Kingston University London

Human fall detection methodologies: from machine learning using acted data to fall modelling using myoskeletal simulation

by

GEORGIOS MASTORAKIS

in the

Faculty of Science, Engineering and Computing

Computer Science and Mathematics

July 2018

This thesis is being submitted in accordance with the requirements for the degree of Doctor of Philosophy (PhD) to the Kingston University London

Declaration of Authorship

- This thesis is submitted as requirement for a Ph.D. Degree in the School of Computer Science and Mathematics (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 Jan 2011 to Feb 2012 and from Nov 2015 to July 2018 under the supervision of Prof. D Makris and Prof. T Ellis in the Digital Information Research Centre (DIRC) of Kingston University, London.

Copyright statement

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial Share Alike licence (CC BY-NC-SA). Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and if remix, transform, or build upon the material, they must distribute their contributions under the same license as the original. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Georgios Mastorakis

16.07.2018

Kingston University London

Abstract

Faculty of Science, Engineering and Computing
Computer Science and Mathematics

Doctor of Philosophy

Human fall detection methodologies: from machine learning using acted data to fall modelling using myoskeletal simulation

by GEORGIOS MASTORAKIS

Human Fall Detection is a research area with interest from many disciplines and aims to perform for many assisted-living monitoring applications to promptly identify life-threatening situations. A fall occurs when a person is unable to maintain balance due to a variety of issues; physical; mental or environmental. The accurate detection of the fall is crucial as a missed detection can be fatal. Variability of human physiological characteristics is currently unstudied as to the impact on a fall detector's performance as young adults and elderly are expected to fall differently. Another important issue is the scene occlusions. In the use of visual sensors, an occluded fall is treated as a missed detection as the whereabouts of the person is unknown when occluded. Finally, current studies are based on acted fall datasets on which algorithms are trained. These dataset are unrepresentative of real fall events and illustrate the events without occlusions or other scene influences.

Several fall detection algorithms were developed during the study aiming to achieve accuracy in detection falls while fall-like actions such as lying down remain undetected. Human fall datasets were used for training and testing purposes of A machine learning algorithm using data from depth cameras which captured the fall events from different views. A new pathway was introduced tackling the issues of availability issues of data-driven machine learning approaches which was achieved with the use of simulation data. The use of myoskeletal simulation was then selected as a closer representation of the human body in terms of structure and behaviour. With the use of a simulation model, a personalised estimation of the fall event can be achieved as it is parametrised on a physical characteristic such

as the height of the falling person. Alternative technologies such as accelerometers have been used for fall detection to prove the validity of this approach on other modalities. A study regarding the impact of occlusions for fall detection which is one of the issues not properly investigated in current work is proposed and examined. Synthetic occlusions were added to existing depth data from publicly available datasets.

The research methodologies were evaluated using the most representative depth video and accelerometer data from existing datasets, as well as YouTube videos of real-fall events. The machine learning methodologies achieved good results on similar body variability datasets. A discussion regarding the proof of concept of the simulation-based approach for fall modelling is mentioned given the comparative results against existing methodologies which achieves better than any existing work evaluated against known datasets. The simulation approach is also evaluated against occluded fall and non-fall event data, proving the further robustness of the approach. This platform can be expanded to analyse any type of fall, or body posture (e.g. elderly), without the use of humans to performs fall events.

Acknowledgements

This work would not have been possible without the support of my supervisors Prof. D Makris and Prof T Ellis. I am grateful for their patience, advice and guidance over the last three years. It was definitely not an easy journey but with their support I remained determined and motivated. The depth of their knowledge, their approach and expertise added significantly to my own personal development and the experience has been invaluable.

I wish to express my thanks to the members (past and present) of the Digital Information Research Centre. I specifically wish to thank Prof. Graeme Jones, Dr Gordon Hunter, Dr Jean-Christophe Nebel and Dr Francisco Florez Revuelta, for their useful insight. I would like to extend my thanks to Rosalind Percival. Also, my interns Xavier Hildenbrand, Kevin Grand, for their helpful collaboration and colleagues Ioannis Kazantzidis, Dr Juan Fernandez Montenegro and Matthias Pilz for their friendship and guidance throughout my PhD.

Last but not least, I owe a huge thanks to a very special person in my life, my wife, Mina for her continued and unfailing love, unending support and immense understanding during my pursuit of PhD degree that made our common goal, the completion of this thesis, possible.

Contents

Abstract **iii**

Acknowledgements **vi**

List of Figures **xiii**

List of Tables **xvii**

Abbreviations **xix**

Symbols **xxi**

1 Introduction **1**
1.1 Challenges 4
1.2 Aims & Objectives 6
1.3 Contributions and Thesis Overview 6

2 Literature Review **9**
2.1 Fall detection features 10
2.2 Wearable sensors 11
2.3 Fixed location sensors 12
2.3.1 RGB systems 13
2.4 Combinatory Systems 14
2.5 Algorithms 15
2.5.1 Use of RGB data 15
2.5.2 Usage of depth data 17
2.5.3 Use of accelerometer data 18
2.6 Discussion 19
2.7 Conclusion 22

3 Datasets **23**
3.1 Introduction 23
3.2 Fall Types 25

3.3 Public camera (2D) datasets . . . 26
3.3.1 Single camera LE2i dataset . . . 26
3.3.2 Multiple cameras fall dataset . . . 26
3.4 Depth Sensor data . . . 28
3.4.1 Depth sensors, OpenNI and Microsoft Kinect SDK . . . 29
3.5 Public RGB-D datasets . . . 29
3.5.1 TST Fall Detection v2 . . . 29
3.5.2 UR Fall Detection . . . 30
3.5.3 SDUFall . . . 31
3.5.4 University of Texas datasets . . . 31
3.5.5 ACT42 dataset . . . 32
3.5.6 Daily Living Activity Recognition . . . 33
3.5.7 NTU RGB+D Action Recognition Dataset . . . 33
3.5.8 UWA3D Multiview Activity dataset . . . 34
3.6 Accelerometer based dataset . . . 35
3.6.1 Sisfall dataset . . . 35
3.7 GM depth dataset . . . 35
3.8 Limitations of existing datasets: A discussion . . . 37
3.8.1 Age of participants . . . 38
3.8.2 Health of participants . . . 39
3.8.3 Types of falls . . . 39
3.8.4 Size of datasets . . . 39
3.8.5 Variability of subjects . . . 40
3.8.6 Hesitation . . . 40
3.9 Scene set-up . . . 40
3.9.1 Occlusions . . . 41
3.9.2 Sensor location . . . 42
3.9.3 Data quality and adaptation . . . 42
3.10 Comparison of real vs acted falls . . . 43
3.11 Performance Evaluation Measures . . . 45
3.12 Conclusion . . . 46

4 Learning to detect falls 49
4.1 Introduction . . . 49
4.1.1 Review of 3D vision systems . . . 50
4.1.2 Technical Criticism of 3D methods . . . 50
4.2 Detecting rigid falls . . . 52
4.2.1 Overview . . . 53
4.2.2 3D Bounding box data analysis . . . 54
4.2.2.1 Fall initiation by velocity . . . 55
4.2.2.2 Completion state of a fall by inactivity detection . 57
4.3 Detecting collapsing and rigid falls . . . 58
4.3.1 Overview . . . 59
4.3.2 Conservative Bounding box analysis . . . 59

4.3.2.1 The ϕ angle . . . 60
4.3.2.2 Angular velocity . . . 62
4.3.2.3 Fall detection after inactivity . . . 63
4.4 Experimental Results and Discussion . . . 64
4.4.1 Training . . . 64
4.4.2 Algorithm 1 Results . . . 65
4.4.3 Algorithm 2 Results . . . 68
4.4.4 Algorithm 1 vs Algorithm 2 . . . 68
4.5 Conclusion . . . 69

5 Simulation: Modelling Fall 71
5.1 Introduction . . . 71
5.2 Review . . . 72
5.2.1 Physics simulation - Synthetic approaches . . . 73
5.3 Modelling falls . . . 74
5.3.1 Falling rod simulations . . . 75
5.3.1.1 Rigid Falling Rod . . . 75
Derivation of 5.1 . . . 76
5.3.2 Myoskeletal human model simulation . . . 78
5.3.2.1 Fall simulations . . . 79
Simulated model preparation . . . 79
Rigid fall . . . 80
Collapsing fall . . . 80
5.3.2.2 ADL simulation . . . 82
5.3.2.3 Scaling the model . . . 83
5.4 Evaluation of the fall simulation models . . . 84
5.5 Conclusion . . . 85

6 Fall detection based on myoskeletal simulation 87
6.1 Introduction . . . 87
6.2 Fall detection using a Hybrid approach . . . 89
6.3 Methodology . . . 90
6.3.1 Estimating Body Orientation . . . 90
6.3.2 Vertical velocity of CoM . . . 94
6.3.3 Hybrid Fall Detection Algorithm . . . 94
6.4 Experimental Results . . . 97
6.4.1 Evaluation of Body Orientation Estimation . . . 97
6.5 Evaluation of the Hybrid Detection algorithm . . . 98
6.6 Discussion . . . 100
6.7 Fall detection using Myoskeletal simulation . . . 100
6.7.1 Use of Hausdorff distance . . . 101
6.7.2 Methodology . . . 102
6.7.2.1 Data pre-processing . . . 102
6.7.2.2 Fall detection . . . 103

6.8 Experimental Results and Discussion . . . 106
6.8.1 Validation of Hausdorff Distance . . . 106
6.8.2 Comparison of the fall simulation models . . . 107
6.8.3 Evaluation of myoskeletal simulation based fall detection . . 108
6.8.4 Evaluation of using a customised simulated model . . . 109
6.8.5 Use of simulation data in machine learning . . . 111
6.9 Acceleration based fall detection using myoskeletal simulation . . . 112
6.9.1 Methodology . . . 112
6.9.2 Results and discussion . . . 113
6.10 Conclusion . . . 115

7 Occlusion robust fall detection 117
7.1 Review . . . 118
7.2 Modelling occlusions . . . 120
7.2.1 Simple Occlusion models . . . 120
7.2.2 Complex Occlusion models – partial/realistic . . . 120
7.2.3 Truncated fall measurements . . . 122
7.3 Fall detection under occlusions . . . 122
7.3.1 Evaluation of algorithm under occlusion . . . 124
7.4 Conclusion . . . 127

8 Conclusion 129
8.1 Summary . . . 129
8.1.1 Evaluation of data . . . 129
8.1.2 Learning approaches . . . 130
8.1.3 The use of simulation for fall detection . . . 130
8.1.3.1 Simulation approaches . . . 131
8.1.4 Occlusion robustness . . . 132
8.1.4.1 Occlusion evaluation protocol . . . 132
8.2 Future Work . . . 133
8.2.1 Other Fall Types and Scene Simulation . . . 133
8.2.2 Other types of myoskeletal models . . . 133
8.2.3 Further personalisation . . . 133
8.2.4 Applied to other technologies . . . 134
8.3 Epilogue . . . 134

A AppendixA 157
A.1 Visual results of Algorithm 1 from Chapter 4 . . . 157

B AppendixB 163
B.1 Feet resistance rod model . . . 163
B.2 Two piece rod models . . . 164

C AppendixC 167

C.1 Published Work 167

List of Figures

3.1 Visuals from the LE2i dataset [1]: fall events at top row, ADLs on lower row . . . 27
3.2 Visuals from the Multiple cameras fall dataset [2]: fall events at top row, ADLs on lower row . . . 27
3.3 Visuals from the TST Fall Detection v2 dataset [3] . . . 30
3.4 Visuals from the UR Fall Detection dataset [4] : a hesitated fall . . 30
3.5 Visuals from the SDUFall dataset [5] : fall events at top row, ADLs on lower row . . . 31
3.6 Visuals from EDF [6] and OCCU [7] datasets. Top row: falls repetition over four angles, lower row: occluded fall behind a bed 32
3.7 Visuals from the ACT42 dataset [8]. Events captured from four views in RGB-D . . . 33
3.8 Visuals from the Daily Living Activity Recognition dataset [9]. Several ADLs and fall samples of RGB-D and skeleton . . . 34
3.9 Visuals from the NTU RGB+D Action Recognition dataset [10]. Several ADLs and fall samples in RGB-D . . . 34
3.10 Visuals from the UWA3D dataset [11]. Performance of a bending over (top row) and falling (low row) . . . 35
3.11 Visuals from the dataset. Row a) a detected fall event, b) walking and picking up an object, c) lying down . . . 36
3.12 Three view-angle examples of the GM dataset. a) 45 view , b) side view, c) front view . . . 36
3.13 Typical occluded scene. The camera view is partially blocked by the red box. Half the person is occluded. . . . 41
3.14 Velocity profiles of collapsing falls. This Figure shows the velocity variation between a hesitated fall ($GM2_A$ hes.) and how this compares with an actual fall caused by hyperventilation . . . 44
3.15 Plot of a Gaussian pdf fitted to the distribution of Hausdorff distances: red curve denotes the HDs of YouTube to non-hesitant falls, while blue curve the HDs between YouTube and hesitant falls . . . 44
3.16 Hesitation while collapsing. Subject uses an arm to balance, then expands her legs to sit on the floor . . . 45

4.1 Depth map of the scene. User is identified by OpenNI . . . 53

4.2 2D Bounding box during a fall; the height reduces while the width increases (a) as seen in [12, 13], where the initial and final bounding box dimensions are required. The proposed approach using a 3D bounding box of the height and the composition of width and depth (b) 55
4.3 Bounding box dimensions and velocities. a) width, height, depth distances and width–depth graph. Vertical lines denote the initiation by velocity step and the fall detection confirmation step. b) $\dot{L}$ raw signal in green and filtered using Kalman in red. Similarly in (c) $\dot{H}$ of raw and filtered signal. 56
4.4 Side view of a sideways fall. Bounding box already detects the user (a), fall initiated by calculating velocity (b), inactivity detected (c), fall detected (d) 57
4.5 Conservative bounding box (red), ordinary bounding box (blue) ρ angle of 3D bounding box top corners and CoM, ϕ angle of conservative bounding box corners and CoM 60
4.6 The angle from the two opposite top corners of the 3D bounding box to the centre of mass 60
4.7 Applying standard deviation on subject's depth pixels to determine the depth dimension on the conservative bounding box. Z is the depth size of the ordinary 3D bounding box, while the depth of the conservative bounding box equals twice the SD value. Three examples: a)standing, b)lifting arm and leg, c)extend arms to the opposite direction. Graph shows the accumulation of depth pixels . 61
4.8 3D angle during a collapsing fall. 62
4.9 3D conservative bounding box angle of a collapsing fall. Notice how the angle increases as the bounding box height reduces 62
4.10 Collapsing fall detection example using Algorithm 2. (a) detection of bounding box, (b) person collapses and knees hit the floor, (c) angular velocity exceeds threshold while person inclines towards the floor, (d) fall detected when inactivity and angle size conditions are fulfilled 63
4.11 Circles indicate 100 triplets estimated by random search for training the rigid fall algorithm. Their median ($t_vH = 1.18m/s, T_vDW = 1.20m/s, N = 8frames$) is marked as a bold circle and is used for the experiments 66

5.1 Falling rod, of length L with uniform mass m end-point vertical velocity V_y, CoM vertical velocity V_{CoM_y} and ω 75
5.2 Typical rod model velocities. The final velocity is proportional to the length of the rod. This velocity is measured from the top point until the rod reaches the horizontal position 77
5.3 Sequence of an actual fall event as captured by a depth camera and of a fall simulated by OpenSim 78

5.4 Images sampled from a YouTube video of a) person acting a rigid fall and b) person suffering a collapsing fall following hyperventilation which includes a faint . . . 79
5.5 Feet position of simulated model before fall . . . 80
5.6 The Full Body Model given by OpenSim engine. Blue marker denotes the head location point, while pink markers denote the MoCap relevant markers . . . 81
5.7 Three types of rigid fall, backward (top), forward (middle) and sideways (bottom) as simulated on OpenSim . . . 81
5.8 Collapsing fall as simulated by OpenSim . . . 82

6.1 Angles of knees and elbows. Green line shows the bisector of the left body side and red of the right . . . 91
6.2 Angle thresholds as defined in Eq. 6.1 . . . 92
6.3 Falling angle thresholds denoted by colour. Blue: front, green: back, yellow: left, magenta: right . . . 93
6.4 The velocity profiles of vertical CoM of three types of falls as simulated by OpenSim. A standard model represents a typical male body of 1.78m height and 78Kg mass was used for all three simulations . . . 94
6.5 Polynomial fit of three fall types . . . 96
6.6 Three examples of body orientation as detected by the algorithm. Notice the green and red lines showing the direction of the angle in 3D. . . . 98
6.7 Fitting error (ϵ) of falls and non-falls for GM dataset: Notice how well separated falls and non falls appear . . . 99
6.8 Blue dot indicates location of the top bounding box point and red dot to indicate the head location . . . 103
6.9 The pipeline of the myoskeletal simulation fall detection system. Red box encloses the data preprocessing, blue box the model simulation and the green box the fall detection. ONI: depth data format, C3D: standard mocap data format, TRC: OpenSim motion format . 104
6.10 Selection of larger gradient of an ADL (lying down) . . . 105
6.11 Selection of the larger gradient of a fall event . . . 105
6.12 Plot of a Gaussian pdf fitted to the distribution of Hausdorff distances: blue, red and green curves denote the pdfs of HDs between falls, between non-falls and between falls and non-falls respectively . 106
6.13 Simulated ADL velocity profiles of sitting down actions . . . 107
6.14 Velocity profiles of four simulated models of a sideways fall . . . 108
6.15 Velocity profiles of four Opensim simulated falls. We can see the visual similarity of the profiles. . . . 109
6.16 Two C_8 profiles as selected by the Hausdorff distance. Blue graph: ADL simulated profile (sitting down), red graph: simulated fall . . 113
6.17 Typical examples of C_8 profiles of falls (top row) and ADLs lower row . . . 114

7.1 Three occlusion modes. The black rectangle in the image is a synthetic occlusion applied to the depth image. The degree of occlusion is expressed proportionally to the person's height: (a) 40%, (b) 50%, (c) 70% occlusion measured from the ground 121
7.2 Segmented images of furniture 121
7.3 Complex occlusions from a coffee table a chair and a sofa 122
7.4 Truncate Y location of top bounding box point after a cut-off based on 50% occlusion showing extrapolated profile 123
7.5 Location profile estimation in an occluded rigid fall event 124
7.6 Results across datasets **UR** and **GM** using simple occlusions. Red line denotes the sensitivity and green line shows the specificity at various occlusion degrees. Note that both fall and ADL events are detected by the myoskeletal approach 6.7 when occlusion degree is at least 50% 125
7.7 Results from SDU dataset using truncation of head location. Red line denotes the sensitivity and green line shows the specificity at various occlusion degrees 125

A.1 Fall detected in a front (angled) view 157
A.2 Lying on the floor 158
A.3 Sitting vigorously on a chair 158
A.4 Picking up an item from the floor in fast motion 159
A.5 Sweeping activity 159
A.6 Vigorously sitting on a sofa 160
A.7 Sideways fall towards the sensor 160
A.8 Picking up and dropping a box 161
A.9 Picking up and dropping a chair 161

B.1 Feet force schematic 164
B.2 Two piece rigid fall model 165

List of Tables

2.1 Pros and Cons of current fall detection approaches 20
2.2 Fall detection approaches and their performance 21

3.1 RGB-D Fall datasets. R: RGB data, IR: infrared data, D: depth data, A: accelerometer data, S: Kinect skeleton data. The table shows the different fall event datasets of several sensor technologies. Noticeable is the number of fall events if compared with the ADLs as well as how small the fall number is in general 37
3.2 Performance of fall detection and comparison against previous studies across 2 public datasets 46
3.3 Type and number of events from each dataset 47

4.1 Performance of Algorithm 1. A→B: Training on A, test on B datasets, N/C: Not Converges, † Converges after re-sampling of training set 67
4.2 Performance of Algorithm 2. A→B: Training on A, test on B datasets, † Converges after re-sampling of training set 69

6.1 Performance of the Hybrid Algorithm 98
6.2 Mean error ($\log \epsilon$) from height customised and Trained based approaches 99
6.3 Performance of fall detection algorithms developed in this thesis and comparison against previous studies across 2 public datasets . . 110
6.4 Simulated model height variability over **UR, SDU, GM** datasets combined. The table presents results of the algorithm for each height models applied to the height-labelled acted datasets (**UR, SDU**) and the in-house depth dataset (**GM**) showing the sensitivity and the specificity for each combination of simulated model height and approximate human height. If height selectivity is applied then detection is 100% for both sensitivity and specificity (main diagonal). Values in bold denote either missed detections (for sensitivity) or false positives (for specificity) 111
6.5 Performance of fall detection and comparison against SisFall dataset 114

7.1 Performance of myoskeletal approach 6.7 against in-house and public datasets 126

Abbreviations

2D	**2 D**imentions
2.5D	**2.5 D**imensions
3D	**3 D**imentions
ADL	**A**ctivities (of) **D**aily **L**ife
Accu	**A**Accuracy
C3D	Biomechanics motion capture data
CCTV	**C**losed-**C**ircuit **T**ele**V**ision
CoM	**C**entre (of) **M**ass
f-o-v	**f**field **o**f **v**iew
FN	**F**alse **N**egatives
FP	**F**alse **P**ositives
GMM	**G**aussian **M**ixture **M**odel
GPU	**G**raphics **P**rocessing **U**nit
HD	**H**ausdorff **D**istance
HMM	**H**idden **M**arkov **M**odel
IR	**I**nfra **R**red
k-NN	**k-** **N**earest **N**eighbors
KFD	**K**ernel **F**isher **D**iscriminant
LOWESS	**LO**(cally **WE**ighted **S**catterplot **S**moothing
MFCC	**M**el-**F**requency **C**epstral **C**oefficients
ML	**M**achine **L**earning
MoCap	**Mo**tion **Cap**ture
MRI	**M**agnetic **R**esonance **I**maging
NN	**N**earest **N**eighbor

ONI	**O**PEN **NI**
PCA	**P**rincipal **C**omponent **A**nalysis
PNG	**P**ortable **N**etwork **G**raphics
PIR	**P**assive **I**nfrared **S**ensor
Prec	**P**recision
RS	**R**andom **S**earch
RGB	**R**ed **G**reen **B**lue
RGB-D	**R**ed **G**reen **B**lue **D**epth
RF	**R**andom **F**orest
RMSE	**R**oot **M**ean **S**quare **E**rror
SBC	**S**parse **B**ased **C**lassifier
SD	**S**tandard **D**eviation
SDK	**S**oftware **D**evelopment **K**it
Sens	**S**ensitivity
Spec	**S**pecificity
SVM	**S**upport **V**ector **M**achine
TN	**T**rue **N**egatives
TP	**T**rue **P**ositives
TRC	**T**rack **R**ow **C**olumn
WHO	**W**orld **H**ealth **O**rganisation

Symbols

X, Y, Z	world coordinates
vH	height velocity of the bounding box
vZ	height velocity of the bounding box
vD	height velocity of the bounding box
vWD	height velocity of the bounding box
L	Length
g	gravitational acceleration
θ	angle
V_y	vertical velocity of the head
$Vcom_y$	vertical velocity of the CoM
Ek	kinetic energy
Ep	potential energy
$\alpha, \beta, \phi, \rho$	angular positions
m	mass
$\vec{OG}$	position vector of the centre of the mass
y_G	coordinate of the centre of mass
c	constant of integration, c = 0 in practice
ω	angular frequency
$Y_m(t)$	top boudning box vertical location profile
$Y(t)$	FILTERED top boudning box vertical location profile
$V^+(t)$	simulated fall profile
$V^-(t)$	simulated nonfall profile

Publications

Journals:

- **G. Mastorakis** and D. Makris. "Fall detection system using Kinect's infrared sensor". Journal of Real-Time Image Processing, 9(4):635–646, 2014, DOI: https://doi.org/10.1007/s11554-012-0246-9
- **G. Mastorakis**, T. Ellis, D. Makris. "Fall Detection without People: A Simulation Approach Tackling Video Data Scarcity". Expert Systems With Applications, June 2018, DOI: https://doi.org/10.1016/j.eswa.2018.06.019

Conferences:

- **G. Mastorakis**, X. Hildenbrand, K. Grand, D. Makris. "Customisable Fall Detection A hybrid approach using physics-based simulation and machine learning". ECCV 2016 - 1st Workshop on Action and Anticipation for Visual Learning Amsterdam, Oct 2016

Technical Presentations

- **G. Mastorakis** "Fall detection system using Kinect sensor". Microsoft Faculty Connection, Kinect for Life, Oct 2012
- **G. Mastorakis** and D. Makris. "Fall detection: types, solutions, weaknesses. A review of computer vision-based fall detection systems". IET Event, Human Motion Analysis for Healthcare Applications, Savoy Place, May 2016
- **G. Mastorakis** and D. Makris. "Studying accidental falls". DIRC / CEESR Student Conference, Kingston University, June 2016
- **G. Mastorakis**, T. Ellis, D. Makris. "Fall detection using myoskeletal model simulation". BMVA Symposium, Human Activity Recognition and Monitoring, Oct 2017
- **G. Mastorakis**, D. Makris, T. Ellis. "Detecting falls using accelerometer data based on myoskeletal fall modelling". BMVA symposium, Computer Vision for smart environments and assisted living, June 2018

Chapter 1

Introduction

Ever since our humanoid ancestors evolved to bipedal walking, nearly 2 million years ago, we have been vulnerable to falling over. This brings a significant risk of injury, according to the severity of the fall and the well-being of the individual. A fall is an incident which results in a person coming to rest inadvertently on the ground or floor or other lower level as defined by the World Health Organisation (WHO) [14]. A fall occurs due to internal or external factors when a person loses conciousness or accidentally slips or stumbles while walking or standing. Internal causes include underlying medical conditions, such as neurological, cardiac or other disabling conditions; side effects of medication, physical inactivity and loss of balance, particularly among older people; poor mobility, cognition, and vision, particularly among those living in an institution, such as a nursing home or chronic care facility. External factors are also responsible for inducing falls such as overcrowded housing, poorly maintained footpaths, banana skins etc. Other unsafe environments may particularly affect those with poor balance and limited vision and also those working at elevated heights or other hazardous working conditions. More types of falls are observed in the working environment, due to walking surface condition, low visibility and lack of concentration, tiredness, etc. There are some clearly identifiable groups of people who are more likely to fall such as athletes. Some people may also fall after a violent attack.

A person who is young and healthy can experience a fall without it resulting in serious injury and they can heal relatively fast. Those who are more vulnerable, such as the elderly or disabled and patients in rehabilitation, may be more susceptible

to serious injuries and take longer to heal. In some cases immediate assistance is required yet the faller may not have the ability to call for help.

The elderly though are the most vulnerable to fall and due to their living circumstances of isolation are more prone to have a fatal progression after falling. Age is a significant factor that is closely linked to severe falls [15]. Several studies have shown [16] that elderly people experience at least one fall every year. Also, falls are the main cause of accidental death in adults aged 65 or more, based on a review of 90 epidemiological studies [17]. Other resources show the injuries caused by falls in the general population [18].

Other studies characterise the severity, frequency, risk factors [19] and cost [20] of fall incidents which are attributed to be the leading cause of fatal [21] and non-fatal injuries among adults over the age of 65 [22]. Other studies discuss the acceptance of applied fall detection systems for the elderly [23]. [24] discuss the different types of accidental walking falls (slip, trip, and step) and their potential causes. [25] discuss the various health conditions that may cause falls in relation to falls in the elderly population. A study discussing the potential of video-based fall detection is given in [26] where participants with a fall history approve the life-saving benefits of a monitoring system.

Currently deployed technologies for alerting a fall incident are manual and self-activated, based on push-button devices which the person wears as a pendant or bracelet. This device is given to the elderly living alone and also those who are vulnerable to and have a history of falls. It is required to be worn constantly during day and night. In the event of a fall, the fallen person has to push the button to generate a signal that is then transmitted to a crisis telecare centre. However, this approach relies on the individual being conscious, in full possession of their mental faculties and cognisant of the required response, otherwise it is simply an unattractive ornamentation.

Similar modalities exist in assisted living houses, where push-button alarms are supplied to the occupants. Constant surveillance is available in such places and in the case of a fall incident, the response time is expected to be less than for those independently living in their own homes.

Automatic human fall detection systems are required to take over these manual technologies for monitoring vulnerable people who are prone to falling. A plethora of new methodologies for fall detection have been developed in the past few years,

with a strong motivation to enable the elderly and the infirm to live independently in their own homes, whilst providing an unobstructed and non-invasive means of monitoring their well-being. A large-scale monitoring strategy is required to maintain those in need in their homes. Ideally, a home monitoring system would provide a comprehensive detection capability, whilst preserving the privacy of the individuals being monitored.

The variety of fall types is ignored by many of these fall detection systems, which produce a solution without assessing the human individual or the scene characteristics. People fall for a variety of reasons related with their physical or mental health (e.g. ageing/disability) or due to abnormalities of the walking surface (e.g. slippery/uneven floor). An effective fall detection system should accurately and robustly detect a fall when it occurs, without false detections (e.g. lying on the floor for the purpose of an exercise) for application in the general population. The investigation in this study will first focus on data-driven trained algorithms and then proceed to physics-based myoskeletal algorithms performing fall modelling.

Several sensor technologies were used for detecting falls such as cameras, infra-red sensors, acoustic and pressure sensors accelerometers and others. A further discussion of these is given in Chapter 2. The development of these sensors contributed to the development of fall detection algorithms, particularly with the falling cost of cameras and the development of new image acquisition technologies such as the inexpensive depth cameras (Kinect, Xtion, etc.). Also, the maturity of the computer vision domain has simplified the development and deployment of computer vision applications. The use of depth data was beneficial for this work since the level of certainty and accuracy has increased due to the introduction of an extra dimension (in reality it is 2.5D) and also the high level of data provided by OpenNI and Kinect SDK which simplifies person segmentation. With the arrival of Kinect in 2011, new fall detection research utilised the functionalities of depth sensors such as 3D analysis and privacy protection. As a result, using Kinect (depth) data provided a more acceptable solution of a fall detection system.

Another major issue of fall detection systems is the use of human fall data for training which introduces many complications and results in questionable algorithmic performance. This is due to the difficulty in performing such falls in a realistic manner which imitates the actual behaviour, particularly for samples of elderly falling – which is one of the target groups of this study. Current fall data are acted by young adults, acting pre-defined patterns of how a fall should look, causing the

minimum risk on impact. Also, the size and variability of these samples is limited as it is performed by a small group of people for a specific geographical region.

Current fall detection algorithms employ a generic one-size-fits-all solution and overlook the individual characteristics such as the height, weight or posture. Every fall event appears to be different according to the height, weight distribution, centre of mass (CoM) location, the orientation of the person or other parameters. In this study, the investigation will focus on the fall events (i.e. rigid or collapsing) which are introduced by internal physical issues caused by lack of consciousness and the faller does not recover from the fall i.e. remains on the ground. This work investigates several fall detection algorithms which try to overcome the issues of accurate and robust detection of falls using depth sensors while the privacy of the fallers is preserved. It also evaluates alternative approaches to modelling falls taking into account the human variability of falling people, as well as indoor scene occlusions. An evaluation of accelerometer-based fall detection is also briefly studied in order to provide a general applicability of a proposed approach on non-vision based studies.

New methods are proposed: a) a machine learning approach where the fall decision is learnt by training on fall and non-fall data, b) a machine learning method where the decision is learnt by using physics-based myoskeletal data and human non-fall data, which is customised to the person's height only for falls and c) a myoskeletal fall modelling approach which relies on a single observation without a learning procedure that is fully customised on the person's height. The simulation based techniques were developed to eliminate the use of unrepresentative human-fall-data for training purposes. Furthermore, occlusion protocols are also proposed to evaluate the later approach against simple and complex synthetic occlusions.

1.1 Challenges

The accuracy and robustness of a fall detector is crucial when a human life relies on it. Several actions, such as lying down may confuse the detection algorithm and as a result, a non-fall may be detected as a fall and cause an unnecessary alert to be raised. These are described as activities of daily living (ADL) or non-falls. Therefore, algorithms should minimise the effect these events have, to increase the reliability of the system, reducing false positives (FPs). Different types of falls

exist which depend on the direction the body has when falling and each of them has different motion. Hence, a detection algorithm is required to detect all types and minimise missed-detections. Monitoring in general and particularly at home can be invasive and consideration of the issues of personal privacy may negatively influence the choice of a vision-based fall detection system. The monitoring system must have some privacy features such as to hide recognisable characteristics as the face.

Existing research into fall detection systems usually relies on (or are adjusted/-trained) using limited quantities of human fall data (that is non-representative of the real fall events). Capturing real-life fall events is a rigorous process which involves a costly infrastructure in order to capture the event – whenever it happens. A few research groups have succeeded in recording falls in hospitals and care homes, but such data is limited and not publicly available due to privacy and copyright constraints. The alternative approach uses human-simulated fall video recordings where participants attempt to act the fall following specific guidance from a researcher. The approach of data collection is quite common in data science, nevertheless, collecting fall detection videos raises the following issues in terms of how representative data are:

1. **Demographic**. Data samples from falls and ADL should include people of different ages. Similarly, people with different physical characteristics (e.g. height, weight, posture) should be participating in these datasets. Finally, further samples should be included of people with behavioural characteristics such as gait patterns.

2. **Sample quality, quantity and availability** These recordings should show real fall events or human-simulated ones which are representative. The size of these datasets should be enough in permutations, fall types, visual scenes (e.g. home), etc. Finally, the data (either real fall or human-simulated) should be publicly available and easily adaptable/readable.

3. **Scene conditions**. Visual occlusion is another issue to be taken into account. Coffee tables, chairs, or sofas and other furniture can act as obstructions in a home scene.

These issues on the quality and quantity of fall data were only recognised later in the research, and the initial work exploited conventional datasets focusing mainly

on the accuracy, robustness and privacy of approaches. The more recent work for this thesis focuses on the use of simulation technologies to address the issues.

1.2 Aims & Objectives

The aim of this project is to investigate robust methods for detecting falls, and particularly their applicability to support independent living for elderly and infirm people. Specific technical objectives that will guide the research are the following:

1. Investigate features for fall detection and develop reliable and robust detection algorithm(s) for rigid and collapsing falls.
2. Investigate the use of fall simulation to obviate the need for acted fall datasets and the effect of physical body characteristics and fall direction on the fall behaviour
3. Investigate the impact of occlusion in the detection of fall and ADL events
4. Investigate the use of simulation for other modalities such as accelerometer-based fall detectors

1.3 Contributions and Thesis Overview

The initial research developed a machine learning approach to detecting rigid fall events using video depth data [27]. Subsequently other fall types such as collapsing were investigated as part of a generalised fall detector and as a result alternative features were examined to deliver a robust multi-fall-type fall detector. Public datasets have become available and a thorough examination of their videos [28] raised several issues in terms of how representative data is. It was also found that such datasets were relatively small in the number of subjects and fall permutations, with a minimum variation of falling behaviour and lack of realism. This introduced the search for other means in order to replace the human factor from the training of fall detection algorithms. The use of physical models starting with a falling rod and extending to a full myoskeletal model are introduced and investigated [29]. The use of velocity profiles rather than a single value (e.g. peak velocity) is utilised

as the simulation modelling can provide a full velocity profile. The detection of a fall is expressed as a comparison between velocity profiles of simulated falls and non-falls using the Hausdorff distance. The simulation using a myoskeletal model is shown to be a closer representation of a human performing falls without worrying about injuries. Also, it provides continuous data when an event occurs, that makes the simulation approach occlusion robust. Finally, a validation protocol against occlusions is proposed in order to assess the performance of simulation based approach under occlusions [30].

The following list itemises the contributions:

- **Data driven Approaches**

  Two real-time machine learning based fall detection algorithms: 1. A rigid fall detection algorithm based on the analysis of fall and ADL depth data from Microsoft Kinect I with the use of X,Y,Z velocities of the 3D bounding box; 2. A rigid and collapsing fall detection algorithm based on the analysis of fall and ADL depth data with the use of 3D angular velocity derived from a modified bounding box.

- **A review, critique and <u>evaluation</u> of fall data**

  A study examining the issues of current human fall datatets with an evaluation of acted data against real falls. This evaluation determined the poor acting observed within the fall samples, particularly for collapsing falls.

- **Simulation based approaches**

  Demonstrate the capability of a physics-based myoskeletal model to simulate a fall, and to use this simulation to replace the need for recorded human fall data to detect falls. Thus, three personalised myoskeletal simulation based fall detection algorithms: 1. A hybrid fall detection algorithm using myoskeletal simulation and human ADL data which is customised by the person's height and by using the estimated falling direction. This approach is applied on depth data; 2. An occlusion robust fall detection algorithm based explicitly on myoskeletal simulation which is only customised by the person's height and utilises the comparison of velocity profile using the Hausdorff distance. The evaluation used depth and YouTube real fall data. 3. An approach similar to 2, applied on accelerometer data.

- **A framework to evaluate the impact of occlusion**

  A methodology of evaluating fall detection algorithms under occlusions using synthetic ones which are inserted on videos of existing fall datasets. Such synthetic occlusions tested the performance under variable degree (i.e. rigid shaped) of occlusion as well as the performance when complex occlusions were used such as chairs, coffee tables etc.

The chapters structure outline of the remainder of this thesis is presented:

Chapter 2 — Literature review

A literature survey of fields related to this research including existing fall detection systems using a variety of technologies.

Chapter 3 — Datasets

A study examining the issues of current human fall datatets with an evaluation of acted data against real falls.

Chapter 4 — Learning to detect falls

Methodologies based on machine learning to detect rigid and collapsing falls captured by a depth camera.

Chapter 5 — Simulation: Modelling falls

Demonstrates the capability of a physics-based myoskeletal model to simulate a fall.

Chapter 6 — Fall detection using myoskeletal simulation

Considers three personalised myoskeletal simulation based fall detection algorithms.

Chapter 7 — Occlusion robust fall detection

A framework of evaluating fall detection algorithms under occlusions using synthetic ones.

Chapter 8 — Conclusion and future work

A summary of contribution and future work.

Chapter 2

Literature Review

This chapter will serve as a general review of fall detection systems as each following chapter includes a chapter specific review of the covered topic. Research into fall detection systems have studied a wide range of sensor modalities, including accelerometers, acoustic, ultrasonic, infra-red, radar, RGB cameras, depth sensors etc. The bulk of the methods rely on those principal approaches; detecting the velocity of the falling body or the changing shape of a person's projected silhouette e.g. from an upright to prone position, on the floor.

The chapter covers several features, hardware technologies and algorithmic techniques reported for fall detection. The discussion of features specifies the most dominant approaches developed in computer vision. Another discussion specifies the technologies used and divides systems into two groups, one for wearable solutions and another based on static sensors. Since this work was initially based on 3D vision and then on both 3D vision and then extented to include accelerometer measurements, this topic is included in the discussion. Since the development of Kinect in 2011, the topic of fall detection flourished under the use of cheap depth sensors, mobile phone gyroscopes and accelerometers and also the developed small accelerometer devices, which were then embedded on a range of wearable devices for monitoring heart rate, activity etc. Finally, a discussion will focus on the algorithmic approaches which classify the events as falls or ADLs.

2.1 Fall detection features

Several features were employed for fall detection and other studies have performed feature selection in order to select the best ones [31]. Such features include bounding boxes, shape descriptors, velocity, height of the person from the floor, fall angle and more as discussed in recent review studies [32]. The use of bounding boxes, either 2D [12] where changes in the motion of the bounding box are analysed and 3D [27], where the velocities of the height, width and depth of a 3D bounding box are calculated. The aspect ratio is discussed in [33] and [34]). The aspect ratio is computed as the ratio of the width and height of the bounding box around of the extracted person. A small aspect ratio implies that person's posture is upright, whereas a high aspect ratio means a lying down posture.

Ellipse detection appears in several studies: in [35] the 3D position of the center of the ellipse was employed as a feature defined as the distance between the center of the ellipse and the plane floor in 3D space. In [36] an ellipse is fitted over the person in order to calculate the fall angle which is found between one of the major axis (e.g. long axis) of the ellipse and the floor. A similar approach using fall angle is also described in [33] where a small angle is used to decide that a person has fallen.

The studies in [37–39] use the head location to measure the distance from the floor, the velocity of the head, the distance to the ground and the 3D velocity. The velocity of the head is found to exceed certain thresholds for the event to be classified as a fall. A simplier approach [37] uses the head's height from the floor which classifies an event as a fall if the head is located below the threshold.

The shape of a person is analysed in several studies on 2D or 3D data and an event is assessed using data samples during the event or by using data from the initial/final state. In [36] the method determines the direction and position of the individual based on the shape of the human silhouette. The centroid of the silhouette and the angle between the human body and the floor plane are also calculated for fall incident detection. In [5] curvature scale space features are extracted from the depth maps of the human silhouette. The features appear as approximations of the silhouette edge and are recorded during the event in order to capture the shape change over time. In [40], the authors use the silhouette to fit a bounding box, where the aspect ratio is calculated as well as a covariance matrix, which provide adequate features for fall detection.

Skeleton data, either derived from 2D or 3D data of conventional cameras or Kinect sensors was used for this subject. In [41] Kinect data were processed using the Microsoft Kinect SDK where the skeleton joints are extracted and tracked. The joint velocities are measured together with the distances of these joints to the floor. The fall is detected when velocities and distances are below defined thresholds. In [42] a 2D skeleton is extracted by running the well-known graph traversal Depth-first search algorithm on the human contour which is partitioned into triangular meshes. In [43] the 3D skeleton information is also used. The orientation of the major axis is calculated using the coordinates of the head, shoulder, spine, hip and knee joints. Then the angle between this line and the horizontal line is calculated which determines the inclination of the body after the fall.

