A facial imitation framework for the simultaneous face control of a virtual avatar and a humanoid robot

Mattia Bruscia[1], Graziano A. Manduzio[1], Lorenzo Cominelli[1] and Enzo Pasquale Scilingo[1]

[1]University of Pisa, Pisa, Italy

Abstract

Facial expression imitation (FEI) for humanoid robots is an active research field in the context of human robot interaction (HRI). Virtual avatars can enhance and simplify the experimental HRI setup in terms of cost and performance, avoiding possible long-term mechanical degradation of the physical robot in use. Moreover, the presented framework allows to conduct comparison studies aimed at investigating the role of embodiment in the interaction with a robot versus its digital twin, which is a critical factor to establish a successful social bond with the robot, as in the case of numerous clinical applications.

Keywords

Human-robot interaction, facial expression imitation, virtual avatar, Facial Action Coding System (FACS)

1. Introduction

In recent years, the advent of anthropomorphic social robots, increasingly similar in physical features to human beings, has lead researchers and engineers to endow these robots with even more advanced human-like abilities. Among these, we can easily consider the real-time ability to recognize and mimic the facial expressions of another human being. Methods for automated facial expression recognition (FER) have been a research field in human-robot interaction for several years (see Li and Deng (2020) [1]; Canedo and Neves (2019) [2], for a survey). Research activity related to facial expressions, refers to the studies of Paul Ekman about the action units (AUs), a set of anatomical basis facial movements to compose all the others, described in the FACS [3]. However, facial expression imitation (FEI) for humanoid robot is a younger field of research. For example, Breazeal et al., built a robot capable of learning how to imitate facial expressions from simple imitative games played with a human [4]. Wu et al., developed a system to make the robot face "Einstein" able to learn expression facial patterns coding a map from detected action units (AUs) to servos, solving an inverse kinematic problem [5]. Boucenna et al., developed a neural network model able to control a robot head and learn online to recognize the facial expressions of the human partner [6]. A similar approach was used by Meghdari et al. [7], and Kobayashi and Hara [8]. Teaching a robot these skills is challenging. Frequently, the learning methodologies employed may necessitate substantial exertion from the robot's joints, potentially leading to a swift degradation in performance or even the fracturing of servos. In this context, the use of virtual avatars can speed up the task learning process, without imposing excessive mechanical strain on robot's mechanisms. Furthermore, a virtual avatar is easier and cheaper to use than a physical robotic counterpart, opening up a range of possible application scenarios, not only in the development context but also in the clinical one. Dongxiao et al., moved a virtual face, using a 3D facial video of the user captured with Kinect [9]. Rawal et al., introduced ExGenNet, a novel deep generative approach for facial expressions on humanoid robots. They trained the system using a robot Alfie's simulator [10]. In this manuscript, we introduce an innovative methodology for concurrently manipulating the facial expressions of a virtual avatar [11, 12] and a sophisticated expressive robot [13], utilizing a cutting-edge Action Units (AUs) detector with an high detection accuracy [14]. The simultaneous control of a digital avatar and a highly expressive humanoid robot is a fundamental aspect of the study described, as it

10th Italian Workshop on Artificial Intelligence and Robotics (AIRO 2023)

m.bruscia@studenti.unipi.it (M. Bruscia); grazianoalfredo.manduzio@phd.unipi.it (G. A. Manduzio); lorenzo.cominelli@unipi.it (L. Cominelli); enzo.scilingo@unipi.it (E. P. Scilingo)

provides the opportunity to assess the value of embodiment in the context of emotional communication between a human being and an artificial interlocutor.

