Fall Detection Based on Body Part Tracking Using a Depth Camera

Zhen-Peng Bian, *Student Member, IEEE,* Junhui Hou, *Student Member, IEEE,* Lap-Pui Chau, *Senior Member, IEEE,* and Nadia Magnenat-Thalmann

***Abstract*—The elderly population is increasing rapidly all over the world. One major risk for elderly people is the fall accidents, especially for those living alone. In this paper, we propose a robust fall detection approach by analyzing the tracked key joints of the human body using a single depth camera. Compared to the rivals that rely on the RGB inputs, the proposed scheme is independent of illumination of the lights and can work even in a dark room. In our scheme, a pose-invariant Randomized Decision Tree (RDT) algorithm is proposed for the key joint extraction, which requires low computational cost during the training and test. Then, the Support Vector Machine (SVM) classifier is employed to determine whether a fall motion occurs, whose input is the 3D trajectory of the head joint. The experimental results demonstrate that the proposed fall detection method is more accurate and robust compared with the state-of-the-art methods.**

***Index Terms*—Computer vision, 3D, monocular, video surveillance, fall detection, head tracking.**

I. Introduction

THE proportion of the elderly population is rising rapidly in most countries. In 2010, the elderly population (60+ years old) is 759 million (11 percent of the total population) all over the world [1]. Many studies have indicated that falls in elderly people are one of the most dangerous situations at home [2]. Approximately 28-35% of elderly people fall one time or more per year [3].

When an elderly person is living alone and has a fall accident, he/she may be lying on the floor for a long time without any help. This scenario mostly will lead to a serious negative outcome. Therefore, a fall accident detection system, which can automatically detect the fall accident and call for help, is very important for elderly people, especially for those living alone.

In [2], [4], [5], the authors reviewed principles and methods used in existing fall detection approaches. Nowadays, fall detection approaches could be classified as two main categories: non-vision-based method and vision-based method. Most methods of fall detection employ inertial sensors, such as accelerometers, since they are low costs. However, the methods based on inertial sensors are intrusive. As the vision technologies developed fast during the past few years, the vision-based methods, which are non-intrusive, have become a focal point in the research of the fall detection. They can capture the object's motion and analyse the object environment and their relationship, such as the human lying on the floor. The feature of the vision-based systems can be posture [6], [7], [8], [9], shape in-activity/change [10], [11], spatio-temporal [10], 3D head position [12], [13], and 3-D silhouette vertical distribution [14]. To improve the accuracy, some researchers combined non-vision-based method and vision-based method [15]. Most of the fall detection methods based on vision try to execute in real-time using standard computers and low cost cameras. The fall motion is very fast, taking few hundred milliseconds, and the image processing is high computational complexity. Thus, most vision-based existing methods cannot capture the specific motion during the fall phase. Recent researches on fall detection based on computer vision showed some practical frameworks. However, the robustness as well as accuracy of vision-based methods still leave a wide open room for further fall detection research and development.

The depth camera, such as Kinect [16], was used for fall detection [17], [18], [19], [20]. Thanks to the infra-red LED, the depth camera is independent of illumination of lights and can work well in weak light condition even in a dark room. It can also work well when the light condition significantly changes such as switching on or off the lights. As we know, some falls are caused by the weak light condition. In the depth image, each pixel value represents the depth information instead of the traditional color or intensity information. The depth value is the distance between the object and the camera. The depth information can be used to calibrate each pixel to a real world 3D location point. Compared with the traditional intensity or color camera, the depth camera provides several useful advantages in the object recognition. Depth cameras are useful for removing ambiguity in size scale. The object size in the color or intensity image is changed according to the distance between the object and the camera. That introduces ambiguity in size scale since the distance is unknown. In color or intensity images, the shadow greatly reduces the quality of background subtraction. Depth cameras can resolve silhouette ambiguity of the human body. The depth cameras simplify the tasks of background subtraction and floor detection. That can improve the robustness of the object recognition and can offer some useful information about the relationship between the human and the environment, such as the human hitting the floor. Furthermore, the realistic depth images of human can be much easier to synthesize. Therefore, a large and

Z.-P. Bian, Junhui Hou and L.-P. Chau are with the School of Electrical and Electronics Engineering, Nanyang Technological University, 639798 Singapore (email: zbian1@e.ntu.edu.sg, houj0001@e.ntu.edu.sg, elpchau@ntu.edu.sg).

N. Magnenat-Thalmann is with the Institute for Media Innovation, Nanyang Technological University, 639798, Singapore (email: nadiathalmann@ntu.edu.sg).

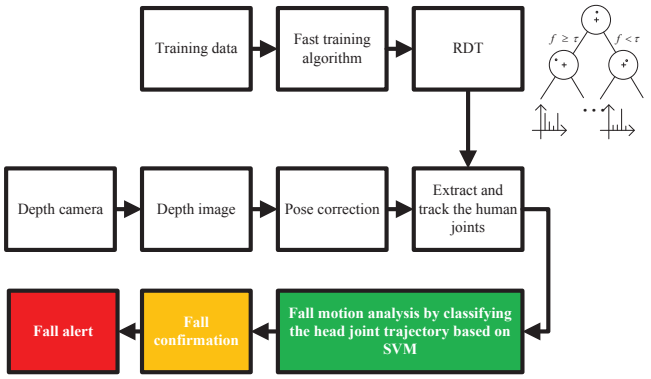


Fig. 1. Flow chart of proposed fall detection.

reliable training dataset can be built up for markerless motion capture [21].

In this paper, a robust fall detection system based on human body part tracking using a depth camera is proposed. To capture the fall motion, an improved randomized decision tree (RDT) algorithm is employed to extract the 3D body joints. By tracking the 3D joint trajectory, a support vector machine (SVM) classifier for fall detection is proposed. The 3D joint trajectory pattern is the input of the SVM classifier. Fig. 1 shows the flow chart of proposed fall detection algorithm.

