Front. Mech. Eng. 2018, 13(3): 427–441
https://doi.org/10.1007/s11465-017-0465-7

RESEARCH ARTICLE

**Sajjad TAGHVAEI, Kazuhiro KOSUGE**

# Image-based fall detection and classification of a user with a walking support system



**Abstract** The classification of visual human action is important in the development of systems that interact with humans. This study investigates an image-based classification of the human state while using a walking support system to improve the safety and dependability of these systems. We categorize the possible human behavior while utilizing a walker robot into eight states (i.e., sitting, standing, walking, and five falling types), and propose two different methods, namely, normal distribution and hidden Markov models (HMMs), to detect and recognize these states. The visual feature for the state classification is the centroid position of the upper body, which is extracted from the user's depth images. The first method shows that the centroid position follows a normal distribution while walking, which can be adopted to detect any non-walking state. The second method implements HMMs to detect and recognize these states. We then measure and compare the performance of both methods. The classification results are employed to control the motion of a passive-type walker (called "RT Walker") by activating its brakes in non-walking states. Thus, the system can be used for sit/stand support and fall prevention. The experiments are performed with four subjects, including an experienced physiotherapist. Results show that the algorithm can be adapted to the new user's motion pattern within 40 s, with a fall detection rate of 96.25% and state classification rate of 81.0%. The proposed method can be implemented to other abnormality detection/classification applications that employ depth image-sensing devices.

Received November 26, 2016; accepted May 13, 2017

Sajjad TAGHVAEI (✉)
School of Mechanical Engineering, Shiraz University, Shiraz 71936-16548, Iran
E-mail: sj.taghvaei@shirazu.ac.ir

Kazuhiro KOSUGE
Department of Bioengineering and Robotics, Tohoku University, Sendai 980-8579, Japan

**Keywords** fall detection, walking support, hidden Markov model, multivariate analysis

## 1 Introduction

The need to provide dependable and self-assistive devices for the elderly is increasing due to population aging [1]. Among the different daily-life activities, physical support in mobility-related tasks, such as walking, is still the main challenge that should be tackled [2]. One problem that must be considered is the frequent occurrences of hospital admissions and injuries due to falls reported for this vulnerable population. Approximately 30% of elderly people fall at least once a year [3]. This percentage includes 87% of users with walking support systems, who are treated at hospital emergency departments for injuries from falls that involve walkers and canes [4].

Researchers have proposed numerous fall detection algorithms with various sensory setups for different applications [5]. The use of accelerometers [6,7], gyroscopes [8], single and multiple cameras [9], laser range finders [10], depth sensors [11,12], wireless networks [13], and other novel sensory equipment (e.g., as sensing floors [14]) has been investigated. The proposed algorithms utilize threshold-based methods [15], multi-sensor data fusion methods [16], and machine learning algorithms [6,17]. A two-layer support vector machine has been proposed in Ref. [18] to identify human posture (standing vs. walking). Another recent work [19] proposed a combination of threshold-based and machine learning in a hierarchical framework for fall detection. However, the detection occurs after the person has rested on the ground (i.e., the rest phase) similar to other common algorithms [19].

The aforementioned methods can be evaluated according to accuracy, availability of sensory setup, and applicability to different platforms. Many approaches have improved the accuracy of results to a satisfactory extent, and a quantitative comparison of different methods

has been presented in Ref. [20]. However, the use of a commonly wearable and expensive sensory setup [6,10] questions the applicability of the proposed methods. In addition, the lack of focus on real-time algorithms [9,21] also disqualifies the methods for walking assistive devices, such as the commonly utilized walkers.

Despite the variety in sensor technologies and classification algorithms, few studies have investigated the following concepts: Real-time fall prediction or detection before the person has already fallen; implementation of fall detection in assistive technologies for possible fall prevention, and a dynamic model of walking that can generate different falling scenarios.

Previous authors have proposed methods for the state classification of a user by visual features extracted by principal component analysis [11], in which the feature extraction has comparatively higher computation cost and requires a database. The upper body centroid of the user is considered as the visual feature in Ref. [22]. However, the state classification is not proposed.

Using the Microsoft Xbox sensor called Kinect, the current authors propose several methods for real-time abnormal state detection (i.e., sitting, standing, or falling) based on depth image data. A recent study [23] compared three types of common non-wearable sensor technologies for fall detection among those living in elderly home care facilities. The said study used a depth camera for an ongoing investigation, in which depth camera systems have been installed in 94 additional older adult apartments [23].

In this work, the state classification algorithms are implemented and tested on a previously designed passive walker robot called "RT Walker" [24]. The robot is equipped with servo brakes, which can be activated in any non-walking state. The system can prevent the user from falling and assist him/her in sitting down, standing up, and regaining stability after falling in this manner.

The position of the upper body centroid is utilized as the visual feature in the present work. Two different probabilistic models are employed for the state classification. First, it is proved that the upper body centroid position during walking with a walker can be fitted by a multivariate normal distribution function. The model is adopted to detect any non-walking state and control the walker's motion accordingly. Considering the time dependence of human behavior, a hidden Markov model (HMM) is also utilized for the state classification. The model is then applied for the non-walking states of detection and recognition.

Among the different classifiers for visual human action recognition, we specifically require an algorithm with low computational cost for time-series data, in order to provide fall detection/classification before the user has actually fallen. HMMs have been reported, such as a classifier for time-series data analysis in one-class classification [25] (i.e., fall detection [6]) and multi-class classification [26] (i.e., fall classification [11]). Then, the algorithms are tested with four subjects, including an experienced physiotherapist asked to imitate the falling accidents of real users. The performance results of the methods are discussed and compared.

The current study attempts to utilize a real-time fall detection of an RT Walker user by using a vision system (i.e., Kinect). Multi-class classification, which employs the depth images of the user's upper body, is adopted to increase the dependability of walking support systems. This classification requires experimental data, which are extracted from 5 different users.

The rest of this paper is organized as follows. The developed RT Walker is described in Section 2. Section 3 explains the algorithm framework utilized for the state estimation and motion control of the walker. State classification by normal distribution function fitting is described in detail in Section 4. Aspects of the HMM-based state classification and the walker's motion control are discussed in Sections 5 and 6, respectively. Experimental results and their discussion are presented in Sections 7 and 8, respectively. Finally, the concluding remarks are presented in Section 9.

## 2 RT Walker

Inspired by the proposed concept of passive robotics by Goswami et al. [27], we used servo brakes to control the motion of a passive-type walker. Given that unintentional movements of the system cannot occur, passive robots are intrinsically safe and suitable for systems that physically interact with humans. The developed RT Walker is a prototype with a support frame, two passive casters, and a controller (Fig. 1 [24]). Unlike other passive robots, the RT Walker does not require any servo motor to control the servo brakes. Instead, the rear wheels are equipped with powder brakes, which can be employed to change the brake torques of the rear wheels according to the input current.

The robot is equipped with a Microsoft Xbox sensor (or Kinect) to obtain the user's depth images (Fig. 2). The sensor data are robust and reliable under different optical conditions. Moreover, this option is considerably cheaper than the previously adopted laser range finders [28].

## 3 Framework

Given that the walker is passive, motion control involves the activation of brakes on the rear wheels for any non-walking state. These states can be classified into six types shown in Fig. 3 based on the common accidents that involve walker users (i.e., four falling scenarios, sitting, and standing). Figure 3 are taken from an experienced physiotherapist who imitated typical accidents that befall

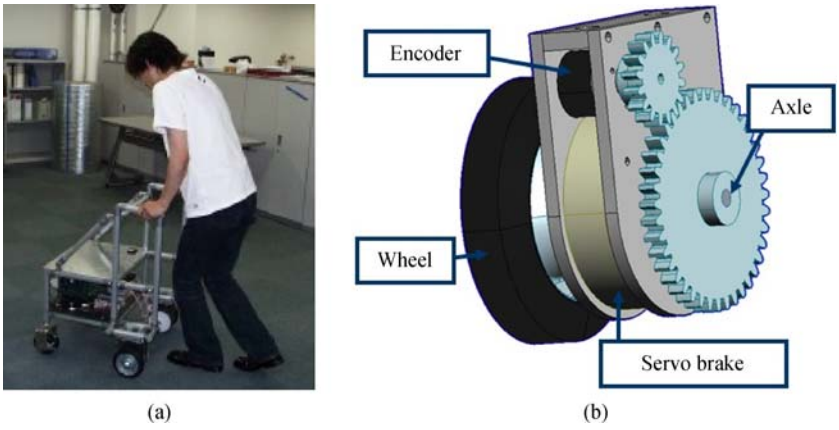


**Fig. 1** (a) Prototype of the RT Walker; (b) the rear wheel and servo brake system [24]

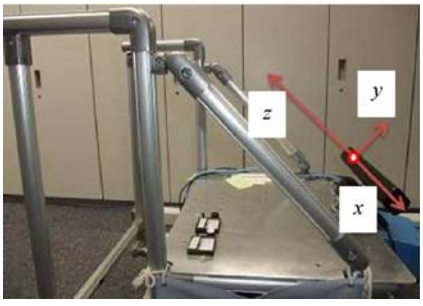


**Fig. 2** Depth sensor installed on the RT Walker and the attached coordinate system

real walker users. Falling can occur in many ways, but it can be roughly classified as one of these typical cases. Standing is referred to as either getting up from a sitting state or regaining one's stability after falling.

