CHAPTER X

FACIAL GESTURE RECOGNITION AND POSTURAL INTERACTION USING NEURAL EVOLUTION ALGORITHM AND ACTIVE APPEARANCE MODELS

J. GARCÍA, M. GONZÁLEZ-FIERRO, L. MORENO, C. BALAGUER

Robotics Lab, Universidad Carlos III de Madrid {jgbueno, mgpalaci, moreno, balaguer}@ing.uc3m.es

In this paper we address the problem of human-robot interaction from the point of view of expressing emotions. The HOAP-3 humanoid robot is able to discriminate between three emotional states: happy, sad and normal (no emotion). When any of these emotions is found, the robot interacts with the human by performing a sequence of postures and movements, showing happiness or sadness. We make use of Active Appearance Models (AAMs) to determinate the face shape and gesture. We have also developed an algorithm named neural evolution, which is based on neural networks and differential evolution algorithms. This mix of techniques allows us to obtain a ratio of success of 83%.

1 Introduction

Humanoid robot research of last years has produced a remarkable improvement of robot abilities, mobility and interaction capability. Service robots need to be able to interact with humans if they are going to share our environment.

An intelligent and skillful robot requires natural interaction and complex behaviors to perform tasks and offer services to humans. It also needs cognitive models to understand human emotions and expressions, and it has to be able to reply in consonance. Visual recognition of facial gestures can be useful in accomplishing natural and robust human-robot interaction.

However, to develop a system that detects and interprets facial expressions can be challenging. It involves the problem of determinate the relevant facial features and classify the different expressions.

Emotional states and expressions have been characterized in six generic states, such as fear, joy, sadness, surprise, disgust and anger (Ekman, 1999). More complex emotions can be detected by mixing these proposed basic expressions.

Pattern recognition and computer vision techniques have been successfully used for many gesture recognition systems (Mitra, 2007) in very different robots, such as social robots as Maggie (Ramey, 2011) or mobile manipulator as ARMAR robot (Stiefelhagen, 2007). It involves face tracking, shape detection of facial features, clustering, optical flow, optimization and classification. Facial Feature Extractions techniques rely on the detection and tracking of several face parameters, like mouth shape or eyebrows distance (Cerezo, 2007).

An approach that has been probed very effective is based on Hidden Markov Models (HMMs). In (Otsuka, 1998) facial muscles variations are modeled and classified, and in (Arsic, 2006) the facial expression is decomposed in submotions to enhance the performance. Other approach is based on Kalman filtering to predict and track facial features (Zelinsky, 1996).

Principal Component Analysis (PCA) usually reports good recognition rates. The eigenfaces method calculates an approximate representation of the face. It finds the principal components of the facial image distribution (Chung, 1999).

Other methods based on Facial Action Coding System (FACS), Active Appearance Models (AAMs), particle filters or Support Vector Machines (SVMs) have also provided good results (Essa, 1997)(Dornaika, 2008)

In this paper, a facial gesture detection system has been developed and implement in the humanoid robot HOAP-3. Using AAMs (Edwards, 1998), the robot is able to detect the shape of the face that is appearing in front of them. After extracting some characteristic features of the face, a neural network is trained and optimized using differential evolution. This novel algorithm has been named neural evolution. A set of emotional gestures are classified in three different states: happy, sad and normal (no emotional activity).

The interaction of the robot is performed by a set of postural sequences which try also to express emotions. If the robot detects that the human is happy, it responds happily waving the arms and moving the head. On the contrary, if the robot determinate that the human is sad, it replies with a slow movement lowing his head.

The document is structured as follows. In section 2 the method to obtain the face shape is exposed. Section 3 presents the algorithm neural evolution. In section 4 a small presentation of the humanoid robot is showed. In section 5 the results are showed and discussed. And finally, section 6 presents the conclusions.

2 Active Appearance Models

Active Appearance Model (AAM) was introduced by (Edwards, 1998) with multi-resolution, color textures and a better edge finder method. Some of the applications are medical imagery analysis, texture recognition and face tracking. AAM is based on ASM, a proposal which enables the model to automatically recognize if a contour is a good target or not. Furthermore, ASM introduced the texture information by adding the texture of the lines that passes perpendicularly to the control point, fixing the positions of the mesh on each step. The initial contour is found to match the best texture for the control mesh iteratively.

