A Low-Cost Human Locomotion Speed Recognition for Augmented Virtual Environments Exploration

Marina Vela Nuñez, Carlo Alberto Avizzano, Emanuele Ruffaldi and Massimo Bergamasco

Abstract— **This paper presents a novel and low-cost interface designed for real-time human locomotion speed recognition, which fits with the exploration of kinesthetic virtual environments (VE). According to the interface paradigm, the human locomotion recognition feeds VE navigation control. An experimental session has been organized in order to acquire acceleration data related to locomotion of 10 healthy subjects (men and women) aging between 23 and 35 years. A treadmill has been used to capture the velocity at which subjects were moving. Our system was designed to optimize classification performances in human locomotion speed recognition in real-time. The recognized human speed locomotion has been shown to enhance users' sensation of presence in the virtual environment. A simple scenario has been developed to assess the system functionality. The experiments carried out show that our system is excellent at classifying a wide range of human locomotion and can be used both in virtual and augmented reality (VR) environments for improved interaction.**

I. INTRODUCTION

Fig. 1. *The experimental Virtual Reality Environment.*

The trend in VR is to simulate environments which seem increasingly real to humans. However, one of the principal limitations is in the actual size of the physical visualization spaces. Hence, despite the advances in human motion recognition, research attention is still required to develop new devices and algorithms that facilitate and make realistic navigation issues in VR environments. The improvement of these systems make the human-VR interactions more natural, and provide users with a better sensation of full immersion.

Motion recognition systems are based on different types of capturing technologies. We distinguish marked from markless based approaches. The former relies generally on tracking optical markers attached to a person [12], [13], while the latter tracks features appearing naturally in videos [3], [10], [11]. In professional applications a common choice is to adopt optical motion capture systems. These systems have the advantage of being accurate but expensive as they require users to wear markers in proper patterns. As a result, these systems must be set up appropriately in advance. Recently new and accessible devices are emerging on the market, such is the case of the wiimote device that has accelerometer sensors. Accelerometers are easy to handle and they do not need previous setup. The information provided by accelerometers is poorer than the one available from position trackers, yet usable for specific kinds of structured interaction. In [7], Shiratori has shown how elementary classification of patterns applied to low cost device can animate virtual subject motions. Liu [8] used HMM applied to accelerometers path to correlated choreagraphical data. We benchmarked this type of sensor previously, in [6] we presented a system to recognize human gait for augmented VE exploration. The system successfully classified between walking and running gaits, since the implementation of this system was based in a careful study of the human locomotion behavior during walking and running. However, the results of such benchmarks showed a lacks of fidelity in locomotion reconstruction when elementary locomotion as well as constant speed navigation were addressed.

All the existing attempts use classification tools to simulate a virtual gesture of the user without correlating in that specific navigation properties that are induced by acceleration profiles. For such reasons, we addressed the development of a novel algorithm that, besides the detection of walking or running, provide relevant indicators for the navigation speed control. Finally, the goal of our research was to focus on off-the-shelf sensors. The sensors employed take advantage of new and easily accessible technologies in the market, as the wiimote control (Fig. 3). One of the advantages of leg based sensors is that the users do not need to use their hands to navigate in the VE. The hands are free to perform other interaction activities.

In what follows we propose a sensor system to detect different speed variations in human locomotion in real time, based on neural networks (NNs) as classifier. The NN are trained in a calibration phase and hence they can be suitable for real time computation at a reduced computational cost. In addition, as we will see, with respect other classification methods (such as HMM) the proposed approach allows scalability of navigation velocities while optimizing the computational cost.

With respect to existing devices, the proposed system was

This work was not supported by any organization

M. Vela Nuñez, C. Alberto Avizzano, Emanuele Ruffaldi and Massimo Bergamasco are with PERCRO Perceptual Robotics Laboratory, Scuola Superiore Sant'Anna, Piazza Martiri della Libertà, 33 - 56127 Pisa (Italy) m.velanunez@sssup.it, carlo@sssup.it, pit@sssup.it, bergamasco@sssup.it

