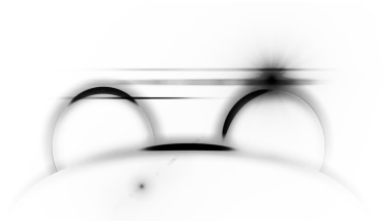

FROG

FUN ROBOTIC
OUTDOOR GUIDE

Deliverable D2.2
Path planning and execution component for efficient and human-aware navigation

Consortium

UNIVERSITEIT VAN AMSTERDAM (UvA)
YDREAMS - INFORMATICA S.A. (YD)
IDMIND - ENGENHARIA DE SISTEMAS LDA (IDM)
UNIVERSIDAD PABLO DE OLAVIDE (UPO)
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE (ICL)
UNIVERSITY OF TWENTE (UT)

Grant Agreement no. **288235**
Funding Scheme: **STREP**

DOCUMENT INFORMATION

Project

Project acronym:	FROG
Project Full Title:	Fun Robotic Outdoor Guide
Grant agreement no.:	288235
Funding scheme:	STREP
Project start date:	1 October 2011
Project duration:	30 September 2014
Call topic:	ICT-2011.2.1 Cognitive Systems and Robotics (a), (d)
Project web-site:	www.frogrobot.eu

Document

Deliverable number:	D2.2
Deliverable title:	Path planning and execution component for efficient and human-aware navigation
Due date of deliverable:	M34 - July 31, 2014
Actual submission date:	August 4, 2014
Editors:	
Authors:	UPO
Reviewers:	All Partners
Participating beneficiaries:	UPO
Work Package no.:	2
Work Package title:	Robot Control, Navigation and Location Based Content
Work Package leader:	UPO
Work Package participants:	All Partners
Estimated person-months for deliverable:	
Dissemination level:	Public
Nature:	Report
Version:	2.0
Draft/Final	Final
No of pages (including cover):	39
Keywords:	Robot Navigation, Social Interaction

Contents

List of Figures

List of Tables

1 Introduction

This document will detail the robot navigation system developed within the FROG project. As the navigation system was already deployed and demonstrated during the second review at the Lisbon Zoo with a reduced functionality, the present document will focus mainly on the new research findings and testing included into the navigation stack since then.

The navigation system shall be flexible enough to work in very different and crowded scenarios. Particularly, the system was tested in the Lisbon Zoo (see Fig. 1) for the second year review of the project as commented, and will be validated in the Royal Alcázar in Seville (see Fig. 2) at the end of the project.

The proposed robot navigation system extends state of the art navigation schemes with some social skills in order to naturally integrate the robot motion in crowded areas. Thus, this document analyses different human interactions datasets in order to model social interactions and how to use such models into the robot navigation stack. The robot makes use of these models together with two different sensors for person detection (stereo vision and laser scanner) for efficient and natural person avoidance. The proposed algorithms have been tested and validated at the UPO university in controlled environments and also in general navigation experiments resulting in a more social robot navigation behavior.

The navigation system design and implementation also focused on safe robot motion, integrating different kind of sensors for obstacle detection such as 360° 2D laser scanner or tilted laser scanner for obstacle detection in the short range. All these sensors are taken into account when building the shortest path to the robot's objective position. The safe navigation has been extensively tested in the last year of the project in both Royal Alcázar and Lisbon Zoo in real situations with good results.

The document is organized as follows: Section 2 makes a review of the state of the art and describes the proposed navigation stack. Later, Section 4 summarizes the techniques used for online person detection, essential information input for social navigation. Sections 5 and 6 present the social cost functions studied in the project and the datasets used for model fitting based on learning approaches. Thus, Section 7 describes the learning approach followed for social costs estimation based on Gaussian Processes and Inverse Reinforcement Learning. Then, Section 8 deals with the validation of the generalization of the social cost function and comparison with a Proxemics-based method, and Section 9 shows the results of the learned behavior in actual experiments involving the FROG robot. Finally, the conclusions and future open lines are discussed.

2 Requirements and high-level design

2.1 Requirements

The main objective of the robot navigation system is to command the robot to move through the environment safely and efficiently. Robot navigation is a huge research topic where many different scenarios and constraints can be considered. The FROG project focuses on the scenarios depicted in Deliverable D1.1 [8]. Two main modalities can be identified in Deliverable D1.1: waypoint navigation (including people approaching) and people guidance. This document focus on waypoint navigation while Deliverable D3.3: "Person guidance navigation component" details the people guidance subsystem.

Figure 1: The Lisbon Zoo. This scenario is mostly outdoors, including less structured buildings and roads, slopes, grass and other challenges.

The following functional requirements for the navigation component are derived from the use cases defined in Deliverable D1.1 [8]:

- Reach next way-point. The robotic guide starts to drive around from one waypoint to the next waypoint following a default circuit. During this process the robot is always detecting obstacles and avoiding collisions.
- Introduce robot. The robot stops close to the visitors and introduces itself.

According to these requirements, an autonomous navigation system will be setup into the robot with special emphasis in the social interaction. Thus, the navigation system developed in this document will command the robot in order to reach the desired position and orientation taking into account a priori information about the environment (a map of the area in which the robot was deployed) and dynamic objects detected while navigating, including persons.

It will be shown how the proposed navigation system is able to, first, detect the position and orientation of the persons surrounding the robot and, second, move accordingly to human interaction rules, reducing the gap between humans and robots in favor to a more natural interaction. This document makes use of existing datasets to estimate such human interaction models, proposing different model parameterizations. These models will be used by the robot during its execution in order to decide which motions are more human-like than others or the minimal distance to persons.

2.2 State of the art

Today, more and more mobile robots are entering our daily lives and coexisting with us. As a result, new challenges for navigation systems arise. The creation of motion plans for robots that share space with humans in dynamic environments is a subject of intense investigation in the robotic field. Robots must respect human social conventions, guarantee the comfort of surrounding persons, and maintain legibility, so humans can understand the robot's intentions [25]. This human aware navigation involves different fields as human perception, cognitive models and motion planning.

In recent years, many different robots have been developed for this purpose. Some examples

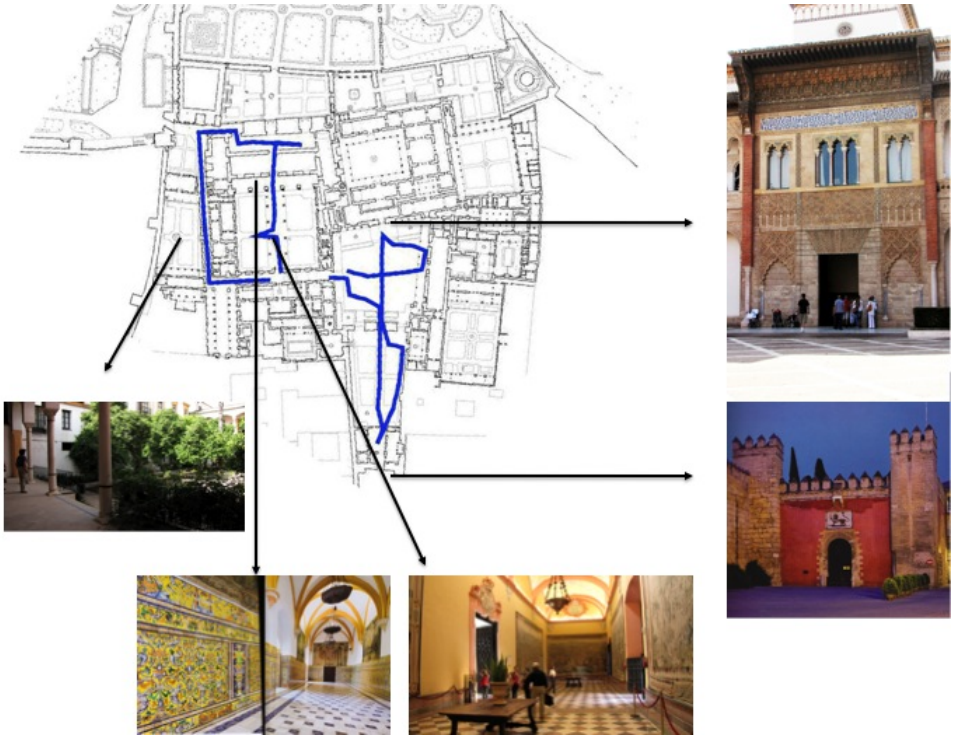

Figure 2: The Royal Alcázar of Seville. This scenario consists of a combination of outdoor and indoor places on man-made structures. The Royal Alcázar is a very crowded scenario.

of interactive museum and exposition tour guide robots, similar to FROG robot, are the pioneers RHINO [5] and Minerva [40], or more recently Robotinho [13].

In scenarios involving interaction with humans, the dynamism of the agents poses several difficult issues for robot motion planning. To ensure a safe and efficient navigation but also social interaction and social awareness when performing the robot tasks, humans have to be taken into account in the entire robot planning and navigation stack, from task planning [2], task supervision and execution [7] to path planning and execution [38, 41, 42].

Social awareness requires, on one hand, that a robot is able to detect persons, estimate their poses and differentiate them from static and dynamic obstacles. Laser rangefinders have been used for person detection and tracking [4, 6, 33]. For indoors environments, the use of RGB-D sensors has been also proposed [39, 31]. Other common approach, for indoors and outdoors, is the use of stereo vision, and in the FROG robot a stereo vision system is able to provide persons positions and orientations in real time [12, 20, 10].

Once the robot has information about the surrounding persons, the navigation stack should consider them in a different way than other obstacles in the environment to achieve a socially normative navigation. Current path planners will not solve the social navigation problem, as planners try to minimize time or length, which does not translate to social paths in general. This requires determining costs related to social compliance. Some authors [38, 21] have included costs and constraints related to human-awareness into planners to obtain socially acceptable paths, but these costs are pre-programmed. However, hard-coded social behaviors may be inappropriate [14]. Many have derived costs from Proxemics theory [18], studying the effects of crossing people in a corridor [22, 32], but the results indicate that while entering the intimate sphere of people is less comfortable, a too significant avoidance is also considered unnecessary. Moreover, as commented in [27], Proxemics is focused on scenarios in which people interact, and it could not be suitable for navigating among people.

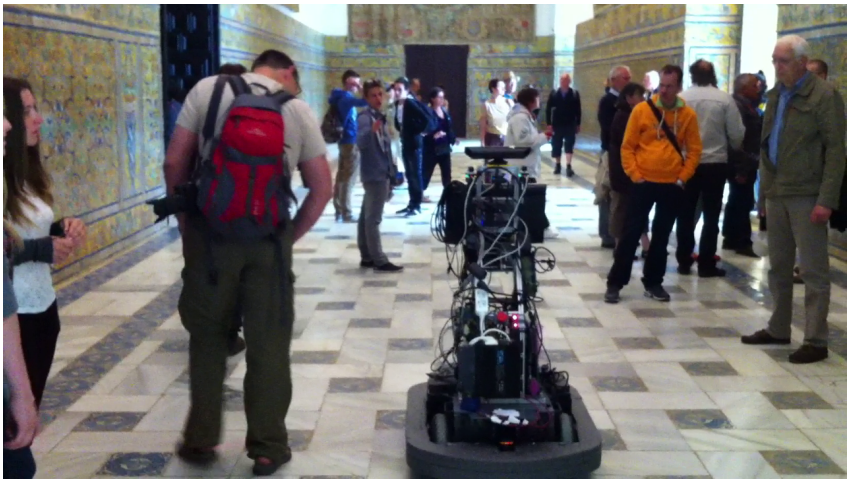

Figure 3: The FROG project aims to deploy a guiding robot with a fun personality, considering social feedback, in the Royal Alcázar of Seville and the Zoo of Lisbon. A typical situation of the first scenario is presented here.

But the benefits of human motion go beyond collision avoidance approaches. It provides cues about the intentions of the subjects that can be used to decide when and where to move. Learning and imitating motion behaviors of people can also bring benefits in terms of navigation efficiency for the robot. Besides that, the robot's motion can become more predictable, improving acceptance by pedestrians [24].

Thus, learning these costs and models from human motion data seems a more principled approach. In the last years, several contributions have been presented in this direction: supervised learning is used in [42] to learn appropriate human motion prediction models that take into account human-robot interaction when navigating in crowded scenarios. Unsupervised learning is used by Luber et al., [27] to determine socially-normative motion prototypes, which are then employed to infer social costs when planning paths. In [15], a model based on social forces is employed. The parameters for the social forces are learnt from feedback provided by users.

An additional approach is learning from demonstrations [3]: an expert indicates the robot how it should navigate among humans. In [23], the behavior of pedestrians is learnt from observed trajectories composed of observations of pedestrians and also trajectories obtained by teleoperating the robot. One way to implement learning from demonstrations is through inverse reinforcement learning (IRL) [1]. The observations of an expert demonstrating the task that we want to learn to perform are used to recover what reward (or cost) function the demonstrator was attempting to maximize. Then, the reward can be used to obtain a corresponding robot policy.