AAM was improved with weighting steps and extra normalization. Models are generated by combining a model of shape variation with a model of the appearance variations in a shape-normalized frame. Using a training set of images, landmarks of enhanced points are extracted, giving a statistical model of shape variation. The alignment of these features can be introduced into a PCA to reduce the amount of information details.

$$x = \tilde{x} + P_s \cdot b_s \qquad (1)$$

where $\tilde{x}$ is the mean shape P_s is a set of orthogonal modes of variation and b_s is a set of shape parameters. Each triangle of the Delaunay mesh is warped so that their control points match the mean shape by means of a barycenter property of the triangles. Each texture is normalized to reduce the global lighting variation applying a scaling α , and offset, β ,

$g = (g_{im} - \beta 1) / \alpha$

That is done recursively giving as a result after applying PCA

$$g = \tilde{g} + P_g \cdot b_g$$

where $\tilde{g}$ is the mean normalized gray vector, P_g is a set of orthogonal modes of variation and b_g is a set of grey-levels parameters. Therefore, the shape and appearance can be summarized by the vectors b_s and b_g . Correlations between texture and shape can be found, that is why PĈA is once again performed to the data, giving as a result:

$$b = \left(\frac{W_s b_s}{b_g} \right) = \left(\frac{W_s P_s^T (x - \tilde{x})}{P_g^T (g - \tilde{g})} \right)$$

where W_s is a diagonal matrix of weights for each shape parameter. Applying PCA to the previous equation, the obtained model

$$b = Q \cdot c$$

where Q are the eigenvectors and c is a vector of appearance parameters which depends on shape and gray-levels of the model. Because of the linearity of the problem, any face image can be represented by means of parameter c so

$$x = \tilde{x} + P_s \cdot W_s \cdot Q_s \cdot c$$

$$g = \tilde{g} + P_g \cdot Q_g \cdot c$$

a grey-level image can be generated from the vector g and warped using control points described by x.

3 Neural Evolution

In this paper, a novel method for pattern recognition is presented based on a genetic algorithm named Differential Evolution (DE) in conjunction with a Neural Network in charge of the evaluation process. The genetic optimizer minimizes the global error by tuning the weights and biases of the NN.

3.1 Differential Evolution

3.1.1 Background and State-Of-Art in optimization problem

The differential evolution is mainly an optimization method invented in 1995 based on the genetic algorithm developed by Kenneth Price. It is based on population that attacks the initial problem by means an evaluation of multiple initial points selected randomly and it evolutions over the previous populations randomly.

As it is stated in (Price, 2005), there exists several ways to solve the minimization problem in multimodal functions. As it is logical, the selection of the starting points is the first issue to be solved. Before genetic algorithms were used, several alternatives have been studied, precise-less and performing low robustness:

1. *Simulated Annealing* – Performs an heuristic search where, on each iteration, closest points are evaluated and probabilistically it

is decided if a new state s' is chosen or not looking for points with less energy. This procedure is realized until the energy is lower than a certain value. This method has a transition probability greater than zero, eliminating the chance to get locked in a local minimum. Furthermore, as long as global minimum is reached, probability is reduced asymptotically.

2. *Multi-Point, Derivative-Based Methods* – Several initial points are proposed and energy is estimated based on their values. Normally, these methods apply a derivative function, even not been strictly necessary. Being possible to apply direct search techniques where the function cannot be derived.

3. *Multi-Point, clustering methods* – Other possibility is to cluster the initial points based on their attraction. With this method, minimums can be taken as hyper-ellipsoids. It is possible to estimate the center of the hyper-ellipsoids and decides which the global minimum is. There is an important memory consumption issue using this method, so other alternatives are actually chosen.

