

Fall detection system using Kinect's infrared sensor

Georgios Mastorakis · Dimitrios Makris

Received: 5 September 2011 / Accepted: 20 February 2012 / Published online: 15 March 2012
© Springer-Verlag 2012

Abstract This paper presents a novel fall detection system based on the Kinect sensor. The system runs in real-time and is capable of detecting walking falls accurately and robustly without taking into account any false positive activities (i.e. lying on the floor). Velocity and inactivity calculations are performed to decide whether a fall has occurred. The key novelty of our approach is measuring the velocity based on the contraction or expansion of the width, height and depth of the 3D bounding box. By explicitly using the 3D bounding box, our algorithm requires no pre-knowledge of the scene (i.e. floor), as the set of detected actions are adequate to complete the process of fall detection.

Keywords Kinect · Fall detection · Home assistance · Real-time processing · 3D bounding box analysis

1 Introduction

Computer vision-based fall detection is a very important application that has been used to save lives [21]. A fall occurs when a person accidentally falls/slips while walking or standing. Age is a significant factor that is closely linked

to severe falls [18]. Several studies have shown [12] that elderly people experience at least one fall every year. Also, falls are the main cause of accidental death in older adults aged 65 or more, based on a review of 90 epidemiological studies [10]. Other resources show the injuries caused from falls in the general population [19].

A fall may be due to health and ageing-related issues, abnormality of walking surface or even lack of concentration. A falling person requires immediate assistance after the incident. Therefore, 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.

This work introduces a real-time algorithm that utilises the human 3D bounding box, expressed in world coordinates. Depth data are acquired using infrared (IR) signal from Kinect which is not affected by lighting conditions. Using the 3D bounding box our algorithm calculates the first derivative (velocity) of width, height and depth to determine whether a particular activity is a fall or not. Our algorithm does not require any pre-knowledge of the floor plane coordinates or the detection and tracking of a particular body part as other systems do [11, 17, 23].

Also, we have tested our algorithm to detect a range of falls (backward, forward and sideways), while setting the sensor to different positions (side view, frontal view and back view) and different type of actions performed at different speeds. Those non-fall activities could cause a false positive (FP) detection especially when a person is lying down, crouching down, picking up an item from the floor, etc. The main algorithm is designed as a simple two step Boolean decision tree where several output data are checked sequentially. The parameters of the decision tree are estimated by random search optimisation. Furthermore,

Electronic supplementary material The online version of this article (doi:10.1007/s11554-012-0246-9) contains supplementary material, which is available to authorized users.

G. Mastorakis (✉) · D. Makris
Faculty of Computing, Information Systems and Mathematics,
Digital Imaging Research Centre, Kingston University,
Surrey, UK
e-mail: g.mastorakis@ieee.org

D. Makris
e-mail: d.makris@kingston.ac.uk

the usage of OpenNI [3] significantly helps the pre-processing of the depth data in terms of background subtraction and user identification.

Other existing approaches share limitations regarding privacy issues since the captured video contains visual information of the involved person in a falling incident. A solution towards protecting privacy is the analysis of data that does not reveal any facial characteristics. Depth data that are derived from the Kinect sensor and used in our system do not contain any identifiable visual information.

1.1 Related work

There is an endless list of fall detection systems which use different technologies and techniques. We will discuss some of those technologies and list them in two different groups, one for non-vision-based solutions and another for vision based. Since our work is specifically based on 3D vision we will also extend our discussion to this particular area.

1.1.1 Non-vision systems

Normally, such systems use wearable motion detectors with accelerometers and gyroscopes [7, 20] capable of detecting the rapid motion changes of the person who wears them. The problem with such detectors is that the person who is supposed to use the device usually forgets or ignores the importance of wearing it. Therefore, no fall is detected as the device is not activated.

Push-alarms [15] are also devices carried by the person prone to falls and can be activated by pushing the alarm after the fall. Similarly, this technology can be very weak as the person may not be carrying the device or may be unable to push the button if the person is unconscious due to the fall.

Acoustic and ambience sensors systems use microphones or vibration sensors. Such systems detect the loudness and height of the sound to recognise a fall [22]. Others, detect the floor vibration [5]. Such systems are limited to indoor use only due to their restrictive application range.

1.1.2 Vision systems

Such systems use image analysis to detect falls for the elderly and general population. They require one [8, 33] or several cameras [6, 9]. 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 to detect a fall's characteristic dynamics and then to monitor target's inactivity [26].

One approach of fall detection is to analyse the velocity of the falling person as proposed by biomechanics [30]. In [24], head's velocity is used to detect a fall using 3D tracking. Their approach may not be robust as they detect 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 [14]. In that study, the authors focus on three types of falls (forward, backward and sideways). While their approach is robust as they can differentiate between falling and lying/sitting, it is considered also limited as the raw data used for their analysis are captured only from the side-view.

1.1.3 3D vision systems

Vision depth image systems use 3D cameras or depth sensors to track and analyse the human motion. Depth image analysis has an important advantage regarding identity protection and privacy, since the delivered data reveal no facial characteristics. There are only a few other previous studies that use a 3D/depth camera/sensor [11, 17, 23]. In the next section, we will further discuss those approaches, as our work lies within this particular area.

1.2 Technical criticism of 3D methods

Since our system analyses the depth information using Kinect's IR sensor we will give a more detailed analysis to emphasise the benefits and weaknesses of the existing approaches. In [17] the authors use a 3D camera to develop an elderly monitoring system, 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) to have resulting images with fewer blobs (assuming that the biggest blob defines the human silhouette). Further, 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 [16].

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 and objects laid). Also, the description of falls or other activities is not defined in their work i.e. one can brutally sit on a sofa; the viewing position may be different; the "lying sequence" comprised of several different postures not properly 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. [11] describe an approach based on the distance of a falling person from the floor, inactivity and pose estimation. The floor is detected using RANSAC [13] 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: first detecting large enough planes and second 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 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 [31]). However, the latter is computationally expensive.

Rougier et al. [23] propose a Kinect-based system to detect falls. Their system first 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 the 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 [32]. The authors claim that their algorithm is able to identify a fallen person while occluded, based on the velocity detection. However, their evaluation is limited to a small number of experiments with no indication about the number of subjects performing. In addition, there is no clear description of the type of falls and their experiments do not include fall-like activity patterns i.e. when someone is picking up something very fast from the floor or lying very fast on the floor or brutally sitting on a sofa. Therefore, there is no proof that a FP is avoided when a person performs a fall-like activity.

As an overall criticism, we can say that all the described 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. Also, another important point is the rather limited number of experiments (raw dataset) used for evaluating those methods except [11].

