Using Clustering Techniques for Intelligent Camera-based User Interfaces

Zorana Banković, José M. Moya, Elena Romero, Javier Blesa, David Fraga, Juan Carlos Vallejo, Álvaro Araujo, Pedro Malagón, Juan-Mariano de Goyeneche, Daniel Villanueva, Octavio Nieto-Taladriz

Abstract. The area of Human-Machine Interface is growing fast due to its high importance in all technological systems. The basic idea behind designing Human-Machine interfaces is to enrich the communication with the technology in a natural and easy way. Gesture interfaces are a good example of transparent interfaces. Such interfaces must identify properly the action the user wants to perform, so the proper gesture recognition is of the highest importance. However, most of the systems based on gesture recognition use complex methods requiring high-resource devices. In this work we propose to model gestures capturing their temporal properties, which significantly reduces storage requirements, and use clustering techniques, namely self-organizing maps and unsupervised genetic algorithm, for their classification. We further propose to train a certain number of algorithms with different parameters and combine their decision using majority voting in order to decrease the false positive rate. The main advantage of the approach is its simplicity, which enables the implementation using devices with limited resources, and therefore low cost. The testing results demonstrate its high potential.

Keywords. Gesture recognition, intelligent environments, self-organizing maps, unsupervised genetic algorithm

1 Introduction

Human-machine interaction has been the subject of intense research over the past few decades. According to the trends of ubiquitous computing, the human-computer interaction must be designed as naturally and as easily as possible, without resulting in the perception of an intrusive technology.

User interaction should be designed in a way that does not require the user to adapt to special conventions or rules, but the environment should be the one to adapt to the natural way of user interaction. In the recent past, new natural and flexible interfaces, embedded in the objects the people use on everyday basis, have been developed. These new interfaces have been designed and adapted for the end users' needs. One of

the most natural and comfortable ways to interact with the environment is by hand gestures, as vision is an attractive tool since it is rich in information that is provided in a low-cost and non-intrusive manner.

Nowadays, there are a number of tools that are able to track a person's movements and find out what gestures are being performed. Although there is a large amount of research done in image and video based gesture recognition, the variety of the tools used for this purpose is not very wide, and they fall into one of the following groups:

- *Depth-aware cameras*. Specialized cameras, such as time-of-flight (TOF) cameras, that provide depth information can be used to approximate a 3D representation of the hand gestures. The main advantage of this approach is easy segmentation of the hand from its environment [16].
- *Stereo cameras*. The outputs of two cameras with know interrelation can also be used to approximate 3D representations of the monitored objects.
- *Controller-based gestures*. This approach assumes existence of additional controllers that act as body extensions, and the gestures are captured by software. A typical example is the Wii Remote, which uses changes in time acceleration to represent gestures [17]. However, these extensions do not fulfill the necessity of transparency, since the user has to hold or wear them while performing gestures.
- *Single camera*. A normal camera can also be used for gesture recognition. Although not necessarily as effective as stereo or depth aware cameras, their cost is much lower which is the decisive factor when it comes to the massive deployment, such as in the ambient intelligence systems.

For the above reasons, we chose to use a single camera for capturing and distinguishing hand gestures. However, most of the systems based on gesture recognition use complex methods or algorithms, that require high-resource devices to be efficiently performed. Thus, a lot of work has been done on gesture recognition using general purpose computers. However, we have to take into account that the embedded systems connected to the cameras usually exhibit very limited resources. We need to do as much processing as possible in the camera processor to try to reduce the spectrum occupation, but we must be aware that this may lead to the increase of the camera processor resource consumption. Bearing this in mind, the main advantage of our proposal is its simplicity and its low resource consumption. This permits us to perform our algorithm on a device with limited resources, and therefore low cost, as embedded systems usually are.

