Analyzing the relevance of features for a social navigation task

Rafael Ramon-Vigo[1], Noe Perez-Higueras[1], Fernando Caballero[2] and Luis Merino[1]

[1] Pablo de Olavide University
Seville, Spain
[2] University of Seville
Seville, Spain

Abstract. Robot navigation in human environments is an active research area that poses serious challenges in both robot perception and actuation. Among them, social navigation and human-awareness have gained lot of attention in the last years due to its important role in human safety and robot acceptance. Several approaches have been proposed; learning by demonstrations stands as one of the most used approaches for estimating the insights of human social interactions. However, typically the features used to model the person-robot interaction are assumed to be given. It is very usual to consider general features like robot velocity, acceleration or distance to the persons, but there are not studies on the criteria used for such features selection.
In this paper, we employ a supervised learning approach to analyze the most important features that might take part into the human-robot interaction during a robot social navigation task. To this end, different subsets of features are employed with an AdaBoost classifier and its classification accuracy is compared with that of humans in a social navigation experimental setup. The analysis shows how it is very important not only to consider the robot-person relative poses and velocities, but also to recognize the particular social situation.

Keywords: Human-Robot Interaction, Supervised Learning, Social Robots

1 Introduction

Telepresence systems allow a human controller (the visitor) to interact remotely with people. Called by some "Skype on a stick", in such systems the visitor pilots a remotely located robot that results in a more physically presence than with standard teleconferencing. One of potential problems of telepresence systems is the cognitive overload that arises by having to take low (navigation commands) and high level decisions (interaction) at the same time. This may lead to mistakes at low level and to give less attention to the high level tasks [13]. To allow the visitor focusing in the interaction with other people, we aim to enhance

Fig. 1: The picture depicts a telepresence robot with partial autonomy in terms of navigation and body pose control, considering social feedback, in elderly centres. A typical situation is presented here.

the autonomy of the telepresence robot to perform low-level decisions for the controller regarding navigation and body pose in social settings (see Figure 1).

Actually, partial autonomy in terms of navigation is a feature requested by telepresence users [4]. Enhancing the autonomy of the telepresence robot in terms of navigation involves not only ensuring a safe and efficient navigation but also social interaction and social awareness when performing the robot tasks. For instance, approaching a person should be performed in a socially appropriate manner. In addition, when accompanying a person, some social rules must be maintained.

To this end, novel approaches are based on learning socially acceptable behaviors from real data collected under various social situations, avoiding manual explicit formulation of the behaviors. This is particularly interesting in the setup of telepresence robots, as there is a controller from which we can obtain information. In the last years, several contributions have been presented in this direction: supervised learning is used in [12] 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., [9] to determine socially-normative motion prototypes, which are then employed to infer social costs when planning paths. In [5], 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 [2]: an expert indicates the robot how it should navigate among humans. One way to implement it is through Inverse Reinforcement Learning (IRL) [1], in which a reward (or cost) function is recovered from the expert behavior, and then used to obtain a corresponding robot policy. In [7], a path planner based on inverse reinforcement learning is presented. As the planner is learned from exemplary trajectories involving interaction, it is also aware of typical social behaviors. Inverse reinforcement learning for social navigation is also considered in [10]. However, while in [7] the costs are used to path plans, in [10] the authors employ these techniques to learn local execution policies, thus providing direct control of the robot. This can be combined with other planning techniques at higher levels, while alleviating the complexity associated to learning.

Most of those works assume that the learned cost function depends on a set of predefined and hand-coded features of the state, like person distance and others. This paper presents a procedure to discriminate between features in order to choose those ones that could better describe the task of navigating among other people. Closest to this work, in [14] the authors present a software framework to select the features in the design choice in IRL by means of investigating the effect of selecting several feature sets in the evaluation of two different IRL approaches. However, while they compare different IRL methods for different set of features, in this work we leverage feedback provided by the telepresence users to employ a supervised learning method to determine the importance of the features.

The paper presents an analysis of the feature importance, as well as the data used for learning. A dataset of the robot navigating through other persons in different social configurations is employed here to learn how to classify different state/action pairs as socially normative behavior or not, using AdaBoost. This approach has been used by [11] to determine the features to take into account when the robot selects and follows a human leader to take advantage of their motion. Here we analyze a different task, and a different set of features. A similar approach has been used by [3] in order to determine the most important features on which a person 2D range-based classifier could rely on.

