Real time gesture recognition using Continuous Time Recurrent Neural Networks

Gonzalo Bailador
Fac. de Informática (UPM)
Univ. Politécnica de Madrid
Madrid, Spain
gonzalo.bailador@upm.es

Daniel Roggen
Wearable Computing Lab
ETH Zürich
Zürich, Switzerland
droggen@ife.ee.ethz.ch

Gerhard Tröster
Wearable Computing Lab
ETH Zürich
Zürich, Switzerland
troester@ife.ee.ethz.ch

Gracián Triviño
European Centre for Soft Computing
Edificio Científico Tecnológico
Asturias, Spain
gracian.trivino@softcomputing.es

ABSTRACT

This paper presents a new approach to the problem of gesture recognition in real time using inexpensive accelerometers. This approach is based on the idea of creating specialized signal predictors for each gesture class. These signal predictors forecast future acceleration values from current ones. The errors between the measured acceleration of a given gesture and the predictors are used for classification. This approach is modular and allows for seamless inclusion of new gesture classes. These predictors are implemented using Continuous Time Recurrent Neural Networks (CTRNN). On the one hand, this kind of networks exhibits rich dynamical behaviour that is useful in gesture recognition and on the other, they have a relatively low computational cost that is interesting feature for real time systems.

Keywords

Gesture Recognition, Recurrent Neural Networks

1. INTRODUCTION

Wearable computers are intelligent devices seamlessly integrating in clothing or objects we carry around everyday. By being "on the body" wearable computers are at an ideal location to detect important informations about the "state" of the user, such as his position, his activities or gestures or even his social interactions. This *context awareness* [7, 17] allows wearable computers to e.g. become the personal health assistant of the user [23] (e.g. by monitoring the physical activity) or to deliver context-based information [20]. User gestures are an important aspect of the context. They can be used for human-computer interactions [14], to detect social interactions [15], or even provide an insight into affective disorders or depression [12].

One of the challenge of gesture recognition in wearable computing is to offer good recognition accuracy on miniature wearable devices (e.g. [21]) which offer long battery life, and consequently limited computational power.

Hidden Markov models, dynamic programming and neural networks have been investigated for gesture recognition [6] with hidden Markov models being nowadays one of the predominant approach to classify sporadic gestures (e.g. classification of intentional gestures [5]).

Fuzzy expert systems has also been investigated for gesture recognition[8] based on analyzing complex features of the signal like the doppler spectrum. The disadvantage of these methods is that the classification is based on the separability of the features, therefore two different gestures with similar values for these features may be difficult to classify.

In this article we describe a method to classify gestures from inexpensive accelerometers. This approach is based on signal *predictors* for each gesture class that is to be classified. These signal predictors forecast future acceleration values from current ones. The errors between the measured acceleration of a given gesture and the predictors are used for classification. Signal predictors have the advantage of operating directly on the raw sensor signal. They do not require feature extraction that is common in Bayesian classification or hidden Markov models. Therefore they may avoid designer bias in feature selection and thus have the potential of being more general. This prediction approach has been used previously to recognize gestures obtaining high recognition rates. In [22] several gestures captured by a magnetic motion tracker are classified using predictors based on kalman filter. Neuro-Fuzzy systems have also been studied to create such predictors [1, 13]. In particular, in [1] we used Neuro Fuzzy systems to create acceleration signal predictors obtaining high recognition rates which showed the suitability of the prediction approach with acceleration signals.

The objective of this paper is to investigate how this predictor approach performs when using a more general signal predictor than in our previous work with Neuro Fuzzy

BodyNets '07, June 11-13, 2007, Florence, Italy

systems [1]: a Continuous Time Recurrent Neural Network (CTRNNs) is used instead as predictor. CTRNNs are networks of continuous model neurons without constraints placed on their connectivity [10] that exhibit rich dynamics [2]. It has been proved that the internal state of a CTRNN neuron can approximate any dynamical system [9]. The dynamic and non-linear nature of CTRNN makes them suited for temporal information processing. Sequential behaviors and learning with CTRNNs have been illustrated in bit sequence tasks [25] and in robotics [3].

CTRNN may be therefore well suited as universal signal predictors. They may cope with the dynamics of any gesture signal with the appropriate time constants and interconnection weights.

The rest of the paper is organized as follows: in section 2 the device used for signal capture is described. Section 3 explains the structure of the CTRNN predictors and how to use them to recognize gestures. In section 4 we describe the gesture recognition experiment carried out to validate the approach. In section 5 we discuss the results and highlight future work. And finally, section 6 concludes this paper.

