ARTIFICIAL NEURAL NETWORKS BASED SYMBOLIC GESTURE INTERFACE

C. Iacopino, Anna Montesanto, Paola Baldassarri, A. F. Dragoni and P. Puliti
DEIT, Università Politecnica delle Marche, Italy

Keywords: Gesture recognition, real time video tracking, neural networks, user interface.

Abstract: The purpose of the developed system is the realization of a gesture recognizer, applied to a user interface. We tried to get fast and easy software for user, without leaving out reliability and using instruments available to common user: a PC and a webcam. The gesture detection is based on well-known artificial vision techniques, as the tracking algorithm by Lucas and Kanade. The paths, opportunely selected, are recognized by a double layered architecture of multilayer perceptrons. The realized system is efficiency and has a good robustness, paying attention to an adequate learning of gesture vocabulary both for the user and for system.

1 INTRODUCTION

The event detection and, in particular, the human action recognition gained in last years a great interest in the computer vision e video processing research.

Gesture recognition aims to understand human gestures using mathematical algorithms; normally is focused on hands and head gesture. This subject will allow computers to understand human body language filling the existent gap between humans and machines. The uses move from virtual reality to robot guide until sign language recognizers. The purpose of the developed system is the realization of a gesture recognizer, applied in the specific case to a user interface. We tried to get fast and easy software for user, without leaving out reliability and using instruments available to common user: a PC and a webcam. We can have different uses of this technology; it can be helpful wherever there is need of a richer and more expressive interface than the existent one (textual or GUI). In this paper we aim for the realization to a symbolic gesture interface that uses a vocabulary of codified commands and known by both the user and the system. We try to make the learning and the use as simple as possible like systems based on natural gestures. The assessment of the noticed gesture at final is based exclusively on the hand actions; the biological organisms give meaning to space-time configurations and recognize the scene thank to the way the temporal movement is made.

The gesture detection is based on well-known artificial vision techniques. We use a filter to segment the regions with skin chromaticity. This help to focus the next tracking phase and to reduce the computational load. The algorithm used for the tracking is based on the feature tracking method proposed by Lucas and Kanade in 1981 (Lucas and Kanade, 1981). We follow the approach of Tomasi and Kanade (Tomasi and Kanade, 1991) that derived the criterion of features selection from a tracking performance optimization. The paths, opportunely selected, are recognized by a double layered architecture of multilayer perceptrons, a quick and robust system, able to understand the cinematic characteristics of these paths.

This paper is divided in two main parts: in the first we give a clear description of the realized system, telling about the three main components: the vision system, the recognition system and the user interface.

In the second part there is the description of real working and of system using. We tell about the codified gestures, the learning net procedure and at last the testing phase with results and problems.

Iacopino C., Montesanto A., Baldassarri P., F. Dragoni A. and Puliti P. (2008).
ARTIFICIAL NEURAL NETWORKS BASED SYMBOLIC GESTURE INTERFACE.
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 364-369
DOI: 10.5220/0001932803640369

2 SYSTEM REALIZED

The built system is essentially a symbolic gesture recognizer used for a user interface. It has to be underlined that the main aim was the creation of the recognizer; the interface is just a natural application of this technology.

The development process can be divided in three main elements:

- vision system
- artificial neural net architecture
- user interface

2.1 Vision System

The vision subsystem has to capture the frames and to process them until obtaining a pattern: a set of values understandable by the neural net. The whole process starts with the capture.

We decided to separate by time the saving phase of frames from the elaboration phase realized in the following tasks. We acted in this way to guarantee a constant and quick frame's sample, considering that the tracking process requires a changeable execution time. Once the frames are saved, the webcam turns into stand-by and the process moves to the skin color filtering.

2.1.1 Skin Filter

The use of color information was introduced in these last year to solve problems concerning the localization of hands and face with great results. It was shown the color represents very good information to extract some parts: in particular, the skin color can be used to find the hands and the face of a person. The human skin has, indeed, a peculiar distribution that significantly differs from the background.

This is extremely important because it allows you to focus the object to track in the following tracking phase. Tracking is made with no marker or referring point, so skin filter became the first way to shrink the area that has to be analyzed. This allows reducing the computational load to which the system is submitted, considering that tracking procedure is the most grave.

