Imitating Personalized Expressions in an Avatar through Machine Learning

Cassondra Puklavage, Alexander Pirela, Avelino J. Gonzalez and Michael Georgiopoulos
Intelligent Systems laboratory
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816 2362
USA

Abstract

This paper describes a method for automatically reproducing personalized facial expressions in a computer generated avatar and a proof of concept test is implemented to evaluate the preliminary viability of the method. The method employed is a combination of an expression recognition system derived from the Facial Action Coding System, mapped by a Particle Swarm Optimization algorithm onto a computerized facial rig with 'musculature' based on the Facial Action Coding System. This approach is analyzed in depth, and an initial test of the algorithm is implemented and tested with promising initial results.

Introduction

Computer animation of humans have made great progress in recent years, allowing their role to expand beyond 'canned' animations in movies or video games, to dynamic interactions that are generated in real-time with substantial realism. Furthermore, advanced computer graphics now permit the reproduction of the face of a specific human individual in sufficient detail to be quite convincing to a human viewer. [1] At present, however, even the most realistic avatar is limited by its ability to simulate the expressions of its living human subject. The face of an avatar may look quite similar to a given human subject, but if that face is animated with expressions in a generic and/or artificial manner, the illusion of the likeness can be shattered. At present this is typically overcome using complex and expensive motion-capture technology to record an actor's motions and expressions down to the level of individual nuance, allowing these to be recreated in the avatar. These approaches require many hours of work by highly skilled digital artists and programmers. While these approaches are generally successful, the extensive labor by trained professionals for even simple sequences make them beyond the reach of many potential users of such avatar technology.

The term *avatar* can refer to a variety of things, even when narrowing the field to computing. The specific avatar referred to here is a virtual representation of an artificially intelligent entity, intended to make user interaction with the computer more effective and/or efficient.

In general, most people can be said to be more comfortable dealing with other people than with machines, particularly when using the machine requires learning a new and often unintuitive system. This is particularly true for young children, the elderly and those not computer literate. Utilizing an avatar can provide a more familiar and intuitive interface when the human can interact with the avatar in spoken language. Using the system would be as simple as carrying on a conversation.

However, while the end result is simple to use, implementing an effective, convincing avatar is a substantial technical challenge. In addition to requiring effective natural language processing and intelligent dialog management - a non-trivial feat in itself - an avatar's effectiveness depends largely on the realism of the animation, which must pass the user's scrutiny. Modeling humans, especially human faces, has long been considered among the most challenging computational tasks, not because it is inherently more complex than other objects, but rather because, as humans in a society, we have substantial experience with actual humans. Because of this long-term experience, we easily pick up on even minute dissimilarities in avatars.

When these dissimilarities become evident, the avatar becomes more noted for its differences than for its realism, and rather than producing improved empathy, will appear 'zombie-like' or 'off' and tend to inspire mistrust or even revulsion. This region between cartoon-like animation and photorealism, where an avatar's empathic nature abruptly drops off, is called the *uncanny valley* [11].

The intent of this project is to develop a machine learning method by which a particular human's personal facial expressions (i.e., emotional facial contortions) are recorded and automatically analyzed to extract their distinguishing characteristics, all without the expensive motion capture approaches mentioned above. These are then parameterized so that a dynamic avatar may reproduce and combine them as needed, all with minimal external control. Our application, appropriately implemented, could reduce this to a few hours of runtime on a common desktop

workstation—no more than would currently be invested in rendering a single frame via motion capture methods.

The most unique characteristic of our approach is the comparative simplicity and (eventual) autonomy of the process. The key element is not to recreate the emotional expressions of an individual, which has been accomplished to varying degrees in other applications. Rather, the key utility—and key difficulty—of our work is to create a system that can achieve this on command, through a (comparatively) simple process that only requires the input and control of minimally trained end-users, using still images of the individual in question. Such an application would present great advantages to the fields of entertainment and computer-based education.