The bounding box alone does not provide enough information regarding the human motion and the performance of this technique relies on the camera view angles, particularly for the 2D methods. The aspect ratio can be inaccurate due to the position of the person, camera, and occluding objects, if present. The silhouette based features have the same issues as they can be occluded and rely on the viewing angles. The head location appears a more stable feature as it is occlusion robust due to its location and does not rely on the rest of the body to be detected. But, head detection can be problematic in cases where there is rotation or tilt of the head or camera viewing angle. Skeleton data derived by Kinect are unstable especially when the person is falling and it is noted that the skeleton shape does not recover to its original shape after the fall. In Chapter 4 several features are discussed and a re-evaluation of a feature-selection study for fall detection analyses the significance of fall data (i.e. fall types) for selecting and developing a feature.

2.2 Wearable sensors

The most common approach to generating an alert in event of a fall is a push-alarm such as [44]. Those devices are carried by a person prone to falls and are activated by pushing the alarm after the fall. This technology can be very weak as the person may not be carrying the device or may be unable to push the button if they are unconscious. Nevertheless, this technology is widely used by many Councils in the UK as they are easily installed, maintained and are cost-effective.

In general, wearable devices are automatic in terms of data processing. They capture motion continuously using motion detectors with accelerometers and gyroscopes [45, 46]. Such sensors are capable of detecting the rapid motion changes of the person who wears them. A study in [47] discusses the use of a wearable sensor to detect unseen falls. In the same context, other studies use a mobile phone's [48] accelerometer and magnetic field sensor data, accelerometer and data from the wearable camera [49], and energy sensor such as a triboelectric generator [50].

Although this seems promising, it is questionable how effective the wearable approaches are when it comes to such a life-threatening event, as the person has to wear the device continuously. If the person who is supposed to use the device forgets (e.g. elderly due to memory issues) or ignores the importance of wearing it, a fall is not detected rendering the approach useless. Other issues of portable device is the requirement to recharge or replace the batteries every now and then in order to continuously operate or even remove them when in shower due to lack of waterproof capabilities. Mobile phones already contain the technology to detect the fall (i.e. an accelerometer, gyroscope exists in most of the phones) while having the capability to call for help whether the user is indoors or outdoors. Nevertheless, the location of the phone is crucial as the user may hold it on a jacket's pocket, a pocket near the waist or inside a purse. Benefits of wearables are that the sensor is personalised and moves with the person, therefore there is no need to have a sensor in every room. Also, these devices are entirely private and not affected by occlusions, as are camera-based ones, but they can be affected by wireless communications.

2.3 Fixed location sensors

Acoustic and ambient sensors systems use microphones or vibration sensors. Such systems detect the loudness and height of the sound to recognise a fall [51]. Other approaches detect the floor vibration [52] or use features extracted from the radar signal [53]. A microphone array system is presented in [54] where it is found that the fall signal has highest frequency component around 1000 Hz. Using the height of the sound source, sound classification techniques such as mel-frequency cepstral coefficients (MFCCs) and a nearest neighbour (NN) algorithm are used to classify falls from non-falls. A ceiling of infrared sensors is proposed in [55], where each

sensor produces a binary response given the existence of a person underneath. The set of signals produce an 5 by 4 pixel image where an assessment is performed on how different a stream of pixel values are from the previous frame in order to determine a fall event.

Smart tiles containing force sensors and 3-axis accelerometers is presented in [56]. The force sensors allow the detection of falls as well as recognition of other human activities such as walking, standing, sitting, lying down, and the transitions between them. However, the detection accuracy on human fall data returns false positives caused by lying postures. This issue is resolved by a fusion between the force sensor measurements and the accelerometer sensor decisions. Another floor based sensor is proposed in [57] where pressure-sensitive fibre sensors are embedded underfloor with an application focus on fall detection in the bathroom. The fibre sensor is low cost, unobtrusive, and waterproof, making it especially useful in a bathroom. The assumption of the system relies on the fact that when a fall occurs, the target must be lying on the floor, as people do not normally lay on the bathroom floor to exercise or sleep.

Static mounted sensors may have the advantage of monitoring without the requirement to wear a device constantly while surveillance can be unobstructed and continuous without the person's concern. The assumption for many studies is that the monitoring area has to be clean from objects and occlusions. Such systems are limited to indoor use only due to their restrictive application range. Also, the cost and complications of deployment and maintenance are discouraging factors for using such approaches.

2.3.1 RGB systems

Some systems use image analysis to detect falls. They require one [58, 59] or several cameras [2, 60]. They do not require a device attached to the person as they are able to detect the human motion, using computer vision algorithms. Thermal cameras are also used to locate and track a thermal target and analyse its motion in order to detect a fall's characteristic dynamics and then to monitor a target's inactivity [61]. One approach to fall detection is to analyse the velocity of the falling person as proposed by biomechanics [62]. In [39], head velocity is used to detect a fall using 3D tracking. Their approach is not robust as they detect only

two out of three falls but it can differentiate between the actual falls and the fall-like events i.e. sitting. Other vision approaches focus on posture-based events as in [63]. In that study the authors focus on three types of falls (forward, backward, sideways). While their approach is robust as they can differentiate between falling and lying/sitting, it is limited as the raw data used for their analysis is captured only from a side-view.

2.4 Combinatory Systems

Some studies suggest a combination of hardware–vision solutions for fall detection. In [64] Ambient Assisted Living platforms are discussed as wearable, ambient, vision and multimodal. Also, in [65] several fall detection systems are compared mainly for wearable sensors. The work in [4] uses data from accelerometer and depth video from 2 sensors. Acoustic sensors [66] or PIR sensors together with thermopiles [67] and depth with accelerometer and acoustic sensors [68]. Finally, in [69] a combination of camera and heart monitor system is proposed. Such systems provide a more reliable result based on the given experiments but their complexity is higher and some may still be invasive. One way or another, vision solutions may still be relevant in the designing of such combinatory solutions.

Thresholding techniques where signals from floor pressure data and infra-red images are processed and a fall is detected when a set threshold is met [70]; this approach reports 90% accuracy on a dataset of 120 samples.

Bayesian filtering is used to determine the pose of the person as the probability of falling or getting up using data from a near-field imaging floor sensor [71]; the authors propose a floor sensor based on near-field imaging. The shape, size, and magnitude of the patterns are collected for classification from a set of features that are computed from the cluster of observations. The postural estimation is implemented using Bayesian filtering instead of the features being classified directly. The system has problems with test subjects falling onto their knees as this produces a pattern very similar to a standing person. 650 events and ten people yielded a sensitivity and specificity of 91%.

The discussed approaches in the previous two sections (2.3, 2.3) provide fall detection solutions where authors have used human fall data for tuning their approaches. In general, the performance of these approaches is linked with the data which were

used for training, a complication discussed in Chapter 3. Another issue is the complexity of the combinatory systems to analyse signals from different modalities. The camera systems are mainly monocular and prone to occlusions from furniture or other objects within the scene. The event of an occluded fall is not used in these studies and will be further discussed in Chapter 3.

2.5 Algorithms

A number review papers summarise and discuss computer vision based fall detection systems such as: [6][72][73]. Zhang et al. [6] discuss the recent methodologies and categorise them in terms of acquisition (single RGB, multi RGB cameras and depth sensors), where [65, 74] discuss the different accelerometer and other wearable approaches. Another distinguishing factor is whether these algorithms are ad-hoc methods based on empirical observations or pattern recognition methods that are trained using machine learning (ML). The majority of the algorithms discussed in the following review are based on ML approaches as researchers use classification algorithms to justify whether an event is a fall or ADL. In either case, debating on whether an ad-hoc method is less reliable than a ML is out of the scope of this study, since the complication as discussed in Chapter 3 is more related with the quality of fall datasets. Therefore, a critique to discuss the complications of data-driven approaches will occur on Chapter 3, where quality of fall data is proposed as one of the issues of ML performance. The following provides a thorough discussion of the use of RGB, depth and accelerometer based detectors.

2.5.1 Use of RGB data

A wide range of techniques for fall detection are found in the literature which use cameras and other sensors. In [75] a Gaussian Mixture Model method is used to classify the different activities as a fall or not, based on shape deformation during the fall followed by a lack of significant movement after the fall. Segmentation is performed to extract the silhouette and additional edge points inside the silhouette are extracted using a Canny edge detector for matching two consecutive human shapes using the shape context. The mean matching cost and Procrustes analysis are applied for shape analysis. Both of these methods contribute in quantifying the abnormal shape deformation. A fall is characterised by a peak on the smoothed

full Procrustes distance curve or mean matching cost curve followed by a lack of significant movement of the person just after the fall. A Gaussian Mixture Model (GMM) classifier is implemented to detect falls. Further computation of the sensitivity, specificity, accuracy and the error rate obtained from the GMM classifier is performed for the analysis. An ensemble classifier is used to combine the results of all cameras. The mean matching cost and the Procrustes analysis reduce the error rate to 4.6% and 3.8%, respectively.

Rule-based techniques determined by a set of features from the subject and its bounding box such as aspect ratio, horizontal and vertical gradient distribution of object in XY plane and fall angle are used to assess the fall event [13]. An adaptive approach for the detection of moving objects by using background subtraction as well as bounding boxes is used. The described fall model is based on feature extraction analysis, detection and classification. Features extracted include horizontal and vertical gradients, aspect ratio and the centroid angle to the horizontal axis of the bounding box. Falls are confirmed when the angle reaches a value less than 45 degrees. The algorithm reports 100% accuracy, specificity and sensitivity, evaluated on their dataset (40 videos).

A multi-frame Gaussian Classifier is used to determine the direction of the body and the head location over a predefined frame window [76]. The method is aimed at incidents involving falls in unobserved home situations by presenting the design and real time implementation of a fall detection system. The design involves segmentation of foreground objects in the image streams obtained from two fixed, uncalibrated, perpendicular cameras. The direction of the main axis of the body and the ratio of the variances in x and y directions are calculated through principal component analysis (PCA). A head tracking module is used for human detection as well as increasing the robustness of the system. Head position is estimated as a blob using the Gaussian skin-colour model and is tracked by searching for skin-coloured blobs nearby the head position. The classification is performed through a Gaussian multi-frame classier. The system shows accuracy of 100% on unoccluded video sequences but the addition of occlusion on 4 video samples reduces the accuracy to 44%.

A nearest-neighbour rule, where postures using the ratio and difference of human body silhouette bounding box height and width are used together with the time difference between events to classify a fall [77]. The authors proposed a fall detection system in which a statistical scheme and vertical projection histograms of

the silhouette image are used to reduce the effect of upper limb activities of human body. This approach used k-NN classification to classify the postures using the difference and height-width ratio of human body silhouettes bounding box. The k-NN classifier and the critical time difference are used to detect fall incident events. The study reports an accuracy of 84.44% based on 15 subjects.

Hidden Markov Models where falls can be detected by analysing the person's posture and detecting sudden changes in posture (e.g. from standing to lying) are described in [60]. The authors applied a multi-camera system for image stream processing. The processing includes recognition of hazardous events and behaviours, such as falls, through tracking and detection. The cameras are partially overlapped and exchange visual data during the camera handover through a novel idea of warping people's silhouettes. The video server (multi-client, multi-threaded transcoding) transmits sequences for further processing to confirm the validity of received data. The bandwidth usage is optimised through event-based transcoding and semantic methods.

Fuzzy Logic is used to determine the state (e.g. upright, lying) of the person at each frame using voxels derived by silhouettes of people captured by infrared cameras [78]. This study also used a multi-camera system where the authors applied silhouettes to form a 3D model of the human object. The membership degree of the object is measured using fuzzy logic to a pre-determined number of states at each image. The fall detection method consists of two levels. The first level deduces the number of states for the object at each image. The second level deals with linguistic summaries of the object's states called Voxel Person. Further derivations are performed regarding the activity. The study reports a specificity 93.75% and a sensitivity of 100%.

2.5.2 Usage of depth data

Attempts [27] in detecting falls by processing depth data arose from the research in this thesis.

In [41] the authors use the skeleton tracking capabilities of their own algorithm. Nevertheless, the skeleton works accurately when the sensor is placed at a specific range and location. The approach aims to detect falls by extracting skeleton data from Kinect depth images based on the fast randomized decision forest algorithm.

This algorithm produces more accurate detection by properly rotating frames to match human orientation. This approach achieved 100% accuracy on a small dataset of 20 sample falls.

In more recent studies such as in [79, 80], the authors used Riemannian manifolds of fall velocity statistics and a combination of RGB and skeleton data respectively. Both studies have evaluated their approaches based on publicly available datasets and achieved nearly perfect performance in terms of accuracy and false positive rate. Nevertheless, the evaluation process in all the above studies does not consider an individual's physical characteristics or falls subject to occlusions.

The initial critique of the depth based fall detectors (pre 2011) can be found in Chapter 4 as the proposed algorithms were compared to these studies.

2.5.3 Use of accelerometer data

In such studies, an accelerometer device is placed on, or near the waist - a location near the CoM, or in other locations. In [81] Igual et al. discusses different datasets of accelerometer data recorded via mobile phones which were placed in the pockets or purses of the participants during fall and ADL scenarios. A review paper [82] discusses the different approaches used. These can be grouped into threshold based (in pre-fall, impact, post-fall, velocity, acceleration magnitude and signal change, angular velocity, critical incline based on pre-fall phase) and machine learning (One-class SVM, KFD, k-NN). It is noted that the preference is given to threshold techniques.

A fall occurrence is determined via the k-nearest neighbour algorithm as discussed in [83]. The authors used a cell phone with a tri-axial accelerometer embedded in it. Data pre-processing is performed using a 1-class support vector machine (SVM) and the wireless channel for Internet connection. Classification is achieved through the k-nearest neighbour (k-NN) algorithm and kernel Fisher discriminant (KFD) analysis. Their algorithm was tested on a variety of scenarios of ordinary daily activities, (i.e. walking, walking down the stairs at normal speed) and different fall types, as well as high-intensity daily activities(i.e. running, jump and gymnastics). They report a specificity of 97.5 % and a sensitivity of 84.4 %.

Chien-Cheng et al. [84] proposed a home-based, real-time fall detection system that not only can distinguish up to 4 different kinds of fall events (forward, backward, rightward and leftward), but is also portable, low-cost and with high accuracy rate. The system includes a real-time fall detection band, a home server, and GSM instant messaging function which can transfer fall alert, send emergency help messages. Four male subjects performed 120 fall events and the accuracy rate of the algorithm was 95.83%.

The significance of accelerometer-based fall detectors is still high as recent studies have evaluated the behaviour of elderly performing ADL events [85]. In this study, the authors use accelerometer data on a novel non-linear classification feature that allows one to obtain high accuracy values with a simple threshold. Their work reports 99.4% accuracy.

2.6 Discussion

A summary of the available approaches using single RGB, multiple RGB cameras, depth sensors, accelerometers, ambient sensors and the fusion of some of these sensors is shown in Table 2.1 itemising their pros and cons. A representative reference is included for each approach. The Table includes off-the-shelf technologies such as cameras and infra-red sensors, accelerometers, pressure and sound sensors. Other researchers have used a combination of technologies to increase performance [86].

Fall detection approaches as discussed utilise a variety of sensors such as wearables (accelerometers, gyroscopes), fixed location sensors (i.e. cameras, radars, acoustic, pressure) to detect or protect (i.e. airbags) the person from the fall impact. Fixed location sensors as discussed, can miss the detection of a fall due to occlusion, interference, memory limitations or avoidance of the user to trust the use a monitoring device (e.g. switch the device off). Other reasons include the removal of such devices before sleep or a shower/bath and as a result, a continuous surveillance of someone can be interrupted, especially on occasions where a fall is more likely to happen, such as getting up from the bed or coming out from the shower. As discussed previously, a wearable device has a number of complications, but it also has benefits. Briefly, the wearable provides a continuous signal if worn, hence,

Table 2.1: Pros and Cons of current fall detection approaches

Approach	Pros	Cons
Monocular camera [87]	Easy to setup cheap	Privacy not preserved Occlusion ineffective
Multi-cam [2]	Occlusion robust 3D scene analysis	Difficult setup, cameras require syncing, privacy not preserved, 3D calibration
Infra-red [27]	Privacy preserved, 3D scene analysis, person segmentation ready	Interference, noisy data
Wearable accelerometer [45] gyrospope [46]	Occlusion robust, privacy preserved,	Intrusive, must be worn
Ambient sensors acoustic [51] floor vibration, [52]	Privacy preserved, occlusion robust	Expensive, can be applied on small areas
Fusion 3D vision & wearable [4] 2D & heart monitor [69] acoustic & depth [66]	Higher accuracy and performance	Complex setup, requires syncing

is occlusion robust, if compared to a camera sensor. Also, privacy is assured as recorded/processed signal from such a device is not an image or audio.

Utilising fixed location sensors can invalidate the above issues but introduce further ones. The use of video cameras introduces privacy issues since cameras are likely to monitor wet areas and bedrooms. Acoustic sensors are prone to error and pressure floor sensors are very expensive to cover the entire floor of a house. Depth cameras from the other hand can provide the related privacy as the depth map at a given resolution obscures facial characteristics and other body details. Such depth sensors are now inexpensive to use. Nevertheless, when used in an array of several sensors projecting on the same area, such sensors are prone to interference. This has been addressed in [88] where a vibrating device attached to the sensor disturbs the projection of the laser signal allowing minimal interference with the laser signal of another sensor placed in the same area. Another benefit of the depth sensors is the data processing in the 2.5D space which makes the detection of particular actions possible without calibration as if using an uncalibrated RGB camera. The use of depth data is selected for this work due to the above reasons.

Table 2.2: Fall detection approaches and their performance

study	year	sensor	features	algorithm	evaluation	performance
[12]	2005	RGB,S	2D bbox	HMM	39F 25ADL	100%
[83]	2006	3axial accel	accel	KFD, k-NN	N/A	sp 97.5% sen 84.4%
[39]	2006	3D	head velocity	Decision tree	9F 10ADL	78.9%
[13]	2007	RGB	gradient distribution, aspect ratio	threshold	40	100%
[51]	2008	PIR	Differential voltage	HMM		80%
[78]	2009	RGB	voxels	fuzzy logic	N/A	sp 93.75% se 100%
[77]	2010	RGB	body silhouette	k-NN	N/A	84.44%
[71]	2010	Electric near-field	electrodes, body dimension size, magnitude	Markov chain	650	91%
[34]	2011	depth	user height body velocity	decision tree	N/A	98.7%
[27]	2011	depth	3D BBOX	RS	48F 136ADL	100%
[43]	2013	depth	3D skeleton	threshold	49F 24ADL	95.8%
[5]	2014	depth	curvature scale space	ELM	200F 800ADL	86.83%
[49]	2016	Smartphone camera	Gradient patterns edge orientations	Decision tree	N/A	93.78%
[80]	2017	depth	skeleton, motion map	rule based, SVM	30F 40ADL	99.37%

Furthermore, one of the benefits of the accelerometry approaches discussed in this thesis is the applicability of simulation fall data on accelerometer based fall detectors as discussed in Section 6.9.

A performance evaluation of the systems is given in Table 2.2, where the discussed systems are listed given their accuracy and the number of events (falls and ADLs) on which they have been evaluated. Given these results, it is noticeable that most of the studies use their own data to evaluate and in the majority of them, the sample is small, while the performance is very high. It will be inconclusive to decide the robustness based on such small datasets. This is due to the lack of such fall data, for reasons discussed in the next chapter. The performance will be discussed further in the next chapter where public datasets have been used to evaluate vision-based depth algorithms.

The use of a the velocity of a particular point or bounding box is more feasible as a feature due to the fact that a fall is an event where higher velocities are expected. This is a derivation of most of the threshold based techniques discussed in this Chapter. Furthermore, the machine learning techniques also define a threshold for velocity assessment. Also, the velocity is less complex to measure when using a head detector/bounding box in RGB or depth data and provides -given the existing studies- a good detection rate. It is also widely used for accelerometry based algorithms.

This Chapter leaves an open question - answered in the next Chapter - regarding the suggested robustness of each existing work. Regardless of methodology, a data-driven approach always relies on how representative data is - the selected feature or features fit the purpose of this particular set of data. It is then unnecessary to propose the best algorithms in terms of their performance, knowing that this performance is mainly achieved based on the training used data. An evaluation of assessing representative data is produced for Chapter 3.

2.7 Conclusion

A thorough discussion of technologies, approaches and algorithms was presented. The performance in some of the systems is near 100%, nevertheless, this is questionable if we consider the size and acting behaviour of the participants in these data samples. A thorough discussion about these shortcoming will be discussed in the next Chapter. The use of depth video is selected as the most feasible type of data for this study as discussed. The use of accelerometer application will also take place in order to show the applicability of a proposed method (6.9). The use of features (i.e. velocity) will be discussed in each and following chapter as the detection features vary according to the methodology.

Existing studies are based on the pre-knowledge of how fall events appear on healthy adults, e.g. acted by a falling person where the body has known starting and concluding states. However, in reality these states can be different within people particularly for the target group (i.e. elderly and infirm). However, all these methods are constrained by the type of available data, which is not representative of real fall events of elderly people. Next chapter will investigate the data availability constraints and propose the possible solutions.

Chapter 3

Datasets

3.1 Introduction

Researchers in computer vision and data sciences in general, require a significant amount of data to develop and validate their algorithms. Several characteristics define a good dataset to cover the range of relevant objects, actions or scenarios. A good dataset has a sufficient number of examples to be representative of the variability of actions, human subjects, scene and light conditions, environmental changes and more. It will also provide annotated ground-truth of these objects, actions etc. An example of how these datasets are acquired is the capture of videos of real-world scenes by CCTV cameras. The events shown in these datasets are in most cases accurate and of representative quality of a real-life event.

Similarly, fall detection algorithms require a representative set of recorded examples of people falling for algorithmic training and testing purposes. Ideally, one scenario for capturing such fall events would be to use cameras or other sensors in hospital wards, care homes, assisted living accommodation, in the homes of elderly, rehabilitation centres, inpatient wards etc. Several of these data recording centres would be located around the globe in order to detect the human physiological characteristics variated by height, weight etc. Continuous recordings - day and night - of data over several months or years would have captured a significant number of falls as well as other activities of daily life (ADLs) with some having a similar motion pattern to a fall, such as lying down. The recordings would be in a format that protects personal privacy and allows public access and redistribution for scientific purposes. Unfortunately, the above scenario is imaginary as such

recordings - even if they exist - are confidential and limited in terms of variability and ethically unsuitable as discussed later in this Chapter.

To overcome the issue of data availability, researchers have implemented their own versions of fall events - acted by volunteers. Such datasets are discussed in this Chapter in an attempt to show the limitations and pitfalls of acquiring and using them. Recorded fall data acted by people are not as representative in terms of falling behaviour when compared with datasets of other types of actions. In other words, an acted walking behaviour is likely to be representative as it involves usual daily activities while acting a fall is an artificial action subject to inaccurate behaviours. Given this reason, there is a scarcity of realistic fall samples due to hesitation [28] and the risk of injury performing a fall event. The use of actors guided by researchers aims to bridge the gap between real and human-simulated falls. Also noticeable is the small number of actors participating in these datasets for the above reason and as a result the human variability is limited. The age and health of actors also play a significant role as the vulnerable population (e.g. elderly) is missing from such samples due to ethical complications which prohibit the data-recording of risky actions performed by an infirm person.

A variety of different sensors such as RGB, RGB-D, accelerometers, gyroscopes and radars have been used to record fall events and a range of common activities of daily life (ADLs) as discussed in Chapter 2. This Chapter will focus on visual datasets and more particularly on the RGB-D datasets captured by depth sensors since such data provide access to 3D motion processing. One benefit of depth data is that compared with RGB imagery, facial characteristics are less visible, providing greater protection of personal privacy. Accelerometer data recorded by sensors will be also used for evaluation since such data also describe a person's 3D motion and provide even greater degree of privacy protection.

Recent developments with the use of deep learning exploit the availability of large datasets for training but for fall detection as discussed, such data are not representative and are limited in their number and variability. A deep learning algorithm will suffer in the same way as existing machine learning algorithms that are data-driven. Therefore, a new approach is required to a non-data-driven approach.

This Chapter reviews the publicly-accessible datasets that have been created to support fall detection. The Chapter discusses the early RGB datasets; next is a discussion about the benefits of depth data over RGB, followed by a section on

depth sensors and the human segmentation and tracking software that is available. Two depth datasets (GM, GM2) have been specifically developed for this study in which different types of fall and non-fall events were captured using depth sensors. The GM dataset was possibly the first recorded fall event data intended for the design and evaluation of [27] which is discussed in Chapter 4. Since [27], a number of publicly available datasets have been created and are discussed in section 3.5. In 3.6, an accelerometer dataset is discussed as an evaluation against such data will be presented in Chapter 6. Section 3.8 provides a discussion on the limitations of existing fall datasets considering their suitability in real environments, subject to occlusions and the representative nature of their demographic. Visuals from each dataset are included in order to provide a recognisable image of the data type. To highlight some of the issues with existing datasets (particularly for hesitation), a comparison is presented where acted falls are compared with actual fainting falls, using videos from YouTube.

3.2 Fall Types

A fall has many variations as initially discussed in the Introduction. Internal and external factors contribute to a fall, with the internal ones caused by the physical or mental state of the individual whilst the external are associated with clothing, footwear [89] and the environment. More specifically the physical factors, particularly for the elderly [90], are related to blood pressure, brain atrophy [91], low vision [92], diabetes [93], medication side-effects [94], muscle weakness [95], vitamin deficiencies [96], injury or the lower limbs, gait irregularities [97] and balance issues. Mental conditions [98, 99] may particularly affect cognition causing confusion, lack of attention, reduced sense of risk etc.

Falls have a direction according to the prior body motion or the centre of mass [100]. These falls are directed towards the front, side or the back of the body where the body stays relatively rigid and falls as a stick, or can have a vertical direction where the knees fold over and hit the ground first and the rest of the body falls to the floor afterwards (collapsing fall). After such incidents, the person may remain unconscious on the ground or crawl to seek help. Trips and slips are considered as fall events which are caused by external factors such as elevated or slippery floor surfaces. These incidents may or may not conclude in an unconscious state, depending on the severity and location of the impact.

Another group of falls is observed during sports events, where athletes unintentionally or intentionally fall to prevent an incident or create one. A fall may also be caused by an aggressive attack by another person or an animal.

This study will focus on several fall types that effect the elderly and infirm. It will particularly consider rigid and collapsing falls which conclude on an unconscious state of rest on the ground.

3.3 Public camera (2D) datasets

Many early studies utilised RGB 2D cameras to record falls. Such datasets are discussed in this section which discusses the early challenges of fall detection using such video data.

3.3.1 Single camera LE2i dataset

The LE2i dataset [1] contains 191 videos, 143 falls and 48 ADL of 9 subjects of unrecorded age, weight or height. The recordings are made in different types of room (home, coffee room, office and lecture room) as seen in Fig. 3.1 and according to the authors, this is done in order to evaluate the robustness of the method to different locations. However, they fail to distinguish the actions related to each room, e.g. there is a mattress in the office setting where subjects lay down to sleep. Only one type of fall is shown: a rigid fall event with visible hesitation as actors pull their hands towards the floor to minimise the impact. The capture uses a single RGB camera and the video sequences contain variable illumination and typical difficulties like occlusions due to furniture or cluttered and textured background. Occlusion is found in only 8 videos and it has minimal impact on the scenario as seen in the first column of images in Fig. 3.1.

3.3.2 Multiple cameras fall dataset

The Multiple camera fall dataset [2] is one of the early attempts to record video data for the study of fall detection. One subject (of unknown age or other physical characteristics info), performed 24 falls and 99 ADLs RGB videos. Such actions

FIGURE 3.1: Visuals from the LE2i dataset [1]: fall events at top row, ADLs on lower row

FIGURE 3.2: Visuals from the Multiple cameras fall dataset [2]: fall events at top row, ADLs on lower row

include walking in different directions, housekeeping and actions with characteristics similar to falls (sitting down/standing up, crouching down, picking up an object from the floor). Falls include different types with a direction to the front or back of the body or when failing to sit down properly, or due to loss of balance. The data collection used 8 cameras, mounted around a room to record activity. Although there are objects that will potentially occlude the subject, details of the size and location are only available from the images and the results assume a non-occluded view of the fall and non-fall events is available from at least one of the cameras. Fig. 3.2 shows several frames from this dataset, showing examples of falls and ADL events.

3.4 Depth Sensor data

Depth data provide access to an extra dimension, hence, activity is also captured when the person is moving towards or away from the sensor. The depth sensors are calibrated and the silhouette of the objects or people is measured in millimetres without any extra software development as these sensors come with their own software.

Minimal camera setup is required in terms of calibration and synchronisation to avoid complications discussed in [2] where several cameras were used. Without a calibrated camera it can be difficult to measure the physical characteristics of a person such as their height or the bounding box in 3D space. Also, a monocular calibrated camera can provide depth estimation when intrinsic camera parameters are known and tracked objects are of known size. Stereo cameras also provide a solution to the above issues, such as the ZED [101]. The main issue using this particular camera is its cost as well as the required hardware (GPU) to run it. Furthermore, RGB (2D) video processing is a difficult task when it comes to track human motion and maintain shape information on a cluttered background. Data processing of depth videos using software such as OpenNI, Kinect SDK, and Orbbec Body Tracking SDK provides such measurements of 3D bounding boxes since the person tracker is taking into account only depth data and not colour information, regardless of camera location.

Normal light conditions are required to use an RGB camera system, meaning that lights are required to be continuously on. The effect of this is to have a system where falls happening at dark areas are missed e.g. a fall occurs at night when the person avoids to turning the lights on. The Kinect depth camera for example relies on infra-red illumination and therefore, the signal contains valid information even in complete darkness. A depth camera can be used continuously indoors without complications and changes of light conditions.

Depth data hides the person's facial and other physical characteristics as the format of the video stream (disparity map) as well as the distance from the sensor contribute to maintain personal privacy.

Such inexpensive depth sensors developed by PrimeSence and sold Microsoft in 2010 will be briefly discussed. Kinect I, II and Orbbec Astra are some of these

sensors which researchers used for data collection. The next section discusses the technical details and deployed software.

3.4.1 Depth sensors, OpenNI and Microsoft Kinect SDK

Several depth sensors have been developed since the arrival of Kinect I released in 2010 by Microsoft. It uses three types of sensors: an RGB camera, an IR-based depth sensor and an acoustic sensor. The maximum range of Kinect's IR sensor is 10 metres though the actual effective range depends on the environment. Practically, depth images are noisy beyond 7 metres and may lead to misinterpretations. The Kinect has been widely used to develop numerous applications [102] for action recognition.

OpenNI [103] is an open source tool from PrimeSense [104] which provides access to the depth information regarding human subject's detection and tracking and articulated pose estimation as well as gesture and motion recognition.

3.5 Public RGB-D datasets

More recently RGB-D datasets have become publicly available for evaluating fall detection algorithms. The following briefly summarises this composition.

3.5.1 TST Fall Detection v2

The TST Fall Detection v2 [3] is an RGB-D dataset recorded using Microsoft Kinect v2 and two accelerometers placed on the wrist and waist of the subjects. Each subject performed 4 different ADLs (i.e. sitting down, walking and picking up an object from the floor, walk back and forth, lie down on the mattress) and 4 fall types (i.e. falling flat to the floor towards the front, side or backwards or seated on the floor after a backward fall). Nevertheless, the different types of falls conclude on the floor and actions appear rigid and staged. This particular dataset is delivered by 11 subjects of unknown age, height or weight although the authors record some variation in height (1.62-1.97 m). The actions are extracted from a long sequence, i.e. the fall event is isolated from any other actions such as walking,

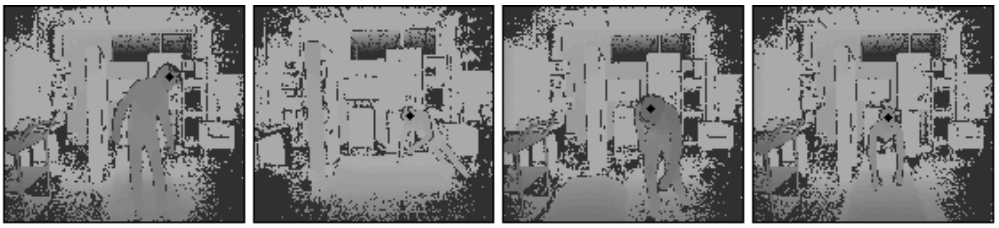

FIGURE 3.3: Visuals from the TST Fall Detection v2 dataset [3]

FIGURE 3.4: Visuals from the UR Fall Detection dataset [4] : a hesitated fall

before falling. The format of data makes this dataset difficult to process and as a result, this dataset is not used for evaluating other algorithms in the literature (i.e. is the less preferable dataset). Fig. 3.3 shows visuals from this dataset where fall or ADL events are happening in front of the sensor.

3.5.2 UR Fall Detection

The UR Fall Detection [4] is another dataset providing acceleration fall data and video (RGB and depth). It has been collected using a two camera configuration, one parallel to the floor and the other mounted on the ceiling. Annotations of other features, e.g. those characterizing the bounding box around the person, are also provided. The dataset consists of falls belonging to two categories: falls from standing position and falls from sitting on a chair. This is one of the most popular datasets that has been used by many other researchers for their evaluation and comparisons as it is very easy to process as its format is in PNG where pixel intensity denotes the correct depth (this involves a calculation according to sensor type). Nevertheless, it is only a small dataset of 5 subjects performing only 15 walking to rigid falls and 15 seated falls. Subjects clearly hesitate when performing a fall as seen in Fig. 3.4 where falls terminate on the floor without any cushioning mattress. Notice how the actor tries to reduce the impact by resisting using his arms.

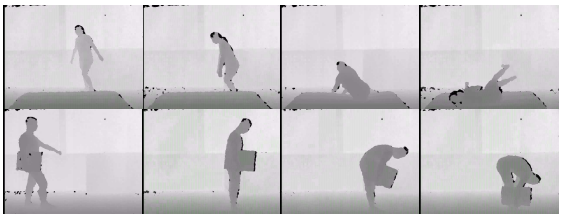

FIGURE 3.5: Visuals from the SDUFall dataset [5] : fall events at top row, ADLs on lower row

3.5.3 SDUFall

The SDUFall dataset [5] is one of the largest datasets comprising data captured from 20 people performing different types of falls (backwards, sideways) and 5 different ADLs (bending, squatting, sitting, lying and walking), with each subject repeating each action 10 times. In each repetition, the actors may or may not carry large objects, turn a light on or off, or change direction and position relative to the camera. This is another dataset where although there is a larger set of participants, the physical characteristics of each is not recorded. Many researchers have used this dataset as it has 200 fall samples in depth, RGB and skeleton, distributed in .avi format and text files. Fig. 3.5 shows visuals from this dataset wherein the top row a fall occurs and in the lower row is picking up an object from the floor while holding a briefcase.

3.5.4 University of Texas datasets

Three different datasets were collected at the University of Texas:

The Falling Detection dataset [105] has been collected in a laboratory-based simulated apartment set-up, with two Kinects mounted at opposite upper corners of the room. Six subjects perform 26 falls and several ADLs such as picking up a coin from the floor, sitting down on the floor, tying shoelaces, lying down on the bed, opening the low drawer which is close to the floor, jumping on to the floor, and lying down on the floor. The recording provides only depth data and there is no information about the participants or the camera setting.

The EDF dataset [6] extends the previous datasets in terms of data collection. The setting has remained the same in a simulated apartment where two Kinects have been installed to capture the events from two different directions, leading

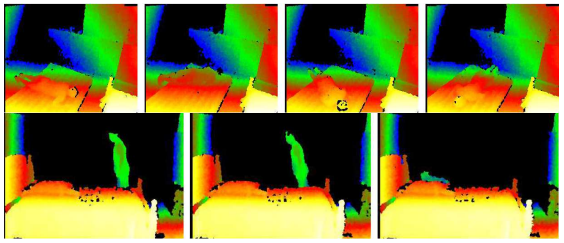

FIGURE 3.6: Visuals from EDF [6] and OCCU [7] datasets. Top row: falls repetition over four angles, lower row: occluded fall behind a bed

to a total of 320 sequences. In addition, 100 sequences of 5 different ADLs that could be associated with falls are recorded such as "pick up an object", "sit on the floor", "lie down on the floor", "tie shoelaces", and "do a plank exercise".

The OCCU dataset [7], as the EDF set, also uses the same setting. The main feature of this dataset is the presence of occluded falls for which the end of the action is completely occluded by an object such as a bed. Five subjects simulated 12 falls, 6 for each of the two viewpoints. Similarly to the EDF dataset, 80 sequences of actions that can be confused with falls are also provided. This is the only dataset where occlusions are introduced. Fig. 3.6 shows visuals from these datasets, where on the first row a fall occurs with different direction towards the sensor, whilst in the lower row an occluded fall occurs.

3.5.5 ACT42 dataset

The ACT42 dataset [8] mainly focuses on facilitating practical applications, such as smart house or e-healthcare, and contains 14 daily activities such as: Drink, Make Phone Call, Mop Floor, Pick Up, Put On, Read Book, Sit Down, Sit Up, Stumble, Take Off, Throw Away, Twist Open and Wipe Clean. Two categories of falls are considered, namely Collapse (fall due to internal factors i.e. heart attack, stroke etc.) and Stumble (fall due to external obstacles). The dataset was captured by 4 Kinect sensors from different heights and view angles. This is one of the first datasets showing data of collapsing fall event videos. Nevertheless, in the majority of those videos, it is noticeable how subjects hesitate to fall in a vertical direction towards the ground. Data regarding participants physical characteristics is not available and the sensors positions, although different in every capturing

FIGURE 3.7: Visuals from the ACT42 dataset [8]. Events captured from four views in RGB-D

scenario, is not recorded (e.g. height of sensor). Visuals from this datasets are seen in Fig. 3.7 where every event is captured by four cameras.

3.5.6 Daily Living Activity Recognition

The Daily Living Activity Recognition dataset [9] has data of subjects performing five activities related to falling event including standing, fall from standing, fall from sitting, sit on a chair, and sit on the floor, captured using a Kinect sensor. RGB, depth and skeleton data were provided in this dataset in 150 different data files, nevertheless, only 50 of those are available for public retrieval. Subjects perform events in-front of the sensor and without any occluded scenes. Other data is not recorded from the participants or sensor location. Fig. 3.8 shows several events from the dataset.

3.5.7 NTU RGB+D Action Recognition Dataset

This dataset [10] appears to have the most video samples of any set discussed in this chapter. This is a dataset not particularly prepared for falls as it contains only 40 fall events captured from different angles. The falls are not as realistic as we have seen in other studies which focus on the subject. The authors claim that there is a human variability on subjects such as age, height and weight, but this information is not made available. There are videos where the fall event does not conclude to a resting place on the floor, but the subject stops the fall and holds on with their hands. Falls appear to be conducted with minimum risk and hesitation is obvious. The fall actions appear without occlusions from objects. Currently, at the time of writing this work, this dataset has not been used for evaluating any fall detection algorithm. Fig. 3.9 shows visuals from the dataset where the first two images show ADL events and the last image a sideways fall.

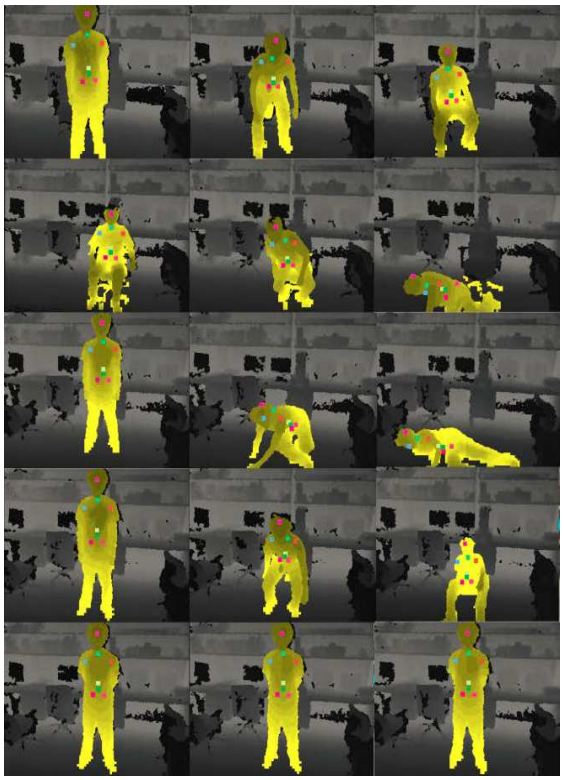

FIGURE 3.8: Visuals from the Daily Living Activity Recognition dataset [9]. Several ADLs and fall samples of RGB-D and skeleton

FIGURE 3.9: Visuals from the NTU RGB+D Action Recognition dataset [10]. Several ADLs and fall samples in RGB-D

3.5.8 UWA3D Multiview Activity dataset

Dataset [11] consists of 30 ADLs and a falling down event performed by 10 subjects (hand waving, one hand punching, sitting down, standing up, holding chest, holding head, holding back, walking, turning around, drinking, bending, running, kicking, jumping, moping floor, sneezing, sitting down (chair), squatting, two hands waving, two hand punching, vibrating, irregular walking, lying down, phone answering, jumping jack, picking up, putting down, dancing, and coughing). To achieve multi-view, five subjects performed 15 activities from four different side views. Nevertheless, only the front view is available at the time of this study for retrieval. Subjects' physical characteristics data do not appear anywhere in the information related to participants. Visuals are shown in Fig. 3.10 where on the

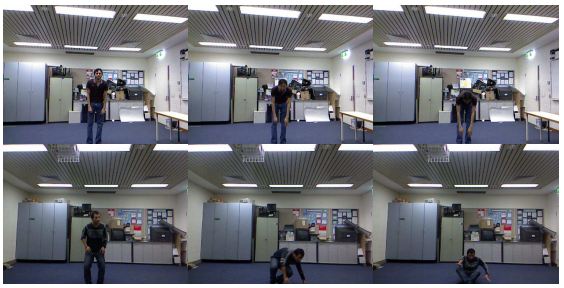

FIGURE 3.10: Visuals from the UWA3D dataset [11]. Performance of a bending over (top row) and falling (low row)

first row a person is bending over, while on the lower row the person performs a collapse with noticeable hesitation.

