Facial Emotion Recognition and Adaptative Postural Reaction by a Humanoid based on Neural Evolution

Jorge García Bueno, Miguel González-Fierro, Luis Moreno & Carlos Balaguer

Manuscript
Received:
30, Apr., 2013
Revised:
17, Jul., 2013
Accepted:
24, Jul., 2013
Published:
15, Oct., 2013

Keywords
HRI, Neural Evolution, AAM, Posture interaction.

***Abstract*— We present a method of bidirectional interaction between a human and a humanoid robot in terms of emotional expressions. The robot is able to detect continuous transitions of human emotions that ranges from very sad to very happy using Active Appearance Models (AAMs) and Neural Evolution Algorithm to determinate the face shape and gestures. As a response of the human emotions, the robot performs postural reactions that dynamically adapt to the human expressions, performing a body language which changes in terms of intensity as the human emotions vary. Our method is implemented in the HOAP-3 humanoid robot.**

1. Introduction

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 [1]. More complex emotions can be detected by mixing these proposed basic expressions.

Gesture and facial emotion recognition system has been a wide field of research in recent years [2]. Some emotion recognition systems use Emotional Text To Speech (ETTS) to express emotions [3-5]. They control speed, pauses and volume of the speech to detect between different emotions.

Some systems like [6] combines speech recognition, face tracking, shape detection of facial features, clustering, optical flow, optimization and classification and gesture detection to perform multimodal Human-Robot Interaction.

Facial Feature Extractions techniques rely on the detection and tracking of several face parameters, like mouth shape or eyebrows distance [7].

An approach that has been probed very effective is based on Hidden Markov Models (HMMs). In [8] facial muscles variations are modeled and classified, and in [9] the facial expression is decomposed in sub-motions to enhance the performance. Other approach is based on Kalman filtering to predict and track facial features [10].

Principal Component Analysis (PCA) usually reports good recognition rates. The eigen-faces method calculates an approximate representation of the face. It finds the principal components of the facial image distribution [11].

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 [12-13].

On the other hand postural interaction is one of the more basic ways of communication. Postural interaction has a great importance in the first years of the human development, when an infant wants to communicate with his/her mother [14-15]. The lack of these abilities is usually related with some mental disorders like autism [16].

This phenomenon is also studied in animals. Some studies address the intentional meaning of gestures in apes [17], or how human gestures are interpreted by animals [18].

Robots, in a similar way, should be able to communicate and express emotions through gestures and movements.

Some works address methods of emotional interaction with robots through dancing [19-21], fear expressions [22-25], happiness or sadness [23-28].

An interesting approach in which our work is based is the so called emotional body language [23-24]. The authors use the small humanoid robot NAO to express emotions with the robot body. They found that the position and movement of the head influence in the transmission of happiness or sadness.

The work with the humanoid robot KOBIAN addresses the robot emotional communication in terms of facial expressions and postural movements [26-28]. In [26-27] a whole body emotional routine is computed for every emotion and a group of people is selected to determine which emotion the robot is performing.

In our previous work [29], we developed a bidirectional interaction system that recognizes three discrete emotional states, happy, sad and neutral. The robot was able to reply in accordance to the inputs of the human. In this paper, a facial gesture detection system has been developed and implement

This work was supported by the Department of Systems and Automatics in Universidad Carlos III de Madrid.

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

in the humanoid robot HOAP-3. Using AAMs [30], 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 neutral (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 a facial gesture procedure for the AAMs is explained, section 3 describes a novel algorithm for learning how to determine the state of emotion of the human. Then section 4 explains the humanoids postural interaction and the different movements adaptively changed with the emotion degree. Lately in section 5 the proposed system is defined, section 6 contains the experimental results and finally section 7 gathers the results, conclusions and comparisons with related work.

2. Facial gesture acquisition

In the daily life, face-to-face communication plays a very important role in the expression of character, emotion and/or identity [31]. In [32] it is shown that only 7% of affective information is transferred by spoken language, that 38% is transferred by paralanguage and 55% of transfer is due to facial expressions. Then facial expression is the principal way to transmit emotions between people.

A. Active Appearance Models

Active Appearance Model (AAM) was introduced by [30] 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.

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

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,

$$g = \left(g_{im} - \beta\right) / \alpha \qquad \text{(Equ. 2)}$$

That is done recursively giving as a result after applying PCA

$$g = \tilde{g} + P_g \cdot b_g \qquad \text{(Equ. 3)}$$

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 PCA is once again performed to the data, giving as a result:

$$b = \left(\begin{array}{c} W_s b_s \\ b_g \end{array} \right) = \left(\begin{array}{c} W_s P_s^T \left(x - \tilde{x}\right) \\ P_g^T \left(g - \tilde{g}\right) \end{array} \right) \qquad \text{(Equ. 4)}$$

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

$$b = Q \cdot c \qquad \text{(Equ. 5)}$$

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

$$x = \tilde{x} + P_s \cdot W_s \cdot Q_s \cdot c \qquad \text{(Equ. 6)}$$

$$g = \tilde{g} + P_g \cdot Q_g \cdot c \qquad \text{(Equ. 7)}$$

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

B. Initialization of AAM

Due to human bodies are complex and very variable objects, it is complicated to find features or heuristics that could confront the huge variety of instances of the object class (*e.g* faces, arms, legs …) that may rotate in any direction, captured in different light conditions or the simple apparition of glasses, jumpers or any other external object such as umbrellas, shopping bags or even a mum carrying a pram. For such as objects, statistical models (here called classifiers) may be trained and used to detect the desired targets.

To do that, statistical models will be taught using multiple instances of the object to be recognized (these instances are called *positive*) and also multiple samples of *negative* instances where the object does not appear. The collection of all these samples, positive and negative, form a training set. During the training process, face features will be extracted from the training set and unique features of each image will be used to classify the object. It is important to remark that using this method, if the cascade does not detect an existing object it is possible to add the sample to the classifier training set and correct it for the next time.

The statistical approach used in this project has been defined using the OpenCV libraries based directly on the Viola and Jones publication [1]. This option applies individual *Haar-Like* features and a cascade of boosted tree classifiers as a statistical model. The classifier must be trained on images of the same size and detection is done using a window of that size moved along the whole picture. For each step, the algorithm checks if the region looks like the desired object or not.

Furthermore, to include possible sizes of the images to be detected, the classifier has the ability to scale the patterns. To make this method works, it is necessary just a batch of Haar-like features and a large set of very simple classifiers to classify the image region as the *desired object* or as a *non-desired object*. Each feature is determined by the shape of the feature, its position relative to the search window origin and the scale factor applied on the feature. As a result, 14 templates shown in Figure 1 are used for this project.

