Received September 30, 2020, accepted October 13, 2020, date of publication October 26, 2020,
date of current version November 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033550

Developing a Lightweight Rock-Paper-Scissors Framework for Human-Robot Collaborative Gaming

HEIKE BROCK[1], (Member, IEEE), JAVIER PONCE CHULANI[2], LUIS MERINO[2], (Member, IEEE), DEBORAH SZAPIRO[3], (Member, IEEE), AND RANDY GOMEZ[1], (Member, IEEE)
[1]Honda Research Institute, Japan Company, Ltd., Wako-Shi 351-0188, Japan
[2]Systems Engineering and Automation Department, University Pablo de Olavide, 41013 Seville, Spain
[3]School of Design, University of Technology Sydney, Sydney, NSW 2007, Australia

Corresponding author: Heike Brock (h.brock@jp.honda-ri.com)

ABSTRACT We present a novel implementation of a Rock-Paper-Scissors (RPS) game interaction with a social robot. The framework is tailored to be computationally lightweight, as well as entertaining and visually appealing through collaboration with designers and animators. The fundamental gesture recognition pipeline employs a Leap motion device and two separate machine learning architectures to evaluate kinematic hand data on-the-fly. The first architecture is used to recognize and segment human motion activity in order to initialize the RPS play, and the second architecture is used to classify hand gestures into rock, paper or scissors. The employed tabletop robot interacts in the RPS play through unique animated gestural movements and vocalizations designed by animators which communicate the robot's choices as well as cognitive reflection on winning, losing and draw states. Performance of both learning architectures is carefully evaluated with respect to accuracy, reliability and run time performance under different feature and classifier types. Moreover, we assess our system during an interactive RPS play between robot and human. Experimental results show that the proposed system is robust to user variations and play style in real environment conditions. As such, it offers a powerful application for the subsequent exploration of social human-machine interaction.

INDEX TERMS Animation, gesture recognition, human-robot interaction, intelligent robots, motion segmentation, social robotics.

I. INTRODUCTION

The classic Rock-Paper-Scissors (RPS) game constitutes a simple action and effect interaction with long-standing fascination for various research domains. For example, RPS-like patterns are used by sociology to explain the dynamics of collective efforts [1], and by biology to investigate the evolution of ecosystems [2]. For psychology, analysis of the basic gestural play can be interesting to identify patterns or strategies in human decision-making process [3]. In engineering and robotics, RPS game interactions between a human and a robot player build the base for two main research questions: basic technological development [4], [5] on the one hand, and its deployment in Human-Machine-Interaction studies [6]–[9] on the other. Here, we seek to combine both aspects, with the primary goal to create a machine learning-driven RPS playmate whose smoothness and procedure of play interaction is highly appealing to the human opponent. Most importantly, we target a system that is flexible towards individual variations in movement and game timing without predefined game constraints. Under a high level of flexibility and robustness, our system should then subsequently be expanded to include intent prediction and game control via reasoning or reinforced learning, as well as be employed for various social exploratory studies.

Kanda *et al.* [10] were the first to discuss a RPS interaction between a human and a robot. They included a RPS (here in Japanese: Janken) game module in their proposed robot control architecture. Subsequently, most works that introduced RPS robot designs focused on the creation of an interactive human robot hand which can form the three RPS hand shapes. Examples are the robot presented by Hasuda *et al.* [11], the robot of Ahn *et al.* [12] and the model by Lin *et al.* [13]. The most recent prototype is the 'Janken robot' developed by the Ishikawa Watanabe laboratory [14]. Equipped with a high-speed camera setup, its sensing system

The associate editor coordinating the review of this manuscript and approving it for publication was Maani Ghaffari.

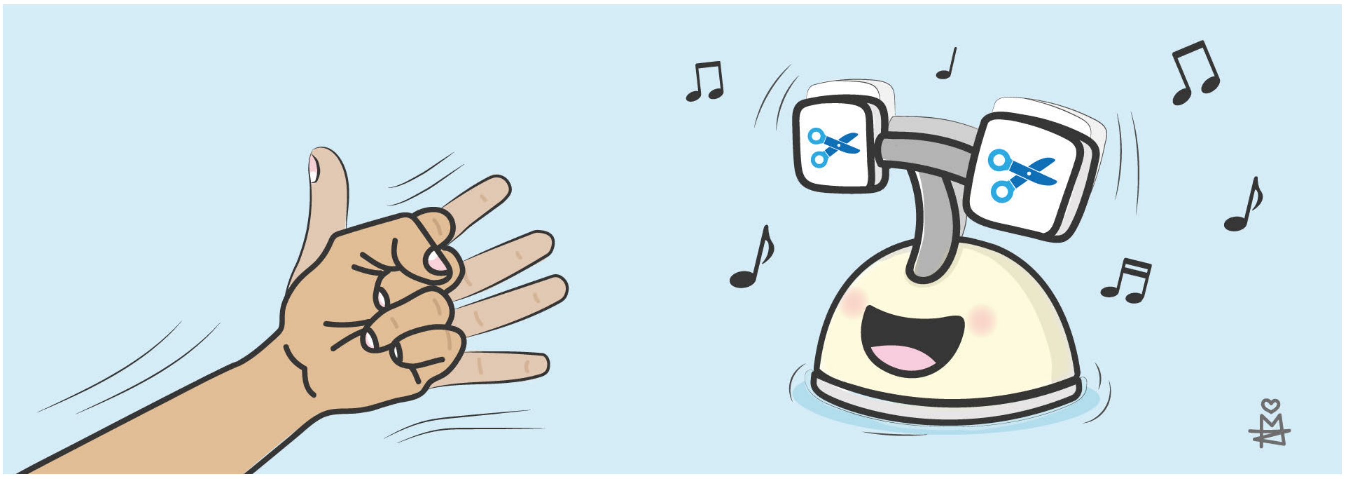


FIGURE 1. **Illustration of the envisioned Rock-Paper-Scissors game interaction with the employed social tabletop robot Haru.**

can recognize and react to human play within fractions of a second, enabling the robot to achieve a winning rate of 100%. While the system is highly impressive from a technical side, interactivity is largely restricted to predefined play conditions such as the number of upper-lower hand movements before play. Moreover, fixed technological settings of predefined camera and mirror arrangements make the overall system difficult and costly to deploy outside a laboratory. Overall, the RPS games mentioned above are very hardware-focused and lacking of design considerations in terms of movements, sounds and interactivity. For example, the representation of the robot opponent via a human-like hand might hinder the exploration of social and emotive aspects of a two-player interaction. For these reasons, we present a distinct technical solution that we combine with input from designers and animators to add creativity to the system implementation process. We furthermore perform a rigorous analysis of the employed learning processes, in order to create a smooth and reliable environment for real-time human-robot play interaction.

First of all, we propose a lightweight and portable RPS framework. To achieve mobility, all of the necessary motion perception is handled by a single Leap Motion device [15], which we connect to the basic robot hardware. In contrast to full-body pose tracking camera sensors, the Leap Motion specializes in the tracking of hand and finger movements, making it the best sensing device for our targeted ubiquitous RPS play scenario. Within the area of hand movement analysis, the sensor has previously been utilized to explore the recognition of gestural movements [16] and isolated sign language expressions [17]. Moreover, it has been employed for gesture based control of robots [18]–[20] and evaluated for use in close-distance non-verbal human-robot communication [21]. In addition, we integrate our RPS interaction pipeline into an emotive robot platform designed for social robot research. For this, we utilize the exploratory tabletop robot Haru [22] as depicted in Fig. 1. Haru is equipped with a set of pre-defined animated behavior routines created in line with the robot's unique actuators and design [23]. This allows for the the implementation of a completely novel human-robot RPS game experience.