The structure of this paper is as follows. Section II introduces the joint extraction method. Section III introduces the fall detection methods based on SVM. Experimental results are presented in Section IV. Conclusions are presented in Section V.

II. Joint Extraction

To capture the human fall motion, a markerless joint extraction is employed. The joint extraction is based on the proposed RDT algorithm, which is trained by large depth images dataset.

A. The feature for joint extraction

In [22], the recognized feature is based on the difference of intensities of two pixels taken in the neighbourhood of a key point. This feature was further developed in the depth image by Shotton et al. in [21]. They employed a simple depth comparison feature instead of the intensity comparison feature, resulting in a wonderful success.

Only one or two comparison features are very weak for discriminating objects, such as discriminating body parts. However, a RDT based on these comparison features are sufficient to discriminate objects. It can handle the noise of the depth image and can even work using the 2D silhouette [21]. The formulation of comparison feature in [21] can be described as:

$$f((x_0,y_0)|(\Delta x_1,\Delta y_1),(\Delta x_2,\Delta y_2)) = z((x_0,y_0)+\frac{(\Delta x_1,\Delta y_1)}{z(x_0,y_0)}) - z((x_0,y_0)+\frac{(\Delta x_2,\Delta y_2)}{z(x_0,y_0)}) \quad (1)$$

where (x_0,y_0) is the test pixel of the depth image, $(\Delta x,\Delta y)$ is the offset related to the test pixel (x_0,y_0), $(\Delta x_1,\Delta y_1)$ and $(\Delta x_2,\Delta y_2)$ are two different offset values, $z(x,y)$ is the depth value of the pixel (x,y). $1/z(x,y)$ is used to normalize the offset value, so that the feature can resolve the ambiguity in depth variation. As shown in Equation 1, this feature is three dimension translation invariant.

The depth information around the test pixel describes the geometric surface around this pixel. The geometric surface around the test pixel can be described well enough by the depth differences between the neighbour pixels and the test pixel. Thus, the following feature can be used to describe the same geometric surface as Equation 1 to recognize the test pixel:

$$f((x_0,y_0)|(\Delta x,\Delta y)) = z(x_0,y_0) - z((x_0,y_0)+\frac{(\Delta x,\Delta y)}{z(x_0,y_0)}) \quad (2)$$

Compared with Equation 1, this feature is with a higher computational efficiency: there are only one division, one addition and one subtraction; it only looks up two depth pixels. That can save time and leave more time for real-time fall detection.

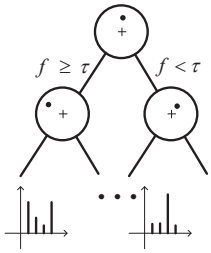


Fig. 2. Randomized decision tree.

Fig. 2 shows a randomized decision tree consisting of split and leaf nodes. A randomized decision forest includes several randomized decision trees. In many applications, the task of multi-class classifiers can be implemented in a high efficiency and high speed by RDT [21], [22]. In each tree, each split node n has a parameter $p_n = ((\Delta x_n,\Delta y_n),\tau_n)$. $(\Delta x_n,\Delta y_n)$ is the offset value. τ_n is a scalar threshold for comparing with the feature value of the test pixel. The evaluating function of comparison is

$$E((x_0,y_0);p_n) = B(f((x_0,y_0)|(\Delta x_n,\Delta y_n)) - \tau_n) \quad (3)$$

where $B(\cdot)$ is a binary function. When the value of $(f((x_0,y_0)|(\Delta x_n,\Delta y_n)) - \tau_n)$ is greater than or equal to 0, there will be $B(f((x_0,y_0)|(\Delta x_n,\Delta y_n)) - \tau_n) = 1$; otherwise $B(f((x_0,y_0)|(\Delta x_n,\Delta y_n)) - \tau_n) = 0$. When $E((x_0,y_0);p_n)$ is one, the test pixel is split to the left branch child of node n. When $E((x_0,y_0);p_n)$ is zero, the test pixel is split to the right branch child of node n. The operation is repeated, and stop when it meets the leaf node l. There are some classification information in the leaf node l, such as the probability of body parts. To classify a pixel (x_0,y_0), one starts at the root and repeatedly evaluates Equation 3, branching left or right according to the value of E until reaching the leaf node.

After classifying each pixel, the joint position can be predicted by body part classification [21]. In the body part classification method, the joint position is presented by the mean of the same class pixels. Since the head and hip, which

are used in the proposed fall detection, are the most visible body parts, we can use body part classification to extract them. Fig. 3 shows a classification result of the human body. Each color in Fig. 3 stands for the respective part of the human body with the highest probability.

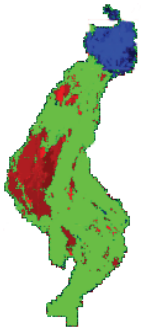


Fig. 3. The classification result of each depth pixel, each color stands for the respective part of the human body with the highest probability. Blue: head, red: hip, green: other part.

B. Aggregating predictions

The result of RDT algorithm is the classification of each pixel. Then the joint position is found by a mean shift method. The mean shift is based on a Gaussian kernel and a weighting factor. For the mean shift, the density estimator [21] of each body part c is defined as

$$J_c(L) \propto \sum_{i=1}^{N} w_{ic} \exp(-\| \frac{L - L_i}{b_c} \|^2) \tag{4}$$

where c is the body part label, L is the 3D location in the 3D real world, w_{ic} is a pixel weighting, N is the number of total test pixels in the test image I, L_i is the 3D location of the test pixel (x_i, y_i). b_c is a bandwidth for body part c. w_{ic} includes two factors: (1) the probability P from the RDT algorithm; (2) the world surface area related to the depth value z. The formulation of w_{ic} is

$$w_{ic} = P(c|(x_i, y_i), I) \cdot z(x_i, y_i)^2 \tag{5}$$

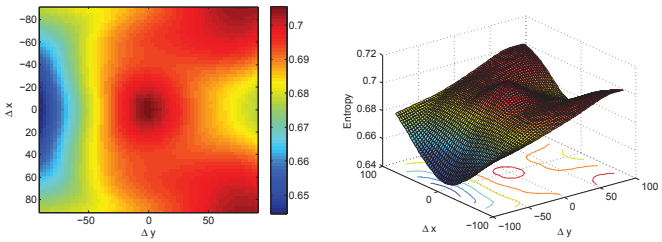
This mean shift method result is more accurate than the result of the global centre of the same body part since there are some outlying pixels as shown in Fig. 3.