In order to quantify the facial expressions, our machine learning approach first analyzes input images of an actor's face on a digital photograph, and uses facial recognition algorithms (camera study) to locate reference points, such as the eyes, lips, brows, etc. In subsequent images, the actor models the six basic emotions, and the system assesses the changes in the previously identified reference points. It then proceeds to build an expression using these data for an avatar that represents the same specific human.

In keeping with this objective, this paper describes a method for achieving such results and evaluates a proof-of-concept prototype to determine the viability of this proposed method. The paper is organized as follows: In the Literature Review section we review the pertinent prior work carried out in this and related fields. In the third section (Facial Action Coding System) we discuss the Facial Action Coding System that is central to our approach. In the section after that we review the intent and development of avatars as interactive aids, and discuss the avatar utilized in this project. The next section provides an overview of the Particle Swarm Optimization machine-learning algorithm implemented in this project. The experimental method that was used is then discussed along with the results. In the final section we summarize our findings and present our conclusions.

Literature Review

Several projects have been reported in the literature that have some similarities to our work. Their results were invaluable in the formulation of our approach. However, we were unable to find an exact precedent to our work.

The problem of quantification of facial expressions has been addressed in several studies. For example, a 2004 study by Bartlett et al [2] utilizes several machine-learning methods to classify still images. Videos are checked in a similar manner, going through and classifying the expression for each frame. These findings are then charted versus time, and a smooth graph is created based on changes in expressions recognized. This system claims over 90 percent accuracy in both still and video input in recognizing facial expressions. Nevertheless, this system can only identify expressions—unlike our application, it cannot reproduce them.

The facial recognition model chosen for our work, however more closely resembles those described in [12] and [17], which utilize swarm intelligence algorithms, although other methods were also evaluated. However, there are subtle but significant differences. With regards to [12], the data source used in this study is a 3D facial scan, modeled as a point cloud, whereas our study analyzes a simple 2D image. The more fundamental difference, however, is that the application in this study is to identify the emotion corresponding to the facial expression being made, by comparison with existing criteria. Our study takes a nearly diametrically opposed focus: Our end application is to identify the unique nuances that differentiate a given expression. In other words, the cited study would take a set of individuals and determine that they are all smiling; our study would take a set of smiles and determine what makes each one unique.

The focus of [17] is on identifying the presence of faces in a given image (using PSO); our paper, in contrast, deals with analyzing the facial expression shown in a given image, which is already known (or assumed) to contain a face. Our objective is not to identify or classify the face or expression in the image, but rather to determine its distinguishing characteristics in a quantitative manner that can then be mapped onto a digital avatar.

In other related work, Teller and Veloso [16] selected a genetic programming derivate, called Parallel Algorithm Discovery and Orchestration,. However, genetic programming generally requires too much external control, so this method was not selected for our program. Similarly, Khashman [8] uses a novel back-propagation learning algorithm, augmented by a confidence-nervousness variable, with excellent results. The back-propagation algorithm, however, is a supervised learning system, and would therefore most likely require skilled operators. A method with possible utility to further developments of our application is proposed by Mpiperis et al [13], with the specific advantage that it is designed for law enforcement and security use and is largely self-contained. Their approach, however, requires an unnecessarily elaborate method for our purposes.

The concept of imitation learning, both in the colloquial sense and more formally as scientific method, has notable analogues with the method used in our study to replicate facial expressions. Nevertheless, the overwhelming majority of research on imitation learning regards the development of robotic behavior, with control of robotic hands making up the bulk of studies. Among those studies with a less common focus, none was found to apply to the reproduction of facial expressions, much less to individualized expressions. Oztop et al [18] investigate the mechanisms by which an infant learns gestures and behaviors through imitation, and applies these concepts to the semi-autonomous generation of robotic behaviors. Wood and Bryson [19] investigate social imitation learning of complex, "program-level" actions between digital avatars in a virtual reality environment (the game Unreal Tournament). Ariki et al [20] focus on whole-body level

actions, though primarily lower-body, with the added input of balancing efforts by analysis of the reaction force on the imitator's (robot's) feet.