2. SENSOR HARDWARE

In order to capture the gestures used in the following experiments, a tri-axial accelerometer is used [1]. This device was selected because it has good features for wearable applications. It is small sized, so it can be worn easily (see figure 1). It has as well a very low power consumption so it can operate for long periods of time (12 hours) with a single battery that is inside the case. This accelerometer module is connected over Bluetooth to the personal computer capturing the data.

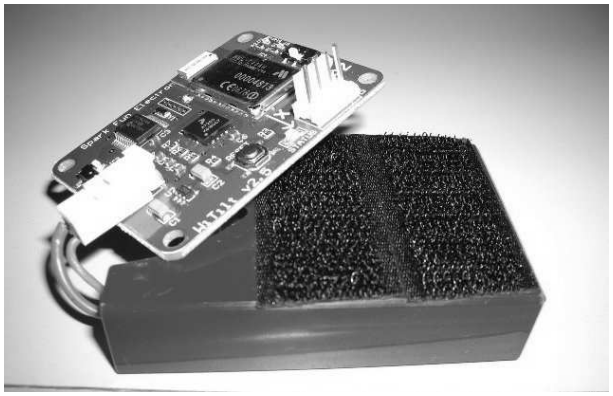

Figure 1: Tri-Axial Acceleration Sensor 64x40x15 milimeters

The data provided by this sensor consists in an acceleration vector with three components: one for each axis (ax, ay, az). Their values are measured in gravity units (g) in the range of [-6g,6g] encoded with 10 bits. This vector is captured with a sampling rate of 100 Hz, which is fast enough for our purpose since the maximum frequency of hand gestures is about 10 Hz [24].

During the experiments, this sensor was held in the hand in a vertical orientation. Furthermore, to segment some gestures we used a wireless mouse that was held in the other hand. Data from the wireless button was recorded on the personal computer together with the acceleration data. This button was pressed before doing the gesture and released after finishing it. An example of the acceleration signals for a circular hand motion is illustrated in figure 2.

[1]The acceleration sensor used in this work is the module Witilt v2.5 provided by Sparkfun electronics http://www.sparkfun.com

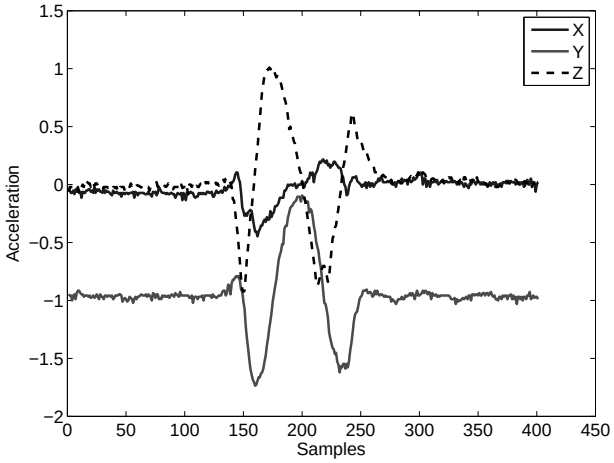

Figure 2: Acceleration signals recorded at the hand when performing a circular hand motion.

3. GESTURE RECOGNITION USING SIGNAL PREDICTORS

The recognition system presents a high modularity because it is completely composed of the basic component represented in figure 3. There are as many components as number of different class of gestures. This component consists of three main blocks: a memory block in order to delay one time step the input signal, a predictor block that is the core of the method and an error block that is used to calculate the prediction error.

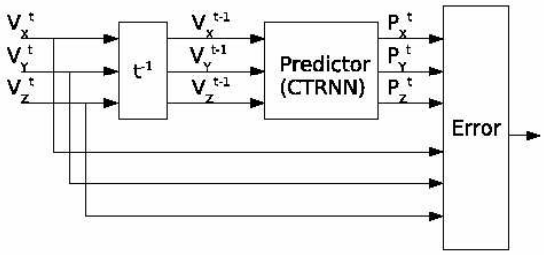

Figure 3: Basic Component

3.1 Predictor block using CTRNN

As stated in introduction, the predictor block is implemented with CTRNN. The specific architecture of the neural network used in this work consists of five fully connected neurons in which each neuron can be connected to every input (see figure 4). The input of this predictor block is the acceleration vector in the previous time step $V_{[X,Y,Z]}^{t-1}$ and

the output is the prediction of the acceleration vector for the present time $P^t_{[X,Y,Z]}$ that are the activations of three neurons.

