

Formalizing Normative Robot Behavior

Billy Okal¹(✉) and Kai O. Arras^{1,2}

¹ Department of Computer Science, University of Freiburg, Freiburg, Germany

okal@cs.uni-freiburg.de

² Bosch Corporate Research, Robert-Bosch GmbH, Gerlingen, Germany

Abstract. We address the task of modeling, generating and evaluating normative behavior for interactive robots. Normative behavior is essential for coherent deployment of these robots in human populated spaces. We develop a first unifying, intuitive and general formalism of the task that subsumes most previous approaches which have focused mainly on specific tasks. We present concrete and practical definitions of norms and show how to generate and evaluate behavior that adheres to such norms. We then demonstrate the formalism on a socially normative navigation task for service robots. Further, we discuss the key challenges in realizing such behaviors, and in particular, the role of perception and uncertainty.

1 Introduction

As more service robots are deployed for various functions in public spaces such as hotel lobbies, airports, hospitals, care homes, etc.; the need to demonstrate additional social, cultural capabilities beyond primary functionality arises. This is because having such robots in human populated spaces change the underlying dynamics of social and cultural interactions [13, 18]. Additionally, most human interactions often influenced by deeper social and cultural standards or norms which vary across environments, making it hard to explicitly model them. Robots operating in these spaces nonetheless need to exhibit behavior that takes into account such social and cultural aspects. We call such behavior a *normative behavior* although other terminology such as compliance in the case of “socially compliant navigation” or human-awareness are also commonly used. In this paper, *normative* is taken to mean according to a set of norms, which can be either formal such as traffic rules, or social-cultural such as politeness when navigating. The behavior resulting from adherence to the norms may not be efficient in terms of classical task metrics like path length in planning, but may sometimes lead to efficiencies in other regimes. For example a norm requiring a robot to execute a slipstream maneuver (tailing a person heading in similar direction in a dense crowd) may generate longer paths; but in the process also improve reliability in reaching goal by avoiding situations leading to getting stuck. These norms are often characterized by the entities the robot is interacting with and the structure of the environment. For example Kruse et al. [10, 17] provide a summary of navigation behaviors with respect to human-awareness norm generated by considering various geometric relations in the robot’s environment.

The key task therefore, is figuring out how to equip robots with decision making capabilities that result in normative behavior. By taking care of norms in the decision making stage, we can build anticipatory behavior as opposed to reactive one. Anticipatory behavior is better suited for coordination and can avoid damages [1]. Incorporating norms into decision making involves the following key steps: (i) a formal and practical understanding of norms, (ii) techniques to generate behavior that adheres to the norms and (iii) effective evaluation methods for assessing the result. There has been a growing interest in various instantiations of this task, especially with respect to social navigation norms for service robots. For example [7, 16] all seek to develop normative behavior for navigating in crowded scenes, while [4, 9, 21, 22] focus on normative pairwise interactions such as passing one another on either side. Most of these approaches focus on different aspects of the task, defining their own metrics and understanding of norms, thereby making it difficult to compare, evaluate and select methods to deploy in practice. It is with this realization that we develop a unified formalism for addressing this task in this paper. It is our hope that this new formalism will help organize the efforts to tackle this task under a common setup.

While developing the new formalism, we highlight some key challenges associated with realizing normative behavior. As a concrete example, consider a robot providing service in a hotel lobby; the robot’s normative navigation decision making capabilities are highly dependent on the performance *perception* components, that is, if people cannot be reliably detected, planning around them normatively is rendered impossible. In general, the more properties of the environment that can be reliably perceived, the more norms can be taken into account in decision making. Additionally, *predictions* of future states and actions of other agents in the environment is also crucial. Finally, *uncertainty* arising from action execution using noisy controllers also needs to be considered. Altogether, these challenges form a tightly coupled perception-action-control loop that requires clear interfaces between components when developing a wholesome solution. Consequently, the proposed formalism should be able to define such interfaces clearly on a common framework. Thus, the main contribution in this paper is a first unifying formalism for modeling norms and normative behavior for interactive robots. The presented formalism also admits natural ways of generating and evaluating resulting normative behavior.

