A Motion Classification Approach to Fall Detection

Shanfeng Hu[1,2], Worasak Rueangsirarak[3], Maxime Bouchée[4], Nauman Aslam[1], and Hubert P. H. Shum[1]

[1]Northumbria University, UK, Email: {shanfeng.hu, nauman.aslam, hubert.shum}@northumbria.ac.uk
[2]Beihang University, China, Email: hu_shan_feng@buaa.edu.cn
[3]Mae Fah Luang University, Thailand, Email: worasak.rue@mfu.ac.th
[4]Seatech University of Toulon, France, Email: maxime.bouchee.18@seatech.fr

***Abstract*—The population of older people in the world has grown rapidly in recent years. To alleviate the increasing burden on health systems, automated health monitoring of older people can be very economical for requesting urgent medical support when a harmful accident has been detected. One of the accidents that happens frequently to older people in a household environment is a fall, which can cause serious injuries if not handled immediately. In this paper, we propose a motion classification approach to fall detection, by integrating the techniques of motion capture and machine learning. The motion of a person is recorded with a set of inertial sensors, which provides a comprehensive and structural description of body movements, while being robust to variations in the working environment. We build a database comprising motions of both falls and normal activities. We experiment with several combinations of joint selection, feature extraction, and classification algorithms, showing that accurate fall detection can be achieved by our motion classification approach.**

***Keywords*—**fall detection, motion capture, motion analysis, motion classification, machine learning

I. Introduction

According to the 2017 Revision of World Population Prospects [1], there are approximately 962 million older people (aged 60 or above) in the world today, which makes up 13% of the entire population. Moreover, the population of older people is estimated to be growing at a rate of 3% per year, and the trend of population aging is irreversible in almost all parts of the world. One of the significant consequences of population aging is the rising burden on health systems. Aiming to alleviate this burden, automated in-home monitoring of older people is an economical way to dispatch medical care and support when a harmful accident has been detected.

In a home environment, one of the accidents that happens frequently to older people is a fall [2], which can cause serious injuries if the accident cannot be reported and handled immediately. Due to complex layouts of furniture and home appliances, a fall can happen to an old person in various situations. A natural requirement for a fall detector would be to reliably identify most kinds of real falls, while issuing as few false alarms as possible during normal activities. The challenge, however, is that in some situations, a normal activity such as bending down to pick up an item on the floor may be mistaken as a fall, since both activities resemble each other in terms of movement speed. Conversely, a fall may be misinterpreted as a normal activity such as sleeping on the floor, due to the similarity in body postures.

The above difficulties arise from the intrinsic complexity of human activity in a physical environment. As a result, the difference between falls and normal activities in terms of body posture may not be as prominent as the intra-variations of either one. To accurately distinguish falls from diverse kinds of normal activities, it is important to acquire the body movement data of a person in a comprehensive way, so that subsequent analysis can be applied to extract more informative features for more accurate classification.

In this paper, we represent the motion of a human body as a sequence of body postures, each of which is described by a set of body joints with measured angles and positions in the 3D space. These joints are organized as a hierarchy to capture full body posture information as shown in Figure 1, while respecting the physical rigidity constraints induced by human bone structure. A motion sensor is attached to each joint to measure its angle and position. Compared to traditional accelerometers mounted on parts of the body [3], [4], [5], these sensors provide more comprehensive and structural information about human activity. Meanwhile, they are more robust to working environment than color and depth cameras [6], [7], [8], [9], [10], as the latter are known to suffer from cluttered backgrounds, self occlusions, and environment occlusions.

Using these sensors, we build a database comprising motions of both fall and normal activities. A male adult subject is recruited to perform different types of motion including falling, walking, grabbing, sleeping, sitting on a chair, and sitting on the floor. After preprocessing the motion including registration, segmentation, alignment, and scaling, we obtain a database of 69 motion clips, each of which contains 380 frames of normalized body postures. With a frame rate of 60 frames per second, the duration of each motion clip is roughly 6.3 seconds. This database allows us to investigate a machine learning approach to fall detection.