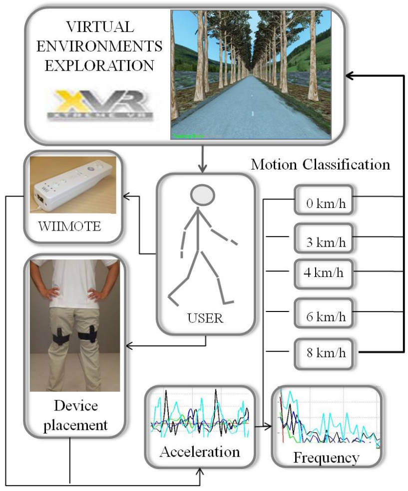


Fig. 2. *Block diagram of the speed locomotion recognition.*

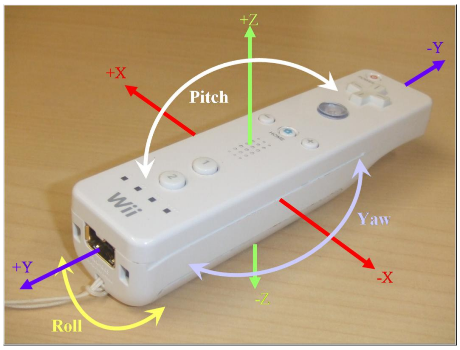


Fig. 3. *Wiimote controller device with the axes reference.*

calibrated and benchmarked with subjects moving in a real condition in order to have an optimal estimation of the speed. Ten subjects, pacing at different velocities contributed to collect a dataset for the NN training. A treadmill, and an external positional reference, were employed to ensure that the dataset was collected at known constant velocities. A set of accelerometers were attaced to user's legs during data acquisition. The constant velocity of the treadmill do not affect the accelerometer signal profiles.

In order to improve the robustness of the reconstruction, the acceleration data were analyzed in the frequency domain and selected features fed the classification network. Once trained the Network was converted in an online estimator of the speed locomotion. The estimated locomotion speeds were then provided to the VE rendered (XVR). A schematic block diagram that describes how the locomotion classifier were integrated in the VE system is given in Fig. 2.

Among the advantages of the devised approach, we would also highlight the fact that the presented system does not need any calibration approach to adapt to different users. This is a direct consequence of the inter-subject optimization achieved in the design phase.

The quality of the developed algorithm was tested through a 3D virtual environment, which has the finality to give the user visual feedback of his/her locomotion and create the user's sensation of moving in a park. To create this illusion, we used objects such as virtual trees, a road, blue sky and mountains. (Fig. 1).

The paper is organized as follows: Section 2 gives an overview of related works. Section 3 provides details of the algorithm employed to recognize human locomotion speed in real time.

II. RELATED WORKS

A. Motion Recognition

Using accelerometers attached to the body is a method that has been proven effective for human motion recognition. Ling Bao [19] proposed a system for activities recognition using five accelerometers worn on different parts of the body. Features such as energy, frequency-domain entropy, and correlation were computed on the acceleration data, over 6.71 s sliding windows. Several classification methods were then applied and their performance tested on 20 everyday tasks activities of daily living (walking, running, bicycling, reading, stretching, etc.). A. Mannini and M.Sabatini [31] proposed a method for classifying human physical activities. Using five accelerometer, they computed data frame lasted at 6.7 s with every new frame available every 3.35 s. Features such as DC component, energy, the frequency-domain entropy, and correlation coefficients were computed. N. Ravi et al. [21] developed a system to recognize activities using a single accelerometer worn near the pelvic region. The authors computed features over a sliding window of 5.12 s which was sufficient to capture cycles in the different activities considered. As for the classification, they evaluated the performance of the base-level classifiers as well as the meta-level-classifiers such as boosting [24].

T. Huynh and B. Schiele [25] studied the effect of computing several features over different window lengths (0.25, 0.5, 1, 2 and 4 sec) on the recognition rates of common activities using acceleration data. The authors concluded that a better recognition occurs when selecting different window lengths for different activities. For example, the 1 second window has been chosen for the activities 'jogging' and 'walking'; the 2 and 4 second windows are more adapted for 'skipping' and 'hopping', and the 0.25 and 0.5 second lead to a better recognition rate for the activity 'standing'.

Takeuchi et al. [22] investigated, between the frequency and the time domain, the best features parameters for human action recognition. The authors employ the acceleration information in three axes and their derivatives as the base-line method in the time domain. They use Mel-Frequency Cepstral Coefficients as feature parameters in the frequency domain. They found that the best recognition rates are obtained when the features of the axis that contained most of gravitational acceleration information were included.

For the best locomotion interpretation, our method employs both time and frequency domain analyses on the acceleration information in order to recognize the changes in the locomotion speed in real-time. The algorithm is conceived to operate in real-time in order to allow Virtual

Reality Exploration (VRE). In this condition even a small delay negatively impacts the interaction. During a treadmill walking the simultaneous, locomotion accelerations were measured. Data were sampled at 20ms period, accelerations were interpreted in frequency domain.

In order to maintain low the computational burden of the motion detection algorithm, frequency features properties were analyzed by visual inspection. Successively, frequency data were used to train a NN. Finally the trained NN was embedded as a real time sensor that delivers the user speed when locomotion are detected in the VE. This information were sent to the VR engine(XVR) coherently with the locomotion of his or her.