3.1.2 DE method

DE is an optimizer based on population that solves the initial point selection by means of sampling the objective function in random initial points. During the **initial step**, the input parameter's domains $x_m^{min} \cdot x_m^{max}$ are established, generating N_P vectors over the initial population as shown in figure 2. Each vector is indexed taking a value between 0 and $N_P - 1$

As in other population based methods, DE generates new points (perturbations) based on previous points. Those deviations are not reflections as other solutions such as CRS or Nelder-Mead. The main difference comes with a selection of those new points, which are randomly selected from three individuals. Two of the elements x_{r1}, x_{r2} are subtracted and multiply by a weight (weight and mutation) F and a third point is added x_{r_3} giving the trial vector

$$u_0 = x_{r3} + F \cdot (x_{r1} - x_{r2}) \tag{3.1}$$

as it is shown in figure 3

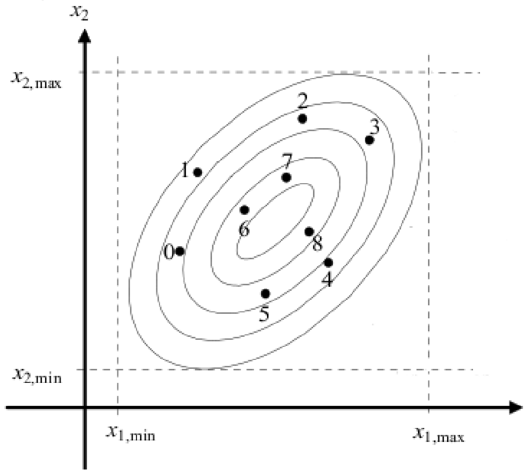

Figure 1: Differential Evolution. First Approach

Afterwards, in the selection step, the u_0 trial vector is compared with the rest of the vectors with the same index, where in figure the number is 0. This representation is shown the **selection and storage** where the lowest cost vector is taken as the member for the next generation. This process is repeated until a population N_p has competed against the trial vector randomly generated. Once the last vector has been evaluated, the survivor vectors of the N_p come the predecessor of the next iteration.

When an exit condition is achieved, the algorithm finishes. Usually, the boundary conditions are: time, number of iterations/generations or achieved precision. For this paper, due to the fact that the search is performed once, convergence speed is not crucial, being the maximum priority for an optimum the precision (once the network is optimized, the optimal values for weights and biases is the same and do not need to be changed).

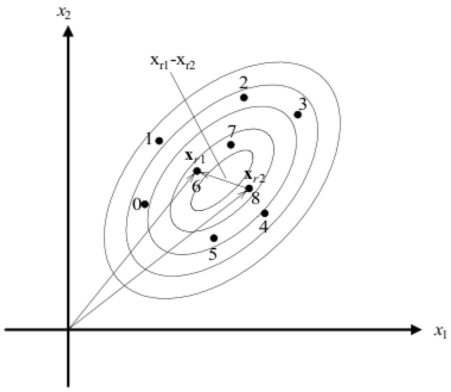

Figure 2: Differential Evolution. Selection of the population with random values and generation of vector u0

3.2 Proposed system: Neural Evolution

The proposed algorithm mixes three well known systems: DE optimizer, NN system and AAM for feature face extractions. Figure resumes the whole structure. Basically, a NN is trained with DE instead of classical methods such as back-propagation. The optimized values are weights and biases of input layer, hidden layer and output layer, giving as a result the optimal NN. The input parameters are the location of the features for each face obtained using AAM's location algorithm.

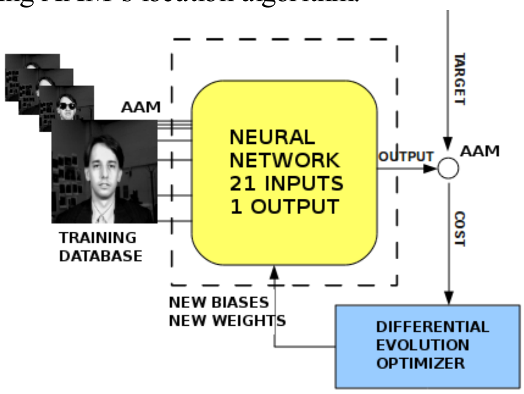

Figure 3: Proposed System mixing AAM with a NN optimized with Differential Evolution. Weights and biases are tuned by the optimizer.