Different aspects to tackle the IRL problem have been proposed. An probabilistic method based on the principle of maximum entropy is presented in [43]. The computational cost problem is managed in [28] by using a Bayesian nonparametric mixture model to divide the observations and obtain a group of simpler reward functions. From another point of view, the authors in [26] use Gaussian processes to represent the reward instead of a linear combination of a set of features.

In [19], a path planner based on inverse reinforcement learning is presented. As the planner is learned for exemplary trajectories involving interaction, it is also aware of typical social behaviors. The authors have also considered inverse reinforcement learning for social navigation. However, while in [19] the costs are used to path plans, here we employ these techniques

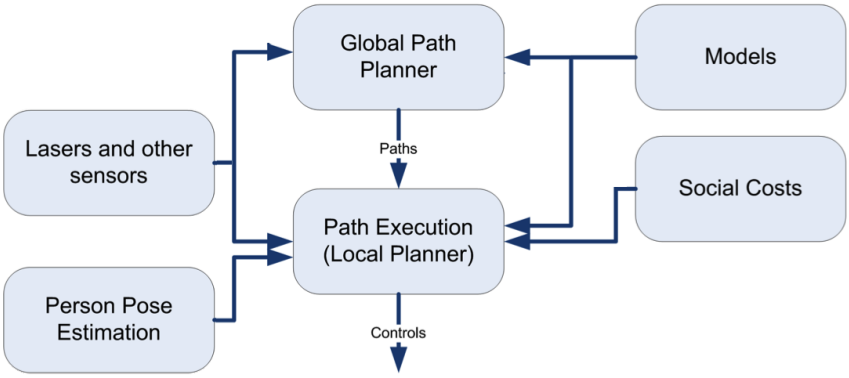

Figure 4: The navigation stack consists of a global planner, acting on global models; and a local planner acting on the most up to date information at a higher frequency.

to learn local execution policies, thus providing direct control of the robot [35]. This can be combined with other planning techniques at higher levels, while alleviating the complexity associated to learning. Furthermore, the methodology to extract the reward function from a public dataset is also described [36],.

In this document, a thorough analysis of the learning procedure is described, as well as the data used for learning. Two datasets of person motion in different scenarios are employed here to learn the cost functions. We study the generalization of the obtained reward functions by comparing the motion behavior learned from one scenario when applied in the other one. We also explore if the combination of the two training sets improves the general behavior. Furthermore, we propose a model with a simple set of features on which the reward function is depending on. Then, we augment the proposed model adding high-level features based on persons densities in different regions around the robot. Finally, we analyze and compare these approaches with a Proxemics-based cost function.

2.3 High-level design

The navigation stack of the FROG robot follows the classical separation between a global path planner and a local path execution module (see Fig. 4). The global planner employs the robot global pose and global models of the obstacles and potentially other models in order to determine a path to the goal. The local planner receives the global path and tries to follow it, by considering the most up to date sensorial information on the robot frame. This local planner generates the controls (angular and linear velocities) commanded to the robot platform.

The modules of this architecture are implemented considering the scenarios of the project. In the following section, different details on the local and global planners and sensors will be described. Furthermore, human-awareness and social behaviors will be incorporated into those modules through features and cost functions related to the persons surrounding the robot. This will be described from Section 4 ahead.

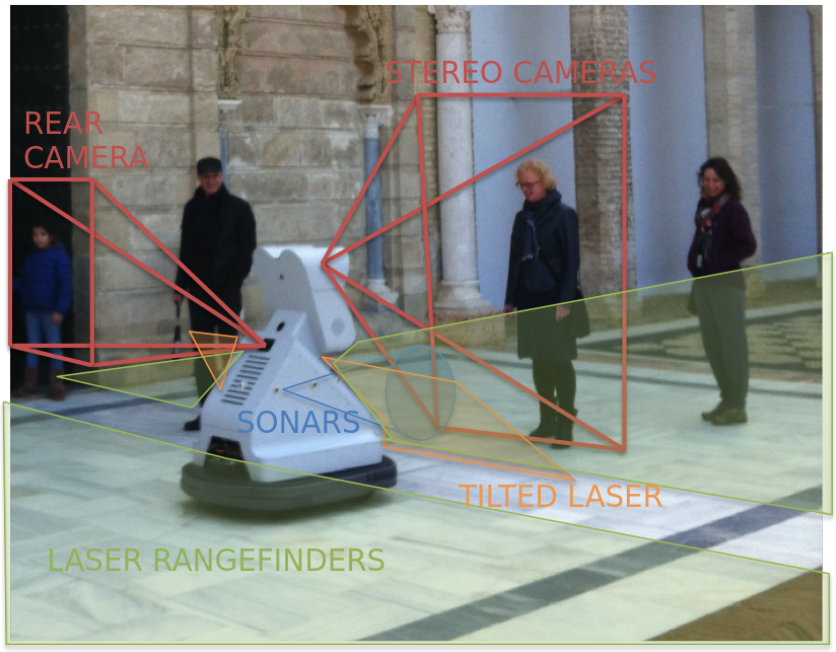

Figure 5: Robot platform and placement of sensors. Approximate localization and field of view of the sensors mounted in FROG robot. Green planes denote the front and rear laser scanner planes, it can be seen how the scanners cover 360° approximately around the robot. Orange plane stands for the 45° tilted laser scanner for obstacle detection. Red fields denote the field of view of the front stereo camera and back camera. Blue areas stand for sonar sensing areas.

3 Robot navigation stack

3.1 Robot platform and sensors for navigation

Figure 5 shows a picture of the FROG robot as deployed in the Royal Alcázar for a demonstration of its capabilities. The FROG robot consists of a skid-steering platform, with 4 wheels adapted to the scenarios considered in the project. It has an autonomy of two to four hours depending on the type of ground and the number of embedded PCs running, up to three. The robot weights 80Kg approximately and its maximum velocity is 1.6 m/s (software limited to 0.8 m/s).

The robot is equipped with a wide range of sensors for safety, localization and navigation. Deliverable D1.3 [9] describes the final position of the sensors, as well as some further details about the robot platform. Among them, the following sensors are considered for person detection and navigation:

- Odometry is computed by reading encoders and angular velocities from an MTi-G IMU from XSense
- In the final version of FROG, three laser rangefinders are considered. Two deployed horizontally forward and backwards, employed for localization and obstacle avoidance. The third laser is placed in front of the robot and tilted 45°, it is used for 3D obstacle avoidance.
- A stereo camera pair is employed for person detection, pose estimation and 3D perception.
- An additional camera is used for low-range affective computing of the interacting persons.

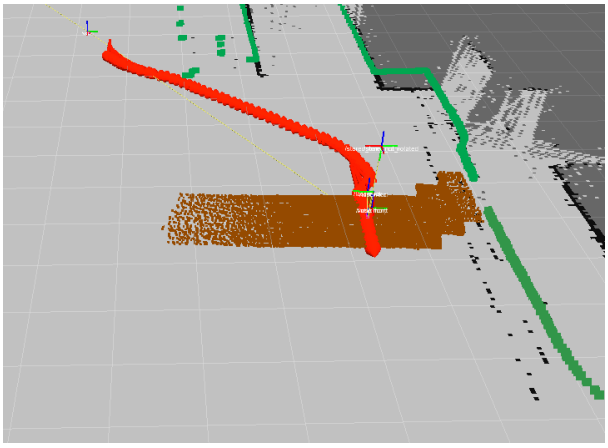

Figure 6: Integration of tilted laser for obstacle detection. The image shows the robot trajectory (red), the tilted laser integration in the last 3 seconds based on odometry (brown) and the obstacles detected with the front/rear lasers (green). It can be seen how the tilted laser allows the detection of the stairs on the robot's left-hand.

- A sonar ring surrounding the robot.

In the FROG robot we tried to dispose the sensors in order to cover as much area as possible around the robot. The frontal and real lasers cover a total angle of nearly 360° around the robot. Moreover, the sonars are employed to detect obstacles in the lateral areas of the robot which the lasers can not cover as well as elements at different height than the lasers.

A tilted laser was also installed in the robot in order to detect short range obstacles not visible by the frontal lasers (obstacles upper or below the scan plane). The laser is placed right below the screen and tilted 45° approximately. This configuration allows for detecting close objects in front of the robot. However, this sensor only provides measurements in a single plane; in order to build a small 3D map in front of the robot the laser scans are integrated through time for a fixed interval (last 3 seconds). Thus, the 6DoF robot odometry is used as laser pose estimation to build the map with low computational cost. This 3D local map serves as input to the local planner to detect possible obstacles (positive and negative) and avoid them. Fig. 6 shows an example where tilted laser (brown in the figure) allows the detection of some stairs that are below the front/rear laser level (green in the figure), this map is updated as the robot moves, keeping only the last 3 seconds of information to reduce the computational requirements of the local planner.

The stereo camera system is also used for obstacle detection and avoidance in the navigation system. The disparity maps computed by UvA software are back-projected into a point cloud of objects and these maps are included into the local planner. The sensor is suitable for detection of long and mid range obstacles overhanging obstacles in front of the robot, mainly for obstacles that are not visible at the front/rear laser scan plane. As with the tilted laser, the stereo map input to the navigation system only considers the information integrated in the last 2 seconds in order to decrease the computational resources needed to deal with the 3D point-cloud. Furthermore, the original disparity map is downsampled to a quarter of the original size. Fig. 7 shows and example in which the integration time has been significantly incremented in order to visualize the level of detail of the map.

Concerning the sensors for social navigation, the stereo camera pair is employed, as described by UvA in Deliverable D3.1 [10], to obtain the person location and body orientation estimation in front of the robot. Furthermore, an algorithm for people detection based on 2D range data from lasers will be employed to detect people in the rear and aside areas of the robot. This

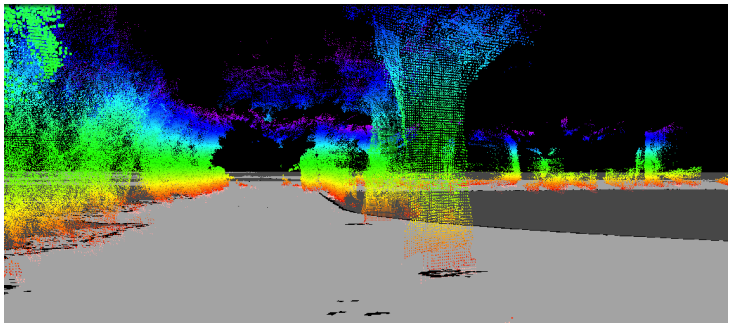

Figure 7: Example of the 3D map computed using stereo vision at Lisbon Zoo. The map used for navigation only includes the last seconds of information, because they are the most relevant for local planning and execution

way, we can be aware of all people surrounding the robot. This topic will be addressed in detail in the Section 4.

3.2 Navigation stack

The current implementation of the navigation stack extends the Robot Operating System (ROS) navigation architecture. We are mainly concerned with adapting the local planner, although significant modifications have been carried out to adapt the global planner to the FROG requirements. The global planner is based on a Dijkstra's algorithm to search through the available working area and find the best path. Using the predefined navigation map of the area, this graph search algorithm produces a shortest path tree solving the single-source shortest path problem for a graph with non-negative edge path costs (in future work we plan to consider also social constraints at this level).

Many adaptations have been implemented in the global planner to fulfill FROG requirements mainly in terms of efficiency. ROS global planner works well with small and medium size global maps but becomes slow when large maps (or with high resolution) are considered. Thus, very frequent methods as map cleaning have been refactored and improved to work with large maps. Also the recovery actions (actions taken when the robot do not find a solution for going from the current position to the specified way-point) have been refactored in order to perform faster and efficient actions. Also some aspects as the global map update policy were changed.

We consider as local planner an extension of the Trajectory Rollout algorithm [17]. The algorithm has been almost reimplemented considering computational efficiency as a major constraints. This controller predicts possible trajectories with a discrete-time simulation over a receding horizon. To ensure safe and feasible motion, the robot's kinodynamic constrains and accelerations have to be indicated correctly. The controller choses the best trajectory among the predicted trajectories by evaluating different cost functions to balance the robot different goals, such as distance to global path, distance to local goal or obstacle cost among others. We modify this technique to include additional cost terms considering social costs, which are then learned from data, as will be explained in later sections. Based on the person pose estimations received, the controller adds the corresponding social costs, which modifies the motion commands performed by the robot.