B. Environment Navigation

True locomotion base is not yet uncommon in VE navigation. The most common alternatives are: flying, via the use of a wand; simulated walking, with uniform velocity determined through a position or inclination tracker; and normal walking without any speed control. In the latter case, the location and orientation of the larger virtual environment is typically corrected at regular timings in order to match the position of the subject in the real environment [23].

Immersive virtual environments (IVE) have several applications, ranging from architectural design to situational awareness training. Such VR systems require that the proprioceptive information, that we use unconsciously to form a mental model of the body, be overlaid with sensory data that are supplied by computer-generated displays [29]. The main problem in VE research is still the continuing search for most natural ways of human interaction (full immersion). Previous studies have demonstrated that real walking in virtual environments is better than virtual walking; it is more natural and produces a higher sense of presence than other navigation methods [17], [18], [29]. The actions and perception of the user should be in sequence with the VE. Therefore, the new generation of sensor systems used for virtual human interactions should be more robust in human action recognition. In our days a sensor system is higher rated when it is designed to respond to several unexpected events during the interaction. These systems should be designed using devices easy to setup previous to the interaction and more accessible.

III. REAL-TIME HUMAN LOCOMOTION SPEED RECOGNITION

A. Methodology

The acceleration behavior during human locomotion is crucial in selecting relevant features for gait recognition. When we walk or run, our movement is cyclic and our legs are constantly transitioning between two phases: stance and swing [12]. The stance phase is active when the foot is placed on the terrain, while the swing phase is active during the foot air floating.

It is typical that in the floating phase the acceleration of the foot is in the forward direction, while in the stance phase the foot is almost resting. This repeats cyclically during walking or running. The swing phase is characterized by a down-up course of the acceleration, ending with the heel-strike. The latter is usually clearly visible as a negative peak.

These conditions allow us to perform a clear phase segmentation of the user locomotion as described in [6].

1) Data Acquisition: An experimental champaign was carried out to extract relevant features from the acceleration signal. The ideal situation is to get acceleration samples from subjects who walk and run in a regular pattern (constant speed). However, during a natural walk or run, the person velocity presents small variation from regularity. To minimize these variations, to reduce the space required for the capture campaign and to ensure that the average velocity was correctly collected, a treadmill with preset velocities was used to impose the speed at which the users had to move. Therefore, we have asked 10 healthy subjects, aging between 23 and 35 years, to walk and run on the treadmill.

In this manner the locomotion references for 3km/h, 4km/h, 6km/h and 8km/h were collected. During the test campaign two portable devices were bound to user's thighs (Fig. 2), close to the knee joint, which performs most of the work during human locomotion [5] and is the most useful information to discriminate activities [19].

The acceleration data acquired from the wiimote at sampling frequency of 50Hz was then sent to a PC via Bluetooth and analyzed in a Matlab/Simulink environment.

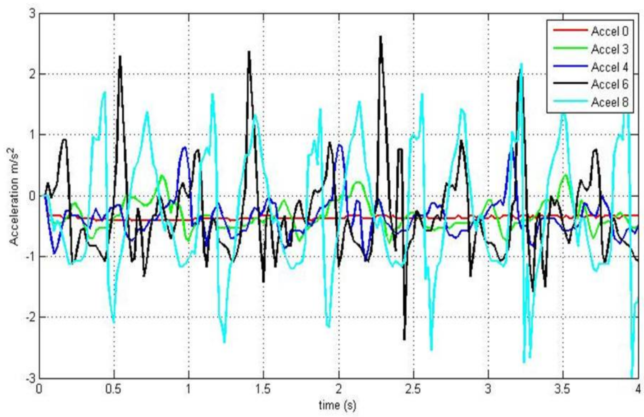


Fig. 4. *Acceleration data obtained at 4 min from the velocities of: 0km/h, 3km/h, 4km/h,6km/h and 8km/h.*

2) *Feature Extraction:* To compute the subject's speed, we transformed the acceleration signal (fig. 4) in the frequency domain using for it the absolute value of the Fourier transform (FFT). The acceleration information were chunked in series of N samples overlapping windows which are updated each sampling time (20 ms). The FFT reduces the number of computations needed for N points from $2N^2$ to 2N lg N, where lg is the base-2 logarithm. In our case we used N=200 for the training phase, and N= 64 during the navigation phase (Fig. 5). In both cases we trained the following NN to recognize the velocity at the end of the sequence, thus minimizing the perception of the delay to 20ms (plus computational time) in total.

After a visual comparison of the several transforms, we chose the most characteristics data of the frequency as the features used to training the NN for the human locomotion classification. Our algorithm uses the features from the frequency components shown in the vertical axis acceleration that reflects the acceleration provided by the rising and lowering of the body and give more information about the person's locomotion [19], [22]. Therefore, to get translation invariant operator in the signals, consistent features from different parts of the walking cycle were extracted. We create a matrix with the following relevant features to train the NN:

- The mean of the 6 values of discrete frequencies close to the first sampling border.
- The 5 values obtained from the coefficients of a 4th degrees polynomial computed to interpolate the absolute values of FFT data.
- The mean of the 6 values of discrete frequencies close to the end sampling border.

