life, movies and sports.

2.5.1.1 Scripted Scenarios

A popular method of collecting data is to instruct the participant to perform the desired actions in controlled environments. The first scripted scenarios such as the KTH [10] and Weizmann [80] datasets (see Figure 2-18) contained simple actions and each video sequence only contained one class of action. Motion capture datasets [81] [82][83] capture high quality skeleton data (see Figure 2-19) and contain a much wider variety of actions including sports and locomotion with multiple action classes in a sequence making them more applicable to real-world scenarios. Gaming actions may include sports and locomotion actions but there are subtle differences such as the manner the action is performed and the viewpoint of the camera. Even simple actions such as walking are different in the gaming environment as the player will walk on the spot. The HDM05 Motion Capture Database [82] database does include locomotion on the spot but not a full range of gaming actions.



Figure 2-18 KTH [10] intensity data

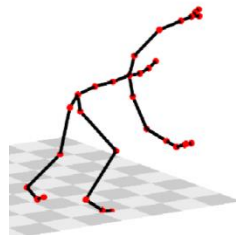


Figure 2-19 HDM05 [82] mocap data

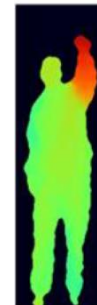


Figure 2-20 Action3D [40] depth data

Microsoft research specifically developed a gaming action database, MSR Action3D Database [40] which initially consisted of a sequence of depth maps (see Figure 2-20) and was later extended by a third party to include skeleton data, however the skeleton data is very noisy. Subsequently, Microsoft research released another gaming dataset MSRC-12 [71] captured with the Kinect which contained much more reliable skeleton data than their previous dataset. Similarly, Masood et al. [57] also captured skeleton data using the Kinect for a gaming dataset with actions based on the game Mirror's Edge (see Figure 2-21 for example actions). Nevertheless, the existing gaming datasets only contain one action class for each sequence and no corresponding video data is available. There are no publicly available gaming action recognition databases that contain multiple action classes and all three modalities (video, depth and skeleton). Furthermore, the existing gaming datasets are single person whereas most commercial games are multiplayer.

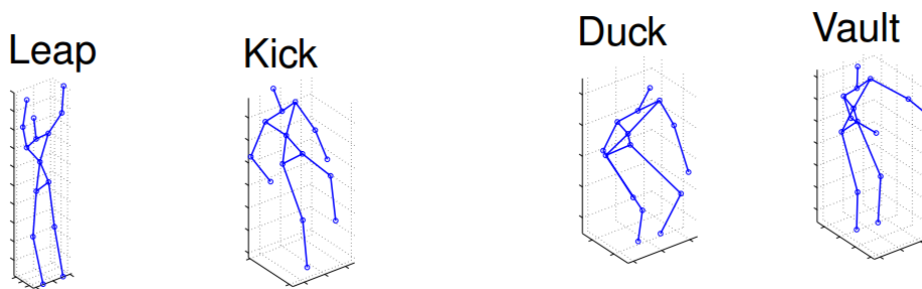


Figure 2-21 UCF Kinect dataset [12]

2.5.1.2 Real-life scenarios

The general trend especially with video datasets has been to move away from scripted scenarios in controlled environments to real-life scenarios such as surveillance, daily life, movies and sports. In surveillance, datasets such as PETS [84] and i-Lids [85] are obtained using security cameras in real outdoor environments such as car parks, airports and train stations (see Figure 2-22). Similarly, home cameras can be used to capture daily living tasks such as sleeping, cooking and watching TV for the purposes of assisted home living and smart homes.



Figure 2-22 PETS [84]



Figure 2-23 Hollywood 2 [86]



Figure 2-24 UCF sports action dataset [27]

An alternative approach to capture real-world scenarios is to extract footage from movies and TV. This footage naturally has diverse and cluttered backgrounds and frequently moving camera viewpoints. Popular movie datasets are Hollywood [35] and Hollywood2 [86] datasets (see Figure 2-23). Similarly, sports datasets have been extracted from TV footage such as YouTube Action Dataset [59] and UCF sports action dataset (see Figure 2-24) [27]. The individual actions are realistic but the major limitation of these datasets is that they have been segmented into sequences containing a single action.

2.5.1.3 Datasets used in this thesis

The focus of this thesis is gaming scenarios so the datasets extracted from movies or sporting events are not applicable. There are several existing scripted gaming datasets but they only contain one action class for each sequence and no corresponding video data is available. Furthermore, the existing gaming datasets are single person whereas most commercial games are multiplayer. Therefore, in this thesis two new multi-modal datasets containing video, depth and skeleton are proposed to overcome the existing limitations. Nevertheless, to compare to existing online action recognition algorithms the MSRC-12 dataset [71] will also be used.

The MSRC-12 dataset comprises of 30 people performing 12 gestures. These gestures are categorised into two categories: iconic and metaphoric gestures. The iconic gestures directly correspond to real world actions and represent first person shooter (FPS) gaming

actions. There are six FPS gaming actions: crouch, shoot, throw, night goggles, change weapon and kick as shown in Figure 2-25. In contrast to the iconic gestures, the metaphoric actions represent abstract concepts for manipulating a music player e.g. raise volume of the music. The same gesture is repeated 10 times by each subject, so each sequence contains multiple instances of the same gesture. The participants were instructed using different instruction modalities such as images, video and text. The instruction modality that produced the most accurate results was video plus text so this thesis uses this particular subset of the dataset. The dataset was captured using the Kinect but only the skeleton data was made publicly available.

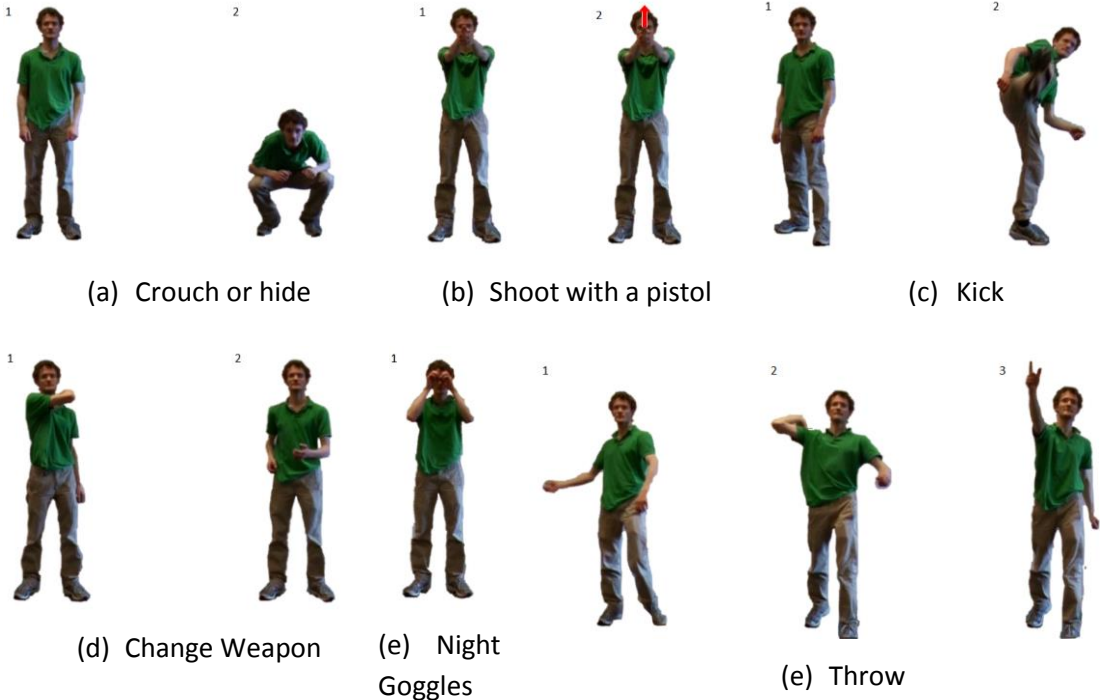


Figure 2-25 MSRC-12 Gaming Actions instructions provided to subjects (image modality) [71]

2.5.2 PERFORMANCE METRICS

Depending on the application action recognition algorithms may have very different constraints and requirements and therefore must be evaluated accordingly. If the video data is pre-segmented into action instances and processed offline, as in retrieval applications for movies, then it is sufficient to evaluate the algorithm purely in terms of accuracy. In contrast, online recognition systems process a continuous data stream in real time which means the evaluation must incorporate the latency of the detection as well as the accuracy.

2.5.2.1 Annotation

There are three types of temporal ground truth that are commonly used for action recognition: sequence-level, frame-range and action point (as depicted in Figure 2-26). There are also spatial annotations which are more relevant to video and depth data than skeleton data.

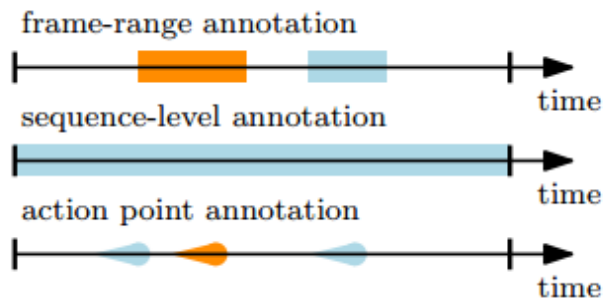


Figure 2-26 Annotation [13]

The sequence level annotation is the simplest form of annotation which provides an action label for each sequence. However, this annotation is only for sequences that are pre-segmented to contain one type of action that are intended to be processed offline. The frame range annotation labels each frame according to the action depicted at that point in time and can therefore be used for sequences containing multiple actions. Typically, the

label range includes all the frames from the onset (start) to the offset (end) of the action but it does not contain a temporal anchor to precisely measure latency which is a critical evaluation criterion in human-computer action and gaming. To measure latency Nowozin and Shotton [13] introduced action points, temporal anchors for action instances within a sequence. An action point has the following formal definition: “An action point of an action is a single time instance at which the presence of the action is clear and that can be uniquely identified for all instances of the action” [13]. Action Points themselves are not about the semantics of a particular action but allow application specific definitions to enable reproducible ground truth that has temporal anchors for measuring latency. In this thesis, an action point explicitly represents the peak of an action (as introduced in section 2.2).

2.5.2.2 Performance metrics

The performance metrics correspond directly to the annotations: classification accuracy is commonly used for sequence level annotation, F_1 -score for frame-based annotation and Action Point F_1 -score for action point annotation. These metrics can be calculated using four base cases shown in Table 2-2 for two class problems. Classification accuracy and the Frame F_1 -score can be evaluated as though they are two class problems even when more classes are being recognised. For each sequence (or frame) and for each action there is a positive label if the sequence contains the current action and negative label if it does not. Similarly, if the recognised action is the same action class as the label this is a positive detection and if it is another action class it is a negative recognition. For a positive label if the recognised action is also positive, this is a true positive (tp). If the recognised action is negative for a positive label this is a false negative (fn). For a negative label, if the recognition is also negative, it is a true negative (tn) and it is a false positive (fp) if a negative label is detected as positive. This approach however does not work for the Action Point F_1 -score which is discussed separately in section 2.5.2.2.3.

Table 2-2 Confusion matrix for two-class problems

Recognised class		
Ground truth	Positive	Negative
Positive	tp : True positive	fn : False negative
Negative	fp : False positive	tn : True negative

2.5.2.2.1 Classification accuracy (sequence level)

A common performance measure for action recognition is classification accuracy. Classification accuracy represents the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications, as shown in Eq. (2-14). Confusion matrices are frequently used to breakdown the number of correct classifications by class. The attractive feature of the confusion matrix is that the correct classifications are displayed along the diagonal axis making it clear when a class is misclassified.

$$\text{accuracy} = \frac{tp + tn}{tp + fp + fn + tn} \quad (2-14)$$

Applying the performance metric to simple action recognition datasets is straightforward as each sequence contains a single action. The simplest case is when the method recognises the action at the sequence level and outputs a single action class for each sequence. If the actions are recognised at the frame level an action label is output for each frame in the sequence and a majority decision over all frames is taken to decide the action label for the complete sequence. In either case, the recognised action label for the sequence is compared to the ground truth label for the sequence. This simple metric is

even used on more complex datasets such as the movie datasets using the false assumption that the sequence contains only one action, even if it actually contains multiple actions. This simple approach of applying the performance measure to the entire sequence does not measure latency required by real-time applications and is not an accurate measure if multiple actions occur in a sequence.

2.5.2.2.2 *Frame F_1 -score (frame level)*

In a realistic case, with multiple actions within a sequence the classification accuracy or F_1 -score is more suitably applied to each frame. The F_1 -score is more robust than accuracy when classes are imbalanced as it is the harmonic mean of precision p_r and recall r_e as shown in Eq. (2-15). However, frame level metrics do not measure latency which is required to make action recognition methods suitable for a range of real-world applications.

$$F_1 = 2 \frac{p_r \cdot r_e}{p_r + r_e} \quad (2-15)$$

$$p_r = \frac{tp}{tp + fp} \quad (2-16)$$

$$r_e = \frac{tp}{tp + fn} \quad (2-17)$$

2.5.2.2.3 *Action Point F_1 -score (instance level)*

Low latency is critical in interactive gaming to appear responsive to the player's actions. An action performed by the player must be detected as soon as possible to prevent poor gameplay. Nowozin et al. [13] proposed a latency aware performance metric for online human action recognition. They introduced 'action points' as temporal anchors for the detection and evaluation of actions in real time. An action label is deemed correct if it is detected within a specific time window of size 2Δ which is centred around the ground truth action point as illustrated in Figure 2-27. The correct detections are counted as tp ,

whilst ignoring multiple correct detections. In the case where no action or the incorrect action class is detected within the ground truth window then a fn is counted and in the latter case a fp is also counted. Similarly, to object based metrics for motion detection [87] there is no case of tn therefore popular evaluation metrics such as accuracy and the ROC curve cannot be applied.

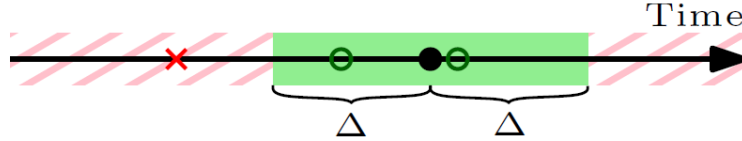


Figure 2-27 Action point F_1 metric for a single action: a fixed time window of size 2Δ is centered around the ground truth action point annotation (marked \bullet) and used to split the three detected action points into correct (marked \circ) and incorrect detections (marked \times). If there is more than one detected action point within the ground truth window only one prediction is counted. All incorrect detections are counted.

For a specified amount of latency (Δ) the action point F_1 score [13] determines whether a detection made at time t_{d_a} for action a is correct in relation to a ground truth action point at time t_{g_a} by using the following formula:

$$\Phi_a(t_{d_a}, t_{g_a}, \Delta) = \begin{cases} 1 & \text{if } |t_{g_a} - t_{d_a}| \leq \Delta \\ 0 & \text{otherwise} \end{cases} \quad (2-18)$$

For a specified amount of latency (Δ) precision p_r and recall r_e are measured for each action a and combined to calculate a single F_1 -score.

$$F_1(a, \Delta) = 2 \frac{p_{r_a}(\Delta) r_{e_a}(\Delta)}{p_{r_a}(\Delta) + r_{e_a}(\Delta)} \quad (2-19)$$

As online action recognition algorithms need to detect multiple actions, the mean F_1 score over all actions is used, defined as:

$$F_1(A, \Delta) = \frac{1}{|A|} \sum_{a \in A} F_1(a, \Delta) \quad (2-20)$$

2.5.3 CROSS VALIDATION

The ability of action recognition algorithms to correctly classify new examples that differ from those used during training can be measured by its generalisation error. If a large amount of data is available then the training and test set can be created by taking independent samples and a third set, the validation set can be created to tune the model's parameters. In many real world applications, it is expensive and time consuming to collect a large dataset and segmenting the data into training and testing is inappropriate. In gaming datasets, these difficulties are reflected by the small number of users captured. In such scenarios where a limited amount of training data is available, a hold-out procedure can be applied to obtain a reliable estimate of the algorithms generalisation error.

One of the most common hold out procedures is cross validation which involves portioning the dataset into complementary subsets, training the model on one subset (training set) and validating the model on the other subset (the test set). Multiple rounds of cross validation can be performed and the results averaged to reduce the variability of the generalisation error. K-fold cross validation involves partitioning the original sample into randomly partitioned sub-samples. A typical value of K is 10 and the extreme version is leave-one-out cross validation which leaves out one training sample each time, however in the case of time series data involving human these approaches can provide optimistic estimates that may cause overfitting as the data samples in the training and testing sets are not independent. Leave-one-subject out cross validation (LOSOCV) overcomes this problem by leaving out all observations from the same subject providing an unbiased

estimate of the generalisation error ensuring the algorithm will perform as expected on unseen subjects [88]. Generalisation to unseen subjects is a typical requirement for real-world applications so LOSOCV will be used for experiments in this thesis.

2.5.4 EXPERIMENTAL SETUP

Many personal computers have two or four cores that enable multiple threads to be executed simultaneously and in the near future computers are expected to have significantly more cores. To take advantage of these developments in hardware, the algorithms developed in this thesis have been designed to use parallel programming where appropriate to decrease training and testing time. The PC used for experiments in this thesis has the following specification:

Hardware	Software
Processor: Intel Core i7-2600 CPU @ 3.40GHz	Operating System: Windows 7
Memory: 6.00GB	IDE: Visual Studio 2010, Matlab 2011a
Number of cores: 4	Programming languages: C# and Matlab
Number of logical processors: 8	Libraries: EmguCV v2.3, OpenCV v2.4.3, Kinect SDK v1.7

2.6 Conclusion

Many state-of-the-art action recognition algorithms are appearance based which have the benefits of little to no high level processing and encoding contextual information. The problem is that most of these algorithms are far from being real-time and the lack of

contextual information in a gaming scenario may mean these approaches underperform. Due to recent advances in depth camera technology and a reliable pose estimation algorithm, alternative approaches based on depth maps and skeleton data have been proposed. Skeleton features have reduced the viewpoint and anthropometric variations and have outperformed colour features and therefore will be used exclusively in the experimental sections of this thesis. Therefore, in this thesis the focus is developing learning algorithms that address execution rates and personal style.

There are many types of machine learning algorithms that have been applied to action recognition but the majority of approaches have been applied offline and even the online approaches have high latency. Notable exceptions are AdaBoost and Random Forests that have been successfully applied online with low latency, so, they will be used as baselines in this thesis. Both observational and computational latency should be considered when developing algorithms to ensure that they are suitable for real-world applications. Approaches to simplify existing algorithms need to be investigated in addition to selecting algorithms that are suitable for parallelisation to ensure low computational latency. Continuous classification is preferable over automatic segmentation to ensure low observation latency and sliding window approaches need further investigation to determine their effect on execution rate invariance.

There are many public datasets containing video sequences for action recognition but none specifically containing gaming actions which differ from sports and locomotion actions in the manner they are executed and the viewpoint of the camera. There are several existing scripted gaming datasets recorded with the Kinect but they only contain one action class in each sequence whereas real games contain a variety of different actions. Furthermore, the existing gaming datasets are single person whereas most commercial games are multiplayer. Therefore, in this thesis two new multi-modal datasets containing video, depth and skeleton are proposed to overcome the existing limitations. Nevertheless,

to compare the proposed algorithms to existing online action recognition algorithms the MSRC-12 dataset will also be used.

Classification accuracy is the performance measure used to compare the state-of-the-art offline action recognition algorithms. The simple manner in which it is applied to the entire sequence does not incorporate latency constraints required by real-time applications and is not an accurate measure if multiple actions occur in a sequence. Nowozin et al. [13] proposed a latency aware performance metric for online human action recognition. They introduced ‘action points’ as temporal anchors for the detection and evaluation of single person actions in real time. The Action Point F_1 -score will be used in this thesis to evaluate the single person action recognition algorithms and it will be extended to evaluate interaction recognition.

The existing online action recognition methods use action points and the algorithms have been specifically designed to detect actions that are momentary and discrete in nature. The existing approaches cannot detect the duration of the action peak which is critical for detecting interactions and is addressed in chapter 5. Additionally, the existing approaches cannot detect multiple concurrent actions performed by the same subject such as walking and waving. In chapter 5 progress toward overcoming this limitation is made by detecting actions that are performed in quick succession and temporally overlap. The existing approaches cannot detect continuous activities such as walking or running or a sequence of movements such as dancing. Detecting long range temporal dependencies is out of the scope of this thesis but detecting individual walking steps or dance movements can be considered the same as detecting a punch or kick which are the main focus of this thesis.

CHAPTER 3

ACTION RECOGNITION USING DYNAMIC FEATURE SELECTION

3.1 Introduction

Action recognition algorithms suitable for real-world applications must be capable of processing a continuous stream of multiple actions in real-time. The latency of the recognition can vary, depending on the application. For example, a sign language recognition system may delay recognition until a sequence of words or an entire sequence is parsed [11]. Such systems can benefit from increased accuracy by delaying the recognition. However, applications such as interactive computer games based on human actions do not have this luxury, as they require recognition with low latency. Nevertheless, the majority of existing action recognition approaches have been applied offline and even the online approaches have high latency. Notable exceptions are AdaBoost [23] and Random Forests [70] that have been applied online with a sliding window approach to achieve low latency.

Dimensionality reduction techniques have been used in conjunction with machine learning algorithms to reduce the number of considered features to improve computation time and reduce memory requirements. Furthermore, when the dimensionality of the feature set is high, some features may be irrelevant or noisy and therefore removing these features can improve accuracy. There are many different dimensionality reduction techniques that can be divided into feature selection and feature transformation. Feature selection methods choose a subset of important features whereas feature transformation methods form new features, that are fewer in number than the original. Due to the large

number of existing dimensionality reduction techniques this chapter will focus on feature selection approaches and the following chapter on feature transformation techniques.

There are many public datasets containing video sequences for action recognition [10], [27], [35], [59], [80], [84]–[86] but none specifically containing gaming actions. There are several existing scripted gaming datasets recorded with a depth sensor [40], [57], [71] but they only contain one type of action in each sequence whereas commercial games contain a variety of different actions. Therefore, in this chapter a new multi-action, multi-modal dataset (G3D) containing video, depth and skeleton is captured to evaluate the proposed algorithm and made publicly available for other researchers.

The contributions in this chapter are two-fold; (1) a novel algorithm for online action recognition, Dynamic Feature Selection which combines the discriminative power of Random Forests for feature selection with an ensemble of AdaBoost classifiers for dynamic classification to improve accuracy [89] and (2) a new gaming action dataset, G3D [90] which is the first public gaming action dataset to contain multi-actions and multi-modal data.

3.2 Related Work

A review of both offline and online action recognition algorithms in addition to relevant datasets and evaluation metrics are presented in section 2. In this review the focus is on feature selection in general and then specifically how it has been applied to action recognition. The aim of feature selection is to find the most discriminative subset of features that contribute most to the performance of the classifier. Numerous feature selection methods have been developed which can be divided into wrappers, filters and embedded methods [91].

Filter methods select subsets of variables by ranking individual variables with scoring functions such as correlation coefficient or mutual information criterion. The variable ranking is performed as a pre-processing step independent of the classifier. The benefits of these approaches are their simplicity and computational efficiency. However, wrapper and embedded methods may give a better performance improvement over filter methods [91].

Wrapper methods are a simple and powerful way to address the feature selection problem. They use the prediction performance of a given classifier to assess the relative usefulness of subsets of features. The optimal feature subset can be found by testing all possible subsets. However, as there are $(2^D - 1)$ possible combinations of D features, it is computationally unfeasible for large numbers of features [92]. A wide range of search strategies have been proposed to address this issue, including forward selection, backward selection, best-first, branch-and-bound, simulated annealing and genetic algorithms (see [93] for a review). In pose-based action recognition genetic algorithms have been used to determine the optimum set of skeleton joints which improved recognition rates [94].

Embedded methods incorporate variable selection in the process of training and can be more efficient than wrapper methods. Decision trees [75] and Random Forests [70] contain a built-in mechanism to perform variable selection that can estimate the importance of each feature during the classification process.

Random Forests were employed by Negin et al. [95] as a discriminative feature selection tool to improve the action recognition performance of a Support Vector Machine (SVM) with a small fraction of the original pose-based features. It should be noted that [95] was published around the same time as the method proposed in this chapter [89] and at the time they were both the first to employ Random Forests as a feature selection mechanism for action recognition. The key difference is [95] used features extracted from the entire sequence which were processed offline whereas the proposed, Dynamic Feature Selection

is online [89]. A couple of years later Sharaf et al. [73] achieved state-of-the-art results for online action recognition with a similar feature selection approach that combined Recursive Feature Elimination with a SVM. Sharaf et al. selected features at multi-scales to improve execution rate invariance but their approach is computationally limited to a couple of levels which limits the execution rate invariance.

In contrast to selecting hand-crafted features, deep learning approaches have been used to learn features from unlabelled video data [96]–[98]. The benefit of deep learning is that the features can be automatically selected without the use of prior knowledge and they have achieved comparable or even better accuracy than engineered features for offline action recognition. Nevertheless, deep learning approaches require large amounts of training data.

3.3 Methodology

The main contribution of this chapter is a new approach to dynamically select the most discriminative pose based features for online action recognition. Specifically, Random Forests are used for feature selection, while a novel ensemble of AdaBoost models is proposed for dynamic classification. The classifiers work as local experts as different features sets are better able to discriminate different actions. In contrast to existing approaches [13], [71], [95] where the features are extracted from a fixed number of frames (e.g. 1 second) the proposed features represent a single frame to improve execution rate invariance. Execution speed may differ between action classes and between subjects and it is also important in the gaming scenario where different actions may be performed in quick succession. The proposed method has two key phases: an offline training phase and an online testing phase.

3.3.1 TRAINING PHASE

The training phase consists of two key steps: Random Forests for feature selection and a novel ensemble of AdaBoost models for dynamic classification, as depicted in Figure 3-1 and summarised in Table 3-1. The proposed approach is generic so it could use features from any modality but in this thesis pose-based features are used for their viewpoint and anthropometric invariance.

The feature vector for a given pose is represented by $\mathbf{x} \in \mathbb{R}^D$, where $D = 297$ features. The features are a combination of 57 position difference features, 60 position velocity features, 20 position velocity magnitude features, 80 joint angle features and 80 angle velocity features (for more details see section 2.3.4).

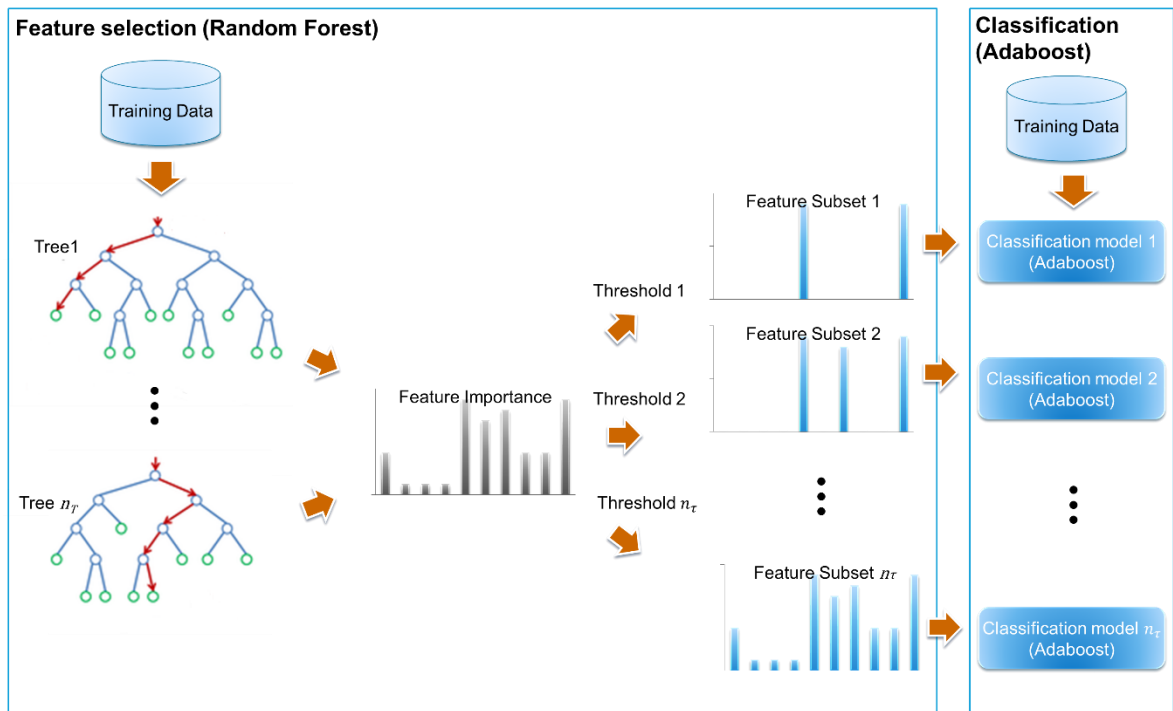


Figure 3-1 Dynamic Feature Selection (Training)

Table 3-1 Dynamic Feature Selection Algorithm (Training)

1. Feature Selection

- a. Train a Random Forest on the training set to obtain a set of n_T decision trees.
 - b. For each feature over all the trees in the forest calculate the average importance score using Breiman's [70] algorithm.
 - c. Rank the features in order of importance.
 - d. Group features into subsets of an increasing number of important features.
 - i. The first feature subset is obtained by selecting the features with the importance values higher than threshold τ_1 .
 - ii. Then add the features with the importance values higher than τ_2 to get the next feature subset.
 - iii. Repeat step (ii) until all features have been added to a subset and you have n_τ feature subsets, where $n_\tau \ll D$.
-

2. Classification

- a. For each feature subset ($1: n_\tau$):
 - i. Train an AdaBoost classifier using only the features selected for that subset
-

3.3.1.1 Feature Selection

Random Forests [70] are a collection of Decision Trees [75], where each tree is randomly grown. Details on these classifiers are provided in the background section (see sections 2.4.3.1.1 and 2.4.3.1.2). Random Forests are generally used as a discriminative classifier however in this chapter they are proposed as a discriminative feature selection tool, to estimate the importance of each feature.

Specifically, Breiman's [70] variable importance algorithm is calculated for each feature, by randomising this feature in each tree and measuring the percentage increase in the test

set error rate of the "out of bag" permuted features in comparison to the original features. The greater the increase in percentage error then the greater the importance of the feature for that tree. The average of this number over all trees in the forest is the importance score for the feature. The original D features can then be ranked in order of importance.

A simple static feature selection mechanism is to learn an importance threshold and discard features of lower importance, the reduced set of features can then be used to train another Random Forest or other classifier. Negin et al. [95] employed Random Forests as a discriminative feature selection tool to improve the action recognition performance of a Support Vector Machine (SVM) with a small fraction of the original pose-based features.

In contrast, the method proposed in this chapter uses a dynamic feature selection mechanism to improve accuracy by combining the discriminative power of Random Forests for feature selection with an ensemble of classifiers for dynamic online classification. The novelty of the proposed feature selection is the creation of multiple feature training sets instead of one feature set as in previous work. Given D features, the proposed approach creates n_τ feature subsets, where $n_\tau \ll D$ by thresholding the ranked features from 1 to D into groups as depicted in Figure 3-1 and summarised in Table 3-1.

Sharaf [73] found that different actions have different temporal scales therefore it is conceivable that the features which differentiate actions may change throughout the temporal duration of an action, particularly at the onset and offset of an action. My hypothesis is that different features sets are better able to discriminate different actions than a single feature set. The proposed approach is that multiple feature sets can be employed to train an ensemble of classifiers, which work as local experts at each frame to obtain better combined classification accuracy.

Examining the feature sets selected at each frame, reveals that the first feature set discriminated better between the action classes (punch, kick and defend) whereas the latter

feature sets were better able to discriminate the ‘Other’ action. The ‘Other’ action represents all the non-action frames and can be used to determine the start of an action which is critical for online action recognition. Therefore, selecting the highest confidence from a series of classifiers gives improved discrimination between the action classes and ‘Other’ class in comparison to individual classifiers. Results to support this are demonstrated in section 3.5.4.1.

3.3.1.2 Dynamic Classification

The novelty of the proposed classification is that the optimum feature subset is dynamically selected at each frame by training an ensemble of classifiers with different feature subsets. If the classifiers are run in parallel there should be no significant increase in computational time. The proposed framework is generic but to evaluate the performance AdaBoost [99] was selected as the classifier. Details on this classifier are provided in the background section (see section 2.4.3.1.3). The proposed training for dynamic classification assumes that D features have been grouped into n_τ feature subsets. Then an ensemble of n_τ AdaBoost models, one for each different feature subsets is learnt as depicted in Figure 3-1 and summarised in Table 3-1.

3.3.2 TESTING PHASE

During testing an AdaBoost model is dynamically selected at each frame based on the highest detection to provide real-time classification as illustrated in Figure 3-2 and summarised in Table 3-2.

Table 3-2 Dynamic Feature Selection Algorithm (Testing)

1. For each frame use each feature subset and corresponding multiclass classifier to give n_τ classifications.
2. Store the highest certainty for each action at each frame.
3. Add the highest certainty at each frame to a sliding window and sum results over the window for each action.
4. The action label for the current frame is the most confident classification for all actions.
5. The action points for a sequence are detected by a change in action label.

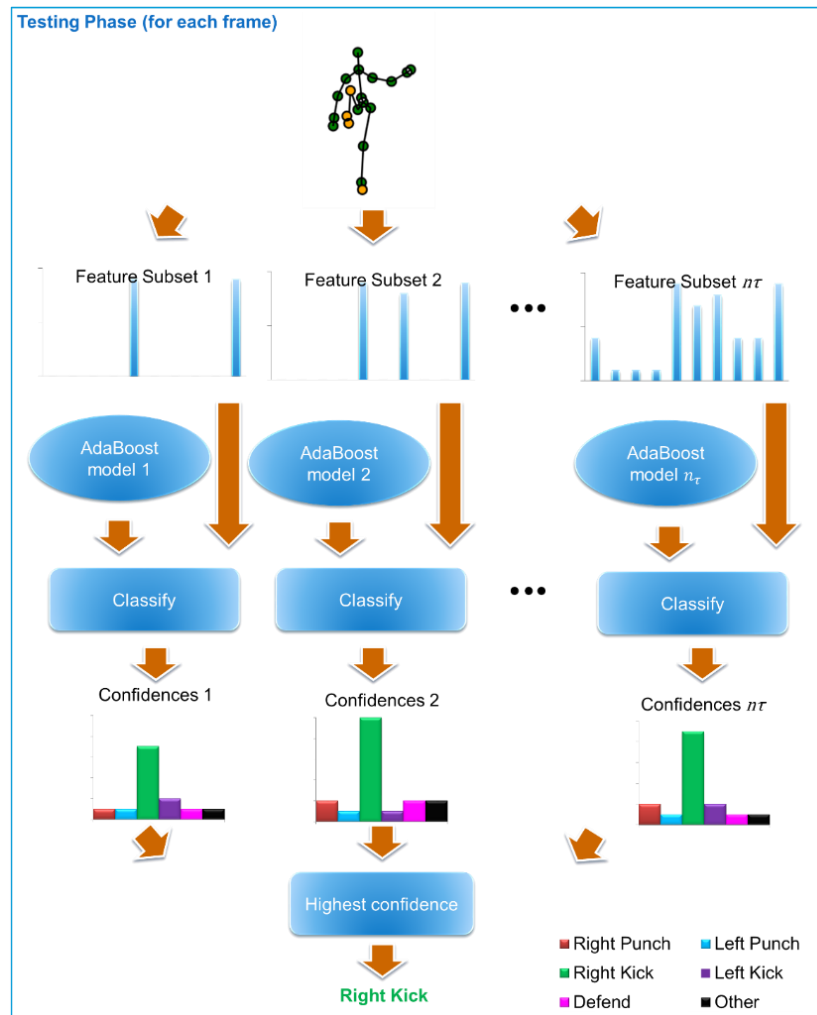


Figure 3-2 Dynamic Classification (testing)

Pose-based features for each frame are split into the same feature subsets learnt during training. Each feature subset is used as input for the appropriately trained AdaBoost model and each model makes individual detections. The most confident (highest) classification from all the models for each action is recorded as shown in Figure 3-3. The ‘Other’ action represents all other frames that are not the actions specifically being detected and are important to detect the point in time a specific action occurs.