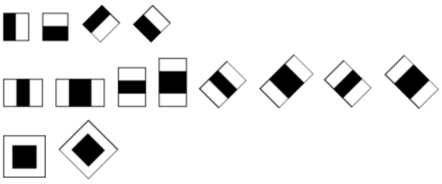


Fig. 1. Set of Haar-like templates used for object detection. Edge features, line features and centre-surround features.

As in the previous Figure, each feature is designed using two or three black or white rectangles horizontal, vertical or rotated by 45º. To compute the Haar feature's value, just a weighted summation of two components is needed: the pixel sum over the whole area of the feature and the sum over the black rectangle. Once this simple calculation is done, the weights of both components are of opposite signs and they are normalized dividing the summation over the black rectangle by the total area. As an example, for the second feature:

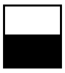


Fig. 2 Example of Haar-like templates used for object detection.

$$2 \times weight_{black} = -4 \times weight_{whole} \qquad \text{(Equ. 8)}$$

or for the thirteenth feature:

$$weight_{black} = -9 \times weight_{whole} \qquad \text{(Equ. 9)}$$

Now, instead of computing directly the pixel sums over multiple rectangles and make the process of detection incredibly slow, [5] defined a way to make the summations faster with at tool named Integral Image.

$$ii(x,y) = \sum_{x'<x,y'<y} i(x',y') \qquad \text{(Equ. 10)}$$

where $ii(x,y)$ is the integral image and $i(x,y)$ is the original image. The summation of pixels over a single window can be done using the surrounding areas as showed in Figure 3.

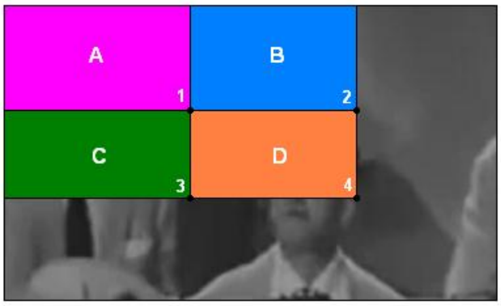


Fig. 3 Area of different regions of pixels used for the integral images procedure.

where the sum of the pixels of the rectangle D can be computed with four array references as is demonstrated in [1]. The integral image at 1 corresponds to the summation of all the pixels included in rectangle A. Just as the previous example, the value at 2 is A + B, the value at 3 is A + C and finally the value at 4 is A + B + C + D. Therefore, the sum within D can be done as 4 + 1 - (2 + 3). That means that the pixel sum over a rectangle can be done regardless of the size, just taking into account the corners of the rectangle.

$$RecSum(r) = \underbrace{ii(x_0+w, y_0+h)}_{4} + \underbrace{ii(x_0, y_0)}_{1} - \underbrace{ii(x_0+w, y_0)}_{2} - \underbrace{ii(x_0, y_0+h)}_{3} \qquad \text{(Equ. 11)}$$

If a decision tree classifier is created taking into account each one of the feature value computed over each area of the image, as in the following Equ.4 for two terminal nodes or the next equation Equ.5 for three, where each f_i will give *+1* if the obtained value is inside a threshold predefined and *-1* otherwise.

$$f_i = \begin{cases} +1, & x_i \geq t_i \\ -1, & x_i < t_i \end{cases} \quad \text{(Equ.12)}$$

$$f_i = \begin{cases} +1, & t_{i,0} \leq x_i \leq t_{i,1} \\ -1, & else \end{cases} \quad \text{(Equ. 13)}$$

Where it can be stated that

$$x_i = w_{i,black} \cdot RecSum(r_{i,black}) + w_{i,whole} \cdot RecSum(r_{i,whole}) \quad \text{(Equ. 14)}$$

It is important to notice that each classifier is not ready to detect an object by itself. It notices simple feature in the image (like a border or a point). For instance, the eye region is often darker than the cheeks, or the iris is darker than the rest of the eye (supposing a correct size and orientation of the feature as in Figure 4).

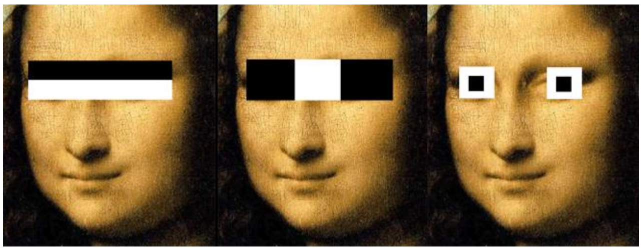


Fig. 4 Area of different regions of pixels used for the integral images procedure.

After the creation of these classifiers (called *weak classifiers)* a list of complex robust classifiers is built out with the union of all the weak classifiers iteratively as a weighted sum of weak classifiers, being each one increasingly more complex. Afterwards, a cascade is created where first positions are for simple classifiers and final positions for the most complex. As far as the window is scanned by each classifier, it can be rejected or sent to the next F$_i$ as stated in Figure 5.

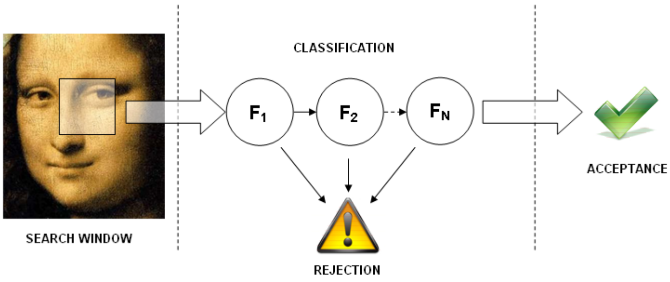


Fig. 5 Cascade of Haar. Each classifier can reject or pass to the next classifier the image.

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 (NN) in charge of the evaluation process. The genetic optimizer minimizes the global error by tuning the weights and biases of the NN directly based on the performance results for each iteration.

One of the principal contributions of this work is the recognition of emotions or any other patterns based on a NN. In this section a complete comparison between back-propagation and differential evolution algorithms for the optimization of the net will be done. In both cases the same network will be selected, the difference will be on the learning methodology.

To compare both alternatives several experiments have been processed. On each experiment the error (as it will be defined below) has been analyzed in order to try the best configuration parameters for both alternatives.

C. Differential Evolution

1) Background and State-Of-Art in optimization problem: Differential evolution is mainly an optimization method invented in 1995 based on the genetic algorithm developed by Kenneth Price [33]. 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 [34], 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:

- *Simulated Annealing* – Performs a heuristic search where in every iteration the closest points are evaluated and probabilistically it is decided if a new state 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.
- *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.
- *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.

2) *Differential Evolution 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_{r3} giving the trial vector

$$u_0 = x_{r3} + F\cdot\left(x_{r1} - x_{r2}\right) \qquad \text{(Equ. 15)}$$

as it is shown in the following Figure 6.