To realize a skin filter we decided to isolate in some way the skin tone in the color range YCbCr, this allows considering just the chroma component. The most important problem is that light condition has effect on how the color appears: let's think, for instance, to a warm color artificial lighting.

The approach we used in this paper uses a rectangular mask on the bi-dimensional chromatic level. This keeps the computational load low. About the value of the thresholds we relied on results of Chai (Chai, 1999), who took an experiment on a various cluster finding a precise concentration range Cr=[133,173] e Cb=[77,127], as you can see in the following graph:

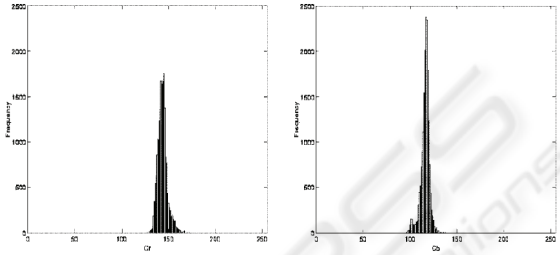

Figure 1: Component Cr e Cb of skin color.

To fix the illumination problem we choose to give entirely to the user the possibility to correct the chromatic correction that has to be applied, using a very easy and intuitive set up phase. At the end the so obtained mask is submitted to another filtering process deleting the isolate pixels. This operation is necessary to make the regions more continuous.

To increase the tracking efficiency we preferred to filter all the frames, giving a constant reference to the tracking process. The result of the elaboration is shown by the following image.

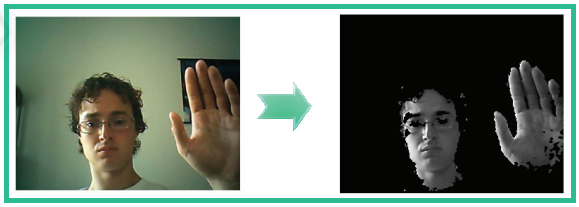

Figure 2: Frame before and after skin filter.

2.1.2 Tracking

Because of the noise it is not possible to determine the correct position of a single pixel that can be confused with adjacent points. Due to this problem we do not track single pixel but pixel window, feature. We based our work on a feature tracking method. The entire phase can be divided in three main moments.

- feature selection
- tracking
- feature recovery

The core of the algorithm is given by the tracking method thought by Lucas and Kanade. The main element of this technique is the definition of the comparison measure between characteristic, fixed dimension windows, in the two frames that have to be compared, like the square of the difference of the window intensity. Different points in a window can have different behaviours. They can have different intensity variation or, when the window is on an edge, different speed or appear and disappear. So, in the window, transformations from an image to another happen very often. We have to represent the situation using the affine motion field:

$$\delta = \mathrm{AX} + \mathrm{D} \text{ con } \mathrm{A} = \begin{bmatrix} a_{xx} a_{xy} \\ a_{yx} a_{yy} \end{bmatrix} \qquad (1)$$

where δ is in general the displacement, A is the deformation matrix and D is the translation vector (Shi and Tomasi, 1994).

To verify we are tracking always the same window, we check the dissimilarity every steps; it is defined in the following way:

$$\in = \int_w \left[I(X) - J(AX + D) \right]^2 w dX \qquad (2)$$

where I(X) e J(X) are image functions of the two frames in exam. If dissimilarity is too high the window is discarded. We obtain as result the displacement that optimizes this sum.

If we consider frames so temporally close to make the shift very short, we will have a deformation matrix very small. In this case, it can be considered null. Mistakes on the displacement can be due to the determination of these parameters in these conditions. If the aim is the shift determining, it is better to determine just the spatial components of the movement. The result is even more simple and fast to calculate.

The affine model is useful in features monitoring. In the features selection mistakes can be made; so monitoring is important to do not obtain contradictory results.

In small shifts we consider a linearization according to Taylor series of the image intensity; it allows using Newton-Raphson method for minimization.

The window feature can be rounded to the simple translation of the same in the previous frame. Besides, for the same reason, the intensity of the translated window can be written as the original one adding a dissimilarity term depending almost linearly from the displacement vector. Starting from the based solution, a few iterations are enough to make the system converge.