In this case, the NN could have a lineal threshold activation function, because in this case it is not necessary as back propagation algorithm does. The source code for the NN evaluation has been ported from Carnegie Mellon University. The followed algorithm and some of the experiments performed come from (Mitchell, 1997).

4 Humanoid platform

HOAP-3 robot (Figure 4) is a small humanoid of 60 cm and 9 kg., designed and developed by the Japanese Fujitsu. It has 28 degrees of freedom which provides it with high movement capability. All motors can be controlled in position or velocity.

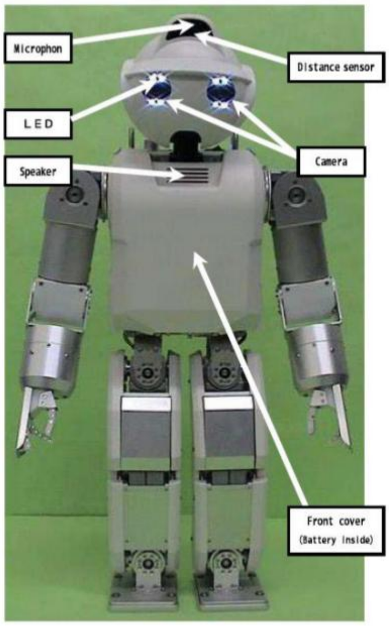

Figure 4: HOAP-3 robot architecture.

The robot incorporates an embedded PC-104, with 1.1 GHz, 512 Mb of RAM and wireless connection. Inside the robot runs a RT-Linux based on Fedora (2003 edition).

To complete the functionality of this humanoid, a set of sensors are added. It has two usb cameras, grip sensors, accelerometers, gyros and ZMP sensors.

5 Experimental results

To compute the AAM algorithm[1], a database of images, each of them with different faces and expressions, has been used. To obtain the average model of the face, 63 characteristic points has been selected. These landmarks correspond to the important features that define a face, such as eyes, chin, nose, mouth and eyebrows (Figure 5).

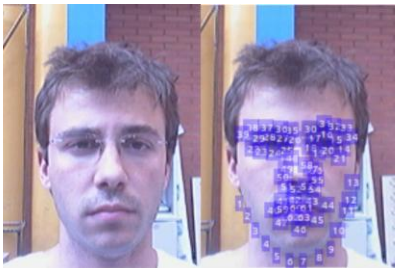

Figure 5: Original image and landmarks of 63 charasteristic points

The system is trained to obtain a statistical model of the face shape and texture, and produces a face tracker. This face tracker is represented as a mesh using Delaunay triangles (Figure 6a). The representation of the face model is made of splines using the face tracker (Figure 6b).

[1] The AAM algorithm used is based on the code of Dr. Radhika Vathsan at BITS Pilani Goa Campus, India.

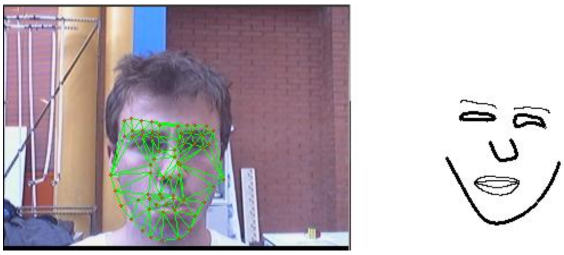

Figure 6: Face tracker mesh (a) and face spline representation (b)

Once the face model has been obtained, it can be used to find facial expressions. To do so, a neural network has been trained with some features of the face model mesh. The features selected have been the relative position of the mouth shape, nose and eyebrows. A set of images of several people in each of the tree states studied, happy, sad and normal, have been provided to the neural network. To optimize the network, a differential evolution algorithm has been used. We have called this neural evolution.

Once the neural network is trained, it has been implemented in HOAP-3 robot, so it can be able to interact with humans. The process occurs as follows, HOAP-3 robot detects, in real time, the face of the user and computes the AAM model of his face. Then, it introduces the selected features as the input of the neural network. The result is one of the proposed states: happy, sad or normal.