In this article we present a low-cost gesture interface that can control different systems of an environment with simple and fast processing in the embedded systems, minimizing the need for communications. First we propose to model gestures capturing their temporal properties, and after that we deploy clustering techniques for gesture classification. This work is extension of our previous work where we deployed self-organizing maps (SOM) algorithm for clustering the gestures [1]. SOM algorithm, as well as other typical clustering algorithms such as k-means, needs to have the number of clusters fixed from the start, when it is not possible to know the optimal value. For this reason, we have developed an unsupervised genetic algorithm (GA) [15], which in essence searches for the optimal clustering. The benefit of this

approach is that we do not have to fix the number of clusters from the beginning. Finally, we propose to have a certain number of classifiers, and make the final decision based on majority voting. Bearing in mind that we do not know the optimal parameter setting from the start, the majority voting can help overcome the issues of particular classifiers and obtain a strong classifier.

The paper is organized as follows. In Section 2 we present the previous work on the subject. Section 3 details the characterization of gestures. In Section 4 we give further details of the implementation of clustering algorithms and their combination. Finally, results are presented in Section 5 and conclusions are drawn in Section 6.

2 Previous Work

Vision-based techniques for hand gesture recognition usually take the following steps [18]: segmentation, i.e. the extraction of the hand from the background data, tracking of the motion that captures temporal properties and finally gesture recognition that can be based on a machine learning technique. The extensive survey of all the steps is given in [18].

For this reason, the common characteristic of the techniques that deploy a clustering technique [2, 3, 4 and 5] is that they are usually deployed together with another technique that captures temporal properties of the gestures. Furthermore, all of them deploy rather standard characterization that contains information such as trajectory of the hand, resultant direction of the movement, or velocity of the movement. Extraction of these features introduces additional computational overhead, and with their complexity, i.e. the need of training two learning algorithms, total computational overhead can be too high for their implementation in devices with limited resources.

On the other hand, our characterization implicitly contains above information and its calculation is straightforward. Since it also captures the temporal properties of gestures, we do not need two learning algorithms. Hand segmentation is not necessary, as we characterize the whole frame, rather than only the hand. This makes our approach less complicated and more appropriate for implementations in devices with limited resources.

A similar idea is deployed in [22], where the authors want to provide better traffic monitoring by using methods of image recognition from video sequences. However, with appropriate adjustments it could also be deployed for hand gesture recognition. The approach uses the concept of space depth and is based on modelling human vision and bio-inspired recognition techniques. The information about resource consumption is not available, so we are not able to say if it would be an appropriate solution for implementation in embedded systems. Furthermore, any comparison with hand gesture techniques cannot be provided at this stage, but the approach seems promising as it also captures temporal properties of sequences.

3 Gesture Characterization

Each gesture is captured as a set of frames of variable size, as presented in the figure below for the gesture up-down.