2 Modeling Normative Robot Behavior

In order to have a concise yet flexible formalism for normative behavior, we discuss three key components that are needed to realize such behavior. These components will then enable us to make formal definitions of norms and normative behavior, and lead to what we think of as natural ways of generating such behavior. We also emphasize that in all of these components, uncertainty plays a key role in the success of any endeavor, hence the formalism needs to admit possibility to reason about uncertainly at all levels.

2.1 Environment

The environment \mathcal{C} that the robot is operating in is made up of the space and all the entities $\mathcal{E} = \{e_i\}$ present in it, which together define the structure of the environment. These entities could be interactive and even adversarial like humans or simply artifacts like general obstacles in the scene. Each entity’s anchoring in the environment is summarized using a pose vector \mathbf{x}_i . Additionally, each entity possesses a set of *attributes* $\mathcal{A} = \{a_i\}$, any subset of which is represented using a vector $\mathbf{a}_i = (a_j, \dots, a_{j+k})^T \subseteq \mathcal{A}$. For example, a person i can be represented using a pose $\mathbf{x}_i = (x, y, \dot{x}, \dot{y})^T$ and may have simple attributes such as age, gender, carrying-luggage, etc. given in a vector $\mathbf{a}_i = (25.0, M, T)^T$. Further, there are pairwise relations $f : \mathcal{E} \times \mathcal{E} \mapsto [0, 1]$ between some of the entities. Such pairing can be as a result of attributes. For example two people may belong to a group like a couple, to which their gender attributes as well as geometric reasoning may generate a pairing probability of say $f_{\text{group}}(e_1, e_2) = 0.7$, which is interpreted as the strength of the relation. We eschew the details how to define and detect such relations to the designer, and only emphasize that the formalism presented is general enough to admit many choices. Figure 1 shows a example of such an environment in the case of a navigation task. We argue that this minimalistic representation is sufficient to capture all aspects needed for decision making that culminates in normative behavior of any complexity.

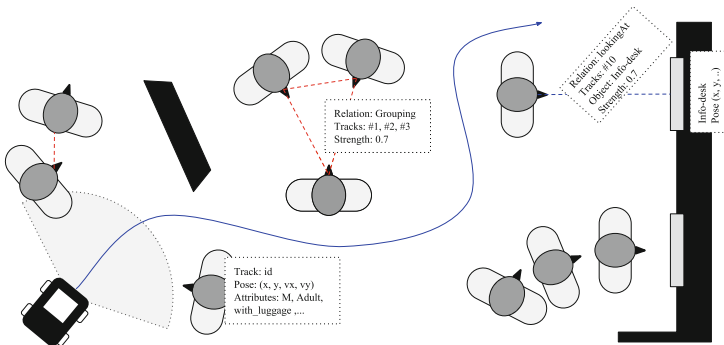


Fig. 1. A normative navigation example setup. Entities shown include people, desk, queues and general obstacles (in black). Potential perceptual attributes are shown alongside people. Pairwise relations between people and between person (e. g. grouping) and other objects (e. g. looking at a screen) are also illustrated with dotted red and blue lines. A sample socially normative navigation path is shown in blue curve. (Color figure online)

2.2 Perception

A crucial component for realizing any normative behavior in the environment we just described above, is the ability of observe the different aspects of the

said environment with reasonable accuracy. In fact, we argue that the difference between normative behavior and non-normative counterpart lies solely in which of the perceived aspects of the environment are taken into account in decision making. In this work, we require that any robot intending to exhibit normative behaviors be equipped perception modules for observing entities in the environment, a subset of their attributes and relations. The richness of this perceived subset of attributes directly affects how complex a normative behavior may be realized. For example, we cannot develop a behavior adhering to norms relating to gender if we cannot reliably perceive gender attribute.