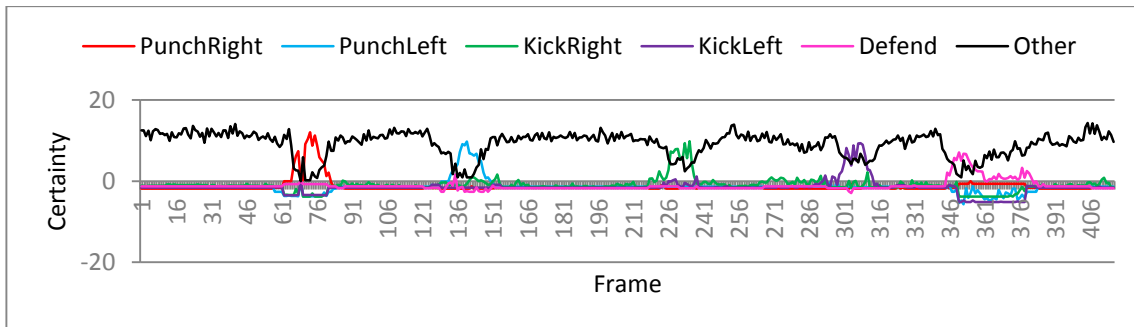


Figure 3-3 Frame based certainties for a fighting sequence from the G3D dataset

Frame based certainties are summed over a sliding window of n_w frames to smooth results to reduce false positives and increase accuracy as shown in Figure 3-4. The most confident classification determines the action label for a frame. A change in a frame based action label detects the action points for the sequence as shown in Figure 3-4.

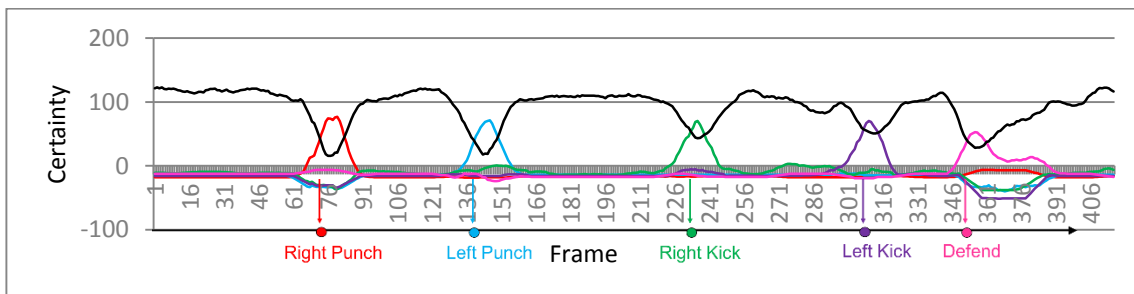


Figure 3-4 Smoothed results for a fighting sequence from the G3D dataset and detected action points for $n_w = 10$

The novel dynamic method for feature selection presented in this section can improve accuracy for online action recognition at low latency. To evaluate its performance in a gaming scenario a new dataset is required.

3.4 G3D Dataset

A new dataset, G3D for real-time action recognition in gaming, containing synchronised video, depth and skeleton data has been captured. This dataset is publicly available at <http://dipersec.kingston.ac.uk/G3D/> to allow researchers to develop new action recognition algorithms for video games and benchmark their performance. Due to the formats selected it is possible to view all the recorded data and tags without any special software tools.



Figure 3-5 Colour image



Figure 3-6 Depth map

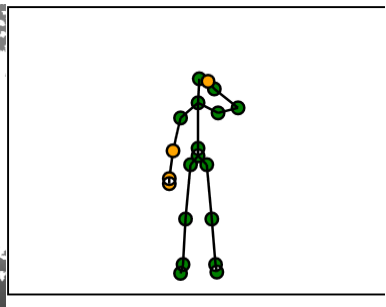


Figure 3-7 Skeleton data

The Microsoft Kinect enables easy capture of synchronised video, depth and skeleton data. The three streams were recorded at 30fps in a mirrored view so Figure 3-5 to Figure 3-7 are actually a right punch. The PNG image format was selected for storing both the depth and colour images as it is a lossless format, suitable for online access and is open source. The resolution used to store both the depth and colour images was 640x480. The raw depth information contains the depth of each pixel in millimetres and was stored in 16-bit greyscale (see Figure 3-6) and the raw colour in 24-bit RGB (see Figure 3-5).

The 16-bits of depth data contain 12 bits for the depth distance (0-4096mm), 1 bit reserved for the sentinel values (which was not used and fixed at 0) and 3 bits to identify the player. The player index can be used to segment the depth maps by user, as illustrated in Figure 3-8 where the player (blue pixels) can be easily distinguished from the background. The depth information was also mapped to the colour coordinate space and stored in a 16-bit greyscale. Combining the colour image with the mapped depth data allows the user to also be segmented in the colour image.

The XML text format was selected for storing the skeleton information as it is human readable and again suited for online access. The root node the XML file is an array of skeletons. Each skeleton contains the player's position and pose. The pose comprises of 20 joints. The player and joint positions are given in x, y and z co-ordinates in meters. These positions are also mapped into the depth (see Figure 3-8) and colour co-ordinates spaces. The skeleton data includes a joint tracking state, displayed in Figure 3-7 as tracked (green), inferred (yellow) and not tracked (red). In many cases the inferred joints will be accurate as in Figure 3-8 but certain situations where limbs are occluded the inferred joints may be inaccurate as in Figure 3-9. Consequently, pose data may need to be combined with colour or depth data to improve accuracy.

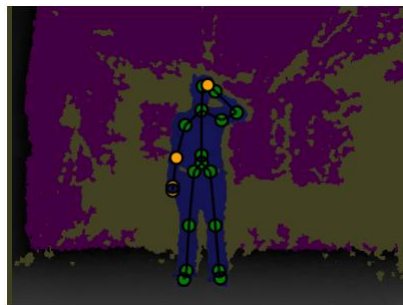


Figure 3-8 Correctly inferred joints (yellow).

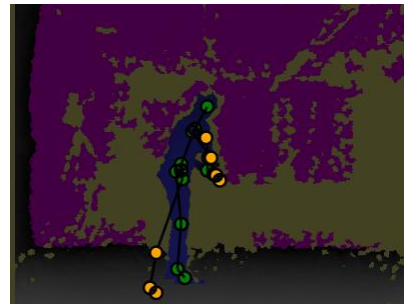


Figure 3-9 Incorrectly inferred joints (yellow).

This dataset contains 10 subjects, individually performing 20 gaming actions : punch right, punch left, kick right, kick left, defend, golf swing, tennis swing forehand, tennis

swing backhand, tennis serve, throw bowling ball, aim and fire gun, walk, run, jump, climb, crouch, steer a car, wave, flap and clap, grouped into seven categories: fighting, golf, tennis, bowling, FPS, driving and miscellaneous. Most sequences contain multiple actions in a controlled indoor environment with a fixed camera, a typical setup for gesture based gaming. The subjects were given basic instructions as to how to perform the action, similar to those issued in a Kinect game. Nevertheless, the subjects were free to perform the gesture with either hand or in the case of a side facing action stand with either foot forward to create a diverse dataset. Each sequence is repeated three times by each subject. However, in contrast to the MSRC-12 dataset [71] different actions for the same category are mixed together within a sequence and the sequence is repeated three times. Figure 1 shows example skeleton data for a fighting sequence. This resulted in over 80,000 frames of video, depth and skeleton data. All the frames in the dataset that contain actions were manually labelled in a separate file with an appropriate tag. Each tag represents a single action and contains the action class e.g. Punch Right and frame number representing the peak of the action (as shown in Figure 3-10). The XML tags are also publicly available.

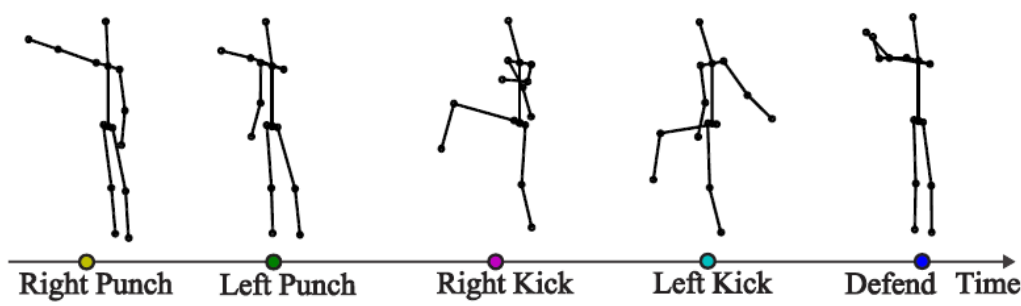


Figure 3-10 A fighting sequence from the G3D dataset with "action point" ground truth

In contrast to the existing gaming datasets, the G3D dataset is more realistic as it contains multiple actions within each sequence rather than repeating the same action multiple times as in MSRC-12 [71] and MSRAction3D [40] datasets (for a review of these datasets see

section 2.5.1.1). Additionally, G3D is the only gaming dataset to provide synchronised colour, depth and skeleton data, although in this thesis only the skeleton data is used.

3.5 Results

The proposed Dynamic Feature Selection framework was tested with publicly available gaming datasets against state-of-the-art approaches for online action recognition with the same experimental setup.

3.5.1 DATASETS

Existing gaming datasets are limited so G3D (introduced in section 3.4) was specifically captured for real time action recognition containing multiple actions in each sequence and also for a comparison with existing methods the publicly available gaming dataset MSRC-12 [71] (summarised in 2.5.1.3) was used. Both datasets provide sequences of skeleton data captured using the Kinect pose estimation pipeline at 30fps. Action point annotations of the peak poses are available for the MSRC-12 dataset and G3D dataset to precisely measure the latency of action recognition methods as well as the accuracy (described in section 2.5.2.1).

A “leave-person(s) out” cross validation protocol (described in section 2.5.3) was used where a set of people is removed to obtain the minimum test set that contains instances of all actions. For the MSRC-12 dataset this may be more than one actor as not every actor performs all the actions for the video + text modality². For the G3D dataset this is simply one actor as all actors perform all the actions. The remaining large set is used for the training. This process is repeated 10 times with different subsets to obtain the general

² This is the instruction modality used to teach the subjects how to perform the actions in the MSRC-12 dataset.

performance. The total number of training and testing instances for each dataset used in the following experiments is shown in Table 3-3.

Table 3-3 The total number of training and testing instances for gaming action datasets

Dataset	Actions	Subjects	Repetitions	Cross Validation	Training Action Instances	Testing Action Instances
G3D	5	10	3	10	1350	150
MSRC-12	6	10	10	10	5400	600

3.5.2 PERFORMANCE METRICS

For a fair comparison with existing approaches the same latency aware metric was used as initially proposed by [71] and later adopted by [73]. The detected action points are compared to the ground truth action points using the action point metric (described in section 2.5.2.2.3) to obtain a mean F-score at a fixed latency Δ , where $\Delta = 333ms^3$.

3.5.3 COMPARATIVE STUDY

The following is a brief summary of the comparison algorithms and the parameters used. For all the experiments the number of positive training samples selected around the action point was ± 8 and all other samples were used as negative training samples. The optimal positive sample size was found by varying this parameter between ± 1 and ± 20 on the training set.

³ A fixed latency of 333ms was already used for online action recognition and was adopted for a fair comparison with existing methods.

- **Random Forests:** is a state-of-the-art approach for low latency online action recognition [13], [71]. The 3 parameters that affect the performance of the Random Forest are the number of trees in the forest, the depth of each tree and the number of selected features at each node. Exhaustive searching of every combination of these 3 parameters is computationally prohibitive so in order to find the optimal forest configuration, 27 forests were trained with a combination of (10, 50 and 200) n_T trees, of depth (4, 6 and 8) with (10, 100 and 297) features selected at each node. Parameter selection was performed using cross validation on the training set, results of the cross validation are shown in Figure 3-11, Figure 3-12 and Figure 3-13. The best values of 200 trees, of depth 8 and 10 features at each node were found.
- **AdaBoost:** A comparison of Random Forests and AdaBoost in a different field [74] showed that AdaBoost can provide higher classification accuracy at the cost of less efficient computation. The standard version of AdaBoost is sensitive to noise in the dataset so Gentle AdaBoost [100] was selected as it gives less weight to outlier data points. As AdaBoost is also based on Decision Trees it has similar parameters: the number of weak classifiers which is the number of trees and the depth of the trees. Similarly, exhaustive searching is computationally prohibitive so in order to find the optimal configuration, 16 models were trained with a combination of (10, 50, 100 and 200) trees of depth (1, 3, 5 and 8). Parameter selection was performed using cross validation on the training set, results of the cross validation are shown in Figure 3-14 and Figure 3-15. The best values of 100 trees and depth 5 were found.
- **Dynamic Feature Selection:** The proposed method in this chapter combines Random Forests for feature selection with a novel dynamic variation of AdaBoost for online classification. The optimum parameters for Random Forest and

AdaBoost as described above were used for these experiments. The feature importance thresholds τ were set every 10% between 10% and 100%, so there were 10 feature sets n_τ in these experiments.

A smoothing window n_w of size 10 frames was applied to the frame based certainty results, which were provided by all the approaches except Random Forest which produced a direct classification for each frame. The final output from the algorithms tested is the set of detected action points for each sequence.

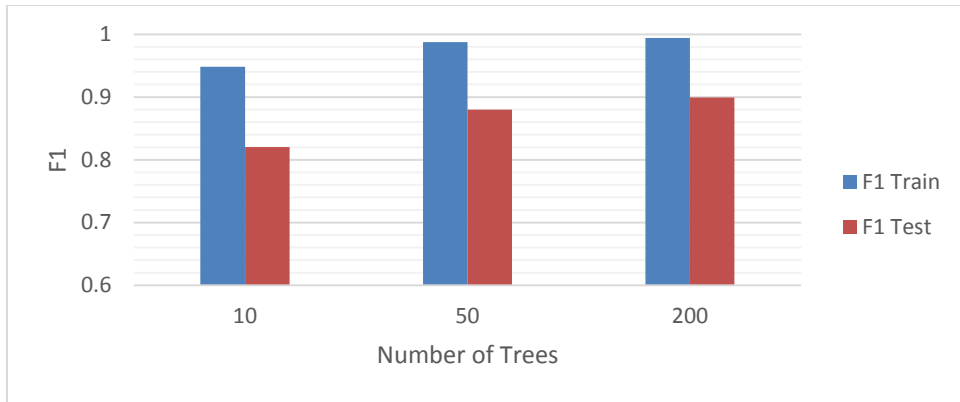


Figure 3-11 F1 results when varying the number of trees in the Random Forest, with depth fixed at 8 and number of selected features fixed at 10.

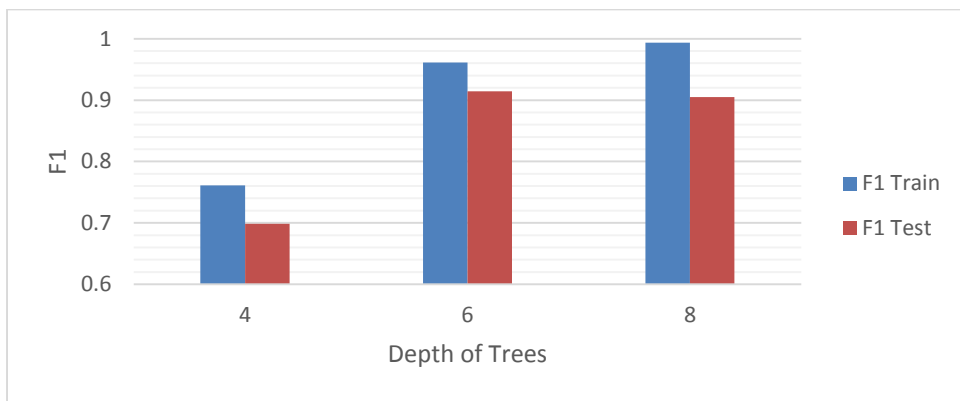


Figure 3-12 F1 results when varying the depth of the trees in the Random Forest, with number of trees set at 200 and number of selected features set at 10.

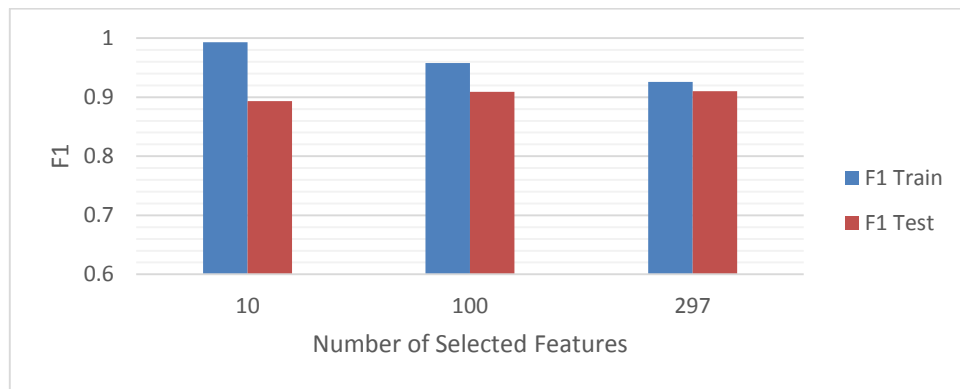


Figure 3-13 F1 results when varying the number of selected features at each node of a tree in the Random Forest, with depth fixed at 8 and number of trees set at 200.

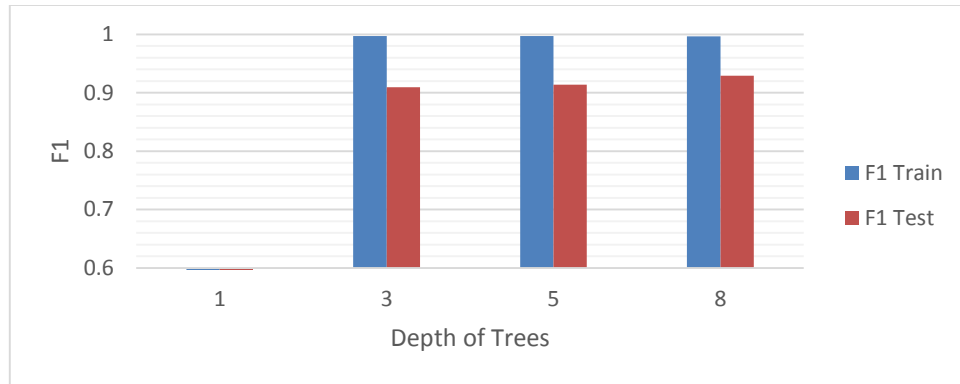


Figure 3-14 F1 results when varying the depth of the trees in Adaboost, with the number of trees set at 100.

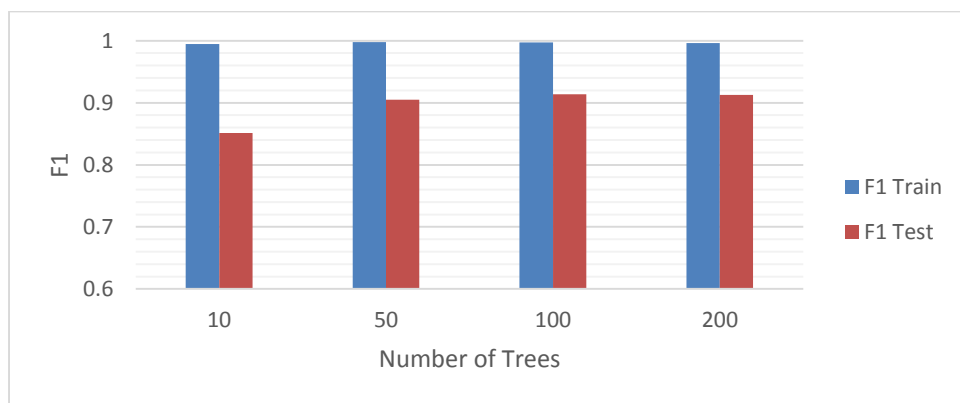


Figure 3-15 F1 results when varying the number of trees in Adaboost, with depth fixed at 5.

3.5.4 PERFORMANCE EVALUATION

The experimental results show that the proposed Dynamic Feature Selection framework improves accuracy across both datasets in comparison to AdaBoost and Random Forest without any feature selection. For the MSRC-12 and G3D datasets there is a 7% and 3% increase in performance respectively to the baseline AdaBoost method (see Table 3-4). The smaller increase on the latter dataset is because the F-score is already much higher for the AdaBoost method so there is less scope for improvement. The Dynamic Feature

Selection accuracy is not significantly different to state-of-the-art results with a 95% confidence and the computation time is 30% faster (see Table 3-4 for details).

Table 3-4 Action Point F1-scores at $\Delta=333$ ms and computation times, the average and standard deviations over ten leave-persons-out runs are shown. The results shown in italics were published by the method authors, all other results were re-created.

	Random Forest <i>[71]</i>	Random Forest	Ada-Boost	Dynamic Feature Selection	SVM-RFE <i>[73]</i>
Feature Vector	Multi-frame	Single-frame	Single-frame	Single-frame	Multi-frame
No. of features	<i>4550</i>	297	297	297	<i>100-10220</i>
Action Point F1-scores					
G3D	-	0.894 ± 0.155	0.884 ± 0.147	0.910 \pm 0.128	<i>0.937</i>
MSRC-12	<i>0.765</i> ± 0.070	0.619 ± 0.148	0.675 ± 0.156	0.744 \pm 0.270	-
Computation Time (per frame)					
G3D	-	1.029ms ± 0.014	1.088ms ± 0.02	1.001ms ± 0.038	-
MSRC-12	-	0.398ms ± 0.016	0.808ms ± 0.329	1.846ms ± 0.035	<i>2.63ms</i> <i>(100 features)</i> <i>2.704ms</i> <i>(200 features)</i> <i>2.779ms</i> <i>(300 features)</i> <i>11.908ms</i> <i>(10220 Features)</i>

Comparing the accuracy achieved by Fothergill et al. [71] on the MSRC-12 dataset using a Random Forest approach and the baseline Random Forest method presented in this chapter it can be noted that there is a significant drop in performance. The main difference between these Random Forest implementations is that Fothergill et al. [71] used a fixed feature vector of 35 frames whereas in this chapter a single frame feature vector was used. A more detailed analysis of the results by action (as shown in Figure 3-18 and Figure 3-19) reveals that the single frame feature vector performs poorly on the change weapon action. The action point of the change weapon as illustrated in Figure 3-16 is similar to poses in the night goggles as illustrated in Figure 3-17 so without the temporal history it is difficult to discriminate between these actions. Therefore, these experiments demonstrate that the temporal history of the action is important to differentiate between similar actions.

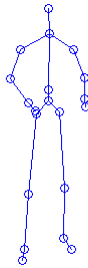


Figure 3-16 Change Weapon Action Point frame

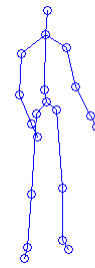


Figure 3-17 Night Goggles frame near the end of the action

Although a fixed size feature vector [71] incorporates temporal history, it is not invariant to changes in execution rate. Sharaf [73] were able to achieve state-of-the-art results on the G3D dataset by performing action detection across different temporal scales to improve execution rate invariance but their approach was computationally limited to a couple of temporal scales. In conclusion, to improve on the existing state-of-the-art approaches a method is required that can incorporate temporal history and be execution rate invariant.

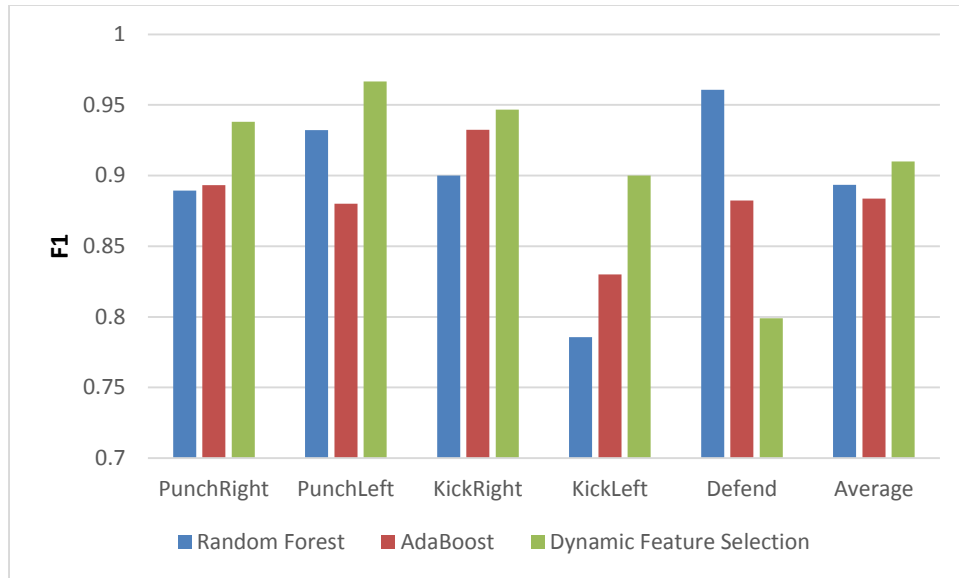


Figure 3-18 G3D Fighting results by action, all experiments conducted with a single-frame feature vector

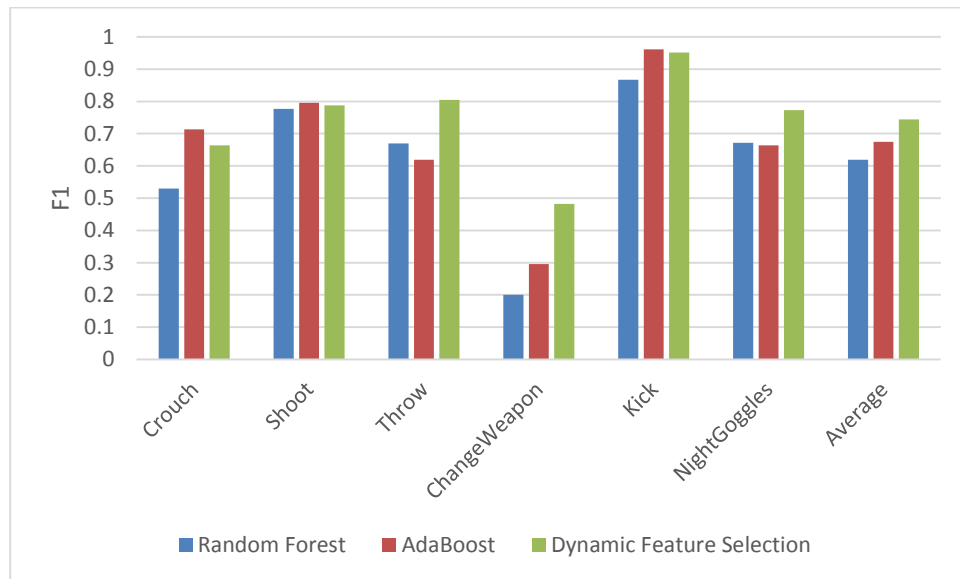


Figure 3-19 MSRC-12 Fighting results by action, all experiments conducted with a single-frame feature vector

3.5.4.1 Insights on Feature Selection

Existing feature selection approaches for action recognition [73], [95] incorporated temporal information from the entire action which resulted in very large initial feature vectors (13300 and 10220 features). These approaches focused on reducing features to enable real-time performance and were able to dramatically decrease the number of features to 10% of the original, which resulted in a decrease in computation time whilst maintaining or even improving the results.

However, the Dynamic Feature Selection approach began with a single frame feature vector (297 features) for execution rate invariance and real-time performance so the focus was on improving the accuracy by using an ensemble of AdaBoost classifiers. The intuition behind the increase in accuracy is that the classifiers work as local experts at each frame as different features sets are better able to discriminate different actions. This is supported by Sharaf [73] who found that different actions have different temporal scales. An analysis of which feature set is selected at each frame also supports this hypothesis, as in the G3D dataset the punching, kicking and defending actions favour the top 10% of features whereas the other action favours all the features (as shown in Figure 3-20). Similarly, in the MSRC-12 dataset the majority of actions selected the top 10% of features and the other action selected all the features (as shown in Figure 3-21).

In conclusion, the Dynamic Feature Selection approach is comparable to state-of-the-art results provided on the G3D and MSRC-12 datasets with a reduced computation time. The proposed method has an improved execution rate performance over existing approaches but at the expense of not being able to detect similar actions.

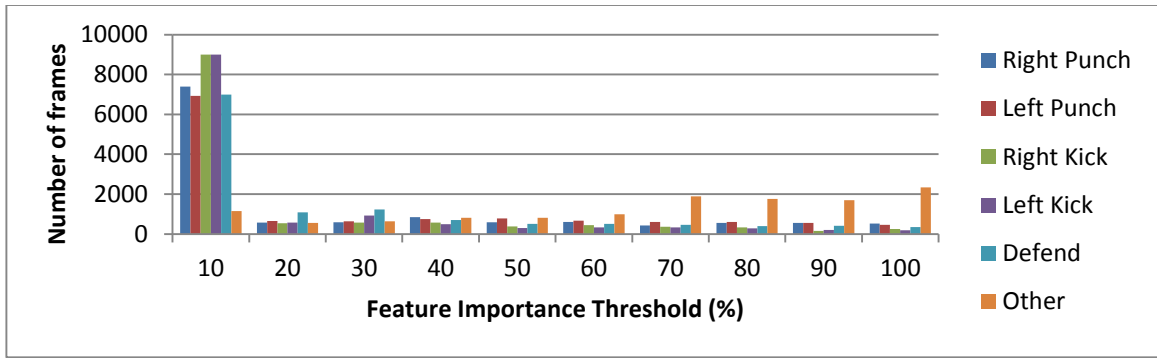


Figure 3-20 Feature Importance (G3D)

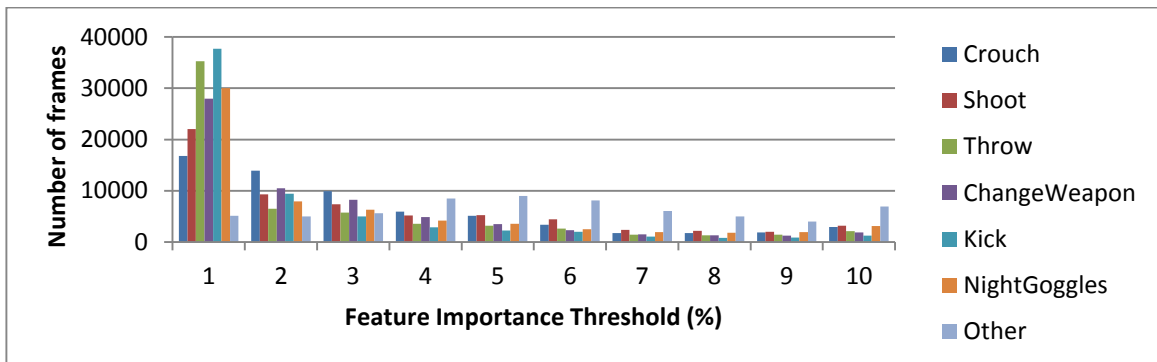


Figure 3-21 Feature Importance (MSRC-12)

3.6 Summary

This chapter introduced a novel method for Dynamic Feature Selection for online action recognition that combines the strengths of feature selection with local expert classifiers. Specifically, the feature selection method built into Random Forest was used to determine feature subsets and then the reduced feature vectors used to train an ensemble of AdaBoost classifiers. In contrast to existing approaches using feature selection, recognition occurs dynamically at each frame to select the most confident classification.

Additionally, a new dataset G3D for gaming action recognition was captured and made publicly available containing synchronised video, depth and skeleton data. This multimodal dataset has enabled researchers worldwide to evaluate action and pose

recognition algorithms. In contrast to the existing gaming datasets, the G3D dataset is more realistic as it contains multiple different actions within each sequence rather than repeating the same action multiple times.

Experiments on G3D and MSRC-12 publicly available datasets demonstrate that the new Dynamic Feature Selection algorithm for real-time action recognition improves the accuracy of baseline algorithms at low-latency. The results are also comparable to state-of-the-art algorithms and further analysis indicates that the proposed method improves execution rate invariance over existing approaches but at the expense of confusing actions that contain similar poses. In conclusion, temporal history is important as actions with similar poses must be distinguished. However, the existing state-of-the-art approaches are not invariant to execution rate changes and require seeing the entire action. The next chapter investigates an alternative approach based on dimensionality reduction that includes temporal history and is also execution rate invariant.

CHAPTER 4

ACTION RECOGNITION USING CLUSTERED SPATIO-TEMPORAL MANIFOLDS

4.1 Introduction

The previous chapter demonstrated that feature selection can improve accuracy and reduce computation time of online action recognition. However, the existing approaches fail to address execution rate invariance (section 3.2). This chapter focuses on feature transformation that maps the original high dimensional feature space to a much lower dimension, resulting in fewer features that are a combination of the original features. The advantage of feature transformation is that it handles the situation in which multiple features collectively provide good discrimination even if they provide relatively poor discrimination individually. Specifically, this chapter investigates the use of spatio-temporal manifolds that have been previously used for offline action recognition. The benefit of these manifolds is that they maintain the temporal history of the action to improve accuracy for action recognition and enable action prediction. Action prediction is a recent development in human action recognition, it involves forecasting future occurrences based on recent observations.

Action prediction is a very difficult problem for machines but is naturally performed by humans to coordinate their actions in time and space to accomplish their goals. Experimental results in human-human interaction in a table tennis game showed that action prediction improves performance [101]–[103]. Action prediction can enhance many applications with a human-machine interface in a range of domains including home entertainment, healthcare, sports, and robotics. For example, a personal robotic assistant

for the elderly can enable independent living by assisting with a range of cognitive and physical tasks to improve their quality of life. Natural social interaction between the robot and patient is important for acceptance of the robot in the patient's home and can also provide vital social contact for the patient [104].

Action recognition can be categorised into four distinctive approaches: offline, online, early and prediction (as illustrated in Figure 4-1). Traditionally, action recognition is performed offline using pre-segmented action sequences containing a single action and all the observations are used to classify the action. Similarly, early action recognition is typically performed on pre-segmented sequences but using as few observations as possible from the start of the sequence. In contrast, online action recognition approaches have the more complex task of classifying a continuous stream of actions in real-time. Additionally, temporal localisation of the action peak before the action is complete is required in applications that demand low latency (see section 2.5.2.1 for definitions and examples). Action prediction aims to estimate future action occurrences based on recent observations. Prediction on a continuous stream with temporal localisation of the action peak before it occurs is a very challenging scenario.

In this chapter a novel algorithm is presented that models the dynamics of human actions with Clustered Spatio-Temporal Manifolds (CSTM). The core of the algorithm creates novel style, invariant action templates that when matched with a sliding window variant of Dynamic Time Warping (DTW) provides execution rate invariance for continuous action classification in real-time, for early recognition. The proposed action templates provide the ability to follow the progression of an action and combined with new Peak Key Poses enable action detection with low latency. Furthermore, future progress can be estimated using regression for action prediction.

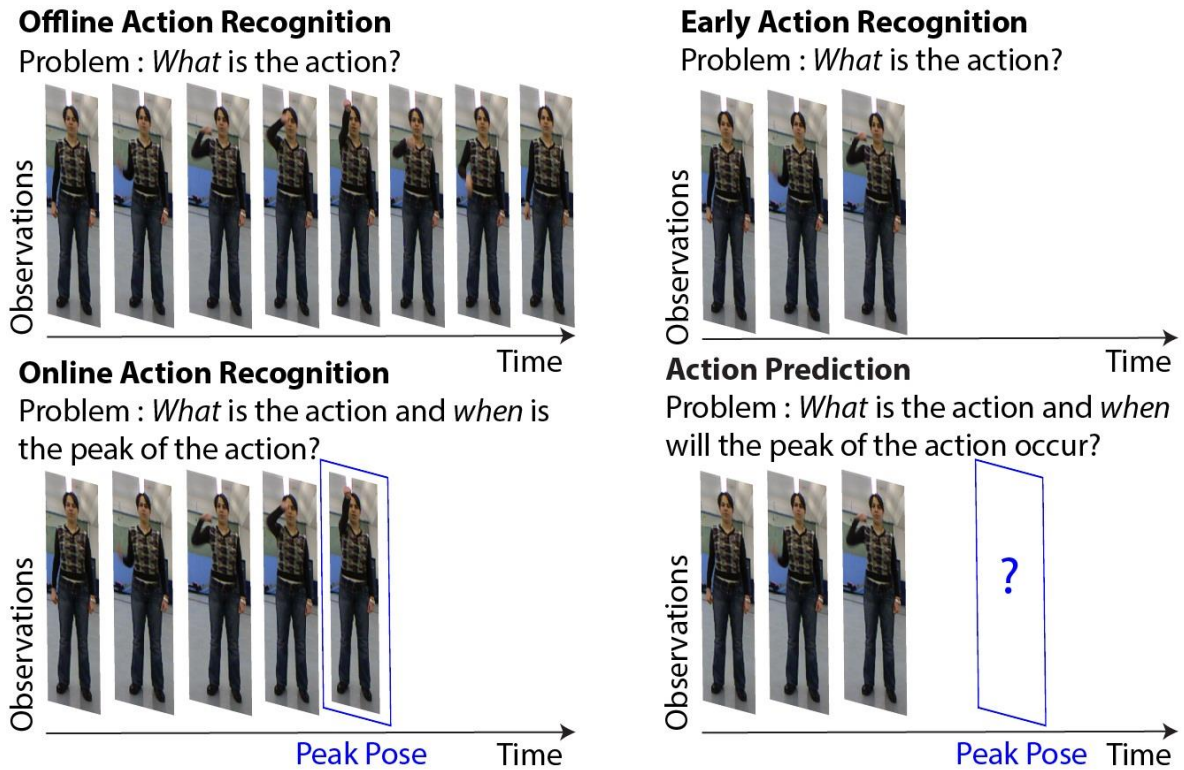


Figure 4-1 Observations required for offline, early, online action recognition and prediction

4.2 Related Work

Existing feature transformation approaches are analysed first, which have only been applied to offline action recognition. Then, a general review of the more recent research into early, online action recognition and prediction approaches is provided. For a broad review of online action recognition approaches see section 2.4.2.