2. Proposed work

As shown in Fig. 1, the proposed framework is composed of three main systems: the real-time acquisition of images from a camera, the analysis of the extracted images to obtain the Action Units (AUs) of the detected subject, the send and execution of the facial movements to the avatar and to the physical robot. In the acquisition phase, the `Frame Grabber` (FG, Algorithm 1), when executed, asks the user to input the desired frame acquisition rate r (r = 5 fps if not specified) and the name of the folder where the frames will be stored. The `setupFolder()` function is then called, creating both a main folder and a temporary one for storing the frames. If folders already exist, the program informs the user that specified folders already exists. Next, the `setupCamera()` function is called, initializing the webcam and configuring its resolution (width w = 640 pixel, height h = 480 pixel) and frame rate. Finally, the `getFrames()` function starts capturing frames from the webcam. For each captured frame, a unique name is generated, including the frame number and timestamp. The frame is then saved as an image in the temporary folder and copied to the main folder. This process continues until the user interrupts the program with a keyboard interruption. When this occurs, the program disconnects the webcam, closes all `OpenCV` windows, and removes both the main and temporary folders. If an error occurs during the frame capture (e.g., if the webcam is unavailable), the program notifies the user that it cannot initiate a new webcam recording. The second program, i.e., the `Event Handler` (EH, Algorithm 2) is a file monitoring system that responds to the creation of new files by sending them to a server for processing and subsequently forwarding the results to another server. It uses the `watchdog` module to monitor a specified directory for new files. The program starts by defining the directory to monitor and then enters a waiting loop until the specified directory exists. Once this condition is met, it begins monitoring it using the `OnMyWatch` class. This class utilizes the Observer class from the `watchdog` module to monitor the directory. The `OnMyWatch` class has a run method that initiates the observation and waits for file system events. When a file system event is detected, the `on_any_event()` method of the `Handler` class is called. This method checks if the event corresponds to the creation of a new file. If a new file is detected, its path is passed to the `emaution()` function, which sends the file in a binary representation I to a local server for processing and returns an XML response. The XML response is then sent to another server using the `sendToAbel()` and `sendToAvatar()` function. This process continues until the user interrupts the program or an error occurs. Two Flask web applications [15], i.e., `ReceiverAvatar` (RAv, Algorithm 3) and `ReceiverAbel` (RAb, Algorithm 4), are structured as web services that receive XML data, extract AUs values, and send them in the appropriate format to the avatar and the robot using the `sendAUsAbel()` and `sendAUsAvatar()` functions. Using Flask applications enable the system with a high versatility and scalability, because, when they receive a post request to the relative write endpoint, they call the write function, which extracts the data from the request, performs some formatting steps, sends the data to the avatar or to Abel using the relative `sendAUs()` function, and returns a response to the original request. Another Flask application using the `Emotiva` API is responsible for predicting facial AUs and estimated emotions from a single image I. `Emotiva` is a Facial Expression Recognition (FER) software able to analyze human attentive and affective states [14]. A post call is made each time the event handler detects the capture of a frame in the specified folder. The virtual avatar used in this framework is based on the *OpenFACS* project, an open-source *FACS*-based 3D face animation system [11, 12]. It is a software that enables the simulation of realistic facial expressions by manipulating specific AUs as defined in the FACS. *OpenFACS* includes an API suitable for generating real-time dynamic facial expressions for a three-dimensional character. It can be easily integrated into existing systems without requiring prior experience in computer graphics.

Algorithm 1 Frame Grabber

```
function setupFolder(folder_name)
    if folder_name is not specified then
        folder_name ← '/correct/path/to/frame_folder'
    if folder_path doesn't exist then
        create folder_path
    return folder_path
function setupCamera(r)
    if r is not specified then
        r ← '5'
    camera_port ← 0
    w ← 640
    h ← 480
    wait_FPS ← 1000/r
    camera ← initialize camera at camera_port
    open camera
    set frame dimensions for camera with dimensions w and h
    return camera, wait_FPS
function getFrames(camera, wait_FPS, folder_path)
    i ← 1
    try: until no Interruption from user
    while camera ready to record do
        I ← capture frame from camera
        frame_path ← frame_folder + 'frame_' + i
        save I in frame_path
        i ← i + 1
        do nothing for wait_FPS/1000
    except: KeyboardInterrupt
    close camera
    delete folder_path
```