Concretely, for every entity in the scene, a perception module produces tuples of poses and associated uncertainty estimates $(\mathbf{x}_i, \delta_i^p)$. For example, this could be a people detector module for persons or a localization module for providing obstacle poses. Similarly, another high level perception module would provide attributes values and associated uncertainty $(\mathbf{a}_i, \delta_i^a)$. An example of such attribute detectors in practice is given in [11] for age groups, gender and clothing related attributes. Finally, the pairwise relations can be perceived using relational reasoning modules so as to provide relation probabilities for every pair of entities. Altogether, the perception modules are seen as black boxes \mathcal{P} which produce signals for each entity, attributes and relations in the environment. The exact form of these uncertainty estimates δ_i^p, δ_i^a depend on the sensor instrumentation used and algorithms for the various perception tasks involved.

2.3 Execution

Normative behavior is usually targeted at robots which interact with humans or other robots. As such, these robots modify the environment they operate in and potentially alter future percepts, thus we need to also formalize the nature of their effects through their actions \mathcal{U} . Concretely, for most decisions made by the robot, a series of actions are performed, but executions are often imperfect hence the need to explicitly model uncertainty. We argue that the execution of a series of actions can be effectively assessed by examining the trajectory $\xi = \{\mathbf{x}^t\}$ resulting for such execution. Such a trajectory can have a ‘band’ in the space of poses, capturing the uncertainty in the execution. This simplistic formulation is sufficient for purposes of normative behavior realization.

Finally, using the three components above, we can now formally define norms and normative behavior before we set about on finding techniques to generate and evaluate them.

Definition 1 (Norm). *A norm N^μ in the context of robot behavior is a property of the robot’s environment \mathcal{C} , percepts \mathcal{P} and actions \mathcal{U} with an associated set of M tests $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_M\}$ for assessing adherence.*

Definition 2 (Norm Test). *Given a norm N^μ over some environment and a trajectory $\xi_j \in \Xi$, a norm test $\mu_i : \mathcal{C} \times \mathcal{P} \times \Xi \mapsto \{0, 1\}$ is a function that evaluates adherence of the trajectory to the norm.*

The adherence to a norm is collectively assessed by all the norm’s tests, meaning all norm tests must pass. This formulation of norms ensures that practical assessment is algorithmically possible.

Definition 3 (Normative Robot Behavior). *Given a collection of K norms $N_1^{\mu_1}, \dots, N_k^{\mu_k}$; a behavior exhibited by a trajectory ξ is said to normative if and only if the application of all the norm tests to the trajectory pass. i. e. Given $C \in \mathcal{C}, P \in \mathcal{P}$, the behavior is normative if and only if $\bigwedge_{k=1}^K \bigwedge_{m=1}^M \mu_{m,k}(C, P, \xi) = 1$.*

3 Generating Normative Behavior

In this section, we provide the technical means for realizing the normative behavior defined in Sect. 2. This entails specification of decision making approaches which are incorporated in task and motion planning modules of such robots. The most common approach is formulate a cost function which encodes the desired normative behavior, and then use such function to guide solution search in planning algorithms. However, it is often very difficult to manually design such cost functions, especially because of inherent ambiguity in the specification of these behaviors due to dependence on social and cultural aspects of involved parties. A common simplification used in practice is to model the cost functions are mixtures of basis features of the environment and the entities in it. The formalism presented here is particularly well suited for such endeavor as these features can simply be based on the poses, attributes and relations of entities. This also helps lighten the burden of coming up with features. However, the mixing ratios of such features still need to be figured out.

A promising technique for learning the mixing of features is learning from demonstration (LFD). In particular, inverse reinforcement learning (IRL) formally introduced in [14] has been used successfully in many applications such as crowd navigation [7, 9, 16]. The IRL approach assumes the robot’s decision making is carried out using a Markov decision process (MDP) with an unknown cost function (equivalently reward function), usually assumed to encode the behavior. In practice IRL involves demonstrating the desired behavior, usually by manually driving the robot say using a joystick, and then using typically iterative algorithms to recover cost function that “explains” the demonstration. This cost function can then be used to either generate costmaps over planning domains or is integrated directly into a planning algorithms objective function. The main challenge in using IRL approaches is lack of computationally and data efficient algorithms used to recover the cost function and practical representations suited for most real world tasks.

Regardless of the procedure used for realizing the normative behaviors, it is imperative that uncertainty in both perception and execution be taken into account in decision making. For example, when cost functions, this could mean inflating cost regions around potential configurations by a factor proportional to the uncertainty in say people detection. Finally, for successful normative behavior generation, it is imperative to predict and take into account the future actions

of other decision making agents in the same environment. These could entail predicting future positions of people and then generating costmaps that already take the prediction into account, resulting in anticipatory behavior. For certain tasks such as navigation, this often reduces stop-and-go motions caused by too reactive a planner which acts myopically.

4 Evaluating Normative Behavior

Normative robot behavior can be evaluated in two different ways; firstly by checking the adherence of behavior to norms using the associated norm tests, and secondly using task specific metrics. The evaluation based on norm tests is fully dependent on the specification of the norms. Evaluation using task specific metrics could lead to discoveries of potential trade-offs between say task efficiency and normativeness. These trade-offs, if any, could be helpful to service robot designers who are often confronted with choosing either functionality or normativeness in practical settings.

A number of task metrics have been proposed in the past including these human robot interaction metrics [2, 20]. These include path lengths, time to goal, idle operating time, human comfort as measured by qualitative questionnaires. We defer the exact choice of such task metrics to the designer as it is difficult to list a complete set of metrics for all possible tasks.

5 Case Study: Socially Normative Crowd Navigation

In order to demonstrate how to use the formalism practically, we show how to define simple social norms for a mobile service robot navigating in a crowded space and generate the required socially normative navigation behaviors. The environment is a place $\mathcal{C} \triangleq \mathbb{R}^2$, entities are people, shops, walls, etc. Some of the people in the scene are engaged in groups, others are engaged with various activities such as queuing or looking at information boards. Our service robot is required to efficiently navigate in this scene while respecting the various social norms, and in effect treat people as more than just dynamic obstacles.

We define the following basic norms for our case study.

Personal spaces, $\mathcal{N}_P^{\mu_P}$: Always minimize intrusions into personal spaces around people. These personal spaces are derived from Proxemics theory [6] with these radii (Personal: 0.45 m to 1.2 m, Social: 1.2 m to 3.6 m). We define $\mu_P = \{\mu_P, \mu_S\}$, where test $\mu_P = \zeta_P \leq \alpha_P$ and $\zeta_P = \sum_t \sum_i \mathbb{1}(\|\mathbf{x}_r^t - \mathbf{x}_p^i\|_2 \leq 1.2)$ for some appropriate threshold α_P which can be experimentally identified. $\mathbf{x}_r^t, \mathbf{x}_p^i$ denote robot pose and person i respectively at time t , while $\mathbb{1}(\cdot)$ is the indicator function. Other test, μ_S are defined analogously.

Interaction spaces, $\mathcal{N}_I^{\mu_I}$: While interacting with people, minimize disturbance on the relations between them, e. g. do not cross through a group. We define $\mu_I = \{\mu_r\}$ with $\mu_r = \zeta_r \leq \alpha_R$ and $\zeta_r = \sum_t \sum_k \text{dist}(\mathbf{x}_r^t, \mathbf{x}_s^k)$, where $\text{dist}(\cdot)$ is shortest distance to an interaction area (which can be represented as polygon),

α_R is a threshold and \mathbf{x}_s^k is the pose of k -th interaction area. For a pairwise relation, this is a line.

