Avatar Reaction to Multimodal Human Behavior

Baptiste Chopin, Mohamed Daoudi, Angela Bartolo

Avatar Reaction to Multimodal Human Behavior*

Baptiste Chopin[1], Mohamed Daoudi[2,3], and Angela Bartolo[4,5]

[1] Université Côte d'Azur, Inria,France
[2] Univ. Lille, CNRS, Centrale Lille, Institut Mines-Télécom, UMR 9189 CRIStAL, Lille, F-59000, France
[3] IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, Lille, F-59000, France
[4] CNRS, UMR 9193-SCALab-Sciences Cognitives et Sciences Affectives, University of Lille- SHS, Villeneuve d'Ascq, 59000, Lille, France
[5] Institut Universitaire de France (Paris)

Abstract. In this paper, we propose a virtual agent application. We develop a virtual agent that reacts to gestures and a virtual environment in which it can interact with the user. We capture motion with a Kinect V2 camera, predict the end of the motion and then classify it. The application also features a facial expression recognition module. In addition, to all these modules, we include also OpenAI conversation module. The application can also be used with a virtual reality headset.

Keywords: avatar reaction · human motion prediction · facial expression.

1 Introduction

The development of a virtual agent that can interact with users in a virtual environment has potential applications in fields such as gaming, education, and healthcare. Interaction with these virtual agents is provided and has usually been studied, through several modalities: expression recognition and generation [7], action recognition and interaction with objects [6], multi-modal input and output [7]. Some methods can even be used for both virtual agents and robots [9]. However, the challenge of building an interactive virtual agent differs from that of building an interactive robot. Heting et al. [14] discuss the use of virtual agents, specifically avatars, in human-computer interactions. The paper presents an emotionally responsive avatar named Diana, which recognizes human affect and responds with natural facial expressions to improve the user experience in the interaction. The more recent work focus on facial expression generation in dyadic interactions [12], while our work is interesting by the reaction of an avatar to action.
Daoudi et al. [5] propose a new computational approach to identify human

* This work was supported by French government funding managed by the National Research Agency under grants ANR-21-ESRE-0030 (CONTINUUM), by CNRS through the 80—Prime program, and ANR-16-IDEX-0004 ULNE.

social intention in action by analyzing the trajectories of the human arm. The proposed method for predicting social intention is based on the trajectories of the human arm, which has been used to improve social communication between humans and virtual agents. To our knowledge there exist no virtual agent that is able to react to communicative gesture. Our virtual agent is able to react to communicative gestures using non-intrusive data acquisition devices while having realistic behavior.

2 Overview

We set several constraints to the application we build: the avatar appearance must be realistic, the environment must be believable, the user's motion data must be collected in an non-intrusive manner, the reaction from the agent must be related to the action of the user and realistic, finally the reaction motion must start quickly after the user perform a gesture. To respect the constraints we set for our virtual agent application, we must first be able to react quickly to the user gesture. This implies that, in a short time, we must understand the motion the user is performing and produce a reaction to this motion. In these conditions, waiting for the user to complete the motion is not ideal. Therefore, we choose to start by predicting the end of the user motion based on its start. Then we classify the predicted motion and send the information to the avatar that will react accordingly by selecting the corresponding prerecorded reaction motion. The result of the classification will let the system decide which reaction to perform. However, as we will see in Section.3.1, all the gestures we choose indicate discomfort for the user. Depending on the degree of said discomfort, a reaction might not be needed. To take this possibility into account, we add a facial expression recognition module to decide if the virtual agent has to react. Fig.1 shows the overall architecture of the application.

