Computer Vision and Image Understanding 149 (2016) 65–77

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

A real-time Human-Robot Interaction system based on gestures for assistive scenarios

Gerard Canal[a,b,d,*], Sergio Escalera[c,a], Cecilio Angulo[b]

[a] Computer Vision Center, Campus UAB, Edifici O, 08193 Bellaterra (Cerdanyola), Barcelona, Spain
[b] Dept. Automatic Control, UPC - BarcelonaTech, FME Building, Pau Gargallo 5, 08028 Barcelona, Spain
[c] Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via de les Corts Catalanes 585, 08007 Barcelona, Spain
[d] Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028 Barcelona, Spain

ARTICLE INFO

Article history:
Received 15 April 2015
Accepted 2 March 2016
Available online 10 March 2016

Keywords:
Gesture recognition
Human Robot Interaction
Dynamic Time Warping
Pointing location estimation

ABSTRACT

Natural and intuitive human interaction with robotic systems is a key point to develop robots assisting people in an easy and effective way. In this paper, a Human Robot Interaction (HRI) system able to recognize gestures usually employed in human non-verbal communication is introduced, and an in-depth study of its usability is performed. The system deals with dynamic gestures such as waving or nodding which are recognized using a Dynamic Time Warping approach based on gesture specific features computed from depth maps. A static gesture consisting in pointing at an object is also recognized. The pointed location is then estimated in order to detect candidate objects the user may refer to. When the pointed object is unclear for the robot, a disambiguation procedure by means of either a verbal or gestural dialogue is performed. This skill would lead to the robot picking an object in behalf of the user, which could present difficulties to do it by itself. The overall system — which is composed by a NAO and Wifibot robots, a Kinect™ v2 sensor and two laptops — is firstly evaluated in a structured lab setup. Then, a broad set of user tests has been completed, which allows to assess correct performance in terms of recognition rates, easiness of use and response times.

1. Introduction

Autonomous robots are making their way into human inhabited environments such as homes and workplaces: for entertainment, helping users in their domestic activities of daily living, or helping disabled people in personal care or basic activities, which would improve their autonomy and quality of life.

In order to deploy such robotic systems inhabiting unstructured social spaces, robots should be endowed with some communication skills so that users can interact with them just as they would intuitively do, eventually considering a minimal training. Besides, given that a great part of the human communication is carried out by means of non-verbal channels [1,2], skills like gesture recognition and human behavior analysis reveal to be very useful for this kind of robotic systems, which would include viewing and understanding their surroundings and the humans that inhabit them.

Gesture recognition is an active field of research in Computer Vision that benefits from many machine learning algorithms, such as temporal warping [3–5], Hidden Markov Models (HMMs), Support Vector Machines (SVMs) [6], random forest classifiers [7] and deep learning [8], just to mention a few of them. Moreover, gesture recognition personalization techniques have also been proposed in [9] to adapt the system to a given user. Studies in Human Computer Interaction (HCI) and more specifically Human Robot Interaction (HRI) take advantage of this field. Hence, many recent contributions [10–14] consider Kinect™-like sensors to recognize gestures given the discriminative information provided by multimodal RGB-Depth data. A Kinect™ based application is introduced in [15] for taking order service of an elderly care robot. Static body posture is analyzed by an assistive robot in [16] to detect whether the user is open towards the robot interaction or not. Communicative gestures are contrasted from daily living activities in [17] for an intuitive human robot interaction. A novice user can generate his/her gesture library in a semi-supervised way in [18], which are then recognized using a non-parametric stochastic segmentation algorithm. In [19], the user can define specific gestures that mean some message in a human-robot dialogue, and in [20] a framework to define user gestures to control a robot is presented. Deep neural networks are used in [21] to recognize gestures in real time by considering only RGB information. Pointing gestures, similar to the one we propose in this paper, have been studied mostly focusing

* Corresponding author.
E-mail addresses: gerard.canal@cvc.uab.cat (G. Canal), sergio@maia.ub.es (S. Escalera), cecilio.angulo@upc.edu (C. Angulo).

http://dx.doi.org/10.1016/j.cviu.2016.03.004
1077-3142/

in hand gestures [22], using the hand orientation and face pose [23]. The pointing direction is estimated in [24,25] using gaze and finger orientation, and deictic gesture interactions that people use to refer to objects in the environment are studied in [26]. Related pointing interactions have also been used for robot guidance [27].

In this work we introduce a real time Human Robot Interaction (HRI) system whose objective is to allow user communication with the robot in an easy, natural and intuitive gesture-based fashion. The experimental setup is composed by a humanoid robot (Aldebaran's NAO) and a wheeled platform (Wifibot) that carries the NAO humanoid and a Kinect™ sensor. In this set-up, the multi-robot system is able to recognize static and dynamic gestures from humans based on geometric features extracted from biometric information and dynamic programming techniques. From the gesture understanding of a deictic visual indication of the user, robots can assist him/her in tasks such as picking up an object from the floor and bringing it to the user. In order to validate the system and extract robust conclusions of the interactive behavior, the proposed system has been tested in offline experiments, reporting high recognition rates, as well as with an extensive set of user tests in which 67 people assessed its performance.

The remainder of the paper is organized as follows: Section 2 introduces the methods used for gesture recognition and Human Robot Interaction. Section 3 presents the experimental results including the offline and user tests and, finally, Section 4 concludes the paper.

2. Gesture based Human Robot Interaction

With the aim to study gestural communication for HRI, a robotic system has been developed able to understand four different gestures so a human user can interact with it: wave (hand is raised and moved left and right), pointing at (with an outstretched arm), head shake (for expressing disagreement) and nod (head gesture for agreement).

The overall robotic system involves several elements: an Aldebaran's NAO robot, a small size humanoid robot which is very suitable to interact with human users; a Microsoft's Kinect™ v2 sensor to get RGB-Depth visual data from the environment and track the user; and, given that the vision sensor exceeds NAO's robot capabilities (in size and computing performance), a Nexter Robotics' Wifibot wheeled platform is used to carry the sensor as well as the NAO, easing its navigation and precision at long ranges.

In fact, the proposed robotic system takes inspiration from the DARPA Robotics Challenge 2015[1] in which a humanoid robot should drive a car towards an interest place and exit the car in order to finish its work by foot. In a similar way, the wheeled robot was added to the system in order to carry the sensor along with the little humanoid, which should also exit it to complete its task by walking. This multi-robot setup allows the NAO to use the information from the Kinect's™ v2 sensor and eases its navigation. And for its side, the NAO is the one in charge of directly interacting with the user, also being able to act on the environment, for instance, by grasping objects. The overall setup is shown in Fig. 1, with the NAO seated on the Wifibot. The setup also includes a laptop with an Intel i5 processor to deal with Kinect™'s data and another Intel Core 2 Duo laptop, which sends commands to the robots using the Robot Operating System (ROS)[2] [28]. The depth maps are processed using the Point Clouds Library (PCL)[3] [29], and body tracking information is obtained using the Kinect™ v2 SDK.

[1] theroboticschallenge.org
[2] ros.org
[3] pointclouds.org

Fig. 1. The robotic system designed for this work.

The system has been programmed as an interactive application, and tested with several users of different ages and not related with the robotics world (see Section 3.2).

2.1. Real time gesture recognition: Interaction with a robot

This section explains the methods used to perform the gesture recognition and image understanding. Given that the application of the system is to enhance the interaction between a human user and a robot, the defined gestures should be as natural for the user as possible, avoiding user training or learning of a specific set of gestures. Instead, the robot should understand gestures as a human would understand another human's gestures, and should reply to that visual stimulus in real time.