3.6 Accelerometer based dataset

3.6.1 Sisfall dataset

This dataset [106] records accelerometer and gyroscope fall data including the subjects' height, weight and sex alongside falling data provided by 3 different devices (two accelerometers and one gyroscope) mounted on the waist of the participants. Unfortunately, only a few videos are provided from this study just for viewing purposes and to distinguish the different types of falls. This is also a dataset where elderly subjects perform ADL events. Data from this study has been used for evaluating several algorithms and assumptions made in Chapter 6.

3.7 GM depth dataset

At the start of this study (2011), no depth-based public datasets were available for training and validation purposes. Fall events and scenarios were simulated by humans using falling scenarios seen in real-life as well as fall videos available online in RGB video.

Hence an in-house dataset was created using Kinect I sensor. The Kinect IR depth sensor captures videos at 640 $\times$ 480 resolution at 30 fps. The fall actions were recorded from a direct view of the scene making sure that the fall event is fully captured. For that reason, the Kinect has been attached to a tripod at the height

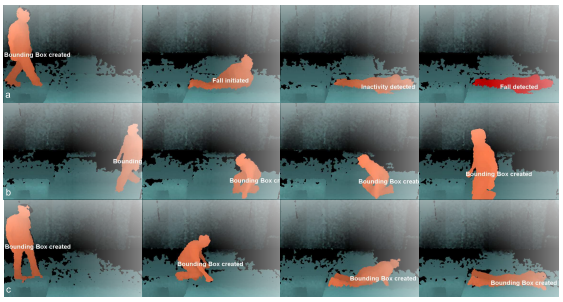

FIGURE 3.11: Visuals from the dataset. Row a) a detected fall event, b) walking and picking up an object, c) lying down

FIGURE 3.12: Three view-angle examples of the GM dataset. a) 45 view , b) side view, c) front view

of 204 cm and inclined to the floor plane. The Kinect monitored an area within 7 metres of the acted events which is within the range of the IR sensor.

The dataset is comprised of 184 video sequences, of actions that include: 48 falls (backward, forward, sideways), 32 seating actions, 48 lying down actions on the floor (backward, forward, sideways) and 32 "picking up an item from the floor" actions, performed by 8 different subjects (6 male, 2 female). Visuals of the dataset are shown in Fig. 3.11. 24 other activities that change the size of the 3D bounding box were also performed (e.g. sweeping with a broom, dusting with a duster, picking up a char or a box and placing it back). Table 3.1 lists the number of different types of actions included in this dataset.

Those videos were captured from three different view-angles (captured separately) in order to provide several different views of an activity in a real environment, as seen in 3.12. That is, for each trial, the person was changing their direction towards the sensor. Subjects performed the fall actions on a 30 cm thick mat in order to prevent injury and to capture more realistic falls.

Further investigations required the addition of videos of collapsing falls, due to the fact that limited number of examples of such fall type were included on the public available datasets. The $GM2_A$ dataset has 40 collapsing fall samples from 3 subjects. After the comparison discussed in 3.10, the selected samples from the $GM2_A$ dataset will be merged with the existing **GM** data to form the **GM2** dataset. For

Table 3.1: RGB-D Fall datasets. R: RGB data, IR: infrared data, D: depth data, A: accelerometer data, S: Kinect skeleton data. The table shows the different fall event datasets of several sensor technologies. Noticeable is the number of fall events if compared with the ADLs as well as how small the fall number is in general

Dataset	Subjects	Actions	Fall Samples	ADL Samples	Data Type
Multiple cameras [2] 2010	1	9	24	99	R
LE2i [1] 2013	9	7	143	48	R
TST v2 [3] 2016	11	5			D, S, A
UR [4] 2014	5	6	30	40	R, D, A
SDUFall [5] 2014	20	6	200	1000	R, D, S
Fall Detection [105] 2012	6	8	26	61	D
EDF [6] 2015	10	6	160	50	D
OCCU [7] 2014	5	5	30	80	D
ACT42 [8] 2012	24	14	48	672	D, R
Daily Living [9] 2012	5	5	10	40	D, R, S
NTU RGB+D [10] 2016	40	60	80	4720	R, D, S, IR
UWA3D [11] 2014	10	30	10	290	R, D

$GM2_A$ dataset, events were captured using a Kinect I and an Astra Orbbec sensor. To avoid interference a shake 'n sense [88] approach was adopted where one or both sensors use a vibrating means to disturb the laser signal interference.

3.8 Limitations of existing datasets: A discussion

In general, as discussed, computer vision algorithms require a significant amount of data for training which in this particular field is sparse and of questionable quality in terms of how realistic the fall event is. Table 3.1 summarises the samples found in public datasets, which specifies the number of subjects and samples of each dataset. In the above-discussed datasets for action recognition, falls are a very small class of experiments as it is easier and less risky to perform common actions, such as walking, sitting, greeting etc.

Genuine fall data is not readily available, particularly for vulnerable people as there are complications in collecting and distributing it. There are ethical reasons which prohibit elderly and infirm from participating in data collections that involve falls due to the fragility of the body. The few genuine data recorded from actual scenes recorded from hospitals or assisted living homes, is not available, mainly, due to

reasons of privacy and ethical approval. As a result, researchers have implemented human-simulated falls in order to develop fall detection algorithms and fill the data availability gap. Acting participants are asked to perform an event which in reality is performed without our consciousness (e.g. is a non-conscious action when we fall due to dizziness or when we stumble on something). Such implication makes the data collection a difficult task as the actors find it unpleasant to perform.

A fall is not an event that occurs often within the day and a result we are not prepared or willing to perform such an action for the sake of data recording. Even when we decide to perform such staged falls, we may be reluctant to fall realistically, due to the risk of injury and it was never a case to hire professional actors (stunt men) who are trained to simulate realistic falls.

The following sections discuss in detail the issues with existing datasets and recording practices and provide the reader with an insight into their limitations.

3.8.1 Age of participants

All the datasets consulted (excluding the SisFall) provide limited data regarding the age of participants. In general, the elderly are not represented in any of the datasets - even if those actions are not fall related. The available fall event data recordings are performed by young people, mainly students and researchers from an academic institution under instruction from the researcher to collapse "normally". In such circumstances, self-preservation takes over and the fall event will be unrepresentative of genuine falls, particularly if the aim is to acquire data representative of the vulnerable population (i.e. elderly people).

Particular emphasis is to be given to the elderly as the target group where this study can potentially find an application. As discussed in Chapter 1, the elderly are more prone to fainting due to a number of causes. Loss of balance can be due to muscle weakness or other medical conditions such as reduced brain functionality, blood pressure issues, visual impermanent, confusion and disorientation due to mental issues. Nevertheless, current datasets are intended to apply to algorithms which aim to detect elderly falls.

3.8.2 Health of participants

To participate in these datasets and perform a fall, actors are asked about their physical condition or even mental state. If issues exist which conceal a risk, such participants will be excluded due to restrictions set by ethics committees when human subjects are asked to perform tasks for data recording. Therefore, only the healthy participate in these datasets and vulnerable population are excluded from the study. Most of the existing work in fall detection discusses the applicability of their approach without investigation of such populations. In reality though, falls can cause death mainly from an ill-health or old-age person.

3.8.3 Types of falls

The discussed datasets have samples from mainly one type of fall (i.e. rigid). The collapsing fall type is collected in a few datasets, but the falling behaviour is unrepresentative of a real collapsing event. One possible reason for avoiding the performance of such fall is the risk of injury, particularly for the knees [107]. Section 3.10 tries to justify and assess the hesitation in some of the collapsing falls.

3.8.4 Size of datasets

The small number of human actors performing fall events may not be sufficient to represent the entire population. For example, one of the largest datasets [10] for fall detection consists of only 40 fit, young and healthy male and female subjects performing falls and other ADLs. Whilst, the number of participants may be sufficient, they are not sufficiently varied. However, even compared with the number of recorded falls every year in a given country such as the US or the UK, it can be considered as small. The small number of samples may play a significant role in the accuracy of a fall detection algorithm when applied to a real scenario, as the algorithm will have been trained on a small amount of data, limiting its robustness.

3.8.5 Variability of subjects

Variability in human body morphology assotiated with as the height, weight, age, or gender are factors which are generally ignored. An elder has different posture from a young person and a pregnant woman may walk differently from someone with a broken leg. The answer is not to ask any of these groups to perform falls, but to give an appreciation of how the existing datasets lack in variability. Therefore, algorithms based on these limited datasets may have questionable performance when applied to a broader demographic. For example it is noted in [108] that men and women have a different centre of mass. This is subjected to height, weight and anteroposterior depths of a person.

3.8.6 Hesitation

Human subjects performing staged falls may have difficulty in acting realistically due to hesitation associated with the concern of having an injury. A *hesitated fall* is defined for this study as a fall event where the person undergoes the fall but extends arms to minimise the impact against the head or turns on the side to avoid knee impact.

The risk of injury is an important factor when permission is sought to conduct fall experiments, hence, the type of falls may be organised to follow a restricted protocol specified by regulations of health and safety or ethical considerations. Researchers then have no option but to warn the participants of any complication in the case of injury and may request disclaimers, particularly if they were deployed in a real environment. As a result, data from such non-realistic recordings may have a negative impact on an algorithm's performance.

3.9 Scene set-up

Some of the discussed datasets including the RGB ones provide example events from actual home scenes. Although such scenes appear realistic, it is far from a usual home scene as rooms are sparsely furnished and unrealistically configured, where few occlusions are visible since most of the furniture is located near the walls.

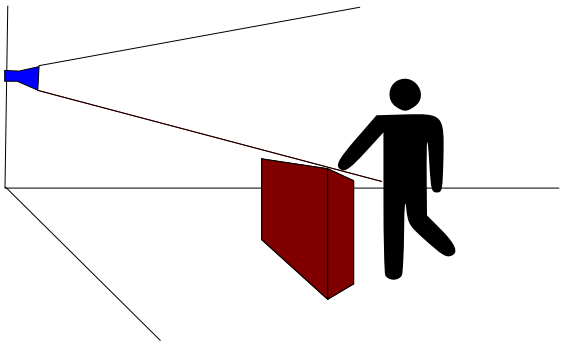

FIGURE 3.13: Typical occluded scene. The camera view is partially blocked by the red box. Half the person is occluded.

3.9.1 Occlusions

As noted from the analysis of the datasets in Sections 3.3 3.5, fall events appear fully visible in the video scenes without any scene occlusions. Datasets generally include fall event videos without other objects appearing nearby unless this object is used, (such as a chair, stool, or a bed) and as a consequence, occlusion scenarios are rarely represented. The lack of occlusions in most existing datasets is unrealistic for virtually all indoor (home) environments. Therefore, in the event of an occluded fall, current algorithms are generally untested for such scenarios. In a home scene we may get non-occluded views, but as people move around a cluttered environment there may be frequent occasions during which they are part-occluded, to various degrees. Fig 3.13 illustrates an occlusion obstructing the view of a person.

Although many studies discuss the application of fall detection for the elderly, at home or in hospital, occlusion is rarely mentioned, hence, methods are not evaluated to provide occlusion-robust solutions.

In an occluded home scene, a fall detector should rely on features that are visible and stable. The issue is that many fall detection algorithms may use one or more features that are more adversely affected by an occlusion on the ground plane (e.g. CoM); the head location would seem to be the single feature least susceptible to occlusion. An approach to dealing with occlusions is to use several cameras as seen in several datasets in order to maintain a continuous view of the scene, though this is still not guaranteed to eliminate the possibility of occlusion.

An attempt to evaluate current algorithms under occlusions is discussed in [7] where authors have developed an occluded dataset and evaluated several state-of-the-art algorithms. Five subjects perform fall events which conclude with the fallen person completely occluded behind a bed. Yet, the level of occlusion that is normally caused by a bed is fairly small (approximately 30%) and even this dataset fails to provide a proper evaluation of partially occluded events.

3.9.2 Sensor location

Only a few studies/datasets make note of the sensor location. The position of the sensor plays a significant role as to where the best field of view is achieved in order to maintain a clear view of the home scene. This is unrelated to the minimisation of occlusions as even if the sensor is located higher, occlusions may still occur. The sensor location in some cases plays a significant role in how the person appears, hence, an algorithm is designed in order to detect a fall using that particular data. See the example in [4] where the depth sensor is located on the ceiling, pointing downwards. In other cases the sensor is placed on a table, which seems unrealistic for a home scene. Obviously, the location, in this case, aims to detect the height variation of the falling person, rather than how depth/length of human body changes during a fall. Also, by placing the sensor at a low height, the view is more prone to self-occlusions. In this scenario, a fall may start near the sensor and conclude on the floor in front of the sensor and possibly under the f-o-v of the sensor - implying that the fall is outside the viewing window.

3.9.3 Data quality and adaptation

One of the issues in using public datasets is the recording format and how other researchers can use the data. In a few cases, depth data were compressed resulting in poor depth information, or in other cases, the depth information was less reliable and as a result, further time was required to address such issues. Different depth sensors or OpenNI/Microsoft Kinect SDK versions were delivering different video/image formats which were time-consuming to use or convert.

3.10 Comparison of real vs acted falls

This section discusses the level of realism in terms of how actors behave during a fall. A comparison of real falls and staged ones is necessary to show how different real falls are. To investigate this, falls were taken from public video channels, such as YouTube. In recent years YouTube has become an increasingly useful source of video data. Searching for appropriate falls is still a challenge as some of the fall videos are not of the right quality for processing. Such videos do not require ethical consideration and can be used as is. For this study hyperventilation videos were used, where young people hyperventilate themselves until they faint. Some of these videos show the realism and how violent a fall can be.

Using a camera calibration feature tracking software, YouTube hyperventilation videos were processed in order to measure the person's head vertical velocity V_y. Velocity is selected as a comparative feature as actors who hesitate try to slow down the fall by applying force to their knees or extend their hands to the ground in order to minimise the impact. This behaviour will cut-off the action and the velocity of the head will be different from the real fall event.

Noticeable hesitation is observed on collapsing fall videos found in ACT42 [8] (48 samples) and $GM2_A$ (40 samples) datasets. The head's vertical velocity is measured from videos from these two datasets. To measure the similarity between the velocity profiles of the real and acted falls, the Hausdorff distance (HD) is calculated (validation of HD is discussed in section 6.8.1).

Fig. 3.14 shows the velocity profiles of three different falls; from the $GM2_A$ dataset (blue and green graph) and a real YouTube fall event (red graph). The Hausdorff distance between the YouTube and $GM2_A$ hesitated profiles was 2.87 m/sec, while the HD of YouTube and GM2 was 0.673. To show the difference between the realistic and hesitated falls, a pdf is plotted as seen in Fig. 3.15. Two classes are visible, one with 16 examples of realistic fall events when compared with YouTube ones and 24 samples which when compared with YouTube videos are classified as a similar to a non-fall event (see Fig. 6.12).

These results are selected in order to show a valid collapsing fall according to this evaluation and a hesitated/unrealistic one from the $GM2_A$ dataset. Samples with a small HD distance in velocity, when compared with a fall velocity profile, denote

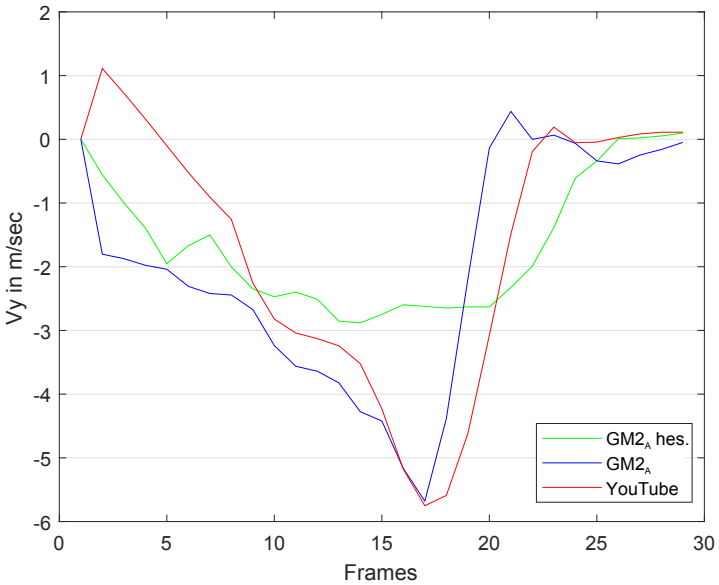

FIGURE 3.14: Velocity profiles of collapsing falls. This Figure shows the velocity variation between a hesitated fall ($GM2_A$ hes.) and how this compares with an actual fall caused by hyperventilation

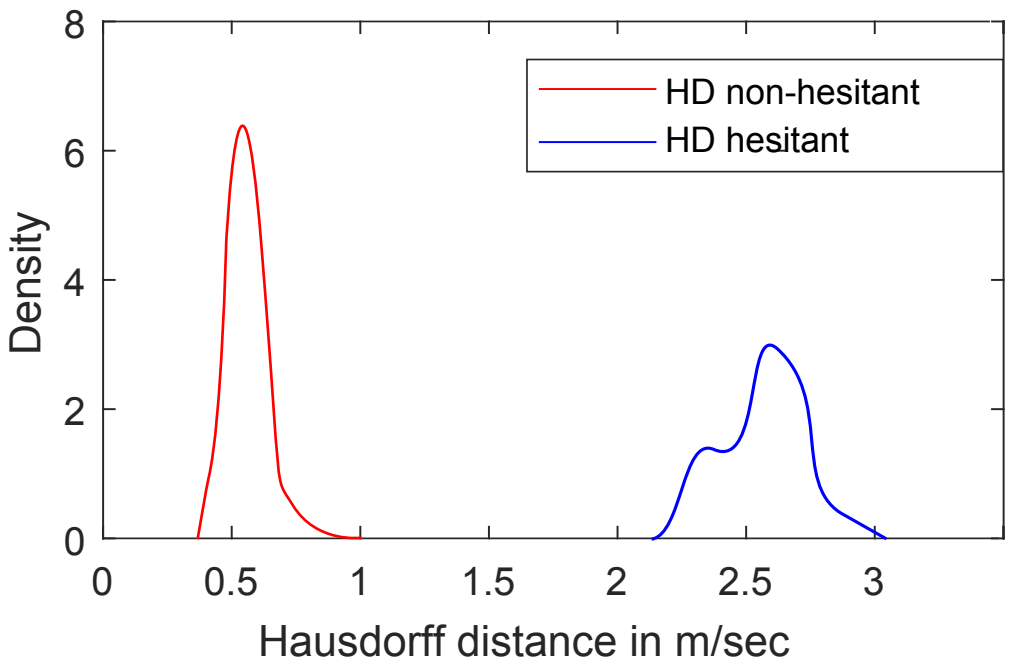

FIGURE 3.15: Plot of a Gaussian pdf fitted to the distribution of Hausdorff distances: red curve denotes the HDs of YouTube to non-hesitant falls, while blue curve the HDs between YouTube and hesitant falls

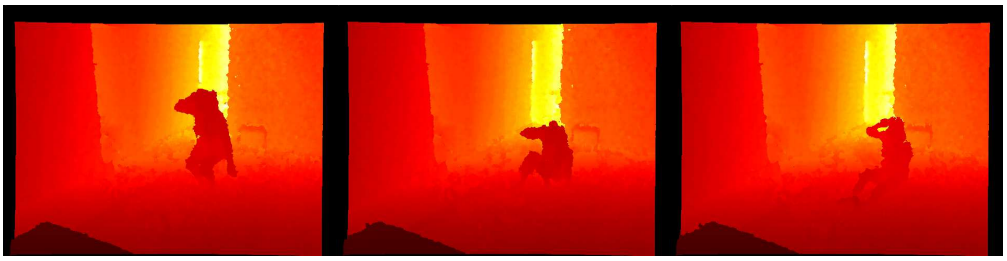

FIGURE 3.16: Hesitation while collapsing. Subject uses an arm to balance, then expands her legs to sit on the floor

that their falling behaviour is more similar to a realistic fall as evaluated in Section 6.8.1.

Visuals in Fig. 3.16 show hesitation due to self-preservation, resulting in an unrealistic fall example as the actor uses an arm to balance, then extends her legs to sit on the floor. Only 7 examples from ACT42 and 16 examples from $GM2_A$ were classified as realistic collapsing fall samples. This result indicates how actors hesitate in performing a fall in a realistic manner and hence, only a small sample of 16 examples from the $GM2_A$ dataset is usable for evaluation purposes. These 16 videos will be used as an addition to the GM dataset in order to provide samples of collapsing falls to form the **GM2** dataset.

3.11 Performance Evaluation Measures

Results are presented using the following performance measures: the number of correctly detected falls (TP), missed fall detections (FP), ADLs detected as falls (FN) and ADLs that are not detected as falls (TN). Accuracy (Eq. 3.1) gives the proportion of true events that were correctly classified across all measurements. Precision (Eq. 3.2) is the proportion of positive results that were correctly classified. Sensitivity (Eq. 3.3) is the proportion of actual positive event results correctly classified and specificity (Eq. 3.4) is the proportion of negative results correctly classified.

$$Accu = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

$$Prec = \frac{TP}{TP + FP} \tag{3.2}$$

Table 3.2: Performance of fall detection and comparison against previous studies across 2 public datasets

Method	Accu (%)	Prec (%)	Sens (%)	Spec (%)
(a) Dataset UR (15 fall, 40 ADLs)				
[45]	95.00	90.91	100.0	90.00
[4]	98.33	96.77	100.0	96.67
[79]	-	-	100.0	97.25
[80]	99.37	96.77	100.0	99.23
(b) Dataset SDU (200 fall, 800 ALDs)				
[5]	86.83	-	91.15	77.14
[109]	91.89	-	-	-
[110]	92.98	-	93.52	90.76

$$Sens = \frac{TP}{TP + FN} \tag{3.3}$$

$$Spec = \frac{TN}{TN + FP} \tag{3.4}$$

Two public datasets UR [4], SDUFall [5] and the GM and GM2 of people performing fall events and ADLs are used to evaluate the algorithms developed in this thesis. These public datasets were selected as they are frequently used by other researchers and hence a comparison measure is available for other fall detection methods. Table 3.2 compares the performance of other researches using these public datasets. For brevity, these datasets are labelled as **UR**, **SDU**, **GM**, **GM2** respectively. These datasets contain a variety of fall events and non-fall actions, such as picking up objects, lying down or other actions that can trigger a false positive decision. Also, the sensor mounting position for these datasets was at varied heights and falls were occurring at varied locations in the scene. Table 3.3 summarises the number and type of actions in these datasets. Note that **GM2** dataset is merged with **GM** whenever an evaluation is required, therefore, **GM2** has 64 fall events, 32 sitting events etc.

3.12 Conclusion

As discussed, the public datasets provide insufficient information regarding the actors' physical characteristics such as age, height, weight etc. in order to assess

Table 3.3: Type and number of events from each dataset

	Datasets				
Actions	**UR**	**SDU**	**GM**	**GM2**	Total
Fall	15	200	48	16 + (48)	279
Sitting	9	200	32	(32)	241
Picking up	-	-	32	(32)	32
Squatting	8	200	-	-	208
Lying	16	200	48	(48)	264
Bending	7	200	-	-	207
Other	-	-	24	(24)	24

the impact of these characteristics on fall detection algorithms. Sometimes, they mention that their datasets contain males and females without specifying further information. Furthermore, the number of fall events or the participating actors is limited when compared to other action recognition datasets. These public datasets do not include data of the specific group for which they are intended, such as the elderly and infirm. Hesitation is described as one of the issues associated with unrealistic fall behaviour. This is observed mostly on all types of falls but it is more severe in collapsing events. Therefore, an evaluation of these collapsing videos was undertaken in order to filter out the hesitated fall events using comparative induced falls via hyperventilation from YouTube.

Also, these datasets lack visual occlusions hence available algorithms are not intended to be occlusion robust. A change in furniture may require sensor relocation to overcome a new scene occlusion and a change of habits may require signal replication of the wearable receiver. Somehow, methodologies are required to bridge the gap between these inconsistencies and improve performance without human intervention. Such approaches will be discussed mainly in Chapters 5, 6 where the falling behaviour is not trained by a data-driven approach.

RGB data provides detailed views of faces (as seen in 3.1), hence to preserve privacy, such technology is inappropriate. The use of a depth camera resolves this issue since disparity images are of such resolution that identifiable characteristics such as the face are obscured.

Two datasets have been selected [4, 5] for evaluation as they are widely used for evaluation in other studies. These datasets consist of falls by actors performing several types of falls (backwards, forwards etc.) as well as ADLs.

Chapter 4

Learning to detect falls

4.1 Introduction

This Chapter describes an initial investigation of fall detection using depth data for training and testing a Random Search [111] machine learning algorithm. Two real-time algorithms were developed that utilise a 3D bounding box to parameterise a body shape, expressed in world coordinates: the first method is described in Section 4.2 and published in [27] and the second detailed in Section 4.3. The first study focuses on a rigid fall whilst the later considers rigid and collapsing falls. Discussion about fall types is given in Section 3.2.

From the 3D bounding box, the first algorithm calculates the first derivative (velocity) of width, height and depth in order to determine whether a particular activity is a fall or not. The algorithm does not require pre-knowledge of the floor plane coordinates or the detection and tracking of any particular body part, as used by some other systems [112–114]. Several studies discuss the fact that during the fall, the width of the 2D bounding box is expanding, while the height is contracting [12, 13]. Those studies require the initial and final aspect ratio of the 2D bounding box to confirm a fall, while the proposed approach does not measure the initial/final bounding box dimensions.

The second algorithm utilises a conservative 3D bounding box which filters out the motion of the arms and legs. Instead of length, width and height, this method uses an angle. The angular velocity and inactivity, as well as the size of the angle at the end of every event, are used to distinguish a fall from a non-fall.

These algorithms were tested to detect a range of falls (backward, forward, sideways), while setting the sensor to different orientations, providing different views of the human body (side view, frontal view, back view) and different type of actions performed at different speeds. These non-fall activities are challenging because they could lead to false positive (FP) detections especially when a person is lying down, crouching down, picking up an item from the floor, etc. This is due to the similarity these actions have to the fall according to the velocity profiles. The main algorithm is designed as a two-step Boolean decision tree where several output variables are checked sequentially. The parameters of the decision tree are estimated by Random Search optimisation [111]. Furthermore, the usage of OpenNI [103] significantly helps the pre-processing of the depth data in terms of background subtraction and user identification.

The Chapter is organised as follows: a review section of the 3D vision systems, followed by sections describing the two algorithms. Experimental Results and Discussion section explains how datasets were used to evaluate the algorithms via different protocols.

4.1.1 Review of 3D vision systems

Vision depth image systems use 3D cameras or depth sensors to track and analyse human motion. This review section focuses on fall detection algorithms using depth data developed prior to publication of this work [27] and is more sensibly discussed here rather than in Chapter 2 where a generic review of algorithms is given. At the time of this work, only a few studies ([34, 112, 113]) using depth/infrared data for fall detection existed and a criticism of those was required to pinpoint their complications and possible performance issues.

4.1.2 Technical Criticism of 3D methods

Since this system analyses depth information using Kinect's IR sensor a more detailed analysis will be given in order to emphasise the benefits and weaknesses of the existing approaches. In [113] the authors use a 3D camera to develop a monitoring system for elderly people, which is also capable of detecting falls. Their approach involves fitting an ellipse around the subject after a series of pre-processing steps (image thresholding, smoothing, eroding and dilating) in order to

have images resulting with fewer blobs (assuming that the biggest blob defines the human silhouette). Next, their algorithm maps the centre of the blob into world coordinates by a linear calibration method. For distinguishing activity patterns of fall-like actions the authors use an online learning method described in [115].

However, their methodology requires considerably more processing time due to the online learning process; it requires pre-knowledge of the scene (world coordinates), which depends on the visibility of the floor (occlusions, objects laid). Also, the description of falls or other activities is not defined in their work i.e. one can vigorously sit on a sofa; the viewing position may be different; the "lying sequence" are comprised of several different postures not clearly defined. Finally, there is no proper evaluation of their algorithm, as it is tested only on one subject, without consideration of FPs or missed detections.

Diraco et. al. [112] describe an approach based on the distance of a falling person from the floor, inactivity and pose estimation. The floor is detected using RANSAC [116] which fits a plane to a 3D point cloud that covers the largest area. This off-line process requires extra time to perform and is required whenever the camera is installed. It is a complex process that requires the detected planes and the external calibration parameters and is performed in two steps: firstly detecting large enough planes and secondly filtering those planes. Next, their method calculates the 3D centroid of a person and measures its distance from the floor. If this distance is below a certain threshold the algorithm checks whether there is any further motion/activity. A fall is detected by combining the distance of the body's centre of mass from the floor, the inactivity of the fallen person and the orientation of the body spine as derived by a 3D pose estimation (Reeb Graph [117]). However, the latter is computationally expensive.

Rougier et. al. [34] propose a Kinect based system to detect falls. Their system firstly uses the subject's centroid to measure the distance from the floor. Then, they use the centre of mass to calculate the velocity. A fall is detected when velocity is above a certain threshold while the distance of the centre of mass to the floor is below another certain threshold. The floor is detected by a histogram analysis of a V-disparity image [118]. The authors claim that their algorithm is able to identify a fallen person while occluded, based on the velocity detection, although this is stated without justification or evaluation against some video samples. Their evaluation is limited to a small number of samples with no information about the number of subjects performing the falls. In addition, there is no clear description

of the type of falls performed and their experiments do not include fall-like activity patterns i.e. when someone is picking up something quickly from the floor, lying down quickly on the floor or vigorously sitting on a sofa as seen on examples from GM dataset. Therefore, there is no evidence that an FP is avoided when a person performs a fall-like activity.

As an overall criticism, one can say that all these systems require floor coordinates to operate. Furthermore, they do not provide any specific information regarding tracking the subject or any further information regarding how the activity patterns have been defined such as the CoM velocity. Also, another important point is the rather limited number of experiments (raw dataset) used for evaluating those methods (except [112]).

4.2 Detecting rigid falls

This section describes the techniques for fall detection. The proposed algorithm analyses the depth information of the subject (3D bounding box). OpenNI provides a method (UserGenerator) to analyse the depth information of the scene. UserGenerator performs background subtraction and motion tracking. For this analysis only three parameters were used as estimated by OpenNI, that is the width, height and depth of the human posture, which defines a 3D bounding box. This simplified set of parameters delivers a more reliable result than articulated pose estimation. From the early experiments, it is found that pose estimation may fail during a fall and is not possible to recover a fallen posture at its final state. Also, further analysis of the 3D articulated model requires significantly more computational power than the 3D bounding box analysis. The next subsection discusses in more detail the 3D bounding box extraction, while the following subsections describe how the 3D bounding box's parameters are used to detect a fall.

It is not a requirement for the proposed algorithm to calculate and use the floor coordinates as previous approaches do (see Section 4.1.2). Further to that, it is noted that a fall is a fast action and a high frame rate in real-time systems is advisable to avoid missed detections. The development of a real-time algorithm is required to capture these rapid changes over short time.

FIGURE 4.1: Depth map of the scene. User is identified by OpenNI

4.2.1 Overview

Pose estimation for segmenting the person is performed by the algorithm described in [119], where randomized decision forests are trained on randomly synthesized depth images containing annotated body parts of each person. The 3D bounding box is created using OpenNI's DepthMetaData process to contain the depth map of the user with world coordinates X_{max}, Y_{max}, Z_{max}, and X_{min}, Y_{min}, Z_{min}. The width, height and depth of the 3D bounding box are estimated as the differences between the maximum and minimum points along the X, Y and Z dimensions respectively. Hence, width $W = |X_{min} - X_{max}|$, height $H = |Y_{min} - Y_{max}|$ and depth $D = |Z_{min} - Z_{max}|$. The initial subject's detection and tracking are operated by a standard OpenNI function as seen in Figure 4.1. Traditionally, the position of the 3D bounding box is tracked to estimate the motion of humans or other objects. In the proposed approach, a fall is detected by analysing the 3D bounding box's width, height and depth and ignoring the global motion of 3D bounding box. The depth of the bounding box is denoted as the difference of the closest to the farthest point of the person segmentation point cloud. Hence, the person's depth is an approximation based on the visible-to-the-sensor side of the body.

4.2.2 3D Bounding box data analysis

As described in the previous Section, each user is wrapped into a 3D bounding box and the dimensions of the 3D bounding box are the only input the algorithm requires to operate with. OpenNI fits a new 3D bounding on each frame, extracting a new set of width, height and depth values. The algorithm analyses those values, and their first derivatives (e.g. $\dot{H}$) to detect a fall.

The diagram in Figure 4.2 (a) shows a 2D bounding box of a falling person when the sensor is recording the scene from a side view. In this method, a 3D bounding box (Figure 4.2 b) is used which behaves similarly but uses three dimensions instead of two. The height of the 3D bounding box will contract during the fall and the width and/or the depth will expand. A combination of the two expanding dimensions of the 3D bounding box W and D is calculated to define the motion of the person in the horizontal plane. The composition of depth–width is given $L = \sqrt{D^2 + W^2}$. This combination eliminates the use of three different thresholds for height, width and depth into two (i.e. H and L) hence, will optimise the training procedure. Also, the combination of depth and width normalises the noisy depth signal and provides a smoother signal for further processing.

Figure 4.3 shows the change of width, depth, height and width–depth composition of the bounding box as well the first derivatives of the height and the composition of width–depth during a fall. Notice how the height (pink line) drops while the width (green line) expands during the fall and the velocity increase as the person is falling. The composition of depth and width has similar values to the width as depth remains relatively unchanged during this type of fall. It is also observed that the signal delivered by OpenNI is quite noisy, especially in regards to Z dimension, therefore, a discrete Kalman filter [121] is applied in order to smooth the velocities as seen in Figure 4.3. The filter is defined by the following estimations:

$$prior \begin{cases} \hat{X}_k^- = A\hat{X}_{k-1} + B \\ P_k = AP_{k-1}A^T + Q \end{cases} \tag{4.1}$$

where $\hat{X}_k^-$ is the state prediction, A is the state transition, B is the control signal, P_k the error covariance prediction, Q process noise covariance

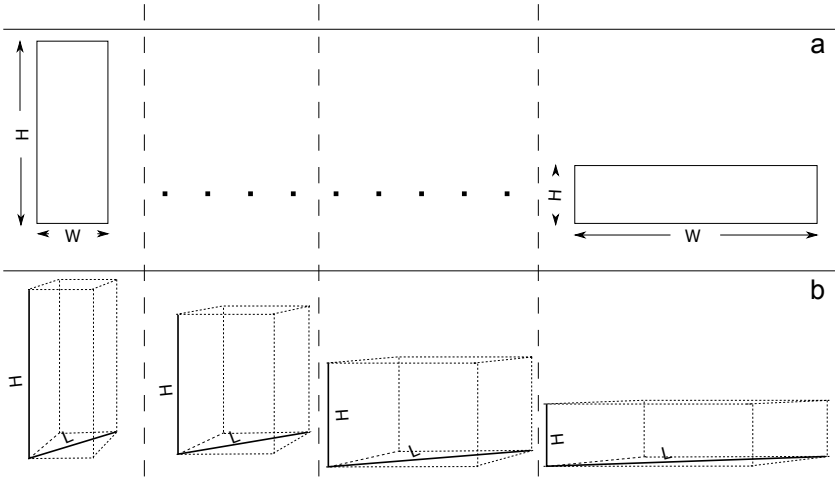

FIGURE 4.2: 2D Bounding box during a fall; the height reduces while the width increases (a) as seen in [12, 13], where the initial and final bounding box dimensions are required. The proposed approach using a 3D bounding box of the height and the composition of width and depth (b)

$$posterior \begin{cases} \hat{X}_k = \hat{X}_k^- + K(z - H\hat{X}_k^-) \\ P_k = (1 - \hat{X}_k H)P \end{cases} \tag{4.2}$$

where, $\hat{X}_k$ is the state correction, $\hat{X}_k^-$ is the state prediction, K is the Kalman gain, z the actual measurement

4.2.2.1 Fall initiation by velocity

Humans are articulated *objects* and hence their motions can be complex. However, it has been seen that a falling activity can be differentiated from other activities such as sitting, bending or lying mainly by the velocity of the centre of mass [62]. However, estimating the true centre of mass may be complex. Instead, the algorithm measures the velocities from the changes in H and L. The resulting $\dot{L}$ and the $\dot{H}$ are checked during N sequential frames. The velocity thresholds for the height $T_{\dot{H}}$ and the width–depth composite vector $T_{\dot{L}}$ of the bounding box, as well the duration of the fall (N frames) are estimated by performing Random Search [111] that optimises the classification score in a training dataset. The training procedure is described in 4.4.1.

When both velocities $(\dot{L}, \dot{H})$ exceed particular thresholds ($T_{\dot{H}}$ and $T_{\dot{L}}$) (e.g. 3D bounding box's height velocity, etc), a fall initiation is detected. Alternatively,

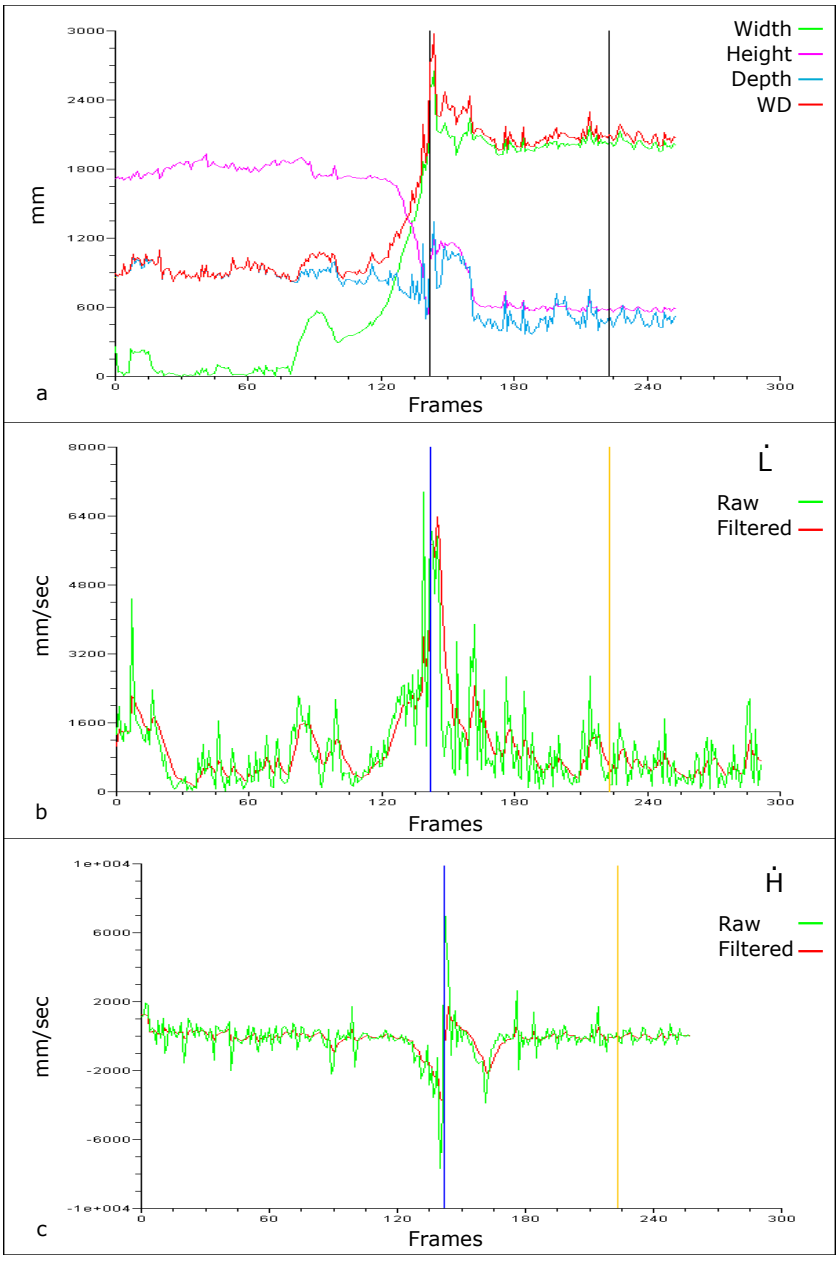

FIGURE 4.3: Bounding box dimensions and velocities. a) width, height, depth distances and width–depth graph. Vertical lines denote the initiation by velocity step and the fall detection confirmation step. b) $\dot{L}$ raw signal in green and filtered using Kalman in red. Similarly in (c) $\dot{H}$ of raw and filtered signal.