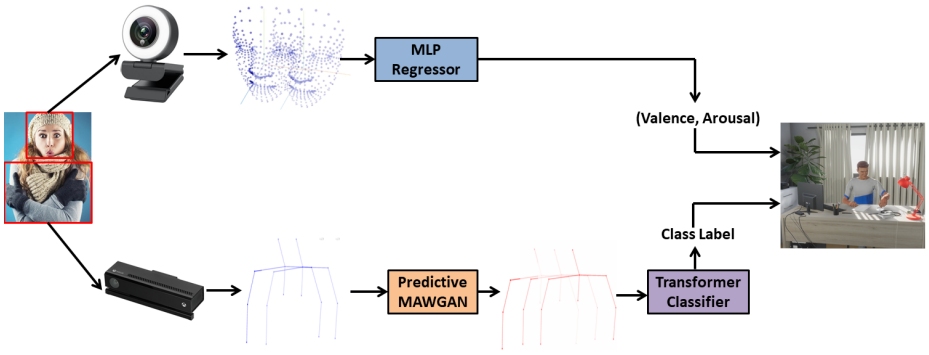

Fig. 1. Overview of the virtual agent application architecture

Scenario	Motions	number of samples MoCap	number of samples Kinect
I am cold	rub both arms with hands	115	109
I am hot	wave one or both hands near the neck/ mouth	115	115
This is too loud	cover both ears with hands	113	99
This smells bad	pinch nose with one hand or cover mouth and nose with both hands	115	110
It is too bright	cover eyes with one or both hands	111	109
Hello	wave with one or both hands	100	121
Idle	random and meaningless motions	59	65
Total		728	728

Table 1. Content of the gesture dataset

A final issue to address is the need for data. While facial expression databases are plentiful and we do not need data to generate the virtual agent's response, we do need data to train networks to predict and recognize the user's gesture. In summary, the main contributions of our work are:

1. Development of a virtual agent that reacts to gestures and interacts with the user in a virtual environment.
2. A new dataset of 3D gestures of 11 subjects to train our models that are used with the virtual agent, is proposed. Each 3D gesture is recorded by motion capture. The motions were performed by 11 healthy volunteer participants who were asked to perform gestures indicating their mental state.

3 Proposed Method

3.1 Motion Database

There is currently no database containing the kind of communicative gestures that we are looking for when interacting with the avatar (e.g "I'm cold", "My ears hurt"...). Most motion databases contain motions that represent action performed alone (walking, kicking...) or in interaction with other persons (exchanging an item, fighting...). In the latter, the interactions require physical contact between the two parties, either directly or through an object. Gesture databases are also available, however, the gestures included are gestures that do not require a reaction from another person. Furthermore, the existing gesture databases that could be used to make another person react are culturally coded, the motion performed might not have the same meaning or no meaning at all for a French person. To get data that we could use in our scenarios we build a database of communicative motions.

We show in Table 1 the different scenarios and motions used in our database as well as a description and the number of samples recorded. Since different motions can be used for each of the scenarios given to the participants, they were also instructed to avoid certain types of motion (*e.g* they were not allowed to pull the collar of their shirt during the "I am hot" scenario). We added an additional class of motions containing random idle motions to train our networks. It is used to recognize when no motion of particular significance is being performed. By doing that, we can avoid the misclassification of random motions as meaningful ones leading to an unwanted reaction from the avatar. The scenarios were selected from situations where people might perform a gesture to indicate

their mental state but that can also be performed to elicit a response from another person (e.g. the gesture of "coming here" oblige observers to modify their behavior. In this case, people move toward the executor). Mental state gestures are gestures that communicate a feeling (*e.g.* I am cold). In all our scenarios the motion indicates a discomfort and the avatar must react by helping the user.

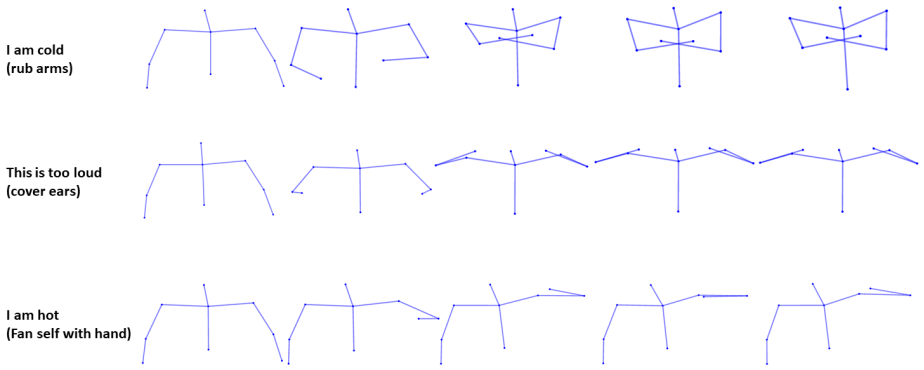

Fig. 2. Examples of motion from the new database (motion capture). We show three different classes. Visualization obtained after motion sequences were cut and resampled to have a duration of 3*s* (see Sec.3.3)

