Workshops at 18th International Conference on Intelligent Environments (IE2022)
H.H. Alvarez Valera and M. Luštrek (Eds.)

doi:10.3233/AISE220017

A Novel Human-Awareness Solution for Person-Following Robot's Behavior Problem Based on Proxemics

Javier DURÁN [a], Félix VIDARTE [a], Araceli VEGA [a], Pablo BUSTOS [a], Pedro NÚÑEZ [a,1]

[a] *Universidad de Extremadura. RoboLab research Lab*

Abstract. Soon robots will cooperate with humans in everyday tasks. These robots must be endowed with social skills so that their behavior will be similar to that of people. One of these behaviors is navigation: how the robot plans the route and moves through ubicomp environments. For example, a social behavior during navigation consists of detecting the position of people and evaluating with proxemics those areas where the robot can move and with what velocity. This work presents a new controller for the following ability of a socially aware person. The robot is equipped with RGB-D and laser sensors and navigates through an ubicomp environment that provides the person's position at every moment. The system initially estimates the person's position and its interaction regions at a future instant and then adjusts its path and velocity based on this estimate. Experimental results in simulated environments are included and discussed as initial results to show the performance of this proposal. We include a set of social metrics to validate the proposed results.

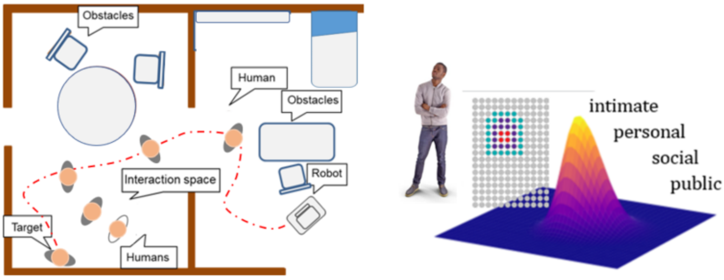
Keywords. social awareness robot, proxemics, social navigation

1. Introduction

Social navigation is a topic of enormous interest in the robotics community. How robots navigate in an environment with people determines their acceptance [1]. Many authors have focused on how the robot moves and the main factors determining this social awareness. Among them, the path planned by the robot and the velocity of its movement stand out (see recent review in [2]). In this respect, many authors rely on the use of proxemics to address these problems of social robot navigation [3,4]. Proxemics studies the spatial relationship between people during an interaction [5]; more specifically, proxemics can help us define those regions where people may feel uncomfortable during robot navigation. Thus, if we define these regions, we can introduce parameters in our navigation algorithms that improve their social behavior [4].

This paper focuses on a typical problem in many social robotics applications: the person-following task. In this behavior, a human user walks in front of the robot, and the robot has to follow the user along the path (see Fig. 1a). This behavior must consider the

[1]Corresponding Author: Pedro Núñez; E-mail:pnuntru@unex.es



(a) (b)

Figure 1. Graphical description of the person-following behavior for social robots; Navigation must take into account personal and interaction spaces; b) An asymmetric Gaussian defines personal spaces (intimate, personal, social, and public).

robot's position from the person, the velocity, and the path planning itself. In addition, it must solve the problem of detecting and tracking the person, estimating its movement and the velocity at which the robot moves. This last is not a simple task and requires the combined use of different agents ranging from perception to navigation [3].

This proposal presents our ongoing work for human-following behavior. As the main novelty, our algorithm uses the theory of proxemics to compute: i) the path that best fits the expected behavior of the robot; ii) the linear and rotational velocity of the robot controller; iii) the future position of the person at the instant when the robot replans its path. All these novelties are integrated within the SNAPE framework [6] and the CORTEX cognitive architecture [7]. Our solution is agnostic of the implementation and may be feasible to implement in other robotics frameworks.