The Lucas-Kanade‘s algorithm is well known to be very accurate and robust. These two characteristics are in contradiction: considering the tracked window size we can assume that, to increase accuracy, we should use a smaller as possible window so that we do not loose too many details. On the other hand if we want to maintain a particular robustness during alterations of light and of the size of displacement, in particular for wide movements we have to use a bigger window.

In this paper, we use the pyramidal representation (Bouguet, 2000). In this way it is easier follow wide pixel movements that, at the level of the main image, are larger than the integration window while, in a lower level, can be confined in it. The pyramidal representation halves image resolution in each level. The algorithm starts the analysis from higher level, small images and with few details, to go down to the next level so that the accuracy will increase. This technique gives the advantage to follow wide movements; dissimilarity of displacement vector remains very small not breaking the hypothesis of Taylor series.

It is possible improve the convergence range, doing a suppression of the high spatial frequency, doing then the smoothing of the image. In this way the details will be reduced and then the accuracy: if the smoothing window is bigger than the size of the subject that you want to track, this will be totally erased. Smoothing will be applied to higher level image, the one with low resolution, so that the information will not be lost.

For feature selection, we noted not all parts of image contain complete information about movement.

This is the aperture problem: along a straight border, we can determinate only the border orthogonal component. This is typical problem of Lucas-Kanade algorithm; it derives from lack of constraints for the main equation of optical flow. Generally strategy to overcome these problems is to use only regions with enough rich texture. In this paper we follow Tomasi and Kanade approach (Tomasi and Kanade, 1991). They use a choice criterion to optimized tracking performance. In other words, a feature is defined “good” if it can track “well”. So we considered only features with gradient of eigenvalues enough big.

Tracking algorithm presented stop when a feature is declared “lost”. This can happen because the system can’t recognized it enough well. It is necessary recovery of that feature. The idea is to

forecast where should be basing on shifts of nears features. This forecasting must satisfied particular conditions: minimum distance from other features, value of gradient of eigenvalues and matching skin filter mask.

2.1.3 Pre-Processing

To use a neural net for recognition is necessary to manipulate the results of tracking phase to provide a constant size input. We chose to extract exactly 5 features. We think this number is the minimum necessary to feel same tinges between different gestures. It is necessary to choose the 5 paths that describe better the gesture.

Because typology of gestures used in this paper, and because it is important only the hand movement, we considered only the movement size to eliminate external features from hand, for example face.

We also considered, as parameter of "goodness", the eigenvalue associated to feature. The following image shows the result of this process.

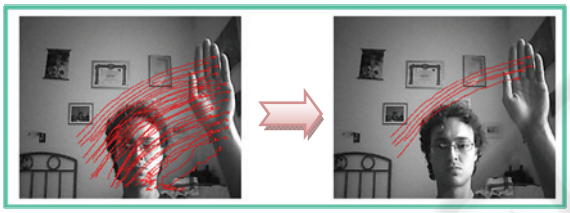

Figure 3: Pre-processing process.

2.2 Artificial Neural Net Architecture

During the building of recognition system, the attention is focused on a single object movement. In particular the intention is to detect the hand movement.

The identification process is based on the "schema" concept, a macrostructure in which the whole action is organized (Rumelhart and Ortony, 1964).

The schema, used in this paper, is a temporal sequent, divided into 10 steps. The parameters that characterized the path of movement are spatial coordinates x and y, derived form pre projection on a bi-dimensional plane (unique point-of-view), and the instant speed (average speed in a sample interval). We suppose that these tree parameters correspond to the minimum information of the semantic of action. This choice reduces the dimensionality of research space and reflects the user oriented representation.

Action semantic correspond to the human perception of covariance from cinematic and geometry (Runeson, 1994).

2.2.1 The Architecture

Recognition system consists of a double layered neural architecture. As you can see from figure 5, the first layer receives inputs of extracted feature from video sequence. Second layer provides, in output, the gesture type.

The system is formed by multilayer perceptrons: the first layer consist of 10 nets, one for each step of sequence, the second layer is formed by a unique perceptron, sufficient to synthesize all the outputs of the first layer. Perceptron is a well-known feedforward net, used in classification problems (Rosenblatt, 1962; Rumelhart et al., 1986).

Input pattern consist of 5 paths, each one of them 30 frame long and described by 3 parameters for each frame. In total we have 450 variables for 10 perceptrons; nets formed by 45 units of input.