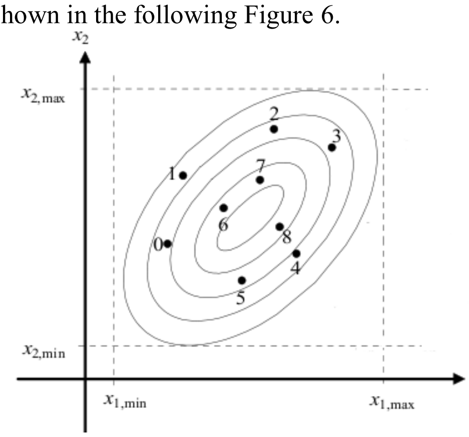


Fig. 6. Differential Evolution optimization algorithm. First approach to the global minimum.

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 has competed against the trial vector randomly generated. Once the last vector has been evaluated, the survivor vectors of the 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).

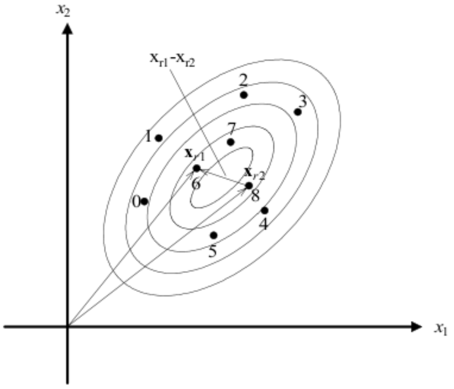


Fig. 7. Differential Evolution. Selection of the population with random values and generation of vector u$_0$.

D. 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.

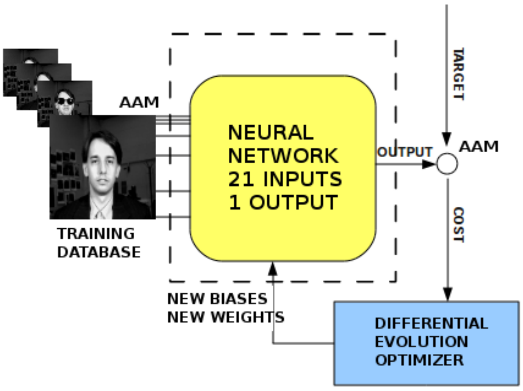

AAM
TRAINING DATABASE
NEURAL NETWORK 21 INPUTS 1 OUTPUT
OUTPUT
TARGET
AAM
COST
NEW BIASES
NEW WEIGHTS
DIFFERENTIAL EVOLUTION OPTIMIZER

Fig. 8 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 [35].

4. Humanoid postural interaction

The interaction with the human is considered to be dynamically changing while the human is expressing different emotions, i.e., as the intensity of the human emotion rises, the postural response of the humanoid changes in accordance with the human expression.

A. Humanoid robot

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

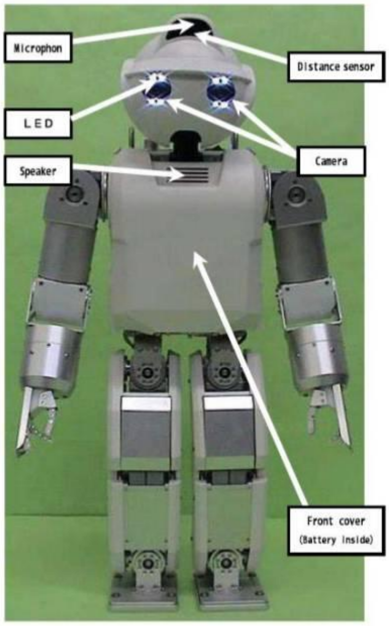


Fig. 9 Humanoid HOAP-3 robot architecture.

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.

A. Trajectory generation based on human demonstrations

The trajectories performed by the robot have been computed based on the demonstrations of a human teacher.

First, a tracking vision system to capture the teacher's movement has been developed. Using a set of 3 tags placed at the shoulder, elbow and hand, the movement of the teacher arms has been tracked. To obtain the 3D trajectories of the tags, we have used colour segmentation and a Kalman filter.

The noisy trajectory obtained is smoothed using a cubic spline, which is defined as a piecewise polynomial fitted to a set of via points.

$$(t_0, q_0^*), (t_1, q_1^*) ... (t_k, q_k^*) \qquad \text{(Equ. 16)}$$

where $q_i^* \in \mathbb{R}^N$ is the articular via points at time $t_i \in \mathbb{R}$.

Given these via points, there is a cubic trajectory that passes through these points and satisfies smooth criteria, given by

$$q_i(t) = a_i(t-t_i)^3 + b_i(t-t_i)^2 + c_i(t-t_i) + d_i \qquad \text{(Equ. 17)}$$

where a_i, b_i, c_i, d_i are the polynomial coefficients optimized. The complete articular trajectory $q(t) \in \mathbb{R}^N$ is a concatenation of Equ. 10 over the time intervals:

$$q(t) = \begin{cases} q_0(t) & \text{if } \mathrm{t}_0 \le \mathrm{t} < \mathrm{t}_1 \\ \vdots & \\ q_k(t) & \text{if } \mathrm{t}_\mathrm{k} \le \mathrm{t} < \mathrm{t}_{\mathrm{k+1}} \end{cases} \qquad \text{(Equ. 18)}$$

The trajectories performed by the human have to be adapted to the robot. To do so, we have obtained the kinematic model of the human arm as a 4 degree of freedom manipulator, with 3 degrees of freedom (yaw, pitch, roll) in the shoulder and 1 degree of freedom in the elbow (yaw). This is the same model as the HOAP arm.

In such way, it is possible to use Inverse Kinematics algorithms in order to get the joint angles of the robot arm. In Fig. 10 the used algorithm is presented.

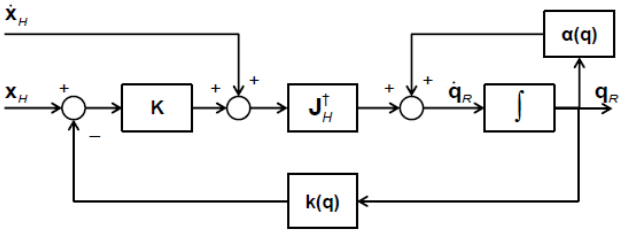


Fig. 10 Humanoid robot trajectory adaptation from human teacher.