Only the discrete frequencies close to the sampling border are consider relevant since they show the highest variance amongst the data. These features are show in the figure 5 in the orange ellipses. The polynomial is use to find the coefficients of the signal of degree N=4 that fits the best data sense, in a least-squares. And return a row vector (P) of length N+1 that contain the polynomial coefficients in descending powers, $P(1)X^N + P(2)X^{(N-1)} + ... + P(N)X + P(N+1)$. In our case we give 64 FFT values and it output 5 frequency values. These values represents half of the FFT signal since the FFT is symmetric. From Bayesian point of view, many regularization techniques correspond to imposing certain prior distributions on model parameters. Regularization involves introducing additional information in order to solve an ill-posed problem or to prevent overfitting. Therefore, the selection of only few features allows a computation-heavy algorithm.

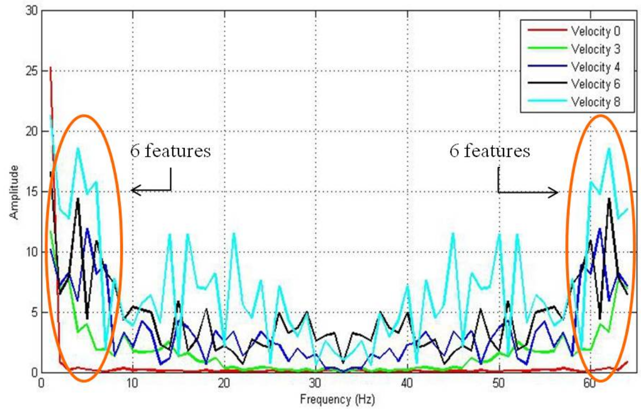


Fig. 5. *Windows of the absolute value of the Fourier transform for each 200 acceleration samples.*

3) *Classifier Implementation:* For the complexity of the data, we decide to design a classifier base on NNs. The most suitable NN for our system is a feedforward Multi-Layer perceptron Neural Network (MLP) [30] some of its characteristics are: a) The model of each neuron in the network includes a *nonlinearity* at the output end. b) The network contains one or more layers of *hidden neurons* that enable the network to learn complex tasks by extracting progressively more meaningful features from the input patterns (vector). c) The network exhibits a high degree of *connectivity*, determined by the synapses of the network. These NNs have two different kind of signals.

- *Function Signals*, its come into the input, propagates forward (neuron-by-neuron), and emerge as an output signal of the network.
- *Error Signals*, originate at an output neuron of the network, and propogate backwards (layer by layer) through the network.

The design of our (MLP) classifier was done with three layers. The first layer consist on 5 components of the input data vector, correspond to the hidden layer that are 20 neurons; the third layer is the output that correspond to the classified of the different velocities; an intermediate hidden layer for data processing. (Fig. 6) The method used to train the network was Backpropagation. This algorithm lets each node's weight change and being more adaptable to the desired output.

By using samples for 10 different subjects we ensure a good intra-subject generalization, while the selection of only 7 features to feed the NN allows an excellent classifier algorithm (Bayesian regularization). The regularization generalization ability is not dependent on using a small number of neurons, but requires more computing power. A MLP network with 20 neurons proved to be balanced between these requirements; increasing the number of neurons has no more impact on the rate and indeed exhibited a loss of selectivity. The NN training was performed off line, this training method is not part of the on-line simulation. In paragraph B) is explained how the on-line model was implemented.

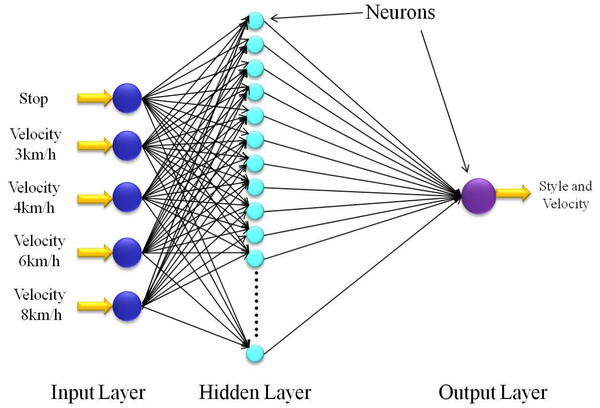


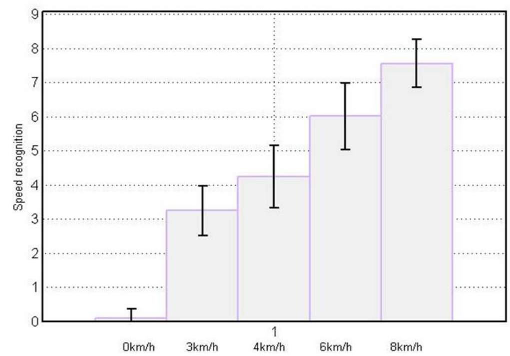
Fig. 6. *A feed-forward multilayer perceptron (MLP) network with 20 neurons.*