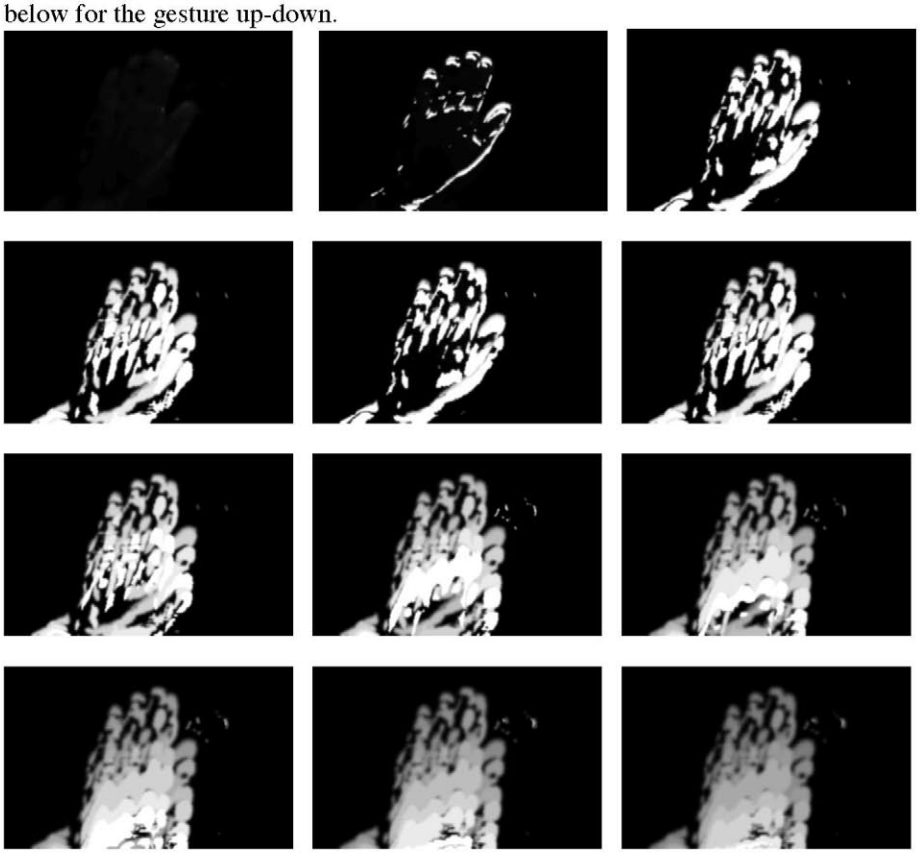

Fig. 1. : Gesture captured in 12 consecutive frames

We propose to divide each frame into (n x n) smaller parts and assign a value to each part that corresponds to its luminosity. After that we characterize temporal evolution of each part in the following way. If, for example, the consecutive values of one part are: 0 0 20 40 50 60 70 50 10, we extract the features that characterize this part with time-windows of a certain size, where the value of each feature is its frequency $\phi(x)$ in the given sequence. For time-window of size 3, the characterization would be the following:

0 0 20	0.16
0 20 40	0.16
20 40 50	0.16
40 50 60	0.16
50 60 70	0.16

70 50 10 0.16

Having in mind that the number of the features extracted in this way does not have to be fixed for every characterization period, in order to find the difference, i.e. the distance between two of them, we deploy distance function proposed in [6] that calculates distance between sequences. Further, the distance between two captured gestures is simply the sum of absolute distances of the corresponding sub-parts. In this way, the gesture characterization process implicitly captures the temporal evolution of the gesture. The following step is classification using clustering, either SOM or unsupervised GA.

4 Clustering Algorithms

4.1 Self-Organizing Maps Algorithm

Self organizing maps (SOM), also known as Kohonen networks, are an unsupervised type of neural networks [7]. Neural networks have been widely deployed in many different areas [20, 21]. As in neural networks, the basic idea of SOM has origins in certain brain operations, specifically in projection of multidimensional inputs to one-dimensional or two-dimensional neuronal structures on cortex. For example, perception of color depends on three different light receptors (red, green and blue), plus the eyes capture information about the position, size or texture of objects. It has been demonstrated that this multidimensional signal is processed by the planar cortex structure. Further, the areas of the brain responsible for different signals from the body preserve topology, e.g. the area responsible for the signals that come from the arms is close to the area responsible for the signals that come from the hand. These are precisely the basic ideas of SOM that consist in the following:

1. Multidimensional data and their dependencies are presented and captured in a lower dimension SOM network (usually 2D).
2. The proximity of the nodes in the lattice reflects similarity of the data mapped to the nodes.

For these reasons, SOMs have been widely deployed for clustering and good visualization of clustering problem. If we project the resulting clusters to RGB space, we can visualize the similarities of the adjacent clusters. SOMs have been successfully deployed in different fields such as image processing [9], robotics [9] (for both visual and motor functions), function approximation in mathematics [10], network security [11], detection of outliers in data [12], etc.