In order to obtain the output values, it is necessary to calculate the activations of all neurons. The activation of the neuron i is calculated using the following differential equation ([3]):

$$\frac{d\gamma_i}{dt} = \frac{1}{\tau_i}\left(-\gamma_i + \sum_{j=1}^{N}\omega_{ij}A_j + \sum_{k=1}^{X,Y,Z}\omega_{ik}V_k\right) \quad (1)$$

$$A_j = \sigma(\gamma_j - \theta_j) \quad where \quad \sigma(x) = \frac{1}{1+e^{-x}} \quad (2)$$

Where N is the number of neurons in this case five, γ_i is the state of neuron i, τ_i is the time constant of neuron i, ω_{ij} is the weight of the synapse from neuron i to j. A_j is the activation of the neuron j that is calculated using the equation 2 where θ_j is the bias of that neuron, ω_{ik} is the weight that neuron i applies to input k and lastly V_k is the value of the predictor inputs for each axis X,Y and Z.

To discretize the differential equation 1 we use the Forward Euler numerical integration as in [3] so, the iterative update rule for the state of each neuron is:

$$\gamma_i^{t+1} = \gamma_i^t + \frac{\Delta t}{\tau_i}\left(-\gamma_i^t + \sum_{j=1}^{N}\omega_{ij}A_j^t + \sum_{k=1}^{X,Y,Z}\omega_{ik}V_k^t\right) \quad (3)$$

Where Δt is the integration step and for this work it was fixed to 0.01s because this is the sampling rate of the sensor. Furthermore, since the outputs of the CTRNN are in the range $[0,1]$, the inputs in range $[-6g,+6g]$ are normalized to this range by a linear mapping.

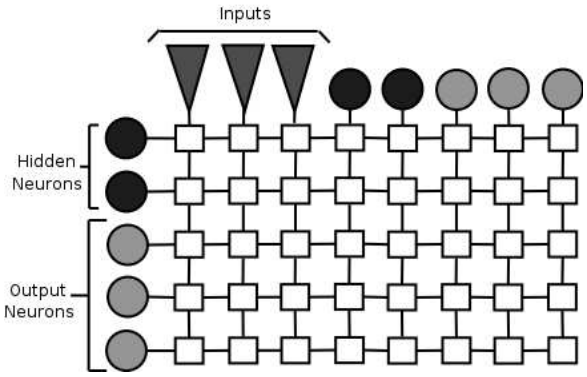

Figure 4: Architecture of the CTRNN

3.2 Prediction error block

The function of the error block is to measure how good is the prediction produced by the predictor. For this, it calculates the error between the real signals and the predicted ones. In particular for this work, the error measurement for a sample is the mean absolute error of each axis of the signal, computed with the following equation:

$$Pred\_Error^t = \sum_{i}^{X,Y,Z} |P_i^t - V_i^t|/3 \quad (4)$$

This is the prediction error for only one sample so, to obtain the prediction error for one gesture, the mean of this error is calculated for all its samples.

Parameter	# per neuron	Range
Neuron weights	5	[-0.25,0.25]
Input weights	3	[-0.25,0.25]
Time constant	1	[0.01,0.1]
Neuron Bias	1	[-0.25,0.25]

Table 1: Parameters of each Neuron of CTRNN

$$Pred\_Gesture\_Error = \sum_{t=1}^{T} Pred\_Error^t/T \quad (5)$$

Where T is the number of samples of the gesture.

3.3 Training of the signal predictors

Once the architecture of the signal predictor is defined (number of neurons, connections), the next step is to find the appropriate parameters to create the predictors of each class. In table 1 are shown all parameters for each neuron that need to be trained. The second column indicates how many parameters of this type there are in each neuron. The last column shows the range used for these parameters that were chosen after several trials with different ranges.

For this type of neural networks, a global optimization of the network parameters using genetic algorithms can be performed [3]. Genetic algorithms allow to do robust global searches in complex search spaces [11]. The basic idea of this method is to represent the neuron parameters as a bit string (the genetic string). A population of such string is then "evolved" using operators of selection, mutation and crossover inspired by biological evolution, in order to maximize or minimize a fitness function. In this case, each genetic string represents the parameters of a CTRNN. The length of this genetic string is 300 bits and each block of 60 bits encodes the parameters of a single neuron. In this neuron block are represented 5 connection weigths between neurons, 3 weights applied to the inputs, 1 time constant and 1 bias of the neuron in this order. All parameters are binary encoded on 6 bits in the range indicated in table 1.