Due to the high dimension nature of motion data, it would be impossible to hand-craft an ad-hoc method for separating falls from normal activities. Therefore, we regard fall detection as a motion classification task, which takes a motion clip as input and predicts a fall/not-fall binary label as output. If a trained classifier has high generalization performance, i.e., it performs well on input motions outside the training dataset, very accurate and reliable fall detection results can then be expected. We consider two factors that have a significant impact on generalization: one is the input features extracted from the original motion data, and the other is the choice of classification algorithms. We experiment with three joint selection strategies, two feature extraction methods, and four choices of kernels in the support vector machine (SVM) classifier, to demonstrate that very high fall detection performance can be achieved by our motion classification approach using the right combination.

The structure of this paper is as follows. Section II reviews literature closely related to fall detection. Section III elaborates on the construction of our fall motion database. Section IV presents our motion classification approach to fall detection. Section V shows the experimental results of fall detection. Section VI concludes this paper and discusses future research directions.

II. Related Work

In this section, we briefly discuss fall detection approaches in literature. Please refer to [11] for a recent survey of this field. Based on the kind of sensor used to capture body movements, existing fall detection approaches can be divided into the following two main categories.

Approaches in the first category mainly use acceleration sensors mounted on parts of the body of a person for fall detection. Mathie et al. [3] identified a fall event when a rapid increase of the negative acceleration measured by a waist-mounted accelerometer was detected. Bianchi et al. [4] enhanced a waist-mounted accelerometer with a barometric pressure sensor for altitude measurement, and trained a decision tree classifier for fall detection. Lai et al. [5] recognized injured body parts of a fallen older person using a combination of several tri-axial accelerometers. These sensors are cheap and easy to use, but have very limited capabilities to record fine-grained, full body movements, and thus can lead to unreliable fall detection results.

Approaches in the second category, compared to those in the first category, are less intrusive, as they mainly utilize color and depth cameras for measurement purposes. Thome et al. [6] introduced a layered hidden Markov model for posture modeling from multiple color cameras. Fall detection was formulated as posture classification in a fuzzy logic context. Rougier et al. [7] tracked human body silhouettes from a video sequence, and then recovered body shape deformations for fall detection using a Gaussian mixture model. Recently, depth cameras such as Microsoft's Kinect have been used for action recognition [8], because they provide direct measurement of geometric and structural information. Example works include [9] and [10], which recovered 3D body structures from depth images for fall detection. The drawback of color and depth cameras is that they both are sensitive to body occlusion, environment occlusion, and cluttered backgrounds, which present a significant challenge to a computer vision system for fall detection.

III. Fall Motion Database Construction

In this section, we elaborate on the construction of a database comprising motions of both falls and normal activities. This database allows us to investigate a motion classification approach to fall detection in the next section.

Motion Representation. In the left of Figure 1, we show that in motion analysis, a human body is abstracted as an articulated figure, which is composed of a set of rigid bones with specified lengths. To parameterize the rigid motion of