Algorithm 2 Image sender

```
function sendAUs(s)
    v ← content of the post request to '/AUs_write_port'
    return response
function detectNewImage(event)
    if new image in the folder then
        send v to Flask receiver server
```

Algorithm 3 Avatar's receiver

```
initialize Flask instance
function write('/write_port', method='POST')
    s ← content of the post request to '/AUs_extraction_port'
    v ← extract list from s
    send v and movement speed to avatar
    return 'Data received'
```

Algorithm 4 Abel's receiver

```
initialize Flask instance
function write('/write_port', method='POST')
    s ← content of the post request to '/AUs_extraction_port'
    v ← extract list from s
    send v and movement speed to Abel
    return 'Data received'
```

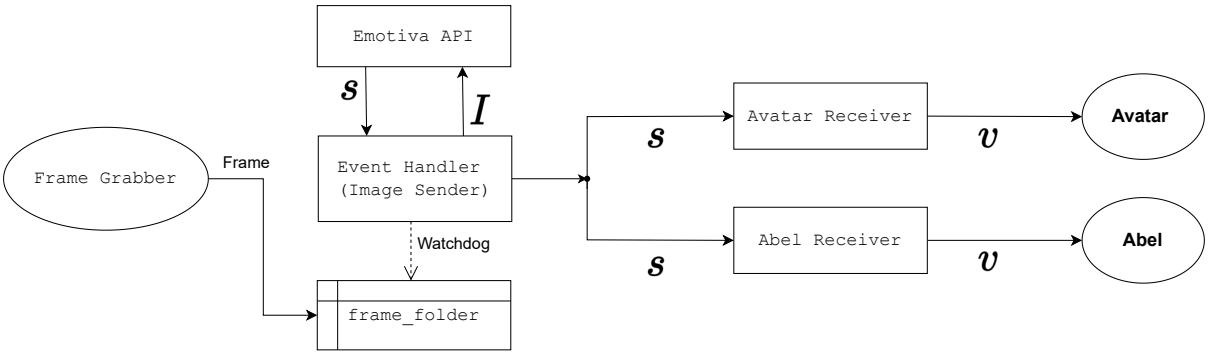
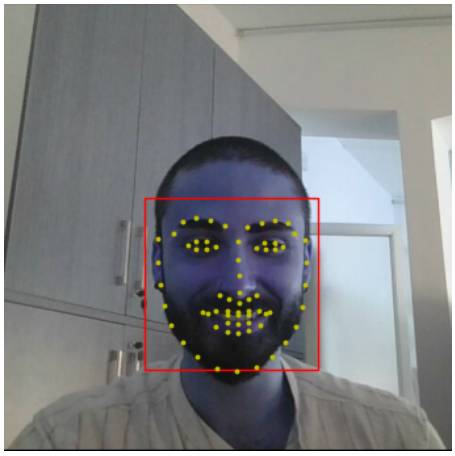


Figure 1: Schematic of the proposed framework.

