Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
November-December 2017: 1–15

DOI: 10.1177/1729881417736950
journals.sagepub.com/home/arx
SAGE

A local user mapping architecture for social robots

Arnaud Ramey[1], María Malfaz[2], José Carlos Castillo[2], Álvaro Castro-González[2], Irene Pérez[2] and Miguel A Salichs[2]

Abstract

User detection, recognition, and tracking is at the heart of human–robot interaction, and yet, to date, no universal robust method exists for being aware of the people in a robot's surroundings. The present article imports into existing social robotic platforms different techniques, some of them classical, and other novel, for detecting, recognizing, and tracking human users. The outputs from the parallel execution of these algorithms are then merged, creating a modular, expandable, and fast architecture. This results in a local user mapping through fusion of multiple user recognition techniques. The different people detectors comply with a common interface called PeoplePoseList Publisher, while the people recognition algorithms meet an interface called PeoplePoseList Matcher. The fusion of all these different modules is based on the Unscented Kalman Filtering technique. Extensive benchmarks of the subcomponents and of the whole architecture demonstrate the validity and interest of all levels of the architecture. In addition, all the software and data sets generated in this work are freely available.

Keywords

User awareness, detection, recognition, tracking, HRI, multimodal fusion, social robotics

Date received: 4 October 2016; accepted: 20 July 2017

Topic: Vision Systems
Topic Editor: Antonio Fernandez-Caballero
Associate Editor: Ismael Garcia Varea

Introduction

Social robotics aims at making daily companion robots that interact with human users, helping and entertaining them in their everyday life. Consequently, they must follow social behavior and rules, spanning a wide range of applications and types of users. Examples include helping children with their homework, taking care of elderly people, giving information and advice to people in public places, and so on.[1,2] The relation between the human user and the robot can be short term, such as a robot giving directions at a shopping mall,[3] or long term, for instance, a robot delivering mail and food to employees in a lab on a daily basis over a period of several months.[4] The relation between human users and robots, called *human–robot interaction* (*HRI*), is at the core of social robotics: A social robot aims at helping users and, as such, it needs to attain a *user awareness*. This consists in the robot's having knowledge about how many users are around it, where they are, and who they are. The goal of the research presented in this article is to endow social robots with these abilities, something which is still a challenging problem for the robotics community.

There are works pointing out that humans attain their user awareness following a divide and conquer strategy,[5]

[1] Ecole Polytechnique, Paris, France
[2] Universidad Carlos III de Madrid, Robotics Lab, Madrid, Spain

Corresponding author:
María Malfaz, Universidad Carlos III de Madrid, Robotics Lab, Madrid, 28911, Spain.
Email: mmalfaz@ing.uc3m.es

that is, splitting this difficult task into independent subtasks of lesser complexity that are easier to achieve. Some patients who have suffered different types of accidents, such as seizures or head trauma, have had some very specific parts of their brain damaged, leaving the rest intact. This helps to understand the way our brain works, and especially where the different functions are located and how the whole system is articulated. For instance, patients of the so-called *prosopagnosia*, also called face blindness, cannot recognize the identity of known faces although the rest of the brain's functionality is intact, such as object recognition or even face detection.[6] The way people awareness is achieved in the brain can be divided into three modules: (i) *people detection*, which consists in locating people around us from the instantaneous data stream of our sensory system; (ii) *people recognition*, dealing with knowing who they are; and (iii) *people tracking and mapping*, which is a higher level understanding of people's motion to maintain spatial and temporal coherency. Therefore, considering this idea as an inspiration, the local mapping of users for a social robot can be split into three subtasks: user detection, user recognition, and user tracking and mapping (by using data fusion). This decomposition constitutes one of the main novelties of this article, which is the fusion of algorithms for user mapping instead of just fusing sensor information.

The structure of this article is as follows: in "Related work" section, we will review how user awareness is achieved in other social robots. In "Problem statement" section, we design our strategy to achieve a generic user awareness architecture as well as hardware and software constraints. In "Approach" section, we present the approach and structure designed to obtain user awareness given the defined strategy and constraints. The experimental results obtained following this approach are presented in "Experimental results" section. Finally, in "Conclusions" section, some conclusions are drawn and future research is outlined.

Related work

Giving user awareness to social robots is a challenging problem that has been tackled already by numerous authors and with a wide range of sensors and techniques. In this section, we will review the current trends in user awareness for social robots. Some of them are barely aware of their environment, similarly to mechanical puppets, while others perceive and recognize their users. This leads to a classification with different levels of user awareness.

There are proposals with no long-term memory about the users interacting with the robot. For example, the robot *Aibo*[7] was equipped with a variety of sensors and buttons on its body. When one of these buttons was pressed, the robot knew that there was a user nearby and started behaving accordingly. *RHINO*[8] is an interactive robot guide for museums. To interact with the robot, the users have to press buttons on the onboard interface, which will make the robot deliver information about the museum in a unidirectional fashion.

Additionally, there are robots that detect users automatically without needing them to take any explicit action. This is the case with the social robot *Kismet*,[9] which can perform a closed-loop active vision by using face and eye detection. *Roboceptionist* helps users to find their way in offices.[10] The interaction is short term, as users usually ask a few questions of the robot and then leave for their destination. Geiger et al.[11] presented the social robot *ALIAS* as a gaming platform for elderly people. The user is detected by using voice detection and a face detection algorithm, but no recognition whatsoever is performed. The HRI is made through the use of the tablet computer. The *STRAND*[12] project includes short-term user awareness using Red Green Blue (RGB)-D and laser information for detection with a Kalman filter that provides the tracks of the users in the space. In a similar line, the project *CompanionAble*,[13] focus on the use of social robots for elderly people, presents a perception approach that uses multiple cues based on histogram of oriented gradient (HOG) and shape models for people tracking through a Bayesian filter. The MOnarCH[14] project proposes another approach for a mobile platform for edutainment activities in a pediatric hospital. In this case, the user awareness of the robot is provided using an RGB-D camera placed on the robot and omnidirectional cameras placed in the environment.

The previous references presented robots that are able to detect and interact with users on a short-term basis, that is, if the same user happened to come back later, the robot would not remember their former interaction. Nonetheless, it is because we are able to identify individuals that we can develop a unique relationship with each of them. *Valerie* is one of the first robot receptionists, used at Carnegie Mellon university.[15] Valerie was involved in long-term interactions, and as it stood for several months in a booth at the entrance to offices. User recognition was made possible by using the use of a magnetic card reader: The users were unambiguously identified by swiping their ID card. In Kanda et al.,[16] the authors study the evolution of the relationship over time in an 18-day field trial between 119 first- and sixth-grade students and a humanoid robot. They chose to perform user recognition using wireless radio frequency identification (RFID) tags and sensors. *Jibo* (Jibo homepage: `https://www.jibo.com`) is a robot that recognizes the users around it by using its vision system (face detection and recognition) and microphones (speech recognition). Jibo gathers information about the users by using the applications and media they consume, so that the interaction is personalized. Portugal et al. presented *Social-Robot*,[17] a service mobile robot for social interaction with elderly people. An RGB-D camera provides information for people detection and face recognition in three dimension (3-D). With this information, the robot can safely approach

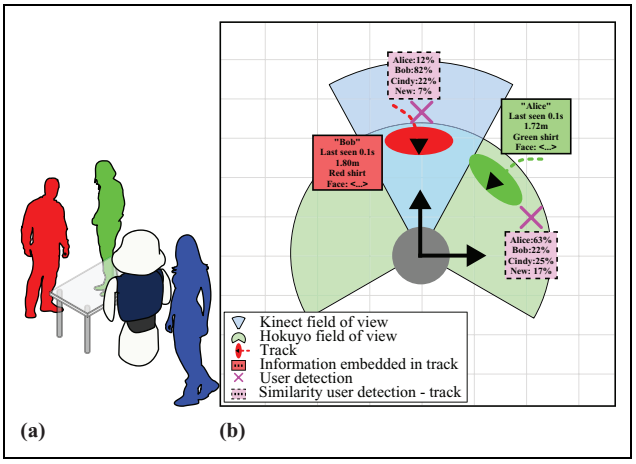

Figure 1. (a) A fictional situation of user awareness around a robot. (b) Knowledge representation of the user awareness in this situation.

people and, what is more, this system makes use of a collaborative network that manages user profiling and care.

Problem statement

The goal of this research is to endow robots with user awareness by detecting the users in the vicinity of the robot, recognizing them, and building a consistent representation of this knowledge on a map. The number of users, their positions, and who they are is a set of unknown variables that we regard as the *state of the system*. For each user around the robot, the system must create and maintain a set of data that we call a *track*: The user's identity, position, and trajectory.

We use an approach based on divide and conquer, as presented in the "Introduction" section, and split the user awareness into three modules: people detection, people recognition, and people tracking and mapping. These three tasks are executed simultaneously when the robot interacts with users. To make things clearer, let us consider the fictional case of Figure 1(a). The robot here, drawn as the dark blue and white shape in the middle, is surrounded by three users. Figure 1(b) illustrates the process of creating user awareness. *User detection* consists in detecting, in the sensor data stream, the users around the robot, and knowing where they are. Each of them can then be identified by a temporary ID. Those detections are shown in Figure 1(b) as pink crosses. The fields of view of the different sensors are drawn as angular ranges.

For each user, its track contains the user's identity, previous positions, and so on. Note that the user highlighted in blue is out of the sensors' field of view; therefore, the robot is not yet aware of that user's presence. In Figure 1(b), an ellipse shows the position of each track, associated with the uncertainty of its position, and a dashed line shows its previous positions. The additional knowledge in each track is depicted next to the track ellipse in the colored frame box with a solid border.

User recognition aims at obtaining a long-term coherency with the way it detects the users, in other words, matching the temporary IDs to a permanent identity. User recognition is used to match detections and tracks: The higher the similarity, the more probable the detection corresponds to the track. The similarity between each user detection (a pink cross) and the set of tracks, obtained by user recognition, is in the pink rectangle with a dashed border next to this cross.

After the user recognition is performed, through Kalman filtering and *multimodal data fusion* algorithms, each track can be updated with the matching user detection, by using the similarities computed by the user recognition. The set of tracks constitutes the *user mapping*.

This work has been carried out using Robot Operating System (ROS),[18] a software architecture specifically developed for use with robots. To sum-up the ROS landscape and glossary, processes are called *nodes*. They are independent and run in parallel. They can exchange data via messages sent on *topics*, data channels uniquely identified by their name. A function of a given node can be called from other nodes using a *service*, uniquely defined by a string name and a pair of strictly typed messages: One for the request and one for the response. ROS topics are many-to-many communication mechanisms, while ROS services are many-to-one.

Approach

The proposed approach is made of several steps, described in the following sections: First, in "A common data structure" section, we define a common ROS data structure, called `PeoplePoseList (PPL)`, that implements the concept of track. In "User detection" section, we present how the user detection algorithms are shaped to publish `PPL`s. The user recognition algorithms are then used as matching tools of a `PPL`s produced by a detection algorithm against a `PPL`-containing tracks, which is presented in "User recognition" section. A one-to-one assignment between the tracks and each `PPL`s produced by a detection algorithm is computed by using the hints given by each user recognition algorithm. These assignments are used to create or update tracks through data fusion algorithms. The updated set of tracks is an up-to-date mapping of the users. This process is presented in "User mapping, using the data fusion based on Kalman filtering" section. Finally, in "Configuration of multiple `PPLP`s and `PPLM`s" section, we present how the developed architecture can be adapted to different social robots, with different sensors. All the source code developed in this research is freely available online under an open source license (https://github.com/UC3MSocialRobots).

A common data structure

Common practice in computer science consists in standardizing the communication layer and the data that are exchanged by the different modules, more so than the particular

structures of these modules. For this reason, we designed a common data structure that describes the information associated to a user detection and this will be the output of any user detector that we want to integrate: the PeoplePose (PP). To ensure the modularity of our user awareness architecture, we exploited ROS messages and services mechanisms.

The PP data structure corresponds to a single user detection and contains all the information worth being shared, such as the 3-D pose of the user, the confidence of the detection, RGB and depth images of the user if they are available (for vision-based algorithms), and so on. The PP data structure is implemented as an ROS msg, a simple message description language for describing a data structure exchanged by ROS nodes. The details of the PP message are in Code listing 1.

```
// the header, useful for the stamp and the frame
std_msgs/Header header
:time stamp
:string frame_id
// person's head estimated position and orientation
geometry_msgs/Pose head_pose
:geometry_msgs/Point position
::float64 x, y, z
:geometry_msgs/Quaternion orientation
::float64 x, y, z, w
// the standard deviation of the estimated pose
float32 std_dev
// person's name (for instance, "Bob").
// Only filled by user recognition methods, such as face
    ↪ recognition or multimodal fusion.
string person_name
// between 0=really unsure and 1=very sure
float32 confidence
// the color mask of the user: a tight crop of the RGB
    ↪ image to the user detection.
sensor_msgs/Image rgb
// the depth mask of the user
sensor_msgs/Image depth
// the binary mask of the user.  Image pixels belonging
    ↪ to the user are > 0
sensor_msgs/Image user
// a list of attributes of this person, for instance her
    ↪ height, her preferences...
string[] attributes_names
// the values of the previous attributes
string[] attributes_values
```

Code listing 1. The People Pose message.

Not all the fields of the message are necessarily filled by all the methods: For instance, a people detector based on the information of a two-dimensional (2-D) laser range finder will not use the image fields. A detector can detect several users at once with a single data input. For instance, a face detector can find several users in the same RGB image. The different PPs generated by each detection are then gathered into a single message: This collection of PPs is then called PPL and is also based on a ROS msg. An example of a PPL is provided in "Architecture output: Sample data based on multimodal fusion" section.

User detection

An algorithm capable of generating PPL messages is called a PeoplePoseList Publisher (PPLP). The integration into the robotic architecture is straightforward as long as each new detector publishes a PPL data structure. An example of a processing flow with several PPLPs is visible in Figure 2. Let us suppose that our robot is equipped with a microphone, a laser range finder, and an RGB webcam. The different input streams can be seamlessly shared between different PPLPs. In this example, PPLP3 and PPLP4 share the color stream of the camera, and as PPLP4 is computationally costly, it is chosen to run on a remote computer. The topics are relayed between the robot and the remote computer using a ROS communication layer.

The integration of the user detection algorithms. The architecture integrates a series of user detection algorithms that make use of the most common sensor technologies in social robots, such as RGB images, 3-D depth, and 2-D lidar information.

Improved Viola–Jones face detection–based PPLPs. The depth data are useful to discard false positive detectionsz given by the classical RGB Viola–Jones classifier for face detection.[19] To ease the integration, this detector is wrapped as a *PPLP*. The improvement consists in reprojecting a given number of 2-D points in 3-D and examines the resulting 3-D bounding box. If it does not comply with some generic given geometric constraints, this detection is classified as a false positive and discarded. We used a maximum face width of 30 cm and height of 40 cm, which are fairly permissive.

Improved HOGs PPLPs. A HOG is a feature used in computer vision for object detection.[20] It has turned out to be a very efficient technique for the detection of human shapes. The basic idea underlying this concept is that objects within an image can be described through the distribution of edge directions or intensity gradients. In a way similar to the face detector presented before, the original algorithm needs an RGB image as input and returns as output the rectangular estimates of the people. Unlike the face detector, the rectangle returned by the 2-D detector is usually bigger than the person and not centered on her. For this reason, we compute the biggest 3-D cluster of the 3-D cloud, then threshold the bounding box of this biggest 3-D cluster. The resulting HOG detector, along with the false positive removal, was wrapped as a *PPLP*.

NiTE-based PPLPs. The patented PrimeSense NiTE middleware,[21] freely distributed under Apache License, version 2.0, allows detecting and tracking human shapes from depth maps. The NiTE middleware supplies a data structure that a system module converts into a *PPL*, the *users multi-mask*, and publishes it using the ROS messaging system. This structure indicates where the users are: if a pixel p of the user's multi-mask has the value 0, this means there is no user in p, whereas if it has the value 1, p corresponds to a pixel of the user 1, and so on. For a given user, a *user mask* is the multi-mask image where all the

