Facial Emoticons

Expression Recognition as a Means of Interaction for Physically Challenged People

Sandra Gama, Daniel Gonçalves
Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, Portugal
Phone: +351 213100289, e-mail: sandra.gama@ist.utl.pt, daniel.goncalves@inesc-id.pt

Abstract — **Facial expressions are highly effective non-verbal means to share emotions among human beings. Effectively, many of these expressions are universally understandable. Nowadays, in the cybernetic panorama, emoticons are used as surrogate facial expressions in electronic-mediated communication. As a new modality to more easily interact with computers, it provides a natural means of interaction that may be particularly useful for physically challenged people, since it opens a feedback channel that makes it possible to interact without using a keyboard or a mouse.**

We present FacialEmoticons, a facial expression recognition library that can be used as the basis for meaningful interactions with motor-impaired users. After face tracking and feature detection, Bayesian Classifiers are used to infer the expression on the user's face. Our methodology requires low computational power yielding good results with low-latency rates and has recognition rates comparable to those of humans. Our results show the viability for facial-expression-based real-time interaction in the context of people with severe motor impairments.

I. Introduction

For human beings, the process of recognizing facial expressions is fairly straightforward. Actually, these consist of a series of specific features that often make them quite easy to understand, even among people of different social and ethnical backgrounds.

Ideally, human-machine interaction must be as similar to communication among humans as possible. Effectively, in the context of facial expressions, computer systems should be able to classify facial expressions and use this information to enhance interaction.

Face movement has been widely used as an interaction modality for people with special needs. Effectively, Frangeskides and Lanitis [7] have created a system that relies on head movements for interaction, showing that it is possible to perform most tasks with a high accuracy, and so have Chen et al. [4], who have created a head-movement-controlled mouse to be used by people with movement restrictions, These studies haven't, however, considered facial muscle movements.

As for facial expressions, some studies have taken these into account as a means of interaction for physically challenged people. Baklouti et al. [3] presents an application that allows people to control a four axis exoskeletal orthosis not only using facial movement, but also lip expression to virtually control the orthodotic limbs. However, they haven't yet made use of complete facial expression classification to provide more information to the system. Faria et al. [6] have developed a prototype for an intelligent wheelchair, which uses facial expressions specifically as driving instructions, allowing people who have severe hand movement restrictions to freely command their wheelchairs.

In order to use facial expressions as a means of interaction, it is necessary to adopt image processing and pattern recognition methodologies. A wide range of studies has been done in this area over the last two decades, such as the ones of Pantic and Rothkrantz [13], Bartlett et al. [1] and Lu et al. [9], which process facial images and associate them with corresponding states of emotion. Furthermore, Bartlett et al. [2] have developed a facial recognition system which, given facial frontal images, classifies the corresponding emotion. This system has been deployed on a range of platforms such as Sony's Aibo, ATR's RoboVie and CU Animator.

In this paper we present FacialEmoticons, a library that, besides taking advantage of the vast information of facial expressions, providing a rich non-conventional modality for interaction, is also a potentially useful tool for people with special needs, allowing a much easier interface for those who are physically challenged. Furthermore, being extremely versatile, FacialEmoticons may be used synergistically with any application. We present two prototype applications in which we have integrated our facial expression recognition library. We will succinctly describe how expression recognition takes place, as well as illustrate the interfaces we have created to show the functionalities of the FacialEmoticons library.

II. Recognizing Facial Expressions

We have followed a top-down approach to facial expression recognition, as illustrated in Figure 1.

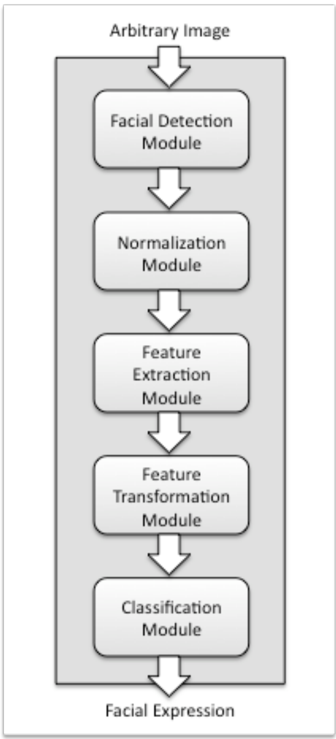

Fig. 1. Methodology for Facial Expression Recognition.