To compute the global path and to evaluate the different trajectories to follow the path locally, two 2D grid maps are used. Each map cell of the grid map contains relevant information for motion, such as the presence of an obstacle, or membership in a recognized path. For global planning, the grid map is built from the information of the predefined navigation map along with

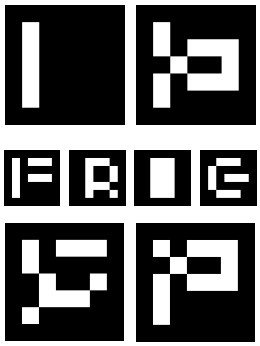

Figure 8: Left: the charging station deployed at the Royal Alcázar. The robot charging station is located in a narrow corridor in the same space as the audio guides. Right: the artificial pattern used for the docking maneuver.

the sensors data. This navigation map is similar to the localization map, but we include some limits in the map in order to restrict the areas where we do not want the robot to navigate. This way, a global path is never calculated crossing these no-go areas.

For local planning, another 2D grid map is built just from sensors data. This is a rectangular grid map of 8x8 meters around the robot. The portion of the global path to follow is mapped onto this area and the grid cells are marked with distance 0 to a path point, and distance 0 to the goal. Then, a propagation algorithm efficiently marks all other cells with their Manhattan distance to the closest point marked with zero. Moreover, the sensors' data is used to mark (insert obstacle information into the grid map) and clear (remove obstacle information from the grid map) simply changing the value of the corresponding cells. The map grid is updated with a frequency of 10 Hz in order to score trajectories efficiently as explained before.

3.3 Docking and undocking maneuvers

One of the low-level navigation behaviors of the navigation stack is devoted to the docking and undocking from the charging station (described in detail in Deliverable D1.3 [9]), see Fig. 8. As this behavior is different from the usual operation of the navigation stack, it is described here separately.

The docking and undocking maneuvers are performed by using artificial visual markers, the frontal cameras and the robot odometry. A multi-scale pattern composed by visual markers placed at different positions have been implemented based on a modified version of the ARUCO library [30] (see Fig. 8, right). The scale and distribution of markers into the visual pattern are designed to keep position and orientation accuracy during the docking/undocking maneuvers. These visual markers are used to guide the final approach to the docking station using visual servoing, using just local information and robot odometry (thus, not relying on global localization).

The undocking is performed following a predefined rectilinear trajectory out of the charging station. Visual pattern and robot odometry are merged to consistently estimate the position of the robot with respect the docking station during the task.

In both cases, local sensors (laser scanners, robot bumpers, etc) are continuously checked in order to detect obstacles in the robot's path. The robot is programmed to not avoid obstacles, instead it will be stopped every time an obstacle is closer than a given threshold. This behavior is motivated by the narrow space around the docking area and the constraints imposing by the docking maneuver (the robot should reach the docking point into an orientation envelope to properly dock-in).

3.4 Robot navigation integration

Finally, the robot navigation stack is integrated into the FROG system. A communication system to receive and send data between the different components of the overall FROG platform has been implemented. It is based on JSON lightweight data-interchange format.

In particular, the navigation and localization system, besides publishing continuously the location of the robot to the other modules, needs to receive the navigation commands from the top-level FROG behavior tree. Moreover, it has to communicate the execution state of the navigation action commanded.

The interface finally defined by the navigation stack involves the following actions:

- Going to a defined goal. The location and the desired orientation of the goal point are received. Then, the path to the goal is calculated and the navigation is initiated.
- Rotating to reach a specific orientation. This behavior is similar to the previous one, but in this case, the location of the goal is the current location of the robot causing only a turning movement.
- Approaching to a particular person. The identification number of the targeted person is used instead of the goal coordinates. Moreover, a value of safety distance is also indicated in order to stop the robot in front of the person keeping the indicated distance between them. The approaching maneuver is done taking into account the orientation of the person. So, the robot try to face the person. Furthermore, we perform a simple tracking of the person by using a time window. This way, we can change the navigation goal if the person is moving.
- Rotating to face a particular person. We only consider the orientation of the targeted person so as to rotate the robot to face the person.
- Cancel the current navigation execution and stop the robot. In this case, the navigation to the current goal is aborted, and the robot movement is stopped.
- Docking and undocking. It commands the robot to look for the markers of the docking station and perform the docking (or undocking) maneuver.

Also, a specific navigation behavior has been programmed to approach the desired goal with the best accuracy permitted. This behavior consists of decreasing the velocities approaching the goal point. A similar behavior has been included to initiate the robot movement smoothly, starting with a low velocity and increasing it to reach the commanded velocity.

4 Human-awareness

Person detection is a key-aspect for robot social navigation. The main sensor considered into the project for person detection is the People Detection algorithm developed by UvA in

Figure 9: Person detection and body orientation estimation based on stereo cameras detection at Lisbon Zoo. Pink ellipses show the full orientation estimation pdf as a polar plot. Lines inside each ellipse represent the current maximum likelihood orientation. Pink rectangle delimits person position and height.

the FROG project [10]. This algorithm has demonstrated to be flexible and accurate enough to detect persons in front of the robot, providing their positions and orientations. However, the limited field of view of the stereo cameras used as image input for this algorithm constraints the amount of information provided to the navigation stack and, more importantly, do not provide person information in the back of the robot.

A second source of information has, thus, been added to increase the detection area for persons, a 2D laser-based algorithm for person detection. This algorithm is not as accurate as the one using the stereo system, but it is reliable enough to provide good estimations of persons around the robot. We use the front and rear lasers as inputs for this algorithm, so we are able to detect persons in 360° around the robot at good frequency (about 10Hz).

Next paragraphs summarizes both algorithms and their integration into the robot navigation system.

4.1 Stereo vision based person detection

The module in charge of Person Detection and Body Orientation Estimation using stereo cameras (see Fig 9) is described in detail by UvA in Deliverable D3.1 [10]. This module captures data employing two calibrated Dalsa HM1400 XDR cameras and using the Library for Efficient Large-scale Stereo Matching (LIBELAS) [16], which offers high speed high quality stereo disparity map computations.

The system uses the stereo information to select ROIs to be processed by the person classifier. ROIs are placed on the ground plane at 46 equally spaced intervals between 2 and 25 meters from the camera. For each ROI, person classification is performed based on Histogram of Oriented Gradients (HOG) features and a linear Support Vector Machine (SVM) classifier [11]

The output of this module to the navigation stack contains the ROI, location of person on ground plane with respect to the camera, person detection probability, maximum likelihood orientation estimate and also full orientation pdf, discretized over 360° (see Fig. 9).

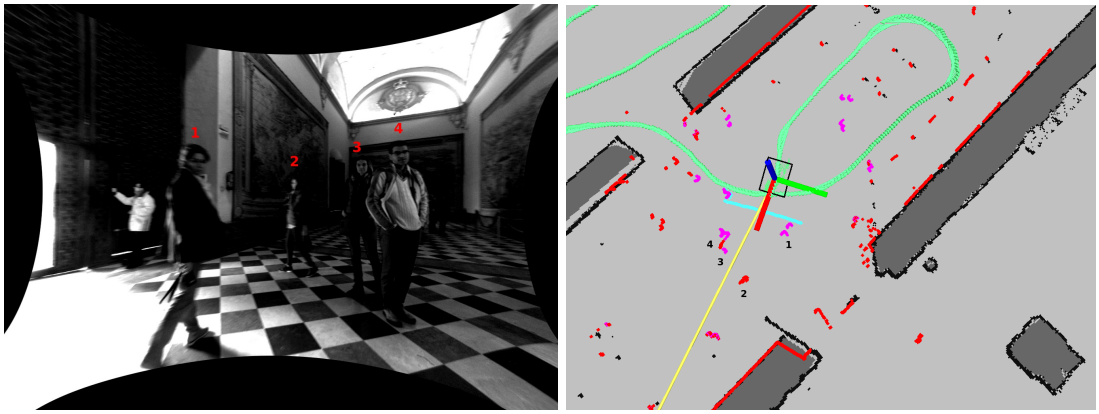

Figure 10: Example of people detection based on range measurements. On the left: stereo capture at Royal Alcazar, with some people standing looking at the robot. On the right: localization of the robot, with 360° people detection (purple), permanent obstacles (red) and robot's trajectory (light green).

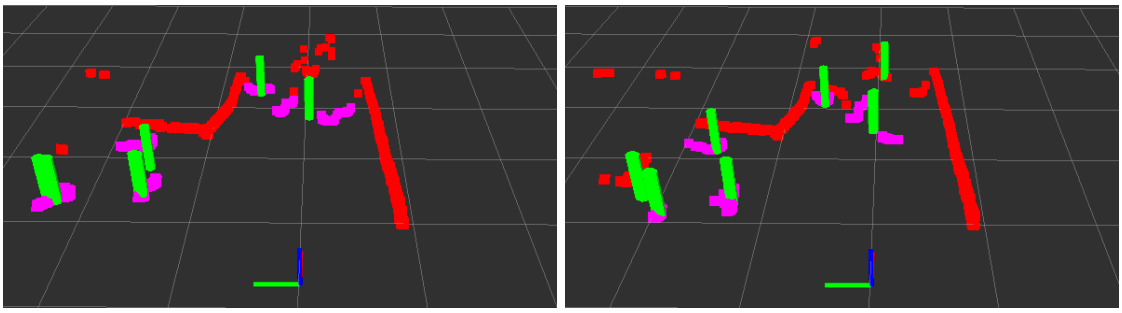

Figure 11: Laser-based people tracking results after 2 seconds. On both: people detected and tracked (green), human legs in laser segments (purple) and permanent obstacles (red).

4.2 Laser based person detection

We also process the information from the 2D lasers information in order to have further information about the surrounding persons. For that, we leverage the technique developed by Mozos et al. [29]. This algorithm is not as accurate as the UvA's module, but it is reliable enough to provide good estimations of persons around the robot. We use the front and rear lasers as inputs for this algorithm, so we are able to detect persons in 360° around the robot at good frequency (about 10Hz).

The laser-based person detection takes advantage of supervised learning [4] to create a classifier for this purpose, paying special attention to geometrical properties of range measurements corresponding to human such as size, circularity, convexity or compactness. This process divides laser measurements into segments and determines if each segment belongs to a human leg or not. Combining this algorithm with the stereo vision allows the robot to get not only a front people detection, but also a 360° human-awareness (see Fig. 10).

As this system provides information about each laser segment, determining if it constitutes a human leg or not, another two layers were developed to complement this information:

- A first layer is in charge of grouping pairs of legs in case of being near enough to be considered as a single person, considering the Euclidean distance between both legs and establishing a maximum threshold. For safety reasons, the remaining single legs are also considered as additional persons, due to situations where people stay with both legs

close enough to be identified as a single segment in laser measurement.

- The second layer performs a simple tracking of people, by using nearest-neighbor association and Kalman filtering to handle with people movement and small miss-detections, as can be seen in Fig. 11.

5 Learning a social cost function for navigation

This section describes the cost functions used by the navigation stack to model the social interaction of persons, up to the level in which we are interested for robot navigation. Instead of defining by hand this cost function, or using measures derived from Proxemics theory, here the objective is to learn the cost function from observations of humans navigating among other humans.

The learning of the cost function is accomplished by using inverse reinforcement learning (IRL, [1, 19]). IRL assumes that the person from which we want to learn can be modeled by a Markov Decision Process (MDP). Formally, a (discrete) MDP is defined by the tuple $\langle S, A, T, R, D, \gamma \rangle$:

- The *state space* S is the finite set of possible states $s \in S$;
- the *action space* A is defined as the finite set of possible actions $a \in A$.
- the state transition function T, which indicates how the state evolves when executing action a, and that is modeled by the conditional probability function $T(s', a, s) = p(s'|a, s)$
- $R(s, a)$, the reward obtained for executing action a at state s

At every step, an action is taken and a reward is given (or cost is incurred). A function $a = \pi(s)$ that maps an state to an action is called a policy (or controller). To each policy, it can be associated a *value* V_π, the expected cumulative reward following that policy $E[\sum_{t=0}^{D} \gamma^t R(s, a)|\pi]$.

$$V_\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S} p(s'|s, \pi(s)) V_\pi(s') \tag{1}$$

Associated to the value function is also the $Q_\pi(s, a)$ function:

$$Q_\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) V_\pi(s') \tag{2}$$

To ensure that the sum is finite when $D \to \infty$, rewards are weighted by a discount factor $\gamma \in [0, 1)$

A policy π^* that maximizes the *value* V^* is called an *optimal* policy. The optimal value function is the fixed point of the recursion:

$$V^*(s) = \max_{a \in A} \left[R(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) V^*(s') \right] \tag{3}$$

and the optimal policy is thus:

$$\pi^* = \arg\max_{a \in A} \left[R(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) V_{\pi^*}(s') \right] \tag{4}$$

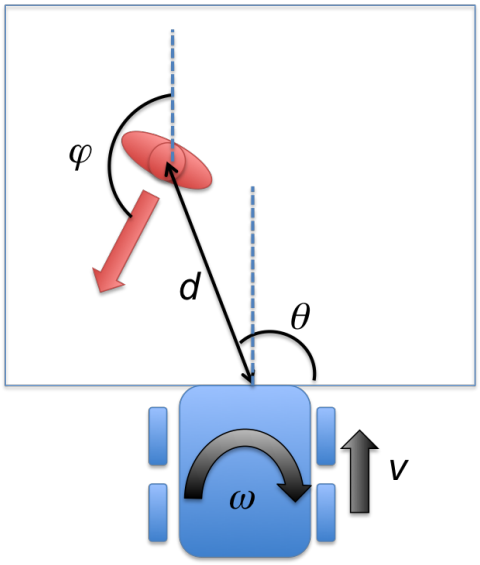

Figure 12: The state space for model 1. The state is defined as the relative pose of the person with respect to the robot, encoded as the relative position of the person in polar coordinates (d, θ), and the approach angle φ. The actions (linear, v, and angular, ω, speeds) affect how this state evolves.