2 Kinect and OpenNI

Released in 2010 by Microsoft, Kinect is the fastest selling consumer electronics device [1]. Kinect is the first game console that does not require any push remote/joystick or any other form of control device as it is designed to

accurately recognise the human motion and translate it into commands/actions. Kinect uses three types of sensors: an RGB camera, an IR and an acoustic sensor; all developed by PrimeSense [4]. Our system uses the IR sensor. Our captured videos have 640×480 resolution at 30 fps, although the maximum resolution delivered by Kinect is $1,200 \times 960$ at 30 fps. The maximum range of Kinect's IR sensor is 10 m but the actual effective range depends on the environment. Practically, depth images appear noisy enough to cause misinterpretations beyond 7 m.

First attempt to use Kinect in non-gaming applications was achieved by reverse engineering at the end of 2010 [2]. Since then, numerous applications have been developed for action recognition and augmenting reality.

One of the most important development tools for Kinect is OpenNI [3]. OpenNI is an open source tool also provided by PrimeSense. With OpenNI, the developer can access the depth information of a human subject and estimate and track its articulate pose which can be used for human tracking, gesture and motion recognition. Furthermore, OpenNI allows the developer to change and add new routines and processes to enhance or extend the capabilities of the existing tools. Nevertheless, OpenNI does not provide access to the motor, hence, no information about the tilt angle or accelerometer can be delivered.

3 Methodology

In this section, we will describe our technique for fall detection. Our 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 our analysis we use only three parameters as estimated by OpenNI, i.e. the width, height and depth of the human posture, which define a 3D bounding box. This simplified set of parameters delivers a more reliable result than articulated pose estimation. From our early experiments we found that pose estimation may fail during the 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 section discusses in more detail the 3D bounding box extraction, while in the following sections we describe how the 3D bounding box's parameters are used to detect a fall.

It is not a requirement for our algorithm to calculate and use the floor coordinates as previous approaches do (see Sect. 1.2). Further to that, we must note that a fall is a fast activity and a high frame rate in real-time systems is advisable to avoid missed detections.

3.1 Overview

The 3D bounding box is created using OpenNI's Depth-MetaData 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 of 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 we see in Fig. 1. Traditionally, the position of the 3D bounding box is tracked to estimate the motion of humans or other objects. In our 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 operation of our algorithm runs inside OpenNI's main loop of the depth map process. In Algorithm 1 we describe the operation of our method. The next section explains the different steps of our algorithm.



Fig. 1 Depth map of the scene. User is identified by OpenNI

values, as well as their first derivatives at each frame to detect a fall.

Algorithm 1 Pseudocode of fall detection algorithm

```

SET threshold  $T_{vH}, T_{vWD}$ 
SET threshold  $T_{iH}$ 
SET counter  $a, b = 0$ 
SET counter  $falling = Nframes$ 
SET counter  $inactivity = Mframes$ 
SET boolean  $activityDetection = false$ 
SET boolean  $inactivityDetection = false$ 
while run do
  if  $(vH > T_{vH})$  and  $(vWD > T_{vWD})$  then
    if  $a = falling$  then
      SET  $activityDetection = true$ 
    end if
    SET  $a++$ 
  end if
  if  $activityDetection = true$  and  $absVh < T_{iH}$  then
    if  $b = inactivity$  then
      SET  $inactivityDetection = true$ 
      SET  $Fall\ detected$ 
    end if
    SET  $b++$ 
  end if
end while

```

3.2 3D Bounding box data analysis

As we described in the previous section, each user is wrapped into a 3D bounding box. The dimensions of the 3D bounding box is the only input our algorithm requires to operate with. OpenNI analyses each frame and a new 3D bounding box is fitted every time with a new set of width, height and depth values. Our algorithm analyses those

Several studies discuss the fact that during the fall, the width of the 2D bounding box is expanding, while the height is contracting [27, 28]. Those studies require the initial and final aspect ratio of the 2D bounding box to confirm a fall, while our approach does not measure the initial/final bounding box dimensions. Figure 2 shows a fall detected by a sensor placed from a side view. In our case, we will use a 3D bounding box which behaves

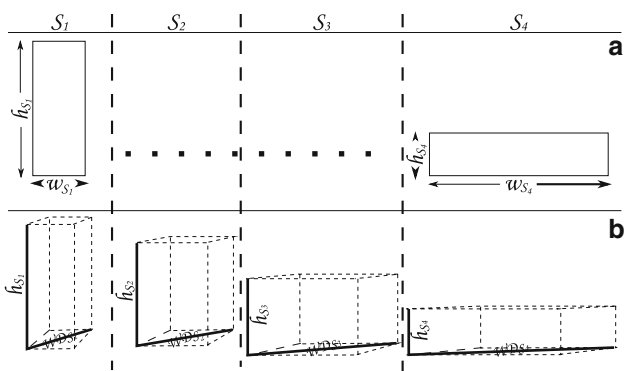


Fig. 2 2D Bounding box during a fall; the height reduces while the width increases (a) as seen in [27, 28], where the initial and final bounding box dimensions are required. Our approach using a 3D bounding box of the height and the composition of width and depth (b)

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. We combine the two expanding dimensions of the 3D bounding box W and D . The composition of depth–width $WD = \sqrt{D^2 + W^2}$.

We have split the fall event into states $S = S_1, S_2, S_3, S_4$. For each state we have the height $H = H_1, H_2, H_3, H_4$ and the width–depth $WD = WD_1, WD_2, WD_3, WD_4$. 3D Bounding box’s height first derivative is defined by $v_H = \frac{H_i - H_{i-1}}{t_i - t_{i-1}}$ in a particular state S_i with H_i . Similarly, the velocity of the composition is defined as $v_{WD} = \frac{WD_i - WD_{i-1}}{t_i - t_{i-1}}$.

In Fig. 3 we see 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. We have also noticed that the signal delivered by OpenNI is quite noisy, especially in regard to Z dimension, therefore, we use a Kalman filter [29] to smooth the velocities as seen in Fig. 3.

3.2.1 Fall initiation by velocity

Human motion is articulated and therefore quite 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 [30]. However, estimating the centre of mass may be very complicated. Instead, our algorithm measures the velocities of height and the composite vector of width and depth. The resulted v_{WD} and the v_H are checked during N sequential frames. The velocity thresholds for the height T_{vH} and the width–depth composite vector T_{vWD} of the bounding box, as well the duration of the fall (N frames) are estimated by performing random search [25] that optimises the classification score in a training dataset.

