The 23rd IEEE International Symposium on Robot and Human Interactive Communication
August 25-29, 2014. Edinburgh, Scotland, UK,

Transferring human navigation behaviors into a robot local planner

Rafael Ramón-Vigo[a], Noé Pérez-Higueras[a], Fernando Caballero[b] and Luis Merino[a]

Abstract— **Robot navigation in human environments is an active research area that poses serious challenges. Among them, social navigation and human-awareness has gain lot of attention in the last years due to its important role in human safety and robot acceptance. Learning has been proposed as a more principled way of estimating the insights of human social interactions. In this paper, inverse reinforcement learning is analyzed as a tool to transfer the typical human navigation behavior to the robot local navigation planner. Observations of real human motion interactions found in one publicly available datasets are employed to learn a cost function, which is then used to determine a navigation controller. The paper presents an analysis of the performance of the controller behavior in two different scenarios interacting with persons, and a comparison of this approach with a Proxemics-based method.**

Fig. 1: The FROG project aims to deploy a guiding robot with a fun personality, considering social feedback, in the Royal Alcazar of Seville and the Zoo of Lisbon. A typical situation of the first scenario is presented here.

I. Introduction

This work is part of the Fun Robotic Outdoor Guide (FROG) FP7 project[1]. It aims to deploy a guiding robot in touristic sites. While robot guides have been developed for more than a decade [1], [2], the project considers as new contributions the development of social behaviors and a wining robot personality by integrating social feedback, as well as the robust operation in outdoors crowded scenarios. The project aims to demonstrate the operation of the robot in the Lisbon City Zoo and the Royal Alcazar in Seville (see Fig. 1). Acting in these crowded scenarios (the Royal Alcazar may have more than 5000 visits per day, totaling 1.5 million-visitors per year) involves not only ensuring a safe and efficient navigation but also social interaction and social awareness when performing the robot tasks.

In scenarios involving interaction with humans, these considerations have to be taken into account in the entire robot planning and navigation stack, from task planning [3], task supervision and execution [4] to path planning and execution [5], [6], [7].

Focusing on the particular case of the navigation stack, current path planners are typically used to determine paths that minimize time or length, which does not translate to social paths in general. This requires determining costs related to social compliance. Some authors [5], [8] 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 behaviours may be inappropriate [9]. Many have derived costs from proxemics theory [10], but as commented in [11], proxemics is focused on scenarios in which people interact, and it could not be suitable for navigation among people.

Thus, learning these costs and models from data seems a more principled approach. In the last years, several contributions have been presented in this direction: supervised learning is used in [7] 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. In [12], 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 [13]: an expert indicates the robot how it should navigate among humans. One way to implement it is through inverse reinforcement learning [14], in which a reward (or cost) function is recovered from the expert behavior, and then used to obtain a corresponding robot policy.

In [15], 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 [15] the costs are used to path plans, in [16] 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.

In this paper, 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

This work is partially funded by the European Commission 7th Framework Programme under grant agreement no. 288235 (FROG) and the project PAIS-MultiRobot, funded by the Junta de Andalucía (TIC-7390)

[a]Rafael Ramón-Vigo, Noé Pérez-Higueras and Luis Merino are with the Pablo de Olavide University, Seville (Spain) {rramvig,noeperez,lmercab}@upo.es
[b]Fernando Caballero is with University of Seville, Seville (Spain) fcaballero@us.es

[1]http://www.frogrobot.eu

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, and we analyze and compare this approach with a Proxemics-based cost function.

The structure of the paper is as follows: next section describes the learning approach of social costs. Then, Section IV deals with the evaluation of the generalization of the social cost function and comparison with Proxemics-based method. Finally, the conclusions and future open lines are discussed.

II. Learning the social cost function

The learning of the cost function is accomplished by using inverse reinforcement learning (IRL, [14], [15]). IRL assumes that the expert 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* is the finite set of possible states $s \in S$; the *action space* is defined as the finite set of possible actions $a \in A$. At every step, an action is taken and a reward is given (or cost is incurred). After performing an action a, the state transition is modeled by the conditional probability function $T(s', a, s) = p(s'|a, s)$. At every time instant then the state is observed. The reward obtained at each step is denoted $R(s, a)$. A function $a = \pi(s)$ that maps a state to an action is called a policy. A policy that maximizes the sum of expected rewards, or *value*, earned during D time steps $E[\sum_{t=0}^{D} \gamma^t R(s, a)]$ is called an *optimal* policy. To ensure that the sum is finite when $D \to \infty$, rewards are weighted by a discount factor $\gamma \in [0, 1)$.

The objective of IRL is to determine the reward function $R(s, a)$ that the expert is following by observing the expert acting in the real world, assuming that it is executing a policy 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.