To evolve the CTRNN, a standard genetic algorithm was used with a population of 100 invididuals, rank selection of the 30 best individuals, one-point crossover rate of 70% and mutation rate of 1% per bit and elitism that copies the best individual without change in the new generation.

The fitness function that guides the search of the genetic algorithm reflects how well the CTRNN predicts a training set T that is composed of several instances $T_{1..N}$ of gestures of the same class. The fitness function is the mean of the measure *Pred_Gesture_Error* (5) for all training instances. Therefore, the genetic algorithm should minimize it because the smaller this measure, the better the predictor.

$$Fitness = \sum_{i=1}^{N}(Pred\_Gesture\_Error(T_i)/N \quad (6)$$

Notice that for each different training instance the activations of the CTRNN start with zero value so we decided to feed the CTRNN with twenty previous samples of the instance in order to initialize it.

3.4 Recognition of gestures

The first step to recognize gestures is to create a signal

predictor for each different gesture class that we want to recognize. This is done using the previous genetic algorithm with one training set for each gesture class. Each training set is composed of several instances of gestures of the same class.

After training, the system is used to recognize gestures in the following way: the acceleration signal that contains the gestures is provided sample by sample to all the predictors. So, the activations of all neurons of the predictors are updated for each sample. In contrast to the training step, where the neuron activations are reset at the beginning of the gesture, no reset is done during recognition to allow for continuous use (i.e. the predictors are provided with all the recorded samples that contain a succession of gestures). When all activations are updated, the prediction error produced by each predictor is computed by the error block. To classify each gesture, the information of segmentation is used to extract the part of the signal that belongs to that gesture. For this part, the measure *Pred_Gesture_Error* is computed for all the predictors. After comparing all these errors, the lowest one indicates the class of the analyzed gesture because usually the predictor trained with gestures of that class will produce the better prediction.

4. PERFORMANCE EVALUATION

A set of eight different gestures represented in figure 5 has been used to test the performance and accuracy of the recognition method. The begin and end of each gesture is marked with a circle and an arrow, respectively. All gestures were performed vertically. In this work, two different datasets were recorded by only one person that held the sensor in a vertical orientation with the right hand. For both datasets, twenty instances of each gesture were recorded in a random sequence. So, in total for each dataset, there are 160 gesture instances.

In the first dataset, the objective was to check the validity of the proposed approach for gesture recognition. In this dataset the noise sources were minimized: the person that performed all gestures was sitting during the recording time. Furthermore, these gestures were isolated: between one gesture and the next, the hand rested in the same position. In a first step, the segmentation was done manually pressing a button but we noticed that this was not accurate because the person sometimes took some time to release the button after finishing the gesture. So, in order to avoid these segmentation problems, in this test the gestures were segmented automatically using the magnitude of the acceleration signal. When the device is not moving the only acceleration present in the sensor device is the gravity that is equal to 1 g, therefore if the magnitude of the acceleration is different from 1 g this means that there is a gesture. In particular it was considered that the device was moving when the magnitude was out of the range $[-1.15g, 1.15g]$.

The second dataset was recorded in order to test how the predictors behave in a realistic environment. As in the previous dataset, the same gestures were done by only one person but this person moved around the room in an uncontrained way. A continuous sequence of motion and activities was carried out like: sitting, standing up, reading books, opening drawers ... during which the 8 gestures were performed at random instants. Furthermore, there was not any rest posture between two following gestures. In this dataset, the automatic segmentation was not used because it cannot discriminate between gestures and other activities. Therefore, the gestures were segmented manually pressing the button of the wireless mouse.

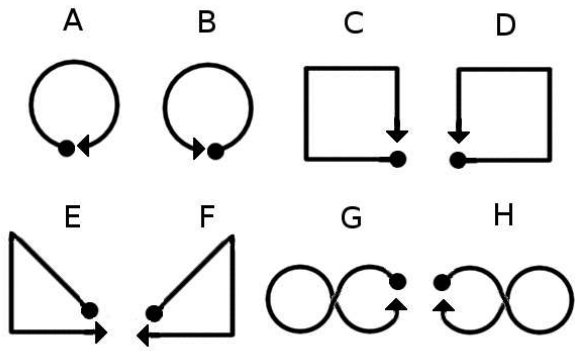

Figure 5: Gestures used to analized the performance of the method

4.1 Results

In both datasets, 5 instances randomly selected for each class of gesture were used to train the signal predictors. As it was explained before, during this training, the genetic algorithm has to minimize the fitness function. As an example of this training, figure 6 shows the evolution of the fitness function during the training of a predictor for gestures of class A. As can be seen from this figure only about 20 generations were necessary for the fitness values to reach a stable level although evolution is always performed for 200 generations.

