



# HHS Public Access

Author manuscript

*Int J Intell Robot Appl.* Author manuscript; available in PMC 2018 February 01.

Published in final edited form as:

*Int J Intell Robot Appl.* 2017 February ; 1(1): 32–42. doi:10.1007/s41315-016-0002-6.

## RGB-D Camera Based Walking Pattern Recognition by Support Vector Machines for a Smart Rollator

He Zhang and Cang Ye

Dept. of Systems Engineering, University of Arkansas at Little Rock, 2801 S. University Ave, Little Rock, AR 72204

### Abstract

This paper presents a walking pattern detection method for a smart rollator. The method detects the rollator user's lower extremities from the depth data of an RGB-D camera. It then segments the 3D point data of the lower extremities into the leg and foot data points, from which a skeletal system with 6 skeletal points and 4 rods is extracted and used to represent a walking gait. A gait feature, comprising the parameters of the gait shape and gait motion, is then constructed to describe a walking state. K-means clustering is employed to cluster all gait features obtained from a number of walking videos into 6 key gait features. Using these key gait features, a walking video sequence is modeled as a Markov chain. The stationary distribution of the Markov chain represents the walking pattern. Three Support Vector Machines (SVMs) are trained for walking pattern detection. Each SVM detects one of the three walking patterns. Experimental results demonstrate that the proposed method has a better performance in detecting walking patterns than seven existing methods.

### 1 Introduction

Walking therapy is a particular physical therapy (or physiotherapy) that assists a motor-impaired patient to recover their walking ability. This treatment requires interaction and cooperation between a therapist and a patient. The patient is offered instructions to perform the physiotherapy exercises in a monitored manner that provides feedbacks to the therapists for evaluating the effectiveness of the exercises and adjusting the therapy parameters. However, due to the lengthy recovery process and the need of travel, one-to-one in-clinic treatment is prohibitively expensive. As a result, the patient is taught in clinic about the therapy exercises and performs the exercises at home. While it is cost-effective and save the patient time in travel, at-home physiotherapy does not provide the therapist with feedback in a timely fashion for evaluation and adjustment of the exercises. Often, a patient uses a rolling walker (aka rollator) [2, 6, 7, 8] as a walking aid and to support the therapy exercises during the recovery process. Our work is therefore to develop a computer vision method for automatic detection of walking patterns and devise a smart rollator system that is able to provide persistent monitor on the user's walking patterns for at-home walking therapy. The system can be used to score a physiotherapy exercise by monitoring the change in the user's walking patterns during the course of recovery.

We define a walking pattern as a sequence of walking postures and speeds. In the course of a walking therapy, a patient undergoes changes in both walking posture and speed. If the

prescribed exercises are effective, the patient's walking gait will change from abnormal to normal and the speed from slow to normal. Otherwise, there will be no noticeable change in the walking pattern. In other words, detection of walking pattern change plays a critical role to the therapist in judging the effectiveness of the at-home walking therapy sessions.

Walking pattern recognition by computer vision involves lower limb detection, gait feature (including gait shape and gait motion parameters) extraction, walking pattern representation (as a sequence of gait features), machine learning for pattern detection.

In the literature, force and moment sensors [6] have been used on a smart rollator to estimate the step count, pace and stride time of the rollator user. However, these sensors cannot measure the walking posture. In [7], a video camera is mounted on the front bar of a rollator to monitor the lower limb behavior of the user for balance control. The system measures the displacements and velocities of the feet and it requires the user to wear markers on the shoes for foot detection. Recently, RGB-D camera has been employed to measure a person's walking postures and speeds [1, 2]. An RGB-D camera provides reliable depth data for lower limb detection. Gritti, *et al.* [1] propose a histogram based lower limb detection algorithm that extracts a person's feet and legs from an RGB-D camera's depth data and tracks the feet and legs over time. However, it does not measure the walking postures. Joly, *et al.* [2] propose a model-fitting method to detect bare legs and bare feet from the depth data of a Kinect sensor. The method models a bare leg as a cylinder and a bare foot as a plane and fit the parametric models to the Kinect data to detect the leg and foot. Although a skeleton representation of the lower limb can be created from the detected leg and foot, the work in [2] mainly focuses on determining foot orientation and ankle angle. However, the cylinder model fitting approach cannot be used in our case where the human legs are covered by the deformable pant. It is not possible to use any parametric model to describe the motion-induced deformation of the pant. In this paper, we propose a new method to determine the leg skeleton by least square plane fitting. Based on the skeletons of the lower limbs, we introduce a new gait feature to describe the gait shape and the gait motion and use it for walking pattern recognition. The gait feature representation resembles the action feature, consisting of shape and motion parameters of a full skeletal system of human body, that has been used in [4] for human action recognition.

Existing methods [4, 5] for human action recognition may be applied to walking pattern recognition. In [4], an image-to-image difference of the action features between a test video and a class—a video representing a particular class of action—is computed and the sum of the differences for all image frames is used for action detection. The sum does not take into account the transitions between action states. The method in [5] allows comparison of one image frame against multiple image frames. However, transitions between action states are not considered. In [18], we propose method to recognize walking pattern for a smart rollator by analyzing the point cloud data stream of an RGB-D camera. This paper is an extended version of [18]. The proposed walking pattern detection method uses a Markov chain model to capture the characteristics of a gait feature sequence. A gait feature consists of the shape and motion parameters of the walking gait. The transition matrix of the Markov chain model records both the state and state transition information of a walking sequence. If a walking sequence has a fixed pattern, the transition matrix should converge and thus the stationary distribution represents the walking pattern. To automatically identify the walking pattern, we

used Support Vector Machine (SVM) [3] because it has been proved efficient in recognizing human actions with discriminative feature descriptors.

This paper is organized as follows. Section 2 briefly describes our RGB-D camera based smart rollator system. Section 3 introduces the data processing pipeline of the walking pattern recognition method. Section 4 presents the method for leg and foot extraction and the construction of gait feature. Section 5 first briefly describes three popular human activity detection methods and then introduces the Markov chain modeling of a gait feature sequence and the SVMs for walking pattern recognition. Section 6 presents the experimental results of the proposed method and the comparisons with seven existing methods. The paper is concluded in Section 7.

## 2 Smart Rollator Setup

As depicted in Fig. 1(a), an RGB-D camera (ASUS Xtion PRO LIVE) is installed on a rollator, facing towards the lower-extremity of the user with a tilt-down angle  $\theta = -20^\circ$ . This view angle ensures that the feet and the lower parts of the legs are inside the camera's field of view when the user is walking. The RGB-D camera provides a color video and a depth video with  $640 \times 480$  pixels at 30 fps. Given a depth image, the 3D point cloud of the user's lower body can be obtained in real time. Fig. 1(b) shows the point cloud of the user's legs and feet from a depth image frame. The camera coordinate system  $X_c Y_c Z_c$  and the coordinate system  $X_w Y_w Z_w$  that is used to analyze the lower extremity motion are depicted in Fig. 1(a).  $X_w Y_w Z_w$  is obtained by rotating  $X_c Y_c Z_c$  around  $X_c$  for  $20^\circ$ . Each point  $q_i$  of the point cloud in  $X_c Y_c Z_c$  is transformed into a point  $p_i$  in  $X_w Y_w Z_w$  by

$$p_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & -\cos(\theta) \end{bmatrix} q_i \quad (1)$$

For each data frame, the floor plane (shown as the purple rectangle in Fig. 1(b)) is extracted from the point cloud  $\{p_i\}$  using a RANSAC plane segmentation method [10] and the points belonging to the plane are removed. The rest of the points are then used for foot and leg extraction as described in Section 4.2.

In this paper, a simulated walking therapy case is used for the development and validation of the proposed method. The rollator user imitates the walking patterns of a patient with knee injury during the course of recovery. Both normal walk and abnormal walk will be performed and video data (stream of RGB and depth data) will be captured by the RGB-D camera for training and testing the walking pattern recognition method. A Normal Walk (NW) is one with a sequence of normal walking gait at a regular speed. An abnormal walk is one with a sequence of normal/abnormal walking gaits at a much slower speed, sometimes near zero (i.e., halt). The abnormal walking gait is a lame walking gait. Abnormal walks include Slow Walk with Halt (SWH), Slow Lame Walk (SLW) in this paper.

### 3 Data Processing Pipeline of the Smart Rollator System

As depicted in Fig. 2, the data processing pipeline of the smart rollator system consists of three main modules: Gait Feature Extraction (GFE), Markov Chain Modeling (MCM), and SVM-based Walking Pattern Recognition (SWPR). The GFE extracts the point cloud of the user's lower-extremity from a frame of a walking video. After locating the data for the foot and leg, the GFE models the lower-extremity as a skeletal system consisting of skeletons and skeletal points. It then computes the position and motion parameters of the left and right skeletal systems' skeletons and skeletal points. Using these parameters, the GFE constructs a gait feature for the frame. The MCM first clusters the gait features extracted from a number of walking videos into six classes, each of which is a key gait feature. It then describes each video frame by one of the six key gait features. This turns the walking video into a Markov chain whose stationary distribution represents a certain walking pattern. The SWPR maps the stationary distribution to a walking pattern by a trained SVM. The technical details of the three modules will be given in the following sections.

### 4 Gait Feature Extraction

In this paper, a gait feature contains parameters describing gait shape and gait motion. The gait shape parameters encode the current information of the user's lower extremity posture while the gait motion parameters describe how one gait shape evolves into another. The gait shape parameters are the positions of skeletal points of each lower extremity and the gait motion parameters are the velocities of these skeletal points. Collectively, these parameters describe a walking state of the rollator user. The process of gait feature extraction is divided into four steps: lower limb detection, leg and foot segmentation, leg and foot skeletons extraction, and gait feature construction.

#### 4.1 Lower Limb Detection

Considering the case of walking on a flat ground with the rollator, we use the RANSAC plane segmentation method [10] to extract the floor plane from the first frame of the camera's point cloud data. The extracted floor plane is then used to initialize the rollator's coordinate systems as mentioned in Section 2. Data points within the view volume clipped by the chassis of the rollator and the depth limit of the ASUS Xtion PRO LIVE (0.8m~3.5m) are identified, out of which we select the clusters within the first 70 cm above the floor plane as lower limb cluster  $P$  containing feet and legs.

#### 4.2 Leg and Foot Segmentation

A 2-stage processing is employed to find the foot and leg segments from each lower limb cluster  $P$ . In the first stage, the minimum  $y$  coordinate  $y_{min}$  of the lower limb cluster's data points is obtained and the coarse foot and leg segments,  $P'_f$  and  $P'_l$ , are located based on the data points'  $y$ -coordinates. Assuming the  $y$ -span between the toe and the ankle of a human's feet is smaller than 0.2 meter, we locate points within  $[y_{min}, y_{min}+0.2m]$  as the foot segment  $P'_f$  and the rest the leg segment  $P'_l$ . In the second stage, the normal vector of each point in  $P'_f$  is first computed. Then, the normal vector based region growing segmentation algorithm

[11] is implemented to extract the accurate foot segment  $P_f$  from  $P_f'$ . The leg segment is determined by  $P_l = (P - P_f) \cap P_l'$ .

Fig. 3(b) depicts the segmentation result on a frame of depth data of the camera. The points of the leg segment are shown in green while the points of the foot segment are shown in red.

### 4.3 Leg and Foot Skeleton Extraction

Three skeletal points are computed and used to form the leg and foot skeletons. The first two skeletal points are the centroids of the leg and foot segments. And the third point—the ankle point—is determined as the point where the leg intersects the foot-plane. Similar to [2], a Least-Square Plane (LSP) to the data points of the foot segment  $P_f$  is first computed and its normal vector  $\bar{n}$  is used to describe the foot orientation. The LSP is called a foot-plane.

The skeleton of the leg can be extracted from the data points  $p_j$  for  $j = 1, \dots, N$ , of the leg segment  $P_l$  where  $N$  is the total number of data points of  $P_l$ . In an ideal case where the pant leg is pleat-free and the data points are noise-free, the orientation of the leg skeleton, denoted  $\mu$ , is orthogonal with the surface normal of  $p_j$  denoted  $\mathbf{n}_j = (n_{jx}, n_{jy}, n_{jz})$ . If we treat  $\mathbf{n}_j$  as a data point,  $\mu$  is the normal of the LSP to point set  $\mathbf{n}_j$  for  $j = 1, \dots, N$ . The least-square problem is equivalent to find the normal of the LSP to a point set  $q_j = (k_j n_{jx}, k_j n_{jy}, k_j n_{jz})$  for  $j = 1, \dots, N$ , where  $k_j$  is a randomly generated non-zero value for  $\mathbf{n}_j$ . By applying  $k_j$ , we spread the data points in a larger area without changing each point's vector direction. This treatment avoids the case where all data points locate in a narrow area, making the LSP sensitive to noise. For a real-world scenario, using all data points to compute  $\mu$  minimizes the effects of the noise and pant-pleats. The LSP problem is solved by the singular value decomposition method. The centroid of  $P_l$  and  $\mu$  are then used to describe the leg skeleton.

The ankle point is determined as the intersection of the leg skeleton and the foot-plane. A lower limb skeletal system, consisting of 2 skeletons and 3 skeletal points is then formed as shown in Fig. 3b.

### 4.4 Gait Feature Representation

The extraction of the two skeletal systems results in 6 skeletal points  $sp_i$  ( $i = 1, \dots, 6$ ). The skeletal points' positions determine the gait shape. Using the 6 skeletal points' centroid  $sp_c = \{x_c, y_c, z_c\}$  as the reference point, the skeletal points' coordinates are re-computed by  $sp_i' = sp_i - sp_c$ , from which a bounding box  $[x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}]$  is created. Finally, a 18-dimensional vector  $f^s$  representing the gait shape is computed from  $sp_i' = \{x_i', y_i', z_i'\}$  by:

$$f_j^s = \begin{cases} (x_i' - x_{min}) / (x_{max} - x_{min}) & \text{for } j=1 \dots 6, i=1 \dots 6 \\ (y_i' - y_{min}) / (y_{max} - y_{min}) & \text{for } j=7 \dots 12, i=1 \dots 6 \\ (z_i' - z_{min}) / (z_{max} - z_{min}) & \text{for } j=13 \dots 18, i=1 \dots 6 \end{cases} \quad (2)$$

The velocity of each skeletal point is computed from its positional change between two consecutive frames. We denote the velocities of a leg point, ankle point and foot point by  $v_l$ ,  $v_a$  and  $v_f$  respectively and use superscripts 1 and 2 to represent left and right, respectively.

Because the ankle point is indirectly computed from the point cloud (as the intersection between the leg skeleton and foot-plane), its position may incur a larger error than the other two skeletal points. This means that its velocity computed from two consecutive frames is not reliable. Fig. 4 shows the motion parameters computed from the image frames of a 12-second walking video clip. Taking the velocity of the right ankle point  $v_a^2$  (Fig. 4a) for instance, we can observe that  $v_{ax}^2$  goes beyond  $\pm 1$  m/s at some frames. This should not occur because  $|v_{ix}^2| < 0.26$  m/s (Fig. 4c) and  $|v_{fx}^2| < 0.2$  m/s (Fig. 4b). The measurement error in  $|v_{ax}^2|$  was caused by the error in extracting the ankle point. However, we found that the measurement of angle between the foot skeleton and the leg skeleton is more accurate, indicating the angular velocity  $\omega_p$  (Fig. 4d) may be used as a more reliable motion parameter. Therefore, we use the velocities of the foot and the leg centroids, denoted by  $v_f = (v_{fx}, v_{fy}, v_{fz})$  and  $v_l = (v_{lx}, v_{ly}, v_{lz})$ , and the angular velocity  $\omega_p$  to form a 14-dimensional vector  $f^m$ :

$$f^m = [v_f^1, v_l^1, \omega_p^1, v_f^2, v_l^2, \omega_p^2] \quad (3)$$

Figs. 4b–4d depict the gait's motion parameters that are used to form vector  $f^m$ . They were computed from a sequence of image frames of a 12-second walking video clip.

By concatenating (2) and (3), a 32-dimensional feature is constructed as  $[f^s, f^m]$ . In order to rule out correlation between the elements of the feature vector, the principal component analysis method [17] is employed to reduce the feature's dimensionality from 32 to 18. The 18 eigenvalues weight over 95%.

## 5 Walking Pattern Detection

A classification algorithm is needed to detect the user's walking pattern from the extracted gait feature. In [14], a comparative study of four well-known classification techniques, namely Nearest Centroid Classifier (NCC), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and K-nearest Neighbors (KNN), are conducted by using a benchmark dataset—UCI OPPORTUNITY dataset [15] for human action recognition. Walking pattern detection methods based on these methods and three state-of-the-art methods, including Naive-Bayes-Nearest-Neighbor (NBNN) Classifier [4], key pose based Dynamic Time Warping (DTW) [5], and Bag-of-Video-Words (BoVW), will be implemented and compared with the proposed method in this paper. In this section, we first give a brief introduction on NBNN, DTW, and BoVW and then describe in details a new Markov chain based classification method for walking pattern detection. In Section 7, the proposed method's performance will be compared with the other above-mentioned techniques.

### 5.1 Naive-Bayes-Nearest-Neighbor (NBNN)

The idea of using NBNN for human action detection [4] is to use a number of feature collections,  $C = \{C_j\}$ , to describe different types of human actions. Each collection of features,  $C_j$ , represents an action of  $j^{\text{th}}$  type. These features are extracted from video clips

that have been labeled to be the  $j^{\text{th}}$  type actions. Given a  $M$ -frame test video, features,  $p_i$  for  $i = 1, \dots, M$ , are first extracted and the classification result, denoted  $C^*$ , of NBNN is given by:

$$C^* = \underset{C_j}{\operatorname{argmin}} \sum_i^M \|p_i - NN_{C_j}(p_i)\|^2, \quad (4)$$

where  $NN_{C_j}(p_i)$  is the nearest neighbor of  $p_i$  in  $C_j$ .

## 5.2 Dynamic Time Warping (DTW)

In [5], DTW is employed for human action detection. The method models human action as a sequence of key features and identify action through sequence matching by using DTW. In training phase, the training data (video clips) is first processed to extract gait features. Then key features are obtained from these gait features by using K-Means and each video is described by a sequence of key features. In action detection phase, a test video is processed and represent by a sequence of key features  $S$ . The distance between  $S$  and the  $k^{\text{th}}$  key feature sequence  $S_k$  is denoted  $D_k(S_k, S) = \kappa(M_k, N)$ , where  $M_k$  and  $N$  are the sizes of  $S_k$  and  $S$ , respectively.  $\kappa(M_k, N)$  is computed by using a dynamic programming rule as follows:

$$\Delta_k(i, j) = \min \left\{ \begin{array}{l} \Delta_k(i-1, j) \\ \Delta_k(i, j-1) \\ \Delta_k(i-1, j-1) \end{array} \right\} + d_{ij}, \quad (5)$$

where  $d_{ij}$  is the distance between key features  $p_i$  and  $p_j$ .  $S$  is classified as one belonging to the class that contain  $S^*$  with minimum  $D(S^*, S)$ .

## 5.3 Bag-of-Video-Words (BoVW) based SVM

In [12], BoVW technique is used for human action recognition. The method's training phase consists of five steps. First, SIFT features [16] are extracted from all images of the training videos and the extended visual features, each of which includes the SIFT descriptor and the  $x$  and  $y$  coordinates of the key point, are formed. Second, the feature descriptors are clustered by using K-means algorithm and the resulted clusters' centers, call visual words, form the word vocabulary. Third, each video's feature descriptors are mapped to the vocabulary to create a word frequency histogram (the video's signature) to represent the video. Fourth, the value of each bin of the histogram is normalized over all the videos. Fifth, a multi-class SVM is trained by using the normalized histograms. In the testing phase, the training histograms are re-normalized along with the histogram of the test video. The re-normalization is performed in such as way that the resultant normalized test histogram is affected by all the histograms (training ones and test one). Afterward, the normalized test histogram is fed to the SVM for classification.

## 5.4 Markov Chain Modeling of Walking

The gait feature extraction process can produce a large number of gait features from a walking video. Similar to the idea of key feature, they are classified into a few representative

gait features, called Key Gait Features (KGFs), to simplify the representation. In this paper, we use K-means classification algorithm to partition the gait features (extracted from a number of walking videos) into a number of clusters. A KGF is then computed as the centroid of a cluster. After the KGFs are determined, each gait feature of a walking video is represented by one of the KGFs and the sequence of KGFs forms a Markov chain whose stationary distribution is used to detect the rollator user's walking pattern.

**5.4.1 Key Gait Feature**—We employ K-means algorithm to partition the gait features into  $k$  clusters and the centroid of each cluster is computed as a KGF [3, 5]. The value of  $k$  is determined by the Bayesian Information Criterion (BIC) [9]. The BIC is a criterion for selecting a model out of a finite set of models. In our case, we choose  $k$  with the lowest BIC. In addition, if the number of gait features belonging to a cluster is smaller than a threshold  $\tau = 50$ , this cluster is treated as an outlier and thus deleted. Using this scheme, we extract 6 KGFs from a number of walking videos and use them to represent all possible gaits for a walking video.

**5.4.2 Markov Chain Model**—Each gait extracted from a walking video is now represented by a KGF if the norm of the difference between the gait feature and the KGF is below a threshold. By treating a KGF as a state, we denote the gait sequence of a walking video by a Markov chain  $S$ . The transition matrix  $P$  of the Markov chain is of  $6 \times 6$  dimensions. Each entry of the matrix  $p_{ij}$  represents the probability, with which state  $i$  evolves into state  $j$ .  $p_{ij}$  can be obtained from state sequence  $S$  by

$$p_{ij} = \begin{cases} n_{ij}/(n_i - 1) & \text{if } j \text{ is the last state in } S \\ n_{ij}/n_i & \text{otherwise} \end{cases}, \quad (6)$$

where  $n_{ij}$  is the number of transitions from state  $i$  into state  $j$  while  $n_i$  is the number of occurrences of state  $i$  in  $S$ . Therefore,  $P$  can be computed from  $S$ . It is noted that (6)

guarantees  $\sum_{j=1}^N p_{ij}$  for  $i = 1, \dots, 6$ , where  $N$  is the total number of states in  $S$ . The following is a Markov chain sample obtained from a portion of a walking video:

$S$ : 11132323344444465652323344444665623233344

$S$  describes how long a gait is held and what gait it transforms into. For this sequence,  $p_{34}$  can be computed by  $p_{34} = n_{34}/n_3 = 3/11 = 0.273$  and  $p_{46} = n_{46}/(n_4 - 1) = 2/(11 - 1) = 0.2$ . The other entries can be computed in a similar way to obtain the transition matrix  $P$ .

Assuming that a walking video has a fixed pattern, the transition matrix  $P$  of the Markov chain should converge with a sufficiently large number of video data frames. In this case, the stationary distribution  $\pi$  of the Markov chain holds the inherent property of the walking pattern.  $\pi$ , a row vector whose entries are nonnegative and sum to 1, is defined by:

$$\pi P = \pi \quad (7)$$

It can be seen that  $\pi$  is a left eigenvector of  $P$  with an eigenvalue of 1. Therefore, it can be computed from  $P$ . In this work, we use  $\pi$  to represent the walking pattern of a walking video.

### 5.5 SVM for Walking Pattern Recognition

As indicated earlier, there are three types of walking patterns to be detected. Therefore, a multi-class classifier is required for pattern recognition. In this paper, we use the one-vs-all strategy to train three Support Vector Machines (SVMs) to detect the walking patterns. One SVM will be trained to recognize a particular type of walking patterns by using the relevant training data  $(\pi_i, y_i); i=1 \dots, N$ , where  $N$  is the number of walking videos used for training the SVM while  $y_i$  is the SVM output for  $\pi_i$ ,  $y_i$  is manually labeled. Taking the training of the 3<sup>rd</sup> SVM (for SLW detection) as an example, feature vector  $\pi_i$  is computed for the  $i^{\text{th}}$  video. If the walking pattern of this video is SLW,  $y_i = +1$ ; Otherwise  $y_i = -1$ . The kernel function of the SVM is Gaussian kernel whose sigma is 0.03 and regularization parameter is 0.01.

## 6 Experimental Results

### 6.1 Data collection

Nine human subjects participated in data collection. They were instructed to perform the three types of walks. For each walk, the image and depth data streams were recorded from the Xtion. The video for each walk is 12–17 seconds long, containing 360–500 data frames. 5 video clips were recorded for the experiments performed by each human subject, resulting in 45 video clips, 9 for NW, 9 for SWH and 27 for SLW. Three SLW videos were recorded for each subject limping on his left leg, right leg, and both legs, respectively.

### 6.2 Performance evaluation and comparison

In our experiments, leave-one-out cross validation technique is employed for performance evaluation. The performance of the proposed Markov Stationary Distribution (MSD) based one-vs-all SVM method is compared with that of the NCC, LDA, QDA, KNN and NBNN classifiers [4] as well as the DTW method and BoVW based one-vs-all SVM [3] method. In spite of their simplicity, NCC, LDA, QDA and KNN have been reported in [14] to perform well on the UCI OPPORTUNITY dataset [15] for human action recognition. Therefore, they are implemented and compared with the proposed method in this paper. The average detection accuracy and the F-measure [15] of each method are used for performance evaluation. The F-measure takes into account the precision and recall for each class and can provide a better performance evaluation in terms of accuracy. The precision for the  $i^{\text{th}}$  class is defined as  $\alpha_i = \frac{TP_i}{TP_i + FP_i}$  and recall as  $\beta_i = \frac{TP_i}{TP_i + FN_i}$ , where  $TP_i$ ,  $FP_i$  and  $FN_i$  are the true positive, false positive and false negative numbers for the class, respectively. Considering class imbalance, the F-measure is computed by using the classes' sample proportion,

$$F_1 = \sum_i 2 * \frac{s_i}{S} * \frac{\alpha_i * \beta_i}{\alpha_i + \beta_i}, \quad (8)$$

where  $s_i$  is the number of samples of class  $i$  and  $S$  is the total number of samples.

The experimental results are tabulated in Tables 1–8. The three types of videos to be tested are indicated in bolded letters (with the number of video clips in the parenthesis). The classification result (NW, SWH and SLW) for each type of test videos is shown in the column. Taking the first column of Table 1 as an example, out of the 9 NW test videos, 4 was detected as NW, 1 as SWH, and 4 as SLW. The average accuracy and the F-measure of each method (over all video clips) are computed and tabulated in Table 9. From the Table 9, it is clear that the proposed method outperforms the other methods in both average accuracy (0.87) and F-measure (0.87). In term of the simple performance index—average accuracy, the performances of the other 7 methods are ranked as BoVW based one-vs-all SVM, DTW, NBNN, KNN, QDA, LDA, and NCC. However, if the more accurate performance index—F-measure—is used, they would be ranked as BoVW based one-vs-all SVM, DTW, KNN, LDA, QDA, NBNN, and NCC.

## 7 Conclusion

This paper presents an RGB-D camera based walking pattern detection method for a smart rollator system. The method extracts the user's lower limbs from the camera's depth data to obtain the gait information represented by a skeletal system with six skeletal points and four skeletons. By combining the parameters of the gait shape and gait motion, a gait feature is constructed to describe a walking state. K-means is employed to cluster all gait features extracted from a number of walking videos into six key gait features. Using the key gait features, a walking video sequence is modeled as a Markov chain, of which the stationary distribution represents the walking pattern. Three SVMs are trained and used to detect the three walking patterns. Experimental results validate that the proposed method outperforms seven existing methods in detecting walking patterns.

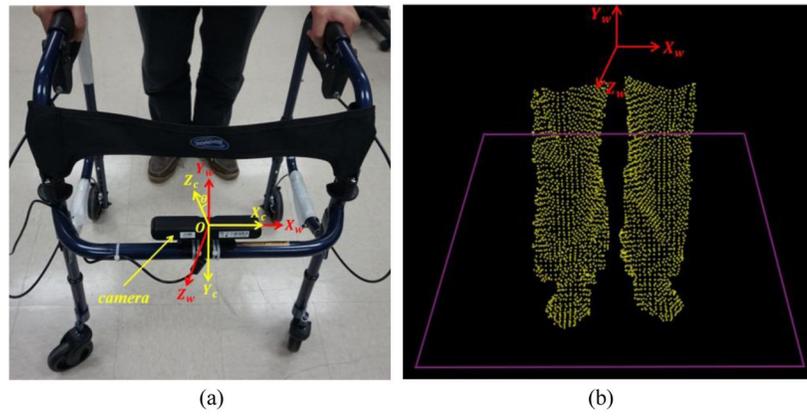
In term of future research, we will use video data collected from real patients' to test the method and compare its performance with that of the other methods. Also, we will define more walking patterns and include them in the proposed method. For real world application, the real-time video stream from the RGB-D camera will be examined by the proposed method segment by segment, each of which contains a fix number of data frames. The user's walking ability will be evaluated based on the accumulative recognition results on the video segments.

## Acknowledgments

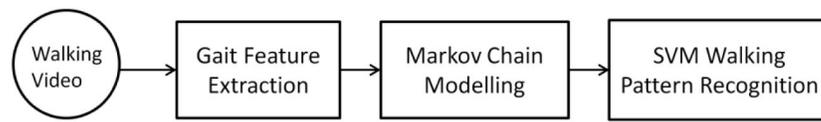
This work was supported by the National Institute of Child Health and Human Development, the National Institute of Nursing Research, and the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health under Award R01NR016151. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

## References

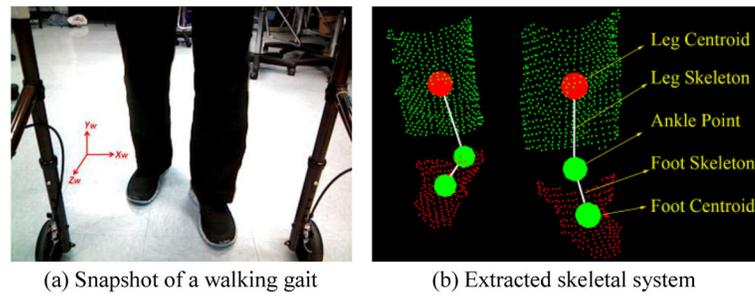
1. Gritti, A., Tarabini, O., Guzzi, J. Kinect-based People Detection and Tracking from Small-footprint Ground Robots. IEEE/RSJ International Conference on Intelligent Robots and Systems; Chicago, IL. 2014.
2. Joly, C., Dune, C. Feet and Legs Tracking Using a Smart Rollator Equipped with a Kinect. IEEE/RSJ International Conference on Intelligent Robots and Systems; Tokyo, Japan. 2013.
3. Laptev I, Caputo B, Schüldt C, et al. Local velocity-adapted motion events for spatio-temporal recognition. Computer Vision and Image Understanding. 2007; 108:207–229.
4. Xiaodong, Y., YingLi, T. Eigenjoints-based Action Recognition Using Naive-Bayes-Nearest-Neighbor. Computer Vision and Pattern Recognition Workshops; Providence, RI. 2012.
5. Chaaaroui AA, Padilla-López JR, Climent-Pérez P, et al. Evolutionary Joint Selection to Improve Human Action Recognition with RGB-D Devices. Expert Systems with Applications. 2014; 41:786–794.
6. Alwan M, Ledoux A, Wasson G, et al. Basic Walker-assisted Gait Characteristics Derived From Forces and Moments Exerted on the Walker's Handles: Results on Normal subjects. Medical Engineering and Physics. 2007; 29:380–389. [PubMed: 16843697]
7. Tung, J. PhD dissertation. University of Toronto; 2010. Development and Evaluation of the iWalker: An Instrumented Rolling Walker to Assess Balance and Mobility in Everyday Activities.
8. Dune, C., Gorce, P., Merlet, JP. Can smart rollators be used for gait monitoring and fall prevention?. IEEE/RSJ International Conference on Intelligent Robots and Systems; 2012.
9. Pelleg, D., Moore, AW. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. International Conference on Machine Learning (ICML); 2000.
10. Qian X, Ye C. NCC-RANSAC: A Fast Plane Extraction Method for 3D Range Data Segmentation. IEEE Transactions on Cybernetics. 2014; 44:2771–2783. [PubMed: 24771605]
11. [http://pointclouds.org/documentation/tutorials/region\\_growing\\_segmentation.php](http://pointclouds.org/documentation/tutorials/region_growing_segmentation.php)
12. Mona MM, Elsayed H, Magda BF, et al. An enhanced method for human action recognition. Journal of Advanced Research. 2015; 6:163–169. [PubMed: 25750750]
13. Csurka, G., Dance, C., Fan, L., et al. Visual categorization with bags of keypoints. ECCV Workshop on Statistical Learning in Computer Vision; 2004.
14. Sagha, H., Digumarti, ST., Millán, JDR., Chavarriaga, R. Benchmarking classification techniques using the Opportunity human activity dataset. IEEE International Conference on Systems, Man, and Cybernetics (SMC); 2011.
15. Ricardo C, Hesam S, Alberto C, et al. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters. 2013; 34:2033–2042.
16. Lowe D. Distinctive Image Features From Scale-Invariant Keypoints. International Journal of Computer Vision. 2004; 60:91–110.
17. Pearson K. On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine. 1901; 2:559–572.
18. Zhang H, Ye C. An RGB-D Camera based Walking Pattern Detection Method for Smart Rollators. Lecture Notes in Computer Science. 2015; 9474:624–633.



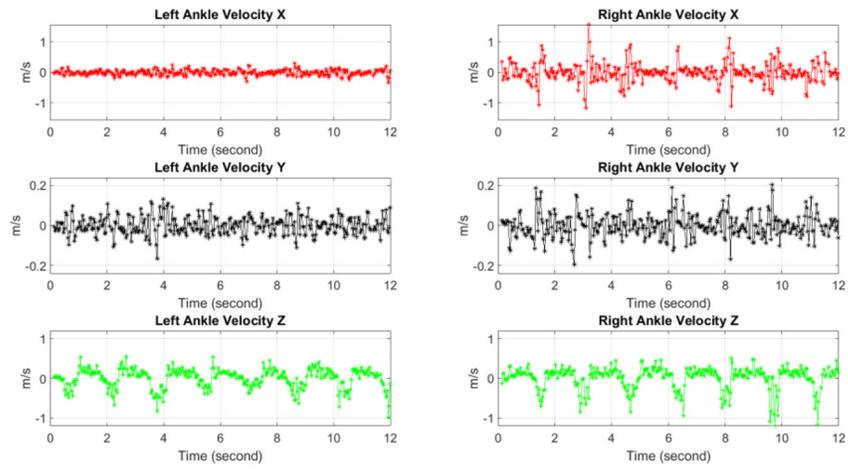
**Fig. 1.** Smart rollator and its coordinate systems: (a) The rollator with the Xtion camera; (b) Floor plane and point cloud of the lower-extremity.



