A real time OSC controlled agent for human machine interactions

Hüseyin Çakmak and Kevin El Haddad and Roberto Pulisci

Abstract

In this paper we talk about our ongoing work on the creation of a tool for Wizard of Oz experiments. These experiments' goal will be to study methods for non-verbal emotional expressive behaviors in human-computer interactions. We therefore present our real-time reactive social agent controllable via an Android application interface.

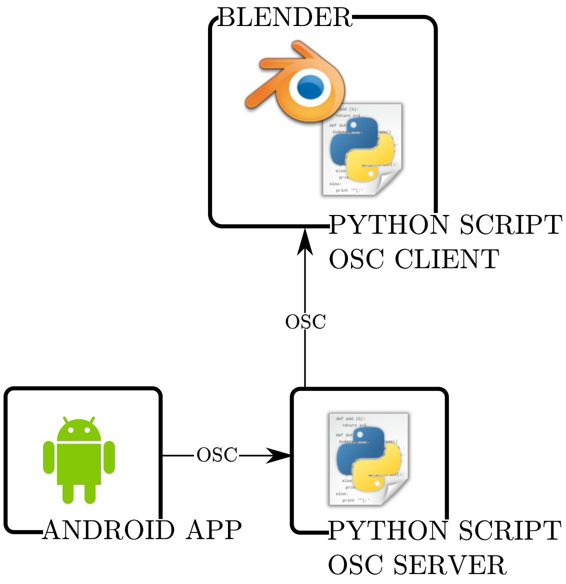

Figure 1: General overview of the system

1. INTRODUCTION

There is a growing interest in the development of more intuitive interfaces to interact with the machines in our everyday tasks. Indeed, interacting with machines have become significantly more common in our daily lives in the past two decades. So, having a more intuitive way of interacting with our daily computers (laptops, mobiles phones, etc) than the usual mouse and keyboards would enhance our experience with the machine and make the interaction more comfortable. An example of such a natural Human-Computer Interaction (HCI) would be controlling the machine's tasks using natural conversational speech or gestures; i.e. being able to talk to the machine just like to a human being. In order to do so, the machine should incorporate human-like features, such as emotional expressions or even having human-like features, such as facial traits.

We present a real-time reactive agent possessing human facial features along with audiovisual (AV) expressions. It can serve in Wizard of Oz (WoZ) [4] experiments and, in the longer term, as an autonomous virtual agent.

Indeed this systems allows to control the agent's facial expressions and head movements (rotation and translation) and also to play the corresponding audio cues synchronously. The agent could thus produce speech as well as non-verbal expressions.

2. IMPLEMENTATION

The reactive agent has been implemented using Python scripting. Part of the implementation is done in Blender which is used as rendering engine and part of it was done outside Blender. The general overview of the system is given in Figure 1.

The Python script in Blender consists of an Open Sound Control (OSC) client that waits for data from the server running as a standalone application. The data consists of a vector of 105 values corresponding to 3 head translations, 3 head rotations and 99 facials landmarks translations (33 landmarks times 3 translation values). When a data frame is received, the

Blender Python scripts applies the deformations to the 3D model in real time. Therefore, currently the data needed to drive the avatar's behaviors should be the 3D coordinates of the previously mentioned landmarks (in our case these recordings were made using the Optitrack system).

Additionally, an Android app is used as interface to use the agent as a WoZ system; predefined animation sequences (sequences of data frames corresponding to a specific animation) can be triggered from the Android app at particular moments chosen by a human operator. When triggered, the animation sequence which is stored in a text file is sent frame by frame by the OSC Server to the OSC Client inside Blender and the animation is played live.

3. IMPROVING WOZ APPLICATIONS

As previously mentioned, one of the use of this system will be in WoZ experiments. The use of an android application to remotely control the virtual agent would facilitate the setup of WoZ experiments and would help us improve previous experiments and recording sessions. In a WoZ experiment, the subject interaction with the system should be unaware that the system is being controlled by a human. Indeed, remotely controlling the agent's reactions via OSC (see Fig. 2 and Fig. 3) would help with the constraint of the user's position when controlling the agent. Also, our system can react quasi instantaneously with predefined animation sequences. To sum up, our system first gives us flexibility in the design of the experiment's setup, and is real-time efficient. Qualities which are important and convenient for a WoZ experiment.