4) Validation of the locomotion speed classification: In order to estimate the accuracy of our predictive model, we applyed the technique of cross-validation, also called rotation estimation [26], [27], [28]. Data is divided into complementary subsets called (training_set), and (not_train_set). (training_set) are the subsets used for training the NN and (not_train_set) is the subset used to test the NN. One round of cross-validation includes training as well as one test of the NN. In our validation process we performed 10 rounds on various (training_set) with (not_train_set). The error of the predictive model is then computed as the normalized absolute value of the difference between inputted style velocity and model(NN) locomotion output classification. The fig. 7 shows the obtained mean and standard deviation og the speed recognition from the 10 rounds cross-validation test for the each velocity (0km/h, 3km/h, 4km/h, 6km/h, 8km/h). The error (ε) was estimated with the formula that is shows in equation 1, where obj represent the desired velocity and $model$ is the obtained output from the NN. The table I shows the obtained error for all velocities overall, it was 4% then, the accuracy of our NN was 96%. After this result we train the NN with all the 10 set samples, the training duration for the NN was 5.47sec and the performance computed with the mean square error method (with regularization) was 0.24%, where the maximum number of repetitions (EPOCHS) reached at 100. The cross-validation tests and the train of the NN were performed in Matlab version 7.10 on an Intel i7-720QM (1,6GHz) CPU mobile and 4.096MB of memory. Once locomotion recognition is successful, we developed a real time sensing component explained in the following section.

TABLE I

PERCENT OF ACCURACY AND ERRORS OF THE 10 ROUNDS CROSS-VALIDATION TEST

Speed	0km/h	3km/h	4km/h	6km/h	8km/h
%Accuracy	99.92	92	94	99.9	94.5
%Error	0.0793	7.93	6	0.1067	5.525



Speed	0km/h	3km/h	4km/h	6km/h	8km/h
Recognition	0.0760	3.2380	4.2400	6.0064	7.5580
Standard deviation	0.2800	0.7400	0.9200	0.9700	0.7100

Fig. 7. *Speed recognition with Cross-Validation test of the 5 velocities (0km/h, 3km/h, 4km/h, 6km/h, 8km/h).*

$$\varepsilon = \left(\frac{|obj - model|}{obj} \right) 100 \qquad (1)$$

5) Extrapolation test: In order to verify if the velocities are property extrapolate, we tested the NN with intermediate velocities, such as 5.5km/h and 7km/h, which are not used for training the NN. The Fig. 8 shows the results for both velocities 5.5km/h(red line) and 7km/h(blue line) during 21 seconds test. The most interesting in the results is that the NN never confuse the given (not train) velocity with one of the train velocities.

B. Online recognition model

1) Feature extration: A Simulink state machine has been built to perform FFT through recursive estimations. See (Fig. 10) magenta block. 1st recognition is delayed due to the lap

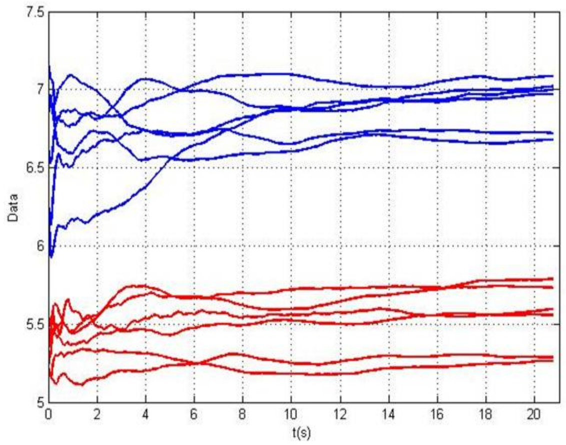


Fig. 8. *Rate recognition of 6 different samples of 5.5km/h (red line) and 7km/h (blue line) velocities during 21 seconds test.*