The SNAPE framework, proposed in [6], describes a complete system for socially aware robot navigation. In our approach, perception agents compute, around people, personal interaction spaces based on proxemics theory. These spaces are defined as asymmetric Gaussians (see Fig. 1b) and take into account interactions between two or more persons from sums of Gaussians. We use this information to plan the robot's path. In addition, we modify the robot controller according to these regions and how they evolve in time. Thus, the robot's velocities will consider these spaces to improve its acceptance.

The structure of this paper is as follows: after a brief summary of the state-of-the-art in Section 2, Section 3 summarizes the SNAPE framework and the CORTEX cognitive architecture. Section 4 describes the person-following algorithm proposed in this paper. Experimental results are described in Section 5, and the main conclusions in Section 6.

2. Related work

The future generation of robots must interact with people just as humans would. Endowing the robot with social behaviors to enhance these human-robot interactions is a challenge many authors currently address. In particular, person-following skill is a part of these Human-Robot Interaction (HRI) that has more research in the autonomous robot navigation field. A person-following robot aims to follow a moving human target in three different positions: behind, to the side, and in front [8]. Our paper initially focuses on following a human user in front of the robot, adapting the robot's velocities to the personal interaction spaces defined by the proxemics [3].

The approach to the human tracking problem involves solving three specific issues: the robot hardware mechanism, the tracking of the person of interest (*i.e.*, leader), and the person-following algorithms that impact the robot controller. A differential base is the most common solution in most of today's robots regarding the hardware mechanism [8,9,10]. This platform facilitates the person's following by turning quickly in all directions. However, omnidirectional robots have been growing due to their smooth and non-invasive movement [11,12]. We present a person-following solution for a specific differential mobile platform, although its extension to any other robot is immediate. The main novelty, the use of proxemics for the definition of interaction spaces, can be applied to any other controller.

Human detection and tracking algorithms are critical components in these person-following robots (see interesting reviews in [13,14]). Some systems rely exclusively on the robot's sensors, such as lasers or RGBD cameras, using data fusion algorithms and classical tracking methods (*e.g.*, Kalman filters or particle filters) or based on neural networks [15,16]. Other techniques use sensors arranged by the environment, which are integrated with those of the robot itself to track the person [17]. Our proposal does not address this issue. For the experiments, we assume that the person's position is known fusing different sensors readings located in the environment and the own robot.

The last of the problems, the person-following controller, is also a complex problem that concerns the robot's movement through the environment. Many algorithms attempt to keep this distance fixed by simply modifying the forward and rotational velocities as a function of the distance to the person. For example, in the work presented in [10], the authors used an algorithm based on virtual spring model to mitigate the difference of movement between the human and the mobile robot. Tarmizi et al., [8] summarizes some of the classical solutions for motion controller. In [18], the authors describe a novel system for dynamic environments, which uses nonlinear model predictive control to optimize the robot's trajectory. In our article, we include as a main novelty human-aware navigation to follow the person, keeping distances according to the personal spaces of interaction. Therefore, our system improves the current ones by considering social connotations during the robot's displacement.

3. Overview of the SNAPE framework and CORTEX cognitive architecture

The solution proposed in this article uses the CORTEX robotic cognitive architecture [7]. This architecture comprises software agents that perform specific robot tasks, such as human detection, face recognition, human-robot interaction, path planning, or navigation. All these agents share information through a distributed working memory, known as Deep State Representation (DSR) [19], which is accessible to all agents. The information is consistent, quickly updated, and easily scalable. Fig. 2 shows an example for a use case of the navigation of an autonomous robot in an environment with people. The nodes and arcs show all agents' information. They include geometric and symbolic data (*e.g.*, the person in the graph has the symbolic attribute 'leader' and its position and orientation concerning the robot).