C. Training

In order to obtain an optimised parameter p_n of each split node n of RDT, the computation complexity was very high in [21], [23]. Based on Equation 2 we propose a fast training algorithm which can reduce the number of candidate offsets significantly in the training phase.

The training pixels with labels for each synthesized depth image are randomly down sampled. The tree is trained using the smallest Shannon entropy to split each node.

At each node, a weak learner parameter $p((\Delta x, \Delta y), \tau)$ ($(\Delta x, \Delta y) \in (\Delta X, \Delta Y)$ is the pixel offset, $\tau \in T$ is the threshold value in Equation 3.) induces a partition of input



(a) (b)

Fig. 4. Entropy map and 3D view. (a) Map of entropy. (b) 3D view of the map.

example set $Q = \{I, (X, Y)\}$ into left and right subsets by Equation 3:

$$Q_L((\Delta x, \Delta y), \tau) = \{E((x_0, y_0); p((\Delta x, \Delta y), \tau)) = 1\} \tag{6}$$

$$Q_R((\Delta x, \Delta y), \tau) = \{E((x_0, y_0); p((\Delta x, \Delta y), \tau)) = 0\} \tag{7}$$

For each offset $(\Delta x, \Delta y)$, compute the τ giving the smallest Shannon entropy:

$$\tau^* = \arg\min_{\tau \in T} S(Q((\Delta x, \Delta y), \tau)) \tag{8}$$

$$S(Q((\Delta x, \Delta y), \tau)) = \sum_{sub \in L, R} \frac{|Q_{sub}((\Delta x, \Delta y), \tau)|}{Q} H(Q_{sub}((\Delta x, \Delta y), \tau)) \tag{9}$$

where $H(Q)$ is the Shannon entropy (computed on the probability of body part labels) of set Q. $S(Q)$ is the sum of Shannon entropy.

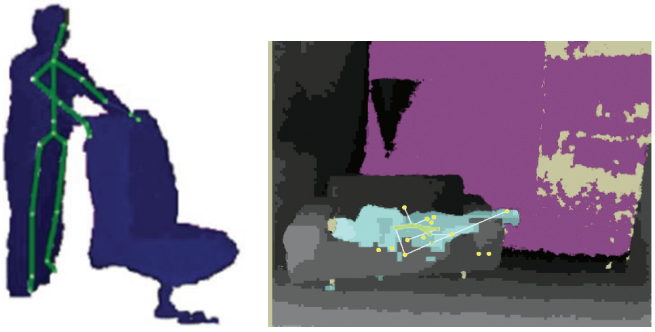
Fig. 4(a) is a map in a node by drawing Shannon entropy (given by $p((\Delta x, \Delta y), \tau^*)$) on the corresponding location $(\Delta x, \Delta y)$. Fig. 4(b) is a mesh view of Fig. 4(a). From Fig. 4(b), it can be noted that the Shannon entropy surface is smooth. The smallest Shannon entropy in this surface can be efficiently searched out by some search algorithms. Thus, just a few offsets need to be trained by Equations 6, 7 for each node. Equations 6, 7 should be tested by the whole set $Q = \{I, (X, Y)\}$ of input examples in a node, and this operation takes a long training time. Therefore, using Equation 2 and a suitable search algorithm, it can dramatically save training cost. In contrast, the feature in Equation 1 requires two offsets, which require randomly sampling 2000 candidate offset pairs among $(2M + 1)^4$ candidate offset pairs in [21], [23], where M is the range of $\Delta x, \Delta y$, i.e., $\Delta x, \Delta y \in [-M, M]$.

The parameter p with the smallest Shannon entropy in all candidate offsets and thresholds is

$$\begin{aligned} p((\Delta x^*, \Delta y^*), \tau^*) &= \arg\min_{(\Delta x, \Delta y) \in (\Delta X, \Delta Y), \tau \in T} S(Q((\Delta x, \Delta y), \tau)) \\ &= \arg\min_{(\Delta x, \Delta y) \in (\Delta X, \Delta Y)} \{\min_{\tau \in T} S(Q((\Delta x, \Delta y), \tau))\} \end{aligned} \tag{10}$$

If the depth of the tree is not too large, the training algorithm recurs for the left and right child nodes with example subsets $Q_L((\Delta x^*, \Delta y^*), \tau^*)$ and $Q_R((\Delta x^*, \Delta y^*), \tau^*))$, respectively, according to $p((\Delta x^*, \Delta y^*), \tau^*)$.

Table I demonstrates the performances of the algorithms based on [21] and ours. Based on Equation 2 and a search



(a) (b)

Fig. 5. Joint extraction results.

algorithm, it takes less than four hours to train one tree with twenty levels from ten thousand images on a standard PC using one core by Matlab. There are 1000 training example pixels in each image, 33 candidate thresholds per offset and 24 dynamic candidate offsets. It takes 310 hours with 2000 pairs of candidate offset based on Equation 1. 2000 images with ground truth of the head joint are used to test the performance of these two algorithms. The test is implemented on a standard PC using one core by C++. Based on Equation 2, the test time per frame is 2.8 ms. Based on Equation 1, the test time per frame is 5.0 ms, which is 79% more than 2.8 ms. The mean error is measured on the head joint. Their mean errors, i.e., 3.2 cm (Equation 1) and 3.1 cm (Equation 2), are very similar without substantial difference.

TABLE I
COMPARISON OF ALGORITHMS BASED ON [21] AND OURS.

	Training time per tree (hour)	Test time per frame (ms)	Mean error (cm)
[21]	310	5.0	3.2
Ours	**3.9**	**2.8**	**3.1**

D. Pose correction

In order to track the joint trajectory well, the frame rate of the camera output should be high and the joint extraction should be fast. The frame rate of Kinect is 30 frames per second and the joint extraction based on RDT takes a few milliseconds. Thus, the joint trajectories can be tracked well.

Based on RDT algorithm, an open license software has been released, i.e., Kinect for windows SDK [16]. Fig. 5(a) shows that the human joint extraction from the SDK is correct when the person is standing even moving a chair at the same time. However, the SDK joint extraction cannot work well under some scenarios, such as the person lying on a sofa as shown in Fig. 5(b). The degraded accuracy problem also appears in the moment of falling down due to the human body orientation changes dramatically while falling.

We propose to rotate the depth image in Fig. 5(b) by 90 degree clockwise, and the key joint positions can be extracted accurately as shown in Fig. 6. As inspired from this, the human torso is always rotated to be vertical in the image before the joint extraction process in the proposed fall detection system. Thanks to the high output frame rate of the depth camera, it is fast enough to track the human motion during falls. The torso orientation can be defined by the straight line through the hip centre and the head. The person's torso orientation changes very little per frame even during the fall. Thus, the torso orientation can be corrected well enough by rotating the torso based on the pose in the previous frame. Therefore, the joint extraction is pose-invariant.

Fig. 6. After rotation, the head and hip centre joints can be extracted correctly.

When the person walks into the view field at the first time, the head and hip centre are extracted. This information is used to rotate the torso orientation for the next frame, and the rotation angle is updated frame by frame. As shown in Fig. 7, when the person is falling down (Fig. 7(a)), the torso orientation can be always rotated to be vertical in the image (Fig. 7(b)) based on the previous frame.

(a)

(b)

Fig. 7. A simulated fall sequence. (a) Original poses. (b) After correction by proposed method.

The re-initialization is required when the torso orientation tracking is lost. As shown in Fig. 8, the input image after the subject segmentation was rotated by several angles, such as 0°, 120° and 240°, to generate several images. For each image, the head and the hip centre are extracted to obtain the torso orientation, and then the extracted torso orientation is corrected to be vertical. The extraction and correction are repeated several times. After the extractions and corrections, three candidate final rotation angles are obtained. It needs to select the best one from the three candidates. The selection metric is defined by the density estimator (Equation 4) during

the processing of joint extraction. For simplicity, the decision can be made by using the density estimator of the head portion.

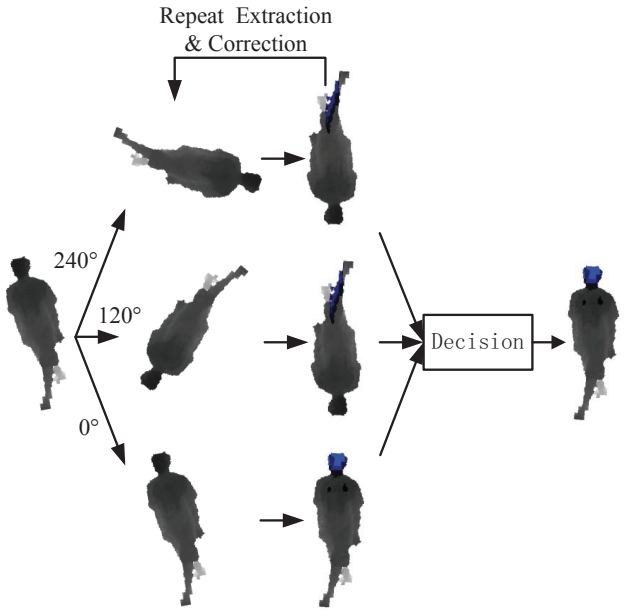


Fig. 8. An example of re-initialization.

To speed up the joint extraction, the rotation of the depth image can be taken place by rotating the offset parameter $(\Delta x_n, \Delta y_n)$ of each split node n of tree. This rotation of parameters can be done off-line after training. The rotation angles are fixed, such as $10°, 20°, ..., 350°$.

III. Fall detection based on SVM

This section describes the fall detection based on SVM, which employs the head joint distance trajectory as input feature vector.

A. Fall motion analysis based on SVM

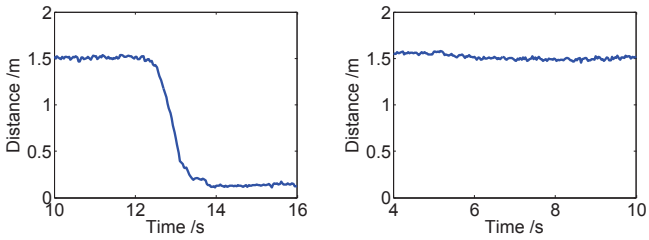
"Inadvertently coming to rest on the ground, floor or other lower level, excluding intentional change in position to rest in furniture, wall or other objects." is the definition of fall by World Health Organizations [24]. After the joint extraction, "coming to" "the ground" can be described in a technical word, i.e., some key joints coming to the ground. This feature can be used for fall detection.

Referring to the above feature, the fall is an activity related to the floor. The floor information should be measured. In [17], they assume that the floor occupies a sufficiently large part in the image. However, this assumption is not always true such as a small bedroom with a bed. In the proposed system, the floor plane is defined by choosing three points from the input depth image when the depth camera is set-up. The floor plane equation can be described as

$$Ax_w + By_w + Cz_w + D = 0 \tag{11}$$

$$A^2 + B^2 + C^2 = 1 \tag{12}$$

where A, B, C and D are coefficients, x_w, y_w and z_w are real world coordinate variables. The coefficients A, B, C and D will be determined after choosing three points.



(a) (b)

Fig. 9. Two patterns of head distance trajectory from a same video sequence. (a) Fall pattern. (b) Non-fall pattern.

The joint distance trajectory is defined as the trajectory of the distance between the joint and the floor. This signal includes the fall motion information, such as the acceleration, the joint velocity, the distance between the human joint and the floor and other hidden information. Fig. 9 shows a fall pattern and a non-fall pattern of the head distance trajectory. The fall motion can be classified by the joint distance trajectory pattern.

A d dimensional feature vector is formed by the distance trajectory in d consecutive frames. d should be large enough to cover all the phases of falling including rapid movement period during the fall, the period before the fall and the period after the fall. The fall detection can be seen as two classes classification problem. SVM is very suitable for two classes classification problem. It can automatically learn the maximum-margin hyperplane for classification. The feature vector as aforementioned described can be used as the input feature vector of SVM classifier.

B. Training dataset

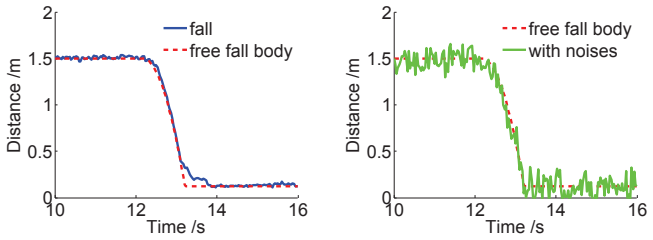
Since the 3D head joint trajectory has been tracking, the head joint motion can be analysed by the physics mechanics principle. During the falling phase, the joint motion can be seen as a free fall body. The free fall body is described as a simple formula

$$h(t) = h_0 + \frac{1}{2}a(t - t_0)^2 \tag{13}$$

where $h(t)$ is the height at the time t, h_0 is the height at the beginning of fall, a is the acceleration, t is the current time and t_0 is the beginning time. The free fall body can be used to simulate the joint fall motion to generate fall patterns by computer. Fig. 10(a) shows a free fall body trajectory fitting the head distance trajectory of a falling person. It can be noted that the free fall body curve fits well. The difference can be considered as Gaussian white noises. In order to improve the robustness, Gaussian white noises are added into the free fall body curve, as shown in Fig. 10(b). Some non-fall patterns can also be simulated by computer, as shown in Fig. 11. Based on the free fall body simulation, a large fall and non-fall patterns dataset can be built up.

C. Fall confirmation

In order to confirm the fall detection, the recover motion analysis after fall motion is required.



(a) (b)

Fig. 10. Free fall body trajectory fits the head distance trajectory. (a) Without noise. (b) Free fall body with noises.

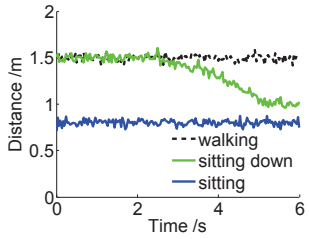


Fig. 11. Non-fall patterns simulated by computer.

There are two recover metrics: (a) the heights of hip and head are higher than a recover threshold value $T_{recover1}$ for a certain time; (b) the height of the head is higher than a high recover threshold value $T_{recover2}$ for a certain time. If one of these two metrics is satisfied, it means that the person is recovered. In our experiment, we set $T_{recover1} = 0.5m$ and $T_{recover2} = 0.8m$.

After the fall motion, if the person cannot recover within a certain time, fall detection will be confirmed. Without a fall confirmation, the fall alert will not be triggered. This stage can avoid the false alert.

IV. Experimental results

To test the proposed method, some normal activities (like crouching down, standing up, sitting down, walking) and falls, which are simulated by the human, have been tested.

A. Performance evaluation metric

The following parameters suggested by [2] are used to analyse the detection results of the proposed algorithm.

(1) True positives (TP): the number of fall events detected correctly.

(2) True negatives (TN): the number of non-fall events detected correctly.

(3) False positives (FP): the number of non-fall events detected as fall events.

(4) False negatives (FN): the number of fall events detected as non-fall events.

(5) Sensitivity (Se): the capacity to detect fall events

$$Se = \frac{TP}{TP + FN} \tag{14}$$

(6) Specificity (Sp): the capacity to detect non-fall events

$$Sp = \frac{TN}{TN + FP} \tag{15}$$

(7) Accuracy (Ac): the correct classification rate

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

(8) Error rate (Er): the incorrect classification rate

$$Er = \frac{FP + FN}{TP + TN + FP + FN} \tag{17}$$

A high sensitivity means that most falls are detected. A high specificity means that most non-falls are detected as non-fall. A good method for fall detection should have a high sensitivity and a high specificity. Besides, the accuracy should be high and the error rate should be low.

B. Dataset

The non-fall and fall activities were simulated as shown in Table II. They are suggested by [2], but with more detailed description. During an impactive fall, the elderly person cannot keep the transient pose kneeling or sitting on the floor. Therefore, the person will lie on the floor after fall as shown in the first row and fifth row of scenarios in Table II. In Table II, "Positive" and "Negative" means fall and non-fall, respectively. In total, there are 20 scenarios. 50% are positive and 50% are negative. Each scenario is simulated several times. Totally, there are 380 samples. There are four subjects, and their heights, ages and weights are: 159-182 cm, 24-31 years and 48-85 kg. Three are male and one is female. The experiments are in a real bedroom, as shown in Fig. 12. The camera is mounted 2.3m height on the wall.

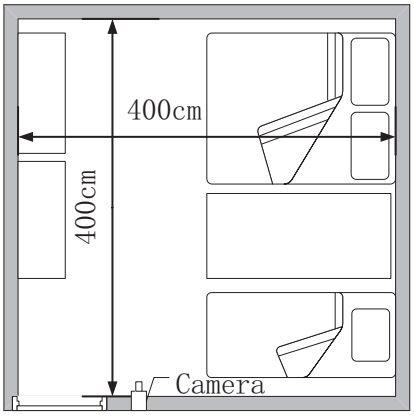


Fig. 12. Room plan for the evaluation.

The research was approved by the Institutional Review Board of Nanyang Technological University, Singapore.

C. Performance validation

To test the performance of SVM classifier method, a large training dataset, which includes about $100k$ non-fall and fall patterns of head distance trajectory, is generated by computer.

TABLE II
SCENARIOS FOR THE EVALUATION OF FALL DETECTION [2].

Category	Description	Outcome
Backward fall	On the hip, then lying	Positive
	Ending lying	Positive
	Ending in lateral position	Positive
	With recovery	Negative
Forward fall	On the knees, then lying	Positive
	With forward arm protection	Positive
	Ending lying	Positive
	With rotation, ending in the lateral right position	Positive
	With rotation, ending in the lateral to the left position	Positive
	With recovery	Negative
Lateral fall to the right	Ending lying	Positive
	With recovery	Negative
Lateral fall to the left	Ending lying	Positive
	With recovery	Negative
Syncope	Vertical slipping against a wall finishing in sitting position	Negative
Neutral	To sit down on a chair then to stand up	Negative
	To lie down on the bed then to rise up	Negative
	Walk a few meters	Negative
	To bend down, catch something on the floor, then to rise up	Negative
	To cough or sneeze	Negative

TABLE III
THE RESULTS OF FALL MOTION DETECTION BASED ON SVM.

TP	TN	FP	FN	Se(%)	Sp(%)	Ac(%)	Er(%)
182	190	0	8	95.8	100	97.9	2.1

After training, the SVM classifier is used to detect falls in the dataset of human simulated scenarios in Table II. The experimental results of fall motion analysis (detection) of SVM classifier is shown in Table III. For the fall motion detection, there is only eight error results, which are FN errors. The head distance trajectory during fall of an FN sample is shown in Fig. 13. There is an air mattress on the floor to protect the subject, as shown in Fig. 14. In this fall event, when the subject falls backward, the air mattress bounces the body of the subject quickly that the head distance trajectory is indicated by a circle in Fig. 13. The SVM classifier cannot have a correct decision on this head distance trajectory. If there is no air mattress, the dramatic rebound would not happen. In the fall confirmation stage, there is one fall event sample lost tracking the subject. In this sample, when the subject is lying on the floor, the system cannot segment the subject and recognises that there is no subject in the view field. The fall motion in this sample has been detected correctly by the SVM classifier. However, the system fails to confirm this fall event in the fall confirmation stage. It is due to failing in subject segmentation operated by Kinect SDK. In the fall confirmation stage, all the confirmation results are correct, except this lost tracking sample. Without a fall confirmation, the fall alert will not be triggered. Though the fall confirmation stage misses the lost tracking sample, this stage can avoid the false alert effectively. Thus, the system misses nine fall alerts, and there is no false alert.

A video demonstration of fall detection based on SVM is available in the website [1].

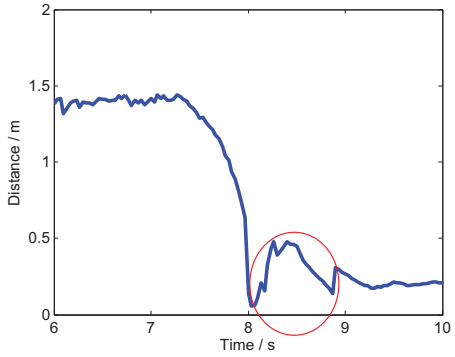


Fig. 13. The head distance trajectory during the fall of an FN sample of fall motion detection by SVM classifier. The circle indicates the rebound behaviour.

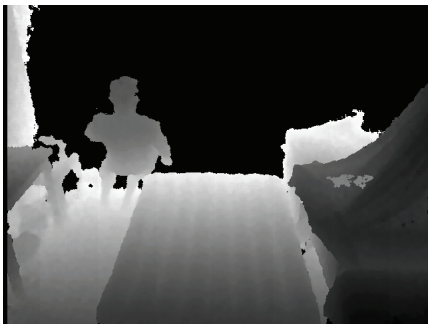


Fig. 14. There is an air mattress on the floor to protect the subject.

D. Comparison

To further evaluate the proposed algorithm, we compared it with two state-of-the-art approaches based on depth camera.

[1] http://www.ntu.edu.sg/home/elpchau/FallDetect2013.wmv

TABLE IV
COMPARISON OF DIFFERENT APPROACHES FOR FALL DETECTION.

	TP	TN	FP	FN	Se(%)	Sp(%)	Ac(%)	Er(%)
[17]	**188**	104	86	**2**	**98.9**	54.7	76.8	23.2
[18]	175	151	39	15	92.1	79.5	85.8	14.2
Proposed	181	**190**	**0**	9	95.3	**100**	**97.6**	**2.4**

The approach of [17] is based on human silhouette centre height relative to the floor. The human silhouette centre is obtained by the whole foreground (with morphological filtering). When the silhouette centre is lower than a threshold, a fall is detected. The experimental results are shown in row [17] of Table IV. This algorithm can detect the most of fall events, but with a lot of false positives (FP). It cannot distinguish the fall accident and the non-impact initiative activities well since it does not consider the motion together. When most part of the foreground object is near to the floor, including slowly lying down or sitting down on the floor or bad segmentation, it is detected as fall. The centre location is easily distorted by moving object and bad segmentation. To these events, the two key joints, head and hip, are still tracked well by the proposed method in our experiment. The proposed joint tracking method has a better robustness.

The approach of [18] makes use of the orientation of the body, which is based on the joints extracted from Kinect SDK, and the height information of the spine. If the orientation of the body is parallel to the floor and the spine distance to the floor is smaller than a threshold, a fall is detected. The experimental results are shown in row [18] of Table IV. The main disadvantage of this approach is the unreliable joints extraction. When the subject falls and is lying on the floor, the joint extraction is inaccurate and the orientation of the body obtained from inaccurate joints provides false information. Figure 15 shows two examples of the angle between the orientation extracted from Kinect SDK and the floor when the subject walked and fell down ending lying. From Figure 15, it can be noted that the orientations extracted from Kinect SDK were wrong when the subject fell and was lying on the floor. Based on un-robust joint extraction and predefined empirical thresholds, this method's capacity of prediction is limited.

Combined with the fall confirmation, the error rate of the proposed method is 2.4%. As shown in Table IV, the proposed approach outperforms the existing state-of-the-art approaches. Furthermore, compared with [18], which is required to extract at least six joints, only two joints are required in the proposed algorithm. Combined with the higher efficiency feature in Equation 2, the proposed algorithm has a lower computational complexity.

V. CONCLUSION

The proposed fall detection approach uses the infra-red based depth camera, so the approach can operate even in the dark condition. The depth camera can measure the human body motion and the relationship between the body and the environment. The floor plane can be extracted from the depth images. To capture the human motion, an enhanced RDT algorithm, which reduces the computational complexity based on the one offset feature, is employed to extract the human joints. Existing fall detection method based on joint extraction cannot extract human joints correctly when the subject lies down. The proposed rotation of the person torso orientation increases the accuracy of the joint extraction for fall detection. After extracting the joints, an SVM classifier is proposed to detect the fall based on the joint trajectory. The proposed approach is based on a single depth camera. Because the proposed motion analysis is based on the head tracking and the depth camera can be mounted close to the ceiling, it can avoid most occlusion situations. In the case of occlusion problem, multiple depth cameras based on the proposed approach can solve the problem. However, the proposed approach cannot detect the fall ending lying on furniture, for example, a wooden sofa, since the distance between the body and the floor is too high.

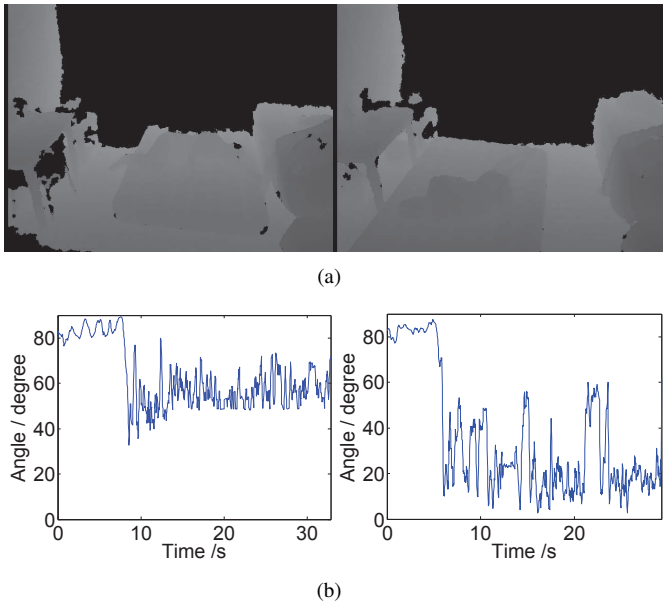


Fig. 15. Two examples of the angle between the orientation extracted from Kinect SDK and the floor when the subject walked and fell down ending lying. (a) Two depth image samples. (b) Angle waveforms corresponded to the two sequences of (a).

The proposed RDT training algorithm based on the one offset feature reduces the number of candidate offsets significantly from 2000 to 24. Therefore, the computational complexity of building RDT for fall detection can be reduced by 83 times under the same training condition.

Based on depth image sequence, by extracting and tracking the joints of the human body as well as investigating the joints' behaviour after fall, the proposed approach can detect and confirm the human fall accurately. Experimental results show that the accuracy of the proposed algorithm is improved by 11.8% compared with the most recent state-of-the-art fall detection algorithm.

ACKNOWLEDGMENT

The authors would like to acknowledge the Ph.D. grant from the Institute for Media Innovation, Nanyang Technological

University, Singapore.

REFERENCES

[1] United-Nations, "World population prospects: The 2008 revision," 2008.

[2] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy, "Fall detection-principles and methods," *29th IEEE International Conferenceon Engineering in Medicineand Biology Society (EMBS)*, pp. 1663–1666, 2007.

[3] W. H. O. (WHO), "Good health adds life to years," *Global brief for World Health Day 2012*.

[4] X. Yu, "Approaches and principles of fall detectionfor elderly and patient," *10th IEEE International Conferenceon e-Health Networking,Applications and Services (HealthCom)*, pp. 42–47, 2008.

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

[6] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A LHMM-based approach," *IEEE Trans. Circuits Syst. Video Technol.(TCSVT)*, vol. 18, no. 11, pp. 1522–1532, 2008.

[7] M. Yu, A. Rhuma, S. Naqvi, L. Wang, and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1274–1286, 2012.

[8] D. Brulin, Y. Benezeth, and E. Courtial, "Posture recognition based on fuzzy logic for home monitoring of the elderly," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 5, pp. 974–982, 2012.

[9] M. Yu, Y. Yu, A. Rhuma, S. Naqvi, L. Wang, and J. Chambers, "An online one class support vector machine based person-specific fall detection system for monitoring an elderly individual in a room environment," *IEEE Journal of Biomedical and Health Informatics*, 2013.

[10] H. Foroughi, B. S. Aski, and H. Pourreza, "Intelligent video surveillance for monitoring fall detection of elderly in home environments," in *11th IEEE International Conference on Computer and Information Technology (ICCIT)*, 2008, pp. 219–224.

[11] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Trans. Circuits Syst. Video Technol.(TCSVT)*, vol. 21, pp. 611–622, 2011.

[12] C. Rougier and J. Meunie, "Fall detection using 3D head trajectory extracted from a single camera video sequence," in *First International Work-shop on Video Processing for Security(VP4S)*, 2006.

[13] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "3D head tracking for fall detection using a single calibrated camera," *Image Vision Comput.*, vol. 31, no. 3, pp. 246–254, Mar. 2013.

[14] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 290–300, 2011.

[15] C. Doukas and I. Maglogiannis, "Emergency fall incidents detection in assisted living environments utilizing motion, sound, and visual perceptual components," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 277–289, 2011.

[16] Microsoft, "http://www.microsoft.com/en-us/kinectforwindows/."

[17] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection from depth map video sequences," in *Proceedings of the 9th international conference on Toward useful services for elderly and people with disabilities: smart homes and health telematics*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 121–128.

[18] R. Planinc and M. Kampel, "Introducing the use of depth data for fall detection," *Personal Ubiquitous Computing*, 2012.

[19] Z. P. Bian, L. P. Chau, and N. Magnenat-Thalmann, "A depth video approach for fall detection based on human joints height and falling velocity," in *International Conference on Computer Animation and Social Agents*, May 2012.

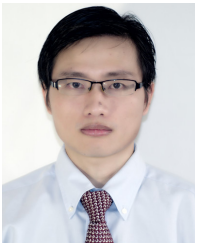
[20] Z.-P. Bian, L.-P. Chau, and N. Magnenat-Thalmann, "Fall detection based on skeleton extraction," in *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. New York, NY, USA: ACM, 2012, pp. 91–94.

[21] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, june 2011, pp. 1297 –1304.

[22] P. Lagger and P. Fua, "Randomized trees for real-time keypoint recognition," in *in Proc. CVPR*, 2005.

[23] K. Buys, C. Cagniart, A. Baksheev, T. D. Laet, J. D. Schutter, and C. Pantofaru, "An adaptable system for RGB-D based human body detection and pose estimation," *accepted for publication in the Journal of Visual Communication and Image Representation*.

[24] W. H. O., "WHO global report on falls prevention in older age," *World Health Organization (WHO) Library Cataloguing-in-Publication Data*, 2007.



Zhen-Peng Bian received the B. Eng degree in Microelectronics from South China University of Technology, Guangzhou, China in 2007. He is currently pursuing the Ph.D degree from the School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore.

His current research interests include fall detection, motion capture, human computer interaction and image processing.

Junhui Hou received the B. Eng degree in Information Engineering (Talented Students Program) from South China University of Technology, Guangzhou, China and the M. Eng in Signal and Information Processing from Northwestern Polytechnical University, Xi'an, China in 2009 and 2012, respectively. He is currently pursuing the Ph.D degree from the School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore.

His current research interests include video compression, image processing and computer graphics processing.

Lap-Pui Chau received the B. Eng degree with first class honours in Electronic Engineering from Oxford Brookes University, England, and the Ph.D. degree in Electronic Engineering from Hong Kong Polytechnic University, Hong Kong, in 1992 and 1997, respectively. In June 1996, he joined Tritech Microelectronics as a senior engineer. Since March 1997, he joined Centre for Signal Processing, a national research centre in Nanyang Technological University as a research fellow, subsequently he joined School of Electrical & Electronic Engineering, Nanyang Technological University as an assistant professor and currently, he is an associate professor. His research interests include fast signal processing algorithms, scalable video and video transcoding , robust video transmission, image representation for 3D content delivery, and image based human skeleton extraction.

He involved in organization committee of international conferences including the IEEE International Conference on Image Processing (ICIP 2010, ICIP 2004), and IEEE International Conference on Multimedia & Expo (ICME 2010). He is a Technical Program Co-Chairs for Visual Communications and Image Processing (VCIP 2013) and 2010 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS 2010).

He was the chair of Technical Committee on Circuits & Systems for Communications (TC-CASC) of IEEE Circuits and Systems Society from 2010 to 2012, and the chairman of IEEE Singapore Circuits and Systems Chapter from 2009 to 2010. He served as an associate editor for IEEE Transactions on Multimedia, IEEE Signal Processing Letters, and is currently serving as an associate editor for IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Broadcasting and IEEE Circuits and Systems Society Newsletter. Besides, he is IEEE Distinguished Lecturer for 2009-2013, and a steering committee member of IEEE Transactions for Mobile Computing from 2011-2013.

Nadia Magnenat-Thalmann has pioneered various aspects of research of virtual humans over the last 30 years. She obtained several Bachelor's and Master's degrees in various disciplines (Psychology, Biology and Biochemistry) and a PhD in Quantum Physics from the University of Geneva in 1977. From 1977 to 1989, she was a Professor at the University of Montreal in Canada. In 1989, she moved to the University of Geneva where she founded the interdisciplinary research group MIRALab.

She is Editor-in-Chief of The Visual Computer Journal published by Springer Verlag, and editors of several other journals. During her Career, she has received more than 30 Awards. Among the recent ones, two Doctor Honoris Causa (Leibniz University of Hanover in Germany and University of Ottawa in Canada), the Distinguished Career Award from the Eurographics in Norrkoping, Sweden, and a Career Achievement Award from the Canadian Human Computer Communications Society in Toronto. Very recently, she received the prestigious Humboldt Research Award in Germany. Besides directing her research group MIRALab in Switzerland, she is presently visiting Professor and Director of the Institute for Media Innovation (IMI) at Nanyang Technological University, Singapore. For more information, please visit http://imi.ntu.edu.sg/AboutIMI/DirectorOfIMI/Pages/CurriculumVitae.aspx