Gestures as Interface for a Home TV Digital Divide Solutions through Inertial Sensors

Stefano Pinardi, Matteo Dominoni

D.I.S.Co. University of Milano-Bicocca, Italy
pinardi@disco.unimib.it; dominoni@disco.unimib.it

Abstract. Seniors are the fastest growing segment of populations not only in many parts of Europe, but also in Japan and the United States. ICT technologies are not very popular among many elderly and also are not designed around their cultural necessities and ergonomic needs. The risk is that in the very near future this growing segment will be digitally isolated, in a society that is more and more based on ICT as infrastructure for service, and communications.

Easy Reach Project proposes an ergonomic application to break social isolation through social interaction to help the elderly to overcome barrier of the digital divide. This paper focuses its attention on the development of the technology and algorithms used as Human Computer Interface of the Easy Reach Project, that exploits inertial sensors to detect gestures.

Many experimental algorithms for gesture recognition have been developed using inertial sensors in conjunction with other sensors or devices, or by themselves, but they have not been thoroughly tested in real situations, they are not devoted to adapt to the elderly and their way of executing gestures. The elderly are not used to modern interfaces and devices, and – due to aging – they can face problems in executing even very simple gestures.

Our algorithm based on Pearson index and Hamming distance for gestures recognition has been tested both with young and elderly, and was shown to be resilient to changes in velocity and individual differences, still maintaining great accuracy of recognition (97.4% in user independent mode; 98.79% in user dependent mode). The algorithm has been adopted by the Easy Reach consortium (2009-2013) to pilot the human machine gesture-based interface.

Keywords: Sensors, Inertial Sensors, Gestures, Human Machine Interfaces, Ambient Intelligence, Assisted Living, Elderly, Social, Digital Divide, Home, TV.

1 Introduction

The goal of this work is to develop an innovative and low cost HMI (Human Machine Interface) to exploit gestures to control a TV Set for smart applications. The HMI interface is part of the "Easy Reach" project, partially supported by AAL JP (Ambient Assisted Living Joint Program). The Easy Reach Project has the aim to help the elderly to interact with each other using ICT technology to break social isolation while at

the same time coping with the ergonomic and cultural difficulties related to aging and the digital divide [1]. Seniors are the fastest growing part of the population in Europe [2], in Japan and in the United States: on one side ICT technologies are not popular among many elderly and on the other ICT applications are not designed around their mentality in order to solve their needs [1]. The risk is that in the very near future the elderly - who are becoming the most numerous segment of the population - will be digitally isolated in a society that is more and more based on ICT as infrastructure both for service, and communications. Easy Reach proposes a solution for social interaction, which is easy from the ergonomic point of view, that help the elderly break the barrier of the digital divide barrier. We use a device with inertial sensors that is handled by the user, to interpret the gestures that pilot the interface. This paper focus his attention about these algorithms and their accuracy.

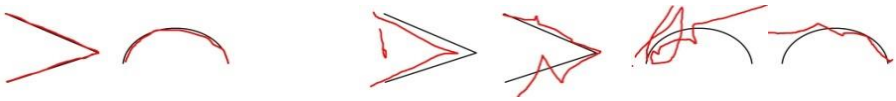

Fig. 1. Gestures: young vs elderly. In black the ideal gesture in red the actual execution. More variations, noise and errors are present in elderly gestures.

The general purpose is to identify gestures considering the time feature, normalizing acquisitions, and making the system robust to variations in gesture execution, introduced both by individual variance and senile deterioration, while maintaining a high accuracy. As we shall see, the proposed method achieves an accuracy of 97.39% on the dataset 1 (8 gestures) in user *independent* mode and an accuracy of 98.79% in user *dependent* mode on the same dataset (both with a threshold of 58%), exploiting only inertial sensors information. The algorithm has been adopted by the Easy Reach consortium to pilot the "Easy Reach" Human Computer Interface.

