

Generation of expressive motions for a tabletop robot interpolating from hand-made animations

Gonzalo Mier¹, Fernando Caballero¹, Keisuke Nakamura², Luis Merino¹ and Randy Gomez²

Abstract—Motion is an important modality for human-robot interaction. Besides a fundamental component to carry out tasks, through motion a robot can express intentions and expressions as well. In this paper, we focus on a tabletop robot in which motion, among other modalities, is used to convey expressions. The robot incorporates a set of pre-programmed motion animations that show different expressions with various intensities. These have been created by designers with expertise in animation. The objective in the paper is to analyze if these examples can be used as demonstrations, and combined by the robot to generate additional richer expressions. Challenges are the representation space used, and the scarce number of examples. The paper compares three different learning from demonstration approaches for the task at hand. A user study is presented to evaluate the resultant new expressive motions automatically generated by combining previous demonstrations.

I. INTRODUCTION

Natural and multimodal interaction is one of the key components of social robots [1], [2]. Among the different modalities that can be used in these interactions, motion is very important and what distinguishes a robot from other interactive virtual agents.

Many of the social robots available incorporate a set of predefined animated motions. For instance, the Choregraphe software [3] used to program Pepper or Nao includes different pre-programmed animations. Here we consider the research robot Haru [4], which also incorporates a set of animations created by designers to convey expressions like agreement or disagreement, happiness, sadness or shyness. These animations can be considered as discrete open-loop macro actions that can be used to build complex interactive behaviors.

In this work we consider the question if such predefined animations can be combined so to generate automatically expressions like, for instance, a shy disagreement. Therefore, the question to address is to determine if it is possible to combine motions so that the resultant combinations are legible and express the desired combined expressions.

Several challenges need to be considered. There is the question on how the motion trajectories should be interpolated. The manifold of motions representing a particular expression is complex and direct interpolation will likely lead to non-meaningful motions. Learning approaches can be applied to extract the relevant information, but only a

handful of animations is available, so the methods should be non-data intensive.

In this work, we analyze three different techniques from the literature to interpolate animated motions using the designed animations as references. The techniques differ on the way the provided trajectories are used to extract relevant information and to generate the desired combined expression. We then perform a user study to evaluate the expressiveness and legibility of the resultant combined motion for the different methods.

The paper is structured as follows. Next section reviews the state of the art. Then, Section III presents the robot employed in the paper, including the motion degrees of freedom that can be used to convey expressions. The methods considered for the combination of expressions are presented next. Section V describes the evaluation methodology and it is followed by the results. The paper ends with a discussion and outlook.

II. STATE OF THE ART

The generation of expressive motion is a relevant topic in animation [5], [6] and also robotics [7], [8], [9], [10], [11]. The authors of [9] analyze how to move the head of an Android Robot while speaking. In [11] the authors consider how to express emotions using a robot swarm. Others use robotics arms to make other people understand their intentions with the movement [7], [12], or to express basic states as Pleasure, Arousal and Dominance using different controllers selected by Fuzzy-logic [10]. The work in [8] presents techniques to express different motion styles with a robotic arm while performing tasks.

Most approaches employ Learning from Demonstration (LfD) techniques [13], [14] to teach robots/animated characters to produce such expressive motion. For instance, [5], [6] learn how to imitate the style of people walking from demonstrations by using Inverse Reinforcement Learning (IRL) [15]. IRL is also used in [8] to transfer motion styles to the robotic arm by learning cost functions. Other approaches encode the data from the demonstrations as a Gaussian Mixture Model (GMM) [16], [17]. Then, Gaussian Mixture Regression (GMR) [18] is used to generate the new motions. In general, most of these approaches make use of hundreds or even thousands of trajectory examples in order to properly learn the behaviours, considering that each demonstration has to be conducted by at least one human, training data are difficult and expensive to produce, and therefore scarce. Thus, this paper focuses in the development of techniques

¹G. Mier, F. Caballero and L. Merino are with the Service Robotics Laboratory, School of Engineering, Universidad Pablo de Olavide, Crta. Utrera km 1, Seville, Spain gmiemuo@upo.es, fcaballero@upo.es, lmercab@upo.es

²K. Nakamura and R. Gomez are with Honda Research Institute, Wako, Saitama, Japan



Fig. 1: Haru is a tabletop robot for research on social robotics. Haru has 5 DoF: base rotation, neck tilt, eyes tilt, eyes roll and eyes stroke. The eyes roll moves synchronously in opposite direction.

able to perform LfD from little data [19], decreasing the need of examples from hundreds to dozens.

In this work, we evaluate several LfD approaches for the generation of expressive motions based on demonstrations provided by animators. As acceleration patterns of movement actually convey emotions [20], we will model not only the motions but also the dynamics by considering time constraints into the learning process. In particular, we consider the use of Task Parameterized Gaussian Mixture Model (TPGMM) [16], the interpolation between encoded GMM motions in the manifold of the Symmetric Positive Definite (SPD) matrices [17], and a supervised learning approach.

III. HARU: A SOCIAL TABLETOP ROBOT

While the approaches presented below for robot motion interpolation are general and easily adaptable to many platforms, this section describes the social robot used for experimentation in order to better understand the methods.

This paper makes use of the Haru robot [4], a new tabletop robot with 5 degrees of freedom (see Fig. 1). This is a platform for research on social robot interaction. It includes speakers to reproduce sounds, microphones, LCD screens in the eyes to reproduce videos on them, and LEDs for the eyebrows and the mouth. There are 5 motors that control base rotation, neck tilt, eyes tilt, eyes roll and eyes stroke.

Haru provides a set of software tools to create/modify animations, and also a simulator to check such animations before transferring them to the actual robot. We call animation to the combination of one or more robot actuation or interaction modalities in a predefined period.

In this work we limit the animations to motor movements purely. Videos and sound produce expressions easier and more recognizable than movement. So we center our work only in the interpolation of motor movements in the articular space. To test those interpolations, all LEDs are turned off, no sound is reproduced and the eyes videos are neutral green eyes that blink every few seconds in loop. This way, we can better evaluate the impact and quality of the produced animations.

Expression	Intensity	Pos_x	Pos_y
Happy	3	0	0.3
Happy	7	0	0.7
Shy	3	0.3	0
Shy	7	0.7	0
Sad	3	0	-0.3
Sad	7	0	-0.7
Agree	3	0.2121	0.2121
Agree	7	0.495	0.495
Disagree	3	0.2121	-0.2121
Disagree	7	0.495	-0.495

TABLE I: Table of the expressions, their intensities and the values of the codification that we are using to classify them in the Plutchik's wheel.

IV. ROBOT ANIMATION INTERPOLATION IN THE ARTICULAR SPACE

The aim of this paper is to automatically generate animations that represent one or more expressions from a few handmade samples. We use 10 samples provided by human designers as training data, each conveying an expression with a different intensity. These expressions are "Happiness", "Shyness", "Sadness", "Agreement" and "Disagreement". Animations are rated according to its intensity, a value between 1 and 10. The larger the intensity, the stronger will be the animation. The sample animations used in this paper have intensity 3 and 7.

Our objective is to automatically generate animations that express the expressions "Happy-Agree" (Happiness and Agreement), "Shy-Agree" (Shyness and Agreement), "Shy-Disagree" (Shyness and Disagreement) and "Sad-Disagree" (Sadness and Disagreement).

All interpolations will be made at the intermediate point between the two animations, in order to ease their later recognition, so the interpolation "Shy-Agree" is equivalent to "Agree-Shy". In this work we will only use interpolations between animations with the same intensity to generate animations that express other expressions, but the interpolation can be done in an equivalent way to obtain other expressions with different intensities.

Each animation consists of the continuous time evolution of the 5 degrees of freedom, and it is labeled with the expression and its intensity. We encode these two later variables as continuous values by mapping them to the "Plutchik's wheel of emotion" model [21] (see Fig. 2), which is extremely used in psychology to classify the human emotions. Thus, each animation has two parameters associated, which are the position of the expression that tries to reproduce in the Plutchik's wheel, in " x " and " y ". According to Plutchik's wheel, expressions with higher intensity should be near $[0, 0]$, but we decide that $[0, 0]$ is a neutral expression, and the unit circle contains the expressions with higher intensity (Table I). While this model is applied for humans, it gives an idea of "closeness" between expressions.

¹Source: <http://www.copyright.com/blog/your-fragile-emotions-illustrated>. On Date: 22/10/2015.

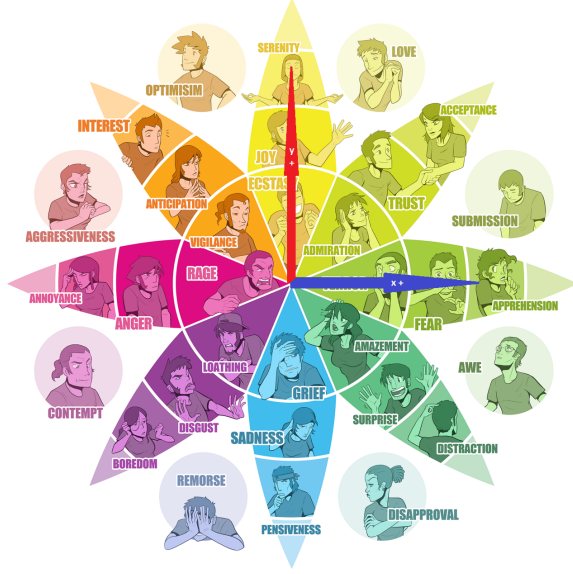


Fig. 2: Plutchik's wheel of emotions. Image extracted from the web¹.

The objective of the interpolation is to generate the trajectories of the 5 joints of the robot given as inputs the desired expression and intensity (using its position in the Plutchik's wheel) and a vector of time instants for the trajectory. Next we present the methods considered.

We select three methods to do the interpolation: the first one, the TPGMM [22], uses a model to map the expressions and a data augmentation technique; the second one, the Binary Regression Decision Tree (BRDT) [23], also uses the map of expressions, but not the data augmentation; the last one, the Geodesic interpolation proposed by Jaquier and Calinon [17], do not use the map, neither the data augmentation. These methods are specially suited for working with small datasets. The TPGMM and Geodesic interpolation exploit the structure of the information in order to decrease the required amount of examples, while the proposed BRDT is simple enough to guarantee good generalization even with scarce data.

A. Task Parameter Gaussian Mixture Model

The first method we consider is the TPGMM [22]. This method employs a Gaussian Mixture Model (GMM) to learn a representation of all the training animations.

In the GMM, we encode the training data as the joint distribution $P(\xi^I, \xi^O) \sim \sum_{i=1}^K \pi_i \mathcal{N}(\mu_i, \Sigma_i)$, being ξ^I the inputs (a vector of time instants and the expression position in the wheel), ξ^O the outputs (the 5 values of the actuators for each of the time instants) and π_i the weights of the different modes in the GMM (each with its mean μ_i and covariance Σ_i). We use the notation of [22]:

$$\xi = \begin{bmatrix} \xi^I \\ \xi^O \end{bmatrix}, \mu_i = \begin{bmatrix} \mu_i^I \\ \mu_i^O \end{bmatrix}, \Sigma_i = \begin{bmatrix} \Sigma_i^I & \Sigma_i^{IO} \\ \Sigma_i^{OI} & \Sigma_i^O \end{bmatrix} \quad (1)$$

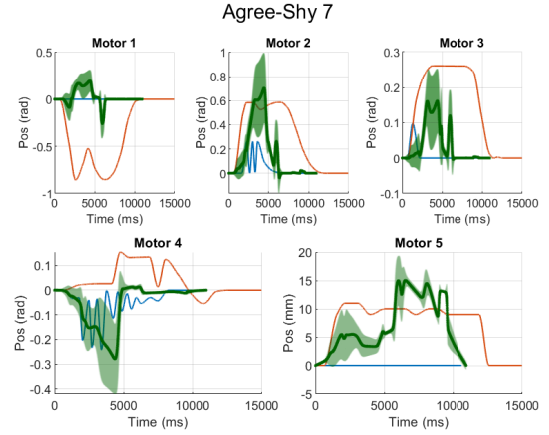


Fig. 3: Shy on red, Agree on blue and the interpolation of both, Agree-Shy, on green. On bright green are the GMM generated from the GMR.

Once the GMM is encoded, we can use it to interpolate new outputs given an input by conditioning on the inputs (Gaussian Mixture Regression -GMR-). In this case conditioning on the desired expression and intensity, and the time instants for the trajectory. The conditional distribution $P(\xi^O|\xi^I)$ is given by:

$$P(\xi^O|\xi^I) \sim \sum_{i=1}^K h_i(\xi^I) \mathcal{N}(\hat{\mu}_i^O(\xi^I), \hat{\Sigma}_i^O) \quad (2)$$

with

$$\hat{\mu}_i^O(\xi^I) = \mu_i^O + \Sigma_i^{OI} \Sigma_i^{II-1} (\xi^I - \mu_i^I) \quad (3)$$

$$\hat{\Sigma}_i^O = \Sigma_i^O - \Sigma_i^{OI} \Sigma_i^{II-1} \Sigma_i^{IO} \quad (4)$$

$$h_i(\xi^I) = \frac{\pi_i \mathcal{N}(\xi^I | \mu_i^I, \Sigma_i^I)}{\sum_{k=1}^K \pi_k \mathcal{N}(\xi^I | \mu_k^I, \Sigma_k^I)} \quad (5)$$

Regarding the parameters of the TPGMM, two frames are used to learn each animation, the initial and the end point. All the animations start and end in the origin. The TPGMM initialization we use to train is the equal bins splitting [22].

To interpolate using this method, we use polar coordinates on the wheel to find the middle point between the two animations that we are trying to interpolate. For instance, if we want to interpolate the animation that correspond with the expression Agree-Shy with intensity 7, we compute the average between the polar coordinates position of Agree with intensity 7 ($[0.7, \frac{\pi}{4}]$) and Shy with intensity 7 ($[0.7, 0]$), and return that value ($[0.7, \frac{\pi}{8}]$) to the Cartesian coordinate space again ($[0.6467, 0.2679]$). This is then used as inputs (with the time instants) to the GMM interpolation. The interpolated trajectory of "Agree-Shy 7" and the two hand-made "Agree 7" and "Shy 7" are represented in Fig. 3.

B. Binary Regression Decision Tree

The second method considered is a Binary Regression Decision Tree (BRDT) [23]. The BRDT is employed here to learn a direct mapping from expressions and their intensities

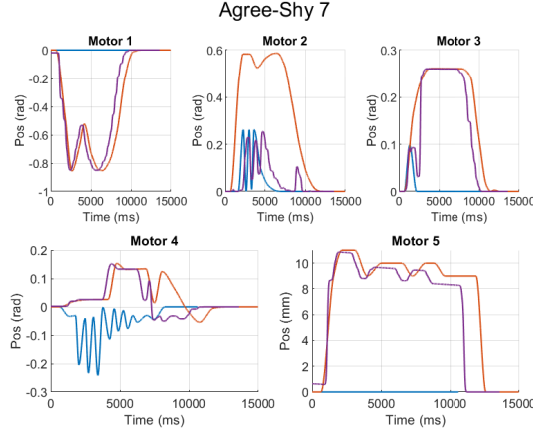


Fig. 4: Shy on red, Agree on blue and the interpolation of both, Agree-Shy, on purple, generated using the BRDT interpolation.

(using the Plutchik’s Wheel-based encoding as before) and time steps to motor commands at the given time steps. BRDT is a Classification And Regression Tree (CART) algorithm, which uses the recursive partition of the input to learn the model, and predict from it².

The interpolated trajectory of “Agree-Shy 7” and the two hand-made “Agree 7” and “Shy 7” using this method is shown in Fig. 4.

C. Geodesic interpolation

The two previous methods use the Plutchik’s Wheel as the model to encode the variables representing expressions and their intensities. This assumes that the model works as well for people and for robots, which has not been tested. To avoid using any model, we consider a third case in which we directly interpolate the trajectories of the hand-made animations that we want to mix.

We only have one animation example for each expression and intensity. In order to allow for some variability, we encode the trajectories of those animations into Gaussian Mixture Models (GMM), one for each animation. The variables that we use to encode the GMM are the time and the joint position of each motor (6 variables).

As the GMMs representing the trajectories related to expressions lie in a complex manifold, we use the Geodesic interpolation proposed by Jaquier and Calinon [17]. The method allows interpolating between the covariances of the GMMs in the Symmetric Positive Definite (SPD) manifold as 4th-order tensors.

By applying directly the method to our data, the interpolation result is not the same in both directions (for instance, between “Agree-Shy” and “Shy-Agree”), as we can see in Fig. 5 for the middle point of the interpolation. Because of this, we modify this method to obtain the same interpolated trajectory when commuting the input trajectories.

Geodesic interpolation of Shy-Agree 7 and Agree-Shy 7

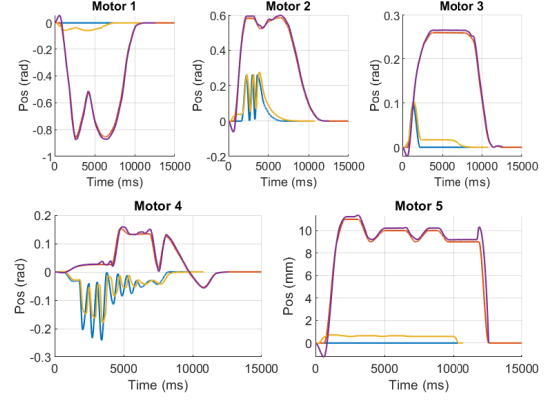


Fig. 5: Geodesic Interpolation between two trajectories Shy (in red) and Agree (in blue). With $w = 0.5$, in purple, the interpolation of Shy to Agree, and in yellow, the interpolation of Agree to Shy. In this interpolation, we have not modified the interpolation of Jaquier and Calinon [17].

To change this, we propose to add to the diagonal of the covariance matrix of each GMM a regularization term. This parameter was used by Jaquier and Calinon [17] to avoid that the covariance matrix is not invertible, but the value is small ($= 10^{-5}$). In this work, we notice even small changes (from 10^{-5} to 10^{-6} or 10^{-4}) of that parameter provokes a relevant modification in the interpolation. Due to that, we optimize the regularization term and number of Gaussians $n_{gauss_{Geod}}$ that are employed in the method³ that make the interpolations generated between the first (A) and the second (B) trajectory the same as the interpolations between the second (B) and the first (A) one, as:

$$traj_{(A \rightarrow B) * \omega_A} = traj_{(B \rightarrow A) * (1 - \omega_A)} = traj_{(B \rightarrow A) * \omega_B} \quad (6)$$

being ω_A the weight of the trajectory A in the interpolation, ω_B the weight of the trajectory B, $\omega \in [0, 1]$, $\sum \omega = 1$.

The similarity between both trajectories is computed using the dynamic time warping [24], [25] (DTW). We use a genetic algorithm to find the regularization term and number of Gaussians. The score used in the genetic algorithm is:

$$\sum_{\omega_A} DTW(traj_{(A \rightarrow B) * \omega_A}, traj_{(B \rightarrow A) * (1 - \omega_A)}) \quad (7)$$

for $\omega_A = \{0.4, 0.5, 0.6\}$. These three values are used because the similarity of the interpolated trajectories are very different. We minimize the score of the candidate solution.

The initialization of the population is completely random for $n_{gauss_{Geod}} \in [10, 4000]$, and the logarithm of regularization term ($\log_{10}(regTerm) \in [-4, 2]$). The selection operator picks with higher probability those candidates with lower score. We use a BLX-alpha crossover [26] with $alpha = 0.5$. The

²To compute this method, we use fitrree from Matlab 2018b

³In particular, we employ the demo.Riemannian_cov_interp03 method at <http://www.idiap.ch/software/pbdlib/>

Geodesic interpolation of Shy-Agree 7 and Agree-Shy 7

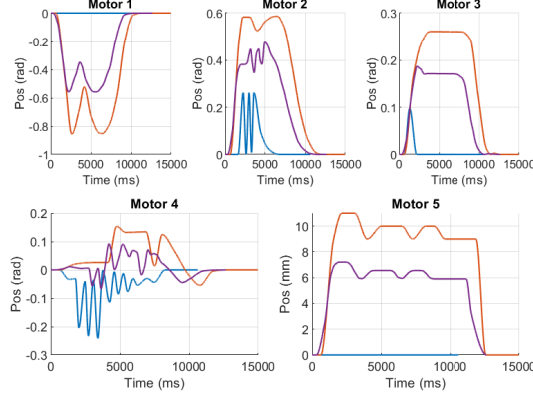


Fig. 6: Geodesic Interpolation between two trajectories Shy (in red) and Agree (in blue). With $w = 0.5$, in purple, the interpolation of Shy to Agree, and in yellow, the interpolation of Agree to Shy. This is the same interpolation as Fig. 5 but with the variances of both GMM adjusted.

mutation operator changes with the probability of 1% the value of a parameter for other value in the same range. The new population is formed by a quarter of the best candidates, half of the new candidates produced by the crossover operator and a quarter of new random candidates. As we can see in Fig. 6, the interpolation is now commutative and (6) holds.

V. TESTING METHODOLOGY

To test how well the three methods can generate interpolated animations and how readable these animations are, we have carried out a user study. As we have mentioned in Section IV, we are using the animations of Happy, Sad, Shy, Agree, and Disagree, each one with two different intensities, as training data; and we generate the interpolations: Happy-Agree, Shy-Agree, Shy-Disagree and Sad-Disagree, for each of the two intensities.

This user study is divided in 3 parts: The first part is an introduction to the robot. We give the participants a quiz where they have to answer, for each animation displayed, which expression do they think Haru is trying to convey. For that, the quiz is structured like a table with 5 empty columns. In the first and third columns, there is a Plutchik's wheel where the participants can mark pure expressions. We ask the participants to mark in the first and second columns which is the most suitable expression of the animation displayed, and with which accuracy they think their answer is right. In the third and four columns, we ask the same but with the second most suitable expression, and its accuracy. The accuracy is a value between 0 and 100, being 100 if they have no doubts. Only the results with an accuracy > 20 will be annotated. In the last column, the participants can write a comment if they want to mention something. We encourage them to select two expressions, unless they are sure they know the answer.

We display each animation 3 times. In this part, only the hand-made animations are shown, to make sure the participants understand the experiment, the limitations of the robot, and the expressions they have to choose between. For each group of participants, we select randomly one intensity for each expression (5 animations in total), which are displayed randomly.

After this, the animations are displayed once again in the same order, and the interviewer explains to which expression corresponds each animation. This is done to avoid problems if the participant classify wrongly one or more of the hand-made animations.

The second part of the user study repeats the same procedure as the first part, but the animations displayed are 4 of the non-used hand-made animations, and 4 interpolated animations (1 intensity for each interpolated expression) for each of the three interpolation methods. In total, 16 animations are presented to the participants ($4 + 4 \cdot 3 = 16$). Also, the order of the animations is different for each group. The participants have to fill the same questionnaire as before.

In the third and last part, the interviewer asks the participants to give a mark to the interpolation methods. Of course, the method is not indicated, but groups of three animations are shown, all three conveying the same interpolated expression (which in this case is indicated by the interviewer) with different methods. The interviewer asks which of the three interpolated animations fits better. The participants have to give an "A" to the best one, a "B" to the second one and "C" to the worst one. This is done for the 4 interpolated expressions. The interpolation method order and the expression order are selected randomly for each group of participants.

VI. RESULTS

The user study has being carried out with 15 people (9 men and 6 women) that have seen the robot before no more than three times. The mean of the participants age is 34 years old with a standard deviation of 11. The nationality of the participant are Japanese, Canadian, American, Mexican, Brazilian and Philippine. The animations were recorded and shown to the participants on video to make the experiment scalable and reproducible.

A. Hand-made animations

As mentioned, in the first step, the interviewer asks the participants which expressions they feel the animation is reproducing. The results for the hand made animations are presented in Fig. 7. The first time the participants saw Haru's animations (Figure 7a), we get a matrix in which the Agree and the Sad animations are really identifiable. Shy and Disagree are misidentified with Sad. The Happy animation shown in the first part is confused with the Shy one. Then, the experimenter explains the shown expressions, and new ones are presented. After that Shy is perceived as Sad, meanwhile Disagree gets better results. The animation of Happy with intensity 7 was removed of the test-bench because the motions were a bit too aggressive and could have

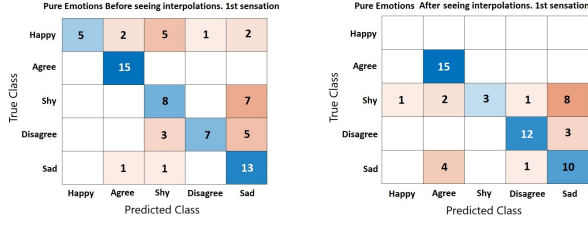


Fig. 7: Confusion matrix of hand-made expressions before and after seeing any of the interpolated expressions.

harm the robot (but it is used as training data). As we do not have the second animation for Happy, we cannot check if this impression changes after seeing some interpolations.

B. Synthetic animations results using the 3 methods

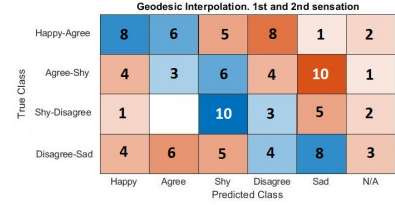
To articulate the data collected in the quiz, the results are presented as a confusion matrix with two diagonals of true classifications instead of one. For each pair of expressions in an interpolation, if one of those expressions is predicted, it is considered as a good prediction.

The Geodesic interpolation (Figure 8a) generates movement which cannot be distinguished among each other. Even the most extreme expressions, Happy and Sad, are hardly identified in the interpolations.

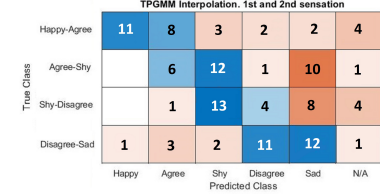
The TPGMM interpolation (Figure 8b) works well for all the interpolations. The worst detected interpolation is Agree-Shy, because Sad is predicted instead. As we see in Fig. 7, Shy and Sad are usually confounded even in the hand-made animations, so it is normal that users guess that both expressions could be possible, but they do not know which one. We see the same effect in the interpolation Shy-Disagree. However, Agree-Shy is more recognizable than Shy-Disagree. Still, all the interpolations have at least one of the expressions well recognised and the other one more recognised than any other expression (excepting the one that classify Shy as Sad).

The last interpolation, made by Binary Regression Decision Tree (Figure 8c), predicts worse than 8b, but wrong predictions are not only located in the Sad expression, as in the TPGMM interpolation. In this method, errors are bigger than in the previous case, but bad predictions are near the diagonals, which means people have the feeling if an animation is 'happier' or 'sadder', but cannot identify them correctly.

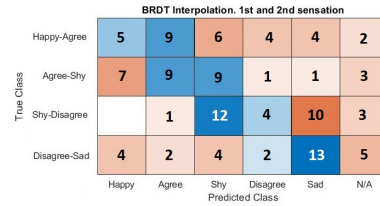
If we compare the results of the three methods, we can see that the best one is the TPGMM interpolation as all the interpolations are better recognized than the other methods. And BRDT have better results than Geodesic interpolations, as people can classify well if an animation is happier or sadder. This is due to TPGMM and BRDT use a model to classify and order the expressions meanwhile the Geodesic interpolation use only the information in the data, which seems not enough information.



(a) Geodesic interpolation.



(b) TPGMM interpolation.



(c) BRDT interpolation.

Fig. 8: Confusion matrix guessing both expressions.

C. What is the preferred method?

In the third part of the study, we analyze which method is better when combining expressions from the point of view of the participants. By comparing these three methods to each other (see Fig. 9), the TPGMM method in Agree-Shy and Disagree-Sad interpolations are the best by far. In those interpolations in which TPGMM is not the preferred one, it has almost the same rate as the second more selected, the Geodesic interpolation. Also, in the case of the two best interpolations of the three of them, the TPGMM method has the same results, being the best method. The Geodesic method is preferred more times than the BRDT method almost always. This may be due it is easier to see the expressions of the animation if you know what expressions are you searching for.

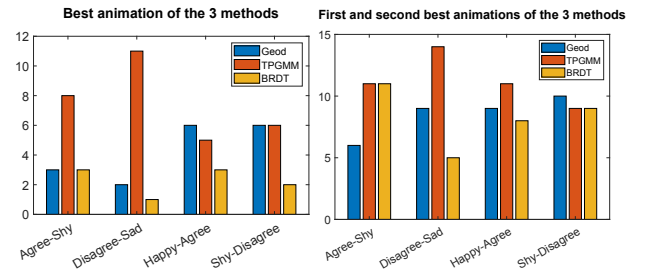


Fig. 9: "What do you think is the animation generated that better describe this interpolated expression?" answers. The first graph is counting only their first option and the left one is using both first and second options.

VII. CONCLUSIONS

Discussion. The paper considers the automatic generation of new expressive motions from a set of exemplary animations provided by humans. We approached this problem as a learning from demonstration scenario, and have analyzed three different LfD methods. First, we used the psychological model of Plutchik's wheel as inspiration to encode the expressions associated to the animations. Relying on a pre-made model makes the solution as good as your model, so we had to choose between which model we use or if we wanted to use any model. Then, we studied three different interpolation methods to learn and generate the trajectories of the animations. The first two methods, TPGMM and BRDT, rely on the indicated model to encode the animations. However, the Geodesic interpolation does not rely on any model. The results in the Geodesic interpolation are worse than the TPGMM and the BRDT interpolations. TPGMM and Geodesic interpolations use GMM to encode the trajectories of the animations. Due to that, the solutions depend on how we approximate the GMM to the animations. In summary, we obtained that the Plutchik's wheel model obtains better results than not using any model, and TPGMM works better than the other two approximations.

Limitations and Future Work. We are working on generation of expressions in the technical terms through interpolation and in the future we will be focusing on analyzing deeper meaning associated to the higher meaning of expressions. In this work, we assume that the Plutchik's wheel is a helpful model to encode the expression-related variables, as it provides a (qualitative) distance between expressions. However, the Plutchik's wheel is made for human expressions, so extrapolating that to robot's movement require a more extended study. Despite we have good results in the user study with only 12 animations, we would like to repeat the user study with a larger population and additional hand-made and interpolated animations, covering a wider spectrum of expressions. Additionally, we will compare the methods with those that encode motion styles as cost functions using IRL, like [6], [8].

The work presented is a first step towards the objective to generalize the production of these expressions automatically. As future work, we would like to extend this work to the combination of motion tasks, like tracking a person while expressing an expression, as these expressive motions are not an end in themselves, but they will be used by the robot when it is performing tasks [8]. The objective is to extract the relevant information from the demonstrations of "static" expressions and combine it in a certain way with the motion commands resulting from the tracking modules.

REFERENCES

- [1] D. Perzanowski, A. C. Schultz, W. Adams, E. Marsh, and M. Bugajska, "Building a multimodal human-robot interface," *IEEE Intelligent Systems*, vol. 16, no. 1, pp. 16–21, Jan 2001.
- [2] A. K. Pandey and R. Gelin, "A mass-produced sociable humanoid robot: Pepper: The first machine of its kind," *IEEE Robotics Automation Magazine*, vol. 25, no. 3, pp. 40–48, Sep. 2018.
- [3] E. Pot, J. Monceaux, R. Gelin, and B. Maisonnier, "Choregraphe: a graphical tool for humanoid robot programming," in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, Sep. 2009, pp. 46–51.
- [4] R. Gomez, D. Szapiro, K. Galindo, and K. Nakamura, "Haru: Hardware design of an experimental tabletop robot assistant," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2018, pp. 233–240.
- [5] C. K. Liu, A. Hertzmann, and Z. Popović, "Learning physics-based motion style with nonlinear inverse optimization," in *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 1071–1081.
- [6] S. J. Lee and Z. Popović, "Learning behavior styles with inverse reinforcement learning," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4. ACM, 2010, p. 122.
- [7] C. Bodden, D. Rakita, B. Mutlu, and M. Gleicher, "Evaluating intent-expressive robot arm motion," pp. 658–663, 2016.
- [8] A. Zhou and A. D. Dragan, "Cost functions for robot motion style," pp. 3632–3639, 2018.
- [9] C. T. Ishi, H. Ishiguro, and N. Hagita, "Analysis of head motions and speech, and head motion control in an android robot," *Geminoid Studies: Science and Technologies for Humanlike Teleoperated Androids*, pp. 89–110, 2018.
- [10] L. Rincon, E. Coronado, H. Hendra, J. Phan, Z. Zainalkefi, and G. Venture, "Expressive states with a robot arm using adaptive fuzzy and robust predictive controllers," in *2018 3rd International Conference on Control and Robotics Engineering (ICCERE)*. IEEE, 2018, pp. 11–15.
- [11] M. Santos and M. Egerstedt, "From motions to emotions: Can the fundamental emotions be expressed in a robot swarm?" *arXiv preprint arXiv:1903.12118*, 2019.
- [12] C. Bodden, D. Rakita, B. Mutlu, and M. Gleicher, "A flexible optimization-based method for synthesizing intent-expressive robot arm motion," *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1376–1394, 2018.
- [13] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*. Springer, 2008, pp. 1371–1394.
- [14] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [15] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.
- [16] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," pp. 323–329, 2012.
- [17] N. Jaquier and S. Calinon, "Gaussian mixture regression on symmetric positive definite matrices manifolds: Application to wrist motion estimation with sEMG," 2017.
- [18] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an em approach," in *Advances in neural information processing systems*, 1994, pp. 120–127.
- [19] Y. Wu and Y. Demiris, "Towards one shot learning by imitation for humanoid robots," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2889–2894, 2010.
- [20] E. I. Barakova and T. Lourens, "Expressing and interpreting emotional movements in social games with robots," *Personal and ubiquitous computing*, vol. 14, no. 5, pp. 457–467, 2010.
- [21] R. Plutchik, *Emotion: A Psychoevolutionary Synthesis*. Harper and Row, 1980. [Online]. Available: <https://books.google.co.jp/books?id=G5t9AAAAAAAJ>
- [22] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, 2015.
- [23] L. Breiman, *Classification and regression trees*. Routledge, 2017.
- [24] H. Sakoe, S. Chiba, A. Waibel, and K. Lee, "Dynamic programming algorithm optimization for spoken word recognition," *Readings in speech recognition*, vol. 159, p. 224, 1990.
- [25] K. K. Paliwal, A. Agarwal, and S. S. Sinha, "A modification over sakoe and chiba's dynamic time warping algorithm for isolated word recognition," *Signal Processing*, vol. 4, no. 4, pp. 329–333, 1982.
- [26] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," in *Foundations of genetic algorithms*. Elsevier, 1993, vol. 2, pp. 187–202.