

Article

A New Approach to Fall Detection Based on the Human Torso Motion Model

Leiyue Yao, Weidong Min * and Keqiang Lu

School of Information Engineering, Nanchang University, Nanchang 330031, China; leiyue_yao@163.com (L.Y.); lukeqiang1111@163.com (K.L.)

* Correspondence: minweidong@ncu.edu.cn; Tel.: +86-0791-8396-9684

Received: 26 July 2017; Accepted: 18 September 2017; Published: 26 September 2017

Featured Application: In our proposed method, a new feature named torso angle is firstly proposed and imported in the human torso motion model (HTMM) for fall detection. By tracking the changing rate of torso angle and centroid height, our fall detection model has a strong capability of differentiating falls from other fall-like activities, such as controlled lying down, quickly crouching and bending, which may be big obstacles for other existing computer vision-based approaches to discriminate.

Abstract: This paper presents a new approach for fall detection based on two features and their motion characteristics extracted from the human torso. The 3D positions of the hip center joint and the shoulder center joint in depth images are used to build a fall detection model named the human torso motion model (HTMM). Person's torso angle and centroid height are imported as key features in HTMM. Once a person comes into the scene, the positions of these two joints are fetched to calculate the person's torso angle. Whenever the angle is larger than a given threshold, the changing rates of the torso angle and the centroid height are recorded frame by frame in a given period of time. A fall can be identified when the above two changing rates reach the thresholds. By using the new feature, falls can be accurately and effectively distinguished from other fall-like activities, which are very difficult for other computer vision-based approaches to differentiate. Experiment results show that our approach achieved a detection accuracy of 97.5%, 98% true positive rate (TPR) and 97% true negative rate (TNR). Furthermore, the approach is time efficient and robust because of only calculating the changing rate of gravity and centroid height.

Keywords: fall detection; human torso motion model; torso angle; centroid height; computer vision; depth image; Kinect

1. Introduction

Falls are one of the major health hazards among the aging population aged over 60. According to the report of the World Health Organization, approximately 28–35% of people aged 65 and over fall each year and 32–42% for those over 70 years of age. In fact, falls exponentially increase due to age-related biological changes, which lead to a high incidence of falls and fall-related injuries in ageing societies [1]. Nowadays, falls are not only life threatening, but also one of the major issues in elderly health. A fall can cause severe consequences, including broken bones, superficial cuts and abrasions to the skin soft tissue [2,3]. If a falling person cannot get help in a short period of time, the situation will be even worse. As many people in this ageing group live alone, it is difficult for them to seek help immediately. For these and many other reasons, the number of research works on fall detection has increased dramatically over recent years. Automatic detection of human falls provides help to reduce the time between the fall and the arrival of medical attention [4,5].

During recent years, different kinds of approaches have been proposed to realize automatic fall detection. Mubashir et al. [6] classified the fall detection methods into three major categories: the wearable device-based method, the ambient sensor-based method and the vision-based method.

Wearable sensors detect the motion and location of the human body. M.J. Mathie et al. [7,8] adopted accelerations as the major feature to identify falls. Sixsmith A. [9] judged falls by checking periods of inactivity. M. Kangas et al. [10] identified a fall by analyzing the posture of the person. Although these approaches did work in detecting falls, the disadvantages are also obvious. The wearable sensors can be worn or damaged due to external factors, and they will not detect falls if the person forgets to wear them. Besides, wearable devices may make people feel uncomfortable.

Another type of approach uses sensors that are deployed in the environment to detect falls. Zhuang, Huang et al. [11] checked whether there was a fall by collecting sounds, while M. Alwan et al. [12] detected falls by monitoring vibrations in a room. In [13], the authors focus on detecting falls in the bathroom by using smart floors to collect floor pressure information. Although the ambient sensor-based method is a creative way for fall detection, the accuracy and false alarm rate are unacceptable in most situations. Moreover, this kind of operation is limited to those places where the sensors have been deployed.

Due to these main drawbacks of the above two approaches, the computer vision-based method has become a hot topic in recent research. The method needs no wearable devices and aims at detecting falls by analyzing video streams from one or more cameras. Compared with other approaches, this kind of method has more advantages. However, privacy is a problem for the people who live under the supervision. Q.M. Rajpoot et al. [14] provided solutions for privacy identification and protection. In addition, Chaaoui, A.A. et al. [15] defined five specific visualization levels corresponding to five visualization models, respectively. The appearance of persons can be protected at each level to a different degree. Our proposed method is based on computer vision, and the usage of only depth images helps maintain privacy. Compared with the method proposed in [15], the usage of depth images reaches the third visualization level. More detailed information about the typical methods based on computer vision technology is discussed in Section 2.

The remaining part of this paper is organized as follows. Section 2 gives an overview of the current state-of-the-art with regard to features and methods used in the vision-based fall detection method. Section 3 describes our model and algorithm of fall detection. Section 4 shows the experiment and gives the results of our approach. Section 5 lists three major limitations of our model. Finally, we present our concluding remarks in Section 5.

2. Related Work

With the development of machine learning, computer vision and image processing techniques, the video camera or computer vision-based method has become a new, but hot topic of fall detection. Compared with the wearable device-based approaches, the camera or computer vision-based methods are less intrusive, more accurate and robust. Moreover, the detection systems can be easily updated by downloading the newest programs.

One typical fall detection method identified and located the person in the video by using a vertical projection histogram with a statistical scheme. Then, the ratio of the width and height of the bounding box and the absolute difference between these two values were used to detect falls [16]. In [17], the authors divided the image into many rectangular areas, then falls can be detected by calculating the area of the body. Kwolek et al. [18] presented a fuzzy system that used the accelerometer and the Kinect sensor to detect the fall event. However, since the human external rectangle is difficult to detect accurately, false alarms and missed detections are the major problems of this method.

Video sequence analysis is another major branch of vision-based fall detection. Khan and Habib [19] used motion history images and detected large motion in a video sequence. The combination of the aspect ratio of the bounding box and the changing rate of width and height were used in their approach to detect falls. In [20], overlapping smart cameras were used for fall detection and localization. In [21], a single camera covering the full view of the room environment is used for

the video recording of an elderly person's daily activities for a certain time period. Fall detection is realized by analyzing the postures that are extracted from video clips. Rougier et al. [22] proposed a new way to track 3D head trajectory by only one calibrated camera. The vertical velocity of the 3D head is used to judge whether a fall happens. Olivieri et al. [23] used optical flow to detect falls and recognize other human activities.

Recently, depth images have been widely used for action recognition and classification, because of the particular advantages in privacy protection and the availability of being easily collected by Kinect or other sensors. Lei Yang et al. [24] proposed a fall detection method based on spatio-temporal context tracking over three-dimensional (3D) depth images captured by the Kinect sensor. Rougier, C. et al. [25] used several cameras to create 3D vision. A measure of the vertical distribution along the vertical axis is calculated, and a fall event is detected when this distribution is abnormally near the ground for a certain length of time. Gasparrini et al. [26] proposed a fall detection method for indoor environments, based on the usage of the Kinect depth sensor in an "on-ceiling" configuration and on the analysis of depth frames. Rodrigo Ibañez et al. [27] proposed a lightweight approach to recognize gestures with Kinect by utilizing approximate string matching. Georgios Mastorakis et al. [28] fetch data from the depth image and detect falls by measuring the velocity based on the contraction or expansion of the width, height and depth of the 3D bounding box. Aguilar et al. [29] presented a 3D path planning system that uses a point cloud obtained from the robot workspace, with a Kinect V2 sensor to identify the interest regions and the obstacles of the environment. Alazrai R. [30] built a view-invariant descriptor for human activity representation by 3D skeleton joint positions to detect falls and other activities.

However, all of these methods have shortcomings. For example, in [31], the ratio of the height and width of the rectangle extracted from a person was used to determine whether a person falls. This method uses three features, i.e., human aspect ratio, effective area ratio and center variation rate. However, the human aspect ratio changes greatly when the relative positions of the camera and the target change, which results in a high false alarm rate and low accuracy of the detection system. Multi-cameras [32] and wide-angle camera-based [33] methods are effective ways to detect falls. However, the pre-works, i.e., camera installation, image calibration and 3D human identification, are difficult and complex. As for the deep learning method [34], the need for a huge number of labeled data makes it complicated in adjustment and poor in flexibility.