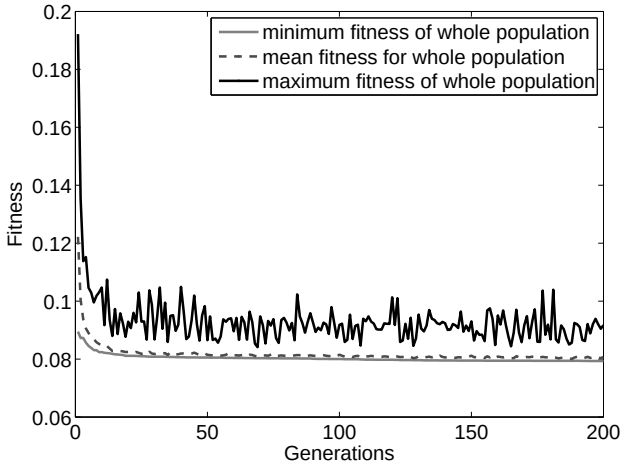

Figure 6: Evolution of the fitness function during a run of the training for the predictor of class A

Using the trained predictors of the dataset of isolated gestures, we can observe how the recognition method works. For example, figure 7 represents the prediction errors obtained in a trial for all the instances of class A. The training instances used for this trial are in the positions 2, 8, 15, 16 and 19. Most of the 20 instances are well classified because the lowest prediction errors are produced by the predictors

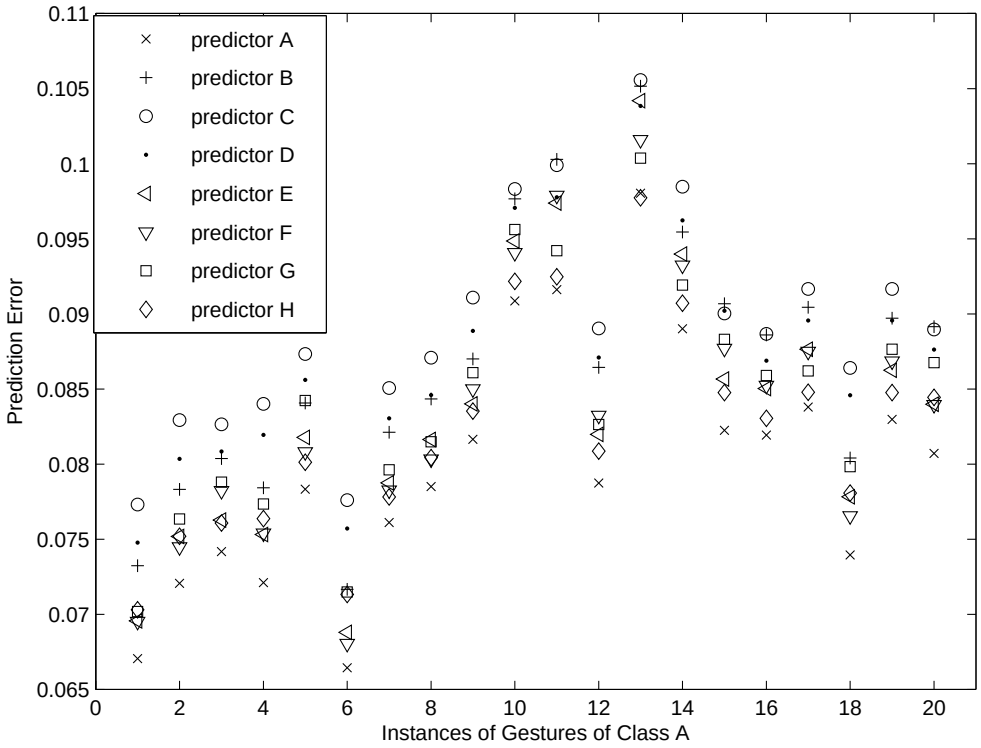

Figure 7: Prediction Errors for individual gestures of class A with the 8 class of predictors

that belongs to the same class. There is only one missclassification in 13^{th} instance, the predictor H obtains a lower error than predictor A. Furthermore, it can be observed that there is a high variability in this error for all instances. This means that it may be difficult to recognize a gesture by comparing the error of its corresponding predictor to an absolute threshold. However in this experiment recognition seems possible due to the comparison of the errors produced by different predictors between them.

