FADES: Behavioral detection of falls using body shapes from 3D joint data

Hee Jung Yoon, Ho-Kyeong Ra, Taejoon Park, Sam Chung and Sang Hyuk Son

Abstract. Many efforts have been made to design classification systems that can aid the protection of elderly in a home environment. In this work, we focus on an accident that is a great risk for seniors living alone, a fall. Specifically, we present FADES, which uses skeletal joint information collected from a 3D depth camera to accurately classify different types of falls facing various directions from a single camera and distinguish an actual fall versus a fall-like activity, even in the presence of partially occluding objects. The framework of FADES is designed using two different phases to classify the detection of a fall, a non-fall, or normal behavior. For the first phase, we use a classification method based on Support Vector Machine (SVM) to detect body shapes that appear during an interval of falling behavior. During the second phase, we aggregate the results of the first phase using a frequency-based method to determine the similarity between the behavior sequences trained for each of the behavior. Our system shows promising results that is comparable to state-of-the-art techniques such as Viterbi algorithm, revealing real time performance with latency of <45 ms and achieving the detection accuracy of 96.07% and 95.7% for falls and non-falls, respectively.

Keywords: Fall detection, Kinect skeletal joint data, home assistance, real-time processing

1. Introduction

As the Baby Boomer generation ages, the need for elderly care significantly increases, especially in emergency situations when they live alone or are left at home unsupervised. Of many unintentional home injuries, falls are the most common cause of harm amongst the elderly population, resulting in hip fractures, severe head injuries, and joint dislocations. Falling is also the leading cause of accidental death in this age group. These incidents have a tremendous impact on medical cost. In U.S., more than 20 billion dollars are spent annually on seniors treating injuries caused by falls [9]. When a person has fallen, it is important that the individual immediately receives help to prevent further damage to his or her health and the need to pay for expensive medical costs. To provide a convenient, cost-effective method of detecting falls for elderly individuals, we specifically focus on real-time fall detection using a 3D depth camera.

Two kinds of sensing techniques are generally used on a state-of-the-art fall detection system: invasive and non-invasive. Invasive techniques use physiological features such as acceleration readings from wearable sensors to detect falls. These sensors are to be worn at all times and are not convenient for long-term use. Non-invasive techniques typically use surveillance cameras. Although these may be more practical for long-term monitoring in home environments, they do not provide information at the skeleton level needed to understand human behavior. An exception is a Kinect sensor [33] which provides skeletal joint information using a non-invasive infrared camera. Based on this observation, in this work, we address the problem of detecting falls using Kinect.
Kinect sensors have been successfully used for fall detection [15,18,19,26,29], gait analysis [11,21,34], and gesture recognition [3,28,40]. Nevertheless, reliably detecting a fall with Kinect is difficult for many reasons. First, when monitoring people in a physical environment, the system should take into account the fact that a given activity may vary from person to person and that there exist different types of falls such as slipping, tripping, and loss of balance. Hence, the system should be flexible enough to precisely specify what constitutes many different types of falls of all directions from the camera. Second, false alarms are very common in any detection system. This is especially important for our study, where sensor readings from camera sensors in a practical deployment can be extremely noisy. Proper handling and reducing false alarms is crucial, specifically differentiating actual falls from fall-like activities such as lying-down and sleeping, and normal daily activities such as walking and cleaning. Third, the detection process must be quick and precise so that the emergency situation can be rapidly resolved. This is crucial for our study where a fall must be detected in real-time to help fallen individuals as quickly as possible.

In this paper, we present FADES, a system that detects different types of falls in a home environment using 3D joint data from Kinect. FADES takes a machine learning-based approach that accepts raw skeletal joint data as an input and performs training in two separate phases: one for body shape detection and the other for behavior sequence detection. The first phase detects the body shapes of the falls, while the second aggregates the results of the first phase to determine the similarity between the behavior sequences of the shapes with the actual behavior. The system is designed to detect falling behavior even when there are partially occluding objects, and allow the system to be capable of accurately distinguishing an actual fall versus a fall-like activity (or a normal daily activity) by testing the input data stream with pre-trained classifiers. For each type of fall, we have considered all possible body shapes that characterize a fall, and therefore, can accurately detect the behavior from any direction of a single camera regardless of the type of fall.

We evaluate FADES in a real-world single-person household and demonstrate how FADES accurately and promptly detects different types of falls, including non-fall and normal activities, in various orientations of the person with respect to the camera, even in the presence of partially occluding objects. To compare our work with a state-of-the-art technique, we test our second phase using Viterbi algorithm. We show that FADES is less complex while providing similar accuracy than that of advanced approaches. Our performance evaluation results demonstrate that it achieves the detection accuracy of 96.07% and 95.7% for falls and non-falls, respectively, while satisfying the requirement of real-time response with latency of <45ms. Moreover, we show that a multi-camera setup would provide more reliability in tracking skeletal joint information. As such, this work provides empirical evidence that FADES is more accurate and more effective for practical deployment than existing fall detection techniques.

The remainder of the paper is organized as follows. In Section 2, we introduce several fall detection approaches. We start our technical discussions by presenting the implementation process of our proposed system in Section 3 and characterizing the falling behavior in Section 4. Using this data and our observations, the design of FADES is presented in Section 5 and performance evaluations in Section 6. Finally, we position our work among others in Section 7 and conclude the paper in Section 8.

