Accurate and Fast Classification of Foot Gestures for Virtual Locomotion

Xinyu Shi † Xiamen University
Junjun Pan † Beihang University
Zeyong Hu Xiamen University
Juncong Lin Xiamen University
Shihui Guo * Xiamen University
Minghong Liao Xiamen University
Ye Pan University College London
Ligang Liu University of Science and Technology of China

Abstract

This work explores the use of foot gestures for locomotion in virtual environments. Foot gestures are represented as the distribution of plantar pressure and detected by three sparsely-located sensors on each insole. The Long Short-Term Memory model is chosen as the classifier to recognize the performer's foot gesture based on the captured signals of pressure information. The trained classifier directly takes the noisy and sparse input of sensor data and identifies seven categories of foot gestures (stand, walk forward/backward, run, jump, slide left and right) without the manual definition of signal features. This classifier is capable of recognizing the foot gestures, even with the existence of large sensor-specific, inter-person and intra-person variations. Results show that an accuracy of ∼80% can be achieved across different users with different shoe sizes and ∼85% for users with the same shoe size. A novel method, Dual-Check Till Consensus, is proposed to reduce the latency of gesture recognition from 2 seconds to 0.5 seconds and increase the accuracy to over 97%. This method offers a promising solution to achieve lower latency and higher accuracy at a minor cost of computation workload. The characteristics of high accuracy and fast classification of our method could lead to wider applications of using foot patterns for human-computer interaction.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction techniques—Gestural input; Human-centered computing—Human computer interaction (HCI)—Interactive systems and tools—User interface programming

1 Introduction

The rapid development of consumer-level devices, such as Oculus and HTC Vive, leads to increasing popularity of Virtual Reality (VR) among the public. The immersion in virtual environments (VEs) offers unique experiences to users and shows its potential in the fields like education, broadcasting, entertainment, etc. The capability of freely navigating in a large VE is a critical function in interactive VR applications. A recent work [40] reviewed the efforts to perform natural walking in VEs. Compared with walking-in-place and joystick-based locomotion, real walking serves as a better mode for virtual locomotion in terms of simplicity, straightforwardness, naturalness [15, 40, 59].

However, it is a challenging task to map the locomotion in the real world (RW) to that in the VE. The first challenge is to capture the locomotion pattern in the real world. Standard motion capture systems, such as Vicon, are expensive and require additional efforts in setting up the external devices (specialized cameras and tracking suits). Although alternative solutions, such as Kinect, Oculus Quest and Vive Focus, allow fast setup and accurate tracking, users are still physically constrained in a limited space and may not be able to use such devices in challenging scenarios such as outdoor environments. The second challenge is to define the mapping from the captured posture in RW to the *referent* action [18] in the VE, particularly, in the case of walking-in-place where there are no direct mapping rules. Researchers have proposed various walking-in-place methods (GUD-WIP [64], LLCM-WIP [17] and SAS-WIP [8]), or hybrid methods (Legomotion [6]), to allow users to explore a large VE by walking in a relatively small RW. However, existing methods generally require manual-tuning of parameters to identify different motion categories and are limited to most-common gestures (walking-stopping). Therefore, it is still important to develop alternative methods to allow intuitive exploration in the VE.

Inspired by the fact that we wear shoes to facilitate long-distance locomotion, this work uses sensors in the insoles to detect the plantar pressure and capture the locomotion patterns in RW (Figure 1). Although foot gestures serve as a promising solution for interaction (see [61] for a comprehensive review in this domain and some recent works [19, 31, 55]), it is significantly difficult to interpret the signal of pressure distributions given a large group of users and a wide range of locomotion patterns. The reasons are threefold. Firstly, there exist **sensor-specific** variations in the manufacturing process, which means two sensors may return different pressure values even though they are pressed with the same force. Secondly, we observe largely **inter-person** and **intra-person** variations, which means such signals of the same pattern may vary significantly for different persons, or even for the same person when making different attempts. Lastly, using an excessive number of sensors may introduce redundant information and increase the manufacture and computing cost, while using a minimal number of sensors may require longer sequences to accurately identify the locomotion pattern. Therefore, developing a stable and fast classification algorithm to handle noisy and sparse data is a non-trivial task.

The goal of this study is to develop novel techniques with sparse distribution of plantar pressure and achieve accurate and fast classification of foot gestures for virtual locomotion. We used consumer-level hardware (for the retail price of 30 USD), with three pressure sensors sparsely embedded in each insole. This classifier should be robust against noises caused by the sensor and individual variations, and capable of identifying the accurate foot patterns within a short time frame. In addition to well-investigated walking/stopping patterns, we also aim to explore a wider range of other gestures for intuitive interaction in VE. To this end, we made the following contributions:

- We develop a pattern classifier, based on the Long Short-Term Memory (LSTM) network, which is generalized across different sensors and individuals. This classifier al-

*Corresponding author: guoshihui@xmu.edu.cn. †These authors contributed equally to this work.

978-1-7281-4765-9/19/$31.00 ©2019 IEEE
DOI 10.1109/ISMAR.2019.000-6

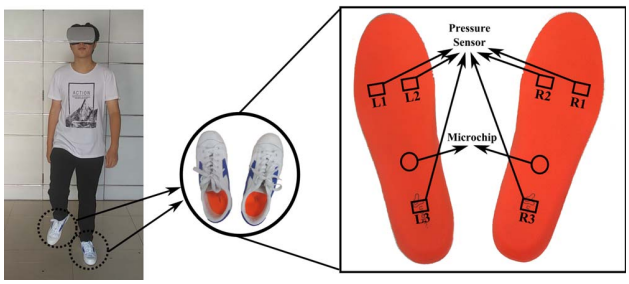


Figure 1: (left) user performing the walking with the device. (middle) shoes with smart insoles. (right) smart insoles with the illustration of the location of pressure sensors and microchip.

lows users to stand, walk forward/backward, run, jump, slide left and right, without the manual definition of signal features. This is, to the best of authors' knowledge, the state-of-the-art numbers ($N = 7$) of foot patterns, considering the large sensor-specific, inter-person and intra-person variations.

- We propose a novel method, Dual-Check Till Consensus (DCTC), to make fast and accurate decisions on gesture recognition. In comparison to the standard LSTM network, DCTC achieves lower latency and higher accuracy at a minor cost of computation workload. The time latency is reduced from 2 seconds of the standard LSTM to 0.5 seconds while the accuracy is improved from 85% to 97.1%.

- We conduct a systematic evaluation of our method, by comparing the existing methods in gesture recognition (the standard LSTM, Hidden Markov Model, Dynamic Time Warping). The results show that our method advances the state-of-the-art performance in terms of both accuracy and time cost.

2 Related Work

2.1 Techniques for Virtual Locomotion

Virtual locomotion refers to the functionality of allowing users to navigate in 3D virtual environments, which critically affects the user's sense of presence. We here focus on existing techniques using body (in particular, foot) movements to trigger virtual locomotion and exclude the discussions on using conventional devices (such as joy-stick, mouse, keyboard or touch screen). So far researchers have proposed the techniques of real walking [15,53,59], walking-in-place [8,17,64,69], or hybrid solutions of real walking and walking-in-place [4,6]. Experiments reveal the subject preferences of virtual locomotion techniques, in the orders of real walking, walking-in-place, flying and using a button-like controller [51,56,59]. Although real walking [59] offers users with high fidelity of presence and immersion, it is limited by the space size of the real world. The technique of redirected walking [15, 53] introduces subtle adjustments to the virtual path by translation, rotation, and curvature gains without affecting the user's perception. These adjustments allow users to walk in a substantially larger virtual space in a relatively smaller physical world.