Our proposed person-following robot uses the SNAPE navigation framework [6]. This framework covers the whole spectrum: from the robot's surrounding perception to behavior planning. SNAPE is composed of five layers which are defined as follows:

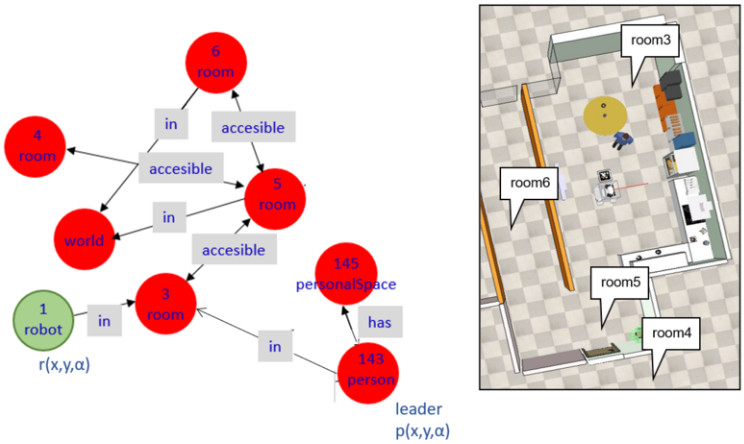


Figure 2. Deep State Representation view associated with the environment on the right. The robot and the person (leader) have symbolic 'in' links, indicating that both are in that room. Other symbolic links represent various situations (*e.g.*, accessible or has). The nodes also have geometric and symbolic information.

- **Perception layer**. This low-level layer gains access to the sensors' data deployed in the environment, including the robot's own devices. This layer comprises specific agents, from laser sensors or RGB-D cameras to audio sensors. The agents dump the information on the DSR, and other agents process it to, among other functions, fuse the measurements and obtain an estimation of the people pose in the environment.
- **Social layer**. Once the architecture detects the people's location, this layer generates a social map of the environment. A social map consists of a replica of the free space map of the robot's surrounding where social information is added. In our case, proxemics theory defines regions (intimate, personal, social, and public) that modify the original map with different costs. First, we start with a free space map that identifies the obstacles in the environment. This map is modified to add personal spaces for interaction. In the case of a single person, this space is modeled by contour lines of an asymmetric Gaussian. In the case of interacting persons, the model is based on a sum of Gaussians. By modifying the parameters of the Gaussian, we can adjust the function to different cultures and social contexts. Formally, the robot's environment is modeled by the grid $G(N,E)$ of n cells, evenly distributed in the space. Each cell n_i has two different parameters: accessibility, a_n, and weight, w_n. The accessibility of a the n_i cell is a flag variable whose value is 1 if the space is occupied and 0 if it is free. w_i, indicates the weight of the cell. Low values of w_i indicates that the robot should follow this route and elevated values of w_i advise not to use this route if there are other path with low weight. In the beginning, all boxes have the same weight 1. Later, G is actualized to add personal spaces around people, considering the type of social interaction: public, social, personal, and intimate. The social map obtained, $G'(N,E)$, also includes all the interaction spaces [3].
- **Navigation layer**. This layer is the core of the algorithm presented in this paper. The social map is used in this layer for several actions aimed at robot navigation. The ultimate goal is to have human-aware navigation, and this involves modifying

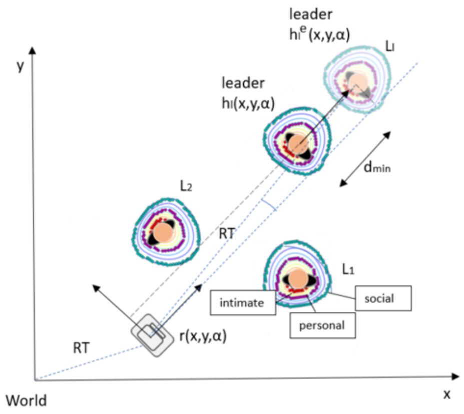


Figure 3. Top view of the socially awareness person-following problem in a dynamic environment.

the routes to respect personal spaces. In addition, it must modify robot velocities and add consecutive targets to keep the person in the robot's focus during the following task. Our solution uses a classical global planning algorithm, followed by an elastic-band based path optimization. The robot controller modifies the velocities based on the interaction spaces defined by proxemics theory. This navigation layer will be described in detail in Section 4.