2. Background – Current fall detection approaches

An injury from an unintentional fall may result in limited or reduced mobility. If the person is unable to get up from a fall, the damage may increase and cause medical illnesses such as blood clots in the legs or general weakness [2]. As rapid mobilization prevents these problems, many different types of approaches have been proposed to find a solution.

2.1. Accelerometer

A common approach to detect falls is to use wearable devices such as an accelerometer [5] which is an electromechanical device that measures acceleration forces. As part of the process of measuring acceleration, the accelerometer yields information of vibration, inclination, and shock that contributes to the detection of falls.

One of the common examples of modern accelerometers is a MicroElectroMechanical System (MEMS) device [24]. This is a simple device that functions mainly with the cantilever beam and circuits that are designed for detecting the presence of deflection senses. A gyroscope [4], which is a device for measuring body orientation based on the principle of angular
momentum, is also used along with the accelerometer in order to detect falls at an earlier stage. Another related fall detection approach is based on applying accelerometers onto Android smartphones [8,35]. The major drawback of these techniques is that the sensor devices (accelerometers and/or gyroscopes) are to be worn or carried everywhere, necessitating the use of batteries that need to be recharged regularly for adequate functioning. If in any case such devices are forgotten to be worn or recharged, which is common for elderly individuals, no fall will be detected as the devices will not be triggered. Although we do not limit the use of \((x, y, z)\) coordinate data from accelerometers for the two phase training, utilizing the Kinect sensor has a merit in addressing the concerns from battery-powered devices, as well as in significantly enhancing the safety of the person in emergency situations.

2.2. Floor vibration

In contrast to accelerometers, a floor vibration-based fall detector [1] is another method for detecting falls. The approach is based on the detection of fall vibration patterns that are significantly different from those generated by normal daily activities. The main advantage is that floor vibration sensors are inexpensive and capable of preserving privacy. However, the performance is insufficient when the detection varies with different floor dynamics [29].

2.3. Video-based fall detection

Video surveillance offers a promising solution for automatic fall detection, as no body-worn devices are needed. There have been many previous researches on detecting falls using image processing techniques. A commonly used method is to analyze the ratio of the moving object’s bounding box [37], which extracts time-series signals describing the motion of a person. Due to problems of occluding objects, Rougier et al. [30] proposed to fit an ellipse on the foreground area, using a ceiling-placed, wide-angle camera. They also proposed a shape matching technique to track the person’s silhouette along the video sequence [31]. Furthermore, the work by Lee and Mihailidis [16] used a person’s silhouette and the 2D image velocity to detect falls with thresholds for normal zones of inactivity. Nait-Charif and McKenna [23], on the other hand, used an ellipse representing the person to detect inactivity outside usual inactivity zones such as chairs and sofas. Despite the achievements that these researches have provided for visual fall detection, some challenges still remain.

– Video surveillance systems need to be robust against image processing difficulties, especially background modeling and object segmentation with issues of high video compression, shadows and reflection, and cluttered background. Problems occur when there is a poor job of extracting all relevant pixels required to track a person.

– Visual fall detection is prone to high levels of false positives. Differentiating falling behavior versus a normal daily activity is difficult to accomplish. As described by Foroughi [10], many current surveillance fall detection systems [7,16,20,36] are unable to discriminate between an actual falling incident and an event when a person is merely lying down.

The challenges from video-based methods have brought many recent studies [15,18,19,26,29] to come up with novel approaches by utilizing Kinect. We compare and evaluate our work with Kinect-based fall detection systems later in Section 7.

3. Implementation

More and more researches are utilizing vision-based techniques to capture, and work with, visual information for various purposes including the development of video surveillance techniques with less human configuration. To tackle the similar motive, we use Kinect, which is a simpler and more effective way of capturing 3D skeletal joint data than using a standard camera.

Included in the device are both infrared (IR) and RGB cameras. The depth image size has a maximum resolution of 640 by 480 and it processes depth data with a frame rate of 30 fps. OpenNI [25] and PrimeSense [27] drivers, along with its libraries, are used for tracking individual skeletal joints with Kinect.

The skeletal joint information is very effective for fall detection as it becomes possible to access data at a skeleton level and precisely track a person’s body shape. Another important advantage is that a person can be detected even in areas of low lighting, and moreover, privacy can be preserved by utilizing only the skeletal information rather than RGB images. For our study we are not storing skeletal information, rather FADES monitors and detects falls in real-time. Note that the Kinect skeletal model is automatically initialized when a person enters the scene without requiring calibration.
For convenient usage of 3D joint data during both the training and the testing processes, we implemented a graphic user interface along with the skeleton tracker video frame. We developed this standalone system by combining the OpenNI and LIBSVM [13] library, both an open source tool, in order to train and classify our data with our approach. OpenNI is a resourceful tool that allows developers to access the skeletal information of a human for tracking purposes. LIBSVM is an integrated software package for SVM, regression, and distribution estimation.

4. Characterizing behavior

Characterizing behavior is a vital part for our system. In this section, we describe the important factors that must be considered for behavior characterization.

4.1. Types of falls

Studies have shown that the common types of falls experienced by elderly individuals in a home environment are slipping, tripping, and loss of balance [6,22,32,39]. Slipping happens when a person loses balance due to low friction between a person’s feet and the walking surface, or when the surface is contaminated with slippery substance such as water, ice, oil, food scraps, or small objects. A trip or stumble is a sudden arrest of movement of a foot with continued motion of the body. Usually, objects on the floor contribute to these events. Good balance is dependent on many factors such as reliable vision, muscle strength, joint mobility, and proprioceptors. Elderly individuals are prone to variety of diseases that affect these systems and result in poor balance which can eventually lead to falls.