To evaluate our framework, we investigate both the data processing functionalities to attain robustness, and the real-time interaction process to facilitate a smooth and natural game interaction. Most importantly, results show that we succeed to implement a system of high recognition accuracy which does not rely on pre-defined play and hand movement actions. Moreover, the system appears robust and practical within the collaborative game setting. Here, its real-time set up shows that a human-like game interaction can be attained with a playmate of non-human actuation and modality, supporting future directions in social robotics research and embodied communication.

II. PROBLEM DEFINITION

We strive to align our framework to the classical sequence of a two-player RPS interaction, whereas the main cognitive processes of the robotic opponent should be invariant towards different types of RPS plays. This is particularly important to successfully handle different user styles, game situations and cultural backgrounds. For example, whereas in Central Europe a RPS round always ends after the third up-down movement (verbal cue:'tic-tac-toe' or 'schnick-schnack-schnuck'), in Japan a first RPS round lasts four up-down movements (verbal cue: 'saisho-gu-janken-pon'), and a round following an initial tie (verbal cue: 'aiko-deshou') two up-down movements. Inconsistencies in a player's movement are prominent in the basic movement characteristics of the hand, as e.g. the palm acceleration shown in Fig. 2, and influence system quality. To establish a universal play intelligence, it is therefore necessary to determine a strategy that specifically addresses this data variability.

Based on the fundamental hand dynamics, we define a single RPS round to constitute of five distinct motion phases P1-P5 as illustrated in Fig. 3. P1 is the static rest phase at the beginning of the play without movement of the hand. P2 is the

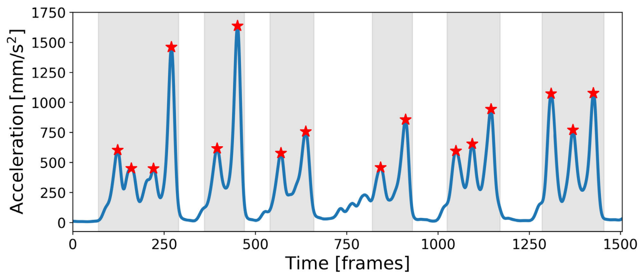


FIGURE 2. Smoothed absolute palm acceleration during a free RPS play composed of 6 rounds (gray zones). The length of one round – as defined by up-down swings of the hand – is variable, whereas every upwards swing (red marks) results in a local curve maximum.

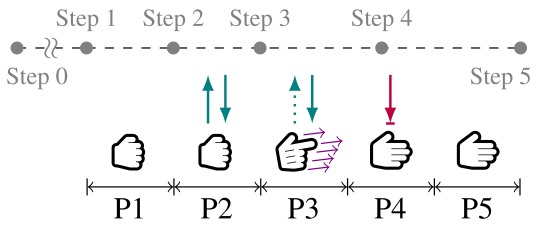


FIGURE 3. The human player's movement is split into five major movement phases P1-P5 within one single RPS round. These phases outline the overall system's perception process (Step 0 to Step 5). The dynamic phases P2-P4 are most relevant for the robot perception and interaction functionalities.

dynamic preparatory phase during which the hand (usually in closed fist) is swung multiple times in upwards-downwards direction. P3 is a highly dynamic formation phase during which the fingers are brought into position to form any of the three RPS shapes. P4 is a reduced dynamic deceleration phase which brings the hand (with fingers formed to the played shape) into a final static position. Lastly, P5 defines the static ending phase used to evaluate the winner of the play interaction.

The sensing and processing of the human movement actions then undergo the following stages:

Step 0 Start the system, process all incoming frames
Step 1 Establish stand-by (zero) state for P1
Step 2 Identify the beginning of P2, switch to active state
Step 3 Recognize the initiation of P3, issue 'go' signal
Step 4 Recognize hand shape during P4, distribute 'shape' information
Step 5 Return to zero state by the end of P5

As discussed in this section, it is evident that the simple RPS play is mired with variability issues that need to be rectified prior to deployment in an actual human-robot interaction application.

III. RECOGNITION SYSTEM

In this section we will discuss the schemes in developing a RPS gesture recognition system using hand movement features such as finger joint positions, finger and palm directions and velocities. We split the recognition process into two tasks, namely **task (1)**: movement segmentation of P2 in order to identify the initiation of P3, and **task (2)**: hand shape classification into RPS throughout P4 to P5.

A. MACHINE LEARNING DATA

We recorded RPS movements of 15 volunteers (11 male, 4 female) in an open space of an office environment with semi-constant light conditions. To address general individual differences in movement style, but also cultural differences in play as discussed under Section II, we strived to collect as variant data as possible. As such, our volunteers had diverse age (average 36.2 years with a variance of 8.30 years) and multiple cultural backgrounds (6 Japanese, 2 Germans, 2 Chinese, 1 Spanish, 1 French, 1 Danish, 1 Filipino, 1 American). Every actor was asked to perform 5 takes with a minimum of 5 repetitions for all three RPS shapes. The actors were free to choose their positioning (seated, standing) and could execute their movement with any of the left or right hand in their own unique style of playing. This led to 1258 action segments that were used as training set $\mathcal{T}$ for the learning of a player-independent perception module. For system validation, we additionally collected independent sequences of 5 to 10 RPS rounds under real, free play conditions. The resulting 214 action segments build our evaluation data set $\mathcal{V}$. All data was annotated with respect to their necessary label information, the beginning and end of movement phase P2 for task (1), and the first and last frames of the hand being held in any of the three shapes for task (2).

B. FEATURE TRACKING AND PROCESSING

We utilize basic motion information provided by the Leap SDK [15] to build distinct feature representations for both tasks (1) and (2), respectively. Features should be simple and robust towards sensing and tracking errors present in the SDK's provided hand information. We engineer the features to be extracted easily and fast in order to reduce computational overload and avoid delay in the overall signal processing. Moreover, to avoid dependence on the SDK's internal feature computation processes, all features base on the hand joint positions only. As such, our proposed game pipeline could also be implemented similarly without a Leap device in the future, e.g. by utilizing video-based systems with hand tracking options such as OpenPose [24] or MediaPipe [25].