3. Our Approach for Fall Detection

In our proposed method, a new feature named torso angle is adopted. This feature together with the centroid height and their motion characteristics form a descriptor for human fall representation, called the human torso motion model (HTMM). Different from machine learning and deep learning methods, HTMM is a threshold-based approach for fall detection with high time efficiency because of the calculation of only the changing rates of the torso angle and centroid height. Before we elaborate the model, some technical terms will be detailed first in the following.

3.1. The Key Concepts and Definitions in HTMM

3.1.1. Torso Line and Torso Vector

There are in total 20 points of a person's skeleton that can be tracked by the Kinect sensor when the person is standing, while 10 points when sitting if used in front view. Additionally, the untracked joints can be estimated by the embedded program in Kinect. Especially when the part of the body that contains the joints can be detected, the positions of these untracked joints can be exactly estimated. Figure 1 shows the details of the joints in standing and sitting postures. A line connecting the shoulder center and the hip center is a key concept in our algorithm to calculate torso angle. This line is named as the torso line and marked in red as shown in Figure 1.

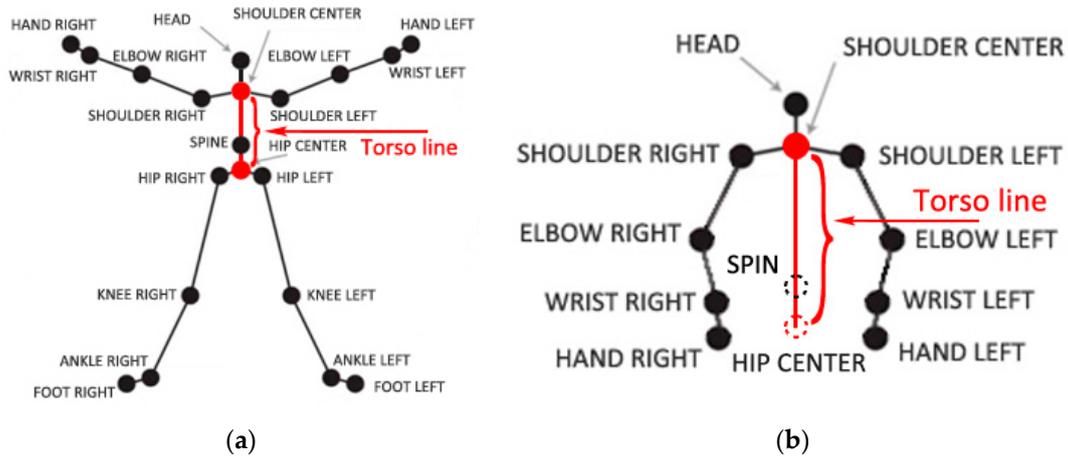


Figure 1. Joints of a person’s skeleton provided by Kinect. (a) Twenty joints of the skeleton that can be tracked when the person is standing; (b) 10 joints and two estimated joints of the skeleton when the person is sitting, where spin and hip center joints marked with two circles of dotted lines are estimated.

In a red, green, blue (RGB) image, each pixel only contains 2D position information. However, the depth image provides each pixel’s 3D position information. Figure 2 shows the joints in 3D depth image space. These joints of a person’s skeleton can be considered as pixels in the depth image. With these 3D joint points of the skeleton, the computer can understand the meaning of human gestures when a set of complex activities is made by human beings in front of the Kinect. As the main purpose of our system is to alarm as soon as a fall happens, we only take the shoulder center and hip center joints into consideration. In other words, our method only pays attention to the human torso.

The torso line can be represented as a vector and calculated by:

$$\overline{HS} = (X_h - X_s, Y_h - Y_s, Z_h - Z_s), \tag{1}$$

where \overline{HS} is the vector of the torso line. $H(X_h, Y_h, Z_h)$ is the position of the hip-center joint in 3D coordinates, and $S(X_s, Y_s, Z_s)$ indicates the position of the shoulder-center joint. We call the vector the torso vector in our approach.

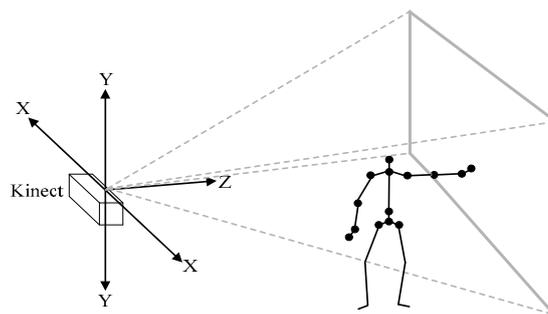


Figure 2. Depth image space created by the Kinect sensor.

3.1.2. Gravity Vector and Torso Angle

As we all know, the gravity line is always vertical to the ground. Therefore, all lines parallel with the y -axis can be seen as the gravity lines. Every two points on the gravity line can form a vector, which is called the gravity vector in our approach.

According to vector translation theory in solid geometry, the start point of the gravity vector and the start point of the torso vector can be moved to origin coordinates. We take a middle frame of a fall video for example. Figure 3 shows the process of forming the torso angle.

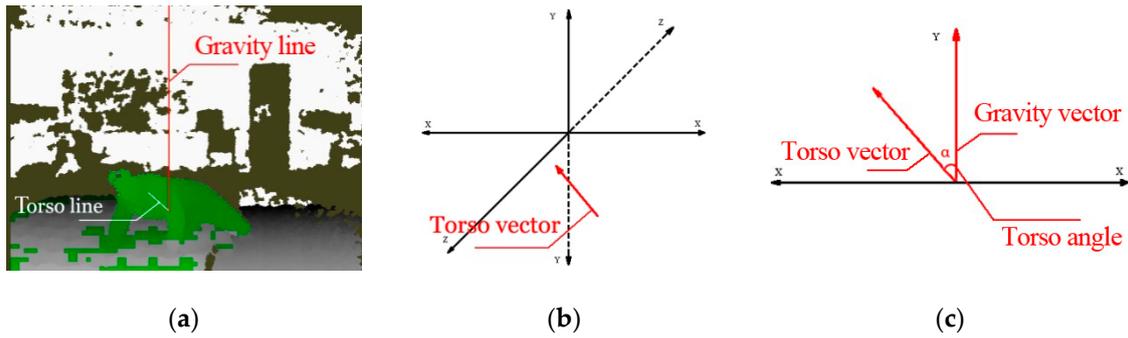


Figure 3. The process of extracting torso angle from a depth image. (a) A depth image fetched from a middle frame in a fall video; (b) torso vector represented in 3D coordinates; (c) torso angle represented in a 2D plane after vector translation.

According to the cosine law, the angle between these two vectors, named the torso angle, can be calculated by:

$$\cos(\alpha) = \cos(\overrightarrow{HS}, \overrightarrow{GH}) = \frac{\overrightarrow{HS} \cdot \overrightarrow{GH}}{|\overrightarrow{HS}| \times |\overrightarrow{GH}|}, \tag{2}$$

where \overrightarrow{GH} means the vector from any point on the y -axis to the hip center joint. Since the point on the gravity line is self-defined and the hip center is moved to the origin coordinates, so $\overrightarrow{GH} = (0, -Y_h, 0)$. Equation (2) can be also represented by:

$$\cos(\alpha) = \cos(\overrightarrow{GH}, \overrightarrow{HS}) = \frac{Y_h - Y_s}{\sqrt{(X_s - X_h)^2 + (Y_s - Y_h)^2 + (Z_s - Z_h)^2}}, \tag{3}$$

Figure 4 shows the torso angles in three common daily activities, standing, walking and sitting. Figure 4a–c shows the RGB images of each video and Figure 4d–f are their depth images marked with the torso line and the gravity line.