A. Model

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 constitute 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 [15], 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 [15], 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

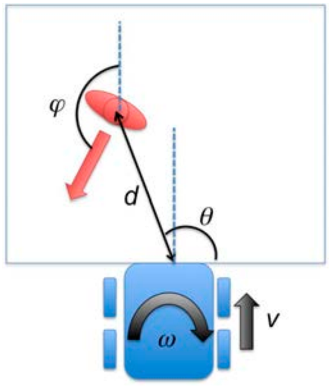

Fig. 2: The state is defined as the relative pose of the person with respect to the robot, encoded as the relative position of the person in polar coordinates (d, θ), and the approach angle φ. The actions (linear, v, and angular, ω, speeds) affect how this state evolves.

(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. 2). As the parametrization is local, the pose of the robot is not considered into the state space, resulting in a better coverage with training data and a better generalization for changing environments. Furthermore, as we are considering a local navigation planner, it is not required to have a prior probability distribution over the target positions on the global map, as required in [15], were social global paths are seek.

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

One hypothesis that will be analyzed in this paper is if 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.

III. Datasets and Training

A. Datasets

As a source of examples on which the pedestrians motion is extracted, The BIWI Walking Pedestrians dataset[2] [17] has been used (see Fig. 3). It consists of a bird view of two outdoors urban environments:

- The first proposed scenario (DS1) is a busy sidewalk next to an hotel entrance in Zurich (see Fig. 3, left).
- The second one (DS2), at the same dataset, is a bird view of the ETH main building, in Zurich as well (see Fig. 3, right).

For both scenarios, the positions and velocities of all persons and the corresponding timestamps are manually annotated.

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

Fig. 3: Example images of the BIWI Walking Pedestrian dataset used for learning [17]. Left: hotel entrance. Right: ETH main building.

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. By contrast, in the second dataset, the people flow may appear from anyway of the top and merge into such narrower corridor.

By considering just local pairwise relative motions between two persons to learn the social avoiding maneuvers in the expert local frame, we will have very similar situations in other scenarios with people. So, we expect that the behavior learned can be transferred between different scenarios, and they could be applied to scenarios like the zoo.

B. Training

We consider the algorithm Gaussian Process IRL (GPIRL) [18] 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. Thus, the GP allows to extrapolate the learnt reward function to other state spaces within the domain of the features considered.

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:

- 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. 2) 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).
- The action performed by the expert at the same time instant. 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 to have equal experiments, the number of samples is equalized for each episodes, by dividing them in several if needed.

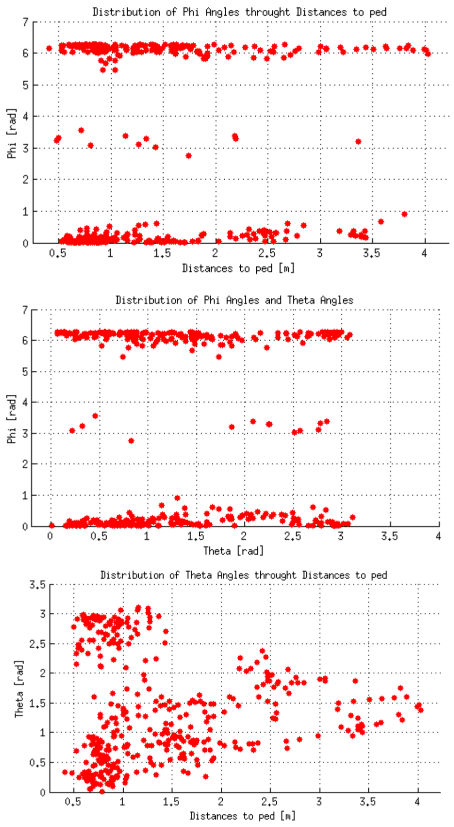

Fig. 4: Training points from all the episodes for scenario DS1. Top: Approach angle φ vs. distance d in the local frame. Middle: approach angle φ vs. θ. Bottom: polar coordinates (d, θ) of the closest person in the local frame.

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”. As we want to learn the task of avoiding people, it is necessary that a pedestrian exists within the local planning area for a certain number of time steps. This value has been tuned for these datasets, and it is a tradeoff between the temporally time horizon of a training episode for the MDP policy resolution and the number of trainings that we get with the current datasets.

Furthermore, we impose that these pedestrians have to move at least 2 meters from their start point. This is an heuristic filter to ensure that the pedestrian selected as expert is really moving and he is not just standing at the same place, i.e. waiting for the bus or for someone, or maybe seeing at some shop window. Both conditions allow us to focus at 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.