The only problem-specific point here is the centre, i.e. node representation and updating. Each centre is implemented as a collection whose size can be changed on the fly and whose elements are the features, i.e. sub-sequences defined in the previous text with assigned occurrence or frequency. The adjustment of nodes (that belong to the map area to be adjusted) is performed in the following way:

- If a feature of the input instance $v(t)$ exists in the node, its $\phi(x)$ is modified according to the centre update given in [13];

- If a feature of the instance $v(t)$ does not exist in the cluster centre, the feature is added to the centre with occurrence equal to 1.

4.2 Unsupervised Genetic Algorithm

The objective of the designed genetic algorithm is to find optimal clustering. The algorithm follows the steps of traditional genetic algorithms [15]. However, there are few aspects that are problem-specific, and thus need to be defined from the start:

- *Chromosome Codification:* Each gene in a chromosome represents a group centre. Since the optimal number of groups in a clustering problem is not known a priori, the chromosomes are implemented as lists of variable size. Each centre is presented as a collection of also variable size whose elements are the features defined above with their corresponding $\phi(x)$ value.
- *Genetic Operators:* We deploy one-point crossover, so each newly formed chromosome has at least two group centres. Also, we deploy destructive mutation which randomly eliminates an element from the list. With this we guarantee that the number of centres does not reach very high numbers.
- *Fitness Function:* Bearing in mind that optimal clustering is not known a priori, it imposes the usage of a clustering validation coefficient as fitness function. We deploy Davies-Bouldin index [14]:

$$DB = \frac{1}{n}\sum_{i=1,i\neq j}^{n} \max\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \quad (1)$$

where n stands for the number of groups, σ_i and σ_j are the average distances between groups i and j correspondingly, c_i and c_j stand for their centres, while $d(c_i, c_j)$ is the distance between the centres. Small values of DB coefficient correspond to the groups that are compact and whose respective distances between the centres are high. Thus, we define the fitness function as $1/DB$. Furthermore, in this step we adjust the group centres in the same way as performed in the case of SOM.

4.3 Detector Combination based on Majority Voting

Since it is not possible to know optimal parameter setting from the start, it is not likely to get a single optimal classifier. In order to overcome this issue, we propose to create a certain number of classifiers that are trained separately. Individual clustering algorithms are further combined using the majority voting, as it has been proven to be successful as any other classifier combination [19], although much simpler. In this way, the issues of separate classifiers get filtered and we obtain a strong classifier, that also results in lower false positive rate.

4.4 Implementation Details

Having in mind that clusters may end up having many features which would introduce significant computational overhead, we discard all the features that have at least 100 smaller value of the maximal feature value of the node, as this does not affect significantly on the final result. After having finished the training, in the current implementation we label the nodes with the label of the gesture from the set of labeled gestures that is closest to the node according to the distance function explained above. The process is depicted in Figure 2.

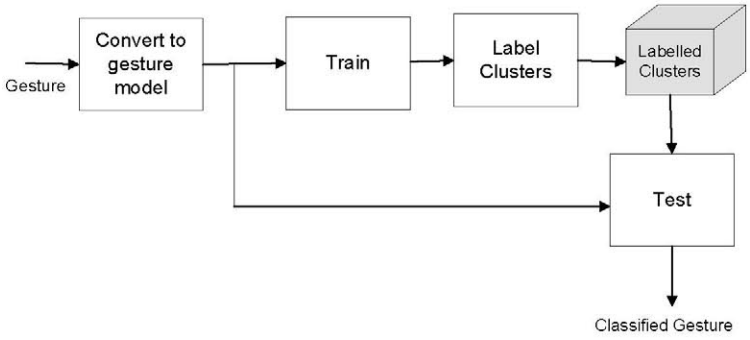

Fig. 2. The process of gesture recognition

4.5 Advantages of the Approach

The main advantage of our proposal is its simplicity. Our characterization of gestures captures the temporal evolution of the gesture, and we distinguish gestures simply by clustering them. This is another advantage, as in essence we do not have to label all the gestures (only those used for cluster labeling). Furthermore, the characterization significantly reduces the memory needed to store a gesture. For example, a captured gesture that occupies 507kB is reduced to 625B (for 5x5 division of the frame), around 1000 times. This permits us to perform our algorithm on a device with limited resources, as embedded systems deployed in ambient intelligence usually are.