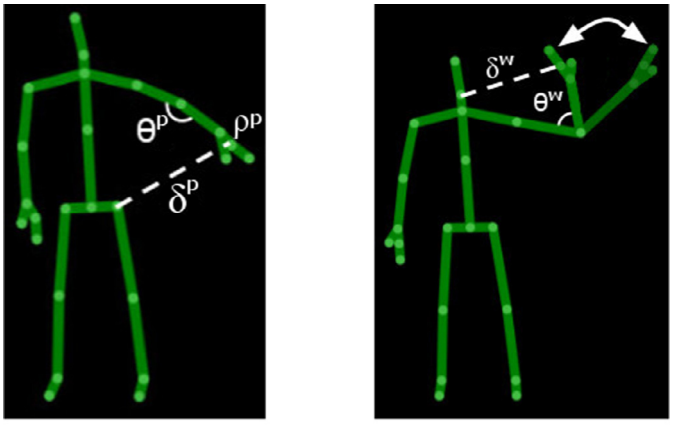
The considered set of human gestures has been divided into two categories, depending on the amount of movement involved in their execution:

- Static gestures are those in which the user places his/her limbs in a specific position and stands for a while, without any dynamics or movement involved. In this case, the transmitted information is obtained through the static pose configuration. Pointing at an object is an example of static gesture.
- Dynamic gestures are, in contrast, those in which the movement is the main gesture's feature. The transmitted information comes from the type of movement as well as its execution velocity. It may also contain a particular pose for a limb during the movement. Examples of dynamic gestures are a wave to salute someone or a gesture with the hand to ask someone to approach to the user's location.

Four different gestures have been included in the designed system to interact with the robot, being three of them dynamic and the remaining one static. The dynamic gestures are the wave, the nod and a facial negation gesture. The static one is the pointing at an object. Both categories are tackled using different approaches. Next we describe the extracted features, the gesture recognition methods and how the gesture's semantic information is extracted.

2.1.1. Definition of gesture specific features

Gesture recognition is performed based on some features extracted from the user body information obtained from depth maps.



(a) Point at gesture features.

(b) Wave gesture features and dynamics.

Fig. 2. Skeletal gesture features.

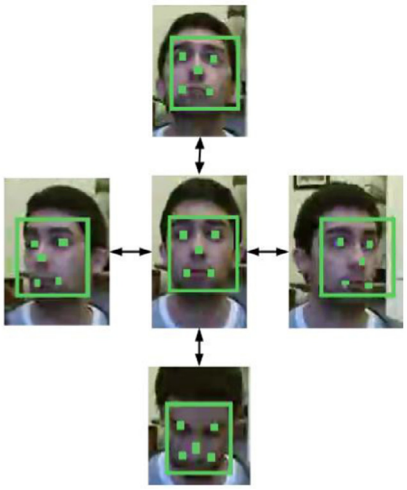


Fig. 3. Facial gesture features and dynamics. The vertical arrows represent the nod gesture and the horizontal ones the negation.

For the included arm gestures or any possible new gestures involving more body parts, skeletal data is obtained from depth images of the Kinect™ sensor using the Kinect™ SDK v2.0.

Given that a limb gesture such as the wave does not depend on the position of other parts of the body such as the legs, the rest of the body is not taken into consideration when the recognition is performed. So, rather than directly using the joint coordinates of the whole body, as in [4,30], our proposed method only takes into account the involved limbs from which some distinctive features are extracted. This approach allows the system to recognize gestures any time the skeletal data is properly tracked from the sensor, including situations such as sitting (for instance a person in a wheelchair), as well as standing up or crouching.

The application is able to recognize four gestures: the pointing at, the wave, the nod and the head negation. The point at gesture's features on the skeleton are displayed in Fig. 2a. They can be described as:

- δ^p, the Euclidean distance between the hand and hip joints of the same body part. This feature discriminates between the pointing position and the resting one in which the arms may be outstretched at the sides of the body but not pointing at a place.
- θ^p, the elbow joint angle, defined as the angle between the vector from the elbow joint to the shoulder one and the vector from the elbow to the hand joint. It defines when the arm is outstretched.
- ρ^p, the position of the hand joint.

Given the presented setup and the overall structure of the robotic system, the above features only accounts for large pointing gestures (with the full arm extended), as the ones one would use to point at something laying on the ground.

The features and dynamics for the wave gesture are shown in Fig. 2b. They are defined as:

- δ^w, the Euclidean distance between neck and hand joints. Although it was not necessary in order to perform the tests with the current set of gestures, this measure could be normalized by dividing it by the longitude of the arm to have a standardized value in the range [0, 1] to handle body variations.
- θ^w, the elbow joint angle, as defined in the point at gesture.

The elbow angle used in the features above does not require from normalization as it is not affected by different body heights.

The orientation of the face provided by the sensor is used to describe the nod gesture (vertical movement of the head) and the negation one (horizontal movement of the head). The three usual angular axes — pitch, roll and yaw — are used but instead of taking the absolute values, its derivatives are employed as frame features, $\Delta O_{i,a} = O_{i,a} - O_{i-1,a}$, where $O_{i,a}$ is the orientation in degrees of the face in the frame i according to the a axis. Moreover, one out of F frames is used to compute the features to filter noisy orientation estimations, and the values are thresholded to a given value D in order to end up with a sequence of directional changes. More formally, the feature of a frame i for the axis a, $f_{i,a}$, is computed as:

$$f_{i,a} = (|\Delta O_{i,a}| \geq D) \cdot sign(\Delta O_{i,a}). \tag{1}$$

Fig. 3 depicts the facial gestures.

2.1.2. Dynamic gesture recognition

A Dynamic Time Warping (DTW) [31] approach is used to detect the dynamic gestures. The DTW algorithm matches two temporal sequences finding the minimum alignment cost between them. One sequence is the reference gesture model of the gesture g, $\mathbf{R}_g = \{\mathbf{r}_1, \ldots, \mathbf{r}_m\}$, and the other is the input stream $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_\infty\}$, where $\mathbf{r}_i$ and $\mathbf{s}_i$ are feature vectors. Features will depend on the gesture to be recognized: for the wave, $\mathbf{r}_i = \{\delta_i^w, \theta_i^w\}$ and $\mathbf{r}_i = \{f_{i,pitch}, f_{i,roll}, f_{i,yaw}\}$ for the facial gestures. Both sequences are aligned by means of the computation of a $m \times n$ dynamic programming matrix $\mathbf{M}$, where n is the length of the temporal window being used to discretize the infinite time, as data keeps entering the system while no gesture has been identified. Provided that gesture spotting is not needed, the minimum value for n is two.

Each element $m_{i,j} \in \mathbf{M}$ represents the distance between the subsequences $\{\mathbf{r}_1, \ldots, \mathbf{r}_i\}$ and $\{\mathbf{s}_1, \ldots, \mathbf{s}_j\}$, so it is computed as:

$$m_{i,j} = d(\mathbf{r}_i, \mathbf{s}_j) + min(m_{i,j-1}, m_{i-1,j}, m_{i-1,j-1}), \tag{2}$$

where $d(\cdot, \cdot)$ is a distance metric of choice. Different distance metrics can be used in our implementation. For instance, the Hamming distance:

$$d_H(\mathbf{r}_i, \mathbf{s}_j) = \sum_{k=0}^{o} \{r_i^k \neq s_j^k\}, \tag{3}$$

with o being the number of features of the gesture, is used for the facial gestures case. The weighted $L1$ distance is employed for the case of the wave gesture, computed as:

$$d_{L1}(\mathbf{r}_i, \mathbf{s}_j) = \sum_{k=0}^{o} \alpha_k |r_i^k - s_j^k|, \tag{4}$$

with α_k a positive weighting constant.

A gesture g will be considered as recognized if a subsequence of the input data stream $\mathbf{S}$ is similar enough to the reference sequence $\mathbf{R}_g$:

$$m_{m,k} \leq \mu_g, \ \forall k, \tag{5}$$

where μ_g is obtained using a training method for each gesture g, detailed in Section 3.1.1.

In order to assure the fulfillment of the real time constraint, the DTW is executed in a multi-threaded way in which the different gestures are spread between different threads that run the gesture recognition method simultaneously, stopping in case one of the methods finds a gesture in the input sequence.

In case of the need of properly segmenting the gesture in a begin-end manner, such as for validation purposes, the warping path can be found to locate the beginning of a gestural sequence. This warping path:

$$\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_T\}, \tag{6}$$

with $max(m, n) \leq T < m + n + 1$, is a matrix of pairs of indexes of contiguous elements in the matrix $\mathbf{M}$ that define a mapping between the reference gesture $\mathbf{R}_g$ and a subsequence of the input sequence $\mathbf{S}$, subject to the following constraints:

- $\mathbf{w}_1 = (1, j)$ and $\mathbf{w}_t = (m, j')$.
- for $\mathbf{w}_{t-1} = (a', b')$ and $\mathbf{w}_t = (a, b)$ then $a - a' \leq 1$ and $b - b' \leq 1$.

The warping path $\mathbf{W}$ that minimizes the warping cost:

$$C_{\mathbf{w}}(\mathbf{M}) = \min_{\mathbf{w} \in \mathbf{W}} \left\{ \frac{1}{T} \sqrt{\sum_{t=1}^{T} \mathbf{M}_{\mathbf{w}_t}} \right\}, \tag{7}$$

can be found for the matrix $\mathbf{M}$ by backtracking of the minimum path from $m_{m,j}$, to $m_{1,k}$, being k the starting point of the segmented gesture and j the ending of it.

2.1.3. Static gesture recognition

A static approach has been selected for static gesture recognition, in the sense that a gesture is considered as recognized when features are within certain values for a given number of contiguous frames and small movement is involved. The number of frames and the feature thresholds are obtained through a similar training method as for the dynamic case.

In our case, the pointing gesture is recognized when, for a certain number of frames F, the elbow angle is greater than a threshold T_{ea} indicating the arm is outstretched and the distance between the hand and the hip is greater than a certain distance T_d meaning that the arm is not in the resting position. Moreover, the hand coordinates are used in order to check the constraint that the position is hold still and not moving. That is, a gesture is recognized if the following constraints are held during F_p frames:

$$\delta_i^p > T_d, \ \theta_i^p > T_{ea}, \ d_E(\rho_i^p, \rho_{i-1}^p) \approx 0, \tag{8}$$

where d_E represents the Euclidean distance.

The system runs the static gesture recognition in parallel with the dynamic one, in a multi-threaded way.

2.1.4. Pointed location estimation

Once a pointing gesture has been recognized, some information needs to be extracted from it in order to perform its associated task and help the user. The main information that this deictic gesture gives is the pointed location, which is the region of the surrounding space that has some elements of interest for the user. To estimate it, a floor plane description, the pointing direction and some coordinates belonging to the ground are needed.

First of all, the arm position has to be obtained in order to know the pointing direction. To do so, the arm joints of the last ten frames of the gesture are averaged to obtain the mean direction and avoid tracking errors. Then, the coordinates of the hand joint H and the elbow joint E are used to get the pointing direction as the $\overrightarrow{EH} = H - E$ vector. Even though the Kinect™ v2 sensor provides information about the hand tip joint, the direction provided by the elbow to hand vector proved to be more precise than the hand to hand tip one in preliminary tests.

The ground plane is extracted using the plane estimation method of the PCL library [32]. A depth image of the Kinect™ is obtained and converted to a point cloud, the planes of which are segmented using a Random Sample Consensus (RANSAC) method [33]. Those planes that have a similar orthogonal vector to a reference calibrated plane are used as floor planes. The reference plane is automatically obtained at system start up by segmenting all the planes in the depth image and keeping the parameters of the plane whose orthogonal vector is the same as the vertical axis (y axis) of the sensor. In case the camera is not in a parallel position with the ground or no plane is found which fulfills this condition, the reference plane is obtained from the user who has to click three points of the ground in the graphical interface, from which the plane is estimated. Then, the ground point coordinates are obtained by picking one element from the floor cloud.

Therefore, let P_f be the ground point and $\vec{N}_f = (A, B, C)$ the orthogonal vector of the floor plane $\pi_f = Ax + By + Cz + D = 0$, the pointed point P_p can be obtained by:

$$P_p = H + \frac{(P_f - H) \cdot \vec{N}_f}{\overrightarrow{EH} \cdot \vec{N}_f} \cdot \overrightarrow{EH}. \tag{9}$$

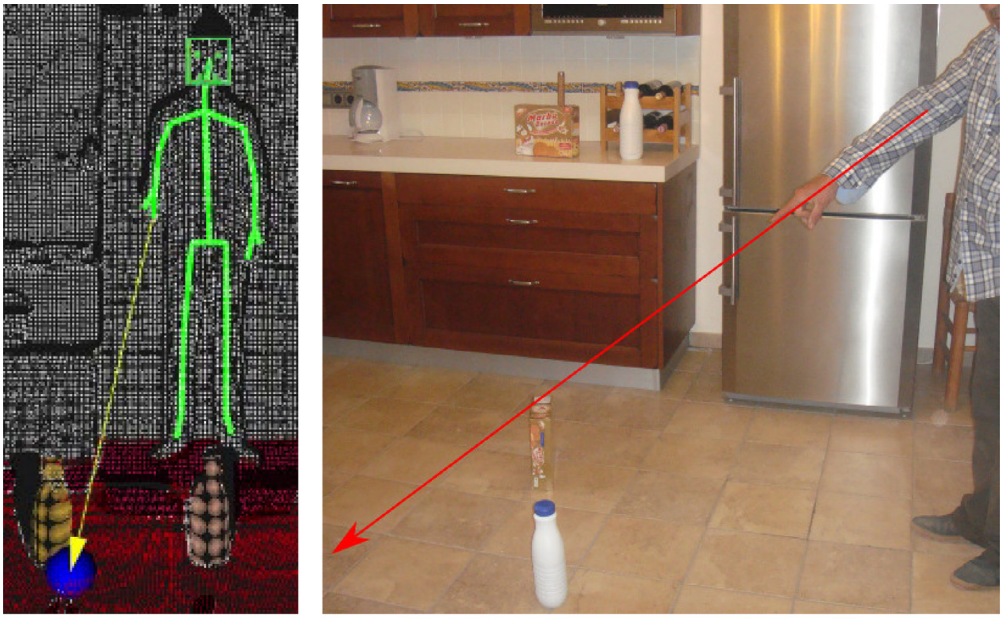
An example of the pointing location estimation is shown in Fig. 4a.

After some tests with users, we observed that the bones were correctly tracked by the Kinect™ sensor but not precisely enough to get an accurate pointing direction. This was more clear when the pointing gesture was performed with the hand in front of the body. Also, the users tended to actually point farther than the objects' location, and the real pointed line did not intersect with the objects, as it can be observed in Fig. 4b. In order to deal with this imprecision, we corrected the pointing position just by slightly translating the pointed location backwards.

2.1.5. Near point object segmentation and disambiguation

Similar to what humans do as a response to a pointing gesture, we want that the robots look at the surroundings of the estimated pointed location to detect possible objects that the user is referring to. Notice that in our case we do not care about recognizing the actual objects but rather detecting their presence.

This is performed by first extracting the set of points X from the scene point cloud in which each $\mathbf{x}_i \in \mathbf{X}$ is selected such that its Euclidean distance d_E to the pointed point is smaller than a certain value r, $d_E(\mathbf{x}_i, P_p) \leq r$, being $\mathbf{X}$ a spherical point cloud of radius r and centered in the pointed point P_p. After the extraction of the floor plane, $\mathbf{Z} = \mathbf{X} \setminus \{\mathbf{x}_i \mid \mathbf{x}_i \in \pi_f\}$, all the objects should be isolated and a clustering algorithm is applied to the sub point cloud $\mathbf{Z}$ in order to join all the points of the same objects in a smaller point cloud per each object. The clustering algorithm that has been used is the Euclidean Cluster Extraction method [32], which starts the clustering by picking a point $\mathbf{z}_i \in \mathbf{Z}$ and joining to it all its neighbors $\mathbf{z}_j \in \mathbf{Z}$ such that $d_E(\mathbf{z}_i, \mathbf{z}_j) < d_{th}$, being d_{th} a user defined threshold. The process is repeated for all of these neighbors until no more points are found, in which case a cluster C_i is obtained. The remaining points of the cloud $\mathbf{Z}$ are processed in the same way to get the other object clusters. Once the objects are found, its centroid point is computed as the mean coordinates of all the points of the cluster, $\frac{1}{|C_i|} \sum_{\mathbf{z} \in C_i} \mathbf{z}$, and then each cluster's convex hull is reconstructed in order to compute its area. This



(a) Pointing location estimation.

(b) Example of user pointing deviation.

Fig. 4. Examples of the point at gesture.

allows the system to get a notion of its position in the space and size (see Fig. 4a).

However, it may be the case in which the pointed location is not clearly near a single object, so there is a doubt on which was the referred one. When this situation arises, a spoken disambiguation process is started in which the robot asks the user about the object. To do so, the robot may ask if the person was pointing at the biggest object if the objects are clearly of different sizes, otherwise it asks about its relative position, for instance asking a question like "is it the object at your right?". The user can respond to the question with a yes or no utterance, recognized using NAO's built in speech recognition software, or by performing the equivalent facial gestures, and the robot will know which was the referred object if there were only two of them, or it may ask another question in case there were three dubious objects in sight. A flowchart of the disambiguation process is included in the supplementary material.

2.2. Robotics interaction with the human

The gesture recognition makes the robotic system able to understand some human gestures. But, the human user must be able to recognize what is the robot doing for the interaction to be successful and pleasant. In our case, this means that the robots must work together in order to fulfill the task and respond to the user in an appropriate way. For instance, the Wifibot is able to perform a more precise navigation, whereas the NAO is ideal to interact and speak to the user as well as to act on the environment. This means that the answer of the system to a visual stimuli made by the person has to be expected for them, thus being a natural response to the gesture. Fig. 5 shows the flow of the application in a normal use case. The application has been programmed using a state machine paradigm to control the workflow. Details of the implemented state machines are shown in the supplementary material.

For the wave gesture, the expected response would be waving back to the user, performing a similar gesture to the one made by him/her and maybe performing some utterance. In the case of the pointing gesture, the robot has to approach the pointed location and analyze which objects are present, trying to deduce which object was the user referring to. Notice that there is no need that the user points to a place which is in the field of view of the sensor, being it possible to point at some objects which are farther away

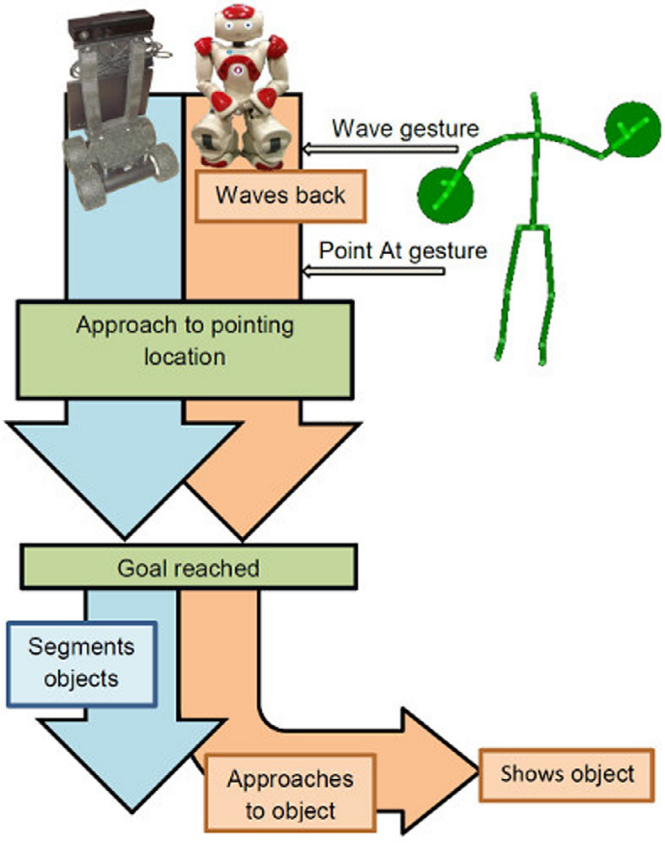


Fig. 5. Example of application's use case.

Fig. 6. NAO's going down of the Wifibot to approach the object.

Fig. 7. NAO showing the pointed object.

which will also make the robot go to the pointed location to check for objects.

Once the object is known and has been disambiguated in case of doubt, the NAO goes down the Wifibot (Fig. 6) and approaches the object, which is then shown to the user performing a gesture with the hand and the head to expose that it understood the object correctly, as it can be seen in Fig. 7. Note that this could be extended to grasp the object and bring it to the user.

3. Experimental results

In order to evaluate the designed system, several experiments were carried out, including offline evaluation of the methods and online evaluation of the whole system with an extensive set of user tests.

3.1. Offline evaluation

The gesture recognition methods were evaluated in an offline setting in order to validate the performance of the methods and tune a set of parameter values. Hence, a small data set "HuPBA sequences" was generated and labeled. It includes 30 sequences of 6 different users (5 sequences per user) in which each of them performs the four gestures that the system is able to recognize, as well as another arbitrary gesture of their choice; all of them performed in a random order. The gesture models used in the dynamic gesture recognition module were specifically recorded for this purpose from one user performing the gesture in an ideal way. This ideal way was taken from the observations of the recorded sequences, and also taking into account observation of other gesture based systems and quotidian interaction with people. This model subject is not part of the subjects in the data set.

In order to evaluate the system, two metrics usually used in this domain have been adopted: the Jaccard index (also known as overlap) and defined as $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$, and the F1 score, which is computed as $F1_{score} = \frac{2TP}{2TP+FP+FN}$.

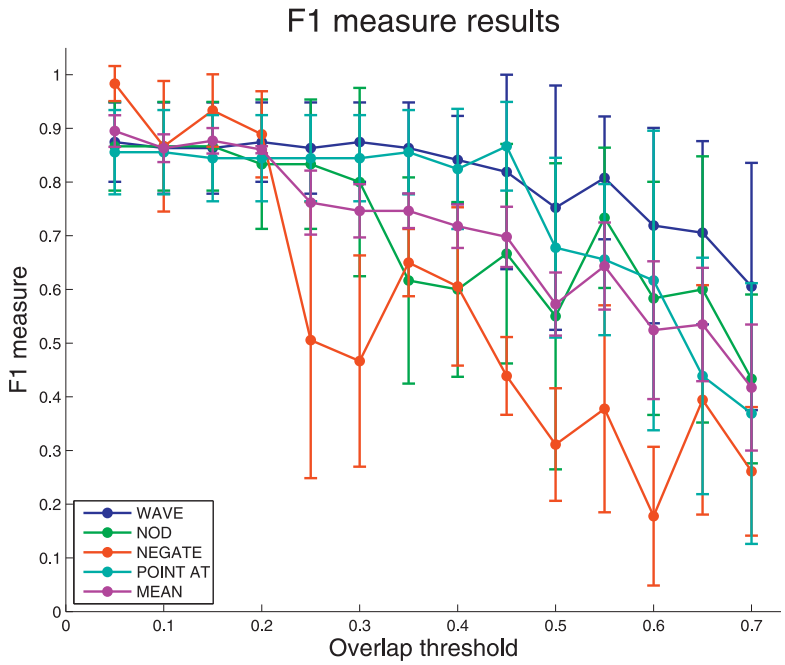
3.1.1. Parameters and evaluation results

In order to compute the performance measure, a Leave-One-Subject-Out cross validation (LOSOCV) technique has been used. In it, a subject of the data set is left out and a grid search is performed in order to tune the best parameters for the different methods and gestures of the system. Then, those parameters are used with the sequences of the left out user and the performance metrics are obtained. This procedure is repeated with all the subjects and their results are averaged for every subject and sequence in order to obtain the final score.

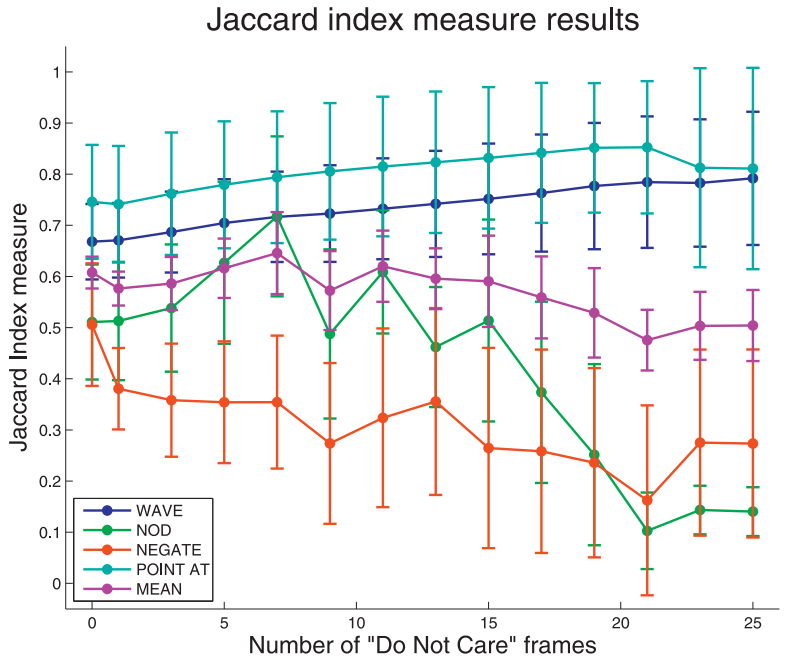
To carry out the parameters tuning, an interval of values for each of them is tested against the set of recordings, keeping those which perform better. The interval of parameters that has been used and tested includes the DTW thresholds $\mu_{wave} \in [6.75, 9.5]$, considering equally distributed values with step 0.25, $\mu_{nod} = \mu_{negate} \in [4.5, 20]$ with step 0.5. The distance weights for the wave gesture were $\alpha \in [0.1, 0.55]$ with step 0.05. The facial gesture's parameters tested were orientation derivative threshold $D \in [5, 30]$ with step 5 and number of frames between samples $F \in [1, 20]$ with increments of 1 unit. For the static gestures, the thresholds and number of frames were $T_d \in [0.1, 0.45]$ with step 0.5 and $T_{ea} \in [2.0, 2.55]$ with a stepping of 0.05. Those ranges were chosen empirically by performing some initial tests using some sequences which included variations in the gestures, recorded for this purpose.

Fig. 8 shows the obtained results with the standard deviation of the different users. Fig. 8a plots the results for the F1 measure with different overlap thresholds to decide which amount of overlapping is enough to be considered a *TP*. Meanwhile, Fig. 8b shows the results using the Jaccard index measure with different number of "Do not care" frames.

As it can be observed, the wave and the point at gestures are the ones which have better recognition rates, being the point at slightly better according to the Jaccard index. As for the facial gestures, the nodding presents a better performance than the negation in both measures. The facial gestures present a worse performance due to the fact that many users perform the gestures very subtly and with different lengths that vary in a considerable way in terms of orientation. It also gets hampered by the distance from the user to the camera as the orientation values are more subtle the farther the user is. Even though, we get a LOSOCV F1 score of 0.6 ± 0.61 (mean $\pm$ standard deviation of the LOSO subjects) for the nod gesture and 0.61 ± 0.15 for the negation one with an overlap threshold of 0.4, which have resulted to be acceptable to get a natural interaction in the real time system.



(a) F1 measure results.



(b) Overlap (Jaccard Index) measure results.

Fig. 8. Offline performance evaluation results.

Focusing on the Jaccard index plot from Fig. 8b, it can be observed that the best mean performance is obtained when 7 "Do Not Care" frames are used, reaching a 0.65 ± 0.07 of overlap. The use of "Do Not Care" frames to compute the Jaccard index makes sense in natural interaction applications because the goal is not to segment the gesture at frame level but to detect the gesture itself, despite which frame the detection started or ended. The use of 7 frames (the three previous to the beginning, the beginning frame and the three after it) is enough to solve any temporal difference between the detection and the labeled data.

3.2. User tests evaluation

In order to evaluate the system's performance, it was tested with different users in a real scenario. Their opinion was collected and use easiness was considered according to the need of external intervention from our part for the communication.

The test users were selected from different age groups and education backgrounds, who might have never seen a humanoid robot before, to analyze their behavior and check the task fulfillment. The tests took place in different environments, trying to keep users

Table 1
Numerical user's answers to the survey (to answer with a number from 1 to 5).

Question	Min	Max	Mean ± SD 9–34 years	35–60 years	61–86 years
Wave's response speed	1	5	3.79 ± 0.74	3.89 ± 0.90	4.00 ± 1.05
Point at's response speed	1	5	3.66 ± 0.91	3.88 ± 1.02	4.00 ± 1.41
Figured out the pointed object	1	5	4.00 ± 1.16	3.76 ± 1.09	3.55 ± 1.75
NAO clearly showed its guess	1	5	4.32 ± 0.97	4.12 ± 0.99	3.82 ± 1.72
Naturalness of the interaction	2	5	3.57 ± 0.63	3.53 ± 1.00	4.14 ± 0.90

in known and comfortable scenarios, including two high schools, a community center and an elderly social association. A total of 67 users participated in the experiments.

The screenplay for the tests is as follows: the user stands in front of the robotic system and two or three objects are placed on the ground, around three meters far. The user first waves to the robot, then points at an object of their election, answering with a facial gesture if the robot asks a question to disambiguate. Otherwise, the users were asked to perform some facial gestures at the end of the test. The procedure was usually repeated twice by each user, and they had to fill in a questionnaire about the experience at the end. A video showing an execution example of the system is included as supplementary material.

The objects were two milk bottles and a cookie box, and the gesture recognition parameters were obtained by using the training mechanism previously explained, but this time all the "HuPBA sequences" were used for the tuning of the parameters. As for the object cluster extraction, a radius of 55 centimeters around the pointed location was used, which was a suitable value for the used objects. Fig. 9 shows some of the users performing the tests in the different environments.

3.2.1. User's survey analysis

This section highlights some interesting results which were obtained from users' questionnaire after the test. Results are analyzed in three age groups. Fig. 10 shows some bar plots of the most relevant questions, aggregated by age group. Table 1 includes some of the answers to numerical questions in the questionnaires.

In summary, users aged from 9 to 86 years, average being 34.8 ± 23.98. They have been divided into three groups: 9 to 34, 35 to 60 and 61 to 86 years, being the youngest user of the last group aged 71. The gender was quite balanced, being 55% of the users males, as seen in Fig. 10a. Moreover, they had zero or very small previous contact with any kind of robots.

The wave gesture was agreed to be natural by most of the users, in all the age groups, even though some users had problems to reproduce it and needed some explanation as they would have waved in another way. The response they obtained from the robot was the one they would expect and was considered quick enough, which means that the robot acted in a natural way and they did not need help to understand the response it gave, as seen in Fig. 10b, c and in Table 1. The results for the point at gesture are quite similar, being it natural and quite fast with equivalent results in the different age groups, even though some users expected the robot to do something with the objects such as grasping or opening a bottle (Fig. 10d, e). Moreover, most of the users thought the pointing time was enough but a 35% of the users felt it was too much time (although some of them kept pointing at the object once the robot said the gesture was already recognized), as shown in Fig. 10f. As for NAO's response, the robot missed the right object in a very few cases, but they thought it clearly showed which object the robot understood without ambiguities, as seen in Table 1.

The facial gestures were not performed by all the users, but again most of them felt comfortable doing them, being the nod too exaggerated for some of them. In fact, 46% of the people from the youngest group that made the nod gesture felt it was unnatural or too exaggerated, as shown in Fig. 10g. The negate gesture had similar response (see Fig. 10h). In general, facial gestures presented a disadvantage with long haired people in which the hair covered the face while performing them (specially in the negation case), which implied that the face tracker lost the face and the gesture was not recognized. The 88% of the users thought that it was easy to answer the yes/no questions to the system.

Finally, the overall interaction was felt quite natural, as seen in Table 1, and not too much users felt frustration due to the system misunderstanding of gestures, as it can be seen in Fig. 10i. Some users did not know what was the robot doing at some moment of the test as shown in Fig. 10j, but most of these cases were due to the language difficulty, as the robot spoke in English[4]. Hence, the 36% of the users did not speak English and they needed external support and translation. The 92% the users stated that they enjoyed the test (100% of the elderly group did), and a vast majority of the users thought that applications of this kind can be useful to assist people in household environments, specially the elder ones or those with reduced mobility, as depicted in Fig. 10l. Moreover, almost all of them thought it was easy to communicate a task in a gesture manner, as Fig. 10k shows. In the last question they were asked about possible gesture additions to the system. The most interesting responses include gestures to call it to come back, start, stop or indicate the NAO to sit again on the Wifibot.

Table 2
Response and execution times and recognition rates for the different gestures and the object detection in 30 tests. The dynamic gesture recognition times span from the end of the gesture to the system response, and the static ones from the start of the gesture to the object response. The gesture times were measured using a standard chronometer operated by the test controller.

Item	Time (seconds) Mean ± SD	Recognition rate
Wave gesture	1.72 ± 0.62	83.33%
Point at gesture	1.91 ± 0.67	96.67%
Nod gesture	1.99 ± 0.49	73.33%
Negate gesture	1.47 ± 0.47	33.33%
Object detection	0.53 ± 0.29	63.33%

Table 3
Error rates by cause in the object detection step for 30 tests.

Cause	Rate
Wrong pointing location estimation	3.33%
Object not detected or wrong object detected	16.67%
Disambiguation failure	3.33%
Navigation error (did not reach the place)	13.33%

3.3. System times and recognition rates

In order to obtain objective evaluation metrics, 30 additional tests performed by six users (five gestures per user) were conducted. The response times of the different gestures along with recognition rates, as well as the execution times of the object detection module were extracted from them. Tables 2 and 3 show the obtained results.

As it can be seen, the response times in Table 2, which span from the end of the gesture to the start of the robot response, are quite suitable for a natural interaction, being all the gestures answered in less than two seconds in average. As for the object detection, comprising the time between the order from the robot

[4] Most of the user's mother tongue was either Spanish or Catalan.

(a) A user in a high school.

(b) A user in another high school.

(c) A user in the community center.

(d) A user in the elderly social association.

Fig. 9. Examples of users performing the tests.

to segment objects and the response from the Wifibot's laptop, which is computed in less than a second.

Looking at the recognition rates, the best recognized gesture was the point at one. The negation gesture was the one with the lowest recognition rates, as it was the case of the offline results, mainly because the face not being well tracked when the face is sidewards the camera.

The system also shows high recognition rates for the object detection even though there were some errors, which are detailed in Table 3.

4. Conclusions

In this work, we presented a multi-robot system designed to interact with human users in a real time gesture based manner. The system is a proof of concept that shows how important is the interaction phase in order to be able to assist users with special needs, such as elderly or handicapped people. Consequently, they could interact with the robot in the way they are used to do with other human beings, and the robot can use the information provided by the users to help them. For instance, the robot could pick

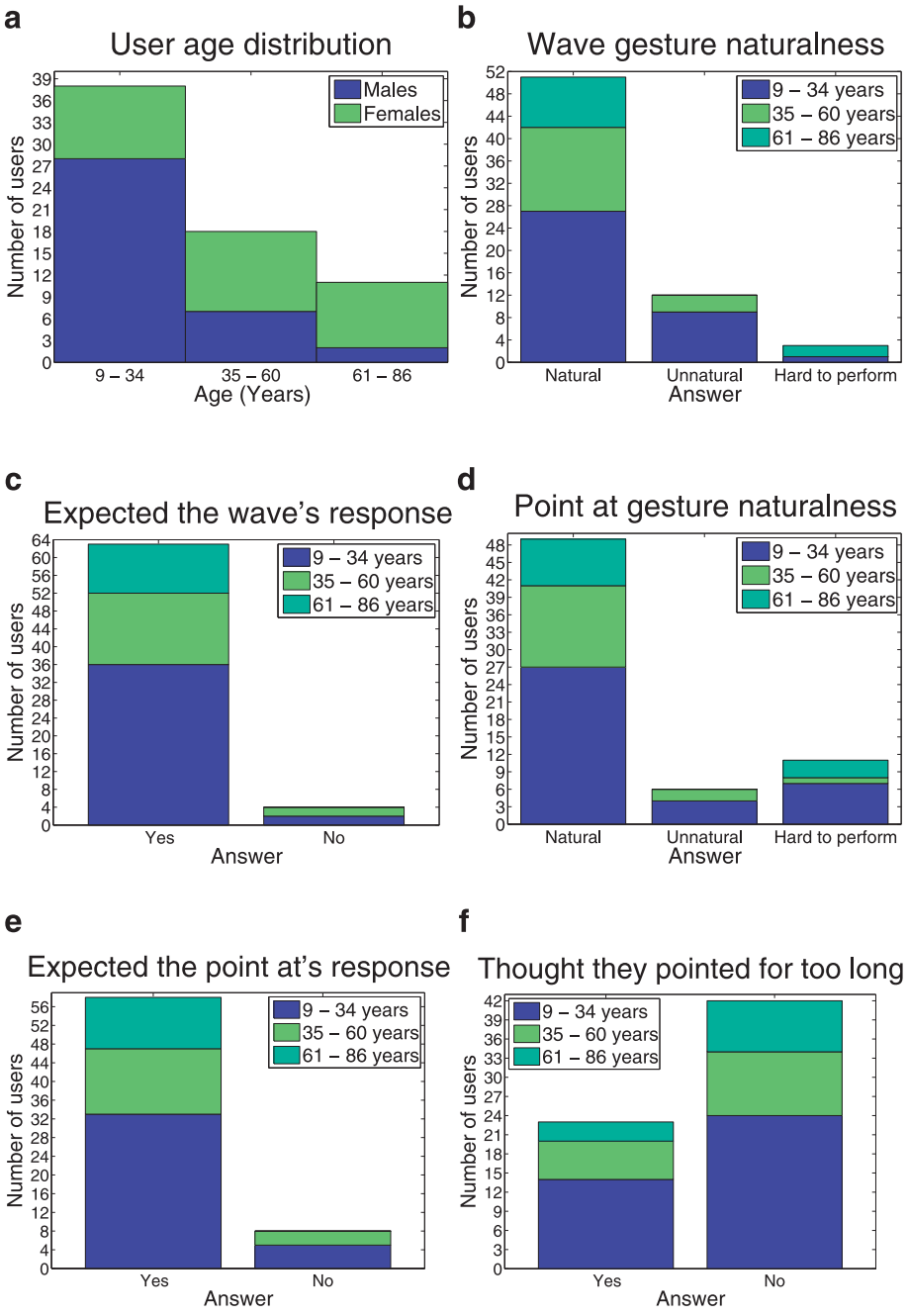


Fig. 10. User answers to the questionnaire.

something up from the floor without the need of actually recognizing the object but just knowing that the person referred it with a deictic gesture.

We included a gesture recognition method based on the Kinect™ v2 sensor which takes into account dynamic gestures, recognized by a DTW using specific features from the face and the body, and static gestures such as deictic ones to refer to something present in the environment.

The multi-robot system is shown as an effective way of combining efforts with specialized robots, one to carry the weight of the sensor and the computing power with a precise navigation, and the other able to speak and interact in a natural way with the user. Their collaboration in performing the tasks leads to the success of the system and the interaction.

Furthermore, an extensive set of user tests was carried out with 67 users who had little contact with robots and that were able to perform the tests with minimal external indications, resulting in a natural interaction for them in most of the cases. Offline tests also showed high recognition rates performing real time gesture detection and spotting in a specifically recorded data set.

