Multimedia Tools and Applications (2022) 81:3421–3438
https://doi.org/10.1007/s11042-021-11289-x

1188: ARTIFICIAL INTELLIGENCE FOR PHYSICAL AGENTS

Modeling and evaluating beat gestures for social robots

Unai Zabala[1] · Igor Rodriguez[1] · José María Martínez-Otzeta[1] · Elena Lazkano[1]

Received: 29 January 2021 / Revised: 4 May 2021 / Accepted: 9 July 2021 /
Published online: 3 August 2021

Abstract

Natural gestures are a desirable feature for a humanoid robot, as they are presumed to elicit a more comfortable interaction in people. With this aim in mind, we present in this paper a system to develop a natural talking gesture generation behavior. A Generative Adversarial Network (GAN) produces novel beat gestures from the data captured from recordings of human talking. The data is obtained without the need for any kind of wearable, as a motion capture system properly estimates the position of the limbs/joints involved in human expressive talking behavior. After testing in a Pepper robot, it is shown that the system is able to generate natural gestures during large talking periods without becoming repetitive. This approach is computationally more demanding than previous work, therefore a comparison is made in order to evaluate the improvements. This comparison is made by calculating some common measures about the end effectors' trajectories (jerk and path lengths) and complemented by the Fréchet Gesture Distance (FGD) that aims to measure the fidelity of the generated gestures with respect to the provided ones. Results show that the described system is able to learn natural gestures just by observation and improves the one developed with a simpler motion capture system. The quantitative results are sustained by questionnaire based human evaluation.

Keywords Social robots · Motion capturing and imitation · Generative adversarial networks · Talking movements · Fréchet gesture distance

1 Introduction

The main aim of social robotics [2] is to endow robots with artificial social intelligence in order to advance in a more natural human-machine interaction and make them participate in

This work has been partially supported by the Basque Government (IT900-16 and Elkartek 2018/00114), the Spanish Ministry of Economy and Competitiveness (RTI 2018-093337-B-100, MINECO/FEDER,EU).

✉ Unai Zabala
unai.zabalac@ehu.eus

[1] Computer Science and Artificial Intelligence, Faculty of Informatics, UPV/EHU, Manuel Lardizabal 1, 20018 Donostia, Spain

complex human contexts. To achieve the desired level of sophistication in robot behaviors, it is necessary to model and implement sensing, processing and interacting capabilities similar to those presented in humans. Emotions, intentions, motivations, and other related cognitive functions are also needed to be taken into account.

Movements, postures and all kind of spontaneous gesticulation are involved in talking and social interactions even if they vary among cultures and are largely subjective. If we want to make people feel confident when interacting with robots, while building trust in the process, a human-like talking gesticulation behavior is highly desirable. Different kind of gestures involving head, arms and hands are used both to reinforce the meaning of the speech and to express emotional state through non-verbal signs. The present work is limited in scope to beats, i.e. movements of body parts that occur during a conversation and that although they are synchronised with the general flow of speech, they do not have a particular meaning associated to them [18]. Generation of a representative set of gesture motions that it is also variate enough to avoid repetitive behavior is a hard task to perform manually with handmade animations [24]. A friendlier way of acquiring such natural interaction ability by social robots would be, undoubtedly, through just observation of human behavior, as in [36]. However, in order to capture human motion markers were needed to record hand information.

The present approach intends to take one step forward to improve social robots gesticulation capabilities, therefore allowing us to capture the naturalness with which we gesticulate when talking and then transfer such properties to a robot. Opposite to [36], a full markerless motion capturing and system is used, which makes it easier to use with no loss of performance. The movements data is given by OpenPose [5], a real-time multi-person 2D pose estimation that uses a non-parametric representation to learn to associate body parts with individuals in a given image. In addition, a gesture generation model capable of generating human talking beat gestures for a humanoid robot is presented. The obtained model is trained with a database captured by recording humans talking in real time. The main contribution of this paper over the work presented in [37] is the analysis of the improvements evaluated by means of different measures, quantitative and qualitative.

The rest of the paper is structured as follows: Section 2 presents the different generative models found in the literature that are used for robot gesture generation. Next, Section 3 defines the baseline used to later on evaluate the performance of the developed system, the capturing, mapping system, together with the generative approach used. Section 4 summarizes the comparative measures used along with the obtained values. The paper concludes with Section 5 pointing to further research.

2 Approaches for gesture generation

Social robots need a mechanism that allow to generate a natural and appropriate body language, in the way people do. Most models for generating non-verbal behavior are based on rules [8, 20]. Therefore, those systems can produce a limited set of movements and usually are tuned for a particular setting. In contrast, data-driven systems are flexible and easily adjustable. Given a set of examples, data-driven methods aim to learn a model P which to sample from, such that P is as similar as possible to the unknown distribution contained in the input data.

Several authors have explored data-driven approaches for motion generation. In [14] the authors propose the combination of Principal Component Analysis (PCA) [33] and

Hidden Markov Models (HMMs) for encoding different movement primitives to generate humanoid motion. Tanwani [31] uses Hidden Semi-Markov Models (HSMM) for learning robot manipulation skills from humans. These approaches mainly focus on the technical aspects of generating motion like arm lifting, arc circling or tracking and reaching a screwdriver, very different motion skills compared to speech related gestures. Regarding on social robotics, some generative approaches are being applied with different objectives. For instance, in [15] Manfrè et al. use HMMs for dance creation.

Deep learning techniques have also been applied to generative models, giving rise to deep generative models. Recently several deep generative models, mainly Generative Adversarial Networks (GAN) [9] and Variational Autoencoders (VAEs) [11], have been applied for gesture generation. For instance, Nishimura et al. [21] propose a long-term motion generation method by using a generative model trained by short-term motions (CNN-GAN). The purpose of their research was to model the non-verbal communication (mainly upper body motion) of human during interaction for the application of the motion generation of the Ibuki humanoid robot.

However, we found few references to beat gesture generation in robots, albeit it is important to provide social robots with an adequate beat generation system.

According to Bremner et al. [3], selecting naively from a library of gestures is unlikely to result in particularly human-like gesture sequences. They perform an analysis of chat show videos and generate rules for the creation of of human-like beat gestures. These gestures are later combined with hand scripted non-beat gestures, to produce monologues with a complete set of accompanying gestures, concluding that having correctly designed gesture sequences improves observer engagement. As mentioned before, hand made rule based approaches are time consuming, hard and difficult to generalize enough. Wolfert et al. also emphasize the importance of beat gesture generationg is social robots in [34]. They compare generated beat gestures with gestures created manually and conclude that users prefer beat gestures generated by an encoder-decoder DNN model. In their work, the learning model is fed with combined speech and motion information. However, in the experimentation no robot is involved, just a 3D model of a human's upper body.

The closest work we found to our approach is given by Marmpena et al. They exploit the latent space of a VAE for generating beat gestures in a Pepper robot. In [17] they state that the latent space of the model exhibits topological features that can be used to modulate the amplitude of the motion, and they propose a structural feature that can be potentially useful for generating animations of specific arousal according to the dimensional theory of emotion [27]. This work is later extended on the one hand, by modifying the VAE into a Conditional Variational Autoencoder (CVAE), and on the other hand, by adding the sequences of eye LEDs patterns to the training data to increase the expressiveness of the animations [16]. Although they adopt a generative approach to beat gesture generation similar to what we do, they feed the learning mechanism with a manually built set of animations. They have no need to map recorded human motion into robot motion at the expense of a more limited and less natural gesture set to learn from.

In Rodriguez et al. [25] we used GANs to generate emotional gesticulation movements for the humanoid robot Pepper. Similarly to Marmpena et al., we used animations learned from a set of predefined talking gestures obtained from the robot's animation library, which later are modified by changing the head position, eyes' led color, arms motion velocity and speech intonation according to the sentiment of the speech. In a later work [36], we improved the beat gesture generation system by feeding the GAN with human talking gestures recorded by a Kinect. In the present work, we improve the mocap system to obtain more precise and richer human motion information and evaluate it using different quantitative measures, sustaining the obtained measures by questionnaire based human evaluation.

3 Experimental baseline

Just like a robot can learn to tighten a nut by demonstration it can also acquire a way to imitate the movements performed by the instructor. Thus, learning to gesticulate by observation should enhance robot naturalness.

In this section the approach employed for learning gestures from observation is described in detail. The proposed approach can be divided in a three-step process: observation, mapping and learning.

The observation step consists on capturing motion data while people talks. Afterwards, the data is filtered and mapped into the robot's motion space. Head, arms and hands features are mapped. Finally, the data is used in a learning step in order to be able to generate new but similar robot motion gestures. Each of these steps are more in deep explained in the consequent subsections (see Fig. 1).

3.1 Observation: motion capturing using OpenPose

Motion capture (MoCap) is the process of recording motion data through any type of sensor. Applications of such systems range from animation, bio-mechanics, medicine to sports, science, entertainment, robotics [39] of even study of animal behavior [29]. MoCap

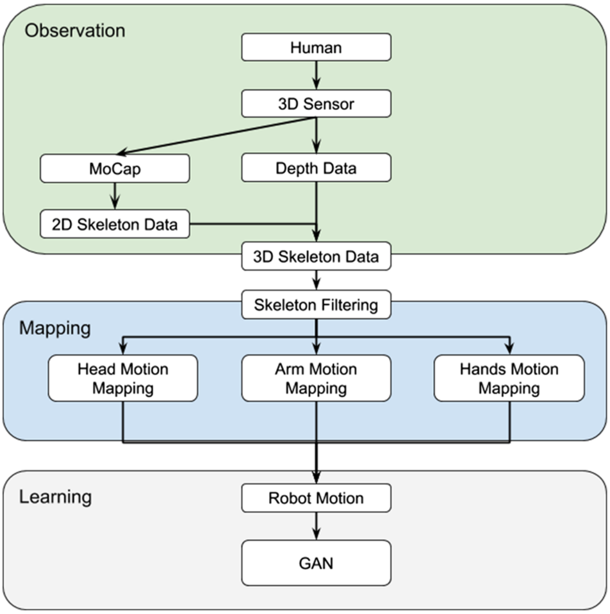
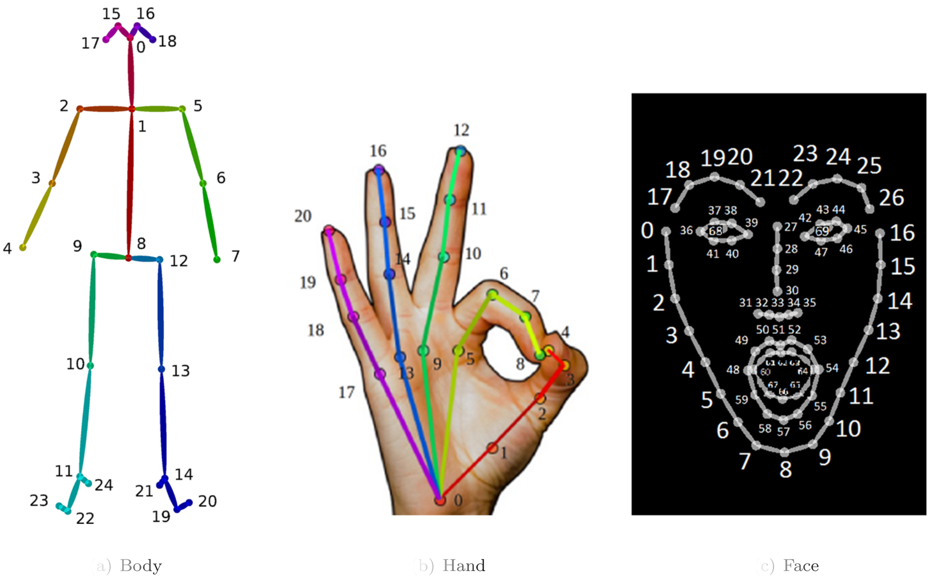


Fig. 1 Gestures learning process



a) Body

b) Hand

c) Face

Fig. 2 Openpose detected body, hand and face keypoints

systems rely on optical technologies, and can be marker-based (e.g. Vicon[1]) or markerless, like RGB-D cameras. While the former ones provide more accurate results, the latter ones are less prone to produce gaps (missing values) that need to be estimated [19, 32]. Besides, markerless approaches are welcomed by users that do not need to wear devices or clothes that might be cumbersome.

Trying to capture human poses as natural as possible, markerless technique seems to be more appropriate due to the freedom of movement of the speaker. Microsoft's Kinect depth sensor has been very popular as markerless skeleton tracker device due to its availability [1, 7]. The Kinect uses a machine learning method for human body pose recovery. Mainly, it infers body parts from depth images using a randomized decision forest learned from over a million training examples [30].

The advent of deep learning systems motivated the development of more powerful model-based approaches for human pose estimation. OpenPose [6] is one of the most popular bottom-up approaches for multi-person human pose estimation. It provides in real-time the body, hands, and facial keypoints (135 keypoints in total) of every person in the image. Figure 2 shows an example of the different parts detected. As most deep learning based techniques, it requires a powerful GPU and needs specific software dependencies in order to run efficiently. However, the face and hand information that provides turns it into a promising alternative for the task we are involved in.

In the proposed approach the technology employed for the observation process includes the Intel RealSense D435 [2] depth camera and OpenPose as MoCap system. The observation step consists of the process of extracting the necessary human motion information, in this

[1] https://www.vicon.com/

[2] https://www.intelrealsense.com/depth-camera-d435/

case the 3D skeleton according to the OpenPose BODY25 model and hand information. Considering OpenPose only estimates 2D human pose, our approach combines the output given by OpenPose and the depth data provided by the camera to obtain the 3D skeleton data of the user. Facial keypoints are not considered since Pepper facial expression is limited to coloured led eyes. Extracted data are low pass filtered before converting into robot motion, in order to stabilize them and avoid trembling.

3.2 Mapping: translating human motion to robot motion

Human motion cannot be directly mapped to a robot because in spite of the humanoid appearance their motion systems differ. Their joints have different degrees of freedom (DOF), movable ranges are not the same, etc. Therefore, original motions must be modified to be feasible by the robot, i.e the captured movements must be correctly mapped by satisfying several constraints (see [22] for a good overview of every aspect of the motion imitation task).

The mapping can be done by inverse kinematics, calculating the necessary joint positions given a desired end effector's pose [1]. Only the information of the end effectors is considered and employed to estimate appropriate positions of every joint. This process is effective but its complexity requires high computational load. Alternatively, direct kinematics adapts captured angles to the robot [36, 38]. As it is a more straightforward and computationally tractable method, this last technique has been used in the system presented here.

During the mapping stage only upper body joints (arms, head and hands) are considered since lower-body parts are not involved in beat gesticulation. The description of the terms used in the equations presented in the following subsections are detailed in Appendix. A to facilitate the readability and comprehension.

3.2.1 Arms mapping

The literature reveals different approaches to calculate the robot arm joint positions [12, 38]. This mapping process depends upon the robot's degrees of freedom and joints range. For the Pepper robot arms we are dealing with, some upper-body link vectors are built through the skeleton points in the human skeleton model, and joint angles are afterwards extracted from the calculation of the angles between those vectors (see Fig. 3). For the sake of simplicity, the formulae involved in that process are not going to be reproduced here (see [36] for more detailed information).

3.2.2 Head mapping

OpenPose detects basic face features such as the nose, the eyes and the ears (see Fig. 4). To map humans head position into the robot, we use the nose position as reference. Nose's position shifts horizontally when moving the head left to right and gets closer to or away from the neck when looking up or down. Thus, head's pitch (H_{ϕ}^{robot}) is proportional to the distance between the nose and the neck ($\overline{NN}$) joint. This distance is enough to adjust the neck's pitch angle (see (1)). The distance value must be converted from the camera frame to the robot head joint range. This is done by the *rangeConv* function. The $RangeHE_{\phi}^{robot}$ in (1) represents the range of values the head pitch can take. Instead, the yaw orientation of the head itself (HE_{ψ}^{robot}) can be calculated by measuring the angle between the vector

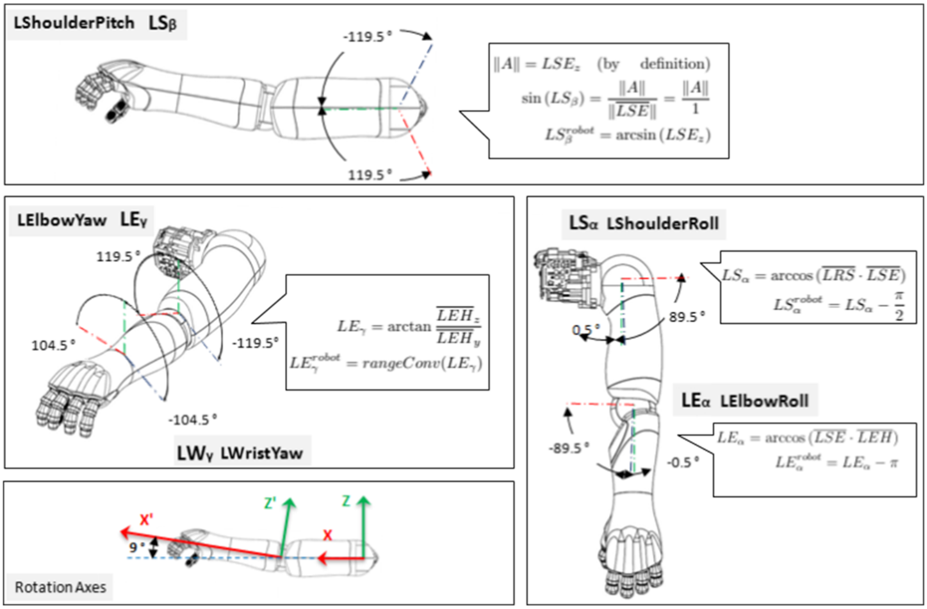

LShoulderPitch LSβ
-119.5°
119.5°
$\|A\| = LSE_z$ (by definition)
$\sin(LS_\beta) = \frac{\|A\|}{\|LSE\|} = \frac{\|A\|}{1}$
$LS_\beta^{robot} = \arcsin(LSE_z)$
LElbowYaw LEγ
119.5°
104.5°
-119.5°
-104.5°
$LE_\gamma = \arctan \frac{\overline{LEH_z}}{\overline{LEH_y}}$
$LE_\gamma^{robot} = rangeConv(LE_\gamma)$
LWγ LWristYaw
LSα LShoulderRoll
$LS_\alpha = \arccos(\overline{LRS} \cdot \overline{LSE})$
$LS_\alpha^{robot} = LS_\alpha - \frac{\pi}{2}$
89.5°
0.5°
LEα LElbowRoll
$LE_\alpha = \arccos(\overline{LSE} \cdot \overline{LEH})$
$LE_\alpha^{robot} = LE_\alpha - \pi$
-89.5°
-0.5°
Z'
Z
X'
X
9°
Rotation Axes

Fig. 3 Left arm joints and angle limits

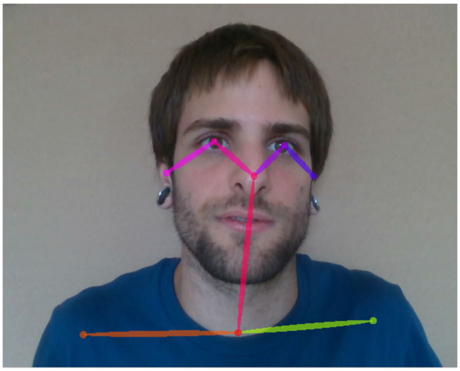
joining the nose and the neck, and the vertical axis as expressed in (2). Again, the measured angle must be converted to the tolerable robot head yaw range $RangeHE_\psi^{robot}$.

$$\begin{aligned} \overline{NN} &= dist(Nose, Neck) \\ HE_\phi^{robot} &= rangeConv(\overline{NN}, RangeHE_\phi^{robot}) \end{aligned} \tag{1}$$

$$HE_\psi^{robot} = rangeConv(-\arcsin(NN_x), RangeHE_\psi^{robot}) \tag{2}$$

Note that using the nose information instead of the more general head central pose allows for a more detailed realistic capture of the head motion.

Fig. 4 Face keypoints detected in OpenPose's BODY25 model

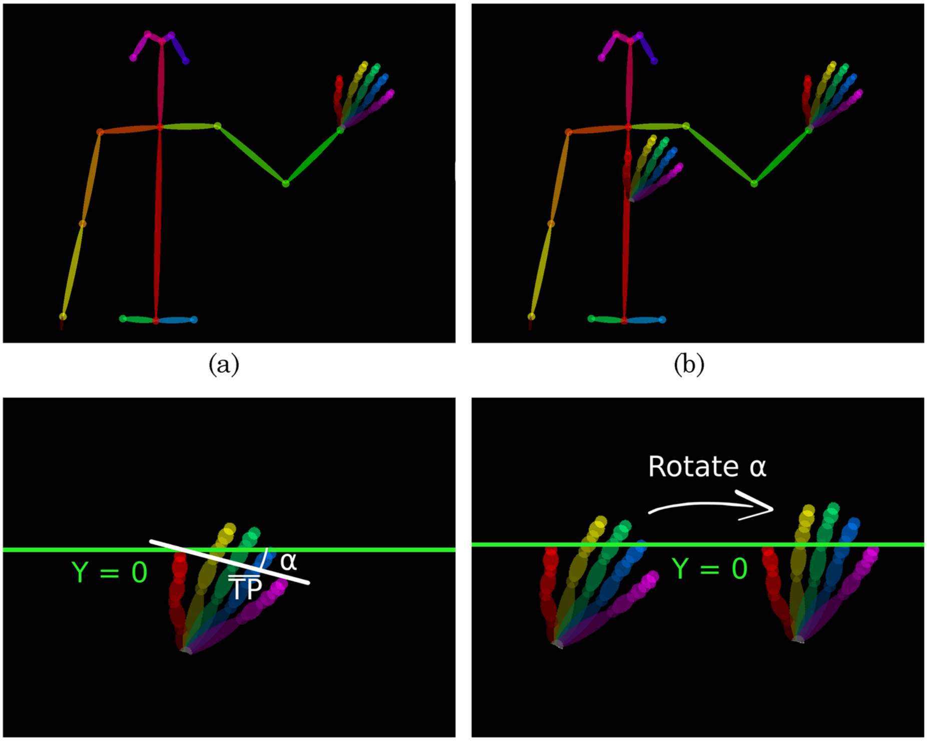


3.2.3 Hands mapping

OpenPose differentiates left and right sides without any calibration and gives 21 keypoints per hand, four per finger plus wrist. Next how hand movements are mapped is detailed. Note that the explanation focuses on the left hand, without any loose of generalization, due to the similarity of the right hand analysis.

1. First, it is determined whether a hand is showing the palm or the back. This requires to calculate the angle between the horizontal line and the line joining the thumb and the pinky fingertips. This is represented in (3), where FT stands for fingertip and OFT represents the new origin of a fingertip (see Fig. 5a and b). Afterwards, the fingers' points are rotated in such a way that the pinky lies at the right of the thumb, and both fingertips are aligned with $Y = 0$, as reflected in (4) (see Fig. 5c). For the right hand, at least two fingers should lie over that line to consider the palm is being showed as described in (5), the opposite condition for the left hand (see Fig. 5d).

$$\begin{aligned} &\forall i FT^i \quad OFT^i_{x,y} = FT^i_{x,y} - FT^{thumb}_{x,y} \\ &\alpha = \arctan(OFT^{pinky}_y, OFT^{pinky}_x) \end{aligned} \tag{3}$$



(a) (b) (c) (d)

Fig. 5 Hand side detection process for the right hand

$$\forall i \;\; FT_x^{\prime i} = OFT_x^i * \cos(-\alpha) - OFT_y^i * \sin(-\alpha)$$
$$\forall i \;\; FT_y^{\prime i} = OFT_x^i * \sin(-\alpha) - OFT_y^i * \cos(-\alpha) \tag{4}$$

$$HandSide = \begin{cases} Back & ((\sum_{i=1}^{3} FT_y^{\prime i}) > 0) \geq 2 \\ Palm & otherwise \end{cases} \tag{5}$$

2. Afterwards, the turn of the wrist, i.e hand's yaw angle (W_{ψ}^{robot}), is set correspondingly by measuring the distance between the thumb and the pinky fingertips ($\overline{TP}$) (6). The minimum and maximum values are adjusted according to the wrist's height so to avoid collisions with the touch screen on the chest of the robot. In this case, $RangeW_{\psi}^{robot}$ in (6) equation corresponds to the range of values the robot hand's yaw can take.

$$\overline{TP} = dist(FT^{\prime thumb}, FT^{\prime pinky})$$
$$W_{\psi}^{robot} = rangeConv(\overline{TP}, RangeW_{\psi}^{robot}) \tag{6}$$

3. Hand opening/closing (HA_{open}^{robot}) is also measured as a function of the distance between wrist and middle fingertip ($\overline{MW}$) (see (7)). The obtained $\overline{MW}$ value is then converted to the robot hand's opening/closing range ($RangeHA_{open}^{robot}$)

$$\overline{MW} = dist(FT^{middle}, Wrist)$$
$$HA_{open}^{robot} = rangeConv(\overline{MW}, RangeHA_{open}^{robot}) \tag{7}$$

3.3 Learning: GAN based gesture generation

GAN networks are composed by two different interconnected networks. The *Generator* (*G*) network generates possible candidates so that they are as similar as possible to the training set. The second network, known as *Discriminator* (*D*), judges the output of the first network to discriminate whether its input data are "real", namely equal to the input data set, or if they are "fake", that is, generated to trick with false data.

The training dataset given to the *D* network contained 2018 unit of movements (UM), being each UM a sequence of 4 consecutive poses, and each pose 14 float numbers corresponding to joint values of head, arms, wrists (yaw angle) and hands (finger opening value). These samples were recorded with the aforementioned OpenPose based MoCap system by registering 4 different persons talking, about 9 minutes overall.

The *D* network is thus trained using that data to learn its distribution space; its input dimension is 56. On the other hand, the *G* network is seeded through a random input with a uniform distribution in the range $[-1, 1]$ and with a dimension of 100. The *G* intends to produce as output gestures that belong to the real data distribution and that the *D* network would not be able to correctly pick out as generated. Figure 6 depicts the architecture of the generator and discriminator networks.

4 Evaluation of the robot's behaviour

In order to evaluate the whole system, we opted to compare some gestural features of the produced behavior with the one in [36]. Both systems differ in the MoCap (OpenNI vs. OpenPose) used to collect the training data. The OpenNI based approach showed difficulties to accurately track hands and head positions as it can be appreciated in Fig. 7.

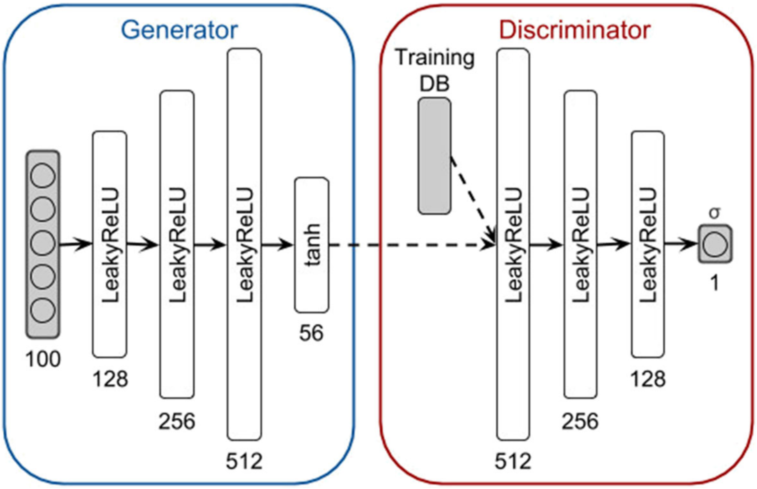
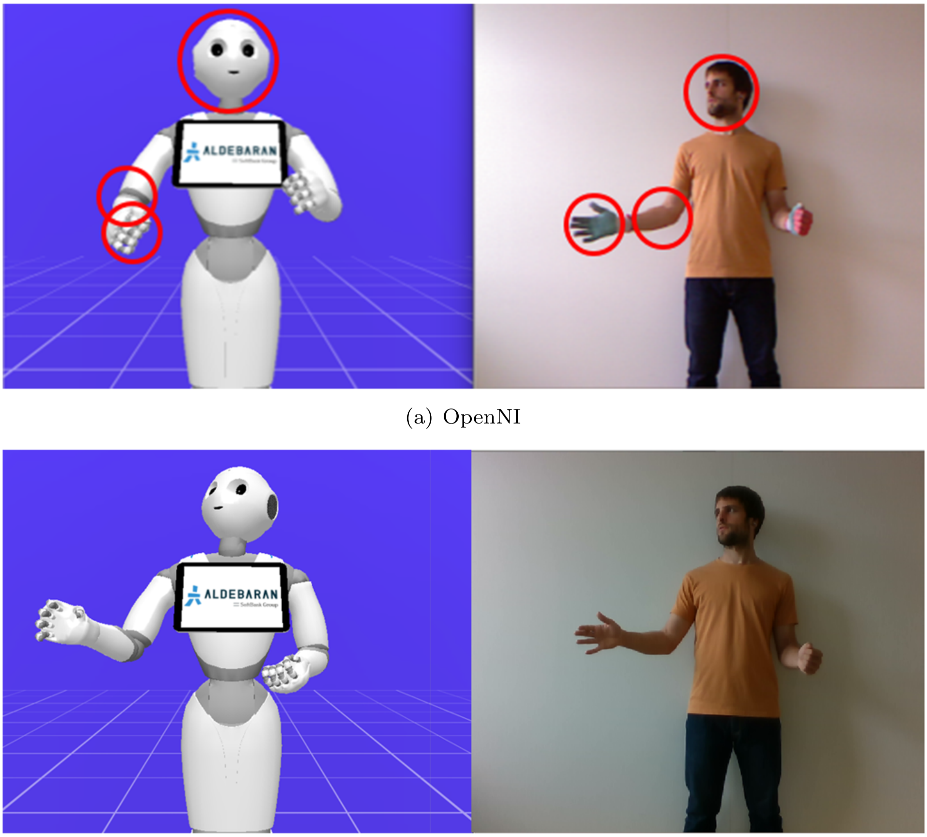


Fig. 6 GAN setup for talking gesture generation



(a) OpenNI

(b) OpenPose

Fig. 7 Reproduction of poses in the simulated robot

Table 1 Mean values for each measure (ϕ: pitch, ψ: yaw)

		OpenPose based	OpenNI based
Lhand	E_{jerk}	0.0369	0.0371
	E_{lpath}	29.6107	31.1931
Rhand	E_{jerk}	0.0304	0.0367
	E_{lpath}	25.3052	29.8871
Lelbow	E_{jerk}	0.0135	0.0162
	E_{lpath}	11.6434	13.4594
Relbow	E_{jerk}	0.0102	0.0161
	E_{lpath}	9.0550	13.9018
Head	E^{ψ}_{jerk}	0.0515	0.0478
	E^{ϕ}_{jerk}	0.0301	0.0248

These difficulties are therefore reflected in the generated gestures, as can be appreciated in this video[3]. The executions of both systems correspond to the models trained to generate movements using 4 consecutive poses as unit of movement. Notice that the temporal length of the audio intended to be pronounced by the robot determines the number of UM required to the generative model. Thus, the execution of those UMs, one after the other, defines the whole movement displayed by the robot.

In this first and naïve qualitative analysis three man differences are detected. On the one hand, head information provided by the OpenNI skeleton tracking package was not enough for preserving head movements and thus, the resulting motion was poor. On the other hand, the tracker only offered wrist positions and as a consequence, a vision based alternative was developed by segmenting red/green colors of the gloves wore by the speaker for tracking palms and backs of both hands. The opening/closing of the fingers was made at random for each generated movement. Lastly, the robot elbows tended to be too separated from the body and raised up. At a glance, it can be seen that the OpenPose based approach overcomes these three main drawbacks.

Further and more thorough qualitative analysis is presented in Section 4.2.

4.1 Quantitative analysis

There is no consensus in the field about which objective measures should be used to evaluate the quality of generated gestures. As a step towards common evaluation measures for the gesture generation field, we primarily use metrics found in the literature [13, 26, 35], namely the mean values of jerk and length path, and the Fréchet Gesture Distance (FGD):

– Norm of Jerk: as mentioned in the introduction, the goal is to generate spontaneous smooth movements. The norm of Jerk is a smoothness measure based on root mean

[3] https://www.youtube.com/watch?v=h9wpMEH8JQc

square (RMS) jerk quantification [4]. It is calculated according to (8), where *accel* stands for the acceleration at time t.

$$jerk = \frac{1}{T}\sum_{t=1}^{T} ||\dot{accel_t}|| \tag{8}$$

- Length of the generated paths: the length of the path ($lpath$) described by the positions of the hands during time ($\overline{x}_t$) is also another interesting measure. Lower $jerk$ values would lead to lower $lpath$ values, as the movements would be smoother. The measure ($lpath$) is computed as (9).

$$lpath = \sum_{t=2}^{T} ||\overline{x}_t - \overline{x}_{t-1}|| \tag{9}$$

- The Fréchet Gesture Distance (FGD) draws its inspiration from the Fréchet Inception Distance (FID) [10] commonly used in the image generation domain. The Inception model used in FID has been built from the predictions of a deep learning algorithm that has been trained with millions of labeled images which could belong to any of one thousand predefined classes. In inference mode the Inception model is presented an image, and then returns a list of one thousand probabilities. The probability the models assigns the image to belong to class C_i is the element P_i of the list. Our aim is to define a similar model for gestures, because to the best of our knowledge there is no

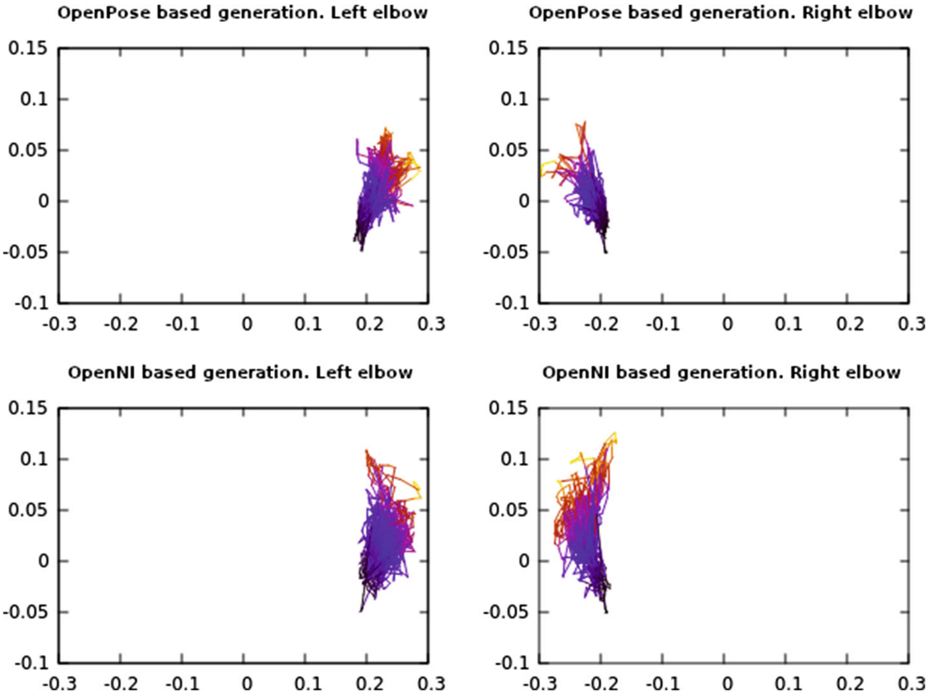


Fig. 8 Elbows' motion paths

Table 2 FGD values ($E \pm \sigma$)

	OpenPose GAN	OpenNI GAN
$E \pm \sigma$	0.1231±0.0108	0.1309± 0.0117

public model which could fill that role. Using Choregraphe,[4] a software designed to create robot animations, we have created a set of synthetic gestures, which in turn has been the data to build a Gaussian Mixture Model (GMM). The number of poses in this synthetic gestures database is a total of 1502. The need for a Choregraphe-based GMM arises from the desirable independence between such model and the data that will be analyzed. This assumption is met with the Inception model, and our goal is to meet it too in our approach. The data used for training the generative models is captured by a MoCap system, which is a different source altogether from the Choregraphe sofware, therefore ensuring the desired independence between both sets of data.

Previous work by the authors [27] supports the suitability of GMM to create the model. They show that, when used as generative model, GMMs rank second after GANs in the quality of generated gestures. As GMMs can also be used as classifiers, they can therefore return the set of probabilities needed to compute the FGD in the same manner that the Inception model does in FID.

Two different GANs trained in the same conditions have been used to generate robot motion for about 300 seconds and during the execution of the movements, the 3D coordinates of the end effectors (i.e. left and right hands) and elbows with respect to the pelvis have been recorded while talking. Head pitch and yaw values were also saved. Table 1 shows the obtained values for each approach.

Overall, OpenPose based motion generation obtains lower (and thus, more desirable) values for every measure. Hands and elbows show smoother motion. Figure 8 allows to compare elbows' motion paths during those 300 seconds. It is noticeable that OpenPose based gesture generator spread less both, left and right elbows' positions during those 300 seconds. This fact, together with the obtained motion measures confirm the strange elbow poses produced by the OpenNI based system and identified in the above referred video.

Head jerk is the exception though. There is an explanation for this outlier. In order to be able to capture head pose using the OpenNI based approach, the talking human needed to exaggerate the head movements so that the mapped pose was perceptible on the robot. This drawback can be appreciated during the replication of the movements of the second actress in the simulated robot in this video[5] (timestamp 0:36 to 0:44s). Small human head movements cannot be captured with fidelity using the OpenNI based approach. As a result, the generated head motion is barely noticeable and hence, produces smaller jerk values.

Table 2 shows the FGD distance values for the two GAN models aforementioned. The motivation behind FGD is to make it fulfill the same role in gestures than FID in images, i.e. measure the distance between the original and the generated movements. Therefore, the smaller the values of FGD, the more similar to the original gestures. Note that OpenPose results in a smaller FGD.

[4] http://doc.aldebaran.com/2-5/software/choregraphe/index.html

[5] https://www.youtube.com/watch?v=iW1566ozbdg

Table 3 Mean values based on five-point Likert scale rating for each gesticulation type

	OpenPose GAN	OpenNI GAN	Wilcoxon p-value
Naturalness	3.72 ± 0.76	3.36 ± 0.80	0.013
Fluency	3.98 ± 0.71	3.47 ± 0.91	0.0007
Appropriateness	3.52 ± 0.89	3.27 ± 0.95	0.12
Variability	3.83 ± 0.79	3.19 ± 0.91	7.71e-5
Synchronization	3.56 ± 0.79	3.49 ± 0.83	0.41
Liking	3.78 ± 0.82	3.44 ± 0.87	0.042

4.2 Qualitative analysis

While there are several successfully approaches on how gestures are produced by social robots, problems arise when it comes to evaluate the behavior of the robot. Usually robot behavior is qualitatively evaluated, and the most common tool used for qualitatively measure the behavior of robots is the questionnaire. Often questionnaires are defined so that participants can rank several aspects of the robot's performance [23, 28]. There seems to be a consensus in presenting the questions using Likert scale and analyzing the obtained responses using some statistical test like analysis of variance, chi-square and so on.

The questionnaire for the study rates the following aspects in a five point Likert scale: the *naturalness* of the gestures, the *fluency*, the *appropriateness* of the gestures for accompanying the speech, the *variability* of gestures perceived, the *synchronization* between the speech and the gestures, and *how much they liked* the gestures performed by the robot.

We conducted an online survey showing two videos[6,7] about 1 minute long each (equivalent to 240 generated movements) to evaluate the perceived quality of the generated movements of both systems.

In order to avoid possible option ordering bias, we prepared two questionnaires with different video ordering and we randomly showed one of them to the participants. 59 volunteers (22 female, 37 male) with an average age of 32 participated in our user study to judge the robot's gesticulation capabilities. Table 3 illustrates the results.

Overall, mean score results are higher for the GAN trained with OpenPose as skeleton tracker. Fluency and variability are the best valued properties and where the differences between both systems differ the most. OpenPose allows for more variable and fluent gesture generation.

However, this vague numerical comparison might be non statistically significant due to the apparently large variances. To find if the qualitative perception of both systems shows a statistically significant difference, we applied the Wilcoxon signed-rank test and the obtained p-values for each evaluated feature are shown in the last column of Table 3. As we are testing six hypotheses simultaneously, it is advisable to apply the Bonferroni correction. Therefore, for the standard α value of 0.05 for each individual hypothesis to be achieved, the threshold value has to be replaced by $\alpha = 0.05/6 = 0.0083$. Again, fluency and variability pass the test and naturalness shows a close p-value $(0.083 < 0.013 < 0.05)$. Appropriateness, Synchronization and Liking, didn't show significant differences, but they all obtained higher mean ratings. It is worth recalling that both are GAN based gesture generation systems that differ only in the MoCAP used to capture the training data and thus, very similar

[6] OpenNI: https://youtu.be/O7q2neEeA9s

[7] OpenPose: https://youtu.be/-ubxvZ2gxtI

in nature. However, these results together with the quantitative analysis results previously showed allow us to conclude that OpenPose tool introduces significant improvements in the generated behavior.

5 Conclusions and further work

In this paper a gesture generating system from human observation has been presented and tested. It has been shown that while a markerless motion based system is more computationally demanding than other methods, the results justify its use, due to the high capture accuracy and comfort for the user.

Markerless motion capture is preferable in order not to constraint the speakers during the motion recording process. It allows to the human to behave naturally due to the non-intrusive nature of the recording process. This fact is not petty since the goal is to generate human like natural robot gestures. Beyond this, OpenPose captures motion more faithfully, i.e. the skeleton obtained better gathers the pose of the human model. The work presented here intended to evaluate if the computational demands are justified by producing data that allows a more natural, credible and non jerky gesturing system. Different measures, quantitative and qualitatives confirm that the OpenPose-GAN based gesture generation improves OpenNI-GAN system. OpenPose based approach induces better replication and, as a consequence, better motion generation as it has been empirically measured.

A more expressive database would broaden the variety of gestures being generated by the robot. Before exposing the robot to a large audience, a wider set of human models need to be recorded. It is of high interest to compare the capability of GAN to generate gestures with another generative approach such as VAEs. Training different learning mechanisms with the same data would allow us to see if there are significant differences among the generated gestures or to choose the most appropriate model for the task in hands. However, the most interesting point in our opinion is that the developed system offers the basis for extending the beats to a generator conditioned on other kind of movements as emotion-based or context related gestures.

Appendix A

Table 4 Description of the terms used in (1)–(7)

Term	Description
$\overline{NN} = (NN_x, NN_y)$	The euclidean distance between the nose and head keypoints detected in OpenPose's BODY25 model.
$RangeHE_{\phi}^{robot}$	Pepper's head pitch joint range values in radians.
$RangeHE_{\psi}^{robot}$	Pepper's head yaw joint range values in radians.
HE_{ϕ}^{robot}	Pepper's head pitch angle in radians obtained after the mapping process.
HE_{ψ}^{robot}	Pepper's head yaw angle in radians obtained after the mapping process.
$FT_{x,y}^{i}$	X and Y components of the fingertip keypoints detected in OpenPose's COCO Hand model, where $i = \{thumb, index, middle, ring, pinky\}$.
$OFT_{x,y}^{i}$	X and Y components of the fingertip keypoints detected in OpenPose's COCO Hand model after the translation to the new origin, where $i = \{thumb, index, middle, ring, pinky\}$.

Table 4 (continued)

Term	Description
FT'^{i}_{y}	X and Y components of the fingertip keypoints rotated in such a way that the pinky lies at the right of the thumb, and both fingertips are aligned with $Y = 0$. $i = \{thumb, index, middle, ring, pinky\}$
$\overline{TP}$	The euclidean distance between the thumb and the pinky fingertips.
$RangeW^{robot}_{\psi}$	Pepper's wrist yaw joint range values in radians.
W^{robot}_{ψ}	Pepper's wrist yaw angle in radians obtained after the mapping process.
$Wrist$	Wrist keypoint detected in OpenPose's COCO model.
$\overline{MW}$	The euclidean distance between the middle fingertip and wrist keypoints detected in OpenPose's COCO model.
$RangeW^{robot}_{open}$	Pepper's wrist yaw joint range values in radians.
HA^{robot}_{open}	Pepper's hand opening/closing actuator range values.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

1. Alibeigi M, Rabiee S, Ahmadabadi MN (2017) Inverse kinematics based human mimicking system using skeletal tracking technology. J Intell Robot Syst 85(1):27–45
2. Breazeal C (2004) Designing sociable robots. intelligent robotics and autonomous agents. MIT Press, Cambridge
3. Bremner P, Pipe AG, Fraser M, Subramanian S, Melhuish C (2009) Beat gesture generation rules for human-robot interaction. In: RO-MAN 2009 - The 18th IEEE international symposium on robot and human interactive communication, pp 1029–1034. https://doi.org/10.1109/ROMAN.2009.5326136
4. Calinon S, D'halluin F, Sauser EL, Cakdwell DG, Billard AG (2004) Learning and reproduction of gestures by imitation. In: International conference on intelligent robots and systems, pp 2769–2774
5. Cao Z, Hidalgo Martinez G, Simon T, Wei S, Sheikh YA (2019) OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. IEEE Trans Pattern Anal Mach Intell 1–1. https://doi.org/10.1109/TPAMI.2019.2929257
6. Cao Z, Simon T, Wei SE, Sheikh Y (2017) Realtime multi-person 2D pose estimation using part affinity fields. In: CVPR
7. Fadli H, Machbub C, Hidayat E (2015) Human gesture imitation on NAO humanoid robot using Kinect based on inverse kinematics method. In: International conference on advanced mechatronics, intelligent manufacture, and industrial automation (ICAMIMIA). IEEE
8. Fernández-Baena A., Montaño R., Antonijoan M, Roversi A, Miralles D, Alías F (2014) Gesture synthesis adapted to speech emphasis. Speech Commun 57:331–350
9. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
10. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) GANS trained by a two time-scale update rule converge to a local Nash equilibrium. In: Advances in neural information processing systems, pp 6626–6637
11. Kingma DP, Welling M (2013) Auto-encoding variational Bayes. arXiv:1312.6114

12. Kofinas N, Orfanoudakis E, Lagoudakis MG (2015) Complete analytical forward and inverse kinematics for the NAO humanoid robot. J Intell Robot Syst 77(2):251–264. https://doi.org/10.1007/s10846-013-0015-4
13. Kucherenko T, Hasegawa D, Henter GE (2019) Analyzing input and output representations for speech-driven gesture generation. In: 19Th international ACM conference on intelligent virtual agents (IVA), pp 97–104. https://doi.org/10.1145/3308532.3329472
14. Kwon J, Park FC (2006) Using hidden Markov models to generate natural humanoid movement. In: International conference on intelligent robots and systems (IROS). IEEE/RSJ
15. Manfrè A., Infantino I, Vella F, Gaglio S (2016) An automatic system for humanoid dance creation. Biologic Insp Cognit Architect 15:1–9
16. Marmpena M, Garcia F, Lim A (2020) Generating robotic emotional body language of targeted valence and arousal with conditional variational autoencoders. In: Companion of the 2020 ACM/IEEE international conference on human-robot interaction, pp 357–359
17. Marmpena M, Lim A, Dahl TS, Hemion N (2019) Generating robotic emotional body language with variational autoencoders. In: 2019 8Th international conference on affective computing and intelligent interaction (ACII). IEEE, pp 545–551
18. McNeill D (1992) Hand and mind: What gestures reveal about thought. University of Chicago Press, Chicago
19. Mehta D, Sridhar S, Sotnychenko O, Rhodin H, Shafiei M, Seidel HP, Xu W, Casas D, Theobalt C (2017) VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. ACM Trans Graph 36(4):44:1–44:14
20. Ng-Thow-Hing V, Luo P, Okita S (2010) Synchronized gesture and speech production for humanoid robots. In: 2010 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 4617–4624
21. Nishimura Y, Nakamura Y, Ishiguro H (2020) Long-term motion generation for interactive humanoid robots using GAN with convolutional network. In: Companion of the 2020 ACM/IEEE international conference on human-robot interaction, pp 375–377
22. Poubel LP (2013) Whole-body online human motion imitation by a humanoid robot using task specification. Master's thesis, Ecole Centrale de Nantes–Warsaw University of Technology
23. Pérez-Mayos L, Farrús M, Adell J (2019) Part-of-speech and prosody-based approaches for robot speech and gesture synchronization. J Intell Robot Syst. https://doi.org/10.1007/s10846-019-01100-3
24. Rodriguez I, Astigarraga A, Ruiz T, Lazkano E (2016) Singing minstrel robots, a means for improving social behaviors. In: IEEE International conference on robotics and automation (ICRA), pp 2902–2907
25. Rodriguez I, Manfré A., Vella F, Infantino I, Lazkano E FPR García Olaya Á, Sesmero Lorente MP, Iglesias Martínez JA, Ledezma Espino A (eds) (2019) Talking with sentiment: Adaptive expression generation behavior for social robots. Springer International Publishing, Cham
26. Rodriguez I, Martínez-Otzeta J. M., Irigoien I, Lazkano E (2019) Spontaneous talking gestures using generative adversarial networks. Robot Auton Syst 114:57–65
27. Russell JA (1980) A circumplex model of affect. J Pers Soc Psychol 39(6):1161
28. Sakai K, Minato T, Ishi CT, Ishiguro H (2017) Novel speech motion generation by modeling dynamics of human speech production. Front Robot AI 4:49. https://doi.org/10.3389/frobt.2017.00049
29. Schubert T, Eggensperger K, Gkogkidis A, Hutter F, Ball T, Burgard W (2016) Automatic bone parameter estimation for skeleton tracking in optical motion capture. In: International conference on robotics and automation (ICRA). IEEE
30. Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: Proceedings of the 2011 IEEE conference on computer vision and pattern recognition, CVPR '11. IEEE Computer Society, USA, pp 1297–1304. https://doi.org/10.1109/CVPR.2011.5995316
31. Tanwani AK (2018) Generative models for learning robot manipulation. Ph.D. thesis École Polytechnique fédéral de Laussane (EPFL)
32. Tits M, Tilmanne J, Dutoit T (2018) Robust and automatic motion-capture data recovery using soft skeleton constraints and model averaging. PLOS ONE 13(7):1–21
33. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemomet Intell Laborat Syst 2(1-3):37–52
34. Wolfert P, Kucherenko T, Kjellström H, Belpaeme T (2019) Should beat gestures be learned or designed?: a benchmarking user study. In: ICDL-EPIROB: Workshop On naturalistic non-verbal and affective human-robot interactions
35. Zabala U, Rodriguez I, Martínez-Otzeta JM, Irigoien I, Lazkano E (2020) Quantitative analysis of robot gesticulation behavior. Autono Robot 1–15

36. Zabala U, Rodriguez I, Martínez-Otzeta JM, Lazkano E (2019) Learning to gesticulate by observation using a deep generative approach. In: 11Th international conference on social robotics (ICSR). Springer, pp 666–675
37. Zabala U, Rodriguez I, Martínez-Otzeta JM, Lazkano E (2020) Can a social robot learn to gesticulate just by observing humans? In: Workshop of physical agents. Springer, pp 137–150
38. Zhang Z, Niu Y, Kong LD, Lin S, Wang H (2019) A real-time upper-body robot imitation system. Int J Robot Cont 2:49–56. https://doi.org/10.5430/ijrc.v2n1p49
39. Zhang Z, Niu Y, Yan Z, Lin S (2018) Real-time whole-body imitation by humanoid robots and task-oriented teleoperation using an analytical mapping method and quantitative evaluation. Appl Sci 8(10). https://www.mdpi.com/2076-3417/8/10/2005

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.