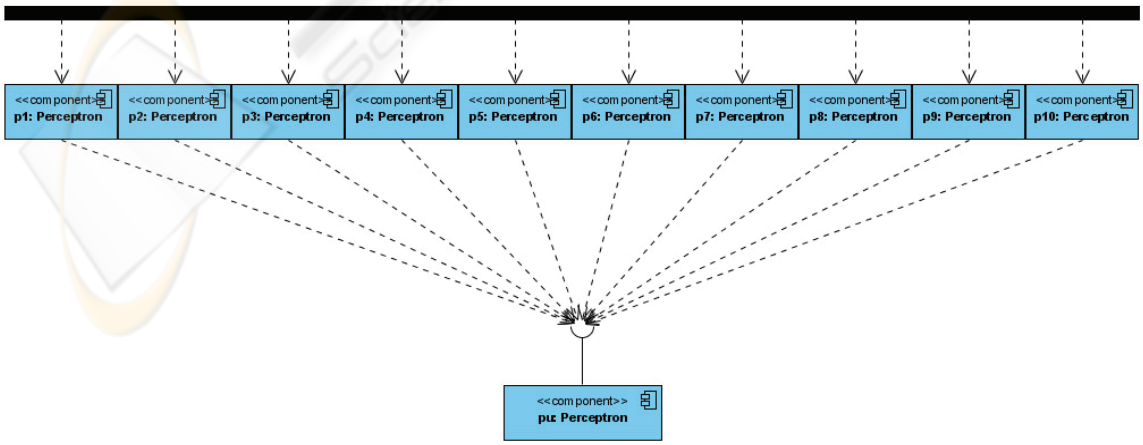

Figure 4: Neural net architecture used.

The following image shows the typical structure of multilayer perceptron.

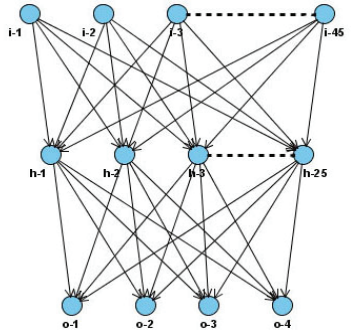

Figure 5: Multilayer perceptron structure used.

It has to be underlined there is a unique hidden layer formed by about half of input units, 25. Output is codified by 4 neurons; so we have the possibility to learn 16 gestures.

Each first layer perceptron recognize 3 frame of sequence. The second layer perceptron helps to merge the information of all the sequence, providing an output based on the global behaviour. Two different movements could have very similar local behaviours.

The dimensions of the second layer perceptron are slightly different: input layer is of 40 units, 4 for each first layer net, while hidden layer has 20 units.

Learning of all nets is supervised and is based on the error-backpropagation algorithm. The pattern that has to be learnt, is presented to net and is calculated the global square error, feedforward phase. After this, error signal is used to modify values of connection coefficients between output units and hidden units, then between hidden units and input units, backpropagation phase.

2.3 User Interface

The term “user interface” refers to what there is between user and machine, what permit to a user to manage system functionalities.

The system communicates only to an external entity, the typical user. The developed interface implements only minimal functionalities required to maintain maximum simplicity: set up, procedure to adjust skin filter parameter, and the recognition, process to recognize gestures.

It is clear how is necessary to associate, to the tactile interface (gesture recognizer), a graphic interface for the functionalities.

It is necessary a common channel, as a GUI, to start these operations and to obtain feedbacks. This assists to control the system but maintain a minimal help from user.

3 TESTING

Testing process consists with a phase for gesture codification, nets learning and at last a testing phase to verify system accuracy and robustness.

Gesture definition is fundamental to finish parameters set up of vision system. We decided to use symbolic gestures because instrument property of gesture recognizer (neural architecture). This choice leads to the typical problems of these systems: segmentation and learning. For the first, a partial solution is the graphic interface. To reduce problems derived from usability and from learning, we kept gestures number to minimum hitting the attention on recognition system efficiency. We gave importance to cinematic nature of gesture e not to static configuration; so we hit on globally hand paths. We use sequence of about 1 minute long to realize a system quite quick in output. These elements surely reduce problems derived from learning and usability; we chose 4 gestures, very easy and not too similar, shown in the following image.