As such, the initial problem has been divided into five sub-problems: *Facial Detection*, which implies tracking the face and extracting its coordinates; *Normalization*, that enhances compatibility and performance by resizing facial images into a standard size; *Feature Extraction*, that determines, using the previous step's output, the coordinates of the main facial features; *Feature Transformation*, that maximizes and selects relevant information to process in the next module; and *Classification* which, using the facial features which have been inferred in the previous step, determines the corresponding facial expressions.

A. From Images to Expressions

A.1. Facial Detection

In the context of the present work, the user's environment consists of an arbitrary room, and his or her only instrument for interaction is a computer with a webcam. As such, the system must be able to process context-independent images; not only facial images with homogeneous backgrounds, but also more complex images where other background elements may be present. Additionally, it must have robustness against slight face rotations.

It was thus necessary to detect a face on an arbitrary image. Viola and Jones' method [15], since it uses Haar classifier cascades (which are computationally light), along with providing a means for real-time facial detection, seemed to be a proper choice.

Thus, not only the problem of facial detection has been solved, but also it is done in real time, which is an important factor.

A.2. Normalization

So the system may interact with any user who owns a webcam, and since devices present different resolutions, and users may be at any distance from the camera, it is important to consider that facial areas may present a variety of dimensions. So that facial feature extraction is done with enough precision, and since algorithms are sensible to image dimensions, it was necessary and pertinent to normalize these. This stage follows face detection. As such, only the face area is normalized, for resource maximization. Images are resized into standard dimensions before being processed for feature extraction.

A facial feature model has been adopted, which consists of facial features of an *average face*, to infer unknown facial values. Model dimensions are taken into account when normalizing a facial image, in order to grant high coherence and flexibility. Consequently, if the model is altered, the image's dimensions will accompany this transformation.

A.3. Feature Extraction

FACS, created by Ekman and Friesen in 1978 [5], is a technical norm that classifies facial behaviors depending on the muscles that are responsible for each facial action. Many other studies have considered FACS as a basis for their work. This method defines some (originally, 46) UA (Unitary Actions), which consist of the actions that correspond to one or more facial muscles. Different combinations of UA define a wide set of facial expressions. Accordingly with this norm, there is a vast spectrum of parameters that must be taken into account. These are predominantly related to eyes, eyebrows and mouth movements. Effectively, other features such as chin and cheeks are responsible for only 4 of the UA.

In order to maximize feature detection, a model of the human face has been created. This model allows not only to restrict the region of interest but also to estimate the position of facial features when these are not detected (caused by extremely poor lighting conditions, extreme face rotation or landmark occlusion). The model is dynamically adjusted to the detected features and, if it is impossible to obtain a certain feature, the model compensates for it. This grants the system an increased robustness.

Investigators from the university of Regensburg in Germany created two average faces (one for each gender) for investigation purposes in the field of psychology. Here, we have created a hybrid model by merging these two faces, which resulted in a realistic start point, well adapted to this study's necessities. It consists of a vector in which every facial feature is represented, according to the standard dimensions defined on the face model. It presents an elastic behavior, being progressively adapted to the facial features that are localized. Additionally, this model is used as a feature detection aid, since it substitutes the coordinates that haven't been correctly detected, preventing from possible errors. In these cases, and since it is an elastic model, the coordinates of undetected features will assume the value which the model presents to its corresponding features. This model is illustrated in Figure 2.

Fig. 2. Face Model.

The methodology that has been adopted for facial feature detection implies five stages:

Detection of a region of interest: Here we take advantage of the OpenCV library functions; this step helps reduce computational weight corresponding to the following steps. In the case of the eyes, four fundamental points are considered: the leftmost, rightmost, topmost and down-most. These make it possible to find out the width and height of the eye, which are fundamental for expression classification. In order to improve performance, only the upper half of the face has been considered. When analyzing the eyebrows, we didn't use any cascade classifier. In spite of that, a tighter region of interest has been defined, which processes only the area that is located right above the uppermost point of the left eye, taking advantage of the symmetric properties of the human face to maximise the efficiency of detection. This decision has been considered carefully. On one hand, it implies a slight decrease in the system's robustness in cases of people who have suffered from an injury or disease that prevents them from moving their frontalis muscle (the one which is responsible for eyebrow raising). On the other hand, given the objective of developing a light, simple and fast method for feature detection, this seemed to be a proper decision. Mouth detection followed the same steps as eyes detection. However, since the cascade classifier didn't have quite a satisfactory performance, it was used only as a basis for detection In this case, special attention has been paid to defining the mouth's region of interest based on the elastic model which has been adopted.