Facial Action Coding System

The *Facial Action Coding System* (FACS) is a system of notation developed by Ekman and Friesen [5] to record facial movements in a quantified manner, providing far more information than a qualitative observation. The parameters exhaustively catalog all possible changes in facial expression caused by individual muscular contractions. Thus, the system can accurately describe any facial movement, except those affected by facial disfigurement or paralysis. This universality is important to our work, as our model must be able to recognize and recreate any possible expression with exceptional fidelity—a more generalized system of descriptors would not be able to satisfy these requirements.

Despite being derived from the results of muscular contractions, FACS does not directly record muscular contractions. Instead, it uses a derived system of "action units", which the creators designed to more accurately reflect facial actions. Such a system is necessary where the contractions of multiple different muscles have indistinguishable effects, and conversely where variations in the contraction of a single muscle produce distinct effects: The action units have therefore been tailored to match the observable end results, rather than rigidly imitating muscular movement. This dovetails neatly with the desired characteristics of our approach.

Experimental Avatar Testbed

Our main focus on this investigation is to animate an avatar, rather than creating one from scratch. Thus, several pre-existing avatars were evaluated for our use. The modeling suite Poser 7 [15] was initially considered. It is capable of substantial detail and minute control over the facial mesh. However, this program is closed-source.

Another program called CANDIDE [4] is open-source, but is no longer supported by its creators, thereby preventing us from obtaining the source code, which by now is somewhat obsolete.

Ultimately, the Object-oriented Graphics Rendering Engine, abbreviated OGRE [14], was selected. See Figure 1. In particular, we used a demo application designed to showcase OGRE's capabilities for facial animation. Like its parent application, Blender [3], OGRE, is open-source and currently in widespread use, contributing to easy availability of the source code and providing considerable support in its technical aspects. A major limitation of OGRE, however, was its restricted control of the demo's facial features. As a result, it may seem like only the mouth is moving. In reality the eyebrows are also being adjusted, but it is difficult to notice these changes in the images that are provided throughout the paper.

As mentioned earlier, the facial mesh can only be manipulated via the adjustment of 17 sliders. Each slider representing a compounded action, such as enunciating a specific phoneme, individual muscles cannot be controlled or manipulated. The interface for these sliders is depicted in Figure 1. The use of these sliders limits the range of expressions capable with the OGRE avatar. An avatar suitable for a finalized implementation should be rigged according to the Action Units described by FACS, in particular those Action Units most closely associated with the expression of emotion in a human face. Nevertheless, the number and location of the sliders does not change the basic functionality of our algorithm, and as such, the OGRE demo was selected as an appropriate platform for an initial experiment. The approach taken and the experiments used to evaluate our work is described next. We begin by describing our machine learning approach.

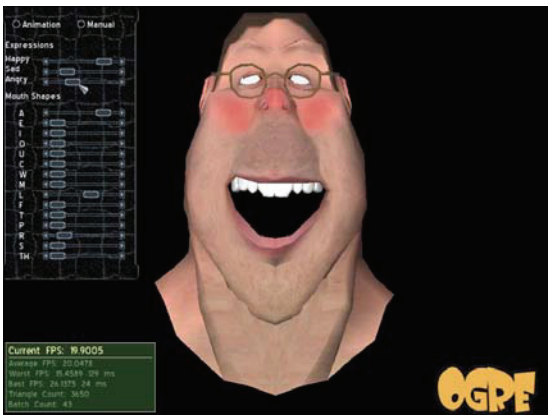

Figure 1 - Original program showing sliders that control facial expression (image acquired from OGRE demo).

Machine Learning Approach

Following the analysis of the work of others, we selected *Particle Swarm Optimization* (PSO) as the machine learning method to be used in our application. PSO was employed successfully in the work described by Wang et al. [17] PSO uses a population of *particles* to collaboratively search for a solution. Each particle has a position and velocity in the feature space. In addition, each particle remembers the highest fitness it has encountered, and where it occurred.