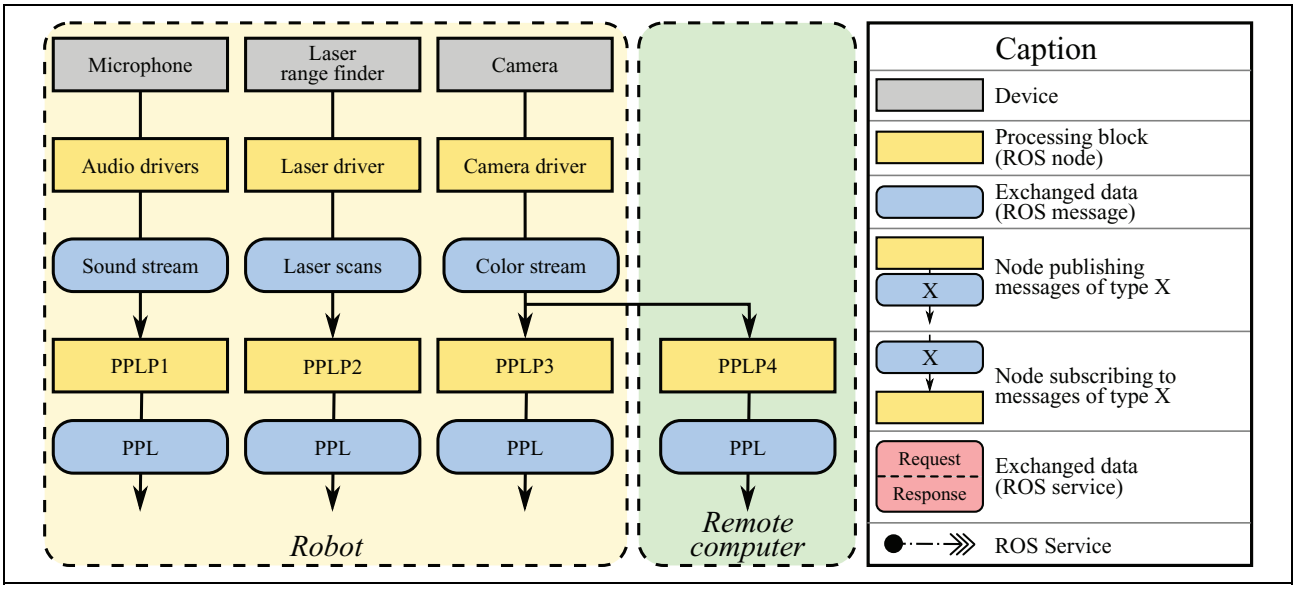

Figure 2. An example of distributed people detection by using several PPLPs. The legend is identical for Figures 2, 3, and 4. As Figures 2, 3 and 4 illustrate different parts of the system, they do not necessarily display all symbols contained in this legend. PPLP: PeoplePoseList Publisher.

pixels that do not belong to this user are set to 0 (i.e. "erasing" the other users). The NiTE middleware already includes tracking capabilities. The same physical user, whether or not visible in successive frames, is identified by the same ID in the resulting successive multi-masks.

Polar-Perspective Map-based PPLP. This people detector, introduced for a pedestrian detection system by,[22] uses the idea that a person appears as a set of points tightly close one to another in the 3-D point cloud given by the range imaging device. When projecting these 3-D points on the ground plane, these clusters will be projected onto the same area, thus generating a sort of high-density blob on the ground plane. Standing persons can then easily be characterized by the size of the blob. The so-called *Polar-Perspective Map* (*PPM*) is an occupancy map based on the polar coordinate system: It uses a regular grid based on the bearing of the points and their inverse distance to the device.

Tabletop PPLP. The tabletop PPLP is based on the idea that people standing on the floor generate a point cloud that is similar to objects standing on a tabletop and detecting objects on a planar surface such as a tabletop is a problem that has already been tackled by other authors (e.g. object grasping[22]). The tabletop PPLP combines some of these techniques in an innovative and straightforward way to find the users in front of the robot. It is based on detecting the ground plane by using the statistical RANSAC method,[23] separating aligned blobs by using a Canny filter applied to the depth image,[24] and then retrieving the pixel blobs of the objects that are on top of the ground plane.

Leg pattern–based PPLP. With the information that has been structured and associated to a metric dimension provided by the 2-D laser range finders, we obtain a direct understanding of the scene in front of the robot in the laser plane. Since laser range finders are typically mounted at the level of the legs of the users, that is, about 40 cm high, user detection is made through the detection of their legs. Many leg pattern–based detection algorithms exist. We chose the one described in Bellotto and Hu[25] for its simplicity and the overall good performance claimed by its authors. It was integrated as a PPLP: It subscribes to the laser scans acquired from the laser range finder and performs the leg pattern–based detection described in the original article.

User recognition

In our architecture, each user detector seen in "User detection" section publishes independently the instantaneous positions of the detected users, shaped as PPLs. As such, two detectors often produce detections at different rates. The user recognition algorithms described in this section tackle this problem, providing matching capabilities between detections and tracks.

Throughout this section and the following ones, we denote by $\mathcal{T}$ the set of tracks (representing the tracked users) and $\mathcal{P}$ a PPL created by a user detection algorithm. We denote by $n_{\mathcal{T}}$ the number of tracks and $n_{\mathcal{P}}$ the number of PPs

$$\mathcal{T} = \{\mathcal{T}_i, 0 \leq i \leq n_{\mathcal{T}}\} \quad \text{where } \forall i \in [1, n_{\mathcal{T}}], \quad \mathcal{T}_i \text{ is a } PP$$
$$\mathcal{P} = \{\mathcal{P}_j, 0 \leq j \leq n_{\mathcal{P}}\} \quad \text{where } \forall j \in [1, n_{\mathcal{P}}], \quad \mathcal{P}_j \text{ is a } PP$$

The idea for integrating the different user recognition algorithms is the following: A user matching algorithm M is defined as a normalized distance that takes a track and a detection PP as inputs and returns their similarity as output

$$M : \begin{cases} \mathcal{T} \times \mathcal{P} & \mapsto & [0,1] \\ (\mathcal{T}_i, \mathcal{P}_j) & \rightarrow & ||\mathcal{T}_i \mathcal{P}_j|| \end{cases}$$

Given tracks $\mathcal{T}$ and detections $\mathcal{P}$, this algorithm M can then compute a *cost matrix* C_M of size $n_\mathcal{T} \times n_\mathcal{P}$ between $\mathcal{T}$ and $\mathcal{P}$. That way, for all $i,j \in [1, n_\mathcal{T}] \times [1, n_\mathcal{P}]$, the more the track PP $\mathcal{T}_i$ from $\mathcal{T}$ matches the detection PP $\mathcal{P}_j$ from $\mathcal{P}$ according to this algorithm, the smaller the corresponding element in the cost matrix $C_M[i,j]$ is set. As such, the algorithm suggests a match between the track $\mathcal{T}_i$ and the detected PP $\mathcal{P}_j$ in a "soft way": There is not always a suggested matching for all detections. If the detection PPL does not contain faces, a cost matrix based on face recognition will be full of ones and then it will not have any weight in the final assignment.

We call a PeoplePoseList Matcher (PPLM) an integrated user recognition algorithm using this formalism that given both detection and track PPLs computes this cost matrix. Each PPLM is shaped as a ROS service providing high modularity: Several PPLMs can run in parallel seamlessly, and the distributed nature of ROS even allows these nodes to run on different computers. A new algorithm is thus easily integrated: It just has to comply with this interface, and the different nodes can be distributed among different machines, thanks to the ROS communication layer. Finally, adding a new PPLM is quick and easy: The new node needs to supply its own MatchPPL service.

Integration of user recognition algorithms. From the information provided by the different user detection algorithms, several recognition techniques have been shaped as PPLMs to achieve a higher level of abstraction. Thus, the architecture is able to perform long-term recognition of the user, improving user awareness. Although some recognizers, such as Euclidean distance or NiTE multimap, only provide short- and mid-term recognition, others, such as face recognition, are able to identify users over a longer time span, even if days have passed since their last identification. Therefore, it is the fusion of all these algorithms that is considered long term rather than the separate methods themselves.

Euclidean distance PPLM. The simplest method to estimate the likelihood of a track against a detected PP is to compare their 3-D positions. In other words, the closer are a track and a detection, the more likely it will be that they correspond to the same person. We have the explicit formula, for $i,j \in [1, n_\mathcal{T}] \times [1, n_\mathcal{P}]$

$$C_{\text{Euclidean}}[i,j] = \min\left(1, \frac{||\mathcal{T}_i, \mathcal{P}_j||_{L_2}}{D}\right)$$

The constant D is a distance threshold, in meters, and is introduced for normalization. This likelihood estimation needs to choose a distance function. We used the Euclidean L_2 norm, as it corresponds more accurately to the standard definition of the distance between 3-D positions.

Face recognition–based PPLM. The visual appearance of the face is key information that humans use extensively to discriminate between people. For this reason, we use the Fisherfaces algorithm,[26] a face recognition method that uses dimensionality reduction.

There are other methods, such as Eigenfaces[27] and Local Binary Pattern Histogram,[28] but based on other publications that compare their performance, Ahmed and Amin[29] and Belhumeur et al.[26] decided to use the Fisherfaces algorithm. The face recognition–based PPLM reuses the results of the face detection PPLP presented in "The integration of the user detection algorithms" section.

As said before, the more similar are two PPs, the smaller their cost should be. If the j th user face in the detection PPL is set, the face recognizer determines the most similar reference PP, and the corresponding cell in the cost matrix is set to zero. We have the explicit formula, for $i,j \in [1, n_\mathcal{T}] \times [1, n_\mathcal{P}]$

$$C_{\text{face}}[i,j] = \begin{cases} 0 & \text{if } i = \text{face}(\mathcal{P}_j) \\ 1 & \text{otherwise} \end{cases}$$

Height-based PPLM. The height of the users is a good metric not only because it helps recognize one from another but also since the height of unknown users may help to determine their gender, as men tend to be taller than women. We used a novel method for estimating the height of the user,[30] which deals with poses beyond standing straight, such as being slighty stooped or lifting an arm for greeting. Provided the depth image and user mask image, the height is obtained by computing the length of a line that goes from the head of the user to the feet, going through the middle of the body shape, and as such, this method requires that the user be entirely visible in the image stream. The vertical field of view (FOV) of the device being 50°, the method requires the user to be further than approximately 2 m. This strong assumption can be checked thanks to the distance of the user in the depth image.

It first performs a morphological thinning on the user mask image,[31] which generates the skeleton of the image. Then, the length of the skeleton from head to feet is computed. This gives us a pixel height of the user, that is converted into a metric one by using the depth information. User matching is then performed by evaluating the user height on both tracks $\mathcal{T}$ and people detections $\mathcal{P}$ and setting the matching cost to the absolute height difference. The constant H is a height threshold, in meters, and is introduced for normalization. We have the explicit formula, for $i,j \in [1, n_\mathcal{T}] \times [1, n_\mathcal{P}]$

$$C_{\text{height}}[i,j] = \min\left(1, \frac{|\text{height}(\mathcal{T}_i) - \text{height}(\mathcal{P}_j)|}{H}\right)$$

NiTE multimap–based PPLM. The raw output of the NiTE algorithm, presented in "The integration of the user detection algorithms" section, is shaped as a user multimap. The cost of matching a given detected PP with the set of tracks is defined as follows: This cost is equal to zero if two NiTE user identifiers are equal, and equal to one otherwise. In other words, the cost matrix of this PPLP is mostly set to one, with some zero values where NiTE names correspond: We have the explicit formula, for $i,j \in [1, n_{\mathcal{T}}] \times [1, n_{\mathcal{P}}]$

$$C_{NiTE}[i,j] = \begin{cases} 0 & \text{if } NiTE\_id_i = NiTE\_id_j \\ 1 & \text{otherwise} \end{cases}$$

PersonHistogramSet-based PPLM. Color histograms have been used extensively for user recognition.[32,33] However, they often do not take into account the fact that this color data in a person is naturally structured: A possible segmentation relies on three parts, the head, the upper body (torso and arms, covered by a shirt or another item of clothing), and the lower body (trousers, skirt, etc.).

Some articles represent the user's color distribution as a set of histograms, but with a constant height step,[32] therefore the slices do not correspond to physical body parts (head, torso, limbs, etc.). We developed a novel method for user recognition based on color histograms,[30] by generating a set of three Hue histograms structured so as to represent the previously mentioned natural segmentation of the human body. Once the PersonHistogramSets (PHSs) are computed for both a track and a PP, their matching cost is obtained by summing the intersection distances between corresponding histograms in both PHSs. We have the explicit formula, for $i,j \in [1, n_{\mathcal{T}}] \times [1, n_{\mathcal{P}}]$

$$C_{\text{PHS}}[i,j] = d_{\text{PHS}}(\mathcal{T}_i, \mathcal{P}_j)$$

User mapping, using the data fusion based on Kalman filtering

We have defined a common structure for matching algorithms for PPLMs that provide cost matrices. For a given detection PPL, the cost matrix describes how each detected PP is similar to the different reference tracks (also structured as PPs). In this section, we focus on the algorithm that uses these PPLMs to create the tracks, that is, user mapping. The state of each tracked user is a nonlinear system and forces us to choose a nonlinear data fusion algorithm. To predict and update each track, we chose an extension of Kalman filtering for highly nonlinear problems: Unscented Kalman Filters (UKFs).[34] Therefore, the UKF fuses information from the recognition techniques described in "Integration of user recognition algorithms" section.

From now on, the processing block in charge of retrieving all cost matrices and performing the multimodal fusion, structured as a ROS node, will be referred to as the *fusion node*. The fusion node has the set of tracks corresponding to the tracked users in memory $\mathcal{T}$. Each of these tracks is structured as a PPs. This node subscribes to a range of topics emitted by PPLPs. As said before, each of these PPLP publishes independently and asynchronously the instantaneous positions of the detected users, shaped as PPLs. *Upon reception of each* PPL *published by each* PPLP, two successive processes called *gating* and *matching* take place in the fusion node.

First, in the *gating* process,[35] user detections that cannot correspond to tracks are kept away, being stored in the so-called gating buffer. This is determined by computing the Euclidean distance between each track and the detection and comparing it with the maximum distance that a human person running could go in the time elapsed since the last update of the track. Because of gating, in each received PPL, not all PPs are necessarily used for the matching by the fusion node, avoiding incorrect matchings.

Once the gating is done, in this *matching* phase, for the remaining PPs, the fusion node needs to determine which track corresponds to which detected PP. We denote by C, the *global cost matrix* used for the update of the Kalman filter of each user. For each PPLMs M, the fusion node requests the cost matrix C_M. As different user matching algorithms run in parallel, each of these provides its own cost matrix. The global cost matrix C is set to the weighted average of all these cost matrices, where the weights, w_M, are chosen by design to reflect the a priori confidence in each matching algorithm

$$C = \sum_{M \in \text{ set of PPLM}} w_M C_M$$

Similarly to each cost matrix, the more a detection PP from $\mathcal{P}$ matches a track PP from $\mathcal{T}$, the smaller the corresponding element in C should be. This is where the different PPLMs are useful: Because of the global cost matrix C, the more similar are a given track PP and a detection PP, the smaller the corresponding element of the cost matrix.

The fusion node computes the *linear assignment* corresponding to this cost matrix with the *Jonker–Volgenant algorithm*.[36] This assignment determines an optimal track-to-detection assignment (an injective function)

$$X : \begin{cases} [1, n_{\mathcal{T}}] & \mapsto & [1, n_{\mathcal{P}}] \\ i & \to & j \end{cases}$$

that minimizes: $\min_X \sum_{i \in n_{\mathcal{T}}} C[i, X(i)]$. In other words, X tells us which detected PP $\mathcal{P}_j$ corresponds to each track $\mathcal{T}_i$. For each track $\mathcal{T}_i$, $i \in [1, n_{\mathcal{T}}]$, the inner UKF embedded is updated with the matching detected PP, $\mathcal{P}_{X(i)}$.[34] That way, the 3-D position of the tracked user is updated with the latest corresponding data.

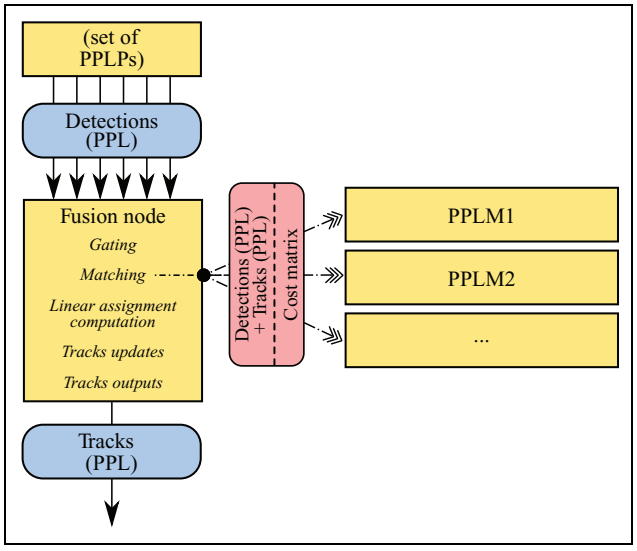

Figure 3. Diagram of the dataflow between the fusion node and the different PPLMs. The legend is the same as that of Figure 2. PPLM: PeoplePoseList Matcher.

Finally, the updated set of tracks, which is also shaped as a PPL, is emitted by the fusion node. If there are more detected PPs than tracks, after the fusion, each PP that is not used to update any track is put in this gating buffer. When a critical number of unassociated measures accumulates at a given spatial position in a window of time in the buffer, a new track is created.

"User recognition" section presented how each PPLM provides a ROS *service* of the same type, which we denote by MatchPPL, but with different names. The fusion node calls only the specific services that it wants to use. That way, the architecture benefits from the advantages of the structure of PPLPs and PPLMs, as pointed out in "User detection" and "User recognition" sections. The data flow is illustrated in Figure 3.

Configuration of multiple PPLPs and PPLMs

The user awareness described in this article can be adapted to different sensors, hardware configurations, and computing capabilities. A set of PPLPs and PPLMs, along with the data fusion node, is called a *configuration*. A configuration needs at least one PPLP, for user detection, and one PPLM, for user recognition. In that case, there is no multimodal fusion: the detections come from a single source. At the other end of the scale, a configuration using all the PPLPs and PPLMs implemented to date is presented in Figure 4. Note that using all the algorithms at the same time corresponds to obtaining the best trade-off between the performance of the tracking and the burden of the computations needed, the set of PPLPs and PPLMs must be chosen according to the characteristics of the robot. For instance, if it has no laser range finder, the leg PPLP can be removed. If the users will not come close, the face recognition PPLM is not needed. To help design a configuration relevant for a given robot setup, the robot Mini, we will assess the performance of each component of the architecture in the next section.

Experimental results

In this section, we present the robotic platform employed in the experiments in "Robotic platform" section and the acquisition of a realistic data set in "RoboticsLab People Dataset (RLPD): A realistic HRI-based people data set" section. The data are used to test the performance of the presented algorithms for user detection (see "Benchmarking of user detection algorithms with RLPD" section) and user recognition (see "Benchmarking of user recognition algorithms with RLPD" section). "Architecture output: Sample data based on multimodal fusion" section presents a sample output of the architecture and, finally, the performance of the overall user awareness architecture is evaluated in "Benchmarking of multimodal fusion configurations with RLPD" section.

Robotic platform

Our tests have been run using the Kinect camera integrated in the social robot *Mini* developed at our lab (see Figure 5 for more details about its sensors and actuators).

Although the computational needs of the architecture would allow performing the whole operation entirely on the robot, we traditionally use a distributed architecture that prevents overloading the robot, which could affect its reactivity. In the evaluation described in this article, two computers are used: the embedded computer of the robot (acting as a master) and a remote desktop computer for computation. The former is an Intel i5 quadcore CPU @ 3.30 GHz and embeds the driver for the sensors and a depth imaging device. The latter is an AMD Athlon 64 Dual Core @ 2.7 GHz and contains the user awareness architectures: PPLPs, PPLMs, and the fusion node. Both are on the same Ethernet network, which alleviates the issues due to network performance. During the development and testing phases, the architecture proposed in the article and the network were fast enough to process the data at the rate of the data set, which is roughly 5 Hz.

RoboticsLab People Dataset (RLPD): A realistic HRI-based people data set

There are academic data sets adapted for the evaluation of people detection, recognition, and tracking, such as DGait[37] and Kinect Tracking Precision (KTP).[38] They allow a faithful measurement of the performance of the user awareness system, make the comparison with other similar systems easier, and ensure that the measurements can be repeated in the same context.

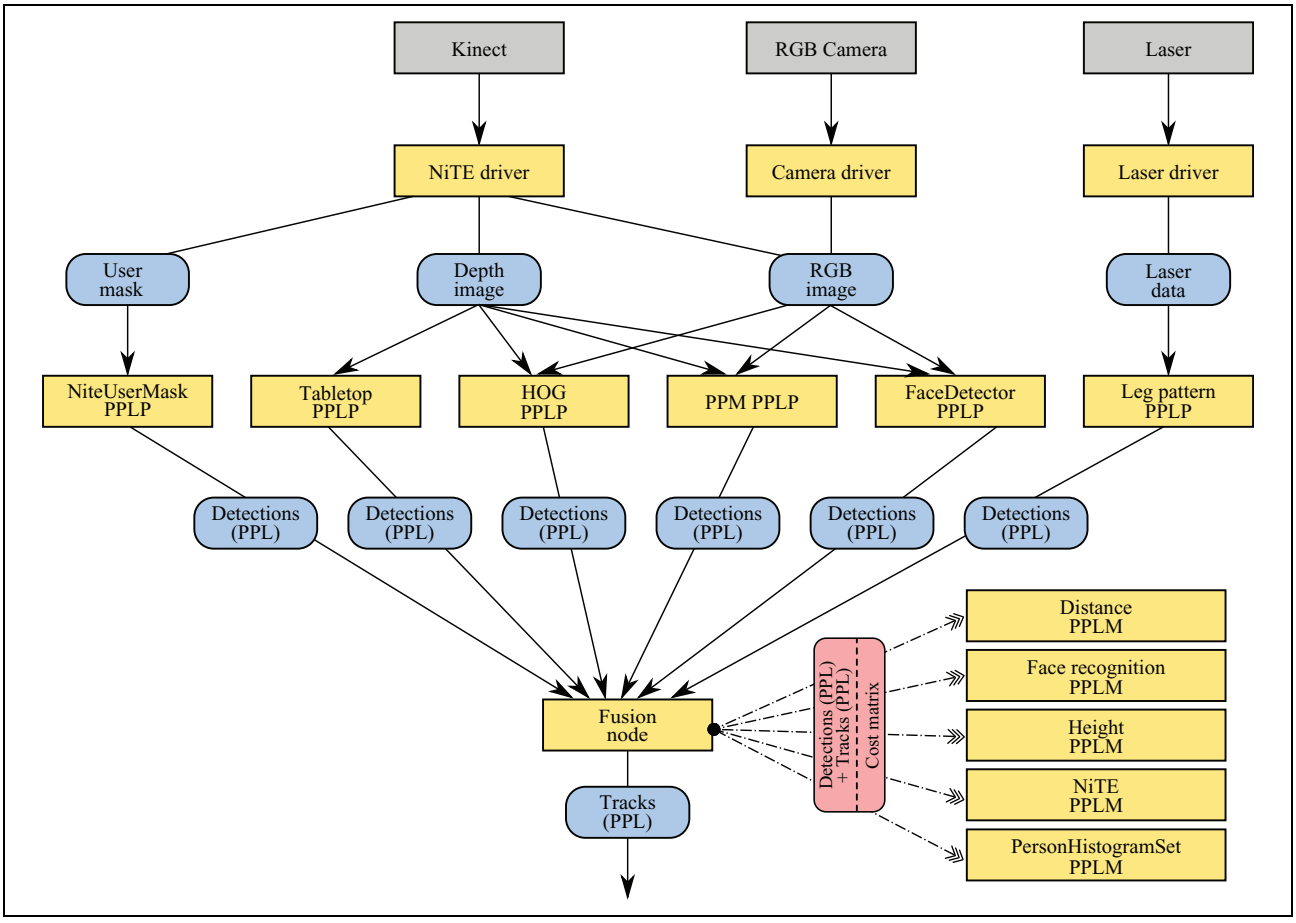

Figure 4. Multimodal fusion using all implemented `PPLP`s and `PPLM`s. The legend is the same as that of Figure 2. The NiTE driver provides three kinds of information: depth, color, and the user mask. The latter contains an initial user detection based on their depth with respect to the camera as seen in Figure 6. `PPLP`: `PeoplePoseList` Publisher; `PPLM`: `PeoplePoseList` Matcher.

Figure 5. Main components of the social robot Mini.

However, even though our system was designed in a fashion as generic as possible, it has been tested in a social robot. For this reason, we decided to acquire real data from the robot *Mini*, and with users that fit best the target audience: people from Spain, with a variety of genders and shapes. In addition, the actors do not wander randomly on the stage, as they do in the KTP data set: We designed scenarios that mimic a realistic HRI situation in which one or several users interact naturally with the robot: addressing the robot, using gestures, respecting the proxemics distance, and so on.

Data set summary. The data set represents three users interacting with a robot that integrates a Kinect camera. They move on the stage according to a script that was previously defined and is made of three scenarios of increasing difficulty. Their motion is challenging: They get in and out of the room, there are occlusions and partial views.

The data set is meaningful if and only if the real positions of the users are known. We first thought of using markers, such as ARToolkit markers.[39] However, the imperfect detections could not guarantee an accurate ground truth concerning the users' positions in each frame. For this reason, in each of the 600+ frames, the ground truth user positions have been manually labeled.

This data set is licensed under the terms of the GNU General Public License version 2 as published by the Free Software Foundation and freely available for downloading along with images and videos (https://sites.google.com/site/rameyarnaud/research/phd/roboticslab-people-dataset).

Input frames. Each frame is labeled with a time stamp. The time stamp is expressed in milliseconds elapsed after the beginning of the recording, using six digits with leading zeros. For instance, frame 065514 was recorded 1 min and 5 s after the beginning of the recording. In total, we have 647 frames for 133 s (2 min 15 s roughly), which is, on average, 5 frames per second.

Acquired data. For each frame, we have four images and one data file: (i) The RGB image ("XXXXX_rgb.png", lossy JPG compression, quality: 85); (ii) The depth image ("XXXXX_depth.png" and "XXXXX_depth_params.yaml", lossy affine depth-as-PNG compression); (iii) The user mask obtained as output of the Kinect API, called NiTE. This image stream is synchronized with the RGB and depth streams of the Kinect and indicates, for each pixel of each frame, whether this pixel belongs to a detected user using the NiTE algorithm presented in "The integration of the user detection algorithms" section ("XXXXX_user_mask_illus.png", lossless PNG compression); and (iv) the hand-labeled ground truth user mask. This image stream has the same purpose as the NiTE user masks, but it has been manually annotated, so that it contains the exact ground truth concerning the position of each user in each depth image ("XXXXX_ground_truth_user.png", lossless PNG compression).

The data set also supplies the camera info of the depth imaging device. It is made up of the camera's intrinsic parameters and allows converting 2-D pixels into 3-D points. The calibrations for both the RGB and depth (infrared) cameras are available.

Data set annotation. The ground truth user positions have been labeled manually in each frame, using both RGB and depth contents to create a "perfect" user mask. We developed a Graphical User Interface shaped as a raster graphics editor that allows the manual labeling of each pixel of a depth image.

Like the data set, these tools are licensed under the terms of the GNU General Public License version 2 as published by the Free Software Foundation, and freely available for downloading at the same URL.

Data set analysis. The data set was recorded on July 2014. It is made of about 650 frame acquisitions, each of them consisting of four images, that is, about 2600 images. The data set is roughly 65 megabytes. These images can be easily imported into any programming language, such as C++ or Matlab. The complete data set takes 133 s and the total number of frames is 647, and 548 of them have several PPs. Some samples are visible in Figure 6.

Figure 6. Some samples of the RLPD. From left to right: column 1: the RGB image; column 2: the depth image; column 3: the manually labeled user map; and column 4: the NiTE (Kinect API) user map. The data set has some challenging features: partial (second row) or complete occlusions (third row), user not fully visible (fourth row). Note how the manual color indexing of the users is consistent (third column): the same user always corresponds to the same color. On the other hand, the NiTE algorithm performs swaps and creates new users, or even merges users (fourth column).

Benchmarking of user detection algorithms with RLPD

We benchmarked the different user detection algorithms on this new data set. These algorithms were all wrapped with a common interface, `PPLP`. The performance of the different `PPLP`s on the RLPD is presented in Table 1.

We calculated the *accuracy* and the *hit rate* of each algorithm. *Accuracy* refers to the percentage of correctly evaluated frames (see equation (1)). The *hit rate* measures the number of people successfully detected considering the frames that certainly contain at least one person (see equation (2))

$$\text{Accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{total number of frames}} \times 100 \quad (1)$$

$$\text{Hit rate} = \frac{\text{true positive}}{\text{total number of frames with people}} \times 100 \quad (2)$$

We can see first that no detector has a very good performance by itself. As explained before, the RLPD has a high level of complexity: The occlusions are frequent, the users are sometimes partially shown in the images, and in general, they are somewhat far away from the camera. This underlines the fact the we cannot only use one `PPLP` and hope it will work in all situations.

The NiTE `PPLP` has a high accuracy and hit rate (both above 90%). This benchmark confirms that the NiTE-based `PPLP` is a useful method for detecting users in a social robot, even in challenging conditions. The tabletop `PPLP` has an average performance: It has a detection rate around 40%. Indeed, as can be seen in the sample images, the ground is hardly visible in the images acquired by the camera, which generates a poor estimation of the ground

Table 1. Benchmark results for the PPLPs with the RLPD.

PPLPs	Face detection	HOG based	NiTE based	PPM based	Tabletop based
Accuracy (%)	22.2	8.89	90.8	15.7	40
Hit rate (%)	18.6	4.76	90.6	12	41.9

PPLP: PeoplePoseList Publisher; RLPD: RoboticsLab People Dataset.

plane. We can then conclude that the tabletop PPLP is not appropriate for the robot spatial configuration used in the RLPD. In a way similar to the tabletop PPLP, the PPM PPLP has limited performance. The reasons are very similar: The limited visibility of the ground provokes a poor estimate of the transform between the camera space and the perspective map, which creates an incorrect projection into the latter. On the other hand, the face detection PPLP has low accuracy and recall. Indeed, the users are most of the time several meters away from the robot and often turn their back to the camera, which are challenging conditions for face detection. Finally, the behavior of the HOG detector is maybe the most surprising. Its detection and recall rates, which are very high in the original paper,[20] fall drastically to under 10%. As can be seen in the sample images, the users are never fully seen, and especially their legs are most of the time out of the picture frame. Since the HOG detector is trained for detecting fully visible pedestrians, it performed very poorly on the RLPD.

In conclusion, concerning the PPLP benchmarking, it turns out that each of the different PPLPs has both strengths and limitations, but no PPLP taken alone is reliable enough for robust people detection. We need to combine them using multimodal fusion, and the experimental setting will help to choose one configuration or another.

Benchmarking of user recognition algorithms with RLPD

In this section, we assess the performance of each user recognition algorithm, structured as a PPLM, as presented in "User recognition" section. To do so, for each PPLM M, for each frame n, we supply to M a pair of ground truth set of users: the current $\mathcal{P}_n$ and the previous one $\mathcal{P}_{n-1}$. Of course, in both, ground truth identities have been stripped (field person_name). PPLM M responses with the cost matrix C_M. The set-to-set assignment X is computed using the Jonker–Volgenant algorithm as presented in "User mapping, using the data fusion based on Kalman filtering" section. This assignment X is compared with the ground truth identities, which allows determining the precision of PPLM M. The results for all PPLMs are in Table 2.

We see that the Euclidean distance PPLM offers a very good compromise between computational needs and precision: It makes only 12 incorrect labelings, which is the best performance, while being very straightforward.

Table 2. Benchmark results of each PPLMs on the RLPD.

PPLM	Total labeling	Incorrect labeling	Overall accuracy
Euclidean distance	1267	12	0.997632
Face recognition	1267	284	0.775848
Height	1267	346	0.723421
NiTE	1267	39	0.958333
PHS	1267	14	0.988791

PPLM: PeoplePoseList Matcher; RLPD: RoboticsLab People Dataset; PHS: PersonHistogramSet.

The face recognition–based PPLM obtains a poor performance: close to 300 incorrect labelings out of 1267, roughly 1 out of 4. Indeed, users' faces are often not detectable by the algorithm (users turning their back or too far away for instance), and then a single face mismatch can be propagated from frame to frame over a long period of time, until the user's face is visible again.

The height-based PPLM is the most error prone. In the roughly two and a half minutes of the video, 346 incorrect labelings are made. This can be explained by users 1 and 3 having very similar heights, and for this reason, their IDs are frequently swapped, as can be seen in the confusion matrices. This result underlines that the use of height information alone, without any spatial or visual additional information, is a clue but is not enough for accurate matching.

The performance of the NiTE PPLM, powered by the NiTE algorithm, is very good considering the overall accuracy, 95.8%. We could expect a good tracking of the users by the NiTE software: Being the algorithm powering the user tracking for the XBox games using the Kinect device, there has been a lot of development and testing to ensure its accuracy. Furthermore, the experimental conditions are close to the optimal use of the Kinect recommended by Microsoft: static device, indoor environment, and limited crowd. This experimentally validates the robustness of the NiTE algorithm for user recognition in these conditions.

Finally, the PHS PPLM that matches users from one frame to another uniquely using the color of their clothes obtains a number of incorrect labelings similar to the Euclidean distance–based matching, while it does not use any spatial information about the users. Their sole color appearance proves to give some meaningful hints about who is who from one frame to another. Furthermore, the use of the Hue color component shall make this method robust to changing lighting conditions, even though this data set does

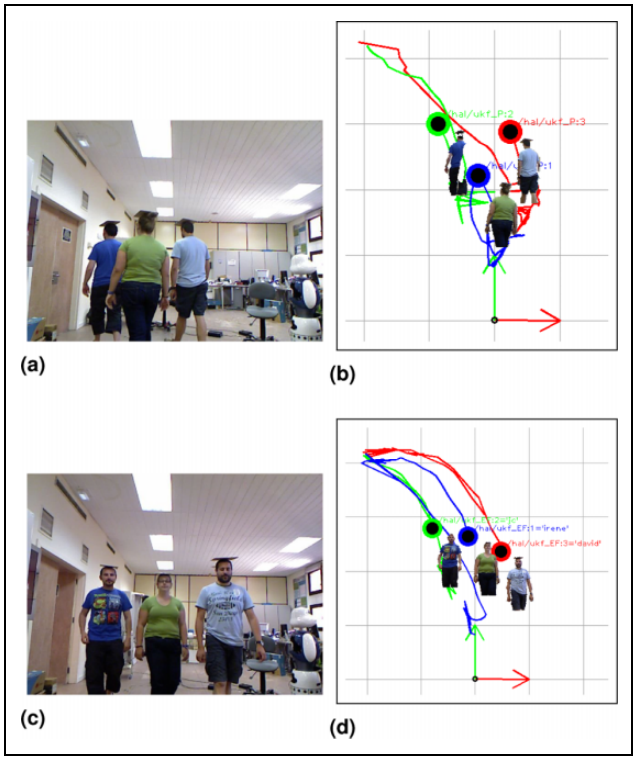

Figure 7. Sample pictures of our user awareness architecture with the RLPD. RLPD: RoboticsLab People Dataset.

not offer such challenges. This confirms the usefulness of the use of color information for user matching.

Architecture output: Sample data based on multimodal fusion

A sample `PPL` message obtained by the multimodal fusion based on both the Euclidean distance `PPLM` and the face recognition–based `PPLM` is shown in Code listing 2. It contains three users, labeled `1="irene"`, `2="jc"`, and `3="david"`. Sample images of the data supplied by some `PPLM` configurations are shown in Figure 7.

Note on user labeling. Depending on the `PPLM`s used in a configuration, this will either perform user recognition against known users or not. Figure 7 illustrates both cases with two different configurations, at two different moments of time, illustrated by the RGB frames shown in Figure 7(a) and (c). Figure 7(b) and (d) presents the different user tracks (colored lines), along with their user mask and names. As user recognition between frames is performed by the `PPLM` configuration, we can know where the user has been and so display the trail. On the one hand, in Figure 7(b), the configuration uses both PHS-based and distance-based matching. The temporary inter-frame user names are `"1"`, `"2"`, and `"3"`. These names are coherent, so that a given user has the same temporary name between frames. On the other hand, in

```
header:
  stamp: secs: 1462463744 nsecs: 723373065
  frame_id: /openni_rgb_optical_frame
method: /hal/ukf_EF
poses:
  - header:
      stamp: secs: 1462463744 nsecs: 555357811
      frame_id: /openni_rgb_optical_frame
    head_pose:
      position: x: 0.137989561449 y: 0.104732589936 z:
          ↪ 1.05867811085
      orientation: 0.822802841663
    std_dev: 1.359126091
    person_name: 1
    confidence: 1.0
    rgb:   [image raw data]
    depth: [image raw data]
    user:  [image raw data]
    attributes:
      names: ['user_multimap_name', 'initial_confidence',
          ↪ 'ukf_orien', 'ukf_speed', 'face_name']
      values: ['1', '3.22197', '2.34697', '0.103122', '
          ↪ irene']

  - header:
      stamp: secs: 1462463744 nsecs: 555357811
      frame_id: /openni_rgb_optical_frame
    head_pose:
      position: x: -0.449980712295 y: 0.379179085209 z:
          ↪ 1.95117391383
      orientation: x: 0.0 y: 0.0 z: 0.0 w: 1.0
    std_dev: 0.822802841663
    person_name: 2
    confidence: 1.0
    rgb:   [image raw data]
    depth: [image raw data]
    user:  [image raw data]
    attributes:
      names: ['user_multimap_name', 'initial_confidence',
          ↪ 'ukf_orien', 'ukf_speed', 'face_name']
      values: ['2', '3.22632', '4.3536', '0.0579029', 'jc'
          ↪ ]

  - header:
      stamp: secs: 1462463744 nsecs: 555357811
      frame_id: /openni_rgb_optical_frame
    head_pose:
      position: x: 0.690787201354 y: 0.397138127132 z:
          ↪ 1.98568838951
      orientation: x: 0.0 y: 0.0 z: 0.0 w: 1.0
    std_dev: 0.822802841663
    person_name: 3
    confidence: 1.0
    rgb:   [image raw data]
    depth: [image raw data]
    user:  [image raw data]
    attributes:
      names: ['user_multimap_name', 'initial_confidence',
          ↪ 'ukf_orien', 'ukf_speed', 'face_name']
      values: ['3', '3.22771', '2.71067', '-0.0156272', '
          ↪ david']
```

Code listing 2. A sample PeoplePoseList message.

Figure 7(d), the configuration uses the face recognition matcher combined with the distance-based matcher. The face recognition `PPLM`, when it is fed with couples of annotated faces and meaningful names (such as `"david"`, `"irene"` or `"jc"`), can set absolute names to the tracks. Consequently, in the `PPL` output message, the `PPLM` associates each temporary name with a meaningful name. Unlike the other configuration, the user labels are displayed along with their names, obtained by using face recognition. Also note that in this example frame, the three users are tracked, even though one user is partially occluding the others.

Table 3. Benchmark results for different configurations of PPLM on the RLPD.[a]

Configuration name	ID swaps	Processed frames (%)
Euclidean distance (E)	16	99
Face recognition (F)	20	98
Height (H)	37	98
NiTE (N)	16	97
PHS (P)	21	95
Euclidean + face recognition (EF)	2	98
Euclidean + height (EH)	12	99
Euclidean + NiTE (EN)	18	98
Euclidean + PHS (EP)	8	98
Euclidean + face recognition + PHS (EFP)	8	99
All (EFHNP)	22	83

PPLM: PeoplePoseList Matcher; RLPD: RoboticsLab People Dataset; PHS: PersonHistogramSet.
[a]The last column shows the rate of frames that could be processed by the fusion node while respecting the real-time constraint.

Benchmarking of multimodal fusion configurations with RLPD

The performance of the data fusion node according to its set of PPLMs was also benchmarked on the RLPD. Over 10 different *configurations* were benchmarked. The five configurations using a single PPLM for track-to-detection matching (face detection, HOG, height detection, NiTE, tabletop) were assessed. On top of these, other configurations based on combinations of these five available PPLMs were also used. The results of the benchmark are gathered in Table 3.

First, the table confirms the conclusions of the benchmarking of the PPLMs made in "Benchmarking of user recognition algorithms with RLPD" section. Among the configurations relying on a single PPLMs, some perform better than others. Namely, the Euclidean distance–based, the NiTE-based, and the face recognition–based PPLMs perform less than 20 ID swaps during the whole sequence. At the other end, the height-based PPLM taken alone is more than twice as error prone. Overall, the number of swaps is lower than the number of frame-to-frame mismatches that were done by each PPLM: Kalman filtering helps get rid of isolated mismatches. Furthermore, as the table shows, each PPLM configuration is able to process almost all PPLs messages; in other words, no message is skipped.

Second, the table also assesses the performance of the configurations relying on several PPLMs. As presented in "Approach" section, the fusion node used with such a configuration calls sequentially each of its PPLMs to obtain its corresponding cost matrix, then computes the global cost matrix. Except one 2-PPLM configuration, EN, all the others (EF, EH, EP) obtain less ID swaps (2, 12, and 8 ID swaps) than the best 1-PPLM configurations, E and N. In other words, the tracking errors made by the multimodal fusion are fewer than for any of the algorithms taken separately. The performance of the EF configuration can be underlined: Over the whole sequence and in spite of the very challenging scenario, only two ID swaps were committed, which is the best performance obtained among all the tested configurations. We can explain this improvement: the Euclidean distance tracker is efficient for most situations and solves the ambiguities correctly. However, in complicated situations, for instance, users disappearing through the door then reappearing, more sophisticated PPLMs, such as face recognition, help to solve correctly the matching.

When we increase the number of PPLMs involved in the multimodal fusion, the performance of the fusion is affected. For instance, merging the two best 2-PPLM configurations, EF and EP, into EFP (Euclidean distance + face recognition + PHS) does not improve further the performance.

Indeed, in the challenging parts of the data set, while one of the trackers correctly matches the current users to the tracks, the others get mixed up: The rate of incorrect hints given to the fusion node increases, and the final result is erroneous. Furthermore, an increased number of PPLMs generates a computational overload. This trend is confirmed when the number of PPLMs increases further: When using simultaneously five PPLMs, the number of ID swaps increases further, while the real-time capability is also affected, as shown by the decrease in the number of processed frames.

Conclusions

In this article, we tackled the challenge of giving user awareness to social robots. To do so, the problem was split into three subproblems: user detection, user recognition, and user tracking and mapping. A common data structure was designed, called PPL and shaped as a ROS message. User detection algorithms are shaped as PPLPs and user recognition algorithms as PPLMs. We integrated state-of-the-art algorithms using this format, for instance, face detection, pedestrian HOG and NiTE for user detection; and face recognition, height computation or Histogram-based matching for user recognition. User tracking and mapping is based on UKF at the user level, and for multiple users, linear assignments are used to determine which detection updates each track. The resulting data mapping is also encapsulated into a PPL. Therefore, the novelty of this article is not related to the development of new detection/recognition algorithms or improving a Kalman filter: It is focused on how different algorithms can be fused to perform a reliable user mapping during HRI. In fact, the results show that by using several parallel algorithms for detecting and matching users, and then merging their outputs by multimodal fusion, our proposal obtains a more reliable local user mapping. Additionally, the number of possible configurations of PPLPs and PPLMs is very large,

and some of them outperform others, most notably the simultaneous use of several PPLMs can increase the precision of the user mapping compared with each of these PPLMs taken alone, as seen in Table 3.