1) MOVEMENT SEGMENTATION FEATURES

According to the characteristics of the different RPS phases, P2 is distinguishable from its preceding and succeeding phases by changes in the hand and finger velocities and motion trajectories. We reflect this to form the set of feature representations $\mathcal{F}_{\text{Seg}}$ utilizing the basic velocity and direction vectors of a player's hand. These are $\vec{v_j}$ representing the velocity in 3D space of any joint $j \in \mathbb{J}$ as

$$\vec{v_j} = \begin{bmatrix} v_{j_x} \\ v_{j_y} \\ v_{j_z} \end{bmatrix}, \tag{1}$$

whereas $\mathbb{J}$ constitutes the set of palm and finger tip joints, and $\vec{u_d}$ representing relative changes of the vectors $d \in \mathbb{D}$ as

$$\vec{u_d} = \begin{bmatrix} u_{d_x,i-1} \\ u_{d_y,i-1} \\ u_{d_z,i-1} \end{bmatrix} - \begin{bmatrix} u_{d_x,i} \\ u_{d_y,i} \\ u_{d_z,i} \end{bmatrix}, \tag{2}$$

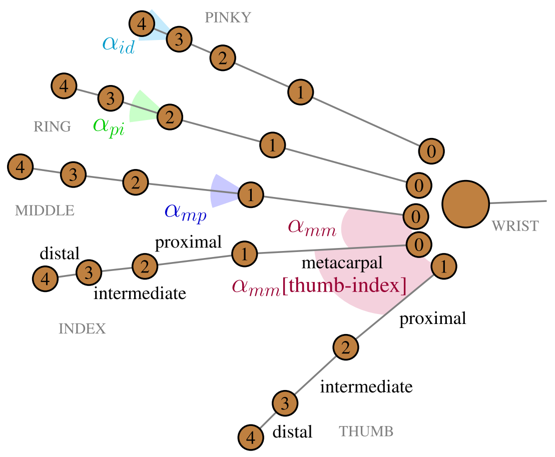


FIGURE 4. Schematic illustration of the human hand and the chosen angular features computed from the joint positions 0, 1, 2, 3 and 4 per finger. Please note that the thumb has no 0 (metacarpal) joint, and hence no metacarpal-proximal bone angle α_{mp}.

whereas $\mathbb{D}$ constitutes the set of normal and direction vectors of the palm and $i \in [2, \ldots, n]$ with n denoting the length of a motion sequence. The final feature set is then defined as $\mathcal{F}_{\text{Seg}} = \{\overrightarrow{v_j}, \overrightarrow{u_d}\}$ for all j and d respectively.

2) SHAPE RECOGNITION FEATURES

Utilizing the positions of finger joints in 3-dimensional space, the classes 'Rock', 'Paper' and 'Scissors' can be uniquely described by their angular and distal relations. The set of feature representations $\mathcal{F}_{\text{Shape}}$ therefore constitutes the normalized inter-joint angles computed from a hand's raw joint positions. For all neighboring finger bones, we compute the angle α of the length-normalized inner product of the two vectors $\overrightarrow{b_{J_1J_2}}$ and $\overrightarrow{b_{J_3J_4}}$ representing the respective bones (J_1, J_2) and (J_3, J_4) as

$$\alpha = \arccos \frac{\overrightarrow{b_{J_1J_2}} \cdot \overrightarrow{b_{J_3J_4}}}{||\overrightarrow{b_{J_1J_2}}|| \cdot ||\overrightarrow{b_{J_3J_4}}||}. \quad (3)$$

This provides us with the angles α_{mm} between metacarpal finger bones (respectively between proximal and metacarpal bone for the thumb and index finger), as well as the angles α_{mp} between metacarpal and proximal bone, α_{pi} between proximal and intermediate bone and α_{id} between intermediate and distal bone of every finger (Fig. 4).

We combine the angles to the final feature set $\mathcal{F}_{\text{Shape1}} = \{\alpha_{mm}, \alpha_{mp}, \alpha_{pi}, \alpha_{id}\}$. For subsequent comparison, we furthermore build a second set $\mathcal{F}_{\text{Shape2}} = \{\alpha_{pp}\text{[thumb-index]}, \alpha_{mp}, \alpha_{pi}\}$ with reduced morphological information.

C. MACHINE LEARNING ARCHITECTURES

We create two different machine learning architectures for task (1) and (2) of the required gesture understanding capabilities. For both tasks, we first define a basic recognition strategy. To find the best balance between accuracy and reliability and computational lightweight and overload, we then train several machine learning classifiers that are commonly employed in similar learning tasks [26]. In concrete, these are a Random Forest (RF) with bootstrapping and 300 decision trees, a Gaussian Naive Bayes (GNB) classifier, a decision tree with Adaptive Boosting (ADA), and a Support Vector Machine (SVM) with RBF kernel. For the segmentation task, we furthermore learn a Convolutional Neural Network (CNN). Here, the idea is to examine whether the intrinsic feature learning abilities of a CNN are superior in describing the phase transitioning between P2 to P3. The CNN is built of two stacks of convolution and pooling layers and two dense layers. The first stack consists of three convolution layers with a kernel of size (3×1) evaluating data along the temporal dimension only, followed by a pooling layer with kernel size (2×2). The second stack consists of three convolution layers with a kernel of size (3×3) and a (2×2) pooling layer. We apply ELU activation to all convolution layers and the first dense layer. The second dense layer has a sigmoid activation for classification.

1) MOVEMENT SEGMENTATION

We choose to adapt and modify the binary strategy for movement segmentation that we previously introduced in the context of continuous sign language recognition [27]. This strategy is learning temporal context information based on the accumulated information within windows of neighboring feature vectors. The idea here is that this contextual frame-wise classification should enable both an immediate and flexible robot reaction to the different time duration in a free RPS play. Precisely, we observe a subtle change of velocity, direction and movement flow before the final play's downward movement (characterizing P3 and P4) within all of our data. This movement pattern can serve to characterize the transition between P2 and P3. To define this phase transition and to learn the temporal characteristics of the swinging hand movement during P2, we consider a window of neighboring feature vectors. As proposed in the original work, all frames that are not annotated as part of P2 are treated as class 0, and all frames that are annotated as part of P2 are treated as class 1. The input to all classifiers then constitutes a concatenation of vectors as given by $\mathcal{F}_{\text{Seg}}$ and their respective frame-wise labels.

The trained classifier returns the probability of an incoming feature vector to belong to class 1 on the base of its smoothness in movement flow. To make the output estimate more robust towards outliers (e.g. stutters and changes of direction in a player's hand trajectory), we smooth the sequence of frame-wise predictions with a Gaussian filter of kernel deviation $\sigma = 3$. Every frame with a smoothed confidence ≥ 0.5 is then labeled as 1, and all remaining frames are labeled as 0. The first frame labeled as 1 marks the beginning of P2 and the next occurrence of a 0 labeled frame marks its end (and elicits the issuing of the 'go' prompt for control of the robot play action).

2) SHAPE RECOGNITION

The shape recognition constitutes of a standard multi-class classifier trained with frame-wise feature vectors taken from

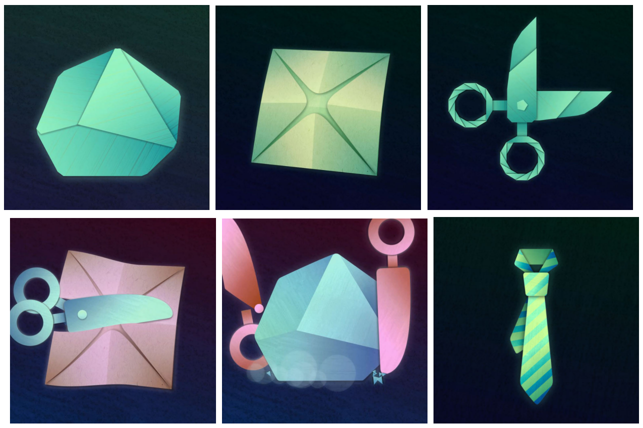


FIGURE 5. Still images taken from the custom-made 3D animations displayed in the robot's two LCD eye panels. Top row: robot choices. Bottom row: game results.

$\mathcal{F}_{\text{Shape1}}$ or $\mathcal{F}_{\text{Shape2}}$ and their three assigned RPS labels. The learned classifier returns the probability of an incoming feature vector as belonging to any of the classes. A hand shape is considered as officially classified (and the appropriate 'shape' message for control of the robot's game result responses distributed) as soon as the confidence for one class reaches a value ≥ 0.8 for more than 15 consecutive frames.

IV. ROBOT PLAY DESIGN

The employed Haru platform allows the composition of open-loop macro-actions to control the robot's game and play behavior [23]. These animation-like performances – so-called animation routines – make use of the robot's motion and audio-visual actuation modalities. Namely, these are: motor-controlled motion (body rotation, neck lean, eyes tilt and roll), sounds, a 3-inch stereo colour LCD screen functioning as the robot eyes, the eye movement within the LCD panel, and addressable LED strips wrapped around the borders of the eye.

Different routines have been designed to express the actions and decisions in the game (i.e. when the robot selects either 'Rock', 'Paper', or 'Scissors'), the robot's awareness of the played game result (i.e. winning, losing, or drawing), as well as the reactions of the robot to the final outcome of the play interaction (i.e. sadness, happiness, or neutrality). Due to the unique morphology of the robot, especially the former two aspects are very important. They ensure to provide a familiar and intuitive setting to the human player, and hence deliver a convincing game experience that encourages a natural RPS play environment. We choose to communicate the robot choices utilizing the prominent LED eye panels. Every selection is displayed via a moving still image of a rock, (a sheet of) paper, and a scissor as illustrated in Fig. 5. Similarly, we convey the subsequent robot's mental states reflecting the game result with a 3D animation displaying either rock beating scissors, paper beating rock, scissors beating paper, or a tie – respectively draw – result. Lastly, the robot's reactions on final game results are chosen from an existing pool of animations available in the robot platform to convey emotional expressions and create an illusion of the robot's emotive behavior. Bottomline, all of the robot's interactivity – starting from choreography through animation routines, icon displays, sounds, etc. – were all custom made through partnership with designers and animators.

V. BASIC GAME PIPELINE

The implementation of the whole RPS game pipeline comprises of a series of steps that fold around the previously described recognition architectures and animated behavior routines. These steps are organized by a behavior tree that structures the flow of all tasks and controls the robot drivers via Robot Operation System (ROS) commands. Fig. 6 illustrates the flow between basic states of the game as it is also described in detail in the following. Variations and adaptations of the interaction flow to specific game interactions can then be incorporated in a flexible way thanks to the modularity of the behavior tree.

A. START GAME STATE

Directly after system start, the start game state initializes two game counters that are used to keep track of the robot's number of losing and wining rounds. At the start of every RPS round, Haru furthermore shows a countdown using its eye screens. During this state, the robot selects its next RPS play option (either 'Rock', 'Paper' or 'Scissors'). Currently this is done randomly, but the strategy could be modified in the future. For example, Haru's choice might change in dependency of the previously seen options of the player to model a more human-like interaction.

B. LISTENING AND SHOW STATES

In these states, the phase segmentation and recognition architectures described above come into play. First, during the listening state, the robot awaits to detect the end of the P2 phase as indicated by the 'go' signal. As soon as the 'go' signal is issued, the robot switches to the show state and displays the animation routine that conveys its current RPS selection. This immediacy ensures to synchronize the selection of the robot with that of the player, and to avoid delays that may give the impression that the robot is selecting its option after seeing the player's selection.

After displaying its play selection, the robot awaits the RPS 'shape' information, which is commonly retrieved before phase P4 of the RPS recognition module ends. The system then moves to the next states, depending on the result of the round.

C. DRAW, WINNING AND LOSING STATES

The draw, winning and losing states show an animation routine reflecting the result of the current round (draw, win or lose), and update the game counters accordingly. If one of the counters is above a defined threshold (meaning either the player or the robot has won a specified number of times), the system evolves to the end game state. If not, or if there is a draw, the system goes back to the state game state for a new RPS round.

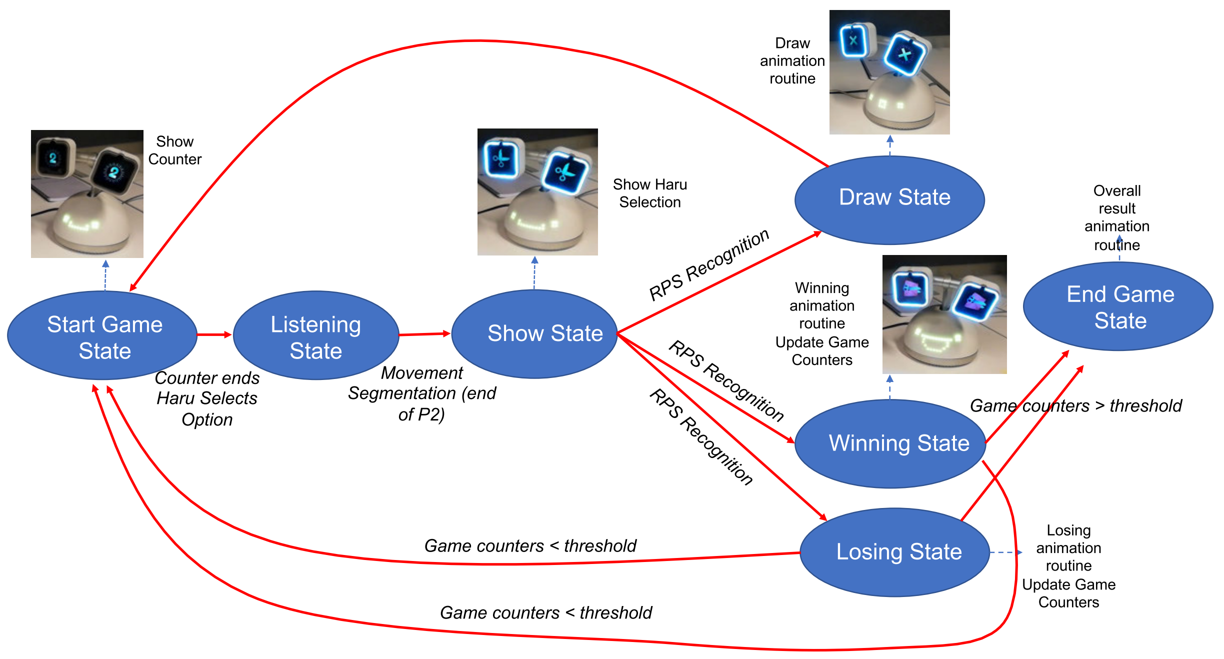


FIGURE 6. **Illustration of the game implementation. The ellipses show the main states. In italic, the main transition triggers are shown. The main actions performed in each state are also shown.**

D. END GAME STATE

In this state, the robot will select an adequate animation routine expressing the robot's emotion, such as happiness or sadness, depending on the overall result of the game. The counter threshold to reach the final pipeline state can be configured before system use.

VI. EXPERIMENTS

We evaluate our framework's perception components utilizing the common metrics accuracy, precision, recall and F1 score. Decision trees were shown to achieve good results in similar learning tasks [28], so that we treat the RF classifiers as base classifier architectures. This means we first analyze variations in feature window size and feature set composition under the RF, and then relate the respective best performing combinations to the remaining classifiers for comparison. Moreover, we determine the lengths of single predictions and plays on a consumer PC (CPU: Intel i7-7700) to draw conclusions on the computational load and lightweight during run time for real-time interaction with the robot.

A. MODEL EVALUATION

The models of the recognition module are analyzed as a function of variation in styles, timing and positions during RPS play. For best support with both common machine learning libraries and ROS, we utilize the Leap Python SDK version 2.3. All classifiers are trained utilizing the scikit-learn library, respectively Keras for the CNN. The Leap Motion perceives and processes the play of the human with a sampling rate of 120Hz.

1) MOVEMENT SEGMENTATION

To begin, we train three RFs that utilize a different window size of neighboring frame observations. RF_{Seg1} evaluates a window of size $w_1 = \pm 4$ frames around the current frame of interest, respectively 9 frames in total. This equals the movement information occurring within 75 ms. The window size of RF_{Seg2} is chosen as $w_2 = \pm 7$ frames, respectively 15 frames in total, or a movement span of 125 ms. Lastly, RF_{Seg3} handles feature vectors built from a window size $w_3 = \pm 10$, meaning 21 frames in total and 175 ms of hand movement information. As could be expected, the average quality of the binary predictions increases with the number of neighboring movement features observed, see Table 1. For RF_{Seg1}, we note early segmentation errors before the completion of P2. These are caused by short frame-wise misclassification during irregular movement executions. Opposedly, misclassifications of RF_{Seg2} and RF_{Seg3} mostly occur around the segment boundaries and the immediate transition frames between movement phases. They hardly affect intra-phase frame predictions and the segment estimates appear robust and reliable. Comparison between the latter two RFs – and particularly their precision scores – furthermore suggests that a larger window size is primarily helpful to better distinguish between class 1 and class 0 frames, therewith reducing the number of false positives. On the deployed hardware, a single frame prediction was of similar average run time for all the RFs. Since $\mathcal{F}_{Seg}$ utilizes data directly provided by the SDK, computational overload for feature extraction is also neglectable. Consequently, it appears meaningful to utilize a RF trained with longer feature vectors.

We next compare the performance of RF_{Seg2} to the performance of the remaining shallow classifiers with same feature window size $w_2 = \pm 7$ frames. We note that the GNB is faster than the RF, but the model fails to predict frames with reliable accuracy. Moreover, ADA augments both accuracy and run time, but cannot provide better predictions than the RF. Lastly, the SVM requires considerable more computation

TABLE 1. Performances of the tested classifiers for phase segmentation (averaged across test actors).

	accuracy	*precision*	*recall*	*F1*	*run time*
RF_{Seg1}	0.897	0.824	0.924	0.868	0.134 ms
RF_{Seg2}	0.911	0.843	0.942	0.887	0.134 ms
RF_{Seg3}	0.921	0.862	0.947	0.900	0.134 ms
GNB_{Seg2}	0.667	0.494	0.567	0.521	0.018 ms
ADA_{Seg2}	0.880	0.836	0.850	0.837	0.728 ms
SVM_{Seg2}	0.910	0.859	0.909	0.881	70.229 ms
CNN_{Seg2}	0.372	0.372	1.000	0.534	0.159 ms
CNN_{Seg3}	0.374	0.374	1.000	0.537	0.169 ms
CNN_{Seg2*}	0.885	0.805	0.912	0.852	0.152 ms
CNN_{Seg3*}	0.898	0.819	0.932	0.868	0.153 ms
RF_{Seg2*}	0.895	0.828	0.881	0.850	0.134 ms

time to achieve an equal level of accuracy as the RF. This makes RF_{Seg2} or RF_{Seg3} the preferred classifier.

Lastly, we determine the performance of the CNN. For this, we undertake minor changes in the classifier feature input: whereas it was necessary to flatten all neighboring feature vectors into one one-dimensional vector for the shallow classifiers, we can now maintain the original two-dimensional data structure with time (frames) being represented along one axis and the number of features along the second. Following the observations that longer features vectors provide better results, and to provide sufficient neighboring information to the convolution kernels, we start our analysis with feature images build from a window of w_2 neighboring frames. We train a CNN with batch normalization (batch size 128) and dropout of 0.5 with the Adam optimizer for 25 epochs. As one can see from the low accuracy and precision and perfect recall values, the classifier CNN_{Seg2} fails to learn relevant information for segmentation. The same holds for CNN_{Seg3} with longer feature window w_3. We suspect this problem to be a cause of the variant and differently sized features within each frame's feature vector. For this reason, we experiment with different feature arrangements until we find a suitable feature composition. The revised feature set $\mathcal{F}_{Seg*}$ then only constitutes of the pruned velocity vectors $\overrightarrow{v_j'}$ along the Leap sensor's relative y (left-right) and z (up-down) axis of any joint $j \in \mathbb{J}$. Training of the CNN with the revised feature set yields CNN_{Seg2*} for w_2 and CNN_{Seg3*} for w_3 with similar run time than the RF, but $\approx 3\%$ smaller performance values.

To assess the effect of potentially irrelevant features, we perform a final evaluation. For this, we train RF_{Seg2*} on the reduced feature set and window w_2. As opposed to the CNN, the novel RF cannot surpass the performance of the initial RF. This suggests that the two classifier methods learn to identify different information within the movement data.

2) HAND SHAPE RECOGNITION

We are interested to determine whether and how the size and selection of shape features influences the overall classifier performance. For this reason, we first train RF_{Shape1} and RF_{Shape2} based on $\mathcal{F}_{Shape1}$ and $\mathcal{F}_{Shape2}$. As shown in Table 2, the choice of feature transformation does not impact the performance of the shape classification. Data visualization reveals that the few occurring misclassifications are mostly caused by tracking errors within the underlying finger

TABLE 2. Performances of the tested classifiers for shape recognition (averaged across test actors).

	accuracy	*precision*	*recall*	*F1*	*run time*
RF_{Shape1}	0.993	0.993	0.992	0.993	40.745 ms
RF_{Shape2}	0.992	0.992	0.991	0.991	24.215 ms
GNB_{Shape2}	0.956	0.957	0.956	0.956	24.201 ms
ADA_{Shape2}	0.968	0.969	0.968	0.968	24.234 ms
SVM_{Shape2}	0.978	0.980	0.977	0.978	24.352 ms

joint positions. This suggests that a reduced set of hand shape descriptors is sufficient for the given task. Every feature vector is evaluated in 0.015 ms, whereas run time differences of a single frame prediction between the feature sets average to 0.00001 ms. However, we observe that $\mathcal{F}_{Shape2}$ requires less computation time than $\mathcal{F}_{Shape1}$. This makes $\mathcal{F}_{Shape2}$ the better choice for subsequent system deployment.

As before, we compare the performance of the best performing RF to the performances of the remaining shallow classifiers with equal parameter setting, meaning under $\mathcal{F}_{Shape2}$. Again, the RF achieves the highest prediction accuracy within reasonable duration. Since our chosen shape classifier furthermore consists of frame-wise hand shape evaluations, we do not train any spatial content-aware CNN classifier. Among all evaluated shape classifiers, we therefore consider the RF architecture to offer the best balance between computational lightweight and accuracy for the given scenario.

B. RPS PIPELINE EVALUATION

We evaluate the effect of our approach by running a complete RPS game pipeline from segmentation to hand recognition under $RPS_{RF2,S2}$ (RF_{Seg2} and RF_{Shape2}), $RPS_{RF3,S2}$ (RF_{Seg3} and RF_{Shape2}), $RPS_{CNN2,S2}$ (CNN_{Seg2*} and RF_{Shape2}), and $RPS_{CNN3,S2}$ (CNN_{Seg3*} and RF_{Shape2}). For every run, we set the results in relation to a ground truth GT given by manual sequence annotation. We also evaluate a baseline method in which the phase segmentation is obtained by simple threshold-based decision-making utilizing the smoothed absolute palm accelerations shown in Fig. 2. We then combine the proposed phase segments (Vel) with our shape classifier to $RPS_{Vel,S2}$ (Vel and RF_{Shape2}).

Although the RF achieved best performances in the frame-wise segmentation evaluation, the full pipeline results averaged over all actors do not follow this observation as shown in Table 3. Instead, the smoothed CNN predictions obtain even marginally better precision, recall and F1 values on average. From the detailed actor-wise results given in Fig. 7, we can see that the performances of all four RF and CNN pipelines are close to GT with base value 1.0. They furthermore hardly differ among each other. Lastly, all 4 classifier combinations are considerably better than the ones that could be obtained with the minimal system implementation of $RPS_{Vel,Shape2}$.

C. INTERACTION ANALYSIS

We perform a simple preliminary study to assess user impression of the proposed framework utilizing the Haru platform depicted in Fig. 8. The Leap motion is directly connected to a

TABLE 3. RPS pipeline performances (averaged across test actors) as compared to the ground truth defined by manual annotation with value 1.0.

	accuracy	*precision*	*recall*	*F1*	*run time*
GT	1.000	1.000	1.000	1.000	nA.
$RPS_{RF2,S2}$	0.925	0.906	0.916	0.907	24.349 ms
$RPS_{RF3,S2}$	0.931	0.917	0.924	0.917	24.349 ms
$RPS_{CNN2,S2}$	0.912	0.912	0.920	0.912	24.367 ms
$RPS_{CNN3,S2}$	0.926	0.921	0.930	0.922	24.368 ms
$RPS_{Vel,S2}$	0.645	0.698	0.630	0.641	24.215 ms

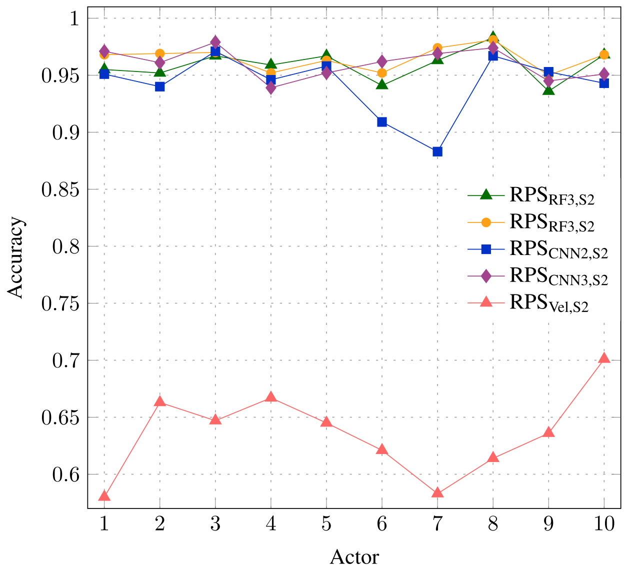


FIGURE 7. Actor-specific prediction accuracies of the full RPS classifications relative to the ground truth GT with value 1.0.

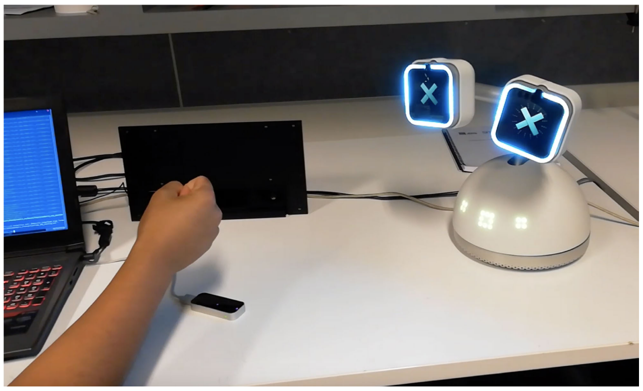


FIGURE 8. The mobile play setup with a single Leap sensor connected to the micro-processor of the robot platform executing all real-time interactions.

LattePanda Alpha 864s single board computer as found inside the robot's body. All motion understanding functionalities with phase segmentation and shape recognition are run on the LattePanda. All 'go' and 'shape' messages are then sent to a behavior tree with repetitive flow for control of the robot action and reaction. The flow follows the game pipeline described in Section V, whereas the threshold for the maximal number of won or lost games is set to 3. After reaching the End Game State, the overall system is restarted from the beginning.

The previous pipeline evaluation could not clearly reveal a best performing RPS system. Therefore, we evaluate both the RF and the CNN-based system set up. For this, we ask ten volunteers (six male, four female, mean age: 41 years with standard deviation 9.81) to freely play RPS for five minutes each under both $RPS_{RF3,S2}$ and $RPS_{CNN3,S2}$. Both configurations served as first play option for five of the ten participants in order to reduce order-induced bias. After completion of both play interactions, we then obtain user feedback via an anonymous questionnaire. The questionnaire constitutes of three short and simple assessments that can be answered within less than five minutes in total.

The first poll consists of one to three questions to determine whether the users show a preference towards any of the system implementations. In concrete, we ask the following questions with their respective set of possible answer selections:

- Did you feel a difference between the two games?
  Choices: Yes/No
- If yes, which of the two games did you prefer?
  Choices: Game 1/Game 2
- If yes, what kind of difference did you experience? Multi selection is possible.
  Choices: Robot response time/Robot response accuracy/Fun of interaction/Other (please specify)

50% of the participants, i.e. five persons, confirm to have experienced a difference in the play interaction, whereas all of them prefer $RPS_{CNN3,S2}$ over $RPS_{RF3,S2}$. Additionally, one participant commented to slightly tend towards $RPS_{CNN3,S2}$ without being sure enough to firm a concrete decision. Out of the potential reasons for preference, robot response time was chosen four times, response accuracy three times, increased fun of the interaction two times and other reasons zero times.

The second poll is composed of four quantitative questions thought to reveal the general user attitude towards the proposed game pipeline. We collect user sentiment on the following aspects of game interaction:

Q1 I think the overall game is fun.
Q2 I think the flow of the interaction is smooth.
Q3 I like the design of the robot reactions.
Q4 If I had a robot, I could imagine to play rock-paper-scissors with it in my free time.

The four statements have to be put into context to personal impression following a 5-point Likert scheme, with 1 denoting disagreement, 2 denoting slight disagreement, 3 denoting neutrality, 4 denoting slight agreement and 5 denoting agreement. Overall user rating is very positive towards the proposed game set up as shown in Fig. 9. The fun factor of the interaction achieves highest ratings with 4.2 points (standard deviation 0.6 points) on average, and the smoothness of interaction achieves lowest ratings with 3.0 points (standard deviation 1.0 points) on average.

The evaluation closes with a short multi-selection field querying the user's opinion on potential points of improvement. This provides us with feedback for future enhancement of the RPS framework, and might suggest reasons for the previous selection. Issues that can be chosen by the participants are the robot response time, the robot response accuracy,

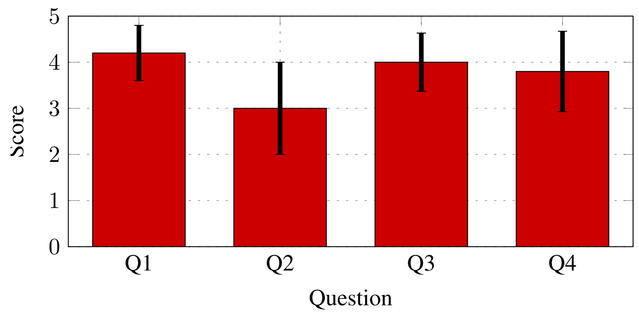


FIGURE 9. **Averaged user responses and their standard deviation to the four overall evaluation aspects Q1, Q2, Q3 and Q4.**

the length of animation (choice 1: make it shorter, choice 2: make it longer), and other issues (to be specified by the participant in written form). Most frequently chosen issues are response time and length of animation, with eight occurrences each. Regarding the latter, seven participants prefer shorter animations, and one participant prefers longer animations. Improved accuracy was selected by five participants, and other reasons four times. Here, the most frequently mentioned issue with three occurrences is the provision of clearer feedback on the information detected by the system. Further issues mentioned by one participant each are an overall too long interaction and too repetitive robot behavior.

VII. DISCUSSION AND FUTURE WORK

Our proposed RPS framework bases on multiple technologies previously developed for related tasks, such as the movement segmentation by Farag and Brock [27]. These were shown to achieve reliable results within their specific sub-domains and suggest the overall reliability of our combined system. To draw a conclusion about the usability and significance of the full framework, we evaluated the game interaction within the natural interaction environment constrained by the available robotic hardware.

Most importantly, we investigated the accuracy and run time of multiple conventional machine learning methods to identify the best setting for subsequent system deployment. On the whole, our evaluations shows that the proposed framework is not only lightweight and mobile, but also reliable and fast for real-time play interaction with a social robot. The hand movement features extracted for both the segmentation and the classification task are universal and can be used with different hand tracking modalities, as well as for the implementation of different hand gesture recognition systems. This makes our proposed system extendable to similar human-robot application scenarios (as e.g. the hand gestural communication discussed in [21]).

Our analysis furthermore suggests that in order to achieve best performance, the chosen classifier should be a RF or a CNN. In between the two classifier, significant differences could not be determined. Whereas the RF classifier appears to be of superior performance with respect to basic data analysis and run time, overall game evaluation indicates higher performance of the CNN classifier. However, more extensive examination, in particular with the support of human players, would be necessary to confirm this observation. With respect to run time aspects, it should furthermore be noted that the reported metrics of the CNN constitute run time measurements obtained from a non-compressed model. Novel extensions of deep learning environments, such as Tensorflow light for mobile devices, considerably speed up the prediction process and might further enhance performance of the system with CNN-based segmentation.

Our preliminary study reveals that users like the game experience of the present framework. However, results should be verified in more extensive studies in the following. In concrete, we plan to conduct studies with a larger number of participants that include clearly defined descriptors of subjective aspects such as robot likeability, and that provide comparison with baseline RPS playmates. Collected user responses indicate that further improvements can be made to make the system more attractive for actual play interactions. For example, the majority of our participants experiences the length of the animated robot behavior routines as too long. Long animations result in unnatural waiting times between games, and also reflect in the comparatively low user scores of Q2. As such, shortening the animations could easily enhance the overall perception of the game flow and its fun factor. The same holds for the addition of information on the detected human action, or similar robot response design aspect.

Enhancement of the RPS play interaction is an iterative process between user evaluation and system adjustment which has just been initiated with the presented first preliminary user assessment. As a next step, we plan to further explore the dynamics behind the proposed RPS play interaction, both in itself, as well as among various types of users, in detailed evaluations. Here, it is important to investigate how robot actions and reactions should be designed and modeled to allow for the most entertaining social interactions. This includes the addition of automatic decision-making within the robot play, as defined by procedural reasoning or reinforced learning for social interaction. Solutions to the previous questions shall then be investigated within extensive Human-Robot-Interaction case studies and qualitative user-based play assessments. We furthermore plan to include psychological research on human RPS play and other aspects of human-robot interaction to maintain interest within the interaction over an extended period of time.

VIII. CONCLUSION

In this paper, we presented a novel framework for a social and entertaining RPS play interaction between a robot and a human player. As opposed to previous systems, the overall design is computationally lightweight and ubiquitous and only relies on a single Leap Motion controller. Data from the Leap controller is used to track kinematic information of the human hand that is then further processed for evaluation by a pre-trained machine knowledge. By separate handling of two different functions, namely the recognition of the initialization of play and the recognition of a played hand shape, the trained machine knowledge can be flexibly applied to different variations of play and user movement. This makes the play pipeline available to various future interaction scenarios. The perception module is fully integrated to be run on tabletop robot Haru, a non-humanoid robot with high emotive expressivity. Detailed analysis and evaluation of multiple machine learning architectures enabled the set up of an accurate, reliable and actor-invariant perception pipeline. The framework can be run in real-time without obvious delay and provide a basic interactive play experience. First user evaluation shows that the collaborative game setting is well perceived. User feedback furthermore indicates points of improvement regarding the flow of interaction and robot behavior design. In the following, the framework should now be refined accordingly, and then be evaluated in more detail with respect to its social interaction potential.

REFERENCES

[1] D. Semmann, H.-J. Krambeck, and M. Milinski, "Volunteering leads to rock–paper–scissors dynamics in a public goods game," *Nature*, vol. 425, no. 6956, pp. 390–393, 2003.

[2] T. Reichenbach, M. Mobilia, and E. Frey, "Mobility promotes and jeopardizes biodiversity in rock–paper–scissors games," *Nature*, vol. 448, no. 7157, pp. 1046–1049, Aug. 2007, doi: 10.1038/nature06095.

[3] Z. Wang, B. Xu, and H.-J. Zhou, "Social cycling and conditional responses in the Rock-Paper-Scissors game," *Sci. Rep.*, vol. 4, no. 1, p. 5830, May 2015.

[4] K. Hoshino and I. Kawabuchi, "Pinching with finger tips in humanoid robot hand," in *Proc. 12th Int. Conf. Adv. Robot.*, Jul. 2005, pp. 705–712.

[5] K. Hoshino, E. Tamaki, and T. Tanimoto, "Copycat hand—Robot hand imitating human motions at high speed and with high accuracy," *Adv. Robot.*, vol. 21, no. 15, pp. 1743–1761, 2007.

[6] E. Short, J. Hart, M. Vu, and B. Scassellati, "No fair!! an interaction with a cheating robot," in *Proc. 5th ACM/IEEE Int. Conf. Human-Robot Interact. (HRI)*, Mar. 2010, pp. 219–226.

[7] A. Litoiu, D. Ullman, J. Kim, and B. Scassellati, "Evidence that robots trigger a cheating detector in humans," in *Proc. 10th Annu. ACM/IEEE Int. Conf. Human-Robot Interact. (HRI)*, 2015, pp. 165–172.

[8] A. Castro-González, J. C. Castillo, F. Alonso-Martín, O. V. Olortegui-Ortega, V. González-Pacheco, M. Malfaz, and M. A. Salichs, "The effects of an impolite vs. a polite robot playing rock-paper-scissors," in *Social Robotics*, A. Agah, J.-J. Cabibihan, A. M. Howard, M. A. Salichs, and H. He, Eds. Cham, Switzerland: Springer, 2016, pp. 306–316.

[9] E. B. Sandoval, J. Brandstetter, and C. Bartneck, "Can a robot bribe a human? The measurement of the negative side of reciprocity in human robot interaction," in *Proc. 11th ACM/IEEE Int. Conf. Human-Robot Interact. (HRI)*, Mar. 2016, pp. 117–124.

[10] T. Kanda, H. Ishiguro, M. Imai, T. Ono, and K. Mase, "A constructive approach for developing interactive humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 2, Sep./Oct. 2002, pp. 1265–1270.

[11] Y. Hasuda, S. Ishibashi, H. Kozuka, H. Okano, and J. Ishikawa, "A robot designed to play the game 'Rock, Paper, Scissors,'" in *Proc. IEEE Int. Symp. Ind. Electron.*, Jun. 2007, pp. 2065–2070.

[12] H. S. Ahn, I.-K. Sa, D.-W. Lee, and D. Choi, "A playmate robot system for playing the rock-paper-scissors game with humans," *Artif. Life Robot.*, vol. 16, no. 2, p. 142, 2011.

[13] C.-Y. Lin, L.-W. Chuang, L.-C. Cheng, and K.-J. Lin, "An interactive finger-gaming robot with real-time emotion feedback," in *Proc. 6th Int. Conf. Autom., Robot. Appl. (ICARA)*, Feb. 2015, pp. 513–518.

[14] K. Ito, T. Sueishi, Y. Yamakawa, and M. Ishikawa, "Tracking and recognition of a human hand in dynamic motion for janken (rock-paper-scissors) robot," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2016, pp. 891–896.

[15] L. Motion. (2019). *Documentation Development SDK*. Accessed: Jan. 23, 2019. [Online]. Available: https://developer.leapmotion.com/documentation/

[16] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with jointly calibrated leap motion and depth sensor," *Multimedia Tools Appl.*, vol. 75, no. 22, pp. 14991–15015, Nov. 2016.

[17] L. E. Potter, J. Araullo, and L. Carter, "The leap motion controller: A view on sign language," in *Proc. 25th Austral. Computer-Human Interact. Conf., Augmentation, Appl., Innov., Collaboration*, 2013, pp. 175–178.

[18] S. Chen, H. Ma, C. Yang, and M. Fu, "Hand gesture based robot control system using leap motion," in *Intelligent Robotics and Applications*, H. Liu, N. Kubota, X. Zhu, R. Dillmann, and D. Zhou, Eds. Cham, Switzerland: Springer, 2015, pp. 581–591.

[19] D. Bassily, C. Georgoulas, J. Guettler, T. Linner, and T. Bock, "Intuitive and adaptive robotic arm manipulation using the leap motion controller," in *Proc. 41st Int. Symp. Robot.* Munich, Germany: VDE, Jun. 2014, pp. 1–7.

[20] L. Peppoloni, F. Brizzi, C. A. Avizzano, and E. Ruffaldi, "Immersive ROS-integrated framework for robot teleoperation," in *Proc. IEEE Symp. 3D User Interfaces (3DUI)*, Mar. 2015, pp. 177–178.

[21] H. Brock, S. Sabanovic, K. Nakamura, and R. Gomez, "Robust real-time hand gestural recognition for non-verbal communication with tabletop robot haru," in *Proc. 29th IEEE Int. Conf. Robot Human Interact. Commun. (RO-MAN)*, Aug. 2020, pp. 891–898.

[22] R. Gomez, D. Szapiro, K. Galindo, and K. Nakamura, "Haru: Hardware design of an experimental tabletop robot assistant," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interact.*, Feb. 2018, pp. 233–240.

[23] R. Gomez, K. Nakamura, D. Szapiro, and L. Merino, "A holistic approach in designing tabletop robot's expressivity," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Aug. 2020, pp. 1970–1976.

[24] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1145–1153.

[25] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Guang Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A framework for building perception pipelines," 2019, *arXiv:1906.08172*. [Online]. Available: http://arxiv.org/abs/1906.08172

[26] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2015.

[27] I. Farag and H. Brock, "Learning motion disfluencies for automatic sign language segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 7360–7364.

[28] I. A. S. Filho, E. N. Chen, J. M. da Silva Junior, and R. da Silva Barboza, "Gesture recognition using leap motion: A comparison between machine learning algorithms," in *Proc. ACM SIGGRAPH Posters*, New York, NY, USA, 2018, pp. 1–2.

HEIKE BROCK (Member, IEEE) received the B.Eng. degree in audiovisual media from Stuttgart Media University, Stuttgart, Germany, in 2008, the M.S. degree in visual computing from Saarland University, Saarbrücken, Germany, in 2011, and the Ph.D. degree in sport informatics from Keio University, Fujisawa, Japan, in 2016.

In December 2016, she joined Honda Research Institute Japan Company, Ltd., as a Scientist. Her research interest comprises the analysis, recognition, and generation of human movement data for the development of non-verbal communication skills of intelligent and autonomous embodied systems.

JAVIER PONCE CHULANI received the B.Eng. degree in electronics, robotic and mechatronic from the University of Seville, Seville, Spain, in 2020.

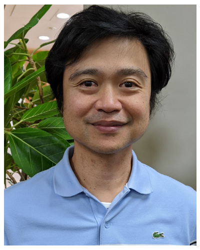
In December 2018, he joined the research group Service Robotics Lab, Universidad Pablo de Olavide. His research involves the development of perception modules for human-robot interaction and the application of behavior trees to social robots.

LUIS MERINO (Member, IEEE) received the M.Sc. degree in telecommunications engineering and the Ph.D. degree in robotics from the University of Seville, Spain, in 2000 and 2007, respectively.

He is currently an Associate Professor with Universidad Pablo de Olavide (UPO), Seville, where he leads the Service Robotics Laboratory. He has been the Vice-Dean of the School of Engineering for five years. His interest includes robot autonomous navigation, including human-aware navigation, decision-making and control for social robotics, planning under uncertainties and multi-robot systems. He has published more than 70 papers and has led UPO's team in several national and international projects on those topics.

Dr. Merino is member of the Spanish Committee on Automation and of the International Socially Intelligent Robotics Consortium. He is Associate Editor of major robotics conferences like IEEE ICRA and IROS conferences, and of the *Image and Vision Computing* journal. His Ph.D. won the ABB Award to the Best Doctoral Dissertation on Robotics in Spain, given by the Spanish Committee of Automation.

DEBORAH SZAPIRO (Member, IEEE) is an award-winning designer, animator, and academic who teaches in the School of Design at the University of Technology Sydney. Her research looks to design and animation's potential as an agent for positive social impact and innovation in relation to new technologies. Her research applies principles of embodied communication and aesthetics to maximize a robot's social presence via emotional and empathetic engagement as well as the design of creative content that supports longitudinal user engagement and learning for specific user scenarios in the home, office, educational, and healthcare environments.



RANDY GOMEZ (Member, IEEE) received the M.Eng.Sci. degree in electrical engineering from the University of New South Wales (UNSW), Australia, in 2002, and the Ph.D. degree from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2006. He pursued postdoctoral works at Kyoto University through the Japan Society for the Promotion of Science (JSPS) Fellowship. He is currently a Senior Scientist with the Honda Research Institute Japan. He oversees the research in embodied communication, which explores the synergy between communication and interaction in order to create meaningful experiences with embodied agents.