At the start of a run, each particle is initialized to a random guess—in this case, a 17-dimensional vector representing a particular combination of slider settings—and evaluated with the fitness function. At each iteration, the particles communicate their fitness and determine the global best—that is, what is the highest fitness any of the particles have encountered up to that point. Using this, combined with the particle's local best—the highest fitness yet encountered by that individual particle—each particle adjusts its velocity towards the global best and its local best, with the rate of this acceleration determined by the

distance, relative fitness, and a predetermined 'inertial' parameter. The basic formulae are described by

$$v_{new} = C_0 v_{old} + C_1 r_1 \left(x_{globalbest} - x_{old}\right) + C_2 r_2 \left(x_{localbest} - x_{old}\right)$$

$$x_{new} = x_{old} + v_{old}$$

where C_0, C_1, and C_2 are constant parameters, and r_1 and r_2 are random vectors. The fitness score is based on how closely the PSO-generated face matches the user-created face, produced through the OGRE demo tool.

Description of Method Employed

Next we describe how the PSO in the OGRE demo was implemented. This was completed in two phases. The first phase was designed to verify that the PSO approach can work on a simplified system. Phase 2 is the implementation of the PSO using the pixel fitness.

Before phase 1 could begin, some manipulations to the OGRE demo were needed to get the rendered face into a suitable form. This was done by interfacing several of the functions provided in the demo. OGRE has the ability to generate different light sources that affect the 3D scene. This light source was moved from its initial spot to where the "camera" was to give even lighting to the entire face. The camera position was also changed to focus more on the face, allowing better detail to be made out.

Once this was done, a method to obtain the slider values from the demo was developed. A function was built into OGRE that provided each of the slider values (between 0 and 1) that were then recorded in a file. These slider values were the parameters that the PSO manipulates in later phases to generate facial expressions corresponding to the emotions manually created through the sliders.

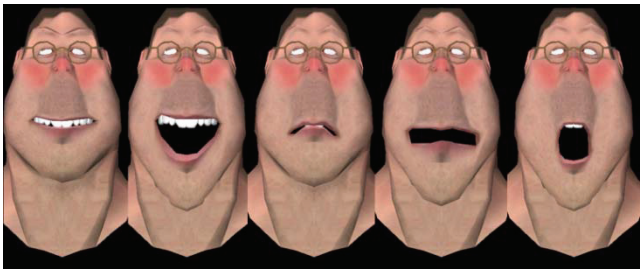

Figure 2. Faces created for experiment to represent individualized expressions. Anger, happiness, sadness, fear and surprise (left to right).

Now that values for specific faces could be recorded, the expressive faces could be created. Given that OGRE is a cartoon-like avatar, our objective of using a photograph of an actual human was not relevant in this testbed. Instead, we adjusted the sliders manually to create faces through OGRE showing five different basic emotions - anger, happiness, fear, sadness and surprise. To make it as realistic as possible, multiple slider settings were used for each emotional face. The emotions created are shown below in Figure 2. This was done to make the created individualized emotional faces unique and to mimic the individually expressed emotions of a person. The sliders were manually adjusted into place, the values were recorded, and a bitmap image was taken of this final face. At the end of this process, the five "individual" emotional faces of OGRE were created as shown in Figure 2.

The final step before running phase 1 was to insert the PSO machine-learning algorithm into the OGRE demo. It was placed in the class where the face was generated. This was done because that section of the demo is where the slider values that change the face were stored.