2 Former Works

One of the first dynamics gesture recognition systems based on HMM and inertial sensors was created by Hofmann et al. in the mid-90s [3]. The application included the use of discrete Hidden Markov Models to reduce complexity, but the recognition algorithm took hours to arrive to its end. The work of Mäntylä et al. [4] is one of the first that uses only accelerometers to recognize both static and dynamic gestures using a sensor box installed on a portable device. The algorithms used for the recognition exploit the HMM and SOM (Self Organizing Map) of Kohonen, the first for dynamics gesture, the second for static ones. The recognition accuracy of the dynamic gestures is quite high, around 97% on average; it is to emphasize that this system use a dataset based on only two people1.

[1] It must be highlighted that results presented must be weighed according to dimension, complexity and cardinality of the dictionaries.

In more recent works Schlömer et al. [5] created a classifier for four distinct gestures using a Nintendo Wiimote exploting a K-means, HMM, and Bayes classifier pipeline, reaching an accuracy of 89.72% on average (84.0%-93.4%). Prekopcsák [6] has an accuracy of 97.4% using an HMM, and 96.0 % using SVM (Support Vector Machines) interpreting the accelerometers data of a Sony Ericcson W910i. These accuracies are high, unfortunately very few is explained about the gestures used, a part from the fact that they involve the wrist and the arm, that is not very informative. Also, these results are obtained on datasets of only four different users. One interesting result is obtained by [7] which has an accuracy of 99.2% on average, using a Bayesian network model on a dataset of 13 different classes of gestures created by 15 people. These results are influenced by an appropriate choice of form of the gestures, adaptably modified to make them easily discernible from each other. For example, the number 7 and 1 are designed to be easily separable, and the number 4 is written not in the usual natural way. Another notable work is [8] that identifies the feature in the frequency domain. The authors reach an accuracy of 98.93 % in a group of 4 gestures, and an accuracy of the 89.29% in a set of 12 gestures using a method called FDSVM (Frame-based Descriptor and multi-class SVM) in user independent mode. In the document it is also shown that DTW (Dynamic Time Warping) and Naive Bayes have lower accuracy than FDSVM in groups of gestures of larger cardinality. Regarding recent papers that describe recognition methods based only on inertial data, we can report the work of Kratz and Rohs [9] that have the accuracy of 80% using 10 gestures created by 12 users, and Chen et al. [10] with an accuracy of 98.8% in user dependent mode using only inertial sensors (implicit) and 85.24 % in user independent mode using only the inertial sensors.

3 Euler Angles

Inertial sensors usually provide two different types of data i) acceleration, angular velocity, magnetic north, or ii) Euler angles. The first type of information is useful to determine the strength and quality of a gesture, for example, to verify the presence of tremors, apparent forces, peaks (e.g. for the identification of falls). This type of data are particularly suited to recognize the dynamic aspects of an action and its quality (cfr. [4] [5] [11]). In this work we use only the Euler angles to detect and classify the executed gestures.

3.1 Individual Variance and Noise

The execution of a gesture is conditioned by three factors:

- Thermal noise: each sensor produces a Gaussian noise due to temperature and electromagnetic fluctuations.
- Position of the sensor: slightly differently modifications of device handling or sensor position sensitively affect data of tilting and positioning of the sensor
- Model noise (articulated body complexity): the body varies and changes from day to day and react differently all the time, this introduces a significant change in the

execution of any gesture. The same gesture repeated in consecutive times, even by the same person, with the same device positioned in the same way, always looks different (see **Fig. 2**). It is the same effect we have when one person signs his name on paper.

Fig. 2. On the left: same gesture executed by different people. Right: the same gesture executed by the same person. Every gesture is different even if repeatedly executed by the same person.

4 Correlation Methodologies

To correlate gestures we test three different algorithms. The first exploits a Pearson correlation on Euler Angles' Yaw and Pitch. The other two are Hamming and Levenshtein distance. To increase the accuracy we have combined Pearson alternatively with one of the other two algorithms. The correlation algorithms are used both during the extraction phase of the centroids[2] , and the recognition phase. Any new gesture is compared with the previously extracted centroids, and the higher scored centroid is referred to as "the recognized gesture". If we do not pass a certain score of confidence (see 5.1, Rejection Threshold) the gesture is consider invalid, this is interpreted as a poorly performed gesture or as a non-intentional gesture.