The motion capture was performed at the Equipex Continuum using the facilities of the Fédération de Recherche "Sciences et Cultures du visuel" at Tourcoing, France. We recruited 11 healthy volunteer participants among the students and staff of the lab team (8 males, 3 females; age between 20-60 yrs). Participants did not receive financial compensation for their participation in the study. Participants were asked to stand in front of the Kinect at a place marked by a cross on the ground. Fig.4 shows the view participants had. Their initial position was then adjusted depending on their height (to allow the capture of all markers by the motion capture system). They were asked to perform the gestures from Table.1. The motions from class "Idle" were only performed by 2 participants as they are random, meaningless, motions. The participants performed each motion at least 10 times. Each participant spent between 30 and 45 minutes to complete the entire recording session. The motions are captured on two different systems. The first is a high-precision motion capture system with 6 Qualisys motion capture cameras and nine markers on the participant's body: one on the forehead between the eyebrows, one at the base of the neck, and one above the navel, for each arm we have markers on the top of the shoulder, on the elbow and on the outer side of the wrist. Fig.3 shows the position of the markers on a participant. The motion capture system is calibrated to have an average positional error of less than 2 millimeters. Each sample is recorded for exactly 5 seconds regardless of the actual duration of the motion at 100 fps.

Fig. 3. Motion capture markers position

The second capture system we use consists of a single Kinect V2 camera. We decided to also capture the motion with a Kinect camera since the motion capture system would give us extremely precise data that we will not be able to obtain with the system we have put in place for our application. The Kinect allows us to capture real-world data of the motion present in the motion capture database which will help our models become more resilient to the noise there is in real-world data. With the Kinect, we capture joint positions of the entire body but then only keep the nine joints corresponding to those from the Motion capture. The capture setup is illustrated in Fig.4. Due to the difficulty of setting up synchronous capture between the Kinect and the motion capture system, the capture from the Kinect is started and stopped by an experimenter following the instruction from the experimenter working with the motion capture system. Due to this, the duration of the Kinect capture varies between 73 and 530 frames with a median of 155 frames and a mean of 157 frames with a framerate of 30 fps which means that on average the motion duration average is close to 5s. Samples that were too noisy or where joint positions were lost for too long were removed. When a joint position was lost for a short amount of time we estimate its position to be the last known position. All recorded samples were checked by a human curator before being used in the database. This amount to 728 motion samples for the MoCap data and 728 samples for the Kinect data.

3.2 Data Capture

To avoid having to wear captors when using the application, we can capture the motion data using depth cameras or RGB cameras. Both allow for the capture

Fig. 4. Motion capture and Kinect setup. Point of view of the captured subject.

of skeletons through different toolkits. For example, OpenPose [3] for RGB camera or the Kinect SDK for the depth cameras. We tried a RGB camera with BlazePose [1], Kinect depth camera, and Intel real sense depth camera with NuiTrack SDK [6]. The result led us to choose the Kinect camera. Blazepose, while good for 2D joint detection, fails to correctly estimate the depth coordinates and causes the recorded data to be noisy. The intel real sense with NuiTrack often fails completely to detect the joint position of the arm and the recorded motion looks nothing like the actual motion. On the other hand, Kinect gives an accurate recording of the motion even if it is quite sensitive to occlusion.

For the face data, we use a RGB camera with facial landmark detection software. Out of the many libraries that exist (OpenFace [7], dlib[8]...) we choose to use Mediapipe FaceMesh[9] which uses a facial detector [2] coupled with a face mesh predictor [10]. We choose to use this method due to the speed and light weight of the model but also because of its accuracy and higher resistance to occlusion. Even if it uses a face mesh instead of proper landmarks, the results on expression recognition are similar to other landmarks detectors.

3.3 Action Prediction and recognition

Action Prediction. Prediction Module is based on [4]. We have shown that the prediction speed is fast enough to be used in real time which is mandatory to be used in our application. In [4] we predicted the same number of frames as what we were given as input. Here, however, we take an input of 25 frames ($1s$) to predict 50 frames ($2s$). The Predictive Network is trained on our dataset,

[6] `https://nuitrack.com/`
[7] `https://github.com/TadasBaltrusaitis/OpenFace`
[8] `http://dlib.net/`
[9] `https://google.github.io/mediapipe/solutions/face_mesh`

Table 2. Recognition accuracy on our dataset. Comparison between $1s$ of motion captured by the Kinect, $3s$ of motion captured by the Kinect, and $1s$ of motion captured by the Kinect concatenated to $2s$ of predicted motion

	Capture $1s$	Capture $3s$	Capture $1s$ + Prediction $2s$
Accuracy	47.7%	63.6%	60.2%

where we mix the Kinect and Mocap data. The motion sequences were cut and resampled to have a duration of $3s$.