The strength of the proposed architecture lies in its modularity: It is easy to add or remove modules, and more generally to design a configuration that fits both the robot's hardware and software requirements, as well as the properties of its environment. This modularity allows the use of the architecture on a variety of platforms that differ in both their hardware capabilities and the way in which they interact with users.

A specific data set of images was created, called RLPD, which corresponds to scenarios of user detection and recognition in a realistic HRI context, acquired on a real robotic platform. The performance of all modules was assessed by using this data set. We also demonstrated that the performance of the mapping is improved by using several algorithms in parallel.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The research leading to these results has received funding from several projects: from the project called: *Development of social robots to help seniors with cognitive impairment—ROBSEN*, funded by the Ministerio de Economía y Competitividad (DPI2014-57684-R); and from the *RoboCity2030-III-CM* project (S2013/MIT-2748), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.

References

1. Jeong GM, Park CW, You S, et al. A study on the education assistant system using smartphones and service robots for children. *Int J Adv Robot Syst* 2014; 11(4).
2. Shiomi M, Shinozawa K, Nakagawa Y, et al. Recommendation effects of a social robot for advertisement-use context in a shopping mall. *Int J Soc Robot* 2013; 5(2): 251–262.
3. Kanda T, Shiomi M, Miyashita Z, et al. An affective guide robot in a shopping mall. In: *Proceedings of the 4th ACM/IEEE international conference on human robot interaction*, La Jolla, CA, USA, 9–13 March 2009, pp. 173–180.
4. Lee MK, Forlizzi J, Rybski PE, et al. The snackbot: documenting the design of a robot for long-term human-robot interaction. In: *4th ACM/IEEE international conference on human-robot interaction*, La Jolla, CA, USA, 9–13 March 2009, pp. 7–14.
5. Banich MT and Belger A. Interhemispheric interaction: How do the hemispheres divide and conquer a task? *Cortex* 1990; 26(1): 77–94.
6. Damasio AR, Damasio H and Van Hoesen GW. Prosopagnosia: anatomic basis and behavioral mechanisms. *Neurology* 1982; 32(4): 331–331.
7. Moon Y, Dutta S and Oundhakar S. *Sony AIBO: The world's first entertainment robot*. Harvard Business School Pub, 2005, Boston, USA.
8. Burgard W, Cremers A and Fox D. The interactive museum tour-guide robot. In: *Proceedings of the 15th national/10th conference on artificial intelligence/innovative applications of artificial intelligence*, Madison, Wisconsin, 26–30 July 1998, pp. 11–18.
9. Breazeal C and Scassellati B. A context-dependent attention system for a social robot. In: *International joint conference on artificial intelligence*, vol. 2., Stockholm, Sweden, 31 July–6 August 1999, pp. 1146–1151.
10. Kollar T, Vedantham A, Sobel C, et al. A multi-modal approach for natural human-robot interaction. In: *International conference on social robotics*, Chengdu, China, 29–31 October 2012, pp. 458–467.
11. Geiger J, Leykauf T and Rehrl T. The robot ALIAS as a gaming platform for elderly persons. *Lebensqualität im Wandel von Demografie und Technik—6 Deutscher AAL-Kongress mit Ausstellung*, Berlin, Tagungsbeiträge, 22–23 January 2013.
12. Dondrup C, Bellotto N, Jovan F, et al. Real-time multisensor people tracking for human-robot spatial interaction. In: *ICRA W on machine learning for social robotics*, Seattle, Washington, 26–30 May 2015, IEEE.
13. Volkhardt M, Weinrich C, Schroeter C, et al. A concept for detection and tracking of people in smart home environments with a mobile robot. In: *2nd CompanionAble workshop co-located with the 3rd European conference on ambient intelligence*, Salzburg, Austria, 18–21 November 2009.
14. Aldinhas MI and Sequeira J. Designing a robotic interface for children: the monarch robot example. In: *Proceedings of the 19th International Conference on CLAWAR*, London, UK, 12–14 September 2016, pp. 652–659.
15. Gockley R, Bruce A, Forlizzi J, et al. Designing robots for long-term social interaction. In: *2005 IEEE/RSJ IROS*, Edmonton, Canada, 2–6 August 2005, pp. 2199–2204.
16. Kanda T, Hirano T, Eaton D, et al. Interactive robots as social partners and peer tutors for children: a field trial. *Human Comput Int* 2004; 19: 61–84.
17. Portugal D, Santos L, Alvito P, et al. Socialrobot: an interactive mobile robot for elderly home care. In: *IEEE/SICE International Symposium on System Integration*, Nagoya, Japan, 12–13 December 2015, pp. 811–816.
18. Quigley M, Conley K, Gerkey B, et al. ROS: an open-source robot operating system. In: *ICRA workshop on open source software*, vol. 3., Kobe, Japan, 12–17 May 2009, p. 5.
19. Viola P and Jones MJ. Robust real-time face detection. *Int J Comput Vis* 2004; 57(2): 137–154.
20. Dalal N and Triggs B. Histograms of oriented gradients for human detection. In: *IEEE computer society conference on computer vision and pattern recognition*, San Diego, USA, 20–26 June 2005, pp. 886–893.

21. Berliner T, Hendel Z, Shpunt A, et al. Modeling of humanoid forms from depth maps, 2012. US Patent 8,249,334.
22. Howard A, Matthies L, Huertas A, et al. Detecting pedestrians with stereo vision: safe operation of autonomous ground vehicles in dynamic environments. In: *Proceedings of the 13th international symposium of robotics research*, Hiroshima, Japan, 26–29 November 2007, p. 27.
23. Fischler MA and Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm ACM* 1981; 24: 381–395.
24. Canny J. A computational approach to edge detection. *IEEE Trans Patt Anal Mach Int* 1986; PAMI-8(6): 679–698.
25. Bellotto N and Hu H. Multisensor-based human detection and tracking for mobile service robots. *IEEE Trans Syst Man Cybern B* 2009; 39(1): 167–181.
26. Belhumeur P, Hespanha J and Kriegman D. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans Patt Anal Mach Int* 1997; 19(7): 711–720.
27. Turk M and Pentland A. Eigenfaces for recognition. *J Cognit Neurosci* 1991; 3(1): 71–86.
28. Ahonen T, Hadid A and Pietikäinen M. Face recognition with local binary patterns. In: *European conference on computer vision*, Prague, Czech Republic, 11–14 May 2004, pp. 469–481.
29. Ahmed MT and Amin SH. Comparison of face recognition algorithms for human-robot interactions. *J Teknol* 2015; 72(2): 73–78.
30. Ramey A. *Local user mapping via multi-modal fusion for social robots*. PhD Thesis, Un. Carlos III, Madrid Spain, 2015.
31. Zhang T and Suen CY. A fast parallel algorithm for thinning digital patterns. *Comm ACM* 1984; 27(3): 236–239.
32. Mittal A and Davis LS. M2 tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. *Int J Comput Vis* 2003; 51: 189–203.
33. Saldivar-Piñon L, Chacon-Murguia M, Sandoval-Rodriguez R, et al. Human sign recognition for robot manipulation. In: *Mexican conference on pattern recognition*, Huatulco, Mexico, 27–30 June 2012, pp. 107–116.
34. Wan EA and Van Der Merwe R. The unscented Kalman filter for nonlinear estimation. In: *Adaptive systems for signal processing, communications, and control symposium*, 2000, pp. 153–158.
35. Mobus R and Kolbe U. Multi-target multi-object tracking, sensor fusion of radar and infrared. In: *IEEE intelligent vehicles symposium*, Parma, Italy, 14–17 June 2004, pp. 732–737. IEEE.
36. Jonker R and Volgenant A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Comput* 1987; 38(4): 325–340.
37. Igual L, Lapedriza À and Borràs R. Robust gait-based gender classification using depth cameras. *EURASIP J Image Video Proc*, Paris, France, 10–12 June 2013, 2013(1): 1–11.
38. Munaro M and Menegatti E. Fast RGB-D people tracking for service robots. *Auton Robot* 2014; 37(3): 227–242.
39. Kato H and Billinghurst M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: *Proceedings 2nd IEEE and ACM international workshop on augmented reality*, San Francisco, CA, USA, 20–21 October 1999, pp. 85–94.