4.1 Pearson Correlation

This measure expresses a linearity correlation between the covariance of two random variables and the product of their standard deviations. The coefficient range in the interval [-1, 1], where 1 indicates a complete correlation between the two variables, and -1 indicates the random variables are inversely related. The higher the correlation, the more probable it is that two gestures are reciprocally similar.

4.2 Hamming and Levenshtein Distance

The Hamming and Levenshtein distance are the well know algorithms for string comparison, used to calculate the grade of difference of two gesture. The algorithm divides the signals in sub-segments, the segmentation algorithm find the local maxima and minima to determine beginning and end of every segment, then it builds up a string in which the four combinations of the Yaw and Pitch directions (up-up; up-down; down-up; down-down) are associated to an equivalent symbol (A,B,C,D) . Then, we use Hamming or Levenshtein to calculate the minimum number of changes required to transform one signal into the other: the resulting value reflects the edit distance between two gestures, i.e. to their "geometrical similarity".

[2] We assumes that gestures of the same class follow a Gaussian distribution with similar variance: the centroid is the gesture that has the greater intra-class similarity

4.3 Pearson-Hamming and Pearson-Levenshtein

To increase the accuracy we combine Pearson with Hamming and Levenshtein. Given a gesture, at first we "eliminate" all the dictionary centroids that are under the rejection threshold (see section 5.1) in the Pearson correlation, then we extract the best gesture assuming the Levenshtein or Hamming distance. In case of even results, we consider valid the gesture-centroid having the higher Pearson correlation.

5 Datasets

We carried out our tests on two data sets. The first one formed by 8 gestures inspired by the commands of a video player. The second one that contains 14 gestures representing the digits "0 to 9 ", and the symbols "plus", "minus", "multiply", and "divide". These two datasets have been created by 8 people between the ages 22 and 75, that performed every gesture 7 times. It was decided to acquire data without allowing the user to familiarize with the device, to simulate the same condition in which the application will be used.

In section 5.1 we present the results of tests by varying the rejection threshold (both in user dependent and user independent mode) to measure the effect on the accuracy. In section 5.3, more detailed results are presented in the form of a Confusion Matrix using the rejection threshold of 58%. In the Conclusions (see paragraph 6) we discuss our results.

Dataset 1 –Multimedia Player. The first dataset is the smallest one and is formed by a set of 8 natural gestures to interact with a video player. Legends are: (-1)-Rejection class; 1-Play, 2-Stop, 3-Previous, 4-Next, 5-Volume Up, 6-Volume Down, 7-Rewind, 8-FastForward.

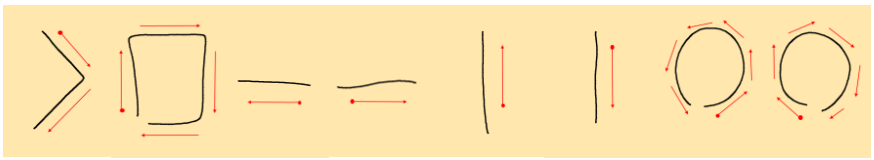

Fig. 3. The eight gestures of dataset-1 ordered 1 to 8 from left to right.

Dataset 2 – Numbers and Operations. The second dataset contains 14 gestures. Legends are: (-1)-Rejection class; 1-One, 2-Two, 3-Three, 4-Four; 5-Five, 6-Six, 7-Seven, 8-Eight, 9-Nine, 10-Zero, 11-Plus, 12-Minus, 13-Multiply, 14-Divide.

Fig. 4. The 14 gestures ordered 1 to 14 left to right. Leftmost "1-one", rightmost "14-Divide".

5.1 Rejection Threshold Test.

We state that a gesture is recognized only if it passes a given threshold T. The threshold value must be at least T > 1 / ||Dictionary|| to perform better than a random algorithm; a too high value creates too many false negatives. A reasonable number of false negative is acceptable (e.g. less than 2%) if it contributes to diminish the numbers of false positives, but we do not want the user to repeatedly redo gestures, as we prefer the use of interface to be easy and "natural".