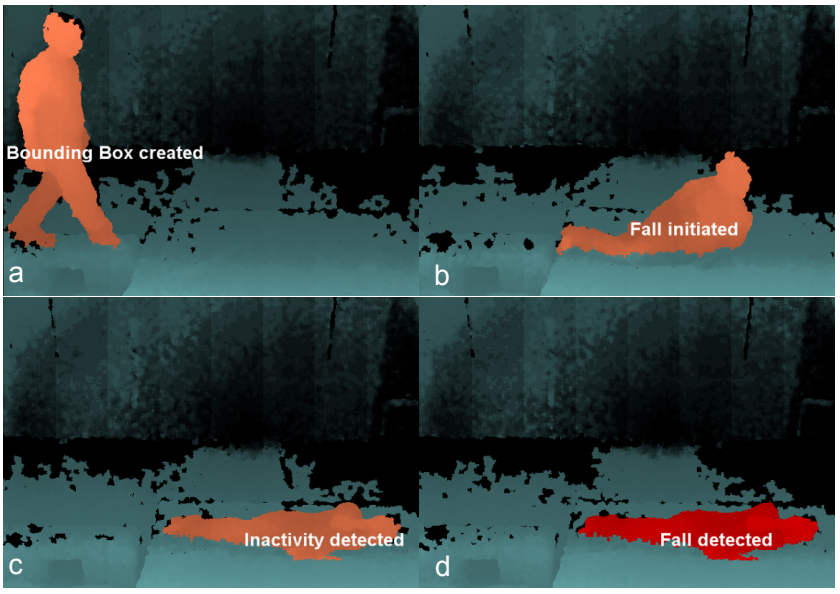

FIGURE 4.4: Side view of a sideways fall. Bounding box already detects the user (a), fall initiated by calculating velocity (b), inactivity detected (c), fall detected (d)

as the two velocities are correlated, it would be possible to train a 2D linear discriminator [120] to learn the critical threshold for detecting falls. The next paragraph discusses the final step. Figure 4.4 (b) shows the visual result of velocity detection for a side fall, captured from a side-on view.

4.2.2.2 Completion state of a fall by inactivity detection

A fall is expected to end in an inactivity state where no motion is detected as the person falls unconscious (i.e resting place). Therefore, the fall completion is detected by checking the appropriate velocity condition. Specifically, the method involves monitoring the subject for some time (e.g. two seconds) to detect any motion (Figure 4.4(c)). If no motion is detected then the algorithm is flagged as "Fall Detected" (Figure 4.4(d)). It is only required for the height velocity ($\dot{H}$) to be less than a certain threshold $T_{i\dot{H}}$ to declare the state as inactive.

The operation of the proposed algorithm runs inside OpenNI's main loop of the depth map process and runs in real-time with computational time of 0.3-0.4 msec per frame. Algorithm 1 describes the operation of the method.

Algorithm 1 Pseudocode of fall detection algorithm

```
SET threshold T_Ḣ, T_L̇
SET threshold T_iḢ
SET counter a, b = 0
SET threshold Nframes
SET threshold Mframes
SET boolean activityDetection = false
SET boolean inactivityDetection = false
while run do
  if (Ḣ > T_Ḣ) and (L̇ > T_L̇) then
    if a = N then
      SET activityDetection = true
    end if
    SET a++
  end if
  if activityDetection = true and |Ḣ| < T_iḢ then
    if b = M then
      SET inactivityDetection = true
      SET Fall detected
    end if
    SET b++
  end if
end while
```

4.3 Detecting collapsing and rigid falls

A fall as an event may have different attributes according to the person's position or pre-fall motion pattern. As discussed in Chapter 3, data availability may be a limiting factor in developing fall detection algorithms. Hence, the fall detection algorithms can only detect the fall types available in these datasets. The issue arises when fall types existing in these datasets are of mainly one type, such as the rigid fall. A collapse is a fall event different from those appearing in virtually all publicly available datasets. The following Sections will look closely at this fall type and investigate a new algorithm capable of detecting both rigid and collapsing falls. A fall of this type occurs when a person collapses without any pre-existing velocity due to a prior activity i.e. walking. The person is falling vertically - at first - and then on the side before coming to rest on the ground.

Furthermore, the algorithm will be evaluated to other everyday tasks such as lying on the floor or sitting down in order to measure the robustness of the approach. The evaluation of the algorithm will use data from the GM, GM2 and public datasets.

4.3.1 Overview

The combined algorithm is designed to detect both fall types and minimise FPs when a fall-like event occurs. Authors in [31] perform a feature selection procedure to obtain the best feature for fall detection using depth data. Nevertheless, the selected feature vector, fails to detect a collapsing fall when tested against GM2 dataset (e.g. Sensitivity = 0). On the re-implementation of these features for this study, it was observed that fall events are missed by several cases. Therefore, a new feature is required to capture the behaviour of this type of fall and maintain a good detection rate for the rigid type. Before discussing the feature, the issues of bounding box analysis are discussed (something missed by many studies).

4.3.2 Conservative Bounding box analysis

One novelty of this work is the conservative 3D bounding box. Generally, the 3D bounding box is calculated based on the position of the body, as well as, the position of the hands and legs as in Figure 4.5. When the subject moves their hands/legs, the bounding box can change dramatically, although the torso remains almost motionless. Measures which comply with the bounding box will become problematic if, for example, a person moves their hands rapidly. In order to filter out the motion of arms and legs, a new conservative bounding box is developed which contains the torso area, which is generally rigid.

The points of the conservative bounding box are calculated as follows: By knowing the subject's centre of mass (CoM) in 2D space (OpenNI provides 2D and 3D bounding boxes) the method runs a horizontal cut (left and right). The cut points (see 4.7 pink dots near left/right of CoM) are the horizontal boundaries of the conservative 3D bounding box in X coordinates. For the Y coordinate, the same points as for an ordinary 3D bounding box are used - derived from OpenNI.

For Z (depth coordinate) a standard deviation (SD) of the full body depth pixels is used. This is to contain the extrema of arms and legs while they move towards the sensor. Figure 4.7 shows three examples of the SD for a standing person, when lifting an arm and a leg or extending their arms in opposite directions. Notice how values of SD for sub-figures (b), (c) remain near the values of SD for (a). Also, there are cases as in (a) where gaps are observed in the graph meaning that these depth pixels are not as many as the others. It is observed that SD on the depth

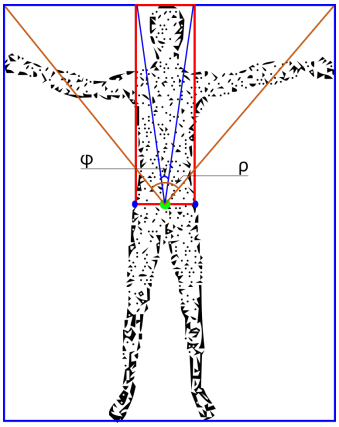

FIGURE 4.5: Conservative bounding box (red), ordinary bounding box (blue) ρ angle of 3D bounding box top corners and CoM, ϕ angle of conservative bounding box corners and CoM

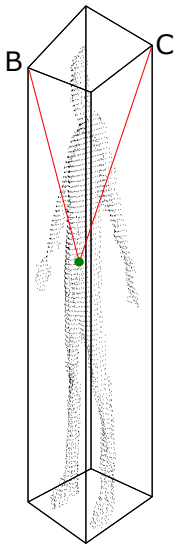

FIGURE 4.6: The angle from the two opposite top corners of the 3D bounding box to the centre of mass

pixels remains low when compared with the actual values of Z. Hence, when the person performs actions such the ones in these figures, the bounding box filters out these values and takes into account the SD values which reflect the similarity concentration of depth pixels of the human subject.

4.3.2.1 The ϕ angle

As already discussed regarding [31], a new robust feature is required to detect the collapsing fall type as well as the rigid one. It is observed that during a collapsing fall, the falling body has a vertical direction until the knees reach the ground and then the body inclines and reaches the ground. The use of $\dot{H}$ could be used alone for the detection of a fall, but this is insufficient as discussed in Section 4.2 and also in [31]. In order to capture the expansion of the 3D bounding box when the body inclines to the ground but also capture the change in height, a new feature is designed to capture both behaviours. An angle is selected drawn from the centre of mass to the two opposite corners of the conservative bounding box. When the person falls until reaching the floor with their knees, the CoM shifts upwards as the bounding box becomes smaller, making the angle larger as seen in Figure 4.9. This captures the change in height. While the person inclines, further expansion of the bounding box is observed, hence, the angle increases until the person is completely fallen. Fig 4.8 shows the angle's change as the person falls.

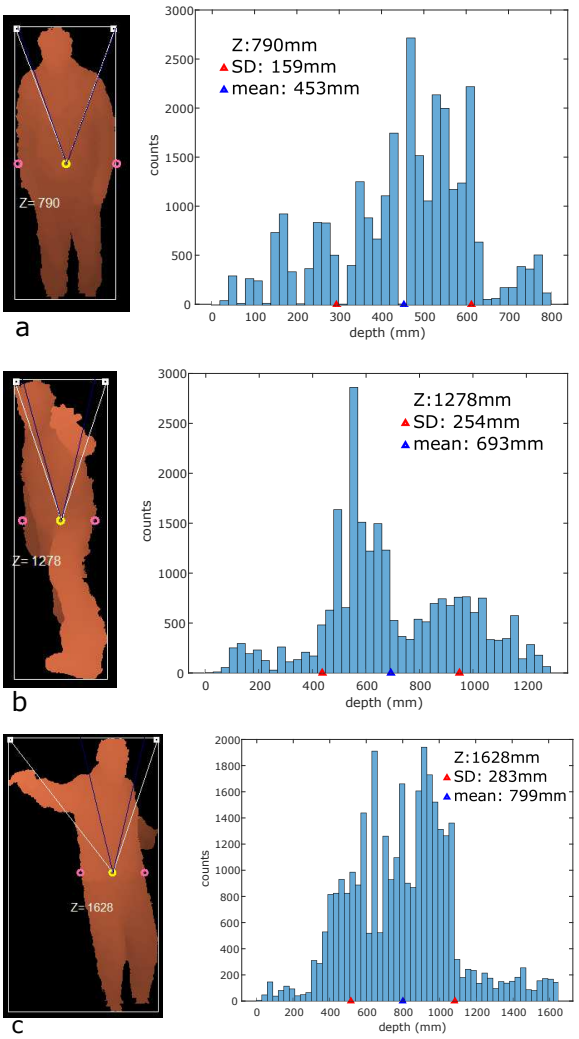

FIGURE 4.7: Applying standard deviation on subject's depth pixels to determine the depth dimension on the conservative bounding box. Z is the depth size of the ordinary 3D bounding box, while the depth of the conservative bounding box equals twice the SD value. Three examples: a)standing, b)lifting arm and leg, c)extend arms to the opposite direction. Graph shows the accumulation of depth pixels

This algorithm relies only on the values of an angle (ϕ) as seen in Fig. 4.5. The 3D representation of the angle in the bounding box is visualised in Fig. 4.6

As discussed, this bounding box relates only to the torso and leg motion and any movement of arms is filtered out. Using this approach, the angle based on the conservative bounding box is a reliable measure of change when a person moves. This is due to the fact that rapid arms motion can alter the bounding box while walking or doing an exercise. Alternatively, if ρ (from Fig. 4.5) was the measuring feature of this approach, false detections, as well as missed detections of fall events, would be expected.

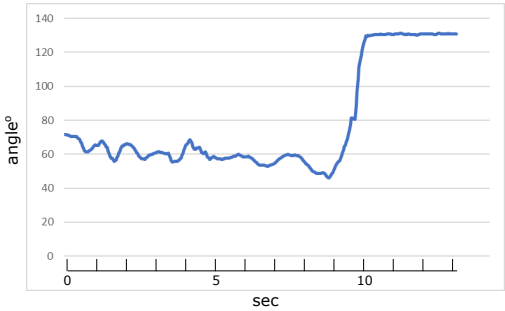

FIGURE 4.8: 3D angle during a collapsing fall.

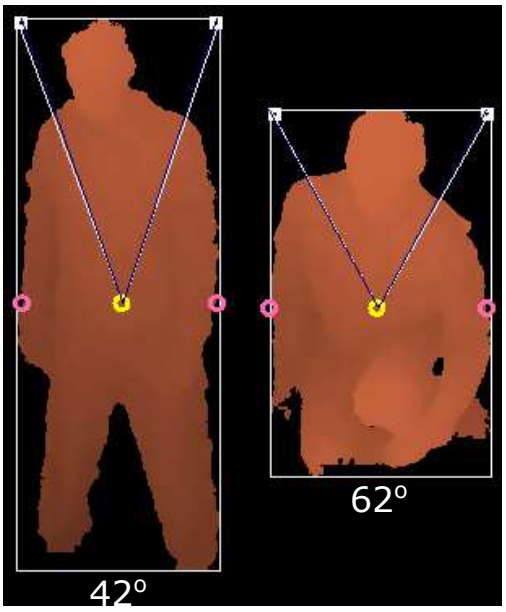

FIGURE 4.9: 3D conservative bounding box angle of a collapsing fall. Notice how the angle increases as the bounding box height reduces

4.3.2.2 Angular velocity

During a fall, the angle ϕ changes its value from acute to obtuse. Nevertheless, this pattern is not enough to determine a fall, since other actions (e.g. lying down) may have the same result. One solution is to measure the angular velocity of ϕ to determine whether the person is falling or not. When this velocity is above a certain threshold during N sequential frames then the fall is initiated. The duration (i.e. number of frames) on which the angular velocity is above the threshold, as well as the angular velocity threshold, are determined by a machine learning algorithm that uses Random Search (see 4.4.1).

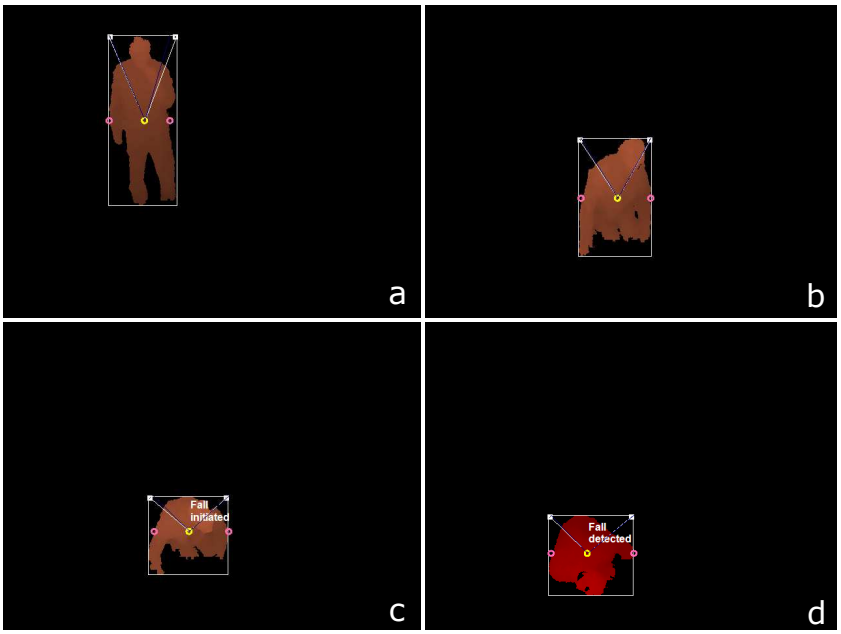

FIGURE 4.10: Collapsing fall detection example using Algorithm 2. (a) detection of bounding box, (b) person collapses and knees hit the floor, (c) angular velocity exceeds threshold while person inclines towards the floor, (d) fall detected when inactivity and angle size conditions are fulfilled

4.3.2.3 Fall detection after inactivity

When the falling person finally reaches the floor, the angular velocity suddenly drops to near zero. The last part of the algorithm will detect the pattern of velocity inactivity. Additionally, the algorithm detects if the angle exceeds 120° (determined by the maximum value the angle has when ADLs of the training set were measured) and flags the detection of fall after a period of time (i.e. 2 sec). Figure 4.10 shows the result of fall detection using the angular velocity.

Algorithm 2 uses OpenCV and OpenNI libraries. Training the algorithm and determination of thresholds follow the same procedure as in Section 4.2 using a Random Search but now using two thresholds $T_{i\dot{\Phi}}$ for setting the angular velocity and the threshold N denoting the number of frames for which this velocity must be maintained in order to decide a fall.

Algorithm 2 Pseudocode of fall detection algorithm using angular velocity

```
SET threshold T_Φ̇
SET threshold T_iΦ̇
SET counter a, b = 0
SET threshold = Nframes
SET threshold Mframes
SET boolean activityDetection = false
SET boolean inactivityDetection = false
while run do
  if (Φ̇ > T_Φ̇) then
    if a = N then
      SET activityDetection = true
    end if
    SET a++
  end if
  if activityDetection = true and |Φ̇| < T_iΦ̇ and φ > 120° then
    if b = M then
      SET inactivityDetection = true
      SET Fall detected
    end if
    SET b++
  end if
end while
```

4.4 Experimental Results and Discussion

4.4.1 Training

Datasets (UR, SDU, GM, GM2) were used for training and test purposes. Three experimental protocols are suggested according to the selection of training and testing sets: i) each dataset is split with a fifth of the samples for training and the rest for testing following a Monte Carlo cross-validation [122] sampled 20 times; ii) train using all samples from one dataset, then test on a different set (this is abbreviated as a $\rightarrow$ between the two datasets); iii) all the datasets are combined and then split as in (i). In some cases (mainly for (ii) and (iii) protocols), large datasets such as SDU or the combined set did not converge during the training procedure. This was resolved using a smaller proportion for training on an sub-sample from the SDU when using protocol ii or use another random training sample (i.e. 20% of the combined set) from the combined set when using protocol (iii). These adjustments are further discussed in Sections 4.4.2, 4.4.3.

For example, the GM dataset was split into a training (12 falls and 22 non-falls) and the testing set consisted of the remaining samples. Collapsing fall videos from the GM2 dataset were used in order to identify the limitations of the rigid fall detection 1 algorithm which was designed for detecting rigid falls as discussed. The remaining datasets UR, SDU were used in the same manner.

The threshold values for velocity $T_{\dot{H}}, T_{\dot{L}}, T_{\dot{\Phi}}$ as well as the duration N of the fall in frames were estimated by performing Random Search on the training dataset multiple (100) times. Separable result values for the thresholds were produced since the fall and non-fall sequences of the training dataset have different velocity and duration values. For example, the values determined after training on the GM dataset using protocol (i) are $t_vH = 1.18m/s, T_vDW = 1.20m/s, N = 8frames$.

The testing set was analysed using the median values of those triplets for Algorithm 1 ($T_{\dot{H}}, T_{\dot{L}}, N$). Figure 4.11 shows the median of the triplets when GM dataset is used in training Algorithm 1. Each triplet is derived by comparing all triplets during the random search optimisation and selecting the one which maximises the accuracy of detection in the training dataset and returns the best match values of velocities and durarion. The velocities derived from the training confirm the values obtained from [62] where fall-related velocities are above 1 m/s. For Algorithm 2, the same search algorithm was used and doublets were found ($T_{\dot{\Phi}}, N$) as reliable estimates of the method parameters.

4.4.2 Algorithm 1 Results

Numerical results are shown in Table 4.1 where accuracy, precision, sensitivity and specificity were calculated. The first four rows of the table shows results from protocol (i). The next six rows show results of the protocol (ii) and the last row from (iii). The algorithm performs better when the test set is a subset of a single dataset as in evaluation scenario (i) when compared with the results of the other protocols, due to the fact that actions of the same subject are included in both training and testing sets. The protocols (ii, iii) when tested do not provide the same results, as the falling and ADL activities are different since they are performed by different people, e.g. picking up an object or sitting down in GM and SDU datasets appears in several cases to be done quite fast, while the UR dataset has several hesitated falls.

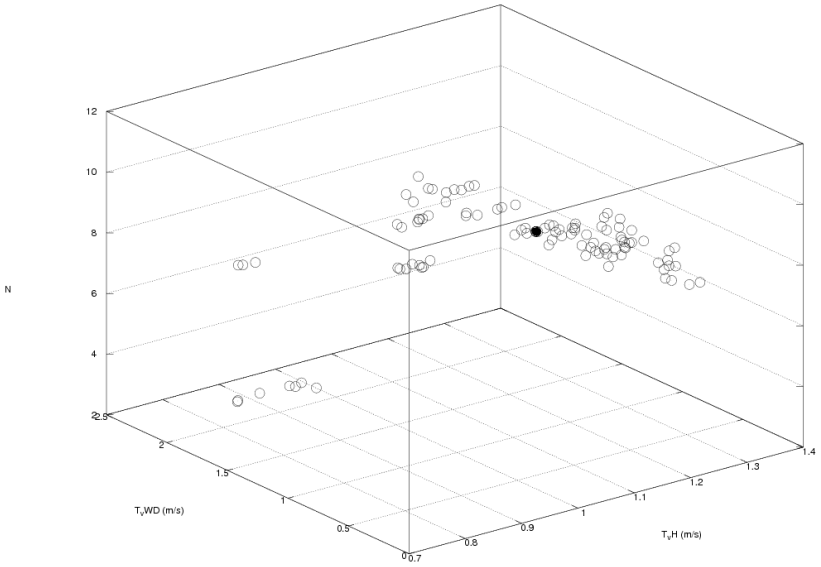

FIGURE 4.11: Circles indicate 100 triplets estimated by random search for training the rigid fall algorithm. Their median ($t_vH = 1.18m/s, T_vDW = 1.20m/s, N = 8frames$) is marked as a bold circle and is used for the experiments

The SDU dataset is quite large and in some of the examples subjects are marching i.e. walking with abrupt arm movements, which the algorithm cannot filter out using the Kalman filter. This is the main reason for the solution not converging as noted in the Table even though this particular set was randomly selected by 80% of its original size (i.e. SDU→UR, SDU→GM).

The same issue was observed when training on the combined set, where the 20% of the sample was randomly selected over 20 times, where convergence succeeded in only 6 of these trials. This particular experiment shows the complication of using a parameter set for the assessment of a large dataset where human subjects have different physical characteristics and behaviour patterns. A solution to this issue is discussed in the following two Chapters where a personalised approach is proposed.

Another noticeable result is when a small dataset (UR) is used to train the algorithm while testing occurs on a large dataset (SDU) where samples have different variability in their physical appearance, shows a drop in performance (e.g. GM→SDU, UR→SDU). The impact of human variability is discussed further in Chapter 5 and 6.

A set of visualisations are shown in Appendix A to demonstrate the variety of several experiments: Forward fall – 45 view (Figure A.1), sideways fall – front

TABLE 4.1: Performance of Algorithm 1. A→B: Training on A, test on B datasets, N/C: Not Converges, † Converges after re-sampling of training set

Dataset	Accu (%)	Prec (%)	Sens (%)	Spec (%)
GM	100.00	100.00	100.00	100.00
GM2	92.50	81.25	94.54	91.72
UR	94.54	80.00	100.00	93.03
SDU	74.40	66.50	41.30	90.11
GM → UR	92.72	73.33	100.00	90.90
GM → SDU	57.00	37.00	19.57	79.74
UR → GM	81.52	87.50	60.00	94.73
UR → SDU	78.80	56.50	47.47	88.58
SDU → GM	N/C	N/C	N/C	N/C
SDU → UR	N/C	N/C	N/C	N/C
Combined †	78.80	59.49	52.03	87.92

view, Figure A.7) lying on the floor – front view (Figure A.2), sitting on a sofa – side view (Figure A.3), picking up an item from the floor – side view (Figure A.4). Another set of experiments includes more specific actions, such as sweeping (Figure A.5) and vigorously sitting (Figure A.6). Sweeping changes the 3D bounding box mostly in Z and X although the velocity fails to reach any of the thresholds $(T_{\dot{H}}, T_{\dot{L}})$. Sitting vigorously is a case where the motion is not long enough in time to be detected as a fall, as the subject's motion is halted when sitting on the sofa. Therefore, no fall is detected in either actions.

Finally, the algorithm was tested against additional non-fall scenarios to see how it behaves when the subject is lifting an object and then placing it back on the floor or on a table. For those experiments, 24 additional videos were captured from three subjects in actions such as lifting a chair and placing it back, lifting and rotating a chair and similarly placing it back, lifting a box and either placing it on the floor or on a table and then moving away. During these experiments, although the bounding box may increase or decrease in width and/or depth, no significant change in the height dimension is observed. Therefore, although the $\dot{L}$ velocity may be increased the $\dot{H}$ remains at normal levels, hence, no fall detection is initiated. A large box is used in order to investigate how the method performs in those scenarios since the box would dramatically change the size of the 3D bounding box when lifted and carried. Figures A.8, A.9 show two of the set of images from this experiment.

The algorithm was proved stable, even when half of the subject's body was occluded by the box. This is because the $\dot{L}$ remains at normal levels (i.e. well

below the $T_{\dot{L}}$) while the $\dot{H}$ exceeds the $T_{\dot{H}}$. Therefore, a fall detection will not be initiated since both $\dot{L}, \dot{H}$ must be above their thresholds.

The bounding box as seen in Figures A.8, A.9 will split into two different bounding boxes (one for the subject and one for the object) when the user places the object on the floor/table. This is caused by the fact that the current OpenNI version initialises separate bounding boxes using a motion detector. The system is still able to track the subject and if a fall occurs, it will raise an alarm. However, if for any reason the object (i.e. box) drops, this may also be detected as a fall.

4.4.3 Algorithm 2 Results

The same evaluation scenarios were used to test Algorithm 2 and the results follow those of Algorithm 1 as seen in Table 4.2. As seen from the Table, the algorithm is capable of operating (i.e. the training algorithm converges) in all scenarios meaning that the angular velocity feature derived from the conservative bounding box performs better than the 3D bounding box velocities. Notice how small sized datasets perform better. SDU dataset when used for training only performs and converges due to use of the conservative bounding box. Also, the combined set failed to converge without split into a smaller set (i.e. 80% of the dataset). Further results are better justified and compared them with Algorithm 1, in the next subsection.

4.4.4 Algorithm 1 vs Algorithm 2

The main difference is the robustness of this algorithm on the GM2 dataset where collapsing events were tested, as well as when tested against the SDU dataset. This dataset is a challenging one, as the subjects enter the scene marching; the arms motion is captured by the ordinary bounding box of Algorithm 1, but is filtered out by the conservative bounding box of Algorithm 2. Comparison against the GM2 dataset is improved against Algorithm 1 due to the use of the new algorithm which captures the vertical motion of the bounding box as the person drops. The other noticeable benefit of Algorithm 2 is that training converges on SDU dataset when 80% of the sample is used. This was achieved 60% of the time (i.e. converges 12 times out of 20 for a randomly selected sample). The same percentage is observed when all datasets are combined, showing again the issues of having one threshold

Table 4.2: Performance of Algorithm 2. A→B: Training on A, test on B datasets, † Converges after re-sampling of training set

Dataset	Accu (%)	Prec (%)	Sens (%)	Spec (%)
GM	97.82	97.91	94.00	99.25
GM2	97.00	95.31	95.31	97.79
UR	96.36	86.66	100.00	95.23
SDU	89.50	82.38	71.48	95.18
GM → UR	96.36	86.66	100.00	95.23
GM → SDU	73.80	63.50	40.18	89.32
UR → GM	87.50	93.75	69.23	97.47
UR → SDU	88.30	84.50	66.27	95.83
SDU → GM	80.43	81.25	59.09	92.37
SDU → UR	89.09	73.33	84.61	90.47
Combined †	87.41	81.36	68.16	94.36

(i.e. one-fits-all solution) for every person's activities as discussed on the results of Algorithm 1.

4.5 Conclusion

Two fall detection systems were developed that require no pre-knowledge of the scene. The first focused on the rigid (forward, backward and sideways) and the second on the collapsing fall type. The fall event is analysed in isolation as an independent activity without specifying or detecting any external parameter set such as the floor plane coordinates. These simple and lightweight algorithms run in real-time with negligible computational time (0.3–0.4msec) with the Kinect's GPU doing most of the heavy computation and are capable of detecting falls with a variety of accuracy according to each evaluation scenario. It is proven to be robust on cases such as sitting vigorously on a chair, lying on the floor or crouching down (i.e. fast action).

This Chapter proposed two machine learning approaches, one based on the analysis of the 3D bounding box's velocities, while the other is based on an angle from the CoM to the two upper corners of the 3D bounding box. These algorithms perform in real-time as they were developed within the OpenNI architecture. It is shown from the results in Tables 4.1 4.2, that Algorithm 2 performs better in overall comparison against Algorithm 1 which is expected as the first algorithm

was developed without talking into account the collapsing falls, or filter out a rapid arms motion (e.g. marching).

Taking the above into account, the algorithms can be characterised as reduced complexity that requires three or two parameters to operate; the width, height and depth or the 3D bounding box angle of the subject. Nevertheless, although results are promising, they are not sufficiently robust when applied to all of the fall datasets. These algorithms were tuned by human fall data i.e. they are data-driven, which as discussed in Chapter 3 are non-representative. Also, difficulty in convergence was found when training on a dataset where the samples have different physical characteristics and moving behaviour from the testing one. Also, the same issue was encountered when using a combined set of all data, where training did not always converge. This is due to the fact that algorithms using a set of parameters for all data (i.e. of people with different physical characteristics) cannot address the variability within the sample and personalise the detection algorithm. The next chapter will try to resolve the issue of using human fall data with the use of physics-based myoskeletal simulations where algorithms will either train on such simulation data or use the simulation to model falls. These new approaches discussed in Chapters 5 and 6 aim to deliver a customisable/personalised fall detector.

Chapter 5

Simulation: Modelling Fall

5.1 Introduction

Existing computer vision fall detection systems are either ad-hoc or learning based and tend to ignore the physical characteristics that contribute to falls. In either case, the algorithms are dataset-driven and detect the fall events which are contained in the fall event datasets. Several important issues were discussed in Chapter 3 regarding these datasets. The realism is a major issue as acted falls are performed by young and healthy people, from a narrow demographic population. Acted falls are limited in number and type due to risks of injury, while hesitation is another factor which contributes to unrealistic falling behaviour.

In addition, such data will be unrepresentative of a vulnerable population of those with health-related issues such as the elderly and the infirm. The physical characteristics of elders, associated with posture, gait, height and weight are different from the samples in the current public fall datasets. Actors from these groups are not part of any dataset for fall detection since there are ethical issues in performing such risky experiments. One can say that data from human-simulated acts like these will never happen and only long-term recordings from hospitals and care homes would fill the gap of data collection. In contrast, young actors are typically used to simulated the fall events.

This is mainly discussed in this Chapter where alternative means are investigated in order to model fall events without the participation of human subjects. The use of simple and complex physics-based approaches will be discussed where the

fall event is modelled using falling rods and myoskeletal simulations. The latter approach can customise its model to fit the person's physical characteristics such as the height. Using models, no human intervention is required, hence, no risks or hesitation in performing, while the models can imitate a close similarity of characteristics with an elder, such as an arched back posture. Therefore, this framework will provide realistic and customisable fall events that match the individual's build.

A fall may be associated with a wide variety of physiological conditions such as low blood pressure, brain ageing [25] and brain atrophy [91] or the consequence of a walking accident such as tripping, slipping or stumbling [24]. In [123] authors discuss the severity of injury where 4.9% of the falls concluded without any serious injury, but 19.9% with serious ones. Somehow, data which are associated with virtually any of these problems do not exist since their collection is extremely difficult. Asking actors to slip or trip would require a scenario of asking participants to walk blindfolded or in a dark room in order to properly imitate the real behaviour. But how risky or even inappropriate this seems. Nevertheless, fainting is one of the few cases where real data can be found via YouTube as have discussed in Chapter 3 where hyperventilation caused fainting on purpose.

Other influences could change the direction or type of a fall which may depend on the walking direction and/or the incline of the person's centre of mass (CoM). In some cases, persons may fall rigidly, whilst in others they collapse vertically. The age of the faller is also a factor which contributes to the kinematics of the fall. The gender could also be a factor since males have a higher CoM than females [124]. The health of the person may impact the falling event such as a broken arm or a leg which temporarily unbalances the natural human movement, or if the person is carrying an object. The above characteristics can be implemented as simulation parameters for producing fall events.

5.2 Review

Only a few studies discuss the use of a physics-based simulation to track human motion. This is due to the fact that current research is based mainly on actual human data (i.e. data driven) and to model a human action mathematically can be trivial especially when such actions are derived from the articulated body.

5.2.1 Physics simulation - Synthetic approaches

With the rise of machine learning, the requirement of sufficiently large and variable datasets has become an issue, as such datasets may be laborious and expensive to acquire and label. One of the issues in fall detection datasets is that we cannot acquire such data from real events or that such data are not released due to privacy protection. One approach to deal with this problem is to generate synthetic data based on a combination of actual observations and physical models. Whilst such simulation has been used by other researchers, the work reported in this thesis is the first to apply it to fall detection.

A number of studies employing computer vision and physics-based modelling exist in the literature. The most relevant studies [125, 126] discuss how tracking a walking person can be achieved with the use of a bipedal model based on physics simulation. Brubaker [126] discuss the use of a simple model for predicting the walking behaviour of a person. The authors evaluate their approach for varied walking speed and with occlusion, but also discuss the limitations of this approach and how a more complex model incorporating myoskeletal capabilities would provide a more accurate representation of human motion. Other studies describe and propose physics-based frameworks for tracking articulated objects. In [127] Lagrange equations of motion are used for models which can synthesise physically correct behaviours in response to applied forces and imposed constraints. Based on a previous study, the work in [128] presents a mathematical formulation and implementation of a system capable of accurate general human motion modelling. The work in [129] uses an off-the-shelf physics simulator to track the behaviour of a rigid object. Another framework is presented in [130], where a method estimates human motion from monocular video. This is done by reconstructing three-dimensional controllers (models) from the video which are capable of implicitly simulating the observed human behaviour. This behaviour is then replayed in other environments and under physical perturbations. Synthetic human data for activity monitoring are presented in [131]. A dataset incorporating rigid poses is produced and used for the purpose of human behaviour recognition as well as scene understanding. Out of context of computer vision related studies, the work in [132] discusses the use of a physics-based simulation engine capable of detecting the stability and falling likelihood of a rigid object.

Recent developments in deep learning [133] have increased the need for larger datasets. An example of synthetic data for action recognition can be found in the SOURREAL dataset presented in [134], consisting of 6 million image frames together with ground truth pose, depth maps, and segmentation masks. The amount of data is achieved by adding people images of variable size as a foreground over a variety of background images. Other examples include synthetic datasets for pedestrian detection [135] and synthetic urban scenes from the SYNTHIA dataset [136].

Previous attempts using simulation/synthetic data show an active pathway in terms of creating data where real data is not available. The synthetic approaches require less time in preparation as they do not incur human interaction in terms of performing actions or scenarios. In terms of fall detection, a simulation-based approach would resolve the data availability problem in terms of fall realism and human risks. That is not only filling the gap of lack of such fall data but also simulate the fall with a model which is personalised using an individual's characteristics, such as their height. The main difference with this study is in terms of how synthetic data on one fall model are used, rather than the many examples of acted falls needed for training purposes.

5.3 Modelling falls

This section explores different types of models and modelling approaches from mechanics to biomechanics simulation for the simulation of fall events. Initially, the fall is modelled with a rigid rod which is a simplified model that may be appropriate for rigid falls. The reasoning of using a simulation to model a fall as discussed in earlier is the fact that current studies use human subjects acting fall events which suggest an improbable falling behaviour due to risk factors and hesitation of performing such acts. The question then is whether we can use simulations to model fall events and avoid the use of human fall data.

This section presents two classes of simulation-based approaches that imitate a falling person, one inspired by mechanics, based on a falling rod and one inspired by biomechanics, based on myoskeletal modelling. An evaluation of those simulation examples is discussed in Section 5.4.

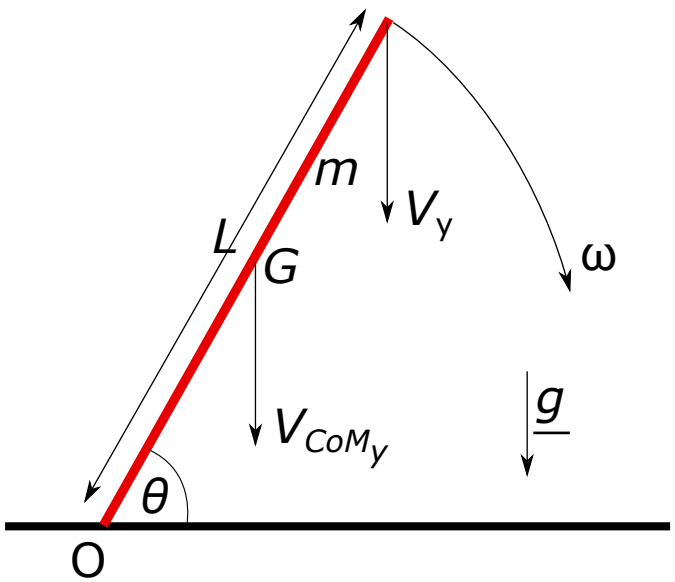

FIGURE 5.1: Falling rod, of length L with uniform mass m end-point vertical velocity V_y, CoM vertical velocity V_{CoM_y} and ω

5.3.1 Falling rod simulations

This section investigates the use of a simple rigid falling rod to simulate the motion of the person experiencing a rigid fall. Several rod simulation types will be discussed: a rigid rod falling at an angle without any resistance (5.3.1.1); a similar rod with applied resistance imitating the feet balancing force (B.1); a two piece rod, imitating the upper and lower body (B.2). These rod models were developed in order to achieve an approximation of falling velocity which is further discussed in Section 5.4. Nevertheless, only the rigid rod simulation is further discussed and the other two methods are presented without further evaluation as their application required further and complex justification.

5.3.1.1 Rigid Falling Rod

A rough approximation of the motion of a person falling with a rigid motion is given by modelling a rod of length L with uniform mass distribution falling from a vertical position, as seen in Fig. 5.1. The following formulas show the angular velocity as recursively defined by ω in Eq. 5.1, the velocity of the centre of mass $Vcom_y$ in Eq. 5.2 and the end-point V_y in Eq. 5.3 of the rod.

$$\omega_{n+1} = \sqrt{\omega_n^2 + \frac{3g(\sin(\theta_n) - \sin(\theta_{n+1}))}{L}} \tag{5.1}$$

$$V_y = L\omega\cos(\theta) \tag{5.2}$$

$$Vcom_y = \frac{L}{2}\omega\cos(\theta) \tag{5.3}$$

where, θ is the orientation of the rod, n is the number of steps and g the gravitational acceleration ($9.81m/sec^2$). Equation 5.1 is derived by solving the Kinetic and Potential energy formulas under the assumption of energy conservation.

Derivation of 5.1 To begin with, we are going to establish the equation of the falling rod. We suppose that the rod is an isolated system, so we can use the principle of energy conservation. The rod is subjected to the weight $\vec{P}$ and a support reaction $\vec{R}$. The following parameters characterise the model:

Ek is the kinetic energy
Ep is the potential energy
α is the angular position
m is the mass of the rod
g is the gravitational acceleration
ω is the angular velocity
$\vec{OG}$ is the position vector of the centre of the mass
y_G is the coordinate of the centre of mass
c is a constant of integration, (c = 0 in practice).

We have:

$$\begin{cases} dEp = -\vec{P}d\vec{OG} \\ Ek = \frac{1}{6}mL^2\omega^2 \end{cases} \tag{5.4}$$

$$\begin{cases} Ep = mgy_G + c = mg\frac{L}{2}sin(\alpha) + c \\ Ek = \frac{1}{6}mL^2\omega^2 \end{cases} \tag{5.5}$$

We have conservation of mechanical energy (E_m), therefore:

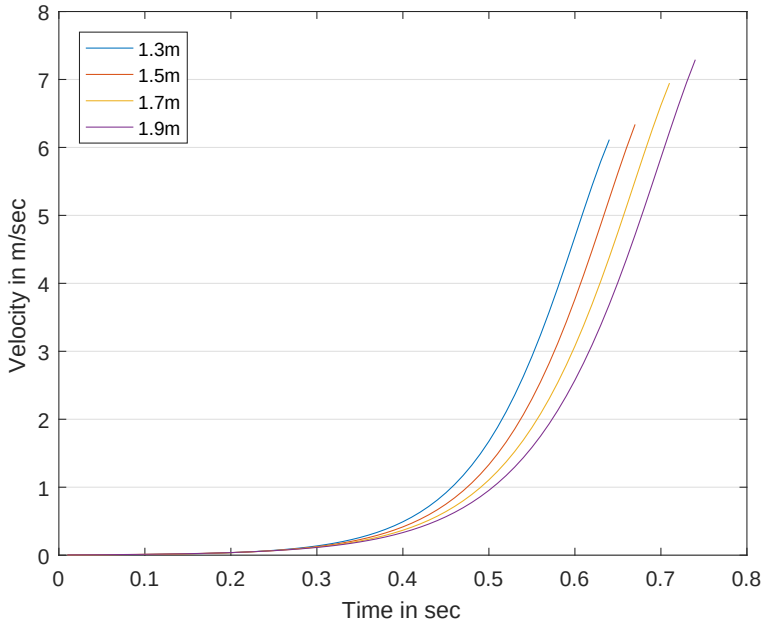

FIGURE 5.2: Typical rod model velocities. The final velocity is proportional to the length of the rod. This velocity is measured from the top point until the rod reaches the horizontal position

$$
\begin{aligned}
&Em_{(n+1)} = Em_n \\
&Ek_{(n+1)} + Ep_{(n+1)} = Ek_n + Ep_n \\
&\tfrac{1}{6}mL^2(\omega^2_{(n+1)} - \omega^2_n) = mg\tfrac{L}{2}(sin(\alpha_n) - sin(\alpha_{(n+1)})) \\
&\boxed{\omega_{n+1} = \sqrt{\omega_n^2 + \frac{3g(\sin(\theta_n) - \sin(\theta_{n+1}))}{L}}}
\end{aligned}
\tag{5.6}
$$

The topmost end of the falling rod represents the location of the head while the middle-point is the centre of mass (CoM). Fig. 5.2 shows velocity profiles for rods of 4 different lengths corresponding to a variety of height ranges of an adult (1.3-1.9m), indicating that the velocity profile is increased proportionally to the length of the rod. That is the taller the person the higher the final velocity and the longer time to fall. The same obviously happens for both CoM and the free end of the rod.