- **HRI interaction layer**. This layer is in charge of starting and keeping dialogues with people in order to achieve the navigation goal. Although in our proposal we do not contemplate human-robot interaction, initiating dialogues with the person if the robot has lost the person's position is an option to be considered in future implementations.
- **Planning layer**. The last layer, more deliberative, is responsible for planning all the actions and their order of execution to achieve the defined objective. In our proposal, this layer is responsible for defining the mission of following the leader and activating each of the agents responsible for detecting and following the person, planning the route, and navigating the environment in a safe and socially accepted way.

4. Person-following robot's behavior based on Proxemics

The person-following behavior we propose in this paper acts on the controller of a differential-drive mobile robot, modifying its forward and rotational velocities. The approach's main goal is to maintain a social distance between the robot and people in the environment, following a leader to an unspecified target. The robot is kept at a distance d_r from leader, such that $d_r \geq d_{min}$, with d_{min} being the personal distance. Unlike other algorithms, our main contribution is that the robot follows the person with social awareness, *i.e.*, while navigating, the robot moves around, avoiding disturbing people during its trajectory. Fig. 3 depicts the person-following problem by a top view representation for modeling purposes. Next, we define the pipeline of our approach:

1. **Person detection and tracking**. In our system, we assume that the CORTEX architecture addresses this problem. Given a set of sensors deployed by the en-

vironment and a data fusion algorithm, the position and orientation of the leader are accessible through the DSR. Let $h_t(x,y,\alpha)$ be the position and orientation of the leader at time instant t. This position is referenced according to the robot's reference system (the edge in the DSR has the rotation R and translation T matrices to transform the person's pose into the robot's reference frame). Let H be, in addition, the set of all persons detected by the CORTEX architecture, $H = \{h_l, h_1, h_2, ..., h_n\}$. Each person h_n is defined by its position and orientation, $h_n = (x,y,\alpha)_n$. We assume that the tracking algorithm correctly discriminates the leader from the rest of the people in each frame.
Instead of working with the vector H, our algorithm estimates the future people pose, H^e to better adapt to these dynamic environments. For this purpose, first we calculate the velocity vector for each person around the robot. Let h_n^t, and $h^{t+\Delta t}$, be the positions of human n in two instants, then the velocity vector is defined as:

$$\vec{v_{h_n}} = \frac{h_n^{t+\Delta t} - h_n^t}{\Delta t} \tag{1}$$

which is characterized by its modulus $|\vec{v_{h_n}}|$ and phase $\beta_{v_{h_n}}$. Then, the estimated people pose at a particular instant of time $\Delta t'$, h_n^e is obtained from this information.

2. **Social mapping estimation and building**. This step aims to use the estimation of people's position to define those areas where robot navigation should be penalized or prohibited. First, we compute the personal interaction spaces for each $h_n^e \in H$ and then modify the grid G [3]. Each person h_n^e is modeled with an asymmetric Gaussian of the form:

$$g_{h_n^e}(x,y) = e^{-(k_1(x-x_n)^2 + k_2(x-x_n)(y-y_n) + k_3(y-y_n)^2)} \tag{2}$$

where k_1, k_2 and k_3 are coefficients used to considering the rotation of the function β_n, defined in [3].
The next step is to create groups of people whenever they interact and then define the interaction spaces in these cases. Our clustering algorithm is based on a Gaussian mixture that finds k region of interactions between people according to a proxemics-dependent function [3]. This set of regions defines where navigation is to be penalized or prohibited. Thus, the final step is to update the free space graph G values.
The contours of these forbidden areas are defined by k polygonal chains (*i.e.*, polyline) $L_k = \{l_1, ..., l_k\}$, where k is the number of interaction regions. Each curve l_i is described as $l_i = \{a_1, ..., a_m\}$, being $a_i = (x,y)_i$ the vertices of the curve, which are located in the contour of the region. The algorithm dynamically adjusts the number of vertices, m. In our proposal, contours consider the distances defined by the proxemics: intimate, personal, social, and public, and are defined according to the distances concerning the person's center. The availability a_i of all the nodes $N_i \in G$ contained in the space formed by $L_k^{intimate}$ is set to occupied, $a_i = occupied$. This means that the robot will not be able to invade this space, as it would disturb the person. For personal and social spaces, the availability of the nodes of the graph will not be modified, but its cost will be changed. The cost c_i

of all the nodes $n_i \in G$, contained in the space formed by $L_k^{personal}$ and L_k^{social} are modified, this cost being higher in the personal area than in the social area. The public space will be the rest of the graph whose costs remain unchanged.

3. **Social Path planning**. The classical Dijkstra algorithm is employed to determine the shortest path $P = \{p_1, p_2, ..., h_l^e\}$ between an initial position and a target (last leader's position estimate) to which the robot must follow. Given the robot's origin, the algorithm calculates the cost from this point to the target node, taking into account the cost of the nodes in G'. The cost of a path is the sum of the cost of the nodes that compose it. This path P is updated each time the robot visits one of the nodes p_i of the trajectory, eliminating this point from the trajectory and adding a new target that coincides with the new estimated position of the leader, h_l^e. The global planning algorithm is executed each time a new position is inserted so that the social path is continuously computed, avoiding invading personal interaction spaces.
Once the global path is planned, the trajectory responds to the dynamic changes according to an elastic band algorithm [6]. For each point p_i^t on the trajectory, the laser sensor readings define at this point a repulsive force f_r that is related to the distance from the nearest object. At the same time t, the position of the leader generates an attractive force f_a on p_i. The combination of both forces dynamically transforms the position of each point p_i^{t+1} in the robot's trajectory as:
$$p_i^{t+1} = p_i^t + f_r + f_a \tag{3}$$
4. **Person-following controller**. The last step is to calculate the appropriate forward and rotational speeds, v_f and v_r, respectively. Given the trajectory P, the robot must move between two consecutive points (p_i, p_{i+1}). In addition, the robot's velocities must be such that it can follow the person adequately. First, the controller described in this article finds the rotational velocity by considering the leader's pose estimate in its reference frame. Then, it calculates the forward velocity, also considering the personal interaction space. We define the gain κ^{v_r}, which modifies the rotational velocity. The robot initially aligns itself with the next point on the trajectory and then adjusts its forward speed. This forward velocity is multiplied by a gain defined by a sigmoid between 0 and 1 :
$$\kappa^{v_f} = 2/(1 + e^{(d_h^t - d_{min} \cdot \lambda)}) - 1 \tag{4}$$
where d_h^t is the distance to the leader, d_{min} is the distance at which the robot must stop and the gain λ is associated with the slope of the sigmoid.

5. Experimental results

This section describes the experiments that validate the proposed person-following behavior. We first describe the robotic platform and the environment used in our tests. Then we outline the simulated scenarios: i) the robot's navigation following a person in a small

(a)

(b)

Figure 4. a) First scenario. From left to right: initial set-up, social map, and path optimized by the elastic band algorithm. In this case, there are no other people in the environment; b) First scenario with people around the robot's trajectory.

environment with and without people, and ii) a complex scenario where people move around the robot's trajectory. In both cases, we will evaluate the robot's socially aware behavior considering the set of metrics defined in [3]. These metrics are (i) average minimum distance to a human during navigation, d^l_{min}; (ii) Cumulative Heading Changes, CHC; (iii) and personal space intrusions, Ψ. First, parameter tuning (*e.g.*, costs of intimate, personal, public, and social spaces) is performed by optimizing these specific social navigation metrics in controlled environments. The experiments are carried out on a PC with the following characteristic: Intel Core i7 processor with 8Gb of DDR3 RAM and Ubuntu GNU/Linux 20.10[2].