The reference position and velocities of the human arm are used as input. The human arm angle velocities can be calculated using the equation

$$\dot{\mathbf{q}}_H = \mathbf{J}_H^\dagger \left[\dot{\mathbf{x}}_H + \mathbf{K}\left(\mathbf{x}_H - k(\mathbf{q})\right) \right] \qquad \text{(Equ. 19)}$$

where the pseudo-inverse of the Jacobian Matrix is used since only the position of the arm is considered. The remaining degrees of freedom can be used in order to adapt the different range of movements of the HOAP-3 robot with respect to the human arm. Then, the robot joints velocities are calculated as:

$$\dot{\mathbf{q}}_R = \mathbf{J}_H^\dagger \left[\dot{\mathbf{x}}_H + \mathbf{K}\left(\mathbf{x}_H - k(\mathbf{q})\right) \right] + \alpha(\mathbf{q}_R) \qquad \text{(Equ. 20)}$$

where

$$\alpha(\mathbf{q}_R) = \left[\mathbf{I} - \mathbf{J}_H^\dagger \mathbf{J}_H \right] \dot{\mathbf{q}}_0 \qquad \text{(Equ. 21)}$$

The vector $\dot{\mathbf{q}}_0$ can be calculated in order to get a solution of joint angles being far from the HOAP-3 joints limits, while getting the same end-effector trajectory:

$$\dot{\mathbf{q}}_{0,i} = -k_l \frac{\mathbf{q}_{H,i} - \bar{\mathbf{q}}_{R,i}}{(\mathbf{q}_{R,i,M} - \mathbf{q}_{R,i,m})^2} \qquad \text{(Equ. 22)}$$

with $k_l > 0$.

In Fig. 11 and Fig. 12 the right and left joint trajectories are plotted.

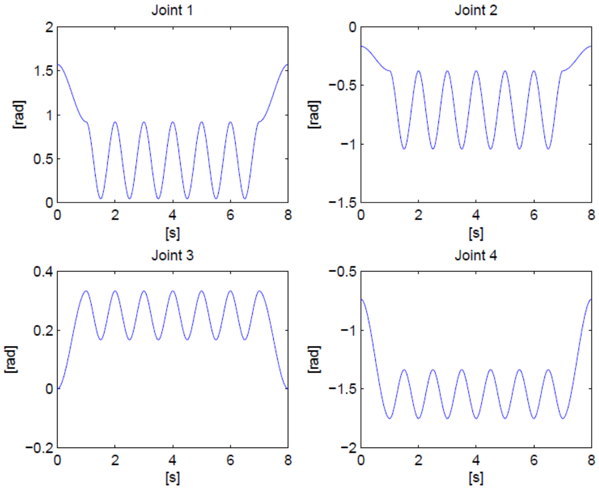


Fig. 11 Adapted trajectories for the humanoid's right arm.

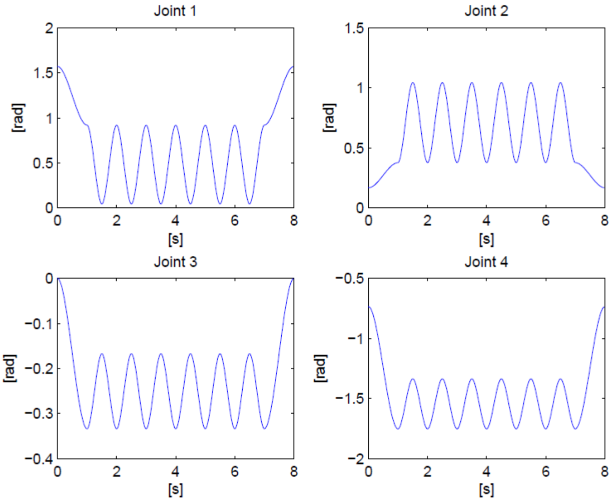


Fig. 12 Adapted trajectories for the humanoid's left arm.

Once the trajectory of sadness and the trajectory of happiness are obtained from the human demonstration, the resulting smoothed spline is parameterized as a function of the time. Then, the trajectories can be modified to go faster or slower.

If the robot detects that the human is in the state *very sad*, the spline is computed to have the maximum time allowed for the trajectory of sadness, which is 8s. As the system detects that the human is moving from *very sad* to *neutral*, the trajectory time rises linearly to 4s, then the trajectory is faster. In the case of the trajectory of happiness, the computation is similar. When the system detects that the human is in the state very happy, the time is the minimum allowed, which is 4s. As the system detects that the human emotion moves from very happy to neutral, the trajectory time increases linearly until 8s, going slower.

B. Torque limit evaluation through inverse dynamics

The trajectories are designed to do not pass the joint limits. Furthermore, a torque analysis needs to be done in order to verify that the torque limits are not surpassed and the trajectory is safe. To obtain the torque trajectory of the robot arms the Newton-Euler formulation has been used.

The algorithms based on this approach are faster than those based on Lagrange formulation [36]. On the contrary, Lagrange formulation has the advantage that only the kinetic and potential energy need to be computed, so they reduce the number of equations to derive and are less prone to errors.

Newton-Euler formulation is based on the balance of all the forces acting on the robot links. This implies that the equations can be expressed in a recursive way, which produces a big advantage, the algorithms based on this formulation are faster than non-recursive ones. Newton-Euler method are described by two equations, the first one is related to the translational movement of the center of mass.

$$f_i - f_{i+1} = m_i \ddot{r}_{CM} - m_i g \qquad \text{(Equ. 23)}$$

where f is the force passing through the link, $\ddot{r}_{CM}$ is the center of mass acceleration, m is the link mass and g is the gravity acceleration.

The second equation is based on the rotative movement of the link.

$$T_i - T_{i+1} = I_i \alpha_i + \omega_i \times (I_i \omega_i) \qquad \text{(Equ. 24)}$$

where T is the torque produced by the link, I is the inertia tensor of the link, α is the angular acceleration and ω the angular velocity.

Building the dynamic model of a high degree of freedom robot can be tedious. If we are working with a humanoid robot, the problem is more difficult due to the numerous joints and the closed kinematic chains. Spatial formulation of dynamics provides a compact and easy to implement notation. This formulation makes use of $6D$ vector and tensors to describe velocity, acceleration, inertia and force [36]. Using these components, a set of dynamic algorithms can be developed.

The equation of motion of a rigid body system is defined using the spatial notation as:

$$\mathbf{f} = \frac{d}{dt}(\mathbf{Iv}) = \mathbf{Ia} + \mathbf{v} \times \mathbf{Iv} \qquad \text{(Equ. 25)}$$

with

$$\mathbf{f} = \begin{pmatrix} n \\ f \end{pmatrix} \in F^6 \qquad \text{(Equ. 26)}$$

$$\mathbf{v} = \begin{pmatrix} \omega \\ v \end{pmatrix} \in M^6 \qquad \text{(Equ. 27)}$$

$$\mathbf{a} = \begin{pmatrix} \dot{\omega} \\ \ddot{c} - v \times \omega \end{pmatrix} \in M^6 \qquad \text{(Equ. 28)}$$

$$\mathbf{I} = \begin{pmatrix} Ic & 0 \\ 0 & m \end{pmatrix} \in M^{6\times 6} \qquad \text{(Equ. 29)}$$

where $\mathbf{f} \in F^6$ is the net spatial force applied in the body, which is compose by $3D$ vectors force f and torque n, $\mathbf{v}$ is the spatial velocity, composed by the linear and angular velocity of the body center of mass, $\mathbf{a}$ is the spatial acceleration and $\mathbf{I}$ is the spatial inertial, composed by the inertia tensor I_c and the mass m.