The algorithm employed for the state estimation and motion control of the walker is shown in Fig. 4. These states enable us to validate the accuracy of the state classification algorithm by indicating whether the algorithm detects all states as non-walking states. The occurrence of falling accidents can be significantly decreased by stopping the walker in non-walking states. The walker can then be utilized as a support for standing and sitting.

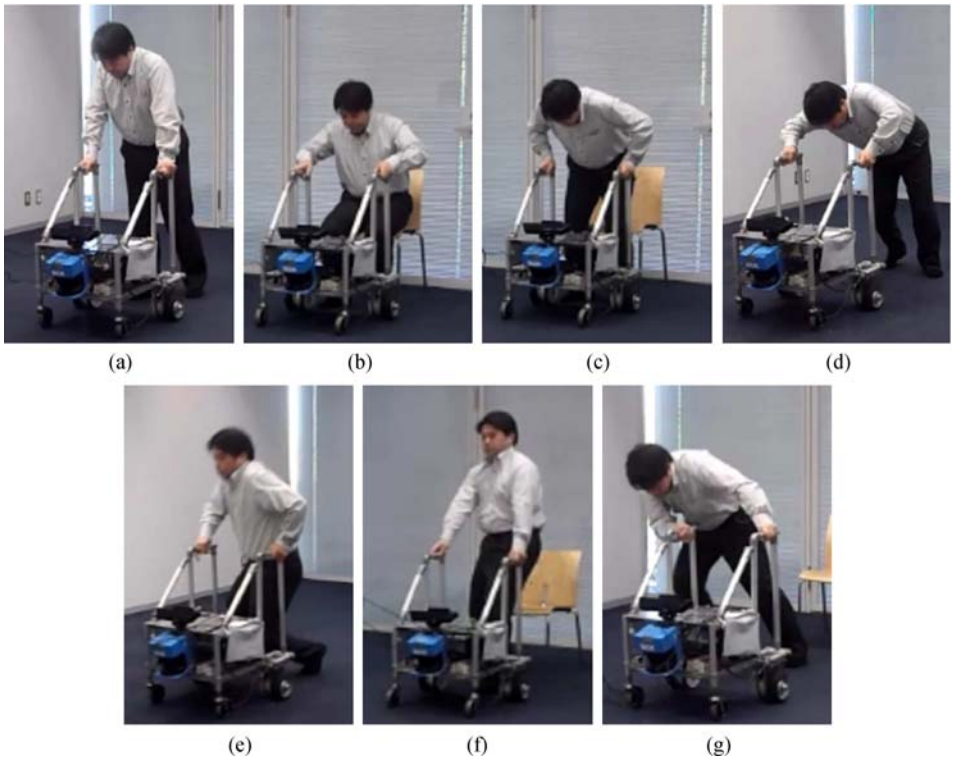


**Fig. 3** Human states while using a walker, as demonstrated by an experienced physiotherapist. *Fall side* is categorized into *fall right* and *fall left*. (a) Walk; (b) sit; (c) stand; (d) fall forward; (e) fall down; (f) fall back; (g) fall side

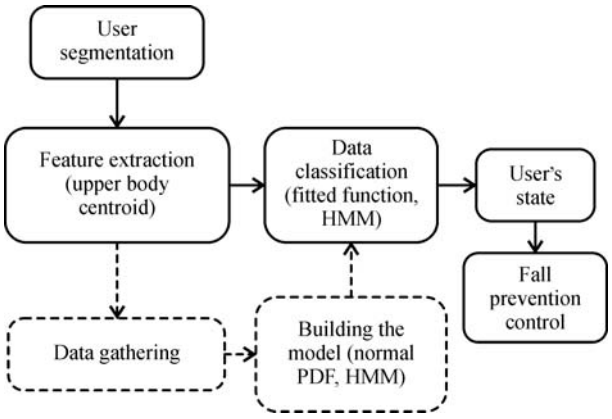


**Fig. 4** Framework for the state estimation and control of the walker. PDF: Probability distribution function

We use two different methods for state classification at this point. The upper body centroid position is the feature extracted to classify a user's state. With respect to the classification methods, the first approach is based on a multivariate normal distribution function fitted on the upper body centroid position during walking. Classification is performed by trained HMMs using data gathered from users in the second approach.

The first step for both approaches is user segmentation, which is mainly conducted by distance slicing. In particular, a threshold is set on the distance of each pixel from the sensor, which is given as a value in millimeters by Kinect. All the pixels closer to the sensor than the threshold are considered as the foreground, whereas the rest of the pixels are set to zero (Fig. 5). Several parts of the walker handle may get into the foreground. These parts are omitted by basic morphological transformations [29], because the handle position and shape are known beforehand. The feature is then extracted from the user's depth image, and a probabilistic model is built for a specific state. State classification involves verifying whether the extracted feature from the current motion belongs to the probabilistic model. Classification may occur at two levels. The first level determines whether the user is walking, regardless of the type of falling, sitting, or standing. Consequently, the probabilistic model is built by using only walking data, as is the case in any one-class classification problem [25].

A second level of classification can be performed to detect the actual type of non-walking state (i.e., sit, stand, fall forward, and fall down) to improve the estimation accuracy of the user's state and the flexibility of the robot response. In this case, a probabilistic model (i.e., HMM) is built for each state to detect the state class that can be the best match for the current motion state.

## 4 State classification by normal distribution function fitting

Given that the upper part of the body does not move significantly when walking with a walker and that the centroid can be a suitable approximation of the center of mass, we employ the upper body centroid position as the main feature for the state classification. By fitting 3D normal distribution functions on the obtained $x$, $y$, and $z$ values of the upper body centroid, we can detect any non-walking state and control the walker's brakes. For this purpose, we must verify whether normality is a valid assumption for a multivariate distribution fitting. Therefore, several normality tests are conducted to validate this approach. Moreover, a statistical analysis of the human upper body centroid during walking is a potential task for further applications that involve human motion analysis.

### 4.1 Extraction of the upper body centroid

The user must first be segmented from the background to extract the feature. For this application, the user is always within a specific range from the sensor, and no occlusions or sources of interference are observed in the image. However, the system is mobile, and the background changes continuously during motion, thus making user segmentation difficult. This scenario can be resolved by applying a depth sensor. User segmentation is performed

(a) (b)

**Fig. 5** (a) Color image captured by Kinect; (b) the segmented user's mask image obtained by distance slicing

by adopting the distance slicing method with the depth information from the sensor and probable distance of the user from the robot during walking. Although the foreground may still contain several pixels from the environment, such as robot parts, they can be removed using simple image-processing filters. The color image and segmented user's mask of the depth image captured by Kinect are shown in Fig. 5. The results are robust and are insensitive to light conditions or to the color of the user's skin/clothing.

The $x$, $y$, and $z$ values of the foreground pixels (Fig. 4) are taken from the sensor, and the average of these values is taken as the centroid of the human upper body. The values are calculated in accordance with the coordinate system attached to the walker, as shown in Fig. 2.

### 4.2 Data mapping

The $x$, $y$, and $z$ values of the upper body centroid are calculated and obtained for a user while he/she is walking normally with the walker. State classification is based on a 3D probabilistic model of the centroid location. Therefore, we should determine whether a proper cumulative distribution function can be fitted on the collected data. Individual checks for the univariate normality of the $x$, $y$, and $z$ samples cannot guarantee the multivariate normality of the sets unless the samples are independent from each other [30].

The multivariate normal distribution function of order $p$ is defined as

$$g(x)=\frac{1}{(\sqrt{2\pi})^p\sqrt{|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \quad (1)$$

where $\boldsymbol{\mu}$ is the mean vector, and $\boldsymbol{\Sigma}$ is the $p\times p$ covariance matrix of the samples. The contours of constant probability are defined by $(x-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})=\text{const}$. For $p=2$, they are ellipses centered at $\boldsymbol{x}=\boldsymbol{\mu}$ with axes parallel to the eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$. For the 3D case, the contours are ellipsoids. For an independent dataset, all non-diagonal values of $\boldsymbol{\Sigma}$ are equal to zero, and the axes of the ellipses (2D case) or ellipsoids (3D case) are parallel to the $x$, $y$, and $z$ axes.

The covariance matrix is calculated, and a contour of constant probability is drawn to check the dependence of the data samples. The distribution of collected data and ellipsoid of constant probability for a normal fitted function in the $xy$, $xz$, and $xyz$ planes are shown in Fig. 6. The data collected from a user while walking with the RT Walker indicate that the $x$, $y$, and $z$ coordinates of the upper body centroid are dependent on one another. This scenario can be obtained by plotting the contours of constant probability of the data. If the data are dependent, then the axes of the ellipsoid contour are not parallel to the $x$ and $y$ axes (Fig. 6). In this scenario, the univariate normality test for these data cannot guarantee the validity of fitting a 3D normal distribution. Therefore, the data are mapped such that they are independent, and univariate normality tests are conducted on the mapped data. Linear mapping is carried out based on decomposition of the covariance matrix by its eigenvalues, and the test results can be utilized to verify the validity of the multivariate normality fitting.

