Facial Expressions and Speeches Obtained Through Rear Projection using the Socibot Desktop Robot

Gheorghe Gîlcă, Nicu George Bîzdoacă

Abstract—This article aims to present a few practical applications using the Socibot Desktop social robot. To begin, we need to connect the Socibot Robot to the Ethernet or WiFi network. Then, in the IDE (Integrated Development Environment) browser we can control the robot directly, write programs it can execute, load a guise for it, or get access to the robot sensors. We mean to realize two applications: making a new guise to be loaded into the robot's memory in order to be projected onto its face and creating a program in the Virtual Robot browser interface. The first application aims to change the robot's appearance with the guise we created. The guise was created in Adobe Photoshop and then loaded into the robot's memory. The second application is more complex, the completed program has a length of 57 seconds, contains 21 head movements, 4 speeches in French and 2 guises of the robot Socibot.

Index Terms— browser interface, guise, moods, sequence, social robot

I. INTRODUCTION

Socibot Desktop is a humanoid robot built by the Engineered Arts Company in the United Kingdom in 2014. It is presented in Fig. 1, and its physical structure is composed of the following elements [1]:

- Integrated high-definition webcam;
- High-fidelity incorporated loudspeaker;
- 3 electrically-regulated axes at the neck level;
- PC Incorporated;
- Integrated Infra-red depth sensor for gesture detection.
- Digital projector.

The Socibot robot can follow simultaneously up to 12 persons using the depth sensor. The Socibot robot runs based on rear projection, using a digital projector. Expressions can be projected on the plastic cast face of the robot, each expression being created directly through the browser interface [1].

Figure 1. Structure of the Socibot Desktop [1]

Pre-installed facial expressions are in concordance with FACS (Facial Action Coding System) in order to establish differences between facial expressions.

FACS is a system used to classify the movements of the human face according to how they look on the face, based on the system developed by Paul Ekman and Wallace Friesen [2]. Paul Ekman, Friesen and Joseph C. Hager published an important update for FACS in 2002 [3].

We can connect an android tablet or any phone using this system through the browser interface in order to realize the tele-presence (remote-presence) operation, that is, to command the robot from a distance.

Engineered Arts offers a complete content management browser-based interface and a programming interface of the Socibot. To access this interface we have to key in the url field of the browser //sb-yyyy/, where yyyy stands for the robot serial number. If this does not work, we could try inputting the robot's IP directly in the url field.

The Socibot robot and the calculator connect to the router using an ethernet network cable. The router is connected to the Internet and also ensures access to the robot through the browser interface, as explained above.

The general connection structure of the computer in order to allow the Sociobot robot to realize applications is:
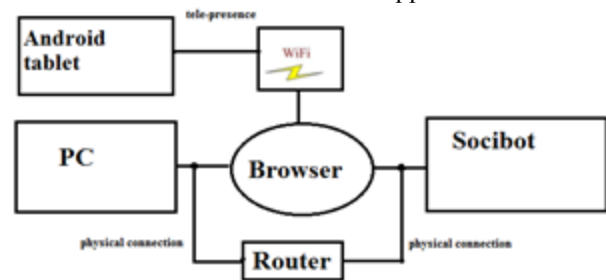

Figure 2.   Proposed Block Diagram

The Socibot robot also integrates the facial motion capture technology through markers, using the Visage SDK marker and aiming to record in real-time the 3D position of the head, line of sight, and facial features coordinates, including mouth, chin, eyebrows contours and the closing of the eyes [1].

II. APPLICATIONS USING THE SOCIBOT DESKTOP ROBOT

A. Creating a guise for the Socibot Robot

In Ya Face is a graphic 3D application, as shown in Fig. 3, that uses a series of facial models in order to generate images for rear-projection onto a surface of similar form to the human face. If we choose to select a new expression we click on the menu list (marked as 1), and the values of the facial units actions, AUs, change according to the facial expressions

(marked as 2), and in the desktop area (marked as 3) the guise that will be projected on the robot's face is shown. We can realize the following in the In Ya Face graphic application:

- Real-time animated facial expressions;
- Apply textures (bitmap images of the face) to a neutral facial model;
- Add expression modifiers to change the form of the basic model;
- Add mesh modifiers for the shape of the mouth in order to create mouth movements according to the speech;
- Add texture to the eyeball ;
- Controls the line of sight and the dilatation of the pupil;
- Maps the connation of the presented animated face model to a fixed-form network that fits the physical structure of the robot;
- Uses a geometric network of the actual face of the robot to correct the distortion;

- Provides controls for the rear-field projection and position in order to correct the distortion;
- Accepts JSON codified commands to control the animation;
- Corrects the color difference for the projector lenses.