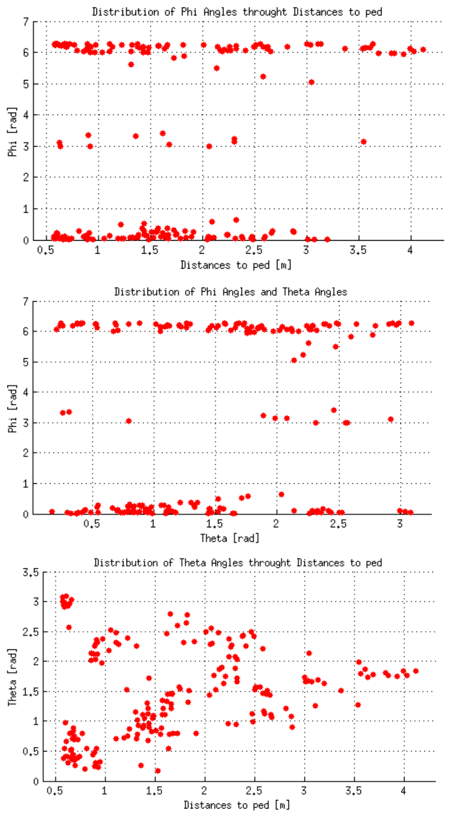

Fig. 5: Training points from all the episodes for scenario DS2. Top: Approach angle φ vs. distance d in the local frame. Middle: approach angle φ vs. θ. Bottom: polar coordinates (d, θ) of the closest person in the local frame.

C. Analysis of the data

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

Figure 4 shows the values of the features for the scenario 1 (DS1), the building entrance. It should be recalled that all the features are computed locally to the expert.

Fig. 4, 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. 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 $\theta \sim \pi$).

Fig. 4, top, shows the distance vs. approach angle φ, and θ vs. φ respectively. What it can be seen there is that typically in the 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 contrary direction ($\varphi \sim \pi$). There are nearly 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.

The same aspects can be observed in the training set for DS2, shown in Fig. 5.

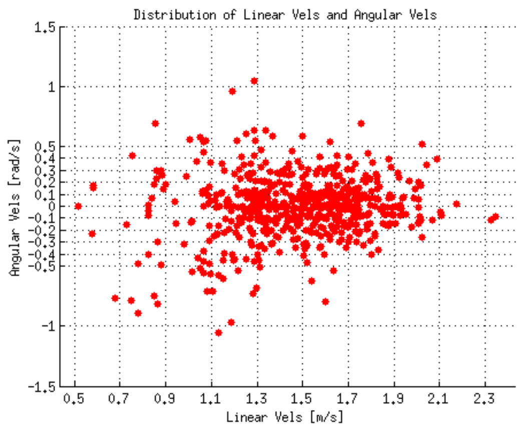

Fig. 6: Actions (linear and angular velocities) of the training samples for both scenarios DS1 and DS2. The discretization bins employed are also shown.

D. 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 discretized as follows:

- 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 $\phi \in [0\ 2\pi)$ is discretized into 8 bins of 0.69 rads.

Figure 6 shows the linear and angular velocities of 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. 6). 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.

IV. Evaluation

By using the 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.

TABLE I: IRL-based policy vs. Proxemics-based policy. Mean errors and standard deviations.

	Linear Vel(m/s)		Angular Vel(rad/s)	
	IRL closest	PRX closest	IRL 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

We compute the mean errors in 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.

A. *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 [19]. A cost function modeling the personal space is implemented as two Gaussians distributions as in [10]. The first function is asymmetric and placed 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 way.

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

B. *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 sum of the value function of the MDP for all the pedestrians on the local navigation area.

TABLE II: IRL-based policy using just the closest pedestrian (closest) and all pedestrians. Mean errors and standard deviations.

	Linear Vel(m/s)		Angular Vel(rad/s)	
	IRL closest	IRL all	IRL closest	IRL 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

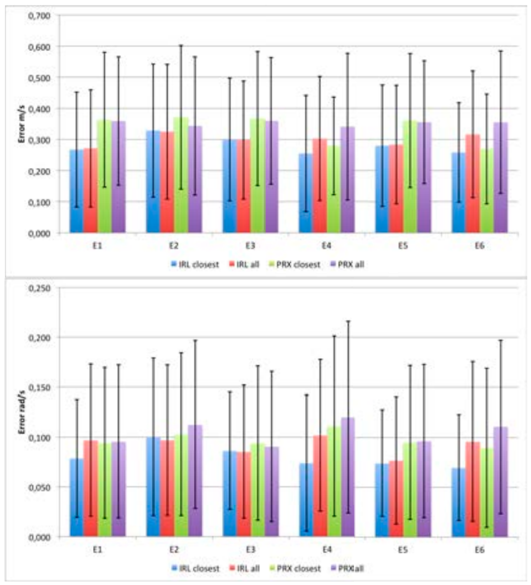

Fig. 7: Comparison of the different policies and evaluations. Top: error on linear velocities with respect to the ground truth actions. Bottom: error on angular velocities. IRL all and PRX all denote the IRL-based and Proxemics-based policies respectively, when considering all the pedestrians within the local navigation area.