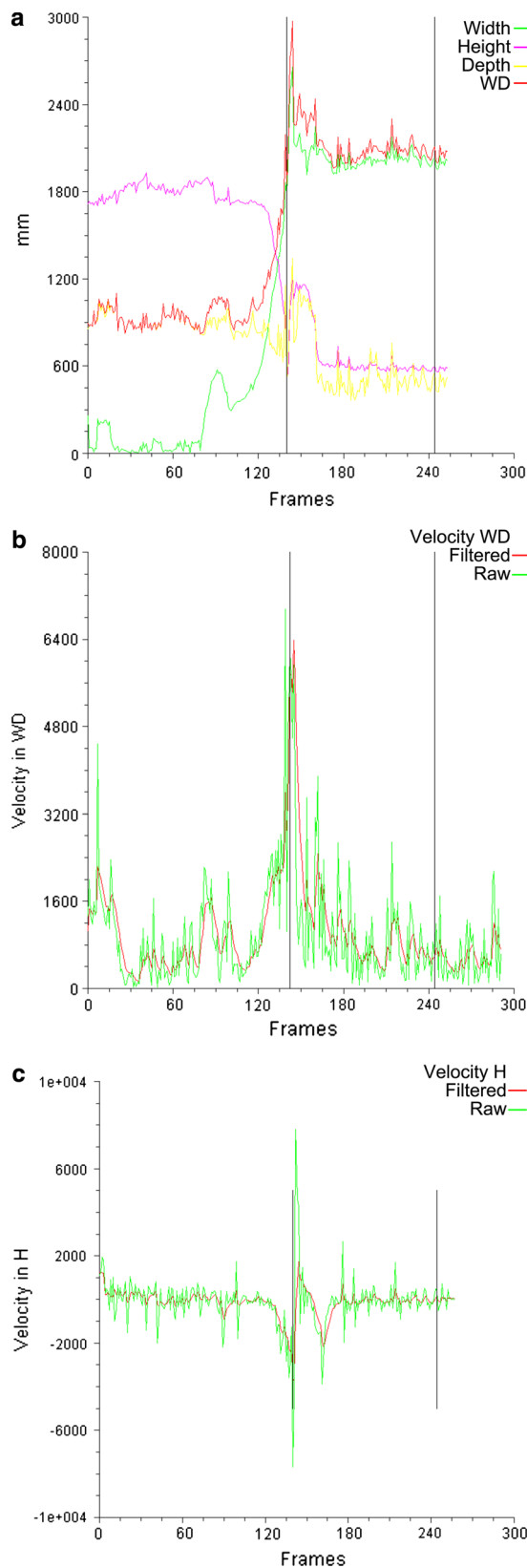
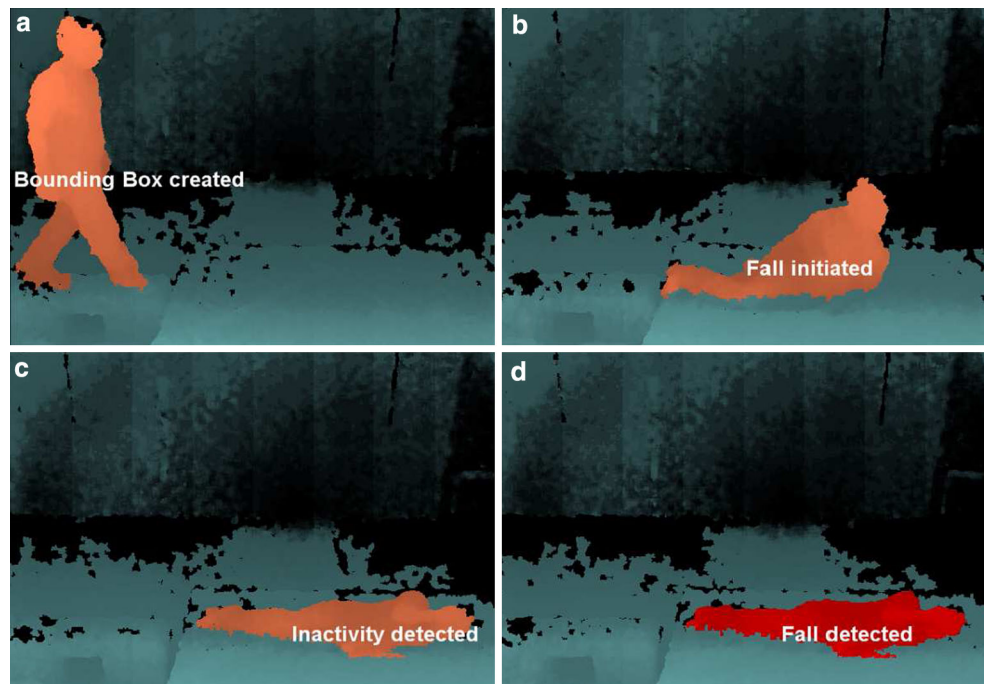


Fig. 3 Width, height, depth distances and width–depth, height velocities of the 3D bounding box during a sideways view fall. Smoothed velocities for WD and H show the improvement of the signal

Fig. 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)



When both velocities (v_{WD} , v_H) exceed particular thresholds (e.g. 3D bounding box's height velocity, etc.), fall initiation is detected. The next paragraph discusses the final step. In Fig. 4b we see the visual result of velocity detection for side fall, captured sideways.

3.2.2 Completion state of a fall by inactivity detection

A fall always ends at an inactivity state where no motion is detected (i.e. resting place). Therefore, the fall completion is detected by checking the appropriate velocity condition. Specifically, our method involves monitoring the subject for some time (e.g. 2 s) to detect any motion (Figs. 4c, 6c, 12c). If no motion is occurred then the algorithm is flagged as “Fall Detected” (Figs. 4d, 6d, 12d). It is only required for the height velocity (v_H) to be less than a certain threshold to declare the state as inactive.

4 Experimental results and discussion

4.1 Experimental setup and dataset

The initial step for setting such a system requires direct view of the scene where a fall is possible to occur. For that reason, Kinect has been attached to a tripod at the height of 204 cm and inclined to the floor plane. Due to the sensitivity of the sensor, Kinect has been placed at a distance no farther than 7 m from the area of a possible fall.

For our evaluation we captured 184 video samples of actions that included: 48 falls (backward, forward and

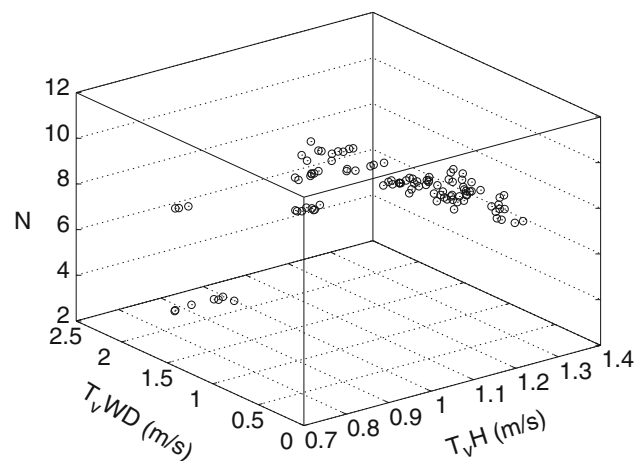


Fig. 5 Circles indicate 100 triplets estimated by random search. Their median ($t_{v,H} = 1.18\text{m/s}$, $T_{v,WD} = 1.20\text{m/s}$, $N = 8$ frames) is marked as a cross and is used for our experiments

sideways), 32 seating activities, 48 lying activities on the floor (backward, forward and sideways) and 32 “picking up an item from the floor” activities, performed by eight different subjects. Other activities that change the size of the 3D bounding box were also performed (i.e. sweeping with a broom, dusting with a duster).

In addition, we instructed two subjects to perform in slow motion to imitate the behaviour of an elderly person. Therefore, slow falls and other actions were performed to demonstrate how the algorithm operates in such cases. We believe that this approach (i.e. adult subjects performing

Fig. 6 Fall detected in a front (angled) view

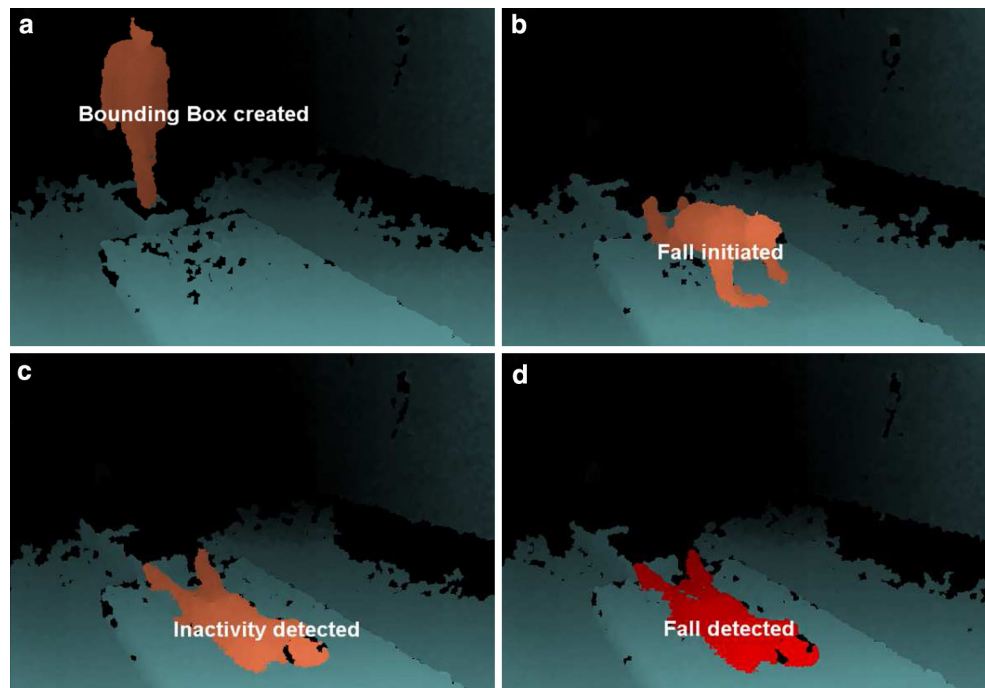
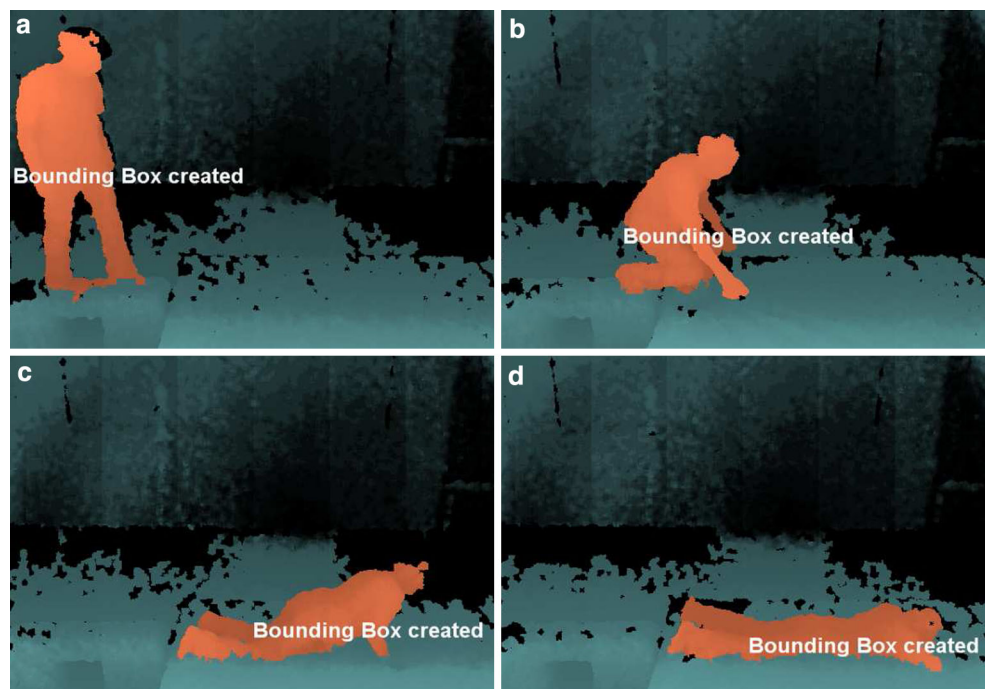


Fig. 7 Lying on the floor



slow activities) is more feasible and ethical in an effort to simulate falls within the elderly population. We captured 12 such videos of slow falls and another 12 videos of other slow activities (sitting, sweeping, lying down etc).

Those videos were captured from three different angles to imitate several different views of an activity in a real environment. Subjects performed the fall actions on a 30-cm thick mat to allow realistic performance of falls.

4.2 Training

The dataset was split into a training and a testing set; the former consisted of 12 falls and 22 non-fall video samples from four subjects, while the latter consisted of all the rest. We ensured that some extreme cases (slow/fast falls, sitting, lying etc.) were included in the training set to cover all the intermediate case activities.

Fig. 8 Sitting brutally on a chair

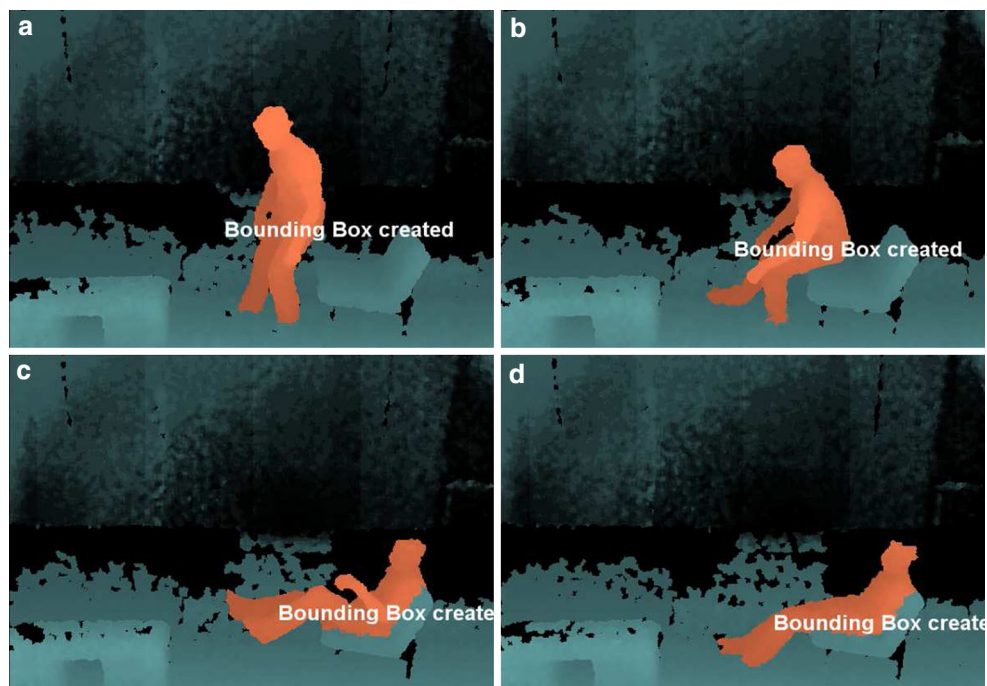
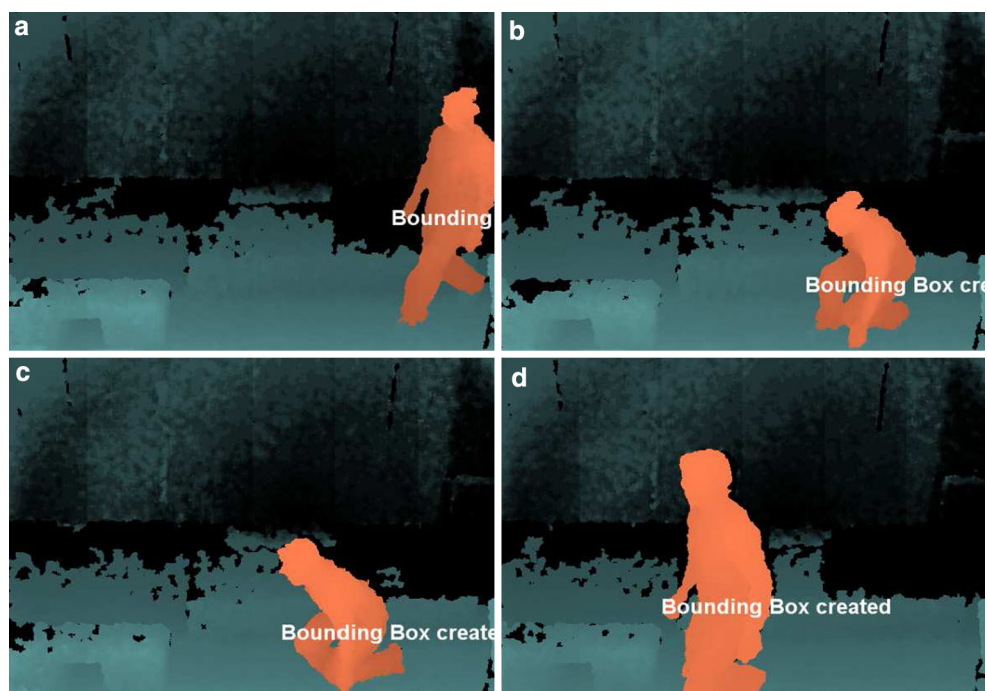


Fig. 9 Picking up an item from the floor in fast motion

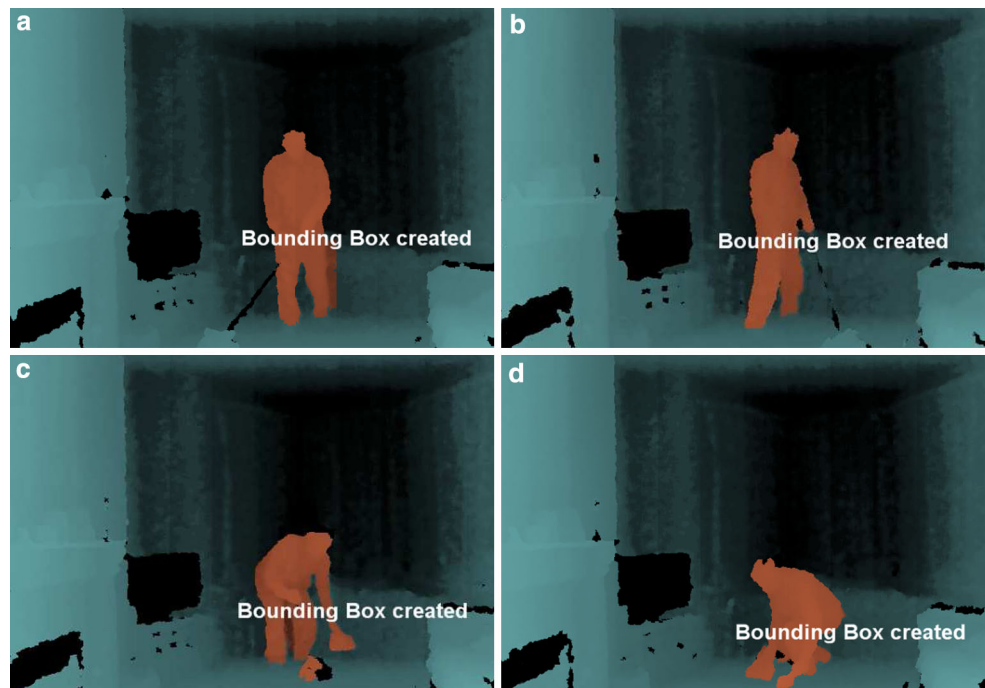
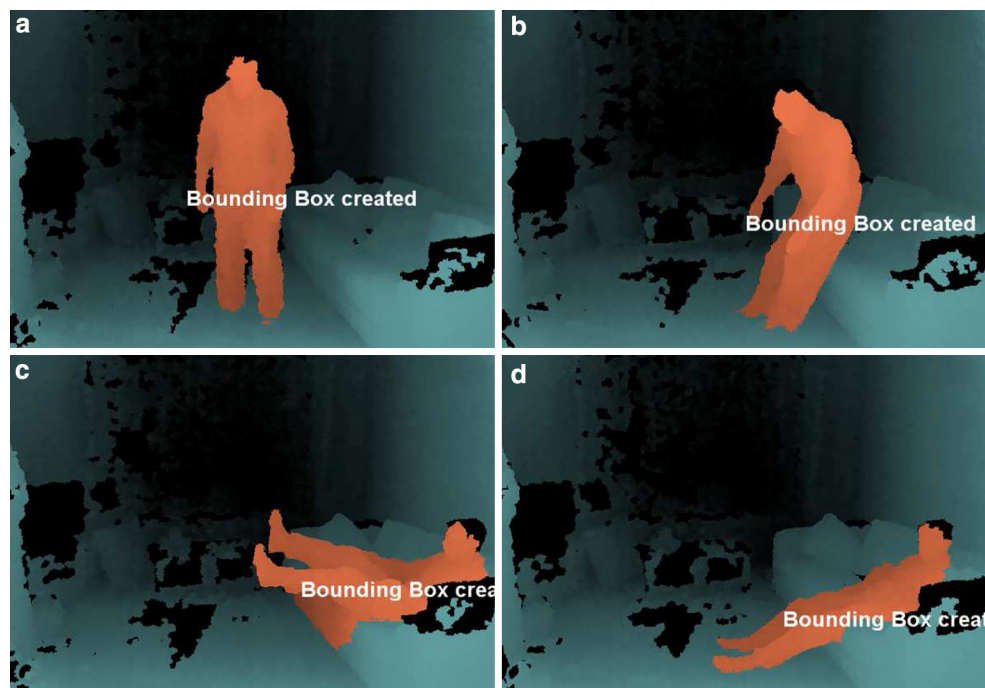


The threshold values for velocity T_{vH} , T_{vWD} as well as the duration N of the fall in frames were estimated by performing random search on the training dataset multiple (100) times. Since the fall and non-fall sequences of the training dataset were separable, many triplets that gave 100% classification score were found. We analysed the testing set using the median values of those triplets which are considered reliable estimates of our method's

parameters (see Fig. 5). The velocities derived from our training confirm the values obtained from [30] where fall-related velocities are above 1 m/s.

4.3 Results

We have measured the per frame processing time and we found it to be around 0.3–0.4 ms (Intel Core Duo,

Fig. 10 Sweeping activity**Fig. 11** Brutally sitting on a sofa

2.4 GHz). We have produced a series of visual results to demonstrate the variety of our experiments: forward fall, front view (Fig. 6), lying on the floor, front view (Fig. 7), sitting on a sofa, side view (Fig. 8), picking up an item from the floor, side view (Fig. 9). All falls were accurately detected (i.e. no missed detections).

Another set of experiments includes more specific actions, such as sweeping (Fig. 10) and brutally sitting

(Fig. 11). Sweeping, changes the 3D bounding box mostly in Z, X while the velocity is not reaching any of the thresholds (T_{vH}, T_{vWD}). Brutally sitting is a case where the motion is not lengthy 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.

To further test our algorithm we have performed a set of slow falls (Fig. 12) to imitate an elderly person's actions.

Fig. 12 Slow fall

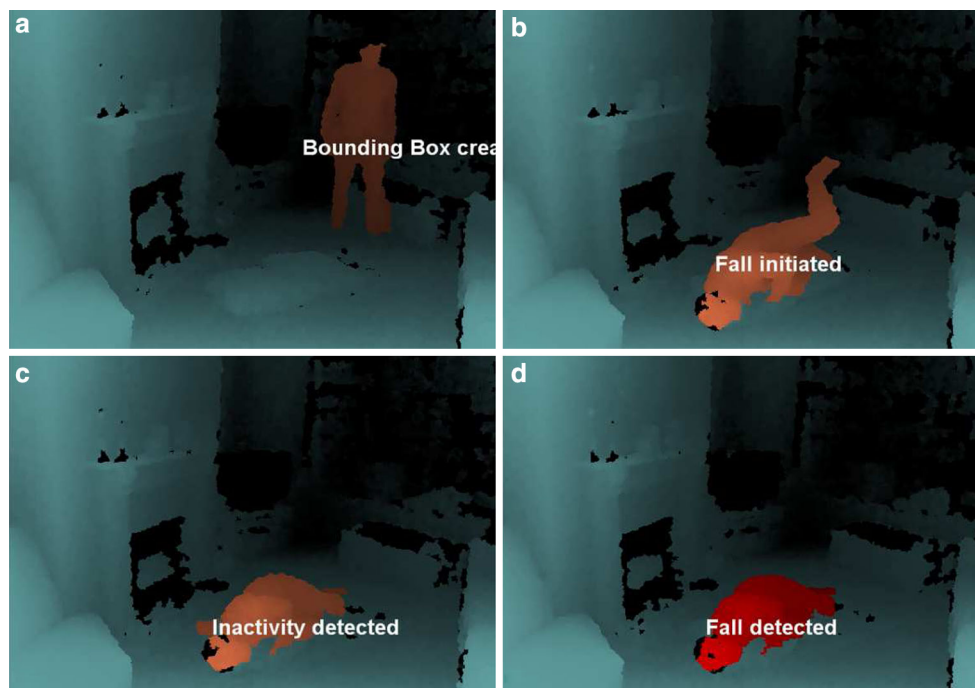
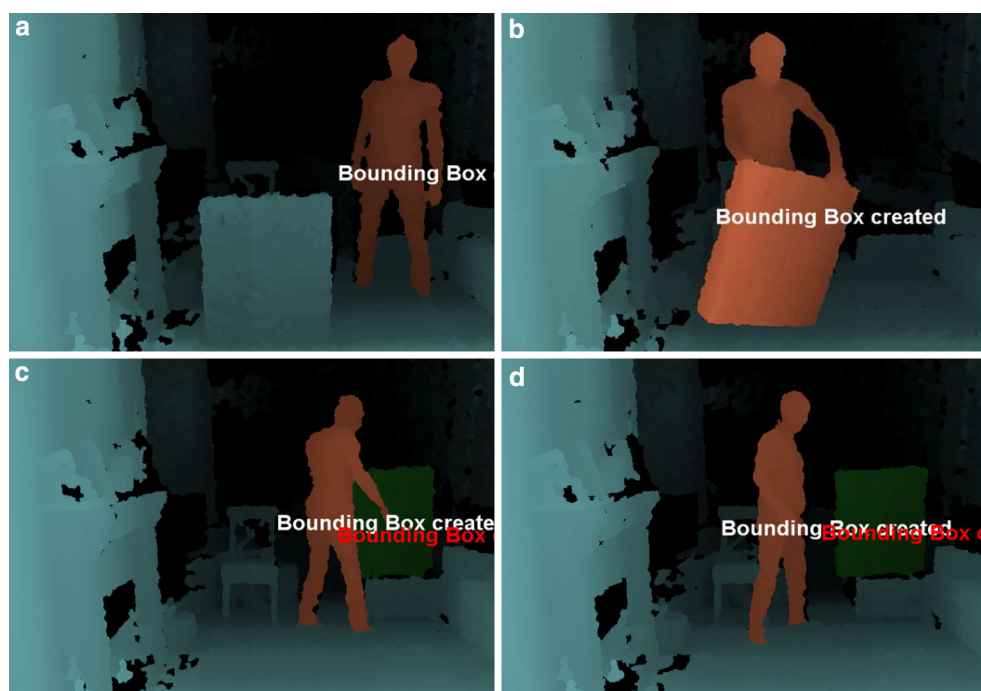


Fig. 13 Picking up and dropping a box

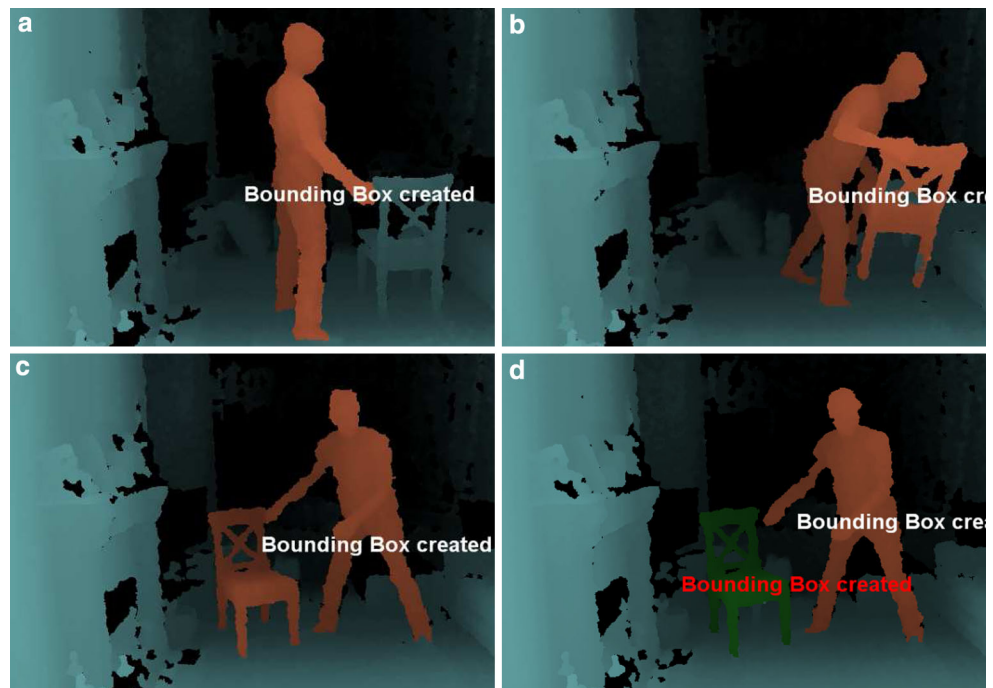


An elderly person's fall is slower at the beginning of the action. However, as the falling action progresses, gravity and lack of balance increases the velocity of the fall and therefore, these kind of falls are detected by our algorithm.

Finally, we tested our algorithm with 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, we captured 40 additional videos

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 the 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 height dimension is observed. Therefore, although the v_{WD} velocity may be increased the v_H remains at normal levels,

Fig. 14 Picking up and dropping a chair



hence, no fall detection is initiated. We used a large box to investigate how our method performs in those scenarios, since the box would dramatically change the size of the 3D bounding box. Figures 13, 14 show two of the set images from the above experiments.

The algorithm was proved stable, even when half of the subject's body was occluded by the box. This is because the v_{WD} remains at normal levels (i.e. well below the $T_{v_{WD}}$) while the v_H exceeds the T_{v_H} . Therefore, a fall detection will not be initiated since both v_{WD} , v_H must be above their thresholds.

The bounding box as we see in Figs. 13, 14 will split into two different bounding boxes (one for the subject and the other 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 will still be 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) falls too, this may also be detected as a fall.

5 Conclusion

We have developed a robust walking fall detection system that requires no pre-knowledge of the scene. We have managed to isolate and analyse the fall event as an independent activity without specifying or detecting any external parameter set such as the floor plane coordinates. The simple and lightweight algorithm has negligible computational time (0.3–0.4 ms) and is capable of

detecting any walking fall without FPs caused by non-fall actions, i.e. sitting brutally on a chair, lying on the floor or crouching down (i.e. fast action).

While previous fall detection approaches use a particular point of the body such as the head or the centre of mass to measure the falling velocity, our approach is based on the analysis of the 3D bounding box's first derivatives.

Taking the above into account, our algorithm can be characterised as one of the reduced complexity that requires three parameters to operate; the width, height and depth of the subject. Our system is fast, robust and uses an inexpensive sensor, therefore, it can be easily applied on a large scale for reliable fall detection. With its generic application, our system can be used in the general population and also contribute in supporting independent living of the elderly.

Acknowledgments We thank Dr. Jesus Martinez del Rincon and Andy Strong for their invaluable assistance in setting up the experimental environment for the capturing sessions.

References

1. BBC News-Microsoft Kinect 'fastest-selling device on record'. <http://www.bbc.co.uk/news/business-12697975>. Last accessed 17 May 2011
2. Openkinect. <http://openkinect.org/>. Last accessed 17 May 2011
3. OpenNI. <http://www.openni.org>. Last accessed 17 May 2011
4. Primesense. <http://www.primesense.com/>. Last accessed 17 May 2011
5. Alwan, M., Rajendran, P., Kell, S., Mack, D., Dalal, S., Wolfe, M., Felder, R.: A smart and passive floor-vibration based fall

- detector for elderly. In: *IEEE Proceedings of International Conference on Information and Communication Technologies: From Theory to Applications*, pp. 1003–1007 (2006)
6. Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection with multiple cameras: an occlusion-resistant method based on 3-d silhouette vertical distribution. *IEEE Trans. Inf. Technol. Biomed.* **15**(2), 290–300 (2011)
 7. Bourke, A., O'Brien, J., Lyons, G.: Evaluation of a threshold based tri-axial accelerometer fall detection algorithm. *Gait Posture* **26**(2), 194–199 (2007)
 8. Cucchiara, R., Prati, A., Vezzani, R.: An intelligent surveillance system for dangerous situation detection in home environments. *Intell. Artif.* **1**(1), 11–15 (2004)
 9. Cucchiara, R., Prati, A., Vezzani, R.: A multi-camera vision system for fall detection and alarm generation. *Exp. Syst. J.* **24**, 334–345 (2007)
 10. Deandrea, S., Lucenteforte, E., Bravi, F., Foschi, R., La Vecchia, C., Negri, E.: Risk factors for falls in community-dwelling older people: a systematic review and meta-analysis. *Epidemiology* **21**(5), 658–668 (2010)
 11. Diraco, G., Leone, A., Siciliano, P.: 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*, pp. 1536–1541 (2010)
 12. Donald, I., Bulpitt, C.: The prognosis of falls in elderly people living at home. *Age Ageing* **28**(2), 121–5 (1999)
 13. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**, 381–395 (1981)
 14. Foroughi, H., Aski, B., Pourreza, H. (2008) Intelligent video surveillance for monitoring fall detection of elderly in home environments. In: *International Conference on Computer and Information Technology*, pp. 219–224
 15. Hori, T., Nishida, Y., Aizawa, H.: Sensor network for supporting elderly care home. In: *IEEE Proceedings of Sensors* **2**, 575–578 (2004)
 16. Jansen, B., Deklerck, R.: Context aware inactivity recognition for visual fall detection. In: *Proceedings of International Conference on Pervasive Computing Technologies for Healthcare*, pp. 1–4 (2006)
 17. Jansen B., Temmermans F., Deklerck R.: 3D human pose recognition for home monitoring of elderly. In: *IEEE International Conference of the Engineering in Medicine and Biology Society*, pp. 4049–4051 (2007)
 18. Kim, K.J., Ashton-Miller, J.A.: Biomechanics of fall arrest using the upper extremity: age differences. *Clin. Biomech.* **18**(4), 311–318 (2003)
 19. Leclercq, S.: In-company same- and low-level falls: from an understanding of such accidents to their prevention. *Int. J. Ind. Ergonomics* **25**(1), 59–67 (2000)
 20. Li, Q., Stankovic, J., Hanson, M., Barth, A., Lach, J., Zhou, G.: Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In: *Proceedings of International Workshop on Wearable and Implantable Body Sensor Networks*, pp. 138–143 (2009)
 21. Londei, S.T., Rousseau, J., Ducharme, F., St-Arnaud, A., Meunier, J., Saint-Arnaud, J., Giroux, F.: An intelligent videomonitoring system for fall detection at home: perceptions of elderly people. *J. Telemed. Telecare* **15**, 383–390 (2009)
 22. Popescu, M., Li, Y., Skubic, M., Rantz, M.: An acoustic fall detector system that uses sound height information to reduce the false alarm rate. In: *IEEE International Conference of the Engineering in Medicine and Biology Society*, pp. 4628–4631 (2008)
 23. Rougier, C., Auvinet, E., Rousseau, J., Mignotte, M., Meunier, J.: Fall detection from depth map video sequences. In: *Abdulrazak, B., Giroux, S., Bouchard, B., Pigot, H., Mokhtari, M. (eds.)* *Toward Useful Services for Elderly and People with Disabilities. Lecture Notes in Computer Science*, vol. 6719, pp. 121–128. Springer, Berlin (2011)
 24. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Monocular 3D head tracking to detect falls of elderly people. In: *IEEE International Conference of the Engineering in Medicine and Biology Society*, pp. 6384–6387 (2006)
 25. Schumer, M.A., Steiglitz, K.: Adaptive step size random search. *IEEE Trans. Autom. Control* **13**(3), 270–276 (1968)
 26. Sixsmith, A., Johnson, N.: A smart sensor to detect the falls of the elderly. *IEEE Pervasive Comput* **3**(2), 42–47 (2004)
 27. Toreyin, B.U., Dedeoglu, Y., Cetin, A.E.: HMM based falling person detection using both audio and video. In: *Sebe, N., Lew, M.S., Huang, T.S. (eds.) IEEE International Workshop on Human-Computer Interaction. Lecture Notes in Computer Science*, vol. 3766, pp. 211–220. Springer, Berlin (2005)
 28. Vishwakarma, V., Mandal, C., Sural, S.: Automatic detection of human fall in video. In: *Proceedings of International Conference on Pattern Recognition and Machine Intelligence, PReMI'07*, pp. 616–623. Springer, Berlin (2007)
 29. Welch, G., Bishop, G.: *An Introduction to the Kalman Filter*. University of North Carolina at Chapel Hill (1995)
 30. Wu, G.: Distinguishing fall activities from normal activities by velocity characteristics. *J. Biomech.* **33**(11), 1497–1500 (2000)
 31. Xiao, Y., Siebert, P., Werghi, N.: A discrete reeb graph approach for the segmentation of human body scans. In: *IEEE Proceedings of International Conference on 3-D Digital Imaging and Modeling* (2003)
 32. Zhao, J., Katupitiya, J., Ward, J.: Global correlation based ground plane estimation using v-disparity image. In: *IEEE Intl. Conf. on Rob. and Aut.*, pp. 529–534 (2007)
 33. Zouba, N., Boulay, B., Bremond, F., Thonnat, M.: *Cognitive vision. In: Monitoring Activities of Daily Living (ADLs) of Elderly Based on 3D Key Human Postures*, pp. 37–50. Springer, Berlin (2008)

Author Biographies

Georgios Mastorakis studied vehicular engineering in a technical school in Athens, Greece in 1998. He moved to UK to start his studies in Computer Science at Royal Holloway in 2002. He worked with Prof. Fionn Murtagh on his Image Processing dissertation regarding a surveillance application. He later on joined the Vision and Signal Processing group in Physics Department at Royal Holloway in 2006 working closely with Prof. E.R. Davies. Georgios completed his research on an on-board vision based vehicle detection and tracking system and has published relevant publications. In 2011 he became a Visiting Researcher at Kingston University, London joining the Digital Imaging Research Centre (DIRC), during which time he worked closely with Dr Makris to develop non-gaming applications using Xbox Kinect.

Dimitrios Makris received the diploma in Electrical and Computer Engineering from Aristotle University of Thessaloniki, Greece in 1999 and the PhD in Computer Vision from City University, London in 2004. He is currently a Reader (Associate Professor) in the School of Computing and Information Systems, Kingston University, London. His research interests are in the area of image processing, computer vision and machine learning and especially in motion analysis.