Walking-in-place enjoys its advantages in allowing users to navigate in a large virtual space within a physically-constrained real world. Advanced omni-directional treadmill has been developed to allow users to walk in all directions, however, the high price of this device indicates that it is currently limited to professional scenarios, rather than consumer-level applications. So far, representative walking-in-place techniques include GUD-WIP [64], LLCM-WIP [17], SAS-WIP [8]), or hybrid methods (Legomotion [6]). In addition to the use of the foot, virtual locomotion has been achieved with non-critical body parts, including tapping [37], hip bending and leaning [20, 23, 47], shaking-head [57], arm swing [38]. Researchers used other body-relevant devices, such as the Wii balance board [14] or human-scale joystick [29]. Researchers also explored the use of hip bending and leaning, for the purpose of ground navigation in scenarios of large display [20] and head-mounted display [47]. Leaning-based motion cueing interfaces mentioned in [22, 63] were also designed for an immersive experience in 3D virtual space. We kindly refer our readers to recent surveys [39, 40] on details of natural walking in virtual reality. Another research direction is to reconstruct 3D human pose [21, 35, 72] and achieve pose estimation [62] using body-worn IMUs. Researchers also proposed a hybrid character control interface mapping the user's pose to the character [36]. The implementation of existing methods for walking-in-place involves manual efforts of tuning the threshold values for recognizing different actions, such as start and stop. This process can be labor-intensive if the number of gestures increases. Our work automatically identifies the signal features embedded in a large database and avoids the manual tuning of model parameters for the purpose of gesture classification.

2.2 Applications of Foot Interactions in Virtual Reality

Foot interactions in VR cover a wider range of channels, in addition to the aforementioned virtual locomotion. Audio and haptics are two common interactions explored in existing studies [41–44, 46]. Researchers developed shoes with pressure sensors, actuators, and markers and provided physically-based audio-haptic feedback to users when they are virtually walking on different surfaces [42]. This study reveals that although auditory and haptic feedback leads to a more realistic experience (reported by participants), this claim is not supported by experimental data and questionnaires. A pair of haptic shoes, named RealWalk [52], adopted MR fluid (Magnetorheological fluid) actuators to simulate four different scenarios: grassland, snow, concrete floor, and dry sand. MR fluid actuators adaptively adjust the viscosity of MR fluid by varying the magnetic field intensity based on the type of materials in virtual ground surfaces and the foot pressure distribution. Researchers also explored the use of foot gestures [28], or in combination with hand [27], as the interaction technique for mobile games. The information of foot pressure captured from a sensor pad is also used to interactively control avatars [71]. Some locomotion devices for Virtual Reality such as Cyberith Virtualizer [10] and Virtuix Omni [5] offer a new form of treadmills. They are based on the low friction principle and leverage the sensor system to detect user's movements. It is a kind of solution for intuitive player motion in VR gaming, allowing users to create interactive animations without the cost or inconveniences of a full-body motion capture system.

Another application of foot patterns in the VR domain is an unobtrusive and immersive mobility training system for stroke rehabilitation called VRInsole [45]. This system utilized the patients' motion information collected from the smart insole and thus provided the input for the VR application to perform corresponding exercise animations. Researchers [27] also designed an immersive football game on the platform of mobile phones and used hand/foot gestures to

interact in this VE. A proof of concept, named as ShoeSole-Sense [30], was developed which enabled location independent hands-free interaction through the feet. This device allows movement control, including moving straight, turning and jumping, in a virtual reality installation. Researchers conducted user experiments to find out the best areas for measuring pressure. Our work differentiates from existing works in that we focus on accurate and fast classification of foot patterns across different individuals, and apply such an interaction technique for exploring the 3D VEs.

2.3 Hardware and Software for Foot Pattern Recognition

In order to capture foot patterns, popular solutions of sensor set-up include embedding the pressure sensors on the insole, attaching an accelerometer for measuring linear acceleration and a gyroscope for measuring orientation and angular velocity. Researchers embedded 48 pressure sensors into the insole to monitor the walking pattern and acquired the information including the walking speed, stride length, and the cadence during the locomotion [68]. FootStriker [13], an EMS-based foot strike assistant, used force-sensitive resistors (FSR) in the insole to detect the running style of users and Electrical Muscle Stimulation (EMS) as a real-time assistant to intuitively aid the runner in adapting a mid- or fore-foot stride pattern. Researchers also utilized other devices, such as Kinect, to track the foot gestures for Desktop 3D interaction [50, 60]. An in-shoe electronics system is developed to monitor temperature, pressure, and humidity for patients with diabetes and peripheral neuropathy [34].

Continuous collection and transmission of the pressure information require a substantial degree of power supply, which greatly limits the wide applications of the smart insole. The straightforward solution is reducing the sampling rate at the expense of data accuracy. More recent research [66] increased the battery life from 2 to 10 hours with an energy-efficient adaptive sensing framework. Their solution is adaptively reducing the sampling density while preserving information fidelity based on the gait cycle analysis.

As the technology of sensor manufacture gradually matures, researchers and entrepreneurs have made remarkable progress in offering commercial products of smart insoles to the public. The choices include Sennopro InsoleX [2], StepRite Insole [49], Brilliant Sole [1] for various purposes of rehabilitation and training. Instead of proposing a new system and duplicating the efforts of hardware design, our work uses an existing commercial product and focuses on developing a capable classifier with high accuracy and low latency. Common algorithms in existing works of foot pattern recognition include peak detection and signal denoising. The peak detection method [12, 33, 58] is widely used to detect the timing of foot strike and liftoff and calculate the time difference between two consecutive steps. Researchers compared the average pressure level between the left and right feet and evaluated the rate of the distortion in walking [12]. The location, velocity, and trajectory of Center-of-Pressure are computed as weighted formulas of the original pressure distributions [26]. A major issue of the existing methods in recognizing foot gesture is the challenge to tackle the variations from different persons and sensors. Our work uses the deep neural network to improve this capability of generalization.

3 Method Overview

The goal of this work is to develop a method to accurately and fast classify foot gestures for virtual locomotion. This section first explains the hardware and software implementation of our framework, followed by explanations on selected gestures in this work. Section 4 presents a novel method, DCTC, which improves the standard LSTM model by reducing the time latency and increasing the classification accuracy. The validation experiment and its findings are presented in Section 5. We compared the performance of our work with state-of-the-art and discussed the limitations of our work as well as our future research directions in Section 6. Section 7 concludes this work and points out its potential in VR applications.

3.1 Hardware and Software Implementation

We use the smart insole from Podoon Technology Ltd. to complete this research. Their product is originally designed for athletes and enthusiasts of running, to record and improve their running gait using the information of plantar pressure distribution. The retail price of the insole is 30 USD, which costs far less than the specialized treadmill platforms for VR interaction or even the consumer-level devices for body tracking such as Kinect. Each insole has three pressure sensors and one onboard processing chip, with their locations indicated in Figure 1.

The onboard chip uses the technology of Bluetooth Low Energy and sends the pressure information to other processing devices (such as a smartphone). Each sensor returns an integer value within the range of $[0, 255]$ to indicate the magnitude of the foot pressure. The sampling frequency of the pressure information is 50Hz and the transmission frequency from the onboard chip is 10Hz. The battery life is 1000 hours of running, as reported by the manufacturer.

The training and testing of the classifier are conducted on a server computer with Intel Core i7 (6 cores), 16G memory and NVIDIA GTX 1080Ti. The server computer receives the sensor data, conducts the prediction and sends the results to the VR helmet. All data are transmitted based on TCP protocol. A unity3D application is developed to render the virtual scene on Pico Goblin, which is an all-in-one VR device with Qualcomm Snapdragon 820 CPU and 3G memory. All source code and dataset of this work can be found in the supplementary file.

3.2 Selection of Foot Gestures

Foot gestures, as a proxy for walking-in-place, have been extensively investigated in existing works [8, 17, 39, 40, 64, 69]. An elicitation study was performed to provide thorough understanding of foot gestures which are linked with the corresponding actions in three conditions: standing up in front of a large display, sitting down in front of a desktop display and standing on a projected surface [18].

Researchers define *Referents* [18] as common actions of an avatar in VE, which our proposed gestures (in RW) are planned to trigger. The selected referents are based on existing works and set as: walking forward/backward, running, jumping, sliding left/right [18, 25].

We follow the mapping from gestures to *referents* as proposed in a most recent work [18] (shown in Figure 2). In the existing work [18], participants are confined on a limited horizontal surface, which implies the use of walking-in-place. Different from this work, we decide to offer two choices to walk forward in the VE. One option is real walking (Figure 2(b)), which ensures the most realistic locomotion experience and is suitable for scenarios with sufficient space. Another option is walking-in-place (Figure 2(a)), which is suitable for real environments with limited space. This hybrid approach offers users flexibility to choose real walking when the perceived immersion is preferred and the size of physical space allows, but

Figure 2: Motion patterns to trigger the *referents* of walking forward, backward, running, jumping, sliding left and right in VE.

walking-in-place when there is no sufficient physical space. It is worth noting that the focus of this work is to develop a pattern classifier to identify each gesture, instead of exploring the best gestures to trigger the corresponding referents. Selections of alternative gestures can be seamlessly integrated into the workflow of our method.