4.2.1 FEATURE TRANSFORMATION

Schwarz et al. [105] use feature transformation with Laplacian Eigenmaps (LE) to suppress individual style. LE considers the spatial relationships between poses, but ignores the temporal relationships which are critical for recognising similar actions. This limitation has been overcome by spatio-temporal action manifolds [106]–[109].

Lewandowski et al. [106], [108] proposed Temporal Laplacian Eigenmaps (TLE) that extend LE by preserving the temporal structure and suppressing the stylistic variations of the data in the low dimensional space. Gong and Medioni [107] proposed a directed traversing path on a spatial manifold to incorporate the temporal dimension. They proposed Dynamic Manifold Warping for temporal alignment followed by spatial similarity of sequences on their manifold. Vemulapalli et al. [109] proposed a new representation of skeleton data as a Lie Group which is a 6D curved manifold. Human actions were modelled as curves on this manifold. DTW was used for execution rate invariance and additionally Fourier Temporal Pyramids to handle noise. The final classification was performed with linear SVM and achieved state-of-the-art results for offline action recognition. The spatial-temporal manifolds [106]–[109] are invariant to personal style and execution rate invariant but as the whole sequence is used for classification the action recognition has high observational latency and requires the action to be pre-segmented.

In related work, Paiement et al. [110] performed online quality assessment of human movement using a Diffusion Map (DM) to model normal movements. Like LE, DMs are a feature transformation approach that preserve the spatial structure in the low dimensional space but ignore the temporal relationships. Nevertheless, this is addressed in this approach by a separate pose and dynamic model and importantly the assessment is frame-by frame for continuous quality assessment making it suitable for online applications.

4.2.2 EARLY ACTION RECOGNITION

Early action recognition aims to determine the action class based on as few observations as possible, even when only part of the action has been seen. Existing early activity recognition approaches extend popular activity recognition methods such as bag-of-words

(BoW) [111], [112], sequential state models [113], [114] and maximum margin methods [115]–[117].

Ryoo [111] proposed two extensions to the bag-of-words paradigm for early activity recognition: Integral bag-of-words (Integral BoW) and Dynamic bag-of-words (Dynamic BoW). The integral histogram models spatial changes in the visual words but the temporal relations are ignored. Dynamic BoW overcomes this limitation by splitting an activity into subsequences and using a sequential matching algorithm. Dynamic BoW outperforms Integral BoW which highlights the importance of temporal modelling for early recognition. Both approaches determined accuracy on sequences that were manually pre-segmented to contain a single action where results were calculated after observing ratios from 0.1 to 1.0, where 0.5 represents half the action and 1.0 the full action. Dynamic BoW achieves reasonable accuracy when half the activity has been observed. However, the accuracy of both approaches is significantly reduced in the early part of the activity. Similarly, Cao et al. [112] use the bag-of-visual-words technique on video segments to incorporate local spatio-temporal features. Each video is uniformly divided into equal length segments and a mixture of segments of varied length and temporal shifts is used to improve execution rate invariance. However, this approach is limited to the number of scales and shifts that can be computed.

Sequential state models [113], [114] are effective at early recognition as they intrinsically preserve temporal order. Davis and Tyagi [113] proposed a Hidden Markov Model (HMM) for rapid and reliable early action recognition on manually pre-segmented sequences. Li and Fu [114] propose ARMA-HMM, an integrated autoregressive moving-average model (ARMA) with a HMM for early activity recognition on pre-segmented sequences. ARMA-HMM predicts future poses to enrich the partially observed activity sequences and improve early recognition. However, the reliance on manual pre-

segmentation which has to be performed offline, negates the benefit of the early detection of these approaches.

Lan et al. [116] developed a max-margin framework for early action recognition that achieves state-of-the-art results when half the action has been observed in a manually pre-segmented sequence but the accuracy is significantly reduced in the early part of the activity. Kong et al. [117] extend the max-margin approach to multiple temporal scales and achieve state-of-the-art results when the full action has been observed which is equivalent to the classic offline action recognition problem but accuracy is lower than Lan et al. [116] when observing half of the action.

Hoai and De la Torre [115] proposed max-margin early event detectors for early detection of a range of human activities i.e. facial expressions, gestures and actions. They extended Structured Output SVM to accommodate sequential data. Their learning formulation is a constrained quadratic optimisation problem to ensure monotonicity of the detection of partial activities. To evaluate their approach Hoai and De la Torre [115] concatenated manually pre-segmented sequences to form longer sequences containing multiple actions to temporally detect the action as soon as possible which is an improvement over the previous scenarios in this section of single action evaluation. However, they considered each action individually by placing the action of interest at the end of the sequence and lowering the false positive rate until it reached 0% to ensure their algorithm did not detect the action of interest before it started. Due to these artificial conditions it is not clear how their approach would perform in a real-world scenario of detecting multiple actions in a continuous stream.

The majority of existing approaches [111]–[114], [116], [117] for early activity recognition focus on classifying the action as soon as possible using pre-segmented sequences. These approaches achieve reasonable accuracy after observing half the action

but manual pre-segmentation simplifies the task of early detection which inflates accuracy and limits the applicability of these approaches to real-world scenarios.

4.2.3 ACTION PREDICTION

Action prediction is a recent development in human action recognition, which has received relatively little attention and is also the most difficult task as it involves forecasting future occurrences based on recent observations.

Sequential state models [67], [114] are able to predict future poses as they intrinsically preserve temporal order. Li and Fu [114] proposed ARMA-HMM, which predicts future poses to enrich the partially observed activity sequences. The focus of their work was to improve early recognition so the accuracy of the predicted poses was not evaluated. Also, Galata et al. [67] proposed variable-length Markov models (VLMM) to encode high-order temporal dependencies for animation of human activities. They synthesised hypothetical activity sequences using the VLMM as a stochastic generator to create realistic animations with statistically accurate variations. However, the aim of their work was to generate synthetic poses rather than predict actual future poses.

Vondrick et al. [118] demonstrated the difficulty of predicting actions by demonstrating that human subjects also fail to accurately predict actions in 30% of the cases when given a single frame one second before the action starts. To handle this ambiguity they develop a deep network architecture to produce multiple predictions and use large amounts of unlabelled video data to capture common sense knowledge about the world. Although they are still far from human performance on this task they are able to achieve reasonable accuracy for such a complex task. However, further analysis of their training frames shows that the start of an action is also an ambiguous concept as some examples do contain pose information that reveal the intended action and others contain contextual information that may be used to determine the action. In a gaming scenario, there is no contextual

information which may make prediction more difficult. Also, as gaming datasets are more difficult to collect than YouTube videos there is currently not enough training data available for the gaming scenario to train deep networks.

There is relatively little research into action prediction and the approaches vary widely in their goals, ranging from improving early action recognition, through generating synthetic sequences to predicting the action class before the action starts. The last is the most interesting and challenging especially in scenarios where there is no contextual information.

4.3 Methodology

Three algorithms are proposed in this section with the same core but different extensions to enable early action recognition, online action recognition and action prediction. The core of these methods are the proposed Clustered Spatio-Temporal Manifolds, which are compact style invariant models of the dynamics of human actions. They enable action classification in a continuous stream for early action detection in addition to the ability to follow the progress of the action so that the peak can be detected with low latency or even predicted.

The spatio-temporal manifolds are created by feature transformation to reduce style variance whilst still maintaining the temporal dynamics of the action. The first contribution is to generate key poses by clustering the manifolds and projecting the cluster centres. These key poses reduce computation time and in contrast to existing approaches are not selected from the training data but are style invariant as they are generated from the manifold. Another benefit of generating the key poses from the manifold is that they can be temporally ordered to form original action templates.

The action templates are effectively matched using DTW for execution rate invariance. The second contribution is to reduce the high observational latency of template matching by employing a sliding window approach to match template fragments with low latency. Peak key poses are the third contribution to enable explicit location of action peak for low latency action recognition and even action prediction.

The proposed methods all consist of the same training phase which generates the action templates and a unique testing phase that depends on the task: early action recognition, online action recognition and prediction.

4.3.1 TRAINING PHASE

To create the spatio-temporal action templates there are four key stages: feature transformation, clustering, ordering and projection (as shown in Figure 4-2). Human actions are represented by a large number of spatio-temporal features, so the first stage is to reduce the dimensionality. Temporal dynamics are critical for action recognition and prediction so a dimensionality reduction method that preserves the temporal structure of the data in the embedded space is employed. Temporal Laplacian Eigenmaps (TLE) [106] is a nonlinear feature transformation technique, that finds a new set of dimensions that are combinations of the original dimensions. TLE has previously been used for offline action recognition from video sequences [106] and is suited to any time series data that contains repetitions.

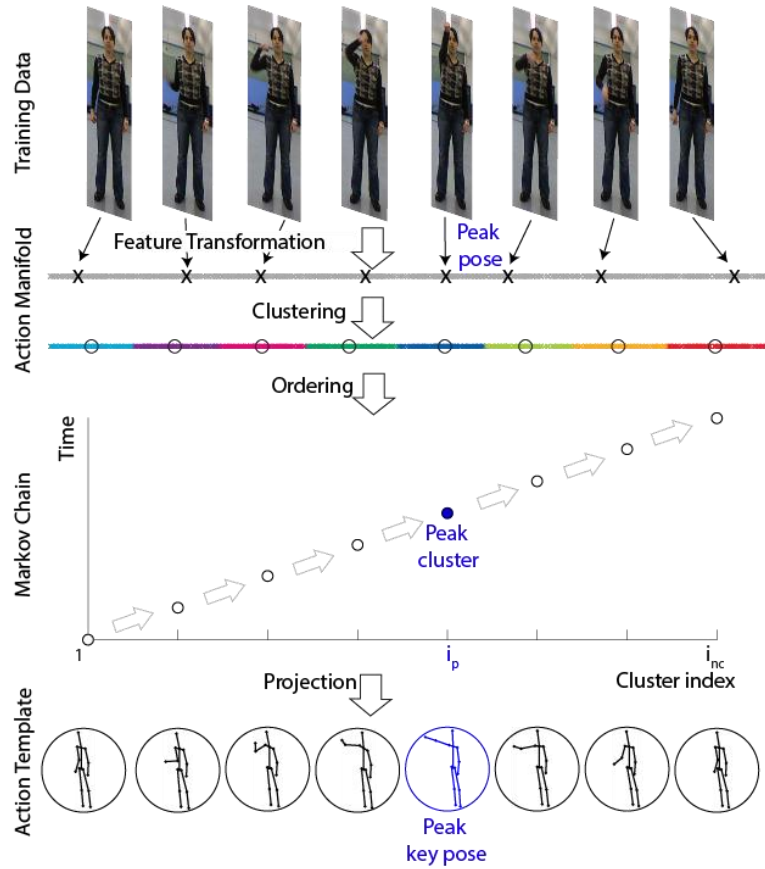


Figure 4-2 Action templates with four key stages: dimensionality reduction, clustering, ordering and projection.

Pose-based features can be viewpoint and anthropometric invariant as well as generated in real-time with a pose estimation method [6]. Normalising the skeleton poses and obtaining the joint angles removes the viewpoint variations. Similar to Lewandowski et al. [106] quaternions $f^q \in \mathbb{C}^4$ (as described in equation (2-9), were calculated for 13 joint angles for each skeleton pose, so each high dimensional feature vector has 52 dimensions. Although the proposed framework is evaluated with skeleton data, the method can also be applied to other time series data.

4.3.1.1 Dimensionality reduction

Temporal Laplacian Eigenmaps (TLE) algorithm [106] is an unsupervised nonlinear method for dimensionality reduction for time series data. Given a set of points

$\mathbf{X} = (\mathbf{x}_{i_r})_{(i_r=1\dots n_r)}$ distributed in high dimensional space ($\mathbf{x}_{i_r} \in \mathbb{R}^D$), which in this chapter $D = 52$, TLE is able to discover their low dimensional representation $\mathbf{Y} = (\mathbf{y}_{i_r})_{(i_r=1\dots n_r)}$, ($\mathbf{y}_{i_r} \in \mathbb{R}^d$) where $d \ll D$ and n_r is the number of points in the time series, as shown in Figure 4-4. The key feature of the embedded manifolds is that the temporal structure of the data is implicitly preserved in the low dimensional space.

Two neighbourhood graphs are constructed during the process of dimensionality reduction, one with adjacent temporal neighbours and another with geometrically similar neighbours, as illustrated in Figure 4-3. The adjacent temporal neighbours are the $2n_u$ closest points in the sequential order. Repetition neighbours are the n_v points similar to \mathbf{x}_{i_r} , extracted from repetitions of time series fragments, based on the minimum DTW distances using the Euclidean metric.

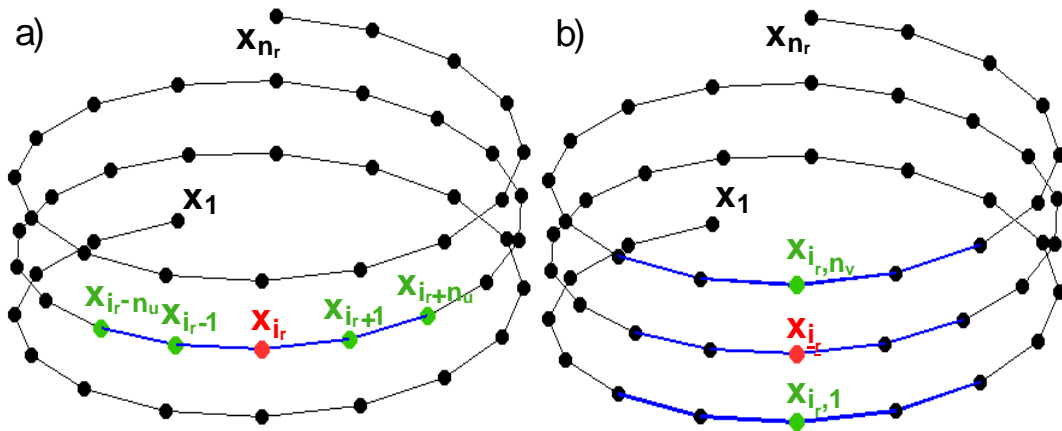


Figure 4-3 TLE: temporal neighbours (green dots) of a given data of a given data point, \mathbf{x}_{i_r} , (red dots) in a) adjacent and b) repetition graphs. [106]

Neighbourhood connections defined in the Laplacian graphs place neighbours from the high dimensional space nearby in the embedded space. Consequently, the temporal neighbours preserve the temporal structure and the spatial neighbours reduce style variability by aligning the time series in the embedded space.

The number of low dimensions d is a key parameter in the dimensionality reduction process but there is no consensus on how this should be determined. Cross validation is the simplest solution but computationally prohibitive. An estimate of the intrinsic dimension is the most computationally efficient solution and various approaches have been proposed. The Maximum Likelihood Estimation (MLE) [124] is a state of the art approach that applies the principle of maximum likelihood to the distances between close neighbours. MLE provides good estimates of the intrinsic dimensionality on simulated and real datasets, furthermore the source code is available⁴ making the implementation trivial.

4.3.1.2 Clustering

Clustering is then performed on the embedded manifold to remove redundant information. k -means [119] is applied to cluster the n_r low dimensional points \mathbf{Y} into n_c clusters $\mathbf{C} = \{\mathbf{c}_{i_c}\}_{(i_c=1\dots n_c)}$, $\mathbf{c}_{i_c} \in \mathbb{R}^d$, where $n_c \ll n_r$ as shown in Figure 4-4. Removing redundant information reduces the computational time of the subsequent action recognition and may also improve accuracy. Additionally, the clusters provide key points throughout an action's lifecycle that can be used to determine the current and even predict future

⁴ <http://www.mathworks.com/matlabcentral/fileexchange/40112-intrinsic-dimensionality-estimation-techniques/content/idEstimation/MLE.m>

progress. The number of clusters ($n_c = 35$) was set based on existing experiments for offline action recognition [106].

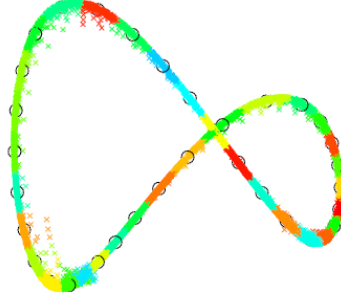


Figure 4-4 Clustered Spatio-Temporal manifold with the low dimensional points Y shown as points, coloured according to their cluster and the cluster centers C as black circles

4.3.1.3 Ordering

The clusters discovered by k -means are unordered so the temporal relationships from the embedded manifold are exploited to order the clusters. A first-order Markov chain [120] is constructed for each action to chronologically link the clusters. The Markov chain is defined by the transition matrix $\Lambda = (\lambda_{i_c j_c})_{(i_c=1\dots n_c, j_c=1\dots n_c)}$ where $\lambda_{i_c j_c}$ are the cluster transition probabilities. The transition probability from cluster i_c to cluster j_c is found by counting connections between temporal neighbours on the manifold. If transitions to the same cluster are ignored, the maximum transition probability for each cluster will represent the temporal order $\mathbf{o} = (o_{i_c})_{(i_c=1\dots n_c)}$ between the clusters as shown in Figure 4-5 and in Eq. (4-1), where $i_c \neq j_c$.

$$o_{i_c} = \arg \max_{j_c} (\lambda_{i_c j_c}) \quad (4-1)$$

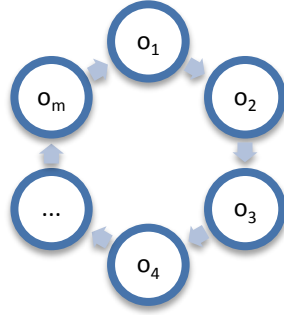


Figure 4-5 Cyclic clustered action manifold and highest probability transitions

4.3.1.4 Projection

Selecting key poses removes redundant information to improve classification accuracy and reduce the computational latency of template matching. In similar work, key poses were created by k -means clustering of training poses and selecting the closest pose to the cluster centre [61]. This reduces stylistic variation by selecting an average pose but as the key pose represents an individual some personal style will remain. To eliminate personal style the proposed method uses the clusters from the low dimensional action manifolds and projects their centres to the high dimensional space, using the Radial Basis Function Network (RBFN) mapping to generate new poses that are not present in the training dataset.

One limitation of TLE is that it places the n_r points in a low-dimensional space but it does not learn general mapping functions that will allow new points to be projected from the low to the high dimensional space. RBFN mapping functions allow projecting new data between the low and high dimensional spaces [106]. Using $\chi = \{y_{i_r}, x_{i_r}\}_{(i=i_r \dots n_r)}$ as a training set, RBFN⁵ are trained to learn the mapping between the low and the high dimensional space [106]. Then using the RBFN mappings the cluster centres \mathbf{C} are

⁵ Matlab function newrbe() was used to design an exact radial basis network with the training set χ .

projected into the high dimensional space to generate key poses $\mathbf{k} \in \mathbb{R}^D$, that form the action templates $\mathbf{K}^a = (\mathbf{k}_{i_o})_{(i_o=o_1 \dots o_{n_c})}$, by using the temporal order \mathbf{o} found between clusters, as illustrated in Figure 4-6.

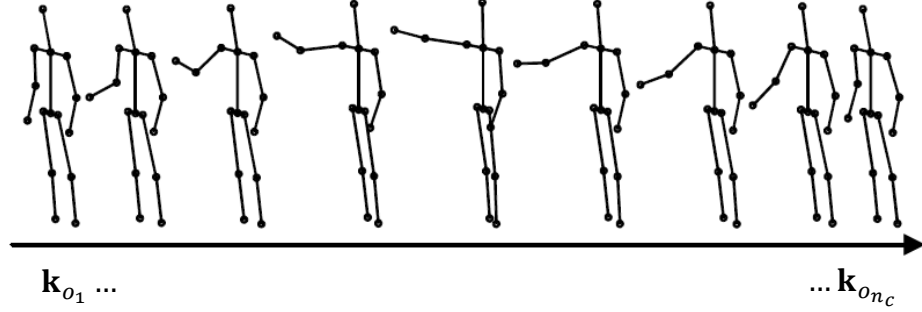


Figure 4-6 Right punch action template

4.3.1.5 Peak key pose selection

Key poses have been used with template matching for offline action classification [61] but the novel contribution is to select the key pose that represents the peak of the action for online classification. Peak key poses are a novel concept, which are related to but are not the same as action points [13] or canonical poses [12]. Peak key poses also represent a single pose but in contrast to existing approaches they are selected from the key poses rather than the training poses so they are invariant to individual style.

To select the peak key poses, the peak poses from the training data are matched against the key pose templates (for the definition of a peak pose see section 2.5.2.1). To increase robustness, fragments of poses are matched rather than single poses which enables actions with similar poses to be correctly matched based on the temporal pose history before the action peak. To extract a fragment f^G from a sequence of poses $\mathbf{S} = (\mathbf{s}_{i_s})_{(i_s=1 \dots n_s)}$, ($\mathbf{s}_{i_s} \in \mathbb{R}^D$), Eq. (4-2) is used, where n_f is the required number of poses in the fragment, i_f is the index of the last pose, n_s is the number of poses in the sequence and $i_f \leq n_s$ and $i_f - n_f \geq 0$.

$$f^G(\mathbf{S}, i_f) = (\mathbf{s}_{j_f})_{(j_f=i_f-n_f, i_f-n_f+1, \dots, i_f)} \quad (4-2)$$

Assuming the peak poses in the training data have been manually selected for each action and their indices stored: $\boldsymbol{\eta} = (\eta_{i_\eta})_{(i_\eta=1 \dots n_\eta)}$, the peak key poses are selected as follows: for each action a and for each peak pose index η_{i_η} , the matching key pose index i_m is found by minimising the DTW distance (described in the section 4.3.1.5.1) between the peak pose fragment from the training poses \mathbf{X} and the key pose fragments from the action templates \mathbf{K}^a , as in Eq. (4-3 and shown in Figure 4-7.

$$i_m(\eta_{i_\eta}) = \arg \min_{i_k \in 1 \dots n_c} f^D(f^G(\mathbf{X}, \eta_{i_\eta}), f^G(\mathbf{K}^a, i_k)) \quad (4-3)$$

To find the peak key pose index i_p for the action a , $\boldsymbol{\zeta}$ is initialised ($\boldsymbol{\zeta} = 0_{1, n_c}$) and each time a matching key pose index i_m is found ζ_{i_m} is incremented. The peak key pose index i_p for the action is the key pose index, with the maximum number of matches ($i_p(a) = \arg \max \boldsymbol{\zeta}$).



Figure 4-7 Template fragment matching: peak pose fragment (left), matched key pose fragment (right)

4.3.1.5.1 Dynamic Time Warping Algorithm

DTW [121] is a well-known algorithm for matching time-series data that allows “elastic” transformation to gain execution rate invariance, as illustrated in Figure 4-8.

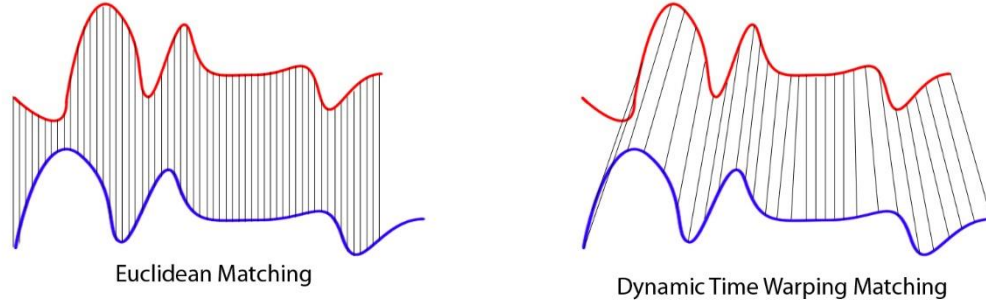


Figure 4-8 Comparison of the Euclidean and DTW matching. (a) The Euclidean matching compares the samples at the same time instants, whereas (b) the DTW measure compares samples with similar shapes to minimise the distance [122].

The similarity of any two time series data, a query sequence $\mathbf{Q} = (\mathbf{q}_{i_q})_{(i_q=1\dots n_q)}$, $\mathbf{q}_{i_q} \in \mathbb{R}^D$ and a reference sequence $\mathbf{R} = (\mathbf{r}_{i_r})_{(i_r=1\dots n_r)}$, $\mathbf{r}_{i_r} \in \mathbb{R}^D$ can be computed using the standard DTW distance metric as follows.

Initially, a local dissimilarity function is used, in this work Euclidean distance f^δ is employed, to create a cross-distance matrix $\mathbf{\Gamma} \in \mathbb{R}^{n_q \times n_r}$ between \mathbf{Q} and \mathbf{R} . Specifically, for any pair of \mathbf{q}_{i_q} and \mathbf{r}_{i_r} :

$$\mathbf{\Gamma}(i_q, i_r) = f^\delta(\mathbf{q}_{i_q}, \mathbf{r}_{i_r}) \quad (4-4)$$

Then warping paths are created so that the distortion along the matrix can be minimised, as demonstrated in Figure 4-9.

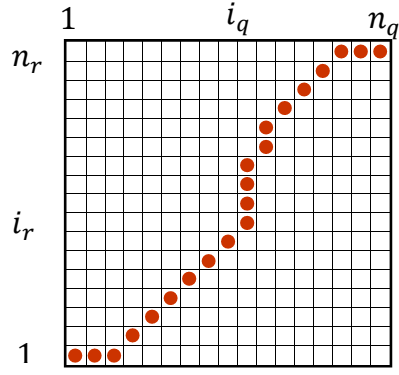


Figure 4-9 Cross-distance matrix Γ between sequences \mathbf{Q} and \mathbf{R} , showing the optimum warping path \mathbf{L} , that minimises the distance between \mathbf{Q} and \mathbf{R} [123]

A warping path \mathbf{L}_{i_L} ($i_L = 1 \dots n_L$) is defined as:

$$\mathbf{L}(i_L) = (f_Q^L(i_L), f_R^L(i_L)) \quad (4-5)$$

where $f_Q^L \in \{1 \dots n_q\}$, $f_R^L \in \{1 \dots n_r\}$ are functions that stretch the time axis of \mathbf{Q} and \mathbf{R} respectively, and n_L is the length of the path.

For any \mathbf{L} the accumulated distortion on the path δ , is calculated as:

$$\delta(\mathbf{Q}, \mathbf{R}) = \sum_{i_L=1}^{n_L} \Gamma(f_Q^L(i_L), f_R^L(i_L)) \quad (4-6)$$

Finally, the DTW distance is calculated by choosing the path \mathbf{L} that stretches the time index as to minimise the Euclidean pair-wise distance between \mathbf{Q} and \mathbf{R} as described below:

$$f^D(\mathbf{Q}, \mathbf{R}) = \min_L \delta(\mathbf{Q}, \mathbf{R}) \quad (4-7)$$

4.3.2 TESTING PHASE

The testing stages depend on the task (online, early action recognition or prediction) but there is a common base of online template matching with Dynamic Time Warping for execution rate invariance [121]. Existing approaches for offline action recognition use the entire action template which inherently has high latency [61]. To enable online recognition a sliding window approach matches recent test poses with action template fragments, as illustrated in Figure 4-10.

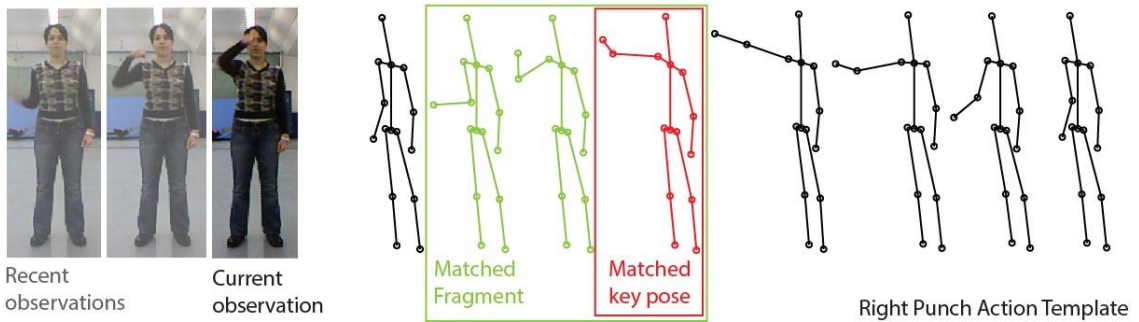


Figure 4-10 Template fragment matching: observed test poses and matched action template

4.3.2.1 Early Action Recognition

Early action recognition aims to determine the action class, based on as few observations as possible, even when only part of the action has been seen. The majority of research in this area is on activities from video sequences [111]–[117]. In existing work [111]–[114], [116], [117] the sequences are pre-segmented to contain a single activity and evaluation is performed at different observation ratios, from 0.1 to 1. So an observation ratio of 0.5 represents the first half of the action and an observation of 1 is the conventional offline action recognition approach. Since the test sequences in this thesis are not pre-segmented, as they consider the real-time application of action recognition, the proposed method assigns an action label for each frame in a continuous stream using a sliding window. The

sliding window contains the recent and current observations from the test stream to ensure no future information is incorporated into the method.

The proposed method for early action recognition is online template matching where the current test pose fragment is matched against sliding windows on each of the different action templates to obtain key pose fragments. The action class of the most similar key pose fragment is used as the action classification label for the current frame. DTW allows "elastic" transformation so actions in the test stream performed at different speeds to the action templates can be matched. Formally, early action recognition for each sequence of test poses $\mathbf{Z} = (\mathbf{z}_{i_t})_{(i_t=1\dots n_t)}$, $\mathbf{z}_{i_t} \in \mathbb{R}^D$ is performed as follows: to find the action classification label a' for the current pose \mathbf{z}_{i_t} , the normalized DTW distance between the test pose fragment and test poses from all the action templates are minimised according to:

$$a^*(i_t) = \arg \min_{a \in 1\dots A} (\min_{i_k \in n_f \dots n_c} f^D(f^G(\mathbf{Z}, i_t), f^G(\mathbf{K}^a, i_k))) \quad (4-8)$$

The minimum normalised DTW distances for each frame of a sample sequence in the G3D dataset against each action template are shown in Figure 4-11. The lowest distance over all the actions represents the matched action class as illustrated in Figure 4-11.

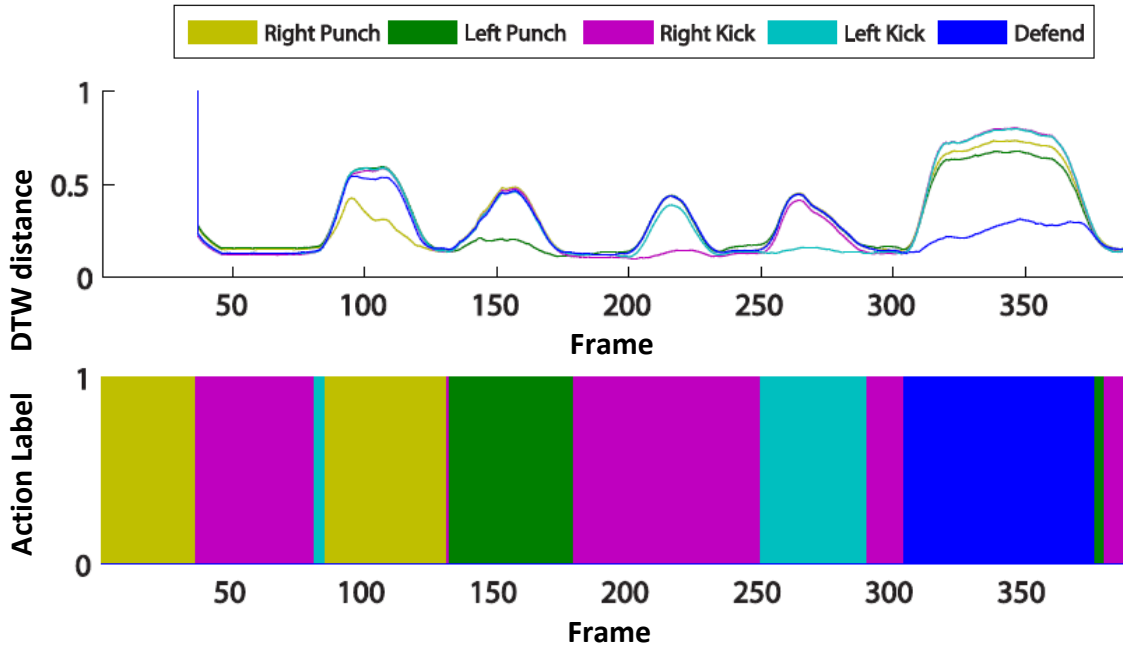


Figure 4-11 (Top) Normalised DTW distance for each frame (Bottom) Action classification label for each frame. At this stage all frames are classified as an action, even the neutral frames. To overcome this limitation action points are detected at the next stage to only classify the peak frame of each action.

4.3.2.2 Online Action Recognition

To enable continuous action recognition to be suitable for real-world applications a single point needs to be identified for each action, rather than classifying individual frames. For this reason action points [13] were introduced which are action labels with temporal anchors. Action points are used in this section to detect the peak of the action and each action point is represented by an action label a and a timestamp t_d .

Combining online template matching with peak key poses enables online action recognition with high accuracy and very low latency. To explicitly locate the moment where an action reaches its peak, poses are followed as they progress through the early

stages of the action and the peak is detected by comparing the matched poses with the peak key pose.

For each test pose stream $\mathbf{Z} = (\mathbf{z}_{i_t})_{(i_t=1\dots n_t)}$ online action recognition is performed as follows: the first step is to find the action classification label a^* for the current test pose \mathbf{z}_{i_t} using the online template matching described in section 4.3.2.1. The second step is to determine the progress of the current action by locating the key pose on the action template that is the closest match to the current test pose. To find the matching key pose index i_m for the current test pose index i_t , the normalised DTW distance for the test pose fragment against test poses from all the action templates are minimised according to Eq. (4-9).

$$i_m(i_t, a^*) = \arg \min_{i_k \in n_f \dots n_c} f^D (f^G(\mathbf{Z}, i_t), f^G(\mathbf{K}^a, i_k)) \quad (4-9)$$

The third step is to determine if the action has reached its peak. The peak key pose can be conceptually projected onto the clustered action manifold to illustrate that the peak pose is detected when the matched key pose index i_m is the same as (or slightly greater) than the peak key pose index i_p (as shown in Figure 4-13) and is formally defined in Eq. (4-10).

$$\varphi(i_m, i_p, n_k) = \begin{cases} 1 & \text{if } 0 \leq i_m - i_p \leq n_k \\ 0 & \text{otherwise} \end{cases} \quad (4-10)$$

where i_m is the matched key pose index for the current test pose \mathbf{z}_{i_t} , i_p is the index of the peak key pose and n_k is the maximum number of poses after i_p allowed to detect a peak pose. This can also be illustrated in graph format as shown in Figure 4-12 where the key pose index i_k , is plotted for each frame and where this cluster index line crosses the peak key pose line (dotted horizontal line) for the corresponding action an action point is detected (o).