Editing the expressions is realized as the notations in Fig. 4 show: to create and store a new expression, we click on the „new expression" button (marked as 1) and we input a name for the expression in the text bar (marked as 2). We can adjust the facial action units using the cursors (marked as 4). The symmetry is activated implicitly for all slides – to disconnect the action units on the left side from those on the right side, we click the yellow tab placed on the right of the cursor. Once we are satisfied with the expression, we can save or delete it (marked as 3).
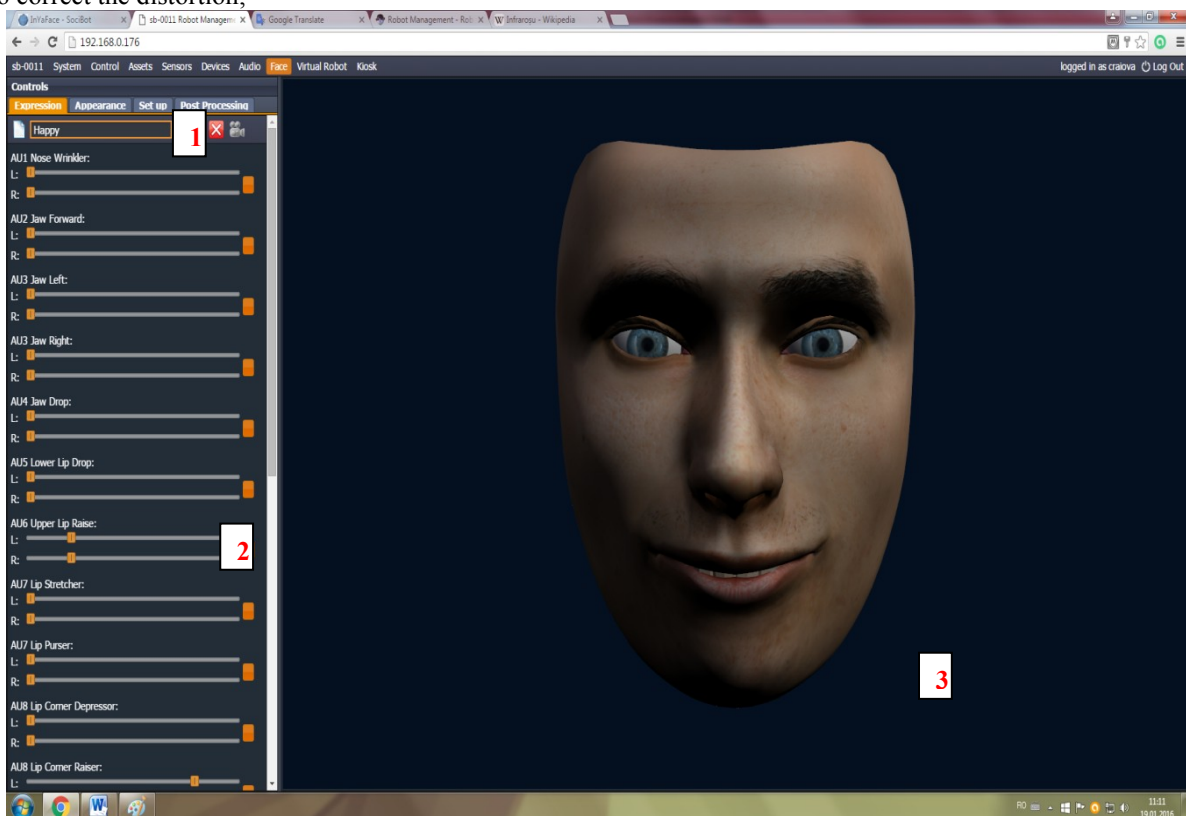