A common characteristic is shared among all these types of falls. The upper part of a person’s body is unintentionally falling toward the ground in high velocity. We use this analysis to take our study a step further by considering all possible orientations of human body during a fall with respect to the single Kinect. To do this, we introduce four directions in which any types of falls can be directed towards: forward, backward, right, and left falls. We further divide these directions into four subtypes, each facing: front of camera, back of camera, right of camera, and left of camera. This yields 16 orientations having similar characteristics of falling downwards. These orientations are used in our training process so that the system can fully understand different types of falls such as loss of balance, slips, and trips, which we evaluate in a later section.

4.2. Skeletal joints

When defining a fall, the 3D skeletal joints must be efficiently utilized so that the system can clearly understand the behavior. We found that all joints are not necessarily required when detecting a fall and hence use four primary joints: head, right shoulder, left shoulder, and torso. Listed below are some of the reasons why we use these specific joints.

– The primary joints are fixed when moving towards a downward direction, and therefore, they are effective when defining succinct descriptions of the behavior. Using all joints in the body builds enormous data complexity because not everyone has similar body motions when falling down. For example, depending on how a person would fall, the wrist joint can be positioned near the body or swinging away from the body. Using joints that flexibly move in all different directions are not effective in fall detection.

– A fraction of the falling interval may not be detected due to the occluding object as illustrated in Fig. 2d. This is a serious problem when detecting a fall since the features being used are covered by the occluding object. We observe that the heights of most household furniture such as couch, chairs, and tables are typically less than an average height of a human. Therefore, using joints on the upper part of the body can significantly minimize the occlusion issues. We show through our evaluation how a fall can be detected even with a partially-occluding object in front of the subject. Specifically, our system is capable of detecting the fall even though the lower half of the body is completely hidden behind the object.

– When one of the joints is not monitored correctly, other joints help support the detection of the joint. The OpenNI software predicts where the incorrect joint might be using surround joints. For example, during a forward fall facing the camera, the head joint may occlude the torso joint. However, the head and the shoulder joints help by predicting where the head joint might be.

4.3. Body shape features and behavior sequence

Human behavior is defined as an observable activity or movement. In our study, a fall can be recognized by
observing the detailed movement of how the person is falling over a certain period of time. During the interval of falling down, there are two important elements that are required to formulate the falling behavior: features and behavior sequences. Features are prominent characteristics that describe the specific pose. A behavior sequence, on the other hand, is a sequence of poses that describes a certain activity. We believe that features reveal valuable characteristics that are used to generate the behavior sequence, which in turn defines complete behavior.

4.3.1. Body shape features

The features that we use are body shapes. A fall is not only characterized as a body moving downward, but also as a body proceeding in a consecutive order due to the gravity pushing the body down in high velocity. For example, when slipping, the person loses balance of the body and abruptly falls down. We categorize the body shapes that are created during the time the body hits the ground as features.

As shown in Fig. 1, we divide the falling interval into five sub-intervals characterized by different body shapes, which are described using \((x, y, z)\) coordinate data of the primary joints. For example, the first body shape, Shape 1, is defined in a position when a person is standing or walking. In this case, the heights of the primary joints would be at the highest positions and would move in parallel to the floor. The second body shape, Shape 2, is in the position when a person is tilting towards the ground. Depending on the type of fall, the primary joints would move around 30 degrees downward. The third body shape, Shape 3, is a person in a leaning position. This body shape can be detected by the joints midway from the ground. The near floor position is the fourth body shape, Shape 4, which is characterized by a person just before a fall with the primary joints approximately 20 degrees up from the floor. The last body shape, Shape 5 is in a fallen position, which is described by a person on the ground. All joints would be near the floor. These five body shapes are used for the first phase of our method later described in Section 5.

We specifically use five body shapes because having too many shapes would hinder the capability of clearly detecting all the shapes while having too little shapes would hinder the usage of the shapes for the second phase of our method to determine fall-like activities. For example, using less than five shapes would be difficult to characterize a fall-like activity such as picking something on the floor.

4.3.2. Behavior sequences

Unlike human eyes, a computing system cannot detect a certain activity merely with feature information. In other words, an actual fall cannot be detected simply with body shapes. The motion must be understood to recognize the behavior. Hence, we construct a set of behavior sequences utilizing a flag mechanism of detected body shapes. The summed-up flag data is passed onto the second phase to process the behavior of the user. Since the primary goal of our research is to detect the falling behavior, our system must first differentiate the falling behavior with other behavior not associated with falls. We design our system to understand three specific behavior sequences: an actual fall, a non-fall, and a normal daily activity. The actual fall is when a person unintentionally falls down and may require assistance to get up or receive further care. The non-fall behavior is a fall-like activity such as sitting, lying, or crouching to pick up an object. The normal daily activities include behavior such as walking, running, and doing chores. The body shapes constituting each of these behavior sequences will be described in detail in the next section to explain the detection process for the second phase of our method.

5. Two-phase support vector machine

We now detail how our system realizes real-time detection using shape information. A stream of skeletal joint data is categorized into three different classes of behavior (fall, non-fall, and normal behavior) by executing both body shape and behavior sequence classifications (at the 1st and 2nd phases, respectively) and
outputting the detection of the observed behavior. We found that a vast majority of falls ends on the ground or near the ground. Therefore, when the fifth body shape is detected from the first phase, the system flags an initialization of decision process until the second phase confirms the behavior. Although techniques such as Viterbi algorithm [14], which is a method for determining the hidden state sequence that best explains an observation sequence given the model, can be used for the second phase, we specifically choose to use our approach because it is a simple but effective way of utilizing the shape information.