Grayscale conversion: This implies a pre-processing for the application of the Canny operator, which manipulates single-channeled images. It also simplifies Gaussia Blur processing, since applying its convolution matrix to one channel is more efficient than applying it to three color channels (RGB).

Gaussian Blur: This algorithm reduces noise and other artifacts that are unnecessary for Canny edge detection. Although Canny, by default, applies a Gaussian Blur filter before detection, there are many artifacts that present high dimensions. This additional Gaussian Blur operator eliminates these artifacts. In preliminary tests, it proved its consistency, especially in the eye region, where dark circles or lines represent additional elements, and in the mouth area, where teeth or the junction between lips may difficult the detection process.

Canny: A Canny algorithm has been invoked for tracking facial features with more precision. This operator has been chosen for many reasons. Firstly, because it made sense to use an algorithm which was fast and implemented by OpenCV. This constraint reduced our choices into Canny, Sobel and Laplace. Secondly, because there was a need for a noise-robust algorithm. Sobel and Laplace algorithms presented a great sensibility to noise. Lastly, because the position of the detected edges should be as precise as possible. Also through this perspective, Canny presented better results when compared to Sobel or Laplace.

Coordinate extraction: By analysis of last step's results, this stage consists of obtaining values corresponding to the detected facial feature. For eyes, extreme points are calculated based on topmost, down-most, leftmost and rightmost tracked points, and normalized so that the final result is symmetric, which is done through the average of the corresponding coordinates. The process is very similar for the mouth. Extreme points define the maximum and minimum values in both axes and the average between them makes it possible to generate a geometric form. For eyebrows, their average point is calculated, since their height is the most important factor. After that, the rest of this feature is approximated through the use of the model we have adopted throughout this study.

A.4. Feature Transformation

The features used for classification have been selected in order to maximize the information available to this process. These features are based in the concept of Action Units, defined in FACS. Although they may provide a wide set of information, if looked at separately, little can be inferred on their corresponding expression. Therefore, a combination of several UAs, as defined by Ekman and Friesen [5], is used.

In order to improve the expressiveness of the features that have been collected in former stages, these have transformed into another set of features. These features are: (1) Vertical distance between eyes and eyebrows; (2) Eye aperture; (3) Mouth aperture; (4) Mouth width; (5) Average vertical distance from mouth corners to eyes; and (6) Average vertical distance from mouth corners to mouth center.

The vertical distance between eyes and eyebrows, as pictured in Figure 3, determines eyebrow elevation and depression, thus providing valuable information on the detection of facial expressions such as surprised, angry or sad.

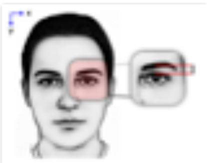

Fig. 3. Vertical distance between eye and eyebrow.

Eye aperture, as illustrated in Figure 4, helps distinguish facial expressions such as surprised, where eyes are wide open, from angry and happy, in which eyes are more closed.

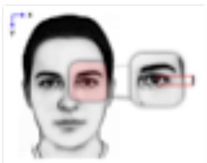

Fig. 4. Eye aperture

Mouth aperture, measured in accordance to Figure 5, provides information for determining facial expressions such as surprised or angry. In the first, the mouth is usually open and, in the latter, the mouth is usually completely closed.

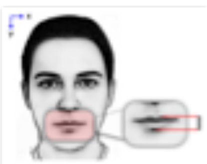

Fig. 5. Mouth aperture

Mouth width, as seen in Figure 6, allows for additional information on several facial expressions, like angry, happy, or sad. While in the case of an angry expression the mouth is usually compressed, when considering happy or sad expressions, it is usually more extended.

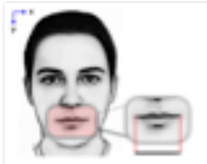

Fig. 6. Mouth width

The average vertical distance from mouth corners to eyes is measured as shown in Figure 7. Its main goal is to distinguish between sad and happy facial expressions. In a sad face, this value is considerably higher than on a happy face.