A first test have been done on the given two datasets, changing the threshold value in a range of 48% to 68%, adding 5% at each iteration. Below we can see the most significant results related to the maximum (68%), medium (58%), and minimum (48%) value of the threshold T.

5.2 Results

The results below denote (highlighted in green) that the best accuracies are reached using a combination of Pearson and Hamming using a threshold T = 68%, in user *dependent* mode.

Threshold T	Algorithm	Set 1 Usr-Indep.	Set 1 Usr-Dep.	Set 2 Usr-Indep.	Set 2 Usr-Dep.
68%	*P*	0.9596	0.9677	0.8708	0.9262
	P-L	0.9516	0.9778	0.8642	0.9288
	P-H	0.9616	1.0	0.9064	0.9433
58%	*P*	0.9596	0.9677	0.8708	0.9262
	P-L	0.9717	0.9778	0.8554	0.9380
	P-H	0.9738	0.9879	0.9288	0.9578
48%	*P*	0.9596	0.9677	0.8708	0.9262
	P-L	0.9838	0.9778	0.8906	0.9407
	P-H	0.9516	0.9798	0.9301	0.9450

Table 1. Accuracies varying the rejection threshold T. Used combinations: P: Pearson; P-L: Pearson-Levenshtein; P-H: Pearson-Hamming.

The values in accuracy, in user *independent* mode, show a trend towards higher values as the threshold is decreased. The user *dependent* tests, instead show an opposite behavior as regards the two datasets: in fact, in the dataset 1, the recognition accuracy tends to decrease, while in the dataset 2 these values tend to increase.

On the basis of these results, we chose the Pearson-Hamming combination using a rejection threshold of 58% to promote a better balance in the dataset1 and dataset2, and in both user modes (*dependent* and *independent*), as long as we prefer maintain a certain flexibility in the dictionary choice.

5.3 Tests on Dataset1 and Dataset2 with Rejection Threshold of 58%

In this section we present more in detail the test results using dataset1 and dataset2 once fixed the threshold T = 58%. Results are presented in the form of a Confusion Matrix. Ground truth (GT) are in columns, while rows represent what has been actually recognized (R). In diagonal the correct results (true positive, in green), outside the diagonal the invalid recognitions: in the leftmost column the false negatives (in blue), out of this column and not in the diagonal the false positives (red).

GT\R	-1	1	2	3	4	5	6	7	8
1	0	70	0	0	0	0	0	0	0
2	0	1	54	0	0	1	2	0	0
3	0	0	0	52	0	1	2	0	0
4	0	0	0	0	58	1	0	0	0
5	0	0	0	0	0	58	0	0	0
6	3	0	0	0	1	0	55	0	0
7	0	0	0	0	0	0	0	69	0
8	1	0	0	0	0	0	0	0	67

GT\R	-1	1	2	3	4	5	6	7	8
1	0	70	0	0	0	0	0	0	0
2	0	0	58	0	0	0	0	0	0
3	0	0	0	53	0	0	0	2	0
4	0	0	0	0	58	0	0	0	1
5	0	0	0	0	0	55	0	1	2
6	0	0	0	0	0	0	59	0	0
7	0	0	0	0	0	0	0	69	0
8	0	0	0	0	0	0	0	0	68

Table 2. Dataset 1: Pearson-Hamming. Left user-independent; right user-dependent mode

In **dataset 1** (8 gestures) the best combination using threshold T=58% in user *independent* mode is "Pearson and Hamming". *Accuracy* is 98.79%, precision is 98.17% , recall is 99.178% . We have only 4 false negatives out of 496 instances (cfr. **Table 2**, left). In user *dependent* mode the *accuracy* is 98.79% , precision is 98.79% , recall is 100% . We do not have any false negative (cfr. **Table 2**, right).