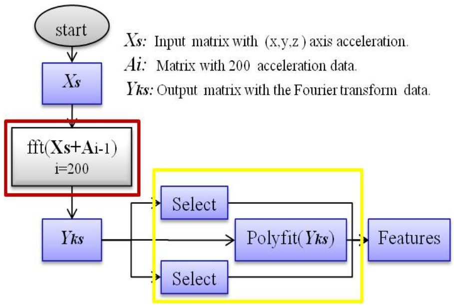


Fig. 9. *Feature extraction process, the red block represents the recursive process to estimate the FFT (cf. fig. 10, the state machine block), while the yellow block represent the feature extraction process (cf. fig. 10, the feature extraction block).*

of time necessary to achieve data for FFT (N=64 or 200) at the beggining. To solve startup issue, a startup sequence, describing a stationary subject, is preloaded. Therefore, the first output at 20ms for the NN will be a non-moving feature as subjects are usually locomotionless when starting an interaction. Then, values from the recursive cycle are progressively replaced with values from the subject's locomotion without delay. According to figure 9, X_s represents the input values (acceleration data), into the state machine. The state machine implements a recursive estimation of the most adherent FFT this prosses is represented in the (Fig. 9) by the red block. Each output from the latter is a $Y_k s$ matrix from which are selected the current features, in order to feed the NN. The FFT ($Y_k s$) for each 200 acceleration data are shown in fig. 5. The online feature extration are reprecented in (Fig. 9 yellow block), the two select blocks extract from the FFT the mean of the first and last six values, while the POLYFIT block extract from FFT the 5 values of the polynomial degree of 4.

2) Implementation of the embedded sensor: In order to read more than one Wii controller and to enable the PC to handle the acceleration data from the wiimote a "C" S-funtion was designed. The function relies on the Wiiuse library [4]. The S-funtion sends acceleration data to the state flow, wich is shown in the figure 10 together with the features extraction polynomial, and the classification network. The use of the embedded coder and the real time workshop finally helped us to package everything in a single executable module.

Once the gait is identified it sends the information to the virtual environment by a UDP protocol. Therefore, the velocity of the moving environment mirrors the speed locomotion of the user during interaction. The Simulink setup provides a flexible framework for handling the recognition in real-time. The models used in the VR environment were designed in 3D Studio Max. The virtual environement application was developed in XVR, which provides facilities to operate with 3rd party software without complexity[16].

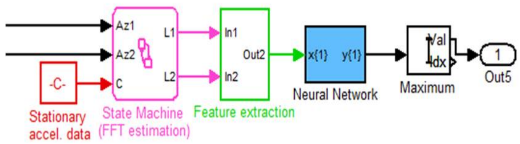


Fig. 10. *Simulink implementation of the phase classifier.*

C. Test Setup

Two portable devices (wiimote controllers) were used to register the acceleration signals [2] from the user's thighs (Figure 11). The wiimote communicated with the computer via a Bluetooth adapter, which provided a suitable operating range (from 1m to 100m). These devices sensed both rotational and translational accelerations by the triple-axis accelerometer (ADXL330) embedded in the controller: backward-forward (x-axis), lateral (y-axis), and vertical (z-axis) (Fig. 3).

The accelerometer has a measurement range of ± 3g (where g is the gravitational acceleration) and output analog voltage signals proportional to the acceleration. Four video projectors were used to display the images of the 3D Virtual environment. A treadmill was used for the walking sessions. The projectors were connected to the two scene rendering PC and a scene master which provides environment synchronization on each different rendering image[9]. Infitec

TABLE II

PERCENT OF ACCURACY AND ERRORS OF THE ONLINE TEST

Speed	3km/h	4km/h	6km/h	8km/h
%Accuracy	95.2033	95.85	97.045	94.4
%Error	4.7967	4.15	2.955	5.605

polarizing lens were adopted to achieve stereography through the use of wavelength polarized light. The embedded master receives the embeded sensor data through the common socket connection and provides to move coherently a virtual camara in the designed scenario.

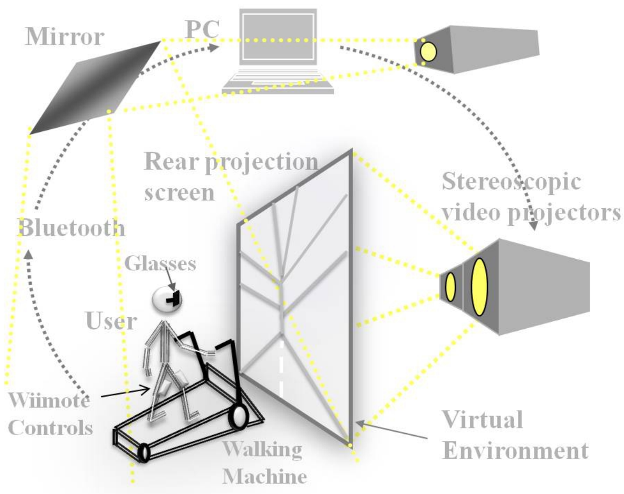
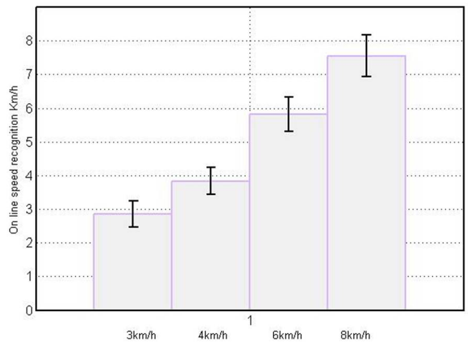