5.1. Training and testing

For the first phase, FADES has five binary classifiers, each recognizing one of the five body shapes. Each of these classifiers recognizes a pre-specified body shape built from coordinate data of the primary joints. We start by collecting a number of data samples from 18 people to abstract body shape information during intervals of multiple falls. Each participant performs 5 different poses for each fall, where each pose represents the body shape of interest. Additionally, for each of the falls, we consider 16 orientations of the falling behavior mentioned in Section 4.2. This results in 1440 (18 people × 5 shapes × 16 orientations) body shape information. Although we run the training in a fixed location, the body shapes are not position-dependent because we can transform/normalized the coordinates of joints of a person (at a arbitrary position) to move them to the center of sight with respect to the Kinect.

For the second phase of FADES, the binary sequences of the body shape detection from the first phase is used to determine a fall, a fall-like activity, or a normal activity. The following describes the detection process:

- When one or two shapes are detected including Shape 5, the behavior will be determined as a normal daily activity.
- When three shapes are detected including the Shape 5, the behavior will be determined as a non-fall.
- When four or five shapes are detected including the Shape 5, the behavior will be determined as a fall only if the time from the second body shape to the last body shape is within a given time.
- When Shape 1 or Shape 2 are detected followed by Shape 3 or Shape 4, the sequence is filtered and resets the observation.

In any falling movement, there is always acceleration due to gravity. Therefore, we set a time of 700 ms to measure an actual fall. The behavior will otherwise be considered as a non-fall. Note that this interval is an adjustable parameter.

The latency between the state when the person falls down and when our system detects the fall is 45 ms, which shows that the delay is negligible. Kinect requires 30 ms processing time to send information to the computing unit and remaining time is required to process the body shapes by the first phase of our method. After detecting a fall, our system continues to observe if a fallen individual is moving in order to make sure if he or she is in an emergency situation for 5 seconds. If the fallen individual is able to get up and move after falling down, it is likely one can contact someone for help. However, if the person is seriously injured and remains motionless on the ground, the system classifies the fall as an emergency situation.

5.2. Support Vector Machine

The use of Support Vector Machine (SVM) [13, 38] reveals a large number of empirical successes among conventional classifiers such as a naïve Bayes classifier, a k-nearest neighbor classifier, and a decision tree [12]. In particular, we chose an SVM-based approach given that a study shows that the SVM outperforms other machine learning algorithms when classifying various data sets containing coordinate data [17].

We are given a labeled dataset \((t_1, c_1), \ldots, (t_N, c_N)\) to be used to train the SVM. The \(t_i \in R^n\), \(i = 1, \ldots, N\), is an \(n \times 1\) vector representing the training data fed to the SVM. In our case, \(t_i\) would symbolize the collection of \((x, y, z)\) coordinates of the joints. The \(c_i\) is a class label representing the five classes of body shapes. \(R\) represents the real number and \(n\) shows the number of dimensions of the input to the SVM.

In general, SVM seeks to define a decision surface which provides the largest margin separating among the data classes while at the same time minimizing the number of errors. The resulting model is nonlinear, and the training is performed by the use of kernel functions. The kernel function \(k\) indicates a measure of similarity between a pattern \(t\) to be tested, and a pattern \(\tilde{t}\) from the stored training set. For example, given two pattern vectors \(t, \tilde{t}\) of dimension \(n \times 1\), the kernel \(k\) can be represented as a canonical dot product:
Fig. 2. Joint data of behavior: (a) normal walking behavior, (b) non-fall going-to-sleep behavior, (c) actual fall behavior, and (d) actual fall with an occluding object.

\[ k(t \times \bar{t}) = \sum_{j=1}^{n} t_j \times \bar{t}_j \]  

(1)

where \( t_j \) and \( \bar{t}_j \) denote the \( j \)th elements of corresponding vectors. Note that the dot product representation of \( k \) allows geometrical interpretation of the vectors.

To apply our study using multiclass classification with \( m \) classes (\( m = 5 \)), the one-versus-all approach is used. The underlying basis of this approach is to reduce the multiclass problem to a set of binary problems, enabling the basic SVM to be utilized. Particularly for \( t_i \), there are \( m \) decision functions. The data \( t_i \) then belongs to the class for which the above decision function has the largest value. Overall, classification for one-versus-all case is done by a winner-take-all strategy, in which the classifier with the highest output function assigns the class.

6. Performance evaluation

6.1. Visualization of joint data

In order to evaluate the overall system performance, we start off by evaluating the pattern of the joints during exemplified cases of fall, non-fall, and normal behavior. A fall with an occluding object is also shown to demonstrate the strength of our approach (Fig. 2). Using only the \( y \) coordinate of the primary joints, we separately observe each behavior within a 10 s interval. Note that plotting each of the body shapes in Fig. 2 is done in a maximum period of 45 ms. Figure 3 demonstrates a visual system view of each of these examples in snap-shot images. Comparing the patterns with the images allows a better understanding of how our system detects each behavior utilizing the joint data and the body shapes.