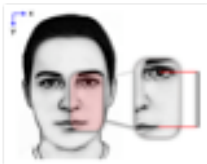

Fig. 7. Vertical distance from mouth corners to eyes.

As for the average vertical distance from mouth corners to its center, which is determined as seen in Figure 8, this feature is used in conjunction with the previous, to differentiate between sad and happy expressions.

Fig. 8. Vertical distance from mouth corners to mouth center.

A.5. Classification

Bayesian classifiers have been adopted, mainly because they make it possible to obtain good results despite their quite low computational weight. Two classifiers have been taken into account. Firstly, a more conventional approach to the Bayesian classifier, using discreet decision intervals, has been chosen. These intervals are associated with the values of samples' features, being the whole classification process based on a discreet set of intervals. However, and since the index of performance wasn't sufficiently satisfactory, we afterwards adopted a Gaussian Bayesian classifier, in which Gaussians take the place of intervals (in this case, only a Gaussian for each feature) for representing training values for each feature.

Bayesian Classifiers using Discreet Decision Intervals: The process of classification begins by testing many facial images in order to obtaining feature reference values and understand their distribution along the domain. Using these values, we esteemed that 5 intervals would be sufficient for discretizing the domain, taking into account reference values, in order to afterwards proceed for the Bayesian classification. These intervals were created so that each one would contain about 20% of the reference samples. When trying to adjust the number of intervals, we verified that, with a decrease, the classifier's quality decreased, and with an increase it did not bring consistent improvements.

The training process was automated through the development of an application for this intent. During it, both 10 positive samples and 10 negative samples are selected. Each one of these is introduced in the classifier's training module together with additional data (denomination of the class for training and information on whether the samples are positive or negative). The training module, using this data, fills in, for each class of emotion, a set of positive and negative structures which belong to each interval of each feature. The distribution of samples along the features intervals presents the probabilities that are presented to the Bayes classifier.

Subsequently, classification consists of calculating the likelihood of a sample feature corresponding to a given emotion. This value is calculated based on the formula

$$L_c = \prod_{f=1}^{N} \frac{HP_f}{TP_f}, \qquad (1)$$

in which L_c is class c's classification, HP_f is the number of positive hits for the feature f and TP_f is the total number of training samples for class f.

This classification method has a disadvantage, which is related to the continuous nature of features. In the process of assigning sample values to intervals it sometimes happens that, while an interval k has a large number of samples and an interval k+2 also has a large number of samples, an interval k+1 has zero samples. This greatly influences the classification process. As such, we decided to model an infinite set of training samples through the use of a Gaussian, i.e., an approximation to the normal distribution.

Gaussian Bayesian Classifiers: A Gaussian Bayesian classifier is very similar to the previous one. However, it doesn't use the values of training samples for classification. Instead, it uses an estimation of values for infinite samples, assuming that these follow a normal distribution As such, the likelihood of a certain value belonging to a given class is calculated through the cumulative distribution function (c.d.f.) of the Gaussian distribution that is generated from training samples. Gaussian distributions are estimated through the mean and standard deviation of test samples. Thus, c.d.f. is given by

$$cdf(x) = \frac{1}{2}\left(1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right), \qquad (2)$$

where μ is the sample values' mean and σ is their standard deviation. erf (z) is the error function associated with the integration of the normalized form of the Gaussian function given by

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt, \qquad (3)$$

The training of the classifier is, once more, done through both positive and negative samples, so that it is possible to determine the likelihood of a sample belonging or not to given class. In this stage, two main values are stored for each feature: the sum of all samples' values, which is used to calculate the samples' mean, and the sum of the squares of all samples' values, which is used to calculate the samples' standard deviation. The number of samples isn't known until the training process has been concluded. As such, both mean and standard deviation are calculated at the classification stage. Consequently, the calculation of the standard deviation is done using the equation

$$\sigma = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right) - \bar{x}^2} =$$

$$= \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right) - \left(\frac{1}{N}\sum_{i=1}^{N} x_i\right)^2} = \qquad (4)$$

$$= \frac{1}{N}\sqrt{N\left(\sum_{i=1}^{N} x_i^2\right) - \left(\sum_{i=1}^{N} x_i\right)^2},$$

where N is the number of samples, x_i is sample i's value and $\bar{x}$ is the samples' mean.