Phase 1: Testing PSO

Phase 1, PSO was used to recreate the slider values that were used for each of the five emotional expressions. In effect, the machine learning algorithm seeks to learn the slider settings for each of these five faces, using the actual slider settings as the standard of fitness. While seemingly trivial, this experiment provided us with a sanity check on the usefulness of PSO for this application. A population of 50 individuals running for 100 iterations was used for this test. The 17 sliders for each particle were randomly generated. The fitness function used calculates the Manhattan distance and then sums these errors for each slider for every particle in the population. This error is the difference from the desired slider value to the generated slider values. After several tests, equations for the velocity and position generation were found. Below is pseudo code to illustrate the PSO.

```
Population is 50
While (iterations less than 100)
{  For loop (through population)
      {  Check fitness (17 sliders)
      {  For loop (through population)
      {  Check Local Best
         Save 17 Slider Values
         Check Global Best
         Save 17 Slider Values
      }
   For loop (through population)
      {  Update Velocity
         Update Position
      }
}
```

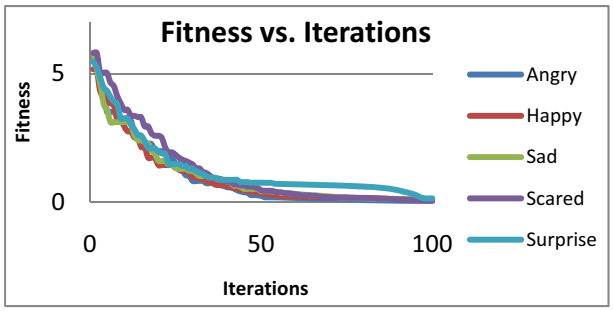

Figure 3 - Test Fitness vs. Iterations for phase 1. Error converges towards zero

At these values the PSO consistently performed well. It was able to recreate all five faces with minimal fitness error, due to the allowed decimal accuracy. With the final values chosen for the PSO, the faces were consistently able to be regenerated. After the 100 iterations there were only minimal errors of the slider values, due to the precision of the decimal. Figure 3 shows that the test fitness almost reached zero error for each face.

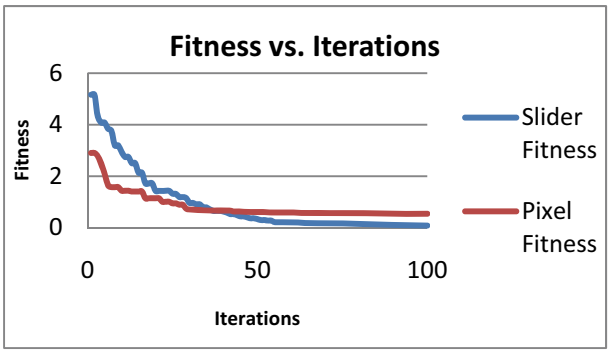

Figure 4 - As the test fitness decreases to zero, so does the pixel fitness. This shows that the photograph fitness can be used in Phase 2.

Phase 2: Implementing Pixel-by-Pixel Fitness

The results of Phase 1 established the PSO as a feasible candidate for the project. Phase 2 introduces the use of the "photograph" of the OGRE demo as the standard of fitness. Unlike the fitness function of Phase 1, this new fitness function compares (pixel-by-pixel) the original face, created by manual manipulation of the sliders, to the PSO generated faces. We refer to this as the *pixel-by-pixel fitness*. It then sums this difference between the two faces and outputs a number from 0 to 100. The OGRE demo was modified once more so that every face in the population is generated before Phase 2 can be tested. Figure 4 shows how both the slider and pixel-by-pixel fitness converge to zero error, showing that the pixel-by-pixel fitness can be used for phase 2. Previously, the face was not regenerated each time the slider values were updated. This was done so that the program could run faster. However, for Phase 2 this became necessary because unlike the test fitness of Phase 1, the pixel-by-pixel fitness is based on the comparison of a generated face and the original face being reproduced. This allows the program to take in each face and compare them using the pixel-by-pixel fitness via the FACS values computed from the images. For this updated program the only adjustment was to increase the iterations to 250. Because in this phase each face in the population is being rendered on the screen, the run time increases from several minutes in Phase 1, to 2 hours for a population of 50 in Phase 2. The resulting PSO-generated generated faces can be seen in Figure 5 below.