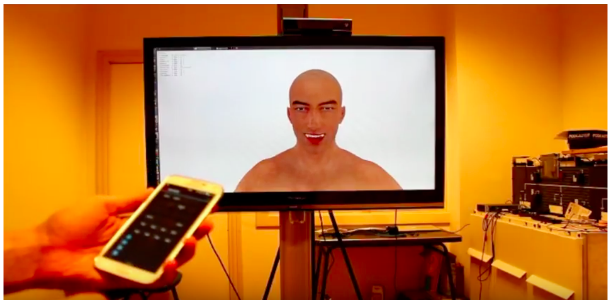

Figure 2: Control of the agent's reactions via and android application and OSC.

4. HCI APPLICATIONS

This system will be part of a larger framework of creating an emotionally expressive agent. Indeed, an AV laughter synthesis system has already been developed [1] and a multilevel AV affect burst [5] database collected [3]. So, ongoing work are made to be able to synthesize in real-time different emotional expressions and at different intensity levels. With this, the real-time reactive agent will contribute to study and implement methods correctly incorporating emotional expressions into HCI applications.

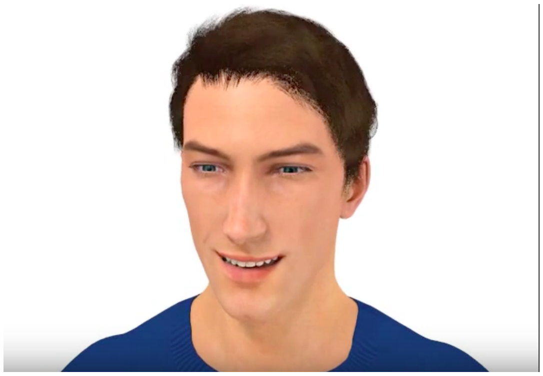

Figure 3: Virtual agent with hair and clothes.

Also, one of the future goals, would be to study whether an environment-aware social agent [2] might create a better HCI. Indeed we are interested to see if taking into account the environment during an interaction and therefore the context would contribute to have a more human-like and therefore a more natural interaction. For example, the change of direction of the agent's head and therefore, of his visual focus, could create the impression of an interaction not only between the user and the virtual agent, but also between the virtual agent and the environment. Indeed, an agent looking away during an interaction with a user while also expressing an emotion such as amazement might enhance the user's experience by making the interaction more natural. Such a scenario could be used, for instance, to point out to a user an event or object outside his visual field and which the virtual agent "knows" is interesting for this user. The companionship of the agent might thus be enhanced with such behavior.

5. CONCLUSION

In this paper, we presented our ongoing work of creating an emotionally, real-time reactive virtual agent and studying the use of an environment-aware social agent. This system is currently not linked to our synthesis systems previously cited. In future works, we intend to build a system in which our synthesis systems' output would be the input of the reactive agent. Improvements of our system will also be attended by, for example, being able to drive the avatar's facial expressions using different types of motion capture data such as Kinect data. Also an interface will be created to more conveniently control the different aspects of the agent's behaviors; i.e. emotion expression, intensity of the expressed emotion and avatar's gaze orientation. Several WoZ experiments will also be designed to study methods of emotional and non-verbal expressive behavior.

The successful methods could finally be used to build an autonomous virtual agent.

A demonstration video of the reactive agent can be found at `http://tcts.fpms.ac.be/~cakmak/personal/?p=640`.

6. REFERENCES

[1] Hüseyin Çakmak. *Audiovisual Laughter Synthesis - A statistical Parametric Approach.* PhD thesis, Polytechnic Faculty of Mons, 2016.

[2] Hüseyin Çakmak, Kevin El Haddad, Nicolas Riche, Julien Leroy, Pierre Marighetto, Bekir Berker Türker, Hossein Khaki, Roberto Pulisci, Emer Gilmartin, Fasih Haider, Kübra Cengiz, Martin Sulir, Ilaria Torre, Shabbir Marzban, Ramazan Yazıcı, Furkan Burak Baĝcı, Vedat Gazi Kılıç, Hilal Sezer, and Sena Büşra Yengeç. Easa : Envireonment aware social agent. Technical report, Polytechnic Faculty of Mons, 2015.

[3] Kevin El Haddad, Hüseyin Cakmak, Stéphane Dupont, and Thierry Dutoit. AVAB-DBS: Audio-Visual Affect Bursts Database for Synthesis. in press.

[4] J. F. Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Trans. Inf. Syst.*, 2(1):26–41, January 1984.

[5] K.R. Scherer. *Emotions: Essays on emotion theory.* Hillsdale, USA, 1994.