The proposed mapping function is given by

$$\boldsymbol{X}=\boldsymbol{A}\boldsymbol{U}\boldsymbol{x},\ \boldsymbol{U}=[\,\boldsymbol{v}_1\quad\boldsymbol{v}_2\quad\boldsymbol{v}_3\,],$$

$$\boldsymbol{A}=\begin{bmatrix}\frac{1}{\sqrt{\lambda_1}} & 0 & 0\\ 0 & \frac{1}{\sqrt{\lambda_2}} & 0\\ 0 & 0 & \frac{1}{\sqrt{\lambda_3}}\end{bmatrix}, \quad (2)$$

where $\boldsymbol{X}=[X\quad Y\quad Z]^{\mathrm{T}}$ is the mapped vector of samples $\boldsymbol{x}=[x\quad y\quad z]^{\mathrm{T}}$, and $\boldsymbol{v}_i$ and $\lambda_i$ $(i=1,2,3)$ are the eigenvectors and eigenvalues of the covariance matrix, respectively. The distribution of the mapped data with the contours of constant probability is shown in Fig. 7. As can be seen, the contours of constant probability for the mapped data are ellipsoids with horizontal/vertical axes. By multiplying matrix $\boldsymbol{A}$ in Eq. (2), the ellipsoid becomes a sphere, because the elliptical radii length is proportional to the square root of the eignevalues.

### 4.3 Normality tests

Several univariate normality tests are employed to verify the normality of the mapped data. However, relying merely on the plots cannot ensure the normality of the data. Therefore, we employ other quantitative tests to confirm the validity of the normality hypothesis. With the test results, we can determine whether the multivariate normal distribution fitting can be applied on the data. This section explains and applies the tests to the experimental results gathered from four users while walking with the RT Walker. The test results will be presented in Section 7.1.

#### 4.3.1 Graphical methods

Apart from the bivariate scatter plots of the data (Fig. 7), histograms and Q-Q plots are the most commonly adopted graphical approaches for normality testing. Although histograms provide a highly suitable visualization of the distribution, their sole use is insufficient in determining data normality. Therefore, we utilize several statistical tools to ensure the normality of the collected data. Meanwhile, the Q-Q plot compares the quantiles of a sample against the population quantiles of the univariate normal. A quantile denotes the point below which a given fraction (or percentage) of samples lies. If the points are

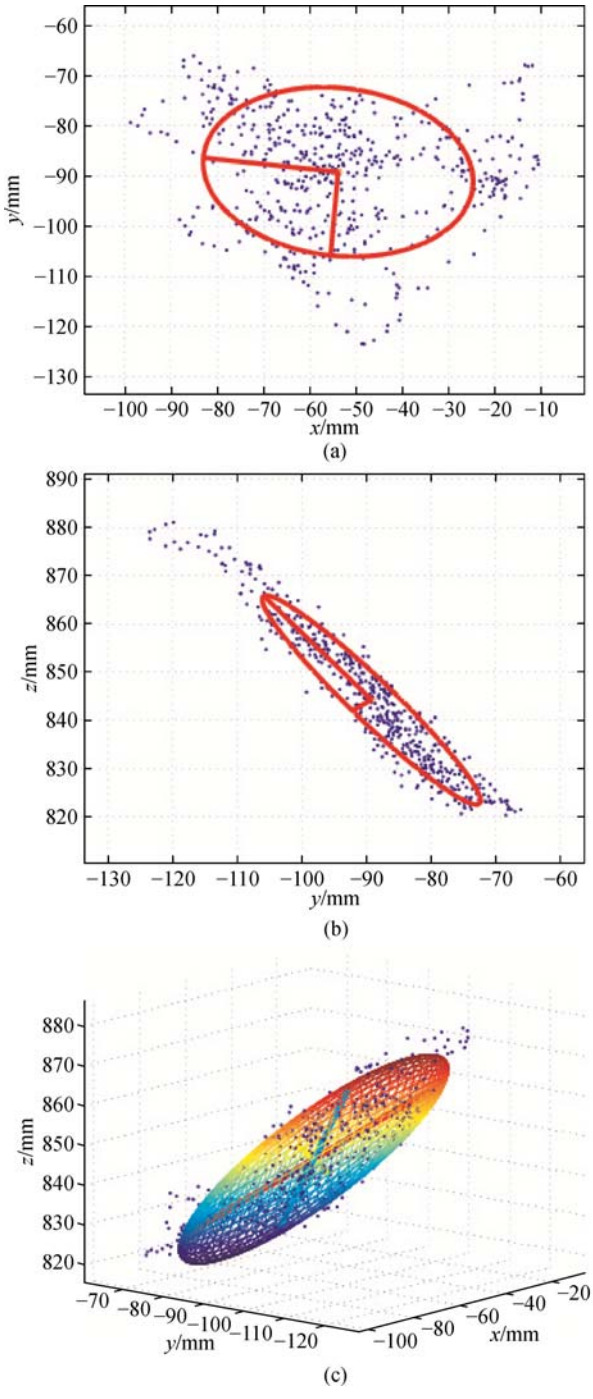


**Fig. 6** Distribution of the data and contours of constant probability for the normal fitted function in the (a) *xy*, (b) *xz*, and (c) *xyz* planes

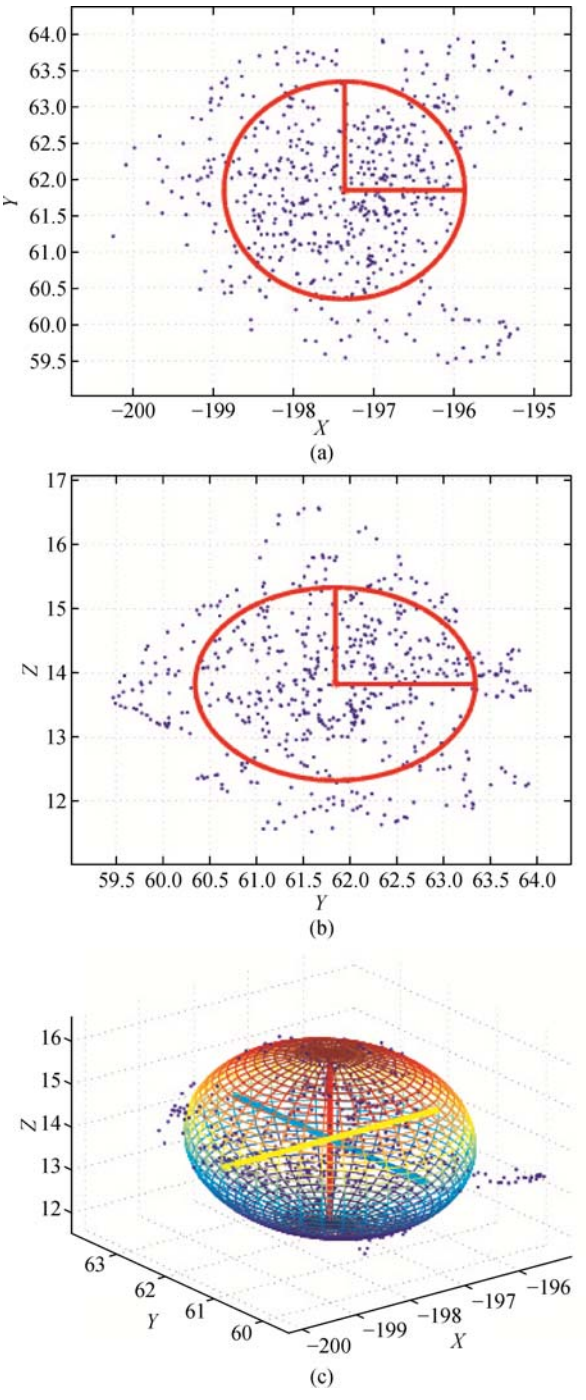


**Fig. 7** Distribution of the mapped data and contours of constant probability for the normal fitted function in the (a) *XY*, (b) *XZ*, and (c) *XYZ* planes

close to a straight line, then no indication of deviation is observed from the norm [30].

#### 4.3.2 Skewness and kurtosis

Other than graphical methods, several quantitative tests can be used to calculate statistical measurements to evaluate the normality of the dataset. These quantitative measurements can be employed to calculate the adaptation of the control system for new users. As this is a largely important issue for such a system, this topic shall be discussed later. Meanwhile, skewness and kurtosis are two

common parameters that quantitatively evaluate the symmetry and flatness of the distribution [30]. They are respectively computed by using the equations

$$\sqrt{b_1} = \sqrt{n}\sum_{i=1}^{n}(x_i-\mu_i)^3[\sum_{i=1}^{n}(x_i-\mu_i)^2]^{-\frac{3}{2}}, \quad (3)$$

$$b_2 = n\sum_{i=1}^{n}(x_i-\mu_i)^4[\sum_{i=1}^{n}(x_i-\mu_i)^2]^{-2}, \quad (4)$$

where $x_i$, $i = 1, 2, ..., n$ are the data samples with a mean value $\mu$. For normal data, the skewness and kurtosis are given by $\sqrt{b_1} = 0$ and $b_2 = 3$, respectively.