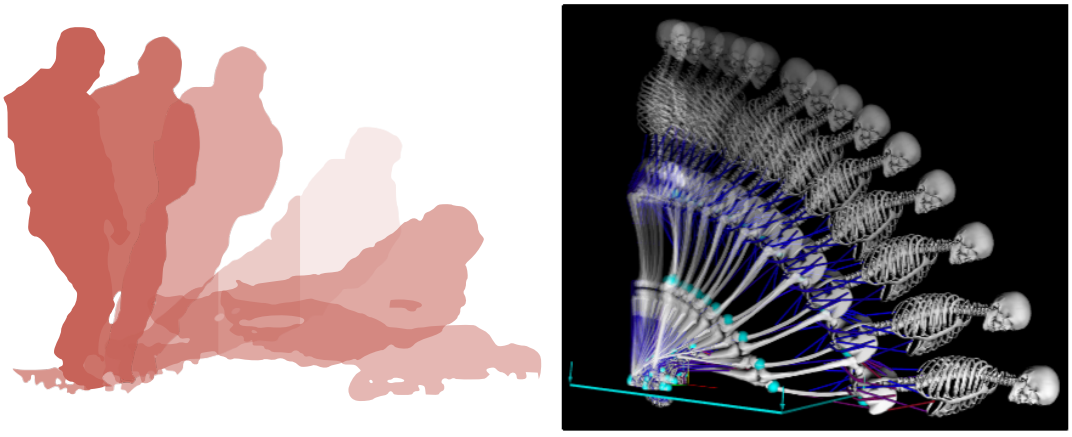

FIGURE 5.3: Sequence of an actual fall event as captured by a depth camera and of a fall simulated by OpenSim

5.3.2 Myoskeletal human model simulation

A more sophisticated model of the human body for the purpose of fall modelling is derived by OpenSim [137], an open-source simulation myoskeletal software initiated by Stanford University. Note that OpenSim is not a simulation software for falls only, but for experimenting with various motion patterns with a use of different myskeletal models. A number of different pre-defined myoskeletal models are available for OpenSim. Figure 5.3 shows the simulated model to be comparable to the falling behaviour of a person.

Biomechanical studies have developed several applications to simulate the human motion during activities such as walking, sitting, jumping etc, in an attempt to understand further the capabilities and limitations of the human body. Their aim is to study human motion and how this can be reproduced as a simulation in order to perform measurements of how different body parts behave during an action. OpenSim provides a detailed myoskeletal simulation model of the dynamics of the human body. It is based on the samples of 21 cadaver and 24 young subject's MRI samples for their musculotendon parameter derivation. The differences between those muscle-generated and inverse dynamics joint moments of the derived models were shown to be within 3% (RMSE) of the peak inverse dynamics joint moments in both walking and running [138], therefore the model is considered suitable for generating muscle-driven simulations of healthy gait. Later, in Section 5.4, several experiments are conducted to validate our assumption that a real fall event has

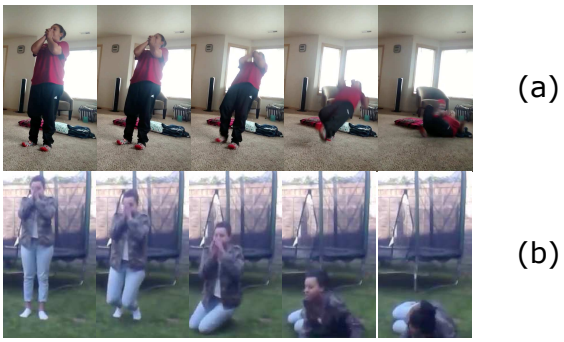

FIGURE 5.4: Images sampled from a YouTube video of a) person acting a rigid fall and b) person suffering a collapsing fall following hyperventilation which includes a faint

similar velocity patterns (e.g. head vertical velocity) with a fall simulated by a myoskeletal model.

5.3.2.1 Fall simulations

The motion of the body varies when the person is falling. Two types of falls are observed in real-life which start from a standing position: rigid and collapsing falls depicted in Fig. 5.4. Using myoskeletal simulation, it was possible to simulate these falling behaviours with accuracy. Each fall type is different and can be parametrised by adjusting parameters such as the model's body inclination or the location of the centre of mass.

Simulated model preparation In order to prepare the model simulation for fall events, the myoskeletal model, as well as the contact area in OpenSim, are constrained by the following conditions:

- The subject/person should be in a static and standing (not walking, not seated etc.) position when the fall occurs
- Objects on the floor, uneven/slippery floor will not cause the person to fall, i.e. slipping, tripping, stumbling are not part of this particular part of the study.
- Simulated model falls only due to gravity - there are not any external dynamic forces.
- Model inclination is responsible for the side of fall in rigid fall types

FIGURE 5.5: Feet position of simulated model before fall

- The torso's centre of mass is the parameter to cause collapsing falls
- The feet (see Fig. 5.5) of the simulated model should be parallel as in (a). Using examples as in (b) may change the falling direction of the model during the simulation. Nevertheless, this does not have any complication towards the validity of the approach as discussed in Section 6.8.2 where the direction (i.e. different fall types) of the fall is validated.

Rigid fall

To perform a fall, the model stands stationary on a platform that has a small inclination towards either the front, back or side. This inclination will trigger the fall event as no other parameter has changed on the model, such as muscle tension or the centre of the torso's mass. The gravitational force will pull the model towards the ground, as the only applied force on the myoskeletal model. The behaviour of the model is represented as a rigid fall as seen from the examples in Fig. 5.7.

As we will discuss later in Chapter 6, section 6.7.2.1, the top bounding box point will be used as a feature for the proposed fall detection method. Here, the simulation model and engine are capable of introducing a number of markers from where measurements can be taken such as distances from the ground platform, velocity and acceleration. A marker on the top of the head will be used for this purpose (blue sphere) as seen in Fig. 5.6.

Collapsing fall This type of fall is subjected to the person falling vertically towards the ground while their knees at first then the torso bends. Via simulations, it is found that altering the placement of the centre of mass of the torso plays a significant role in this falling behaviour. Figure 5.8 shows the impact that mass distribution has on how the body falls. The parameter for initiating the fall is the

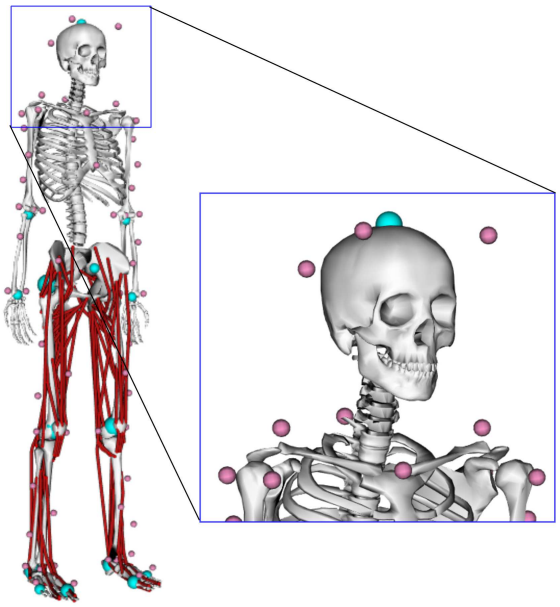

FIGURE 5.6: The Full Body Model given by OpenSim engine. Blue marker denotes the head location point, while pink markers denote the MoCap relevant markers

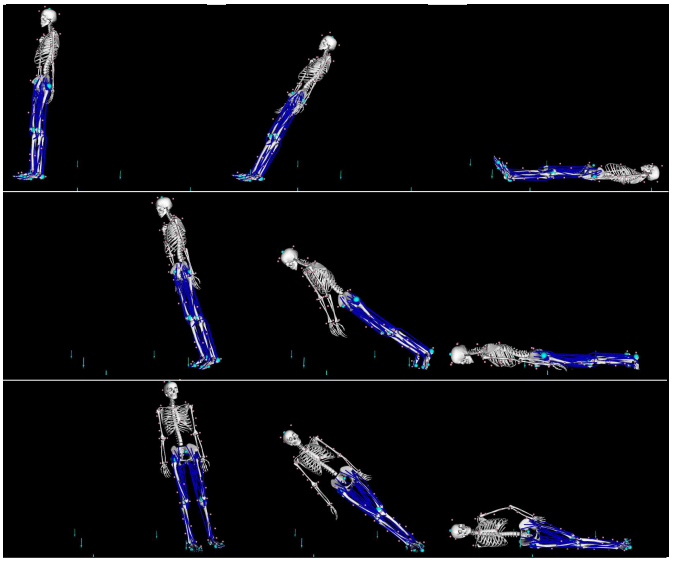

FIGURE 5.7: Three types of rigid fall, backward (top), forward (middle) and sideways (bottom) as simulated on OpenSim

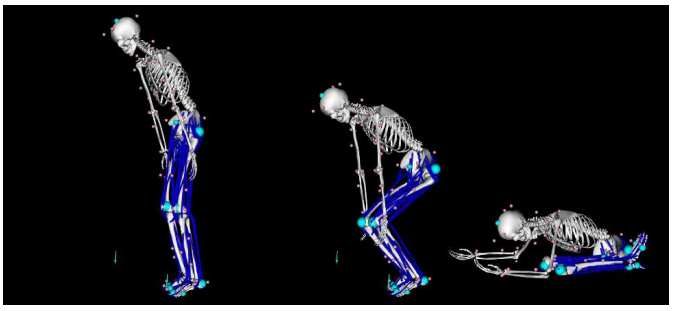

FIGURE 5.8: Collapsing fall as simulated by OpenSim

weight distribution of the torso's CoM which is altered towards the front of the body to initiate the fall. This example in the Figure shows how the model behaves when this parameter is adjusted. Other parameters such as muscle tension were left intact to their original states as defined by the Full Body Model and processed by forward dynamics routine of OpenSim. The posture of the model as seen from the previous Figure shows how torso's CoM contributes to incline the body towards the front, producing a collapsing fall. One type of collapsing fall (e.g. lean forward) is observed in the sparse data selected for the evaluation (i.e. GM2 dataset) of this type of fall, and as a result only this type was modelled by the simulation.

5.3.2.2 ADL simulation

Apart from creating fall events for modelling, other events are required in order to distinguish the two classes. The focus is particularly given to the events which have a similar motion to the fall such as lying down. The discussion about the validity of ADLs was presented in Chapter 3 (3.11).

A number of activities of daily life (ADLs) such as sitting down, lying down, picking up an object, walking quickly, turning around quickly and raising hands are generated using motion capture (MoCap) data from a Motion Capture Database [139]. Such data is processed as seen in [140] in order to model the dynamics of the body e.g. how particular body parts move when a person sits. Non-fall data are required in order to capture the class of actions which have different motion. More interestingly are the fall-like actions such as lying down, aggressively sitting on a sofa, picking up an object. We would expect fall detection algorithms to confuse these events as falls resulting in false positives i.e. non-fall actions which are classified as falls. Also, collecting data of ADL events is much easier and

realistic as there is no hesitation or risk in performing them. Simulated ADLs will provide a negative class of samples of human motion which is required for the algorithm to differentiate from the fall events. It is much easier to use MoCap data in order to translate the motion into a myoskeletal model, than justify the model itself to do so. This is required by our method in order to collect the V_y profiles of such non-fall events. The captured biomechanical data is converted into OpenSim [141], to align MoCap with model markers and allow transfer of the articulate motion from the actors who performed MoCap to any human subject that can be parameterised within the OpenSim model [142]. The benefit of this conversion is to derive simulated models that have the motion of ADL activities while allowing the physical characteristics such as the height of the model to be adjusted separately.

5.3.2.3 Scaling the model

The human body variation discussed previously plays a significant factor in the falling behaviour. Body physical characteristics such as the weight, height, posture alters the CoM of the body. Elders may have an arched back, or be overweight due to the lack of movement. Other variations include different CoM location for men, women, especially during pregnancy, or due to disabilities related to the lower limbs etc. These parameters can be taken from the actual subjects and implemented as parameters for the myoskeletal model. To prove the concept of this approach, the height is selected (as already seen with the falling rod) to act as the customisable parameter.

As seen on the simple rod model, the final fall velocity depends on the height and similarly, the same parameter is responsible for the velocity of the myoskeletal model. The myoskeletal model will be scaled to represent human height variability. This is selected in order to investigate the impact on fall modelling and also is a feature which is easily measured from depth data. Scaling is performed proportionally to all body parts to maintain their ratios to height. A set of such models will be created to perform the fall and non-fall simulations discussed previously. Each scaled model will approximate the height of an actual person.

Studies suggest [143, 144] that body height declines with increasing age hence elders are expected to have a shorter height than the average suggested by [145] given a standard deviation of approximately 7cm in both females and males [146].

The range of height model variations is chosen to provide variation (i.e. 130 – 190cm), given the minimum and maximum average heights found in the [145]. Furthermore, the model sampling (i.e. four models 130, 150, 170, 190cm) was selected as no further improvement in robustness was observed in considering fine gradations of the height model. A single simulated model provides the necessary data for the algorithm described in the next Chapter. It is not required to simulate different models as proposed in the next section.

5.4 Evaluation of the fall simulation models

The fall velocity profiles derived by physics-based simulation are compared with genuine profiles. Specifically, 20 different fall events from YouTube videos are selected where actors faint after hyperventilation [147] as seen in Chapter 3. Those videos are the closest representations of fainting where actors fall rigidly unconscious to the ground and are a genuine source of falls. The videos were processed using [148] for calibration and vertical velocity (V_y) measurement. The process of calibration fits a mesh to the ground plane and the user selects landmarks (i.e. ball, brick, fence, lamp post etc.) near the ground on at least three remote points on the image. A KLtracker is used to track the head and with the use of tracking points, the algorithm calculates its velocity.

An evaluation is conducted to measure the similarity of profiles of both the OpenSim model and falling rod, where, Hausdorff distances are measured against V_y of actual Youtube fall events. The usage and evaluation of the Hausdorff distance is discussed in 6.7.1 and 6.8.1. The average HD of the actual falls, when compared with an OpenSim model performing a forward fall, were 0.365m/s and 1.944m/s when compared with the falling rod. The reason why a forward fall was selected is discussed in 6.8.2. Also, the standard deviation was 0.078m/s when compared with the OpenSim model and 0.782m/s when compared with the rod model. These measurements show that a complex myoskeletal mode (OpenSim) provides more realistic simulated rigid falls more than the rod.

5.5 Conclusion

This Chapter has discussed a new methodology of using myoskeletal simulation for modelling falls and ADLs for the purpose of fall detection.

Different simulation models were evaluated, from a falling rod to a myoskeletal complex model. The human body is significantly different from the simple falling rod model, due to its articulation and muscular reflexes, whilst the rod model is a completely rigid object. A more complex model such as one derived by a myoskeletal simulator provides a more accurate representation of the human body.

Both rod and myoskeletal fall models were evaluated against genuine fall data from YouTube to prove their validity. Given these experiments, myoskeletal model simulation is feasible for the use of fall detection discussed in the next Chapter.

Chapter 6

Fall detection based on myoskeletal simulation

6.1 Introduction

This Chapter discusses the use of myoskeletal simulation as described in Chapter 5, applied for fall detection. With the use of simulation, the new algorithms described here try to overcome the issues of data scarcity of human fall data. The machine learning approaches used in Chapter 4 require a significant amount of data for training. The simulation tries to overcome the issue of data-driven approaches by modelling the fall events which are then used by a detection algorithm. Another issue of current algorithms is the lack of personalisation, that is, a classifier able to deal with different people falling irrespective of their physical characteristics or the type of fall.

Many studies on fall detection have been published in recent years, driven by the need for monitoring vulnerable independent livers and detecting accidents. Further impetus comes from the availability of cheap and easy-to-use depth cameras and other mobile sensors. Two broadly accepted approaches for detecting falls are summarised in recent review studies [72, 149]: i) ad-hoc methods based on empirical observations and ii) pattern recognition methods that are trained using machine learning (ML). Both approaches require pre-recorded training data of falls that are normally staged and performed by volunteers or actors to tune their performances for fall detection. Nevertheless, human subjects may hesitate

to perform a fall and also the acting of a fall might be directed in such a way that is not realistic, or similar to an actual fall event, e.g. fainting [28]. However, the quantity and availability of fall event data is low compared to other tasks of action/event recognition.

A common approach to detect fall events is usually performed by a single model which ignores physical body characteristics (such as a person's height) and as a result the dynamics of the fall may differ accordingly. Existing fall datasets (as discussed in Chapter 3) are based on a small number of human subjects with limited body variability in sex, age, height and weight distribution. In a trainable algorithm, the requirement is to have a dataset that is large enough to capture natural variations of individual characteristics, not only to cover the data requirements of the machine learning algorithm but also to properly cover a range of people's physical characteristics and fall types. All data samples of fall and ADL events are tested via the same procedure that has been trained using a small set of data from human subjects of limited body variability (e.g. height). Hence, algorithms trained on limited datasets have questionable performance when applied to the wider population. A physics-based myoskeletal simulation (as discussed in Chapter 5) provides the opportunity for customising the activity based on the body characteristics and the environment in which the fall occurs.

One solution to address the lack of data is to use an approach that is customised to a person's physical characteristics. [150] use accelerometers to make a personalised fall detector recording the acceleration patterns of ADLs during a calibration. An anomaly detection algorithm is then used to identify falls. However, this approach determines its detection decisions based on human subjects with small differences in their physical characteristics (e.g. an 8cm height variation) raising doubts about performance if the differences were larger.

Three novel approaches are discussed in this Chapter which promote the use of simulation to address the issue of the scarcity and quality of training data to improve fall detection algorithms by customising fall events using myoskeletal simulations and for the purpose of personalisation. One of the proposed methodologies extracts a person's height and pre-fall body orientation from depth cameras to simulate customisable falls (with height and orientation as parameters). The derived data from the simulations are then used as training examples or models of falling behaviour. The second approach is capable of using data from simulating

falls and ADLs customised by the model's height and performs the detection procedure without a machine learning technique. The evaluation will use depth and YouTube data. A third approach uses a different feature to apply the myoskeletal simulation fall detection approach on accelerometer data from a wearable device.

Since the only source of video recordings of falls are based on acted falls, this data are used to evaluate the performance of the detector based on simulation. Experiments are presented based on three methodologies: (i) a hybrid approach, which describes a simple methodology using simulation data and acted data (ii) a fully simulation-based on velocity measurements method and (iii) an impact based fall detection using myoskeletal simulation.

6.2 Fall detection using a Hybrid approach

The hybrid approach utilises the height and the pre-fall orientation of a human body derived by a depth camera. The height is measured from the 3D bounding box and the orientation is estimated using a procedure based on data from skeleton estimation by OpenNI. These two parameters are used to simulate three fall events (i.e. forward, backward, sideways) using OpenSim. The measurements from the falling models are taken from the CoM, where velocity profiles are calculated and then processed by a polynomial regression algorithm. To train the algorithm, acted-falls and ADLs from existing datasets are fitted against the curve produced by the regression algorithm. The fitting error of acted-falls and ADLs shows a small, but significant separation denoting that falls and non-falls are separable (see Fig. 6.7).

Several existing studies have shown how researchers distinguish a fall according to the direction[73]. This fall characteristic is discussed to show that an algorithm is robust to detect these types of falls. Nevertheless, the orientation of a falling body is not embedded in a fall detection algorithm. Here, the approach includes the orientation information in order to examine any beneficial impact on the fall detection.

Existing work has investigated this aspect of human body orientation. In [151] authors use RGB-D data to assess the orientation of a person. This work uses colour and depth for superpixel calculation of each human subject. The results from the temporal and spatial analysis (feature extraction) of those superpixels is then

fused into a dynamic Bayesian network for the final orientation assessment. The implementation of this method is complex and slow for a real-time process which is using orientation as a sub-routine. In [152] authors investigate the head and upper body orientation classification (discrete classes) based on RGB and depth image features, and linear and nonlinear classifiers. Their work relies on RGB and depth features such as a histogram of oriented gradients, depth local binary patterns, and a histogram of depth difference. Also, for the classification task, three different multiclass classifiers are considered: Random forest (RF), linear support vector machine (SVM), and sparse based classifier (SBC). A Convolutional Neural Network approach is presented in [153]. The authors use the colour image data from existing datasets for training purposes but their own data for evaluation. They claim 94% accuracy on their in-house test set validation.

Body orientation is a useful objective of this study since the falling direction can be simulated using OpenSim. As this study uses depth data, there are difficulties in estimating the orientation of the body particularly when the person is facing away from or towards the sensor. In those cases, it is hard for an algorithm or a human to distinguish the two poses without other cues as previous studies require the RGB/colour signals to detect the face. By estimating the body orientation towards the sensor and the falling direction w.r.t. the orientation, the algorithm estimates the type of fall (i.e. forward, backward, sideways).

6.3 Methodology

The fall models are set according to Chapter's 5 myoskeletal simulation preparation for rigid falls as these were investigated for the hybrid approach. Therefore, three fall models are used to process a polynomial regression method. The height of the person is measured directly from the bounding box and the orientation from the skeleton estimation – mechanisms derived by OpenNI.

6.3.1 Estimating Body Orientation

An important part of this algorithm is the necessity of the algorithm to detect the orientation at which the person is falling. The falling direction, when detected, will

be passed as a parameter to the OpenSim simulation engine in order to simulate the same type of fall.

Depth data are processed using OpenNI 2, which has automatic skeleton tracking capabilities. The skeleton mechanism is capable of tracking legs, arms and torso's motion, as soon as the person appears in the scene. The issue previously discussed with skeleton data is the lack of accurate estimation when the person is towards or away from the sensor. Each skeleton segment is connected with a joint, therefore an angle can be measured in the 3D space for each joint. The orientation towards the sensor can be inferred from joint angles extracted from skeleton tracking. An evaluation of the Kinect's skeleton capabilities is discussed in[154] where authors primarily use the Kinect SDK for their experiments. Apparently, an OpenNI evaluation is not available as a publication but is discussed in several websites, which compare the accuracy of the sensor and the software capabilities. However, an evaluation of Kinect's software falls outside the purpose of this study.

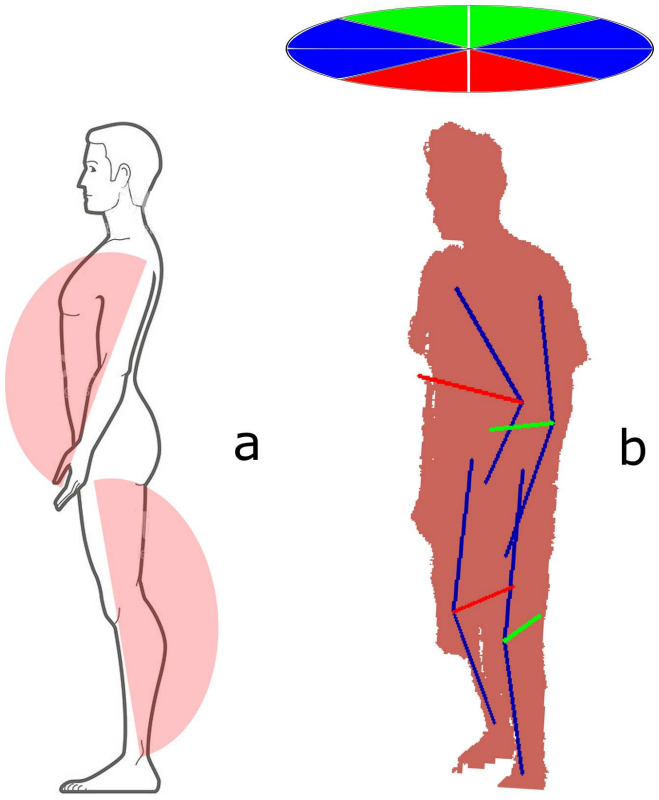

FIGURE 6.1: Angles of knees and elbows. Green line shows the bisector of the left body side and red of the right

The method works on the assumption that when a person is walking or standing, the elbow angles (between forearm and upper arm) and the knee angles (between femur and crus) have a minimum of 0 and maximum of 180 degrees (see Fig. 6.1

(a)). The direction of the bisector of these angles is towards the front for elbow angles, and towards the back for knee angles (see Fig. 6.1 (b)). Notice, that direction of the angle is measured from the horizontal plane. Figure 6.2 shows the thresholds used to classify whether the person is facing the sensor, facing away, or side-on. If two of these four angles have the same direction, then the body orientation towards the sensor is defined. Hence, the algorithm estimates whether the subject is facing forwards, backwards or sideways based on the direction of the bisectors of two of these angles. The following arguments regarding body orientation are true according to the angle α of each bisector assuming that $-180 \leq \alpha \leq 180$ in formula 6.1

$$\begin{aligned} |\alpha| \leq 45^\circ &\rightarrow front \\ 45^\circ < |\alpha| < 135^\circ &\rightarrow side \\ 135^\circ \leq |\alpha| &\rightarrow back \end{aligned} \tag{6.1}$$

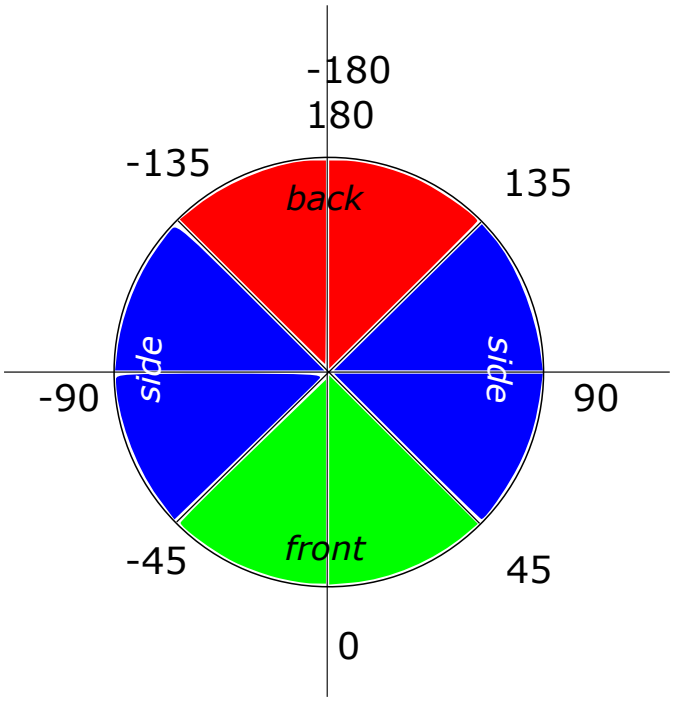

FIGURE 6.2: Angle thresholds as defined in Eq. 6.1

Having the orientation toward the sensor is the first step in estimating the falling direction. The direction of the fall is estimated by measuring the direction of motion of the CoM on the horizontal plane. This is calculated using the arc tangent of two points $atan2(z, x)$ taking these point samples after the initiation

of a fall. This is triggered by a significant change in the vertical velocity of the CoM $Vy_{CoM} > 1m/sec$, defined in this case empirically, with the assumption that this velocity does not exceed the mean velocity of sitting down ADL events. This measure is selected as sitting down (not aggressively) is one of the actions where the velocity profile has the lowest values.

The angle (β) is defined by the first point **A** as seen in Figure 6.3 assigned in the centre of the circle, while the second point assigns the direction (i.e. **B** : event towards the left, **C**: towards the back, **D**: towards the front).

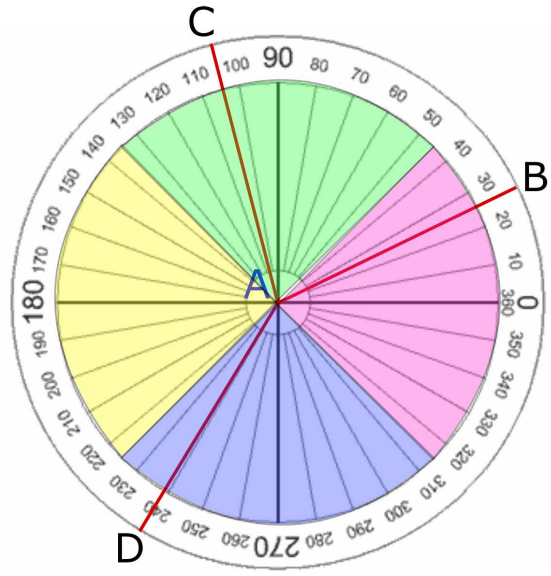

FIGURE 6.3: Falling angle thresholds denoted by colour. Blue: front, green: back, yellow: left, magenta: right

The method takes into account the pre-fall and the falling direction towards the sensors to assess the actual direction given the following rules where α is the body orientation with respect to the sensor and β the falling direction with respect to the sensor. Then, the fall direction $\mathcal{FD}$ is given by Eq. 6.2

$$\mathcal{FD} = \begin{cases} \text{forward fall} & \text{if direction } \alpha \text{ same } \beta \\ \text{backward fall} & \text{if direction } \alpha \text{ opposite } \beta \\ \text{sideways fall} & \text{otherwise} \end{cases} \tag{6.2}$$

As an example: if the person's orientation is estimated as *front* and the direction of the fall is the same using the same thresholds as in Fig. 6.2, then the fall is in the forward direction. If the falling is in the opposite direction to the body orientation then it is a backward fall, otherwise it a sideways direction fall.

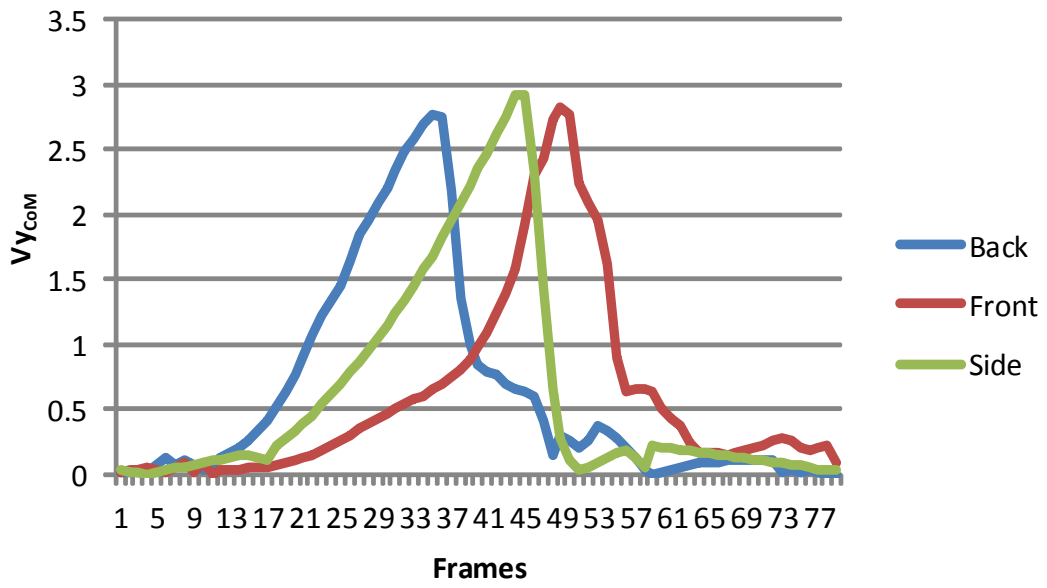

FIGURE 6.4: The velocity profiles of vertical CoM of three types of falls as simulated by OpenSim. A standard model represents a typical male body of 1.78m height and 78Kg mass was used for all three simulations

6.3.2 Vertical velocity of CoM

The dynamics of the fall are quantified by the vertical velocity of the centre of mass Vy_{CoM}. A number of simulations assuming different model heights are shown in Fig. 6.14. Both Eq. 5.3 and these results agree that the taller the body, the higher the maximum of the vertical fall velocity.

Figure 6.4 demonstrates the dynamics of different falls, based on the body/myoskeletal model orientation of the fall (forward, backward, sideways), as derived by OpenSim. These dynamics are related to the balancing forces, affected mainly by the support of the feet which differs depending on the fall orientation, e.g. maximal feet support in forward falls and minimal in backward falls. Another observation is that the backward and sideways falls have a steady velocity profile, while in the forward fall the velocity increases halfway in time between zero and the maximum velocity.

6.3.3 Hybrid Fall Detection Algorithm

The proposed detection algorithm considers the height of the person as estimated by the calibrated depth camera and the orientation of the person before the fall. An

OpenSim simulation is generated for each of the three directions of fall and for four specific model heights (as discussed in Chapter 5) though the models have different min and max as well as sampling (1.2 - 2.1 m with 0.3m sampling). An OpenSim simulation is generated for each of the fall directions and four model heights using the standard (default) myoskeletal model which is simplified by excluding the arms. The upper weight of the body (i.e. torso) was adjusted in order to compensate for the missing weight of the arms. The vertical velocity $Vy_{CoM}(t)$ of the CoM is measured from these simulations to create the velocity profiles. These profiles are processed using a polynomial regression algorithm (Eq. 6.3) where t is the feature, a are weights assigned to a particular feature and i ($i = 0 \ldots N$) the degree of the polynomial.

$$Vy_{CoM}(t)_{(pol)} = \sum_i a_i t^i \tag{6.3}$$

The result of the process returns a fitted curve which approximates the velocity profile of each fall type from the simulation as seen in Fig. 6.5. Using actual fall and ADL examples, the algorithm measures the vertical velocity profile of the CoM. Each profile is fitted with the derived curve of the polynomial regression of a model with known height. The fitting error between the polynomial curve and actual examples ϵ is calculated by (Eq. 6.4) where the log of polynomial fitting error is produced in order to fit results into scale for viewing purposes.

With the use of regression, the algorithm processes a form of a velocity profile to a mathematical formulation of this profile which is then easier to be compared with other profiles. The degree of polynomial was adjusted manually in order to fit the best possible curve of the velocity profile. Furthermore, other fitting methodologies could be used here in order to fit a Gaussian or GMM to the velocity profiles. Nevertheless, such fitting methods were not tested at this stage of the study and an off-the-shelve regression algorithm was preferable for this hybrid approach. Each regression curve is fitted on velocity profiles of a model with a known height, therefore, the regression curve is parametrised by height. Also, the falling orientation can be expressed as an angle as its initially measured parameters (in degrees) of the body orientation towards the sensor and falling direction.

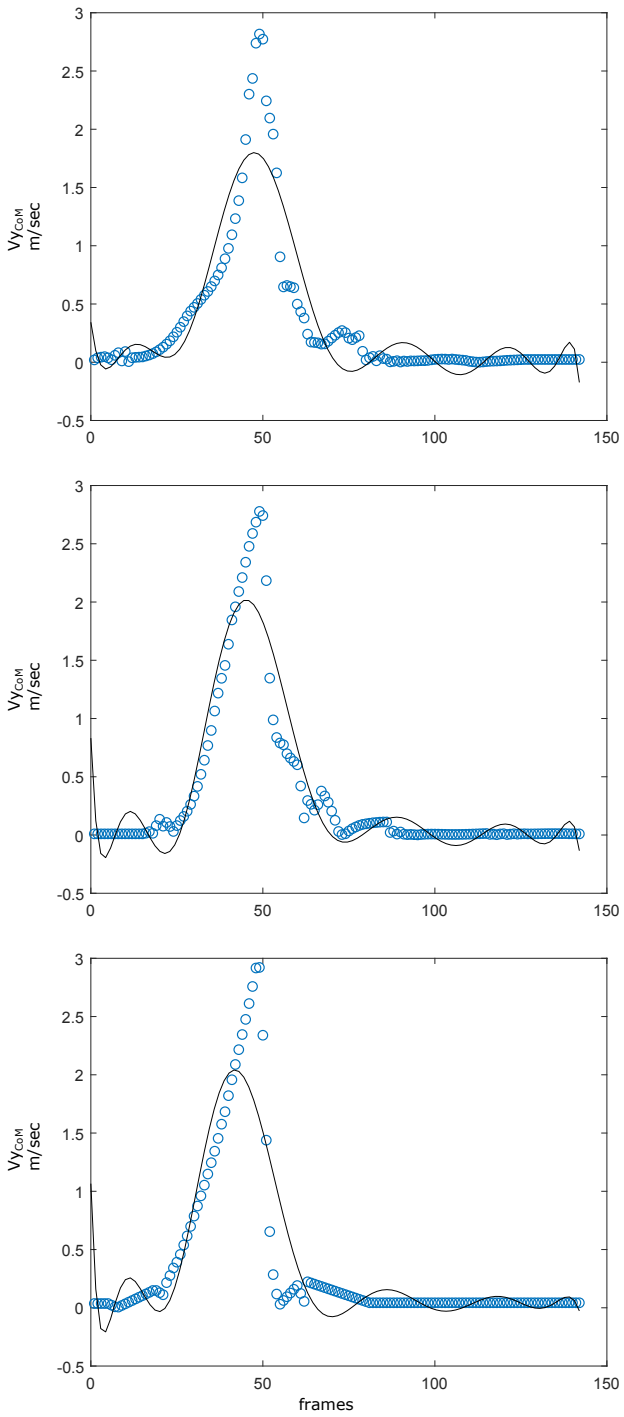

FIGURE 6.5: Polynomial fit of three fall types

$$\epsilon = \log[\sum_{t}(Vy_{CoM}(t)_{(pol)} - Vy_{CoM}(t)_{(meas)})^2] \tag{6.4}$$

When actual events are evaluated, each derived error is shown in Fig. 6.7, where non-falls and falls (i.e. front, side and back) are accumulated. These error values are processed using a linear SVM which specifies a decision boundary between the two classes of falls and non-falls. The detection of falls is therefore determined by the separation of the two classes.

6.4 Experimental Results

6.4.1 Evaluation of Body Orientation Estimation

A simple procedure for testing this approach is conducted by recording and analysing depth sequences of human subjects performing turn and stop actions. Every subject had to start the action by facing the sensor, then turn approximately 90 degrees and stop (this is where the sample is taken), then perform again until is back on facing the sensor (providing 5 samples). 7 subjects (5 male, 2 female) performed 3 trial videos at different locations within the scene. In each trial, they had to perform 3 rotations (315 samples). The *accuracy* is calculated as the ratio of the number of orientation types correctly classified ($N_{cor} = 292$), divided by the total number of samples ($N_{tot} = 315$) as in Eq 6.5.

$$accuracy = \frac{N_{cor}}{N_{tot}} \tag{6.5}$$

Here, the approach delivers 92.7% accuracy. As seen in Figure 6.6 the algorithm can distinguish between the side, front, back human orientation towards the sensor. To evaluate the falling direction, the algorithm used data from datasets UR, GM to estimate the orientation. One of the complications using these datasets is that the skeleton detection operated by OpenNI does not initialise on time to perform the pre-fall estimation. Therefore, due to this reason, only 61.39% were accurately detected showing the limitations of the approach on short videos, where only the fall event is included in the sample.

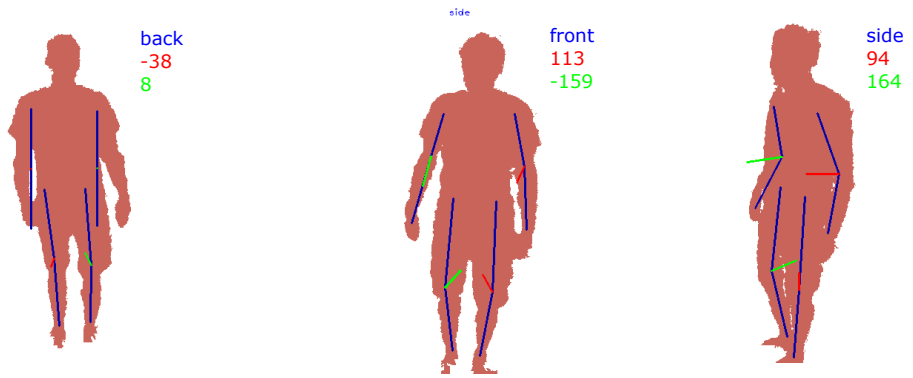

FIGURE 6.6: Three examples of body orientation as detected by the algorithm. Notice the green and red lines showing the direction of the angle in 3D.

TABLE 6.1: Performance of the Hybrid Algorithm

Method	Accu (%)	Prec (%)	Sens (%)	Spec (%)
GM	100.00	100.00	100.00	100.00
GM2	98.50	95.31	100.00	97.84
UR	94.54	80.00	100.00	93.02
SDU	92.00	84.50	77.52	96.03

6.5 Evaluation of the Hybrid Detection algorithm

The datasets GM GM2 UR and SDU are used to evaluate the detection performance. The protocol requires all samples to be compared with the regression curve of the specific simulation according to height. This implies that each sample from these datasets is used for testing as training is using only data from the simulation of falls. For each sample from the dataset, the height of the person is measured from the 3D bounding box and the body orientation is estimated using the method described in 6.3.1, to select the closest simulation model and properly fit a 12-degree polynomial. Table 6.1 summarises the results.

The algorithm has 100% classification accuracy on the GM dataset and the measured error (ϵ) can be seen in Table 6.1 and in Fig. 6.7, where falls and non-falls are linearly separable for this dataset. The red line represents the decision boundary, as specified by a linear SVM. As seen from the Table 6.1, the algorithm improves the performance reported in Chapter 4 (Table 4.1), showing the benefit of using this algorithm. This is especially noticeable in SDU dataset for Specificity (95.18) and how non-falls are not incorrectly detected as falls when the myoskeletal model is selected properly to have an approximate height to the actual observation from the datasets. This point will be further justified in Section 6.8.4.