**Fig. 2.**  
Data Processing Pipeline



**Fig. 3.** Skeletal system extracted from the point cloud data of a normal walking gait.



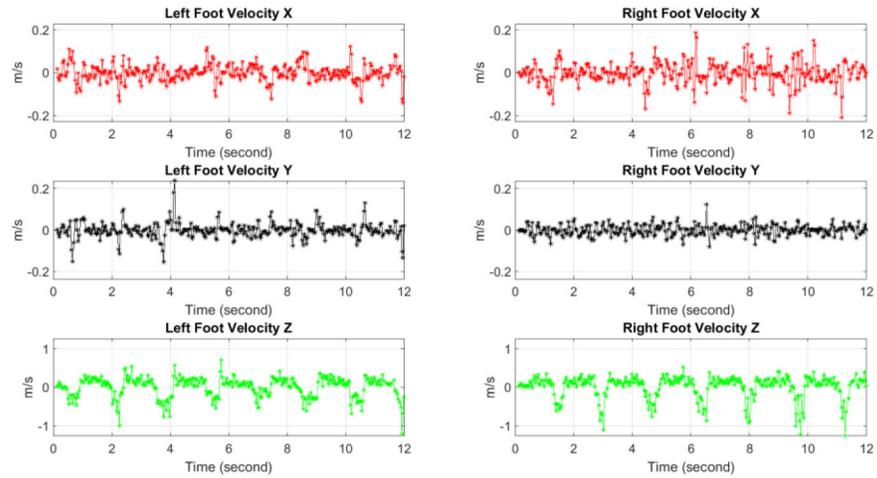
(a) Velocity of the Ankle Point

Author Manuscript

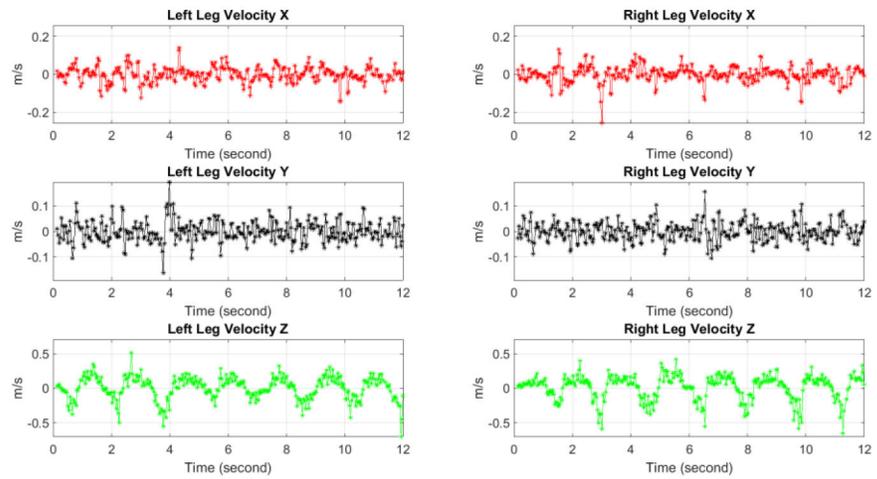
Author Manuscript

Author Manuscript

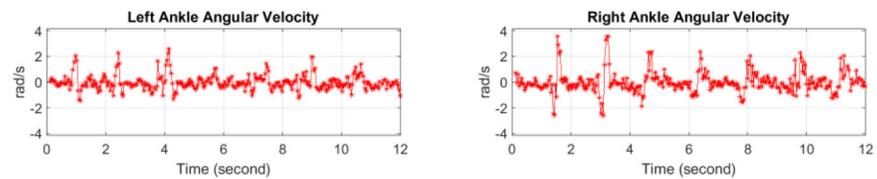
Author Manuscript



(b) Velocity of the Foot Centroid



(c) Velocity of the Leg Centroid



(d) Angular Velocity of the Ankle

**Fig. 4.** Gait motion parameters computed from a 12-second walking video clip.

**Table 1**

Detection result using NCC

	NW(9)	SWH(9)	SLW(27)
NW	4	3	7
SWH	1	0	5
SLW	4	6	15

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 2**

Detection result using KNN

	NW(9)	SWH(9)	SLW(27)
NW	3	5	0
SWH	2	1	4
SLW	4	3	23

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 3**

Detection result using LDA

	NW(9)	SWH(9)	SLW(27)
NW	4	3	1
SWH	3	2	8
SLW	2	4	18

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 4**

Detection result using QDA

	NW(9)	SWH(9)	SLW(27)
NW	2	4	2
SWH	4	0	2
SLW	3	5	23

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 5**

Detection result using NBNN

	NW(9)	SWH(9)	SLW(27)
NW	1	0	0
SWH	1	0	0
SLW	7	9	27

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 6**

Detection result using DTW

	NW(9)	SWH(9)	SLW(27)
NW	2	4	2
SWH	4	2	1
SLW	3	3	24

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 7**

Detection result using BoW

	NW(9)	SWH(9)	SLW(27)
NW	5	1	0
SWH	4	8	2
SLW	0	0	25

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 8**

Detection result using MSD

	NW(9)	SWH(9)	SLW(27)
NW	7	2	2
SWH	2	7	0
SLW	0	0	25

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 9**

Result of the Walking Pattern Recognition

Classifier Performance	NCC	KNN	LDA	QDA	NBNN	DTW	BoYW	MSD
	Accuracy	0.42	0.60	0.53	0.56	0.62	0.62	0.84
F-measure	0.41	0.58	0.55	0.52	0.50	0.60	0.85	<b>0.87</b>