The structure of the paper is as follows: next section describes the experimental set up followed to retrieve both the human demonstrations of the robot navigating and the feedback signal from an user observeing that. The features considered are explained in Section 3 and the results of the evaluations are showed at Section 4. Finally, the conclusions and future developments are detailed in Section 5.

2 Experimental Setup

As a robot social behavior is very difficult to describe mathematically, we aim to learn adequate behaviors in social situations by observing real demonstrations of the task to be accomplished. In particular, in this paper we analyze a social navigation task consisting on approaching a person, called interaction target, with a telepresence robot. Thus, we perform a set of social navigation episodes in which we can create and control specific social situations for navigation. This data will be used to associate the robot and its environment context, such as the position of people in the room, to various types of direct and indirect feedback.

The experiments carried out involve the use of two rooms (see Figure 2): the interaction room and the visitor room. The interaction room is where all the pre-defined social interactions between the robot and the persons present at the scene take place. Due to control and repeatability issues, we propose to use confederates[3] that can conduct the social situation on an established plan.

[3] A confederate in this context is any person who takes part in the experiment but is not a subject. Even if the subjects themselves are aware of this fact.

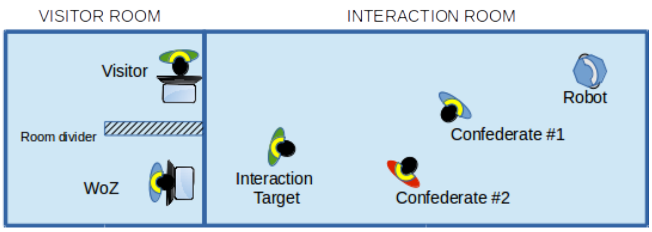

Fig. 2: Overview of the visitor and interaction rooms.

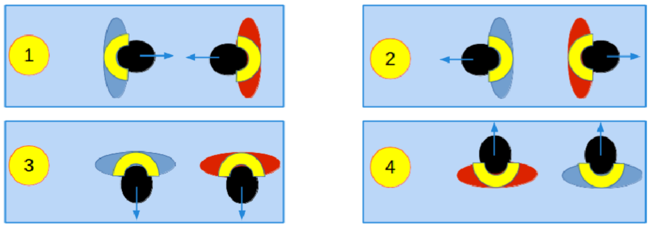

Fig. 3: Different social scenarios regarding to the pose adopted by the confederates.

Among the confederates, we have an interaction target, which is the subject that the visitor intends to interact with using the telepresence robot. Finally, we also have the telepresence robot itself.

In the visitor room we have a subject (the visitor) that is observing, through the telepresence robot, what is going on in the interaction room. The visitor was instructed to provide an instantaneous feedback signal based on the behavior played by the robot in terms of what could be a normative socially behavior or not. This feedback signal will be detailed in the next sections.

Also present in the visitor room is a Wizard of Oz (WoZ) [8]. The WoZ is responsible of the behavior and low level control of the robot, but the visitor is misinformed that the robot is autonomous. In fact, the WoZ is physically separated from the visitor by a room divider and all the time conducts the experiment, not only driving the robot, but also informing the visitors about the current attempt of the robot and carrying the execution timing.

2.1 Experiments

The task of the robot was to reach the interaction target while dealing with the social scenarios (depicted in Figure 3) performed by the other two confederates, and then return to its starting position. Although the scenarios performed include static and dynamic configurations, this work deals only with the static ones as a preliminar study. In the static scenarios, the social obstacles and the interaction target remain standing in the scene at the same place. The robot has to navigate towards the interaction target (and then moving back), performing

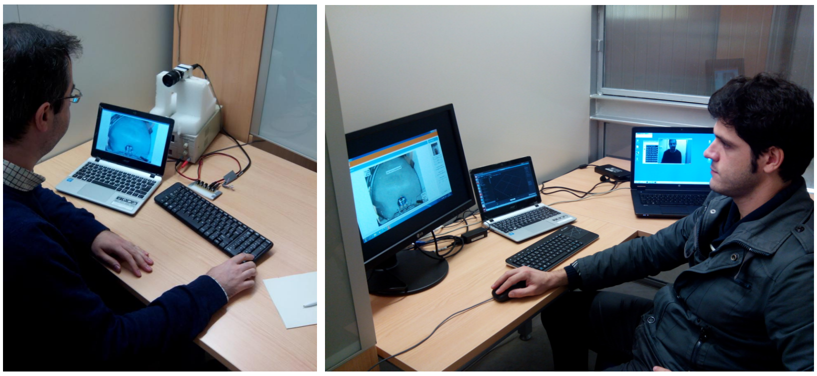

Fig. 4: Left: the visitor is depicted. He observes the scene through the telepresence robot interface in a laptop; he has a keyboard to enter his feedback by pressing a key. Right: the WoZ performing the navigating teleoperating the robot through the telepresence interface. There was no visual contact between them.