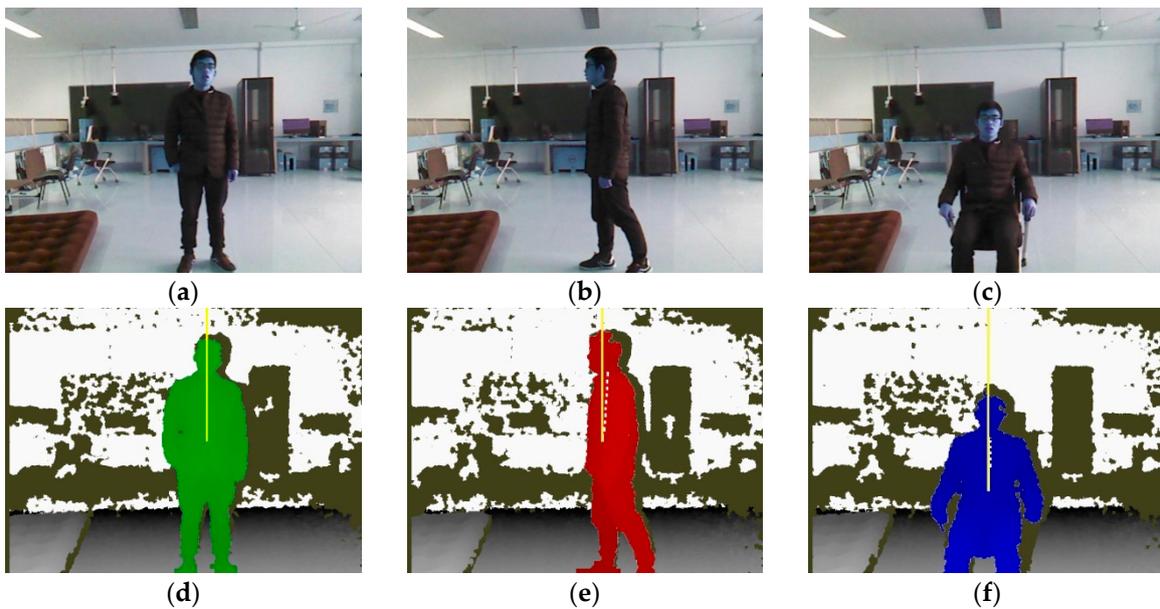


Figure 4. The torso angles in three common daily activities. (a–c) the RGB images of standing, walking and sitting; (d–f) their depth images, where the solid line is the gravity line and the dotted line is the torso line.

3.1.3. Centroid Height

Centroid height means the distance from the centroid point of a person to the ground. In our approach, we adopt the hip center joint as the rough centroid of a person. There are three reasons for using the hip center joint as the centroid, rather than calculating the human shape pixels and finding out the centric position. First, the exact centroid of the human shape is not necessary for our method. The approximated point is also a good choice because of the consideration of only the descending rate of its height. Secondly, the hip touching the ground is a common scene in nearly all kinds of falls. Therefore, although the position of the hip center joint is not the exact position of the centroid of the human shape, it is more representative for fall detection. Thirdly, the Kinect software development kit (SDK) provides a set of user-friendly and effective functions to track or estimate the position of the hip center.

Before calculating the centroid height, ground plane should be found out first. Although finding out the ground plane is a complex work, Kinect SDK provides four parameters for us, which can be used to exactly calculate the height from any points in the depth image to the ground plane. Equation (4) shows the four parameters.

$$Ax + By + Cz + D = 0, \tag{4}$$

where A, B, C and D are the ground parameters that can be used to calculate point height.

The height from the hip center to the ground H_c is calculated using Equation (5).

$$H_c = \frac{|Ax_c + By_c + Cz_c + D|}{\sqrt{A^2 + B^2 + C^2}}, \tag{5}$$

where $C(X_c, Y_c, Z_c)$ is the hip center joint in the depth image.

Through our experiments, we found that the centroid height can be exactly calculated by Equation (5) only when the ground is in the scene. When the Kinect sensor is installed too high to detect the ground, parameters cannot be accurately provided. All four parameters will be set to zero as their default values. Obviously, it will be a big obstacle to estimate one's centroid height. To address this issue, we use the right foot point to estimate the height of centroid when the ground parameters are zero. Therefore, Equation (5) is enhanced as:

$$H_c = \begin{cases} \frac{|Ax_c + By_c + Cz_c + D|}{\sqrt{A^2 + B^2 + C^2}} & \text{ground can be detected} \\ \sqrt{(x_c - x_f)^2 + (y_c - y_f)^2 + (z_c - z_f)^2} & \text{ground cannot be detected} \end{cases}, \tag{6}$$

where $f(X_f, Y_f, Z_f)$ is the position of the right foot joint in the 3D coordinate.

Figure 5 shows the value of centroid height in two typical daily activities, crouching and walking. From the experiment data, it can be concluded that the deviation caused by calculating based on the right foot joint is acceptable.

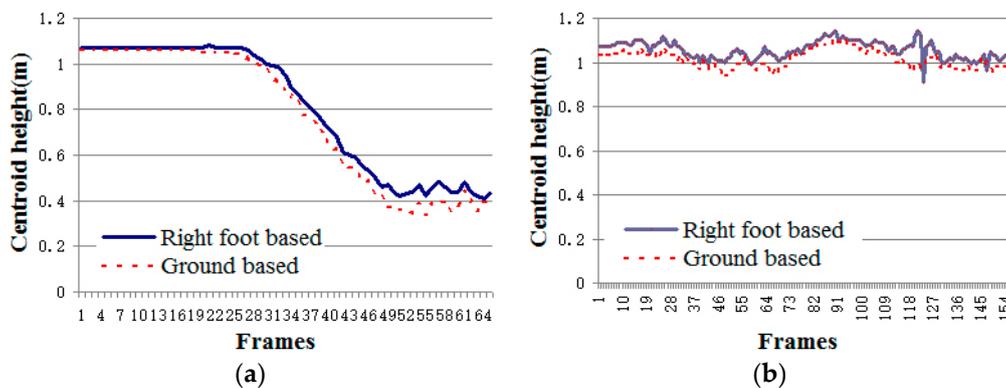


Figure 5. The values of centroid height recorded in two daily activities videos: crouching and walking. (a) Centroid height in crouching activity video; (b) centroid height in walking activity video.

3.2. Human Torso Motion Model

Fall activity is the balance loss of a person. When a fall happens, it is always accompanied with sharp changes in the torso angle and the centroid height. In the proposed method, we take the torso angle and the centroid height as two key features. There are our thresholds in our fall detection model, where T_α is the threshold for the start key frame detection, $T_{v\alpha}$ is the threshold of the changing rate of the torso angle, T_{vh} is the threshold of the velocity of the centroid height and T is the tracking time after the torso angle exceeded T_α . In HTMM. The changing rates of the torso angle and centroid height are calculated frame by frame after the torso angle exceeded T_α . The max values of these two features in the given period of time, T , are compared to their thresholds, respectively. When both rates exceed their corresponding thresholds $T_{v\alpha}$ and T_{vh} , a fall is detected.

For the result shown in the limit of stability test (LOST), an adult person can keep his/her balance with the body leaning forward/backward no more than 12.5 degrees and leaning left/right no more than 16 degrees [35]. In LOST, the person is asked to keep the whole body in a line. However, there is always an angle between the lower body and the upper body in daily activities. However, the person's torso always keeps parallel with the gravity line. Therefore, we take the torso line rather than the body line to form the torso angle with the gravity line. In our experiments, the torso angle exceeding 13 degree is the start of our detection model. The usage of only the torso angle to detect falls is insufficient and will result in low accuracy and a high false alarm rate. For example, bending or controlled lying down will be judged as a fall. To address this issue, the centroid height is imported in our model as the second feature.

Since a fall is an activity that usually happens in a short period of time (in our experiments, most of the fall samples last 1.1~1.6 s), for a video captured with 30 frames per second, there are 33~48 images during a fall. Thus, we built a motion model called HTMM for fall detection. In this model, we pay special attention to the changing rate of torso angle and centroid height in a given period of time.

Assume that the given period of time is represented as T , and $N(n \mid n \in Z \wedge n \leq T \times 30)$ denotes the frames' order in T .

Then, the person's torso angle (α), centroid height (H) and recorded time (T) in each frame can be represented as: $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$, $H = \{h_1, h_2, \dots, h_n\}$, $T = \{t_1, t_2, \dots, t_n\}$.