An obvious problem can be seen with the angry face, where its teeth are not shown. Despite less than a 1% error, the angry face did not converge to the proper features. Through visual inspection and with an understanding of the underlying avatar system, it can be seen that some of these faces are easier to generate than others. Because the population is initialized with random values, there are combinations of sliders that are difficult to reproduce through random generation. Because of this, there are many other combinations of sliders that produce an open mouth with exposed teeth. If just one member of the population can create a face with an open mouth, there is a very good chance it will converge to the right answer. Because the chance of exposing teeth is much less, the PSO may converge to another region that is not the answer, and thereby become stuck because of a lack of momentum. One remedy is to increase the population size. This will increase the chance of a face that will lead to the correct answer. This method was used to solve the angry face problem and the population was increased from 50 to 100. This allowed for a greater variety in the initially generated faces and was able to converge. See Figure 6 below. It can easily be seen by inspection that it compares much better with the left-most face in Figure 2. The only drawback was that the runtime had doubled to 4 hours.

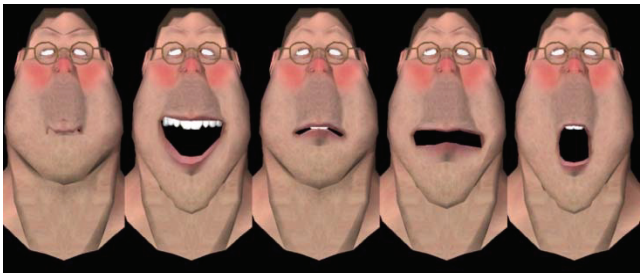

Figure 5 - Emotional faces generated by PSO using photograph fitness after 250 iterations for anger, happiness, sadness, fear and surprise (from left to right).

Figure 6 - Angry face after running the PSO using a population size of 100.

Figure 7 shows the fitness of each run, including the angry face with a population of 50 and 100

Summary and Conclusions

Based on the literature review and experiments conducted it seems that the PSO-based machine learning approach to creating avatars is possible and may serve to save time in generation of expressions in a human-like avatar. Future work includes creating an avatar based on FACS Action

Units. This would allow more freedom in the avatars face and produce more accurate expressions. If that is achieved, further research can be performed by changing the "photograph" in this project to an actual photograph of a person. A suitable avatar engine that represents humans and not cartoon characters would be used instead of OGRE. The main modification to the current program would be the fitness function. Assumptions on how the fitness function will need to be edited can be made. Instead of a pixel-by-pixel comparison, more complex methods would be implemented. The use of biometrics or methods of identifying individuals could be one of such complex methods. Furthermore, assigning weights to important regions of the face including the eyes and mouth may increase the accuracy of the fitness function. Lastly, using video clips instead of photographs would allow the avatar to learn the transition between expressions, rather than just the final expression. The research completed so far has helped opened the door to this mostly unexplored territory.

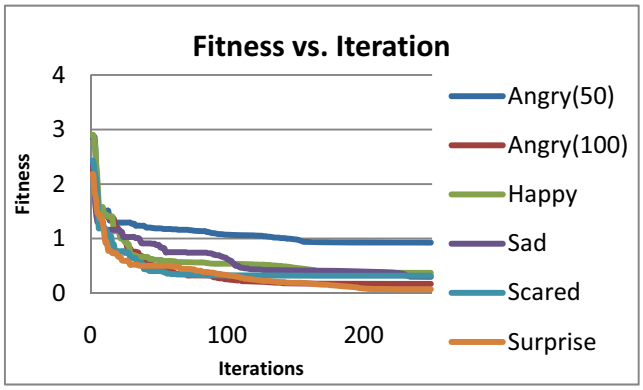

Figure 7 - Pixel-by-pixel fitness for each face with error converging to zero.

References

[1] Bartlett, M. S., Littlewort, G., Lainscsek, C., Fasel, I. and Movellan, J., "Machine learning methods for fully automatic recognition of facial expressions and facial actions", IEEE International Conference on Systems, Man and Cybernetics, October 2004, vol.1, pp. 592 597.