4 Gesture Classifier Training

4.1 Data collection

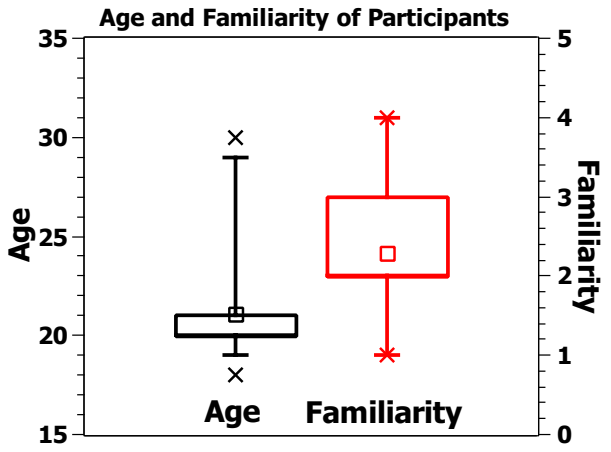


Figure 3: Age and familiarity score of participants in the session of data collection.

Participants Experiments have been carried out with a group of 38 volunteers with an average age of 21.34 and the standard deviation of 3.45. The age distribution is plotted in Figure 3. The participants are students and faculties from our university, ranging from first-year undergraduate to associate professors. The distribution of shoe sizes and male/female is presented in Table 2. Participants involved in the stage of data collection are not recruited for latter validation studies.

Each participant is asked to rate their familiarity of Virtual Reality from 1 to 5 (Table 1). The average score for such familiarity is 2.28, with the SD of 0.82. We believe this group of participants is appropriate for this task, since they have moderate understanding and experience of VR applications, but not influenced by the conventional techniques of VR interaction. All experiments were approved by the Research Ethics Committee Panel at Software School, Xiamen University. Written consent was obtained from each subject after explanation of the experiment.

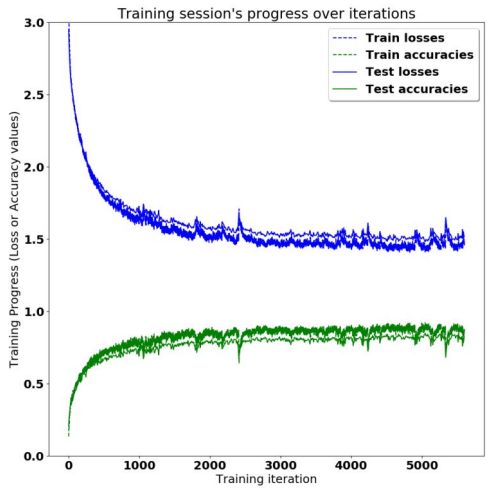
Score	Description of Criteria
1	Never heard of VR concept/application/device
2	Have heard of VR but no hands-on experience
3	Have hands-on VR experience (<= 2 times)
4	Have hands-on VR experience (> 2 times)
5	Experienced user of VR applications, or engaged in related development and research

Table 1: Questionnaire of user familiarity of interactive VR applications.

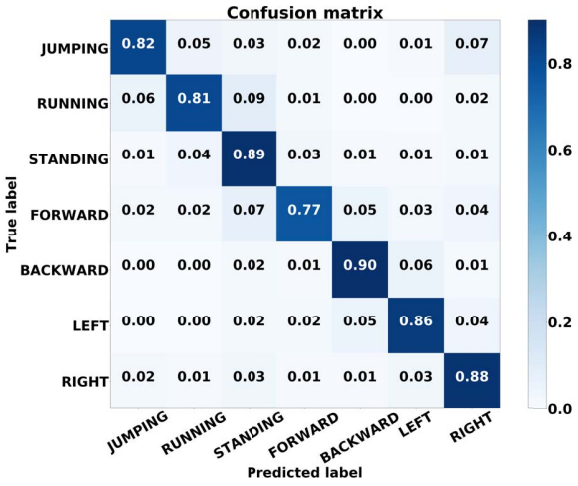
Procedures Each participant performs eight activities (standing, real walking, walking-in-place, walking backward, running, jumping, sliding left and right) wearing the smart insoles and the VR helmet. The helmet displays a virtual environment of an open square which allows users to navigate around. At the beginning of the experiments, participants are informed of the experiment purpose. Participants are presented with a virtual scene which guides them to perform each activity individually. The activities are performed in short duration (10 seconds) with repeated attempts. Participants are instructed to familiarize themselves with the system before the start of data collection. This pre-collection step costs around 2 minutes. The capture process is conducted in a sufficiently large space, which allows the users to perform these activities in a non-stop attempt. A timer is presented to the participant on the VR screen, informing him/her the elapsed time since the start of the current capture session. A participant is free to terminate the capture process at any time if he or she feels tired. Participants are given sufficient rest time during collection sessions. The collected data from each session are automatically uploaded to our server and manually annotated with the corresponding gesture label. The complete collection for each user costs around one hour, including the rest time for participants.

4.2 The standard LSTM Classifier

Data Processing For each participant, we normalize the corresponding data with the maximal sensor-specific pressure value in the collection dataset for individual participants. This technique is designed to neutralize the effect of different body weights of users. Our method works directly with noisy input and no filtering is required for the collected data. After



(a) Training session's progress over iterations



(b) Confusion matrix

Figure 4: Results from training the standard LSTM classifier (sequence size $\mathbf{S}$=100, equivalently 2 seconds). (a) Losses and accuracies for both training and testing dataset. (b) Normalized confusion matrix for the selected behavior patterns.

Shoe Size (US)	Num. of Participants	Num. of Female	Total Duration
6	13	13	699.5
7.5	12	7	714.45
8.5	13	0	780.5

Table 2: Number of participants and length of durations in the process of data collection. The unit of duration is minutes in the collected dataset.

Pattern	Num. of testing samples	Num. of training samples
standing	2724	14580
real walking	2241	11481
walking in place	2709	13656
walking backward	2790	13371
jumping	2712	13863
running	2724	13437
sliding left	2850	15195
sliding right	2847	14487
total	21597	110070

Table 3: Statistics of selected foot patterns in testing and training datasets. All data are provided as the supplementary file of this publication.

that, the time-series data are divided into samples, each of size $\mathbf{N} \times \mathbf{S}$, where $\mathbf{N}$ is the number of sensors and $\mathbf{S}$ is the sequence size of each sensor signal. It is worth noting that the sequence size $\mathbf{S}$ critically affects the prediction accuracy. Multiple sequences sizes, between 10 to 100, are tested in our method (see detailed analysis in Section 6.2).

The complete dataset has been randomly partitioned into two sets on the level of individual participants. For shoe sizes of 6, 7.5, 8.5, 11, 10, 11 participants respectively are selected as the training data and the rest participants as the testing data. The separation between training and testing datasets on the individual level allows us to prove the effectiveness of our classifier in extension to individuals who are not included in the session of data collection. The statistics of the collected database are presented in Table 3.

Training the Standard LSTM Classifier Long Short-Term Memory (LSTM) is an improved sub-category of Recurrent Neural Network and could avoid the problem of vanishing gradient at small computational extra-costs. The LSTM network takes a sample matrix $\mathbf{X}$ of size $\mathbf{N}$ (6 sensors) $\times\mathbf{S}$ (sequence size of each sensor signal), as input, and outputs corresponding inference gesture label vector $\mathbf{Y}$ (7 kinds of gestures). The network model has 3 LSTM layers, each of 64 hidden units, with 1 softmax layer as the output. The network loss function is defined as:

$$\mathcal{L} = ||\mathbf{Y} - \mathbf{Y}_p||^2 \quad (1)$$

where $\mathbf{Y}_p$ is the prediction from the network. We use the Adam Optimizer with the learning rate of 0.0025, the batch size of 1500. Compared with conventional methods in classification of time-series data like Hidden Markov Model (HMM) and Dynamic Time Warping (DTW), we do not require manual feature engineering and avoid the problem of user-specific parameters. Further comparison with existing classification methods can be found in Section 6.2.