A constant, motionless pattern from Fig. 2a shows normal walking behavior. Notice that in Fig. 3a, Shape 1 remains constant due to the upright position when the subject is walking. Figure 2b, on the other hand, reveals a gradual slope of a lying-down movement. Although Shape 5 is triggered, as shown in Fig. 3b, the system understands the behavior as a non-fall activity due to the gradual movement of lying down. In contrast to these examples, the narrow slope in Fig. 2c demonstrates abrupt falling behavior. The images shown in Fig. 3c are similar to those of Fig. 3b; however, due to the difference in velocity, the system understands the movement as actual falling behavior. Figure 2d shows another example of a fall but with a partially-occluding object covering the bottom half of the subject’s body. As shown in both of Figs 2d and 3d, our system can detect a fall even though some joints are covered by occluding objects.
6.2. Accuracy of the first phase

In formulating the sequence of falling behavior, the accuracy of detecting each body shape in the first phase of our method is very crucial. Therefore, to quantify the accuracy, each shape is inspected individually using the parameter for probability estimates in the LIBSVM library. In this experiment, 10 sets of 16 primary falls have been tested in a controlled setting. Falling was done on a 24 cm thick mat to allow realistic performance of the falls. We collected a total of 160 data samples with an average-sized male and a female in order to efficiently compute the accuracy of body shape detection.

For evaluation purposes, we place one Kinect on each side of the room to carefully extract data from every experimental falls. An important point to consider is that when observing a single type of fall, simultaneously gathering the data from all four directions is crucial. By constantly changing directions of the camera to measure fallen data, the accuracy may differ depending on how a person would fall. In doing so, we organize the data so that the classifiers are identical in all the Kinects used. The
trained data are combined together to build the training model.

In demonstrating the results, we divide the experiment into four different sections by the direction of the camera. Figure 4a shows the accuracy comparison of body shapes from forward falls facing the camera; Fig. 4b from back of the camera; Fig. 4c from right of the camera; and Fig. 4d from left of the camera. After computing the average accuracy value for each of the 10 testing sets, the results reveal that the accuracies (all higher than 96%) slightly vary for each of the body shapes. The variation is caused by the imperfect match of the training data and the testing data. However, the high accuracies prove that the first phase of our method is very efficient in triggering the second phase.

6.3. Overall performance evaluation

We evaluate the overall performance of our system, which covers the analysis of the second phase since the output of the second phase is the result of the detection. To do this we deployed multiple FADES in a real apartment setting, as shown in Fig. 5a. Figure 5b shows a 3D floor plan of where we placed FADES in the entire home. Note that we use a single Kinect to run each of the experiments. However, we evaluate our system in various different areas of the home to test the falls in the most realistic environment rather than a fixed setting.

The purpose of this analysis is to evaluate FADES in a natural setting, but we do identify the general script for a person to follow in order to perform the evaluation. Each of the experiments is performed by five people and we ask each individual to fall on a thin mat to allow realistic performance and to prevent getting hurt from the hard impact. However, the individual is not constrained to how many behavior to perform and how to perform the behavior. A human observer keeps records of the number of falls (or non-falls and normal behavior) and the detection result, and stops each experiment after 30 minutes.

We measure the accuracy of our system by separately testing four experiments: (1) the accuracy of detecting different types of falls, non-falls, and normal behavior, (2) the accuracy of detecting falls in different orientations of the person with respect to the camera,
(3) the accuracy of detecting falls in different locations of a room, and (4) the accuracy of detecting falls with an object occluding the person.

6.3.1. Accuracy of detecting different types of falls, non-falls, and normal behavior

For the first experiment, the individual performs various types of falls, non-falls, and normal activities. Specifically, the individual is asked to act out slipping, tripping, losing balance while standing, and losing balance while getting up from a chair as realistic as possible. Note that for this particular experiment, we also test how our system would perform when we use a Viterbi algorithm in the second phase rather than using our frequency-based approach. The Viterbi algorithm is a well-known dynamic programming technique for finding the optimal path. In the present context, the body shapes from the first phase are used to classify the most probable falling sequences responsible for the shape observations and outputting the detection of the observed behavior. We run this comparison to show that our approach is simple yet comparable to a state-of-the-art technique.

Moreover, correct classification of these falls as a fall, which we define as true positives rate (TPR), is of utmost importance to FADES since classifying a fall as an non-fall, which we identify as false negative rate (FNR), can be a significant fault that can block FADES from being used in practical home applications. By contrast, false positive rate (FPR), i.e., detecting a non-fall as a fall, can issue false alarms in a home. For this reason, we also keep track of movements like laying down and picking up something on the floor for the non-falls, and movements like walking for the normal behavior. From this, we measure the true negative rate (TNR), i.e., detecting a non-fall as a non-fall, to observe the performance of fall-like activity detection.

Figure 6 shows the results of using our approach while the results of using the Viterbi method in the second phase is shown in Fig. 7. Using our approach, the weighted average of the 5 participants recognizing slips, trips, and loss of balance while standing is 96.07% whereas the Viterbi algorithm achieves 97%. This shows that FADES possesses very similar performance to that of the state-of-the-art technique, yet simplifies the execution of system with less computation. Additionally, this also demonstrates that our framework of using body shape information in two phases can be applied to other applications and can simply be optimized by replacing the second phase with other state-of-the-art techniques.

Moreover, we observe our system has a very low FPR of 4.13%. Despite higher accuracy of detecting falls performed while standing, the performance of our system gets worse in terms of detecting falls performed while sitting. Our system is designed to detect falls that are performed from a standing position and therefore will need further improvement to detect falls even from sitting positions. Nevertheless, the flexible design allows our system to be adaptive to various learning data sets.