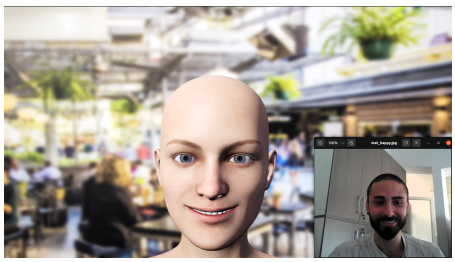
3. Experimental results

The proposed framework was developed with Python on PC platform. We run the application with a frame rate $r = 5$ fps using the integrated webcam, taking images of size 640 x 480 pixel. In Table 1, AUs used for the experiment are listed. In our tests, the facial expressions assumed by the digital avatar as well as by the robot successfully followed the AUs extracted by the Emotiva API and, although they were noisy data acquired with non-specific equipment, it was possible to effectively control the movement of both the virtual and physical agents simply changing the user facial expression. This is shown in Fig. 2c and 2d in the case of the avatar, and in Fig. 2e and 2f in case of the robot. The landmarks are represented as yellow dots superimposed on the two images of the subjects (Fig. 2a and 2b). Additionally, a rectangle is used to identify the faces present in the field of view. The values of the AUs in both cases are also displayed in Figures 2g and 2h. To improve the quality of control, we plan, instead of using the PC camera, to use a Kinect camera directly connected to Abel for image acquisition, which has a higher resolution, and to increase the number of AUs involved in the agent control. To set up the experiment accurately it is necessary to be in a very bright environment, preferably under direct light to increase the contrast of the acquired image. It is also advisable to choose a sufficiently high frame rate to achieve real-time control of the robot and the avatar. Selecting values that are too low can lead to latency issues in the avatar system. Additionally, during the experiment, the subject's face should not exit the camera's field of view or rotate more than an angle of about 30° from the central position. Processing a partial face is not supported. If this occurrence happens, the results are deemed unreliable, and the user receives an alert message. In case of detection of multiple subjects in the field of view, a processing is performed for each visible face. In the context of the presented framework, this could generate conflicts in the control of the avatar and the robot, since there is no decision-making algorithm included in this framework. This problem is solved by the integration of the proposed framework with the high-level cognitive processing (i.e., the Plan block of Abel's control architecture [13]) that makes the artificial agents able to focus their attention on a specific subject according to specific attention rules [16].

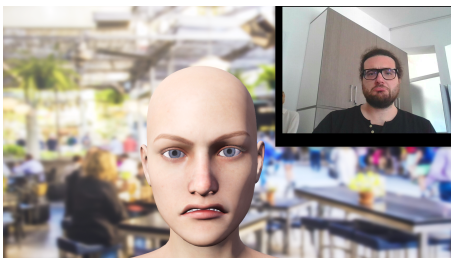


(a) Happy expression

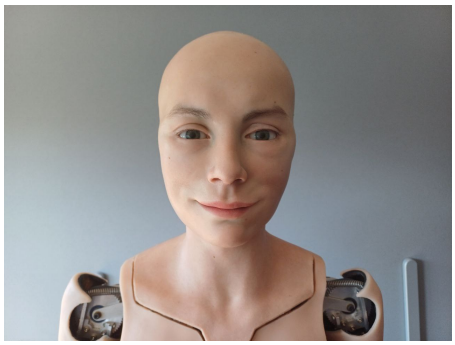
(b) Sad expression



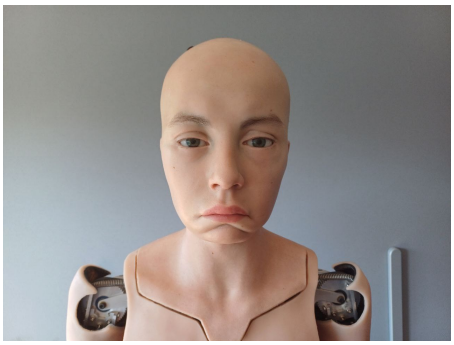
(c) Happy expression, avatar face



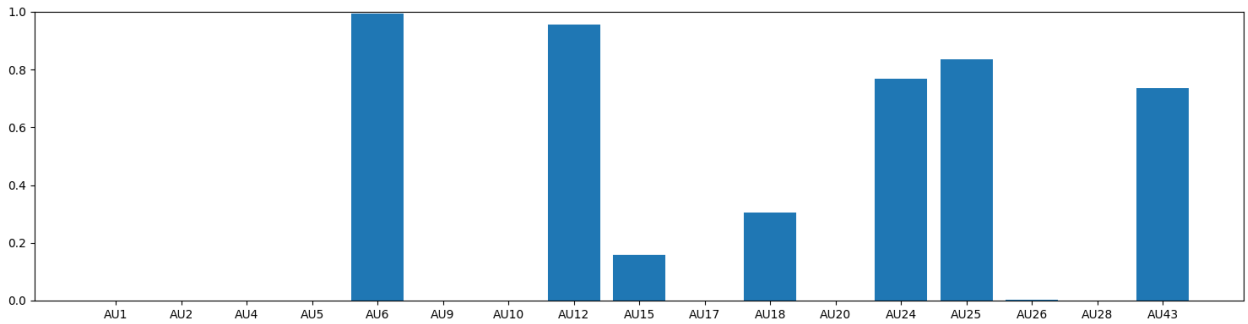
(d) Sad expression, avatar face



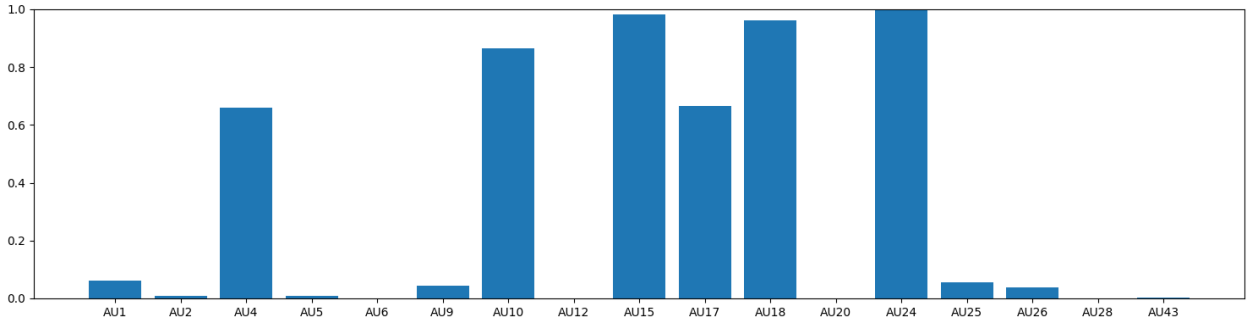
(e) Happy expression, Abel face



(f) Sad expression, Abel face



(g) Happy expression AUs and emotions values



(h) Sad expression AUs and emotions values

Figure 2: Examples of analyzed expressions of different users and related control of the facial mimicry of the digital avatar and Abel robot.

AU index	AU description
1	Inner brow raiser
2	Outer brow raiser
4	Brow lowerer
5	Upper lid raiser
6	Cheek raiser
9	Nose wrinkler
10	Upper lip raiser
12	Lip corner puller
15	Lip corner depressor
17	Chin raiser
18	Lip puckerer
20	Lip stretcher
24	Lip pressor
25	Lips part
26	Jaw drop
28	Lip suck
43	Eyes closed

Table 1
AUs used in the proposed paper.

4. Conclusions and future developments

The use of a digital avatar introduces several advantages, such as simplifying certain phases of development and testing which would normally involve the correspondent physical robot, helping to prevent the inevitable wear and tear of the robot's electronic and mechanical components, and the high scalability and affordability of this technology. On the other hand, we are aware of the importance and the influence of a social robot corporeality in HRI, especially in several clinical applications (e.g., [17, 18, 19, 20]. The presented architecture allows exploratory interaction studies where it will be possible to compare two systems in which the perception and information processing remain identical, with the only differing variable being the representation and embodiment of the artificial agent. These studies will lead to a methodological evaluation using standard scales (e.g., the Godspeed methods [21]) comparing the cases of Abel and the digital avatar. Moreover, the degrees of freedom of the avatar are limited e.g., it cannot make asymmetric expressions, and it cannot move any part of the body but the face. To improve this aspect, the next steps of the project will be also to modify the code of the avatar at a lower level, separating the right and left part of its face, and to build the graphical component and the control of other expressive body parts, such as neck, arms and hands.

Acknowledgments

Thanks to the developers of Emotiva https://emotiva.it/ and openFACS https://github.com/phuselab/openFACS. Research partly funded by PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI", funded by the European Commission under the NextGeneration EU programme.

References

[1] S. Li, W. Deng, Deep facial expression recognition: A survey, IEEE transactions on affective computing 13 (2020) 1195–1215.

[2] D. Canedo, A. J. Neves, Facial expression recognition using computer vision: A systematic review, Applied Sciences 9 (2019) 4678.

[3] P. Ekman, W. V. Friesen, Facial action coding system, Environmental Psychology & Nonverbal Behavior (1978).