a trajectory that avoids in some way the social obstacles. As mentioned before, the different configurations that the confederates could adopt are depicted in Figure 3.

In the adjacent room, the WoZ was controlling the robot while the visitor was observing and evaluating its actions, thinking that it was autonomous. Figure 4 describes the realization of the visitor room.

2.2 Data gathering

The effective area of the interaction room is $6x4$ meters, and it is covered by a motion capture system, in particular the OptiTrack[4] system. This tool allowed us to collect detailed information about the positions and orientations of all the elements present in the interaction room, i.e. the robot, the confederates and the interaction target. Sensing the robot environment during the experiments is necessary in order to derive the states and features that can be later used for learning.

Another important source of data is the feedback signal that the visitor provides. During the experiments, the visitor gives direct feedback to the robot's instantaneous behavior using timestamped button presses (as showed in Figure 4 (left)) whenever he/her feels that the robot is behaving wrongly. The visitor was instructed to press a button when he observed such an action and keep pressing it until he thought that the robot had returned to normal behavior. During the trajectories, the WoZ would deliberately execute socially unacceptable actions sometimes.

Before the integration and collection of the data from the motion capture system Optitrack and the labeling information provided by the visitor subject

[4] https://www.naturalpoint.com/OptiTrack/

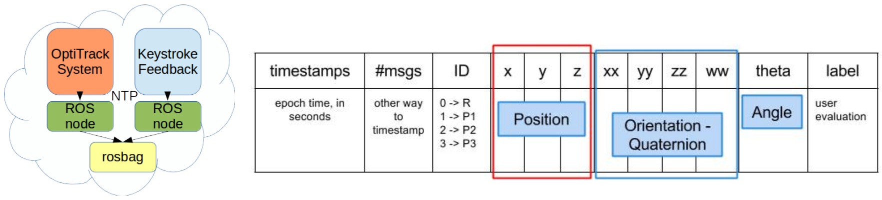

Fig. 5: Left: Integrating the state capture data and labeling information into rosbag log utility. The NTP ensures the time synchronization. Right: Formatting of the extracted data for each trajectory into the txt files from bags. For each ID (robot, R; interaction target, P1; confederates, P2 and P3) the position and orientation are stored.

through the keystroke system (see Figure 5 (left)), a data post-processing step was performed by applying a smoothing interpolation. These allowed us to deal with small drop outs and tracking errors while performing the experiments.

Based on this data, several features can be derived and employed to represent states used in the learning process (see Section 3). For all the features, additional Gaussian filters were used in order to suppress any residual noise. This was required because we are also interested in some features that are obtained by differentiating previous ones.

It is important to notice that the observed feedback signal in the experiments is very sparse. In addition we have very limited knowledge about the meaning of the feedback and its duration. Thus, the dataset was pre-processed in order to increase the duration of the feedback signal, making it less sparse while keeping its interpretation meaningful. This is done by extending the feedback duration forward and backwards. This filter is reasonable under the assumption that the evaluator had some delay time in their reaction (backwards justification) and that the robot actions are smooth, i.e., the robot does not escape from a situation instantaneously (forward justification). The chosen extended duration was a second, centered at the exact time in which the visitor stamped his label. This value was settled empirically after analyzing the total time duration of each single trajectory and the usual length of 'train of keystrokes' observed when a person evaluate a bad behavior.

Despite the feedback signal extension performed, the number of bad examples and good examples are clearly unbalanced on behalf of good ones. However, the Adaboost algorithm implemented [3] deals with such kind of unbiassed datasets.

3 Features considered

The objective is finally to transfer the good behaviors of the WoZ into the robot navigation stack, so that the robot is able to execute the task by itself in a socially adequate manner. The first important issue is to determine which information about the state is relevant for such task.