Continuing onto other behavior, Fig. 8 illustrates that the laying down behavior has an TNR of 92.6%. Some of the laying down activities observed from the participants were very similar to falls, where a person laid down quickly. However, FADES was able to differentiate most of them due to the difference in the speed during a real fall versus fast laying down behavior. When the gravity is pushing down the body, the
Fig. 6. Accuracy of detecting 4 types of falls, i.e., slip, trip, loss of balance while standing, and loss of balance while getting up from a chair, using our approach.

Fig. 7. Accuracy of detecting 4 types of falls, i.e., slip, trip, loss of balance while standing, and loss of balance while getting up from a chair, using a Viterbi Algorithm in the 2nd phase.

velocity of the movement is faster than that of a person quickly laying body down because the body is held by the individual’s intention to place their body down with careful impact. Unless a person forcefully pushes their body onto the ground, FADES can accurately differentiate the two behavior. Also shown in Fig. 8 is an increased accuracy on the behavior to pick up something from the floor. For this case, not all the joints on the upper-part of the body reflect what we have trained for Shape 5, which is required to trigger a detection of a fall. Therefore, this non-fall behavior was detected with better accuracy than that of the laying down behavior. Overall, results show that the TNR of non-falls is 95.6% with a FNR as low as 3.93%, and a 100% detection rate of the walking behavior.

6.3.2. Accuracy of detecting falls in different orientations

The second experiment examines the performance of detecting 16 different orientations of a fall. In Fig. 9, we see that falls that are performed towards the cam-

era has the highest accuracy and the falls that are performed most away from the camera has the lowest accuracy. We realize that this is due to the joints that might be occluding one another when falling in the farthest direction. However, utilizing the skeletal joints is an advantage for this kind of situation by having each of the joints being tracked to support the prediction of the missing joint in the skeleton outline. Although one joint might be occluding another, a fall can be effectively detected with the predicted joint. Our results show an average accuracy of detecting the falls most away from the camera is 92.8% and the overall orientations of the falls 96.36%.

Fig. 8. Accuracy of our approach to detect laying down and picking up something from the floor (non-falls) and walking (normal) performed by 5 people.

Fig. 9. Accuracy of detecting falls performed by 5 people with 16 different orientation of the body with respect to the Kinect, specifically: forward fall facing front of camera (FF), backward fall facing back of camera (BB), left fall facing left of camera (LL), forward fall facing back of camera (FB), right fall facing front of camera (RF), left fall facing front of camera (LF), right fall facing left of camera (RL), right fall facing right of camera (RB), left fall facing right of camera (LR), left fall facing back of camera (LB), right fall facing left of camera (BL), right fall facing back of camera (RB), left fall facing front of camera (LF).
6.3.3. Accuracy of detecting falls in different locations of a room

For the next experiment, we vary the relative distance between the Kinect and the performer in 9 locations, forming a $3 \times 3$ grid. We label the front row, from the left location as 1, 2, and 3; the next row as 4, 5, and 6, and the last row as 7, 8, and 9. Each individual performs a single type of fall with a single orientation of a fall for each of the locations.

The results shown in Fig. 10 reveal that locations 1 to 3 has the lowest accuracy compared to the rest of the locations. This is mainly due to the primarily four joints not being fully detected when a person falls from an area that is too close to the camera. Similarly to the issue with the orientations of the falls, even though the upper part of the body is not fully detected, other joints in the body are held onto these joints by the tracking skeleton. Hence, the accuracy does not drop significantly, having the weighted average accuracy of 90.07% and 93.11% for three locations and overall nine locations, respectively.

6.3.4. Accuracy of detecting falls in the presence of an occluding object

FADES is designed to best suit the detection of falls in a home environment, where furniture such as chair, table, and couch is usually shorter than the height of an adult. For this experiment we analyze the detection of falls with the participant partially covered by a small object, a medium object and a large object. For the small object, we use a $13.25 \times 12.5 \times 7.25$ inch paper shopping bag; for the medium object, we use a $21.8 \times 17.8 \times 39$ inch dining chair; and for the large object, we used a $84.2 \times 39.5 \times 36$ inch living room couch. Note that when analyzing falls with the occluding couch, we locate the subject towards the end of the couch where only the head joint is slightly visible to the camera when the subject has fallen.

Figure 11 shows that the accuracy of detecting falls with an occluding shopping bag and an occluding chair did not make much of a difference to the accuracy of detecting falls without any occluding object. This is because the upper part of the body is fully visible even though the bottom part of the body is covered by the object. However, the accuracy of detecting falls with a couch has dropped significantly with an average detection rate of 64.2%. Some of the falls were detected while some of the falls were not due to the skeleton being lost in times when the body got fully covered by the couch. We show in our next experiment how we can solve this type of occlusion problem dealing with large household objects.

6.4. Reliability of tracking skeletal data using multiple cameras in a room

We acknowledge the fact that an occlusion of the entire body will not be detectable using a single camera, which is an issue for most independent visual monitoring system. This hence requires the deployment of multiple cameras to expand the field of view and fully capture every activity of the user, as shown in our smart home design in Figs 5a and 5b. Important challenges in adopting a multi-camera setup is (1) to overcome occlusion problems dealing with large furniture in a room, (2) to verify that skeletal tracking is reliable, considering the range in which a Kinect can effectively track skeletal joints, (3) to ultimately support the reliability of fall detection in a practical deployment.