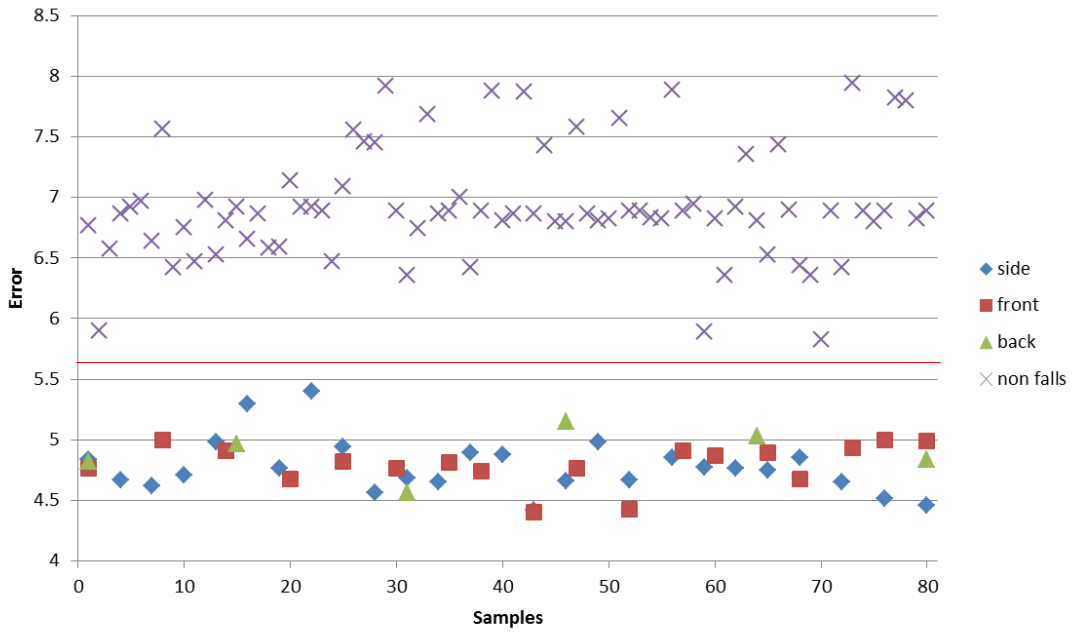

FIGURE 6.7: Fitting error (ϵ) of falls and non-falls for GM dataset: Notice how well separated falls and non falls appear

Fall type \ Approach	Customised	Trained based
Back	4.93	5.08
Front	4.82	5.12
Side	4.84	4.98

TABLE 6.2: Mean error ($\log \epsilon$) from height customised and Trained based approaches

The approach is compared against a similar baseline method where the polynomial regression is applied on three acted falls of the same person and then evaluated on the remaining samples of the GM dataset. This is referred as the Trained based approach (See results in Table 6.2). That is from the 48 fall examples, three examples are taken from the same person acting three different falls and the velocity profiles are processed by the regression algorithm. The evaluation also uses SVM applied to the error values derived from Eq. 5.4, but lacks the customisation in height, provided by the simulation. Specifically, for the baseline approach, $Vy_{CoM}(t)_{(pol)}$ is estimated by a polynomial (12th degree) regression using three fall examples from the dataset. The mean error for the proposed customised approach is lower for all different types of falls as seen in Table 6.2. Meaning that the error from using actual data is not as low as when using simulated data for training.

6.6 Discussion

This approach presented a method for customising the expected human body falls, using their height and fall direction, based on physics-based simulations, derived by the OpenSim software. Customised simulation provides a lower error when used to train the algorithm compared with a similar approach where polynomial regression is applied to human recorded fall events. The simulated model and its synergy with existing machine learning algorithms could be further investigated, with the aim to further customise and therefore optimise fall detection algorithms, but also to address the issue of the scarcity of realistic fall recordings.

The physics-based approach presented here provides valuable outcomes regarding the use of simulated data applied on fall detection. It shows that a fall detection algorithm is feasible using a customisable simulation of a person's height. This approach also utilises the type of fall (forward, backward, sideways) to fit the data with better results than the trainable machine learning approaches discussed in Chapter 4.

6.7 Fall detection using Myoskeletal simulation

Having discussed the hybrid approach we can still observe a trainable methodology of a single velocity profile using a regression algorithm. Here, a new algorithm compares the velocity of single fall and non-fall simulated profiles. A different observation point is selected in order to have an occlusion robust feature such as the head's vertical velocity. Features discussed in Chapter 4 which are relying on the bounding box, particularly the composite of width and depth (L) would perform poorly in the event of occlusions covering a large area of the scene.

A new metric in comparing the velocity profiles was investigated which takes into account the entire velocity profile without using a fitting (i.e. regression) method as previously in the hybrid approach. The Hausdorff Distance (HD) will be discussed for its use and will be evaluated for its feasibility to distinguish falls from ADL.

The differences of this approach from the hybrid is the use of a myoskeletal simulation to model falls and ADLs hence no recorded human fall data is required

to develop height customisable myoskeletal models for the ADLs in order to have both negative and positive classes. Also, no need for a machine learning technique is required. The use of fall direction is not used given an evaluation of the fall events described in Section 6.8.2, demonstrating that the HD metric for the three directions indicates a narrow variation.

6.7.1 Use of Hausdorff distance

Most of the existing research in fall detection uses features related with the person's silhouette, posture, body part, pre-post body location/orientation, velocity etc. measured or detected instantaneously or over a fixed time window. OpenSim provides simulation results of location, velocity and acceleration profiles. A velocity profile for fall and ADL events provides an entire sequence of samples from the beginning to the end of the event. Such full profiles can be beneficial for the classification of an event when compared with, at their full length. A novel approach is introduced here which takes into account the full velocity profile instead of the features noted in Chapter 2.

The observed velocity profile will be measured for similarity against simulated selected profiles (i.e. from a fall and an ADL). In Chapter 4, the use of velocity during a specific frame was used instead of the complete velocity profile as in this algorithm. The use of HD similarity was used in Chapter 3 to assess the hesitation of a fall, and in Chapter 5 to assess the quality of simulation. It is also used for measuring the similarity between simulated events models as well as in for the use of fall detection. The Hausdorff distance (HD), is defined as the distance between two profiles A, B,

$$HD(A,B) = max\{\max_{a\in A}\min_{b\in B} dist(b,a), \max_{b\in B}\min_{a\in A} dist(a,b)\} \tag{6.6}$$

Several distance measures exist for this application, such as the Euclidean, Frechet and Hausdorff. The Hausdorff distance is used in [155] where authors employ it to determine the similarity of trajectories. A useful property of the Hausdorff distance is that it does not require for the profiles to have the same length which is not the case for the Euclidean distance measure, which uses a point-for-point pairing of the trajectory coordinates, as seen in [156]. The distance measure used by Frechet [157] relies on the segment measurements which produce a longer distance than the

Hausdorff and is calculated as the minimum of maximum distances between two curves - effectively is the opposite to the Hausdorff distance. More importantly, the use of the HD to compare velocity profiles (see Section 6.7.2.2) overcomes the need of a threshold.

6.7.2 Methodology

This implementation is based on processing depth data, hence, falls can be represented by a single profile, although the proposed methodology can be adapted for RGB modality, if the scene is calibrated, or even for non-visual modalities, e.g. accelerometer data, because velocity profiles can be similarly generated. Subsection 6.7.2.1 focuses on the use of depth data and particularly on the feature used for this approach.

6.7.2.1 Data pre-processing

The choice of feature for fall detection is important, especially when occlusions are expected, as the visibility of relevant points should be maintained for as long as possible during the fall. Existing studies [37, 38] detect and track the head centroid as they consider it as the most suitable landmark point for this purpose, as this is the highest and most visible point on the body. Nevertheless, these head detectors are not rotation or scale robust and therefore, the top of the head location will be considered, approximated by the top bounding box coordinate [31]. For this work, depth data recorded by a Kinect I sensor and depth data analysis was implemented on the OpenNI platform [103]. The bounding box is estimated from the 3D point cloud of the segmented human tracker in OpenNI. The top point is found at the location where the bounding box touches the head of the person as shown in Figure 6.8 where the point is observed during the fall. Notice how closely estimated this position is with the head location. The depth data, which is generated from an infra-red sensor, can be noisy due to the interaction with hair, where the infra-red signals are absorbed rather than reflected. Therefore, the vertical location is filtered using LOWESS (Locally Weighted Scatterplot Smoothing) [158]), which suppresses the noise whilst maintaining the shape of the vertical location profile.

The estimated person's height is used as an input parameter for the simulation (Section 6.7.2.2) and measured from the bounding box (in metres). The height is

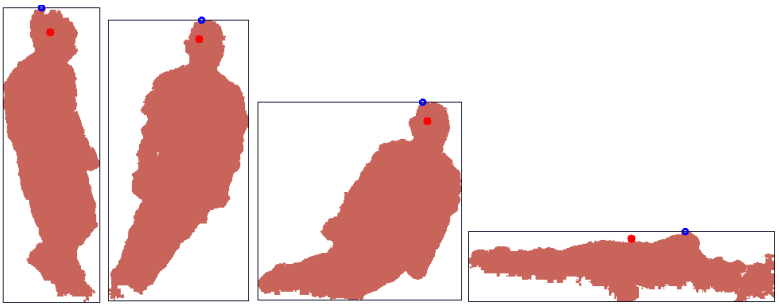

FIGURE 6.8: Blue dot indicates location of the top bounding box point and red dot to indicate the head location

estimated from the depth data because public datasets do not provide such data. Alternatively, height may be determined as a pre-set parameter in many situations (e.g. at home), for independent livers.

6.7.2.2 Fall detection

The proposed fall detection algorithm is summarised by the flow diagram shown in Fig. 6.9 and described in detail in this section. Fall detection is performed by comparing the velocity profiles between observed events and simulated activities of fall or ADL events. The red box contains the pre-processing steps discussed in 6.7.2.1 where depth video samples are processed to derive the top bounding box coordinate and person's height. The elements contained in the blue box compute the simulation of falls and ADLs, as discussed in section 5.3.2.1. A rectification step is required before processing the simulation and actual signals (shown as a black box) in order for the simulation and actual data samples to have the same bit rate.

The green box encloses the fall detection algorithm. Inputs of the algorithm are the $V^{+}(t)$ and $\left\{V_i^{-}(t)\right\}$ which represent the velocity profiles of the simulated fall and non-fall events respectively and $Y(t)$ is the top bounding box location and the person's height. There are N simulated ADLs $(\left\{V_i^{-}(t)\right\}, i = 1 \ldots N)$ such as sitting down, lying down, etc., which are processed in order to contain the active part of the motion (i.e. keep only the velocity profile where there is activity). A non-fall velocity profile $V^{-}(t)$ which is compared to the remaining ADL velocity profiles, is selected by minimising the Hausdorff distance against the profile of the simulated fall $V^{+}(t)$:

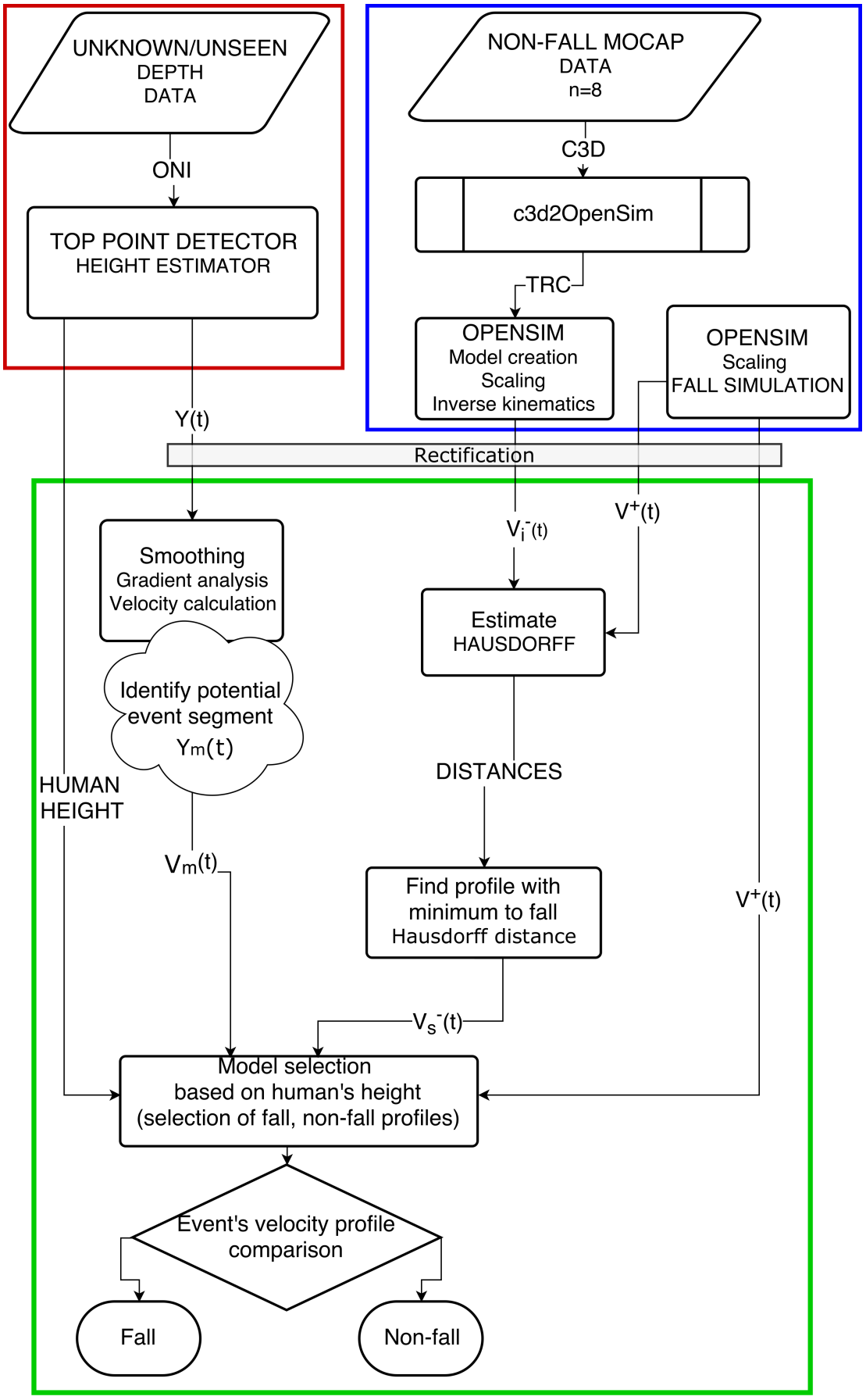

FIGURE 6.9: The pipeline of the myoskeletal simulation fall detection system. Red box encloses the data preprocessing, blue box the model simulation and the green box the fall detection. ONI: depth data format, C3D: standard mocap data format, TRC: OpenSim motion format

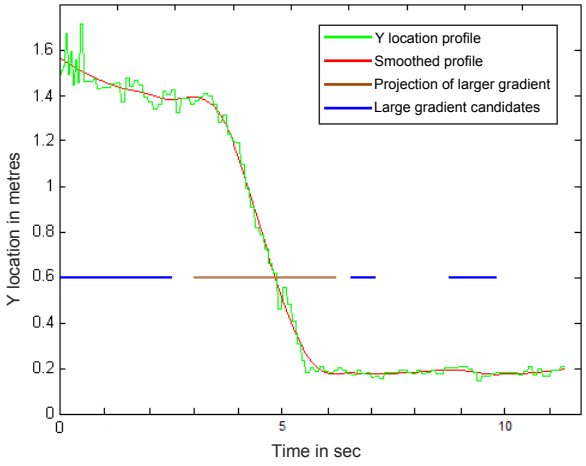

FIGURE 6.10: Selection of larger gradient of an ADL (lying down)

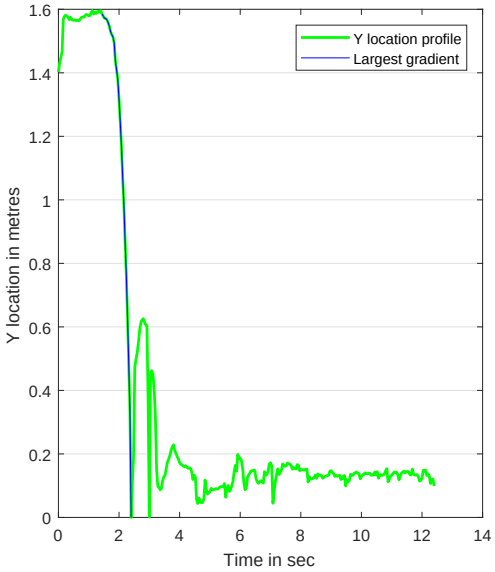

FIGURE 6.11: Selection of the larger gradient of a fall event

$$s = \underset{(i=1...N)}{argmin}(HD(V_i^-(t), V^+(t))) \tag{6.7}$$

The fall profile $V^+(t)$ will also serve as a comparison measure in the next stage when actual human events will be tested. Evaluation of the Hausdorff distance can be found in Section 6.8.1.

The signal containing the bounding box top $Y(t)$ sequence of the detected person is processed in a similar way to identify potential fall segments. A single $Y_m(t)$ location profile containing the event (either a fall or non-fall) is selected by extracting the fragment that encloses the longest and steepest negative gradient. To perform the gradient analysis the algorithm detects a change of height by comparing the current and previous y location. Two further checks are required to select the profile segment. First, the algorithm measures the duration of each negative gradient (if there is more than one) and selects the longest one. Second, it measures the start and end y coordinates of the location and selects the tallest one. The process is illustrated in Fig. 6.10 for the selection of the largest gradient of an ADL and in Fig. 6.11 for the largest gradient of a fall event.

The event's velocity profile ($V_m(t)$) will then be estimated based on this segment. In the last process of the pipeline, $V_m(t)$ is compared against the non-fall and fall simulated profiles ($V_s^-(t)$, $V^+(t)$). These simulated profiles are pre-generated as

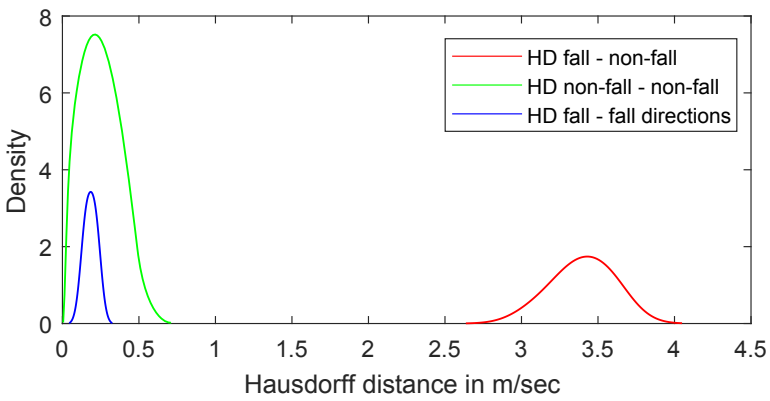

FIGURE 6.12: Plot of a Gaussian pdf fitted to the distribution of Hausdorff distances: blue, red and green curves denote the pdfs of HDs between falls, between non-falls and between falls and non-falls respectively

discussed previously in Section 5.3.2.3 at a discrete set of heights from simulated models with the height approximately equal to the subject's height. The minimum distance from this comparison will determine whether the actual event E is a fall or a non-fall and is given by Equation 6.8:

$$E = \begin{cases} fall & if \quad (H(V^{+}(t), V_m(t)) \leqslant H(V_s^{-}(t), V_m(t))), \\ non-fall & if \quad (H(V^{+}(t), V_m(t)) > H(V_s^{-}(t), V_m(t)). \end{cases} \tag{6.8}$$

6.8 Experimental Results and Discussion

6.8.1 Validation of Hausdorff Distance

A benefit of using the Hausdorff distance is the capability to compare two profiles of different sample lengths. The capability of HD to differentiate between fall and non-fall profiles (Fig. 6.13) is validated by intra-class (fall vs fall, non-fall vs non-fall) and inter-class (fall vs non-fall) comparisons. The velocity profiles of eight different ADL events and four different falls (forward, backward, sideways, collapsing) are considered and Figure 6.12 shows the probability density functions (pdf) of the above HDs. For intra-class comparisons, the values of HDs are in the range 0-0.5 m/sec, while for inter-class comparisons they cluster around 3.5 m/sec. Therefore the intra-class HDs are small compared to the HDs between a fall and non-fall events, which is nearly an order of magnitude larger. This justifies the

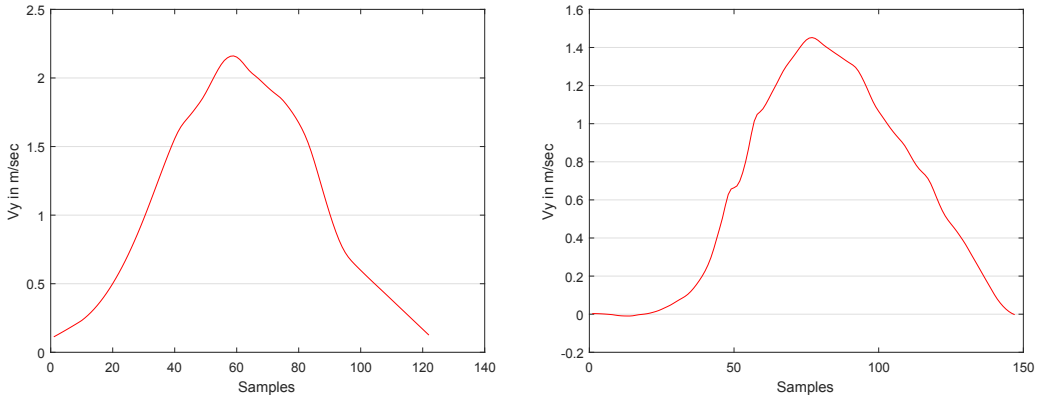

FIGURE 6.13: Simulated ADL velocity profiles of sitting down actions

choice of HD as a distance metric to discriminate between the velocity profiles of events.

6.8.2 Comparison of the fall simulation models

This section evaluates the similarity between simulated models within falls of rigid and collapsing types. The evaluation concludes with using a single type of fall (i.e. forward fall simulation).

The base for our experiments is the Full Body Model [159] in OpenSim with the properties of an average male with a height of 1.78m and weight of 78kg. This model is then scaled (as described in 5.3.2.3) to cover the range of people's heights between 130-190cm in 20cm steps. Fig. 6.14 shows the four velocity profiles simulated by these four models with heights 130, 150, 170 and 190 cm ($V^{+}(t)$) executing a sideways fall. Notice the difference in peak velocity that varies from 5-6.5 m/sec according to the model's height. The customisable model will use this principle to simulate falls according to the person's height.

Previous studies discuss different types of falls as defined by the falling direction, such as forward, backward and sideways fall [73]. Collapsing falls are also discussed in Chapter 3, where the person falls vertically then leans forward towards the ground. Fig. 6.15 shows the simulations of these types and show the variation in V_y. Some noise is observed due to elasticity of the Contact Geometry (OpenSim parameter) of the model, before the initiation of the fall as the model touches the platform and before it comes to rest. To compare these velocity profiles the HD is used. The HD between a forward and a backward fall is 0.215 m/sec, between

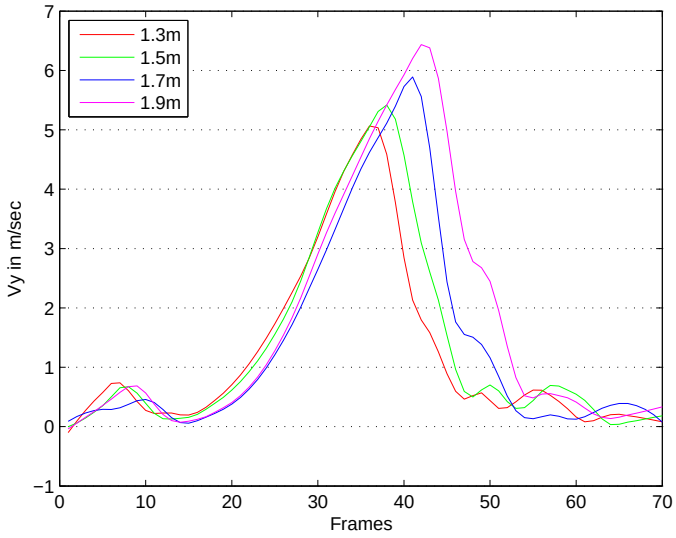

FIGURE 6.14: Velocity profiles of four simulated models of a sideways fall

forward and sideways 0.216 m/sec, and between sideways and backwards 0.154 m/sec. Also, the HD between the collapsing fall and backward is 0.27 m/sec, forward 0.303 m/sec and sideways 0.323 m/sec. To show how small the HD is, a pdf is plotted for this comparison in Fig. 6.12 denoted with the blue curve. Effectively, there is a little benefit in simulating different types of falls as the HD is small, hence, there is similarity of the V_y across the four types of falls. For that reason, only the forward fall simulation produced by the OpenSim engine is used.

6.8.3 Evaluation of myoskeletal simulation based fall detection

Since our methodology does not require samples for training, the evaluation results are reported for the full datasets where other studies use a sample of about 70-80% of the dataset. Table 6.3 summarises the results of the proposed method when tested against two public datasets alongside the performance of other methods tested on the same datasets. The proposed methodology outperforms previous works on both datasets in terms of accuracy, precision and specificity and its sensitivity is similar to all but [110] where more falls are detected, but more ADLs are also detected as falls (FPs). These results show that the simulated approach has almost equal or better performance when tested against these public datasets. In conclusion, the simulated model is shown to provide a more accurate representation of a falling body than actors can simulate, while the whole velocity profile of

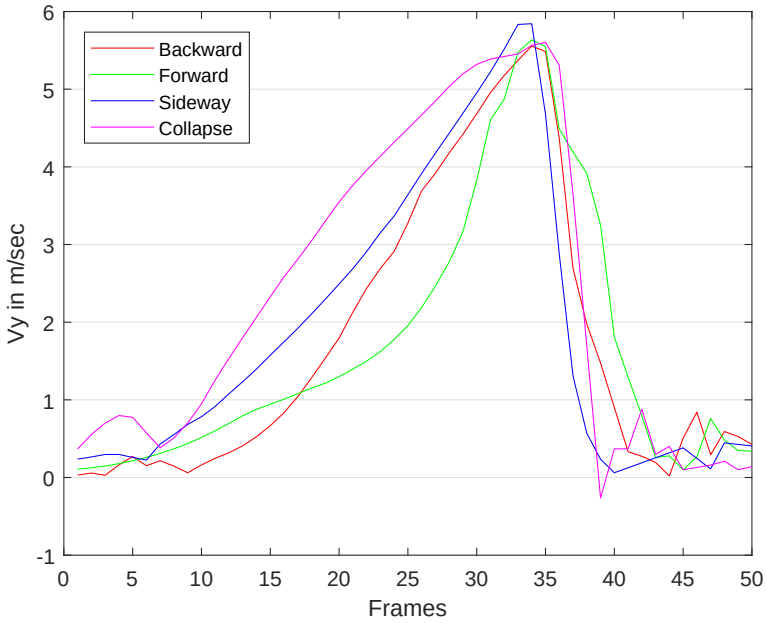

FIGURE 6.15: Velocity profiles of four Opensim simulated falls. We can see the visual similarity of the profiles.

the simulation is taken into account. Hence our method shows that it is possible to perform robust fall detection using only simulated data for fall modelling without the need for human fall data to train the detector.

6.8.4 Evaluation of using a customised simulated model

In order to evaluate the benefits of height customisation, datasets **UR, SDU, GM** are merged into a single dataset with a total of 263 fall and 1212 non-fall events. The algorithm is then tested against this set four times, each time with a different simulated model (130, 150, 170, 190cm tall) to assess each model's capability to detect falls and filter out non-fall events. Table 6.4 shows the results of using fall/non-fall data from across all three datasets **UR**, **SDU** and **GM**.

These events are categorised according to a subject's height, resulting in 10 samples in the range (120-139) cm, 110 samples (140-159 cm) etc. The results show that detection fails when 130cm tall subjects are tested using the 170 and 190cm simulation models, and also when the 150cm subjects are tested against the 190cm model. One can conclude that the 130 or 150cm models can be used to detect all fall events, but this will lead to detecting more non-fall events as falls (FNs) since the velocity profile is lower in comparison with the 170 and 190cm models. This

Table 6.3: Performance of fall detection algorithms developed in this thesis and comparison against previous studies across 2 public datasets

Method	Accu (%)	Prec (%)	Sens (%)	Spec (%)
(a) Dataset UR (15 fall, 40 ADLs)				
Bourke et al.	95.00	90.91	100.0	90.00
[4]	98.33	96.77	100.0	96.67
[79]	-	-	100.0	97.25
[80]	99.37	96.77	100.0	99.23
Algorithm 1	94.54	80.00	100.00	93.03
Algorithm 2	96.36	86.66	100.00	95.23
Hybrid	94.54	80.00	100.00	93.02
Myoskeletal	**100.0**	**100.0**	100.0	**100.0**
(b) Dataset SDU (200 fall, 800 ALDs)				
[5]	86.83	-	91.15	77.14
[109]	91.89	-	-	-
[110]	92.98	-	**93.52**	90.76
Algorithm 1	74.40	66.50	41.30	90.11
Algorithm 2	89.50	82.38	71.48	95.18
Hybrid	92.00	84.50	77.52	96.03
Myoskeletal	**96.90**	**94.00**	90.88	**98.48**

is reflected in Table 6.4 where several events are detected as falls because those events have velocity profiles similar to fall events of the 130 or 150cm models. For example, the bottom-left value of **0.20** denotes that only 2 fall examples were correctly detected, and 8 were missed. Specificity % (values after the dash) denote the TPs correctly detected as non-falls. The values in the top-right corner of the table (**0.66**) denotes that only 8 out of 12 examples are detected as non-falls, meaning that 4 ADLs are classified as falls. In other words, a tall person abruptly sitting or lying down has similar a velocity profile as for a short person falling; this problem is addressed by our approach which is height sensitive and, hence, the algorithm selects the appropriate myoskeletal model. The comparison of the velocity profile of a human event, either a fall or ADL, is against two velocity profiles derived from the simulated model (e.g. the vertical velocity profiles of an ADL and a fall). These simulated profiles are derived from models which approximate the height of the person. The proposed approach does not permit cases where a tall person sits down abruptly and their velocity profile is compared with simulated profiles of a short person unless the height is incorrectly estimated (in many circumstances, the height may be preassigned or would not rely on instantaneous measurements). Hence, the improvement of the detection rates against TNs (Specificity) in Table 6.4.

Table 6.4: Simulated model height variability over **UR, SDU, GM** datasets combined. The table presents results of the algorithm for each height models applied to the height-labelled acted datasets (**UR, SDU**) and the in-house depth dataset (**GM**) showing the sensitivity and the specificity for each combination of simulated model height and approximate human height. If height selectivity is applied then detection is 100% for both sensitivity and specificity (main diagonal). Values in bold denote either missed detections (for sensitivity) or false positives (for specificity)

	Approximate human height number of samples (falls, non-falls)			
Sim model height	130 (10, 40)	150 (110, 440)	170 (108, 441)	190 (8, 12)
130	1.00 — 1.00	1.00 — 1.00	1.00 — **0.88**	1.00 — **0.66**
150	1.00 — 1.00	1.00 — 1.00	1.00 — 1.00	1.00 — **0.91**
170	**0.80** — 1.00	1.00 — 1.00	1.00 — 1.00	1.00 — 1.00
190	**0.20** — 1.00	**0.88** — 1.00	1.00 — 1.00	1.00 — 1.00

This experiment tests each myoskeletal model against all the data (regardless of person's height) and observes its performance. The main diagonal of the table corresponds to results where the appropriate height model is used for the detection and operates without error (i.e. 100% Sens, Spec). As can be seen, the detection produces FPs and FNs if the incorrect height model is selected (results in bold).

Note that these datasets have a limited number of fall data, and the results would be expected to degrade if more data from low height people were included. Also, the experimental data used for this experiment is derived from the correctly classified samples when customisation is enabled. This experiment confirms that model variability is important.

6.8.5 Use of simulation data in machine learning

The use of myoskeletal simulation velocities can be used to train a machine learning algorithm such as the ones in Chapter 4 or a nearest neighbour algorithm. Such an approach should take into account the height of the person as it relates to the same issues discussed in the previous Section (6.8.4) where the sitting down action of a tall person can be confused with a short person falling. Therefore, in the use of machine learning personalisation should be taken into account to avoid such issues.

6.9 Acceleration based fall detection using myoskeletal simulation

To evaluate the approach of fall modelling using simulation, experiments were conducted using acceleration based data of falls and ADLs. With the use of a myoskeletal model, we can measure the acceleration of a falling person and compare it with the acceleration produced from a falling model. To perform this, the algorithm requires the acceleration of a particular point such as the CoM or the head location of the person in order to take such measurements from the simulation. The evaluation of this approach uses data from accelerometer based fall datasets, such as the SisFall [106] which provides a variability in physiological characteristics. The acceleration from this dataset is measured from the CoM using an accelerometer fixed on the waist.

There are pros and cons in using a wearable accelerometer as discussed in Chapter 2. Briefly, the use of wearables is invasive and the user must wear it continuously to avoid a missed-detection of a fall. The benefits are that only a single sensor is needed (per person) as opposed to the solution of an in-house network of cameras or sensors, hence is cost-effective and less complex to setup and maintain. Also, this approach is occlusion robust as the user is not expected to be in the f-o-v of a sensor to be detected. It is generally observed [45], that accelerometers are fixed on the waist rather than other places on the body in order to become less invasive. This is one of the reasons why accelerometers are not fixed on the head though this would give a better detection of a fall event. Another reason is that accelerometers are quite sensitive and the head motion is expected to have more rapid movement than the waist under normal motions.

6.9.1 Methodology

A modification of the previous fall detection algorithm is used. Instead of the vertical velocity of the head/top bounding box point, the algorithm uses the feature described in [106] where the standard deviation on the horizontal plane (C_8) is calculated from accelerometer data. The formula for this feature is given by

$$C_8[k] = \sqrt{\sigma_x^2[k] + \sigma_z^2[k]} \tag{6.9}$$

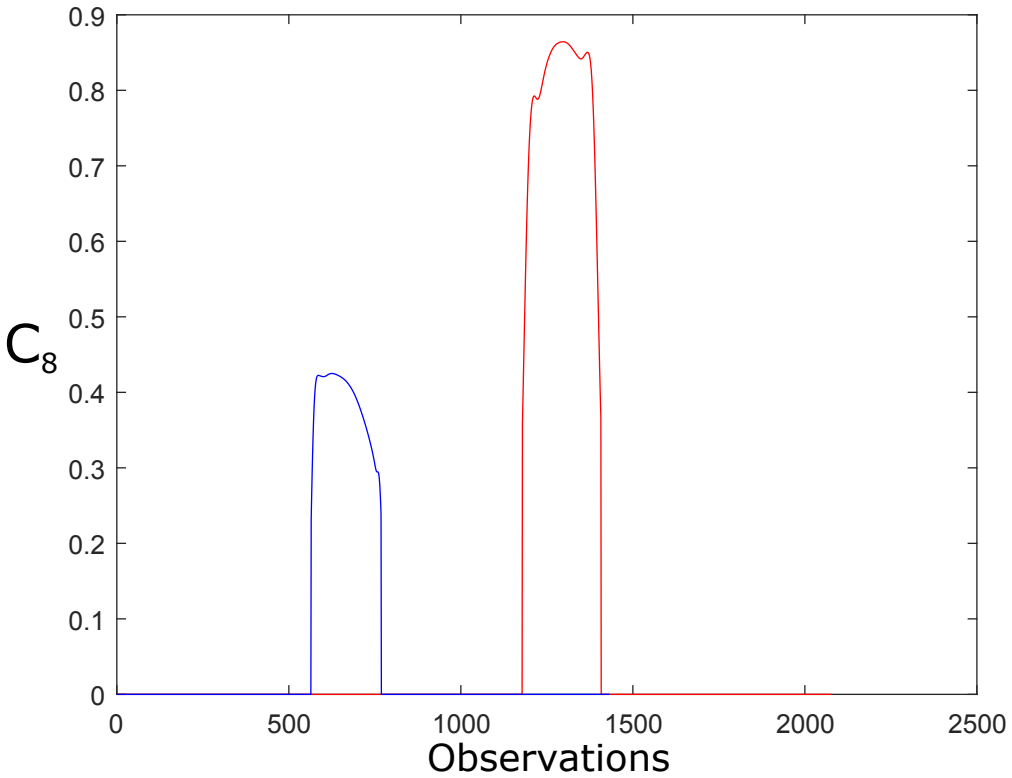

FIGURE 6.16: Two C_8 profiles as selected by the Hausdorff distance. Blue graph: ADL simulated profile (sitting down), red graph: simulated fall

where $\sigma_i = std(a_i[k])$, k is the sample window of 100 observations

Using the same simulations from OpenSim of falls and ADLs, the algorithm calculates this feature and performs a similar evaluation to select the two profiles as described in Section 6.8.1 where the fall event and the nearest in HD ADL event are selected. Fig. 6.16 shows a simulated fall profile and the closer simulated ADL to it via comparison using the Hausdorff distance. These two profiles are the measurement against an actual event from the Sisfall dataset [106]. The height is assigned to the algorithm as this information is supplied with the dataset but note that there is not any means of measuring the height from accelerometer data.

6.9.2 Results and discussion

To evaluate the validity of acceleration based fall detection data from [106] were used. This dataset covers a range of human variabilities such as heights, weights, age and sexes. Therefore, this dataset provides the necessary data for evaluation.

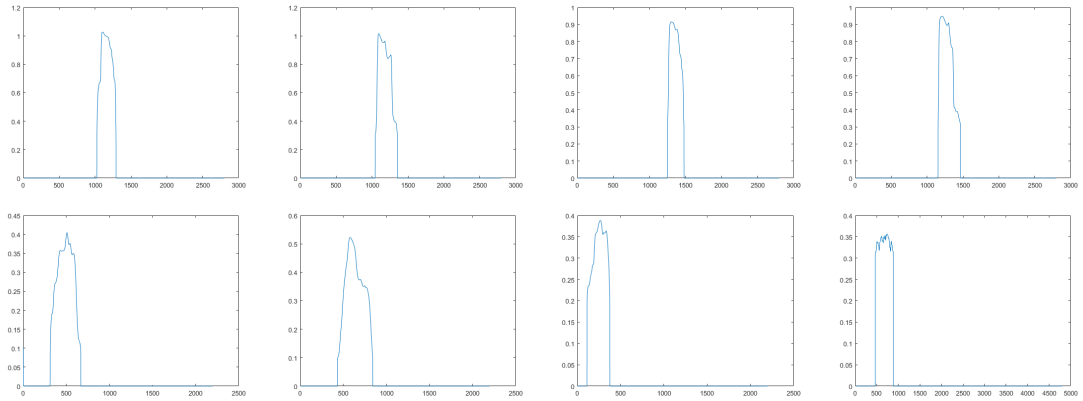

FIGURE 6.17: Typical examples of C_8 profiles of falls (top row) and ADLs lower row

TABLE 6.5: Performance of fall detection and comparison against SisFall dataset

Method	Accu (%)	Prec (%)	Sens (%)	Spec (%)
Mezghani [160]	96.60	-	99.60	94.44
Abdelhedi [161]	70.87	-	54.14	82.85
Nguyen [162]	98.55	-	97.70	99.73
Proposed	99.40	100.00	96.91	100.00

The dataset has several fall types such as collapsing and rigid falls, but also seated falls which have occurred when a person falls while seated. Since simulations of the seated-fall type are not performed by the proposed approach, such fall types will not be used for the evaluation. As a result, six fall types will be used as for evaluation (F01 – F06 from the Sisfall dataset). The fall trials are performed by 23 young adults, of these 6 types, trialled 5 times (690 fall samples). Also, all 3002 ADLs were used from this dataset performed by 38 subjects. Fig. 6.17 shows examples from subject SA21 performing falls and ADLs. The difference is noticeable as the falls have about twice the height of ADLs in the C_8 profiles.

Table 6.5 reports recent algorithms tested on the SisFall dataset. From these results, it is noticeable the benefit of the proposed approach on the Specificity as all falls are detected when the C_8 feature is used. Issues occur when ADLs include events walking upstairs and downstairs quickly and quickly sitting onto a low height chair which causes the Sensitivity to drop as the profiles are much more similar to the fall profile. Also, Mezghani et al. [160] study performs better in detecting falls due to a different feature used for their study.

6.10 Conclusion

This Chapter has reported an investigation of simulation methodologies to validate the development of robust algorithms for detecting people falling over or collapsing. It provides a proof-of-concept for the use of a myoskeletal simulation model as a replacement for real data in the improvement of detector performance.

The fall models were evaluated using real fall data from Youtube, showing the realism of fall modelling. The fall detection system is evaluated on depth and accelerometer datasets. In addition, a new approach to creating fall events based on simulation by a myoskeletal model has been successfully demonstrated, negating the need for real or acted fall-training-data.

Fall velocity profiles differ significantly when compared with ADL velocities which shows a separation between the two classes. Different fall types were also compared (front, back, side and collapsing) and were shown to have similar profiles in the same way that ADL profiles when compared with each other returned small values of the Hausdorff distance.

This approach is innovative in utilising the person's height as a parameter to simulated myoskeletal fall models. The use of simulated data enables data to be acquired easily, negating the need to use actors attempting realistic falls and risking injuries. It provides a tool that is capable of significant flexibility, potentially enabling a wide range of a variation in the physical characteristics and the type of falls that can be simulated. The existing system only covers the customisation of models in terms of height in order to show the capabilities of the simulation-based approach. By using a customisable model based on the height of the falling person, better sensitivity and specificity are achieved. The evaluation against public datasets comprising more than 1000 sample videos clearly demonstrates its robustness and improved performance compared to those achieved by other researchers.

Chapter 7

Occlusion robust fall detection

Computer vision systems are dataset-driven and tend to ignore the scene attributes that contribute to effective fall detection. As discussed in Chapter 3, only one study [7] focuses on the how occlusion affects the detection of falls, while another study tries to overcome the effect of occlusions by monitoring the scene using multiple cameras [2] whilst the rest of the public datasets contain fall incidents without including any environmental elements (i.e. furniture). Every home scene has numerous objects which could cause partial or full occlusions as the person moves around, including furniture such as beds, sofas and chairs which could obscure a fall event.

Current studies use a variety of features (as discussed in Chapter 2) such as the bounding boxes, silhouettes, shapes, skeleton tracking (offered by Kinect platforms), head velocity etc. Many of these would fail to detect a fall under occlusion, particularly if the degree of occlusion covers a large proportion of the body. This is because the observation time of these features is limited, as the event is interrupted due to the occlusion and the algorithm is not capable of recovering the missing motion of the event. Methods requiring the visual location of the body after the fall would also fail to detect the fall (as well as the faller), as the resting location of the faller is occluded. Algorithms which use the head position would have better chances due to the greater visibility of this feature, as discussed in Chapters 5 and 6.

Current datasets do not contain representative occlusion data for a quantitative and qualitative evaluation of fall detection algorithms, therefore, new videos are

required of occluded scenes of falls and non-falls. Nevertheless, to avoid the complications of acquiring a new dataset, existing video data from public datasets can be used. One method was the introduction of synthetic occlusions by masking areas of the image. Two types of these synthetic occlusions were introduced; rectangular block areas and images of real furniture. Real furniture have different sizes and most of them will provide partial occlusions such as sofas, chairs and coffee tables. Another approach was simply to define vertical cut-offs denoting the vertical degree of occlusion. This is an efficient and effective means to implement a full occlusion of the person and avoid the need of requiring actors to simulate falls multiple times, behind various occluding pieces of furniture. The degree of occlusion can be quantified, as a ratio of the person's height, allowing a clear and quantitative assessment of when the fall detection fails.

Simulated falls provide continuous data for the duration of each fall event. The vertical location of the top bounding box point is recorded throughout, from the beginning of the fall until the myoskeletal model reaches the ground. In the event of occlusion, the location profile stops at the last measurement before the occlusion. A truncation process (linear extrapolation) predicts the remaining points toward the floor.

The use of simple and complex synthetic occlusions in depth data and the evaluation of fall detection algorithms against the synthetic occluded data is a novel contribution developed in this thesis. This Chapter investigates different types of occlusion and how to simulate them in a real-life scenario. The fall detection algorithm (6.7) described in the previous chapter will be tested against the new data.

7.1 Review

Synthetic occlusions were used in other studies for several purposes with possibly the earliest discussed in [163] for a medical imaging application. Their study discusses the use of occlusion by reducing the pixel intensity on various regions within Magnetic Resonance Imaging (MRI) images etc. In [164], authors propose an occlusion robust face recognition algorithm. Their evaluation uses synthetic occlusions based on a scarf image overlay applied to a public dataset. Recognition rate succeeds for up to 70% occlusion. A more complex synthetic occlusion

modelling is discussed in [165], where a large number of synthetic occlusion configurations was generated for car detection and address the issues of car-to-car occlusions and car self-occlusions. Car type, orientation, relative position and camera view are some of the parameters for the occlusion models. A dataset comprising of images of occluded vehicles is presented in [166]. This dataset was developed to address the detection of semantic parts on partially occluded vehicles. Synthetic occlusions were also modelled for people as in [167] where a large collection of synthetic occlusion data was developed by compositing segmented objects over a base training data set that has been annotated with part locations and figure-ground masks in order for the occlusion to be applied on the correct position. This was approach was applied on a public dataset. Occlusions applied for people were also implemented in [168] where human figures were imported on existing video of public datasets. The derived data can be used for objective evaluation of human tracking algorithms in the presence of occlusion. In [169], authors implement a simple rectangular occlusion over the image to assess the robustness of an ear detection algorithm, for the purpose of biometric analysis. The occlusion degree variated from 0 to 50%. A similar use of synthetic occlusion is found in [170] where authors use a rectangular occlusion which varies in size and location within the image. Their algorithm tests an occlusion-robust face recognition algorithm. Another face recognition study [171] uses synthetic occlusions with various boundaries and various occlusion levels (0 - 90%) over existing public datasets. A 3D head tracking algorithm was evaluated under a sphere shaped occlusion in [172]. Their evaluation videos contain videos of a moving head while the occlusion also changes its position.

The use of synthetic occlusion has key benefits when compared with videos where occlusions already exist. Synthetic occlusions can be used on existing datasets and be applied on variable areas and sizes. Occlusion sizes are much more manageable when the degree of partial occlusion is required for the performance of a detection/tracking algorithm. They can have different shapes, such as vehicles, trees, people etc. The implementation is much faster than recording videos of variable occlusion types.

7.2 Modelling occlusions

Occlusion is one of the issues discussed in Chapter 3 that current datasets fail to address. In a typical home scene, furniture may obscure a camera's visibility and give limited views of people moving around. An occluded fall may be missed if the algorithm is not capable of detecting the event from a partial view of the person. Simple and complex models of occlusion were inserted in video data of existing datasets or applied directly to truncate the head's location data.

7.2.1 Simple Occlusion models

To evaluate fall detections under various occlusion scenarios, the current datasets are augmented by adding synthetic occlusions to the depth videos. The impact of occlusions is limited to only two studies [2, 7] where multiple cameras are used to provide full coverage of a room and cope with occlusions by selecting an occlusion-free viewpoint. In the second referenced work, authors use a bed as a real occlusion which is limited in terms of providing an evaluation measure for assessing the impact of occlusion on fall detection. Their study, evaluates current (of the time of their publication) algorithms under occlusions using 30 occluded falls and 80 occluded ADLs. Their evaluation is poorly presented (e.g. it is quoted that "false positives are more than missed detections") without much of constructive discussion regarding the reason's of the performance of the discussed algorithms.

In detail, occlusions were inserted as rectangular areas and frame-by-frame obscuring the original depth data in the lower portion of the image. Those frames are then reconstructed as a depth video and the full video analysis is applied to the fall detection. Fig. 7.1 shows the types of occlusion applied for the evaluation of fall detection, where the occluded areas are created by masking out the lower portion of the image.

7.2.2 Complex Occlusion models – partial/realistic

Occlusions in a home environment or general in real-life scenes are more complex than the rectangular boxes applied in the previous section. Here we will implement a realistic/complex set of occlusions by using real images of objects found in a home

FIGURE 7.1: Three occlusion modes. The black rectangle in the image is a synthetic occlusion applied to the depth image. The degree of occlusion is expressed proportionally to the person's height: (a) 40%, (b) 50%, (c) 70% occlusion measured from the ground

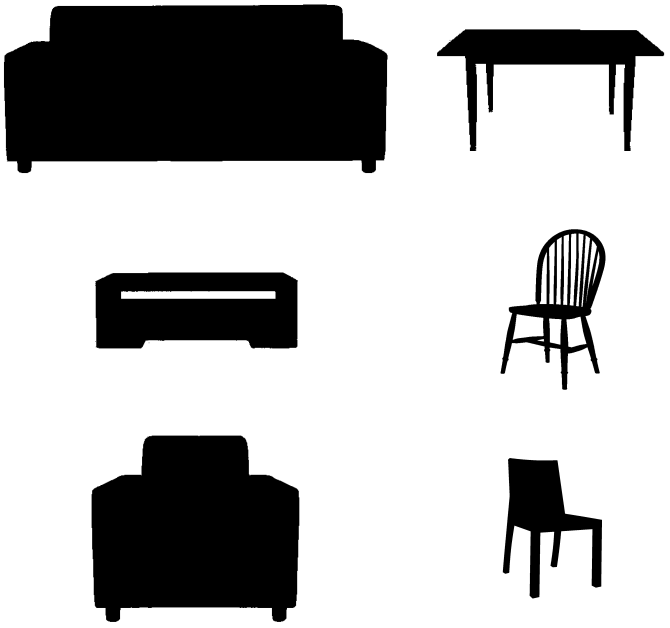

FIGURE 7.2: Segmented images of furniture

such as a chair, sofa and coffee table. Those objects will be scaled and embedded in the depth video in the same manner as the rectangular synthetic occlusions 7.2.1. Every image is segmented in order to contain only the object while the rest of the image is transparent as seen in Fig. 7.2.

The scaling of each object is adjusted according to the person's height in order for the objects to appear as realistic as possible within the scene. The result otherwise would be super-sized or under-sized objects in an unrealistic home scene. For the experiment, the object is placed in the centre of the scene as the fall event is also centred in the view as shown in Fig. 7.3. However the placements of the furniture can be tailored according to any actual room layout. As discussed, these furniture images are placed manually as foreground objects to obscure the fall.

FIGURE 7.3: Complex occlusions from a coffee table a chair and a sofa

7.2.3 Truncated fall measurements

A different approach to create occlusion disturbances on existing video is the interruption of observations (e.g. top bounding box point) at a certain height. The location profile as seen in Fig 7.4, is interrupted at relevant heights to imitate occlusions according to the person's height. In this particular example, the person's height is 1.6 m and the cutoff point for a 50% occlusion is at 0.8m. The degree of occlusion follows the same protocol used in Section 7.2.1 in terms of size (i.e. 40%, 50%, 70%).

7.3 Fall detection under occlusions

When processing human data, the vertical location profile $Y(t)$ of a fall is expected to conclude near the ground, if the whole event is visible. In the event of occlusion, the $Y(t)$ profile will be truncated at the occlusion boundary. Since the $Y(t)$ profile is interrupted due to an occlusion the remaining location profile can be extrapolated to the ground to allow fall detection under occlusion.

Simulation of a fall provides a continuous observation of the fall event from the start to the concluding position of the myskeletal model on the ground. The

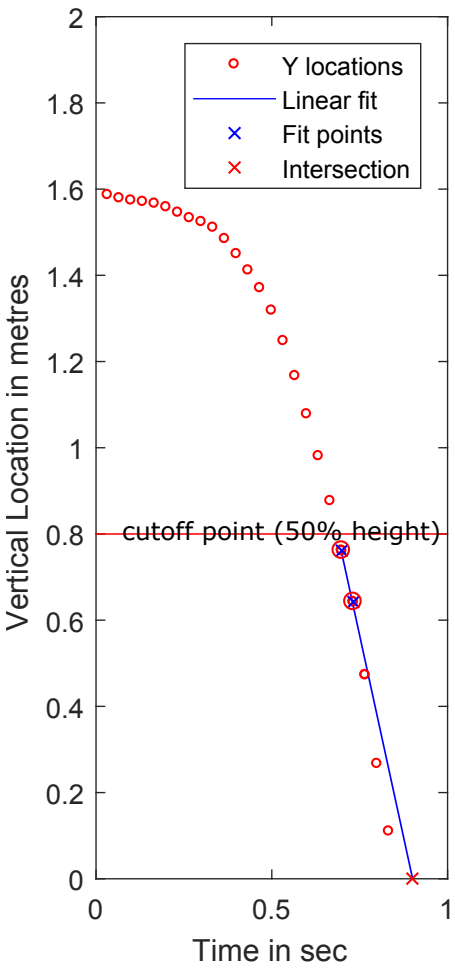

FIGURE 7.4: Truncate Y location of top bounding box point after a cut-off based on 50% occlusion showing extrapolated profile

head location is observed and measured throughout the process. When a fall occurs under occlusion in a real video, the top bounding box location is visible up to certain height. A segment of the visible trajectory ending at the occlusion boundary is identified, containing as many sample points as possible that satisfy a strict linearity condition. The segment is linearly extrapolated towards the ground ($Y_m = 0$) as seen in Fig. 7.5. The new location profile will include those calculated points in order to provide the algorithm with a continuous location profile. In order to assess the robustness of detecting falls under occlusions, a protocol was developed which evaluates the impact of various levels of occlusion, discussed in the next subsection.

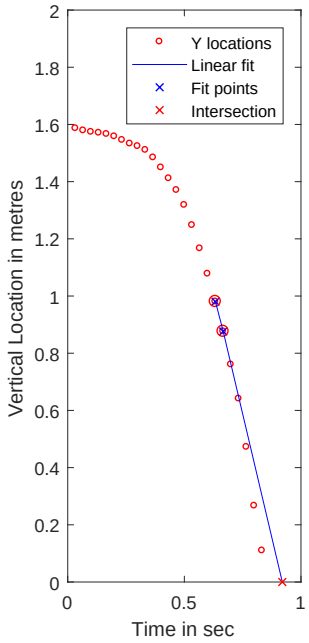

FIGURE 7.5: Location profile estimation in an occluded rigid fall event

7.3.1 Evaluation of algorithm under occlusion

The evaluation protocol using simple synthetic occlusions was applied to **UR** and **GM** datasets consisting of 63 fall and 154 ADL events. Results are summarised in Fig. 7.6 where sensitivity (red line) and specificity are shown to achieve 100% for occlusions up to 50%. Sensitivity drops significantly to 76.5% at 60 % occlusion, with only (30.1%) of the sample detected at 70% occlusion. Specificity drops significantly than faster with falls and is only 43.2 % at 60 % occlusion.

The **SDU** dataset was used for the truncated protocol as it was difficult to split and merge the videos of this dataset. The occlusion results as seen in Fig 7.7 using truncation on the SDU dataset conclude a similar pattern in terms of Sensitivity. Specificity though, remains at 100% on 60% occlusion and holds at 11 % when 70% occlusion degree due to the fact that actions were performed higher (i.e. the camera was located lower than on other datasets at approximately 1.1 m height) from the ground. Also most lying down actions observe participants to first sit on the mattress and then lay down, hence, the person's motion is interrupted. The per dataset Sensitivity and Specificity across **GM**, **UR** and **SDU** datasets is shown in Table 7.1.

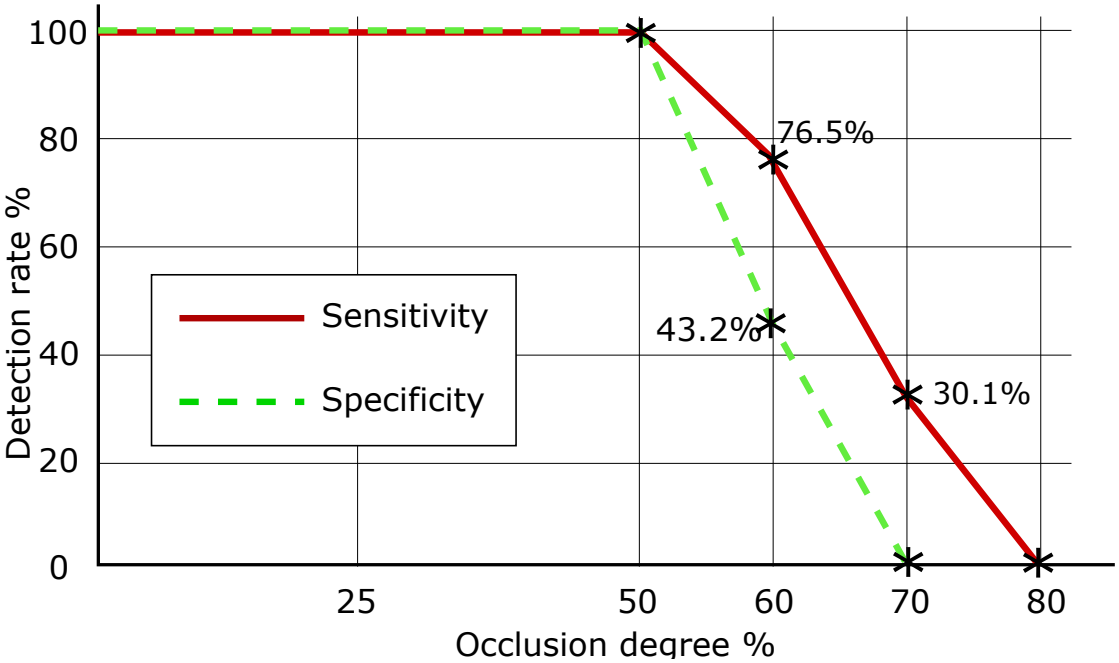

FIGURE 7.6: Results across datasets **UR** and **GM** using simple occlusions. Red line denotes the sensitivity and green line shows the specificity at various occlusion degrees. Note that both fall and ADL events are detected by the myoskeletal approach 6.7 when occlusion degree is at least 50%

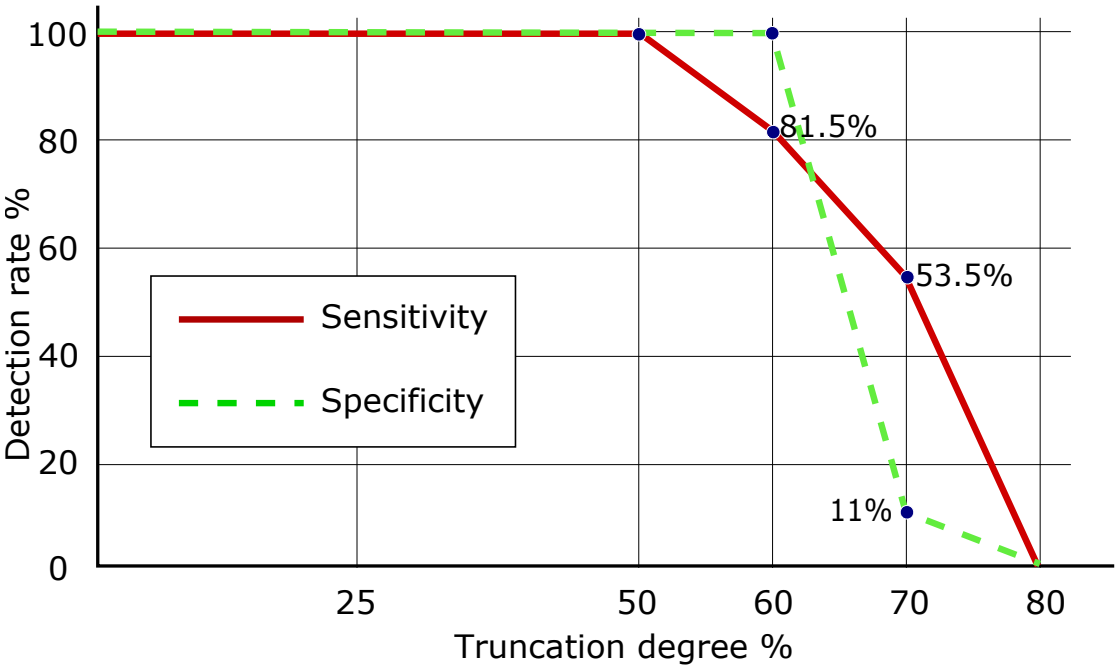

FIGURE 7.7: Results from SDU dataset using truncation of head location. Red line denotes the sensitivity and green line shows the specificity at various occlusion degrees

Table 7.1: Performance of myoskeletal approach 6.7 against in-house and public datasets

	GM		UR		SDU	
Occlusion degree %	Sens	Spec	Sens	Spec	Sens	Spec
50	100	100	100	100	100	100
60	83.4	42.7	69.5	43.7	81.5	100
70	28.6	0	31.6	0	11	53.5
80	0	0	0	0	0 0	0

Data from **UR** and **GM** evaluate complex occlusions, where an object from the list is inserted into each video. Therefore, 63 fall videos have produced 63 cases of occlusions (e.g. 21 examples each of which applied with occlusion from a chair, a sofa and a coffee table) tested against the algorithm. ADL videos were also tested and complex occlusions were inserted. From these complex occlusions, the chair appears to be the tallest object and is expected to cause issues to the algorithm. Nevertheless, the overall size of the chair does not obscure the top bounding box point and using the given examples from these datasets, the falls are detected accurately. In the case of a sofa, the scenario is similar to the simple synthetic occlusions, where the body is occluded by 50% pre-fall and fully post-fall therefore occlusions of these types do not cause missed detections.

It is important to note that in a real home environment, the degree of occlusion associated with furniture depends on the height of the camera, as well as the ratios of distances of the person and the occlusion from the camera. To relate the occlusion degrees to actual objects, one can postulate that 30-40% occlusion is the height of a coffee table, 40-50% a sofa or armchair, a stool, a bed, 50-60% a dining table always depending on the person's height. Having this visual approximation we can say that given this set of examples, the 6.7 algorithm is capable of detecting occluded falls in a typical home scene containing furniture such as coffee tables, desks, setters/sofas, beds etc.

7.4 Conclusion

This chapter introduced the use of synthetic occlusions of rectangular shapes and furniture images to try to reuse existing data in an attempt to assess the robustness of algorithms against occluded scenes.

Synthetic occlusions bridge the gap between the actual home environments and lab scenes where datasets were captured. Several protocols were developed in order to show how the occlusion impacts the detection of a fall according to the size of the body that is occluded. Another protocol uses furniture images such as chairs, tables and sofas to show the impact of occlusion by realistic objects such as furniture.

Last but not least it is demonstrated that the method presented in Section 6.7 is capable of detecting visually occluded falls. A novel evaluation framework based on synthetic occlusions have been proposed in order to establish an understanding of how occlusions impact the detection algorithm. Our method robustly detects falls when body occlusion is up to 50% (measured at a standing position).

Chapter 8

Conclusion

This Chapter summarises the contributions of this thesis, reflecting on the research hypotheses outlined in Chapter 1. Directions for future work are briefly outlined.

8.1 Summary

This thesis explored fall detection methodologies using machine learning and myoskeletal simulation approaches. It has outlined the issues of data availability in data-driven algorithms and the alternative directions. The study also developed a framework in which occlusions can be evaluated. Also, the myoskeletal approach was applied on accelerometer data in order to prove the general use of this method. The following sections specify further these contributions.

8.1.1 Evaluation of data

Chapter 3 discussed several problems with current fall datasets that are not addressed by previous studies in fall detection: limited demographic, small human physical characteristic variation, young age of participants, hesitation, selective fall behaviour.

Nevertheless, a further assessment was required to assess hesitation during falling, particularly for the collapsing falls. The novelty here is the development of an evaluation tool for assessing this hesitation of falls using YouTube fall data. This

is proves how acted falls are unrepresentative of real fall events due to the risk of performing them. The evaluation concluded that the majority of collapsing fall events were poorly performed when compared with YouTube real fall events caused by hyperventilation. The sample of the non-hesitant falls formed the GM2 dataset.

8.1.2 Learning approaches

Two machine learning techniques were developed using depth data. The first approach utilises a 3D bounding box where velocities of expansion and contraction during actions are used to assess a fall. A composition of depth and width velocities was used in order to calculate the horizontal plane velocity. The height velocity together with the composition were measured in real-time during the events and a fall was classified when these velocities exceed thresholds and an afterwards inactivity. A random search algorithm used to learn the thresholds and duration of when these velocities are met. This approach was novel (at the time) and evaluated on a comprehensive depth data dataset which utilised several fall-like ADLs to confuse the classification.

The second learning approach uses a novel feature capable of detecting rigid and collapsing falls. A new bounding box (i.e. the conservative bounding box) was developed which filters out the motion of the arms. This minimises the change of the bounding box when the person moves their arms such as raising arms or extreme walking (e.g. marching), hence is more susceptive to the torso's motion. A new feature based on the bounding box was developed which uses the angle of the two opposite diagonal top edge points of the conservative bounding box and the centre of mass point. The angular velocity is measured that captures the person's motion. A fall is detected if the velocity exceeds a threshold for a number of consecutive frames. Similarly, the angular velocity and frame number values are determined by a random search algorithm.

8.1.3 The use of simulation for fall detection

Several simulation approaches were investigated and developed. Simple fall models such as falling rods were developed in order to imitate a human fall. Parameters such as feet resistance or a two piece rod model where investigated in the fall

modelling. A myoskeletal model was investigated, capable of performing several rigid and collapsing fall types. An evaluation of the different fall types showed that the velocity profile is similar when backward, forward, sideways and collapsing falls are occurring, hence there is no need to represent separately these different fall types.

The rod model did not perform with the same velocity as the myoskeletal when compared with actual YouTube hyperventilation fall events. Also, the rigidity of the rod model would be difficult to adapt to other fall types. The myoskeletal model was evaluated against real-fall data from hyperventilation videos to prove its validity. The use of these videos was novel for the evaluation of fall similarity and fall detection evaluation.

8.1.3.1 Simulation approaches

Two myoskeletal simulation approaches were developed. The first used an approach where falls utilise myoskletal simulation, while for ADLs, human data was used. The hybrid approach uses the standard myoskeletal model to perform three customisable fall types (i.e. forward, backward, sideways). A polynomial regression algorithm is fitted to the vertical velocity profile of the centre of mass. Human fall data was used to test the approach where the CoM is processed for vertical velocity which was fitted with the polynomial derived in the previous step. The fitting error was measured for both fall and ADLs. It is found that there was a linear separation using an SVM between the error values of the falls and ADLs which unambiguously identifies two classes.

The second method utilised only myoskeletal simulation for falls and ADLs, where models in both classes were personalised by the person's height. For the decision of fall or non-fall, a novel method using the Hausdorff distance was used. In this method, the simulated falls and ADLs were compared in order to select the velocity profile which is most similar to the fall. The fall and the most similar (to fall) ADL velocity profile are then compared with every sample from the evaluation datasets to classify falls and ADLs. The personalised fall detector is capable of detecting falls of people with different heights more accurately than existing methods.

An evaluation of different fall types took place in order to show that forward, backward, sideways and collapsing falls have the same vertical velocity profile.

Therefore, the requirement of having different types of simulation falls is diminished. Also, this shows that the data recordings of different types of falls are somehow unnecessary as the falling velocities are very similar, hence one possible type of fall is also enough to evaluate the approach from real data (such as YouTube).

The use of fall modelling requires one myoskeletal simulation to take place for a particular person, without using any learning procedure on human or simulated data. Hence, the algorithm works independently of existing data and allows the modelling of other types of falls which are not available in the current datasets.

8.1.4 Occlusion robustness

The use of simulation for an occlusion robust approach was proposed for the first time in this study. An occlusion robust feature was used, such as the head point, estimated by the top bounding box location, in order to avoid head miss-detections. Similarly, the observation from the myoskeletal model was taken from a marker on the head. The simulation provides the recovered signal which is then compared with the actual human velocity profile.

Linearly interpolating the missing signal in order to complete the falling trajectory of the person is a novelty of the algorithm, resulted in improved performance for occlusions above 50%, when evaluating simple occlusions. The complex (furniture) occlusions were similarly tested using the myoskeletal simulation. These occlusions, although they do not always cover the same area as the simple ones, they simulate a more realistic scenario of occlusion.

8.1.4.1 Occlusion evaluation protocol

An evaluation protocol for occlusions was first used in this study for fall detection. This is also the first time depth data are processed with occlusions. Complex objects were also examined as synthetic occlusions for the first time. By blending square blocks or furniture shaped overlays (e.g. chair, sofa, coffee table, table, couch) on the depth data, the process reuses existing data from depth datasets to create an occluded scene. The size of these occlusions was assigned according to the person's height, in order to measure the degree of occlusion on which the algorithm can detect a fall or an ADL.

8.2 Future Work

8.2.1 Other Fall Types and Scene Simulation

This study outlined the use of myoskeletal simulation models for fall detection and addressed some of the fall types where it can be utilised. As a future direction, the fall modelling would extend to other fall types such as slip, slide, stumble. Falls more common amongst elders or people with mobility issues could be realistically simulated.

More complex scenes can be simulated using OpenSim, with more furniture types or other objects such as floor ornaments, steps or stairs (to climb), slippery surfaces etc. More realistic occlusions (e.g. hospital beds) to investigate the impact of partial occlusion and to evaluate the detector performance on real fall data recorded from hospitals and independent livers, when such data may become available.

8.2.2 Other types of myoskeletal models

This study used the standard (for the hybrid method) and Full Body myoskeletal models. These models are based on the average (perfect) myoskeletal representation. OpenSim has the capabilities to develop other types of body postures, such as the arched back posture of an elder. This will require an elaborative study similar to the one on which the existing models were based.

Alterations on the existing models can imitate the posture of a disabled person where one limb is shorter than the other. Also, the use of prosthetics in the model is possible showing how a person with mobility issues will behave during a fall.

8.2.3 Further personalisation

Alternative body morphologies and other physical characteristics of the faller may be assessed through the modelling. The height of the person was used as a customisable parameter for the simulations in this study. Other physical characteristics could be used, such as weight or posture. As discussed, elders have different physical characteristics such as an arched back which could be modelled. The pre-fall gait is a factor in assessing fragility. Walking speed is a factor which may

increase the fall impact. The footwear type can also be parametrised as is one of the fall factors reported in the literature [89].

8.2.4 Applied to other technologies

Whilst the modelling methods used in this thesis have been applied to depth and accelerometer data, they could also be adapted to other sensor technologies. Calibrated RGB cameras can provide velocity profiles when the head is detected using face recognition or similarly, to use the top bounding box. Sound sensors could correlate the noise of a fall. A Hausdorff comparison of the two spikes from the sound signal and acceleration produced by OpenSim would possibly determine a fall event. Sensors from a mobile phone such as gyroscopes an accelerometers can be used. A proposed algorithm will detect the location of the phone near the body and run a simulation. The recording data from the model will have the samples taken from a point which matches the phone location.

8.3 Epilogue

A reliable and robust fall detector is an important tool in helping to support and maintain the independence of the elderly and the infirm. The goal of this thesis was to make advancements in the field of fall detection. Several data-driven algorithms based on depth data were developed. These algorithms were evaluated against the combined house and public datasets which revealed the need for a personalised approach. Hesitation, lack of fall data (particularly of elderly) and unrepresentative demographic were some of the issues of data-driven approaches (in general). The simulation approach using myoskeletal models overcomes the previous issues data-driven algorithms have and performs better in the occluded scene and on different fall types. It requires the height of the person as the only parameter for personalisation which is the key to outperform existing work.

Such approach can provide support to the vulnerable people at home and promote independent living. The personalisation offered by the simulation approach can be tailored to the individual such as an elder with given physical characteristics. This would increase the performance of the algorithm against this particular user.

Furthermore, the simulation framework can handle the challenges posed by a real home environment (cluttered home).

This thesis has investigated a number of approaches to detecting people falling over, and in particular, distinguish them from actions that may falsely trigger a fall detection (i.e. lying down). It has highlighted the problems of assessing fall detectors using data that is unrepresentative of real events, and proposed the use of simulation to fill the gaps. Fall detectors will be embedded in smart-house technologies which is a growing industry and will become an integral part of life in the near future. These concepts will be required in the emulation of further complex myoskeletal models and will produce an initial step in the development of systems that can make a real difference in supporting vulnerable independent livers.

Bibliography

[1] Imen Charfi, Johel Miteran, Julien Dubois, Mohamed Atri, and Rached Tourki. Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and adaboost-based classification. *Journal of Electronic Imaging*, 22(4):041106–041106, 2013.

[2] Edouard Auvinet, Franck Multon, Alain Saint-Arnaud, Jacqueline Rousseau, and Jean Meunier. Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution. *IEEE Transactions on Information Technology in Biomedicine*, 15(2):290–300, 2011.

[3] Samuele Gasparrini, Enea Cippitelli, Ennio Gambi, Susanna Spinsante, Jonas Wåhslén, Ibrahim Orhan, and Thomas Lindh. Proposal and experimental evaluation of fall detection solution based on wearable and depth data fusion. In *ICT innovations 2015*, pages 99–108. Springer, 2016.

[4] Bogdan Kwolek and Michal Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer methods and programs in biomedicine*, 117(3):489–501, 2014.

[5] Xin Ma, Haibo Wang, Bingxia Xue, Mingang Zhou, Bing Ji, and Yibin Li. Depth-based human fall detection via shape features and improved extreme learning machine. *IEEE journal of biomedical and health informatics*, 18(6): 1915–1922, 2014.

[6] Zhong Zhang, Christopher Conly, and Vassilis Athitsos. A survey on vision-based fall detection. In *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, page 46. ACM, 2015.

[7] Zhong Zhang, Christopher Conly, and Vassilis Athitsos. Evaluating depth-based computer vision methods for fall detection under occlusions. In *International Symposium on Visual Computing*, pages 196–207. Springer, 2014.

[8] Zhongwei Cheng, Lei Qin, Yituo Ye, Qingming Huang, and Qi Tian. Human daily action analysis with multi-view and color-depth data. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 52–61. Springer, 2012.

[9] Chenyang Zhang and Yingli Tian. Rgb-d camera-based daily living activity recognition. *Journal of Computer Vision and Image Processing*, 2(4):12, 2012.

[10] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016.

[11] Hossein Rahmani, Arif Mahmood, Du Q Huynh, and Ajmal Mian. Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition. In *European Conference on Computer Vision*, pages 742–757. Springer, 2014.

[12] B Uğur Töreyin, Yiğithan Dedeoğlu, and A Enis Çetin. Hmm based falling person detection using both audio and video. In *International Workshop on Human-Computer Interaction*, pages 211–220. Springer, 2005.

[13] Vinay Vishwakarma, Chittaranjan Mandal, and Shamik Sural. Automatic detection of human fall in video. In *International conference on pattern recognition and machine intelligence*, pages 616–623. Springer, 2007.

[14] World Health Organization. Falls. http://www.who.int/news-room/fact-sheets/detail/falls, 2018. Accessed 11-April-2018.

[15] Kyu-Jung Kim and James A Ashton-Miller. Biomechanics of fall arrest using the upper extremity: age differences. *Clinical Biomechanics*, 18(4):311–318, 2003.

[16] Ian P Donald and Christopher J Bulpitt. The prognosis of falls in elderly people living at home. *Age and Ageing*, 28(2):121–125, 1999.

[17] Silvia Deandrea, Ersilia Lucenteforte, Francesca Bravi, Roberto Foschi, Carlo La Vecchia, and Eva Negri. Risk factors for falls in community-dwelling older people:" a systematic review and meta-analysis". *Epidemiology*, pages 658–668, 2010.

[18] S Leclercq. In-company same-and low-level falls: From an understanding of such accidents to their prevention. *International journal of industrial ergonomics*, 25(1):59–67, 2000.

[19] Anne Felicia Ambrose, Geet Paul, and Jeffrey M Hausdorff. Risk factors for falls among older adults: a review of the literature. *Maturitas*, 75(1):51–61, 2013.

[20] Sven Heinrich, Kilian Rapp, Ulrich Rissmann, Clemens Becker, and H-H König. Cost of falls in old age: a systematic review. *Osteoporosis international*, 21(6):891–902, 2010.

[21] Pekka Kannus, Jari Parkkari, Seppo Niemi, and Mika Palvanen. Fall-induced deaths among elderly people. *American journal of public health*, 95(3):422–424, 2005.

[22] Gwen Bergen. Falls and fall injuries among adults aged 65 yearsunited states, 2014. *MMWR. Morbidity and mortality weekly report*, 65, 2016.

[23] Florian Feldwieser, Florian Feldwieser, Michael Marchollek, Michael Marchollek, Markus Meis, Markus Meis, Matthias Gietzelt, Matthias Gietzelt, Elisabeth Steinhagen-Thiessen, and Elisabeth Steinhagen-Thiessen. Acceptance of seniors towards automatic in home fall detection devices. *Journal of Assistive Technologies*, 10(4):178–186, 2016.

[24] Carol J Lehtola, William J Becker, and Charles M Brown. Preventing injuries from slips, trips and falls. *Institute of Food and Agriculture Science, University of Florida*, 1990.

[25] Stephen N Robinovitch, Fabio Feldman, Yijian Yang, Rebecca Schonnop, Pet Ming Leung, Thiago Sarraf, Joanie Sims-Gould, and Marie Loughin. Video capture of the circumstances of falls in elderly people residing in long-term care: an observational study. *The Lancet*, 381(9860):47–54, 2013.

[26] Sophie Turgeon Londei, Jacqueline Rousseau, Francine Ducharme, Alain St-Arnaud, Jean Meunier, Jocelyne Saint-Arnaud, and Francine Giroux. An

intelligent videomonitoring system for fall detection at home: perceptions of elderly people. *Journal of telemedicine and telecare*, 15(8):383–390, 2009.

[27] Georgios Mastorakis and Dimitrios Makris. Fall detection system using kinects infrared sensor. *Journal of Real-Time Image Processing*, 9(4):635–646, 2014.

[28] Georgios Mastorakis and D. Makris. Fall detection: types, solutions, weaknesses. a review of computer vision based fall detection systems. In *IET Human Motion Analysis for Healthcare Applications*, 2016.

[29] Georgios Mastorakis, Xavier Hildenbrand, Kevin Grand, and Dimitrios Makris. Customisable fall detection: a hybrid approach using physics based simulation and machine learning. In *1st Workshop on Action and Anticipation for Visual Learning, European Conference on Computer Vision (ECCV)*, 2016.

[30] Georgios Mastorakis, Tim Ellis, and Dimitrios Makris. Fall detection without people: A simulation approach tackling video data scarcity. *Expert Systems with Applications*, pages –, 2018. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2018.06.019. URL `https://www.sciencedirect.com/science/article/pii/S0957417418303658`.

[31] Carolina Maldonado, Homero Ríos, Efrén Mezura-Montes, and Antonio Marin. Feature selection to detect fallen pose using depth images. In *Electronics, Communications and Computers (CONIELECOMP), 2016 International Conference on*, pages 94–100. IEEE, 2016.

[32] Pranesh Vallabh and Reza Malekian. Fall detection monitoring systems: a comprehensive review. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–25, 2017.

[33] Glen Debard, Peter Karsmakers, Mieke Deschodt, Ellen Vlaeyen, Eddy Dejaeger, Koen Milisen, Toon Goedemé, Bart Vanrumste, and Tinne Tuytelaars. Camera-based fall detection on real world data. In *Outdoor and large-scale real-world scene analysis*, pages 356–375. Springer, 2012.

[34] Caroline Rougier, Edouard Auvinet, Jacqueline Rousseau, Max Mignotte, and Jean Meunier. Fall detection from depth map video sequences. In *International Conference on Smart Homes and Health Telematics*, pages 121–128. Springer, 2011.

[35] Leila Panahi and Vahid Ghods. Human fall detection using machine vision techniques on rgb–d images. *Biomedical Signal Processing and Control*, 44: 146–153, 2018.

[36] Lei Yang, Yanyun Ren, and Wenqiang Zhang. 3d depth image analysis for indoor fall detection of elderly people. *Digital Communications and Networks*, 2(1):24–34, 2016.

[37] Lei Yang, Yanyun Ren, Huosheng Hu, and Bo Tian. New fast fall detection method based on spatio-temporal context tracking of head by using depth images. *Sensors*, 15(9):23004–23019, 2015.

[38] Anh Tuan Nghiem, Edouard Auvinet, and Jean Meunier. Head detection using kinect camera and its application to fall detection. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pages 164–169. IEEE, 2012.

[39] Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. Monocular 3d head tracking to detect falls of elderly people. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 6384–6387. IEEE, 2006.

[40] Derek Anderson, James M Keller, Marjorie Skubic, Xi Chen, and Zhihai He. Recognizing falls from silhouettes. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 6388–6391. IEEE, 2006.

[41] Zhen-Peng Bian, Lap-Pui Chau, and Nadia Magnenat-Thalmann. Fall detection based on skeleton extraction. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 91–94. ACM, 2012.

[42] Yie-Tarng Chen, Yu-Ching Lin, and Wen-Hsien Fang. A hybrid human fall detection scheme. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3485–3488. IEEE, 2010.

[43] Rainer Planinc and Martin Kampel. Introducing the use of depth data for fall detection. *Personal and ubiquitous computing*, 17(6):1063–1072, 2013.

[44] Toshio Hori, Yoshifumi Nishida, Hiroshi Aizawa, Shinichi Murakami, and Hiroshi Mizoguchi. Sensor network for supporting elderly care home. In *Sensors, 2004. Proceedings of IEEE*, pages 575–578. IEEE, 2004.

[45] AK Bourke, JV Obrien, and GM Lyons. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & posture*, 26(2):194–199, 2007.

[46] Qiang Li, John A Stankovic, Mark A Hanson, Adam T Barth, John Lach, and Gang Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pages 138–143. IEEE, 2009.

[47] Shehroz S Khan and Babak Taati. Detecting unseen falls from wearable devices using channel-wise ensemble of autoencoders. *Expert Systems with Applications*, 2017.

[48] Jiangpeng Dai, Xiaole Bai, Zhimin Yang, Zhaohui Shen, and Dong Xuan. Mobile phone-based pervasive fall detection. *Personal and ubiquitous computing*, 14(7):633–643, 2010.

[49] Koray Ozcan and Senem Velipasalar. Wearable camera-and accelerometer-based fall detection on portable devices. *IEEE Embedded Systems Letters*, 8 (1):6–9, 2016.

[50] Sungmook Jung, Seungki Hong, Jaemin Kim, Sangkyu Lee, Taeghwan Hyeon, Minbaek Lee, and Dae-Hyeong Kim. Wearable fall detector using integrated sensors and energy devices. *Scientific reports*, 5, 2015.

[51] Mihail Popescu, Yun Li, Marjorie Skubic, and Marilyn Rantz. An acoustic fall detector system that uses sound height information to reduce the false alarm rate. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 4628–4631. IEEE, 2008.

[52] Majd Alwan, Prabhu Jude Rajendran, Steve Kell, David Mack, Siddharth Dalal, Matt Wolfe, and Robin Felder. A smart and passive floor-vibration based fall detector for elderly. In *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, volume 1, pages 1003–1007. IEEE, 2006.

[53] Liang Liu, Mihail Popescu, Marjorie Skubic, Marilyn Rantz, and Paul Cuddihy. An automatic in-home fall detection system using doppler radar signatures. *Journal of Ambient Intelligence and Smart Environments*, 8(4): 453–466, 2016.

[54] Yun Li, KC Ho, and Mihail Popescu. A microphone array system for automatic fall detection. *IEEE Transactions on Biomedical Engineering*, 59(5): 1291–1301, 2012.

[55] Shuai Tao, Mineichi Kudo, and Hidetoshi Nonaka. Privacy-preserved behavior analysis and fall detection by an infrared ceiling sensor network. *Sensors*, 12(12):16920–16936, 2012.

[56] M Daher, AEBE Diab, M El Badaoui El Najjar, M Khalil, and F Charpillet. Elder tracking and fall detection system using smart tiles. *Sensors*, 15800 (1), 2016.

[57] Guodong Feng, Jiechao Mai, Zhen Ban, Xuemei Guo, and Guoli Wang. Floor pressure imaging for fall detection with fiber-optic sensors. *IEEE Pervasive Computing*, 15(2):40–47, 2016.

[58] Rita Cucchiara, Andrea Prati, and Roberto Vezzani. An intelligent surveillance system for dangerous situation detection in home environments. *Intelligenza Artificiale*, 1(1):11–15, 2004.

[59] Nadia Zouba, Bernard Boulay, Francois Bremond, and Monique Thonnat. Monitoring activities of daily living (adls) of elderly based on 3d key human postures. In *ICVW*, pages 37–50. Springer, 2008.

[60] Rita Cucchiara, Andrea Prati, and Roberto Vezzani. A multi-camera vision system for fall detection and alarm generation. *Expert Systems*, 24(5):334–345, 2007.

[61] Andrew Sixsmith and Neil Johnson. A smart sensor to detect the falls of the elderly. *IEEE Pervasive computing*, 3(2):42–47, 2004.

[62] Ge Wu. Distinguishing fall activities from normal activities by velocity characteristics. *Journal of biomechanics*, 33(11):1497–1500, 2000.

[63] Homa Foroughi, Baharak Shakeri Aski, and Hamidreza Pourreza. Intelligent video surveillance for monitoring fall detection of elderly in home environments. In *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on*, pages 219–224. IEEE, 2008.

[64] Velislava Spasova and Ivo Iliev. A survey on automatic fall detection in the context of ambient assisted living systems. *International journal of advanced computer research*, 4(1):94, 2014.

[65] Yueng Santiago Delahoz and Miguel Angel Labrador. Survey on fall detection and fall prevention using wearable and external sensors. *Sensors*, 14 (10):19806–19842, 2014.

[66] Yun Li, Tanvi Banerjee, Mihail Popescu, and Marjorie Skubic. Improvement of acoustic fall detection using kinect depth sensing. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 6736–6739. IEEE, 2013.

[67] Damien Brulin and Estelle Courtial. Multi-sensors data fusion system for fall detection. In *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pages 1–4. IEEE, 2010.

[68] Allesandro Leone, Giovanni Diraco, Cosimo Distante, Pietro Siciliano, Mattia Malfatti, L Gonzo, M Grassi, A Lombardi, G Rescio, P Malcovati, et al. A multi-sensor approach for people fall detection in home environment. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008*, 2008.

[69] Shadi Khawandi and Bassam Daya. Applying neural network architecture in a multi-sensor monitoring system for the elderly. 2012.

[70] Huan-Wen Tzeng, Mei-Yung Chen, and Jai-Yu Chen. Design of fall detection system with floor pressure and infrared image. In *System Science and Engineering (ICSSE), 2010 International Conference on*, pages 131–135. IEEE, 2010.

[71] Henry Rimminen, Juha Lindström, Matti Linnavuo, and Raimo Sepponen. Detection of falls among the elderly by a floor sensor using the electric near field. *IEEE Transactions on Information Technology in Biomedicine*, 14(6): 1475–1476, 2010.

[72] Raul Igual, Carlos Medrano, and Inmaculada Plaza. Challenges, issues and trends in fall detection systems. *Biomedical engineering online*, 12(1):66, 2013.

[73] Natthapon Pannurat, Surapa Thiemjarus, and Ekawit Nantajeewarawat. Automatic fall monitoring: a review. *Sensors*, 14(7):12900–12936, 2014.

[74] Sharwari Kulkarni and Mainak Basu. A review on wearable tri-axial accelerometer based fall detectors. *J. Biomed. Eng. Technol*, 1(3):36–39, 2013.

[75] Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. Robust video surveillance for fall detection based on human shape deformation. *IEEE Transactions on circuits and systems for video Technology*, 21 (5):611–622, 2011.

[76] Lykele Hazelhoff, Jungong Han, et al. Video-based fall detection in the home using principal component analysis. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 298–309. Springer, 2008.

[77] Chien-Liang Liu, Chia-Hoang Lee, and Ping-Min Lin. A fall detection system using k-nearest neighbor classifier. *Expert systems with applications*, 37(10): 7174–7181, 2010.

[78] Derek Anderson, Robert H Luke, James M Keller, Marjorie Skubic, Marilyn Rantz, and Myra Aud. Linguistic summarization of video for fall detection using voxel person and fuzzy logic. *Computer vision and image understanding*, 113(1):80–89, 2009.

[79] Yixiao Yun and Irene Yu-Hua Gu. Human fall detection in videos via boosting and fusing statistical features of appearance, shape and motion dynamics on riemannian manifolds with applications to assisted living. *Computer Vision and Image Understanding*, 148:111–122, 2016.

[80] Thanh-Hai Tran, Thi-Lan Le, Van-Nam Hoang, and Hai Vu. Continuous detection of human fall using multimodal features from kinect sensors in scalable environment. *Computer Methods and Programs in Biomedicine*, 2017.

[81] Raul Igual, Carlos Medrano, and Inmaculada Plaza. A comparison of public datasets for acceleration-based fall detection. *Medical engineering and physics*, 37(9):870–878, 2015.

[82] Lars Schwickert, C Becker, U Lindemann, C Maréchal, A Bourke, L Chiari, JL Helbostad, W Zijlstra, K Aminian, C Todd, et al. Fall detection with body-worn sensors. *Zeitschrift für Gerontologie und Geriatrie*, 46(8):706–719, 2013.

[83] Tong Zhang, Jue Wang, Ping Liu, and Jing Hou. Fall detection by embedding an accelerometer in cellphone and using kfd algorithm. *International Journal of Computer Science and Network Security*, 6(10):277–284, 2006.

[84] Chien-Cheng Lan, Ya-Hsin Hsueh, and Rong-Yuan Hu. Real-time fall detecting system using a tri-axial accelerometer for home care. In *Biomedical Engineering and Biotechnology (iCBEB), 2012 International Conference on*, pages 1077–1080. IEEE, 2012.

[85] Angela Sucerquia, José David López, and Jesús Francisco Vargas-Bonilla. Real-life/real-time elderly fall detection with a triaxial accelerometer. *Sensors*, 18(4):1101, 2018.

[86] Gregory Koshmak, Amy Loutfi, and Maria Linden. Challenges and issues in multisensor fusion approach for fall detection. *Journal of Sensors*, 2016, 2016.

[87] Hammadi Nait-Charif and Stephen J McKenna. Activity summarisation and fall detection in a supportive home environment. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 323–326. IEEE, 2004.

[88] D Alex Butler, Shahram Izadi, Otmar Hilliges, David Molyneaux, Steve Hodges, and David Kim. Shake'n'sense: reducing interference for overlapping structured light depth cameras. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1933–1936. ACM, 2012.

[89] Jennifer L Kelsey, Elizabeth Procter-Gray, Uyen-Sa DT Nguyen, Wenjun Li, Douglas P Kiel, and Marian T Hannan. Footwear and falls in the home among older individuals in the mobilize boston study. *Footwear science*, 2 (3):123–129, 2010.

[90] Mary E Tinetti, T Franklin Williams, and Raymond Mayewski. Fall risk index for elderly patients based on number of chronic disabilities. *The American journal of medicine*, 80(3):429–434, 1986.

[91] Minoru Yamada, Hajime Takechi, Shuhei Mori, Tomoki Aoyama, and Hidenori Arai. Global brain atrophy is associated with physical performance and the risk of falls in older adults with cognitive impairment. *Geriatrics & gerontology international*, 13(2):437–442, 2013.

[92] Stephen R Lord and Julia Dayhew. Visual risk factors for falls in older people. *Journal of the American Geriatrics Society*, 49(5):508–515, 2001.

[93] Carolyn Wallace, Gayle E Reiber, Joseph LeMaster, Douglas G Smith, Katrina Sullivan, Shane Hayes, and Christy Vath. Incidence of falls, risk factors for falls, and fall-related fractures in individuals with diabetes and a prior foot ulcer. *Diabetes care*, 25(11):1983–1986, 2002.

[94] Sirpa Hartikainen, Eija Lönnroos, and Kirsti Louhivuori. Medication as a risk factor for falls: critical systematic review. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 62(10):1172–1181, 2007.

[95] Julie D Moreland, Julie A Richardson, Charlie H Goldsmith, and Catherine M Clase. Muscle weakness and falls in older adults: a systematic review and meta-analysis. *Journal of the American Geriatrics Society*, 52(7):1121–1129, 2004.

[96] Hennie CJP Janssen, Monique M Samson, and Harald JJ Verhaar. Vitamin d deficiency, muscle function, and falls in elderly people. *The American journal of clinical nutrition*, 75(4):611–615, 2002.

[97] Aner Weiss, Marina Brozgol, Moran Dorfman, Talia Herman, Shirley Shema, Nir Giladi, and Jeffrey M Hausdorff. Does the evaluation of gait quality during daily life provide insight into fall risk? a novel approach using 3-day accelerometer recordings. *Neurorehabilitation and neural repair*, 27(8): 742–752, 2013.

[98] Jürgen Härlein, Theo Dassen, Ruud JG Halfens, and Cornelia Heinze. Fall risk factors in older people with dementia or cognitive impairment: a systematic review. *Journal of advanced nursing*, 65(5):922–933, 2009.

[99] Etsuo Horikawa, Toshifumi Matsui, Hiroyuki Arai, Takashi SEKI, Koh IWASAKI, and Hidetada SASAKI. Risk of falls in alzheimers disease: a prospective study. *Internal Medicine*, 44(7):717–721, 2005.

[100] Yi-Chung Pai and James Patton. Center of mass velocity-position predictions for balance control. *Journal of biomechanics*, 30(4):347–354, 1997.

[101] STEREOLABS. Zed stereo camera. https://www.stereolabs.com/, 2016. Accessed 19 May 2017.

[102] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013.

[103] Occipital. Openni, 2016. URL `http://structure.io/openni`. [Last Accessed 05-April-2016].

[104] Wikipedia. Primesense, 2013. URL `https://en.wikipedia.org/wiki/PrimeSense`. [Online; accessed 05-April-2016].

[105] Zhong Zhang, Weihua Liu, Vangelis Metsis, and Vassilis Athitsos. A viewpoint-independent statistical method for fall detection. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3626–3630. IEEE, 2012.

[106] Angela Sucerquia, José David López, and Jesús Francisco Vargas-Bonilla. Sisfall: A fall and movement dataset. *Sensors*, 17(1):198, 2017.

[107] WebMD. Knee problems and injuries. https://www.webmd.com/pain-management/knee-pain/knee-problems-and-injuries-topic-overview, 2018. Accessed 11-Feb-2018.

[108] Charles E Clauser, John T McConville, and John W Young. Weight, volume, and center of mass of segments of the human body. Technical report, ANTIOCH COLL YELLOW SPRINGS OH, 1969.

[109] Fairouz Merrouche and Nadia Baha. Depth camera based fall detection using human shape and movement. In *Signal and Image Processing (ICSIP), IEEE International Conference on*, pages 586–590. IEEE, 2016.

[110] Erdem Akagündüz, Muzaffer Aslan, Abdulkadir Şengür, Haibo Wang, and Melih Cevdet İnce. Silhouette orientation volumes for efficient fall detection

in depth videos. *IEEE journal of biomedical and health informatics*, 21(3): 756–763, 2017.

[111] LA Rastrigin. The convergence of the random search method in the extremal control of a many parameter system. *Automaton & Remote Control*, 24: 1337–1342, 1963.

[112] Giovanni Diraco, Alessandro Leone, and Pietro Siciliano. An active vision system for fall detection and posture recognition in elderly healthcare. In *Proceedings of the conference on design, automation and test in Europe*, pages 1536–1541. European Design and Automation Association, 2010.

[113] Bart Jansen, Frederik Temmermans, and Rudi Deklerck. 3d human pose recognition for home monitoring of elderly. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4049–4051. IEEE, 2007.

[114] Bessam Abdulrazak, Sylvain Giroux, Bruno Bouchard, Mounir Mokhtari, and Hélène Pigot. *Towards Useful Services for Elderly and People with Disabilities: 9th International Conference on Smart Homes and Health Telematics, ICOST 2011, Montreal, Canada, June 20-22, 2011, Proceedings*, volume 6719. Springer Science & Business Media, 2011.

[115] Bart Jansen and Rudi Deklerck. Context aware inactivity recognition for visual fall detection. In *Pervasive Health Conference and Workshops, 2006*, pages 1–4. IEEE, 2006.

[116] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, pages 726–740. Elsevier, 1987.

[117] Yijun Xiao, Paul Siebert, and Naoufel Werghi. A discrete reeb graph approach for the segmentation of human body scans. In *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, pages 378–385. IEEE, 2003.

[118] Jun Zhao, Jayantha Katupitiya, and James Ward. Global correlation based ground plane estimation using v-disparity image. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 529–534. IEEE, 2007.

[119] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. Ieee, 2011.

[120] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *Advances in neural information processing systems*, pages 1569–1576, 2005.

[121] Gary Bishop, Greg Welch, et al. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41, 2001.

[122] Werner Dubitzky, Martin Granzow, and Daniel P Berrar. *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media, 2007.

[123] Mary E Tinetti and Christianna S Williams. Falls, injuries due to falls, and the risk of admission to a nursing home. *New England journal of medicine*, 337(18):1279–1284, 1997.

[124] Ronald F Kirby and John A Roberts. *Introductory biomechanics*. Mouvement Pubns, 1985.

[125] Yujiang Xiang, Jasbir S Arora, and Karim Abdel-Malek. Physics-based modeling and simulation of human walking: a review of optimization-based and other approaches. *Structural and Multidisciplinary Optimization*, 42(1): 1–23, 2010.

[126] Marcus A Brubaker, David J Fleet, and Aaron Hertzmann. Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, 87(1-2):140, 2010.

[127] Dimitris Metaxas and Demetri Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

[128] L Kakadiaris and Dimitris Metaxas. Model-based estimation of 3d human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1453–1459, 2000.

[129] Damien Jade Duff, Thomas Mörwald, Rustam Stolkin, and Jeremy Wyatt. Physical simulation for monocular 3d model based tracking. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5218–5225. IEEE, 2011.

[130] Marek Vondrak, Leonid Sigal, Jessica Hodgins, and Odest Jenkins. Video-based 3d motion capture through biped control. *ACM Transactions On Graphics (TOG)*, 31(4):27, 2012.

[131] Nadia Zouba, Francois Bremond, Monique Thonnat, and Van Thinh Vu. Multi-sensors analysis for everyday activity monitoring. *Proc. of SETIT*, pages 25–29, 2007.

[132] Wenbin Li, Seyedmajid Azimi, Aleš Leonardis, and Mario Fritz. To fall or not to fall: A visual approach to physical stability prediction. *arXiv preprint arXiv:1604.00066*, 2016.

[133] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[134] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.

[135] Hadi Keivan Ekbatani, Oriol Pujol, and Santi Seguí. Synthetic data generation for deep learning in counting pedestrians. In *ICPRAM*, pages 318–323, 2017.

[136] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.

[137] Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman, and Darryl G Thelen. Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering*, 54(11):1940–1950, 2007.

[138] OpenSim. Intro to musculoskeletal modeling, 2017. URL `https://simtk-confluence.stanford.edu/display/OpenSim/Tutorial+1+-+Intro+to+Musculoskeletal+Modeling`. [Online; accessed 11-April-2016].

[139] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007.

[140] Frank C Anderson, Chand T John, Eran Guendelman, Allison S Arnold, and Scott L Delp. Simtrack: Software for rapidly generating muscle-actuated simulations of long-duration movement, 2006.

[141] Felipe Alvim. Convert .c3d and .csv files to opensim. https://simtk.org/projects/c3d2opensim, 2016. Accessed 1 May 2017.

[142] OpenSim. Intro to musculoskeletal modeling. https://simtk-confluence.stanford.edu/display/OpenSim/, 2017. Accessed 11-April-2016.

[143] PM Lührmann, R Bender, B Edelmann-Schäfer, and M Neuhäuser-Berthold. Longitudinal changes in energy expenditure in an elderly german population: a 12-year follow-up. *European journal of clinical nutrition*, 63(8):986, 2009.

[144] Piotr Chmielewski, Krzysztof Borysławski, Krzysztof Chmielowiec, and Jolanta Chmielowiec. Height loss with advancing age in a hospitalized population of polish men and women: magnitude, pattern and associations with mortality. *Anthropological Review*, 78(2):157–168, 2015.

[145] Wikipedia contributors. List of average human height worldwide — wikipedia, the free encyclopedia, 2018. URL `https://en.wikipedia.org/w/index.php?title=List_of_average_human_height_worldwide&oldid=825528696`. [Online; accessed 15-February-2018].

[146] LENORE J LAUNER, TAMARA HARRIS, Ad Hoc Committee on the Statistics of Anthropometry, and Aging. Weight, height and body mass index distributions in geographically and ethnically diverse samples of older persons. *Age and Ageing*, 25(4):300–306, 1996.

[147] The Daily Dot. The pass-out challenge has teens knocking themselves out on purpose. https://www.dailydot.com/irl/pass-out-challenge/, 2014. Accessed 23 Nov 2017.

[148] Roger Y Tsai. An efficient and accurate camera calibration technique for 3d machine vision. *Proc. of Comp. Vis. Patt. Recog.*, pages 364–374, 1986.

[149] Shomir Chaudhuri, Hilaire Thompson, and George Demiris. Fall detection devices and their use with older adults: a systematic review. *Journal of geriatric physical therapy (2001)*, 37(4):178, 2014.

[150] Lingmei Ren and Weisong Shi. Chameleon: personalised and adaptive fall detection of elderly people in home-based environments. *International Journal of Sensor Networks*, 20(3):163–176, 2016.

[151] Wu Liu, Yongdong Zhang, Sheng Tang, Jinhui Tang, Richang Hong, and Jintao Li. Accurate estimation of human body orientation from rgb-d sensors. *IEEE Transactions on Cybernetics*, 43(5):1442–1452, 2013.

[152] Laurent Fitte-Duval, Alhayat Ali Mekonnen, and Frédéric Lerasle. Combination of rgb-d features for head and upper body orientation classification. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 591–603. Springer, 2016.

[153] Jinyoung Choi, Beom-Jin Lee, and Byoung-Tak Zhang. Human body orientation estimation using convolutional neural network. *arXiv preprint arXiv:1609.01984*, 2016.

[154] Mark A Livingston, Jay Sebastian, Zhuming Ai, and Jonathan W Decker. Performance measurements for the microsoft kinect skeleton. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pages 119–120. IEEE, 2012.

[155] Imran N Junejo and Hassan Foroosh. Euclidean path modeling for video surveillance. *Image and Vision computing*, 26(4):512–528, 2008.

[156] Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata, and Alfredo Pulvirenti. Similarity measures and dimensionality reduction techniques for time series data mining. In *Advances in data mining knowledge discovery and applications*. InTech, 2012.

[157] Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang, and Carola Wenk. Fréchet distance for curves, revisited. In *European Symposium on Algorithms*, pages 52–63. Springer, 2006.

[158] William S Cleveland. Lowess: A program for smoothing scatterplots by robust locally weighted regression. *American Statistician*, 35(1):54, 1981.

[159] Apoorva Rajagopal, Christopher L Dembia, Matthew S DeMers, Denny D Delp, Jennifer L Hicks, and Scott L Delp. Full-body musculoskeletal model for muscle-driven simulation of human gait. *IEEE Transactions on Biomedical Engineering*, 63(10):2068–2079, 2016.

[160] Neila Mezghani, Youssef Ouakrim, Md R Islam, Rami Yared, and Bessam Abdulrazak. Context aware adaptable approach for fall detection bases on smart textile. In *Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on*, pages 473–476. IEEE, 2017.

[161] Sahar Abdelhedi, Riad Bourguiba, Jaouhar Mouine, and Mouna Baklouti. Development of a two-threshold-based fall detection algorithm for elderly health monitoring. In *Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on*, pages 1–5. IEEE, 2016.

[162] LP Nguyen, M Saleh, and R Le Bouquin Jeannès. An efficient design of a machine learning-based elderly fall detector. In *International Conference on IoT Technologies for HealthCare*, pages 34–41. Springer, 2017.

[163] Amit Chakraborty, Lawrence H Staib, and James S Duncan. An integrated approach to boundary finding in medical images. In *Biomedical Image Analysis, 1994., Proceedings of the IEEE Workshop on*, pages 13–22. IEEE, 1994.

[164] Zihan Zhou, Andrew Wagner, Hossein Mobahi, John Wright, and Yi Ma. Face recognition with contiguous occlusion using markov random fields. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1050–1057. IEEE, 2009.

[165] Tianfu Wu, Bo Li, and Song-Chun Zhu. Learning and-or model to represent context and occlusion for car detection and viewpoint estimation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1829–1843, 2016.

[166] Jianyu Wang, Cihang Xie, Zhishuai Zhang, Jun Zhu, Lingxi Xie, and Alan Yuille. Detecting semantic parts on partially occluded objects. *British Machine Vision Conference*, 2017.

[167] Golnaz Ghiasi, Yi Yang, Deva Ramanan, and Charless C Fowlkes. Parsing occluded people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2401–2408, 2014.

[168] Aditi Roy, Pratik Chattopadhyay, Shamik Sural, Jayanta Mukherjee, and Gerhard Rigoll. Modelling, synthesis and characterisation of occlusion in videos. *IET Computer Vision*, 9(6):821–830, 2015.

[169] Banafshe Arbab-Zavar and Mark S Nixon. On shape-mediated enrolment in ear biometrics. In *International Symposium on Visual Computing*, pages 549–558. Springer, 2007.

[170] Hyun Jun Oh, Kyoung Mu Lee, and Sang Uk Lee. Occlusion invariant face recognition using selective local non-negative matrix factorization basis images. *Image and Vision computing*, 26(11):1515–1523, 2008.

[171] Xiao-Xin Li, Dao-Qing Dai, Xiao-Fei Zhang, and Chuan-Xian Ren. Structured sparse error coding for face recognition with occlusion. *IEEE transactions on image processing*, 22(5):1889–1900, 2013.

[172] Ye Zhang and Chandra Kambhamettu. 3d head tracking under partial occlusion. *Pattern Recognition*, 35(7):1545–1557, 2002.

Appendix A

AppendixA

A.1 Visual results of Algorithm 1 from Chapter 4

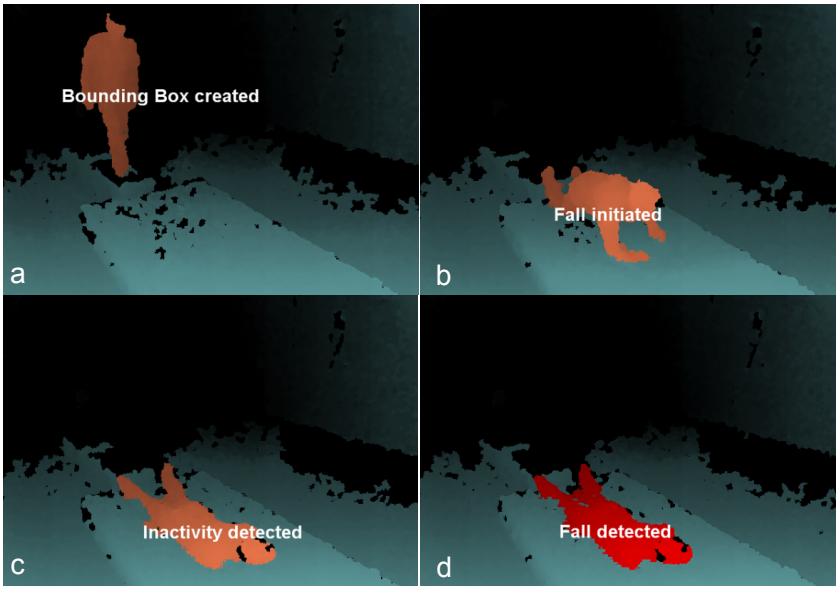

FIGURE A.1: Fall detected in a front (angled) view

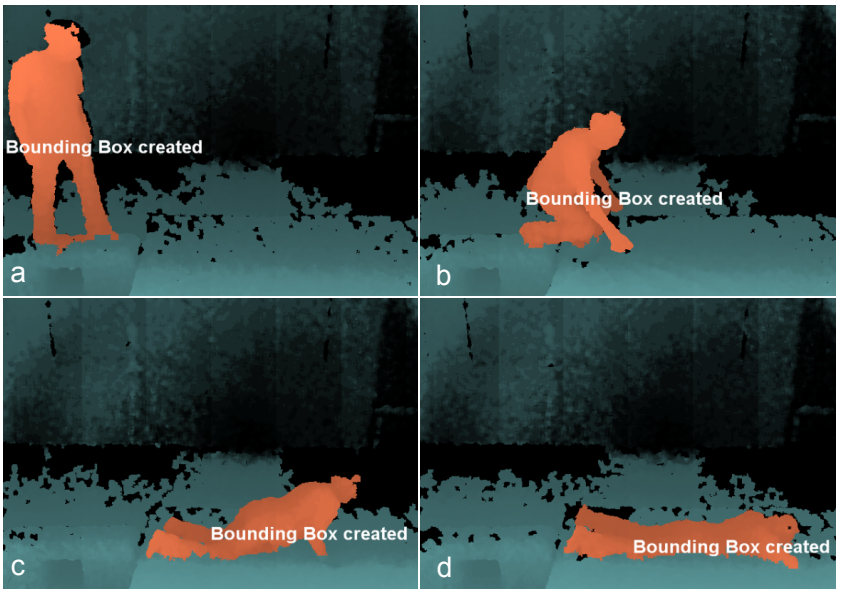

FIGURE A.2: Lying on the floor

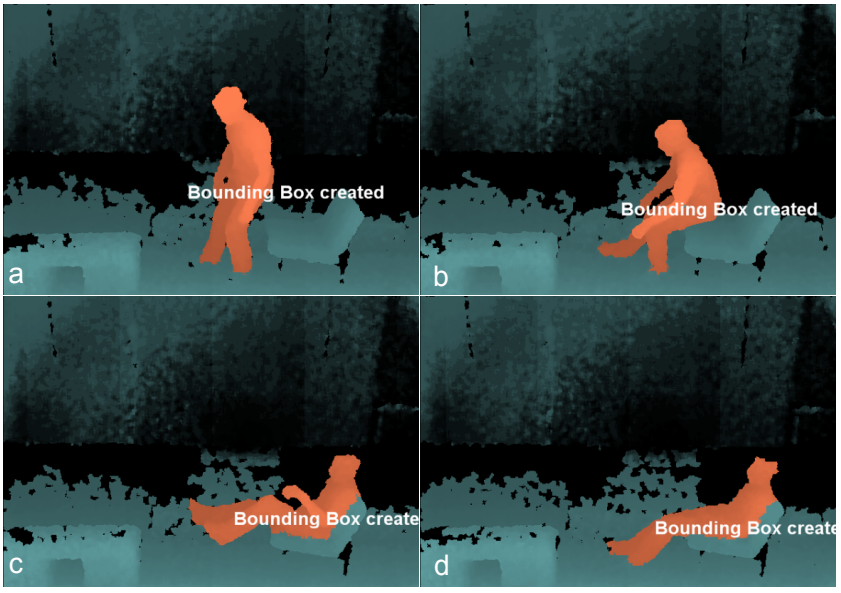

FIGURE A.3: Sitting vigorously on a chair

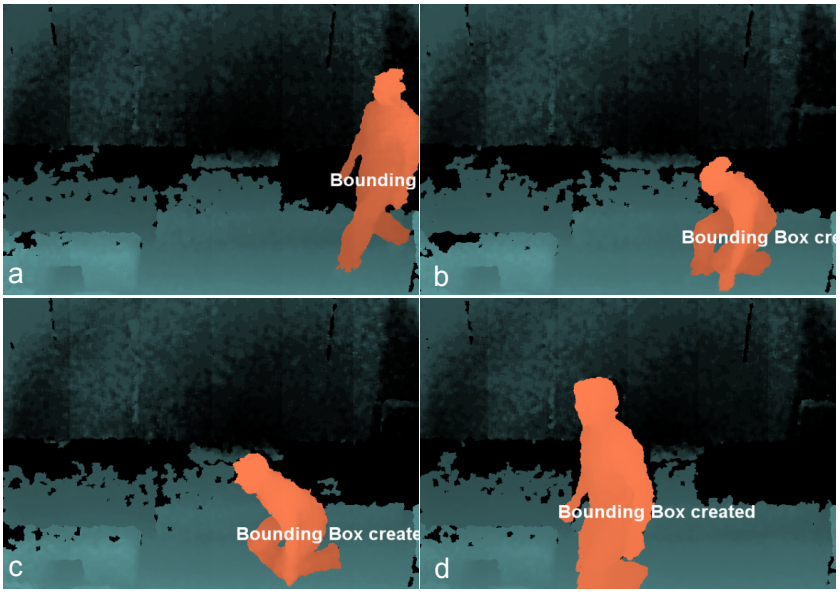

FIGURE A.4: Picking up an item from the floor in fast motion

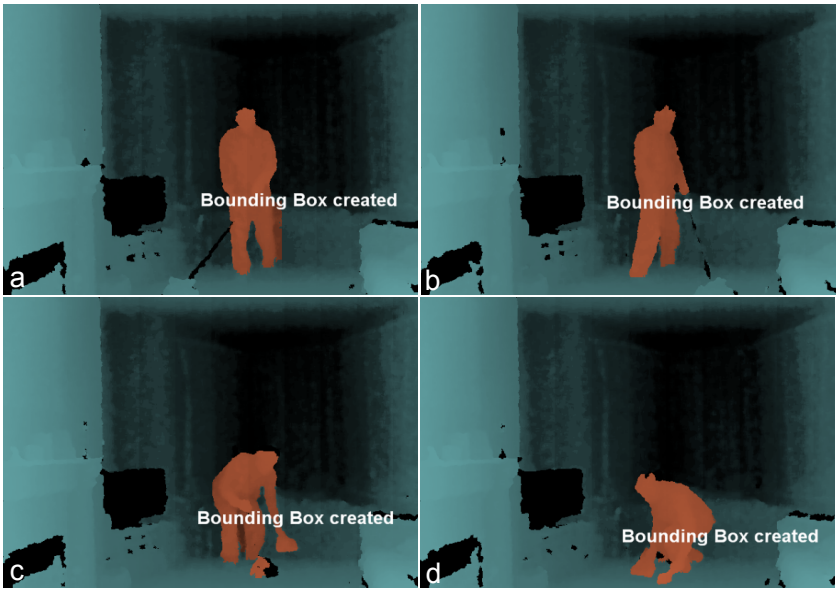

FIGURE A.5: Sweeping activity

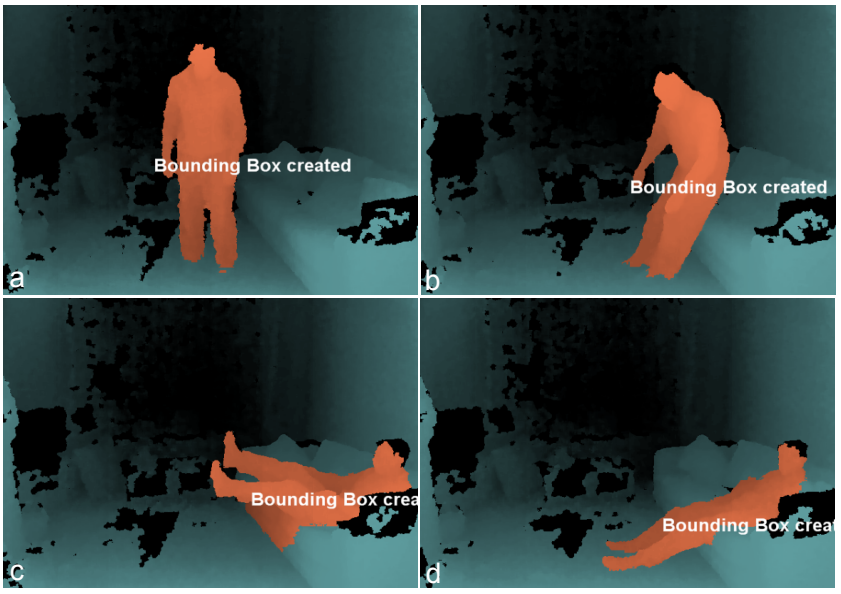

FIGURE A.6: Vigorously sitting on a sofa

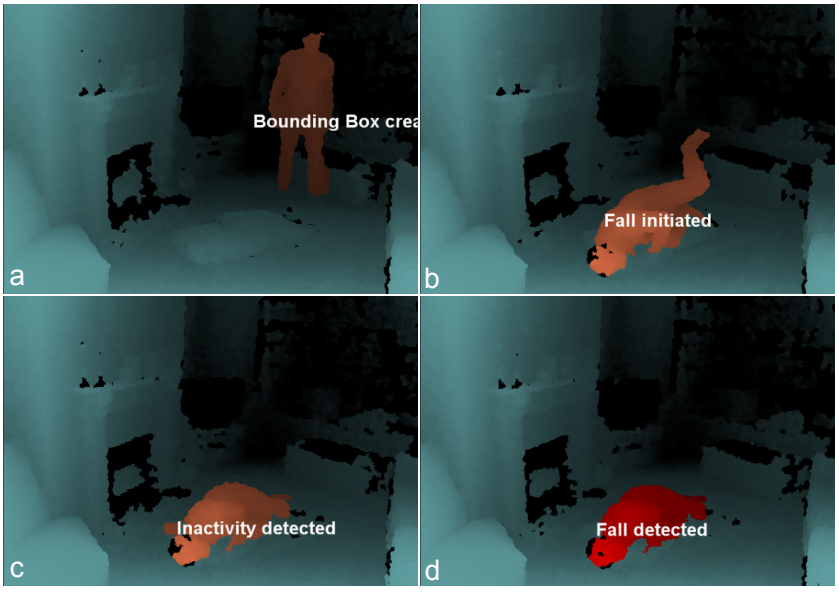

FIGURE A.7: Sideways fall towards the sensor

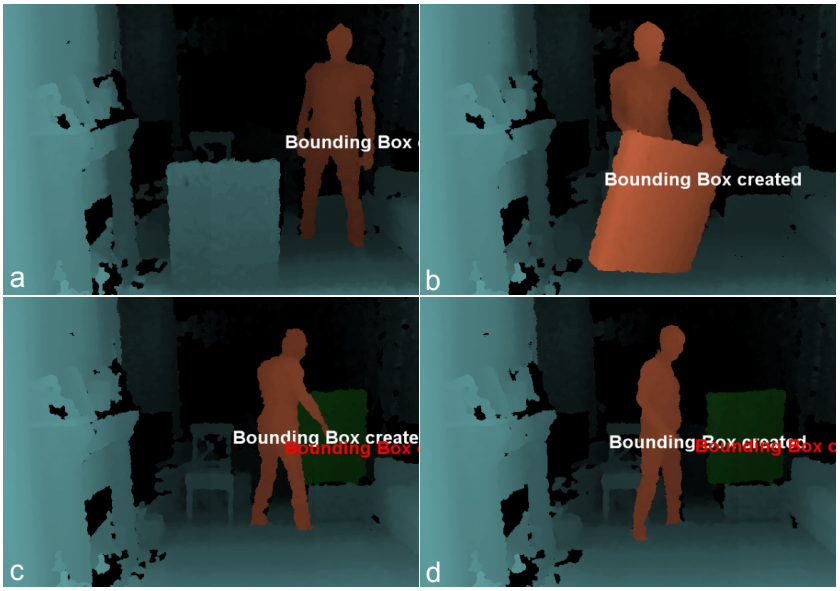

FIGURE A.8: Picking up and dropping a box

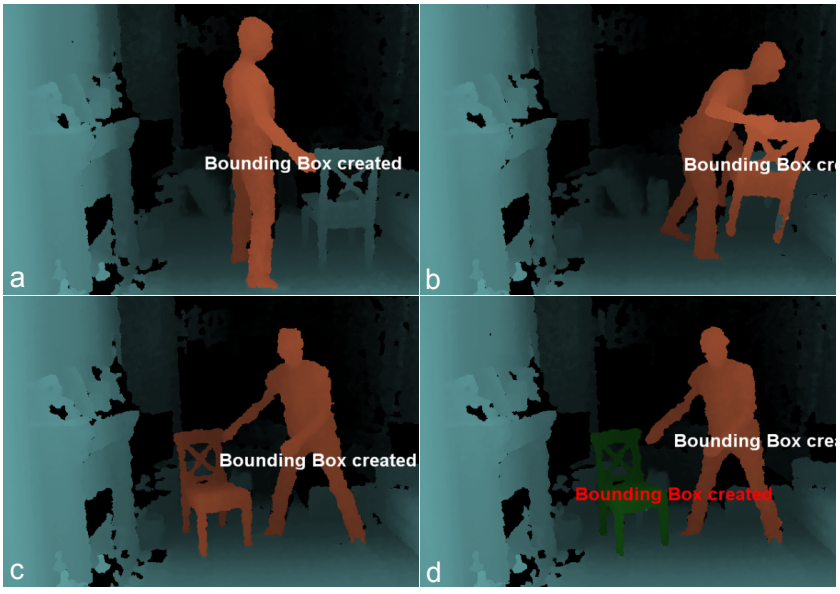

FIGURE A.9: Picking up and dropping a chair

Appendix B

AppendixB

B.1 Feet resistance rod model

As the previous model produces velocities which do not apply to real-data (see Section 5.4), additional parameters are introduced. A "feet" force is introduced in the rod model with the aim of delaying the onset of a fall. It is possible to estimate this force using position and force data, together with some moment calculations.

Fig. B.1 (a) shows a free body diagram of the foot of a person balancing on one leg. We use this assumption since our rod model has one contact point with the ground. The forces acting on the foot and their locations relative to the ankle joint are illustrated. Force $\vec{R}$ is the reaction force of the ground in response to the weight of the person. Force mg is the weight of the foot, where m is the mass of the foot and g is the acceleration due to gravity. These data can be used to estimate the muscle moment that must be produced by the calf muscles for equilibrium to be maintained [108] (i.e. for a person to stay balanced).

If the distance from the ankle joint centre to the foot centre of gravity is measured as 0.05 m and the distance from the ankle joint centre to the reaction force $\vec{R}$ is 0.20 m, then it is possible to calculate the moment about the ankle joint centre.
Taking moments about the ankle joint centre:
Ma = (0.20 x $\vec{R}$) (0.05 x mg)

The force $\vec{R}$ is the weight of the subject. If the mass of the person is 78 kg and the mass of the foot is 1.5% of the full body mass according to [108] - that is 1.17

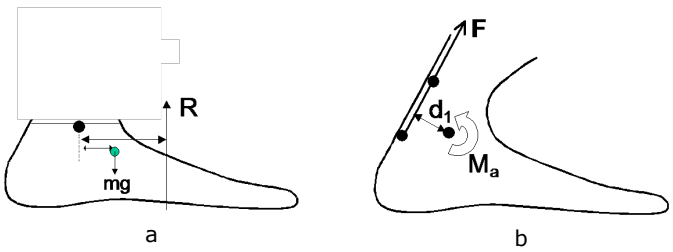

FIGURE B.1: Feet force schematic

kg, the muscle moment about the ankle joint that is required for equilibrium to be maintained can be calculated by:
Ma = - (0.20 x 780) + (0.05 x 11.7) = 156 + 0.585 = 156.585 N.m
This ankle joint moment is generated by the force of the calf muscles, shown as F in Fig. B.1 (b). If the distance from the ankle joint centre to the force $\vec{F}$ is measured, then $\vec{F}$ can be calculated:
Ma = 156.585 = $d_1 F$
If d_1 = 0.04 m, then $\vec{F}$ = 156.585 / 0.04 = 3914.6 N
This force can also be calculated as multiples of bodyweight: 3914.6/780 $\approx$ 5.0 times bodyweight.

Thus Eq. 5.1 becomes:

$$\omega_{n+1} = \sqrt{\omega_n^2 + \frac{3g(\sin(\theta_n)(-F_{feet}) - \sin(\theta_{n+1}))}{L}} \tag{B.1}$$

where F_{feet} is the feet force applied to the rod model for time t and then F_{feet} equals 0 for the duration of the fall.

A complication with applying this model is the duration which this force has to be applied and also, whether this force reduces over time as the rod is falling. Another complication the rate this force reduces linearly, or non-linearly. All these complications made this model difficult to use as these parameters are unknown from the human body perspective.

B.2 Two piece rod models

We can imagine that a human is falling forwards. The human body may have an inclination of the upper body towards the floor. Using a two-piece rod model that

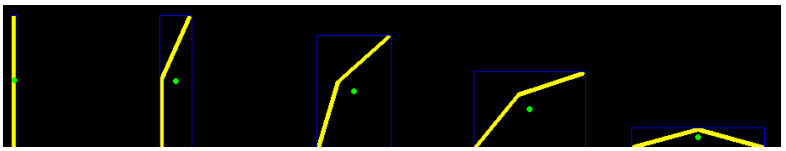

FIGURE B.2: Two piece rigid fall model

is straight at the start of the fall and bends during the fall at a certain angle of 30 degrees. The CoM is located outside this model, hence, there is a different falling velocity. The calculation of ω remains the same using Eq. 5.1, but the CoM point is outside the rod as seen in Fig. B.2.

Similarly as to the previous model, it is difficult to determine the parameters of the model. The bend angle, and when is enabled/disabled or what is its value requires proper justification. Therefore, this models was not evaluated further.

Appendix C

AppendixC

C.1 Published Work

Downloand publications using the links found in page xxiii