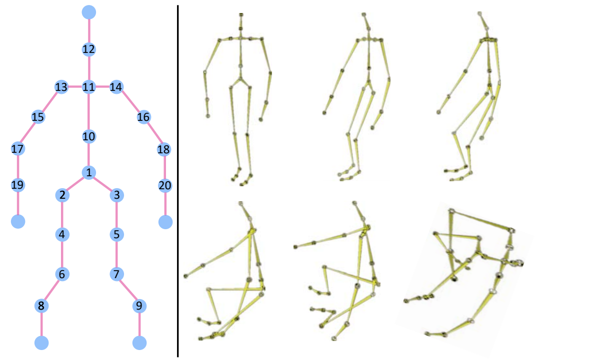

Fig. 1. Motion Representation. **Left**: A human body is abstracted as a tree hierarchy of 20 body joints (circled nodes). Each line segment connecting a pair of joints represents a rigid bone, whose motion is specified by the rotation of the child joint relative to the parent joint. 1: hip; 2: right hip; 3: left hip; 4: right knee; 5: left knee; 6: right ankle; 7: left ankle; 8: right toe; 9: left toe; 10: spine; 11: chest; 12: head; 13: right collar; 14: left collar; 15: right shoulder; 16: left shoulder; 17: right elbow; 18: left elbow; 19: right wrist; 20: left wrist; unnumbered nodes: joints not used in this work. **Right**: The falling motion of a person is represented as a sequence of body postures, which start from the standing posture (top left) and end with the lying posture (bottom right). Each posture is specified by a set of body joints with captured rotation angles and computed spatial positions.

Fig. 2. Motion Capture System. **Left**: The system consists of a set of inertial motion sensors (orange boxes attached to the body of the subject), a remote controller (black box with cables and an aerial), and a host computer (laptop). **Right**: Using the management software run on the computer, we can collect and visualize captured motion data in real-time, as the subject is wearing the sensors to perform an activity.

each bone in a physically valid way, these bones are defined and constrained by a tree hierarchy of 20 body joints, with the hip as the root node of the tree, the head, two toes and two wrists as the leaf nodes (i.e. end effectors), and the remaining joints as the internal nodes. The benefit of this hierarchy is that we can now parameterize the rotation of each joint only relative to its parent, which is analogous to *yaw*, *pitch*, and *roll* of an aircraft in its local frame.

In the right of Figure 1, we show that the motion of a person is represented as a sequence of body postures. Each of these postures is fully specified by the rotation angles of all the joints. To move a body in the 3D space, we also need to specify the position of the root joint, so that the positions of the remaining joints can be calculated easily from the rotation angles and bone lengths, by traversing the joint hierarchy from top to bottom. To restore translational invariance, we subtract the position of each joint from that of the root joint. Consequently, we have two basic kinds of posture representation: one is *joint angle representation*, and the other is *joint position representation*.

Motion Capture System. To obtain the motion data of a person performing an activity, we use the Xsens 3D motion capture system to gather the aforementioned two kinds of posture representations in a time-series way. In the left of Figure 2, we show a snapshot of this system, which comprises a suite of inertial motion sensors worn by a subject, a remote device for controlling these sensors, and a host computer for running a management software. In the right of Figure 2, we show that this software allows us to collect and visualize recorded motion data in real-time, as the person is wearing the motion sensor suite to perform an activity.

Experimental Protocol. We recruited a male adult subject to perform six kinds of activities: falling, walking, grabbing, sleeping, sitting on a chair, and sitting on a floor. The reason we chose these activities in our experiment is that they make up a large portion of the daily life of an older person in a household environment. Before each performance started, we used the motion capture management software to conduct posture calibration, so that subsequent collected motion data was reliable.

Motion Data Processing. After finishing the motion capture experiment, we executed the following steps to process each collected motion, i.e. a sequence of raw body postures, to build our fall motion database:

- First, we retargetted the sequence to a common skeleton template, so that person-specific variations, such as body height and weight, could be eliminated from the captured postures.
- Then, we segmented the sequence into shorter motion clips, each of which only contained postures from the beginning to the end of a single activity.
- Afterwards, we rescaled the length of each clip via linear interpolation, so that all clips in the database had the same number of postures.
- Finally, we annotated each clip with a semantic class label, which could be falling, walking, grabbing, sleeping, sitting on a chair, and sitting on the floor. The first class is considered to be a "fall" label, while the rest is considered to be the "not-fall" labels.

The resulting database comprises 69 well-processed motion clips of six activities. Each clip is a sequence of 380 body postures. With a frame rate of 60 frames per second, the duration of each clip is approximately 6.3 seconds.