Fig. 11. *The full system setup (the use of the treadmill for the interaction is optional).*

1) Online experiments and results: In order to test the online model, we performed experiments using the treadmill to indicate the velocity in which the subject move. We asked 5 subjects aging between 23 and 35 years, (them accelerations were not used to train the NN) to walk at various (3km/h, 4km/h, 6km/h, 8km/h) velocities. We feed our system with the user's thigh acceleration obtained from the two devices. Then we compared the output speed of our model with the real speed locomotion of the user. The mean speed recognition and standard deviation are shown in figure 12. The error of the model was estimated using the equation 1 as explained before in paragraph 4). In the table II is shown the percent of the obtained error for all velocities overall, it was 4.38% consequently, the accuracy of our online classifier was 95.62%. The figure 13 shows the output speed recognition, obtained by an user's locomotion using a treadmil during 42sec online interation.



Speed	3km/h	4km/h	6km/h	8km/h
Recognition	2.8561	3.8340	5.8227	7.5516
Standard deviation	0.3873	0.3975	0.5109	0.6203

Fig. 12. *Online Speed Recognition.*

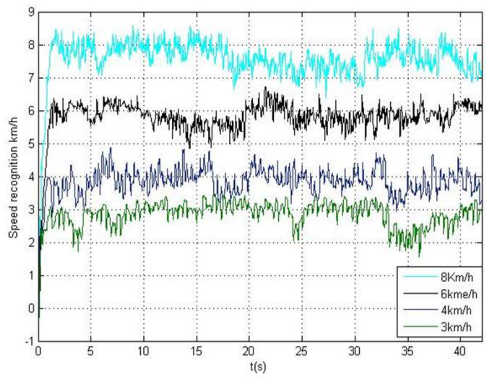


Fig. 13. *Signals of the speed recognition online, obtained by the user's locomotion using a treadmil during 42sec.*

IV. CONCLUSIONS

This paper presented a low-cost system for human locomotion speed recognition for Augmented VEE based on NN. A treadmill has been used to impose subjects velocity during a training phase. The accelerations of the user locomotion were coherently measured and subsequently converted to the equivalent frequency representation. Frequency data were used to train a NN so that it could classify and recognize the subject's global velocity. These features were analyzed carefully in order to develop a system which does not request much computational power, and can be easily adapted to the user needs. Here, we addressed the development of a novel algorithm that, besides the detection of human locomotion style, provides relevant indicators for the navigation speed control.

The carried out tests showed that, given the acceleration data information over time, it is possible to identify the various locomotion speeds with an accuracy of 96%. The sensor system presented here, feeds in real time the VR snd the generated VR environment is perceived by the subject who interacts according to his or her locomotion. This sensor system can be used for several applications, for which the exploration of 3D VR environments is needed. It is designed to be used in VR interactions where users are *walking in place*, use *real walking* or move with any electromagnetic tracker. Our future work will be focused on perform more real experiments in order to test our system in different VR applications.

V. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of: Nicolas Bourdillon, Leonard Johard, Haakon Faste and to all the volunteers that performed the experiments.