To measure the performance of the classifiers for both datasets, a confusion matrix is used. This matrix illustrates the result of the classification. It has as many rows and columns as a classes of gestures. The element placed in the row i and column j indicates the percentage of gestures of type j that are classified as gestures of type i. Therefore, correctly classified gestures will be in the diagonal of the matrix. In the following results each element of the tables have two numbers. The first and second numbers indicate the classification accuracy of the training and testing set respectively.

Optimizing the predictors with a genetic algorithm is a stochastic process. In order to alleviate variability, the experiment has been repeated 10 times with different set of evolved predictors. Furthermore, for each trial the 5 training instances of each predictor were randomly selected. The confusion matrix is the average of this 10 repetitions.

The table 2 shows the confusion matrix for the isolated gestures. A high recognition rate of 98% was obtained for the training set and 94% for the testing set. Another interesting result of this test is that the gestures C and D are sometimes missclassified between them. One explanation may be that on the second and fourth segment both gestures go up and down, respectively. So, this gestures are quite similar during about 50% of the length of the gesture.

The table 3 shows the confusion matrix for the gestures captured in a realistic environment. The total recognition rate for training instances is about 80.5% and for testing instances is 63.6 %. So, there is a noticeable decrease in performance compared to the previous dataset which was recorded in a more constrained way. Besides, the recognition accuracy for the various classes differs much. For example, the recognition rate for class B is very high while for classes A,D and F is relatively low. Lastly, it is interesting to remark that the misclassification between C and D also appears in this dataset and some more like G and H.

Lastly, we noticed that these neural networks presented a robust behaviour for both datasets. They never got saturated or locked in oscillatory patterns, although the predictors were fed directly with the whole signal without resetting the activations between different gestures.

5. DISCUSSION

There is a significant difference between performance of the 2 considered datasets. This could be explained by the higher noise in the signal caused by the person movements and activities in the second dataset. Furthermore, due to the movement it is usual that the device is not held exactly in the same orientation for all gestures. These little variations

Classes	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
A	100.0/96.7	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0
B	0.0/0.0	98.0/95.3	0.0/0.0	0.0/0.0	0.0/2.7	0.0/0.0	0.0/0.0	0.0/0.7
C	0.0/0.0	0.0/0.0	94.0/79.3	6.0/9.3	0.0/0.7	0.0/0.0	0.0/0.0	0.0/2.0
D	0.0/0.0	0.0/0.0	6.0/20.7	92.0/90.7	0.0/0.7	0.0/0.0	0.0/0.0	0.0/2.7
E	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	100.0/96.0	0.0/0.0	0.0/0.0	0.0/0.0
F	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	100.0/100.0	0.0/0.0	0.0/0.0
G	0.0/0.0	2.0/4.7	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	100.0/100.0	0.0/0.0
H	0.0/3.3	0.0/0.0	0.0/0.0	2.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	100.0/94.7

Table 2: Confusion matrix for isolated gestures

Classes	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
A	72.0/50.7	0.0/0.0	2.0/3.3	0.0/0.0	0.0/4.0	18.0/14.0	2.0/0.0	0.0/6.0
B	0.0/0.0	100.0/92.7	0.0/0.0	0.0/0.0	6.0/9.3	0.0/0.0	0.0/0.0	0.0/0.0
C	4.0/16.7	0.0/0.0	88.0/64.0	18.0/34.0	4.0/6.0	10.0/14.7	0.0/0.0	0.0/0.0
D	0.0/0.7	0.0/0.0	6.0/25.3	68.0/48.7	8.0/8.0	2.0/4.0	0.0/0.7	0.0/0.0
E	0.0/6.7	0.0/2.0	2.0/6.0	14.0/16.7	76.0/68.7	0.0/0.0	0.0/0.0	0.0/0.7
F	14.0/12.0	0.0/1.3	2.0/1.3	0.0/0.7	4.0/2.7	62.0/48.0	6.0/8.0	0.0/5.3
G	0.0/1.3	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	4.0/9.3	86.0/66.7	8.0/18.7
H	10.0/12.0	0.0/4.0	0.0/0.0	0.0/0.0	2.0/1.3	4.0/10.0	6.0/24.7	92.0/69.3

Table 3: Confusion matrix for gestures captured in a realistic environment

produce large differences in the signal because, due to the effect of gravity, the signal provided by the accelerometers depends highly in the orientation of the sensor. This could be solved using the gravity to estimate the device orientation and then reorienting the acceleration vector [18].

Another reason for the difference in performance could be the less accurate hand-made segmentation used for the gestures captured in the realistic environment. This coarse segmentation may include parts of the signal that do not belong to the gesture. During training, these parts are also used to train the predictor and this could produce the lower recognition rate. Therefore, a main objective of future work is the search of an automatic segmentation method that also works for noisy datasets. An initial idea for this, could be to detect periods of the signal where there is activity using some features of the acceleration signal like the standard deviation or the magnitude.