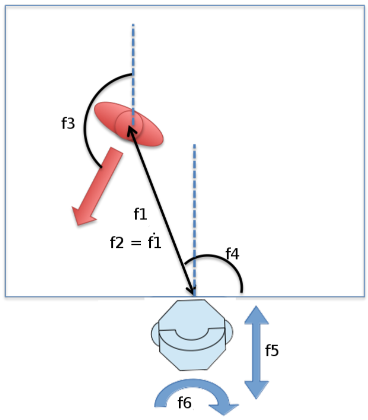

Fig. 6: Features considered in this work. All of them are average values (moving average with a 1-second window) and with respect to the closest person.

We will use the feedback given by the visitor. The idea is to use a supervised machine learning approach to reproduce the labels given by the visitor using as inputs different sets of features from the state. In general, features do not contribute equally to predict the target response; in many situations the majority of the features are in fact irrelevant.

There are some machine learning resources, like individual decision trees, that intrinsically perform feature selection by selecting appropriate split points. This information can be used to measure the importance of each feature; the basic idea is: the more often a feature is used in the split points the more important that feature is. In this work, we use AdaBoost [6], whose core principle is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The input to the algorithm is a set of labeled training data $(e_n, l_n), n = 1, ..., N$, where each e_n is an example and $l_n \in \{+1, -1\}$ indicates whether e_n is positive or negative respectively.

It is interesting to choose a weak learner that is fast to be trained, as usually a large number of them is required. Examples of weak learners are decision trees, multi-layer perceptron and radial basis function. In this work, decision stumps are used, which is a one-level decision tree, making predictions based on a single threshold over a single feature [15].

From the features point of view, one requirement was to keep the model as simple as possible: the level of complexity must be such that the task description is not trivial, which would make learning redundant, or too complex, which would make learning impossible. So thus, the features considered were (see Figure 6):

- f_1: Average Distance To Closest Person
- f_2: Average Relative Velocity To Closest Person
- f_3: Average Relative Orientation To Closest Person

– f_4: Average Angle To Closest Person
– f_5: Average robot's linear velocity
– f_6: Average robot's angular velocity

All the features are computed from the recorded data using a moving average with a 1-second window. Although these features are quite basic, they are sufficient for the initial static scenario described in 2.1. All of the features used here are also employed in the literature, as in [14] and [7], but in this case we do not use a discrete approximation. It should be pointed out that, from the point of view of the approach used in this work, it is plausible to add whatever feature we can measure while the robot is performing the navigation task.

4 Results

A total of 4 sessions of the task described above were carried out. Each session involved 12 different executions, by using different trajectories, locations of the interaction target and configuration of the confederates. That results in 48 trajectories. For each session, a unique visitor evaluated the robot behavior into bad examples (by keystroke) or good examples (by default, no keystroke needed).

Each experiment was tagged with information related with the type of trajectory, target layout, etc. This classification allows us to analyze the feature importance over different set of trajectories, since we can take into account the whole set of trajectories or some subset of them according to the three last tags indicated before. We can also choose between different social navigation tasks, being more specific with respect to what kind of navigation task we attempt to, i.e. navigating towards a target while two persons are interacting (confederates' pose 1 of Fig. 3) or not.

The tags allow distinguishing between approaching the interaction target and getting away from him, called *go* and *return* trajectories respectively. This division is made because the type and nature of the interaction might be different when the robot approaches a person than when the robot leaves him away.

In addition to this, different subsets of features are considered at the learning process of the AdaBoost from those suggested at section 3. Thus, 6 different evaluations regarding to the set of features employed for the AdaBoost learning process will be considered:

– $\mathcal{F}_1 = (f_1, f_2, f_3, f_4)$
– $\mathcal{F}_2 = (f_1, f_2, f_3)$
– $\mathcal{F}_3 = (f_1, f_2, f_4)$
– $\mathcal{F}_{1*} = (f_1, f_2, f_3, f_4, f_5, f_6)$
– $\mathcal{F}_{2*} = (f_1, f_2, f_3, f_5, f_6)$
– $\mathcal{F}_{3*} = (f_1, f_2, f_4, f_5, f_6)$

The evaluations $\mathcal{F}_{\_*}$ consider the last two features described at Section 3, i.e. f_5 and f_6. Both of them could be described as action-based features, because they enclose the robot actions. Thus, the idea is that these features could enclose robot behaviors in terms of bad (or good) maneuvers related to natural or smooth movements that may not disturb the people surrounding the robot but that they can be perceived as strange or unusual. On the other hand, sets $\mathcal{F}_2$ and $\mathcal{F}_3$ (and their $*$ extensions) are intended to study the effects of suppressing some features in the classification task. We focused in f_3 and f_4 in order to determine which of them would be more descriptive for the navigation tasks.