[2] Blender Foundation, http://www.blender.org

[3] DeMara, R. F., Gonzalez, A. J., Leigh, J., Jones, S., Johnson, A., Hung, V., Leon Barth, C., Dookhoo, R. A., Renambot, L., Lee, S. and Carlson, G., "Towards Interactive Training with an Avatar based Human Computer Interface", Proceedings of the Interservice/Industry Training Systems and Education Conference, Dec. 1 4, 2008, Orlando, FL.

[4] Division of Information Coding, Linköping University, http://ww.icg.isy.liu.se/en

[5] Ekman, P. and Friesen, W., "Facial Action Coding System: A Technique for the Measurement of Facial Movement", Palo Alto, CA, Consulting Psychologists Press, 1978.

[6] Gallagher, A.C., Das, M. and Loui, A.C., "User Assisted People Search in Consumer Image Collections", IEEE International Conference on Multimedia and Expo, 2 5 July 2007, pp. 1754 1757.

[7] Kasiran, Z. and Yahya, S., "Facial Expression as an Implicit Customers' Feedback and the Challenges", Computer Graphics, Imaging, and Visualization, 14 17 August 2007, pp. 377 381.

[8] Khashman, A., "A Modified Backpropagation Learning Algorithm with Added Emotional Coefficients", IEEE Transactions on Neural Networks, Volume 19, Issue 11, November 2008, pp. 1896 1909.

[9] Kim, D.J. and Bien, Z. "Design of 'Personalized' Classifier Using Soft Computing Techniques for 'Personalized' Facial Expression Recognition", IEEE Transactions on Fuzzy Systems, Volume 16, Issue 4, August 2008, pp. 874 885.

[10] Liu, J., Chen Y. and Gao W., **"**Mapping learning in eigenspace for harmonious caricature generation", Proceedings of the 14th annual ACM International Conference on Multimedia, 2006, pp. 683 686.

[11] Mori, M., "The Uncanny Valley", trans. K.F. MacDorman, T. Minato, Energy, vol. 7, no. 4, 1970, pp. 33 35.

[12] Mpiperis, I., Malassiotis, S., Petridis, V. and Strintzis, M.G., "3D facial expression recognition using swarm intelligence**",** IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, *ICASSP 2008,* March 31 April 4 2008, pp. 2133 2136.

[13] Mpiperis, I., Malassiotis, S. and Strintzis, M. G., "Bilinear Models for 3 D Face and Facial Expression Recognition", IEEE Transactions on Information Forensics and Security, Volume 12, Issue 3, September 2008, pp. 498 511.

[14] OGRE Open Source 3D Graphics Engine, http://www.ogre3d.org

[15] Smith Micro Software, http://www.smithmicro.com

[16] Teller, A. and Veloso, M., "Algorithm evolution for face recognition: what makes a picture difficult" IEEE International Conference on Evolutionary Computation, Volume 2, 29 November 1 December 1995, pp. 608 613.

[17] Wang, Y., Liu, X. and Suo, P., "Face detection based on pso and neural network"**,** 9th International Conference on Signal Processing, ICSP 2008, 26 29 October 2008, pp.1520 1523Engelmore, R., and Morgan, A. eds. 1986. *Blackboard Systems.* Reading, Mass.: Addison Wesley.

[18] Oztop, E., Chaminade, T., Cheng, G., and Kawato, M., "Imitation bootstrapping: experiments on a robotic hand," Humanoid Robots, 2005 5th IEEE RAS International Conference on, 5 Dec. 2005, pp.189 195.

[19] Wood, M.A. and Bryson, J.J., "Skill Acquisition Through Program Level Imitation in a Real Time Domain," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol.37, no.2, April 2007, pp.272 285.

[20] Ariki, Y., Morimoto, J., and Hyon, S.H., "Behavior recognition with ground reaction force estimation and its application to imitation learning," Intelligent Robots and Systems, IEEE/RSJ International Conference on, 22 26 Sept. 2008, pp.2029 2034.