In Table II it can be seen the comparison between the previous model, considering just the closest pedestrian (closest) and its generalization to all pedestrians (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.

C. *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 of pedestrians.

TABLE III: Scenarios comparison. Mean errors and standard deviations

Scenario DS1

IRL closest	Policy DS1	Policy DS2	Policy DS1+DS2
Linear Vel(m/s)	0.235 ± 0.214	0.242 ± 0.210	0.242 ± 0.226
Angular Vel(rad/s)	0.075 ± 0.071	0.064 ± 0.055	0.069 ± 0.063

Scenario DS2

IRL closest	Policy DS1	Policy DS2	Policy DS1+DS2
Linear Vel(m/s)	0.254 ± 0.234	0.297 ± 0.205	0.250 ± 0.155
Angular Vel(rad/s)	0.082 ± 0.071	0.093 ± 0.095	0.063 ± 0.055

Table III 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 Figures 4 and 5, 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.

Finally, Fig. 7 shows graphically the comparison for all policies and experiments.

V. Conclusions and Future Work

The paper has analyzed the use of inverse reinforcement learning to learn cost/reward functions from examples for the task of navigating among persons. The model employed and the methodology to extract the cost function from a public dataset are described. Furthermore, the cost function is used to derive a navigation policy. This navigation policy is compared to the original human behavior and a policy derived from a cost function derived from Proxemics.

The comparisons show that the IRL policy is better than the policy based on proxemics in all cases. Furthermore, the costs learnt can be used in different setups. On the other hand, a simple extension of the model to deal with more persons in the environment by linearly combining the cost functions associated to each person worsens the behavior. Therefore, other features may be more adequate when crowds are present. Furthermore, the variability of the obtained behaviors indicates that the model should be refined further.

As future work, we will develop models considering features based on densities and flows to compare with this models and analyzed its behavior. Moreover, we will consider unsupervised approaches to learn the underlying structure from the data available and extract exemplary motions.

Furthermore, we will integrate the insights into a real robot and perform experiments in crowded scenarios, performing a qualitative evaluation of the robot behavior.

References

[1] 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*, vol. 19, pp. 972–999, October 2000.

[2] R. Siegwart, K. O. Arras, S. Bouabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor, M. Meisser, R. Philippsen, R. Piguet, G. Ramel, G. Terrien, and N. Tomatis, "Robox at Expo.02: A large-scale installation of personal robots," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 203–222, March 2003.

[3] 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.

[4] A. Clodic, H. Cao, S. Alili, V. Montreuil, R. Alami, and R. Chatila, "SHARY: A Supervision System Adapted to Human-Robot Interaction," in *Experimental Robotics, The Eleventh International Symposium, ISER 2008, July 13-16, 2008, Athens, Greece*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and G. J. Pappas, Eds., vol. 54. Springer, 2008, pp. 229–238.

[5] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Siméon, "A Human Aware Mobile Robot Motion Planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[6] G. D. Tipaldi and K. O. Arras, "Planning Problems for Social Robots," in *Proceedings fo the Twenty-First Internacional Conference on Automated Planning and Scheduling*, 2011, pp. 339–342.

[7] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds." in *IROS*. IEEE, 2010, pp. 797–803.

[8] R. Kirby, J. J. Forlizzi, and R. Simmons, "Affective social robots," *Robotics and Autonomous Systems*, vol. 58, pp. 322–332, 2010.

[9] 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.

[10] R. Kirby, R. G. Simmons, and J. Forlizzi, "Companion: A constraint-optimizing method for person-acceptable navigation." in *RO-MAN*. IEEE, 2009, pp. 607–612.

[11] M. Luber, L. Spinello, J. Silva, and K. Arras, "Socially-aware robot navigation: A learning approach," in *IROS*. IEEE, 2012, pp. 797–803.

[12] 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*, Nov 2013, pp. 1688–1694.

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

[14] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 1–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015430

[15] P. Henry, C. Vollmer, B. Ferris, and D. Fox, "Learning to navigate through crowded environments," in *ICRA'10*, 2010, pp. 981–986.

[16] 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.

[17] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *International Conference on Computer Vision*, 2009.

[18] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with gaussian processes," in *Neural Information Processing Systems Conference*, 2011.

[19] E. T. Hall, *The Hidden Dimension*. Anchor, Oct. 1990.