The changing rates of α and H in T of each frame are calculated using Equations (7) and (8).

$$V_t^\alpha = \left\{ x \mid x_i \in \mathbb{R} \wedge x_i = \frac{\alpha_i - \alpha_1}{t_i - t_1} \right\}, \tag{7}$$

$$V_t^h = \left\{ y \mid y_i \in \mathbb{R} \wedge y_i = \frac{h_i - h_1}{t_i - t_1} \right\}, \tag{8}$$

In HTMM, the max changing rates of torso angle and centroid height in the given period of time are used to compare with their thresholds, $T_{v\alpha}$ and T_{vh} . When both of them exceed their thresholds, HTMM outputs 1one to indicate that the activity is judged as a fall; else HTMM outputs zero to indicate that it is not. Our model can be represented as:

$$\text{HTMM}(\alpha, h, t) = \begin{cases} 1 & \text{Max}(V_t^\alpha) > T_{v\alpha} \wedge \text{Max}(V_t^h) > T_{vh}, \\ 0 & \text{else} \end{cases}, \tag{9}$$

Figure 6 shows the general block diagram of our approach.

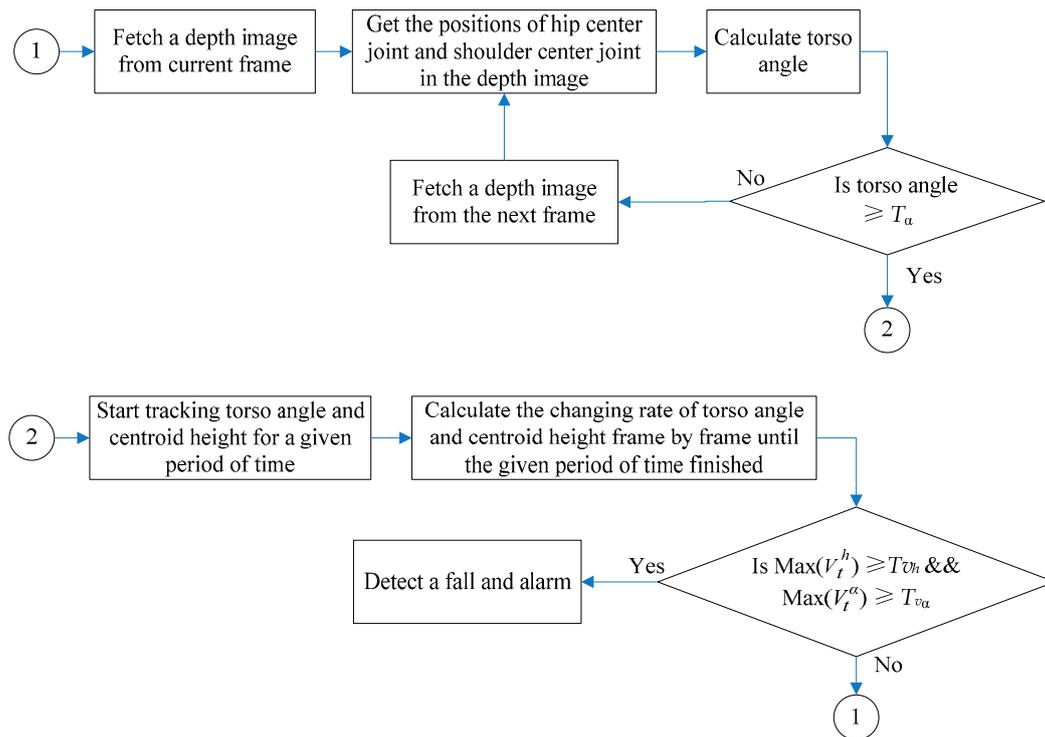


Figure 6. General block diagram of our approach; where $Max(V_t^h)$ is the max changing rate of the centroid height in the given period of time and $Max(V_t^\alpha)$ the max changing rate of the torso angle in the given period of time. The definitions of the given threshold values are elaborated in Section 4.

The general block can be divided into two main steps. The first step is calculating the torso angle of every frame and comparing it with the threshold value. When the torso angle reaches the threshold value, the program turns to Step 2. The second step is tracking the changing rate of torso angle and centroid height frame by frame in a given period of time. Once both rates exceed the threshold value, the activity is judged as a fall, and the system starts alarming; or the activity is considered as a normal activity when the given period of time is over, and the program turns back to Step 1 to start another loop. The pseudocode of our fall detection method is described in the following Algorithm 1.

Algorithm 1: Pseudocode of our proposed method for fall detection.

Input: Sequence of the skeleton frames captured by Kinect sensor

Output: bool bFallIsDetected

Loop 1:

while (torsoAngle < T_α)

{

 joint_shoulderCenter ← fetch shoulder center 3D position from current skeleton frame;

 joint_hipCenter ← fetch hip center 3D position from current skeleton frame;

 torsoAngle ← calculate torso angle of current skeleton frame;

}

Loop 2:

While (trackingTime < T)

{

 V_torsoAngle ← calculate the current torso angle changing rate by current frame;

 MaxV_torsoAngle ← update its value if V_torsoAngle is larger

 V_CentroidHeight ← calculate the current centroid height changing rate by current frame;

 MaxV_CentroidHeight ← update its value if V_CentroidHeight is larger

```

If (MaxV_torsoAngle >  $T_{va}$  && MaxV_CentroidHeight >  $T_{vh}$ )
{
    bFallsDetected = TRUE;
    fallAlarm();
}
}
goto Loop 1;

```

In HTMM, we introduced the torso angle to clearly judge whether the supervised person is in balance or not. Furthermore, the centroid height is used to find out whether the person is falling. These two features make HTMM have high accuracy in fall detection and work well in differentiating fall and fall-like activities. Figure 7 lists three typical fall-like activities that cause a high false alarm rate in previous approaches.

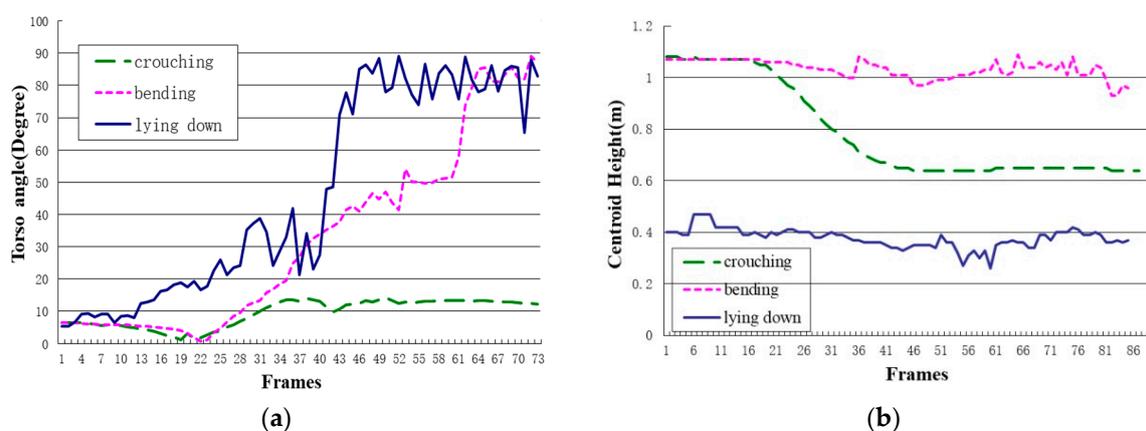


Figure 7. The data changing curve of the torso angle and centroid height. (a) The torso angle data changing curve; (b) the centroid height data changing curve.

In the bounding box ratio analysis approach, the ellipse shape analysis approach or even deep learning approach, the biggest shortcoming of these methods is that there is not a clear line between balance and unbalance. Therefore, when fall-like activity appears, the system will consider it as a fall. However, the torso angle together with the centroid height make it easy for HTMM to distinguish fall-like activities. Take the activities in Figure 7 as examples; each activity reaches only one threshold of two features. For controlled lying down and bending, although the torso angle changes greatly in a short period of time, the centroid height has no changes or changes little during this activity; while for crouching, the centroid height changes dramatically in a short period time, but the torso angle keeps at 12.5 degrees, which is lower than the threshold. Section 4 elaborates detailed statistics of our experiments.