5 Empirical Evaluation

5.1 Training and Testing dataset

In order to test the proposed algorithm, we have captured five types of gesture: left-right, right-left, up-down, down-up and the fifth type are random gestures labeled as unknown. Twelve persons were making the gestures and in total 760 gestures were made. In order to illustrate the memory reduction that our approach provides, we will give the numbers of occupied memories in both cases. The captured gestures occupy 1.08GB of storage space, while after the division of each frame into 5x5 blocks, they occupied 3.11MB, taking around 350 times less storage space.

5.2 Results and Discussion

We have tested our algorithm on different training and testing scenarios (by taking different portions of data explained above). We have performed testing with both 3x3 and 5x5 frame partitions. Furthermore, we have experimented with different sizes of time window mention in the Sec. 3. The main conclusion is that in general smaller window sizes (3 for example) exhibit better performances than higher.

In the following we present results of three separate algorithms and their combination based on majority voting. As we can observe, the classifiers do not perform well on every gesture, above all the detection of gestures left-right and right-left does not give satisfactory results as these have been mostly confused with each other (and sometimes with unknown gestures). Furthermore, the particularities of different users are more prominent in this case, so this case can benefit of user detection, either by additional algorithm for user detection or the existence of context information regarding users. However, the combination made by majority voting performs better than all of the separate algorithms. The results are summarized in Table 1.

Table 1. Maximal Detection Rates for Each Gesture

Gesture	Detection Rate (%)			
	SOM1	GA1	GA2	Combination
Unknown	88	100	0	100
Down-up	100	100	100	100
Up-down	92	50	100	100
Left-right	13	33	33	33
Right-left	13	50	100	50
Overall	61	63	54	76

6 Conclusions

In this work we have presented low-cost algorithm for hand gesture classification. We have proposed the characterization of gestures that captures temporal properties of gesture. In this way, we avoid the steps of gesture segmentation and tracking, as this information is implicitly contained in the extracted features. Moreover, it provides us high level of data compression, which makes possible the implementation using devices with limited resources.

We have further clustered the gestures using clustering algorithm, achieving detection rate of up to 100% for certain gestures. It has also been demonstrated that the combination of classifiers based on majority voting performs better than separate classifiers. However, it has also been observed that certain gestures exhibit user particularities, and their classification could benefit from additional user detection. In the future we plan to work more on this subject. Furthermore, we will work on different ways of classifier combination, especially the low-cost ones.

Acknowledgments. This work was funded by the Spanish Ministry of Industry, Tourism and Trade, under Research Grant TSI-020301-2009-18 (eCID), the Spanish Ministry of Science and Innovation, under Research Grant TEC2009-14595-C02-01, and the CENIT Project Segur@.