The optimal value function is also related to the optimal Q^* function as $V^*(s,a) = \max_a Q*(s,a)$.

The objective of IRL is to determine the reward function $R(s,a)$ that the person (the expert) is following by observing the person acting in the real world, assuming that this expert is executing an (unknown) policy π according to the given MDP. In many cases, the reward function can be assumed to depend on a set of features $\theta(s)$, which are functions of the state.

5.1 Model 1

The most relevant aspect of the approach is to define the MDP model, and, in particular, the state and the features on which the reward function is depending on. This constitutes the main hypothesis considered here.

In principle, the actions of a person navigating among other people will depend on the state of all the persons close to the robot, plus many other factors, like obstacles and the person goal. However, considering all the persons will lead to a large (and time-variant in size) state space. In [19], this is tackled by considering the density and flow direction as features, and using them at the path planning level.

Here, the model considers the generation of the velocity controls of the vehicle. Contrary to [19], we parameterize the state on the local robot/expert frame. This allows reducing the complexity of the problem. Furthermore, in the model we consider just pairwise relative motions between two persons (a robot and a person). The state is then defined by the relative position and orientation of the person with respect to the robot, encoded as $s = \begin{pmatrix} d & \theta & \varphi \end{pmatrix}^T$ (see Fig. 12). As the parameterization is local, the pose of the robot is not considered into the state.

The effects of the actions on the state are modeled by using simple kinematic equations, and are considered to be deterministic. Uncertainties are added on the person motion part, sampling several variations on the speed and angular velocity of the person and determining its

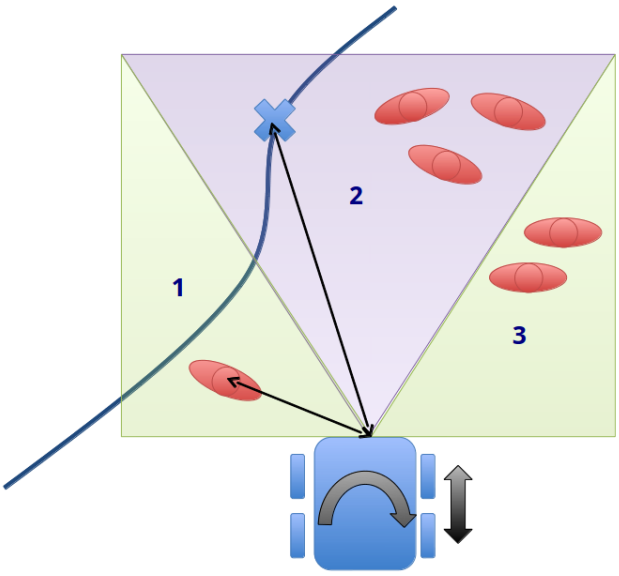

Figure 13: The state in model 2 is defined with the densities ($persons/m^2$) in the three regions in front of the robot.

future position. This way, the transition function $T(s', a, s)$ is determined.

One hypothesis that will be analyzed in this work is whether the model can be extrapolated to cases with more persons by means of the cost function learned applied to all the persons present in the scene.

5.2 Model 2

The second MDP model proposed is based on high level features to define the reward function. In particular, the person densities in different regions in front of the robot are considered to parametrize the state on the robot frame. We use the same area as model 1, but in this case, we divide it into three independent regions; one in front of the robot and two on the left and right sides (see Fig. 13).

Therefore, the state is encoded as $s = \begin{pmatrix} \rho_1 & \rho_2 & \rho_3 \end{pmatrix}^T$. The density value for each region is divided in 5 bins of range $0.25\ persons/m^2$, except the first bin that corresponds to value 0 of density. Then, the transition function $T(s', a, s)$ is determined by considering how the densities in the regions are affected by the robot motion, introducing at the same time uncertainties in the new density values due to the motion of people and the inflow of persons out of the field of view.

The development of this model aims at complementing the first model in a simple way. The idea is to try to capture other navigation behaviors in crowded environments that the first model does not consider taking into account only the closest pedestrian. We alleviate the complexity of the problem and the computational cost by dividing the learning process into different reward functions. Thus, we do not add the new densities features to the previous model state in order to obtain a larger and more complete model. Instead, we develop a new model considering only the densities features, and then obtaining the corresponding reward function. Then, we can use the reward function learned for model 1 and mix it with the new reward function obtained for the model of densities. The results will be showed in Section 8.

It is important to notice that, according to Figs. 12 and 13, only the persons ahead of the expert/robot are considered. This is actually considered during the learning process, as the sensorial space of the persons is limited (and, while persons can move the head an observe also sideways or backwards, we assume that the persons mostly employ the information about the other persons ahead). During robot execution, as indicated in previous sensors, the robot will have information about persons also by means of its rear laser.

5.3 Gaussian Process Inverse Reinforcement Learning, GPIRL

Once the models are defined, and given a set of examples in the form of expert demonstrations, $\mathcal{D} = \{\zeta_1, \cdots, \zeta_N\}$, where each demonstration consists of a path of state-action pairs $\zeta_i = \{s_t, a_t\}_{t=0}^{T}$, the objective is to recover the unknown reward function $R(s, a)$.

We consider the algorithm Gaussian Process IRL (GPIRL) [26] for solving the IRL problem. The main difference with respect to other IRL approaches is that it employs a Gaussian Process to learn a non-linear reward function over the feature space $\theta(s)$. Thus, the GP allows to extrapolate the learnt reward function to other state spaces within the domain of the features considered, if required.

The main characteristics of the algorithm are summarized here:

First of all, it employs the maximum entropy IRL model [43] as model for the demonstrations. Instead of assuming that the examples are sampled from the optimal policy π^* (which usually is not the case, as human demonstrations are often suboptimal), this model considers that the probability of an expert path ζ_i is proportional to the exponential of the differences of the rewards encountered during the path with respect to the optimal expected reward (the value V^*). Thus, given a reward function R, and denoting by Q^{*R} and V^{*R} the optimal Q and value functions for this particular reward R, the probability of executing action $a_{i,t}$ given state $s_{i,t}$ is:

$$p(a_{i,t}|s_{i,t}, R) \propto \exp[Q^{*R}(s_{i,t}, a_{i,t}) - V^{*R}(s_{i,t})]$$

The probability for a full episode is then:

$$p(\zeta_i|R) = \prod_{t=0}^{T} p(a_{i,t}|s_{i,t}, R)$$

and for the full demonstration (considering all the episodes):

$$p(\mathcal{D}|R) = \prod_{i=1}^{N}\prod_{t=0}^{T} p(a_{i,t}|s_{i,t}, R)$$

Thus, the log-likelihood for the full demonstration is given by:

$$\log p(\mathcal{D}|R) = \sum_{i=1}^{N}\sum_{t=0}^{T} \left[Q^{*R}(s_{i,t}, a_{i,t}) - V^{*R}(s_{i,t})\right]$$

This log-likelihood can be differentiated to obtain the reward R that maximizes it.

The second aspect of GPIRL is that it used Gaussian Processes [37] to introduce some structure into the reward R, which can now be expressed as a non-linear function of features. Further details can be found at [26].

Figure 14: Example images of the BIWI Walking Pedestrian dataset used for learning [34]. Left: ETH main building. Right: hotel entrance.

6 Datasets

As indicated above, it is hypothesized that learning the mentioned cost functions by observing how humans navigate among themselves will lead to socially normative behaviors.

As a source of examples and demonstrations, the BIWI Walking Pedestrians dataset[1] [34] has been used (see Fig. 14). The dataset consists of scenes of people walking in a two outdoors urban environments:

- The first proposed scenario (DS1) is a bird view of the ETH main building in Zurich (see Fig. 14, left).
- The second one (DS2), at the same dataset, is a busy sidewalk next to an hotel entrance, in Zurich as well (see Fig. 14, right).

The global data covers about 25+ minutes of observation which has resulted in about 785 observed trajectories. Each of them consists of a set of positions and velocities of all persons and the corresponding timestamps, that are manually annotated at a rate of 0.4 seconds, which is a reasonable time step for the smoothness of the trajectories.

In the first scenario, the people are walking along the sidewalk crossing with people walking in the opposite direction, resulting in two input/output flows of pedestrians in both sides of the images. In this dataset, almost 64% of the captured frames contain at least one pedestrian. In more detail, there is 1 pedestrian in 4% of frames, 2:5%, 3:8%, 4:11%, 5:8%, 6:5%, 7:6%, 8:3%, 9-18:14% of frames. By contrast, in the second dataset, the people flow may appear from anyway of the top and merge into such narrower corridor. In this case, only about the 17% of the total captured frames contain at least 1 pedestrian (1:2%,2:3%,3:1%,4:1%,5:3%,6:1%,7-11:6%).

6.1 Analysis of the data

Before proceeding with the final details on the learning data, a qualitative evaluation of the data is described here.

[1] http://www.vision.ee.ethz.ch/datasets/

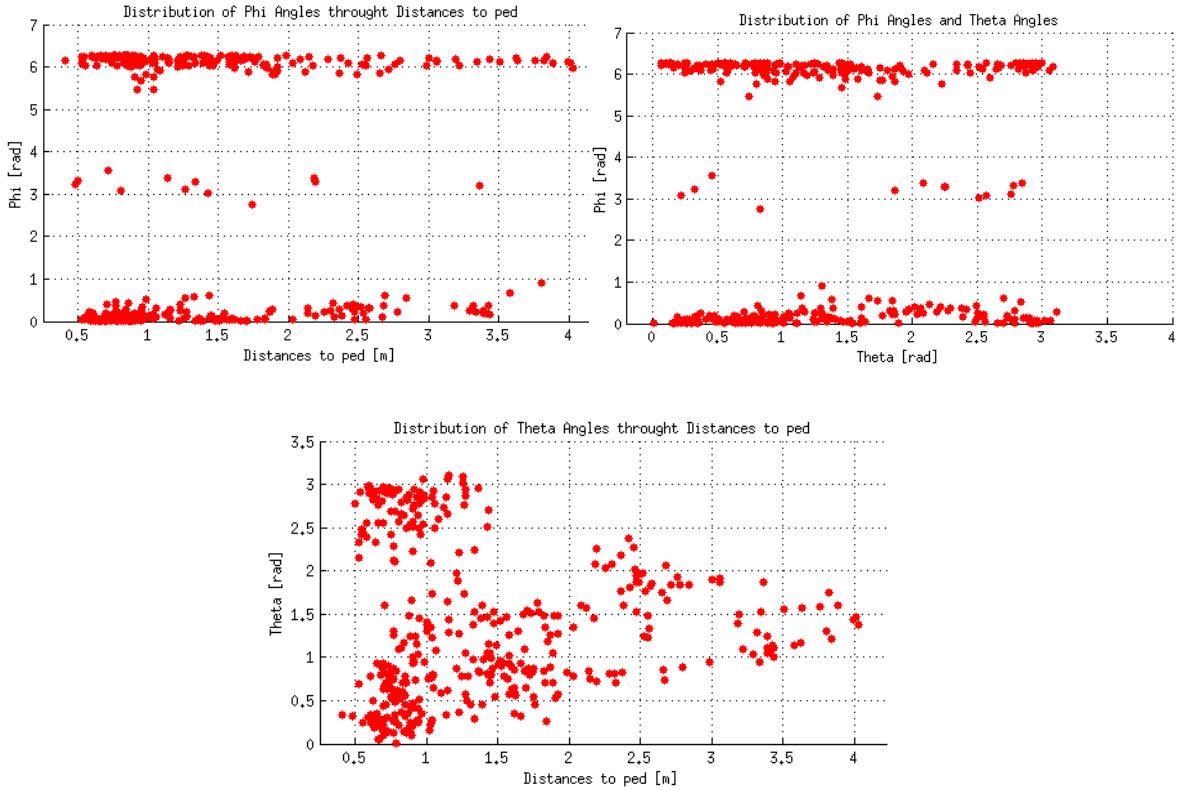

Figure 15: Training points from all the episodes for scenario DS1. Left: Approach angle φ vs. distance d in the local frame. Right: approach angle φ vs. θ. Bottom: polar coordinates (d, θ) of the closest person in the local frame.