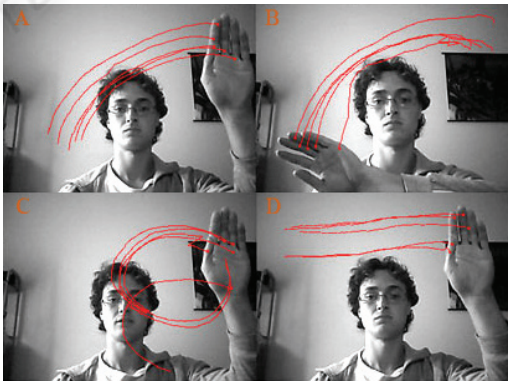

Figure 6: Codified gesture set.

To realize the nets learning, it is necessary to build a training set, a pattern set that mean codified gestures.

For each of them we use 10 different samples done by 5 different people to have a semantic richer set. We noted how each person does each gesture in a different way.

We used learning algorithm in the iterative modality: for each period, connections weights are

modified, MSE is calculated and then process restart. We used a learning rate of 0.1 e 10000 periods.

We noted a really speed of convergence; after only 20000 period we reached a global error about the decimal fifth-figure ($\sim 10^{-5}$).

3.1 Results

System testing in particular is based on robustness test of recognition system, neural architecture. We can do a specific test: generalization phase.

In this phase we produced a pattern set formed by 80 elements, 20 sample for each gesture done by 5 people. We used people different from learning ones. The following table is known as confusion matrix and shows testing results.

Table 1: Confusion matrix.

	A	B	C	D	N. R.
A	18	0	0	0	2
B	0	20	0	0	0
C	1	0	13	0	6
D	3	0	0	15	2

Each row refers to a different gesture; columns indicates how was recognized the particular gesture. "N.R." means "not recognized": happens if nets outputs are too low or if two outputs are quite equal. This case indicates clear nets indecision.

It has to be underlined best result is for gesture B that has the most different cinematic properties, while movement C is most complex, so it is mistaken easily.

4 CONCLUSIONS

System developed shows a quite low computational time although this time is not unimportant; user interface is straightforward and easy to use; finally experiments completed verified about 85 % reliability. In reality we noted the most factor is not only system learning, training set, but it is client learning. Common user needs a time amount to do specific gestures with naturalness even if they are very easy. For a user able to do codified gesture with precision and naturalness, the system has an over 95% of reliability for all the movement typologies.
These results agree with conclusions of the others works about symbolic gesture recognition. The main detected problem from these researches is the user learning time.

Work in this direction surely is not finished; there are a lot of improvements that can be made to this system. We could try to speed up the process to create really real-time software; time of response is reflected on user usability. We could increase gestures vocabulary and verify learning capacities limits. We could improve set up procedure of skin filter to have a completely automatic process and finally we could obtain a GUI-independent system resolving problem derived from movement segmentation. This is the way to have an autonomous gesture interface.

REFERENCES

B. Lucas, T. Kanade, 1981. *An Iterative Image Registration Technique with an Application to Stereo Vision*. Proc 7th Intl Joint Conf on Artificial Intelligence.

C. Tomasi, e T. Kanade, 1991. *Detection and Tracking of Point Feature*. School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA.

D. Chai, 1999. *Face Segmentation Using Skin-Color Map in Videophone Applications*. IEEE Transactions on circuits and systems for video technology, 1999.

J. Shi e C. Tomasi, 1994. *Good Feature to Track*. IEEE Conference on Computer Vision and Pattern Recognition, Seattle.

J. Bouguet, 2000. *Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm*. Intel Corporation Microprocessor Research Labs.

D.E. Rumelhart and A. Ortony, 1964. *The Representation of Knowledge in Memory*. In R.C. Anderson, R.J. Spiro, W.E. Montague (Eds.) Schooling and the acquisition of knowledge, Hillsdale, NJ: Erlbaum.

S. Runeson, 1994. *Perception of Biological Motion: the KSD-Principle and the Implications of a Distal Versus Proximal Approach*. In G. Jansson, W. Epstein & S. S. Bergström (Eds.), Perceiving events and objects.

F. Rosenblatt, 1962. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanism*. Spartan Books, Washington D.C.

D. E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Representations by Back-propagation of Errors", Nature, Vol.323, pp.533-536, 1986.