Next paragraphs present the evaluation of the different features. In the following, only the experiments with configuration 1 for the confederates (see Fig. 3) were considered. The procedure followed consists on cross validation: 4 random sets of training samples have been selected with other 4 sets of testing samples. The ratio of samples was 80% for each set of training and 20% for each set of testing. In the following, the details of the specific trajectories and features employed will be provided.

4.1 Evaluation 1

Table 1 takes into account the whole set of trajectories gathered at the experiments. This case represents the most global description of the task, since no distinction was done between the type of the trajectory, targets' layout and confederates' poses. We intended to classify all the social configurations with a single classifier. We do that with and without considering the action-based features.

Table 1: All trajectories and configurations are considered from the executed examples. Evaluating the overall classification with respect all f_1, f_2, f_3, f_4 features, with and without action-based features.

	All trajectories; All poses			
	$\mathcal{F}_1$		$\mathcal{F}_{1*}$	
	Detected Label		Detected Label	
True Label	No-Social	Social	No-Social	Social
No-Social	46%	**81%**	59%	**85%**
Social	**54%**	19%	**41%**	15%

The results shown in Table 1 indicate that it is not possible to correctly classify all the trajectories. Next, we considered ways to alleviate the complexity of the model being learned, and proposed some easier models by reducing the variance across the examples. During the experiments, the WoZ always introduced the visitor the individual trajectories, differentiating between *go* and *return* trajectories (explained above). Thus, the next natural step was making this distinction. Table 2 evaluates the construction of two different classifiers for both *go* and *return* types of trajectories, taking into account the sets of features $\mathcal{F}_1$ and $\mathcal{F}_{1*}$.

It can be seen that the introduction of the action-based features improves the classification task in all the cases (see Tables 1 and 2). Other important conclusion is the fact that training a single classifier with all the examples performs

Table 2: *go* and *return* trajectories with confederates' pose 1: Evaluating the effects of adding action-based features.

	Type: *go*				Type: *return*			
	Set $\mathcal{F}_1$		Set $\mathcal{F}_{1*}$		Set $\mathcal{F}_1$		Set $\mathcal{F}_{1*}$	
	Detected Label		Detected Label		Detected Label		Detected Label	
True Label	No-Social	Social	No-Social	Social	No-Social	Social	No-Social	Social
No-Social	**93%**	10%	**93%**	3%	**67%**	13%	**94%**	7%
Social	7%	**90%**	7%	**97%**	33%	**87%**	6%	**93%**

poorly with respect to training different classifiers for some specific configurations of the task. Performance differences observed between types *go* and *return* may not be as obvious at a first glance. This could be produced by the lack of some unknown features, like context information, or by a significant difference between both trajectories due to the proximity of the goal to a target in the case of the *go* task.

4.2 Evaluation 2

This evaluation focuses on the analysis of the impact of some features. The main objective is to study how the introduction (or removal) of a specific feature into the classification could affect the performance obtained in $\mathcal{F}_1$ and $\mathcal{F}_{1*}$. Tables 3 and 4 presents the confusion matrix of sets $\{\mathcal{F}_2, \mathcal{F}_3\}$ and $\{\mathcal{F}_{2*}, \mathcal{F}_{3*}\}$, respectively.

Table 3: *go* and *return* trajectories with confederates' pose 1: Evaluating the effects of extracting a feature. No action-based features considered.

	Type: *go*				Type: *return*			
	Set $\mathcal{F}_2$		Set $\mathcal{F}_3$		Set $\mathcal{F}_2$		Set $\mathcal{F}_3$	
	Detected Label		Detected Label		Detected Label		Detected Label	
True Label	No-Social	Social	No-Social	Social	No-Social	Social	No-Social	Social
No-Social	**84%**	6%	**76%**	21%	**90%**	38%	**67%**	20%
Social	16%	**94%**	24%	**79%**	10%	**62%**	33%	**80%**

Table 4: *go* and *return* trajectories with confederates' pose 1: Evaluating the effects of extracting a feature. Action-based features considered here.