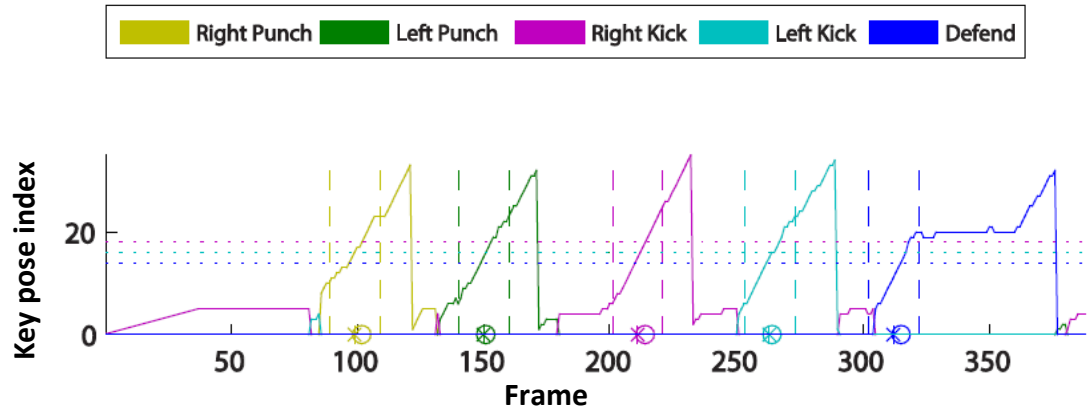


Figure 4-12 Clustered Action Manifold cluster indices for each frame with ground truth action points (*) and detected action points (o)

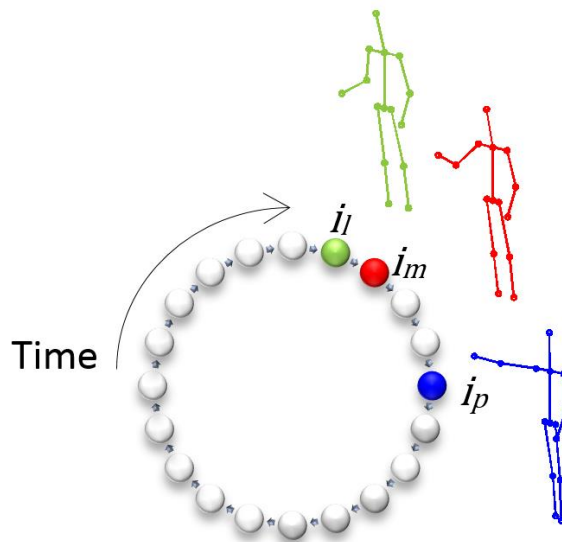


Figure 4-13 Right Punch Clustered Action Manifold with peak key pose index i_p with matched key pose index i_m and last matched key pose index i_l .

4.3.2.3 Action Prediction

There are relatively few approaches to action prediction and the approaches vary widely in their goals, ranging from improving early action recognition [114], through generating synthetic sequences [67] to predicting the action class before the action starts [118]. In

this section a novel approach to action prediction is proposed where action peaks are predicted in a continuous stream before the peak has been observed. Action points are used in this section to represent the action peak and each prediction is represented by an action label a , a timestamp for the predicted action peak t_p to determine the timeliness of the prediction and a timestamp at the time the prediction was made t_d to measure how far in advance the predictions can be accurately made.

For each test pose stream $\mathbf{Z} = (\mathbf{z}_{i_t})_{(i_t=1\dots n_t)}$, online prediction is performed as follows: the first step is to find the action classification label a^* for the current test pose \mathbf{z}_{i_t} using the online template matching, using Eq. (4-8) described in section 4.3.2.1. The second step is to determine the progress of the current action by locating the key pose index i_m on the action template that is the closest match to the current test pose, using Eq. (4-9), described in section 4.3.2.2. The third step is to store the n_m most recent sequential pose matches of the current action class a' to maintain the history of the action progress $\boldsymbol{\theta} = (i_m(\boldsymbol{\theta}_t, a^*))_{(\boldsymbol{\theta}_t=i_t-n_m\dots i_t)}$.

The fourth step is to perform the action prediction using the recent action history and regression. Although the dynamics of human actions are nonlinear in the high dimensional space, their embedded clustered spatio-temporal representation is locally linear. This is demonstrated in Figure 4-12, which shows time along the horizontal axis and the key pose index along the vertical axis. Therefore, linear regression is proposed to quickly predict the action peak. For the current test pose \mathbf{z}_{i_t} , when n_m sequential key pose matches of the same action class a' have been observed, their key pose indices $\boldsymbol{\theta}$, are fitted to a straight line by least-squares regression and the equation of the line is derived by Eq. (4-11).

$$(\alpha'(a^*, i_t), \beta'(a^*, i_t)) = \underset{\alpha, \beta}{\operatorname{argmin}} \sum_{\theta_t=i_t-n_m}^{i_t} (i_m(\boldsymbol{\theta}_t, a^*) - \alpha - \beta t_i)^2 \quad (4-11)$$

where α' is the y-intercept of the least squares line and β' is the gradient.

The least squares line is extended to predict future poses using the derived equation. The peak key pose line is a horizontal line with a y-intercept of the peak key pose index i_p for the corresponding action. The point where the extended least squares line intersects the peak pose horizontal line is the estimated time t_p of the peak with time of detection $t_d = i_t$ (see Figure 4-14). Extreme cases are excluded by setting thresholds on the minimum and maximum gradient of the slope. The gradient of the line represents the execution speed of the current test subject and is independent on the speed of subjects observed in the training set. Fast subjects will match key poses in the action template faster than slower subjects resulting in a steeper slope. A key benefit of the proposed temporal prediction is that it is invariant to execution speed as it utilises the gradient of the slope which is formed based on the speed of the current subject.

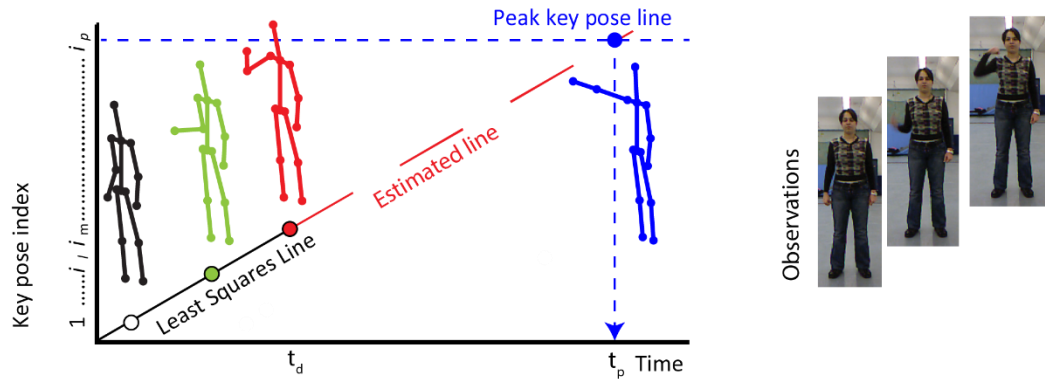


Figure 4-14 Linear regression at time t_d to predict the time t_p at which the partially observed action will reach its peak.

The core of the methods proposed in this chapter are based on style invariant spatio-temporal action templates that can be efficiently matched with DTW for execution rate invariance for early action recognition and combined with peak key poses for reliable

online action recognition. Furthermore, the spatio-temporal templates are suitable for fast linear regression to enable action prediction.

4.4 Results

4.4.1 DATASETS

The performance of the proposed algorithms are evaluated using publicly available datasets designed specifically for real time action recognition: G3D (introduced in section 3.4) and MSRC-12 [71] (summarised in 2.5.1.3). Both datasets provide sequences of skeleton data captured using the Kinect pose estimation pipeline at 30fps. Action point annotations of the peak poses are available for the MSRC-12 dataset and G3D dataset to precisely measure the latency of action recognition methods as well as the accuracy (described in section 2.5.2.1). Comparative studies are conducted separately for performance in the specific tasks of online action recognition, early action recognition and action prediction.

A “leave-person(s) out” cross validation protocol (described in section 2.5.3) was used where a set of people is removed to obtain the minimum test set that contains instances of all actions. For the MSRC-12 dataset this may be more than one actor as not every actor performs all the actions for the video + text modality. For the G3D dataset this is simply one actor as all actors perform all the actions. The remaining large set is used for the training. This process is repeated 10 times with different subsets of people to obtain the general performance. The total number of training and testing instances for each dataset used in the following experiments is shown in Table 3-3.

4.4.2 ONLINE ACTION RECOGNITION

4.4.2.1 Performance Metrics

For a fair comparison with existing approaches the same latency aware metric was used as initially proposed by [71] and later adopted by [73]. The detected action points are compared to the ground truth action points using the action point metric (described in section 2.5.2.2.3) to obtain a mean action point F_1 -score at a fixed latency Δ , where $\Delta = 333ms$.

4.4.2.2 Comparative Study

Clustered Spatio-Temporal Manifolds that are proposed in this chapter are evaluated against the three algorithms discussed in the previous chapter: **Random Forests**, **AdaBoost** and **Dynamic Feature Selection** (see section 3.5.3 for more details on the algorithms and parameters).

Clustered Spatio-Temporal Manifolds: To learn manifolds for each action the algorithm requires manual segmentation of the start and end of the action and all frames are used for training. It is important to note that this segmentation is only required in the training phase and is not performed in the testing phase. The annotated action points are additionally used to learn the peak key poses. The parameters for the proposed approach are the target dimensionality d , the number of clusters n_c in the manifold, the fragment size n_f and the number of clusters n_k that can be skipped at the peak. The target dimensionality ($d = 3$), was determined by applying the maximum likelihood intrinsic dimensionality estimator [124]. The number of clusters ($n_c = 35$) was set based on existing experiments for offline action recognition [106]. The number of poses in the fragment ($n_f = 10$) was set to match the size of the smoothing window S in the previous chapter. To find the value for n_k an exhaustive search was performed within the training

set to maximise the F-score. The optimum value is ($n_k = 0$) for the MSRC-12 and ($n_k = 14$) for the G3D dataset. No smoothing window was applied to the frame based distance results, and the final output from the algorithm was the detected action points for each sequence.

4.4.2.3 Online Recognition Results

Table 4-1 Action Point F1-scores at $\Delta=333$ ms, the average and standard deviations over ten leave-persons-out runs are shown. The results shown in italics were published by the method authors, all other results were generated by my own implementations.

	Random Forest [71]	<i>Random Forest</i>	<i>Ada Boost</i>	<i>Dynamic Feature Selection</i>	SVM-RFE [73]	<i>Clustered Spatio-Temporal Manifolds</i>
Feature Vector	<i>Multi-frame</i>	Single-frame	Single-frame	Single-frame	<i>Multi-frame</i>	Multi-frame
G3D	-	0.894 (0.155)	0.884 (0.147)	0.910 (0.128)	<i>0.937</i>	0.978 (0.026)
MSRC-12	<i>0.765</i> <i>(0.070)</i>	0.619 (0.148)	0.675 (0.156)	0.744 (0.270)	-	0.773 (0.124)

The experimental results show that the proposed Clustered Spatio-Temporal Manifolds achieved state-of-the-art accuracy for online action recognition with low latency. The experiments demonstrate the proposed method achieves the highest accuracy, 77.3% and 97.8% on the MSRC-12 and G3D datasets respectively (see Table 4-1 for a comparison with existing approaches). A breakdown of the results by action shows increased

performance of the proposed method over the comparative methods in every action in the G3D dataset (see Figure 4-16). The graphs show the methods' action point F_1 -score (defined in 2.5.2.2.3) for each action in the dataset and the average across all actions. There is also considerable improvement on actions in the MSRC-12 dataset with similar poses (e.g. change weapon and night goggles) which were difficult to discriminate without the temporal history (see Figure 4-17). The higher accuracy of the proposed method may be attributed to the improved execution rate invariance gained by matching template fragments with DTW instead of fixed size feature windows as used by Fothergill et al. [71] and Sharaf et al. [73]. Although both Zhao et al. [72] and Ellis et al. [12] also perform online action recognition they use the non-gaming actions in the MSRC-12 dataset so a comparison with their accuracy results is not possible.

The proposed method runs in real time (60fps) with low average observational latency of 2 frames (67ms). The observational latency of the proposed approach is very low in comparison to Zhao et al. that have an observation latency of 830-1500ms. The significantly lower observation latency of the proposed method was achieved by using considerably less frames in the sliding window than Zhao et al. in conjunction with the explicit identification of the peak key pose.

Figure 4-15 is an example sequence from the G3D dataset which illustrates the low latency that is achieved by the explicit peak pose (dotted horizontal line). The ground truth action points (*) and the vertical dashed lines represent the time window ($\pm\Delta$) where the action point is deemed to be correctly detected. The detected action points (o) show that the proposed approach has a very low latency and high accuracy.



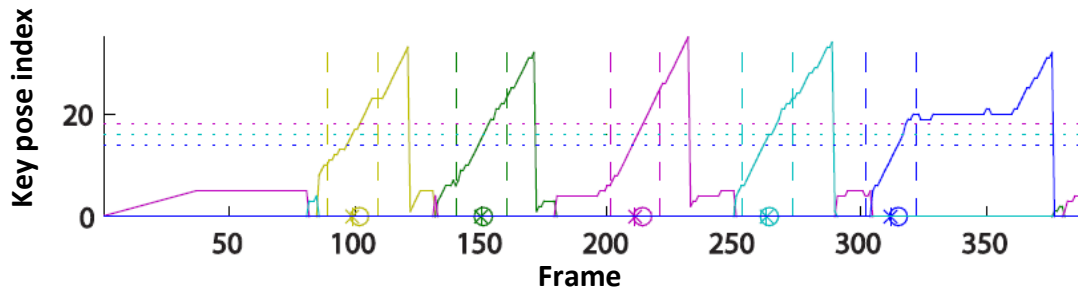


Figure 4-15 Clustered Action Manifold cluster indices for each frame with ground truth action points (*) and detected action points (o)

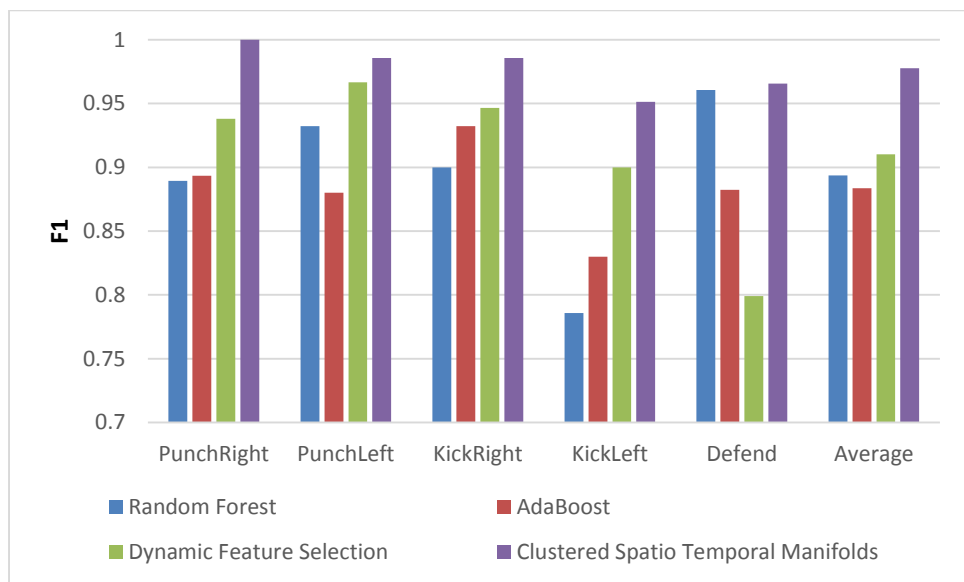


Figure 4-16 G3D Fighting Online Action Recognition Results by Action

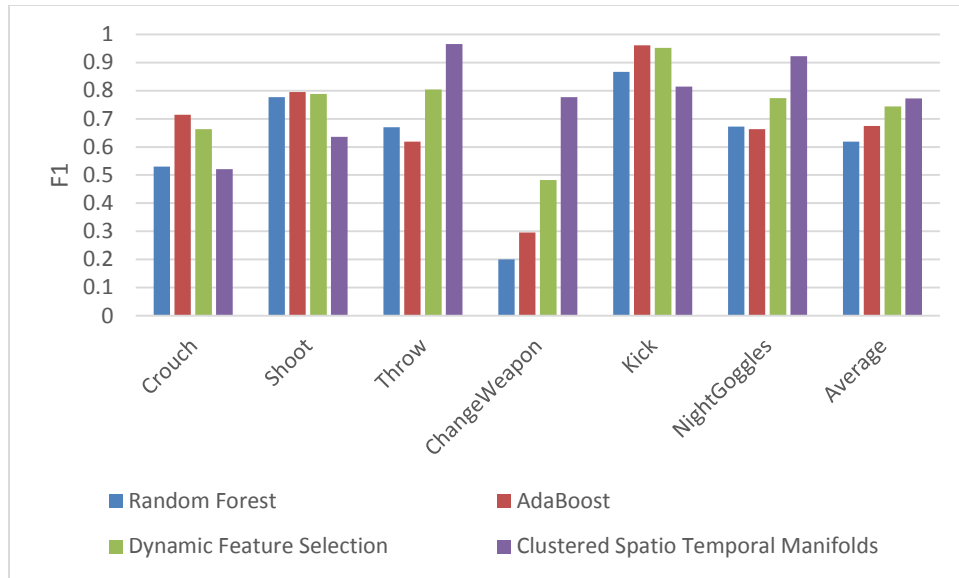


Figure 4-17 MSRC-12 Fighting Online Action Recognition Results by Action

4.4.3 EARLY ACTION RECOGNITION

The existing work on early recognition has been done in the video modality on activities that were pre-segmented [67], [111]–[117] and therefore a direct comparison is not feasible. Instead pose-based approaches for online action recognition have been adapted for early action recognition in a continuous stream to evaluate their effectiveness at a similar task.

4.4.3.1 Performance Metrics

In the video domain, Hoai and De la Torre [115] recorded the F1-scores as the action of interest unrolled from 0.1 to 1 and refer to this as the F1-score curve. However, the percentage of action observed can only be calculated for sequences that have been pre-segmented to contain a single action. Also, in the video domain, Lan et al. [116] use the temporal distance (in frames) to report accuracy. In real world scenarios such as gaming the videos are not pre-segmented, instead action points are provided as temporal anchors

and the latter frame-based metric seems the most appropriate measurement. For example, the methods' performance at a temporal stage -20 describes the classification accuracy given all of the testing frames up to 20 frames before the action peak.

4.4.3.2 Comparative Study

The three algorithms evaluated in the previous chapter are adapted for early action recognition: **Random Forests**, **AdaBoost**, **Dynamic Feature Selection** (see section 3.5.3 for more details on the algorithms parameters). Before the final detection step these algorithms output a frame based classification that is used for early action recognition. Similarly, **Clustered Spatio-Temporal Manifolds**, the algorithm proposed in this chapter, also outputs a frame-based classification before the final action detection (see section 4.4.2.2 for more details on the algorithm parameters).

4.4.3.3 Early Action Recognition Results

The proposed method significantly outperforms all of the comparative methods at all temporal stages across both datasets as illustrated in Figure 4-18 and Figure 4-19. The graphs show the methods' frame F_1 -score (defined in 2.5.2.2.3) at different temporal stages from 20 frames before the action peak -20 to the peak of the action 0. The proposed method reaches 80% accuracy 16 and 10 frames before the action peak on the MSRC-12 and G3D datasets respectively, whereas the comparative methods achieve less than 30% accuracy at similar stages. The significant improvement in classification accuracy especially in the early stages of the action can be attributed to the proposed temporal models. The majority of failure cases were in the neutral or very early stage of the action as shown in Table 4-2 and Table 4-3 where the action is ambiguous. The proposed method achieves 97.8% and 100% accuracy on the MSRC-12 and G3D dataset respectively at the action peak. The failure cases at the action peak in the MSRC-12 dataset were mainly due to the Change Weapon action which in some cases appears very similar to the neutral pose

at the peak as illustrated in Table 4-3. The action peak frame based F1 results are higher than the action point F1 scores reported in the previous section because the frame based metric used in this section is only concerned with classification and not the temporal detection of the action peak of which the latter is a more difficult task. Finally, the proposed approach obtains 76.3% on the MSRC-12 dataset 20 frames before the peak which may be attributed to the fact that the MSRC-12 actions typically have longer onset than G3D actions, especially the Change Weapon, Shoot and Throw actions.

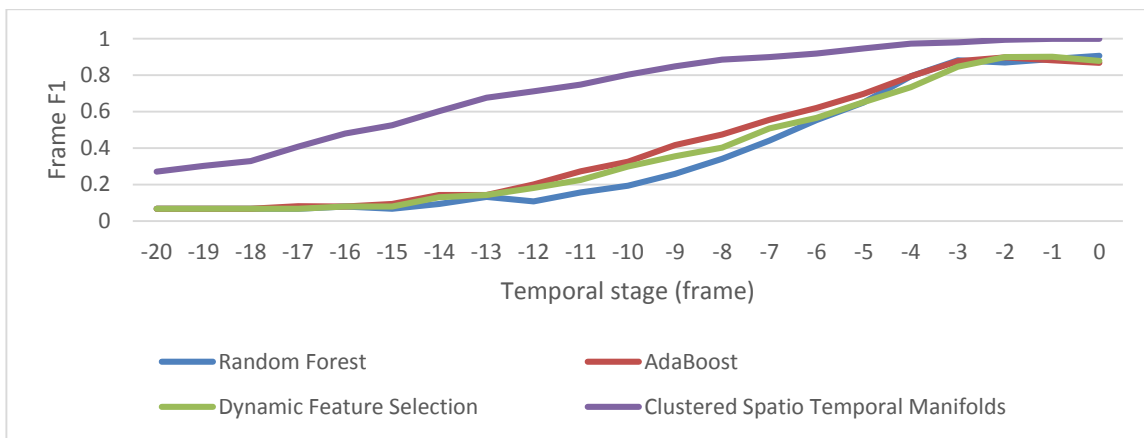


Figure 4-18 G3D Frame F1-scores, the average over ten leave-persons-out runs are shown

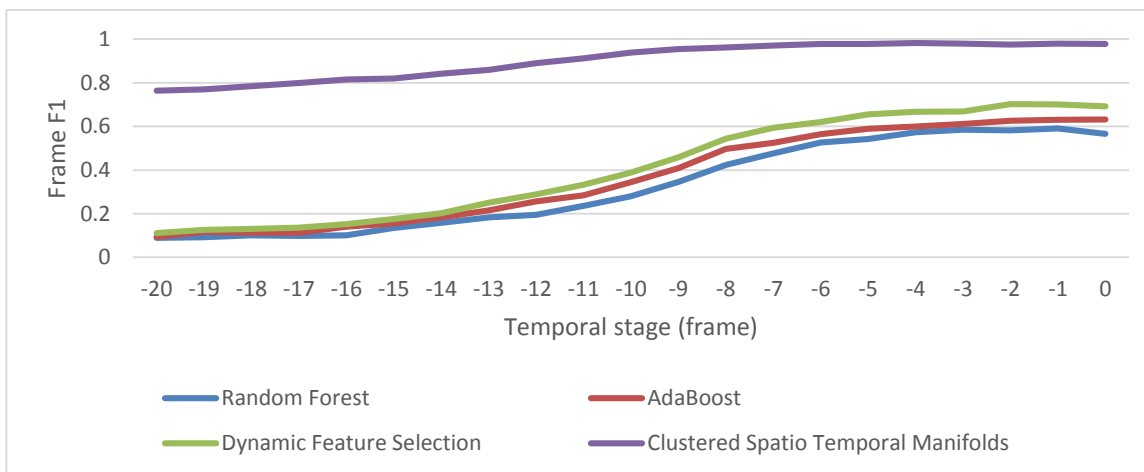


Figure 4-19 MSRC-12 Frame F1-scores, the average over ten leave-persons-out runs are shown

Table 4-2 G3D Temporal Frame Based Results: Correct classifications are shown in green and failure cases in red. The majority of failure cases were in the neutral or very early stage of the action.







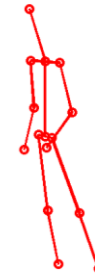







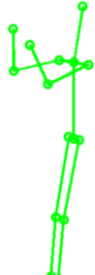




















Frame	-20	-15	-10	-5	-0
Trial_22_s 1					
Actual	Right Punch	Right Punch	Right Punch	Right Punch	Right Punch
Detected	Right Punch	Right Punch	Right Punch	Right Punch	Right Punch
Trial_86_s 2					
Actual	Left Kick	Left Kick	Left Kick	Left Kick	Left Kick
Detected	Right Kick	Left Punch	Left Punch	Left Kick	Left Kick
Trial_171_s s3					
Actual	Defend	Defend	Defend	Defend	Defend
Detected	Kick Left	Kick Right	Kick Right	Defend	Defend

Table 4-3 MSRC-12 Temporal Frame Based Results: Correct classifications are shown in green and failure cases in red. The majority of failure cases were in the neutral or very early stage of the action but there were also some cases at the peak of the action as in some cases the peak pose for Change Weapon is very similar to the neutral pose.

Frame	-20	-15	-10	-5	-0
Trial_p2_1 _8a_p03_f 556-571					
Actual	Throw	Throw	Throw	Throw	Throw
Detected	Throw	Throw	Throw	Throw	Throw
Trial_p2_2 _12A_p25 _f377-397					
Actual	Kick	Kick	Kick	Kick	Kick
Detected	Shoot	Shoot	Kick	Kick	Kick
Trial_p2_2 _6A_p26_ f447-467					
Actual	Shoot	Shoot	Shoot	Shoot	Shoot
Detected	Shoot	Shoot	Shoot	Shoot	Shoot

Trial_p3_2 _10A_p02 _f1123- 1143					
Actual	Change Weapon	Change Weapon	Change Weapon	Change Weapon	Change Weapon
Detected	Change Weapon	Change Weapon	Shoot	Shoot	Kick

4.4.4 ACTION PREDICTION

The existing work on action prediction has also been performed in the video modality and therefore a comparison is not feasible. Instead, the comparative pose-based approaches for early action recognition have been extended with the same linear regression as described in 4.3.2.3 to evaluate their effectiveness at action prediction.

4.4.4.1 Performance Metrics

Huang and Kitani [125] use average frame distance (AFD) to evaluate the accuracy of their predicted poses. AFD is a good measure of the spatial prediction but does not explicitly measure the latency of the temporal prediction. In the proposed method the emphasis is on the temporal prediction of the peak pose, to the best of my knowledge there are no existing metrics for predicting the peak of the action. However, the Action Point F_1 -score (defined in section 2.5.2.2.3) is a latency-aware metric for online action recognition that can be adapted to measure the accuracy of the predicted action points t_{p_a} instead of measuring the accuracy of the detected action points t_{d_a} , by modifying Eq. (2-18) to Eq. (4-12).

$$\Phi_p(t_{p_a}, t_{g_a}, \Delta) = \begin{cases} 1 & \text{if } |t_{g_a} - t_{p_a}| \leq \Delta \\ 0 & \text{otherwise} \end{cases} \quad (4-12)$$

For a new test sequence, the arrival of data can be simulated and the predicted action point F_1 -scores recorded. The predicted action point metric measures instances rather than frame based predictions so it will be referred to as the action point F_1 -score curve.

4.4.4.2 Comparative Study

To extend the early recognition algorithms with linear regression, the methods need to output a certainty measure for each action at each frame, as illustrated in Figure 3-3. This is the case for two out of the three algorithms evaluated in the previous section: **AdaBoost** and **Dynamic Feature Selection**. **Random Forests** could not be adapted for prediction as the frame based result was a classification. **Clustered Spatio-Temporal Manifolds**, the algorithm proposed in this chapter, outputs a cluster index for each frame which can be used in conjunction with the peak key pose index for prediction. The parameter required for prediction is the number of sequential frames for the linear regression. An exhaustive search was performed on the training set and the optimum result for AdaBoost and Dynamic Feature Selection was ($n_m = 2$) and for the Clustered Spatio-Temporal Manifolds the optimum value was ($n_m = 6$).

4.4.4.3 Action Prediction Results

To measure how precisely the peak of the action can be predicted for all subjects the action point F_1 metric was captured as the continuous stream progressed. The proposed method significantly outperforms all of the comparative methods at all temporal stages on the G3D dataset as illustrated in Figure 4-20 and across the majority of temporal stages on the MSRC-12 dataset as illustrated in Figure 4-21. The graphs show the methods' action point F_1 -score (defined in 2.5.2.2.3) at different temporal stages from 20 frames before the action peak -20 to the peak of the action 0. The proposed method works in a continuous

stream, where the prediction is made as early as possible and early incorrect predictions decrease the final F_1 -score. Even at the action peak prediction accuracy is less than online action recognition as the latter approach delays the detection until the peak has been observed. The proposed method reaches 38.1% and 45.6% 10 frames before the action peak. Predicting the point in time at which the peak pose will occur is a much more complex task than early detection of the action class or online action recognition, so a decrease in performance is expected. This is supported by the fact that the comparative approaches only reached a maximum of 24% at 10 frames before the action peak. The improvement in prediction of the proposed method can be attributed to the style invariant temporal model that is learnt for each action which includes explicit identification of a generic peak key pose.

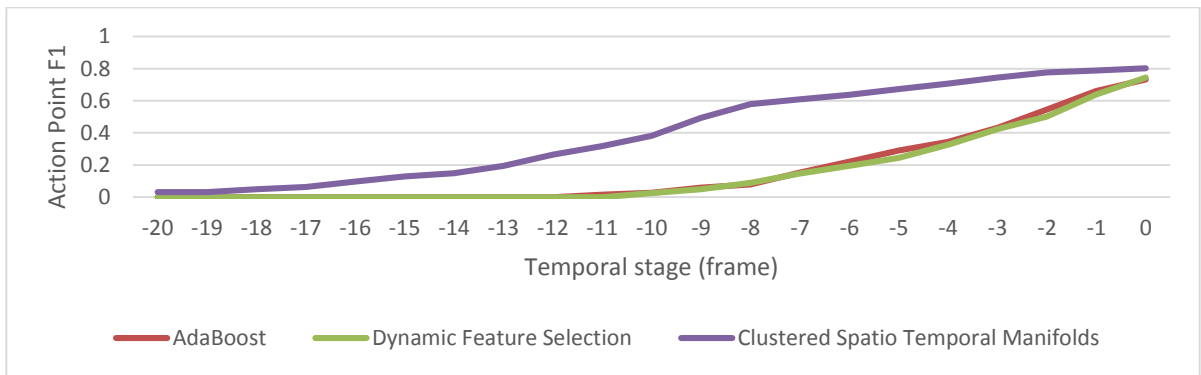


Figure 4-20 G3D Action Point F1-score curves, the average over ten leave-persons-out runs are shown

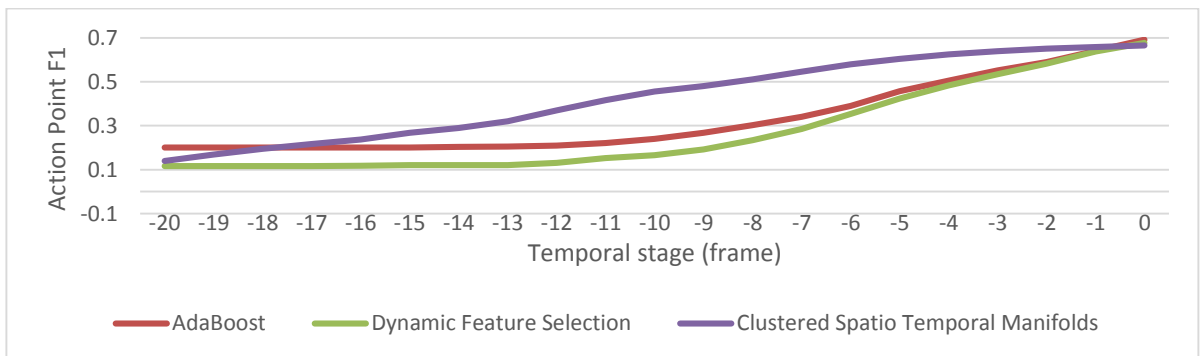


Figure 4-21 MSRC-12 Action Point F1-score curves, the average over ten leave-persons-out runs are shown

A key benefit of the proposed prediction framework is that it is invariant to execution speed; the experimental results show that the regression line for a faster subject has a steeper gradient than the regression line for slower subject performing the same action and in both cases the action peak is detected correctly (see Figure 4-22).

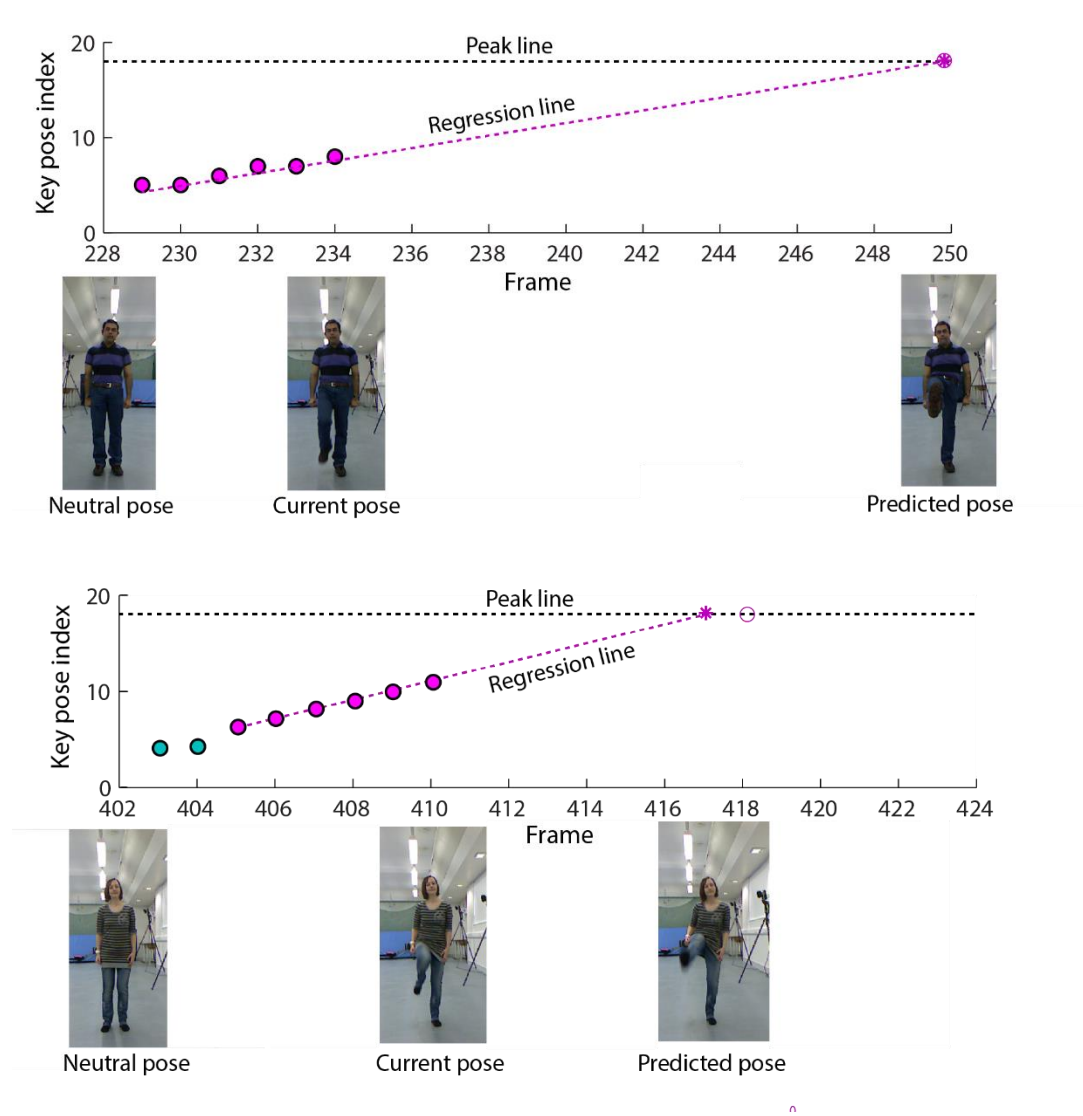


Figure 4-22 Two subjects performing a (right kick), at different speeds (classified right kick poses •, classified left kick poses •, ground truth peak pose *, predicted peak pose ○)

4.5 Summary

The core of the proposed methods in this chapter are the Clustered Spatio-Temporal Manifolds, which are compact style invariant models of the complex dynamics of human actions. They enable action classification in a continuous stream for early action detection in addition to the ability to track the progress of the action so that the peak can be detected with low latency or even predicted.

The spatio-temporal manifolds were created by feature transformation to reduce style variance whilst still maintaining the temporal dynamics of the action. The manifolds were clustered and the cluster centres projected to create key poses which reduced computation time and the key poses were temporally ordered to create action templates.

The action templates were effectively matched using DTW for execution rate invariance. To reduce the high observational latency of template matching a sliding window approach was used to match template fragments with low latency. The proposed approach achieved high accuracy for early action recognition and in contrast to existing approaches can operate in a continuous stream.

Peak key poses were introduced to explicitly locate the moment where an action reaches its peak which enabled low latency recognition before the completion of the action. Experimental results on publicly available gaming action datasets demonstrate state-of-the-art high accuracy with very low latency.

This chapter also introduced the novel and challenging problem of predicting the action peak in a continuous stream. The proposed solution integrates the recent action progress history with regression for fast estimation of the peak. Experiments on public action recognition datasets showed that the proposed method outperforms the comparative

approaches and makes reasonable predictions even when there is a significant variation in the style and execution rate of the subject.

CHAPTER 5

COMPOUND ACTION RECOGNITION USING HIERARCHICAL TRANSFER LEARNING

5.1 Introduction

The previous chapter reported high accuracy and low latency for action recognition but as the existing gaming datasets were recorded with scripted scenarios the actions are temporally isolated and easy to segment. In contrast, this chapter introduces a novel game-sourcing approach for recording realistic actions where the subjects are recorded whilst playing Kinect Sports [126], a commercial video game. Sports games introduced the element of competition so the actions captured were more realistic and challenging in comparison to scripted actions. Subjects in the new game-sourced dataset (G3Di) performed multiple actions in quick succession which resulted in actions with indistinctive boundaries. When multiple actions are performed in quick succession movements from different actions may temporally overlap, which are termed in this thesis as compound actions (see section 5.4 for examples).

Furthermore, none of the existing gaming datasets contain multiple players (MSRC-12 [71], MSR Action3D Database [40] and G3D (introduced in section 3.4)). A wide range of applications could benefit from recognising the actions of multiple users including home entertainment, healthcare, sports, and robotics. For example, a personal robotic assistant for the elderly in a care home could interact with multiple staff and patients to appear more natural. Another example is a training simulation for health care professionals where multiple trainees could interact with a virtual patient, which would emulate the real-life scenario. The Xbox Kinect already has many games titles that are

multi-player. Multiplayer computer games encourage people to interact with other players across the globe or friends and family in the same living room. The interactions can be collaborative or competitive depending on the game and the mode. Boxing is naturally a competitive sport but team sports can be played either collaboratively with friends on the same team or competitively with friends on the opposing team. For example, one can play table tennis alongside a friend in a doubles match or against a friend in a singles match. The players can act simultaneously or after a short delay depending on the sport. For example, in boxing the actions are concurrent but other sports such as table tennis have a delay between one person acting and the other reacting.

Evaluating action recognition algorithms is typically done in isolation, focusing historically on high accuracy and more recently also on low latency. However, in reality most actions form part of an interaction where the duration of the action becomes important. In normal human interaction, people physically interact with each other, like in a real boxing match. Recent technological developments, such as low cost depth sensors, have enabled a new form of interaction which is virtual, for example a full body boxing game illustrated in Figure 5-1.

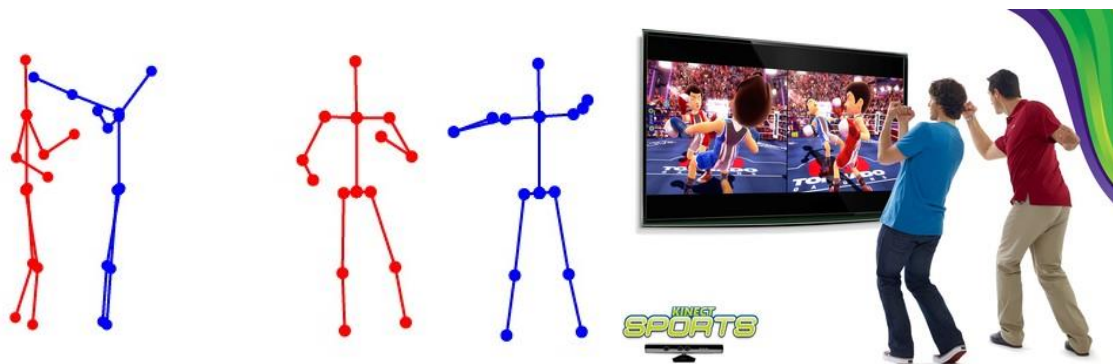


Figure 5-1 Boxing interactions: A real attack (left) occurs when one person punches the other person and makes physical contact, in contrast a virtual attack (middle) and a virtual block (right) occur when both players face the screen and perform actions toward the computer screen so there is no physical contact.

To overcome the challenges presented by realistic multiplayer datasets a Hierarchical Transfer Learning (HTL) algorithm is proposed which is comprised of a hierarchical interaction detection and evaluation framework in addition to a novel transfer learning mechanism for recognition of compound actions.

To enable the recognition of higher-level interactions a hierarchical approach is employed which is based on the recognition of actions. The motivation is that actions are easier to recognise first and can then be used for recognising higher-level interactions. For example, a virtual interaction between two people such as a block in a boxing game could be recognised as a punch action and a defence counter action. The benefits of the proposed hierarchical approach is that it reduces the amount of training data required and interactions are recognised more efficiently as redundancy is reduced in the recognition process by using actions multiple times. Due to the complexities introduced by the compound actions, transfer learning is employed to allow the tasks of action segmentation and modelling to be performed on a related but simpler dataset, combined with model adaptation to improve performance on the more complex dataset. Furthermore, actions are represented by discriminative body parts to provide the flexibility to match test poses that are not in the training dataset by introducing independence between limbs.

5.2 Related Work

Previous chapters have focused on online action recognition of scripted actions whereas this review considers recognising actions from a real world scenario. First, existing datasets are reviewed in terms of the complexity of the actions and the number of subjects (see section 5.2.1). Then, as the diversity and complexity of real-world datasets makes accurate labelling difficult and time consuming techniques for transferring knowledge from simple to complex datasets are reviewed (see section 5.2.2). Next, as commercial

games are often multiplayer, existing work on interaction is reviewed (see section 5.2.3). Finally, interaction performance metrics are evaluated for their suitability for evaluating real-time applications (see section 5.2.4).

5.2.1 DATASETS

The problem with the existing gaming datasets, MSRAction3D [40], MSRC-12 [71] and G3D, is that the scenarios were scripted so the subjects' movements are not realistic. In scripted datasets, the participants are instructed beforehand on how and when to perform the actions, which results in actions that are temporally isolated. In these datasets, the player often returns to the neutral position between actions making them easier to segment and recognise. However, in fast paced competitive computer games like boxing, players skip returning to the neutral position between actions, which results in compound actions. There are no existing gaming datasets containing compound actions.

Existing interaction datasets contain multiple actors and similar to action recognition can be categorised into scripted scenarios [127]–[129] and realistic scenarios. The scripted datasets contain simple interactions such as hand shaking, hugging and kicking performed in staged environments which may be indoors or outdoors, with participants captured from a side view. The majority contain video data [128] and some also contain depth and skeleton data [129], [127]. However, the latter contain noisy and unreliable skeleton data. The realistic scenarios include surveillance [84], [85] and movie / TV datasets [86], [130]. The surveillance datasets focus specifically on surveillance of public spaces for example train stations using a CCTV camera viewpoint [84]. The movie datasets [86] contain a range of activities from various camera viewpoints. Neither of these groups of datasets are suited to gaming scenarios due to the types of activities they contain. There are no known publicly available databases containing gaming interactions with multiple players.

5.2.2 TRANSFER LEARNING

Transfer learning is a machine learning approach to store knowledge gained whilst solving one problem and apply it to a different but related problem. It has been beneficial to many machine learning research areas, including classification, regression and clustering problems to reduce the need to collect and label training data [131]. However, transfer learning applied to action recognition is a relatively new topic with limited research in the computer vision community. Transfer learning has been used for cross-view action recognition [132], [133] to recognise human actions from different views. In both cases the methods were tested offline on a multi-view dataset (IXMAS) [134], which comprised of simple actions with simple backgrounds so it has limited applicability to real world scenarios.

More significantly transfer learning has been used cross-dataset [135], [136] to harness lab datasets to facilitate real-world action recognition. The aim is to generalise action models built from a source dataset to a target dataset, to alleviate the problem of labelling complex sequences. The source dataset typically has a clean background and each video clip may involve only one type of action and a single person, which describes most lab-collected datasets. In contrast, in the target dataset the background may be cluttered and there may be multiple people and multiple actions which may overlap temporally. Cross-dataset learning aims to adapt the existing classifier from a source dataset to a new target dataset, while requiring only a small or even no labelled samples in the target dataset.

Ma et al. [135] built a model within a multi-task framework so the actions of one domain are associated with its own features. The general Schatten p -norm was applied to mine the shared components between the lab data and the real world data. The main advantage of their approach is the ability to share knowledge between the two datasets even if they have different action categories. However, the method was tested offline with sequences containing just a single action.

Cao et al. [136] combine model adaption and action detection into a Maximum a Posterior (MAP) estimation framework for action detection. The advantage of this approach over the previous method is that it can perform spatial-temporal detection of the action within a sequence. However, as a search for the optimal 3D sub-volume is performed across all frames in the target sequence this approach is also offline.

Charkraborty et al. [137] used a probabilistic optimisation model of body parts using HMM. Their method is able to distinguish between similar actions by only considering the body parts that have a major contribution to the actions, for example, legs for walking, jogging and running and arms for boxing waving and clapping. The problem is that the popular action recognition datasets e.g. KTH and Weizmann do not contain body part labels and they are very time consuming to annotate so to overcome this they trained on the HumanEva dataset and tested on the KTH and Weizmann datasets. The detection of body parts took 333ms per frame and additionally HMM has high observational latency which means this approach is not feasible for real-time action recognition.

The existing approaches for transfer learning regarding actions are offline so the knowledge transferred from the source to target dataset is in relation to the action class and both computational and observational latency are high. An idea that has not been considered before is the potential for transfer learning to improve online action recognition, where knowledge about the temporal localisation of the action needs to be transferred in addition to requirements for low latency.

5.2.3 INTERACTION RECOGNITION

Another limitation of the existing gaming datasets that they are single player, whereas commercial games are often multiplayer. The literature reviews in previous chapters have focused on online action recognition of a single subject. In contrast this review considers multiple subjects who are typically researched in terms of their interactions with each

other. In traditional interaction recognition the subjects physically interact in the real world whereas in this chapter the interest is in virtual interaction where multiple subjects interact through a computer.

In human activity recognition there is a vast wealth of research on interaction recognition and traditionally approaches were appearance based as low level features could be quickly extracted from colour sequences. Recent work [24], [40], [43] suggests that human activity recognition accuracy can be improved by using features from 3D data. Pose-based features from skeleton data are a very effective representation for human motion [7], [71], [89], [127], [129] so the focus of this thesis is on pose-based approaches.

Due to the development of a real time pose estimation algorithm [6] from depth streams many recent activity recognition algorithms are based on skeletal joint information. In a recent review of human activity recognition from 3D data [24], the authors concluded that most current approaches only deal with a single human subject. Subsequently, the features are based on joints from a single skeleton such as the pairwise joint location difference feature [7], [71], [89].

These pose-based features were extended to multiple skeletons by Yun et al. [127] to model human interactions. Their experiments showed that the distance between all pairs of joints was the optimum set of joint features for real-time interaction. This feature measures the pairwise joint distance in each skeleton, as well as between the two skeletons. This feature set was specifically designed for person to person interaction where the distance between the joints of the people aids the classification in some cases. For example, the distance between two people can easily be used to differentiate between approaching and departing. However, this feature set is not so relevant for other actions, especially in virtual human interaction where there is no physical interaction between the people. However, if this feature is required it is trivial to rotate the skeletons in a virtual interaction scenario to face each other.

Further research by Hu et al. [129] with pose-based features from multiple skeletons discovered that an interaction can be represented by a positive and negative action. Their results showed that the positive action on its own was discriminative enough to classify the interactions in their dataset, so the interaction recognition was simplified to positive action recognition. This works for simple scenarios where there is only one outcome from an action, such as the punching in their dataset where the first person punches and the second person falls away from the hit. However, in more complex scenarios there are more than one possible reactions from a punch, for example, a hit as just described or a block where the second person defends themselves by raising their hands in front of their face. If the skeletal information from the second person is ignored it will be very difficult to differentiate between these two interactions.

5.2.4 EVALUATION METRICS

Similarly, to action recognition a common performance measure used for interaction recognition is classification accuracy which is applied to the entire sequence. For example, an interaction label is predicted for each frame in the sequence and a majority decision over all frames is taken to decide the interaction label for the complete sequence. However, this approach can only be applied to pre-segmented sequences containing the same interaction which is not the case for many real-world applications.

To overcome this limitation of sequence-based evaluation, frame-based evaluation metrics have been developed [128], [138]. Escalera et al. [138] introduced a Jaccard index that can evaluate sequences with multiple action/interaction classes with respect to time. Ryoo and Aggarwal [128] proposed spatial and temporal bounding boxes to evaluate sequences with multiple interactions with respect to both space and time. Both approaches are evaluated based on the overlap between the system detection and the ground truth labels. These application metrics include temporal constraints but do not explicitly measure the latency of the detection.

An alternative performance metric is the localisation of each action, used when multiple actions occur simultaneously in video or depth data. An action label for each pixel of each frame is predicted, illustrated in Figure 5-2 with actions colour coded. Each action has a ground truth rectangular bounding box for each frame and a correct action localisation occurs if the dominant pixel label within the bounding box matches the ground truth [139]. This level of annotation is more difficult and more time consuming than temporal labels and is not necessary for skeleton data as a simple identity tag can be used to discriminate between multiple people in a scene.

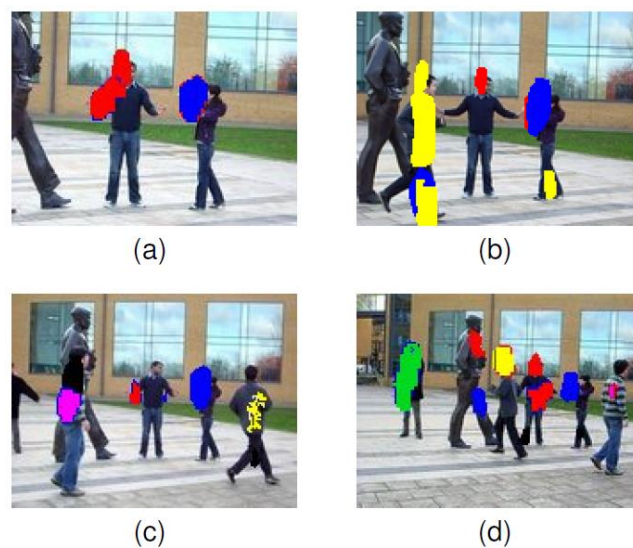


Figure 5-2 Localisation results from the Multi-KTH dataset, red - handclapping, blue - boxing, yellow - running, pink - walking, green - hand waving [139]

Low latency detection is critical for real world applications such as gaming and surveillance. Nowozin et al. [13] proposed the Action Point F_1 -Score which is the latency aware performance metric which has already been used to evaluate the online action recognition algorithms in previous chapters. They introduced ‘action points’ as temporal anchors for the detection and evaluation of actions in real time. However, there are no existing metrics for interaction recognition that explicitly measure latency.

5.3 Methodology

The proposed method Hierarchical Transfer Learning for online action and interaction recognition consists of three phases: offline training and model adaptation, and online testing phase as illustrated in Figure 5-3. The expected input is skeleton data, specifically joint angles which are viewpoint and anthropometric invariant and can be generated in real-time with a pose estimation method [6]. A key contribution of the proposed method in this chapter is that the body part model can be automatically configured to detect actions based on the body parts that are the most discriminative for a particular action. Another key contribution is a transfer learning strategy to allow the tasks of action segmentation and whole body modelling to be performed on a related but simpler source dataset, combined with automatic body part model adaption on a more complex target dataset. The final key contribution is the hierarchical interaction detection framework, which recognises actions first and then infers higher-level interactions.

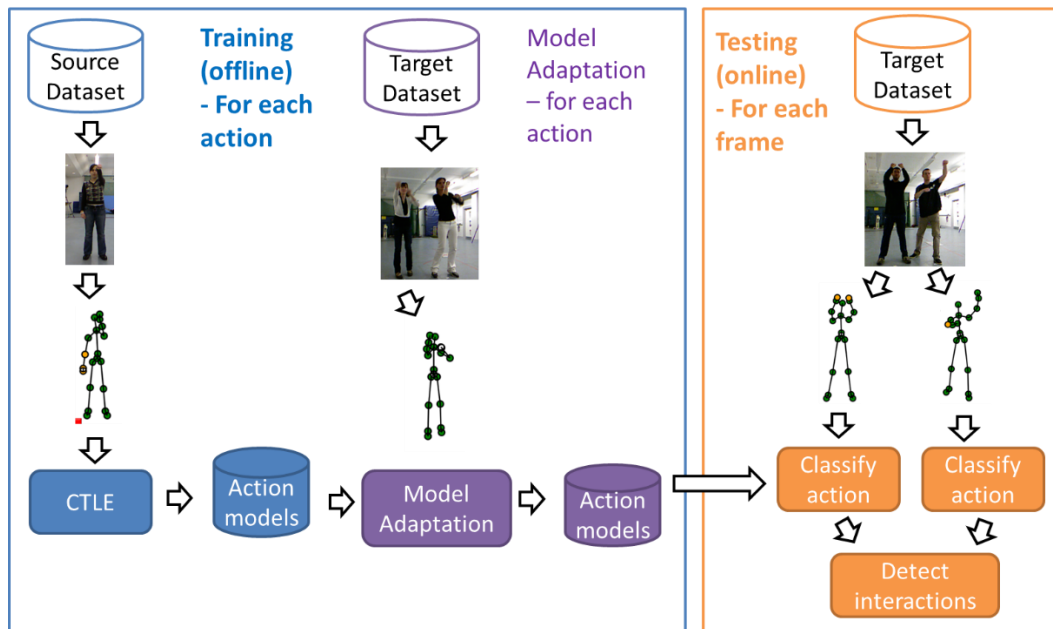


Figure 5-3 Methodology Overview

5.3.1 TRAINING PHASE (SOURCE DATASET)

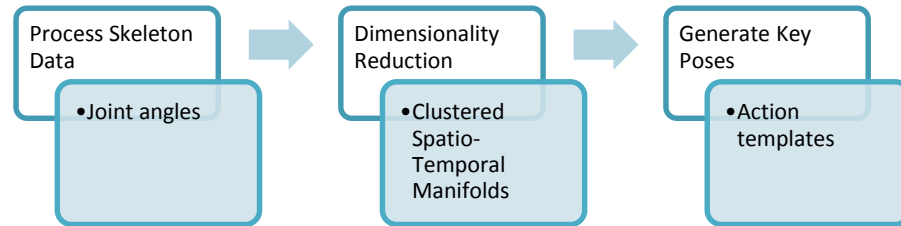


Figure 5-4 Training overview which is performed on the source dataset for each action

The training phase is based on CSTM which was introduced in the previous chapter for online action detection, explained in detail in section 4.3.1 and summarised here. CSTM achieved high accuracy and low latency for multiple actions that were separated temporally. The contribution in this chapter is to adapt the action templates to detect compound actions by representing actions by their most discriminative body parts. The two key stages in training of CSTM are dimensionality reduction and key pose generation. Dimensionality reduction of the skeleton data produces spatio-temporal manifolds which removes individual style whilst maintaining the temporal ordering of the poses. Clustering the manifolds and projecting the cluster centres back to the high dimensional space creates key poses. An individual key pose represents a generic pose from an action at a specific point in time and the sequence of these key poses represent the entire action (as illustrated in Figure 5-5). A major benefit of the clustering is that the number of key poses is significantly lower than the original number of training poses which dramatically reduces the computation time and enables the approach to scale efficiently to much larger datasets.

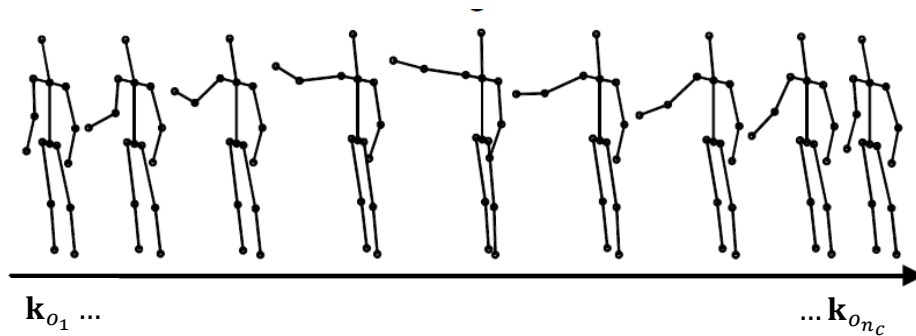


Figure 5-5 Right punch action template, consisting of key poses k_1 to k_m where m is the number of clusters

5.3.2 MODEL ADAPTATION PHASE (TARGET DATASET)

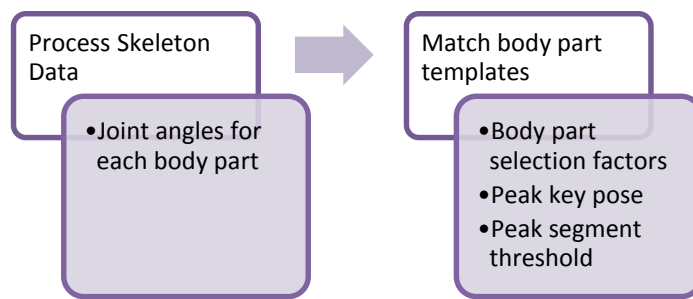


Figure 5-6 Model Adaptation overview which is performed on the target dataset for each action

To detect compound actions a body part template matching algorithm is proposed. Representing actions using body part models allows independence between the body parts $\mathbf{B} = (\mathbf{b}_{i_b})_{(i_b=1\dots n_b)}$ (as illustrated in Figure 5-7). Each body part is represented by joint angles indices, so body parts can be described at any level of granularity, in the proposed approach most of the body parts contain four joints to represent semantic body parts such as arm and leg. The contribution of this section is to automatically select each body part based on their discriminative ability to detect specific actions. Selecting individual body parts, creates flexible body part configurations at different levels of granularity e.g. whole body, upper body or right arm and atypical combinations such as right arm and left leg.



Figure 5-7 Body parts: the skeleton is divided into body parts, right arm (red), left arm (blue), right leg (green), left leg (pink) and torso (black).

There are three main steps to adapt the action templates of the whole body learnt from the source dataset for body part template matching: a) learning the most discriminative body part combinations, b) detecting the most representative peak key pose and c) optimising the peak segment threshold.

All three steps use exemplar matching between the peak poses in the target dataset training poses and the action templates to find the optimum matching parameters. To incorporate the temporal history of the action and increase the robustness of the matching process sequences of poses are matched rather than single poses. To extract a fragment from a sequence of poses Eq. (4-2) is used.

DTW [121] is a well-known algorithm for determining the similarity of time-series data that allows “elastic” transformation to gain execution rate invariance. The similarity of two series of poses, the query sequence $\mathbf{Q} = (\mathbf{q}_{i_q})_{(i_q=1\dots n_q)}$, ($\mathbf{q}_{i_q} \in \mathbb{R}^D$) and the reference sequence $\mathbf{R} = (\mathbf{r}_{i_r})_{(i_r=1\dots n_r)}$, ($\mathbf{r}_{i_r} \in \mathbb{R}^D$), can be computed using the standard DTW distance metric using Eq. (4-4). In the previous chapter the DTW distance was computed for the whole body (see section 4.3.1.5 for more details). To increase flexibility a selective DTW distance measurement is proposed:

$$f^W(\mathbf{Q}, \mathbf{R}, \mathbf{w}) = \sum_{i_b=1}^{n_b} f^D(\mathbf{Q}_{i_b}, \mathbf{R}_{i_b}) w_{i_b} \quad (5-1)$$

For two series of poses, the query sequence \mathbf{Q} and the reference sequence \mathbf{R} , the similarity of body parts is computed independently using the standard DTW distance metric f^D . A selective combination $\mathbf{w} = (w_{i_b})_{(i_b=1\dots n_b)}$, $w_{i_b} \in [0,1]$ of the body part distances provides a discriminative distance metric for compound actions.

5.3.2.1 Body Part Combinations

The most discriminative body part combinations for each action are discovered by maximising the ratio of intra-class matches between the labelled peak poses in the training data of the target dataset and the action templates. This procedure is repeated for all body part combinations, so for computational efficiency binary selection, i.e. $w_{i_b} \in \{0,1\}$ for each of the body parts was employed, which results in 2^{n_b} permutations. For each permutation ε , the intra-class ratio ρ is computed by the number of intra-class matches μ over the number of total training instances in the target dataset n_g . The intra-class matches are counted for each action by exemplar matching between the peak poses from the training data of the target dataset and the key poses from all the action templates. For each action a , if the closest matching action template is the same action this is counted as an intra-class match. The maximum intra-class ratio represents the most discriminative body part combination for each action, as illustrated in Figure 5-8 and summarised in Algorithm 5-1.

Algorithm 5-1 Learn the most discriminative selection factor for each action

Input: Given a set of training poses from the target dataset $\mathbf{G} = (\mathbf{g}_{i_g})_{(i_g=1\dots n_g)}$,

, $\mathbf{g}_{i_g} \in \mathbb{R}^D$ with manually selected peak poses from \mathbf{G} represented by their indices

$\boldsymbol{\eta}^a = (\eta_{i_\eta^a})_{(i_\eta^a=1\dots n_\eta^a)}$ and a set of learnt action templates $\mathbf{K}^a = (\mathbf{k}_{i_k})_{(i_k=1\dots n_c)}$

For each action $a = 1:A$

1. For each permutation $\varepsilon = 1:2^{n_b}$ of body parts
 - 1.1. Initialise $\mu = 0$
 - 1.2. For each peak pose index, $i_\eta^a = 1 \dots n_\eta^a$
 - 1.2.1. Extract the peak pose fragment using Eq. (4-2)
 - 1.2.2. $a' = \min_{a^* \in 1\dots A} f^W(f^G(\mathbf{G}, i_\eta^a), \mathbf{K}^{a^*}, \mathbf{w}_\varepsilon)$ using Eq. (5-1)
 - 1.2.3. If $a' = a$
 - 1.2.3.1. Intra-class match so increment μ
 - 1.3. Compute intra-class ratio $\rho_\varepsilon = \frac{\mu}{n_\eta^a}$
 2. Select the most discriminative selection factors, $\mathbf{w}^a = \arg \max_\varepsilon \rho_\varepsilon$
 3. Store \mathbf{w}^a
-

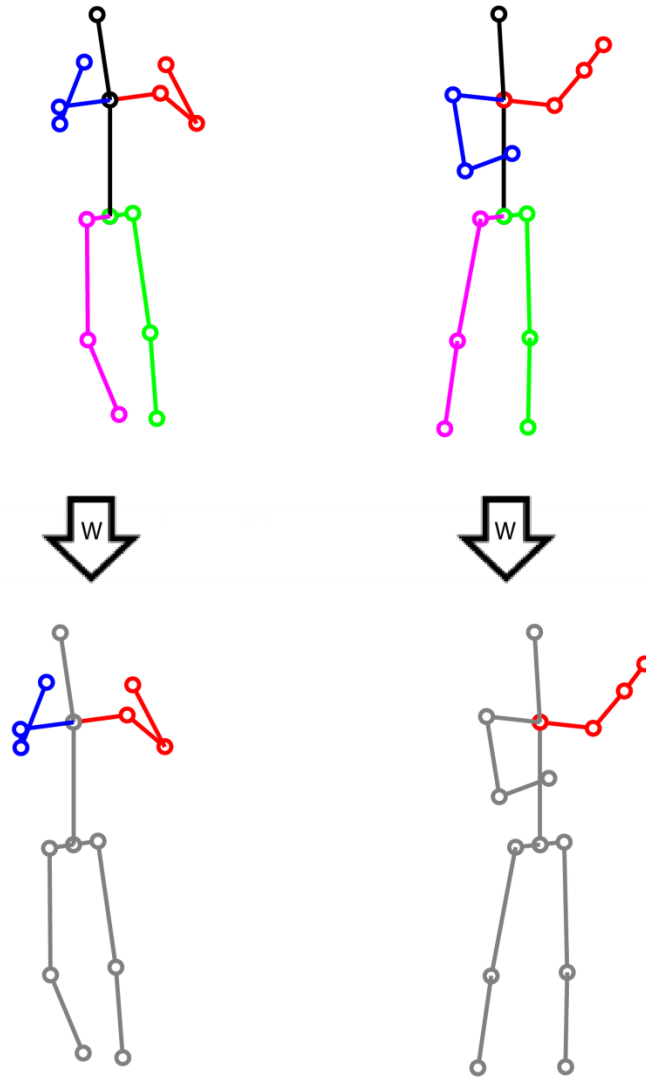


Figure 5-8 Body Part Combinations: The selection factors (W) are optimised for each action based on their ability to discriminate compound actions in the target dataset. The bottom skeletons show potential body parts configurations for the defence (left) and right punch (right) actions.

5.3.2.2 Peak Key Pose Selection

In the previous chapter, peak key poses were proposed as the generic representation of action peak and were automatically selected from the key poses by exemplar matching with the training data (see section 4.3.1.5 for more details). To increase robustness on compound actions the exemplar matching in this chapter is performed using the most discriminative body parts rather than the whole body. The peak key poses are therefore selected as follows: for each action and for each peak pose in the training data of the target dataset, the best matching key pose is found (as shown in Figure 5-9). A peak key pose can be represented by its index i_k in the action template. For each action, the best matching index i_m is found by minimising the distance between the peak pose fragments and the key pose fragments using the most discriminative body part combination. The peak key pose i_p for the action, is the key pose that has the maximum number of matches, as summarised in Algorithm 5-2.

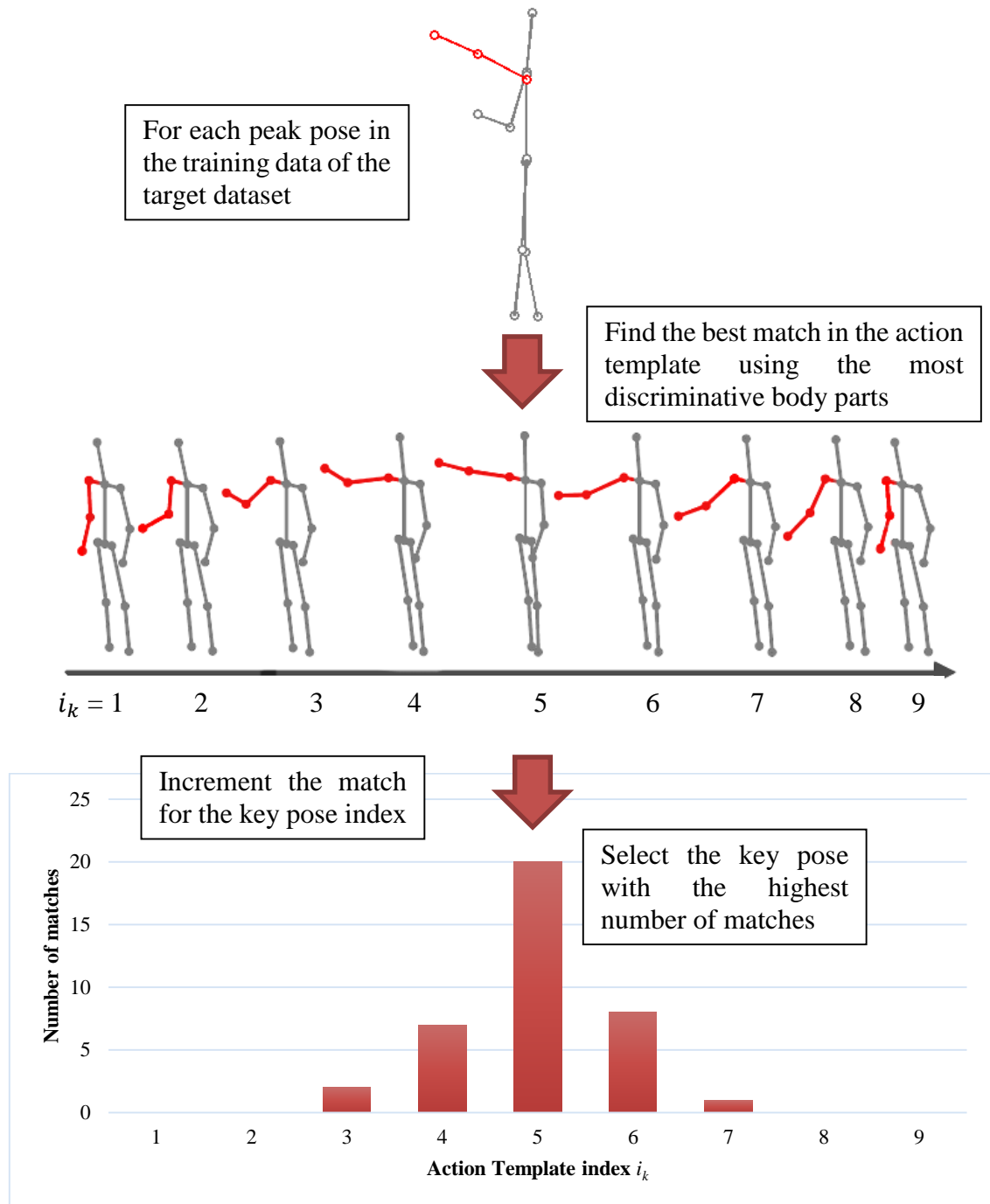


Figure 5-9 Peak key pose selection: each action is considered independently at this stage.

Algorithm 5-2 Learn the peak key pose

Input: Given a set of training poses from the target dataset $\mathbf{G} = (\mathbf{g}_{i_g})_{(i_g=1\dots n_g)}$, $\mathbf{g}_{i_g} \in \mathbb{R}^D$ with manually selected peak poses from \mathbf{G} represented by their indices $\boldsymbol{\eta}^a = (\eta_{i_\eta^a})_{(i_\eta^a=1\dots n_\eta^a)}$ and a set of learnt action templates $\mathbf{K}^a = (\mathbf{k}_{i_k})_{(i_k=1\dots n_c)}$ and learnt selection factors \mathbf{w}^a :

For each action, $a = 1:A$

1. Initialise $\boldsymbol{\zeta} = \mathbf{0}_{(i_\zeta=1\dots n_c)}$
2. For each peak pose index, $i_\eta^a = 1 \dots n_\eta^a$
 - 2.1. Extract the peak pose fragment, $f^G(\mathbf{G}, \eta_{i_\eta^a})$ using Eq. (4-2)
 - 2.2. Find the best matching key pose index

$$i_m = \arg \min_{i_k \in 1\dots n_c} f^W \left(f^G(\mathbf{G}, \eta_{i_\eta^a}), f^G(\mathbf{K}^a, i_k), \mathbf{w}^a \right)$$

- 2.3. Increment ζ_{i_m}
 3. Determine the peak key pose index $i_p(a) = \arg \max_{i_\zeta} \zeta_{i_\zeta}$
-

5.3.2.3 Peak Segment Detection

Some existing methods for online action recognition detect the action as a single point in time [71], [140] whereas others incorporate the duration of the action [141], [142]. A single point in time accurately represents the peak of some actions, for example a punch. However, this is not the case for all actions, such as the defence, whose goal is defined as “when two hands are positioned in front of the face” as in reality the hands remain in front of the face for a significant period of time. To overcome the limitation of action points, action segments are proposed. In contrast to an action point, an action segment has temporal duration. The duration of the action peak is critical for recognising interactions when either subject performs actions with extended action peaks. An example is a multiple player boxing game, where one subject defends whilst the other subject punches

him multiple times. These should all be detected as blocking interactions, but without considering the duration of the defence action, only the first would be detected as a block and the subsequent punches incorrectly as attacks as illustrated by Figure 5-10.

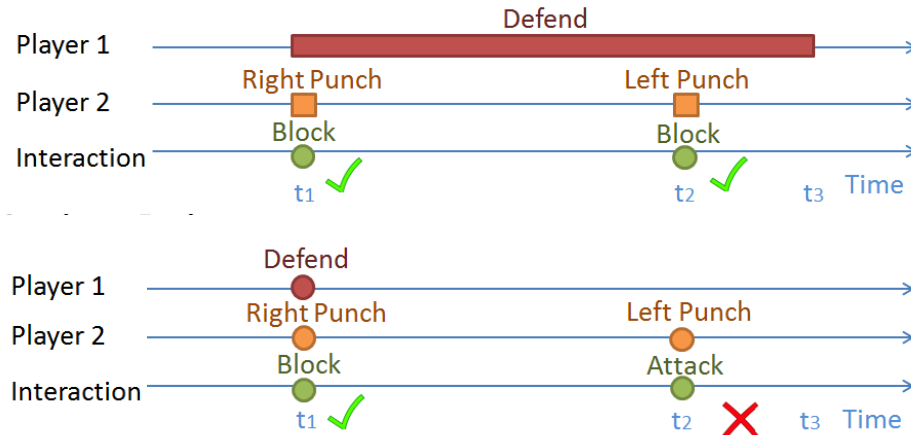


Figure 5-10 (Top) Interaction detection based on action segments which correctly detects actions with long duration. (Bottom) Interaction detection based on action points, which only works if both actions occur at the same time and incorrectly detects interactions if an action has a long duration.

Peak key poses proposed in chapter 4 were limited to detecting a single temporal point so this chapter extends the peak key pose matching to incorporate the duration of the peak. The peak key pose matching is performed using DTW to ensure execution rate invariance and the normalised DTW distances recorded for each frame is illustrated in Figure 5-11. To detect actions in real-time the lowest body part DTW at each frame is compared with the threshold T . If the distance is large ($> T$) then this is not the peak of an action as it is not similar to any of the peak key poses. However, if the distance is sufficiently small ($\leq T$) then this represents the action peak, as shown by the coloured segments on Figure 5-11. The graph shows that selecting a single threshold for multiple actions, can detect actions with both short (punches) and long (defence) duration.

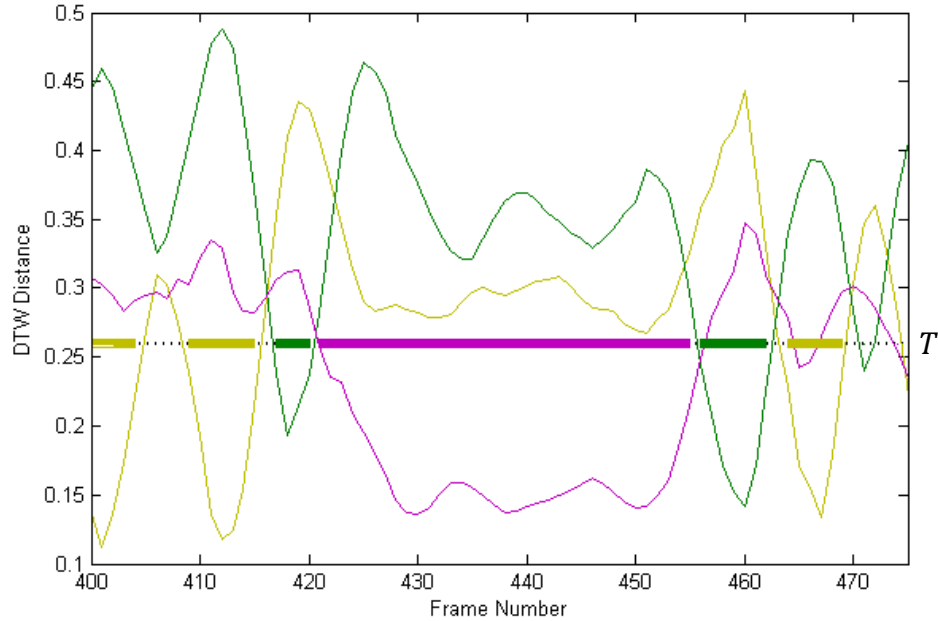


Figure 5-11 Normalised DTW distances: the lowest value represents the most similar action, where this value is lower than the threshold T it represents the detected action. The right punch is displayed in yellow, left punch displayed in green and the defence in magenta.

Similar to [141], [142] a threshold T is introduced but instead of specifically learning a threshold for each action a single threshold for all actions is learnt. Confining the threshold to a single parameter reduces the time taken to adapt the model and this time will not increase even if more actions are considered, providing scalability to larger datasets. The threshold T and fragment size n_f are learnt on the training part of the target dataset by optimising the action point F_1 metric (defined in section 2.5.2.2.3) with the proposed body part template matching algorithm (summarised in Algorithm 5-3) but using the training data from the target dataset rather than the testing data.

5.3.3 TESTING PHASE (TARGET DATASET)

To enable the recognition of higher-level interactions, a hierarchical approach is employed in the testing phase which is based on the recognition of actions, as illustrated in Figure 5-12. The motivation is that actions are easier to recognise first and can be then used for recognising higher-level interactions. The benefits of the proposed hierarchical approach is that it reduces the amount of training data required and interactions are recognised more efficiently as redundancy is reduced in the recognition process by using actions multiple times.

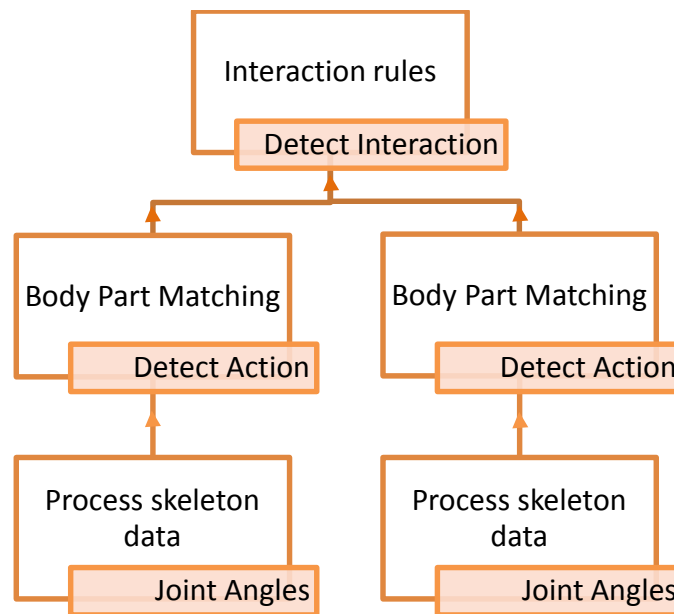


Figure 5-12 Hierarchical view of interaction recognition performed on the target dataset

5.3.3.1 Online Action Recognition

The proposed online Body Part Matching algorithm combines the three elements learnt from the model adaptation phase: Body Part Combinations, Peak Key Pose and Peak Segment Detection threshold to detection compound actions with low latency.

The Body Part Matching algorithm instead of a binary decision for matching a peak key pose uses the threshold T to enable detection of the duration of the peak of an action. The algorithm is summarised here and formalised in Algorithm 5-3. For each test pose in a continuous stream from the target dataset the test pose fragment is extracted. The test pose fragments are matched against the action templates with the selective DTW of the most discriminative body parts (proposed in section 5.3.2). The minimum DTW distance is compared against the threshold to determine if the action peak has been reached. If the action peak has extended duration as in the case of the defence action then the proposed algorithm will keep outputting the same action until the peak has been passed.

Algorithm 5-3 Body Part Matching algorithm

Input: Given a sequence of testing poses from the target dataset $\mathbf{H} = (\mathbf{h}_{i_h})_{(i_h=1\dots n_h)}$, $\mathbf{h}_{i_h} \in \mathbb{R}^D$ and a set of learnt action templates $\mathbf{K}^a = (\mathbf{k}_{i_k})_{(i_k=1\dots n_c)}$ and learnt selection factors \mathbf{w}^a , peak key poses indices $i_p(a)$ the learnt fragment size n_f and the learnt distance threshold τ :

For each test pose index $i_h = 1 \dots n_h$:

1. Extract the current test pose fragment, $f^G(\mathbf{H}, i_h)$ using Eq. (4-2)
 2. For each action, $a = 1:A$
 - 2.1. Extract the key pose fragments, $f^G(\mathbf{K}^a, i_p(a))$ using Eq. (4-2)
 3. $\delta = \min_{a^* \in 1\dots A} f^W(f^G(\mathbf{H}, i_h), f^G(\mathbf{K}^{a^*}, i_p(a^*)), \mathbf{w}^{a^*})$ using Eq. (5-1)
 4. If $\delta < T$
 - 4.1. $a' = \arg \min_{a^* \in 1\dots A} f^W(f^G(\mathbf{H}, i_h), f^G(\mathbf{K}^{a^*}, i_p(a^*)), \mathbf{w}^{a^*})$
 - 4.2. Output action a'
-

5.3.3.2 Hierarchical Interaction Detection Framework

The proposed Hierarchical Interaction Detection Framework enables online interaction recognition between people by detecting their individual actions independently and combining them by a set of interaction rules to infer the interaction. This modular approach is applicable for NUI and enables interaction between people that are not in the same physical location. Actions from different people are detected independently. At each frame, these detections are combined to infer the current interaction. The interaction rules include the valid combinations of actions together with timing constraints. The interactions for the G3Di dataset are depicted in Table 5-1. The action a and counter action a' , are checked at each frame to detect interactions in real time. To check if the action and counter actions temporally overlap two constraints must be satisfied. The first constraint is that the action must start before (or at the same time) the counter action ends ($a_s \leq a'_e$). The second constraint is that the counter action must start before the action ends ($a'_s \leq a_e$). Overlapping examples are shown in Figure 5-13 and non-overlapping examples are shown in Figure 5-14. Finally, a timing constraint t_c is used for scenarios, such as table tennis, a delay is expected between the action and counter action ($t_c > 0$). Action segments are used to represent the peak of the actions and interactions are detected if the action and counter segments overlap either at the same point in time or after a fixed delay, as defined by:

$$\psi(a_s, a_e, a'_s, a'_e) = \begin{cases} 1 & \text{if } a_s + t_c \leq a'_e \text{ and } a'_s \leq a_e + t_c \\ 0 & \text{otherwise} \end{cases} \quad (5-2)$$

where the subscripts s and e represent the start and end of the action segment respectively and $s \leq e$.

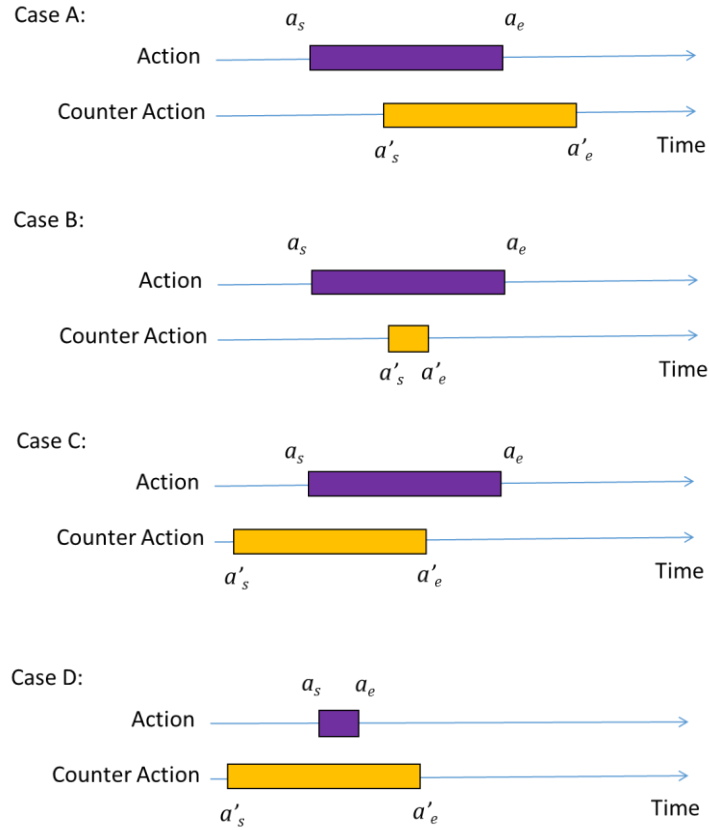


Figure 5-13 Cases A-D, examples of overlapping actions and counter actions, assuming $t_c = 0$

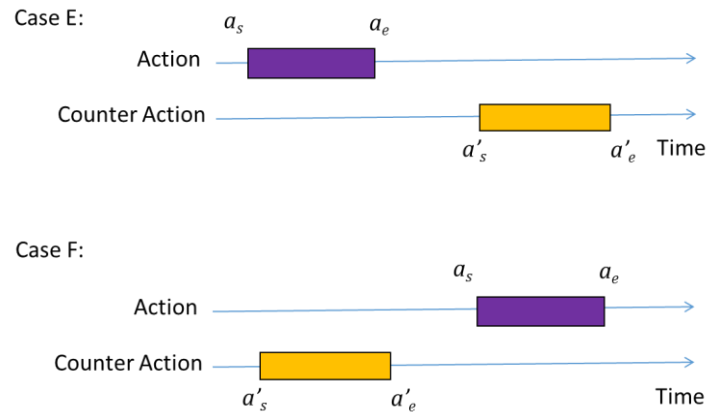


Figure 5-14 Cases E and F, examples of non-overlapping actions and counter actions, assuming $t_c = 0$

Table 5-1 Gaming interactions for the boxing and table tennis scenarios in G3Di.

Sport	Action	Counter Action	Interaction
Boxing	Right Punch	Defend	Block
	Left Punch	Defend	Block
	Right Punch	Other	Attack
	Left Punch	Other	Attack
	Right Punch	Right Punch	Attack
	Right Punch	Left Punch	Attack
	Left Punch	Left Punch	Attack
Table Tennis	Serve	Forehand hit	Rally
	Serve	Backhand hit	Rally
	Serve	Other	Miss
	Forehand hit	Forehand hit	Rally
Table Tennis	Forehand hit	Backhand hit	Rally
	Forehand hit	Other	Miss
	Backhand hit	Backhand hit	Rally
	Backhand hit	Other	Miss
Volleyball	Action	Counter Action	Interaction
	Underhand hit	Underhand hit	Set
	Underhand hit	Overhand Hit	Set
	Overhand Hit	Underhand hit	Set
	Overhand Hit	Overhand Hit	Set
	Jump Hit	Underhand Hit	Set
	Jump Hit	Overhand Hit	Set
	Underhand hit	Jump Hit	Attack
	Overhand hit	Jump Hit	Attack

Sport	Action	Counter Action	Interaction
	Jump hit	Jump Hit	Attack
Football	Action	Counter Action	Interaction
	Kick	Kick	Block
	Kick	Block	Block
	Kick	Save	Block

5.4 G3Di dataset

A new multimodal interaction dataset has been captured, for real time multiplayer gaming and is publicly available⁶. G3Di was captured using a novel game-sourcing approach where the users were recorded whilst playing Kinect Sports [126], a commercial video game. Sports games introduced the element of competition between the players so the actions captured were more realistic and challenging to recognise in comparison to scripted actions. Subjects in the new game-sourced dataset (G3Di) performed multiple actions in quick succession which resulted in compound actions, comprising of movements from different actions. For example, in a full body fighting game a player may throw punches in quick succession, one arm may still be finishing the previous punch whilst the other arm is performing the next punch or a player may leave one arm in the defend position and punch with the other arm (as shown in Figure 5-16). Detecting compound actions is a more complex problem than recognising actions which are temporally isolated.

The proposed recording environment as illustrated in Figure 5-15 allowed the capture of realistic gaming actions. The setup shows two players as the version of depth sensor used

⁶ G3Di can be downloaded from <http://dipersec.kingston.ac.uk/G3D/>

was limited to full skeleton tracking of two people. The recording environment contains two overlapping depth sensors: one for playing a commercial game on a standard games console and the other to capture the colour, depth and skeleton data. The disadvantage of using two sensors with overlapping fields of view is that considerable noise is introduced to the depth data and consequently the skeleton data, due to infrared interference. Specifically, the depth sensor used the Kinect v1, derives depth by projecting a structured light code onto the scene and comparing the projected pattern with the stored pattern. To overcome this problem a motor was attached to one depth sensor to vibrate it and therefore reduce the interference between them as observed in experiments by Butler et al. [20].

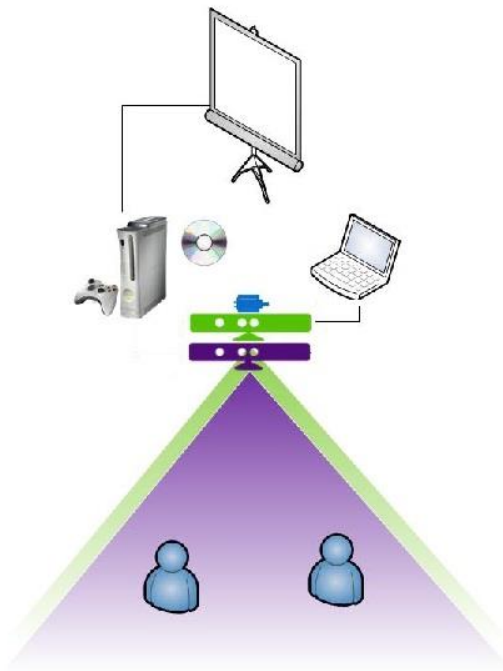


Figure 5-15 Recording environment with 2 depth cameras for simultaneous gameplay and recording.



Figure 5-16 Complex fighting sequences between multiple players, performing multiple actions in quick succession so that the movements temporally overlap (G3Di) [143]. Each row represents a different sequence with visual examples taken every 3 frames.



Figure 5-17 Synchronised colour, depth and skeleton data from a boxing game

Due to the formats selected, it is possible to view all the recorded data and metadata without any special software tools. The three streams were recorded at 30fps in a mirrored view. The depth and colour images were stored as 640x480 PNG files and the skeleton data in XML files. The raw depth information contains the depth of each pixel in millimetres and was stored in 16-bit greyscale and the raw colour in 24-bit RGB. The 16-bits of depth data contain 12 bits for the depth distance (0-4096mm), 1 bit reserved for the sentinel values (which was not used and fixed at 0) and 3 bits to identify the player. The player index can be used to segment the depth maps by user. The depth information was also mapped to the colour coordinate space and stored in a 16-bit greyscale. Combining the colour image with the mapped depth data allows the user to also be segmented in the colour image.

Each skeleton contains the player's position and pose: the pose comprises of 20 joints and the joint positions are given in X, Y and Z coordinates in meters. These positions are also mapped into the depth and colour image coordinate spaces. The skeleton data includes a joint tracking state, displayed in Figure 5-17 as tracked (green), inferred (yellow) and not tracked (red). The joint tracking state provides the confidence of the coordinates for each joint. If the joint is tracked, the confidence in the coordinate data is very high. Whereas, if the joint is inferred by calculating it from other tracked joints, the confidence in the coordinate data will be very low. This is important information for developers of multimodal algorithms fusing data between the skeleton data and other modalities.

To the best of my knowledge this is the first dataset comprised of virtual interactions, meaning that two players interact with each other through a computer interface. This dataset contains 12 people split into 6 pairs. Each pair performed 18 gaming actions from Kinect Sports [126], for six sports games: boxing (right punch, left punch, defend), volleyball (serve, overhand hit, underhand hit, jump hit, block and jump block), football (kick, block and save), table tennis (serve, forehand hit and backhand hit), sprint (run) and hurdles (run and jump). Most sequences contain multiple action classes in a controlled indoor environment with a fixed camera, a typical setup for gesture based gaming. The people played the game in a training mode to become familiar with the movements before they were recorded. The actual game was recorded and particular sections where several different actions were performed multiple times by each player were selected for the dataset. The key features of the gaming datasets are summarised in Table 5-2. G3Di is the only gaming dataset to contain interactions and additionally contains more complex actions as it is the only dataset to be recorded using a commercial game.

Table 5-2 Comparison of gaming datasets.

Dataset	Classes	Subjects	Data sources	Instruction Modality	Scenario
MSRC-12 [71]	12	30	Skeleton	Scripted	Actions
MSRAction3D [40]	20	10	Depth +Skeleton	Scripted	Actions
G3D	20	10	Colour +Depth +Skeleton	Scripted	Actions
G3Di	18	12	Colour +Depth +Skeleton	Game-sourced	Actions +Interactions

5.4.1 DATASET ANNOTATION

The ground truth for the action dataset was conventionally annotated by manually labelling each action point and each action segment, whereas the interaction ground truth was automatically constructed from the action ground truth labels. The ground truth interactions are automatically labelled based on the set of rules that govern the interactions for a particular game (as described in Section 4.2).

5.5 Results

In this section experiments are presented to evaluate the ability of the proposed online action and interaction recognition methods to improve accuracy at low latency in complex scenarios. Previously used algorithms are used to determine the complexity of the new dataset in comparison with the existing gaming datasets and to determine the ability of existing approaches to detect compound actions and interactions between multiple subjects.

5.5.1 DATASETS

The performance of the algorithm is evaluated using publicly available datasets designed specifically for real time action and interaction recognition: G3D (introduced in section 3.4) and G3Di (introduced in section 5.4). Both datasets contain multiple actions in each sequence in a controlled indoor environment with a fixed camera, a typical setup for NUI applications. Both datasets provide sequences of skeleton data captured using the Kinect pose estimation pipeline at 30fps. However, G3D contains scripted actions which are temporally well separated whereas G3Di was captured using a game-sourcing approach where multiple users were recorded whilst playing computer games and consequently contains compound actions which overlap temporally. G3Di also contains noisier skeleton data than G3D as there was interference from multiple Kinects during the recording,

making it more realistic of a home scenario where there may be interference from the sunlight.

The G3D dataset contains 10 subjects performing 20 gaming actions grouped into seven categories. The fighting category was selected as it has the same actions as the G3Di boxing category although there are substantial variations in execution rate as well as personal style between these two datasets due to the different recording environments. The G3D fighting category contains five gaming actions: right punch, left punch, right kick, left kick and defend.

The G3Di dataset contains 12 people split into 6 pairs. Each pair interacted through a gaming interface showcasing six sports: boxing, volleyball, football, table tennis, sprint and hurdles. Boxing is a competitive sport and the interactions can be decomposed by an action and counter action. The boxing actions were right punch, left punch and defend and the interactions between the players are shown in Table 5-1. The total number of action and interaction instances used for the experiments is shown in Table 5-3.

Table 5-3 The total number of action and interaction instances used from each dataset

<i>Dataset</i>	Action Classes	Interaction Classes	Subjects	Action Interaction Instances	/ Frames
<i>G3D (Boxing)</i>	5	NA	10	150 actions	12,870
<i>G3Di (Fighting)</i>	3	2	12	317 actions 257 interactions	6,784

5.5.2 SKELETON DATA

Joint angles are viewpoint and anthropometric invariant and can be generated in real-time with a pose estimation method [6]. More specifically, the skeleton poses are first

normalised and then the three angles defining each joint position are computed and represented by a 4-D quaternion. The skeleton is parameterised as a high dimensional feature vector by concatenating quaternions for all joints. For each pose 13 quaternions are calculated so each feature vector has 52-dimensions (see [141] for more details).

5.5.3 PERFORMANCE METRICS

To evaluate the performance of both action and interaction recognition algorithms on the new dataset, action and interaction online metrics and ground truth annotation are required. For action recognition, the latency aware action point F_1 metric used in previous chapters is employed.

For interaction evaluation the existing frame based metrics [128], [138] include temporal constraints but do not evaluate the latency of the detection. To overcome these limitations the action point F_1 metric which evaluates accuracy and latency is extended to cover interactions. The interactions are evaluated in a similar manner to action points, to obtain a single F_1 -score for an easy comparison of different interaction algorithms. The acceptable latency of the interaction is application specific and can be adjusted with the Δ parameter. The Action Point F_1 -score (defined in section 2.5.2.2.3) can be adapted to measure the accuracy of the detected interaction points t_{d_i} against the ground truth interaction points t_{g_i} , by modifying Eq. (2-18) to Eq. (5-3).

$$\Phi_i(t_{d_i}, t_{g_i}, \Delta) = \begin{cases} 1 & \text{if } |t_{g_i} - t_{d_i}| \leq \Delta \\ 0 & \text{otherwise} \end{cases} \quad (5-3)$$

To clarify the assessment of interaction points a dummy timeline for a boxing game has been created (Figure 5-18), showing the ground truth and the detected points for actions and interactions. The precision and recall are measured for each interaction and both of

these measures are combined to calculate a single interaction F_1 -score. To measure accuracy for multiple interactions, the mean interaction F_1 -score is calculated over all interactions.

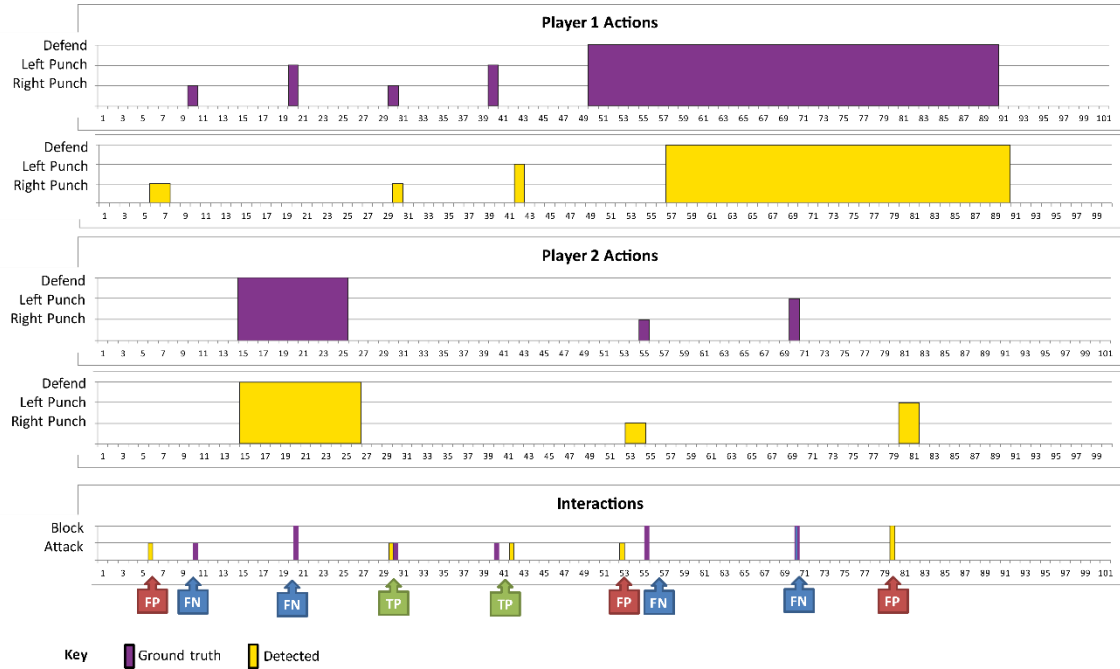


Figure 5-18 A timeline for a boxing game, showing the true positives (TP), false positives (FP) and false negatives (FN). A TP, is a correct interaction identified within Δ frames of the ground truth. A FN, is an undetected interaction on the ground truth.

5.5.4 COMPARATIVE STUDY

The following is a comparison of a range of algorithms which are plugged into the interaction framework illustrated in Figure 5-3. The main methods are AdaBoost, Clustered Spatio-Temporal Manifolds and Hierarchical Transfer Learning. The other methods are used to show specific elements of the proposed method in isolation to validate their effectiveness.

- AdaBoost: AdaBoost has shown high accuracy and low latency for online action recognition [59], [140]. AdaBoost was trained on the source dataset and the parameters: the number of training frames around each peak pose the sliding smoothing window size were optimised on the training part of the target dataset and the method was evaluated on to the target testing data.
- Clustered Spatio-Temporal Manifolds (CSTM): CSTM proposed in chapter 4 was trained on the source dataset and the parameters: the template size and the peak pose detector were optimised on the training part of the target dataset and the method was evaluated on to the target testing data.
- Peak Segment Matching (PSM): is an extension of CSTM which instead of a binary decision for matching a peak key pose introduces a threshold to detect actions with extended duration.
- Body Part Matching (BPM): is an extension of PSM which instead of using the standard DTW in the matching process, uses a selective DTW based on the most discriminative body parts to detect compound actions.
- Transfer Learning Matching (TLM): is an extension of PSM which instead of training and testing on the same dataset. Learns the action templates on a simpler dataset, and performs model adaption on a more complex dataset.
- Hierarchical Transfer Learning (HTL): The proposed method in this chapter, combines the previous three approaches. Transfer learning is applied to Peak Segment Matching, allowing knowledge to be transferred from simple actions in a source dataset to compound actions in a target dataset by adapting the body part models and peak key poses. The parameters: peak segment matching threshold

($T=0.22$) and fragment size ($n_f = 7$) were optimised on the training part of the target dataset and the method was evaluated on the target testing data.

For all the above experiments leave one-person out cross validation on the target dataset was performed; each cross validation fold was trained on 11 subjects and tested on the remaining subject.

5.5.5 ONLINE ACTION AND INTERACTION RESULTS

The proposed method HTL outperforms existing state-of-the-art approaches for fast online action and interaction recognition, as shown in Figure 5-19. Both AdaBoost and CSTM show a significant drop in accurately detecting actions on the G3Di (Fighting) dataset in comparison with previously published results [140] on the G3D (Boxing) dataset. This is significant especially as the G3Di (Fighting) actions are a subset of the G3D (Boxing) actions but confirms the hypothesis that compound actions are more difficult to detect than multiple actions that are temporally well separated.

Additionally, the recognition accuracy for each category of action and interaction is highlighted for a more detailed analysis of each method, as shown in Figure 5-20. A significant outcome is that even though CSTM can detect all of the action categories, it is unable to detect any interactions which are comprised of actions with duration, specifically the block interaction. In addition to showing the limitation of this approach, it also highlights a weakness of the action point metric [13] which does not incorporate the duration of the action peak. Interaction detection is improved by the baseline method Peak Segment Matching (PSM) which instead of a binary decision for matching a peak key pose introduces a threshold which can detect the duration of the peak. The key contributions of this section are the body part template matching (BPM) and the transfer learning strategy (TLM). Individually, applied to the baseline method, these contributions actually decrease the action and interaction recognition but together (HTL) they form a

powerful combination that significantly increases the action and interaction recognition, as shown in Figure 5-19. Intuitively, the body part model is only useful if adapted to the target dataset.

In this thesis, the interest is developing action recognition approaches that are suitable for NUI applications. Research has shown that a delay of 100ms is not perceivable by the user [144]. Therefore, in this section the comparison is against online action recognition methods that are capable of fulfilling this requirement.

Table 5-4 shows that all the methods evaluated are capable of detecting actions with a low average latency of approx. 2 frames, which is equivalent to 66ms. The online action recognition methods with high latency (830-1500ms [14], 2000ms [141]) were not evaluated as they are better suited to other applications.

Figure 5-21 illustrates a typical failure case caused by noisy skeleton data at the action level resulting in an incorrect interaction to be inferred. The main limitation of the proposed approach is that only the skeleton modality is utilised which is subject to interference from sunlight.

The proposed approach outputs a maximum of one action label for each subject for each frame so it cannot manage simultaneous multiple actions at the same time e.g. walking and waving. This limitation does not arise from the underlying algorithm but an implementation decision. Currently, if the distances of multiple actions cross the threshold, the action with the lowest distance is selected. In these cases, the algorithm could be easily adapted to output multiple labels but this would need to be validated on datasets containing multiple simultaneous actions.

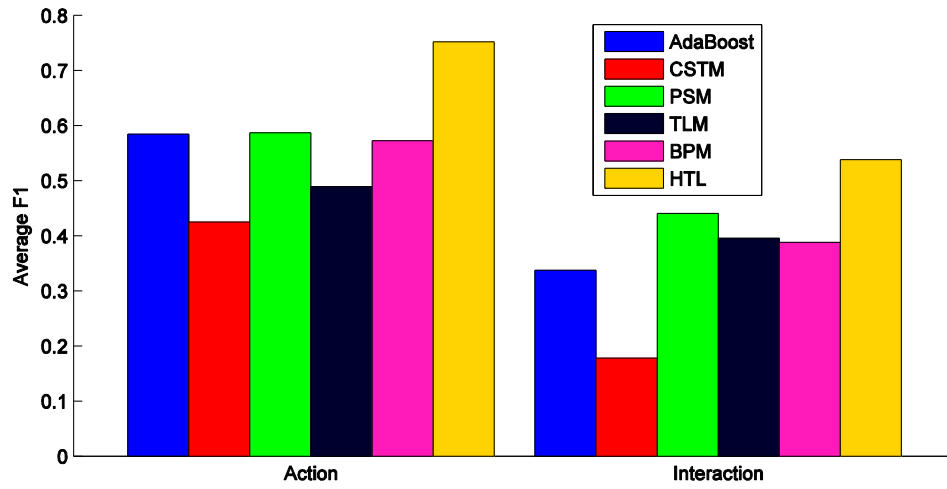


Figure 5-19 Performance comparison of the different approaches. The proposed method (HTL) outperforms the others for both action and interaction detection.

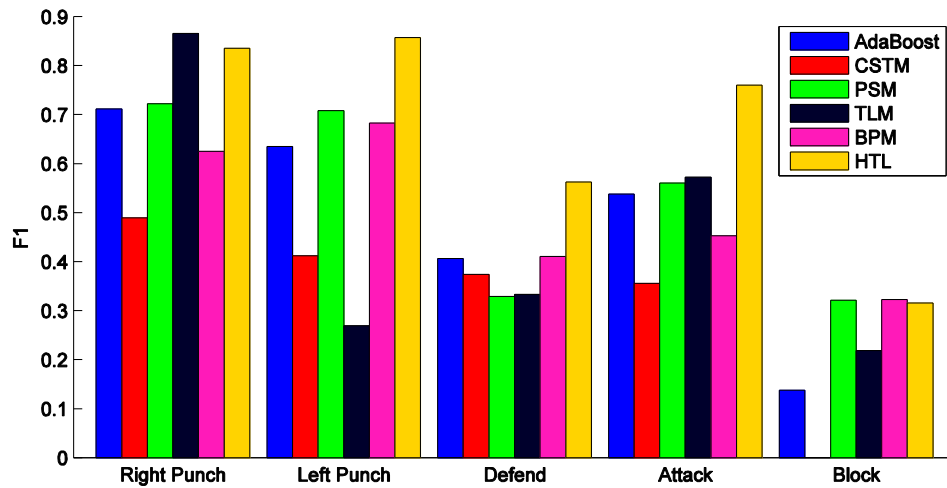


Figure 5-20 Action recognition results (left) and interaction recognition results (right) for each category of the G3Di (Fighting) dataset using different algorithms

Table 5-4 A comparison of the average action latency

Method	Average Action Latency (frames)
AdaBoost	2.12
CSTM	2.00
PSM	1.60
TSM	1.41
BSM	1.94
HTL	2.36

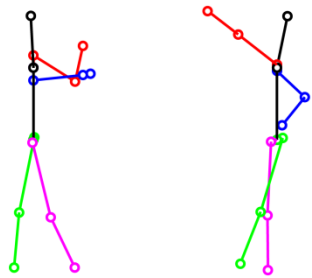


Figure 5-21 Example of a typical failure case caused by noisy skeleton data. The colour image (right) shows that this is a block interaction but the algorithm detects an attack interaction as the defence action is not correctly detected due to incorrect skeleton data for the player on the left. This instance will be penalised twice by the action point metric, firstly a FP for the attack and secondly a FN for the block.

5.6 Summary

In this work a novel Hierarchical Transfer Learning algorithm was proposed for fast online action and interaction recognition. It overcomes the limitations of existing approaches by representing the human body as parts and learning the most discriminative parts needed to detect compound actions. A transfer learning strategy was introduced to allow the tasks of action segmentation and whole body modelling to be performed on a related but simpler dataset. Combined with body part model adaptation on a more complex dataset to introduce independence between limbs and provide the flexibility to match poses that are not in the source dataset. Evaluation on a public target dataset that is more challenging and realistic than the source dataset shows the proposed transfer learning algorithm significantly increases performance at low latency. As the target dataset was recorded whilst users were actually playing a game the actions are more natural than subjects that are given instructions or restrictions and demonstrates the viability of the proposed algorithm for use in real-world applications. The proposed hierarchical interaction framework recognises individual actions with low latency for real-time interaction detection. The incorporation of the action duration in the framework improved both the action and interaction performance.

Furthermore, a novel, realistic and challenging human interaction dataset, G3Di for real time multiplayer gaming was introduced. It overcomes the limitations of existing 3D gaming datasets that only contain a single player with simple action sequences. Sports games introduced the element of competition between the players so the actions captured were more realistic and challenging in comparison to scripted actions. G3Di contains synchronised colour, depth and skeleton and the players were captured from the front view, which improved the quality of the skeleton data. Experimental results indicate higher complexity of the new dataset in comparison to the existing gaming datasets,

highlighting the importance of this dataset for designing algorithms suitable for realistic interactive applications.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Action recognition research historically focused on increasing accuracy on datasets in highly controlled environments. The majority of action recognition algorithms have been applied offline and even the online approaches have high latency. These simplifications have resulted in over-inflated accuracy and action recognition algorithms not suitable for real-time applications. In contrast, this thesis dealt with the more complex problem of online action recognition with low latency in real world scenarios.

6.1 Contributions

In this section the main contributions to fulfil the aim of realistic action recognition are summarised.

6.1.1 REALISTIC GAMING ACTION DATASETS

Perfect or near perfect offline action recognition accuracy on scripted datasets has been achieved. These datasets normally contained a single person that was instructed to perform a single action clearly which over-simplified the task of action recognition.

6.1.1.1 Issues

There are many public action recognition datasets which can be categorised into scripted and realistic scenarios. Movies and sports footage have enabled action recognition from video sequences that are realistic but none of these datasets contain gaming actions. Gaming actions are found within scripted gaming datasets captured by depth sensors but each sequence only contain repetitions of the same action whereas real games contain a

variety of different actions. Additionally, in scripted datasets if there is a delay between actions the subject often returns to the neutral position when changing action. However, in fast paced competitive computer games, like boxing, players do not return to the neutral position between actions, which creates compound actions. Furthermore, the existing gaming datasets are single person whereas commercial games are often multiplayer.

6.1.1.2 Proposed Solutions

Two new gaming datasets, G3D and G3Di were presented for real-time action recognition. G3D was the first public gaming action dataset to contain multiple actions within each sequence making it more like commercial games. G3Di was captured using a novel game-sourcing method so the actions captured were more complex and as realistic as those in commercial games. Additionally, G3D and G3Di are the only two gaming datasets to provide synchronised colour, depth and skeleton data. Experimental results indicate higher complexity of the G3Di dataset, highlighting the importance of this dataset for designing algorithms suitable for real-world applications.

6.1.1.3 Future Work

Due to the technical limitations of the depth sensor used to record both datasets (Kinect for Windows v1), the number of subjects was limited to two players and there was interference from multiple sensors when using the game-sourcing approach. Due to recent technological improvements the Kinect for Windows v2 can track the skeleton of up to six players in real-time and has significantly less interference between multiple sensors. Future work would be to record a new gaming action dataset using the proposed gaming sourcing approach with the latest depth sensor to have more precise colour, depth and joint information as well as more players.

6.1.2 DYNAMIC FEATURE SELECTION

There are many types of machine learning algorithms that have been applied to action recognition but the majority of approaches have been applied offline and even the online approaches have high latency. Both observational and computational latency have been considered when developing the proposed algorithm to ensure that they are suitable for real-world applications.

6.1.2.1 Issues

Most of the existing action recognition algorithms are far from operating online and with low latency. Notable exceptions are AdaBoost and Random Forests with a sliding window to perform continuous action recognition. However, the fixed size of the sliding window in these approaches is a source of error due to execution rate variations. A comparison of Random Forests and AdaBoost showed that AdaBoost can provide higher classification accuracy at the cost of less efficient computation.

6.1.2.2 Proposed Solutions

A novel method for Dynamic Feature Selection for online action recognition was presented that combines the strengths of feature selection with local expert classifiers. Specifically, the feature selection method built in to Random Forest was used to determine feature subsets and then the reduced feature vectors used to train an ensemble of AdaBoost classifiers. In contrast to existing approaches using feature selection, recognition occurs dynamically at each frame to select the most confident classification. Experiments on G3D and MSRC-12 datasets demonstrate that the new Dynamic Feature Selection algorithm for real-time action recognition improves the accuracy of baseline algorithms at low-latency.

6.1.2.3 Future Work

To overcome the fixed sliding window problem of the proposed algorithm the size of the sliding window was reduced to a single pose. However, this resulted in the loss of temporal history of the action and the inability to train the classifier to detect actions with similar poses. There are currently no sliding window approaches that maintain the temporal history and are execution rate invariant, although the multi-scale temporal window is currently the best compromise. Further research into an execution rate invariant sliding window technique could improve the accuracy of existing online action recognition approaches.

6.1.3 CLUSTERED SPATIO-TEMPORAL MANIFOLDS

Existing online action recognition approaches fail to maintain the temporal history of an action in a manner that is execution rate invariant so alternative solutions are investigated. Spatio-temporal manifolds have been previously applied to pre-segmented sequences containing single actions but the key benefit of these manifolds is that they maintain the temporal history of the action which in addition to improving online action recognition could be exploited for early action recognition and even prediction.

6.1.3.1 Issues

Spatial-temporal manifolds are invariant to personal style and execution rate invariant but as the whole sequence is used for classification their observational latency is high which is why they have only previously been applied to offline recognition. The majority of existing approaches for early activity recognition focus on classifying the action as soon as possible and have been applied to pre-segmented sequences. Manual pre-segmentation simplifies the task of early detection which inflates accuracy and limits the applicability of these approaches to real-world scenarios. There is relatively little research into action prediction and it is the most interesting and challenging task especially in scenarios where there is no contextual information.

6.1.3.2 Proposed Solutions

Novel algorithms for early and online action recognition as well as prediction were presented based on Clustered Spatio-Temporal Manifolds. These style invariant compact representation of the dynamics of human action were projected to create action templates. Fragments from the action templates were matched using DTW for execution rate invariance for early recognition of the action. The proposed approach achieved high accuracy and in contrast to existing approaches operates in a continuous stream.

Novel peak key poses were introduced to explicitly locate the moment where an action reaches its peak which enabled low latency recognition before the completion of the action. Experimental results on publicly available gaming action datasets demonstrate state-of-the-art accuracy with very low latency. Furthermore, the peak key poses enabled prediction of the action peak when the recent action progress history was combined with regression.

6.1.3.3 Future Work

The proposed algorithms for early and online action recognition and prediction have only been evaluated with a single player but commercial computer games are often multiplayer. The drop in performance on the task of action prediction tasks highlights the complexity of the problem and is an interesting area for further research.

6.1.4 HIERARCHICAL TRANSFER LEARNING

Evaluation of action recognition algorithms is typically done in isolation, focusing historically on high accuracy and more recently also on low latency. However, in reality most actions form part of an interaction where the duration of the action becomes important.

6.1.4.1 Issues

The diversity and complexity of real-world datasets makes accurate labelling difficult and time consuming. To overcome this, transfer learning has been employed to transfer knowledge from a simpler domain to a more complex target domain. Nevertheless, the existing approaches were limited to offline action recognition. An area that has not been explored before is the potential for transfer learning to improve online action recognition.

6.1.4.2 Proposed Solutions

A novel Hierarchical Transfer Learning framework was proposed for fast online action and interaction recognition. It overcomes the limitations of existing approaches by representing the human body as parts and learning the most discriminative parts needed to detect compound actions.

A transfer learning strategy was introduced to allow the tasks of action segmentation and whole body modelling to be performed on a related but simpler dataset. The transfer learning approach also incorporates body part model adaptation on a more complex dataset to introduce independence between limbs and provide the flexibility to match poses that are not in the source dataset.

Evaluation on G3Di dataset shows the proposed transfer learning algorithm significantly increases performance at low latency. The proposed hierarchical interaction framework recognises individual actions with low latency for real time interaction detection. The incorporation of the action duration in the framework improved both the action and interaction performance.

6.1.4.3 Future Work

Due to computational issues of learning the selection factors to discriminate the body parts were binary, better accuracy may be possible by weighting each body part.. However, an

exhaustive search for the optimum combination of weights would no longer be feasible approach so alternative approaches such as genetic algorithms would need to be investigated.

Another limitation of the proposed algorithms in this thesis is that only the skeleton data is utilised which is subject to interference from sunlight. Future work would be to improve the robustness of the algorithm by fusing features from the depth or colour stream with the skeleton features to evaluate their effectiveness using the G3Di multi-modal dataset.

6.2 Epilogue

This thesis aimed to deal with the problem of complex action recognition in real world scenarios with multiple players. New action recognition datasets were captured that incorporated the challenges of real-world applications. Novel algorithms were developed to overcome these challenges in real-time and advance the study of realistic action recognition. This research is expected to serve as a basis for further study within the research community.

The future of action recognition as demonstrated in this thesis is online rather than offline and recognising multiple rather than single people. This enables a wide range of novel applications including home entertainment, healthcare, sports, and robotics. For example, a personal robotic assistant for the elderly in a care home could naturally interact with staff and patients. In future, it is important not only detect the actions in real-time but to also automatically assess the quality of the action. Automatically assessing the quality of actions using computer vision is a very new topic with limited research. Key challenges are how to determine the quality of an action and how to validate this against expert opinion. Nevertheless, this will extend the range of potential medical applications to

medical diagnosis, home based rehabilitation and ambient assisted living. Similarly, the range of sports applications would increase to include sports training and analysis for improving performance or entertainment. Robust real time action recognition could have a huge impact on society, radically changing the way we interact with machines and revolutionising our lives.

BIBLIOGRAPHY

- [1] J. Devore, "Bowling," 2010. [Online]. Available: <http://bulk2.destructoid.com/ul/176454-e3-10-kinect-sports-in-glorious-picture-form/Bowling-noscale.jpg>. [Accessed: 09-Jul-2015].
- [2] Brontes Processing, "SeeMe system - Professional rehabilitation system with biofeedback," 2015. [Online]. Available: <http://www.virtual-reality-rehabilitation.com/graphics/Set12-cyclops/partSeeMe.jpg>.
- [3] Aldebaran, "Pepper the robot: Intelligent robot." [Online]. Available: <https://www.aldebaran.com/en/a-robots/who-is-pepper>. [Accessed: 11-Aug-2015].
- [4] District-Berne-Knox-Westerlo-Central-School, "The new education 'movement,'" 2012. [Online]. Available: <http://bkwschools.org/elementary/morenews/201213/1024dergositxbox.cfm>. [Accessed: 03-Aug-2015].
- [5] Microsoft, "Kinect for Xbox 360," 2015. [Online]. Available: <http://www.xbox.com/en-US/xbox-360/accessories/kinect>. [Accessed: 09-Jul-2015].
- [6] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, *Real-time human pose recognition in parts from single depth images*, vol. 2, no. 3. IEEE, 2011, pp. 1297–1304.
- [7] A. Yao, J. Gall, G. Fanelli, and L. Van Gool, "Does Human Action Recognition Benefit from Pose Estimation?," *Proceedings of the British Machine Vision Conference 2011*. British Machine Vision Association, pp. 67.1–67.11, 2011.
- [8] Wikiphoto, "Block a Punch Step 1." [Online]. Available: <http://www.wikihow.com/Block-a-Punch#/Image:Block-a-Punch-Step-1.jpg>. [Accessed: 09-Jul-2015].
- [9] H. Liu, R. Feris, and M. Sun, "Benchmarking Datasets for Human Activity Recognition," in *Visual Analysis of Humans*, T. B. Moeslund, A. Hilton, V. Krüger, and L. Sigal, Eds. London: Springer London, 2011, pp. 411–427.
- [10] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004, vol. 3, pp. 32–36 Vol.3.
- [11] T. Starner and a Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," *Comput. Vision, 1995. Proceedings., Int. Symp.*, pp. 265–270, 1995.
- [12] C. Ellis, S. Z. S. Masood, M. M. F. Tappen, J. J. Laviola Jr., and R. Sukthankar, "Exploring the Trade-off Between Accuracy and Observational Latency in Action Recognition," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 420–436, Feb. 2013.

- [13] S. Nowozin and J. Shotton, "Action Points: A Representation for Low-latency Online Human Action Recognition," *Technical Rep.*, pp. 1–18, 2012.
- [14] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng, "Online Human Gesture Recognition from Motion Data Streams," in *ACM Multi-Media 2013*, 2013, pp. 23–32.
- [15] G. W. Records, "Fastest-selling gaming peripheral." [Online]. Available: <http://www.guinnessworldrecords.com/world-records/fastest-selling-gaming-peripheral/>. [Accessed: 09-Jul-2015].
- [16] Microsoft, "Kinect for Windows Sensor Components and Specifications," 2015. [Online]. Available: <https://msdn.microsoft.com/en-us/library/jj131033.aspx>. [Accessed: 14-Jul-2015].
- [17] J. Ashley, "Quick Reference: Kinect 1 vs Kinect 2," 2014. [Online]. Available: <http://www.imaginativeuniversal.com/blog/post/2014/03/05/Quick-Reference-Kinect-1-vs-Kinect-2.aspx>. [Accessed: 03-Aug-2015].
- [18] Asus, "Xtion PRO LIVE." [Online]. Available: https://www.asus.com/uk/Multimedia/Xtion_PRO_LIVE/specifications/. [Accessed: 03-Aug-2015].
- [19] Occipital, "Develop with Depth," 2015. [Online]. Available: <http://structure.io/developers>. [Accessed: 03-Aug-2015].
- [20] A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim, "Shake ' n ' Sense : Reducing Interference for Overlapping Structured Light Depth Cameras," in *Human Factors in Computing Systems*, 2012, pp. 1933–1936.
- [21] E. J. Almazan and G. A. Jones, "Multiple Non-Overlapping RGB-D Sensors for Tracking People," *Robot. Sci. Syst.*, 2013.
- [22] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16:1–16:43, Apr. 2011.
- [23] C. Marais, "Kinect Gesture Detection using Machine Learning," 2011.
- [24] J. K. Aggarwal and L. Xia, "Human Activity Recognition From 3D Data: A Review," *Pattern Recognit. Lett.*, May 2014.
- [25] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, 2001.
- [26] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 405–412.

- [27] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.
- [28] A. Yao, J. Gall, and L. Van Gool, "A hough transform-based voting framework for action recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2061–2068, 2010.
- [29] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A Biologically Inspired System for Action Recognition," in *International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.
- [30] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1932–1939.
- [31] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65–72.
- [32] I. Laptev and T. Lindeberg, "Space-time Interest Points," *Int. J. Comput. Vis.*, vol. 64, pp. 107–123, 2005.
- [33] S. Sempena, N. U. Maulidevi, and P. R. Aryan, "Human action recognition using Dynamic Time Warping," *Proc. 2011 IEEE Int. Conf. Electr. Eng. Informatics*, no. July, pp. 1–5, 2011.
- [34] K. Rapantzikos, Y. Avrithis, and S. Kollias, "Dense saliency-based spationtemporal feature points for action recognition," *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009.
- [35] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [36] A. Kläser, M. Marszalek, and C. Schmid, "A Spatio-Temporal Descriptor Based on 3D gradients," in *BMVC*, 2008.
- [37] G. Willems, T. Tuytelaars, and L. G. An, "Efficient Dense and Scale-Invariant Spatio-Temporal," in *ECCV*, 2008.
- [38] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *British Machine Vision Conference*, 2009, pp. 124.1–124.11.
- [39] E. Yu and J. K. Aggarwal, "Human action recognition with extremities as semantic posture representation," in *Computer Vision and Pattern Recognition Workshops*, 2009, pp. 1–8.
- [40] W. Li and O. M. Way, "Action Recognition Based on A Bag of 3D Points," in *Computer Vision and Pattern Recognition Workshop (CVPRW), 2010 IEEE Conference on*, 2010, pp. 9–14.

- [41] a. Jalal, M. Z. Uddin, J. T. Kim, and T.-S. Kim, "Recognition of Human Home Activities via Depth Silhouettes and R Transformation for Smart Homes," *Indoor Built Environ.*, vol. 21, no. 1, pp. 184–190, 2012.
- [42] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *Proceedings of the 20th ACM international conference on Multimedia - MM '12*, 2012, no. c, p. 1057.
- [43] B. Ni and P. Moulin, "RGBD-HuDaAct: A color-depth video database for human daily activity recognition," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 1147–1153.
- [44] Z. Cheng, L. Qin, Y. Ye, Q. Huang, and Q. Tian, "Human daily action analysis with multi-view and color-depth data," in *ECCV Workshops and Demonstration*, 2012, pp. 52–61.
- [45] Y. Zhao, Z. Liu, L. Yang, and H. Cheng, "Combing RGB and Depth Map Features for human activity recognition," in *APSIPA ASC, IEEE*, 2012, pp. 1–4.
- [46] L. Xia and J. K. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2834–2841.
- [47] J. Wang, "Mining actionlet ensemble for action recognition with depth cameras," *Comput. Vis. Pattern Recognit. (CVPR), 2012 IEEE Conf.*, pp. 1290–1297, Jun. 2012.
- [48] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in *ECCV*, 2012, pp. 872–885.
- [49] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. M. Campos, "STOP: Space-Time Occupancy Patterns for 3D action recognition from depth map sequences," *Prog. Pattern Recognition, Image Anal. Comput. Vision, Appl.*, pp. 252–259, 2012.
- [50] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3D motion understanding," *Int. J. Comput. Vis.*, vol. 95, no. 1, pp. 29–51, 2011.
- [51] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007.
- [52] A. Swadzba, N. Beuter, J. Schmidt, and G. Sagerer, "Tracking objects in 6D for reconstructing static scenes," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, 2008, pp. 1–7.
- [53] S. Fanello, I. Gori, G. Metta, and F. Odone, "Keep it simple and sparse: real-time action recognition," *J. Mach. Learn. Res.*, vol. 14, pp. 2617–2640, 2013.

- [54] A. Letouzey, B. Petit, and E. Boyer, “Scene Flow from Depth and Color Images,” in *Proceedings of the British Machine Vision Conference*, 2011, pp. 46.1–46.11.
- [55] G. Ballin, M. Munaro, and E. Menegatti, “Human Action Recognition from RGB-D Frames Based on Real-Time 3D Optical Flow Estimation,” *Biol. Inspired Cogn. Archit.*, pp. 65–74, 2013.
- [56] M. Müller, A. Baak, and H.-P. Seidel, “Efficient and robust annotation of motion capture data,” *Proc. 2009 ACM SIGGRAPH/Eurographics Symp. Comput. Animat. - SCA '09*, vol. 1, p. 17, 2009.
- [57] S. Z. Masood, C. Ellis, A. Nagaraja, M. F. Tappen, J. J. Laviola, and R. Sukthankar, “Measuring and reducing observational latency when recognizing actions,” in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 422–429.
- [58] J. Sung, C. Ponce, B. Selman, and A. Saxena, “Human Activity Detection from RGBD Images,” *Plan, Act. Intent Recognit.*, 2011.
- [59] J. Liu, J. Luo, and M. Shah, “Recognizing realistic actions from videos ‘in the wild,’” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1996–2003.
- [60] V. Vapnik, *Statistical Learning Theory*. NY: Wiley, 1998.
- [61] A. A. Chaaoui, J. R. Padilla-López, P. Climent-Pérez, and F. Flórez-Revuelta, “Evolutionary joint selection to improve human action recognition with RGB-D devices,” *Expert Syst. Appl.*, vol. 41, no. 3, pp. 786–794, Feb. 2014.
- [62] D. M. Gavrila and L. S. Davis, “3-D model-based tracking of human upper body movement: a multi-view approach,” in *Proceedings of International Symposium on Computer Vision - ISCV*, 1995, pp. 3–8.
- [63] A. Veeraraghavan, R. Chellappa, C. Park, and A. K. Roy-chowdhury, “The Function Space of an Activity,” in *Computer Vision and Pattern Recognition*, 2006, pp. 959 – 968.
- [64] L. R. Rabiner and B.-H. Juang, “An introduction to hidden Markov models,” *ASSP Mag. IEEE*, vol. 3, no. 1, pp. 4–16, 1986.
- [65] J. Yamato, J. Ohya, and K. Ishii, “Recognizing human action in time-sequential images using hidden Markov model,” *Comput. Vis. Pattern Recognit.*, pp. 379–385, 1992.
- [66] Y. Wang, T. Yu, L. Shi, and Z. Li, “Using human body gestures as inputs for gaming via depth analysis,” in *IEEE International Conference on Multimedia and Expo, ICME*, 2008, pp. 993–996.
- [67] A. Galata, N. Johnson, and D. Hogg, “Learning Variable-Length Markov Models of Behavior,” *Comput. Vis. Image Underst.*, vol. 81, no. 3, pp. 398–413, Mar. 2001.

- [68] S. Eickeler, A. Kosmala, and G. Rigoll, "Hidden Markov model based continuous online gesture recognition," in *International Conference on Pattern Recognition*, 1998, vol. 2, pp. 1206–1208.
- [69] P. Natarajan and R. Nevatia, "Online, Real-time Tracking and Recognition of Human Actions," in *IEEE Workshop on Motion and video Computing*, 2008, pp. 1–8.
- [70] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [71] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 1737–1746.
- [72] X. Zhao, S. Wang, X. Li, and H. L. Zhang, "LNCS 7798 - Online Action Recognition by Template Matching," pp. 269–272, 2013.
- [73] A. Sharaf, M. Torki, M. E. Hussein, and M. El-Saban, "Real-time Multi-scale Action Detection From 3D Skeleton Data," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015.
- [74] X. Miao and J. S. Heaton, "A comparison of random forest and Adaboost tree in ecosystem classification in east Mojave Desert," in *Geoinformatics, 2010 18th International Conference on*, pp. 1–6.
- [75] L. Olshen, J. Breiman, R. Friedman, and C. J. Stone, "Classification and Regression Trees," *Wadsworth Int. Gr.*, 1984.
- [76] B. Sadeghi, "An Introduction to Decision Trees with Julia," 2013. [Online]. Available: <http://bensadeghi.com/decision-trees-julia/>.
- [77] D. Benyamin, "A Gentle Introduction to Random Forests, Ensembles, and Performance Metrics in a Commercial System," 2012. [Online]. Available: <https://citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics/>.
- [78] W. Iba and P. Langley, "Induction of One-Level Decision Trees," in *International Conference on Machine Learning*, 1992, pp. 233–240.
- [79] R. Rifkin and A. Klautau, "In Defense of One-Vs-All Classification," *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, Dec. 2004.
- [80] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–53, Dec. 2007.
- [81] C. M. University, "Motion Capture Database," 2011. [Online]. Available: <http://mocap.cs.cmu.edu/>.
- [82] M. Muller, T. Roder, M. Clausen, B. Eberhardt, B. Kruger, and A. Weber, "Documentation Mocap Database HDM05," 2007.

- [83] L. Sigal, A. O. Balan, and M. J. Black, “HumanEva : Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion,” *Int. J. Comput. Vis.*, vol. 87, pp. 4–27, 2010.
- [84] D. Thirde, “PETS: Performance Evaluation of Tracking and Surveillance,” 2005. [Online]. Available: <http://www.cvg.rdg.ac.uk/slides/pets.html>.
- [85] H. Office, *Imagery Library for Intelligent Detection Systems : The i-LIDS User Guide*. 2011.
- [86] M. Marszałek, I. Laptev, and C. Schmid, “Actions in Context,” in *Computer Vision and Pattern Recognition*, 2009, no. i, pp. 2929–2936.
- [87] N. Lazarevic-McManus, J. R. Renno, D. Makris, and G. a. Jones, “An object-based comparative methodology for motion detection based on the F-Measure,” *Comput. Vis. Image Underst.*, vol. 111, no. 1, pp. 74–85, 2008.
- [88] J. A. Rice and B. W. Silverman, “Estimating the Mean and Covariance Structure Nonparametrically When the Data are Curves,” *J. R. Stat. Soc.*, vol. 53, no. 1, pp. pp. 233–243, 1991.
- [89] V. Bloom, V. Argyriou, and D. Makris, “Dynamic Feature Selection for Online Action Recognition,” in *Human Behavior Understanding, Lecture Notes in Computer Science*, vol. LNCS, no. 8212, Switzerland: Springer International Publishing, 2013, pp. 64–76.
- [90] V. Bloom, D. Makris, and V. Argyriou, “G3D: A gaming action dataset and real time action recognition evaluation framework,” in *Computer Vision and Pattern Recognition Workshop (CVPRW), 2012 IEEE Conference on*, 2012, pp. 7–12.
- [91] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.
- [92] E. Amaldi and V. Kann, “On the Approximability of Minimizing Nonzero Variables Or Unsatisfied Relations in Linear Systems,” *Theor. Comput. Sci.*, vol. 209, pp. 237–260, 1998.
- [93] R. Kohavi and G. H. John, “Wrappers for Feature Subset Selection,” *Artif. Intell.*, vol. 97, no. 1, pp. 273–324, 1997.
- [94] P. Climent-Pérez, A. Chaaoui, J. Padilla-López, and F. Flórez-Revuelta, “Optimal Joint Selection for Skeletal Data from RGB-D Devices Using a Genetic Algorithm,” in *Advances in Computational Intelligence*, vol. 7630, 2013, pp. 163–174.
- [95] F. Negin, F. Ozdemir, C. B. Akgul, K. A. Yuksel, and A. Ercil, “A Decision Forest Based Feature Selection Framework for Action Recognition from RGB-Depth Cameras,” in *International Conference on Image Analysis and Recognition*, 2013, pp. 648–657.

- [96] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Human Behavior Understanding*, Springer, 2011, pp. 29–39.
- [97] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 35, no. 1, pp. 221–231, 2013.
- [98] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6316 LNCS, no. PART 6, pp. 140–153, 2010.
- [99] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," 1996.
- [100] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," *Ann. Stat.*, vol. 28, p. 2000, 1998.
- [101] N. Sebanz, H. Bekkering, and G. Knoblich, "Joint action: bodies and minds moving together.," *Trends Cogn. Sci.*, vol. 10, no. 2, pp. 70–6, Mar. 2006.
- [102] S. Streuber, G. Knoblich, N. Sebanz, H. H. Bühlhoff, and S. de la Rosa, "The effect of social context on the use of visual information.," *Exp. brain Res.*, vol. 214, no. 2, pp. 273–84, Oct. 2011.
- [103] K. Verfaillie and A. Daems, "Representing and anticipating human actions in vision," *Vis. cogn.*, vol. 9, no. 1–2, pp. 217–232, Feb. 2002.
- [104] J. Broekens, M. Heerink, and H. Rosendal, "Assistive social robots in elderly care : a review Assistive social robots," *Gerontechnology*, vol. 8, no. 2, 2009.
- [105] L. A. Schwarz, D. Mateus, G. München, and V. Castañeda, "Manifold Learning for ToF-based Human Body Tracking and Activity Recognition," in *British Machine Vision Conference*, 2010, pp. 80.1–80.11.
- [106] M. Lewandowski, D. Makris, and J. Nebel, "Temporal Extension of Laplacian Eigenmaps for Unsupervised Dimensionality Reduction of Time Series," in *International Conference on Pattern Recognition*, 2010, pp. 161 – 164.
- [107] D. Gong and G. Medioni, "Dynamic Manifold Warping for view invariant action recognition," *2011 Int. Conf. Comput. Vis.*, no. 3, pp. 571–578, Nov. 2011.
- [108] M. Lewandowski, D. Makris, S. A. Velastin, and J.-C. Nebel, "Structural Laplacian Eigenmaps for Modeling Sets of Multivariate Sequences.," *IEEE Trans. Cybern.*, Oct. 2013.
- [109] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group," *2014 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014.
- [110] A. Paiement, L. Tao, S. Hannuna, M. Camplani, D. Damen, and M. Mirmehdi, "Online quality assessment of human movement from skeleton data," *Computing*, vol. 27, no. 1, pp. 153–166, 2009.

- [111] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," *2011 Int. Conf. Comput. Vis.*, no. Iccv, pp. 1036–1043, Nov. 2011.
- [112] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. M. Siskind, and S. Wang, "Recognizing Human Activities from Partially Observed Videos," vol. 1.
- [113] J. W. Davis and A. Tyagi, "Minimal-latency human action recognition using reliable-inference," *Image Vis. Comput.*, vol. 24, no. 5, pp. 455–472, May 2006.
- [114] K. Li, Y. Fu, Kang Li, and Yun Fu, "ARMA-HMM: A new approach for early recognition of human activity," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, no. Icp, pp. 1779–1782.
- [115] M. Hoai and F. Torre, "Max-Margin Early Event Detectors," *Int. J. Comput. Vis.*, Dec. 2013.
- [116] T. Lan, T. Chen, and S. Savarese, "A Hierarchical Representation for Future Action Prediction," *Comput. Vision–ECCV 2014*, pp. 689–704, 2014.
- [117] Y. Kong, D. Kit, and Y. Fu, "A Discriminative Model with Multiple Temporal Scales for Action Prediction," in *ECCV 2014 - European Conference on Computer Vision*, 2014, pp. 596–611.
- [118] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating the future by watching unlabeled video," Apr. 2015.
- [119] T. Kanungo, D. M. Mount, N. S. Netanyahu, A. Y. Wu, and C. D. Piatko, "A Local Search Approximation Algorithm for k-Means Clustering," *Spec. Issue 18th Annu. Symp. Comput. Geom. - SoCG2002*, vol. 28, no. 2–3, pp. 89–112, 2003.
- [120] A. W. Drake, "Discrete-state Markov Processes," in *Fundamentals of Applied Probability Theory*, New York: McGraw-Hill, 1967, pp. 163–203.
- [121] P. Senin, "Dynamic Time Warping Algorithm Review," USA, 2008.
- [122] Wikipedia, "Euclidean vs DTW." [Online]. Available: http://upload.wikimedia.org/wikipedia/commons/6/69/Euclidean_vs_DTW.jpg. [Accessed: 07-Feb-2015].
- [123] "DTW algorithm." [Online]. Available: <http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm>. [Accessed: 07-Aug-2015].
- [124] E. Levina and P. J. Bickel, "Maximum Likelihood Estimation of Intrinsic Dimension," *Adv. Neural Inf. Process. Syst.*, vol. 17, 2004.
- [125] D.-A. Huang and K. Kitani, "Action-Reaction: Forecasting the Dynamics of Human Interaction," in *Computer Vision – ECCV 2014*, vol. 8695, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, pp. 489–504.

- [126] Microsoft, “Kinect Sports,” 2010. [Online]. Available: <http://marketplace.xbox.com/en-GB/Product/Kinect-Sports/66acd000-77fe-1000-9115-d8024d5308c9>. [Accessed: 30-Jul-2015].
- [127] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, D. Samaras, and S. Brook, “Two-person Interaction Detection Using Body-Pose Features and Multiple Instance Learning,” in *Computer Vision and Pattern Recognition Workshop (CVPRW), 2012 IEEE Conference on*, 2012, pp. 28–35.
- [128] M. S. Ryoo and J. K. Aggarwal, “UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA),” 2010. [Online]. Available: http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html.
- [129] T. Hu, X. Zhu, W. Guo, and K. Su, “Efficient Interaction Recognition through Positive Action Representation,” *Math. Probl. Eng.*, vol. 2013, pp. 1–11, 2013.
- [130] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. Reid, “High Five: Recognising Human Interactions in TV Shows,” in *BMVC*, 2010.
- [131] S. Pan and Q. Yang, “A survey on transfer learning,” ... *Data Eng. IEEE Trans.*, vol. 22, no. 10, 2010.
- [132] A. Farhadi and M. Tabrizi, “Learning to recognize activities from the wrong view point,” *Comput. Vision–ECCV 2008*, pp. 154–166, 2008.
- [133] J. Liu and M. Shah, “Cross-view action recognition via view knowledge transfer,” *Comput. Vis. ...*, 2011.
- [134] D. Weinland, R. Ronfard, and E. Boyer, “Free viewpoint action recognition using motion history volumes,” *Comput. Vis. Image Underst.*, vol. 104, no. 2–3, pp. 249–257, Nov. 2006.
- [135] Z. Ma, Y. Yang, F. Nie, N. Sebe, S. Yan, and A. G. Hauptmann, “Harnessing Lab Knowledge for Real-World Action Recognition,” *Int. J. Comput. Vis.*, vol. 109, no. 1–2, pp. 60–73, Apr. 2014.
- [136] L. Cao, Z. Liu, and T. Huang, “Cross-dataset action detection,” *Comput. Vis. pattern ...*, 2010.
- [137] B. Chakraborty, A. D. Bagdanov, J. González, and X. Roca, “Human action recognition using an ensemble of body-part detectors,” *Expert Syst.*, vol. 30, no. 2, pp. 101–114, 2013.
- [138] S. Escalera, X. Baro, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, P. V., H. J. Escalante, J. Shotton, and I. Guyon, “ChaLearn Looking at People Challenge 2014: Dataset and Results,” in *ECCV Workshop*, 2014.
- [139] A. Gilbert, J. Illingworth, R. Bowden, and S. Member, “Action Recognition Using Mined Hierarchical Compound Features,” *TPAMI*, vol. 33, no. 5, pp. 883–897, 2011.

- [140] V. Bloom, D. Makris, and V. Argyriou, “Clustered Spatio-temporal Manifolds for Online Action Recognition,” in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2014, pp. 3963–3968.
- [141] A. Chaaraoui and F. Flórez-Revuelta, “Continuous Human Action Recognition in Ambient Assisted Living Scenarios,” in *First International Workshop on Enhanced Living Environments (ELEMENT)*, 2014, pp. 1–8.
- [142] I. Kviatkovsky, E. Rivlin, and I. Shimshoni, “Online action recognition using covariance of shape and motion,” *Comput. Vis. Image Underst.*, vol. 129, pp. 15–26, Dec. 2014.
- [143] V. Bloom, V. Argyriou, and D. Makris, “G3Di : A Gaming Interaction Dataset with a Real Time Detection and Evaluation Framework,” in *European Conf. on Computer Vision Workshops (ECCVW)*, 2014.
- [144] S. K. Card, G. G. Robertson, and J. D. Mackinlay, “The information visualizer: An information workspace,” in *Proc. ACM CHI*, 1991, pp. 181–188.

APPENDIX

8.1 Symbol table

Symbols	Details	Meaning	Method
a	$a \in \mathbb{R}^1$	Action index	General
a'	$a' \in \mathbb{R}^1$	Action index	General
a^*	$a^* \in \mathbb{R}^1$	Action index	General
a_e	$a_e \in \mathbb{R}^1$	Action segment end	HLE
a_s	$a_s \in \mathbb{R}^1$	Action segment start	HLE
a'_e	$a'_e \in \mathbb{R}^1$	Counter action segment end	HLE
a'_s	$a'_s \in \mathbb{R}^1$	Counter action segment start	HLE
\mathbf{b}	$(\mathbf{b}_{i_b})_{(i_b=1..n_b)}$	Low level body parts	HLE
\mathbf{c}	$\mathbf{c} \in \mathbb{R}^d$	Low dimensional cluster	CSTM
d	$d \in \mathbb{R}^1$	Number of dimensions in the low dimensional space	General
d_I		Function to get the depth of a pixel in image I	Pose Estimation
f^G		Function to extract a fragment from a sequence of poses	CSTM
f^{pdx}		Difference feature function for x	Pose-based features
f^{pdy}		Difference feature function for y	Pose-based features
f^{pdz}		Difference feature function for z	Pose-based features

f^{pvx}		Position velocity feature function for x	Pose-based features
f^{pvy}		Position velocity feature function for y	Pose-based features
f^{pvz}		Position velocity feature function for z	Pose-based features
f^{pvd}		Position velocity magnitude function	Pose-based features
f^q	$f^q \in \mathbb{C}^4$	Quaternion	
f^{qd}		Angle velocity function	Pose-based features
f^Ω		OVA function	OVA
f^θ		Depth comparison function	Pose estimation
f^δ		Euclidean distance	DTW
f_Q^L	$f_Q^L \in \{1 \dots n_q\}$	Stretching Q time axis	DTW
f_R^L	$f_R^L \in \{1 \dots n_r\}$	Stretching R time axis	DTW

f^D		Dynamic Time Warping Function	DTW
f^W		Weighted Dynamic Time Warping Function	DTW
g	$g \in \mathbb{R}^D$	Training pose in target dataset	HTL
h	$h \in \mathbb{R}^D$	Testing pose in target dataset	HTL
i		Use with subscript for first index	Reserved
i_b	$i_b \in 1 \dots n_b$	Body part index	HTL
i_c	$i_c \in 1 \dots n_c$	Index of cluster, Row index of transition matrix	CSTM
i_f	$i_f \in 1 \dots n_c$	Index of the last pose in a fragment	CSTM
i_g	$i_g \in 1 \dots n_g$	Index of training pose TARGET dataset	HTL
i_h	$i_h \in 1 \dots n_h$	Index of testing pose TARGET dataset	HTL
i_k	$i_k \in 1 \dots n_c$	Key pose index	CSTM
i_l	$i_l \in 1 \dots n_c$	Last matched key pose index	CSTM
i_m	$i_m \in 1 \dots n_c$	Matching key pose index	CSTM
i_o	$i_o \in 1 \dots n_c$	Ordered index of cluster	CSTM
i_p	$i_p \in 1 \dots n_c$	Peak key pose index	CSTM
i_q	$i_q \in 1 \dots n_q$	Query sequence index	CSTM

i_r	$i_r \in 1 \dots n_r$	Reference sequence index	CSTM
i_s	$i_s \in 1 \dots n_s$	Index of pose in a sequence	CSTM
i_t	$i_t \in 1 \dots n_t$	Index of testing pose	CSTM
i_w	$i_w \in 1 \dots W$	Index of weak classifier	AdaBoost
i_L	$i_L \in 1 \dots n_L$	Warping path index	DTW
i_η	$i_\eta \in 1 \dots n_\eta$	Peak pose index in training data	CSTM
i_ω	$i_\omega \in 1 \dots n_\omega$	Index of gini impurity	Decision Trees
i_Ω	$i_\Omega \in 1 \dots \Omega$	Index of binary classifier	OVA
j		Use with subscript for second index	Reserved
j_c	$j_c \in 1 \dots n_c$	Column index of transition matrix	CSTM
j_f	$j_f \in 1 \dots n_c$		
j_i	$j_i \in 1 \dots n_j$	Column index of transition matrix	CSTM
j_ω	$j_\omega \in 1 \dots n_\omega$	Index of gini impurity	Decision Trees
\mathbf{k}	$\mathbf{k} \in \mathbb{R}^D$	Key pose	CSTM
m		Pixel	Pose estimation
n		Use with index to represent number of items	Reserved
n_b	$n_b \in \mathbb{Z}^1$	Number of body parts	HTL

n_c	$n_c \in \mathbb{Z}^1$	Number of clusters	CSTM
n_f	$n_f \in \mathbb{Z}^1$	Number of poses in a fragment	CSTM/HTL
n_g	$n_g \in \mathbb{Z}^1$	Number of poses in training dataset (TARGET)	HTL
n_h	$n_h \in \mathbb{Z}^1$	Number of poses in testing dataset (TARGET)	HTL
n_j	$n_j \in \mathbb{Z}^1$	Number of joints in a pose	General
n_k	$n_k \in \mathbb{Z}^1$	Maximum number of allowed key poses after peak key pose	CSTM
n_m	$n_m \in \mathbb{Z}^1$	Number of sequential matched poses	CSTM
n_q	$n_q \in \mathbb{Z}^1$	Number of poses in query sequence	CSTM
n_r	$n_r \in \mathbb{Z}^1$	Number of poses in reference sequence	CSTM
n_s	$n_s \in \mathbb{Z}^1$	Number of poses in a generic sequence	CSTM
n_t	$n_t \in \mathbb{Z}^1$	Number of poses in testing set	CSTM
n_u	$n_u \in \mathbb{Z}^1$	Number of temporal neighbours	CSTM
n_v	$n_v \in \mathbb{Z}^1$	Number of repetition neighbours	CSTM
n_w	$n_w \in \mathbb{Z}^1$	Number of frames in smoothing window	DFS

n_L	$n_L \in \mathbb{Z}^1$	Length of the warping path	DTW
n_T	$n_T \in \mathbb{Z}^1$	Number of trees	DFS
n_η	$n_\eta \in \mathbb{Z}^1$	Number of peak poses in the training data	CSTM
n_τ	$n_\tau \in \mathbb{Z}^1$	Number of thresholds / feature subsets	DFS
\mathbf{o}	$\mathbf{o} = (o_{i_c})_{(i_c=1\dots n_c)}$ $o_{i_c} \in \mathbb{R}^1$	Temporal cluster order	CSTM
$p_{j_i,t}$	$p_{j_i,t} \in \mathbb{R}^3$	The 3D location (x^c, y^c, z^c) of joint j_i at time t	General
\mathbf{q}	$\mathbf{q} \in \mathbb{R}^D$	Query pose	CTLE
\mathbf{r}	$\mathbf{r} \in \mathbb{R}^D$	Reference pose	CTLE
\mathbf{s}	$\mathbf{s} \in \mathbb{R}^D$	Pose	CTLE
t		Reserved for time, use subscript	
t_c	$t_c \in \mathbb{R}^1$	Timing constraint	HTL
t_{da}	$t_d \in \mathbb{R}^1$	Detection time of the action point	Evaluation Metrics
t_{ga}	$t_g \in \mathbb{R}^1$	Ground truth time of the action point	Evaluation Metrics
t_{pa}	$t_p \in \mathbb{R}^1$	Estimated time of the action point	Evaluation Metrics
t_{di}	$t_d \in \mathbb{R}^1$	Detection time of the interaction point	Evaluation Metrics
t_{gi}	$t_g \in \mathbb{R}^1$	Ground truth time of the interaction point	Evaluation Metrics

u		Depth image feature offsets	Pose estimation
v		Depth image feature offsets	Pose estimation
w^a	$\mathbf{w}^a = (w_{i_b})_{(i_b=1\dots n_b)}$ $w_{i_b} \in [0,1]$	Weights for low level body parts	HTL
x	$\mathbf{x} \in \mathbb{R}^D$	Feature vector for a pose	DFS/CSTM
x^c	$x^c \in \mathbb{R}^1$	x co-ordinate of 3D joint location	Pose based features
y	$\mathbf{y} \in \mathbb{R}^d$	Manifold points	CSTM
y^c	$y^c \in \mathbb{R}^1$	y co-ordinate of 3D joint location	Pose based features
z	$\mathbf{z} \in \mathbb{R}^D$	Feature vector for a testing pose	CSTM/HTL
z^c	$z^c \in \mathbb{R}^1$	z co-ordinate of 3D joint location	Pose based features
A	$A \in \mathbb{R}^1$	Number of actions	General
B	$\mathbf{B} = (\mathbf{b}_{i_b})_{(i_b=1\dots n_b)}$	Body part model	HTL
C	$\mathbf{C} = \{\mathbf{c}_{i_c}\}_{(i_c=1\dots n_c)}$ $\mathbf{c}_{i_c} \in \mathbb{R}^d$	Low dimensional cluster centers (unordered)	CSTM
D	$D \in \mathbb{R}^1$	Number of features in the high dimensional space	General
F		Reserved	
F₁	$F_1 \in \mathbb{R}^1$	F ₁ - score	General
G	$\mathbf{G} = (\mathbf{g}_{i_g})_{(i_g=1\dots n_g)}$ $\mathbf{g}_{i_g} \in \mathbb{R}^D$	High dimensional poses from TARGET training data set	HTL

H	$\mathbf{H} = (\mathbf{h}_{i_h})_{(i_h=1\dots n_h)}$ $\mathbf{h}_{i_h} \in \mathbb{R}^D$	High dimensional poses from TARGET testing data set	HTL
I		Image	Pose estimation
K^a	$\mathbf{K}^a = (\mathbf{k}_{i_o})_{(i_o=o_1\dots o_{n_c})}$ $\mathbf{k}_{i_o} \in \mathbb{R}^D$	Action templates containing ordered key poses	CSTM
L	$L \in \mathbb{R}^{n_L \times 2}$	Warping path indices	CTLE
M			
N		Node in decision tree	General
P		Function of the fraction of patterns	Decision Trees
Q	$\mathbf{Q} = (\mathbf{q}_{i_q})_{(i_q=1\dots n_q)}$ $\mathbf{q}_{i_q} \in \mathbb{R}^D$	Query sequence of poses	CSTM
R	$\mathbf{R} = (\mathbf{r}_{i_r})_{(i_r=1\dots n_r)}$ $\mathbf{r}_{i_r} \in \mathbb{R}^D$	Reference sequence of poses	CSTM
S	$\mathbf{S} = (\mathbf{s}_{i_s})_{(i_s=1\dots n_s)}$ $\mathbf{s}_{i_s} \in \mathbb{R}^D$	Sequence of poses	CSTM
T	$T \in \mathbb{R}^1$	Threshold for peak segment matching	HLE
W	$W \in \mathbb{R}^1$	Number of weak classifiers	AdaBoost
X	$\mathbf{X} = (\mathbf{x}_{i_r})_{(i_r=1\dots n_r)}$ $\mathbf{x}_{i_r} \in \mathbb{R}^D$	High dimensional poses from training data set	CSTM
Y	$\mathbf{Y} = (\mathbf{y}_{i_r})_{(i_r=1\dots n_r)}$ $\mathbf{y}_{i_r} \in \mathbb{R}^d$	Low dimensional poses from training data set	CSTM
Z	$\mathbf{Z} = (\mathbf{z}_{i_t})_{(i_t=1\dots n_t)}$	Testing poses	CSTM

	$\mathbf{z}_{i_e} \in \mathbb{R}^D$		
α	$\alpha \in \mathbb{R}^1$	Gradient regression line	HTL
β	$\beta \in \mathbb{R}^1$	Intersection regression line	HTL
γ		Gini impurity function	Decision trees
δ	$\delta \in \mathbb{R}^1$	Accumulated distortion on the DTW path	DTW
ε		Permutation	HTL
ζ		Peak pose matches	CSTM/HTL
η^a	$\eta^a = (\eta_{i_\eta})_{(i_\eta=1\dots n_\eta)}$ $\eta_{i_\eta} \in \mathbb{R}^1$	The peak poses indices in training set for each action	HTL
θ	$\theta = (\theta_{i_\theta})_{(i_\theta=1\dots n_\theta)}$ $\theta_{i_\theta} \in \mathbb{R}^1$	Sequential key pose matches of the same class	HTL
λ	$\lambda \in \mathbb{R}^1$	Cluster transition probability	CSTM
μ	$\mu \in \mathbb{Z}^1$	Number of intra-class matches	HTL
ξ		Weighted vote	AdaBoost
ρ	$\rho \in \mathbb{R}^1$	Inter class ratio	HTL
τ	$\tau = \tau_1, \tau_2, \dots, \tau_{n_\tau}$	Thresholds for feature importance subsets	DFS
φ		Peak pose detection function	CSTM
χ	$\chi = \{\mathbf{y}_{i_r}, \mathbf{x}_{i_r}\}_{(i_r=1\dots n_r)}$	Training set for radial Basis Function Network	CSTM
ψ		Weak classifier	AdaBoost

ω		Class	General
Γ	$\Gamma \in \mathbb{R}^{n_q \times n_r}$	DTW Pairwise distance matrix	DTW
Δ	$\Delta \in \mathbb{R}^1$	Latency (ms)	Evaluation metric
Λ	$\Lambda = (\lambda_{i_c j_c})_{(i_c=1 \dots n_c, j_c=1 \dots n_c)}$ $\lambda_{ij} \in \mathbb{R}^1$	Transition matrix of cluster transition probabilities	CSTM
Φ_a		Action point detection evaluation function	Evaluation metric
Φ_p		Action point prediction evaluation function	Evaluation metric
Φ_i		Interaction point detection evaluation function	Evaluation metric
Ψ		Strong classifier	AdaBoost
Ω		Number of binary classifiers	OVA

8.2 Ethics

Research Proposal

1. Applicant Information

Main Applicant: Victoria Bloom, DIRC, Kingston University
Experimental Dates: January 2012 – October 2014

2. Research Proposal

2.1 Background and Rational for Research

The gaming industry in recent years has attracted an increasing large and diverse group of people. A new generation of games based on natural interaction such as dance and sports games have increased the appeal of gaming to family members of all ages. (1)

The latest technological advancement in natural interaction is the Kinect developed by Microsoft for the Xbox 360 games console. Microsoft's slogan is "You are the controller", which captures their concept of a controller free experience allowing the user to control the game with body movements. The titles released to date for the Kinect include driving, dance or sports games that recognise a small set of actions.

There is a vast wealth of research on human action recognition in computer vision and this project will combine it with gaming to advance the state of the art methods for action recognition. These algorithms will be optimised for performance which is one of the main issues in video games. The algorithms will be trained to recognise a wide range of actions including sporting, driving and action-adventure actions such as walking, running, jumping, dropping, firing, changing weapon, throwing and defending. This could increase the complexity and appeal of games that will developed to include action-adventure games similar to Lara Croft.

The restricted environment associated with gaming, typically the users lounge poses unique challenges for human action recognition. The challenges are related to the lack of context in the lounge where the normal background and objects usually associated with a given action are missing. For example, performing a golf swing in a real golf game would require a golf club and may take place on a green field. Performing a golf swing in a Kinect game the user has no golf club and is performing the action in their lounge. This lack of contextual information may mean that the state of the art appearance-based action recognition approaches may under perform.

However, due to recent progress in pose estimation by Microsoft research group (2) early pose based approaches are being revisited by action recognition researchers. Yao et al. (3) experiments showed that pose based features outperform low-level appearance features in a home monitoring scenario. Pose based action recognition approaches may be the solution to the contextual challenges faced in the gaming environment and warrant further investigation.

To compare the performance of both the appearance and pose based approaches a dataset of a range of gaming actions is required containing video, depth and skeleton data. There are already publicly available datasets with sports and locomotion actions containing video and skeleton data (4-6). However, as mentioned previously there is a difference between performing a real action and a gaming action. Even simple actions such as walking are different in the gaming environment as the player will be walking on the spot. The MPI HDM05 Motion

Capture Database (5) database does include locomotion on the spot. However, to get the full range of gaming actions required it is necessary to record our own dataset. To encourage further research in the field of action recognition in gaming it is intended to make the dataset publicly available online.

2.2 Research Design and Protocol

Study Design – Same for full study and pilot study

20-30 healthy subjects between the ages of 18-65 with different morphologies, weights, heights and clothing. This diversity, as well as the considerable size of the test sample, is required in order to obtain activity models capable of generalising over a population of different subjects.

Subjects will report to the laboratory on one occasion to complete the testing process. Subjects will be asked to perform different gaming actions (walking on the spot, running on the spot, kicking, punching etc.). To examine the validity of the proposed protocol during the pilot study, subjects will be asked to perform each of the activities several times.

Participants and Recruitment

Subjects will be recruited from the Kingston University population. An invitation to participate will be sent via a StudentSpace notice. All subjects will participate on a completely voluntary basis and will be asked to complete an informed consent form to check their suitability to participate.

Location

All testing will take place either in the Biomechanics Laboratory (EM03) at Kingston University or if mocap data is not required room SB122 or SB329 at Kingston University.

Procedures

Subjects will be asked to perform several different natural interaction gaming actions, walking on the spot, running on the spot, kicking, punching etc. Subjects will have their activities captured during the trials using Microsoft's Kinect. The Kinect can be used as a motion capture system that does not require the user to wear any markers / special clothing or hold any controllers. The device contains an infrared projector and sensor that measures depth and a video camera for capturing images. It can be used in conjunction with software to produce skeleton data (joint positions and angles) of users in its field of view.

To compare the performance of both the appearance and pose based action recognition approaches the data capture will need to include **the image, depth and skeleton data** provided by the Kinect.

If initial tests prove that the Kinect skeleton data is not yet robust enough then **motion capture data** will also need to be recorded. A motion capture system consists of several infrared cameras and a set of reflective landmarks attached to the body. It is also used in conjunction with software to produce skeleton data (joint positions and angles) of users in its field of view.

Statistical Analysis

Features will be extracted from the captured data and used as input to the machine learning algorithms developed for action recognition. The features extracted will differ for each type of data captured. The features extracted from the image will be low level features such as colour, spatial and temporal gradients and dense optical flow. The features extracted from the depth map will be low level such as pixel depth or higher level such as silhouettes and visual hulls. The features

extracted from the skeleton data will be low level such as joint positions and angles and high level such as qualitative geometric features (3).

The captured features will then be split into three datasets and used to train, validate and test the machine learning action recognition algorithms developed so the performance of the algorithms can be compared.

Data Storage and Confidentiality

All personal data entered on the informed consent forms will be kept in a locked cabinet within the Digital Imaging Research Centre and will conform to the Data Protection Act 1998. Personal data will be kept for further research of the Human Body Group once this project has finished. Personal data will be stored indefinitely.

All recorded data (image, depth and skeleton) will be made publicly available on a Kingston University webpage. No individual personal data will accompany this data. However, as the video contains colour images of the participants they may be identifiable. Recorded data will remain public once this project has finished.

Only the investigators will have access to both personal data and recorded data collected from the study. In order to have access to the personal data, new researchers belonging to the group should ask for permission filling a form and justifying their necessity. In case of approval, they could have access to the personal data during the period of their particular project.

Consent and approval for video, depth and skeleton data to be posted on the internet will be sought after the activity has taken place and the participants have had the opportunity to view their footage.

If a participant decides to withdraw their consent, they can request removing any data that allows their identification at any time. In this case, their video and/or depth and/or skeleton and/or MoCap data will be deleted from the dataset. However, if this data has already been made public there is no guarantee that someone has not already made a copy of the footage concerned.

3 Ethical Considerations

3.1 Informed Consent

Prior to participation, all subjects will be required to complete an Informed Consent and Health Screening questionnaire (please see appendixes below) in order to attain their suitability for the investigation and potential contraindications to exercise. All participants are free to leave the trial at any point without question.

3.2 Risk Assessment

The risks involved in completing this investigation are minimal, since the activities to be performed are similar to those experienced playing Kinect games for the Xbox 360.

3.3 Confidentiality

Please see section 2.2 (*Data Storage and Confidentiality*) for confidentiality measures.

3.4 Conflicting Interests

No researchers involved in this investigation have any conflicting interests or stand to gain financially from the outcome of the testing.

3.5 Bodily Contact

There may be minimal bodily contact such as touching hands to perform co-operative gaming actions, the participants will have a clear idea of the nature of any bodily contact in advance. There will be no force in the bodily contact (no punching or kicking etc.) so there is no risk of any injuries from the contact.

4 Risks and Benefits

All potential risks will be conveyed to the participants clearly through the information sheet and informed consent document.

Subjects may enjoy performing the range of gaming actions and will have the opportunity see a silhouette and skeleton representation of their body.

5 References

- (1) The Entertainment Software Association - The Transformation of the Video Game Industry Available at: <http://www.theesa.com/gamesindailylife/transformation.asp>. Accessed 11/21/2011, 2011.
- (2) Real-Time Human Pose Recognition in Parts from Single Depth Images. ; jun; ; 2011.
- (3) Does Human Action Recognition Benefit from Pose Estimation? Proceedings of the British Machine Vision Conference: BMVA Press; 2011.
- (4) Carnegie Mellon University - CMU Graphics Lab - motion capture library Available at: <http://mocap.cs.cmu.edu/>. Accessed 11/21/2011, 2011.
- (5) Motion Database HDM05 Available at: <http://www.mpi-inf.mpg.de/resources/HDM05/>. Accessed 11/21/2011, 2011.
- (6) HumanEva Available at: <http://vision.cs.brown.edu/humaneva/index.html>. Accessed 11/21/2011, 2011.

Participant Information Sheet

Kingston University London

Validation Study of Action Recognition in Video games

Thank you for showing interest in this current investigation. If you choose to take part in the investigation you will be asked to fill out and sign an informed consent form to make sure there are no current contraindications to your participation. If you choose not to participate in the investigation, thank you for your time. All information obtained during the course of the study will be kept completely confidential. If after reading this sheet you have any questions regarding the project please feel free to ask before you complete the informed consent form.

What is the purpose of the study?

The aim of this study is to examine algorithms for natural interaction with video games performed by applying computer vision techniques.

Why have I been chosen?

We are looking for subjects aged between 18 and 65 years old with different morphologies, weights, heights and clothing styles.

Do I have to take part?

No. Once you have read this information sheet the choice is yours. Even after this time you are free to withdraw from the investigation at any time without any negative effects.

What will happen to me if I take part?

You will be asked to attend the Kingston University Biomechanics Laboratory for one session where you will be asked to perform different gaming actions (walking on the spot, running on the spot, kicking, punching etc.). The study will take place using a Microsoft Kinect video and depth camera that registers the activity and movements that you will perform. In addition, a Motion Capture system may be used which consists of several cameras able to register the position of a set of reflective markers attached to your body.

What are the possible disadvantages and risks of taking part?

The risks involved in completing this investigation are minimal, since the activities to be performed are similar to those experienced playing Kinect games for the Xbox 360.

What happens when the research study stops?

You will be under no obligation for any further testing. Once the data is analysed, you will be able to obtain a full set of data if you wish.

Will my taking part be kept confidential?

All personal information entered on this form will be kept strictly confidential and kept in secure storage.

All recorded data (image, depth and skeleton) will be made publicly available on a Kingston University webpage. They may be used for research purposes by researchers in Kingston University or other institutions. Parts of the data may appear in academic papers or public research presentations. No individual personal data will accompany this data. However, as the video contains colour images you could be identified.

Your consent and approval for video, depth and skeleton data to be posted on the internet will be sought after the activity has taken place and you will have had the opportunity to view your footage. If you wish to withdraw your consent at any time, all recorded data (image, depth and skeleton) will be deleted from the dataset. However, if this data has already been made public there is no guarantee that someone has not already made a copy of the footage concerned.

Who is organising and funding the study?

This study is organised by members of staff from Faculty of Science, Engineering and Computing, Kingston University. No one involved in the study stands to gain financially from the investigation.

What will happen to the results of the research study?

The results may be presented at national and international conferences as well as in scientific journals.

All personal details, as previously stated, will be kept confidential but you may be identified from the video images.

Who has reviewed the study?

The Kingston University Faculty of Science, Engineering and Computing Faculty Research Ethics Committee has reviewed the study.

Contact for further information.

Further information may be obtained from:

Victoria Bloom,
Faculty of Science, Engineering and Computing,
Kingston University,
Penrhyn Road,
Kingston Upon Thames,
Surrey,
KT1 2EE.
Tel: +44 (0) 020 8547 2000 Ext. 62923
Email: k1044104@kingston.ac.uk

Kingston University London

Faculty of Science, Engineering and Computing

Digital Imaging Research Centre

Informed Consent Form for Kinect Recording - Strictly Confidential

Name: Date of Birth:

Contact Number: Email:

Height:..... Weight:

Sex: Left or right handed:

Please answer the following questions truthfully and completely. The purpose of this questionnaire is to establish that you are of sound body to participate in the current investigation. Please answer questions 1-4 now and the remaining questions **after** the activity has taken place.

Q.1) How would you classify your current activity level? Please indicate below:

Low Moderate High Very High

Q.2) Do you suffer from or have every suffered from any injured or condition that will cause changes to the way you walk or move?

Yes No

Details:

.....
.....

Q.3) Do you know of any reason why you should not to participate in the proposed exercise testing protocol? If yes please give details.

Yes No

Details:

.....
.....

Q.4) Please confirm that you have read and fully understand the Participant Information sheet provided.

Yes No

Q.5) Have you had the opportunity to view your recorded video footage?

Yes No

Q.6) Do you permit the usage of the recorded video for the purposes of research on Computer Vision algorithms?.

Yes No

Q.7) Do you permit the publication of the recorded video in scientific papers, conferences, workshops and websites for the purposes of research on Computer Vision algorithms?

Yes No

Q.8) Have you had the opportunity to view your recorded depth data?

Yes No

Q.9) Do you permit the usage of the recorded depth data for the purposes of research on Computer Vision algorithms?.

Yes No

Q.10) Do you permit the publication of the recorded depth data in scientific papers, conferences, workshops and websites for the purposes of research on Computer Vision algorithms?

Yes No

Q.11) Have you had the opportunity to view your recorded skeleton data?

Yes No

Q.12) Do you permit the usage of the recorded skeleton data for the purposes of research on Computer Vision algorithms?

Yes No

Q.13) Do you permit the publication of the recorded skeleton data in scientific papers, conferences, workshops and websites for the purposes of research on Computer Vision algorithms?

Yes No

Statement by participant

I confirm that I have read and understood the information sheet/letter of invitation for this study. I have been informed of the purpose, risks, and benefits of taking part.

(Title of Study)-----

I understand what my involvement will entail and any questions have been answered to my satisfaction.

I understand that my participation is entirely voluntary, and that I can withdraw or request destroying any data that I may be identifiable at any time before the data becomes public without prejudice.

I understand that all information obtained will be treated with the strictest of confidence.

I understand that research data gathered for the study may be published and be disseminated to the scientific community, and that I may be identified as a subject. (please delete if you disagree).

Contact information has been provided should I wish to seek further information from the investigator at any time for purposes of clarification.

Participant's Signature:

Date:

Statement by investigator

I have explained this project and the implications of participation in it to this participant without bias and I believe that the consent is informed and that he/she understands the implications of participation.

Name of investigator:

Signature of investigator:

Date:

Kingston University London

Faculty of Science, Engineering and Computing

Digital Imaging Research Centre

Informed Consent Form for Testing in the Biomechanics Laboratory - Strictly Confidential

Name: Date of Birth:

Contact Number: Email:

Height:..... Weight:

Sex: Left or right handed:

Please answer the following questions truthfully and completely. The purpose of this questionnaire is to establish that you are of sound body to participate in the current investigation. Please answer questions 1-4 now and the remaining questions **after** the activity has taken place.

Q.1) How would you classify your current activity level? Please indicate below:

Low Moderate High Very High

Q.2) Do you suffer from or have every suffered from any injured or condition that will cause changes to the way you walk or move?

Yes No

Details:

.....
.....

Q.3) Do you know of any reason why you should not to participate in the proposed exercise testing protocol? If yes please give details.

Yes No

Details:

.....
.....

Q.4) Please confirm that you have read and fully understand the Participant Information sheet provided.

Yes No

Q.5) Have you had the opportunity to view your recorded skeleton data?

Yes No

Q.6) Do you permit the usage of the recorded skeleton data for the purposes of research on Computer Vision algorithms?

Yes No

Q.7) Do you permit the publication of the recorded skeleton data in scientific papers, conferences, workshops and websites for the purposes of research on Computer Vision algorithms?

Yes No

Statement by participant

I confirm that I have read and understood the information sheet/letter of invitation for this study. I have been informed of the purpose, risks, and benefits of taking part.

(Title of Study)-----

I understand what my involvement will entail and any questions have been answered to my satisfaction.

I understand that my participation is entirely voluntary, and that I can withdraw or request destroying any data that I may be identifiable at any time without prejudice.

I understand that all information obtained will be treated with the strictest of confidence.

I understand that research data gathered for the study may be published and be disseminated to the scientific community, and that I may be identified as a subject. (please delete if you disagree).

Contact information has been provided should I wish to seek further information from the investigator at any time for purposes of clarification.

Participant's Signature:

Date:

Statement by investigator

I have explained this project and the implications of participation in it to this participant without bias and I believe that the consent is informed and that he/she understands the implications of participation.

Name of investigator:

Signature of investigator:

Date:

APPLICATION FORM FOR ETHICAL REVIEW RE4

SECTION A

Project title:

Multiple Action Recognition in Video Games

Name of the lead applicant:

Name (Title / first name / surname):	Miss. Victoria Bloom
Position held:	PhD Student
Department/School/Faculty:	Faculty of Science, Engineering and Computing
Telephone:	+44 (0) 020 8547 2000 Ext. 62923
Email address:	k1044104@kingston.ac.uk

Name of co-applicants:

Name (Title / first name / surname):	Dr. Dimitrios Makris
Position held:	Reader
Department/School/Faculty:	Faculty of Science, Engineering and Computing
Telephone:	+44 (0) 020 8547 2000 Ext. 67082
Email address:	D.Makris@kingston.ac.uk

Name (Title / first name / surname):	Dr. Vasileios Argyriou
Position held:	Senior Lecturer
Department/School/Faculty:	Faculty of Science, Engineering and Computing
Telephone:	+44 (0) 020 8547 2000 Ext. 62591
Email address:	Vasileios.Argyriou@kingston.ac.uk

Is the project

- Student research
- KU Staff research
- Research on KU premises

Yes	X	No	
Yes	X	No	
Yes	X	No	

If it is STUDENT research: Course: PhD Multiple Action Recognition in Video Games

Supervisor/DoS: Dr. Dimitrios Makris _____

SECTION B

Has approval for the project already been granted by another ethics committee?

Yes No

If **NO**, proceed to **Section C**;

If **YES**, please complete the rest of this section before going to the declaration in **Section D**:

Name of the committee: _____ Date of approval: _____

Please attach the submission made to that committee, together with the approval letter. The Faculty Research Ethics Committee (FREC) may require further information or clarification from you and you should not embark on the project until you receive notification from the FREC that recognition of the approval has been granted.

SECTION C

Briefly describe the procedures to be used in this research involving human participants

Subjects will be asked to perform different natural interaction gaming actions, walking on the spot, running on the spot, kicking, punching etc. Subjects will have their activities captured during the trials using Microsoft's Kinect and by a Motion Capture (mocap) system.

Summarise the data sources to be used in the project:

The project will require 4 types of data related to human objects:

1. Motion Capture (MoCap) data of human objects captured at the Biomechanics Lab (Sports Science) in Kingston University the commercial system "Qualysis". MoCap data describes the motion parameters of 3D human articulated motion. The output is a text file with the coordinates of a set of markers that have been attached to the subject. The data will be anonymous and does not allow person identification (e.g. no image is captured)
2. Video data captured using the Kinect from volunteers that have given their consent. Video or image extracts may be published only if volunteers give their consent.
3. Depth data captured using the Kinect from volunteers that have given their consent. The data will be anonymous. Depth data may be published only if volunteers give their consent.
4. Skeleton data captured using the Kinect from volunteers that have given their consent. The data will be anonymous. Skeleton data may be published only if volunteers give their consent.
5. The HumanEva Video and MoCap data (<http://vision.cs.brown.edu/humaneva/>) which has been published for research purposes.
6. The Motion Database HDM05 MoCap data (<http://www.mpi-inf.mpg.de/resources/HDM05/>) which has been published for research purposes.
7. The CMU MoCap Dataset MoCap data (<http://vision.cs.brown.edu/humaneva/>) which has been published for research purposes.

Estimate duration of the project (months): _____ 36 months _____

State the source of funding: ___ PhD studentship funded by SEC Faculty ___

Is it collaborative research?

Yes No

If YES, name of the collaborator institutions:

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____

Provide a brief project description (max. 150 words). This should be written for a lay audience

There is a vast wealth of research on human action recognition in computer vision and this project will combine it with gaming to advance the state of the art methods for action recognition. These algorithms will be optimised for performance and trained to recognise a wide range of actions.

Due to recent progress in pose estimation by Microsoft research group early pose based approaches are being revisited by action recognition researchers. Pose based action recognition approaches may be the solution to the contextual challenges faced in the gaming environment and warrant further investigation.

To get the full range of gaming actions required for training and testing the algorithms developed it is necessary to record our own dataset. To encourage further research in the field of action recognition in gaming it is intended to make the dataset publicly available online.

Risk Assessment: Does the proposed research involve any of the following?

Children or young people under 18 years of age?	Yes	<input type="checkbox"/>	No	<input checked="" type="checkbox"/>
---	-----	--------------------------	----	-------------------------------------

If YES, have you complied with the requirements of the CRB? YES NO

People with an intellectual or mental impairment, temporary or permanent?	Yes	<input type="checkbox"/>	No	<input checked="" type="checkbox"/>
---	-----	--------------------------	----	-------------------------------------

People highly dependent on medical care, e.g., emergency care, intensive care, neonatal intensive care, terminally ill, or unconscious?	Yes	<input type="checkbox"/>	No	<input checked="" type="checkbox"/>
---	-----	--------------------------	----	-------------------------------------

Prisoners, illegal immigrants or financially destitute?	Yes	<input type="checkbox"/>	No	<input checked="" type="checkbox"/>
---	-----	--------------------------	----	-------------------------------------

Pregnant women?	Yes	<input type="checkbox"/>	No	<input checked="" type="checkbox"/>
-----------------	-----	--------------------------	----	-------------------------------------

Will people from a specific ethnic, cultural or indigenous group be involved, or have the potential to be involved in the proposed research?	Yes	<input type="checkbox"/>	No	<input checked="" type="checkbox"/>
--	-----	--------------------------	----	-------------------------------------

Assisted reproductive technology?	Yes		No	x
Human genetic research?	Yes		No	x
Epidemiology research?	Yes		No	x
Stem cell research?	Yes		No	x
Use of environmentally toxic chemicals?	Yes		No	x
Use of radioactive substances?	Yes		No	x
Ingestion of potentially harmful or harmful dose of foods, fluids or drugs?	Yes		No	x
Contravention of social/cultural boundaries?	Yes		No	x
Involves use of data without prior consent?	Yes		No	x
Involves bodily contact?	Yes	x	No	
Compromising professional boundaries between participants and researchers?	Yes		No	x
Deception of participants, concealment or covert observation?	Yes		No	x
Will this research significantly affect the health* outcomes or health services of subjects or communities?	Yes		No	x
Note* health is defined as not just the physical well-being of the individual but also the social, emotional and cultural well-being of the whole community.				
Is there a potential for enduring physical and/or psychological harm/distress to participants?	Yes		No	x
Does your research raise any issues of personal safety for you or other researchers involved in the project? (especially if taking place outside working hours or off University premises)	Yes		No	x
Will the research be conducted without written informed consent being obtained from the participants?	Yes		No	x
Will financial/in kind payments (other than reasonable expenses and compensation for time) be offered to participants? (Indicate in the proposal	Yes		No	x

how much and on what basis this has been decided)				
---	--	--	--	--

Is there a potential danger to participants in case of accidental unauthorised access to data?	Yes	No	x
--	-----	----	---

N.B. If you have answered YES to any of these questions, you should address them fully in your project proposal and show that there are adequate controls in place.

Storage, access and disposal of data

Describe what research data will be stored, where, for what period of time, the measures that will be put in place to ensure security of the data, who will have access to the data, and the method and timing of disposal of the data. *(Reference to the relevant paragraphs of the Ethics Guidance to be added)*

All personal data entered on the informed consent forms will be kept in a locked cabinet within the Digital Image Research Centre and will conform to the Data Protection Act 1998. Personal data will be kept for further research of the Human Body Group once this project has finished. Personal data will be stored indefinitely.

All recorded data (image, depth and skeleton) will be made publicly available on a Kingston University webpage. No individual personal data will accompany this data. However, as the video contains colour images of the participants they may be identifiable. Recorded data will remain public once this project has finished.

Only the investigators will have access to the personal data collected from the study.

In order to have access to the personal data, new researchers belonging to the group should ask for permission filling a form and justifying their necessity. In case of approval, they could have access to the personal data during the period of their particular project.

Consent and approval for video, depth and skeleton data to be posted on the internet will be sought after the activity has taken place and the participants have had the opportunity to view their footage.

If a participant decides to withdraw their consent, they can request removing any data that allows their identification at any time. In this case, their video and/or depth and/or skeleton and/or MoCap data will be deleted from the dataset. However, if this data has already been made public there is no guarantee that someone has not already made a copy of the footage concerned.

SECTION D

To be signed by all applicants

Declaration to be signed by the applicant(s) and the supervisor (in the case of a student):

- I confirm that the research will be undertaken in accordance with the Kingston University *Guidance and procedures for undertaking research involving human participants*
- I will undertake to report formally to the relevant Faculty Research Ethics Committee for continuing review approval.
- I shall ensure that any changes in approved research protocols are reported promptly for approval by the relevant Faculty Research Ethics Committee.
- I shall ensure that the research study complies with the law and University policy on Health and Safety.
- I confirm that the research study is compliant with the requirements of the Criminal Records Bureau where applicable.
- I am satisfied that the research study is compliant with the Data Protection Act 1998, and that necessary arrangements have been, or will be made with regard to the storage and processing of participants' personal information and generally, to ensure confidentiality of such data supplied and generated in the course of the research.
(Note: Where relevant, further advice should be sought from the Data Protection Officer, University Secretary's Office)
- I shall ensure that the research is undertaken in accordance with the University's Single Equality Scheme.
- I will ensure that all adverse or unforeseen problems arising from the research project are reported immediately to the Chair of the relevant Faculty Research Ethics Committee.
- I will undertake to provide notification when the study is complete and if it fails to start or is abandoned;
- (For supervisors, *if the applicant is a student*) I have met and advised the student on the ethical aspects of the study design, and am satisfied that it complies with the current professional (*where relevant*), departmental and University guidelines. I accept responsibility for the conduct of this research and the maintenance of any consent documents as required by this Committee.
- I understand that failure to provide accurate information can invalidate ethical approval.

Signature of lead applicant: *V. Bloom*Date:...01/02/2012.....

Signature of co-applicant: ... *[Signature]*Date:... 01/02/2012.....

Signature of co-applicant: ... *V.Argyriou*Date:... 01/02/2012.....

Signature of co-applicant:Date:.....

Signature of supervisor:.....Date.....

CHECKLIST

Please complete the checklist and attach it to your application:

Project title: ____ Multiple Action Recognition in Video Games _____

Lead Applicant: __ Miss Victoria Bloom _____

Date of application: __ 1st February, 2012 _____

Before submitting this application, please check that you have done the following: (N/A = not applicable)	Applicant			Committee use only		
	Yes	No	N/A	Yes	No	N/A
All questions have been answered	x					
All applicants have signed the application form	x					
The research proposal is attached	x					
Correspondence from other ethics committees is attached			x			
Informed Consent Form is attached	x					
Participant Information Sheets are attached	x					
All letters, advertisements, posters or other recruitment material to be used are attached	x					
All surveys, questionnaires, interview/focus group schedules, data sheets, etc, to be used in collecting data are attached	x					
Reference list attached, where applicable	x					

RESEARCH PROPOSAL GUIDELINES

Provide a description of the proposed research plan and procedures, using the following headings. Show clearly that the research protocol gives adequate consideration to participants' welfare, rights, beliefs, perceptions, customs and that cultural heritage, both individual and collective, will be respected in the course of your research.

Research plan and protocols

- What is the rationale for the research?
- What is the research design/method?
- Where will the project be conducted?
- What is the participant group(s) and why has it been selected?
- How many participants will be recruited and what is the rationale for that number?
- How, by whom, and where, will potential participants be selected and approached to receive the invitation to participate? (*Attach a copy of letters, advertisements, posters or other recruitment material to be used*)
- How much time will potential participants have to consider the invitation to participate?
- What is required of participants? (*Attach a copy of any testing protocols, interview schedules, data sheets, informed consent, etc to be used.*)
- *Relevant experience of researchers*
- Data storage and access to data
- *Explain how the information you receive will be analysed/interpreted and reported. What specific approaches or techniques (statistical or qualitative) will be employed?*
- Dissemination

Ethical consideration

- How will voluntary participation be ensured?
- Is active consent being sought from all participants for all aspects of the research involving them? If No, why not?
- How will participants' privacy be protected during the recruitment process, or access to tissue samples, or access to records?
- What are the benefits and risks to participants and how will risks be minimised?
- Are there any potential conflicts of interest for the researchers?
- Do the researchers have any affiliation with, or financial involvement in, any organisation or entity with direct or indirect interests in the subject matter or materials of this research? Do the researchers expect to obtain any direct or indirect financial or other benefits from conducting this research?
- Are there any restrictions on the publication of the results of this study? If yes, who has imposed them and what are they?
- Will the research involve payments/rewards/inducements to participants?
- How will confidentiality/anonymity of information received be ensured?
- Any other ethical issues specific to your research?

Risk/benefit analysis

- Clearly justify any potential risks to participants (however minimal) by the potential benefits of the research.
- Disclose any foreseeable risks (for example the discomfort of having your views challenged by others in a focus group, or that associated with negative feedback about a learning assessment).
- Direct benefit to participants
- How risks and benefits identified here will be communicated to the participants (e.g., through the informed consent document)?
- Identify any costs and compensation