Figure 4a shows the loss and accuracy for both training and testing datasets. The learning fast converges to an optimal solution after 1000 iterations and reaches an accuracy of 70%, taking around 20 minutes. As the learning progresses, the final accuracy reaches over 85% for both training and testing

datasets. The result shows that the accuracy of the testing dataset is close to the training dataset, which implies that the problem of over-fitting is avoided and our method is capable of generalizing to variations from individual users.

Figure 4b shows the normalized confusion matrix of different behavior patterns with a sequence size of 100 (2 seconds). The result shows that the action of walking forward (walking-in-place) is recognized with the lowest accuracy of 77%. Walking forward is incorrectly labeled as standing (7%) and walking backward (5%). This error is caused by the similarity of these foot patterns. It is worth noting that when real-walking is adopted for walking forward, the accuracy is improved to 81%. Real walking allows users to lift off their feet for an extended duration of time and thus reduces the possibility of mis-labeling as standing. Meanwhile, the patterns of standing, walking backward and sliding left/right are recognized with rather high accuracy of over 85%. The accuracy is further improved with the novel DCTC method proposed in the following paragraphs.

4.3 Dual-Check Till Consensus

Standard LSTM can classify the motion patterns with an accuracy of ∼80% given a large sequence of data (2 seconds) (Figure 4). This indicates that the algorithm can only correctly identify the pattern of jumping possibly after the jumping is finished. This latency could critically lead to negative user experiences. We propose a novel method, Dual-Check Till Consensus (DCTC) (Figure 5), to fast predict the pattern label while improving the accuracy performance.

Our initial observation is that the forward computation of LSTM model is efficient (less than 1 millisecond). Based on such an observation, we apply an iterative procedure to compare the predictions $\mathbf{Y}$ using the samples of sequence duration of both T and T+δT.

$$T, \mathbf{Y} = \underset{T,\mathbf{Y}}{\arg}(< \mathbf{Y}_p^T > \doteq < \mathbf{Y}_p^{T+\delta T} >) \tag{2}$$

We here define the operator $\doteq$ as the identification of the first element-wise equality in two vectors $< \mathbf{Y}_p^T >, < \mathbf{Y}_p^{T+\delta T} >$.

$$< \mathbf{Y}_p^T > = \cup \mathbf{Y}_p^t,\ t \in [0, T] \tag{3}$$

Therefore, we are searching for a pair of parameters $T, \mathbf{Y}$, which leads to the first-time equality of two predictions $\mathbf{Y}_p^T, \mathbf{Y}_p^{T+\delta T}$. T starts from the smallest segment of 0.1 second and increases to 1 second, while δT is 0.1 second. If the predictions from both samples reach the consensus, the algorithm returns this result; otherwise, T is increased until the consensus is made or the maximum value of T is reached. For the latter case, the pattern with the highest probability in previous predictions is selected.

If we assume the probability distribution of the standard LSTM network as $\mathbf{P}(\mathbf{Y}|\mathbf{X})$, the probability $\mathbf{P}^*$ from our DCTC method is:

$$\mathbf{P}_T^*(\mathbf{Y}|\mathbf{X}_T, \mathbf{X}_{T+\delta T}) = 1 - \prod_{t=0}^{T} (1 - \mathbf{P}(\mathbf{Y}|\mathbf{X}_t)\mathbf{P}(\mathbf{Y}|\mathbf{X}_{t+\delta t})) \tag{4}$$

This indicates that increasing the variable T leads to a higher prediction accuracy, or clamping the accuracy with a lower threshold reduces the timecost for iterative verification.

We train both the standard LSTM and DCTC classifiers for each shoe size, and the general shoe size (Table 4). The results show that the proposed DCTC method increases the accuracy by at least 10%, in comparison to the standard

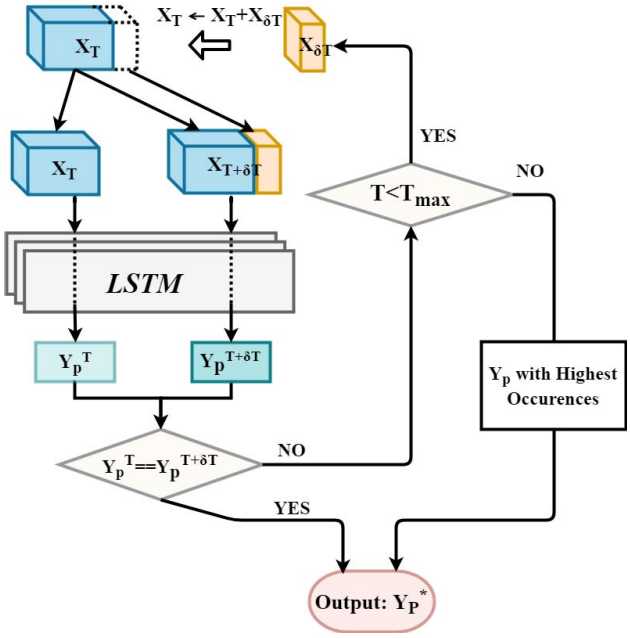


Figure 5: Flowchart of our DCTC algorithm.

LSTM method. To reach an accuracy of 95%, the proposed DCTC method only requires the sequence duration of 0.5 seconds. This is significantly reduced in comparison to the sequence duration of 2 seconds of the standard LSTM method (only achieving an accuracy below 85%). The performance advantage is analyzed in detail, in comparison to the LSTM, HMM, DTW methods (see Section 6.2)).

Additionally, the results (Table 4) show that training the individual classifier achieves better accuracy, over the general classifier. The sensitivity to shoe sizes is potentially caused by male/female distribution (Table 2). The factor of gender has been reported to cause a difference in plantar pressure [48]. We draw a conclusion that using the corresponding classifier for a specific shoe size is a direct solution to raise the accuracy. As everyone is aware of his/her own shoe size, users are prompted to provide their shoe size when they use this application for the first time. The file size of the network model is around 9 megabytes, which is sufficiently small to be downloaded from the remote server.

Shoe Size (US)	LSTM	DCTC
6	0.83	0.97
7.5	0.84	0.94
8.5	0.83	0.96
General	0.78	0.88

Table 4: Accuracy for training the individual classifier for different shoe sizes and a general classifier for all shoe sizes.

5 Validation in Real World

We conduct the validation experiment to evaluate the proposed interaction technique of using foot gestures for virtual locomotion, in particular, focusing on the actual recognition accuracy and the latency of DCTC.

Participants 10 volunteers (5 males and 5 females) with an average age of 21.25 and SD of 3.54 are recruited in this study. The average familiarity of VR (measured with Table 1) is 2.50 and the an SD is 0.85 (Figure 6).

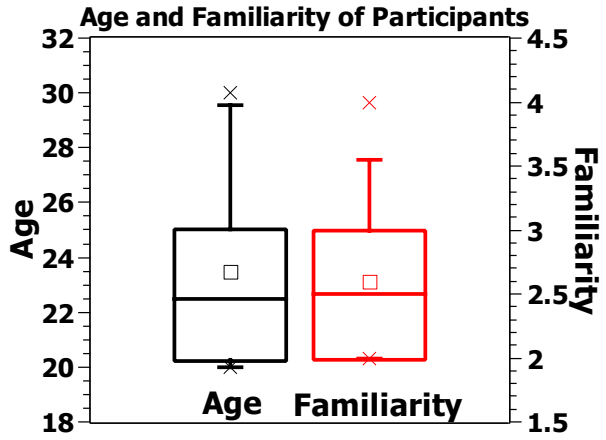
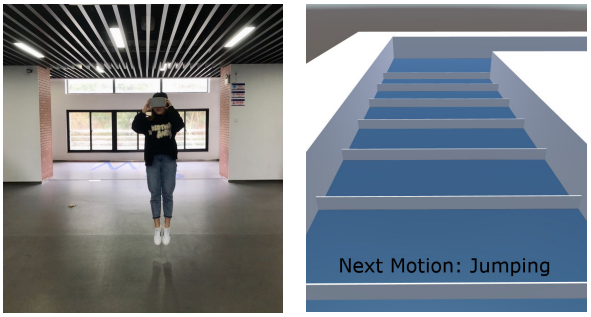


Figure 6: Age and familiarity score of participants in the session of validation.