The robot produces a postural response so that it replies to the human emotion. If the human is happy, the robot moves its arms up and down and moves the head. However, if the human is sad, the robot performs a sad movement, looking down and moving the head from side to side. If the robot detects the normal expression, it waits until a new expression is found. All trajectories are preprogrammed and calculated using inverse kinematics. A set of snapshots are showed in Figure 7, showing the robustness of the system presented.

Figure 7: Snapshots of the experiment. Recognition of normal gesture, happy gesture and sad gesture.

In Table 1, results of the gesture recognition system are shown. It can be seen that it achieves a high rate of accuracy. Training data corresponds to the results of the database image, experimental to the results of the robot implementation and overall to the media of both.

Table 1: Results of gesture recognition

	Training data	Experimental	Overall
Happy	96 %	90 %	93 %
Sad	78 %	67 %	72.5 %
Normal	87 %	79 %	83 %

6 Conclusions

In this paper a new approach of human-robot interaction has been presented. First, AAM models have been used to obtain a statistical representation of the human face. Extracting some important features of the mesh resulting of the AAM algorithm, like mouth and eyebrows shape, a neural network has been trained, to distinguish between three different states, happy, sad, and normal (no emotion). To optimize the network, differential evolution algorithm has been used.

This algorithm has been implemented in HOAP-3 humanoid and tested in the Robotics Lab with several people. The robot has been able to determine the emotional state of people by means of the proposed gesture recognition system. Furthermore, the robot is able to react to those emotions performing a set of pre-programmed human-like movements.

Acknowledgements

We want to thank to Alejandro Balseiro for his inspiration in the experimental phase of this project.

References

Arsic, D., Schenk, J., Schuller, B., Wallhoff, F., and Rigoll, G.: Submotions for hidden markov model based dynamic facial action recognition, Image Processing, 2006 IEEE International Conference on, 673–676, 2006

Cerezo, E., Hupont, I., Manresa-Yee, C., Varona, J., Baldassarri, S., Perales, F., and Seron, F.: Real-time facial expression recognition for natural interaction, Pattern Recognition and Image Analysis, Springer, 40–47, 2007

Chung, K., Kee, S.C., and Kim, S.R.: Face recognition using principal component analysis of Gabor filter responses, ratfg-rts, Published by the IEEE Computer Society, 53, 1999

Dornaika, F., and Davoine, F.: Simultaneous tracking and facial expression recognition using multiperson and multiclass autoregressive models, Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on, 1–6,

Edwards G, Taylor C, Cootes T. Interpreting face images using active appearance models Proceeding of the International Conference on Face And Gesture Recognition 1998, pages 300-305

Ekman, P.: Facial Expression, the Handbook of Cognition and Emotion. John Wiley et Sons, Chichester (1999)

I. Essa and A. Pentland, “Coding, analysis, interpretation, recognition of facial expressions,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 7, pp. 757–736, Jul. 1997.

Mitra, S., and Acharya, T.: Gesture recognition: A survey, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37(3), volume 37, IEEE, 311–324, 2007.

Ramey A, González-Pacheco V, Salichs MA. Integration of a low-cost RGB-D sensor in a social robot for gesture recognition. In: *Proceedings of the 6th international conference on Human-robot interaction - HRI ’11*. New York, New York, USA: ACM Press; 2011.

Stiefelhagen, R., Ekenel, H., F\ügen, C., Gieselmann, P., Holzapfel, H., Kraft, F., Nickel, K., Voit, M., and Waibel, A.: Enabling multimodal human-robot interaction fort the karlsruhe humanoid robot, IEEE Transactions on Robotics, Special Issue on Human-Robot Interaction 23(5), volume 23, 840–851, 2007

Otsuka, T., and Ohya, J.: Spotting segments displaying facial expression from image sequences using HMM, Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on, 442–447, 1998

Zelinsky, A., and Heinzmann, J.: Human-robot interaction using facial gesture recognition, Robot and Human Communication, 1996., 5th IEEE International Workshop on, 256–261, 1996