The process of classification of a sample begins by calculating the likelihood of each feature belonging to each class. This computation is done through the value of the c.d.f. of a normal distribution, which is generated given the mean and standard deviation of training samples. The likelihood of a sample not belonging to a given class is computed as well. The classification process ends with the attribution of the sample to the class that has a higher likelihood, and summarized as

$$Lc = \prod_{j=1}^{N} \frac{0.5 - fda(Z(x_f, \mu_f^p, \sigma_f^p))}{0.5 - fda(Z(x_f, \mu_f^n, \sigma_f^n))} \quad (5)$$

where L_c is the likelihood of class c, x_f the value of feature f, relative to sample x, μ_f^p the mean for feature f's value for positive samples, σ_f^p the standard deviation for feature f's values, μ_f^n the mean of feature f's value for negative samples, σ_f^n the standard deviation for feature f 's value for negative samples and $Z(x, \mu, \sigma)$ the fit to the standard normal, which is given by

$$Z = \frac{X - \mu}{\sigma}. \quad (6)$$

The classifier associates the sample with the class c that presents the highest Lc value for that sample.

B. Evaluating Expression Recognition

In order to evaluate the behavior of facial expression recognition, and with sight to performance enhancement, several user tests have been conducted, using a standard laptop PC with an integrated webcam. Two scenarios have been considered for every test setting: (1) distinguishing between happy and sad faces and (2) distinguishing among five facial expressions: happy, sad, neutral, angry and surprised. The first considers the two most common opposed expressions, and the second all the library recognizes.

The first test setting corresponds to facial images from a University of Dallas' facial database [12]. These images consist of pictures that have been taken in a controlled environment. Tests have been conducted to a total of 936 images, 203 of which correspond to the class 'happy', 41 to 'angry', 570 to 'neutral' 55 to 'sad' 67 to 'surprised'. Success rates have been of 80.76% for the 'Happy vs. Sad' setting and 55.87% for all classes of emotions. Results are summarized in Table I. Through additional testing, we verified that most recognition failures were due to incomplete feature detection, rather than the classifier itself (78% and 72% for each scenario, respectively).

Table I: Hit-rates for the first test setting.

Scenario	Hit-rate (%)
Happy vs. sad	80.76
All expressions	55.87

In order to better understand the behavior of FacialEmoticons in a real-life usage situation, we designed an additional test, where 10 frames per expression, per user, were gathered and classified (instead of just one frame). The most common classification among these 10 frames was then selected and considered to be the facial expression corresponding to that group of frames. 10 users participated in this test, and they were asked to do each one of the 5 facial expressions: happy, sad, surprised, angry and neutral. The lighting conditions were, once more random. In this case, the hit-rate for the "happy vs. sad" scenario was of 85% while the hit-rate for the scenario with all classes of emotions was of 59%, as shown in Table II.

Table II: Hit-rates for the second test setting.

Scenario	Hit-rate (%)
Happy vs. sad	85
All expressions	59

These are not perfect recognition rates. However, some facial expressions are hard to identify or ambiguous. In order to find a comparison baseline, we measured the recognition rate of humans. Each of 10 subjects was presented 335 facial images (the same 65 corresponding to each one of the facial expressions we have collected), and asked to classify each one of these. In the first scenario ("happy vs. sad"), an average hit-rate of 95.89% has been reached. In the second scenario ("all expressions"), the hit-rate was of 64.24%, as shown in Table III.

Table III: Hit-rates for the third test setting.

Scenario	Hit-rate (%)
Happy vs. sad	95.89
All expressions	64.24

Taking these values as a baseline and comparing them with FacialEmoticons' recognition rates shows that our library performs nearly as well as humans. When considering happy and sad faces, it correctly recognizes nearly 90% of human-recognized expressions. For all expressions, the classifier recognizes 92% of them. These results are summarized in Table IV.

Table IV: Results of automatic classification compared with human performance.

Scenario	Hit-rate (%)
Happy vs. Sad	89
All expressions	92

III. Prototype Applications

Two prototype applications have been developed to demonstrate the feasibility of FacialEmoticons as a means of interaction and its usefulness for people who suffer from severe movement restriction.