4. Experiment and Results

4.1. Experimental Setup and Dataset

Our method was implemented using Microsoft Visual Studio 2013 Ultimate 2013 (Redmond, WA, USA) + emgu.cv3.1 (SourceForge, San Francisco, CA, USA) + Microsoft Kinect sensor v1.0 2013 (Redmond, WA, USA) on a PC using an Intel Core i7-4790 3.60-GHz processor, 8 GB RAM clocked at 1333 MHz. Since Kinect is a newly emerging sensor and the feature, the torso angle, is first raised and imported for fall detection, the existing depth action datasets cannot provide the necessary information we need. For example, Cornell Activity Datasets CAD-60/120 [366], the most commonly-used RGB-D dataset for action recognition, contains 12 activities, such as rinsing mouth, brushing teeth, wearing contact lens, etc., which do not provide falling action samples. Therefore, we built a dataset by ourselves for experiments.

This dataset was collected using Microsoft Kinect sensor v1.0, which was installed 1.4 m high from the ground. The people involved in the self-collected dataset are aged between 20 and 36, with different heights (1.70~1.81 m) and genders (four male and one female volunteers). The distance from the monitored person to the Kinect sensor is between 3 and 4.5 m. The actions performed by the single volunteer were separated into two categories: ADL (activity of daily living, including walking, controlled lying down, bending and crouching) and fall (four directions of falls, including forward, backward, left and right). For safety and realistic performance considerations, subjects performed the fall actions on a 15 cm-thick cushion. Each activity is repeated five times by each subject involved. There are in total 100 fall videos (each fall direction contains 25 videos) and 100 ADL videos (each ADL contains 25 videos). In these experiments, the five volunteers were asked to perform in slow motion to imitate the behavior of an elderly person at least one time in different kinds of activities. The joints positions in the 3D coordinates and joint heights were recorded frame by frame in an XML file. The fall samples contain 55~290 frames, and the ADL samples contain 65~317 frames in each video. All of the samples start by standing postures, which last 1~8 s, containing 30~240 frames.

4.2. Result and Evaluation

The human skeleton is used to track the monitored person and to discriminate human objects and other objects. Figure 8 shows that the Kinect-provided skeleton is an effective way to recognize the human object correctly, even when the joints are covered by a skirt or gown or there are other objects in the scene. Additionally, walking devices, such as a walking stick, can also be recognized and excluded from skeleton information by the embedded program of Kinect. Although the joints of the lower part of the skeleton may have a slight deviation where they are covered, the effect on the accuracy of HTMM can be ignored because of the consideration of only the shoulder center and hip center joints.

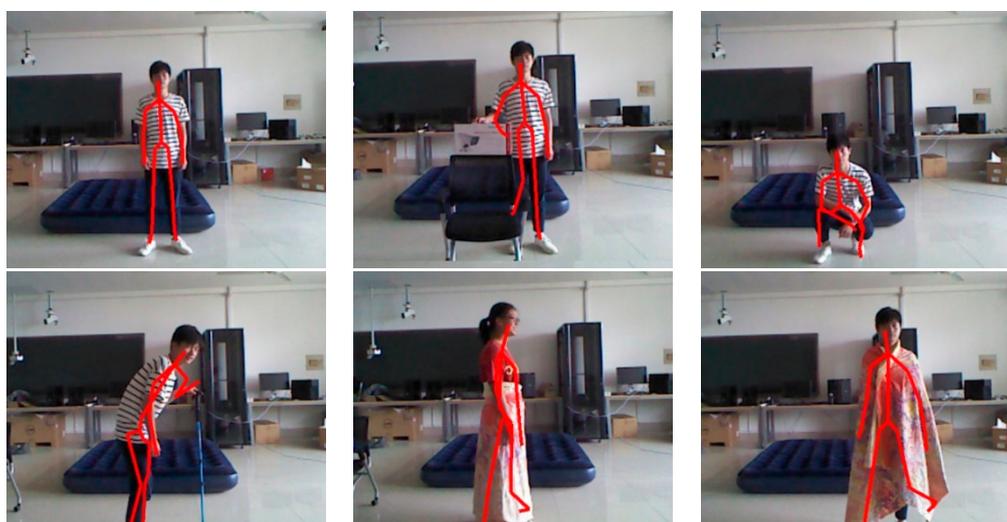


Figure 8. The detected skeletons with different postures, clothes or walking devices.

The four typical falls in Figure 9 show the common torso angle and centroid height changes. Before a fall is detected, the torso angle increases rapidly while the centroid height declines sharply. In HTMM, the changing rate of the torso angle, the changing rate of the centroid height and the tracking time are the three parameters that have a strong effect on the accuracy. In [Error! Reference source not found.7], the range of the decay rate of the centroid height is from 1.21~2.05 m/s. We set 1.21 m/s as its threshold value in our model. Through the experiments, our approach received the best accuracy rate when the parameters were set as follows:

$$T_{va} = 12 \text{ degree}/100 \text{ ms}$$

$$T_{vh} = 1.21 \text{ m/s}$$

$$T = 1300 \text{ ms}$$

where T_{va} is the threshold value of the torso angle increase rate, T_{vh} is the centroid height decay rate and T is the tracking time.

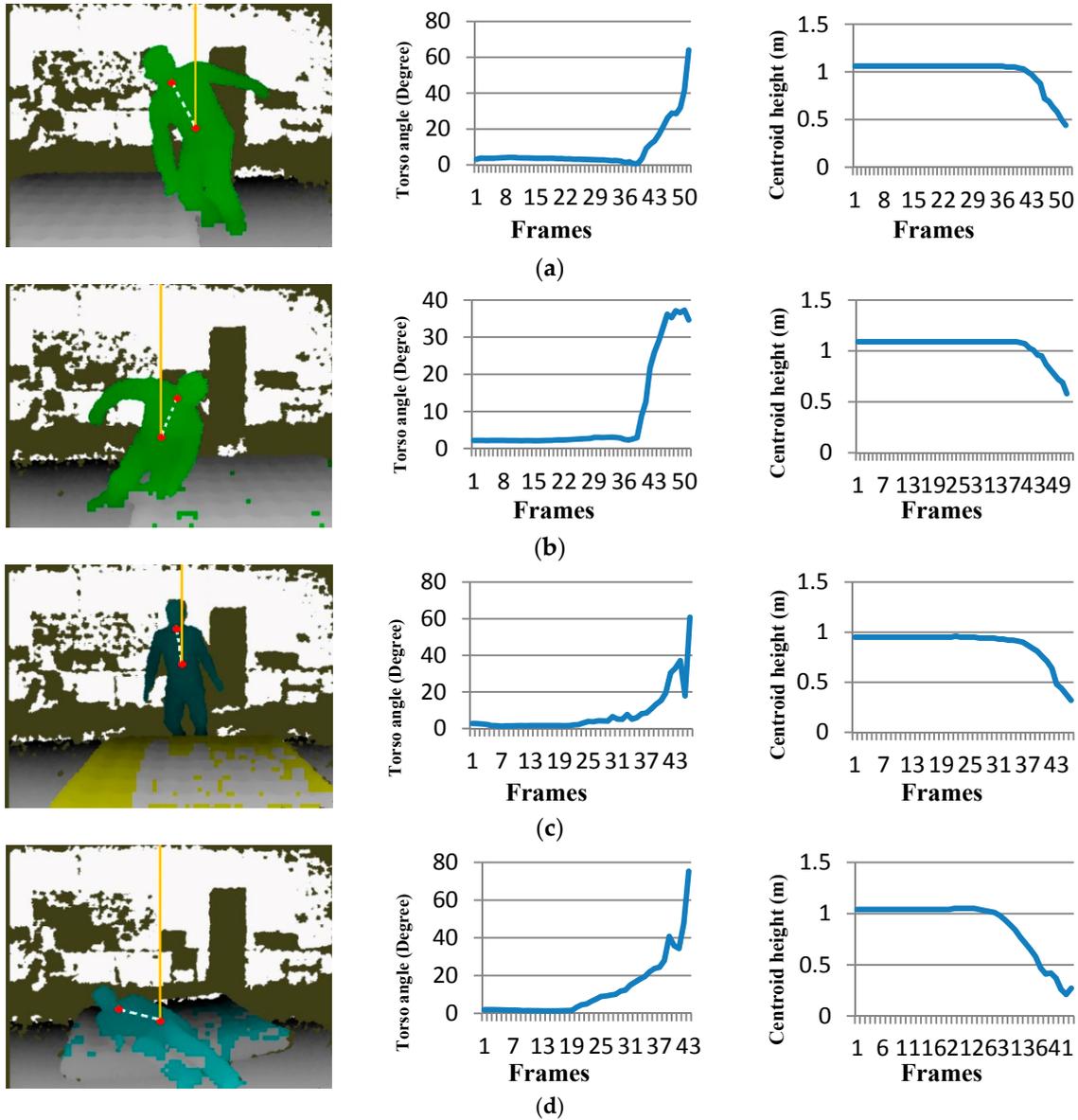


Figure 9. Detected frames of different falls and two features' curves. The left picture of each group is the depth image of the frame when our method detected the fall. The dotted line in the depth image stands for the torso line, while the solid line stands for the gravity line. Shoulder center joint and hip center joint were marked in red. (a) Fall left; (b) fall right; (c) fall forward; (d) fall backward.