The requirements for perception include; reliable detection of people, detection of pairwise relations, in particular grouping affiliations and engagements such as looking at something in the scene. Because there are not many reliable and practical perception modules that can deliver the required attributes for our norms, we first perform experiments using an open source pedestrian simulator¹ described in [15]. We then later deploy the robot in the wild at an airport with the learned socially normative behaviors.

We use the LFD approach presented in [16] to learn behaviors for this task from expert demonstrations. We represent the target cost function as linear combination of features, which we derive from the attributes of entities i. e. distance to persons, distance to pairwise relation lines and relative goal heading. Learning of the cost function is done using an extension of the Bayesian inverse reinforcement learning (BIRL) algorithm that works well in practice as described in [16]. We use 10 trajectories demonstrated by driving the robot using a joystick for learning the cost function. We use the found cost function to generate costmaps which are then used by navigation planners.

We evaluate the learned behaviors by having the robot plan and navigate between a total of 25 different start and goal pairs, and perform the above specified norm tests on the resulting paths plus run additional task specific metrics. These task metrics are: path length, time to goal and cumulative heading changes (CHC); in all of which smaller quantities are preferred. We use *classical* navigation planning where all entities in the scene are simple obstacles as the baseline and compare it to our normative navigation case. In the implementation, we use the `move_base` framework from ROS and add a costmap layer for normative navigation behaviors. We run A* global planner on the generated costmap at 2 Hz, and an elastic band local planner at 12 Hz, with local rolling window costmap of size 8 m². In simulation we use a sensor radius of 8 m; meaning we only consider people tracks within this region for updating the costmap.

Table 1. Evaluation results from the normative and classical behavior trajectories averaged over 25 runs. The norm tests are all statistically significant. RD is the relation disturbance assessed using μ_r norm test.

| | Personal | Social | RD | Path Length | CHC | Time |
|-----------|------------------------|-------------------------|-------------------------|-------------|-----------|------------|
| Normative | 29.8 ± 32.2 | 2056.4 ± 523.5 | 14.9 ± 35.3 | 22.5 ± 7.3 | 5.2 ± 1.8 | 38.9 ± 6.8 |
| Classical | 123.1 ± 69.4 | 2470.8 ± 629.8 | 82.9 ± 45.4 | 19.6 ± 6.1 | 4.5 ± 1.2 | 37.4 ± 7.3 |
| p-value | 1.2 × 10 ⁻⁴ | 1.49 × 10 ⁻² | 4.14 × 10 ⁻⁷ | 0.1276 | 0.1018 | 0.4537 |

As illustrated in Table 1, the resulting learned socially normative behavior performs passes all the norm test with significant difference. The statistical test

¹ https://github.com/srl-freiburg/pedsim_ros.