The minimum range of skeletal tracking is 1.2 meters and the maximum range is approximately 3.5 to 4.5 meters, depending on the software used. We run an experiment using 4 Kinects in a real-world, 4.05 by 3.15 meters dormitory bedroom over a span of 7 days.
To preserve privacy of the volunteer, we only collect data for 6 hours of each day.

We first design each of the four Kinects to capture a person’s body when a motion is detected in the form of RGB information, as well as in the form of joint information. This is done by a program that we implemented which looks at every pixel of the RGB and depth information, and calculates the difference between pixels from the previous frame and the current frame. If the difference projects over a threshold, the system considers the change as body detection. Since we are analyzing the functionality of skeletal tracking information, it is not necessary to collect lots of similar data. We therefore use a 4 minute maximum time window to limit the amount of data.

We then place the cameras in the four corners of the room to collect data throughout the week. Note that we are collecting data any time there is a motion inside the room. It can be immediately after someone enters the room or even if the person moves around after couple hours passed by since entering the room. Large furniture that may cause occlusion problems in the room includes a single sized bed, a desk including a chair, a tall lamp, a bedside drawers, and a large closet.

After collecting the data for each day, we manually count the number of body that were tracked from the RGB camera and the skeletal body that were tracked from the depth camera, and compare the two together.

The RGB count is used as the ground truth since we can visually see the correct number of body captured for each Kinects in the room and can be easily compared with the RGBD count.

Figure 12a shows the number of body detected in the form of RGB from each of the Kinects while Fig. 12b shows the number of body detected in the form of skeletal joints from each, as well as from the aggregate of the four Kinects. We often found that a single Kinect in the room captured a person’s body using the RGB camera but not skeletal body information. These types of problems exist mainly due to occluding objects partly covering the user’s body. For these cases, we observe the other three Kinects in the room during the same time when the skeletal body information was not captured. If any of these three Kinects detected skeletal joint information, the number of body tracked is counted towards the combined or overall count of skeletal body tracked from the depth camera as shown in the red line in Fig. 12b. We see from this that the overall count of the skeletal body is similar to that of the ground truth, meaning that at least one Kinect supports the tracking of the skeletal body although other cameras are blocked by one or more occlusions. Through this experiment, we prove that a multi-camera setup that covers the maximum range of approximately 4.5 meters of a room, provides a reliable skeleton tracking environment to detect falls.

6.5. Computation overhead

For the evaluation of computational overhead, we experiment by varying the number of training samples, and analyze how it affects the overhead and accuracy of our system. The total number of training samples is divided into four sets with: 360, 720, 1080, and 1440
Fig. 13. The number of training sample vs. (a) the number of SVs and (b) the accuracy of detecting falls.

samples. As shown in Fig. 13a, the total number of Support Vectors (SVs), which are the closest data separating the hyperplane, increases with more training samples. This proportionally affects the computational complexity of our system as the complexity of SVM is $O(n)$ where $n$ is the number of SVs. By contrast, naïve pattern matching techniques have a complexity of $O(n^2)$, which causes the computational overhead to increase rapidly. This shows that our system is computationally efficient and highly scalable, making it suitable for real-time processing.

To compare the number of training samples with the true positive detection accuracy of our system, we tested 100 falls for each of the dataset. As shown in Fig. 13b, the accuracy of detecting the falls gradually increased with more training samples. This means that removing training samples may take away possible body shapes that construct the behavior of a fall. This result clearly shows the effectiveness of our proposed approach to maximize the accuracy by considering various types of falls of all directions at the training stage, incurring minimal increase in the overhead of online computation.

In measuring the latency of the detection, we used the data captured to plot Fig. 3 described in Section 6.1. Every point is plotted with the maximum of 45 ms in the largest interval, which indicates that our approach verifies the shape of the user with a maximum of 45 ms latency.

7. Performance comparison

Several Kinect-based fall detection methods were proposed in the past. We evaluate and compare our system with five of recent studies. Table 1 shows a comparison chart that categorizes the feature(s) and data type that each of these works used. Additionally, we indicate on the chart if the work is able to detect various types of falls, distinguish different activities, and handle the occlusion issue, all of which are essential to the robustness of a Kinect-based fall detection system.

A Kinect-based fall detection system by Marzahl et al. [18] places a Kinect 30 cm above the floor and uses an image analysis technique in order to distinguish a fall and other events such as feet in front of the bed, leaving the room, and having an activity in the room. However, their system can only detect the fall under the bed and does not describe the falling behavior by the entire body, which may produce a high FPR. Additionally, this study does not deal with the capability to differentiate an actual fall versus a fall-like activity. In contrast, a study by Rougier et al. [29] combines a human centroid height relative to the ground with body velocity to detect a fall. A fall is detected when the velocity is above a certain threshold while the distance of the center of mass to the floor is below another threshold. Although claiming that their algorithm is capable of detecting a fall with occluding objects based on the velocity, they did not evaluate this capability with various experiments. In addition, they did not consider the capability to detect different types of falls of various directions from the camera.

Kepski et al. [15], on the other hand, utilized a fuzzy inference system using Kinect and a device consisting of an accelerometer and a gyroscope. As inputs their system takes the acceleration, the angular velocity, and the distance of the person’s center of gravity to the altitude at which the Kinect is placed. They stated that employing both the accelerometer and the Kinect can reliably distinguish falls from activities of daily living, and thus have the capability to reduce false alarms. They gave good reasoning why these two devices complement and support each other, and conducted experiments with various types of falls and different kinds of behavior. However, they neither presented quantitative evaluation results nor compared with similar techniques.