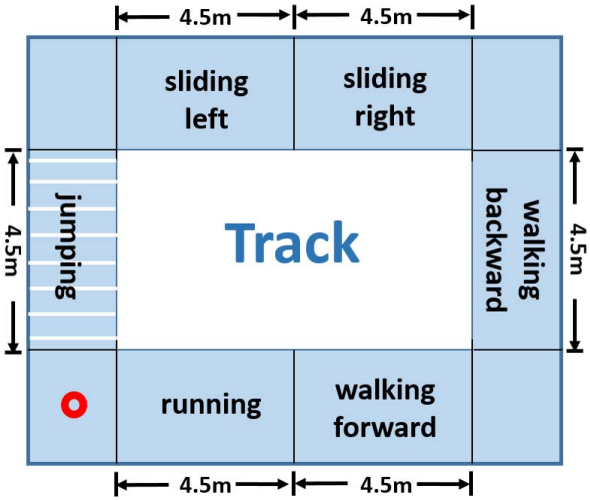
Procedures We develop the application on the VR helmet with the Unity3D framework. The resultant animations are predefined in Unity3D, making sure that validation results are only relevant to the performance of the classification model. A virtual track (Figure 7), with separate segments indicating users to perform the corresponding patterns, is set up to verify our hypothesis that the accuracy and latency of DCTC are acceptable in real-world scenarios. The track area is 86.4×52.8 square meters and the sequence of various action prompts is randomly generated for each individual to eliminate the influence of the order. In particular, 8 hurdles are arranged (on the left side in Figure 7c) so that the participant needs to jump over the hurdles consecutively. For the segment of walking backward, the viewpoint is rotated 180 degrees so the virtual character walks backward and returns to the starting point. The track begins at the red dot in Figure 7c, and terminates at the same position. The room in RW is sufficiently large to allow the participant to walk backwards with no safety concern of object collision. In this scenario, participants are free to choose either real walking or walking-in-place on their own preference to achieve the forward walking.

When running the real-time application, users are guided to perform a few actions (walking-in-place, running-in-place and jumping-in-place) before starting the main application. At this stage, the algorithm estimates the 'maximal' pressure value used for the purpose of weight normalization. These parameters are dynamically updated as a user is engaging in this application. Then, users will enter the track and perform these series of gestures with smart insoles as interaction technique with either LSTM or DCTC classification model without knowing the exact classification algorithm. The same procedure will be repeated with the other model (LSTM or DCTC). All relevant data generated during this process are recorded, including the outputs of the model as well as the surveillance video.

Analysis We collect the mis-labeled actions for each segment showing a mean accuracy of 82% for our DCTC method



(a) Participant In Action (b) Jumping Section



(c) Top View

Figure 7: Interactive demo of navigating in a virtual track.

and 65% for standard LSTM. In more specific terms, the accuracy of DCTC for jumping, running, walking forward and backward, sliding left and right is 83%, 79%, 77%, 85%, 82%, 83%. The results show that gestures with the highest and lowest recognition rates are walking backward and forward respectively. This is consistent with the results shown in the confusion matrix (Figure 4b). We notice that motions which are unclear or vary considerably from the training samples may lead to recognition failure. One example is that users intend to jump, but without their feet fully off the ground. However, this is a common weakness for pattern-recognition models to classify ambiguous and challenging patterns. The gap between the actual recognition rates and the theoretical ones should also be attributed to the 'maximal' parameter used for weight normalization. The only post-processing procedure when preparing the training data-set is to normalize the data with the maximal sensor-specific value for individual participants. When running the real-time application, users are guided to perform a few actions (walking-in-place, running-in-place, and jumping-in-place) before starting the main application. It is challenging to estimate such 'maximal' values within a short time interval, which reduces the recognition accuracy of performed gestures. The problem of body-weight-relevant parameter estimation is also observed in [19]. However, it is worth noting that this problem is part of the data pre-processing of the standard LSTM model, while the major novelty of our work is the iterative approach

to improve the standard one in terms of reducing latency and increasing accuracy.

As for the latency, our DCTC model also distinctly outperforms the standard LSTM. Most participants have complained about the unacceptable delay in the process based on the standard LSTM model, especially in the segment of jumping. Nevertheless, some comments for DCTC are: "I cannot feel the latency for most of the time". Our application runs smoothly at an fps of 30.

6 Discussions

6.1 Comparison with Existing WIP Method

It is worth pointing out that the main purpose of this work differs from existing works in walking-in-place, including LLCM-WIP [17], GUD-WIP [64] and SAS-WIP [8]. These existing methods are proposed to achieve walking-in-place with the amplitude, speed or frequency of foot movements. They differ from our work in the focus of controlling the locomotion speed or achieving the minimal latency between two states (start and stop). The goal of this work is to present a classification strategy which is capable of handling so-far the largest number of locomotion categories while adapting to pattern variations of different participants.

Concerning the latency performance, the metrics of starting and stopping latency of LLCM-WIP [17] which specifically focuses on achieving low-latency interaction are 138 and 96 ms, less than 1/8 of a gait-cycle. In comparison, our method requires more time (0.5 seconds) to detect the transition between 7 categories. Despite the latency comparison, we improve the standard LLCM method in two ways: 1) avoid manual parameter adjustment, 2) address the pattern variations among different individuals. The implementation of LLCM requires manual efforts of parameter adjustment, for example the cutoff frequency of the low-pass filter, as a trade-off between the smoothness of locomotion and low latency. The original work of LLCM focuses on the transition (start, stop) between two locomotion modes, and the challenge of parameter adjustment grows exponentially ($\mathbf{O}(N^2)$) with the number N of locomotion categories. The parameters are also expected to vary across different participants, increasing the complexity of the parameter setting. In contrast, our method avoids the manual settings of parameter values and inherently adapts to the variations across different individuals.

6.2 Accuracy and Timecost Comparison with Existing Gesture Recognition Methods

The recognition algorithm is critical to correctly understand human activity. So far, various methods have been proposed, including Dynamic Time Warping (DTW) [32], Markov Models [9], Conditional Random Fields [7] and Deep Neural Network (DNN) [70], etc. Conventional methods such as DTW require feature selection by manually identifying contributing features during training and thereby reducing computational complexity during classification. Readers may refer to a recent survey [24] for a tutorial on common techniques in feature extraction.

Comparison with HMM Hidden Markov Model (HMM) has been widely used in the field of gesture recognition, achieving over 90% success in KTH or Weizmann datasets [54] [65] [11] [16]. Therefore, we compare with the HMM based approach to model actions using a modified Motion History Images (MHI) for feature extraction proposed by [3]. They report 99% success in Weizmann dataset. They modify the MHI by replacing the linear decay factor with an exponential decay factor emphasizing the recent motion more effectively so that HMM models can achieve a better recognition accuracy. In our implementation, MHI extracts the temporal features by obtaining the difference between the current moment and the previous moment and then computing the gradient direction. The HMM model is defined by λ= (A, B, π) with N number of states (in our problem, N=7). A is the transmission matrix, B comprises of the probability distributions for a feature vector extracted by MHI and π is the initial distribution. λ is trained for each motion categories separately using the observation sequence, the pressure signals from our insoles, with Baum and Welch algorithm. The Viterbi algorithm calculates the probability of each sequence in the test set to tell the final accuracy of this approach. After configuring HMM models with different parameters of hidden states (from 5 to 8) and training iteration (from 100 to 800), we evaluate the results and finally set the hidden states to be 7 and the training iteration to be 500.

The time cost of one prediction is around 3 milliseconds. The results show that the full-size dataset achieves the accuracy around 75% while the datasets with sizes of 6, 7.5, 8.5 achieve the level around 83%, 79%, 81% respectively. The results show that DCTC outperforms HMM in terms of accuracy and capability of coping with noisy and sparse sensor data.