IV. Fall Detection via Motion Classification

In this section, we pose fall detection as a motion classification task, which predicts a fall/not-fall binary label for an input sequence of body postures. Because the dimension of input motion data is very high, it would be impossible to hand-craft an ad-hoc method for this purpose. We need to leverage classification algorithms in machine learning that can robustly work in a high-dimensional feature space and have good generalization performance.

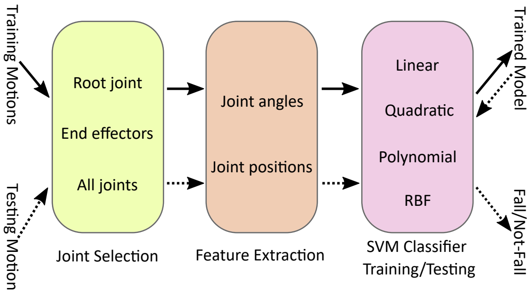

Fig. 3. Method Overview. The solid and dotted line arrows illustrate the steps of the fall motion classifier training and testing stage, respectively. We experiment with multiple options for joint selection and feature extraction. For the SVM classifier, we experiment with four kinds of kernels for identity feature mapping using the linear kernel, or non-linear feature mapping using the polynomial, radial basis function (RBF), and sigmoid kernels.

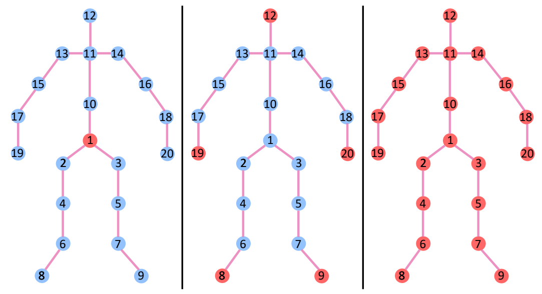

Fig. 4. Joint Selection. Red color is used to indicate the selection of joints. Only the selected joints will be used in the subsequent feature extraction step. **Left**: The root joint, i.e. the hip node, of the hierarchy defined in Figure 1, is selected. **Middle**: The end effectors, i.e. the head, two toes, and two wrists nodes of the hierarchy, are selected. **Right**: All the joints of the hierarchy are selected.

A. *Method Overview*

In Figure 3, we show the main computational steps of our motion classification approach to fall detection. Given an input sequence of body postures, we first select a subset of the body joints, and then extract a feature vector from the selected joints. In the training stage, we can learn a SVM classifier from the training motion data for fall detection. In the testing stage, the trained classifier is used to predict a fall/not-fall binary label for an input motion. For each step, we provide several options to explore the best combination of settings. In the following subsections, we elaborate on each step in more detail.

B. *Joint Selection*

As described in Figure 1, a human body is abstracted as a tree hierarchy of 20 joints in this work. The benefit of this abstraction is that we can represent the motion of a person in a fine-grained way. The downside, however, is that the resulting feature dimension of a whole motion sequence could be very high, because we need to consider all the joints of each posture in the sequence together. This would not only create a non-trivial computational burden, but also impact the generalization performance of a classifier to some extent.

We consider mitigating this problem by manually selecting a subset of key joints, which are expected to have significant correlations with the motion characteristics of a fall activity. As shown in Figure 4, we consider three joint selection strategies:

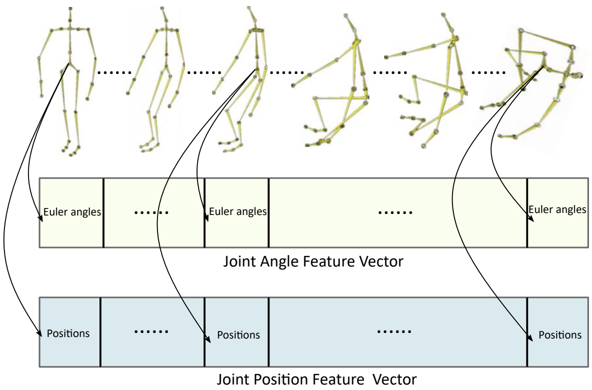

Fig. 5. Feature Extraction. **Top**: The motion of a person is represented as a sequence of body postures. Each posture is described by a set of body joints with measured rotation angles and spatial positions. **Middle**: The joint angle feature vector extracted from the motion contains a sequence of chunks (intervals with line boundaries), each of which is built by concatenating the rotation angles (Euler angle representation) of the selected joints of a posture in the motion sequence. Joint selection is conducted prior to feature extraction, as shown in Figure 3 and 4. **Bottom**: The joint position feature vector has the same layout as shown in the middle, except that we use the positions rather than the rotation angles of the selected joints.

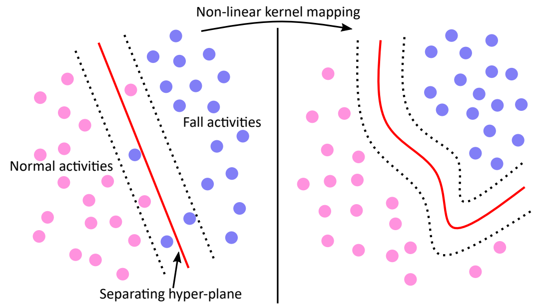

Fig. 6. SVM Classifier. **Left**: After feature extraction as shown in Figure 5, a motion is translated into a feature vector in a high-dimensional space. The feature vectors of motions of normal and fall activities are depicted as pink and blue filled circles, respectively. The goal of linear SVM classifier training is to find a hyper-plane (red straight line) that can separate normal and fall activities in the feature space, while having as wide a margin as possible (two black dotted lines parallel to the red line). In this case, one normal activity is wrongly identified as a fall, and two fall activities are wrongly classified as normal. **Right**: To enhance the linear separability of motion data, we can use a kernel function to perform non-linear feature mapping, so that the data can be better separated in a much higher, and possibly infinite-dimensional, feature space. The resulting hyper-plane is thus non-linear in the original space.

- The first strategy is the simplest, which only selects the root joint of the hierarchy. The reason that the hip node could be relevant to fall detection is that it is able to reflect the rapid drop of body centroid in a short time duration when a fall does happen.
- The second strategy only selects the end effectors of the hierarchy, i.e. the head, two toe, and two wrist nodes. Because the motion of the end effectors are fully driven by that of the root and intermediate joints, they may convey discriminative features of the whole motion that are relevant to fall detection.
- The third strategy is to trivially select all the joints of the hierarchy, which is intended to be a baseline for comparison with the former two strategies.

For each strategy, only the selected joints are included in the subsequent feature extraction step explained in the next section, where we evaluate the impact of joint selection on fall detection accuracy.

C. Feature Extraction

After joint selection, we proceed to the feature extraction step. The purpose of this step is to embed a motion sequence into a high-dimensional feature space, where fall detection is performed by differentiating feature vectors of normal and fall motion sequences. As shown in Figure 5, we consider two feature extraction methods in this work:

- The first method concatenates the rotation angles of the selected joints of each posture in the motion sequence into a feature vector. The rotation of a joint relative to its parent in the hierarchy is represented as a Euler angle triplet (ϕ, θ, ψ), where ϕ is the yaw angle, θ is the pitch angle, and ψ is the roll angle. Such local representation allows us to parameterize the motion of a human body in a rigid, physically valid way. Their measured values are directly available from the motion capture system.
- The second method generates a feature vector with the same layout as that generated by the first method. The only difference is that the content of the feature vector now is the positions of the selected joints. The position of a joint is represented as a coordinate triplet (x, y, z), which can be easily computed from the rotation angles of all the joints and the bone lengths, by specifying the position of the root joint. To restore translational invariance, we subtract the computed positions from the specified position.