In previous work we investigated predictors based on Neuro Fuzzy systems [1]. Each predictor consisted of a fuzzy rule based system that only forecast one axis of the acceleration signal. So, for each gesture, three predictors were needed (one for each axis). The inputs were the previous values of the acceleration signal (t-8, t-16, t-24) of one axis and its output was the predicted signal in current time (t) for that axis. Using this approach, high recognition rates were obtained however it presents some limitations. On the one hand, the distribution in time of the previous values of inputs favours the prediction of specific frequencies of the signal. On the other, if there are correlations between the signals of different axis, this predictor cannot use them because it only has information of one axis. Lastly, these fuzzy predictors do not present any time dynamics. Therefore, these problems and the search for a more general predictor were the reasons that motivated us to look for another kind of predictors

Neural networks have been used in previous research to perform gesture classification [19, 4, 16]. In these approaches complex hierarchical neural networks are often used (Kohonen map combined with a recurrent network in [4], multi-network approach in [16]). Although this added complexity may show benefits in terms of recognition accuracy, our approach was to use a comparatively simple and compact network, with the main motivation of minimizing the computational requirements for implementation in miniature wearable sensor nodes. The network that we chose for this purpose however allows for rich temporal dynamics which is required for the task at hand. Murakami et al. used a discrete-time recurrent neural network for gesture classification (Elman network) [19]. Although our work bears some similarity, the key advantage of CTRNN is their richer temporal dynamics which can be controlled by the neuron time constants. In addition Murakami et al. relied on absolute as well as relative hand position for gesture recognition. This requires a complex sensing device: here inexpensive accelerometers are directly used instead.

To train the predictor, the fitness function that is currently used tends to generate CTRNN that maximize the prediction accuracy of gestures of their own class. As a consequence, lower fitness value means higher prediction accuracy, but may not necessarily translate into higher classification accuracy. Another fitness function may be investigated that specifically tries to maximize the recognition accuracy. This has not been investigated, because the inclusion of new gesture classes obliges to recalculate the whole system. With the current fitness function new gesture classes can be added by simply training the corresponding predictor, which is a more scalable approach.

Lastly, the system described here currently does not include a null class. In other words a gesture that is not in the training set will nevertheless be classified instead of being discarded. This is due to the classification is being only based on the election of the predictor that produces the minimum prediction error. A line of work in the future is the search for a measurement based on the prediction error that allow to classify a gesture in a null class.

6. CONCLUSION

The use of signal predictors to recognize gestures is a novel approach in this field. The results obtained in this paper and the previous ones presented in [1] show that this approach is promising and works with different predictor types: CTRNN and Neuro Fuzzy Systems, respectively. The predictors have interesting features especially for real time systems. They are fast, simple and modular which allows to incorporate easily new gestures in the recognition method.

CTRNN provides a more general predictor that can deal with the dynamics of the gesture signal. The classification of 8 gestures of an isolated dataset achieved very high success rates (94%). In an unconstrained recording the recognition accuracy was 64%, which is nevertheless a high accuracy regarding the number of gesture classes. The recognition rates reached by this method show its suitability for this problem. Besides, these neural networks presented a robust behaviour during gestures prediction because they did not saturate or get locked in oscillatory behaviors.

One of the limitations of this recognition method is its dependency on segmentation and device orientation, but this will be improved in future work. In addition, although changes in device orientation was shown to decrease recognition accuracy, this problem is shared by other methods relying on acceleration data to perform classification.

A comparison of recognition performance with different predictors and different dataset complexity is needed in order to quantify the benefit of using a generic predictor such as a CTRNN instead of simpler Neuro-Fuzzy systems. Furthermore, it is necessary to validate this approach using data from different subjects in order to prove the system robustness because all experiments of this paper were done only by one person. These points remain the object of future work.

7. ACKNOWLEDGEMENTS

The work presented in this paper was completely carried out at the ETH Wearable Computing Lab. The first author would like to thank Universidad Politecnica de Madrid for their support during the stay at this laboratory.

8. REFERENCES

[1] G. Bailador, G. Trivino, and S. Guadarrama. Gesture recognition using a neuro-fuzzy predictor. In *International Conference of Artificial Intelligence and Soft Computing*. Acta press, 2006.

[2] R. D. Beer. On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 13(4):469–509, 1995.