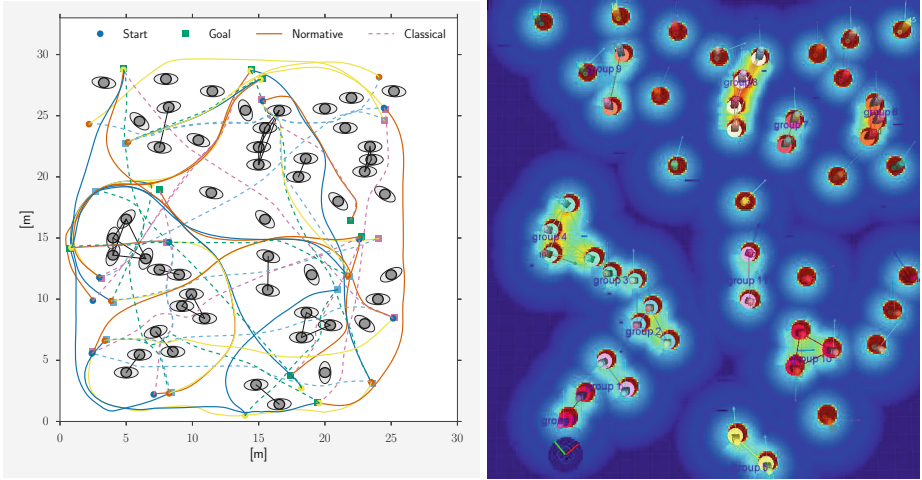


Fig. 2. Left: Example paths realized by the socially normative behavior vs classical (simple obstacle avoidance) behavior in a 30 m^2 crowded area. People are shown in top-view with head and shoulders, Relations between people are shown in black lines. Start locations are filled circles while goals are filled squares. **Right:** Socially normative costmap using the learned cost function. Jet colored (red — highest, blue — lowest). Arrows indicate velocity vectors while lines connecting people are pairwise relations. (Color figure online)

was done using t -test with null hypothesis being; no difference in norm tests between normative and classical behaviors. This null hypothesis is successfully rejected as shown in Table 1 in the first three columns. Additionally, the normative path takes longer paths and make more heading changes as expected but this difference is not statistically significant. While this does not necessarily confirm or deny that normative behavior results in similar performance with respect to task metrics, our intuition tells us that this maybe the case. Figure 2 (right) shows example paths from the two behaviors, and (left) the resulting costmap of the computed using the learned cost function, which enables the robot to drive in a socially normative manner.

6 Related Work

Efforts to generate normative behaviors for robots have only recently began, and as such, most attempts focus of very specific aspects of the task. To the best of our knowledge, this is the first comprehensive attempt at unifying these disparate approaches into one formalism. Nevertheless, we highlight here some of the recent works touching on the different aspects of the task. Learning cost functions for normative behavior is most studied, especially using LFD techniques as in [7, 9, 16, 23], and also manually designed cost functions [12, 19]. The formalism presented here subsumes the approaches in [7, 12, 16] among others,

while still providing a general picture of the task. Other attempts to realize a framework for robot behavior such as [8] are very limited to simple interaction experiments. The framework of [5] is the closest our formalism, though it is a very preliminary effort, and is limited to robot navigation tasks with no explicit treatment of norms. Other works like [3] are seen as too broad, leaving practical implementation aspects still undefined.

7 Conclusions

We have presented a unified formalism for normative robot behaviors, giving practical yet precise definitions for norms and normative behaviors while also the technical means for generating and evaluating such behaviors. We highlighted the key technical requirements needed to realize normative behavior and in particular the dependence on perception and uncertainty reasoning. We have also demonstrated in a case study, how the formalism can be used to model, generate and evaluate socially normative behaviors for a mobile service robot operating in public spaces. In the future, we plan to incorporate the formalism into life-long learning systems for automatic learning of these normative behaviors.

References

1. Castelfranchi, C.: Modelling social action for AI agents. *Artif. Intell.* **103**(1), 157–182 (1998)
2. Chakraborti, T., Talamadupula, K., Zhang, Y., Kambhampati, S.: Interaction in human-robot societies. In: *AAAI Workshop on Symbiotic Cognitive Systems* (2016)
3. Dautenhahn, K.: Socially intelligent robots: dimensions of human-robot interaction. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **362**(1480), 679–704 (2007)
4. Dondrup, C., Bellotto, N., Hanheide, M., Eder, K., Leonards, U.: A computational model of human-robot spatial interactions based on a qualitative trajectory calculus. *Robotics* **4**(1), 63–102 (2015)
5. Gómez, J.V., Mavridis, N., Garrido, S.: Social path planning: generic human-robot interaction framework for robotic navigation tasks. In: *International Workshop on Cognitive Robotics Systems: Replicating Human Actions and Activities* (2013)
6. Hall, E.T., Birdwhistell, R.L., Bock, B., Bohannon, P., Diebold Jr., A.R., Durbin, M., Edmonson, M.S., Fischer, J., Hymes, D., Kimball, S.T., Barre, W.L., Lynch, J.S., McClellan, J.E., Marshall, D.S., Milner, G.B., Sarles, H.B., Trager, G.L., Vayd, A.P.: *Proxemics [and comments and replies]*. *Current anthropology*, pp. 83–108 (1968)
7. Henry, P., Vollmer, C., Ferris, B., Fox, D.: Learning to navigate through crowded environments. In: *Proceeding of International Conference on Robotics and Automation (ICRA)* (2010)
8. Huang, C.M., Mutlu, B.: Robot behavior toolkit: generating effective social behaviors for robots. In: *Proceeding of ACM/IEEE international conference on Human-Robot Interaction (HRI)* (2012)
9. Kretzschmar, H., Spies, M., Sprunk, C., Burgard, W.: Socially compliant mobile robot navigation via inverse reinforcement learning. *Int. J. Robot. Res.* (2016)
10. Kruse, T., Pandey, A.K., Alami, R., Kirsch, A.: Human-aware robot navigation: a survey. *Robot. Auton. Syst.* (2013)

11. Linder, T., Arras, K.O.: Real-time full-body human attribute classification in RGB-D using a tessellation boosting approach. In: *Proceeding of International Conference on Intelligent Robots and Systems (IROS)* (2015)
12. Lu, D.V., Allan, D.B., Smart, W.D.: Tuning cost functions for social navigation. In: Herrmann, G., Pearson, M.J., Lenz, A., Bremner, P., Spiers, A., Leonards, U. (eds.) *ICSR 2013. LNCS (LNAI)*, vol. 8239, pp. 442–451. Springer, Heidelberg (2013). doi:[10.1007/978-3-319-02675-6_44](https://doi.org/10.1007/978-3-319-02675-6_44)
13. Mutlu, B., Forlizzi, J.: Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction. In: *Proceeding of ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2008)
14. Ng, A.Y., Russell, S.J.: Algorithms for inverse reinforcement learning. In: *International Conference on Machine Learning (ICML)* (2000)
15. Okal, B., Arras, K.O.: Towards group-level social activity recognition for mobile robots. In. *IROS Assistance and Service Robotics in a Human Environments Workshop* (2014)
16. Okal, B., Arras, K.O.: Learning socially normative robot navigation behaviors using bayesian inverse reinforcement learning. In: *Proceeding of International Conference on Robotics and Automation (ICRA)* (2016)
17. Rios-Martinez, J., Spalanzani, A., Laugier, C.: From proxemics theory to socially-aware navigation: a survey. *Int. J. Soc. Robot.* **7**(2), 137–153 (2015)
18. Sardar, A., Joosse, M., Weiss, A., Evers, V.: Don't stand so close to me: users' attitudinal and behavioral responses to personal space invasion by robots. In: *Proceeding of ACM/IEEE international conference on Human-Robot Interaction (HRI)* (2012)
19. Sisbot, E., Marin-Urias, L., Alami, R., Simeon, T.: A human aware mobile robot motion planner. *IEEE Trans. Robot. Autom. (TRO)* **23**(5) (2007)
20. Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., Goodrich, M.: Common metrics for human-robot interaction. In: *Proceeding of ACM/IEEE international conference on Human-Robot Interaction (HRI)* (2006)
21. Trautman, P., Ma, J., Murray, R.M., Krause, A.: Robot navigation in dense human crowds: the case for cooperation. In: *Proceeding of International Conference on Robotics and Automation (ICRA)* (2013)
22. Turnwald, A., Althoff, D., Wollherr, D., Buss, M.: Understanding human avoidance behavior: interaction-aware decision making based on game theory. *Int. J. Soc. Robot.* **8**(2), 331–351 (2016)
23. Vasquez, D., Okal, B., Arras, K.O.: Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison. In: *Proceeding of International Conference on Intelligent Robots and Systems (IROS)* (2014)