	Ratio of Complete Dataset	Accuracy	Timecost per sequence
DTW	100%	0.87	7585.13
DTW	50%	0.85	4021.45
DTW	25%	0.82	1936.34
DTW	12.5%	0.79	1013.11
DTW	5.0%	0.75	390.00
DTW	1.0%	0.57	77.55
DTW	0.5%	0.44	44.39
LSTM	N/A	0.83	0.46
DCTC	N/A	0.97	13.64

Table 5: Comparison of the accuracy and timecost for methods of DTW&KNN and our method on the dataset of shoe size 8.5. Unit for the timecost is milliseconds and the sequence size is 100 for the standard LSTM and 25 for DCTC.

Comparison with DTW & KNN The combined recipe of Dynamic Time Warping (DTW) and K Nearest Neighbors (KNN) is a representative method in the domain of time-series classification [32]. This method is offline but is capable of achieving high accuracy given a large database. We use this as a benchmark in terms of data size and accuracy. The collected data is first processed by computing a vector of [mean, median, max, min, standard deviation], given a segment of sensor data. DTW aligns two vectors which are originally out of phase, then computes the corresponding distance between these aligned vectors. The label of the test sequence is predicted by finding the closest neighbor (K=1) in the training dataset. Research shows that this method achieves satisfactory accuracy for the task of time series classification [67]. However, this method is computationally too demanding for real-time applications, as in our case. One solution is to reduce the size of the dataset, which the incoming sequence is compared against. We use the technique of numerosity reduction [32, 67] to reduce the dataset and accelerate the computation process. The results show that

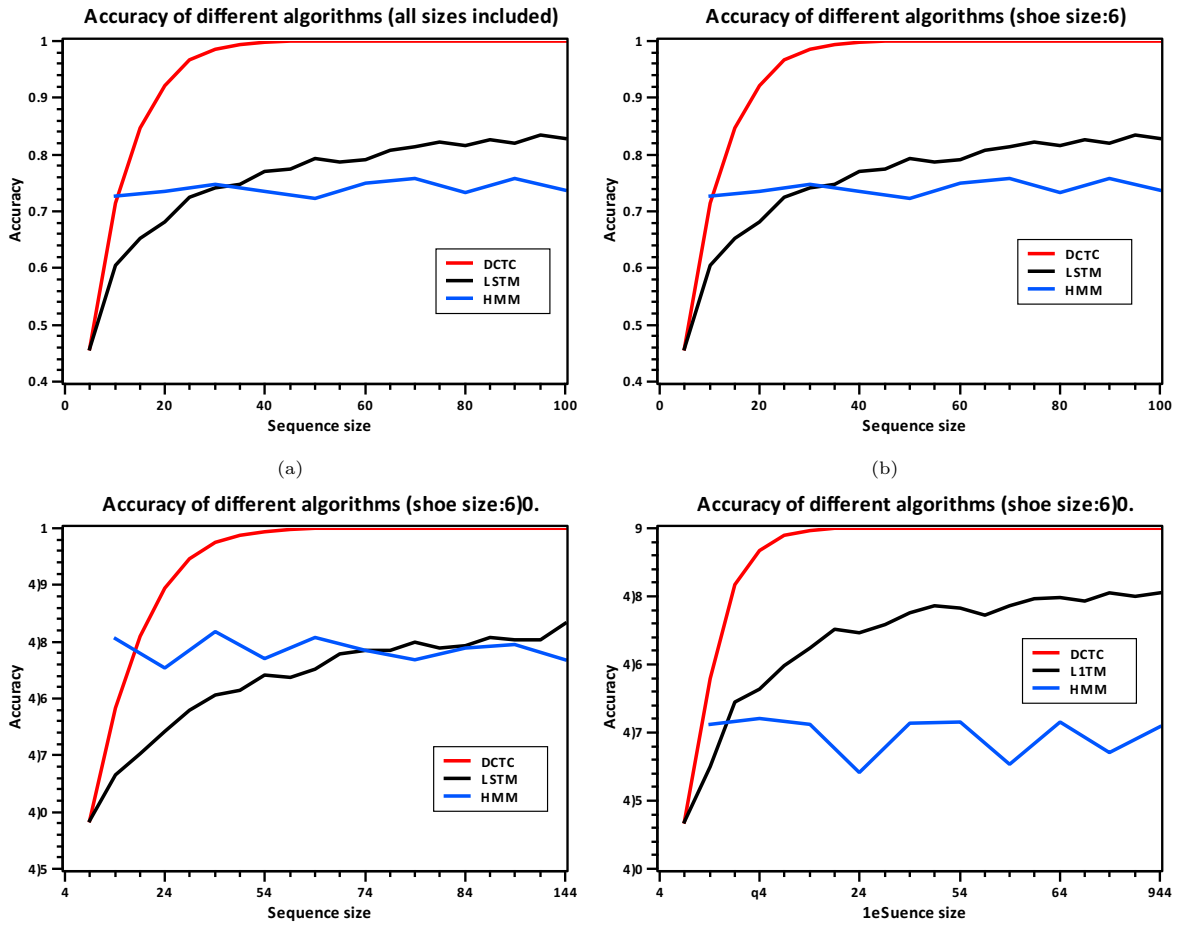


Figure 8: Accuracy comparison of HMM, LSTM and DCTC, with different shoe sizes.

although the full-size dataset achieves the accuracy around 90% (Table 5), each attempt to find the closest neighbor in the dataset costs >7 seconds. When reducing the dataset to speed up the computation, the accuracy drops significantly. Using a larger number of features may potentially increase the accuracy but definitely lead to an explosion of the computing timecost. In comparison, our method achieves an accuracy rate of 83% at the cost of 0.46 milliseconds, in comparison to 82% at the cost of 2 seconds for the method of DTW&KNN when the dataset is maintained at 25%. This shows that the classifier built by our LSTM model captures the patterns embedded in the large dataset, thus can successfully detect the motion pattern without the need to individually compare against the samples in the complete database.

Timecost Comparison Different sequence sizes critically affect the computation load and accuracy of the neural network. For the standard LSTM method, the timecost increases from less than 0.1 to 0.6 millisecond when the sequence size increases from 10 (0.2 seconds) to 100 (2 seconds) (Figure 9). The accuracy rate improves significantly from 50% to over 80% as the sequence size increases from 10 to 100 (Figure 8). This indicates a longer sequence of data signal allows the classifier to better understand the embedded pattern and make the correct recognition. However, it is worth noting that it

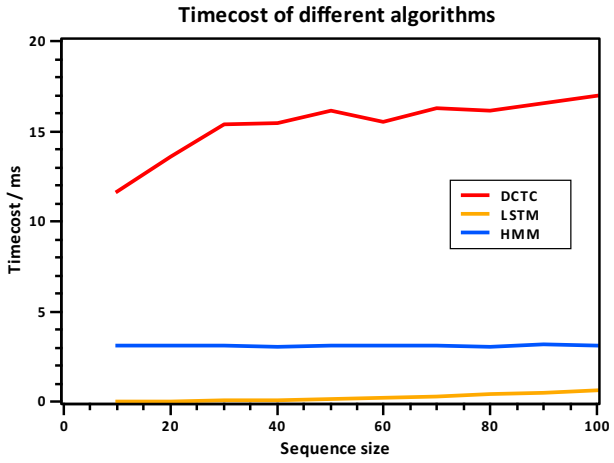


Figure 9: Timecost comparison of HMM, LSTM and DCTC with different sequence sizes.

requires the sequence size to be over 75 (1.5 seconds, times with the sampling frequency of 50 Hz.) in order to reach an accuracy of 80%. Our proposed method, DCTC, improves the standard LSTM model by significantly shortening this time latency and increasing the accuracy rate. Meanwhile, we found out that timecost of HMM is not significantly affected by the sequence size, though its accuracy improves when the sequence size increases.

With our iterative process in DCTC, the timecost increases 10 times as the standard LSTM model. However, the maximal timecost is around 15 milliseconds (Figure 8c), still far less than the time gap between two data transmission (0.1 second). Meanwhile, the accuracy improves significantly for smaller segments of sequences. A sequence size of 15 (0.3 seconds) can achieve an accuracy of 85.3% (Figure 8d), which is a comparable performance to the best accuracy from the standard LSTM model. Furthermore, a sequence size of 25 improves the accuracy to 97.1%. This indicates that for most cases, the algorithm can produce the consensus of the predictions within a time window significantly smaller than the maximum sequence size (100).