As shown in Figure 10, most falls of our self-collected dataset happened in 1.1~1.6 s. Therefore, the values from 1100~1600 ms are all suitable for T_a . The max changing rates of the torso angle and the centroid height in this period are used in HTMM to check whether there is a fall. In the program, we took 1300 ms as the default value.

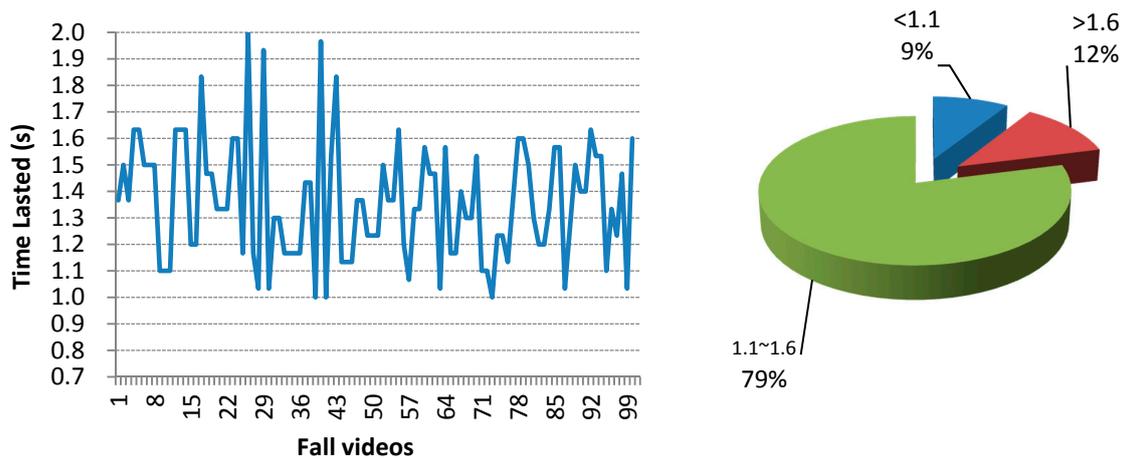
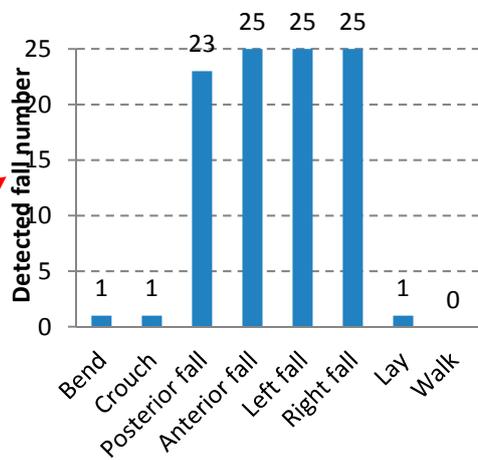


Figure 10. The duration of each fall in the self-collected dataset. There are in total 100 fall videos in the dataset.

Eleven groups of experiments were tested on the self-collected dataset with different T_{va} to find out its best default value. According to the results, when T_{va} was set 12 degree/100 ms, a fall can be detected in the middle of the activity; while a fall may not be detected if T_{va} was set too large, and false alarm rate increases when T_{va} was set too small. Table 1 shows the test results on our self-collected dataset with different values of T_{va} .

Table 1. The test results on the self-collected dataset with different T_{va} . TP: true positive; FN: false negative; FP: false positive; TN: true negative

T_{va}	Fall		Non-Fall	
	TP	FN	FP	TN
0	98	2	4	96
3	98	2	4	96
6	98	2	4	96
9	98	2	4	96
12	98	2	3	97
15	93	7	2	98
18	92	8	1	99
21	89	11	1	99
24	86	14	1	99
27	80	20	1	99
30	70	30	1	99
33	67	33	1	99



Figures 11 and 12 record the changing curves of the torso angle and centroid height of the five wrongly judged samples. By reviewing the data, we found that there are two major factors that affect the accuracy of our method. First and foremost, in rare situations, joint positions may be improperly provided by the embedded program of the Kinect sensor. For example, in Figure 11, from Frame 22~Frame 24 of the “lay” curve, the torso angle increased from 20~69.8 degrees. This abnormal transformation let our method make false judgments in two fall samples and one lay sample. Secondly, some samples were too quick to be detected correctly. For examples, “crouch” and “bend” curves show that the actions were accompanied with a quick lean of the upper body. For the first factor, the average filter or other filters can be employed to address the issue. For the second factor, the centroid height can be employed as the third feature to improve the accuracy of our method. These two improvements are under further investigation for our future work.

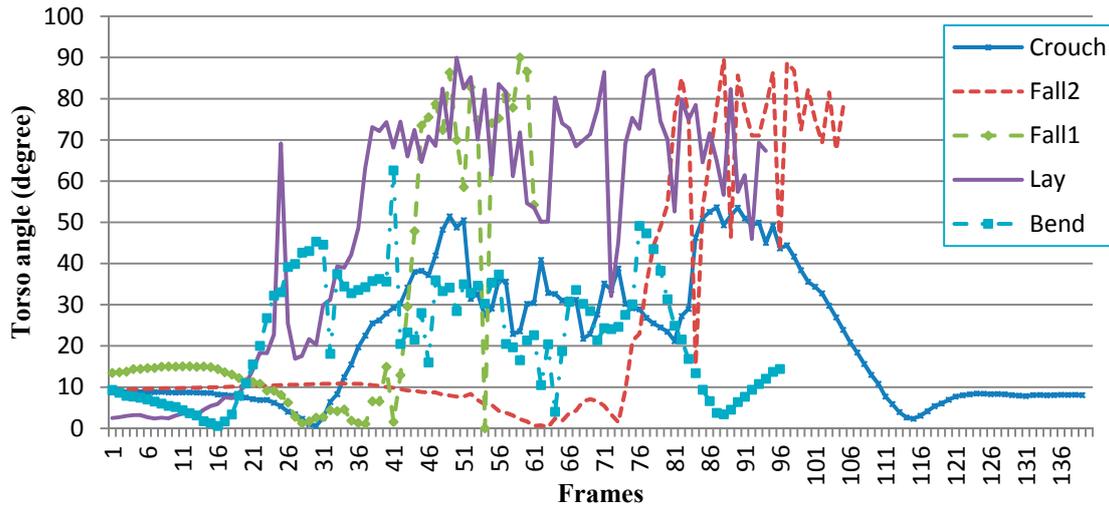


Figure 11. The torso angle changing curves of falsely judged samples.