Inverse dynamics deals with the problem of obtaining the torques applied in every joint starting from the acceleration of the rigid body system. The generic formula can be expressed as

$$\tau = ID(model, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \qquad \text{(Equ. 30)}$$

The most used algorithm to calculate inverse dynamic is the Recursive Newton Euler Algorithm (RNEA) [37] whose spatial formulation can be found in [36]. This algorithm has a complexity of $O(n)$, where n is the number of degrees of freedom.

RNEA has two phases. First, it calculates recursively the velocity and acceleration of every joint, and then, using Equ.25 it calculates the force transmitted in every joint. In a second stage, it computes the joint forces starting at the terminal links and working towards the base.

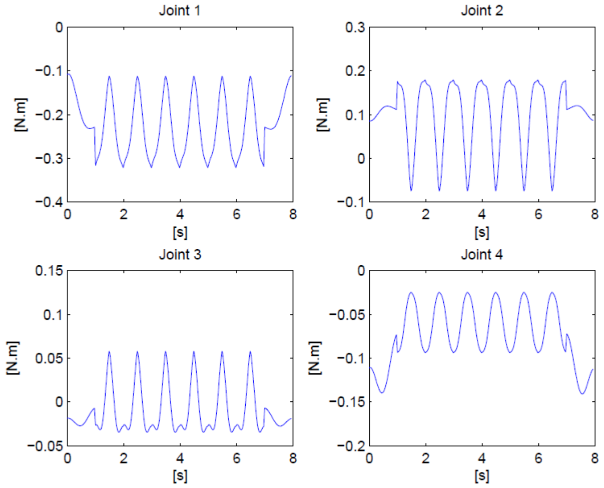


Fig. 13 Torque graphics for the left arm joints in a non-exaggerated motion

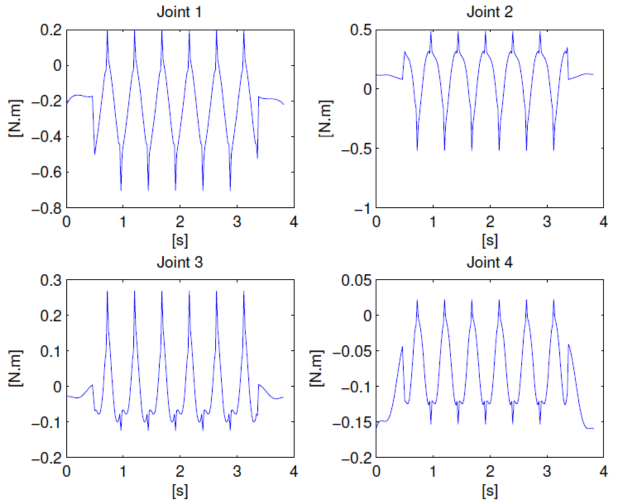


Fig. 13 Torque graphics for the left arm joints in an exaggerated motion

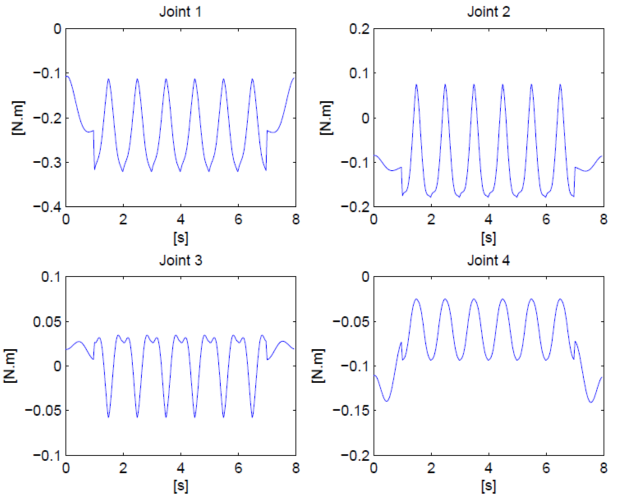


Fig. 15 Torque graphics for the right arm joints in a non-exaggerated motion

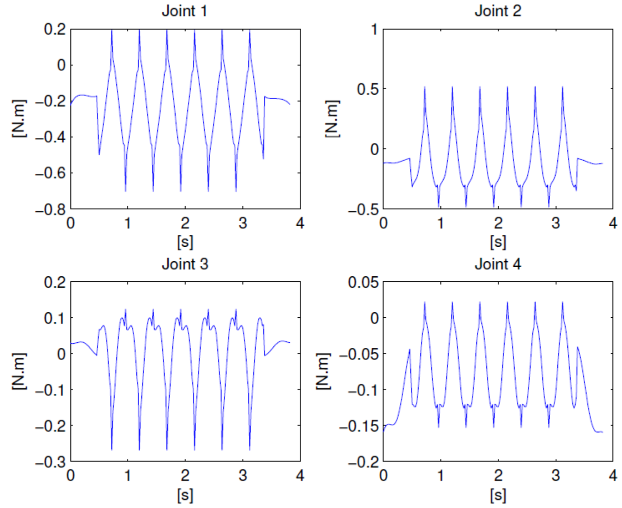


Fig. 16 Torque graphics for the right arm joints in an exaggerated motion

5. Proposed architecture

The aim of this work is to provide an innovative Human-Robot interaction that gathers a solution for each step of the problem.

The humanoid is provided with a set of two RGB webcams permitting to create a perception model with a rate of 25 frames per second. Furthermore, the perception system can feed a machine learning sub-system responsible for the face gesture recognition. The result of this evaluation is a degree of emotion that is levelled between 1 (very sad) and 5 (very happy) being number 3 a neutral state.

Also, robot kinematics is well known and compared with human kinematics so a kinematic adaptation can be done to make the robot imitate the human movements. This, in conjunction with the perception of the robot, makes possible to create a real-time robot interaction with a human reacting dynamically to the facial gestures played by a human in front of him. Fig. 17 covers the steps followed by the proposed method.

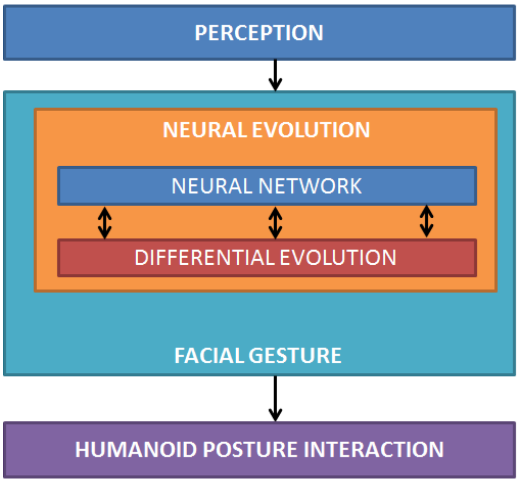


Fig. 17 Adapted trajectories for the humanoid's left arm.

A. Face expressions

The proposed method has been trained to recognise five different emotional states based on the face features. The following Fig.18 represents the variety of gestures that the algorithm is ready to classify and then react to.

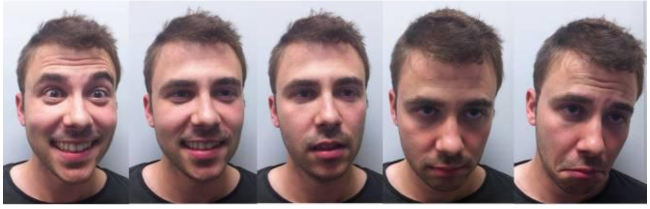


Fig. 18 Range of possible emotional states that the humanoid is prepared to identify. From left to right: very happy, happy, neutral, sad and very sad.

6. Experimental results

In the following section several experiments will be presented for each of the contributions. First of all a comparison between back-propagation and Neural Evolution for the NN optimization will be explained to determine their performance and error estimations. Afterwards the gesture recognition rates using values and experimental results.

A. Back-propagation vs. Neural Evolution

For the optimization comparison a face database has been used containing faces in different postures, gestures, orientations, with and without glasses for 20 people. This database has been provided by Carnegie Mellon University. The dataset is in PGM format with a fixed image size of 32x30 pixels. The desired target is to obtain the maximum recognition rate.

Error function for the learning step will be defined as the norm-1 difference between the estimated grade of happiness and the expected one. So, if a face is labelled as a level of happiness 3 and the result given by the NN is 4.65 the error for this sample is 1.65.

1) *Back-propagation learning.* For this experiment two parameters were modified. Number of hidden layers (between 1 and 4) and learning rate η (between 0.1 and 0.4)

TABLE 1
BACK PROPAGATION PERFORMANCE

Layers	Learning rate η	Success (%)
1	0.1	56.25
	0.2	55.289
	0.3	55.192
	0.4	58.173
2	0.1	61.058
	0.2	60.576
	0.3	62.981
	0.4	59.135
3	0.1	55.769
	0.2	57.212
	0.3	**56.730**
	0.4	55.769
4	0.1	58.173
	0.2	58.653
	0.3	67.788
	0.4	66.346

2) *Neural Evolution.* For this experiment two parameters were modified. Number of population (3, 10, 20, 50 and 100) and number of generations (2, 20 and 200)

TABLE 2
NEURAL EVOLUTION PERFORMANCE

Population	Learning rate η	Success (%)
3	2	75.4807
	20	75.4807
	200	75.4807
10	2	72.1153
	20	75.2403
	200	75.2403
20	2	75.4807
	20	73.557
	200	75.7807
50	2	72.1153
	20	72.1153
	200	**76.9230**
100	2	73.557
	20	70.1923
	200	70.432

As it is shown in the previous tables, best performance for gesture recognition is obtained using Neural Evolution. Therefore, learning step for the following experiments will be done using the proposed method.

3) *Neural Evolution convergence.* The performance of the optimization strategy is very significative and it has been analysed. As it is shown in Figure 19, the downward slope for the convergence of the algorithm is highly reduced as long as the number of iterations is increased. When the algorithm is close to the global minimum the speed is reduced. This might be solved using a hybrid method that mixes the evolutive optimization with a conventional optimization during the last part of the process.

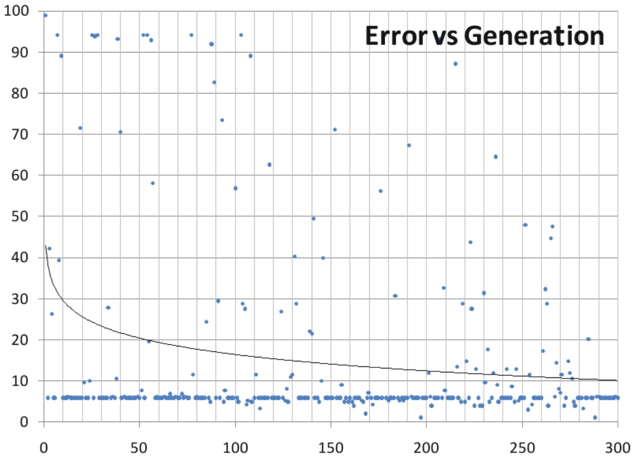


Fig. 19 Error distribution depending on the number of iterations of the optimization algorithm (Neural Evolution). Note the downward slope variations between the first iterations and the last.

B. Active Appearance Models

To compute the AAM algorithm, a public dataset of images, each of them with different faces and expressions, has been used as it has been explained before. 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 (see Fig.20).

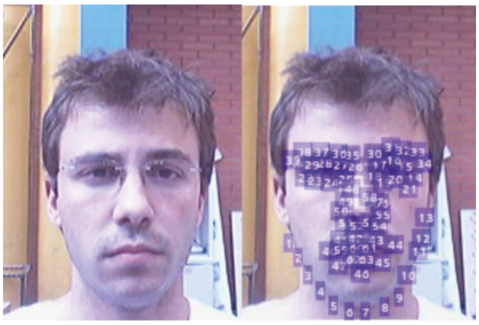


Fig. 20 Original face image on the left and the same image with landmarks of 63 characteristic points superposed on the right.

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 as in [38-39] (Fig. 21.a). The representation of the face model is made of splines using the face tracker (Fig. 21.b).

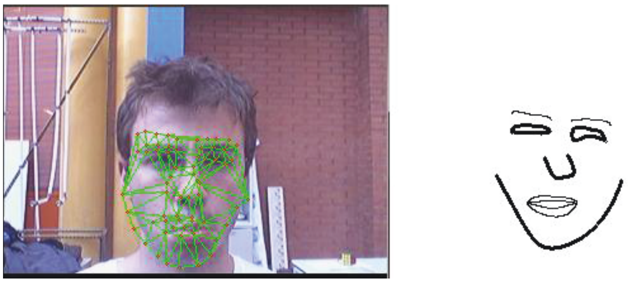


Fig. 21 Face tracking mesh based on Delaunay triangles on the left image (a) and face representation using splines on the right image (b).

C. Trajectory generation

Once the Neural Network is trained, it has been implemented in HOAP-3 robot a list of arm gestures so it is be able to interact with the person autonomously. 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 varies between very happy and very sad passing through neutral face gesture state.