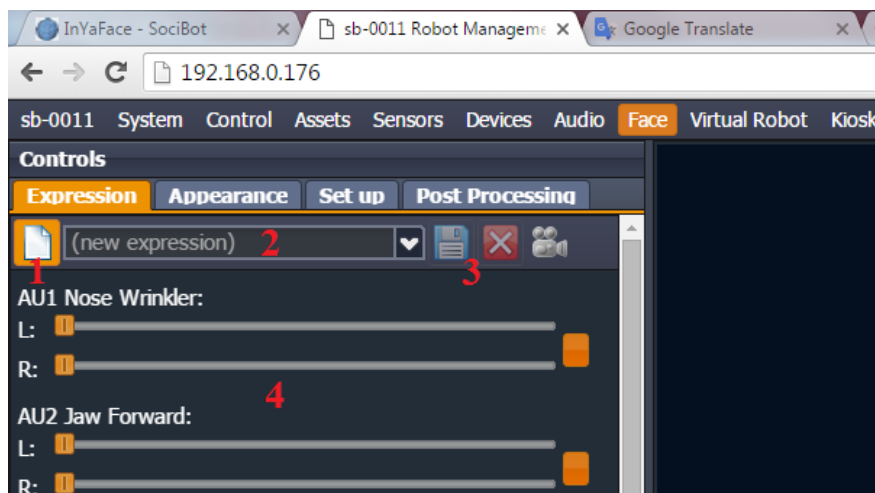**Figure 3. The InYaFace interface and the configuration instrument**

Figure 4. Editing expressions in the InYaFace interface

The guise realized in Adobe Photoshop is shown in Fig. 5. This is constructed based on the model presented on the manufacturer's website [4]. The model contains the UV positional map of the face in the model, a representation of the teeth and tongue (marked as 1), a basic layer for the face (marked as 2) and the face (marked as 3), positioned on top of all the layers. The created image has a square form and we need to save in the png format, to maintain the transparency of the positional map of the face.

If we want to project one of our guises, we can load it through pushing the green, „+" form button (marked as 3 in Fig. 7) and then a window will open, as shown in Fig. 6, where

we can write the name and path of the image so that it is loaded into the robot's memory.

To modify the robot's guise we click the „Appeareance" button, as show in Fig. 7, and we select a guise for it (marked as 1) from the toggle list. We can also change the color of the eyes and their size (marked as 2). In the Fig. 8 is shown the created guise on the Socibot robot.
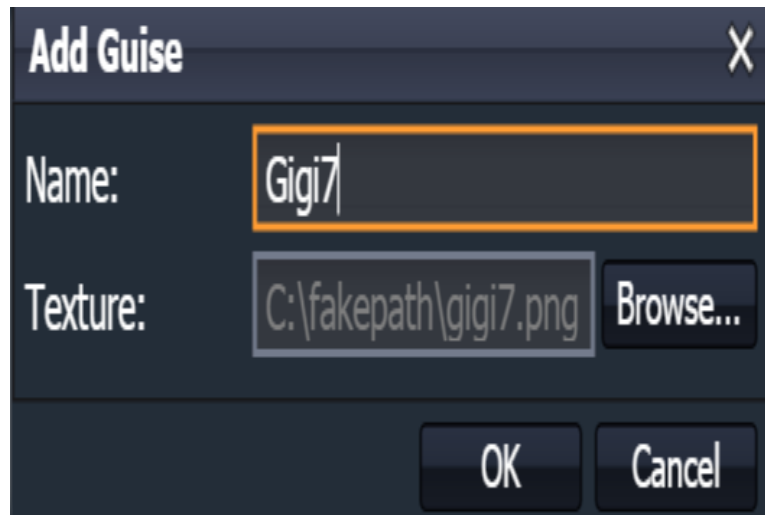