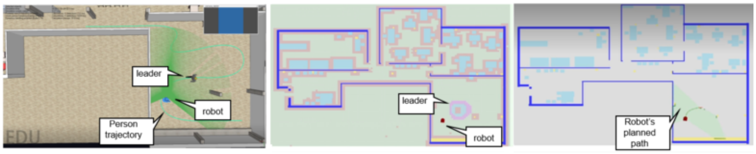
5.1. Simulated scenarios

The first scenario, simulated with the Coppelia simulator software, is a $65m^2$ apartment with a living room, open kitchen, and a corridor. In the simulated ubicomp environment, we have installed different RGBD cameras. The social robot is a directional base equipped with an RGBD camera. For people detection, we have used the CORTEX architecture. We have used the SNAPE framework in its classical version for navigation, which uses a classical Elastic Band algorithm to optimize the robot's path. In the beginning, leader is placed in the environment, and we perform two tests, one without more people in the environment (Fig. 4a) and the other with a person walking in the apartment (Fig. 4b). From left to right, we present the initial set-up, the path optimized by the classical elastic band algorithm, and, finally, the trajectory optimization with the social elastic band.

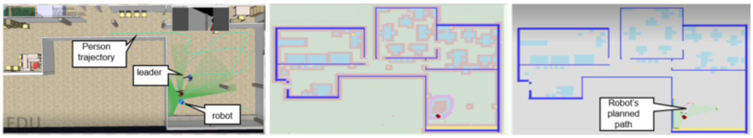
The second scenario is a larger environment, consisting of five rooms and several corridors and furniture distributed throughout the different rooms (see Fig. 5a and Fig. 5b). The experiments were conducted in a similar way to those described above. Once the leader walks, the person-following behavior begins. The robot navigates socially through the environment, adapting its speed and trajectory.

The results of our approach are shown in Table 1, where we represent the behavior of the robot during navigation in all cases. As we can observe, both the CHC values and the

[2]Readers can find a link to all the experiments in the video: `https://youtu.be/DwhXyYoINA4`



(a)



(b)

Figure 5. a) Second scenario. From left to right: initial set-up, social map, and path optimized by the elastic band algorithm. In this case, there are no other people in the environment; b) second scenario with people around the robot's trajectory.

Table 1. Result of the navigation experiment shown in Figs. 4a-5b

	Fists scenario		Second scenario	
Parameter	test1	test2	test1	test2
d^l_{min} (mm)	1757.13	2676.2	2615.82	3113.31
CHC	7,13	9.46	15.9965	15.7323
Ψ (Intimate) (%)	0	0	0.00	0,00
Ψ (Personal) (%)	0	7.23514	0.00	0.315
Ψ (Social) (%)	0	18.6047	0.00	4.50161
Ψ (Public) (%)	100	74.1602	100.00	95.18

average distance to leader d_{min} take acceptable values according to the bibliography. The heading changes are low, and $d^l_m in$ are never below the $d_m in$ (set in our experiments to one meter). The Table 1 also shows how the percentage of time Ψ that the robot invades intimate and personal spaces is zero, and most of the time, the robot navigates in the social and public regions.

6. Conclusions and future works

Human-aware person-following in populated environments is a complex problem that is currently unsolved. The present work focuses on using proxemics theory to i) plan a socially accepted route during person tracking; and ii) design a differential robot controller that applies forward and turning speeds as a function of distance to the leader and their personal spaces. The SNAPE framework provides the basis for person-following behavior. The CORTEX cognitive architecture facilitates data integration from different sources in a ubiquitous environment. We have successfully validated the algorithm in simulated environments. All software is open-source and available to the scientific community.

Future work is aimed at testing this following-person approach in real environments with people. In addition, we will work on the person tracking system with cameras distributed throughout the environment.

ACKNOWLEDGMENT

This work has been partially supported by the Feder funds and by the Extremaduran Goverment projects GR21018, IB18056, and by the MICINN RTI2018-099522-B-C42.

References

[1] Althaus, P., Ishiguro, H., Kanda, T., Miyashita, T., Christensen, H.I.. "Navigation for Human-Robot Interaction Tasks", in Proceedings of the IEEE International Conference on Robotics and Automation. Vol 1, pp. 1894, 1989, 2004.