### 4.4 State classification

This study assumes that the upper body centroid position follows a normal distribution while someone is walking using a walker. If the normality test results of the mapped independent dataset confirm the validity of this assumption, we can then determine whether the user is in a normal walking situation. This mechanism is performed by fitting a 3D normal distribution function on the data of the upper body centroid during walking. The non-normal case for the user can be categorized in the different states shown in Fig. 3. Falling may occur in many ways, and other possible falling cases may be detected given that the criteria are based on a multidimensional distribution of the upper body centroid and that any other falling scenario would require significant deviations from these distributions.

In this case, the state classification is a one-class type [25], which is based on setting a threshold for the value of $P_{xyz}(x,y,z)$. The value $P_{xyz}(x,y,z)$ indicates the probability that the current centroid position belongs to the fitted 3D normal distribution function on normal walking data. For the classification, we have

$$\begin{cases} P_{xyz} > P_{xyz}^{\mathrm{c}} & \text{walk} \\ \text{otherwise} & \text{fall or sit} \end{cases}, \quad (5)$$

where $P_{xyz}^{\mathrm{c}}$ is the critical threshold that separates normal walking from other states. The threshold is similar for the detection of all fall types; this must be sufficiently low so that it does not hinder the walker during normal walking and sufficiently high so that it does not miss any falling incident. The user is asked to walk normally using the walker to record $i = 1, 2, ..., n$ successive frames. The probability distribution value for such data is expressed as $P_i^{\mathrm{walk}}$. The proposed formula for calculating this value is given by

$$\log P_{xyz}^{\mathrm{c}} = (1+\beta)\min_{i=1}^{n}\log P_i^{\mathrm{walk}} - \frac{\beta}{n}\sum_{i=1}^{n}P_i^{\mathrm{walk}}. \quad (6)$$

The threshold value is adjusted based on each user's walking behavior, and $P_{xyz}^{\mathrm{c}}$ is calculated by using a safe distance from the minimum probability distribution value of normal walking. The distance is adjusted with the coefficient $\beta$, which is set based on the experimental results. More on this will be discussed in Section 7. Meanwhile, the parameter $n$ is the number of frame samples obtained to set the threshold probability $P_{xyz}^{\mathrm{c}}$ of the new user.

## 5 HMM-based state classification

HMMs have been widely employed by researchers in vision-based human action recognition [6]. This approach adopts a time sequence of features (observations) to classify human actions. Falling accidents cannot be accurately referred to as *actions*, because they indicate a user's unintentional behavior when falling. However, each category could consist of similar patterns in the upper body centroid motion by categorizing falling accidents (i.e., fall forward, fall down, and fall right).

As discussed in Section 3, HMMs are utilized in this study to classify the states for two levels. After explaining the mathematical structure, the classification algorithm is described for the one-class and multiclass classification approaches. One-class classification refers to the detection of any non-walking state regardless of its type. However, a classifier that differentiates the non-walking states is required to increase the system accuracy for possibly more flexible motion control (Fig. 3).

The mathematical model applied in the present study is based on that described in Ref. [21]. The observed data at frame $t$, $\boldsymbol{o}(t) \in R^3$, are defined by Eq. (7).

$$\boldsymbol{o}(t) = [\,x(t) \quad y(t) \quad z(t)\,]^{\mathrm{T}}. \quad (7)$$

The time series of the observed data, $O = \{\boldsymbol{o}(t)|\ 1 \leqslant t \leqslant T\}$, comprises the data stored from $T$ successive frames. The model is composed of hidden states $S = \{s_1,s_2,...,s_K\}$ that undergo a transition at every time step with a Markov model. Note that these hidden states are mathematical terms and do not refer to the motion states. The observed dataset, $O$, is related to the hidden states through a parameter set, which can be expressed as $\lambda = (\Pi,D,B)$. Here, parameter $\Pi$ is the probability distribution for the initial state, $D$ is the probability distribution for state transitions, and $B$ is the probability distribution for the observed data. Given that the observed data are assumed to have a mixed Gaussian distribution, then parameter $B = \{b_i(t)|1 \leqslant i \leqslant K, 1 \leqslant t \leqslant T\}$ can be formulated by using the equation

$$b_i(t) = \sum_{j=1}^{M} c_{ij}\aleph\left(\boldsymbol{o}(t),\mu_{ij},\sigma_{ij}\right), \quad (8)$$

where $\mu_{ij} \in R^D$ and $\sigma_{ij} \in R^{D\times D}$ are the components of

Gaussian mean vector and variance matrix for the $j$th mixture component in state $S_i$, respectively, and $D$ indicates the dimension of the observed data.

The HMM training involves determining the parameter set values $\lambda = (\Pi, A, B)$, which are required to maximize the probability function of the observation sequence $P(O|\lambda)$. Obtaining the set of initial values for the HMM parameters is important. The number of hidden states, $K$, is set to be equal to the length of the time series $T$. The training samples are clustered by using the $K$-means algorithm, and the number of clusters is considered to be equal to the number of hidden states, $K$. The parameters for a mixture of Gaussians are calculated with the expectation maximization algorithm. Elements of the transition matrix $\boldsymbol{A}$ and initial probability distribution $\Pi$ are randomly selected to satisfy the conditions of Eq. (9)

$$\sum_{j=1}^{K} a_{ij} = 1, \sum_{i=1}^{K} \pi_i = 1, \tag{9}$$

where $\alpha_{ij}$ and $\pi_i$ are the elements of the transition matrix $\boldsymbol{D}$ and initial probability distribution $\Pi$, respectively. The first condition should be satisfied for all $1 \leqslant i \leqslant K$.

For one-class classification, the normal walking data should be obtained. However, the data for all states are required for the next level. The models, which are specified by parameters $\lambda_i$, $1 \leqslant i \leqslant 8$, refer to each of the eight states that are labeled as $C = 1$ (walk), $C = 2$ (sit), $C = 3$ (fall forward), $C = 4$ (fall down), $C = 5$ (fall right), $C = 6$ (fall left), $C = 7$ (fall back), and $C = 8$ (stand), respectively. The model parameters are updated with the gathered data and Baum-Welch algorithm [31]. The observation sequence is updated at each frame by replacing the last one with the current frame data and shifting all others by a step back. Given the model parameters, $P(O|\lambda_i)$, which is the probability of the observation sequence, is calculated at each frame by employing the forward algorithm [31]. The state classification can be performed with a high frame rate in this manner, which is required for the present application.

Considering $\lambda_1$ as the model parameters trained with normal walking data for the first level of classification, we have

$$\begin{cases} P(O|\lambda_1) > P^{\mathrm{c}} & C = 1 \quad (\text{ walk }) \\ \text{otherwise} & C = 2,3,...,7 \quad (\text{ fall, sit or stand }) \end{cases}. \tag{10}$$

The threshold $P^{\mathrm{c}}$ is set based on the experimental results similar to the procedure in Eq. (6). Eight models are trained for the next classification level. When the observation sequence is detected to be either falling or sitting by Eq. (10), other model probabilities are also calculated, and the maximum value shows the state type. This scenario can be described by

$$\begin{cases} P(O|\lambda_1) > P^{\mathrm{c}} & C = 1 \quad (\text{ walk }) \\ \text{otherwise} & C = \underset{i=2}{\overset{8}{\operatorname{argmax}}} P(O|\lambda_i) \end{cases}. \tag{11}$$

## 6 Walker's motion control

The results of state classification are utilized to control the motion of the passive walker for fall prevention and sit/stand support. The control algorithm is designed based on the results of the one-class classification approach, where any non-walking state is detected regardless of the type.

As soon as the non-walking motion state is detected, the brakes are applied to stop the walker and prevent the user from falling. Even after applying the brakes, the user can still possibly fall, but the safety level is increased and can be employed as a support by the user to regain stability. Moreover, the control configures the walker to be a sit/stand support for the user in case of standing or sitting.

The control system is similar to that proposed in Ref. [15], and is designed such that the brake force generates a maximum torque when the user's state is shown to be non-walking, based on Eqs. (5) and (10), for more than a certain time period. The brake force $f_{\mathrm{b}}$ to control the passive walker is designed similar to Ref. [15]

$$f_{\mathrm{b}} = \frac{\mathrm{e}^{\alpha} - 1}{\mathrm{e}^{\alpha_{\max}} - 1} f_{\max}, \tag{12}$$

where $\alpha$ is a variable that decreases or increases by $\Delta\alpha$ if the user is in a walking or non-walking states, respectively, and $f_{\max}$ is the maximum brake force generated when $\alpha$ is at its maximum value, $\alpha_{\max}$. The said authors utilize the notation $C$ in Eq. (10) to refer to the states. Changes in $\alpha$ are described as

$$\alpha = \begin{cases} \alpha - \Delta\alpha & (C = 1;\ \alpha > 0) \\ \alpha + \Delta\alpha & (C = 2,3,...,8;\ \alpha < \alpha_{\max}) \\ 0 & (\alpha \leqslant 0) \\ \alpha_{\max} & (\alpha \geqslant \alpha_{\max}) \end{cases}. \tag{13}$$

Based on Eq. (13), the brake force $f_{\mathrm{b}}$ increases or decreases between 0 and $f_{\max}$ in an exponential manner.

## 7 Experimental results

The classification and motion control algorithms are tested with four subjects, all of whom are healthy and have no disabilities. One of the subjects is an experienced physiotherapist who can skillfully imitate different walking abnormalities and falling accidents. The subjects' characteristics are listed in Table 1, in which the physiotherapist is labeled as User C.