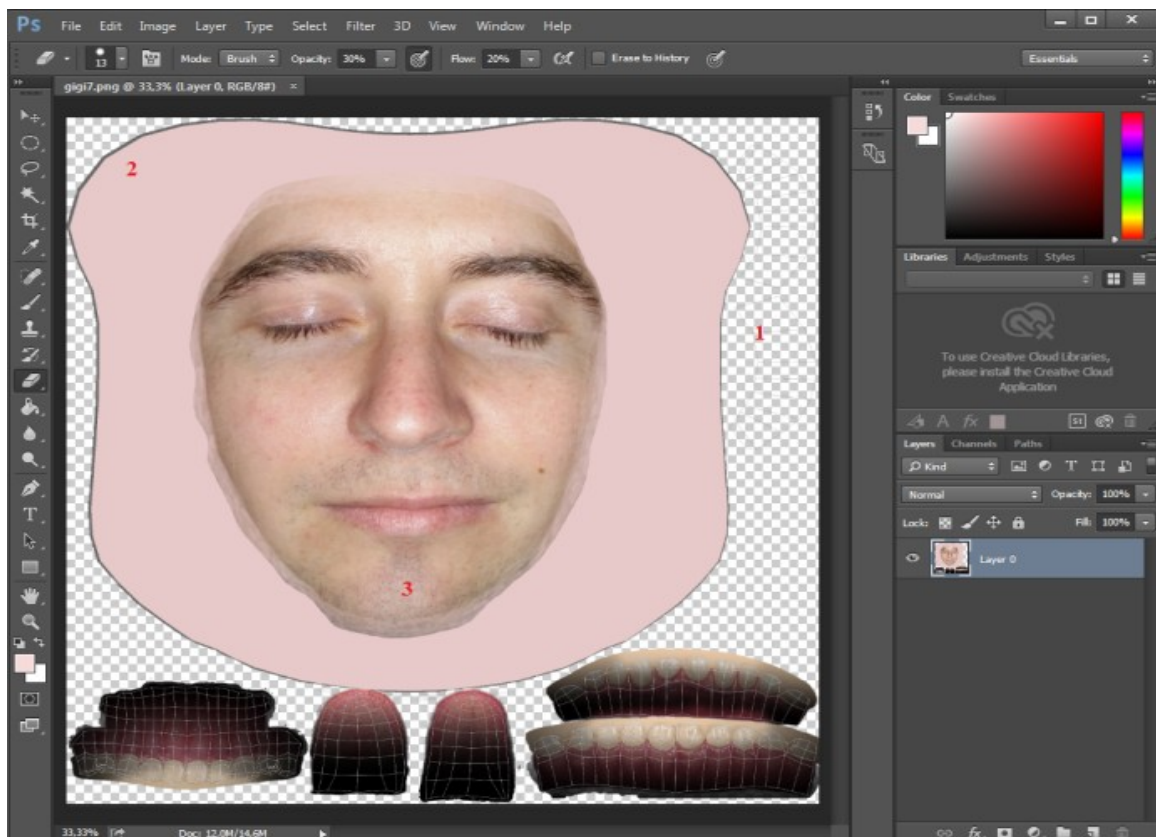**Figure 6. Loading a guise on the robot memory**

