A Framework for Modelling Local Human-Robot Interactions Based on Unsupervised Learning

Rafael Ramón-Vigo[1(✉)], Noé Pérez-Higueras[1], Fernando Caballero[2], and Luis Merino[1]

[1] School of Engineering, Universidad Pablo de Olavide, Crta. Utrera km 1, Seville, Spain
{rramvig,noeperez,lmercab}@upo.es
[2] Department of System Engineering and Automation, University of Seville, Camino de Los Descubrimientos, s/n, Seville, Spain
fcaballero@us.es

Abstract. This paper addresses the problem of teaching a robot interaction behaviors using the imitation learning paradigm. Particularly, the approach makes use of Gaussian Mixture Models (GMMs) to model the physical interaction of the robot and the person when the robot is teleoperated or guided by an expert. The learned models are integrated into a sample-based planner, an RRT*, at two levels: as a cost function in order to plan trajectories considering behavior constraints, and as configuration space sampling bias to discard samples with low cost according to the behaviors. The algorithm is successfully tested in the laboratory using an actual robot and real trajectories examples provided by an expert.

1 Introduction

In the TERESA Project[1], telepresence robots are enhanced to navigate autonomously in social settings. The project considers the development of techniques for safe and efficient obstacle avoidance while reaching navigation goals. This task becomes more challenging when people are considered and explicitly modeled into the navigation approach.

Human-aware, or social, navigation is a complex task that has been addressed using different approaches in the robotics community. Many novel approaches are based on learning socially acceptable behaviors from real data collected under various social situations, avoiding the need of a handcrafted explicit formulation of the behaviors. For instance, supervised learning is used in [16] 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., [11] to determine socially-normative motion prototypes, which are then employed to infer social costs when planning paths. A model based on social forces is employed in [7], where the parameters for the social forces are learnt from feedback provided by users.

[1] http://teresaproject.eu/.

A. Agah et al. (Eds.): ICSR 2016, LNAI 9979, pp. 32–41, 2016.
DOI: 10.1007/978-3-319-47437-3_4

An additional approach is learning from demonstrations, or imitation learning, [2]: an expert teaches the robot how it should navigate among humans. We can leverage the fact that we have a telepresence robot in the TERESA Project, so we can extract useful information from the users of the robot (on the pilot side). Having examples of (teleoperated) robot paths and the relevant configurations (features) of the performed task opens the door for extracting the relevant relations or constraints that best represent such kind of paths. The main hypothesis is that these paths enclose the social implications that a human takes into account when he is performing such task, at least in the same situations performed in the experiments.

Inverse Reinforcement Learning (IRL) [1] techniques are a good candidate to derive such models: a reward (or cost) function is recovered from the expert behavior, and then used to obtain a corresponding robot policy. In [8], a path planner based on inverse reinforcement learning is presented. IRL for social navigation is also considered in [13].

The above mentioned approaches typically make use of discrete Markov Decision Processes (MDPs) as the underlying model. However, it is complex to encode general problems with MDPs due to its computational complexity [12]. Our objective is to use state of the art sampling-based planners, as optimal Rapidly exploring Random Trees (RRT*), to be able to work on continuous configuration spaces. These planners already reason about obstacles present in the environment, and the goal is to incorporate into them information about the social task at hand from data.

Gaussian Mixture Models (GMMs) offer a flexible framework to model the relationships between the relevant features that arise when the robot is performing a particular navigation task. GMMs are a well-suited representation for unsupervised extraction of continuous feature distributions, and they have also shown their utility as models for robot skills representations in Programming by Demonstration (PbD) settings [3].

In [5], the author presents a PbD framework in which GMMs are used to retrieve the statistical constraints of several demonstrations of a particular task, in a manner similar to the approach in [4]. After that, a sampled-based planner based on RRT is used. In this paper we aim to go a step further and use GMMs to incorporate robot navigation behavior into a cost-based RRT* planner. The goal is to find a safe path which imitates a behaviour by remaining within statistically determined constraints. For doing this, we propose, first, to bias the RRT* random samples towards the regions of the configuration space that comply with the model of the task extracted from data and, second, including a new cost function into the RRT* planner to better account for paths that follow the learned behaviors. At the same time, this permits better generalization to new situations by still finding short length paths for different conditions.