References

1. Bankovic, Z., Romero, E., Blesa, J., Moya, J. M., Fraga, D., Vallejo, J.C., Araujo, A., Malagón, P., de Goyeneche, J., Villanueva, D., Nieto-Taladriz, O.: Using Self-Organizing Maps for Intelligent Camera-Based User Interfaces. HAIS 2010, Part II, LNAI 6077, pp. 486–492, 2010.
2. Ishikawa, M., Sasaki, N.: Gesture Recognition based on SOM using Multiple Sensors. In 9th International Conference on Neural Information Processing, pp. 1300-1304, IEEE Xplore, (2002)
3. Shimada, A., Taniguchi, R.: Gesture Recognition Using Sparse Code of Hierarchical SOM. In 19th International Conference on Pattern Recognition, pp. 1-4, IEEE Xplore, (2008)
4. Caridakis, G., Karpouzis, K., Drosopoulos, A. I., Kollias, S. D.: SOMM: Self organizing Markov map for gesture recognition. Pattern Recognition Letters 31(1): 52-59 (2010)
5. Caridakis, G., Karpouzis, K., Pateritsas, C., Drosopoulos, A. I., Stafylopatis, A., Kollias, S. D.: Hand trajectory based gesture recognition using self-organizing feature maps and Markov models. ICME 2008: 1105-1108
6. Rieck, K., Laskov, P.: Linear Time Computation of Similarity for Sequential Data, in: Journal of Machine Learning Research 9 (2008) 23-48
7. Rojas, R: Neural Networks, Springer-Verlag, Berlin, 1996
8. Littmann, E., Drees, A., Ritter, H.: Neural Recognition of Human Pointing Gestures in Real Images, in: Neural Processing Letters (1996) 61–71, Kluwer Academic Publishers
9. Vleugels, J.M.,. Kok, J.N., Overmars, M.H.: A self-organizing neural network for robot motion planning, in: ICANN 93 Art. Neural Networks Conf. Proc. (S. Gielen and B. Kappen eds.), (1993) 281–284, Springer Berlin Heidelberg

10. Aupetit, M., Couturier, P., Massote, P.: Function Approximation with Continuous Self-Organizing Maps Using Neighboring Influence Interpolation, in: Proc. of Neural Computation (NC2000), Berlin, Germany, May 2000
11. Lane Thames, J., Abler, R., Saad, A.: Hybrid intelligent systems for network security, in: ACM Southeast Regional Conference. Proceedings of the 44th annual Southeast regional conference, pp: 286 - 289, (2006)
12. Muñoz, A., Muruzábal, J.: Self-Organizing Maps for Outlier Detection, in: *Neurocomputing* 18(1-3) (1998) 33-60
13. SOM Algorithm, http://www.ai-junkie.comñ/ann/som/som2.html
14. Cluster Validity Indices, http://www.biomedcentral.com/content/supplementary/1471-2105-9-90-S2.pdf
15. Goldberg, D. E. *Genetic algorithms for search, optimization, and machine learning*. 1st Ed. Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1989.
16. Kollorz, E., Hornegger, J., Barke, A.: Gesture recognition with a time-of-flight camera, Dynamic 3D Imaging, pp. 86- 93. (2007)
17. Schlömer, T., Poppinga, B., Henze, N., Boll, S.: Gesture recognition with a Wii controller. Tangible and Embedded Interaction 2008: 11-14
18. Zabulis, X., Baltzakis, H., Argyros, A. A.: Vision-based Hand Gesture Recognition for Human Computer Interaction, Chapter 34, in "The Universal Access Handbook", Lawrence Erlbaum Associates, Inc. (LEA), Series on "Human Factors and Ergonomics", ISBN: 978-0-8058-6280-5, pp 34.1 - 34.30, Jun 2009.
19. Lam, L.; Suen, S.Y.; "Application of majority voting to pattern recognition: an analysis of its behavior and performance," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on , vol.27, no.5, pp.553-568, Sep 1997
20. Jon Haitz Legarreta, Fernando Boto, Iván Macía, Josu Maiora, Guillermo García, Céline Paloc, Manuel Graña, Mariano de Blas: Hybrid Decision Support System for Endovascular Aortic Aneurysm Repair Follow-Up. HAIS (1) 2010: 500-507
21. Emilio Corchado, Ángel Arroyo, and Verónica Tricio: Soft computing models to identify typical meteorological days, Logic Journal of thel IGPL, Oxford University Press July 21, 2010 doi:10.1093/jigpal/jzq035
22. Alexander Buslaev, Marina Yashina, and Igor Kotovich: On problems of intelligent monitoring for traffic. Logic Journal of IGPL Advance Access published July 29, 2010 doi:10.1093/jigpal/jzq032