The subjects were first asked to walk with the walker for several minutes to update the probabilistic model parameters according to each user's walking data. As discussed below, approximately 40 s of walking is determined to be sufficient for the update, and this is referred to as system adaptation.

Regarding the falling experiments, each user was asked to start from a sitting position, stand up, walk for a while, pretend to fall, and then regain his/her stability with the help of the walker. Each user was then asked to repeat each falling scenario four times. This process is shown in Fig. 8.

**Table 1** Characteristics of the experiment subjects.

| User | Height/cm | Weight/kg | Age |
|---|---|---|---|
| A | 185 | 74 | 28 |
| B | 164 | 52 | 27 |
| C | 173 | 85 | 49 |
| D | 183 | 70 | 28 |

### 7.1 Normality test results

The data gathered from the four subjects are mapped (Section 4.2) and tested for normality by using the methods discussed in Section 4.3.

The histograms in Fig. 9 show the dispersion of the upper body centroid positions of two subjects during normal walking. Similar results are obtained for the other two subjects. Results show an acceptable compatibility with Gaussian distribution. Apart from the distribution, the dispersion of the $x$, $y$, and $z$ values (Fig. 6) indicates that the locations of the upper body centroid for all four subjects do not move significantly during walking. For instance, for the case of User A with a height of 185 cm, the centroid moves within a range of approximately 9, 5, and 6 cm in the $x$, $y$, and $z$ directions, respectively. Although these values depend on the user's characteristics and motion patterns, they show a comparatively small range of centroid displacement during walking (i.e., approximately 4% of the user's height within a 210 $cm^3$ space in this case).

The Q-Q plots described in Section 4.3 can also be employed to verify the sample normality. The plots shown in Fig. 10 are Q-Q plots of the mapped $x$, $y$, and $z$ data for the same two subjects, whose histograms are shown in Fig. 9. The plots show a linear relationship between the sample quantiles and those of a standard normal distribution for majority of the domain.

The skewness and kurtosis values of the mapped data for the $X$, $Y$, and $Z$ values from all four subjects are shown in Fig. 11.

The results indicate that the skewness and kurtosis values are close to those for a normally distributed dataset (i.e., 0 for skewness and 3 for kurtosis).

### 7.2 System adaptation for new users

Given that the motion of the upper body centroid during walking varies for different persons, as shown in Fig. 12, the probabilistic model (either normal distribution or HMM) is also different for each user. A new user was asked to walk with the walker for some time during the *data acquisition period*, after which the probabilistic model was updated based on his/her walking pattern. The system control was then prepared for the falling experiments.

For the normal distribution fitting, the adaptation involves determining the mean value and covariance matrix of the samples from the *data acquisition period*. However, identifying how much data would be sufficient for the normal distribution fitting is important. For a new

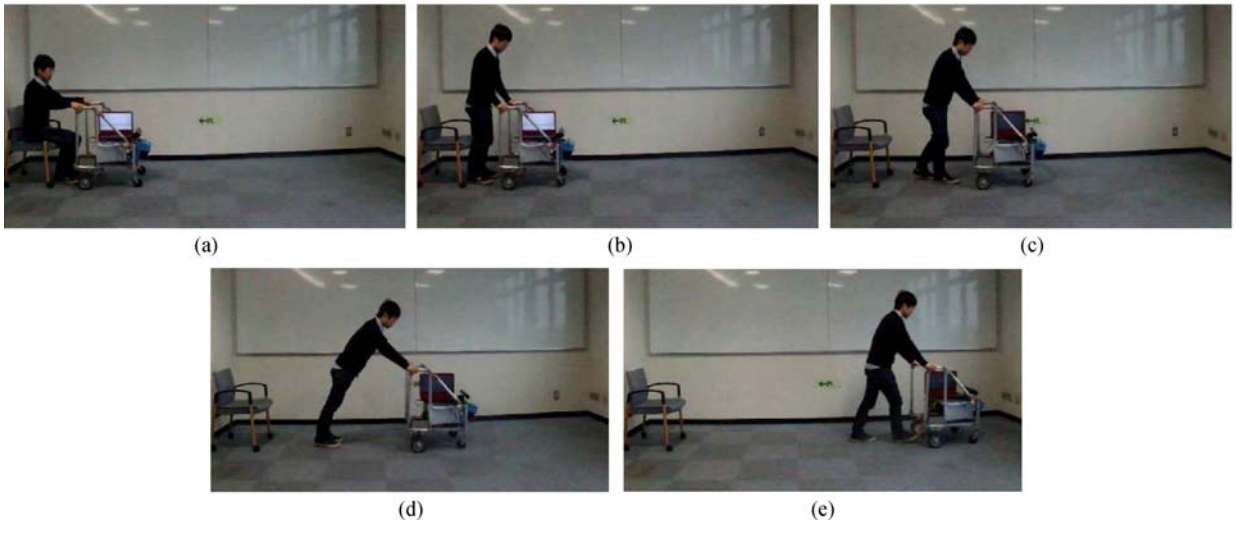


**Fig. 8** Experiments with the RT Walker. (a) 0 s; (b) 6.0 s; (c) 7.8 s; (d) 10.0 s; (e) 14.3 s

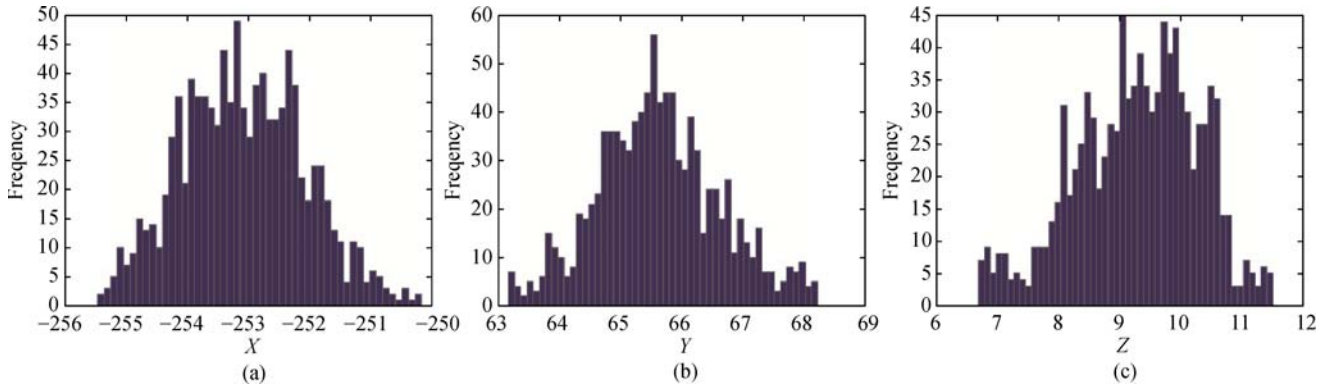


**Fig. 9** Histograms of the mapped data in the *X*, *Y*, and *Z* values for subject B. (a) User B-*X*; (b) User B-*Y*; (c) User B-*Z*

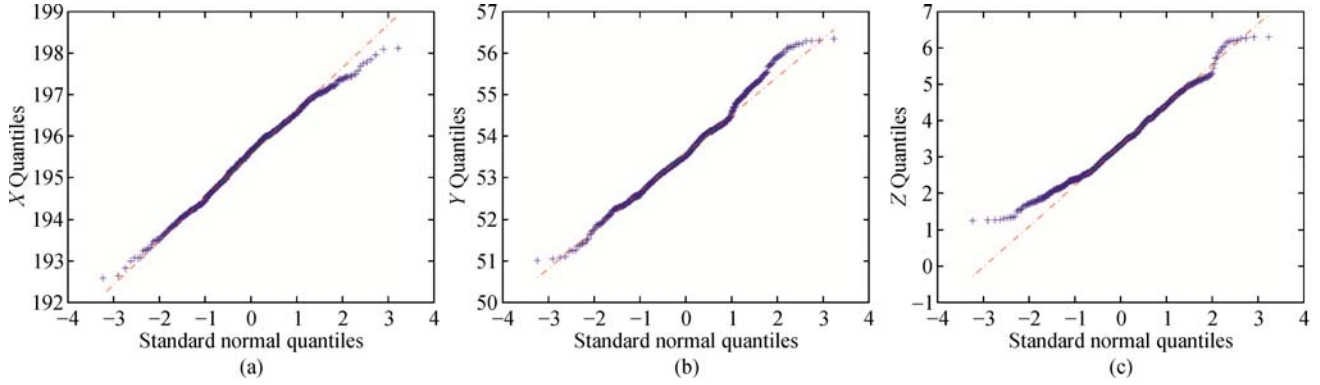


**Fig. 10** Q-Q plots of the mapped data in the *X*, *Y*, and *Z* values for subject D. (a) User D-*X*; (b) User D-*Y*; and (c) User D-*Z*

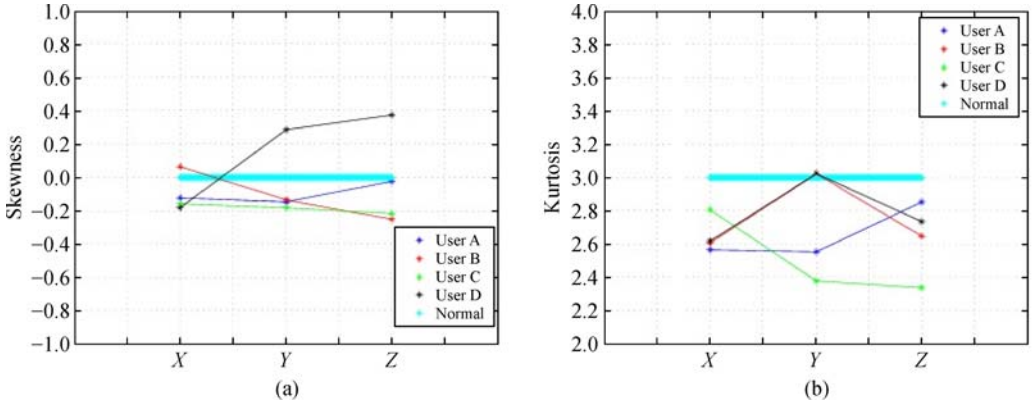


**Fig. 11** (a) Skewness and (b) kurtosis values of the *X*, *Y*, and *Z* data for all four subjects

user, we used the variations in skewness and kurtosis of the mapped version of the obtained data during walking as the criteria to determine when the acquisition period should be stopped. The variations in skewness and kurtosis values for the data from the two subjects are shown in Fig. 13. As can be seen in the figure, and in the results of the other two subjects, the data gathered from 700 frames of normal walking by each subject are sufficient for the normal distribution fitting. Therefore, the system takes approximately 700 frames or 40 s to adapt to the walking pattern of a new user.

In the HMM-based approach, training the HMM for walking can be performed online with the Baum-Welch algorithm [21]. The models for our experiments were trained with an average number of 800 frames of normal walking. Training the models for sitting, standing, and

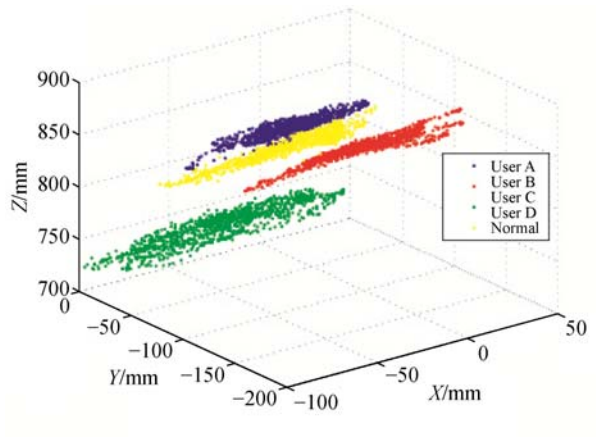


**Fig. 12** Upper body centroid position for all subjects during walking

falling is also conducted by using the Baum-Welch algorithm, after obtaining sufficient training data for each state. For our experiments, the falling-type HMMs were trained with an average of 135 frames.

### 7.3 Normal distribution classification

The experiments for the fall prevention control of the RT Walker by using the normal distribution classification are performed for all four subjects. As previously mentioned, the subjects were asked to walk normally for 2 min to update the distribution and threshold parameters according to their specific walking patterns. For each type of falling accident, the users were asked to resume from a sitting position, stand up, walk with the walker, pretend to fall, and regain their stability with the help of the walker (stand). They were also asked to repeat the falling accident four times.

When any abnormal state other than walking is detected, the brakes are applied according to Eqs. (12) and (13), as well as parameters set at $\alpha_{\max} = 1.5$, $\alpha = 0.2$, $f_{\max} = 150$ N. The results for the five falling types (i.e., fall forward, fall down, fall right, fall left, and fall back) for User C (i.e., the physiotherapist) are shown in Fig. 14.

Significant decreases in the log$P$ values in each experiment can be attributed to sitting, standing, and four successive falling accidents. The braking force increases to stop the walker in these instances. The results from the user's walking data are substituted in Eq. (6) to calculate the threshold for each user. The coefficient $\beta$ has to be the same for all users and is set at $\beta = 3.0$ in the experiments., the algorithm works better for several users than others, but the results generally show that it can detect 96.25% of fall accidents, as seen in Figs. 14 and 15. Approximately 2.5% of false positive falling detections are also noted, where the system activates the brakes even though the user is walking normally.

### 7.4 HMM-based state classification

A similar set of experiments was performed to verify the performance of the HMM-based state classification method. The classification was conducted at two levels. At the first level, any abnormal motion state was detected based on Eq. (10), and the walker was stopped by applying the brake forces (Eq. (12)). At the second level, the type of non-walking state for each motion state was recognized based on the trained HMMs.

The experimental results for the first level of the HMM-based state classification for subject A are shown in Fig. 15. Similar to the case in the previous section, the subjects were asked to start from a sitting position, stand up, start walking, and perform falling actions on four successive occasions. The threshold $P^c$ in Eq. (10) was set by each user's walking data according to Eq. (6), with the coefficient $\beta = 10.0$ for all subjects.

The results from all four subjects show that the method

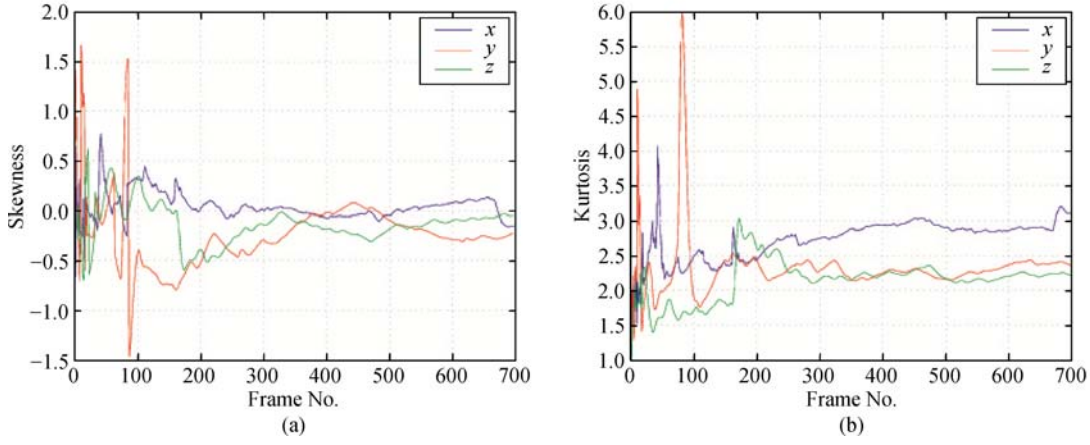


**Fig. 13** Variations in the (a) skewness and (b) kurtosis values during the system adaptation of User C

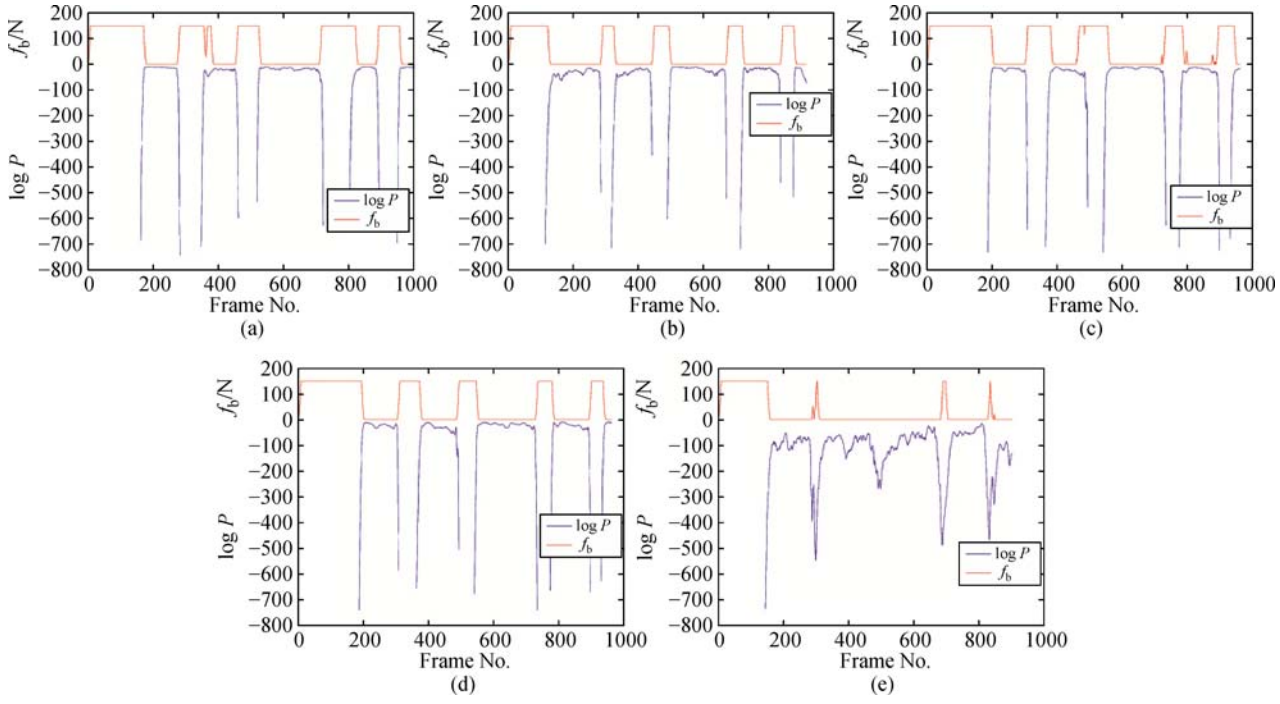


**Fig. 14** Variations in the distribution probability ($\log P_{xyz}$) and brake force $f_b$ for User C. (a) Fall forward; (b) fall down; (c) fall right; (d) fall left; (e) fall back

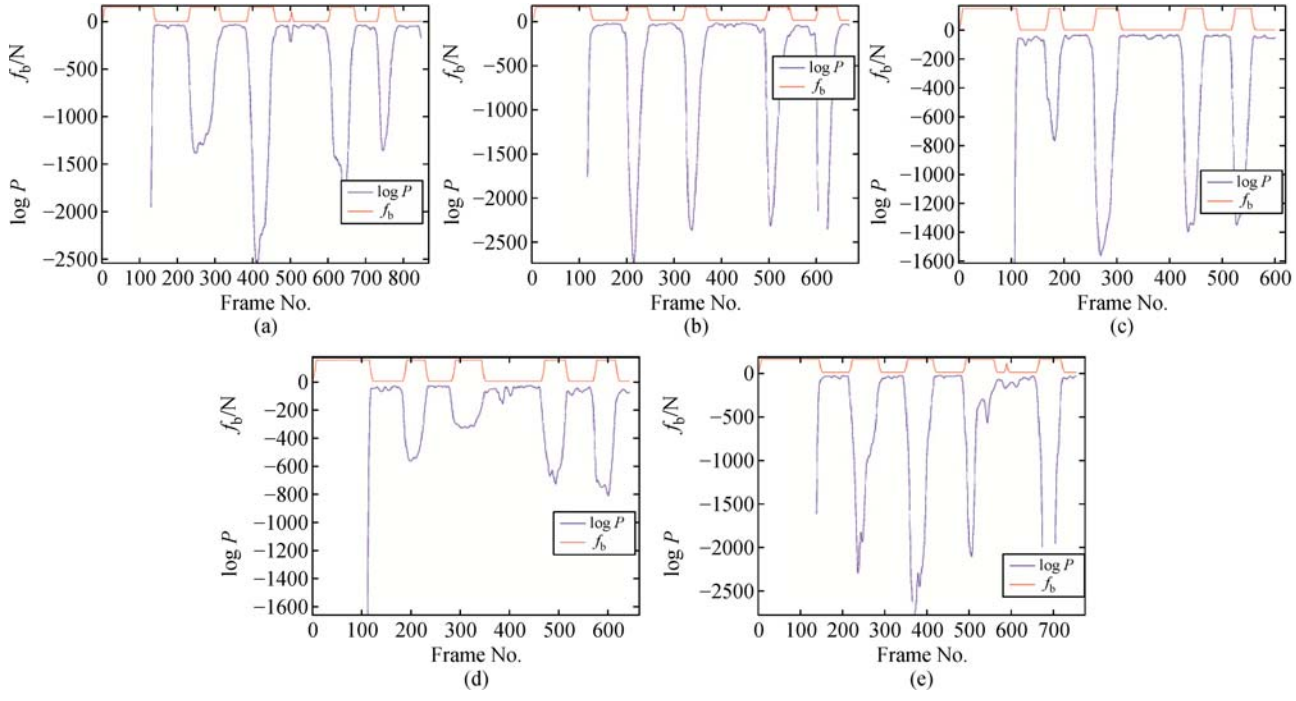


**Fig. 15** Variations in the $\log P(O|\lambda_1)$ and brake force $f_b$ for User A. (a) Fall forward; (b) fall down; (c) fall right; (d) fall left; (e) fall back

detects 98.75% of falling accidents and all of the sitting states. A false positive fall detection rate of 8.75% is also noted.

For the second level of the state classification, eight HMMs were trained for each user by gathering data from each subject. The observation vectors consisted of centroid $x$, $y$, and $z$ values from five successive frames ($T = 5$). Each model had four hidden states, and a Gaussian mixture model of order 3 was fitted on the observation vectors ($K = 4$ and $M = 3$ in Eq. (8)). These values underwent a trail error process to improve the results. The models for the falling states were trained by an average number of 135 frames for each fall type.

As soon as the motion state was detected to be non-walking, the HMM that was most compatible with the observations is associated with the user's state. For each experiment, the sequence of states can be described by the labels discussed in Section 5 as

$$C = \overbrace{2}^{\text{sit}} \rightarrow \overbrace{8}^{\text{siand}} \rightarrow \overbrace{1}^{\text{walk}} \rightarrow \underbrace{\overbrace{(3-7)}^{\text{fall}} \rightarrow \overbrace{8}^{\text{siand}} \rightarrow \overbrace{1}^{\text{walk}}}_{4\ \text{times}}$$

The experimental results of the HMM-based state classification for User B are shown in Fig. 16.

The performance of the recognition algorithm can be evaluated by calculating the confusion matrix of the results, which is calculated based on the number of true and false recognitions for each state. The rows in the matrix refer to true classes, whereas the columns refer to the predicted classes identified by the algorithm. Table 2 shows the confusion matrix based on a dataset with a size of 18733. The row-column number refers to state labels defined in Section 5.

The confusion matrix indicates a total of 81.0% of the correct classifications. The highest rate is for walking (92.8%), whereas the lowest rate is for falling down (34.2%), which is easily mistaken with sitting in 18.2% of the cases.

## 8 Discussion

The normal distribution and HMM-based methods can detect all non-walking states for the four subjects with a high fall detection rate of 96.25% for the normal distribution method and 98.75% for the HMM-based method. The user's normal walking motion should not be interrupted by a high rate of false positive fall detection. Therefore, the normal distribution method shows better results (2.5%) than the HMM-based method (8.75%).

Considering a one-class classification level, the system adaptation for a new user can be performed by approximately 700 frames for the normal distribution method and

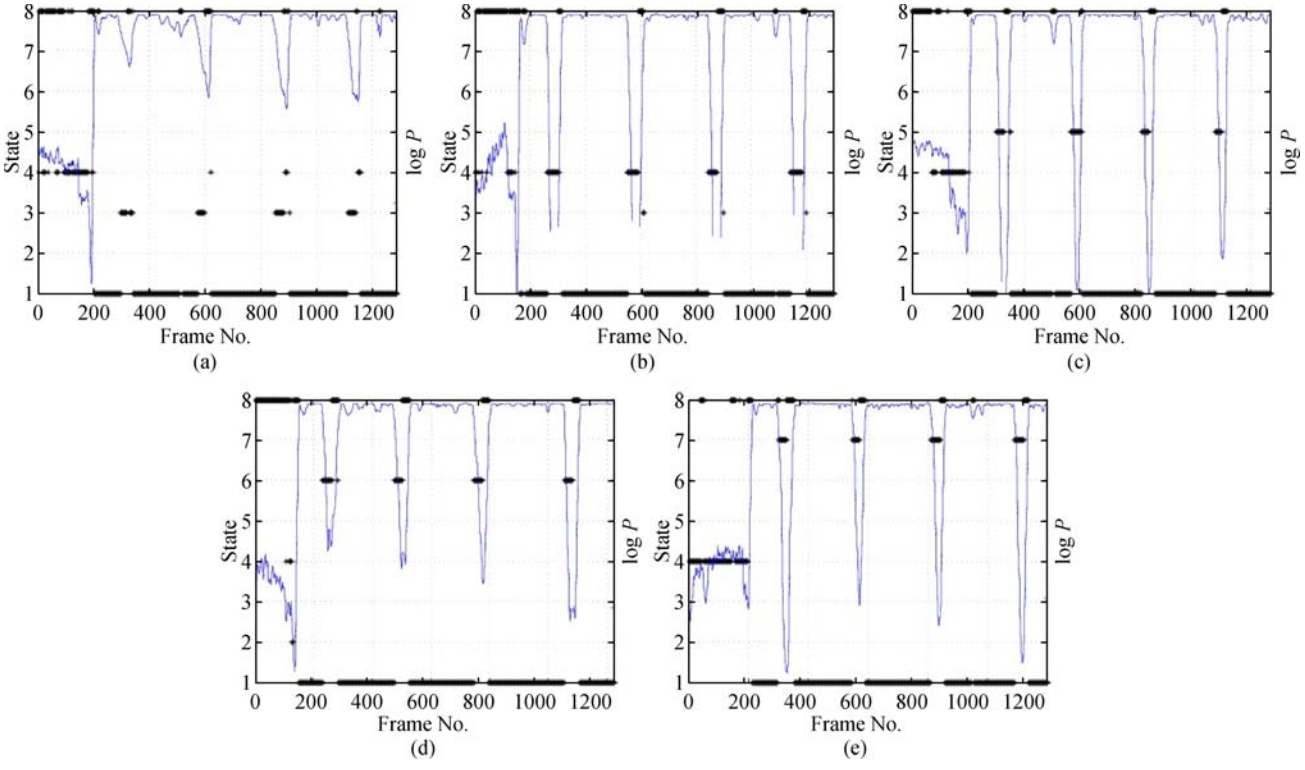


**Fig. 16** Variations in $\log P(O|\lambda_1)$ (blue line) and state recognition results (black *) for User B. (a) Fall forward; (b) fall down; (c) fall right; (d) fall left; (e) fall back

**Table 2** Confusion matrix

| State | Walk | Sit | Forward | Down | Right | Left | Back | Stand |
|---|---|---|---|---|---|---|---|---|
| Walk | 92.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 7.2 |
| Sit | 0.0 | 78.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 21.9 |
| Forward | 3.1 | 1.1 | 70.3 | 0.4 | 0.0 | 0.0 | 0.0 | 25.1 |
| Down | 4.6 | 18.2 | 0.0 | 34.6 | 0.0 | 0.8 | 1.5 | 40.3 |
| Right | 8.4 | 0.0 | 0.0 | 0.0 | 75.1 | 1.1 | 0.0 | 15.4 |
| Left | 22.9 | 0.0 | 0.0 | 0.0 | 0.0 | 68.8 | 0.0 | 8.3 |
| Back | 3.3 | 27.4 | 0.0 | 0.0 | 0.0 | 0.0 | 52.9 | 16.4 |
| Stand | 8.6 | 10.3 | 0.3 | 0.5 | 0.4 | 0.9 | 0.5 | 78.5 |

800 frames for the HMM-based method. Both methods can be obtained within approximately 40 s. However, utilizing the normal distribution method is much easier and computationally simpler.

The HMM-based state classification results for the four subjects show 81.0% of the correct state classification in eight categories. The data were obtained from four healthy subjects for each state, including five falling types. However, building such a database for the elderly and disabled users is much more difficult. The experiments included five different types of 80 fall accidents, which were performed by healthy subjects. However, the results confirm the possibility of such classification, and that the considered falling types are differentiable based on of the upper body centroid motion patterns.

## 9 Conclusions

This study proposed an image-based state classification of a user with a walking support system. We categorized the human motion while using a walker into eight different states: Walking, sitting, standing, and five falling types. Utilizing the upper body centroid, which was extracted from depth images, normal distribution and HMM-based methods were proposed to classify the motion states. State classification results were employed to control the motion of a passive-type walker (i.e., RT Walker) by activating its brakes at any detected non-walking state. The methods were explained, and experimental results obtained from the four subjects who used the RT Walker were presented and discussed.

Performing several normality tests on the data from the four subjects confirmed that the position of the human upper body centroid while walking normally with a walker can be fitted by a multivariate normal distribution function. The fitted distribution function was successfully adopted to detect any non-walking state. Furthermore, the algorithm detected 96.25% of the fall accidents. The system can be adapted to a new user's characteristics after walking normally for approximately 40 s.

The upper body centroid motion pattern was also applied to train HMMs for state classification. The trained HMM with normal walking data was utilized to detect any non-walking state, including sitting, standing, and five falling types. The algorithm detected 98.75% of fall accidents in the experiments, which involved four subjects using the RT Walker. Moreover, the users' data were utilized to train eight HMMs for each user and recognize the motion state at each frame. The state recognition results were evaluated by calculating the confusion matrix, which indicated 81.0% of the correct classification for all users.

The proposed state classification methods can be used in control systems for walking support systems. However, we suggest that further investigation should be conducted with data from real walker users, such as the elderly or disabled people.

**Acknowledgements** The experiments were conducted with the help of Dr. Ryushiro Kawazoe, who is an experienced physical therapist and CEO of Kumasuma Inc., Dr Takuro Hatsukari from the Paramount Bed Company, Tokyo 136-8670, Japan, and the members of the System Robotics Laboratory, Tohoku University, Japan.

## References

1. Alami R, Albu-Schaeffer A, Bicchi A, et al. Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. Beijing: IEEE, 2006
2. Bedaf S, Gelderblom G J, De Witte L. Overview and categorization of robots supporting independent living of elderly people: What activities do they support and how far have they developed. Assistive Technology, 2015, 27(2): 88–100
3. WHO. Good Health Adds Life to Years. Global Brief for World Health Day 2012. 2012. Retrieved form http://www.who.int/ageing/publications/whd2012_global_brief/en/
4. Stevens J A, Thomas K, Teh L, et al. Unintentional fall injuries associated with walkers and canes in older adults treated in US emergency departments. Journal of the American Geriatrics Society, 2009, 57(8): 1464–1469
5. Noury N, Fleury A, Rumeau P, et al. Fall detection-principles and methods. In: Proceedings of 29th Annual International Conference

of the IEEE on Engineering in Medicine and Biology Society. Lyon: IEEE, 2007
6. Tong L, Song Q, Ge Y, et al. HMM-based human fall detection and prediction method using tri-axial accelerometer. IEEE Sensors Journal, 2013, 13(5): 1849–1856
7. Huynh Q T, Nguyen U D, Irazabal L B, et al. Optimization of an accelerometer and gyroscope-based fall detection algorithm. Journal of Sensors, 2015, 2015: 452078
8. Bourke A K, Lyons G M. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. Medical Engineering & Physics, 2008, 30(1): 84–90
9. Auvinet E, Multon F, Saint-Arnaud A, et al. Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution. IEEE Transactions on Information Technology in Biomedicine, 2011, 15(2): 290–300
10. Hirata Y, Hara A, Kosuge K. Motion control of passive intelligent walker using servo brakes. IEEE Transactions on Robotics, 2007, 23 (5): 981–990
11. Taghvaei S, Kosuge K. HMM-based state classification of a user with a walking support system using visual PCA features. Advanced Robotics, 2014, 28(4): 219–230
12. Stone E E, Skubic M. Fall detection in homes of older adults using the Microsoft Kinect. IEEE Journal of Biomedical and Health Informatics, 2015, 19(1): 290–301
13. Wang Y, Wu K, Ni L M. WiFall: Device-free fall detection by wireless networks. IEEE Transactions on Mobile Computing, 2016, 16(2): 581–594
14. Daher M, El Najjar M E B, Khalil M. Automatic fall detection system using sensing floors. International Journal of Computing and Information Sciences, 2016, 12(1): 75–82
15. Purwar A, Un Jeong D, Chung W Y. Activity monitoring from real-time triaxial accelerometer data using sensor network. In: Proceedings of International Conference on Control, Automation and Systems. Seoul: IEEE, 2007, 2402–2406
16. Aguilar P A, Boudy J, Istrate D, et al. A dynamic evidential network for fall detection. IEEE Journal of Biomedical and Health Informatics, 2014, 18(4): 1103–1113
17. Shi G, Chan C S, Li W J, et al. Mobile human airbag system for fall protection using MEMS sensors and embedded SVM classifier. IEEE Sensors Journal, 2009, 9(5): 495–503
18. Zhao K, Jia K, Liu P. Fall detection algorithm based on human posture recognition. In: Pan J S, Tsai P W, Huang H C, eds. Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies, Vol 64. Cham: Springer, 2017, 119–126
19. Hsieh C Y, Liu K C, Huang C N, et al. Novel hierarchical fall detection algorithm using a multiphase fall model. Sensors (Basel), 2017, 17(2): 307
20. Perry J T, Kellog S, Vaidya S M, et al. Survey and evaluation of real-time fall detection approaches. In: Proceedings of 6th International Symposium on High-Capacity Optical Networks and Enabling Technologies. Alexandria: IEEE, 2009, 158–164
21. Rougier C, Meunier J, St-Arnaud A, et al. Fall detection from human shape and motion history using video surveillance. In: Proceedings of 21st International Conference on Advanced Information Networking and Applications Workshops. Niagara Falls: IEEE, 2007, 875–880
22. Taghvaei S, Jahanandish M H, Kosuge K. Autoregressive moving average hidden Markov model for vision-based fall prediction—An application for walker robot. Assistive Technology, 2017, 29(1): 19–27
23. Skubic M, Harris B H, Stone E, et al. Testing non-wearable fall detection methods in the homes of older adults. In: IEEE 38th Annual International Conference of Engineering in Medicine and Biology Society. IEEE, 2016, 557–560
24. Hirata Y, Hara A, Kosuge K. Motion control of passive intelligent walker using servo brakes. IEEE Transactions on Robotics, 2007, 23 (5): 981–990
25. Mazhelis O. One-class classifiers: A review and analysis of suitability in the context of mobile-masquerader detection. South African Computer Journal, 2007, 36: 29–48
26. Yamato J, Ohya J, Ishii K. Recognizing human action in time-sequential images using hidden Markov model. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 1992
27. Goswami A, Peshkin M A, Colgate J E. Passive robotics: An exploration of mechanical computation. In: Proceedings of IEEE International Conference on Robotics and Automation. Cincinnati: IEEE, 1990, 279–284
28. Hirata Y, Komatsuda S, Kosuge K. Fall prevention control of passive intelligent walker based on human model. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. Nice: IEEE, 2008, 1222–1228
29. Gonzalez R C, Woods R E. Image Processing. Digital Image Processing. Upper Saddle River: Addison-Wesley Publishing Co., Inc., 1977
30. Rencher A C. Methods of Multivariate Analysis. 2nd ed. New York: John Wiley & Sons, 2003
31. Rabiner, L, Juang B. An introduction to hidden Markov models. IEEE ASSP Magazine, 1986, 3(1): 4–16