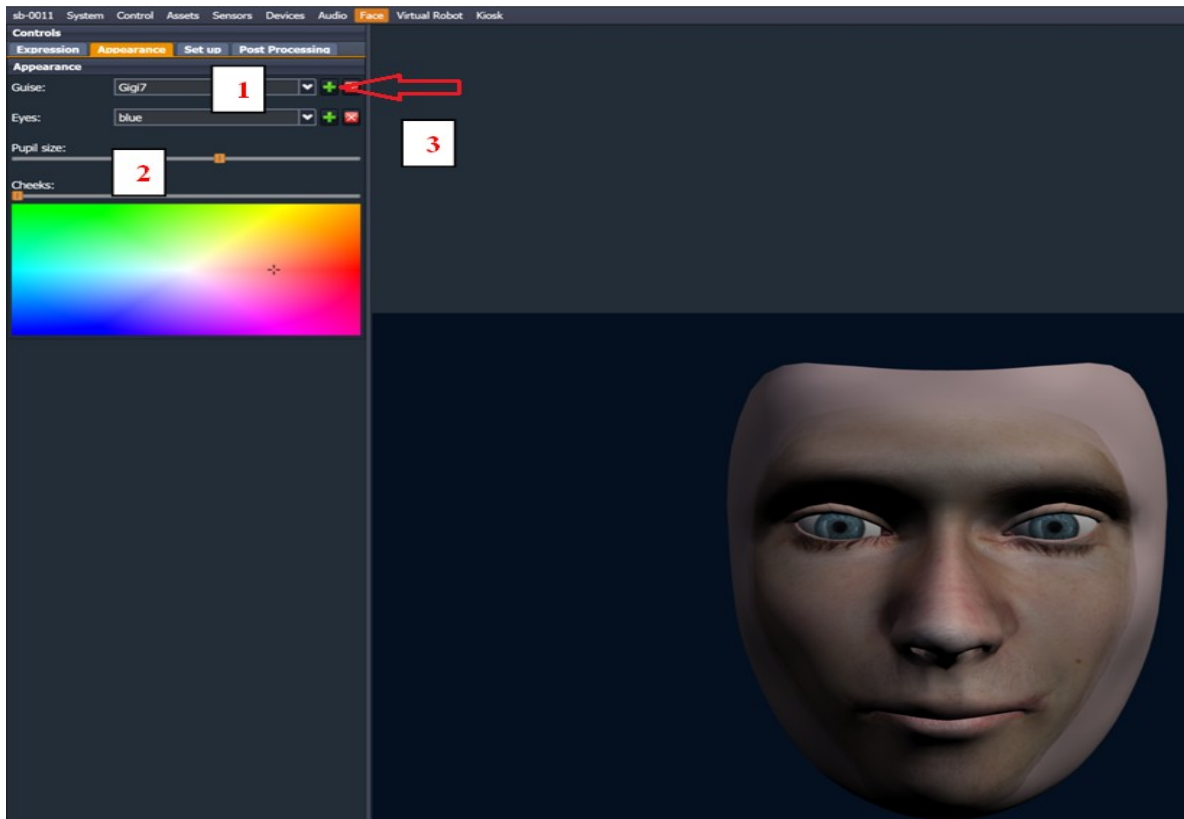**Figure 5. Creating the guise in the Adobe Photoshop for the Socibot Robot**

Figure 7. Showing the guise loaded in the InYaFace application

Figure 8. The created guise displayed on the Socibot robot

B. Creating a program in the virtual robot menu and transferring it to the Socibot robot

Virtual Robot offers an integrated menu in which we can create programs to transfer on the actual robot. We can select files already installed or we could build our own program using this dynamic integrated menu for our type of robot. Fig. 9 presents Virtual Robot programming interface. This consists of the following five important windows:

1 - Menu Bar – here we can realize a few file controls and it displays the following: the name of the created file, authentication details and the IP of the controlled robot;

2 - Staging area – this is the desktop. We can control the robot in this are using only the mouse.

3 - Library – The library of the interface comes with preinstalled content, this being available to copy through drag and drop in the window 4 timeline;

4 - Timeline – here we arrange the aspects in the desired order. We can insert the audio components, speeches, robot head movements;

5 - Inspector – here we can modify robot facial parameters, entries and the moves that we want it to make, these being displayed in real time on the desktop.

In the libraries window we can access the stored data in order

to use them in the timeline sequences. When copied in the timeline, the length of the audio file or movement file will be reflected in the duration of each component as displayed in the timeline.

Figure 9. Creating a program in the Virtual Robot menu

For the speech subclass in the libraries window we use the buttons in Fig. 10 to create a speech, as follows:

1. Language –we select the language;
2. Voice selection – a number of voice samples are available;
3. Voice style – some voices (especially in the US English) have optional speech modulations, for example: mood, age, character;
4. Text input – we input the speech we want in here and press "add Speech" (low-right on the desktop) to insert it in the timeline;
5. Volume – it controls the text-to-speech (TTS) component volume;
6. Speed – it controls the speed;
7. Shaping – controls the inflexion of the text-to-speech component;
8. Lipsync Gain – it controls the range of motion for the maxillary/jaw;
9. Append & Add at Marker – adds the TTS to the voice track;
10. Expand & Contract Library Area – enlarges and shrinks the library zone.

In the Timeline window we can place all the files we want, and then re-arrange them as needed in order to obtain the result of a good sequence for our robot.