[2] R. Möller, A. Furnari, S. Battiato, A. Härmä, G. Maria-Farinella. A survey on human-aware robot navigation, Robotics and Autonomous Systems, vol 145, (2021) doi:10.1016/j.robot.2021.103837. Elsevier.

[3] A. Vega, L.J. Manso, P. Bustos, P. Núñez D.G. Macharet. Socially Aware Robot Navigation System in Human-populated and Interactive Environments based on an Adaptive Spatial Density Function and Space Affordances, in Pattern Recognition Letters, vol. 1, pp. 72-84, (2019).

[4] Rios-Martinez, J.A. "Socially-Aware Robot Navigation: combining Risk Assessment and Social Conventions". PhD Thesis, Inria, 2013.

[5] Hall, E. Proxemics. Current Anthropology, vol. 9, no. 2-3, pp. 83108, 1968.

[6] Vega, A., Gondkar, R., Manso, L., and Núñez, P. Towards efficient human-robot cooperation for socially-aware robot navigation in human-populated environments: the snape framework. Proceedings on IEEE International conference on robotics and automation, China, 2021.

[7] Bustos, P., Manso, L., Bandera, A., Bandera, B., García-Varea, I. Martínez-Gómez, J. "The cortex cognitive robotics architecture: use cases", in Cognitive systems research, vol. 55, pp. 107-123, 2019.

[8] Tarmizi, A., Shukor, A., Sobran, N., and Jamaluddin, M. "Latest Trend in Person Following Robot Control Algorithm: A Review", in Journal of Telecommunication, Electronic and Computer Engineering, vol. 9, num 3, 2017.

[9] Shihao, Yan., Jing, T., Jian H., and Xue, A. "Model Predictive Control for Human Following Rehabilitation Robot", in Proceedings on IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO), Beijing, China, 2019.

[10] Morioka, K., Lee, J., and Hashimoto, H. "Human-Following Mobile Robot in a DistributedIntelligent Sensor Network", in IEEE Transactions on Industrial Electronics, vol. 51, num 1, 2004.

[11] Doroftei, I., Grosu, V., and Spinu, V. "Omnidirectional Mobile Robot - Design and Implementation", in Bioinspiration and Robotics Walking and Climbing Robots, Ed. In-Tech, 2007.

[12] Shabalina, K., Sagitov, A., Magid, E. "Comparative Analysis of Mobile Robot Wheels Design", in Book Series: International Conference on Developments in eSystems Engineering, pp. 175-179, 2018.

[13] Chen, Y., Tian, Y., and He, M. "Monocular human pose estimation: a survey of deep learning-based methods", in Computer Vision and Image Understand vol. 192,pp. 102–897, 2020.

[14] Ansari, M., and Kumar, D. "Human detection techniques for real time surveillance: a comprehensive survey", in Multimedia Tools and Applications volume 80, pp. 8759–8808, 2021.

[15] Algabri, R., and Choi, M. "Deep-Learning-Based Indoor Human Following of Mobile Robot Using Color Feature", in Sensors, vol. 20, num. 9, 2020.

[16] Dargan, S., Kumar, M., Ayagari, M, and Kumar, G. "A survey of deep learning and its applications: a new paradigm to machine learning", in Archives of Computational Methods in Engineering, pp. 1-22, 2019.

[17] Calderita L.V., Vega, A., Barroso-Ramírez, S., Bustos P., and Núñez, P. "Designing a cyber-physical system for ambient assisted living: a use-case analysis for social robot navigation in caregiving centers", in Sensors, vol. 20, iss. 14, 2020.

[18] Febbo, H., Huang, J., and Isele, D. "A Comprehensive Trajectory Planner for a Person-Following ATV", in Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS), 2020.

[19] Garcia, J.C., "G: a low-latency, shared-graph for robotics cognitive architectures" in Master Thesis, University of Extremadura, 2021