6.1.1 Model 1

Figure 15 shows the values of the features of model 1 for the dataset 1 (DS1), the building entrance. It should be recalled that all the features are computed locally to the expert.

Figure 15 bottom shows the polar coordinates of the closest person in the local frame. Several aspects can be highlighted. First of all, the closest person can be as close as 0.5 meters, well within the personal space according to proxemics [18]. It can be also seen that if the person is below 1 meter it is typically located at the sides of the robot ($\theta \sim 0$ or $\theta \sim \pi$).

Figure 15 top shows the distance vs. relative approach angle φ, and θ vs. φ respectively. It can be seen a typically situation in this scenario, the closest person is moving in the same direction ($\varphi \sim 0$ or $\varphi \sim 2\pi$), while there are less cases in which the person cross in the opposite direction ($\varphi \sim \pi$). There are almost no examples in which persons cross with different angles, which indicates that persons try to follow locally the flow of people in terms of direction. Also, it can be noticed how approaching persons ($\varphi \sim \pi$) are located at the sides of the robot.

Similar discussions can be extracted from the training dataset DS2, shown in Fig. 16.

6.1.2 Model 2

For the density analysis (model 2), it could be seen that in ETH building scenario there is a larger number of persons at both sides of the expert than in the hotel scenario (see densities in Fig. 17 and Fig. 18). This is due to the following two main factors:

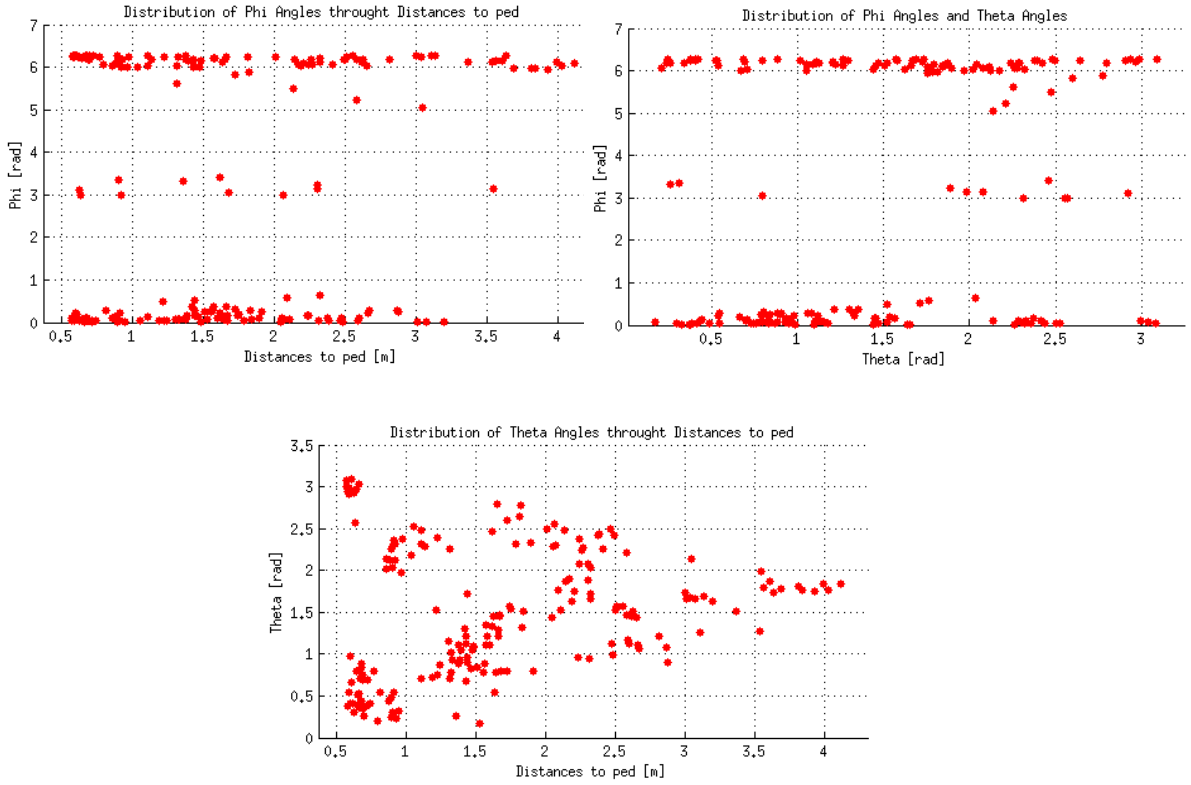

Figure 16: Training points from all the episodes for scenario DS2. Left: Approach angle φ vs. distance d in the local frame. Right: approach angle φ vs. θ. Bottom: polar coordinates (d, θ) of the closest person in the local frame.

- ETH building scenario is more crowded than hotel scenario is.
- The entrance of the ETH building is a narrow place where pedestrians have to deal with each other in a proper way to keep collisions away. Furthermore, this also explains why values of central region densities remains similiar and low at both scenarios.

Finally, it might be seem that densities features are discretized, but this effect is due to the number of persons inside each area is always an integer so the density evolves in steps of $1/area$. Notice that for density values in region 2, ρ_2, the x-axis takes from 0 to 0.5 in contrast with ρ_1 and ρ_3, that ranges from 0 to 1 per/m^2. That is because the central region of the model encompasses double the space than the others (see Fig. 13). Thus, each consecutive horizontal edge to the right corresponds with an increment of one person, up to a maximum of 6 simultaneous for these datasets. It may occur that some of these edges is missing, so the next edge counts for an increment of two persons.

7 Description of the learning process

7.1 Experts

In our case, we employ the dataset to gather the examples from experts in the task of navigating among persons. Some persons are selected as "experts" among the pedestrians that are moving in the dataset. For each point in the trajectory followed by the person we extract:

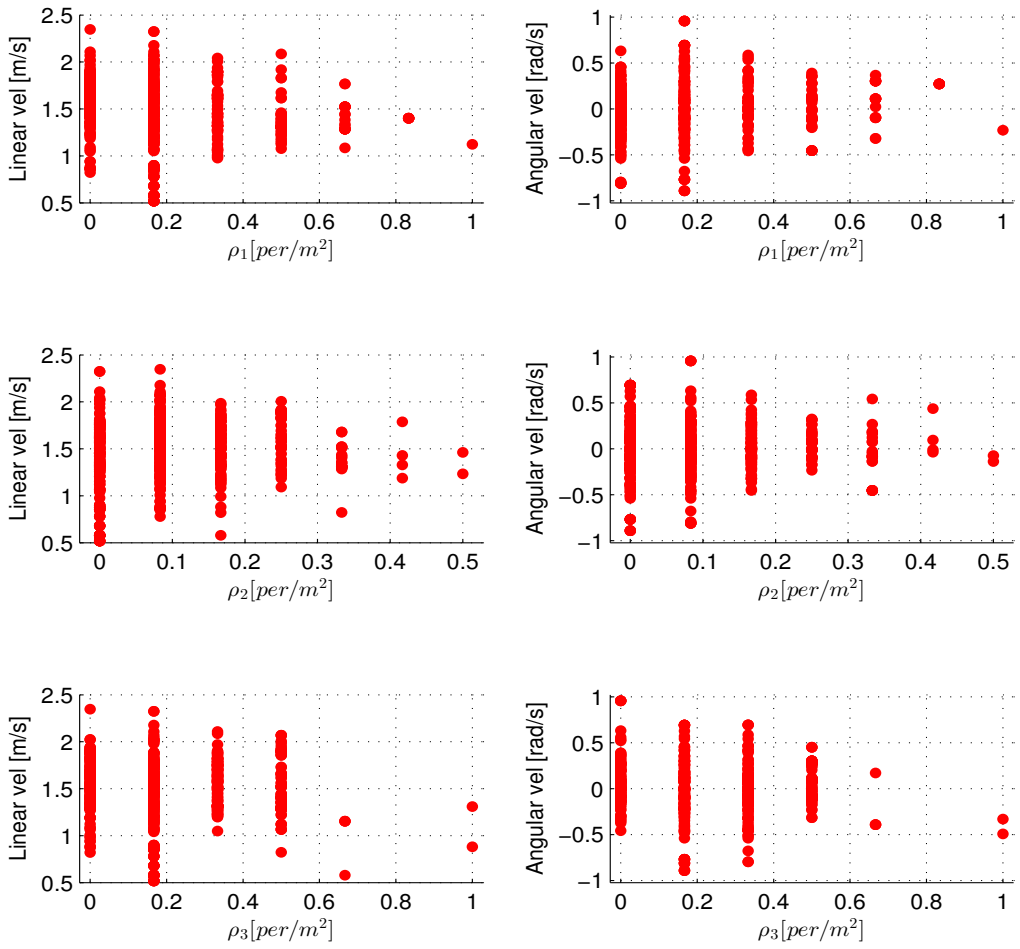

Figure 17: Density and actuation values of the experts for scenario DS1

- For Model 1, the state $s_i = \begin{pmatrix} d & \theta & \varphi \end{pmatrix}^T$ of the closest person within the local planning zone. This local environment (see Fig. 12) is defined as the region used for local planning on the robot, and it is defined as a rectangular region of 4x4 meters (4 meters in front and 2 meters at each side of the robot).
- For Model 2, the state $s_i = \begin{pmatrix} d & \theta & \varphi & \rho_1 & \rho_2 & \rho_3 \end{pmatrix}^T$ is completed with the values of person densities of the three regions considered (see Fig. 13). Here a rectangular region of 4x6 meters is defined (6 meters in front of the robot to gather more pedestrian information).
- The action performed by the expert at the same time instant (for both models). In the particular implementation considered, the action space consists on the linear and angular velocities $a_i = \begin{pmatrix} v & \omega \end{pmatrix}^T$, in order to easily transfer them to the robot. The angular velocity ω is computed by measuring the change of orientation between consecutive poses of the expert.

When the closest person abandons the local planning region, the trajectory $\{s_i, a_i\}_{i=1}^N$ is stored as one episode for the training phase. A new episode is created for the next person. In order

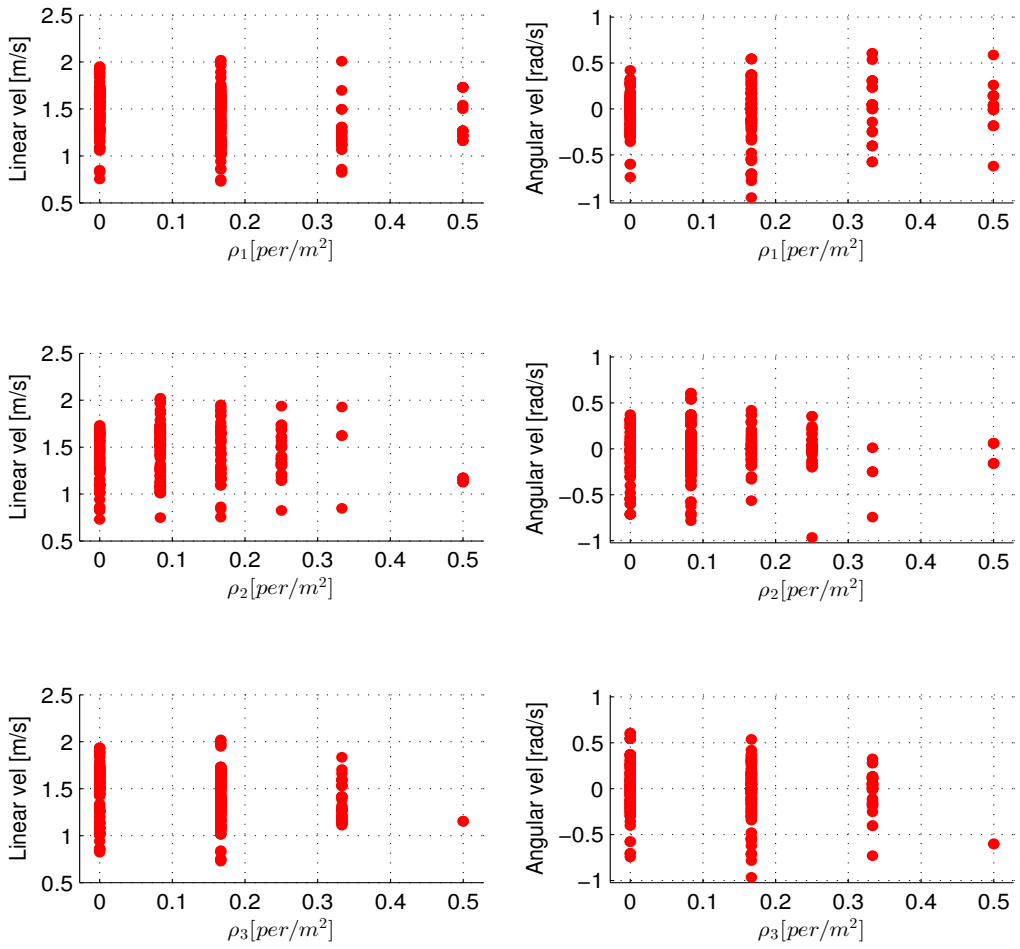

Figure 18: Density and actuation valuesvalues for scenario DS2

to have equal experiments, the number of samples is equalized for each episode, by dividing them into several episodes if needed.

From each dataset, only moving pedestrians for which at least one person is within the local planner region for at least 6 time steps are selected as "experts". Furthermore, we impose that these pedestrians have to move at least 2 meters from their starting point. Both conditions allow us to focus on interesting samples of pedestrians making social navigation. As a result, a training set per dataset is obtained. One of them is a set of 103 episodes from 51 different persons, and the other is a set of 47 episodes of 28 different persons. They are used to learn the reward function and the rest of persons will be used in the evaluation to validate the estimated function.

7.2 Discretization

The GPIRL algorithm uses a discrete MDP as model. Therefore, the state and actions spaces are discretized. The local space used for the local planner is discretized as follows:

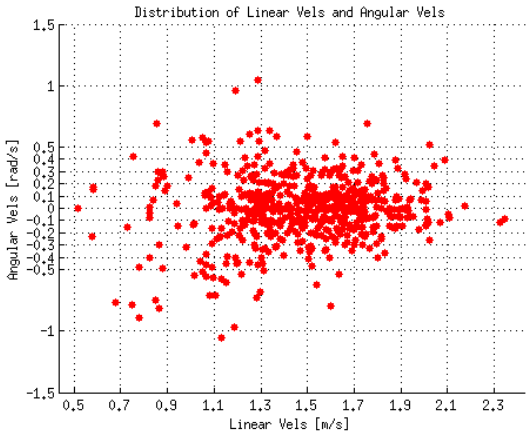

Figure 19: Actions (linear and angular velocities) of the training samples for both scenarios DS1 and DS2. The discretization bins employed are also shown.

- The distance d is discretized into 11 bins of 0.5 meters.
- The relative angle $\theta \in [0\ \pi]$ is discretized into 6 bins of 0.62 rads.
- The person relative orientation $\varphi \in [0\ 2\pi)$ is discretized into 8 bins of 0.69 rads.
- The density value $\rho \in [0\ \infty)$ is discretized into 5 bins of 0.25 $persons/m^2$, starting at zero value that means there is no person into the region considered and grouping greater values than 0.75 $persons/m^2$ at a unique bin. This could be seen as a coarse approximation, but it is enough to enclose a valid behavior in order to avoid those regions with high density values.

Figure 19 shows the linear and angular velocities for all the persons considered as experts in the dataset, for both scenarios. The angular velocity is computed by looking at the change of orientation of the linear velocity vector between two time instants. The action space is discretized considering the behavior of experts in the dataset (see Fig. 19). As we are learning how to move among other people, only persons moving over certain velocity are selected as experts; the linear velocity is discretized into 8 values in $v \in [0.7\ 2.1]$ m/s. The angular velocity is discretized in other 11 values in $\omega \in [-0.5\ 0.5]$ rad/s. Finally, in our case the state is used directly as features to learn the reward function.

8 Validation

By using the previous examples and the IRL algorithm, a reward function is obtained that associates a scalar value to each state. As a first evaluation of the learnt reward function, we compare the actions taken by a person of the dataset with the commands given by the optimal policy obtained by solving the MDP model described above using the learned reward function.

The comparison is performed as follows: at each point of the trajectory of the selected person, the state is computed, as well as the action that should be applied according to the policy, and the actual action performed by the person. If the policy fits perfectly with the person behavior, the actions of the MDP will be very similar to the actual ones. The actions from the MDP are not applied so that in the next point the state is the same in both cases.

In order to apply the action given by the policy for Model 2, in which the state is divided into two sub-states, i.e. $\begin{pmatrix} d & \theta & \varphi \end{pmatrix}^T$ and $\begin{pmatrix} \rho_1 & \rho_2 & \rho_3 \end{pmatrix}^T$ (see 5.2), the following procedure is applied: the

Table 1: Model 1-based policy vs. Proxemics-based policy. Mean errors.

	Linear Vel(m/s)		Angular Vel(rad/s)	
	Model 1 closest	PRX closest	Model 1 closest	PRX closest
E1	0.267 ± 0.184	0.363 ± 0.217	0.078 ± 0.059	0.094 ± 0.076
E2	0.329 ± 0.214	0.371 ± 0.231	0.100 ± 0.079	0.102 ± 0.082
E3	0.300 ± 0.198	0.367 ± 0.216	0.086 ± 0.059	0.094 ± 0.078
E4	0.255 ± 0.187	0.279 ± 0.157	0.074 ± 0.068	0.111 ± 0.090
E5	0.280 ± 0.195	0.361 ± 0.216	0.074 ± 0.053	0.095 ± 0.077
E6	0.258 ± 0.160	0.269 ± 0.177	0.069 ± 0.053	0.089 ± 0.080

Table 2: Model 1-based policy vs. Model 2 -based policy. Mean errors.

	Linear Vel(m/s)		Angular Vel(rad/s)	
	Model 1 closest	Model 2 closest	Model 1 closest	Model 2 closest
E1	0.267 ± 0.184	0.217 ± 0.2	0.078 ± 0.059	0.074 ± 0.057
E2	0.329 ± 0.214	0.231 ± 0.201	0.100 ± 0.079	0.102 ± 0.079
E3	0.300 ± 0.198	0.249 ± 0.246	0.086 ± 0.059	0.096 ± 0.060
E4	0.255 ± 0.187	0.269 ± 0.213	0.074 ± 0.068	0.073 ± 0.060
E5	0.280 ± 0.195	0.248 ± 0.222	0.074 ± 0.053	0.091 ± 0.055
E6	0.258 ± 0.160	0.218 ± 0.170	0.069 ± 0.053	0.058 ± 0.053

two sub-states are computed at each point so both policies are indexed by its own. This results in two policy vectors that contain actions likelihoods, so they are multiplied and normalized, and finally the most probable action is selected.

We compute the mean errors for the linear and angular velocities of each person of the dataset that was not used for training. In order to eliminate the effects of discretization on the actions, the actual actions carried out by the person are also discretized. Furthermore, the calculations are performed in 6 different cases, based on the scenario used to obtain the pedestrian motions and the scenario used to test the policy obtained by solving the respective MDP.

- The first case is training with the data of scenario DS1 and evaluation in the same scenario (E1).
- The same evaluation, but with scenario DS2, is denoted E2.
- The two next experiments evaluate the behavior obtained by training in one scenario and testing in the other one (Experiments E3 and E4).
- The last two experiments (E5 and E6), perform a training mixing training samples from scenarios DS1 and DS2, and evaluate the results in both scenarios respectively.

8.1 Local planning comparison

To evaluate the results of the model presented, we first compare it with an heuristic cost based on Hall's Proxemics (PRX) theory [18]. A cost function modeling the personal space is implemented as two Gaussians distributions as in [22]. The first function is asymmetric and placed

Table 3: Comparison for Model 1-based policy generalization: closest pedestrian and all pedestrians. Mean errors.

	Linear Vel(m/s)		Angular Vel(rad/s)	
	Model 1 closest	Model 1 all	Model 1 closest	Model 1 all
E1	0.267 ± 0.184	0.271 ± 0.189	0.078 ± 0.059	0.097 ± 0.076
E2	0.329 ± 0.214	0.325 ± 0.217	0.100 ± 0.079	0.097 ± 0.076
E3	0.300 ± 0.198	0.298 ± 0.190	0.086 ± 0.059	0.085 ± 0.067
E4	0.255 ± 0.187	0.303 ± 0.199	0.267 ± 0.184	0.271 ± 0.189
E5	0.280 ± 0.195	0.316 ± 0.190	0.074 ± 0.053	0.077 ± 0.064
E6	0.258 ± 0.160	0.316 ± 0.204	0.069 ± 0.053	0.096 ± 0.080

in the front of the person with $\sigma_x = 1.20m$ and narrower space in the sides $\sigma_y = \sigma_x/1.5$. The second Gaussian is placed in the back of the person with $\sigma = 0.5\sigma_x$.

A new reward function is then obtained from this cost and used to determine a Proxemics-based policy by solving the proposed MDP model over this reward. This way, we will compare both policies in the same framework.

The errors committed in all those approaches are presented in Table 1. It can be seen how the learnt reward function (the Model 1 IRL-based policy) obtains in mean a closer behavior than the Proxemics approach. The main difference can be observed in the linear velocity commands.

Similarly, in Table 2, it is shown the error comparison between Model 1 and Model 2. In this case, the addition of density values to derive the policy improves the expected behavior in terms of linear velocities, while angular velocities are similar. Thus, these results encourage us to keep tracking a better description of the social navigation task in terms of feature and state spaces. So far, Model 2 is closer to the observed behavior. On the other hand, there is a large variability on the errors, which indicates that the model based on just the closest person cannot account for all the information used by humans to navigate among others.

8.2 Generalization of one pedestrian model to all pedestrian

In the real world, persons do not move considering just the closest pedestrian when walking through the streets. Normally, we take into account all the persons in front of us up to some meters. This is why the reward function presented before must be completed with the information from other pedestrians in the local planning area of the robot. Thus, the previous algorithms are modified such as the action taken by the robot in this case is the one that maximizes the sum of the value function of the MDP for all the pedestrians on the local navigation area.

In Table 3 it can be seen the comparison between the Model 1, considering just the closest pedestrian (Model 1 closest) and its generalization to all pedestrians (Model 1 all). It can be seen that there are no significative differences with respect to the previous cases, and even worsens the performance in some of the cases. These results suggest that just a linear combination of the proposed model of one pedestrian does not account for all the necessary features to be generalized to all pedestrians case and new features should be taken into account.

In case of considering the Model 2, the generalization from 1 to all pedestrians is computed over the first substate $s_i = \begin{pmatrix} d & \theta & \varphi \end{pmatrix}^T$, since density values are always evaluated over all

Table 4: Comparison for Model 2-based policy generalization: closest pedestrian and all pedestrians. Mean errors.

	Linear Vel(m/s)		Angular Vel(rad/s)	
	Model 2 closest	Model 2 all	Model 2 closest	Model 2 all
E1	0.217 ± 0.2	0.215 ± 0.172	0.074 ± 0.057	0.075 ± 0.059
E2	0.231 ± 0.201	0.229 ± 0.208	0.102 ± 0.079	0.108 ± 0.046
E3	0.249 ± 0.246	0.225 ± 0.194	0.096 ± 0.060	0.089 ± 0.050
E4	0.269 ± 0.213	0.198 ± 0.195	0.073 ± 0.060	0.087 ± 0.058
E5	0.248 ± 0.222	0.258 ± 0.195	0.091 ± 0.055	0.072 ± 0.049
E6	0.218 ± 0.170	0.200 ± 0.196	0.058 ± 0.053	0.073 ± 0.066

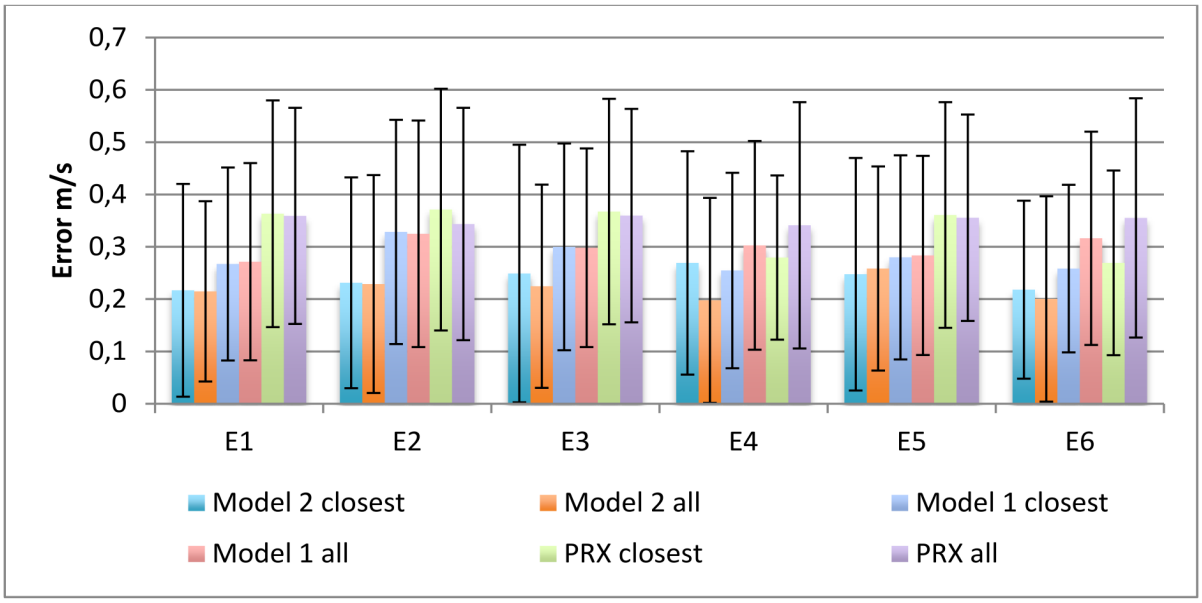

Figure 20: Comparison of the different policies and evaluations: error on linear velocity with respect to the ground truth actions.

pedestrians. Taking this into account, the results are shown in Table 4. It can be seen that the addition of the density features slightly enhances this generalization.

Furthermore, similarly to the comparison made in Table 2, it is worth to highlight that the addition of density features has improved the performance obtained across Model 1-all (second column in Table 1) and Model 2-all (second column in Table 2). This emphasizes the idea that new features could be taken into account to get a closer human-like behavior. Nonetheless, for this first approximation, until experiments will be done, it has been opted for enclose the actual set of features up to new conclusions are extracted.

Finally, Figs. 20 and 21 shows graphically the comparison for all policies and experiments.

8.3 Evaluation in different scenarios

Another aspect that we evaluate is how transferable the reward function is between different scenarios (DS1 and DS2) with different conditions such as space, crowd or crossing directions

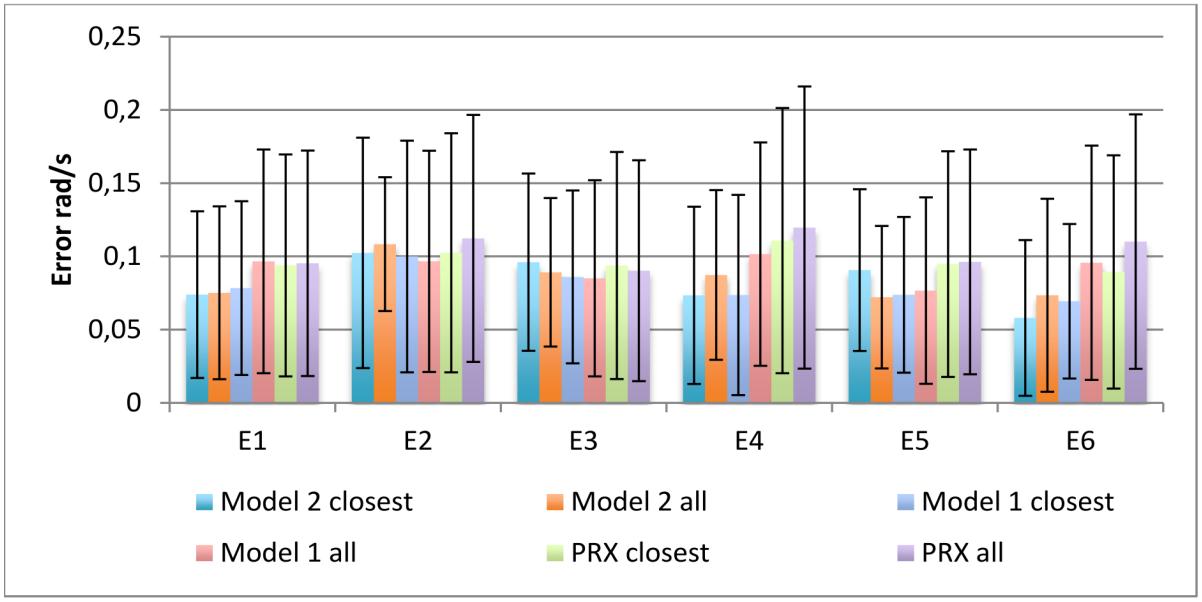

Figure 21: Comparison of the different policies and evaluations: error on angular velocity.

Table 5: Scenarios comparison for DS1

Model 2 closest	Policy DS1	Policy DS2	Policy DS1+DS2
Linear Vel(m/s)	0.217 ± 0.200	0.269 ± 0.213	0.248 ± 0.222
Angular Vel(rad/s)	0.074 ± 0.057	0.073 ± 0.060	0.091 ± 0.055

of pedestrians.

Table 5 shows the errors in actions according to the testing scenario and the scenario in which the policy is learnt. We also include a mixed policy obtained with training samples from both scenarios. It can be observed that there are not relevant differences in the errors between scenarios, and moreover, the policy obtained from the mixed samples does not improve the results significantly. The policy learnt in one scenario can be used in the other. By observing Fig. 15 and 16, it can be seen that, in this particular case, both scenarios are quite similar.

So we consider that a proper evaluation would require further testing with a greater variety of walking conditions between pedestrians and the reformulation of some of the parameters of the model.

9 Experiments

In this section we show actual experiments performed with the robotic platform, integrating the subsystems described above. In these experiments, the robot autonomously navigates from a starting point to a given waypoint, encountering persons in his path during the execution.

The following scenarios are considered to perform the experiments:

- Static scenario. The robot has to cross a controlled scenario with pedestrians standing, talking to each other. It is assumed an static scenario in the sense that people do not move from their initial position (see Fig. 22).

Table 6: Scenarios comparison for DS2

Model 2 closest	Policy DS1	Policy DS2	Policy DS1+DS2
Linear Vel(m/s)	0.249 ± 0.246	0.231 ± 0.201	0.218 ± 0.170
Angular Vel(rad/s)	0.096 ± 0.060	0.102 ± 0.079	0.058 ± 0.053

Figure 22: Experimental setup. This is the scenario selected for the static experiments. It corresponds with an outdoor courtyard at the UPO and ensures a controlled environment for preliminary and safety tests.

- Dynamic scenarios. We use the same scenario than before, but in this case the pedestrians are moving and crossing the area. Moreover, we will show results over another non-controlled scenario as the main corridor at Pablo de Olavide University.
- Lisbon Zoo scenario. We will show some statistics about the results obtained during the 2nd year review in the Lisbon Zoo. It should be recalled that those experiments were performed without including the "social costs".
- Royal Alcázar scenario. We will show some statistics about the results obtained at the Royal Alcázar, using the full navigation stack and social costs.

The experiments will evaluate the approach by comparing a classic local planner [17] with the same planer augmented with the learned costs and the costs based on Proxemics. When weighting the local trajectories, these additional cost functions are considered, taking into account all the persons present. The results are compared using as metrics the total distance traveled towards the goal (TD), the time executing the waypoint (T) and the minimum and medium distance to the persons (Min PD and Mean PD respectively) along the path. By computing the execution time and total distance traveled we can assess the effectiveness to reach the goal, while the distances to persons let us know how the personal space is conserved and its effect in the deviation distance from the global path.

9.1 Experiments with static pedestrians

We have performed 4 runs for each approach, maintaing the same configuration between the different runs. Table 7 summarizes the metrics for each approach. We show the result for the basic navigation without "social component" (No social), results for the proxemics approach (Proxemics), results with the social cost obtained with the model 1 (One Ped), results with the

Table 7: Experimental results in a static scenario

	T (s)	TD (m)	Mean PD (m)	Min PD (m)
No social	50.65 ± 0.21	21.76 ± 0.29	2.82 ± 0.06	0.82 ± 0.03
Proxemics	71.00 ± 1.55	21.09 ± 1.04	3.69 ± 1.69	1.05 ± 0.18
One Ped	54.28 ± 3.39	20.50 ± 0.22	5.19 ± 1.06	1.29 ± 0.18
All Ped	58.85 ± 0.21	22.68 ± 0.19	5.22 ± 1.69	1.69 ± 0.04
Densities	51.05 ± 0.92	24.53 ± 0.01	4.50 ± 0.21	0.87 ± 0.02
OnePed + Dens	53.27 ± 3.17	21.19 ± 1.56	4.00 ± 1.82	1.01 ± 0.29

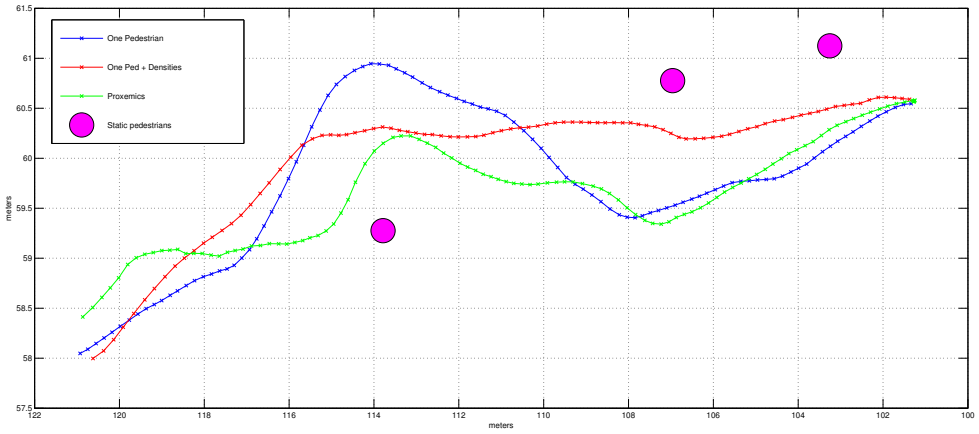

Figure 23: Example of the trajectories performed by the robot according to the approach used.

model 1 generalization from one pedestrian to all (All Ped), results for model 2 (Densities) and finally, results taking into account the costs derived from both models 1 and 2 (One Ped + Dens).

The spatial disposition of the three static pedestrian employed in the experiment is presented in Fig. 23. An example of the trajectories performed by the robot by using different approaches is also showed. At first sight, it can be seen clearly some differences between the trajectories depending on the approach employed. However, we will better explain the results by analyzing all the tests performed in the experiments, which are presented in the table 7:

- No social. The navigation without the social component try to optimize time and distance traveled. So, the average distance to the pedestrian is the lowest of all the approaches.
- Proxemics. This classical approach improves a bit the results of the "No social" navigation. Anyway, it performs a too significant avoidance when the robot is close to a pedestrian, which is considered unnecessary. The excessive execution time may be due to a not very precise programming that causes the robot to take too much time to evaluate the possible trajectories.
- One pedestrian (model 1). The application of the reward function learned with this model improves the results of the proxemics approach. It keeps a long distance to the pedestrians anticipating in time the avoiding maneuvers. The behavior of this model can be suitable for no-crowded scenarios, but the performance can degrade if there are several people surrounding the robot.

Table 8: Experimental results in a dynamic scenario

	T (s)	TD (m)	Mean PD (m)	Min PD (m)
No social	59.22 ± 0.06	20.38 ± 0.29	2.31 ± 0.15	0.20 ± 0.09
Proxemics	68.44 ± 0.02	20.99 ± 0.68	2.59 ± 0.10	0.38 ± 0.04
One Ped	69.45 ± 7.01	21.85 ± 1.89	4.46 ± 0.19	0.20 ± 0.17
All Ped	61.20 ± 0.0	20.50 ± 0.0	2.49 ± 0.16	0.60 ± 0.12
Densities	65.26 ± 10.08	20.75 ± 0.24	2.56 ± 0.17	0.47 ± 0.16
OnePed + Dens	60.20 ± 1.81	20.36 ± 0.35	2.46 ± 0.10	0.33 ± 0.02

- All pedestrian. According to the results, the generalization from one pedestrian to all, has a similar performance to the model for one pedestrian. This result can be different in experiments with more people. Anyway, we think that this generalization cannot retrieve the necessary key aspects for a good navigation in crowded environments.
- Densities (model 2). As we can see in the table, in this case, the distances to the pedestrians are lower than the case of model 1 and its generalization. However, aspects like the orientation or movement direction of the pedestrians are not taken into account, which can produce suboptimal avoiding behaviors.
- One pedestrian + Densities (model 1 and model 2). This approach can anticipate the avoidance maneuvers better than proxemics without performing excessive avoidance. Moreover, it keeps a shorter distance to the pedestrians than only model 1 without being uncomfortable for them. We consider that this behavior can be suitable for crowded environments.

9.2 Experiments with dynamic pedestrians

In this section we present the results of the experiments performed with dynamic pedestrians in two cases. First, we show the results in a controlled scenario, where we can repeat the experiment in similar (or nearly similar) conditions. We performed 4 runs of each navigation approach. The results are presented in Table 8. Secondly, we run the joint model (model 1 and model 2) approach in a uncontrolled scenario. This place is the main corridor of the Pablo de Olavide University, where a lot of students cross everyday (see fig. 24).

The set up of the controlled experiment is like follows: two pedestrians walk in an opposite direction than the robot, forcing it to avoid them. Then, another pedestrian cross in diagonal in front of the robot. Finally, two new pedestrians pass the robot on the left walking in the same direction as it.

Then, the results of table 8 are very similar to the results obtained in case of static pedestrians. Again, all the "social" approaches suggested improve the "no social" behavior. In this case, the model 1 (One Ped) seems to make some excessive avoidance maneuvers. However, the mixed approach of model 1 and model 2 seems to keep an acceptable distance to pedestrians in crowded environments and makes smoother avoiding maneuvers.

Regarding the experiment in the corridor of the University, it lasted about 12 minutes and the traveled distance was 221.18 meters. The average distance to the pedestrians was 4.2 meters. The behavior of the navigation approach was satisfactory according to the results obtained in

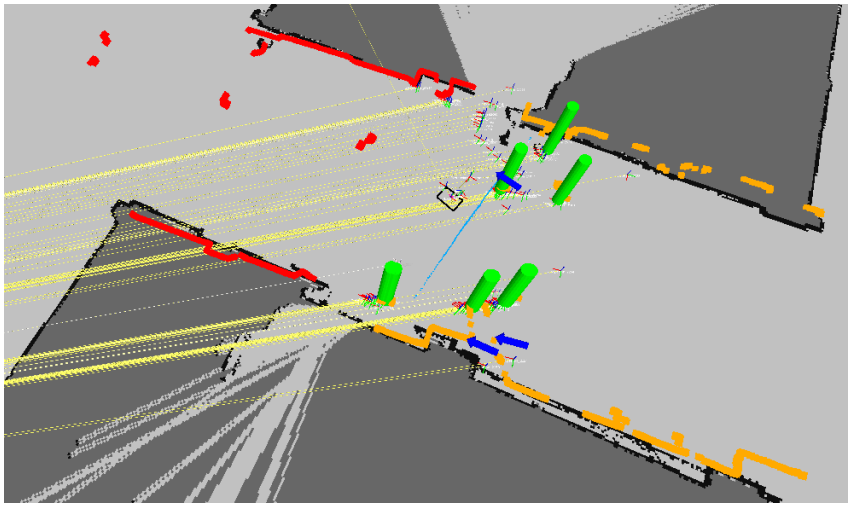

Figure 24: A snapshot of a particular situation in the main corridor at UPO. Green cylinders are people detected by leg detector system based on laser. Blue arrows are people detected by stereo cameras system.

Figure 25: Left: FROG guiding the reviewers and project members at Lisbon Zoo. Right: an overall view of the performed tour.

the controlled experiments. The actions taken to avoid the crossing pedestrians were smooth and sociable well-accepted by the pedestrian.

9.3 Experiments in Lisbon Zoo

In the second-year demo, the robot has guided visitors around in the Lisbon City Zoo, showing the animals, and telling about the species, their normal behavior and the natural environment (see Fig. 25). The demo involved a route of 750 meters, with different points of interest, and it lasted 45 minutes.

The whole demo week the robot was navigating autonomously for more than 3 kms. This demo, where the navigation functionalities were not using the social component, demonstrated its robust and accurate localization for navigation in such challenge outdoor scenario.

Figure 26: An overall view of the performed tours. Three different examples of the trajectories for the tour are shown in red, green and blue (the tour details depend on the time of the day, leading to some differences on the trajectories). Different photographies illustrate the different zones in the trajectory.

9.4 Experiments at the Royal Alcázar

During the third year, additional navigation experiments have been performed at the final demo site, the Royal Alcázar at Seville.

Figure 26 shows the typical mission performed. It involves an undocking maneuver, a tour including 7 Points of Interest, the way back to the charging station and docking again onto the charging station.

In these experiments, all the social navigation functionalities were activated. In particular, between June 16 and June 27, more than 16 guiding missions where performed, totaling more than 6 kilometers in autonomous mode.

Table 9 shows a summary of some of the missions, with information about time and distances. In the final evaluation deliverable, an evaluation of the overall navigation behavior of the robot from the point of view of the user acceptance will be considered.

10 Conclusions

This deliverable summarizes the robot navigation system developed in the framework of FROG project. An efficient and safe navigation system has been implemented according to the FROG

Table 9: Data from several of the tours performed at the Royal Alcázar. The experiments in the bottom rows involved the full tour and the returning to the charging station.

	T (min)	TD (m)	Mean PD (m)
19/06/14 15:02	24,70	223,30	3,42
20/06/14 17:52	21,92	258,23	2,65
26/06/14 11:53	26,60	257,91	2,09
26/06/14 17:35	27,73	275,46	2,25
21/06/14 12:40	38,03	460,22	2,23
22/06/14 16:52	37,39	453,40	2,16
23/06/14 11:55	37,08	468,25	2,30
23/06/14 16:46	35,93	453,77	2,24
25/06/14 18:08	36,51	458,40	2,36
26/06/14 11:08	36,02	434,97	2,20

system specifications, paying special attention to the social interaction aspects of this navigation. The document details the navigation stack implemented in the project and also the integration with sensors and FROG systems (such as the UvA person detection).

An inverse reinforcement learning approach to learn cost/reward functions from examples has been implemented for robot navigating among persons. The document described two different models and the methodology to extract the cost function from a public dataset. These costs function have been used to derive navigation policies for robot social navigation and were compared to the original human behavior and a policy derived from a cost function derived from Proxemics.

The computed policies were also compared and validated in real experiments involving a real robot, online people detection and the whole navigation stack developed in the project. The results included experiments with static and dynamic pedestrians. The experiments show how the computed policies allows a better navigation of the robot in scenarios involving persons, increasing the distance to the pedestrians. Also the results shown how the proposed approaches behaves better than classic Proxemics method which performs too significant avoidances when the robot is close to pedestrian, which is consider unnecessary.

The document also summarized the navigation performed by the robot during the second year review at Lisbon Zoo. The robot was able to navigate within a very complex scenario, with many people surrounding the system without problems. Although the social skills presented in this document were not evaluated at the Lisbon Zoo, the rest of the navigation system (localization, collision avoidance, smooth navigation, Point of Interest integration, etc) where successfully tested in more than 3 Km of fully autonomous robot navigation.

Finally, results obtained at the Royal Alcázar planned for June 2014 are included. This time, all the elements described in the document are employed. During the session, more than 6 kilometers in autonomous mode, guiding persons, were performed.

During the final demo week, the acceptance of the robot as a whole will be evaluated.

Bibliography

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 1–, New York, NY, USA, 2004. ACM.

[2] S. Alili, M. Warnier, M. Ali, and R. Alami. Planning and plan-execution for human-robot cooperative task achievement. In *19th International Conference on Automated Planning and Scheduling*, 2009.

[3] B.D. Argali, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstrations. *Robotics and Autonomous Systems*, 57:469–483, 2009.

[4] K. O. Arras, O. Martinez Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *Proc. International Conference on Robotics and Automation, ICRA*, 2008.

[5] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artif. Intell.*, 114(1-2):3–55, 1999.

[6] Alexander Carballo, Akihisa Ohya, and Shin'ichi Yuta. People detection using range and intensity data from multi-layered laser range finders. In *Proc. International Conference on Intelligent Robots and Systems, IROS*, pages 5849–5854, 2010.

[7] Aurélie Clodic, Hung Cao, Samir Alili, Vincent Montreuil, Rachid Alami, and Raja Chatila. SHARY: A Supervision System Adapted to Human-Robot Interaction. In Oussama Khatib, Vijay Kumar, and George J. Pappas, editors, *Experimental Robotics, The Eleventh International Symposium, ISER 2008, July 13-16, 2008, Athens, Greece*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 229–238. Springer, 2008.

[8] FROG Consortium. Deliverable D1.1: Functional Requirements, Interaction and Constraints. `http://www.frogrobot.eu/wp-content/uploads/2013/05/FROG-ROBOT-D1.11.pdf`.

[9] FROG Consortium. Deliverable D1.3: Customized Robot Plaftorm. `http://www.frogrobot.eu/wp-content/uploads/2013/04/D1.3_Final.pdf`.

[10] FROG Consortium. Deliverable D3.1: Final feature extraction component.

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[12] M. Enzweiler and D.M. Gavrila. Integrated pedestrian classification and orientation estimation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[13] Felix Faber, Maren Bennewitz, Attila Görög, Christoph Gonsior, Dominik Joho, Michael Schreiber, and Sven Behnke. The humanoid museum tour guide robotinho. In *in IEEE Int. Symp. on Robot and Human Interactive Communication*, 2009.

[14] D. Feil-Seifer and M. Mataric. People-aware navigation for goal-oriented behavior involving a human partner. In *Proceedings of the IEEE International Conference on Development and Learning (ICDL)*, 2011.

[15] G. Ferrer, A. Garrell, and A. Sanfeliu. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1688–1694, Nov 2013.

[16] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision – ACCV 2010*, volume 6492 of *Lecture Notes in Computer Science*, pages 25–38. Springer Berlin Heidelberg, 2011.

[17] B. Gerkey and K. Konolige. Planning and control in unstructured terrain. In *Workshop on Path Planning on Costmaps, Proceedings of the IEEE International Conference on Robotics and Automation*, 2008.

[18] Edward T. Hall. *The Hidden Dimension*. Anchor, October 1990.

[19] Peter Henry, Christian Vollmer, Brian Ferris, and Dieter Fox. Learning to navigate through crowded environments. In *ICRA'10*, pages 981–986, 2010.

[20] C. Keller, M. Enzweiler, M. Rohrbach, D.-F. Llorca, C. Schnörr, and D.M. Gavrila. The benefits of dense stereo for pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 12(4):1096–1106, 2011.

[21] R. Kirby, J. J. Forlizzi, and R. Simmons. Affective social robots. *Robotics and Autonomous Systems*, 58:322–332, 2010.

[22] Rachel Kirby, Reid G. Simmons, and Jodi Forlizzi. Companion: A constraint-optimizing method for person-acceptable navigation. In *RO-MAN*, pages 607–612. IEEE, 2009.

[23] Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. Inferring navigation policies for mobile robots from demonstrations. In *Proc. of the Autonomous Learning Workshop at the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[24] Thibault Kruse, Patrizia Basili, Stefan Glasauer, and Alexandra Kirsch. Legible robot navigation in the proximity of moving humans. In *ARSO*, pages 83–88. IEEE, 2012.

[25] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robot. Auton. Syst.*, 61(12):1726–1743, December 2013.

[26] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. In *Neural Information Processing Systems Conference*, 2011.

[27] M. Luber, L. Spinello, J. Silva, and K.O. Arras. Socially-aware robot navigation: A learning approach. In *IROS*, pages 797–803. IEEE, 2012.

[28] Bernard Michini, Mark Cutler, and Jonathan P. How. Scalable reward learning from demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013.

[29] Oscar Martinez Mozos, Ryo Kurazume, and Tsutomu Hasegawa. Multi-part people detection using 2D range data. *International Journal of Social Robotics*, 2(1):31–40, March 2010.

[30] Rafael Munoz-Salinas. ARUCO: a minimal library for augmented reality applications based on OpenCV. `http://www.uco.es/investiga/grupos/ava/node/26`.

[31] Liz Murphy and Peter Corke. STALKERBOT: Learning to Navigate Dynamic Human Environments by Following People. In *Proceedings of the Australasian Conference on Robotics and Automation (ACRA)*, Wellington, NZ, Dec 2012.

[32] E. Pacchierotti, H.I. Christensen, and P. Jensfelt. Evaluation of passing distance for social robots. In *IEEE Workshop on Robot and Human Interactive Communication (ROMAN)*, Hartfordshire, UK, September 2006.

[33] Anand Panangadan, Maja J. Mataric, and Gaurav S. Sukhatme. Tracking and Modeling of Human Activity Using Laser Rangefinders. *I. J. Social Robotics*, 2(1):95–107, 2010.

[34] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *International Conference on Computer Vision*, 2009.

[35] R. Ramon-Vigo, N. Perez-Higueras, F. Caballero, and L. Merino. Learning social cost functions for robot local navigation. In *Proc. International Conference on Intelligent Robots and Systems, IROS*, 2014. submitted.

[36] R. Ramon-Vigo, N. Perez-Higueras, F. Caballero, and L. Merino. Transferring human navigation behaviors into a robot local planner. In *RO-MAN*, 2014. submitted.

[37] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[38] Emrah Akin Sisbot, Luis Felipe Marin-Urias, Rachid Alami, and Thierry Siméon. A Human Aware Mobile Robot Motion Planner. *IEEE Transactions on Robotics*, 23(5):874–883, 2007.

[39] L. Spinello and K. O. Arras. People Detection in RGB-D Data. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[40] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, and C. Hahnel. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *The International Journal of Robotics Research*, 19:972–999, October 2000.

[41] Gian Diego Tipaldi and Kai O. Arras. Planning Problems for Social Robots. In *Proceedings fo the Twenty-First Internacional Conference on Automated Planning and Scheduling*, pages 339–342, 2011.

[42] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *IROS*, pages 797–803. IEEE, 2010.

[43] B. Ziebart, A. Maas, J. Bagnell, and A. Dey. Maximum entropy inverse reinforcement learning. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2008.