6.3 Limitations and Future Work

In this section, we present the limitations of our work and directions for future research. We also discuss the lessons and insights we learned from our experience.

The limitation of our work is rooted in its reliance on building the training dataset. The process of data collection can be accelerated by simultaneously capturing the pressure with our system and gesture with cameras, so the mapping between the pressure and gesture can be built automatically.

We notice that some users hold the headset with their hands (Figure 7a) while some don't (Figure 1 and Figure 2). Recorded videos also reveal that users even dynamically change this holding style (with both hands, just one hand or no hands). We think it is a personal habit since we did not give explicit instructions to users in this regard. This holding style may affect their body balance and thus gesture patterns, but constraining users to a specific style may introduce intervention to user consciousness and affect their sense of immersion. This interesting question requires our future investigation.

For future works, we are developing an intelligent classifier which adapts to a specific user and insoles. Using the current classifier as an initial template, we plan to continuously collect the data flow of the specific user and insoles, and incrementally improve the accuracy of classification. This strategy is promising in completely removing the variations in inter-person patterns and sensor specifics. We expect to achieve a considerably high accuracy of pattern recognition and plan to extend the pattern repertoire to include daily and athletic movements. One fact which is worth noting is that all the subjects in this study are mostly under 30 and in good health. This technique may benefit a broader range of population, including kids, elderly and people with upper body disability. This could lead to a higher impact on these under-represented groups. Another future direction is posture reconstruction based on the information of plantar pressure. We hypothesize that the body posture, in particular, the lower-body posture, critically determines the pressure distribution on the feet, which could be used to reversely infer the current posture. This can be used to track the body movement when the user is wearing the smart insole, and be used in applications including rehabilitation, interaction games etc. However, the technical challenge lies in constructing the mapping from the limited, noisy and sparse pressure information to the high-dimensional body configuration.

7 Conclusion

Our work uses foot patterns as the interaction mode for locomotion in a virtual environment. The proposed method, DCTC, can accurately classify user activities into seven categories: standing, walking forward/backward, running, jumping, sliding left/right. The main contribution of this work is the capability of accurate and fast classification of foot patterns with noisy and sparse inputs. We conducted experiments and showed that using foot patterns can provide intuitive interaction for VR applications.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61702433, 61661146002, 61872020), the Fundamental Research Funds for the Central Universities and Open Project Program of the State Key Lab of CADCG Grant No.A1905 & A1927, Zhejiang University.

References

[1] Virtual reality and video games. `http://www.brilliantsole.com/virtual-reality/`.

[2] Wearable technology platform. `https://smartinsolewearabletechnology.wordpress.com/2016/03/17/vr-game-with-smart-insoleps4/`.

[3] E. C. Alp and H. Y. Keles. Action recognition using mhi based hu moments with hmms. In *IEEE EUROCON 2017-17th International Conference on Smart Technologies*, pp. 212–216. IEEE, 2017.

[4] T. Arnskov, A. Elmholdt, K. Jensen, N. Kristoffersen, J. Litvinas, F. L. Waldhausen, N. C. Nilsson, R. Nordahl, and S. Serafin. A threefold approach for precise and efficient locomotion in virtual environments with varying accessibility. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 507–508. IEEE, 2018.

[5] L. Avila and M. Bailey. Virtual reality for the masses. *IEEE computer graphics and applications*, 34(5):103–104, 2014.

[6] J. Bhandari, S. Tregillus, and E. Folmer. Legomotion: scalable walking-based virtual locomotion. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, p. 18. ACM, 2017.

[7] U. Blanke, B. Schiele, M. Kreil, P. Lukowicz, B. Sick, and T. Gruber. All for one or one for all? combining heterogeneous features for activity spotting. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 18–24. IEEE, 2010.

[8] L. Bruno, J. Pereira, and J. Jorge. A new approach to walking in place. In *IFIP Conference on Human-Computer Interaction*, pp. 370–387. Springer, 2013.

[9] A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster. Robust recognition of reading activity in transit using wearable electrooculography. In *International Conference on Pervasive Computing*, pp. 19–37. Springer, 2008.

[10] T. Cakmak and H. Hager. Cyberith virtualizer: a locomotion device for virtual reality. In *ACM SIGGRAPH 2014 Emerging Technologies*, p. 6. ACM, 2014.

[11] F.-S. Chen, C.-M. Fu, and C.-L. Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and vision computing*, 21(8):745–758, 2003.

[12] H. Cho. Design and implementation of a lightweight smart insole for gait analysis. pp. 792–797, 2017.

[13] F. Daiber, F. Kosmalla, F. Wiehr, and A. Krüger. Footstriker: A wearable ems-based foot strike assistant for running. pp. 421–424, 2017.

[14] G. de Haan, E. J. Griffith, and F. H. Post. Using the wii balance boardâĎć as a low-cost vr interaction device. In

Proceedings of the 2008 ACM symposium on Virtual reality software and technology, pp. 289–290. ACM, 2008.

[15] Z.-C. Dong, X.-M. Fu, C. Zhang, K. Wu, and L. Liu. Smooth assembled mappings for large-scale real walking. *ACM Transactions on Graphics (TOG)*, 36(6):211, 2017.

[16] S. Eickeler, A. Kosmala, and G. Rigoll. Hidden markov model based continuous online gesture recognition. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, vol. 2, pp. 1206–1208. IEEE, 1998.

[17] J. Feasel, M. C. Whitton, and J. D. Wendt. Llcm-wip: Low-latency, continuous-motion walking-in-place. In *3D User Interfaces, 2008. 3DUI 2008. IEEE Symposium on*, pp. 97–104. IEEE, 2008.

[18] Y. Felberbaum and J. Lanir. Better understanding of foot gestures: An elicitation study. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 334. ACM, 2018.

[19] K. Fukahori, D. Sakamoto, and T. Igarashi. Exploring subtle foot plantar-based gestures with sock-placed pressure sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3019–3028. ACM, 2015.

[20] E. Guy, P. Punpongsanon, D. Iwai, K. Sato, and T. Boubekeur. Lazynav: 3d ground navigation with non-critical body parts. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, pp. 43–50. IEEE, 2015.

[21] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll. Deep inertial poser: learning to reconstruct human pose from sparse inertial measurements in real time. In *SIGGRAPH Asia 2018 Technical Papers*, p. 185. ACM, 2018.

[22] A. Kitson, A. M. Hashemian, E. R. Stepanova, E. Kruijff, and B. E. Riecke. Comparing leaning-based motion cueing interfaces for virtual reality locomotion. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 73–82. IEEE, 2017.

[23] E. Langbehn, T. Eichler, S. Ghose, K. von Luck, G. Bruder, and F. Steinicke. Evaluation of an omnidirectional walking-in-place user interface with virtual locomotion speed scaled by forward leaning angle. In *Proceedings of the GI Workshop on Virtual and Augmented Reality (GI VR/AR)*, pp. 149–160, 2015.

[24] O. D. Lara, M. A. Labrador, et al. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013.

[25] C. Lee, S. Ghyme, C. Park, and K. Wohn. The control of avatar motion using hand gesture. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 59–65. ACM, 1998.

[26] F. Lin, A. Wang, Y. Zhuang, M. R. Tomita, and W. Xu. Smart insole: A wearable sensor device for unobtrusive gait monitoring in daily life. *IEEE Transactions on Industrial Informatics*, 12(6):2281–2291, 2016.

[27] Z. Lu, M. S. Lal Khan, and S. Ur Réhman. Hand and foot gesture interaction for handheld devices. In *Proceedings of the 21st ACM international conference on Multimedia*, pp. 621–624. ACM, 2013.

[28] Z. Lv, S. Feng, M. S. L. Khan, S. Ur Réhman, and H. Li. Foot motion sensing: augmented game interface based on foot interaction for smartphone. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pp. 293–296. ACM, 2014.

[29] M. Marchal, J. Pettré, and A. Lécuyer. Joyman: A human-scale joystick for navigating in virtual worlds. In *IEEE Symposium on 3D User Interfaces 2011 (3DUI)*, 2011.

[30] D. J. Matthies, F. Müller, C. Anthes, and D. Kranzlmüller. Shoesolesense: proof of concept for a wearable foot interface for virtual and real environments. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, pp. 93–96. ACM, 2013.