Action Recognition. To classify the predicted motions we use a Transformer encoder that uses spatial attention and the skeleton adjacency module presented in [4]. The Transfomer encoder is followed by a simple multi-layer perceptron to classify the motions. Fig. 5 shows the architecture of the action recognition network. . We then train the network on $3s$ of motion composed of $1s$ of real data and the $2s$ of motion predicted by the prediction module. We train with predicted data instead of real data because the recognition module will always receive predicted data. This module estimates the class of the concatenation of the recorded motion and the predicted motion. Table. 2 shows the accuracy of the network on our dataset. For this experiment, we split the dataset between train and test. Out of the 11 subjects we randomly choose subject 2 and subject 7 to be the test subjects. For class "Idle", since the samples are not connected to a particular subject we randomly take 11 samples from the Kinect data and 11 samples from the Mocap data to create the test set. With this, we have 1192 training samples and 264 test samples. We compare the result of $3s$ of captured data, $1s$ of captured and $1s$ of captured data completed by $2s$ of predicted data. For each modality, we train the network on the corresponding data. As for the prediction network we use both the Mocap and Kinect data. We see that using $3s$ of data the network correctly classifies 63.6% of action. Some actions of our dataset like "Hello" and "I am hot" (see Table 1 for the detailed motions) are similar and often confused which can explain the relatively low score. If we use only $1s$ of data we see a decrease in performance down to 47.7% accuracy. If we use our prediction network to predict $2s$ of motion and combine it with the $1s$ of input data to get a $3s$ sequence we increase the results to 60.2% close to the original 63.6%. This highlights the advantage of using our prediction network. We can are able to classify motion with only one-third of the data nearly as accurately. In practice this allows the avatar to react thrice as fast since the prediction process only takes a few milliseconds.

3.4 Facial Expression Recognition

To avoid reacting to random motion or communicative motion that does not require a reaction from the avatar, we add a facial expression recognition module that estimates the valence and arousal value of the user expression. We use a simple MLP consisting of 4 Dense layers with dropout that we train on the AFEW-VA[11] dataset using landmark data captured with Mediapipe. While the

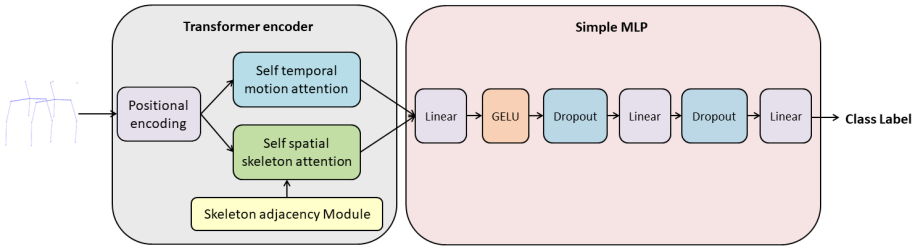

Fig. 5. Architecture of our Interformer-based motion classifier

network might seem extremely simple, it is efficient. On the AFEW-VA dataset, with valence and arousal values comprised between -10 and 10, we obtain an error of only 1. State of the art method might perform even better but since our goal is only to set a threshold for discomfort while being fast we do not need as much accuracy as the state-of-the-art methods. We want to prevent the avatar from reacting when the user is in a neutral or positive mood since the motion performed might not indicate discomfort in that case. To do so we ignore motions performed while $valence > 0$ as it correlates with positive feelings. To avoid reacting when the user is in a neutral mood, we decide to only consider motion performed while $valence < -3$ and $|arousal| > 3$. We capture the face of the user with a second camera independent from the Kinect. The Kinect must be far enough to see the entire upper body and can not be simultaneously used to capture a detailed face.

4 Avatar Environment

Our avatar module is a Unity application where our avatar evolves in a scene consisting of a room with furniture as shown in Figure 7. The room contains a desk behind which the avatar sits, the avatar can interact with several objects in the scene: cigarette, window, curtains, HiFi system, and a keyboard. We choose a simple appearance for the avatar with a tee shirt so that users will not focus on its appearance but rather on its reaction. When reacting to the user gesture the avatar will stand up (except for the "This smells bad" scenario) and perform an action behind the desk (See Table 3 for a complete description of each reaction). This module can also work in virtual reality and the entire room is modelized in unity as seen in Figure 6. When using the virtual reality version, since the user needs to wear a virtual reality helmet we can not use facial expression recognition. In that case, the output of the module is ignored.

The full application combines the aforementioned modules. The motion data is captured with a Kinect V2 camera and the face data with any RGB camera such as a webcam. From the motion data from the Kinect, we keep only the 3D skeleton data that we send to our prediction network every 25 frames. In the prediction network, the data will first be normalized and transformed into

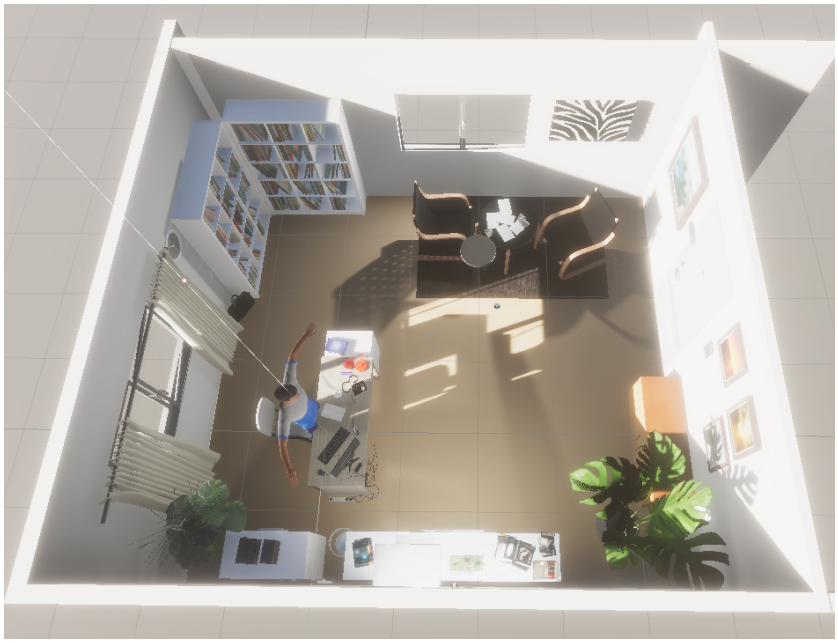

Fig. 6. Aerial view of the scene in the unity builder (ceiling is removed). The entire room is modelized for virtual reality.

Scenario	Motions
I am cold	If the window is open: stand up, close the window, sit down.
I am hot	If the window is closed: stand up, open the window, sit down.
This is too loud	if the volume is high: stand up, turn the volume down on the hi-fi system, sit down
This smells bad	if the cigarette is lit: put out cigarette
It is too bright	if the curtain is open: stand up, close the curtain, sit down
Hello	Stop idle motion and turn to face and look at the user
Before turning attention to the user	look at the computer screen while typing on the keyboard. Sometimes turns its head to look at the notebook.
Attention state	look at the user and sometimes at objects on the desk.

Table 3. Avatar reaction to the gestures described in Table 1

The Square Root Velocity (SRVF) [13], [8], [4]. Then we predict the 50 future frames of the motion in SRVF format and rebuild the corresponding skeleton sequence. The predicted sequence is concatenated with the sequence recorded by the Kinect for a total length of 75 frames. This concatenated sequence is fed to the action recognition network which outputs a class label. Simultaneously 25 frames of the face data are sent to the facial expression recognition module where we extract the facial landmark from the RGB data with MediaPipe's face mesh. The landmarks are treated by the network which outputs arousal and valence values. Our scenarios are all negative ones, the user is bothered by something and wants the avatar to do something about it. We choose valence and arousal intervals that contain expressions that users may perform in these scenarios. If the values obtained by the network are outside these intervals we consider that the motion performed by the user does not show a discomfort needing a reaction or that it is unrelated to the scenarios. In these cases, we prevent the avatar from

Fig. 7. Our avatar in its resting state.

reacting. Depending on the output of the motion recognition and of the facial expression recognition network the application will decide if the avatar needs to react and if so start the appropriate motion. If no reaction motion is required because the predicted class is 'Idle' or due to the valence and arousal values, the avatar stays in its 'Attention' state. This cycle is repeated until the application is closed. When using virtual reality users can move the camera with their heads and can move around in the room. However, the position of the avatar remains unchanged and a change in user position too drastic will interfere with the body data capture from the Kinect. Figure. 1 describes the full application.

In addition to the above modules, we have also developed a module for facial expression with VR headset. Indeed, recognizing emotions while wearing a VR headset is a much more complicated task, as the headset hides information. Vive offers a facial tracker that uses sensors to capture facial movements. Facial expression is defined on the basis of 38 characteristic movements, which are assigned a weight according to their presence in the expression. To recognize certain emotions (joy, anger, surprise, sadness), we selected from the 38 movements those that best defined the desired emotion, then set a threshold on the weight of these movements. If the weight associated with an emotion exceeds this threshold, then the user is considered to be expressing that emotion. To be able to converse with the avatar, we used the OpenAI API integrated into Unity, as well as Microsoft Azure's text-to-speech and speech-to-text services. We convert the user's voice to text and send it to the OpenAI model (we use the gpt-3.5-turbo model) using speech-to-text conversion. We then use text-to-speech conversion to make the model's response audible.

5 Acknowledgments

The authors would also like to thank Hugo Pina Borges and Julian Hutin for performing part of the experimentation.

6 Conclusions and Future Work

In this paper, we propose a virtual agent application that reacts to gestures and a virtual environment in which it can interact with the user. The application captures motion with a Kinect V2 camera, predicts the end of the motion, and then classifies it. It also features a facial expression recognition module and an OpenAI conversation module. The application can be used with a virtual reality headset. In addition, we propose a new database of communicative motions that were recorded using a Kinect V2 camera. The motions were performed by 11 healthy volunteer participants who were asked to perform gestures indicating their mental state. One limitation of our work is to establish criteria to evaluate the human-avatar interaction.

References

1. Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T.L., Zhang, F., Grundmann, M.: Blazepose: On-device real-time body pose tracking. ArXiv **abs/2006.10204** (2020)
2. Bazarevsky, V., Kartynnik, Y., Vakunov, A., Raveendran, K., Grundmann, M.: Blazeface: Sub-millisecond neural face detection on mobile gpus. ArXiv **abs/1907.05047** (2019)
3. Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A.: Openpose: Real-time multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
4. Chopin, B., Otberdout, N., Daoudi, M., Bartolo, A.: 3-d skeleton-based human motion prediction with manifold-aware GAN. IEEE Trans. Biom. Behav. Identity Sci. **5**(3), 321–333 (2023). https://doi.org/10.1109/TBIOM.2022.3215067, `https://doi.org/10.1109/TBIOM.2022.3215067`
5. Daoudi, M., Coello, Y., Descrosiers, P.A., Ott, L.: A new computational approach to identify human social intention in action. In: 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018). pp. 512–516 (2018)
6. Daoudi, M., Coello, Y., Desrosiers, P.A., Ott, L.: A new computational approach to identify human social intention in action. In: 13th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2018, Xi'an, China, May 15-19, 2018. pp. 512–516 (2018). https://doi.org/10.1109/FG.2018.00082, `https://doi.org/10.1109/FG.2018.00082`
7. DiPaola, S., Yalçın, O.N.: A multi-layer artificial intelligence and sensing based affective conversational embodied agent p. 2
8. Drira, H., Ben Amor, B., Srivastava, A., Daoudi, M., Slama, R.: 3D face recognition under expressions, occlusions, and pose variations. PAMI **35**(9), 2270–2283 (2013)

9. Hua, M., Shi, F., Nan, Y., Wang, K., Chen, H., Lian, S.: Towards more realistic human-robot conversation: A seq2seq-based body gesture interaction system. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1393–1400 (2019). https://doi.org/10.1109/IROS40897.2019.8968038
10. Kartynnik, Y., Ablavatski, A., Grishchenko, I., Grundmann, M.: Real-time facial surface geometry from monocular video on mobile gpus. In: CVPR Workshop on Computer Vision for Augmented and Virtual Reality 2019. Long Beach, CA (2019), https://arxiv.org/abs/1907.06724
11. Kossaifi, J., Tzimiropoulos, G., Todorovic, S., Pantic, M.: Afew-va database for valence and arousal estimation in-the-wild. Image and Vision Computing **65** (02 2017). https://doi.org/10.1016/j.imavis.2017.02.001
12. Luo, C., Song, S., Xie, W., Spitale, M., Shen, L., Gunes, H.: Reactface: Multiple appropriate facial reaction generation in dyadic interactions. CoRR **abs/2305.15748** (2023)
13. Srivastava, A., Klassen, E., Joshi, S.H., Jermyn, I.H.: Shape analysis of elastic curves in euclidean spaces. PAMI **33**(7), 1415–1428 (2011)
14. Wang, H., Gaddy, V., Beveridge, J.R., Ortega, F.R.: Building an emotionally responsive avatar with dynamic facial expressions in human—computer interactions. Multimodal Technologies and Interaction **5**(3) (2021)