[3] J. Blynel. Evolving Reinforcement Learning-Like Abilities for Robots. In *5th International Conference on Evolvable Systems (ICES'03)*, 2003. In A. Tyrrell, P.C. Haddow, and J. Torresen (eds.).

[4] K. Boehm, W. Broll, and M. Sokolewicz. Dynamic gesture recognition using neural networks; a fundament for advanced interaction construction. In *Proceedings of the SPIE*, 1994.

[5] G. Chambers, S. Venkatesh, G. West, and H. Bui. Hierarchical recognition of intentional human gestures for sports video annotation. In *Proc. 16th IEEE Conference on Pattern Recognition*, pages 1082–1085, 2002.

[6] J. Cracknell, A. Y. Cairns, C. Ramsay, and I. W. Ricketts. Gesture recognition: an assessment of the performance of recurrent neural networks versus competing techniques. In *IEE Colloquium Applications of neural network to signal processing*, pages 8/1–8/3, 1994.

[7] A. K. Dey and G. D. Abowd. Towards a better understanding of context and context awareness. Technical Report GITGVU-99-22, Georgia Tech, 1999.

[8] T. Frantti and S. Kallio. Expert system for gesture recognition in terminal's user interface. *Expert Systems with Applications*, 26(2):189–202, February 2004.

[9] K. Funahashi and Y. Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6:801–806, 1993.

[10] J. C. Gallagher and J. M. Fiore. Continuous time recurrent neural networks: a paradigm for evolvable analog controller circuits. In *The Proceedings of the National Aerospace and Electronics Conference*, pages 299–304. IEEE Press, 2000.

[11] D. E. Goldberg. *Genetic Algorithms in Search Optimization & Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[12] J. M. Hausdorff, C.-K. Peng, A. L. Goldberger, and A. L. Stoll. Gait unsteadiness and fall risk in two affective disorders: a preliminary study. *BMC Psychiatry*, 4(39), 2004.

[13] C. F. Juang and K.-C. Ku. A recurrent fuzzy network for fuzzy temporal sequence processing and gesture recognition. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 35(4):646–658, 2005.

[14] H. Kang, C. W. Lee, and K. Jung. Recognition-based gesture spotting in video games. *Pattern Recognition Letters*, 25(15):1701–1714, 2004.

[15] M. Kanis, N. Winters, S. Agamanolis, A. Gavin, and C. Cullinan. Toward wearable social networking with iBand. In *CHI '05 - Human factors in computing systems*, pages 1521 – 1524, 2005.

[16] M. V. Lamar and A. Bhuiyan, S. Iwata. T-CombNET - a neural network dedicated to hand gesture recognition. In *Proc. of Biologically Motivated Computer Vision (BMCV)*, pages 613–622, Heidelberg, 2000. Springer-Verlag.

[17] P. Lukowicz, H. Junker, M. Staeger, T. von Bueren, and G. Troester. WearNET: A distributed multi-sensor system for context aware wearables. In G. Borriello and L. Holmquist, editors, *UbiComp 2002: Proceedings of the 4th International Conference on Ubiquitous Computing*, pages 361–370, Heidelberg, Sept. 2002. Springer.

[18] D. Mizell. Using gravity to estimate accelerometer orientation. In *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, page 252, Washington, DC, USA, 2003. IEEE Computer Society.

[19] K. Murakami and H. Taguchi. Gesture recognition using recurrent neural networks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 237–242, 1991.

[20] W. Piekarski and B. H. Thomas. Tinmith-Metro: New outdoor techniques for creating city models with an augmented reality wearable computer. In *Proc. of the*

5th International Symposium on Wearable Computers, pages 31–38, Los Alamitos, CA, 2001. IEEE Computer Society Press.

[21] D. Roggen, N. B. Bharatula, M. Stäger, P. Lukowicz, and G. Tröster. From sensors to miniature networked sensorbuttons. In *Proc. of the 3rd Int. Conf. on Networked Sensing Systems (INSS06)*, pages 119–122, San Diego, CA, 2006. Transducer Research Foundation.

[22] G. S. Schmidt and D. H. House. Towards model-based gesture recognition. In *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, Washington, DC, USA, 2000. IEEE Computer Society.

[23] M. Sung, C. Marci, and A. Pentland. Wearable feedback systems for rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 2(17), june 2005.

[24] C. Verplaetse. Inertial proprioceptive devices: self-motion-sensing toys and tools. *IBM Syst. J.*, 35(3-4):639–650, 1996.

[25] B. M. Yamauchi and R. D. Beer. Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior*, 2(3):219–246, 1994.