A. Facial Emoticons Application

The main objective of this application, depicted in Figure 9, is to illustrate the usage of the FacialEmoticons library as a means for easy insertion of emoticons in any active window.

The prototype runs in background and, visually, consists of a video capture window. Whenever the user wishes to insert an emoticon, he or she only needs to activate a trigger. For preliminary tests, it hasn't been possible to gather enough people who suffer from severe movement restrictions to test our interface. Since we tested with unimpaired users, we defined the F12 key as a trigger for emoticon insertion. With real users other triggers can easily be considered, such as a head movement or a voice commands, depending on context and application.

After gathering image information and processing it, the application inserts a combination of keyboard symbols, corresponding to the user's facial expression, in the active window. It may be used on instant messaging applications, e-mail or on any other software programs. This prototype, being extremely minimalist, proves the simplicity and easiness of inserting an emoticon through a new modality, especially useful for people with motor or visual restrictions, since they only need to activate a very simple trigger to obtain the corresponding facial expression.

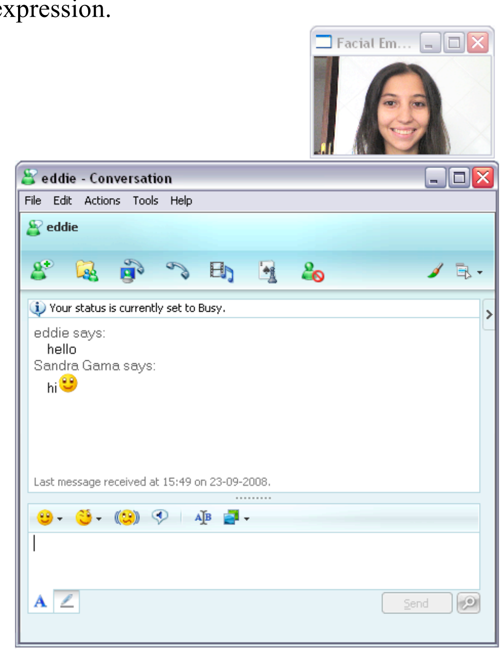

Fig. 9. Facial Emoticons prototype interface.

B. E-Motional Jukebox

This prototype, which may be seen in Figure 10, has been developed with the objective of showing the feasibility of non-conventional modalities for interaction with an audio player. A multimodal interface has been created which consists of gestures to control basic audio functions (such as "play", "pause", etc.) and facial expression recognition for track classification, so that the application has an intelligent behavior when selecting tracks to play. Two cameras are used, to capture both hand gestures and face simultaneously. Hand gesture recognition is done through invocation of the HandVU[8] library functions and facial expression recognition uses FacialEmoticons library. In the context of this application, only happy and sad expressions have been considered. Facial expressions are captured at a time interval of 5 seconds and cumulatively classified, so that the track's classification consists of the user's global appreciation throughout the whole duration of the music track. Although gesture recognition has been used for this prototype (which makes sense for users who don't suffer from severe movement restrictions for both hands, to be laying on their bed, controlling E-motional Jukebox's audio functions), the main idea behind it is that it is possible to use only facial expressions to control the jukebox. Actually, this prototype shows that it is possible to create a player that knows when to stop playing a certain track by classifying the user's expression, being also able to infer the user's taste based on previous classifications and, consequently, play new music tracks based on that. Consequently, it is possible for a person to, without moving, play the music of their preference, which is particularly relevant for people who suffer from severe movement restrictions, such as tetraplegia.

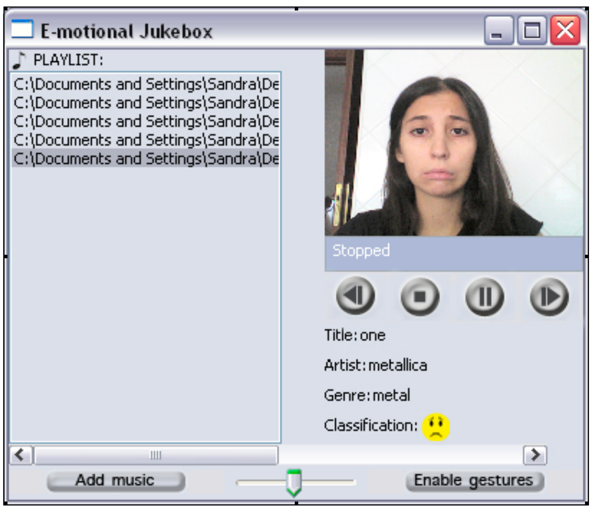

Fig. 10. E-motional Jukebox prototype interface.