In the next section, we evaluate the impact of feature extraction methods on fall detection accuracy.

D. Fall Detection using SVM

After joint selection and feature extraction, we obtain a set of M feature vectors $\{X_k\}_{k=1}^M$ and the corresponding fall/not-fall binary labels $\{y_k\}_{k=1}^M$, where $y_k \in \{-1, 1\}$. The task is to learn a classification function $f : X \rightarrow y$ from the given M pairs of training examples (X_k, y_k), such that it can be reliably generalized to predict a fall/not-fall label for a new, unseen feature vector with high accuracy.

A common practice in machine learning is to represent f with a set of free parameters Θ_f, and then search for an optimal Θ_f^* that approximately solves the following *structural risk minimization* problem:

$$\Theta_f^* = \arg\min_{\Theta_f} \frac{1}{M} \sum_{k=1}^{M} l(f(X_k), y_k) + \lambda R(\Theta_f) \qquad (1)$$

where the loss function $l(f(X_k), y_k)$ penalizes the discrepancy between the predicted label $f(X_k)$ and the true label y_k, and the regularization function $R(\Theta_f)$ penalizes the functional complexity of f, as a simpler function is expected to generalize better than a more complex alternative. λ is a tunable parameter for balancing the effect of the two functions.

		Linear	Polynomial	RBF	Sigmoid
Root Position	**Angles**	61.64%	63.61%	80.42%	73.71%
	Position	-	-	-	-
End Effectors	**Angles**	69.00%	73.19%	80.34%	74.59%
	Position	66.77%	71.33%	80.85%	76.11%
All Joints	**Angles**	70.61%	73.07%	80.33%	76.55%
	Position	70.48%	74.81%	80.33%	72.46%

TABLE I. FALL DETECTION ACCURACY.

In this work, we employ the popularly used SVM classification algorithm [12] for fall detection. SVM is known to work well in the high-dimensional input feature and small training dataset regime. This merit stems directly from the structural risk minimization problem formulation (1), with proper representation of the classification function f, the loss function l, and the regularization function R. As shown in the left of Figure 6, the basic linear SVM represents the classification function as a linear hyper-plane in the feature space, $f(X) = W^T X + b$, where T is the transpose operator, $\Theta_f = \{W, b\}$ are the parameters to optimize, and the sign of $f(X)$ is used to produce +1/-1 binary labels. One of the distinct features that endows SVM with high generalization ability is that it aims to solve for a hyper-plane as wide a margin as possible. This is achieved by using the hinge loss function, $l(f(X_k), y_k) = max(0, 1 - y_k(W^T X_k + b))$ where $max(\cdot, \cdot)$ is the element-wise maximum operator, and the L2-norm regularization function, $R(\Theta_f) = \frac{1}{2}\|W\|^2$ where $\|\cdot\|$ is the Euclidean norm of a vector.

The formulation of the linear SVM is simple and powerful, but would encounter the difficulty of clasifying normal and fall motions that are almost surely non-separable using a linear hyper-plane. As shown in the right of Figure 6, a solution would be to non-linearly transform an original feature vector into a much higher-dimensional space, where it is expected to stay on the correct side of a hyper-plane with higher probability. To implicitly apply this transformation, we only need to define a kernel function, $K(X_i, X_j)$, that measures the similarity of motion X_i and X_j. We consider four kinds of kernel functions in this work:

- Linear kernel: $K(X_i, X_j) = X_i^T X_j$, which corresponds to the basic linear SVM with identity feature mapping.
- Polynomial kernel: $K(X_i, X_j) = (X_i^T X_j)^3$, which implicitly raises a feature vector to the order of 3.
- RBF kernel: $K(X_i, X_j) = e^{-\|X_i - X_j\|^2}$, which implicitly raises a feature vector to an infinite order.
- Sigmoid: $K(X_i, X_j) = \frac{2}{1+e^{-2X_i^T X_j}} - 1$, which implicitly raises a feature vector to an infinite order.