[4] C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, B. Blumberg, Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots, Artificial life 11 (2005) 31–62.

[5] T. Wu, N. J. Butko, P. Ruvulo, M. S. Bartlett, J. R. Movellan, Learning to make facial expressions, in: 2009 IEEE 8th International Conference on Development and Learning, 2009, pp. 1–6. doi:10.1109/DEVLRN.2009.5175536.

[6] S. Boucenna, P. Gaussier, P. Andry, L. Hafemeister, A robot learns the facial expressions recognition and face/non-face discrimination through an imitation game, International Journal of Social Robotics 6 (2014) 633–652.

[7] A. Meghdari, S. B. Shouraki, A. Siamy, A. Shariati, The real-time facial imitation by a social humanoid robot, in: 2016 4th International Conference on Robotics and Mechatronics (ICROM), IEEE, 2016, pp. 524–529.

[8] H. Kobayashi, F. Hara, Facial interaction between animated 3d face robot and human beings, in: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, volume 4, 1997, pp. 3732–3737 vol.4. doi:10.1109/ICSMC.1997.633250.

[9] D. Li, C. Sun, F. Hu, D. Zang, L. Wang, M. Zhang, Real-time performance-driven facial animation with 3ds max and kinect, in: 2013 3rd International Conference on Consumer Electronics, Communications and Networks, 2013, pp. 473–476. doi:10.1109/CECNet.2013.6703372.

[10] N. Rawal, D. Koert, C. Turan, K. Kersting, J. Peters, R. Stock-Homburg, Exgennet: Learning to generate robotic facial expression using facial expression recognition, Frontiers in Robotics and AI 8 (2022) 730317.

[11] V. Cuculo, A. D'Amelio, Openfacs: An open source facs-based 3d face animation system, in: Y. Zhao, N. Barnes, B. Chen, R. Westermann, X. Kong, C. Lin (Eds.), Image and Graphics, Springer International Publishing, Cham, 2019, pp. 232–242.

[12] openFACS, 2023. URL: https://github.com/phuselab/openFACS.

[13] L. Cominelli, G. Hoegen, D. De Rossi, Abel: integrating humanoid body, emotions, and time perception to investigate social interaction and human cognition, Applied Sciences 11 (2021) 1070.

[14] Emotiva, 2023. URL: https://emotiva.it/.

[15] Flask, 2023. URL: https://flask.palletsprojects.com/en/2.3.x/.

[16] L. Cominelli, D. Mazzei, D. E. De Rossi, Seai: Social emotional artificial intelligence based on damasio's theory of mind, Frontiers in Robotics and AI 5 (2018) 6.

[17] S. Shamsuddin, H. Yussof, L. Ismail, F. A. Hanapiah, S. Mohamed, H. A. Piah, N. I. Zahari, Initial response of autistic children in human-robot interaction therapy with humanoid robot nao, in: 2012 IEEE 8th International Colloquium on Signal Processing and its Applications, 2012, pp. 188–193. doi:10.1109/CSPA.2012.6194716.

[18] S. Shamsuddin, H. Yussof, L. I. Ismail, S. Mohamed, F. A. Hanapiah, N. I. Zahari, Initial response in hri-a case study on evaluation of child with autism spectrum disorders interacting with a humanoid robot nao, Procedia Engineering 41 (2012) 1448–1455.

[19] A. Tapus, A. Peca, A. Aly, C. Pop, L. Jisa, S. Pintea, A. S. Rusu, D. O. David, Children with autism social engagement in interaction with nao, an imitative robot: A series of single case experiments, Interaction studies 13 (2012) 315–347.

[20] L. J. Wood, A. Zaraki, B. Robins, K. Dautenhahn, Developing kaspar: a humanoid robot for children with autism, International Journal of Social Robotics 13 (2021) 491–508.

[21] C. Bartneck, E. Croft, D. Kulic, Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots, International Journal of Social Robotics 1 (2009) 71–81. doi:10.1007/s12369-008-0001-3.