GT\R	-1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	75	0	1	0	0	0	0	0	0	0	0	0	0	0
2	0	2	45	1	0	0	0	0	0	0	0	1	0	0	0
3	0	0	1	47	0	0	0	0	0	0	2	0	0	1	0
4	0	1	0	0	49	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	50	2	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	49	0	0	0	0	0	0	0	0
7	0	1	1	2	0	0	0	49	0	0	0	0	0	1	0
8	2	0	0	0	0	0	0	0	47	0	0	0	0	1	0
9	0	0	0	0	0	1	0	0	0	42	0	3	0	1	0
10	0	0	0	0	0	0	0	1	1	0	59	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	58	0	0	0
12	0	0	0	0	0	0	0	1	0	0	0	0	50	0	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	52	2
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55

Table 3. Dataset 2: Pearson – Hamming. User-dependent mode.

In **dataset 2** (14 gestures) the best combination using a threshold T= 58% in user *independent mode* is the "Pearson and Hamming", again. The *accuracy* is 92.885%, precision is 94.631% , recall is 98.052%. In user *dependent* mode the *accuracy* is 95.784%, precision is 96.037%, recall is 99.725% (see **Table 3** for user dep. mode)

6 Conclusions

Comparing our algorithm to others using only inertial systems in user *dependent* mode our recognitions algorithms obtain a better accuracy. The work of Kratz and Rohs 2010 [9] achieves an accuracy of 80% over a 10-gestures dictionary. Our algorithm appears also better in comparison with Chen et al. [10]. In the user *independent* mode their ranking algorithm obtains an accuracy of 85.24% without using optical sensors, against our accuracy of 97.4%. In user *dependent* mode their error rate is much lower, and their accuracy is 98.8% using only the inertial sensors, against our 98.79% (but using threshold 58%). Using a rejection threshold of 68% in user *dependent* mode we reached an accuracy of 100%, on the dataset1 of 8 gestures (see **Table 1**).

Our accuracy results can be also compared to systems that use more complex classification methods such as SVM , SOM and HMM that are in principle more effective, but do not allow a recalculation at "real time " of the class representative (centroid). Wu et al. [8] show that their classification algorithm has good results using a set of gestures that are very similar to our dataset. They have an accuracy of 99.38% in user *dependent* tests on a dataset of only 4 gestures, against our 100% reached using P-H with a threshold 68% on dataset1 (8 gestures) in user *dependent* mode (see **Table 1**). The classifier of Wu et al. has an accuracy of 95.21% on a dataset of 12 gestures, against our 95.78% of accuracy on the dataset2 (but with 14 gestures) in user *dependent* mode (see **Table 3**). Even in user *independent* tests our algorithm outperforms Wu et al. results: we reach an accuracy of 92.88 % on the dataset2 (14 gesture) against the 89.29% obtained by Wu et al. on their dataset of 12 gestures.

In conclusion, our classification algorithm obtains good results and performs better compared to analog recognition systems based only on inertial sensors and also gives better results than most of the multimodal works we found in the literature, reaching accuracies useful for a commercial device. The algorithm have quick time of reaction (less 0.1 sec), it is flexible, and resilient both to individual variations and to differences introduced by ageing. The algorithm has been adopted by the Easy Reach consortium to pilot the human machine gesture-based interface of the Easy Reach Project.