IV. CONCLUSIONS

Facial expressions, being effective means for sharing emotions among humans, provide a potentially rich way of interaction. Despite that, this modality hasn't been widely used as a form of interaction for people with special needs, who have severe difficulties in using either a keyboard or a mouse.

FacialEmoticons, therefore, takes advantage of facial expressions and provides a library that, along with prototype applications, presents much easier means for a non-conventional interaction, especially relevant for people with special needs. We have created a system that can classify arbitrary human facial images with a satisfactory success rate. We have built Facial Emoticons Prototype, which proves the viability of our library for synergistic use with other applications. This application may be used with any other to insert emoticons in the active window. We have also created E-Motional Jukebox, which is a prototype that uses facial expressions for cumulative track classification, exploring the possibility of creating non-conventional audio players that are personalized and easily used by physically challenged people, showing that it is possible to control an audio player only through facial expression recognition.

In the future, we intend to further explore the potential of this modality by performing tests with motor-impaired users, after crafting an interface that addresses some of their problems. Currently, an environmental control system is being developed, that will provide the basis for these tests.

REFERENCES

[1] M. Bartlett, G. Littlewort, I. Fasel and J. Movellan, "Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction", *CVPRW Proceedings*, pp. 53, 2003.

[2] M. Bartlett, G. Littlewort, T. Sejnowski and J. Movellan, "A Prototype for Automatic Recognition of Spontaneous Facial Actions", *Advances in Neural Information Processing Systems*, MIT Press, 2003.

[3] M. Baklouti, S. Couvet and E. Monacelli, "Intelligent Camera Interface (ICI): A Challenging HMI for Disabled People", in *Advances in Computer-Human Interaction Proceedings*, Fev. 2008, pp. 21-25.

[4] Y. Chen, W. Chen, T. Kuo and J. Lai, "A Head Movement Image (HMI)-Controlled Computer Mouse for People with Disabilities", in *Disability and Rehabilitation*, Feb. 2003, Volume 25, pp. 163-167.

[5] P. Ekman and W. Friessen, "Facial Actions Coding System (FACS): Manual", Palo Alto: Consulting Psychologists Press, 1978.

[6] P. Faria, R. Braga, E. Valgôde and L. Reis, "Interface Framework to Drive an Intelligent Wheelchair Using Facial Expressions", in *IEEE Int. Symposium on Industrial Electronics Proceedings*, Jun. 2007, pp. 1791-1796.

[7] F. Frangeskides and A. Lanitis, "A Hands-Free Non-Invasive Human Computer Interaction System", in *Advances in Systems, Computing Sciences and Software Engineering Proceedings*, 2006, pp. 235-242.

[8] M. Kolsch and M. Turk, "Robust Hand Detection", in *IEEE Intl. Conference on Automatic Face and Gesture Recognition Proceedings*, 2004.

[9] H. Lu, Y. Huang, Y. Chen and D. Yang, "Real-time Facial Expression Recognition Based on Pixel-Pattern-Based Texture Feature", *Electronic Letters Proceedings*, 2007, pp. 916-918.

[10] C. Mauri, T. Granollers, J. Lorés and M. García, "Computer Vision Interactions for People with Severe Movement Restrictions", *Interdisciplinary Journal on Humans in ICT Environments*, vol. 2, April 2006, pp. 38-54.

[11] J. Min, K. Lee, S. Lim and D. Kwon "Human-friendly interfaces of wheelchair robotic system for handicapped persons", in *IEEE Int. Conference on Intelligent Robots and Systems Proceedings*, 2002, pp. 1505-1510.

[12] M. Minear and D. Park, "A Lifespan Database of Adult Facial Stimuli", in *Behavior Research Methods, Instruments and Computers*, 2004.

[13] M. Pantic and L.J.M. Rothkrantz, "Expert System for Automatic Analysis of Facial Expressions", in *Image and Vision Computing Journal Proceedings*, 2000, pp. 881-905.

[15] P. Viola and M. Jones, "Robust Real-Time Object Detection", *Technical Report*, University of Cambridge, 2001.