It can be seen that a non-linear kernel function is a natural extension of the linear dot product for data similarity measurement. The extension allows us to easily perform non-linear classification in a new feature space.

V. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our motion classification approach to fall detection.

We use 5-folds cross validation to evaluate the system accuracy, which allows us to reliably assess the generalization

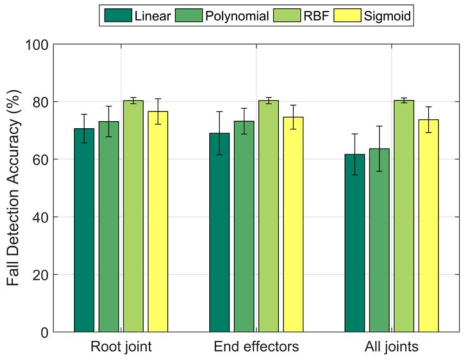

Fig. 7. Three joint selection strategies are compared in terms of their impact on fall detection accuracy (with standard deviations on top of each bar).

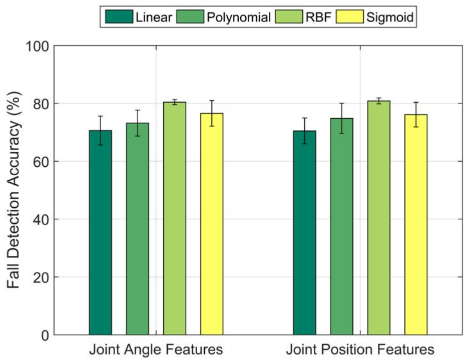

Fig. 8. Two feature extraction methods are compared in terms of their impacts on fall detection accuracy (with standard deviations on top of each bar).

ability of our approach. The fall motion database is randomly split into five subsets of equal size, with four of the five subsets for SVM classifier training and the remaining one for testing. This process is repeated 100 times to estimate the mean value and standard deviation of fall detection accuracy.

Table I shows the accuracy of different setup. The best accuracy is achieved using the 3D joint position of end effectors with the RBF kernal, in which we have 80.85% accuracy. Due to the small size of training data, using a subset of joints instead of all performs slightly better. Notice that since we represent joint positions relative to the root, we cannot extract root positions as a standalone feature. In the following, we evaluate the impact of joint selection, feature extraction, and SVM kernel function on fall detection accuracy.

Impact of Joint Selection. In Figure 7, we compare the three joint selection strategies, i.e. selecting root joint, end effectors, and all joints, in terms of their impact on fall detection accuracy. For each strategy, we show four performance bars corresponding to the four linear and non-linear SVM kernels, respectively. For each combination of strategy and kernel, we only report the accuracy results using the best performing feature extraction methods. It can be seen that the RBF kernel gives the consistently highest accuracy, with the lowest standard deviations, among the four kernels under each joint selection strategy. Also noticeable is that the three strategies give quite comparable results under each kernel, indicating the significance of the root joint and end effectors for fall detection.

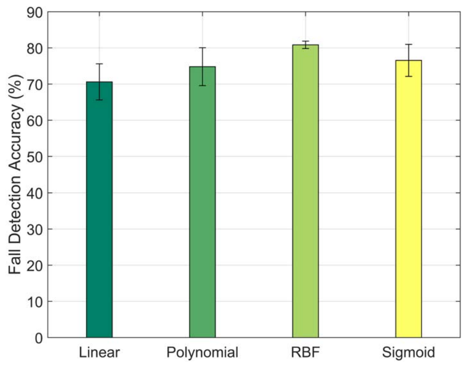

Fig. 9. Four SVM kernel functions are compared in terms of their impacts on fall detection accuracy (with standard deviations on top of each bar).