In the Inspector window we can modify: the facial parameters of the robot, its expressions, the line of sight, the head rotation etc. In Fig. 11 we present an example of a modification we made to the Socibot's "Happy" expression that resulted in a change of the facial expression of the robot also in the desktop area.
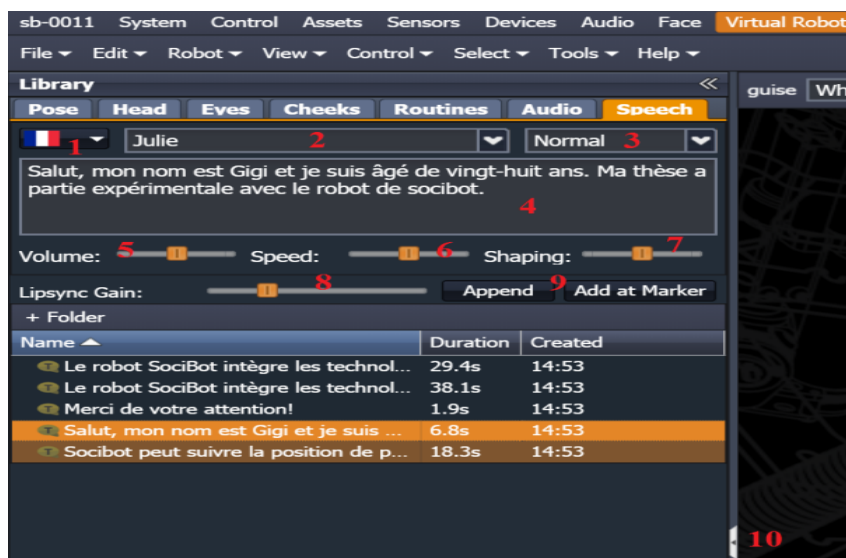

Figure 10. The speech subclass and its functions for editing a speech

Figure 11. Correlation between the changes in the Inspector window and the desktop

The program created (shown in Fig 9) in the virtual robot meniu has a length of 57 seconds, contains 21 head movements, 4 speeches in French and 2 guises of the robot (shown in Fig. 12). Speeches are created as shown in the details of Fig. 10 and then inserted in the timeline. The robot head movements are inserted where we want them in the timeline using the libraries window, and the robot guises are inserted in the timeline at the time mark we need them at. We've transferred it to the real robot and we've captured 2 images: the first at second 22 and the second one at second 49 of the created sequence.
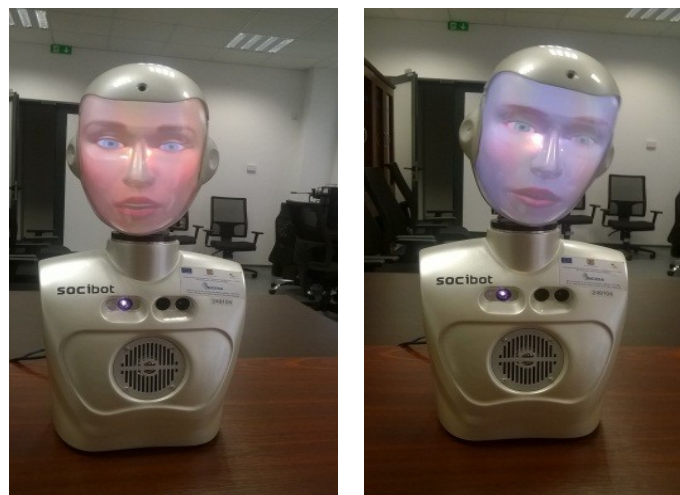

Figura 12. Running the program from the virtual robot menu of the Socibot robot

III. CONCLUSIONS AND FUTURE WORKS

For the first application can be said to be easy to use for loading an appearance, as if inserting a file on a computer, and the design of its guises is based on a model available on the manufacturer's website. The result of application demonstrates that Sociobot robot can project any appearance desired by the user.

For the second application, which is more complex, we can conclude that in the virtual robot programing environment we can do the following: we can change spoken language; we can change the facial action units in order to obtain new expressions created by us, we can change the robot's guises and we can create speech sequences.

In the future we want the Socibot Desktop robot to recognize our facial expressions in real time and to talk with us when it detects our presence using its camera.

REFERENCES

1. https://www.engineeredarts.co.uk/socibot/interface/
2. P. Ekman and W. V. Friesen, Facial Action Coding System: Investigator's Guide, Palo Alto, CA: Consulting Psychologists Press, 1978(a).
3. P. Ekman, W. V. Friesen and J. C. Hager, Facial Action Coding System: The Manual on CD ROM. A Human Face, Salt Lake City, 2002.
4. http://robo-thespian.com/socibot/wiki/index.php/Guises