The paper is structured as follows: Sect. 2 takes care about the tasks being demonstrated by the user while Sect. 3 introduces GMMs. Section 4 details how the GMM can be included into the RRT* planner. Then, Sect. 5 presents the experimental setup, the metrics and some simulations to validate both GMM

learning and its integration with a sampling-based planner. Finally, Sect. 6 shows the conclusions and future works.

2 Demonstrated Tasks

As described before in the introduction, human-awareness is critical for a successful deployment of a robotic application in a space shared with persons. In the TERESA Project we are interested in several social situations, such as avoiding people while navigating, approaching a person to start a conversation, following a person or keep a conversation while moving to another place. For the sake of simplicity, in this work we focus in two particular tasks in order to illustrate this approach.

Avoiding. The robot avoids an standing person that is facing to it. The avoidance maneuver can be performed by passing through by the left or by the right.

Approaching. The robot approaches a standing person in an arbitrary orientation. When the person is looking towards the robot, it performs the shortest path to approach the person. When the standing person is back to the robot, the demonstration trajectories do not follow the shortest path. Rather, the robot tends to take curving paths, also by the left or by the right.

In this work, a Giraff robot is teleoperated by an expert. This user is asked to demonstrate the previous tasks as accurately as necessary a number of times by means of piloting the robot.

3 GMMs for Interaction Modeling

A proper choice of the relevant features $\mathbf{f} = [f_1, f_2, \ldots, f_n]^T$ when encoding a particular navigation task is crucial, as it provides part of the solution to the problem of defining what is important to imitate. In this paper, we have considered as features the distance and the relative angle between the robot and the person in the scene, logged with a timestamp. The features extracted at each time instant are the distance to that person (d) and the relative angle (θ). Thus, a set composed by N datapoints $\zeta = \{\zeta_j\}_{j=1}^{N}$ of $D = 2$ dimension is considered, where time is left out because the dynamics of the behaviours are not considered in this paper.

According to a previous work [14], the features $(d - \theta)$ considered here allows us to model the tasks at hand. However, the selection of features is not a limitation of the technique, and other features can be added for more complicated tasks, including those ones that can describe the dynamics of the task execution, like time and velocities.

From the demonstrated trajectories of the features $\mathcal{D}$, it is possible to extract a GMM, so that the probability of a particular combination of feature values is given by:

$$p(\mathbf{f}|\mathcal{D}) = \sum_{i=1}^{k} \omega_i \mathcal{N}(\mathbf{f}; \mu_{\mathcal{D}}^i, \Sigma_{\mathcal{D}}^i) \tag{1}$$

with k Gaussian *modes*. The GMM is then described by the set of parameters $\left\{\omega_i, \mu_{\mathcal{D}}^i, \Sigma_{\mathcal{D}}^i\right\}_{i=1}^K$, respectively representing the *prior* probabilities, *centers* and *covariance* matrices of the model. The *prior* probabilities, ω_i, satisfy $\omega_i \in [0, 1]$ and $\sum_{i=1}^K \omega_i = 1$.

GMM parameters are learnt by using the *Expectation-Maximization* (EM) algorithm [6] that is seeded with an initial estimate of density centers calculated with the k-means algorithm. A drawback of EM is that the optimal number of components k in a model may not be known beforehand. One usual criterion for model selection is the Bayesian Information Criterion (BIC) [15], but for the experiments presented in this work, we have tested empirically the best number of components that can fit the demonstrated example.

4 The Reproduction Planner

RRT* [10] is a technique for (asymptotical) optimal motion planning. It considers that a cost function is associated to each point $\mathbf{x}$ in the configuration space (a vector representing the position of the robot in our case). The RRT* seeks to obtain the trajectory ζ^* that minimizes the total cost along the path $c(\zeta)$. It does so by randomly sampling the configuration space and creating a tree towards the goal. The paths are then represented by a set of discrete configuration points $\zeta = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$. Each point of the trajectory can be also associated to the values of the features for that point, so that it can be also seen as a trajectory described in feature space $\zeta = \{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N\}$. This paper extend the standard RRT* algorithm with the learned GMM at two levels:

1. Including a new task-similarity cost into the evaluation of the node's costs. The GMM obtained encompasses the most likely configurations of the task. Thus, when a node is proposed to be added to the RRT* tree, a cost based on the GMM is derived and used. The objective is to increase the cost of those configurations that are unlikely according to the learnt GMM. Thus, the likelihood is inverted to obtain that cost. Notice that the likelihood is a density of occurrence, so it is necessary to give a low bound to keep the inverse within the interval [0,1]. To this end, it has been chosen to truncate the likelihood to a certain low value δ.
2. Providing the planner with the most likely subspace to perform the sampling of the RRT*. If the RRT* knows what are the most likely paths, then we can bias the sampling of the configuration space to these areas, reducing the probability of sampling useless states and, hence, reducing the computational costs to obtain a solution. A parameter determines the amount of GMM bias introduced into the planner.

5 Experimental Results

5.1 Gathering the Data for Learning

Several experiments were performed to retrieve exemplary trajectories to feed the GMM learning phase. Those experiments took place inside a clear room, free of obstacles between the person and the robot while performing the scenes described in Sect. 2. A set of 9 trials for each homotopy in each task were logged. The study was recorded using a motion capture system (OptiTrack) and the robot and person's poses were extracted automatically from the data.

The gathered data is used to derive the GMM models of the features using the method described in Sect. 3, one model for each task. Those models are then integrated in the RRT* as explained in Sect. 4. This method is called GMM-RRT*. A set of GMM models in $x - y$ space have been also derived in order to test the GMM-RRT planner [5], used as a baseline. The parameters used was $k = 19$ GMM modes (for each task and method) and a value of $\delta = 0.001$.

5.2 Metrics

We propose two metrics to compare the obtained paths from the different planners and with respect to the demonstrated trajectories, as used in [9]:

The first metric is called *Trajectory Difference Metric* (TDM), which is defined as follows:

$$TDM(\zeta_D, \zeta_P) = \frac{1}{|\zeta_D|} \sum_{i=1}^{|\zeta_D|} \min \overline{\zeta_D(i)\zeta_P} = \frac{1}{|\zeta_D|} \frac{1}{|\zeta_P|} \sum_{i=1}^{|\zeta_D|} \sum_{j=1}^{|\zeta_P|} \min \overline{\zeta_D(i)\zeta_P(j)} \quad (2)$$

where $\zeta_P(j)$ and $\zeta_D(i)$ are the points of the two trajectories ζ_P and ζ_D to be compared, and $\overline{\zeta_D(i)\zeta_P(j)}$ is the distance between two points. $|\zeta_D|$ and $|\zeta_P|$ are the number of samples of each trajectory. This metric gives an idea of the similarity of two given trajectories. The final metric is given by the averaged value of this metric for all the planned and demonstrated trajectories:

$$TDM = \frac{1}{|D|} \frac{1}{|P|} \sum_{D,P} TDM(\zeta_D, \zeta_P) \quad (3)$$

where D and P are the number of Demonstrated and Planned trajectories, respectively.

The second metric is the resulting averaged trajectory length ratio l_e, expressed as the mean of the ratio of the absolute value of the difference between the planned trajectories and the demonstrated trajectories lengths divided by the demonstrated trajectories length:

$$l_e = \frac{1}{|D|} \frac{1}{|P|} \sum_{D,P} \frac{|l(\zeta_D) - l_(\zeta_P)|}{l(\zeta_D)} \quad (4)$$

5.3 Results

Costs Evaluation. This section is oriented to evaluate the convergence speed of the RRT*-base planner to the optimal path in term of costs. Figure 1 shows the evolution of the solution path cost versus the number of iterations using 100 % and 0 % GMM bias, for the "Avoiding a Person" task. The allowed *planning time* for this comparison was 100 s in order to converge to the optimal cost. It can be seen in this example how the planner that includes the GMM *sampling* is faster.

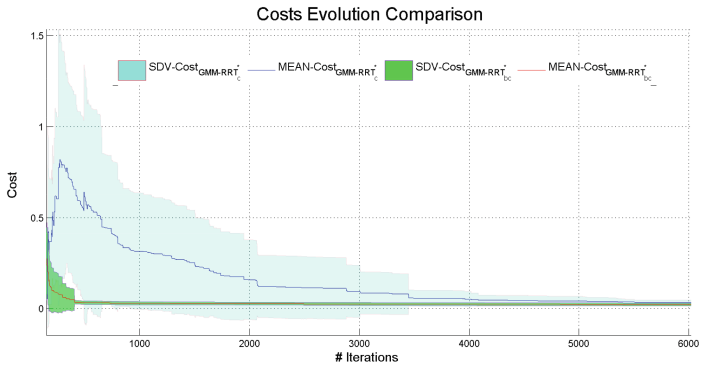
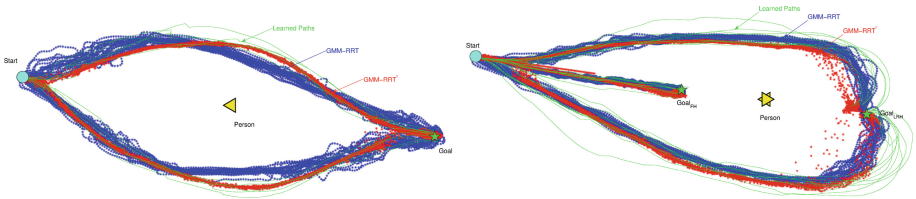


Fig. 1. Mean cost value and the standard deviation obtained for an GMM-RRT* with (red and green, subindex *bc*) and without (blue, subindex *c*) GMM sampling bias. (Color figure online)

Metrics Performance. For the following results a *mixed-sampling* strategy has been adopted: 95 % of the time a GMM sampling is employed, while the remaining 5 % it is the *uniform* sampling. We aim to take advantage of the learned models and also allow a degree of randomness when sampling configurations. This feature is only applicable to the GMM-RRT* planner presented in this paper. The construction and the use of the GMM model for the GMM-RRT planner must satisfy certain restrictions that make sampling possible only on consecutive nodes. For further details please consult [5]. The planner is given enough *planning time* to converge. Figure 2 shows a complete set of 25 trials for each task, homotopy and planner, with the parameters explained before.

Regarding on how well both planners imitate the demonstrated trajectories, Table 1 shows the values obtained when the different planners are used in both tasks based on the metrics related in Sect. 5.2. It can be seen in that table that the planner proposed in this work outperforms in nearly every homotopy considered.

Figure 3 shows the same comparison of the metrics for different *planning times* for the GMM-RRT* approach, in contrast with the metric values obtained for the GMM-RRT (first two columns in the figure).



a) Task 1: Avoid a Person. b) Task 2: Approach a Person.

Fig. 2. Demonstrated (green) and planned trajectories (red for GMM-RRT* and blue for GMM-RRT) are depicted. The "Start" and "Goal" states are also shown, while the "Person's" gaze is represented by a triangle. (Color figure online)

Table 1. Trajectory quality for both tasks. Smaller values are better for all metrics. The best values are highlighted in boldface.

Planner	Task 1: Avoiding a Person		Task 2: Approaching a Person	
	TDM (m)	l_e(%)	**TDM** (m)	l_e(%)
Right Homotopy				
GMM-RRT*	**0.0565** ±0.0152	**3.66** ±2.64	**0.0906** ±0.0176	**3.69** ±2.54
GMM-RRT	0.0803 ± 0.0205	4.00 ± 2.51	0.0917 ± 0.0161	4.96 ± 3.45
Left Homotopy				
GMM-RRT*	**0.0484** ±0.0109	4.96 ± 2.37	**0.0612** ±0.0218	**4.22** ±2.50
GMM-RRT	0.0676 ± 0.0169	**4.31** ±3.1	0.0666 ± 0.0169	4.56 ± 3.62
Frontal Homotopy				
GMM-RRT*			0.0376 ± 0.0141	11.35 ± 5.35
GMM-RRT			**0.0337** ±0.0184	**8.55** ±3.34

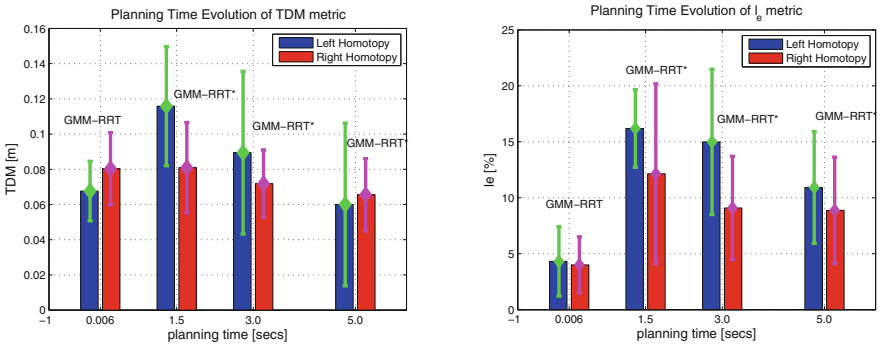


Fig. 3. Mean and standard deviation of the metrics for different planning times and the "Avoiding a Person" task.

We have started with a planning time of 1.5 s because it is the mean time in which the proposed GMM-RRT* planner is able to find the first solution for the tasks presented here and for the bias of 95 %. For 3 s the GMM-RRT* obtain comparable results on the TDM metric.

Managing the Homotopies. As commented in Sect. 2, the tasks being learned by the robot may be composed by several homotopies. Figure 4 illustrates the models of the Approaching a Person task.

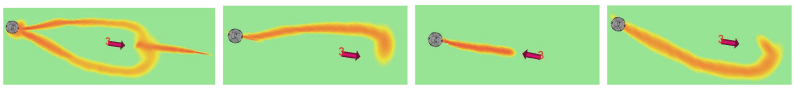


Fig. 4. First-Left: The model includes the three demonstrated homotopies. Second-Left to Right: One model per each demonstrated homotopy.

This is a clear disadvantage of the GMM-RRT planner. Not only the social situation has to be known beforehand to choose between the homotopy models, but also once a model has been selected, it can occur that an obstacle hampers the execution of the plan. In such a situation, both planners can take advantage of the variability in the execution of the demonstrated task, encoded by the GMMs covariance matrices, to avoid the obstacle, but sometimes this could not be enough. For instance, in the situation shown in Fig. 5 it can be seen how the GMM-RRT planner is not able to find a free path to the goal, although it exists. However, the GMM-RRT* planner presented here is able to naturally choose between the available homotopies to reach the goal.

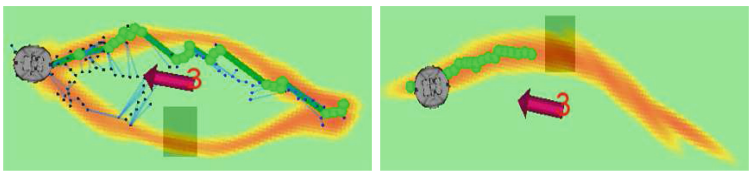


Fig. 5. "Avoiding a Person" task. Shadowed area represents an obstacle. Left: Model includes 2 homotopies, used by GMM-RRT*. Planning time: 3 s. Right: Only right homotopy, used by GMM-RRT. Planning time: 100 s.

Generalization. Regarding to the generalization of the proposed approach to other scenarios, the *approaching a person* task is modified so an obstacle (as a rectangular shadowed area in Fig. 6) is introduced in the robot path to its goal: The GMM-RRT* planner still can reach the goal due to the *mixed-sampling* strategy adopted: the *uniform* bias sampling allows to random explore the state space outside the learned demonstrations.

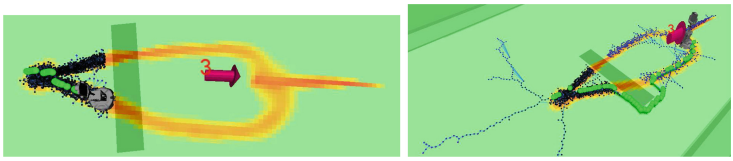


Fig. 6. Left: Simulated trial without *uniform* sampling. Right: 95 % GMM and 5 % *uniform* sampling.

The simulations shown in Fig. 6 were within a 100 s time horizon. A 100 % GMM sampling was not able to find a path that can handle with the obstacle included. If a *mixed-sampling* strategy is allowed, within the same time horizon, it can be seen how the goal is reached successfully. The percentage of bias, thus, offers a trade-off between the imitation capabilities and the planning time required to obtain a path. An adaptive solution that modifies this bias after obtain a first good path is left for future work.

6 Conclusions and Future Work

In this paper we present a planning algorithm based on RRT*, which is capable of infer the most suitable socially-aware paths in two situations (or tasks): avoiding a standing person and approaching forwards or backwards a standing person. Both tasks are statistical characterized from real experiments by using two different GMMs, which are later used in two ways: to guide the state-space sampling step and to include a cost term into the standard RRT* algorithm.

We evaluate the approach presented here jointly with a state of the art algorithm based on RRT and GMM. The comparison includes a metric performance to measure the similarity between the planned trajectories with the learned ones, for different planning times, and how both approaches perform in simulated scenarios that can include slightly differences from where they were learned.

Although the RRT based planner is quicker to get a suitable path, the approach presented here is able to manage homotopies and generalizes better when tackling with unexpected situations (such as obstacles).

Both tasks described in this work are only a simple and specific use case of the proposed planner. As future work we plan to evaluate the use of other features including velocity, time and some environment descriptors that could improve the generalization of the learned behavior to these and new tasks. Also, we will perform real user tests to analyze how the learning affects the readability of the learned behavior.

References

1. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proceeding of the Twenty-First International Conference on Machine Learning, ICML 2004, pp. 1–6. ACM, New York (2004)

2. Argali, B., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstrations. Robot. Auton. Syst. **57**, 469–483 (2009)
3. Calinon, S.: Robot Programming by Demonstration: A Probabilistic Approach. EPFL/CRC Press, Boca Raton (2009)
4. Calinon, S., Billard, A.: A probabilistic programming by demonstration framework handling constraints in joint space and task space. In: Proceeding IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS (2008)
5. Claassens, J.: A RRT-based path planner for use in trajectory imitation. In: Proceeding of the International Conference on Robotics and Automation, ICRA, pp. 3090–3095. IEEE (2010)
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc.: Ser. B **39**(1), 1–38 (1977)
7. Ferrer, G., Garrell, A., Sanfeliu, A.: Robot companion: a social-force based approach with human awareness-navigation in crowded environments. In: Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS. pp. 1688–1694 (2013)
8. Henry, P., Vollmer, C., Ferris, B., Fox, D.: Learning to navigate through crowded environments. In: Proceeding of the International Conference on Robotics and Automation, ICRA, pp. 981–986 (2010)
9. Islas Ramírez, O., Khambhaita, H., Chatila, R., Chetouani, M., Alami, R.: Robots learning how and where to approach people. In: RO-MAN 2016 25th, IEEE International Symposium on Robot and Human Interactive Communication (2016, to appear)
10. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Int. J. Robot. Res. **30**(7), 846–894 (2011)
11. Luber, M., Spinello, L., Silva, J., Arras, K.: Socially-aware robot navigation: a learning approach. In: Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pp. 797–803. IEEE (2012)
12. Okal, B., Gilbert, H., Arras, K.O.: Efficient inverse reinforcement learning using adaptive state-graphs. In: Learning from Demonstration: Inverse Optimal Control, Reinforcement Learning and Lifelong Learning Workshop at Robotics: Science and Systems (RSS), Rome, Italy (2015)
13. Perez-Higueras, N., Ramon-Vigo, R., Caballero, F., Merino, L.: Robot local navigation with learned social cost functions. In: Proceeding of the 11th International Conference on Informatics in Control, Automation and Robotics, ICINCO, vol. 02, pp. 618–625 (2014)
14. Ramon-Vigo, R., Perez-Higueras, N., Caballero, F., Merino, L.: Analyzing the relevance of features for a social navigation task. Robot 2015: Second Iberian Robotics Conference. AISC, vol. 418, 10.1007/978-3-319-27149-119 edn, pp. 235–246. Springer, Heidelberg (2016)
15. Schwarz, G.: Estimating the dimension of a model. Ann. Stat. **6**, 461–464 (1978)
16. Trautman, P., Krause, A.: Unfreezing the robot: Navigation in dense, interacting crowds. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pp. 797–803. IEEE (2010)