[31] D. J. Matthies, T. Roumen, A. Kuijper, and B. Urban. Capsoles: who is walking on what kind of floor? In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, p. 9. ACM, 2017.

[32] T. Mitsa. *Temporal data mining*. Chapman and Hall/CRC, 2010.

[33] M. Mladenov and M. Mock. A step counter service for java-enabled devices using a built-in accelerometer. pp. 1–5, 2009.

[34] R. E. Morley, E. J. Richter, J. W. Klaesner, K. S. Maluf, and M. J. Mueller. In-shoe multisensory data acquisition system. *IEEE Transactions on Biomedical Engineering*, 48(7):815–820, 2001.

[35] C. Mousas. Full-body locomotion reconstruction of virtual characters using a single inertial measurement unit. *Sensors*, 17(11):2589, 2017.

[36] C. Mousas and C.-N. Anagnostopoulos. Performance-driven hybrid full-body character control for navigation and interaction in virtual environments. *3D Research*, 8(2):18, 2017.

[37] N. C. Nilsson, S. Serafin, M. H. Laursen, K. S. Pedersen, E. Sikstrom, and R. Nordahl. Tapping-in-place: Increasing the naturalness of immersive walking-in-place locomotion through novel gestural input. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 31–38. IEEE, 2013.

[38] N. C. Nilsson, S. Serafin, and R. Nordahl. The perceived naturalness of virtual locomotion methods devoid of explicit leg movements. In *Proceedings of Motion on Games*, pp. 155–164. ACM, 2013.

[39] N. C. Nilsson, S. Serafin, and R. Nordahl. Walking in place through virtual worlds. In *International Conference on Human-Computer Interaction*, pp. 37–48. Springer, 2016.

[40] N. C. Nilsson, S. Serafin, F. Steinicke, and R. Nordahl. Natural walking in virtual reality: A review. *Computers in Entertainment (CIE)*, 16(2):8, 2018.

[41] R. Nordahl, A. Berrezag, S. Dimitrov, L. Turchet, V. Hayward, and S. Serafin. Preliminary experiment combining virtual reality haptic shoes and audio synthesis. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, pp. 123–129. Springer, 2010.

[42] R. Nordahl, S. Serafin, N. C. Nilsson, and L. Turchet. Enhancing realism in virtual environments by simulating the audio-haptic sensation of walking on ground surfaces. In *2012 IEEE Virtual Reality (VR)*.

[43] R. Nordahl, S. Serafin, L. Turchet, and N. C. Nilsson. A multimodal architecture for simulating natural interactive walking in virtual environments. *PsychNology Journal*, 9(3), 2011.

[44] R. Nordahl, L. Turchet, and S. Serafin. Sound synthesis and evaluation of interactive footsteps and environmental sounds rendering for virtual reality applications. *IEEE transactions on visualization and computer graphics*, 17(9):1234–1244, 2011.

[45] H. Oagaz, A. Sable, M.-H. Choi, W. Xu, and F. Lin. Vrinsole: An unobtrusive and immersive mobility training system for stroke rehabilitation. pp. 5–8, 2018.

[46] S. Papetti, F. Fontana, M. Civolani, A. Berrezag, and V. Hayward. Audio-tactile display of ground properties using interactive shoes. In *International Workshop on Haptic and Audio Interaction Design*, pp. 117–128. Springer, 2010.

[47] P. Punpongsanon, E. Guy, D. Iwai, K. Sato, and T. Boubekeur. Extended lazynav: Virtual 3d ground navigation for large displays and head-mounted displays. *IEEE transactions on visualization and computer graphics*, 23(8):1952–1963, 2017.

[48] A. Putti, G. Arnold, and R. Abboud. Foot pressure differences in men and women. *Foot and ankle surgery*, 16(1):21–24, 2010.

[49] T. E. Roden, R. LeGrand, R. Fernandez, J. Brown, J. E. Deaton, and J. Ross. Development of a smart insole tracking system for physical therapy and athletics. p. 40, 2014.

[50] A. L. Simeone, E. Velloso, J. Alexander, and H. Gellersen. Feet movement in desktop 3d interaction. In *3D User Inter-*

faces (3DUI), 2014 IEEE Symposium on, pp. 71–74. IEEE, 2014.

[51] M. Slater, M. Usoh, and A. Steed. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(3):201–219, 1995.

[52] H. Son, H. Gil, S. Byeon, S.-Y. Kim, and J. R. Kim. Realwalk: Feeling ground surfaces while walking in virtual reality. *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, p. D400, 2018.

[53] Q. Sun, L.-Y. Wei, and A. Kaufman. Mapping virtual and physical reality. *ACM Transactions on Graphics (TOG)*, 35(4):64, 2016.

[54] A. Sundaresan, A. RoyChowdhury, and R. Chellappa. A hidden markov model based framework for recognition of humans from gait sequences. In *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, vol. 2, pp. II–93. IEEE, 2003.

[55] A. Tajadura-Jiménez, M. Basia, O. Deroy, M. Fairhurst, N. Marquardt, and N. Bianchi-Berthouze. As light as your footsteps: altering walking sounds to change perceived body weight, emotional state and gait. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2943–2952. ACM, 2015.

[56] J. N. Templeman, P. S. Denbrook, and L. E. Sibert. Virtual locomotion: Walking in place through virtual environments. *Presence*, 8(6):598–617, 1999.

[57] L. Terziman, M. Marchal, M. Emily, F. Multon, B. Arnaldi, and A. Lécuyer. Shake-your-head: Revisiting walking-in-place for desktop virtual reality. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, pp. 27–34. ACM, 2010.

[58] K. Tumkur and S. Subbiah. Modeling human walking for step detection and stride determination by 3-axis accelerometer readings in pedometer. pp. 199–204, 2012.

[59] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr. Walking> walking-in-place> flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 359–364. ACM Press/Addison-Wesley Publishing Co., 1999.

[60] E. Velloso, J. Alexander, A. Bulling, and H. Gellersen. Interactions under the desk: A characterisation of foot movements for input in a seated position. In *Human-Computer Interaction*, pp. 384–401. Springer, 2015.

[61] E. Velloso, D. Schmidt, J. Alexander, H. Gellersen, and A. Bulling. The feet in human–computer interaction: A survey of foot-based interaction. *ACM Computing Surveys (CSUR)*, 48(2):21, 2015.

[62] T. von Marcard, B. Rosenhahn, M. J. Black, and G. Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer Graphics Forum*, vol. 36, pp. 349–360. Wiley Online Library, 2017.

[63] J. Wang and R. Lindeman. Leaning-based travel interfaces revisited: frontal versus sidewise stances for flying in 3d virtual spaces. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, pp. 121–128. ACM, 2012.

[64] J. D. Wendt, M. C. Whitton, and F. P. Brooks Jr. Gud wip: Gait-understanding-driven walking-in-place. In *Proceedings/IEEE Virtual Reality Conference; sponsored by IEEE Computer Society Technical Committee on Visualization and Graphics. IEEE Virtual Reality Conference*, vol. 2010, p. 51. NIH Public Access, 2010.

[65] A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE transactions on pattern analysis and machine intelligence*, 21(9):884–900, 1999.

[66] Y. Wu, W. Xu, J. J. Liu, M.-C. Huang, S. Luan, and Y. Lee. An energy-efficient adaptive sensing framework for gait monitoring using smart insole. *IEEE Sensors Journal*, 15(4):2335–2343, 2015.

[67] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*, pp. 1033–1040. ACM, 2006.

[68] W. Xu, M.-C. Huang, N. Amini, J. J. Liu, L. He, and M. Sarrafzadeh. Smart insole: A wearable system for gait analysis. p. 18, 2012.

[69] L. Yan, R. Allison, and S. Rushton. New simple virtual walking method-walking on the spot. In *Proceedings of the IPT Symposium*, 2004.

[70] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Ijcai*, vol. 15, pp. 3995–4001, 2015.

[71] K. Yin and D. K. Pai. Footsee: an interactive animation system. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 329–338. Eurographics Association, 2003.

[72] Z. Zheng, T. Yu, H. Li, K. Guo, Q. Dai, L. Fang, and Y. Liu. Hybridfusion: real-time performance capture using a single depth sensor and sparse imus. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 384–400, 2018.