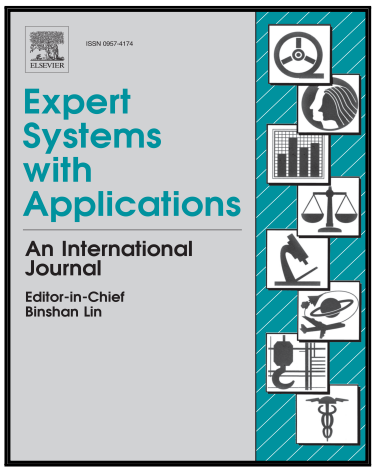

Fall Detection without People: A Simulation Approach Tackling Video Data Scarcity

Georgios Mastorakis, Tim Ellis, Dimitrios Makris

PII: S0957-4174(18)30365-8
DOI: 10.1016/j.eswa.2018.06.019
Reference: ESWA 12013

To appear in: *Expert Systems With Applications*

Received date: 7 December 2017
Revised date: 25 April 2018
Accepted date: 7 June 2018

Please cite this article as: Georgios Mastorakis, Tim Ellis, Dimitrios Makris, Fall Detection without People: A Simulation Approach Tackling Video Data Scarcity, *Expert Systems With Applications* (2018), doi: 10.1016/j.eswa.2018.06.019

Highlights

- Fall detection system based on a myoskeletal (physics-based) simulation
- No need for video recordings of human falls
- Persons height is used to parameterise the simulation, addressing human variability
- State-of-the-art performance, tested in publicly available datasets
- System is robust on occlusions for up to 50

Title

Fall Detection without People: A Simulation Approach Tackling Video Data Scarcity

Author names and affiliations

Georgios Mastorakis (g.mastorakis@ieee.org), Tim Ellis (t.ellis@kingston.ac.uk), Dimitrios Makris(d.makris@kingston.ac.uk)
Kingston University London

Corresponding author

Georgios Mastorakis, g.mastorakis@ieee.org, +44 (0) 7917417446 Faculty of Science, Engineering and Computing Kingston University London — Penrhyn Road — Kingston-upon-Thames, KT1 2EE, UK

Present/permanent address

Faculty of Science, Engineering and Computing Kingston University London — Penrhyn Road — Kingston-upon-Thames, KT1 2EE, UK

Fall Detection without People: A Simulation Approach Tackling Video Data Scarcity

Georgios Mastorakis[1], Tim Ellis, Dimitrios Makris

DIRC, Faculty of Science, Engineering and Computing, Kingston University, London

Abstract

We propose an intelligent system to detect human fall events using a physics-based myoskeletal simulation, detecting falls by comparing the simulation with a fall velocity profile using the Hausdorff distance. Previous methods of fall detection are trained using recordings of acted falls which are limited in number, body variability and type of fall and can be unrepresentative of real falls. The paper demonstrates that the use of fall recordings are unnecessary for modelling the fall as the simulation engine can produce a variety of fall events customised to an individual's physical characteristics using myoskeletal models of different morphology, without pre-knowledge of the falling behaviour. To validate this methodological approach, the simulation is customised by the person's height, modelling a rigid fall type. This approach allows the detection to be tailored to cohorts in the population (such as the elderly or the infirm) that are not represented in existing fall datasets. The method has been evaluated on several publicly available datasets which show that our method outperforms the results of previously reported research in fall detection. Finally, our approach is demonstrated to be robust to occlusions that hide up to 50% of a fall, which increases the applicability of automatic fall detection in a real-world environment such as the home.

Keywords: fall detection, physics simulation, depth video, visual occlusions, myoskeletal modelling, assisted living

*Corresponding author
Email address: g.mastorakis@ieee.org (Georgios Mastorakis)

1. Introduction

Automatic human fall detection is a valuable tool for monitoring vulnerable people who are prone to falling, such as the elderly living at home and those with mobility and other physical impairment. Several studies characterise the severity, frequency, risk factors (Ambrose et al., 2013) and cost (Heinrich et al., 2010) of fall incidents which are attributed to be the leading cause of fatal (Kannus et al., 2005) and non-fatal injuries among adults over the age of 65 (Bergen, 2016). Other studies discuss the acceptance of applied fall detection systems for the elderly (Feldwieser et al., 2016). (Lehtola et al., 1990) discuss the different types of accidental walking falls (slip, trip, and step) and their potential causes. (Robinovitch et al., 2013) discuss the various health conditions that may cause falls.

Two broadly accepted approaches for detecting falls are summarised in recent review studies (Chaudhuri et al., 2014; Igual et al., 2013): i) ad-hoc methods based on empirical observations and ii) pattern recognition methods that are trained using machine learning (ML). Both approaches employ recorded data to tune their detection performances. The tuning is used to ensure clear discrimination between a fall event and normal activities of daily living (ADLs), such as lying down or stooping to pick up an object, that are visibly similar and therefore, several recordings of these actions have been collected.

However, the quantity and availability of fall event data is low compared to other applications in action/event recognition because acquiring this data is subject to risk of injury. Also, the validity of fall data as seen in the current literature (Khan & Hoey, 2017) is questionable in terms of how realistic is the performance of humans acting falls. First, genuine fall data is not readily available, particularly for vulnerable people. Hence, actors are used to simulate fall events. Typically, these will be performed by healthy young people (i.e. students) under instruction from the researcher to fall "normally". In such circumstances, self-preservation takes over and the fall will be unrepresentative

of genuine falls, particularly if the aim is to acquire data representative of the vulnerable population.

Further issues arise from the use of actor-based human fall event data: firstly, the small number of human actors performing fall events may not be sufficient to represent the entire population. Variability in human body morphology such as height and other factors such age and gender are contributing factors to a fall, nevertheless, these are not recorded as part of a data capture. For example, one of the largest datasets (Ma et al., 2014) for fall detection consists of only 20 young male and female subjects performing falls and other ADLs such as sitting down, lying down, picking up an object etc. Therefore, algorithms based on these limited datasets may have questionable performance when applied to a broader demographic. Secondly, human subjects performing staged falls may have difficulty in acting realistically due to hesitation or concern of having an injury. As a result, data from such unrealistic recordings may have a negative impact on an algorithm's performance to capture genuine falls.

Another complication is related with occlusion in the home or other indoor environments. In a real-scene, furniture generally occupies most of the free space in a room, leaving limited space for a clear and unobstructed view of a person navigating the space. In the event of an occluded fall, current algorithms are generally untested for such scenarios. Existing datasets capture fall events without obstructions, except where a chair or stool is used to act an ADL. As a consequence, occlusion scenarios are rarely captured.

A fall may be associated with the fitness or health of the individual or a wide variety of medical conditions such as low blood pressure, brain ageing (Robinovitch et al., 2013) or brain atrophy (Yamada et al., 2013) or the consequence of a walking accident such as tripping, slipping or stumbling (Lehtola et al., 1990). The direction of a fall 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. Gender could also be a factor since males have a higher CoM than females (Kirby & Roberts, 1985).

The health of the person may impact the falling event such as a damaged limb which temporarily unbalances the natural human movement, or if the person is carrying an object. The variety of fall types and their causes is largely ignored by many fall detection systems, which produce a solution without assessing the characteristics of each individual or the scene.

This paper introduces four novel contributions to the task of fall detection; firstly, to 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; secondly, to customise the simulation using a person's height; thirdly, to demonstrate a new method of detecting falls by comparing the velocity profiles with the simulation profiles using the Hausdorff distance and fourthly, to introduce a framework for evaluating detection of falls under varying levels of occlusion.

2. Related Works

A plethora of fall detection algorithms can be found in the literature. The majority of those studies are considered in several recent review papers (Pannurat et al., 2014; Zhang et al., 2015; Spasova & Iliev, 2014). A summary of the available approaches using single RGB, multiple RGB cameras, depth sensors, accelerometers, ambient sensors and a fusion of some of these sensors is shown in Table 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 (Koshmak et al., 2016). Others have used depth data, particularly with the arrival of low cost depth sensors such as the Microsoft Kinect.

2.1. Conventional methods

A wide range of techniques for fall detection are found in the literature which use cameras and other sensors. In (Rougier et al., 2011) 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; 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 to assess the fall event (Vishwakarma et al., 2007); Multi-frame Gaussian Classifier where the direction of the body and the head location are used in by the algorithm over a predefined frame window (Hazelhoff et al., 2008); Bayesian filtering to determine the pose of the person as the probability of falling or getting up using data from a near-field imaging floor sensor (Rimminen et al., 2010); 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 (Liu et al., 2010); 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) (Cucchiara et al., 2007), thresholding techniques where signals from floor pressure data and infra-red images are processed and a fall is detected when set threshold are met (Tzeng et al., 2010), Fuzzy Logic is used to determine the state (e.g. upright, lying) of the person at each frame such using voxels derived by silhouettes of people captured by infra-red cameras (Anderson et al., 2009).

2.2. Depth sensors

This Section describes the use of data derived from depth sensors for fall detection. An early attempt is described in (Mastorakis & Makris, 2014). The algorithm detects a fall when the width, height, depth velocities of the human 3D bounding box exceed a particular set of thresholds derived from training data. This decision threshold has the same value for all subjects. The authors have evaluated their approach on many fall and non-fall events from their own dataset. In (Bian et al., 2012) the authors use skeleton tracking derived by analysis of depth data. However, the reliability of skeleton detection is range-limited, decreasing with depth. In more recent studies (Yun & Gu, 2016; Tran

Table 1: Pros and Cons of current fall detection approaches

Approach	Pros	Cons
Monocular camera (Nait-Charif & McKenna, 2004)	Easy to setup cheap	Privacy not preserved Occlusion ineffective
Multi-cam (Auvinet et al., 2011)	Occlusion robust 3D scene analysis	Difficult setup, cameras require syncing, privacy not preserved, 3D calibration
Infra-red (Mastorakis & Makris, 2014)	Privacy preserved, 3D scene analysis person segmentation ready	Interference, noisy data
Wearable accelerometer (Bourke et al., 2007) gyrospope (Li et al., 2009)	Occlusion robust, privacy preserved	Intrusive, must be worn
Ambient sensors acoustic (Popescu et al., 2008) floor vibration (Alwan et al., 2006)	Privacy preserved, occlusion robust	Expensive, can be applied on small areas
Fusion 3D vision & wearable (Kwolek & Kepski, 2014) 2D & heart monitor (Khawandi et al., 2012) acoustic & depth (Li et al., 2013)	Higher accuracy and performance	Complex setup, requires syncing

et al., 2017) the authors used Riemannian manifolds of fall velocity statistics and a combination of RGB and skeleton data derived from depth 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.

2.3. Public datasets

Lack of genuine fall data. A significant shortcoming of existing fall datasets is that they are universally performed by people acting falls, and there are no publicly-available datasets of real falls, and particularly of people from vulnerable populations. In their review, Khan and Hoey (Khan & Hoey, 2017) discuss public fall datasets and compare their usage. The authors of such datasets provide scant details of the actors' physical characteristics such as age, height, weight etc. though some specify the gender of the participants. Furthermore, the number of fall events or the participating actors is limited when compared to other action recognition datasets. Six datasets (Kwolek & Kepski, 2014; Ma et al., 2014; Cheng et al., 2012; Zhang et al., 2015; Gasparrini et al., 2016; Zhang et al., 2012) that provide depth videos are publicly available.

Lack of person's variability. A common approach is to detect all fall events with a single model, which ignores physical characteristics such as a person's height. All videos are tested via the same procedure that has been trained using a small set of data from human subjects of similar body (i.e. height) variability. Therefore, the robustness of such broad applications to the general population is questionable.

One solution to address the lack of data is to use an approach that is customised to a person's physical characteristics. (Ren & Shi, 2016) 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.

Occlusions. In the majority of the existing datasets, fall events appear fully visible in the scene, without occlusions. This is unrealistic for virtually all indoor (home) environments, as people move around a cluttered environment there may be frequent occasions during which they are part-occluded, to varying degrees. Although many studies discuss the application of fall detection for the elderly, at home or in hospital, occlusion is rarely mentioned, and the methods proposed are not evaluated on realistic data to provide occlusion-robust solutions.

In an occluded home scene, a fall detector should rely on features that are visible and stable. The top of the head location is selected by (Yang et al., 2015) as the highest point on the human body. Still, this approach suffers from missed detections when the head position and orientation is not aligned or simply far away from the sensor. A common approach to dealing with occlusions is to use several cameras and in that way to maintain a continuous view of the scene, though this is not guaranteed to eliminate the possibility of occlusion.

An attempt to evaluate current algorithms under occlusions is discussed in (Zhang et al., 2014) where the 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. However, the level of occlusion that is normally caused by a bed is fairly small (approximately 30%) and even this study (Zhang et al., 2014) fails to provide a proper evaluation of partially-occluded events.

2.4. 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 to acquire and label. One approach to address with this problem is to generate synthetic data based on a combination of actual observations and physical models. This subsection will discuss some of those attempts.

A number of studies employing computer vision and physics-based modelling exist in the literature. The most relevant studies (Xiang et al., 2010; Brubaker et al., 2010) discuss how tracking a walking person can be achieved with the use of a bipedal model based on physics simulation. Brubaker et al. (Brubaker et al., 2010) discusses 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 (Metaxas & Terzopoulos, 1993), 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 (Kakadiaris & Metaxas, 2000) presents a mathematical formulation and implementation of a system capable of accurate general human motion modelling. The work in (Duff et al., 2011) uses an off-the-shelf physics simulator to track the behaviour of a rigid object. Another framework is presented in (Vondrak et al., 2012), where a method estimates human motion from monocular video. This is done by reconstructing three-dimensional controllers (models) from 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 (Zouba et al., 2007). A dataset incorporating rigid poses is produced and used for the purpose of human behaviour recognition as well as scene understanding. Out of the context of computer vision related studies, the work in (Li et al., 2016) 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 (LeCun et al., 2015) have increased the need for large datasets. An example of synthetic data for action recognition can be found in the SOURREAL dataset presented in (Varol et al., 2017), consisting of 6 million image frames together with ground truth pose, depth maps, and

segmentation masks. Other examples include synthetic datasets for pedestrian detection (Ekbatani et al., 2017) and synthetic urban scenes from the SYNTHIA dataset (Ros et al., 2016).

(Mastorakis et al., 2016) describe the use of a simple myoskeletal model parametrised on the height of a person. Three different falls were simulated: forward, backward and sideways. The vertical velocity profile of the centre of mass was recorded and a comparison against real acted data using polynomial regression. The algorithm utilised a mixture of simulation data and real data to learn a decision boundary between falls and non-falls.

2.5. Summary

Previous fall detection methods are trained on acted fall recordings that may constrain the system performance in real-life scenarios, as such data is difficult to obtain, scarce and possibly unrepresentative. In addition, the physical characteristics of the monitored person are not considered, ignoring the possible variation in the fall dynamics. Most methods finally apply a threshold to decide whether a fall has occurred, and so must rely on a means to determine an optimal value to ensure optimum detection performance. Finally, evaluation of fall detection under occlusions is currently limited to using a dataset with rather low occlusions in height and frequency, missing the opportunity to test methods in more challenging scenarios. Nevertheless, acted fall data is currently the only source for evaluating the performance of fall detectors and to provide comparisons with previous work, even though it may be on imperfect data.

3. Methodology

Earlier research by (Mastorakis et al., 2016) had limitations: firstly, it employed the default OpenSim myoskeletal model which is armless; secondly, the polynomial fitting produced only a rough approximation of the profiles and was only used for the fall events as the ADLs where taken from a human acted dataset; thirdly, the CoM feature is not occlusion-robust; fourthly, a SVM was

used to separate falls from ADLs, and hence data were required to train the algorithm; finally, height was only simulated for fall events, as ADLs were taken from the dataset, where the variation was low.

A new fall detection algorithm is proposed that classifies every potential event as a fall or a non-fall by matching with a set of simulated falls. Fall and non-fall data are simulated using OpenSim's Full Body Model (Rajagopal et al., 2016) which provides more realistic physical characteristics. MoCap sequences were used to simulate ADLs, using models with height variation. An occlusion-robust feature based on the top bounding box of the person was selected. Finally, no learning processes (e.g. regression) were needed since the new method is capable of modelling the falls from a single simulation.

Although our implementation is based on processing depth data, the proposed methodology could be adapted to an RGB modality, if the scene is calibrated, or even for non-visual modalities, e.g. accelerometer data. Subsection 3.1 focuses on the use of depth data and particularly on the feature used for our approach.

Two of the public datasets discussed in Section 2.3 (Ma et al., 2014; Kwolek & Kepski, 2014) have been selected for the validation experiments, as they are used frequently for evaluation by other studies. These datasets consist of falls by actors performing several types of falls (backwards, forwards etc.) as well as ADLs.

A myoskeletal simulated engine, OpenSim (Delp et al., 2007), used for creating realistic fall actions is discussed in subsection 3.2.2. Subsection 3.3 analyses the fall detection pipeline where the Hausdorff distance is used to differentiate falls from ADLs and the section concludes with a description of an occlusion resilient methodology.

3.1. 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 (Nghiem et al., 2012; Yang

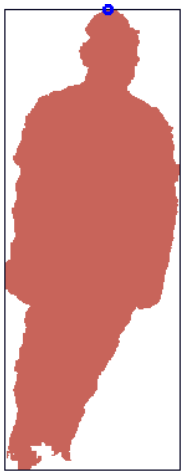

Figure 1: Blue dot indicates location of the top bounding box point that is used to indicate the head location

et al., 2015) 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 (Maldonado et al., 2016). For this work, depth data recorded by a Kinect I sensor and depth data analysis was implemented on the OpenNI (Occipital, 2016) platform. 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 1. The depth data, which is generated by an infra-red sensor, can be noisy due to the interaction with hair, where the infrared signals are absorbed rather than reflected. Therefore, the vertical location is filtered using LOWESS (Locally Weighted Scatterplot Smoothing) (Cleveland, 1981)), 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 3.3) and measured from the 3D bounding box (in metres). The

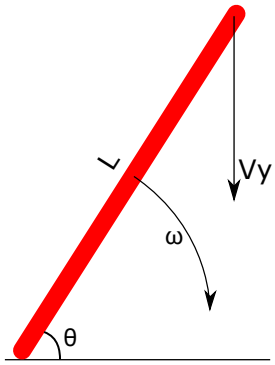

Figure 2: Falling rod, of length L with uniform mass and end-point vertical velocity V_y

height is 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.

3.2. Model simulation

This section presents two physics-based simulation approaches that imitate a falling person, one inspired by mechanics, based on a falling rod and one inspired by biomechanics, based on myoskeletal modelling.

3.2.1. Falling rod simulation

A rough approximation of the motion of a falling person with a rigid falling motion is given by modelling a rod of length L with uniform mass distribution falling from a vertical position, as seen in Figure 2.

The velocity of the end point V_y of the rod, is given by:

$$V_y = L\omega\cos(\theta) \tag{1}$$

where ω is recursively defined by:

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

where, ω is the angular velocity of the falling rod, θ the falling angle, n is the number of steps and g the gravitational acceleration ($9.81 m/sec^2$). Equation 2

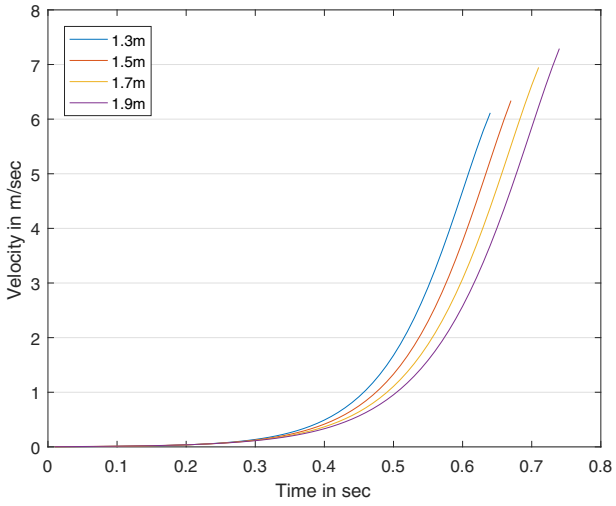

Figure 3: Typical rod model velocities. The final velocity is proportional to the length of the rod. This velocity is measured from the top point until the rod reaches the horizontal position

is derived by solving the Kinetic and Potential energy formulas under the assumption of energy preservation. The topmost end of the falling rod represents the location of the head. Figure 3 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 increases in proportion to the length of the rod – a taller person takes longer to fall and has the higher final velocity.

Nevertheless, 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 and is discussed in the next subsection.

3.2.2. Myoskeletal human model

Biomechanical studies have developed several applications to simulate a human's motion during activities such as walking, sitting, jumping etc, in an attempt to model the dynamics of the motion and to simulate how different

body parts behave during an action. OpenSim (Delp et al., 2007) 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 (Delp et al., 2007), therefore the model is considered suitable for generating muscle-driven simulations of healthy gait. Section 4.2 describes several experiments to validate the assumption that a real fall event has similar velocity patterns (e.g. head vertical velocity) with a fall simulated by a myoskeletal model.

3.2.3. Myoskeletal simulation

To perform a fall, a stationary model is placed on a platform that has a small inclination towards either the front, back or side. This inclination is used to trigger the fall event. 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 Figure 5 where the model falls as a rigid stick without any bending from the back or knees.

As discussed earlier in section 3.1, the top bounding box point is used as a feature for the proposed fall detection method. Here, the OpenSim engine can track the location of markers placed on the myoskeletal model generating point coordinates (e.g. the vertical location from the model's head $Y(t)$), velocity and acceleration values. A marker on the top of the head will be used for this purpose (blue sphere) as seen in Figure 4.

Using motion capture (MoCap) data to drive the model,(Müller et al., 2007) OpenSim can also be used to simulate body motion dynamics of everyday activities such as sitting down, lying down, picking up an object, walking quickly, turning around quickly and raising hands, that generate similar velocity profiles to a real fall motion (Anderson et al., 2006). These simulations yield a set of

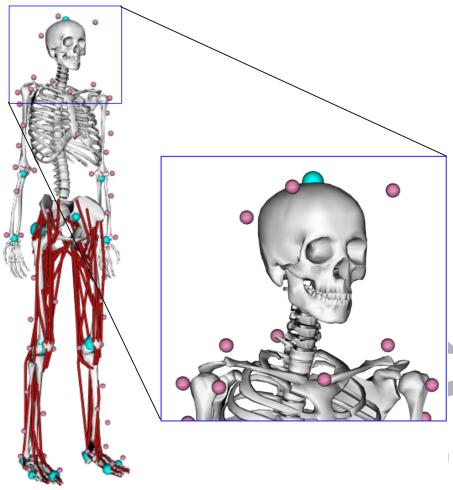

Figure 4: The Full Body Model given by OpenSim engine. Blue marker denotes the head location point, while pink markers denote the MoCap relevant markers

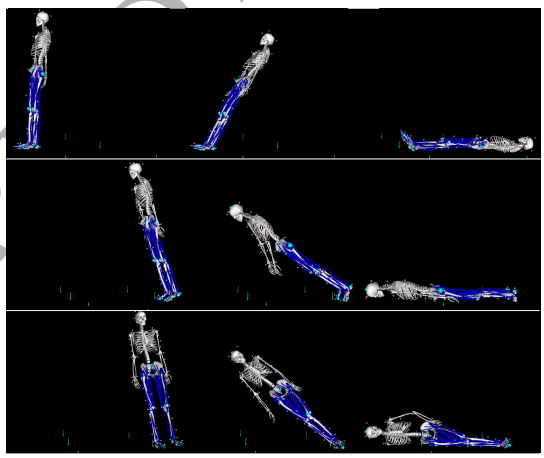

Figure 5: Three types of rigid fall, backward (top), forward (middle) and sideways (bottom) as simulated on OpenSim

non-fall velocity profiles that are used to differentiate falls from non-falls. This is required by our method in order to collect the vertical velocity profiles V_y measured from the head location. The MoCap data (3DC) is converted into OpenSim data format (TRC) using the c3d2OpenSim tools in (Alvim, 2016), to align MoCap markers with myoskeletal model markers. This allows transfer of the articulate motion from MoCap to the simulated model. The model can be parameterised within the OpenSim engine (OpenSim, 2017) to match the person's physical characteristics. The benefit of this conversion is to derive simulated models that have the motion of ADL activities, while the anthropometric parameters such as the height of the model can be adjusted separately.

For this study the model customisation is limited to varying the person's height as this is the most accessible parameter that can be extracted from the subjects in the public datasets. The myoskeletal model is scaled to represent the variation in human height. 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 is matched to the height of the faller.

3.3. Fall detection

The proposed fall detection algorithm is summarised by a flow diagram shown in Figure 6 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 3.1 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 3.2.3.

The green box encloses the fall detection algorithm. Inputs of the algorithm are the $V^+(t)$ and $V_i^-(t)$ which represent the velocity profiles of the simulated fall and non-fall events respectively and $Y(t)$ as the top bounding box location and the person's height. There are N simulated ADLs $(\{V_i^-(t)\}, i = 1 \dots N)$

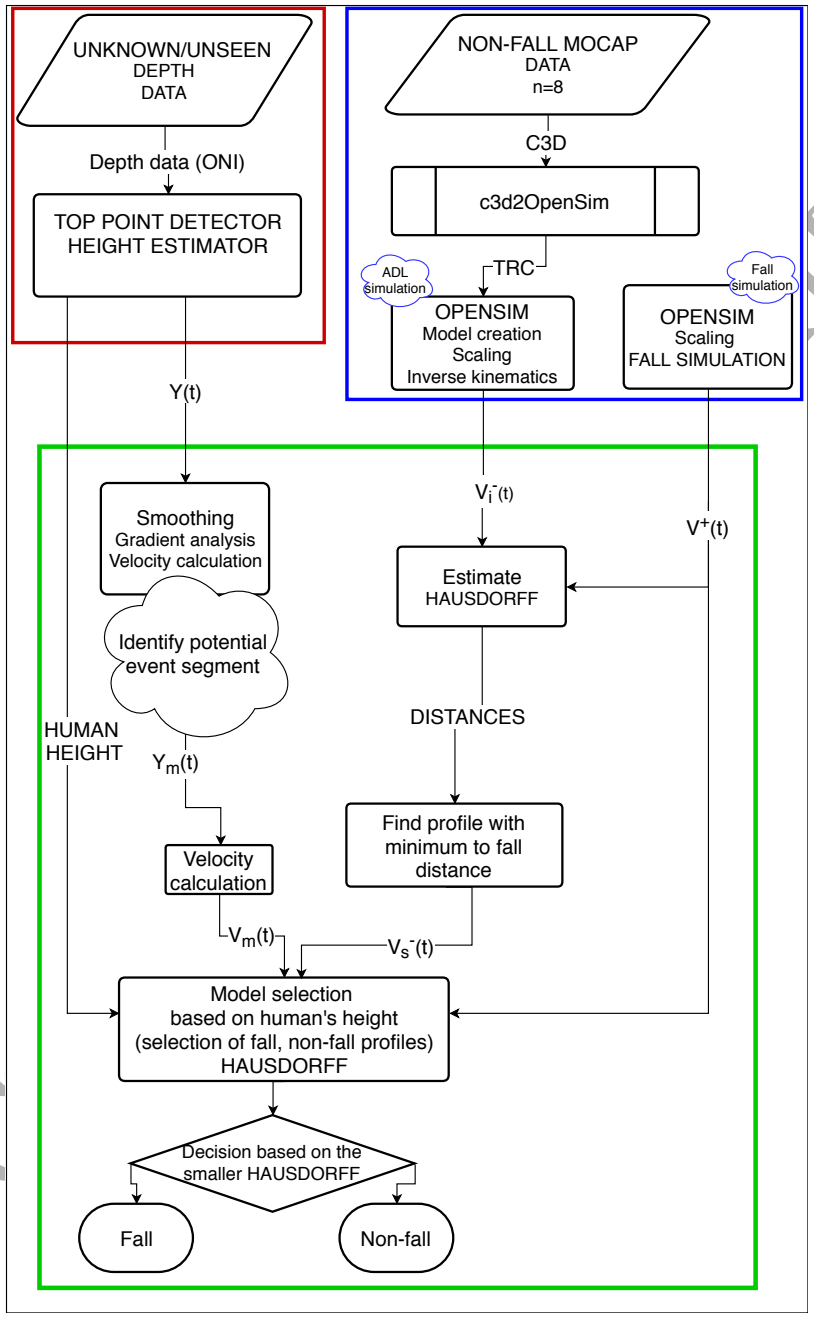

Figure 6: The pipeline of the proposed system. Red box encloses the data preprocessing, blue box the model simulation and the green box the fall detection. ONI: depth data format, C3D: standard mocap data format, TRC: OpenSim motion format

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 $V_s^-(t)$ which is compared with recorded human velocity profiles, is selected by minimising the Hausdorff distance (HD), defined as the distance between two profiles A, B,

$$HD(A,B) = max\{\underset{a\in A}{max}\ \underset{b\in B}{min}\ dist(b,a), \underset{b\in B}{max}\ \underset{a\in A}{min}\ dist(a,b)\} \quad (3)$$

against the profile of the simulated fall $V^+(t)$:

$$s = \underset{(i=1\ldots N)}{argmin}(HD(V_i^-(t), V^+(t))) \quad (4)$$

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 4.1.

The signal containing the bounding box top $Y_m(t)$ sequence of the detected person is processed in a similar way to identify potential fall segments. A single $Y(t)$ 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 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 ($V^+(t)$, $V_s^-(t)$). These simulated profiles are derived 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 5:

$$P(E) = \begin{cases} fall & if \quad (HD(V^{+}(t), V_m(t)) < HD(V_s^{-}(t), V_m(t))), \\ non\text{-}fall & if \quad (HD(V^{+}(t), V_m(t)) > HD(V_s^{-}(t), V_m(t)). \end{cases} \tag{5}$$

3.4. 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 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 sample points to satisfy a strict linearity condition. The segment is extrapolated towards the ground ($Y_m = 0$) as seen in Figure 7. The new location profile will include those calculated points in order to provide the algorithm with a complete location profile. In order to assess the robustness of detecting falls under occlusions, we have developed a protocol which evaluates the impact of various levels of occlusion, discussed in the next subsection.

3.4.1. Synthetic Occlusions

To evaluate fall detection under various occlusion scenarios, the datasets are augmented by adding synthetic occlusions to the depth videos. In detail, the depth video is edited to mask out the lower portion of the image - obscuring the original depth data in the lower portion of the image. These frames are then re-assembled to re-create the depth video and the full video analysis is

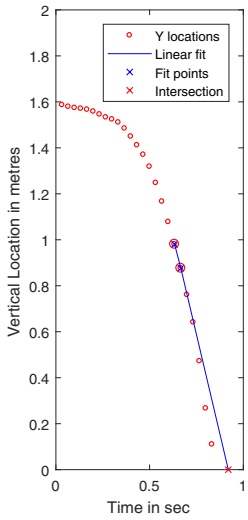

Figure 7: Location profile estimation in an occluded fall event. Occlusion occurs near 0.8m, then two points are used to fit a line which intersects at x point

applied to detect falls. Figure 8 shows the occlusions applied for the evaluation of fall detection. This is an efficient and effective means to implement occlusion, avoiding the need for actors to simulate multiple falls behind various occluding pieces of furniture. The occlusion is quantified as a ratio of its height to the person's height when standing at the pre-fall state.

4. Experimental Results and Discussion

4.1. Validation of Hausdorff Distance

The Hausdorff distance is discussed in (Junejo & Foroosh, 2008), where authors measure the similarity of trajectories. A benefit of using this distance is the capability to compare two profiles of different sample lengths. The capability of HD to differentiate between fall and non-fall profiles 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, as well as three different falls (forward, backward, sideways) are considered and Figure 9 shows the probability

Figure 8: Three occlusion modes. The black rectangle in the image represents a synthetic occlusion applied to the depth image. The size of occlusion is expressed proportionally to the person's height: (a) 40%, (b) 50%, (c) 70% occlusion measured from the ground

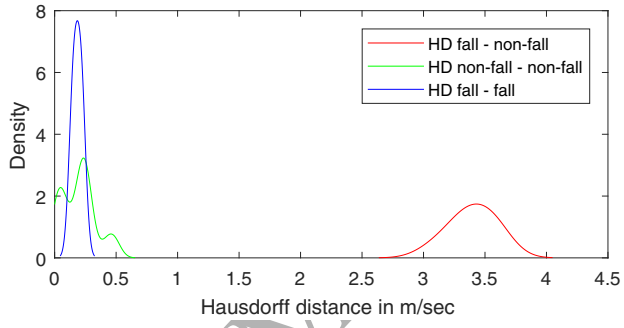

Figure 9: PDF diagram of Hausdorff distances: blue, red and green curves denote the pdfs of HDs between falls, between non-falls and between falls and non-falls respectively

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 event, which is an order of magnitude larger. This justifies choice of HD as a distance metric to discriminate between the velocity profiles of events.

4.2. Evaluation of the simulation models

The base for our experiments is the Full Body Model (Rajagopal et al., 2016) with the properties of an adult male body, in terms of average body mass index (BMI) height (1.78m) and weight (78kg). This model is linearly scaled (as described in 3.2.3) to cover a range of heights between 130-190cm in 20cm

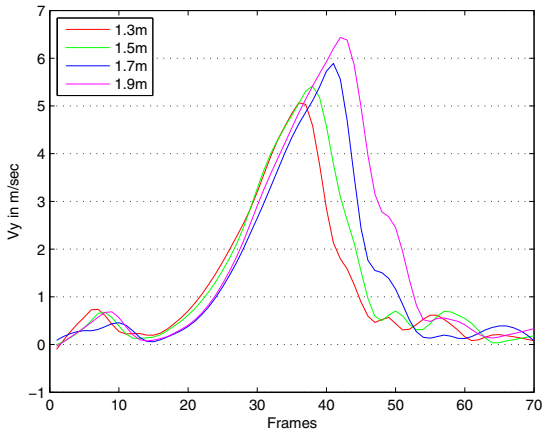

Figure 10: Velocity profiles of four simulated models of a rigid fall. Notice the difference in final velocity according to the model's height

steps. Four model variations were found sufficient, as no further improvement in robustness was observed by considering more height models. For each new actual depth video to be processed, the algorithm estimates the subject's height and selects the body model that is closest to this height. Figure 10 shows the fall velocity profiles ($V^{+}(t)$) for the four models.

Note again, that the height is the only model parameter that is adjusted, along with the inclination towards the ground in order to initiate the fall. Such a choice confirms the validity of model customisation as discussed in Section 1. If other model parameters such as the centre of torso's mass are modified or muscle forces from the knees or ankles are reduced, then other types of falls can also be simulated such as collapsing along the Y-axis (Figure 11). However, this paper focuses on rigid falls, providing a proof-of-concept of the fall simulation approach. This also matches the datasets used for evaluation, as these are largely comprised of enacted rigid falls. The same issue arises in the evaluation of slipping and tripping falls, as such incidents are not covered by public datasets as they involve further risks for the participants. Therefore, future work is required to extend the approach as discussed in the Conclusion.

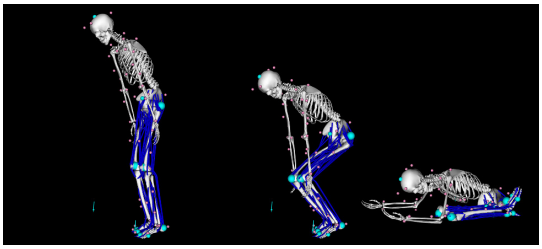

Figure 11: Collapsing fall as simulated by OpenSim

Previous studies discuss different types of falls as defined by the falling direction, such as forward, backward and sideways fall (Pannurat et al., 2014). Figure 12 shows the simulations of those types and indicate the variation in V_y. The HD is used to compare the velocity profiles. The HD between a forward and a backward fall was 0.215 m/sec, between forward and sideways 0.216 m/sec, and between sideways and backwards 0.154 m/sec. The estimated pdf of the HDs of these fall-fall comparisons is plotted in Figure 9 (blue line) denoted with a blue curve. Effectively, there is a little added benefit in matching these different rigid falls with a real fall profile as the HD is small, reflecting the similarity of V_y across the three types of fall. For this reason, only the forward fall simulation produced by the OpenSim engine is used to represent any rigid fall.

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 (The Daily Dot, 2014). Those videos are the closest representations of genuine fainting where actors fall rigidly unconscious to the ground. The videos were processed using (Tsai, 1986) 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 (Bouguet, 2001) is used to track the head's centroid and from these tracked points the algorithm calculates the fall velocity profile.

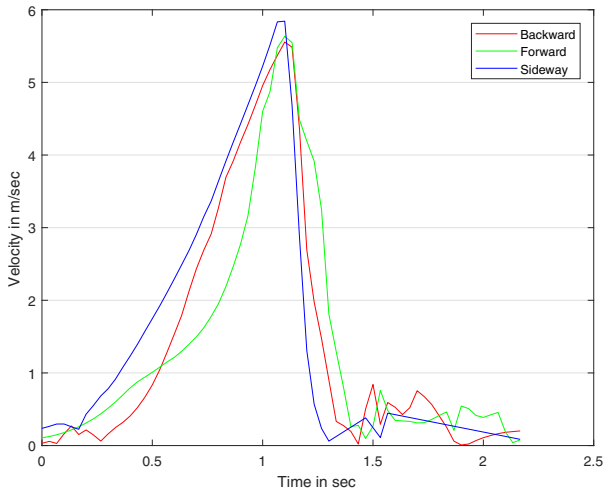

Figure 12: Velocity profiles of three simulated rigid falls using OpenSim. We can see the visual similarity of the profiles. Some noise is observed due to elasticity of the Contact Geometry of the model, before it comes to rest after 2.2 sec

An evaluation is conducted to measure the similarity of profiles of both the OpenSim model and the falling rod model, where, Hausdorff distances are measured against the V_y of actual Youtube fall events. The average HD of the actual falls (when compared with an OpenSim model) were 0.365m/sec and 1.944m/sec (when compared with the falling rod) with standard deviations of 0.078m/sec and 0.782m/sec respectively.

The results show that there is a much smaller distance (0.365m/sec, similar to the HD between different types of falls as seen on Figure 9) between the actual fall profile and OpenSim model, compared to the rod model (1.944m/sec). Therefore, given this comparison, fall events simulated by OpenSim are shown to provide a more accurate match to the recorded fall data.

4.3. Evaluation datasets

Two public datasets (Kwolek & Kepski, 2014; Ma et al., 2014) and our own (Mastorakis & Makris, 2014) of people performing fall events and ADLs are

Table 2: Type and number of events from each dataset

	Datasets			
Actions	**A**	**B**	**C**	Total
Fall	15	200	48	263
Sitting	9	200	32	241
Picking up	-	-	32	32
Squatting	8	200	-	208
Lying	16	200	48	264
Bending	7	200	-	207

used to evaluate our approach. For brevity, we have named those datasets as **A**, **B**, **C** respectively. The datasets have 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. Table 2 summarises the number and type of actions in these datasets.

4.4. Evaluation Measures

Results will be 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). The accuracy (Eq. 6) gives the proportion of true events that were correctly classified across all measurements. Precision (Eq. 7) is the proportion of positive results that were correctly classified. Sensitivity (Eq. 8) is the proportion of actual positive event results correctly classified and specificity (Eq. 9) is the proportion of negative results correctly classified.

$$Accu = \frac{TP + TN}{TP + FN + FN + TN} \tag{6}$$

$$Prec = \frac{TP}{TP + FP} \tag{7}$$

$$Sens = \frac{TP}{TP + FN} \tag{8}$$

$$Spec = \frac{TN}{TN + FP} \tag{9}$$

4.5. Evaluation of simulation based fall detection

Since our methodology does not require samples for training, results are reported for the full datasets. Table 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 (Akagündüz et al., 2017) where more falls are detected, but more ADLs are also detected as falls. 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 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.

4.6. Evaluation of using a customised simulated model

In order to evaluate the benefits of the height customisation, datasets **A, B, C** 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 4 shows the results of using fall/non-fall data from across all three datasets **A**, **B** and **C**.

These events are categorised according to a subject's height, resulting in are 10 samples in the range (120-139) cm, 110 samples (140-159 cm) etc. The results show that detections are missed when 130cm tall subjects are tested using the

Table 3: Performance of fall detection and comparison against previous studies across 2 public datasets

Method	Accu (%)	Prec (%)	Sens (%)	Spec (%)
(a) Dataset A (15 fall, 40 ADLs)				
Bourke et al.	95.00	90.91	100.0	90.00
Kwolek & Kepski (2014)	98.33	96.77	100.0	96.67
Yun & Gu (2016)	-	-	100.0	97.25
Tran et al. (2017)	99.37	96.77	100.0	99.23
Proposed	**100.0**	**100.0**	100.0	**100.0**
(b) Dataset B (200 fall, 800 ADLs)				
Ma et al. (2014)	86.83	-	91.15	77.14
Merrouche & Baha (2016)	91.89	-	-	-
Akagündüz et al. (2017)	92.98	-	**93.52**	90.76
Proposed	**96.90**	**94.00**	90.88	**98.48**

170, 190cm simulation models and also when 150cm tall 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 that will lead to detecting more non-fall events as falls (FNs) since the velocity profile is lower in comparison with the 170, 190cm models. This is reflected in Table 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 are 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 knows which myoskeletal model to use. 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 4.

This experiment tests each myoskeletal model against all the data (regardless of person's height) and observe its performance. The main diagonal of the table corresponds to results where 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.

4.7. Evaluation of algorithm under occlusion

In a home environment, furniture such as chairs, boxes, tables, sofas and beds may result in occlusion of a fall event. Because current datasets do not include examples of occlusions, we have proposed an evaluation protocol discussed in 3.4.1 based on synthetic occlusions, that allow an evaluation of fall detection performance under varying degrees of body occlusion.

The evaluation protocol was applied to datasets A and C consisting of 63 fall and 154 ADL events. Results are summarised in Figure 13 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 (31.9%) of the sample detected at 70% occlusion. Specificity reduces even more rapidly. 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

Table 4: Simulated model height variability over **A,B,C** datasets combined. The table presents results of the algorithm for each height models applied to the height-labelled acted datasets (**A,B**) and the in-house depth dataset (**C**) 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 (estimated from video) number of samples (falls, non-falls)							
Sim model height	130 (10, 40)		150 (110, 440)		170 (108, 441)		190 (8, 12)	
	sens	spec	sens	spec	sens	spec	sens	spec
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

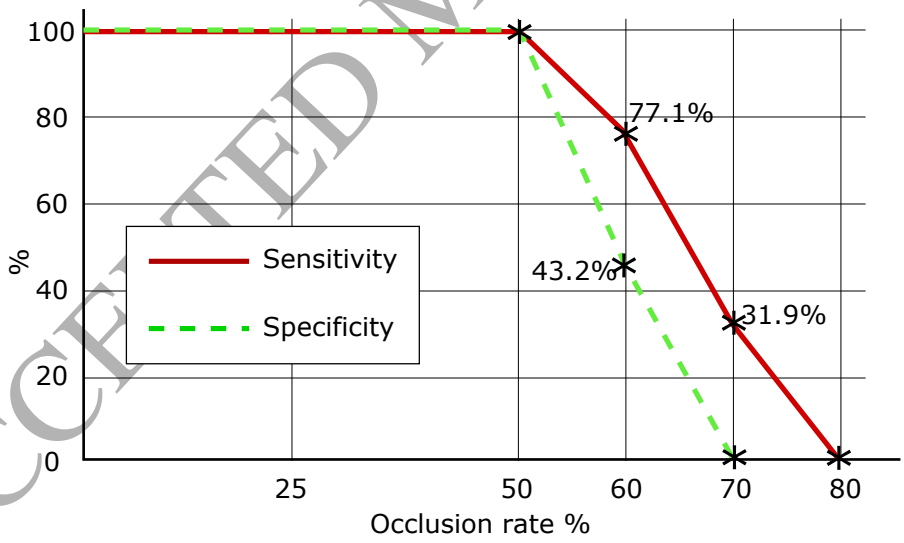

Figure 13: Occlusion results across datasets A and C. Red line denotes the sensitivity and green line shows the specificity at various occlusion rates. Note that both fall and ADL events are detected by our approach when occlusion rate is at least 50%

ratio of distances of the person and the occlusion from the camera; meaning that in small rooms the likelihood and degree of occlusion will be higher. To relate the occlusion rates 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 (taller furniture, such as bookcases and wardrobes are typically placed against a wall, and so do not offer any risk of causing occlusion). Having this visual approximation we can say that given this set of examples, our algorithm will be capable of detecting occluded falls in a typical home scene containing furniture such as coffee tables, desks, setters/sofas, beds etc.

5. Discussion

Current State-of-the-Art approaches to fall detection use a single model of detection, derived from recordings of falls undertaken by actors that are unrepresentative of the general demographic, and particularly of vulnerable populations. They ignore the physical characteristics or conditions of the participants or the intended target group such as the elderly, and also take little account of the scene structure.

Our approach uses a customisable fall model that can be tailored to an individual's height using a myoskeletal simulation, where up to now, the detection of falls was data-driven, based on unrepresentative data. Customising the fall simulation using the height of the faller is shown to improve the classification accuracy of the fall detection. The model can be customised to a wider demographic, optimising detection performance and false detections. This approach has the potential to model other types of falls, different body morphologies, physical impairment and the effects of age on physical capabilities. This is particularly important for vulnerable populations such as the elderly or the infirm, where real fall data are unavailable. Such an approach will have a significant impact on real fall detection applications since it can be tailored to the individual, compared to the generic approach employed by other systems.

The majority of fall detection algorithms rely on human fall data (i.e. videos) or are manually tuned by the researchers. Our approach replaces the need for fall data with simulations of falls. Fall velocity profiles are directly compared with the simulation fall profiles using the Hausdorff distance, allowing the entire duration of the fall to be analysed. Since fall detection is decided by computing the HD a fall and non-fall velocity profiles and selecting the closest, the method is not reliant on determining an optimal threshold. In addition, the combination of customised simulation and profile comparison are also shown to be robust to occlusions, and against synthetic occlusions up to 50% of the person's height. An evaluation protocol is used that provides the means to evaluate occlusions applied to existing public datasets by generating synthetic occlusions. A benefit of using depth data for fall detection, compared with RGB video, is that it provides a degree of privacy, as it hides details such as facial features.

One weakness with the simulation approach is the need to acquire the height of the individual. Height measurement can be inaccurate in an occluded scene where the feet are not visible. In this case the model selected will be shorter than required and may result in an increase of FNs. However, although in the current work height measurement was sampled in an instant prior to the fall, in a real fall monitoring system height can be more accurately estimated over a much longer period of observation. Alternatively, the height of person could be a fixed and pre-determined parameter used by the system. Another weakness with depth data, compared with accelerometer data, is that it is potentially more intrusive, raising concerns of data privacy regarding private body characteristics, or actions.

Another possible complication is related to the location of the head. The algorithm tracks a single point which is expected to be the most recognisable in terms of visibility, such as the top bounding box point, which is located on the head. The reason for using this point is justified in Section 3.1. If the head is not visible at the start of the fall, this point will correspond to some other point on the body, but the algorithm will still function. The problem will result in a miss-measurement of the person's height, which will lead to miss-selection

of an incorrect height for the simulated model.

Another limitation is that the fall simulation models were restricted to rigid falls. This was decided in part because it is the simplest fall to model, but also because the current datasets mainly provide examples of rigid falls. Hence, in order to provide results that are comparable with other researchers that have used these datasets, the rigid model was the clear choice. It should be noted that the method is still reliant on acted fall recordings in order to evaluate its effectiveness and compare its performance with the results of other researchers in the field.

6. Conclusion

This paper reports an investigation of simulation methodologies to validate the development of robust algorithms for detecting falling people, extending previous work (Mastorakis et al., 2016). 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.

A fall detection system based on depth data has been proposed. Falls are detected by comparing the velocity profile of a fall with a simulated fall profile. Its performance has been evaluated using 3 conventional fall datasets, which are recordings captured from acted falls. Results show high levels of performance in distinguishing falls from non-fall events, with higher accuracy when evaluated on public datasets that consist of more than 1000 videos. These results are shown to outperform those reported in the literature.

The results demonstrate the capability of using fall simulation as a replacement for acted fall data, introducing significant flexibility in tailoring the fall model to the faller, enabling a wide range of a variation in their physical characteristics and health, and to the type of falls that can be simulated.

Another key contribution of the paper is with addressing the problem of visually-occluded falls, an issue that is largely unconsidered by other researchers. A novel evaluation framework, based on creating synthetic occlusions, has been

proposed in order to establish an understanding of how occlusions impact the detection algorithm. Results show that our algorithm can reliably detect falls with body occlusions of up to 50% (measured at a standing position).

Whilst the modelling methods used in this paper have been applied to depth data, they could also be adapted to other sensor technologies, such as RGB cameras and accelerometers, from which velocity or acceleration profiles can also be extracted.

Future work will investigate other types of fall, beyond the rigid fall modelled in this paper. In addition, alternative body morphologies and other physical characteristics of the faller will be assessed through the modelling. The HD will then be re-evaluated against these new fall types. We also intend to model more realistic occlusions (e.g. home furniture) 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.

References

References

Akagündüz, E., Aslan, M., Şengür, A., Wang, H., & İnce, M. C. (2017). Silhouette orientation volumes for efficient fall detection in depth videos. *IEEE Journal of Biomedical and Health Informatics*, *21*, 756–763.

Alvim, F. (2016). Convert .c3d and .csv files to opensim. https://simtk.org/projects/c3d2opensim. Accessed 1 May 2017.

Alwan, M., Rajendran, P. J., Kell, S., Mack, D., Dalal, S., Wolfe, M., & Felder, R. (2006). A smart and passive floor-vibration based fall detector for elderly. In *Information and Communication Technologies, 2006. ICTTA'06. 2nd* (pp. 1003–1007). IEEE volume 1.

Ambrose, A. F., Paul, G., & Hausdorff, J. M. (2013). Risk factors for falls among older adults: a review of the literature. *Maturitas*, *75*, 51–61.

Anderson, D., Luke, R. H., Keller, J. M., Skubic, M., Rantz, M., & Aud, M. (2009). Linguistic summarization of video for fall detection using voxel person and fuzzy logic. *Computer Vision and Image Understanding*, *113*, 80–89.

Anderson, F. C., John, C. T., Guendelman, E., Arnold, A. S., & Delp, S. L. (2006). Simtrack: Software for rapidly generating muscle-actuated simulations of long-duration movement.

Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., & Meunier, J. (2011). Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution. *IEEE transactions on information technology in biomedicine*, *15*, 290–300.

Bergen, G. (2016). Falls and fall injuries among adults aged 65 years – United States, 2014. *MMWR. Morbidity and mortality weekly report*, *65*.

Bian, Z.-P., Chau, L.-P., & Magnenat-Thalmann, N. (2012). Fall detection based on skeleton extraction. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry* (pp. 91–94).

Bouguet, J.-Y. (2001). Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, *5*, 4.

Bourke, A., Obrien, J., & Lyons, G. (2007). Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & posture*, *26*, 194–199.

Brubaker, M. A., Fleet, D. J., & Hertzmann, A. (2010). Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, *87*, 140–155.

Chaudhuri, S., Thompson, H., & Demiris, G. (2014). Fall detection devices and their use with older adults: a systematic review. *Journal of Geriatric Physical Therapy (2001)*, *37*, 178.

Cheng, Z., Qin, L., Ye, Y., Huang, Q., & Tian, Q. (2012). Human daily action analysis with multi-view and color-depth data. In *Computer Vision–ECCV 2012. Workshops and Demonstrations* (pp. 52–61).

Cleveland, W. S. (1981). Lowess: A program for smoothing scatterplots by robust locally weighted regression. *The American Statistician*, *35*, 54–54.

Cucchiara, R., Prati, A., & Vezzani, R. (2007). A multi-camera vision system for fall detection and alarm generation. *Expert Systems*, *24*, 334–345.

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., & Thelen, D. G. (2007). OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, *54*, 1940–1950.

Duff, D. J., Mörwald, T., Stolkin, R., & Wyatt, J. (2011). Physical simulation for monocular 3d model based tracking. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5218–5225).

Ekbatani, H. K., Pujol, O., & Segui, S. (2017). Synthetic data generation for deep learning in counting pedestrians. In *ICPRAM* (pp. 318–323).

Feldwieser, F., Feldwieser, F., Marchollek, M., Marchollek, M., Meis, M., Meis, M., Gietzelt, M., Gietzelt, M., Steinhagen-Thiessen, E., & Steinhagen-Thiessen, E. (2016). Acceptance of seniors towards automatic in home fall detection devices. *Journal of Assistive Technologies*, *10*, 178–186.

Gasparrini, S., Cippitelli, E., Gambi, E., Spinsante, S., Wåhslén, J., Orhan, I., & Lindh, T. (2016). Proposal and experimental evaluation of fall detection solution based on wearable and depth data fusion. In *ICT innovations 2015* (pp. 99–108).

Hazelhoff, L., Han, J. et al. (2008). Video-based fall detection in the home using principal component analysis. In *International Conference on Advanced Concepts for Intelligent Vision Systems* (pp. 298–309).

Heinrich, S., Rapp, K., Rissmann, U., Becker, C., & König, H.-H. (2010). Cost of falls in old age: a systematic review. *Osteoporosis international*, *21*, 891–902.

Igual, R., Medrano, C., & Plaza, I. (2013). Challenges, issues and trends in fall detection systems. *Biomedical engineering online*, *12*.

Junejo, I. N., & Foroosh, H. (2008). Euclidean path modeling for video surveillance. *Image and Vision computing*, *26*, 512–528.

Kakadiaris, L., & Metaxas, D. (2000). Model-based estimation of 3d human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 1453–1459.

Kannus, P., Parkkari, J., Niemi, S., & Palvanen, M. (2005). Fall-induced deaths among elderly people. *American journal of public health*, *95*, 422–424.

Khan, S. S., & Hoey, J. (2017). Review of fall detection techniques: A data availability perspective. *Medical engineering & physics*, *39*, 12–22.

Khawandi, S., Chauvet, P., & Daya, B. (2012). Applying neural network architecture in a multi-sensor monitoring system for the elderly. In *Proceedings of the 6th International Conference on Advanced Engineering Computing and Application in Sciences (ADVCOMP 2012), Spain* (pp. 15–22). Citeseer.

Kirby, R. F., & Roberts, J. A. (1985). *Introductory biomechanics*. Mouvement Pubns.

Koshmak, G., Loutfi, A., & Linden, M. (2016). Challenges and issues in multisensor fusion approach for fall detection. *Journal of Sensors*, *2016*.

Kwolek, B., & Kepski, M. (2014). Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer methods and programs in biomedicine*, *117*, 489–501.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

Lehtola, C. J., Becker, W. J., & Brown, C. M. (1990). Preventing injuries from slips, trips and falls. *Institute of Food and Agriculture Science, University of Florida*, .

Li, Q., Stankovic, J. A., Hanson, M. A., Barth, A. T., Lach, J., & Zhou, G. (2009). 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* (pp. 138–143). IEEE.

Li, W., Azimi, S., Leonardis, A., & Fritz, M. (2016). To fall or not to fall: A visual approach to physical stability prediction. *arXiv preprint arXiv:1604.00066*, .

Li, Y., Banerjee, T., Popescu, M., & Skubic, M. (2013). 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* (pp. 6736–6739). IEEE.

Liu, C.-L., Lee, C.-H., & Lin, P.-M. (2010). A fall detection system using k-nearest neighbor classifier. *Expert systems with applications*, *37*, 7174–7181.

Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., & Li, Y. (2014). Depth-based human fall detection via shape features and improved extreme learning machine. *IEEE journal of biomedical and health informatics*, *18*, 1915–1922.

Maldonado, C., Ríos, H., Mezura-Montes, E., & Marin, A. (2016). Feature selection to detect fallen pose using depth images. In *Electronics, Communications and Computers (CONIELECOMP), 2016 International Conference on* (pp. 94–100).

Mastorakis, G., Hildenbrand, X., Grand, K., & Makris, D. (2016). 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)*.

Mastorakis, G., & Makris, D. (2014). Fall detection system using kinects infrared sensor. *Journal of Real-Time Image Processing*, *9*, 635–646.

Merrouche, F., & Baha, N. (2016). Depth camera based fall detection using human shape and movement. In *Signal and Image Processing (ICSIP), IEEE International Conference on* (pp. 586–590).

Metaxas, D., & Terzopoulos, D. (1993). Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*, 580–591.

Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., & Weber, A. (2007). Documentation mocap database hdm05, .

Nait-Charif, H., & McKenna, S. J. (2004). Activity summarisation and fall detection in a supportive home environment. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (pp. 323–326). IEEE volume 4.

Nghiem, A. T., Auvinet, E., & Meunier, J. (2012). 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* (pp. 164–169).

Occipital (2016). OpenNI. http://structure.io/openni. Accessed 05 April 2016.

OpenSim (2017). Intro to musculoskeletal modeling. https://simtk-confluence.stanford.edu/display/OpenSim/. Accessed 11-April-2016.

Pannurat, N., Thiemjarus, S., & Nantajeewarawat, E. (2014). Automatic fall monitoring: a review. *Sensors*, *14*, 12900–12936.

Popescu, M., Li, Y., Skubic, M., & Rantz, M. (2008). 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* (pp. 4628–4631). IEEE.

Rajagopal, A., Dembia, C. L., DeMers, M. S., Delp, D. D., Hicks, J. L., & Delp, S. L. (2016). Full-body musculoskeletal model for muscle-driven simulation of human gait. *IEEE Transactions on Biomedical Engineering*, *63*, 2068–2079.

Ren, L., & Shi, W. (2016). Chameleon: personalised and adaptive fall detection of elderly people in home-based environments. *International Journal of Sensor Networks*, *20*, 163–176.

Rimminen, H., Lindström, J., Linnavuo, M., & Sepponen, R. (2010). Detection of falls among the elderly by a floor sensor using the electric near field. *IEEE Transactions on Information Technology in Biomedicine*, *14*, 1475–1476.

Robinovitch, S. N., Feldman, F., Yang, Y., Schonnop, R., Leung, P. M., Sarraf, T., Sims-Gould, J., & Loughin, M. (2013). Video capture of the circumstances of falls in elderly people residing in long-term care: an observational study. *The Lancet*, *381*, 47–54.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). 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* (pp. 3234–3243).

Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2011). Robust video surveillance for fall detection based on human shape deformation. *IEEE Transactions on circuits and systems for video Technology*, *21*, 611–622.

Spasova, V., & Iliev, I. (2014). A survey on automatic fall detection in the context of ambient assisted living systems. *International journal of advanced computer research*, *4*, 94.

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

Tran, T.-H., Le, T.-L., Hoang, V.-N., & Vu, H. (2017). Continuous detection of human fall using multimodal features from kinect sensors in scalable environment. *Computer Methods and Programs in Biomedicine*, .

Tsai, R. Y. (1986). An efficeint and accurate camera calibration technique for 3d machine vision. *CVPR, 1986*, .

Tzeng, H.-W., Chen, M.-Y., & Chen, J.-Y. (2010). Design of fall detection system with floor pressure and infrared image. In *System Science and Engineering (ICSSE), 2010 International Conference on* (pp. 131–135). IEEE.

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M., Laptev, I., & Schmid, C. (2017). Learning from synthetic humans. *arXiv preprint arXiv:1701.01370*, .

Vishwakarma, V., Mandal, C., & Sural, S. (2007). Automatic detection of human fall in video. In *International conference on pattern recognition and machine intelligence* (pp. 616–623). Springer.

Vondrak, M., Sigal, L., Hodgins, J., & Jenkins, O. (2012). Video-based 3d motion capture through biped control. *ACM Transactions On Graphics (TOG)*, *31*, 27.

Xiang, Y., Arora, J. S., & Abdel-Malek, K. (2010). Physics-based modeling and simulation of human walking: a review of optimization-based and other approaches. *Structural and Multidisciplinary Optimization*, *42*, 1–23.

Yamada, M., Takechi, H., Mori, S., Aoyama, T., & Arai, H. (2013). Global brain atrophy is associated with physical performance and the risk of falls in older adults with cognitive impairment. *Geriatrics & gerontology international*, *13*, 437–442.

Yang, L., Ren, Y., Hu, H., & Tian, B. (2015). New fast fall detection method based on spatio-temporal context tracking of head by using depth images. *Sensors*, *15*, 23004–23019.

Yun, Y., & Gu, I. Y.-H. (2016). 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.

Zhang, Z., Conly, C., & Athitsos, V. (2014). Evaluating depth-based computer vision methods for fall detection under occlusions. In *International Symposium on Visual Computing* (pp. 196–207).

Zhang, Z., Conly, C., & Athitsos, V. (2015). A survey on vision-based fall detection. In *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments* (p. 46).

Zhang, Z., Liu, W., Metsis, V., & Athitsos, V. (2012). A viewpoint-independent statistical method for fall detection. In *21st International Conference on Pattern Recognition (ICPR)* (pp. 3626–3630).

Zouba, N., Bremond, F., Thonnat, M., & Vu, V. T. (2007). Multi-sensors analysis for everyday activity monitoring. *Proc. of SETIT*, (pp. 25–29).