Another Kinect-based fall detection method was proposed by Planinc and Kampel [26]. They presented three different non-invasive technologies: the use of audio, 2D sensors, and 3D sensors, and introduced a
Table 1
Comparison chart of different works

<table>
<thead>
<tr>
<th>Work</th>
<th>Feature(s)</th>
<th>Data type</th>
<th>Detect different types of falls</th>
<th>Distinguish different activities</th>
<th>Consider occlusion handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marzahl et al. [18]</td>
<td>Distance between contour, orientation of contour line, and height and area of contour</td>
<td>3D images</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Rougier et al. [29]</td>
<td>Human centroid relative to the ground and velocity calculation</td>
<td>3D trajectory</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Kepski et al. [15]</td>
<td>Center of gravity</td>
<td>Depth images</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Planinc and Kampel [26]</td>
<td>Orientation of the person’s major axis and the height of the spine calculation</td>
<td>3D skeletal joint (image coordinates and world coordinates)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Mastorakis and Makris [19]</td>
<td>Velocity and inactivity calculation</td>
<td>3D bounding box</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Proposed system</td>
<td>Body shapes and behavior sequences</td>
<td>3D skeletal joints</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

A study by Mastorakis and Makris [19] measures the velocity based on the contraction or expansion of the width, the height, and the depth of the 3D bounding box. The velocity thresholds for the height and the width-depth composite vector of the bounding box, as well the duration of the fall, are estimated by performing a random search that optimizes the classification score in a training dataset. In contrast to other studies, they addressed the problems of detecting various types of falls, distinguishing different activities, and handling occluding objects. However, we implemented their algorithm and found that there are few potential weaknesses.

First, the major drawback is that bounding boxes are sensitive when tested in a realistic environment. Very frequently, random bounding boxes were created on occluding boxed objects. The paper backs this issue by stating that the bounding box of the subject can still be detected even though another bounding box is created on an object. However, this is only true if the subject does not touch the object. As shown in Fig. 14a, even with a slight touch, the system misunderstands the bounding box configuration and identifies the object as the subject or part of it. This not only makes it inconvenient for the user to go through reconfiguration, but also increases potential risks in events of fall. In situations when the subject forgets to go through reconfiguration and happens to fall down, fall detection will not be triggered as the main bounding box is on the object. Our system, by contrast, utilizes skeletal data to identify the subject’s body. As shown in Fig. 14b,
detecting the skeleton of the subject significantly helps reduce occlusion and configuration problems since the system is initializing the human, not a box. Additionally, we found that using simply the velocity of the fall is rather weak. Although it may require less computation, the velocity itself does not provide the behavior of a fall, and therefore makes the system less accurate. Moreover, velocities of the height, the width, and the depth may vary from person to person. New training would always be required to process the threshold for a particular subject. In contrast, we describe the human behavior by using shapes of the falling motion. Monitoring a falling motion helps the system to understand the behavior more clearly. For example, similarly to an observance from a human eye, the brain processes the activity information to recognize a behavior. Likewise, we have our system understand the behavior of the motion rather than only being aware of the velocity information itself.

Lastly, we evaluated the performance of the bounding box method with occluding objects. The analysis is done similarly to the evaluation done in Section 6.3.4 using 5 subjects in the same home setting. A human observer keeps records of the time, the number of falls, and the detection result. Each participant was told to fall in a fixed location and orientation of the fall with the participant partially covered by a shopping bag (small object), a chair (medium object) and a couch (large object). Figure 15 shows that the accuracy was successful when detecting falls with a small occluding object; however, with medium and large objects, the system showed problems, especially when the participant accidentally touched the object.

8. Conclusions

Falling is a leading cause of injury-related morbidity and mortality. As the growing population of elderly increasingly motivates new healthcare services at home, we applied a fall behavior monitoring surveillance technique to a home health care environment by using Kinect. The key novelties of our approach are: (1) the use of 3D skeletal joint data that supports precise detection of a falling behavior, (2) the concrete definition of a fall that holds true for all types of falls facing various directions from the camera, (3) and the two phase approach that accurately detects falls by classifying the body shapes and behavior sequences in a two-step process. By using our approach, we were able to detect an actual fall and distinguish it from a fall-like activity (e.g. lying down), while maintaining a high rate of accuracy (96.07%). Our experimental results demonstrated that the combination of skeletal data and our method significantly increases the robustness of fall detection and verifies a fall under the latency constraint of 45 ms.

Despite FADES showing high detection accuracy, we acknowledge the fact that there are many more issues to consider for a system to be perfectly robust and effective in a home environment. FADES will need further improvements to be more adaptive in a real home setting. Our future work is oriented towards developing the system to: (1) improve accuracy of detecting falls even from a sitting position and in the presence of large occluding objects, and (2) evolve with user feedback to learn new types of unseen falls, non-falls and normal behavior to reduce false alarms and improve its accuracy. We expect that our proposed approach can be extended for a generalized behavior video surveillance and envision that such a system can act as a catalyst to realize useful smart home applications utilizing skeletal joint from 3D depth camera.

Acknowledgement

This research was supported in part by the DGR IST R&D Program of MSIP of Korea (CPS Global Center), and in part by the IT R&D Program of MSIP/IITP (10041145, Self-Organized Software platform for Welfare Devices).

References