REFERENCES

[1] S. Mochon and T.A. McMahon, *Ballistic Walking*, J. of Biomech, vol. 13, pp. 49-57, 1980.
[2] J. C. Lee, *Hacking the Nintendo Wii Remote*, IEEE Pervasive Computing, 2008.
[3] N. Hasler, B. Rosenhahn, T. Thormählen, M. Wand, J. Gall and H.P. Seidel, *Markerless Motion Capture with Unsynchronized Moving Cameras*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), 2009.
[4] http://www.wiiuse.net/?nav=docs
[5] J.M. Donelan, Q. Li, V. Naing, J.A. Hoffer, D.J. Weber and A.D. Kuo, *Biomechanical Energy Harvesting: Generating Electricity During Walking with Minimal User Effort*, Science, Vol. 319. no. 5864, pp. 807-810, 2008.
[6] M. Vela, C. Alberto Avizzano, Emanuel Ruffaldi, and Massimo Bergamasco, *Human Gait Recognition for Augmented Virtual Environments Exploration*, 19th IEEE International Symposium in Robot and Human Interactive Communication,RO-MAN 2010.
[7] T. Shiratori and J.K Hodgins, *Accelerometer-based User Interfaces for tha Control of a Physically Simulated Character*, ACM Transaction on Graphics 27(5), SIGGRAPH Asia 2008.
[8] Y. Liu, X. Liang and Z. He,*Motion Choreographing based on Kinetic Energy with Accelerometer Signal*, Computer Animation and Virtual Worlds, 20(2,3) 2009.
[9] http://www.vicon.co.kr/product/pdf/UltranetLR.pdf
[10] A. Sundaresan and R. Chellappa, *Markerless motion capture using multiple cameras*, IEEE Computer Vision for Interactive and Intelligent Environment (CVIIE'05), 2005.
[11] P. Kaimakis and J. Lasenby, *Markerless Motion Capture with Single and Multiple Cameras*, In Proc. IEEE International Conference on Image Processing (ICIP), vol. 4, pp. 2607-2610, Singapore, 2004.
[12] L.Y. Chang, N.S. Pollard, T.M. Mitchell and E.P. Xing, *Feature selection for grasp recognition from optical markers*, in IEEE International Conference on Intelligent Robots and Systems, 2007.
[13] M.C. Silaghi , R. Plï£¡nkers , R. Boulic , P. Fua and D. Thalmann, *Local and Global Skeleton Fitting Techniques for Optical Motion Capture* , Proceedings of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments, p.26-40, January 1998.
[14] D.G. Smith, *Walking: How Can Something So Simple Be So Complicated?*, in Motion, vol. 14, issue 6, 2004.
[15] A.TH.M. Willemsen, F. Bloemhof and H.B.K. Boom *Automatic Stance-Swing Phase Detection from Accelerometer Data for Peroneal Nerve Stimulation*, IEEE Transactions on Biomedical Engineering, vol. 37, no. 12, december 1990.
[16] M. Carrozzino , F. Tecchia , S. Bacinelli , C. Cappelletti and M. Bergamasco, *Lowering the development time of multimodal interactive application: the real-life experience of the XVR project*, Proc. of International Conference on Advances in computer entertainment technology, ACM SIGCHI, pp. 270-273, Spain, 2005.
[17] M. Slater, M. Usoh and A. Steed, *Influence of a Walking Technique on Presence in Virtual Reality*, ACM Transactions on Computer-Human Interaction, Vol. 2, No. 3, pp. 201-219, 1995.
[18] M. Usoh, K. Arthur, M.C. Whitton, R. Bastos, A. Steed, M. Slater and F.P. Brooks, *Walking > walking-in-place > flying, in virtual environments*, In Proceedings of Computer graphics and interactive techniques, SIGGRAPH, pp. 359-364, NY, USA, 1999.
[19] Ling Bao, *Physical Activity Recognition from Acceleration Data under semi-Naturalistic Conditions*, Thesis submitted to the Department of Electrical Engineering and Computer Science, 2003
[20] J. Mäntyjärvi, M. Lindholm, E. Vildjiounaite, S.-M. Mäkelä, and H. J. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in IEEE International Conference on Acoustics, Speech, and Signal Processing, March 2005.
[21] Nishkam Ravi et al. (2005, July). Activity recognition from accelerometer data. in Proceedings of the Seventeenth Innovative Applications of Artificial Intelligence Conference, 11–18. Menlo Park, Calif.: AAAI Press.
[22] Takeuchi, Shin'ichi, Tamura, Satoshi, Hayamizu, Satoru, "Human Action Recognition Using Acceleration Information Based On Hidden Markov Model", Proceedings : APSIPA ASC 2009 : Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, pp. 829-832, 2009
[23] V. Interrante, B. Riesand, and L. Anderson. Seven League Boots: A New Metaphor for Augmented Locomotion through Moderately Large Scale Immersive Virtual Environments. In Proceedings of Symposium on 3D User Interfaces, pp. 167-170. IEEE, 2007
[24] Y. Freund and R.E. Schapire, Experiments with a new boosting algorithm, In International Conference on Machine Learning, pp. 148-156, 1996.
[25] T. Huynh and B. Schiele, Analyzing features for activity recognition, Proceedings of the joint conference on Smart objects and ambient intelligence, pp. 159-163, 2005.
[26] Picard, Richard; Cook, Dennis, "Cross-Validation of Regression Models". Journal of the American Statistical Association, 1984.
[27] Kohavi, Ron, "A study of cross-validation and bootstrap for accuracy estimation and model selection". Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann. pp. 1137-1143, 1995.
[28] Devijver, P. A., and J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, London, 1982
[29] 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, Vol. 2, No. 3, pp. 201-219, September 1995.
[30] K.k. Aggarwal, Y. Singh, P. Chandra and M. Puri. Bayesian Regularization in a Neural Network Model to Estimate Lines of Code Using Function Points. Journal of Computer Sciences 1 (4): 505-509, 2005
[31] A. Mannini and A.M. Sabatini, Machine learning methods for classifying human physical activity from onbody accelerometers. In Sensors 2010, 10, 1154-1175.