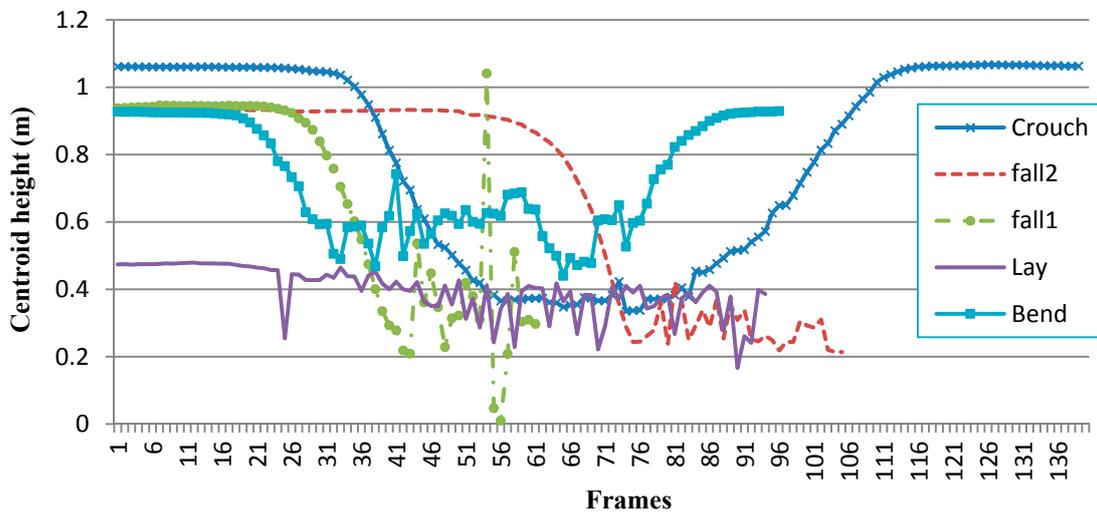


Figure 12. The centroid height changing curves of falsely judged samples.

Fall detection systems are expected to detect falls as soon as possible, so that the most severe consequences of falls will be avoided if the falling person can be assisted immediately. Our method has advantages in time efficiency because of the calculation of only two features. Figure 13 records the time consumptions of each frame in walk and fall.

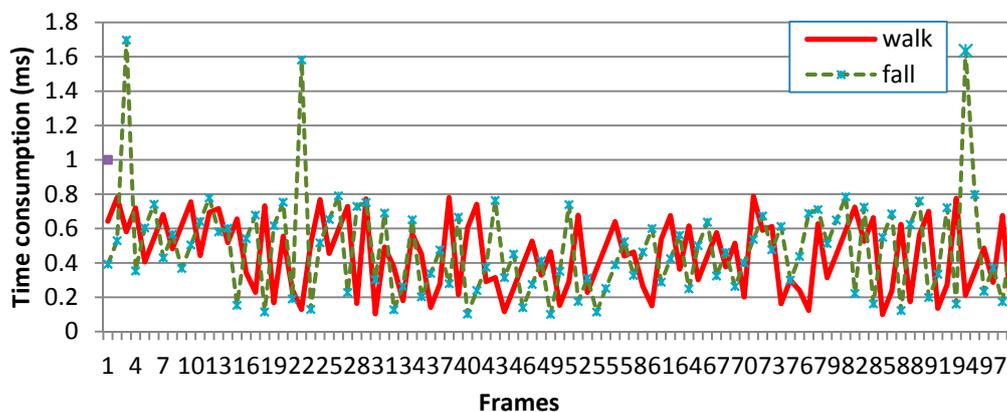


Figure 13. The time consumptions of each frame in walk and fall. The two test videos of this graph were selected randomly from the self-collected dataset.

As shown in Figure 13, most frames were processed in 0.1~0.8 ms. Although a few frames needed 1.6~1.8 ms, the processing time of each frame can be ignored because the time interval between two frames (33.33 ms in our 30 fps videos) is far larger than it.

False alarms comprise one of the biggest obstacles that prevent the computer vision-based fall detection method from being commercialized. Our approach works well in reducing the false alarm rate in fall-like activities' detection. Table 2 records the detailed results.

Table 2. Experimental results and evaluation of our approach.

Activity Type	The Number of Sample Videos	The Number of Detected as Fall Videos
Fall	100	98
Lay	25	1
Crouch	25	1
Bend	25	1
Walk	25	0

In Table 2, TP (true positive) denotes the fall samples that are judged as falls. FP (false positive) means the non-fall samples that are judged as falls. TN (true negative) denotes the non-fall samples that are judged as non-falls. FN (false negative) means the fall samples judged as non-falls. Then, we have $TPR = \frac{TP}{TP + FN}$, $TNR = \frac{TN}{TN + FP}$ and $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$.

We compared our proposed method with other Kinect sensor-based approaches [22,24,26,28]. All of these approaches were tested on our self-collected dataset. The results are recorded in Table 3.

Table 3. Comparison of the fall detection capability on our self-collected dataset with other Kinect-based fall detection approaches.

Approach	Fall				Crouch	Bend	Walk	Lay
	Forward	Backward	Left	Right				
Rougier, C. et al. [22]	25	25	25	25	11	20	0	1
Yang, L. et al. [24]	10	15	23	18	2	8	0	25
Gasparrini, S. et al. [26]	23	22	25	25	12	25	0	25
Mastorakis, G. et al. [28]	8	22	21	25	0	0	0	0
Our proposed method	25	23	25	25	1	1	0	1

As shown in Table 3, all of the approaches worked well in discriminating fall and walk. However, the height-based approaches [24,26] were unable to differentiate fall and other fall-like activities. Especially in terms of controlled lying down, all of the lay samples were wrongly judged as falls by the above two methods. The vertical height velocity-based approach [22] performed much better than height-based approaches [24,26] in discriminating fall and controlled lying down, but it performed poorly in differentiating fall, crouch and bend. The main reason is that both height-based and vertical velocity-based approaches did not lay a clear line between balance and unbalance. Hence, the fall-like activities are difficult or even impossible to detect correctly. As for the bounding box-based approach [28], it performed well in differentiating fall and fall-like activities. However, it performed badly in terms of detecting falling forward. By reviewing these undetected forward fall samples, we found that the main feature, V_{wd} , which was calculated by $V_{wd} = \sqrt{width^2 + depth^2}$ [28], changed more gently than it should be in the front-view condition.

Although the accuracy can be improved by programmatic calibration, it is an uneasy tricky work. After comparing and analyzing these Kinect sensor-based approaches for fall detection, the conclusion was made as given in Table 4.

Table 4. The general comparison of different categories of Kinect sensor-based fall detection approaches.

Approach	Capability of Detecting Falls	Capability of Discriminating Controlled Lying Down	Capability of Discriminating ADLs with Changing slightly in Height	Capability of Discriminating Fall-Like Activities with Changing Sharply in Height
Height based method	High	Poor	High	Poor
Height velocity based method	High	High	High	Poor
Bounding box/ratio of the height and width based methods	Medium	High	High	High
Our proposed method	High	High	High	High

5. Conclusions

In this paper, we proposed a new and fast fall detection method based on the Kinect sensor. A motion model named the human torso motion model (HTMM) is proposed. A new and significant feature named torso angle was firstly adopted in our approach for fall detection. Comparing with existing posture-based approaches, machine learning-based methods and wearable device-based approaches, our proposed approach has obvious advantages in privacy protection and fall-like activities' differentiation. Our method is time efficient and robust because of the calculation of only the changing rate of torso angle and centroid height. Moreover, the usage of only the Kinect sensor is inexpensive, affordable for common families and capable of being easily applied in the elderly person's home. However, compared to a monocular video camera, the limitation of the utilized Kinect sensor is the relatively narrow field-of-view. The best distance from the monitored person to the Kinect sensor should be within 7 m. Beyond this distance, the depth data become unreliable. Nevertheless, the Kinect-based approaches are still effective, efficient and widely used for fall detection.

Although the Microsoft Kinect sensor v2.0 was launched in 2013 and is more powerful than its 1.0 version, Microsoft Kinect sensor v1.0 is sufficient for our proposed method because of the calculation of only two joints and the cost reduction. The mechanism of our proposed method is suitable for Kinect sensor 2.0, as well. In our future work, Kinect sensor 2.0, which is more sensitive than Kinect 1.0, will be adopted to collect depth images that contain more information for more possible improvement. Our fall detection model will be continuously improved by further importing other filters and other features to address exceptional situations.

Acknowledgments: This research was supported by the National Natural Science Foundation of China under Grants 61762061 and 61662044 and the Natural Science Foundation of Jiangxi Province, China, under Grant 20161ACB20004.

Author Contributions: All authors of the paper have made significant contributions to this work. Weidong Min led the project, directed and revised the paper writing. Lei Yue Yao conceived of the idea of the work, wrote the manuscript and analyzed the experiment data. Keqiang Lu collected the original data of the experiment and participated in programming.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ozcan, A.; Donat, H.; Gelecek, N.; Ozdirenc, M.; Karadibak, D. The relationship between risk factors for falling and the quality of life in older adults. *BMC Public Health* **2005**, *5*, 1–6.
2. Yardley, L.; Smith, H. A prospective study of the relationship between feared consequences of falling and avoidance of activity in community-living older people. *Gerontologist* **2002**, *42*, 17–23.
3. Salvà, A.; Bolívar, I.; Pera, G.; Arias, C. Incidence and consequences of falls among elderly people living in the community. *Med. Clin.* **2004**, *122*, 172–1766.

4. Doughty, K.; Lewis, R.; McIntosh, A. The design of a practical and reliable fall detector for community and institutional telecare. *J. Telemed. Telecare* **2000**, *6*, 150–154.
5. Huang, C.; Chiang, C.; Chen, G. Fall detection system for healthcare quality improvement in residential care facilities. *J. Med. Biol. Eng.* **2010**, *30*, 247–252.
6. Mubashir, M.; Shao, L.; Seed, L. A survey on fall detection: Principles and approaches. *Neurocomputing* **2013**, *100*, 144–152.
7. Mathie, M.J.; Coster, A.C.F.; Lovell, N.H.; Celler, B.G. Accelerometry: Providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiol. Meas.* **2004**, *25*, 1–20.
8. Nguyen, T.T.; Cho, M.C.; Lee, T.S. Automatic fall detection using wearable biomedical signal measurement terminal. In Proceedings of the International Conference of the IEEE Engineering in Medicine & Biology Society, Cheongju, Korea, 3–6 September 2009.
9. Sixsmith, A.; Johnson, N. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Comput.* **2004**, *3*, 42–47.
10. Kangas, M.; Vikman, I.; Wiklander, J.; Lindgren, P.; Nyberg, L.; Jämsä, T. Sensitivity and specificity of fall detection in people aged 40 years and over. *Gait Posture* **2009**, *29*, 571–574.
11. Zhuang, X.; Huang, J.; Potamianos, G.; Hasegawa-Johnson, M. Acoustic fall detection using Gaussian mixture models and GMM super vectors. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009.
12. Alwan, M.; Rajendran, P.J.; Kell, S.; Mack, D. A Smart and Passive Floor-Vibration Based Fall Detector for Elderly. *Inf. Commun. Technol.* **2006**, *1*, 1003–1007.
13. Feng, G.; Mai, J.; Ban, Z.; Guo, X.; Wang, G. Floor pressure imaging for fall detection with fiber-optic sensors. *IEEE Pervasive Comput.* **2016**, *15*, 40–47.
14. Rajpoot, Q.M.; Jensen, C.D. Security and Privacy in Video Surveillance: Requirements and Challenges. In Proceedings of the 29th IFIP International Information Security and Privacy Conference, Marrakech, Morocco, 2–4 June 2014.
15. Chaaoui, A.A.; Padilla López, J.R.; Ferrándezpastor, F.J.; Nietohidalgo, M.; Flórezrevuelta, F. A vision-based system for intelligent monitoring: Human behaviour analysis and privacy by context. *Sensors* **2014**, *14*, 8895–8925.
16. Rougier, C.; Meunier, J.; St-Arnaud, A.; Rousseau, J. Robust video surveillance for fall detection based on human shape deformation. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 611–622.
17. Dong, Q.; Yang, Y.; Wang, H.; Xu, J.H. Fall alarm and inactivity detection system design and implementation on Raspberry Pi. In Proceedings of the 17th IEEE International Conference on Advanced Communications Technology, Pyeonghchang, South Africa, 1–3 July 2015.
18. Kwolek, B.; Kepski, M. Fuzzy inference-based fall detection using Kinect and body-worn accelerometer. *Appl. Soft Comput.* **2016**, *40*, 305–318.
19. Khan, M.J.; Habib, H.A. Video analytic for fall detection from shape features and motion gradients. *Lect. Notes Eng. Comput. Sci.* **2009**, *1*, 1311–1316.
20. Williams, A.; Ganesan, D.; Hanson, A. Aging in place: Fall detection and localization in a distributed smart camera network. In Proceedings of the 15th International Conference on Multimedia, Augsburg, Germany, 24–29 September 2007.
21. Yu, M.; Yu, Y.; Rhuma, A.; Naqvi, S.M.; Wang, L.; Chambers, J.A. An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment. *IEEE J. Biomed. Health Inform.* **2013**, *17*, 1002–1014.
22. Rougier, C.; Meunier, J.; St-Arnaud, A.; Rousseau, J. 3D head tracking for fall detection using a single calibrated camera. *Image Vis. Comput.* **2013**, *31*, 246–254.
23. Olivieri, D.N.; Mez Conde, I.; Vila Sobrino, X. Eigenspace-based fall detection and activity recognition from motion templates and machine learning. *Expert Syst. Appl.* **2012**, *39*, 5935–5945.
24. Yang, L.; Ren, Y.; Hu, H.; Tian, B. New fast fall detection method based on spatio-temporal context tracking of head by using depth images. *Sensors* **2015**, *15*, 23004–23019.
25. Rougier, C.; Auvinet, E.; Rousseau, J.; Mignotte, M.; Meunier, J. Fall detection from depth map video sequences. In Proceedings of the 9th International Conference on Smart Homes and Health Telematics, Montreal, QC, Canada, 20–22 July 2011.
26. Gasparrini, S.; Cippitelli, E.; Spinsante, S.; Gambi, E. A depth-based fall detection system using a Kinect® sensor. *Sensors* **2014**, *14*, 2756–2775.

27. Ibañez, R.; Soria, Á.; Teyseyre, A.; Rodriguez, G.; Campo, M. Approximate string matching: A lightweight approach to recognize gestures with Kinect. *Pattern Recognit.* **2016**, *62*, 73–86.
28. Mastorakis, G.; Makris, D. Fall detection system using Kinect's infrared sensor. *J. Real-Time Image Process.* **2014**, *9*, 635–646.
29. Aguilar, W.; Morales, S. 3D environment mapping using the Kinect v2 and path planning based on RRT algorithms. *Electronics* **2016**, *5*, 70–87.
30. Alazrai, R.; Momani, M.; Daoud, M.I. Fall detection for elderly from partially observed depth-map video sequences based on view-invariant human activity representation. *Appl. Sci.* **2017**, *7*, 316–335.
31. Hong, L.; Zuo, C. An improved algorithm of automatic fall detection. *AASRI Procedia* **2012**, *1*, 353–358.
32. Wang, X.; Li, M.; Ji, H.; Gong, Z. A novel modeling approach to fall detection and experimental validation using motion capture system. In Proceedings of the 2013 IEEE International Conference on Robotics and Biomimetics, Shenzhen, China, 12–14 December 2013.
33. Bosch-Jorge, M.; Sánchez-Salmerón, A.J.; Valera, Á.; Ricolfe-Viala, C. Fall detection based on the gravity vector using a wide-angle camera. *Expert Syst. Appl.* **2014**, *41*, 7980–7986.
34. Feng, P.; Yu, M.; Naqvi, S.M.; Chambers, J.A. Deep learning for posture analysis in fall detection. In Proceedings of the International Conference on Digital Signal Processing, Milan, Italy, 19–21 November 2014.
35. Melzer, I.; Benjuya, N.; Kaplanski, J.; Association between ankle muscle strength and limit of stability in older adults. *Age Ageing* **2009**, *38*, 119–123.
36. Sung, J.; Ponce, C.; Selman, B.; Saxena, A. Unstructured human activity detection from RGBD images. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012.
37. Kobayashi, J.; Abdulrazak, L.; Mokhtari, M. Inclusive society: Health and wellbeing in the community, and care at home. In Proceedings of the 11th International Conference on Smart Homes and Health Telematics, Singapore, Singapore, 19–21 June 2013.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).