D. Overall procedure

The complete algorithm corresponds to the face detection using Haar Cascades, then the face is introduced in the AAM algorithm and splines corresponding to the face features are extracted. Then those splines are evaluated as the input for the Neural Evolution algorithm and the happiness state is evaluated. This state makes the robot performs an adaptive response moving both arms. The following Figure represents some steps of the recognition procedure.

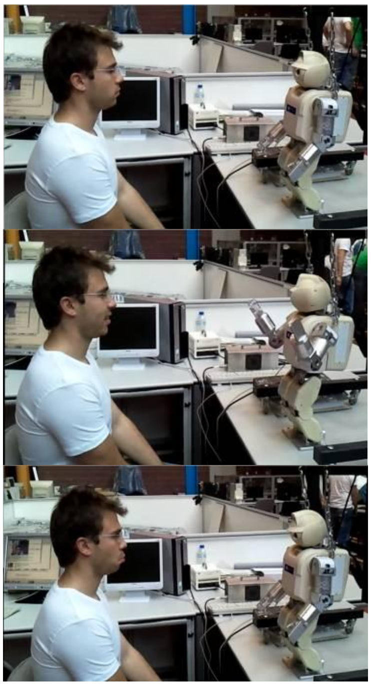


Fig. 22. Some snapshots of the overall experiment. Recognition of normal gesture, happy gesture and sad gesture with the pertinent performance of the robot.

7. Discussion

In this paper we propose a bidirectional interaction system between a human and a humanoid robot regarding facial emotions. The selected emotions are sadness, happiness, neutral (no emotion).

The human expressions are characterized by means of AAMs whose initialization is boosted by Haar Cascades. The output of the AAMs is a set of splines that define some facial features like mouth, eyebrows, eyes and nose, which are used to identify a continuous set of emotions, which ranges from very sad to very happy.

The identification of the emotional level is computed using a Neural Network optimized with a genetic algorithm, Differential Evolution. The main advantage of using DE instead of Back-propagation is because DE produces an improvement in the performance.

To generate the robot postural response we adapted two different teacher movements, sadness and happiness, to the size of the robot, using a visual tracking system. The trajectories are smoothed with cubic splines and then modified as a function of the time to represent the intensity of the postural response. If the human is very happy the robot response is faster and if the human is very sad the response is slower.

The proposed method was implemented in the humanoid HOAP-3 with the participation of several people proving the correct functionality of the system.

A. Key contributions

The proposed system, implemented in a real humanoid robot, allows to detect three different facial emotions in a continuous way that ranges from sadness to happiness, using Active Appearance Models initialized with Haar-Cascades and a Neural Network optimized with Differential Evolution.

Furthermore, the robot modifies the intensity of the postural interaction as a function of the intensity of the human emotion, producing slower postural responses when the human is sad and faster postural responses when the human is happy. All movements take into account joint and torque limits.

B. Comparison with related work

As it is stated by [1], this paper has been focused in determining emotions based on facial expressions with the aim of being replicated by the robot in a natural way.

Instead of acquiring information from hands as proposed by [40] this paper is focused exclusively on facial gestures avoiding the problematic of computing the estimation of occluded fingers.

Contrary to [41] this work achieves five states of expressions without processing finite elements in order to achieve faster processing times. As it is recommended by [23-24] this work has taken into account the head position, orientation and displacement in order to transmit happiness or sadness. Furthermore, the proposed study not only transmits the expression but also interact with the robot in a bi-directional situation.

The work proposed by [42] has an initialization problem when establishing the initial AAM in the image that has been solved in this paper by means of Haar Cascades Classification.

The main improvement with our previous work [29] is the way both human and robot interacts with each other. In [29] we created a bidirectional system that detects discrete emotional states of the human and produced a discrete postural response. The present work produces a more natural interaction because the detection of emotions is continuous and the robot response changes in accordance with the intensity of the human emotions.

Expressions have been based on [26-28] where whole body is used to generate the emotional patterns with the difference that the proposed work finds to recognize, interpret and reproduce the complete emotional response receiving the human facial expression and answering suitably.

Acknowledgment

The research leading to these results has received funding from the COMANDER project CCG10-UC3M/DPI-5350 funded by Comunidad de Madrid and UC3M (University Carlos III of Madrid), and ARCADIA project DPI2010-21047-C02-01 funded by CICYT project grant on behalf of Spanish Ministry of Economy and Competitiveness.

References

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

[2] 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).

[3] V.Gonzalez; A.Ramey; F.Alonso; A.Castro-González; M.A. Salichs. "Maggie: A Social Robot as a Gaming Platform". *International Journal of Social Robotic*. Vol. 3. No. 4. pp.371-381. (2011).

[4] Alm, Cecilia Ovesdotter, Dan Roth, and Richard Sproat. "Emotions from text: machine learning for text-based emotion prediction." *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics (2005).

[5] Schröder, Marc, and Jürgen Trouvain. "The German text-to-speech synthesis system MARY: A tool for research, development and teaching." *International Journal of Speech Technology 6.4 (2003): 365-377.Natural Language Processing*. Association for Computational Linguistics (2005).

[6] 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 2*3(5), volume 23, 840–851, (2007)

[7] 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)

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

[9] 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)

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

[11] Chung, K., Kee, S.C., and Kim, S.R.: "Face recognition using principal component analysis of Gabor filter responses", *Published by the IEEE Computer Society*, 53, 1999

[12] 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.anoid robot, IEEE Transactions on Robotics, Special Issue on Human-Robot Interaction 23(5), volume 23, 840–851, (2007)

[13] 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, (2008)

[14] Bernieri, Frank J., J. Steven Reznick, and Robert Rosenthal. "Synchrony, pseudosynchrony, and dissynchrony: Measuring the entrainment process in mother-infant interactions." *Journal of personality and social psychology* 54.2 (1988): 243.

[15] Beebe, Beatrice, Frank Lachmann, and Joseph Jaffe. "Mother - infant interaction structures and presymbolic self - and object representations." *Psychoanalytic dialogues*(1997): 133-182.

[16] Shumway, Stacy, and Amy M. Wetherby. "Communicative acts of children with autism spectrum disorders in the second year of life." Journal of Speech, Language and Hearing Research 52.5 (2009): 1139

[17] Cartmill, Erica A., and Richard W. Byrne. "Semantics of primate gestures: intentional meanings of orangutan gestures." Animal cognition 13.6 (2010): 793-804.

[18] Miklósi, Ádam, and Krisztina Soproni. A comparative analysis of animals' understanding of the human pointing gesture." *Animal Cognition* 9.2 (2006): 81-93.

[19] M. González-Fierro; D.Hernandez; P.Pierro; C.Balaguer. Dynamic Modelling of Humanoid Robots Using Spatial Algebra. *XXXIII Jornadas de Automática*. Vigo. Spain. Sep, 2012.