References

1. R. Bisiani, D. Merico, S. Pinardi, M. Dominoni, A. Cesta, A. Orlandini, R. Rasconi, M. Suriano, A. Umbrico, O. Sabuncu, T. Schaub, D. D'Aloisi, R. Nicolussi, F. Papa, V. Bouglas, G. Giakas, T. Kavatzikidis and S. Bonfiglio, "Fostering Social Interaction of Home-Bound Elderly People: The EasyReach System," in IEA/AIE 2013, Amsterdam, 2013.
2. K. Hans-Helmut, H. Dirk, L. Thomas, G. R.-H. Steffi, C. A. Matthias, M. Herbert, G. Vilagut, B. Ronny, M. H. Josep, D. G. Giovanni, D. G. Ron, K. Viviane and A. Jordi, "Health status of the advanced elderly in six european countries: results from a representative survey using EQ-5D and SF-12," Vols. Health and Quality of Life Outcomes 2010, 8:143, no. 8, p. 143, 2010.
3. F. G. Hoffman, P. Heyer and G. Hommel, "Velocity Profile Based Recognition of Dynamic Gestures with Discrete Hidden Markov Models," 1996.
4. V. M. Mäntylä, J. Mäntyjärvi, T. Seppänen and E. Tuulari, "Hand gesture recognition of a mobile device user," IEEE, 2000.
5. T. Schlömer, B. Poppinga, N. Henze and S. Boll, "Gesture Recognition with a Wii Controller," in Proceedings of the Second International Conference on Tangible and Embedded Interaction, Bonn, 2008.
6. Z. Prekopcsák, "Accelerometer Based Real-Time Gesture Recognition," Poster, 2008.
7. S. J. Cho, J. K. Oh, W. C. Bang, W. Chang, E. Choi, Y. Jing, J. Cho and D. Y. Kim, "Magic Wand: A Hand-Drawn Gesture Input Device in 3-D Space with Inertial Sensors," in Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition, 2004.
8. J. Wu, G. Pan, D. Zhang, G. Qi and S. Li, "Gesture Recognition with a 3-D Accelerometer," Springer, 2009.
9. S. Kratz and M. Rohs, "A $3 Gesture Recognizer – Simple Gesture Recognition for Devices Equipped with 3D Acceleration Sensors," ACM, 2010.
10. M. Chen, G. AlRegib and B. Juang, "A new 6D motion gesture database and the benchmark results of feature-based statistical recognition," 2011.
11. S. Pinardi and R. Bisiani, "Movements Recognition with Intelligent Multisensor Analysis, a Lexical Approach.," in Proceedings of the 6th Int. Conf. on Intelligent Environments, Kuala Lumpur, 2010.
12. S. Gupta, D. Morris, S. N. Patel and T. Desney, "SoundWave: Using the Doppler Effect to Sense Gestures," Redmond, 2012.
13. R. Xu, S. Zhou and W. J. Li, "MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition," IEEE SENSORS JOURNAL, 5 May 2012.
14. L. Kratz, T. S. Saponas and D. Morris, "Making Gestural Input from Arm-Worn Inertial Sensors More Practical," ACM, 2012.
15. XSens, XM-B Technical Documentation, 2009.
16. STMicroelectronics, "LIS331DLH - MEMS digital output motion sensor ultra low-power

high performance 3-axes “nano” accelerometer," 2009.

17. S. Zhou, Z. Dong, W. J. Li and C. P. Kwong, "Hand-Written Character Recognition Using MEMS Motion Sensing Technology," in Proceedings of the 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2008.
18. P. Keir, J. Elgoyhen, M. Naef, J. Payne, M. Horner and P. Anderson, "Gesture-recognition with Non-referenced Tracking," in Proceedings of the 2006 IEEE Symposium on 3D User interfaces, 2006.
19. P. Fihl, M. Holte, T. Moeslund and L. Reng, "Action Recognition using Motion Primitives and Probabilistic Edit Distance," 2006.
20. T. Pylvänäinen, "Accelerometer Based Gesture Recognition Using Continuous HMMs," Springer-Verlag Berlin Heidelberg, 2005.
21. E. Tuulari and A. Ylisaukko-oja, "SoapBox: A Platform for Ubiquitous Computing Research and Applications," Springer-Verlag Berlin Heidelberg, 2002.
22. C. Vogler, H. Sun and D. Metaxas, "A Framework for Motion Recognition with Applications to American Sign Language and Gait Recognition," IEEE, 2000.
23. D. M. Gavrila, "The Visual Analisys of Human Movement: A Survey," Academic Press, 1998.
24. J. S. Wang and F. C. Chuang, "An Accelerometer-Based Digital Pen With a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition," IEEE, 2011.
25. E. Choi, W. Bang, S. Cho, J. Yang, D. Kim and S. Kim, "Beatbox Music Phone: Gesture-based Interactive Mobile Phone using a Tri-axis Accelerometer," IEEE, 2005.