Impact of Feature Extraction. In Figure 8, we compare joint angle and joint position features in terms of their impact on fall detection accuracy. We report results using the best performing joint selection strategy for each combination of feature type and kernel. Similar to that in Figure 7, the RBF kernel consistently outperforms the others using either one of the two features. The results show that joint angle and joint position features are equally good for fall detection. This is foreseeable since we can convert the representation of the two features into each other without information loss. It appears that the SVM classifier we are using is oblivious to either type of feature representation.

Comparison of SVM Kernels. In Figure 9, we compare the four SVM kernel functions in terms of their impact on fall detection accuracy. For each kernel, we report the results using the best performing combination of joint selection strategy and feature extraction method. Consistent with that in Figure 8, we can see that the RBF kernel achieves the highest accuracy with the smallest standard deviation.

VI. Conclusion and Future Work

In this paper, we proposed a motion classification approach to fall detection. We represented the motion of a person as a sequence of body postures, each of which is described by a set of body joints with measured rotation angles and spatial positions. Under this representation, we built a database of motions of both normal and fall activities using a motion capture system. We experimented with several combinations of joint selection strategies, feature extraction methods, and SVM classifiers of different kernels for fall detection. The results show that we can achieve high detection accuracy.

Currently, we select body joints relevant to fall detection manually. This step can be automated using more involved selection strategies in the future. We also plan to explore feature selection methods that can reduce feature dimension while maintaining sufficient discriminative power. In particular, we are interested in validating if machine learned optimal features for action classification align with human perception.

As a future direction, we would like to explore the possibility of adapting the current framework into other motion capturing/sensing hardware such as the Microsoft Kinect, which does not require the user to wear any sensors and can facilitate more practical in-home usage. The human motion captured by Kinect is noisy, and therefore we will explore noise reduction techniques [13] for more accurate motion identification.

Finally, in our current database, the motion are segmented as a pre-process, and falling motion segments are identified. If the system is to be applied in a real-world scenario, continuous motion identification is necessary. We will research real-time motion segmentation techniques in the future.

Acknowledgment

This work was supported in part by the Erasmus Mundus Action 2 Programme (Ref: 2014-0861/001-001), the Engineering and Physical Sciences Research Council (EPSRC) (Ref: EP/M002632/1) and the Royal Society (Ref: IE160609).

References

[1] Population Division, Department of Economic and Social Affairs, United Nations, "World population prospects: The 2017 revision," *reference NO. ESA/P/WP/248*, 2017.

[2] World Health Organization, "Good health adds life to years: Global brief for world health day 2012," *reference NO. WHO/DCO/WHD/2012.2*, 2012.

[3] M. J. Mathie, A. C. Coster, N. H. Lovell, and B. G. Celler, "Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiological Measurement*, vol. 25, p. R1, 2004.

[4] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 6, pp. 619–627, 2010.

[5] C.-F. Lai, S.-Y. Chang, H.-C. Chao, and Y.-M. Huang, "Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 763–770, 2011.

[6] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A lhmm-based approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1522–1532, 2008.

[7] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 611–622, 2011.

[8] E. S. L. Ho, J. C. P. Chan, D. C. K. Chan, H. P. H. Shum, Y.-m. Cheung, and P. C. Yuen, "Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments," *Computer Vision and Image Understanding*, vol. 148, pp. 97–110, Jul 2016.

[9] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection from depth map video sequences," *Toward Useful Services for Elderly and People with Disabilities*, pp. 121–128, 2011.

[10] Z.-P. Bian, J. Hou, L.-P. Chau, and N. Magnenat-Thalmann, "Fall detection based on body part tracking using a depth camera," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 430–439, 2015.

[11] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[12] C. Cortes and V. Vapnik, "Support vector machine," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[13] Z. Liu, L. Zhou, H. Leung, and H. P. H. Shum, "Kinect posture reconstruction based on a local mixture of gaussian process models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 11, pp. 2437–2450, Nov 2016.