[20] A. Castro - González; M. Malfaz; M. A. Salichs. Learning the selection of actions for an autonomous social robot by reinforcement learning based on motivations. *International Journal of Social Robotics* Vol. 3. No. 4. pp.427-441. 2011.

[21] Shin'ichiro Nakaoka, Atsushi Nakazawa, Fumio Kanehiro, Kenji Kaneko, Mitsuharu Morisawa, Hirohisa Hirukawa, and Katsushi Ikeuchi. "Learning from Observation Paradigm: Leg Task Models for Enabling a Biped Humanoid Robot to Imitate Human Dances". *The International Journal of Robotics Research*, August 2007; vol. 26, 8: pp. 829-844.

[22] A. Castro - González; M. Malfaz; M. A. Salichs. "An autonomous social robot in fear. IEEE Transactions on Autonomous Mental Development". no.99, pp.1,1, (2013)

[23] Beck, A.; Cañamero, L.; Bard, K.A., "Towards an Affect Space for robots to display emotional body language," RO-MAN, 2010 IEEE , vol., no., pp.464,469, 13-15 (Sept. 2010)

[24] Beck, Aryel, et al. "Interpretation of emotional body language displayed by robots." *Proceedings of the 3rd international workshop on Affective interaction in natural environment*s. ACM, (2010).

[25] Saldien, Jelle, et al. "Expressing emotions with the social robot Probo.*" International Journal of Social Robotic*s 2.4 (2010): 377-389.

[26] Zecca, M. ; Mizoguchi, Yu. ; Endo, K. ; Iida, F. ; Kawabata,

Y. ; Endo, N. ; Itoh, K. ; Takanishi, A. "Whole body emotion expressions for KOBIAN humanoid robot - preliminary experiments with different Emotional patterns", 2009. *RO-MAN 2009. The 18th IEEE International Symposium on*, Page(s): 381-386 (2009)

[27] Zecca, M. ; Endo, N. ; Momoki, S. ; Itoh, K. ; Takanishi, A. "Design of the humanoid robot KOBIAN - preliminary analysis of facial and whole body emotion expression capabilities", *Humanoids 2008. 8th IEEE-RAS International Conference on* Page(s): 487- 492. (2008)

[28] Kishi, T. ; Otani, T. ; Endo, N. ; Kryczka, P. ; Hashimoto, K. ; Nakata, K. ; Takanishi, A. "Development of expressive robotic head for bipedal humanoid robot" *Intelligent Robots and Systems (IROS),* 2012 IEEE/RSJ International Conference on; Page(s): 4584 – 4589 (2012)

[29] J. G. Bueno; M.González-Fierro; L.Moreno; C.Balaguer. "Facial Gesture Recognition using Active Appearance Models based on Neural Evolution" 2012 *Conference on Human-Robot Interaction (HRI 2012).* Boston. USA. Mar, 2012.

[30] 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 (1998)

[31] Cole J "About face". *MIT Press*, Cambridge (1998)

[32] Mehrabian "A Communication without words." *Psychol Toda*y 2(4):53–56 (1968)

[33] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359.

[34] Price, K. V., Storn, R. M., & Lampinen, J. A. Differential evolution a practical approach to global optimization. (2005).

[35] Mitchell, Tom M. Machine learning. 1997. Burr Ridge, IL: McGraw Hill, 1997, vol. 45.

[36] Featherstone, R. (2008). Rigid body dynamics algorithms (Vol. 49). New York: Springer.

[37] Luh, J. Y., Walker, M. W., & Paul, R. P. C. (1980). "On-line computational scheme for mechanical manipulators". *J. DYN. SYS. MEAS. & CONTR.,* 102(2), 69-76.

[38] Viola, P., Jones, M. Rapid Object Detection using a Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition, 2001. Proceedings of the 2001 IEEE Computer Society Conference, Vol. 1, p. 511-518, 2001.

[39] Borouchaki, H., George, P. L., Hecht, F., Laug, P., & Saltel, E. (1997). "Delaunay mesh generation governed by metric specifications. Part I. "*Algorithms. Finite elements in analysis and design*, 25(1), 61-83.

[40] Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19(7), 677-695.

[41] Liu, H., Wang, X., & Zhang, Y. (2012, October). Analysis and Research of Humanoid Robot Facial Expression. In Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on (Vol. 2, pp. 315-318). IEEE.

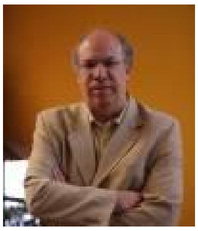
[42] Peyras, J., Bartoli, A., Mercier, H., & Dalle, P. Segmented AAMs improve person-independent face fitting. In BMVC'07-Proc. of the 18th British Machine Vision Conference. (2007).

Authors

Jorge García Bueno received his MSc. Degree in Electronics Engineering in 2009 and his MSc. In Telematics from University Carlos III of Madrid, Spain. In 2011 he received his Master Degree in Robotics and Automation. At the moment, he is a Ph.D. student in the Department of Systems Engineering and Automation of the University Carlos III of Madrid.

Miguel González-Fierro received his MSc. Degree in Electrical Engineering in 2008 from University Carlos III of Madrid, Spain. In 2009 he received his Master Degree in Robotics and Automation. At the moment, he is a Ph.D. student in the Department of Systems Engineering and Automation of the University Carlos III of Madrid.

Professor Luis Moreno received the Degree in Automation and Electronics Engineering in 1984 and the Ph.D. degree in 1988 from the Universidad Politécnica de Madrid, Madrid, Spain. From 1988 to 1994, he was an associate professor at Universidad Politécnica de Madrid. In 1994, he joined the Department of Systems Engineering and Automation, Universidad Carlos III de Madrid, Madrid, Spain, where he has been involved in several mobile robotics projects. His research interests are in the areas of mobile robotics, mobile manipulators, environment modeling, path planning and mobile robot global localization problems.



Professor Carlos Balaguer received his Ph.D. in Automation from the Polytechnic University of Madrid (UPM), Spain, in 1983. From 1983-1994 he was with the Department of Systems Engineering and Automation of the UPM as Associated Professor. Since 1996, he has been a Full Professor of the RoboticsLab at the University Carlos III of Madrid. He is currently the Vice-chancellor for research of the university. His research has included humanoid and assistive and service robots, among others. He participates in numerous EU projects since 1989, such as Eureka projects SAMCA, AMR and GEO, Esprit projects ROCCO and CEROS, Brite project FutureHome, IST project MATS, and the 6FP IP projects ManuBuild, I3CON, and Strep Robot@CWE. He has published more than 200 papers in journals and conference proceedings, and several books in the field of robotics.