Nevertheless, different elements of the system such as the detection of the pointing direction could be improved as future work. For instance, the use of a more accurate hand pose estimator like the ones proposed in [34–36] may allow the direction of the finger to be used to obtain the pointing direction, probably resulting in a more precise location estimation. The facial gestures are another element which could be highly improved, first by trying to use a better facial tracker which can properly handle side views (which

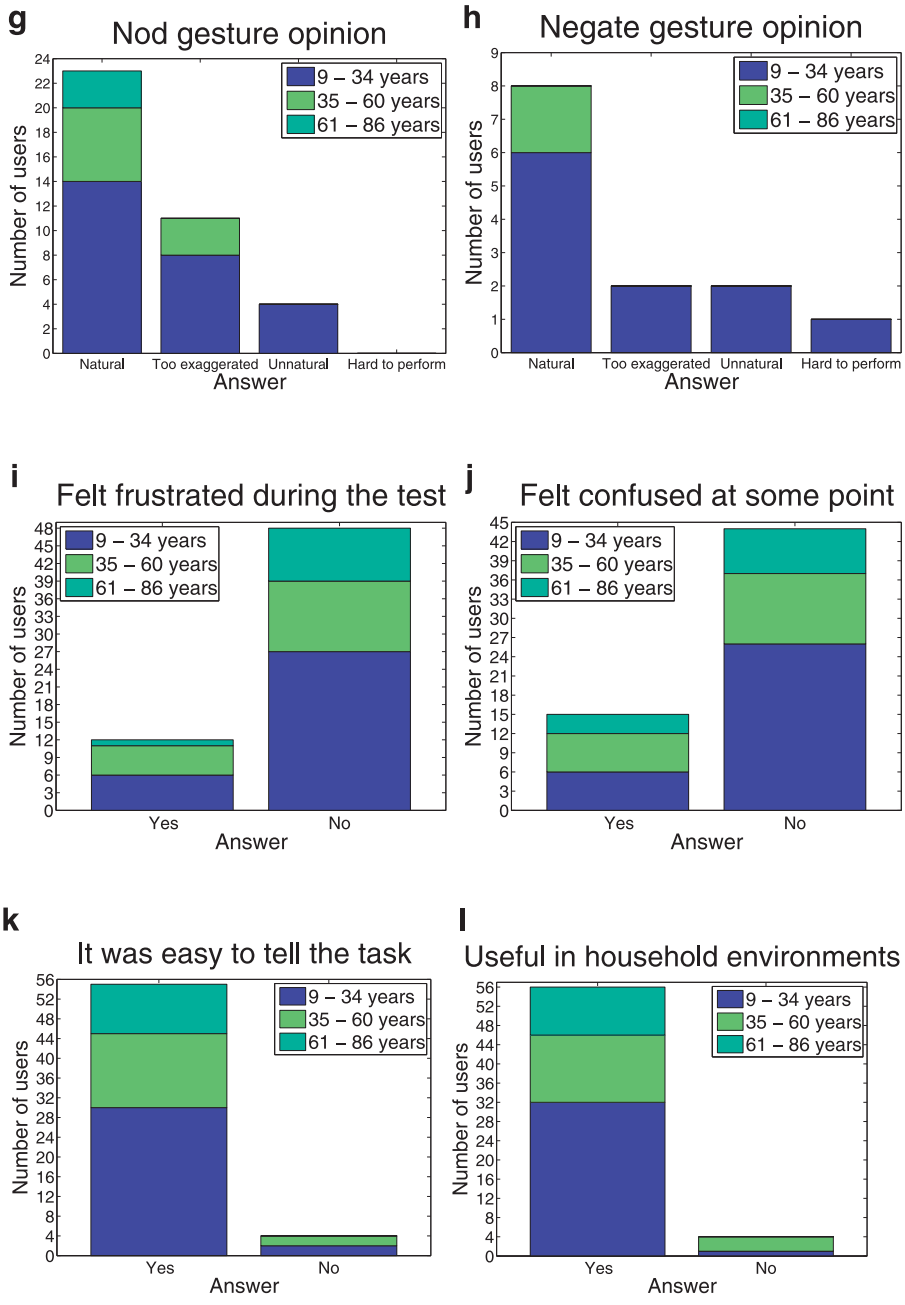


Fig. 10. Continued

clearly affect the detection of the negation gesture), but also by exploring or adding other kind of features.

Acknowledgments

We would like to thank La Garriga's town council, youth center and *Ràdio Silenci* for their help in the user test communication and organization, as well as to the following entities, associations and people located in La Garriga: the *Associació de Gent Gran l'Esplai de l'EspaiCaixa*, "*La Torre del Fanal*" community center, *Institut Vil·la Romana*, *Escola Sant Lluís Gonçaga*, and Pujol-Buckles family for allowing us to perform the tests in their facilities. Special thanks also to Dr. Marta Díaz for her guidelines in the user test analyses, to Joan Guasch and Josep Maria Canal for their help in the Wifibot adaptations, and to Víctor Vílchez for proofreading. This work has been partially supported by the Spanish Ministry of Economy and Competitiveness, through the projects TIN 2012-38416-C03-01 and TIN2013-43478-P. The research fellow Gerard Canal thanks the funding through a grant issued by the Catalunya — La Pedrera Foundation.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.cviu.2016.03.004.

References

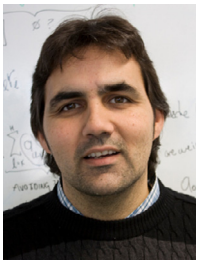
[1] J. DeVito, M. Hecht, The Nonverbal Communication Reader, Waveland Press, 1990.

[2] C. Breazeal, C. Kidd, A. Thomaz, G. Hoffman, M. Berlin, Effects of nonverbal communication on efficiency and robustness in human-robot teamwork, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005), 2005, pp. 708–713, doi:10.1109/IROS.2005.1545011.

[3] A. Hernández-Vela, M.A. Bautista, X. Perez-Sala, V. Ponce-López, S. Escalera, X. Baró, O. Pujol, C. Angulo, Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in RGB-D, Pattern Recognit. Lett. 50 (0) (2014) 112–121, doi:10.1016/j.patrec.2013.09.009.Depth Image Analysis
[4] M. Reyes, G. Domínguez, S. Escalera, Feature weighting in Dynamic Time Warping for gesture recognition in depth data, in: Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 1182–1188, doi:10.1109/ICCVW.2011.6130384.
[5] K. Kulkarni, G. Evangelidis, J. Cech, R. Horaud, Continuous action recognition based on sequence alignment, Int. J. Comput. Vis. 112 (1) (2015) 90–114, doi:10.1007/s11263-014-0758-9.
[6] B. Liang, L. Zheng, Multi-modal gesture recognition using skeletal joints and motion trail model, in: L. Agapito, M.M. Bronstein, C. Rother (Eds.), Computer Vision - ECCV 2014 Workshops, Lecture Notes in Computer Science, 8925, Springer International Publishing, 2015, pp. 623–638, doi:10.1007/978-3-319-16178-5_44.
[7] N. Camgz, A. Kindiroglu, L. Akarun, Gesture recognition using template based random forest classifiers, in: L. Agapito, M.M. Bronstein, C. Rother (Eds.), Computer Vision - ECCV 2014 Workshops, Lecture Notes in Computer Science, vol. 8925, Springer International Publishing, 2015, pp. 579–594, doi:10.1007/978-3-319-16178-5_41.
[8] D. Wu, L. Shao, Deep dynamic neural networks for gesture segmentation and recognition, in: L. Agapito, M.M. Bronstein, C. Rother (Eds.), Computer Vision - ECCV 2014 Workshops, Lecture Notes in Computer Science, vol. 8925, Springer International Publishing, 2015, pp. 552–571, doi:10.1007/978-3-319-16178-5_39.
[9] A. Yao, L. Van Gool, P. Kohli, Gesture recognition portfolios for personalization, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1923–1930, doi:10.1109/CVPR.2014.247.
[10] O. Lopes, M. Reyes, S. Escalera, J. Gonzalez, Spherical blurred shape model for 3-D object and pose recognition: quantitative analysis and HCI applications in smart environments, IEEE Trans. Cybern. 44 (12) (2014) 2379–2390, doi:10.1109/TCYB.2014.2307121.
[11] S. Iengo, S. Rossi, M. Staffa, A. Finzi, Continuous gesture recognition for flexible human-robot interaction, in: Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 4863–4868, doi:10.1109/ICRA.2014.6907571.
[12] H. Kim, S. Hong, H. Myung, Gesture recognition algorithm for moving kinect sensor, in: Proceedings of the 2013 IEEE RO-MAN, 2013, pp. 320–321, doi:10.1109/ROMAN.2013.6628475.
[13] A. Ramey, V. González-Pacheco, M.A. Salichs, Integration of a Low-cost RGB-D Sensor in a Social Robot for Gesture Recognition, in: Proceedings of the 6th International Conference on Human-robot Interaction, in: HRI '11, ACM, New York, NY, USA, 2011, pp. 229–230, doi:10.1145/1957656.1957745.
[14] T. Fujii, J. Hoon Lee, S. Okamoto, Gesture recognition system for human-robot interaction and its application to robotic service task, in: Proceedings of The International MultiConference of Engineers and Computer Scientists (IMECS 2014), I, International Association of Engineers, Newswood Limited, 2014, pp. 63–68.
[15] X. Zhao, A.M. Naguib, S. Lee, Kinect Based Calling Gesture Recognition for Taking Order Service of Elderly Care Robot, in: Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2014), 2014, pp. 525–530, doi:10.1109/ROMAN.2014.6926306.
[16] D. McColl, Z. Zhang, G. Nejat, Human body pose interpretation and classification for social human-robot interaction, Int. J. Soc. Robot. 3 (3) (2011) 313–332, doi:10.1007/s12369-011-0099-6.
[17] A. Chrungoo, S. Manimaran, B. Ravindran, Activity recognition for natural human robot interaction, in: M. Beetz, B. Johnston, M.-A. Williams (Eds.), Social Robotics, Lecture Notes in Computer Science, 8755, Springer International Publishing, 2014, pp. 84–94, doi:10.1007/978-3-319-11973-1_9.
[18] E. Bernier, R. Chellali, I.M. Thouvenin, Human gesture segmentation based on change point model for efficient gesture interface, in: Proceedings of the 2013 IEEE RO-MAN, 2013, pp. 258–263, doi:10.1109/ROMAN.2013.6628456.
[19] D. Michel, K. Papoutsakis, A.A. Argyros, Gesture recognition supporting the interaction of humans with socially assistive robots, in: G. Bebis, R. Boyle, B. Parvin, D. Koracin, R. McMahan, J. Jerald, H. Zhang, S. Drucker, C. Kambhamettu, M. El Choubassi, Z. Deng, M. Carlson (Eds.), Advances in Visual Computing, Lecture Notes in Computer Science, 8887, Springer International Publishing, 2014, pp. 793–804, doi:10.1007/978-3-319-14249-4_76.
[20] M. Obaid, F. Kistler, M. Häring, R. Bühling, E. André, A framework for user-defined body gestures to control a humanoid robot, Int. J. Soc. Robot. 6 (3) (2014) 383–396, doi:10.1007/s12369-014-0233-3.
[21] P. Barros, G. Parisi, D. Jirak, S. Wermter, Real-time gesture recognition using a humanoid robot with a deep neural architecture, in: 2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2014, pp. 646–651, doi:10.1109/HUMANOIDS.2014.7041431.
[22] J.L. Raheja, A. Chaudhary, S. Maheshwari, Hand gesture pointing location detection, Optik - Int. J. Light Electron Opt. 125 (3) (2014) 993–996, doi:10.1016/j.ijleo.2013.07.167.
[23] M. Pateraki, H. Baltzakis, P. Trahanias, Visual estimation of pointed targets for robot guidance via fusion of face pose and hand orientation, Comput. Vis. Image Underst. 120 (0) (2014) 1–13, doi:10.1016/j.cviu.2013.12.006.
[24] C. Park, S. Lee, Real-time 3d pointing gesture recognition for mobile robots with cascade HMM and particle filter, Image Vis. Comput. 29 (1) (2011) 51–63, doi:10.1016/j.imavis.2010.08.006.
[25] D. Droeschel, J. Stuckler, S. Behnke, Learning to interpret pointing gestures with a time-of-flight camera, in: Proceedings of the 2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2011, pp. 481–488.
[26] C. Matuszek, L. Bo, L. Zettlemoyer, D. Fox, Learning from unscripted deictic gesture and language for human-robot interactions, in: Proceedings of the 28th National Conference on Artificial Intelligence (AAAI), Québec City, Quebec, Canada, 2014.
[27] A. Jevtic, G. Doisy, Y. Parmet, Y. Edan, Comparison of interaction modalities for mobile indoor robot guidance: direct physical interaction, person following, and pointing control, IEEE Trans. Human-Mach. Syst. 45 (6) (2015) 653–663, doi:10.1109/THMS.2015.2461683.
[28] M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, ROS: an open-source Robot Operating System, in: ICRA Workshop on Open Source Software, 2009.
[29] R.B. Rusu, S. Cousins, 3D is here: Point Cloud Library (PCL), in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011.
[30] T. Arici, S. Celebi, A.S. Aydin, T.T. Temiz, Robust gesture recognition using feature pre-processing and weighted Dynamic Time Warping, Multimed. Tools Appl. 72 (3) (2014) 3045–3062, doi:10.1007/s11042-013-1591-9.
[31] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoust. Speech Signal Process. 26 (1) (1978) 43–49, doi:10.1109/TASSP.1978.1163055.
[32] R.B. Rusu, Clustering and segmentation, in: Semantic 3D Object Maps for Everyday Robot Manipulation, in: Springer Tracts in Advanced Robotics, 85, Springer Berlin Heidelberg, Ch. 6, 2013, pp. 75–85, doi:10.1007/978-3-642-35479-3_6.
[33] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395, doi:10.1145/358669.358692.
[34] J. Tompson, M. Stein, Y. Lecun, K. Perlin, Real-time continuous pose recovery of human hands using convolutional networks, ACM Trans. Graph. (TOG) 33 (5) (2014) 169:1–169:10, doi:10.1145/2629500.
[35] F. Kirac, Y.E. Kara, L. Akarun, Hierarchically constrained 3d hand pose estimation using regression forests from single frame depth data, Pattern Recognit. Lett. 50 (2014) 91–100. http://dx.doi.org/10.1016/j.patrec.2013.09.003.Depth Image Analysis
[36] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, S. Izadi, Accurate, robust, and flexible real-time hand tracking, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, in: CHI '15, ACM, New York, 2015, pp. 3633–3642, doi:10.1145/2702123.2702179.

Gerard Canal received his bachelor degree in Computer Science at Universitat Politècnica de Catalunya (UPC) in 2013. He obtained his Master degree in Artificial Intelligence at Universitat Politècnica de Catalunya (UPC), Universitat de Barcelona (UB) and Universitat Rovira i Virgili (URV), in 2015. His main research interests include the development of novel assistive technologies based on social robotics involving computer vision. He is currently pursuing a Ph.D. on assistive Human-Robot Interaction using computer vision techniques.

Sergio Escalera obtained the Ph.D. degree on Multi-class visual categorization systems at Computer Vision Center, UAB. He obtained the 2008 best Thesis award on Computer Science at Universitat Autònoma de Barcelona. He leads the Human Pose Recovery and Behavior Analysis Group. He is an associate professor at the Department of Applied Mathematics and Analysis, Universitat de Barcelona. He is also a member of the Computer Vision Center at Campus UAB. He is director of ChaLearn Challenges in Machine Learning. He is vice-chair of IAPR TC-12: Multimedia and visual information systems. His research interests include, between others, statistical pattern recognition, visual object recognition, and HCI systems, with special interest in human pose recovery and behavior analysis from multi-modal data.



Cecilio Angulo received his M.S. degree in Mathematics from the University of Barcelona, Spain, and his Ph.D. degree in Sciences from the Universitat Politècnica de Catalunya - BarcelonaTech, Spain, in 1993 and 2001, respectively. From 1999 to 2007, he was at the Universitat Politècnica de Catalunya, as Assistant Professor. He is nowadays an Associate Professor in the Department of Automatic Control, in the same university. From 2011 he's also serving as Director of the Master's degree in Automatic Control and Robotics. He's currently the Director of the Knowledge Engineering Research Group where he is responsible for research projects in the area of social cognitive robotics. Cecilio Angulo is the author of over 250 technical publications. His research interests include cognitive robotics, machine learning algorithms and social robotics applications.