	Type: *go*				Type: *return*			
	Set $\mathcal{F}_{2*}$		Set $\mathcal{F}_{3*}$		Set $\mathcal{F}_{2*}$		Set $\mathcal{F}_{3*}$	
	Detected Label		Detected Label		Detected Label		Detected Label	
True Label	No-Social	Social	No-Social	Social	No-Social	Social	No-Social	Social
No-Social	**99%**	3%	**93%**	11%	**88%**	12%	**87%**	10%
Social	1%	**97%**	7%	**89%**	12%	**88%**	13%	**90%**

If we compare the Tables 1, 2, 3 and 4, it can be observed that the best performance is obtained by using the set $\mathcal{F}_{2*}$, which performs only slightly better than using the set $\mathcal{F}_{1*}$. Comparing the Tables 3 and 4, we also conclude that the feature f_3 (average relative orientation to closest person) is more relevant than the feature f_4 (average angle to closest person) for the description of the task in both situations (*go* and *return* trajectories).

5 Conclusions and Future Work

This paper presented an analysis of the features used to model the interaction between persons and robots when the latter are performing a particular social navigation task. An experimental setup have been conceived and implemented in order to gather enough data for the validation of the technique. The experiment also included feedback from the user so that supervised learning approaches can be applied to learn such features.

Two main conclusions arise from the data evaluation: First, the type of trajectory followed by the robot plays an important role in the learning phase. A classifier trained with *go* and *return* trajectories performs worst than two different classifiers, one per type of trajectory. We guess this is produced by the different perspectives when the robot needs to approach a goal close to a person (*go*) with respect to a less constrained goal in terms of gaze and body pose expectations (*return*), which basically are considered different navigation tasks. Furthermore, the social situation of other persons in the environment (represented in the experiments by the confederates) is also important, even in such a simple task. Finally, the introduction of features related with the robot action clearly improve the classification. The evaluations showed that considering the robot actions (in terms of velocities and accelerations) help to identify if the robot is behaving normative or not.

Future work considers extending this study to more complex escenarios, including dynamic confederates and targets. In addition, the set of possible features will be increased, considering also some high level information as groups of persons or social situation.

Acknowledgements. This work is partially funded by the European Commission, FP7, under grant agreement no. 611153 (TERESA).

References

1. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the twenty-first international conference on Machine learning. pp. 1–. ICML '04, ACM, New York, NY, USA (2004), `http://doi.acm.org/10.1145/1015330.1015430`
2. Argali, B., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstrations. Robotics and Autonomous Systems 57, 469–483 (2009)

3. Arras, K.O., Mozos, O.M., Burgard, W.: Using boosted features for the detection of people in 2d range data. In: Proc. International Conference on Robotics and Automation, ICRA (2008)
4. Desai, M., Tsui, K., Yanco, H., Uhlik, C.: Essential features of telepresence robots. In: Technologies for Practical Robot Applications (TePRA), 2011 IEEE Conference on. pp. 15–20 (April 2011)
5. Ferrer, G., Garrell, A., Sanfeliu, A.: 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. pp. 1688–1694 (Nov 2013)
6. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119 – 139 (1997), http://www.sciencedirect.com/science/article/pii/S002200009791504X
7. Henry, P., Vollmer, C., Ferris, B., Fox, D.: Learning to navigate through crowded environments. In: ICRA'10. pp. 981–986 (2010)
8. Knox, W., Spaulding, S., Breazeal, C.: Learning social interaction from the wizard: A proposal. In: Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
9. Luber, M., Spinello, L., Silva, J., Arras, K.: Socially-aware robot navigation: A learning approach. In: IROS. pp. 797–803. IEEE (2012)
10. Perez-Higueras, N., Ramon-Vigo, R., Caballero, F., Merino, L.: Robot local navigation with learned social cost functions. In: Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on. vol. 02, pp. 618–625 (Sept 2014)
11. Stein, P., Spalanzani, A., Santos, V., Laugier, C.: On leader following and classification. In: Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. pp. 3135–3140. IEEE (2014)
12. Trautman, P., Krause, A.: Unfreezing the robot: Navigation in dense, interacting crowds. In: IROS. pp. 797–803. IEEE (2010)
13. Tsui, K., Desai, M., Yanco, H., Uhlik, C.: Exploring use cases for telepresence robots. In: Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on. pp. 11–18 (March 2011)
14. Vasquez, D., Okal, B., Arras, K.O.: Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'14). Chicago, USA (2014)
15. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004), http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb