Grant agreement no: FP7-600877

SPENCER:

Social situation-aware perception and action for cognitive robots

Project start: April 1, 2013

Duration: 3 years

DELIVERABLE 5.6

Task planner and robot supervision system

Due date: month 34 (May 2013)

Lead contractor organization: CNRS

Dissemination Level: PUBLIC

Contents

Abstract

This report presents the description of the supervision system and task planner of the SPENCER airport scenario and its implementation. The aim of the supervision system is to be able to refine and execute collaborative tasks with humans in a flexible and robust way, monitoring their actions and adapting its plans to provide a natural and efficient interaction.

1 Introduction

In WP 5.5. we have developed a system to control the robot operation activities, supporting human-robot collaborative tasks and producing socially acceptable behaviors. This goal poses several challenges. The robot needs to be able to reason on human observations, obtained through its sensors, in order to understand what's the best possible behavior, taking into account the overall goal and issues of naturalness and legibility. A first prototype of the components of this WP was developed in the OpenPRS (Procedural Reasoning Systems) [1], and described in [1]. For the final version, we decided to use the well known ROS framework, simplifying the communication with the rest of the architecture.

For a more detailed introduction, we refer to our work for the early prototype of the system [1] and on the second year of the project [2].

2 Situation Assessment

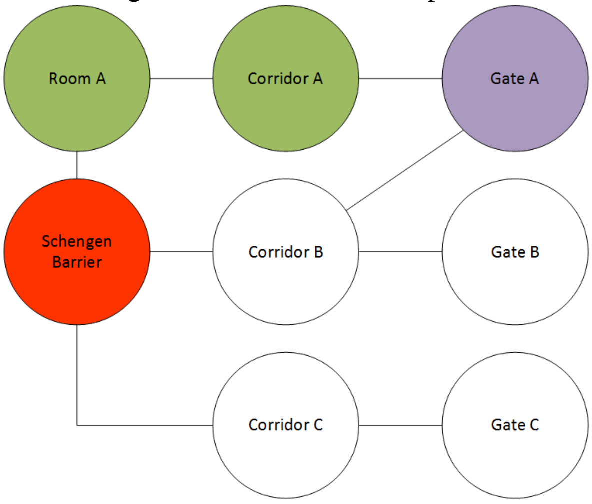
In order to reason and react on human activities, the robot needs to produce symbolic information from data obtained from the sensors. We developed a Situation Assessment component, able to perform different kinds of computations on sensor data and provide information such as: a) the distance and orientation of a human relative to the robot, b) the variation of the distance from a human to the robot c) if a human is currently moving.

In some situations, it's important to link human activities to the current environment. In this way we could estimate that, for example, a human looking at a monitor screen could need information. We perform this kind of reasoning by creating activity areas in the environment and linking them to different kind of computations. An activity area is a polygonal or circular area, which can be fixed or linked and updated with an entity's (object, human or robot) position. We studied two different activity areas: a) Information Screen Area, linked to information screens present in the environment; b) Touristic Point Area, linked to interesting attractions in the environment.

Work on this functionality was started during the second year of the project, with the definition of the component, and completed for the final version of the prototype.

[1] https://git.openrobots.org/projects/openprs

Figure 1: Plan Example. The colored nodes represent the plan, with the red node being the goal, the blue node the starting point and the green nodes intermediate points.



3 Supervision Goals

The supervision system is able to receive goals, either from a terminal or from the robot's graphical interface unit (GUI). The system is able to handle the following goals:

- Guide a group to a destination.
- Navigate to a destination.
- Change the destination of the current mission.

Additionally, the system can be set, using a terminal, to a ”busy” mode, where the robot won't accept goals from the GUI.

In our early prototype, we identified sets of use cases and robot modalities, which were fully implemented, with a working interface, for the final version.

4 Task Planning

The task planner has the objective to create a plan to achieve the current robot's goal. The plan consists of a set of locations, chosen from a high level semantic map that represents different areas of the environment. The map produced by the mapping layer is annotated offline and then read by the Task Planner, which represents it as a graph structure. We developed a first version of the Task Planner for the early prototype of the system. using the dijkstra algorithm, in order to select the best path from the graph structure.. For the final version we decided to use the well known A* algorithm, instead, and integrated the Task Planner fully with the rest of the system. Figure 1 shows an example of a possible plan.

5 Robot Navigation

In the Schengen barrier scenario, the robot will navigate in the Schiphol Airport, a very large area, which can pose problems for the motion planning modules. To handle this situation, we divide the environment map into a set of sub maps, which can be easily handled by the motion planners. As previously said, the map is annotated offline and divided into a graph. In the annotation phase, the map is split into sub-maps, where each sub-map represents three linked nodes $(n1, n2, n3)$ in the graph. This sets are not ordered, meaning that $(n1, n2, n3) = (n3, n2, n1)$. When navigating through a list of semantic nodes, the supervision system will choose a sub-map represented as ($previous\_node\_in\_plan$, $current\_node\_in\_plan$, $next\_node\_in\_plan$).

Special care must be taken in two situations: when the robot is starting a plan (i.e. there is no previous node in the plan) and when the plan contains less than three nodes. In this situation we analyze the semantic map, and look for a node n_prev where $n\_prev \in edges(current\_node\_in\_plan)$ and $n\_prev! = current\_node\_in\_plan$. If n_prev exists we select the sub-map represented by $(n\_prev, current\_node\_in\_plan, next\_node\_in\_plan)$. If not, we look for a node n_succ where $n\_succ \in edges(next\_node\_in\_plan)$ and $n\_succ! = current\_node\_in\_plan$. The function $edges(n)$ represents the list of nodes linked to node n in the semantic map.

For example, the following sub-maps would be generated in the example from Fig. 1:

- Room A - Corridor A - Gate A
- Corridor A- Gate A - Corridor B
- Gate A - Corridor B -Schengen Barrier
- Corridor B - Schengen Barrier - Room A
- Room A - Schengen Barrier - Corridor C
- Schengen Barrier - Corridor C - Gate C
- Schengen BArrier - Corridor B - Gate B
- Gate A - Corridor B - Gate B
- Corridor B - SChengen Barrier - Corridor C

During the example plan in the figure, the robot will choose the following maps:

1. Corridor B - Gate A - Corridor A
2. Gate A - Corridor A -Room A
3. Corridor A - Room A - Schengen Barrier

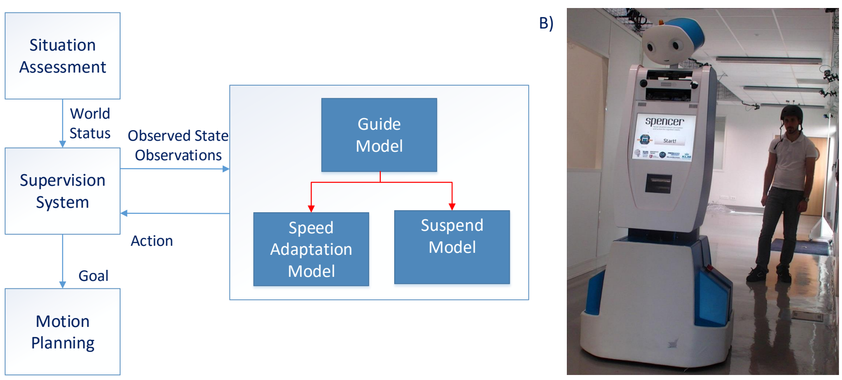
The necessity of switching maps was already identified in year two. At that moment we implemented a first map switching mechanism, based only on sets of two linked sub maps. This choice presented some issues, since the robot could need to go back at switching a map, because the way ahead is blocked, and so we decided, for the final version, to switch to the mechanism described in this paragraph.

5.0.1 Guiding users

Guiding a group is a complex problem, since the robot needs to adapt to its followers needs, which can be ambiguous. To solve this problem, we created a separate planning framework, that we called Collaboration Planners, based on hierarchical MOMDPs (Mixed Observability Markov Decision Process). A MOMDP models the decision process of an agent in situations where the result of an action is partly random, and can lead to several outcomes. In addition, in a MOMDP, the system state is split in an observable set and a hidden set, which cannot be fully observed and must be inferred from observations received from the environment. MOMDPs are a variation of POMDPs (Partially Observable Markov Decision Process), where the system state is completely hidden. Partitioning the system state in a hidden and observable set simplifies the computation of a solution to the model, which is one of the main problems of POMDPs [3].

We use a hierarchical framework [5], where the system model is split into a main MOMDP module and several MOMDP sub-models, each one related to a different action. The models are solved separately, leading to the computation of different, simpler, policy functions. At run-time, the system interacts with the main module, providing values for the set of observations and for the observed variables, and receiving an action as result. Based on this action, the system will contact a different sub-model, receiving the final action to execute. Using hierarchical MOMDPs we can represent a set of models, with a greatly reduced complexity, and easily expand it if we want to implement new actions or to add more complex behaviors. The architecture of our system is shown in Figure 2 A).

Figure 2: A) System Architecture: the main modules of this layer. Situation Assessment reasons on perceptual data, providing symbolic information to the Supervision System. The Supervision System controls the other modules, updating the Collaborative Planners, which compute the next action to perform, and sending goals to the Motion Planning. Blue arrows represent data links between modules, while red arrows represent conceptual links that show the hierarchy of the MOMDPs. B) The robot guiding a user.



When guiding a group, the robot robot should choose if it should proceed guiding, suspend temporarily the task, or abandon it. The Guide Planner is the main MOMDP of our architecture and will make this decision, based on two main variables: the status of advancement of the task (*not_completed*, *completed*), and the quality of commitment of the user (*not_engaged*, *engaged*, *not_interested*). The quality of commitment of the user is an hidden variable, estimated using Situation Assessment, based on the distance of the person toward the robot, its variation, if the user is oriented toward the robot, and if he is moving or still. The robot will abandon the task when it evaluates that it's user has permanently stopped following it.

The system is fully explained in [4]

In our early prototype, the system only presented a single MOMDP used to guide users. For the final version, we added the concept of hierarchical MOMDPS, with the sub-MOMDPS providing better human-aware behaviors.

5.0.2 Adapting the Robot's Speed

We believe that to be socially acceptable, the robot should adapt its speed to the user. By setting its own pace at the start of the scenario the robot would risk of being too slow, annoying the user, or too fast, which would lead the robot to constantly stop to wait for persons, producing an awkward behavior.

The robot defines a desired interval of distances r from the user. The distance of the user from r will influence its actions. 1) If the user is farther than r the robot will *decelerate*. 2) If he is closer to the robot than r, the robot will $accelerate$. 3) If the user is inside r, the robot will continue at its pace.

The robot should also not constantly change speed, in order to give time to users to adapt to its new chosen speed, and so we defined a temporal threshold in which we don't allow the robot to repeat an *accelerate* or *decelerate* action.

In this scenario we also studied the idea that the robot can try to influence the speed of the user. We studied two situations in which this idea can be useful. A) There is a time limit to reach the destination. In this case the robot must balance the desire to satisfy the user with the task urgency. Different situations will require different policies. For example, in an airport scenario, the robot could prioritize arriving on time, warning users if their speed would render the goal not achievable, while in other situations the robot could try to arrive in time but still avoid to adopt speeds that are uncomfortable for the follower. B) The rules of the current environment limit the robot's speed. In this case the robot will avoid accelerating over a set speed even if it detects that its current velocity is considered too slow for the user. For example, the robot could be navigating in a construction zone.

This reasoning is done in the Speed Adaptation MOMDP module, which will be interpreted when the Guide Model chooses to keep guiding the user.

5.0.3 Suspending the task

In some situation, the robot needs to suspend the task, because the user has stopped following it. In this case, the robot should estimate if this suspension of the collaborative scenario is temporary or permanent, and in the latter case abandon the task. We estimate this information using the Suspend Model and the activity areas from Situation Assessment. We link activity areas to the maximum time we expect that the user will be involved in the activity, and with a set of proactive actions that the robot can choose to execute.

In this paper, we investigated a single possible proactive behavior: giving information. In this case, if we detect that the user has stopped following because he is looking at a touristic sight, or at an information screen, the robot can try to engage him and offer related information. At the moment, we just propose a simple routine-based framework for this behavior, and plan to further study it in the future. We believe that the solution of this problem could be rich, and that the robot should estimate the reaction of the user during the execution of its proactive behavior, in order to be able to interrupt if he doesn't want to be helped or to resume the original task if he is satisfied by the robot's actions.

We don't want the robot to be stuck for a long time waiting for a person. If there is a small amount of time to reach the destination, or the user is engaged in the activity for a longer period of time than the one predicted, the Suspend Model can issue a warning action, and eventually abandon the task if the person doesn't start following it again. Sometimes users will stop following without an apparent reason, perhaps outside any activity area. In this case the robot will still allow them some time before issuing a warning and eventually abandoning the task.

References

[1] D5.5: Task planner and robot supervision system early prototype.

[2] Spencer second year report.

[3] Araya-López, M., Thomas, V., Buffet, O., and Charpillet, F. A closer look at momdps. In *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on* (2010), vol. 2, IEEE, pp. 197–204.

[4] Fiore, M., Khambhaita, H., Milliez, G., and Alami, R. An adaptive and proactive human-aware robot guide. In *Social Robotics*. Springer, 2015, pp. 194–203.

[5] Pineau, J., Roy, N., and Thrun, S. A hierarchical approach to pomdp planning and execution. In *Workshop on hierarchy and memory in reinforcement learning (ICML)* (2001), vol. 65, p. 51.

An Adaptive and Proactive Human-Aware Robot Guide

Michelangelo Fiore, Harmish Khambhaita, Grégoire Milliez, and Rachid Alami

Université de Toulouse, LAAS-CNRS institut Carnot, 7 rue du Colonel Roche, 31500, Toulouse, France,
{mfiore,harmish,gmilliez,rachid.alami}@laas.fr

Abstract. In this paper we present a robotic system able to guide a person to a destination in a socially acceptable way. Our robot is able to estimate if the user is still actively following and react accordingly. This is achieved by stopping and waiting for the user or by changing the robot's speed to adapt to his needs. We also investigate how the robot can influence a person's behavior by changing its speed, to account for the urgency of the current task or for environmental stimulus, and by interacting with him when he stops following it. We base the planning model on *Hierarchical Mixed Observability Markov Decision Processes* to decompose the task in smaller subsets, simplifying the computation of a solution. Experimental results suggest the efficacy of our model.

1 Introduction

One interesting problem in human-robot interaction is developing robots able to guide humans, by offering a tour of attractions in an area or simply by helping them to reach a destination.

A generic mobile robot platform should possess a vast set of skills, which includes advanced perception, motion planning, and task planning. These skills are not enough for a robot guide, which is deployed in highly dynamic human environments, and need to be complemented with human-aware behaviors.

Different robot guides have been studied and developed, starting with pioneers like Rhino and Minerva [22]. After these first experiments, several researchers have tried to focus on the social aspects of the problem, which are especially important if the robot needs to offer information. Studies like [23, 5] focus on how the robot should address humans, concentrating on spatial relationships and on how the robot can convey information. Few systems have actually been deployed for long period of time in human environments. Rackhman [4], a museum guide with human-aware behaviors, is an example of such system, and has been deployed in a science museum for several months. Robotic systems must be able to reason on sensor data in order to provide information to the decision layers. In [15], we presented our framework *SPARK*, which is able to maintain a topological description of the world state and to reason on humans' mental states, in order to improve the robot's social behavior. For guiding situations, [9] presents an assessment of human-robot interaction during an exhibition, where perceptual and task related data are used to compute an internal state according to the scenario. With these information, the robot can compute a new emotional state and interact accordingly with users.

Recently there has been emphasis on robot navigation algorithms that explicitly reason about human beings in the environment differently from other static or dynamic obstacles. Starting from *Proxemics*, researchers have investigated explicit social signals based on human-posture and affordance of the environment to improve the legibility of robot motion. For detailed discussion on human-aware navigation algorithms we refer the readers to [13, 18]. Human-aware navigation in a museum situation was studied in [19], where the authors build environmental maps, which include information learnt from human trajectories and postures, in order to plan safe paths that don't disturb humans present in the area.

We consider guiding as a *joint action*, a task where several participants cooperate to reach a common goal [2]. A joint action can be seen as a contract between its participants, that need to fulfill their part of the contract and to continuously monitor the other participants in order to understand what they are doing. Some robotic architectures, such as [6, 7], implement joint actions, explicitly modeling human agents in the system.

Participants in a joint action form a kind of mental representation of the task, which includes the actions that should be performed by every agent [20]. This mechanism can be used to predict what other agents will do, but also to understand when they are deviating from the shared plan. The idea of predicting the will of another agent is linked to the concept of intention, studied in psychology and philosophy literature, such as [3]. This topic is of particular interest in human-robot interaction and has been studied in different kind of scenarios, like [10, 11], or [17], which is related to a museum scenario.

We believe that most robot guide systems are focusing on the social aspects of the problem, and on human-aware navigation, without fully considering the fundamental aspects of joint actions. Guiding is a collaborative task, where the robot doesn't need only to reach a destination, but also to ensure that its follower reaches it, while providing a socially acceptable experience to him. In order to achieve this goal, the robot needs to constantly monitor its user, to adapt to his behaviors and to be ready to proactively help him.

In this paper, we present a robot guide which is able to lead a single person to a destination. More particularly, the originality of our approach is that the robot is able to show both an adaptive and a proactive behavior. The robot will try, while guiding, to select a speed that pleases its user, when adapting, or to propose a new speed, using environmental and task related stimulus. Finally, our system will proactively try to engage a user if it detects he need assistance.

We implement these ideas using a Situation Assessment component, which gathers data from different sources and provides symbolic information, a Supervision System, that controls the other modules, and a planning framework based on hierarchical MOMPDs (Mixed Observability Markov Decision Processes). Finally, a human-aware Motion Planning component allows the robot to navigate populated environments.

2 Situation Assessment

Having data from sensors is not enough, for the robot, to choose which actions to execute. To fill the gap between perception and decision, we use a Situation Assessment component. This module is able to gather data from sensors in input, and to perform different kinds of computations in order to produce symbolic information that can be used by the decision layer.

Our system is able to reason on humans and objects present in the environment, producing different kind of information, such as: a) the distance and orientation of a human relative to the robot, b) the variation of the distance from a human to the robot c) if a human is currently moving.

To be relevant, reasoning on humans should be linked to the environment. The system is able to create activity areas in the environment and link them to different kind of computations. An activity area is a polygonal or circular area, which can be fixed or linked and updated with an entity's (object, human or robot) position. For now, we studied and experimented two different activity areas: a) Information Screen Area, linked to information screens present in the environment; b) Touristic Point Area, linked to interesting attractions in the environment. Using these areas, the system can detect human activities (e.g. human is looking at an information screen, human is looking at an attraction).

Detecting and tracking persons is complex. In this paper, human tracking is done using motion capture. In order to simulate realistic behaviors, we filter data provided by the motion capture in order to account for occlusions from the environment. The system has also been linked in the european project SPENCER [1] to a laser based human tracking component.

3 Planning and Supervision

With the reasoning abilities provided by Situation Assessment, the robot should guide its user toward the goal, which could be predefined or negotiated with him at the start of the scenario. We defined a set of modules, called Collaborative Planners, able to choose which proactive or adaptive actions the robot should perform at each moment.

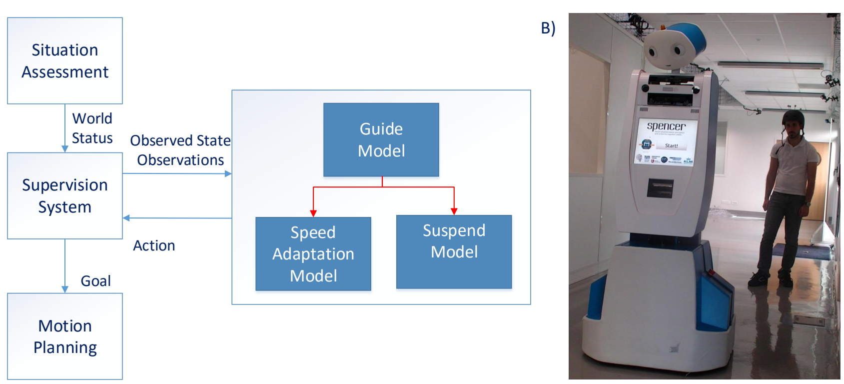
Collaborative Planners The Collaborative Planners form a planning framework, based on hierarchical MOMDPs (Mixed Observability Markov Decision Process) , that enables the system to react in a human-aware way to a user's behaviors. A MOMDP models the decision process of an agent in situations where the result of an action is partly random, and can lead to several outcomes. In addition, in a MOMDP, the system state is split in an observable set and a hidden set, which cannot be fully observed and must be inferred from observations

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

received from the environment. MOMDPs are a variation of POMDPs (Partially Observable Markov Decision Process), where the system state is completely hidden. Partitioning the system state in a hidden and observable set simplifies the computation of a solution to the model, which is one of the main problems of POMDPs [1].

We use a hierarchical framework [16], where the system model is split into a main MOMDP module and several MOMDP sub-models, each one related to a different action. The models are solved separately, leading to the computation of different, simpler, policy functions. At run-time, the system interacts with the main module, providing values for the set of observations and for the observed variables, and receiving an action as result. Based on this action, the system will contact a different sub-model, receiving the final action to execute. Using hierarchical MOMDPs we can represent a set of models, with a greatly reduced complexity, and easily expand it if we want to implement new actions or to add more complex behaviors. The architecture of our system is shown in Figure 1 A).

Fig. 1. A) System Architecture: our system is composed by four main modules. Situation Assessment reasons on perceptual data, providing symbolic information to the Supervision System. The Supervision System controls the other modules, updating the Collaborative Planners, which compute the next action to perform, and sending goals to the Motion Planning. Blue arrows represent data links between modules, while red arrows represent conceptual links that show the hierarchy of the MOMDPs. B) The robot guiding a user.



Guiding users The main problem of the robot is choosing if it should still guide the user, suspend temporarily the task, or abandon it. The Guide Planner is the main MOMDP of our architecture and will make this decision, based on two main variables: the status of advancement of the task (*not_completed*, *completed*), and the quality of commitment of the user (*not_engaged*, *engaged*, *not_interested*). The quality of commitment of the user is an hidden variable, estimated using Situation Assessment, based on the distance of the person toward the robot, its variation, if the user is oriented toward the robot, and if he is moving or still.

The robot will abandon the task when it evaluates that it's user has permanently stopped following it.

Adapting the Robot's Speed We believe that to be socially acceptable, the robot should adapt its speed to the user. By setting its own pace at the start of the scenario the robot would risk of being too slow, annoying the user, or too fast, which would lead the robot to constantly stop to wait for persons, producing an awkward behavior.

The robot defines a desired interval of distances r from the user. The distance of the user from r will influence its actions. 1) If the user is farther than r the robot will *decelerate*. 2) If he is closer to the robot than r, the robot will *accelerate*. 3) If the user is inside r, the robot will continue at its pace.

In this paper, r was a predefined set, but its values could be learnt and adapted to users during the task, since different people could prefer following the robot at different distances and positions. The robot should also not constantly change speed, in order to give time to users to adapt to its new chosen speed, and so we defined a temporal threshold in which we don't allow the robot to repeat an *accelerate* or *decelerate* action.

In this scenario we also studied the idea that the robot can try to influence the speed of the user. We studied two situations in which this idea can be useful. A) There is a time limit to reach the destination. In this case the robot must balance the desire to satisfy the user with the task urgency. Different situations will require different policies. For example, in an airport scenario, the robot could prioritize arriving on time, warning users if their speed would render the goal not achievable, while in other situations the robot could try to arrive in time but still avoid to adopt speeds that are uncomfortable for the follower. B) The rules of the current environment limit the robot's speed. In this case the robot will avoid accelerating over a set speed even if it detects that its current velocity is considered too slow for the user. For example, the robot could be navigating in a construction zone.

This reasoning is done in the Speed Adaptation MOMDP module, which will be interpreted when the Guide Model chooses to keep guiding the user.

Suspending the task In some situation, the robot needs to suspend the task, because the user has stopped following it. In this case, the robot should estimate if this suspension of the collaborative scenario is temporary or permanent, and in the latter case abandon the task. We estimate this information using the Suspend Model and the activity areas from Situation Assessment. We link activity areas to the maximum time we expect that the user will be involved in the activity, and with a set of proactive actions that the robot can choose to execute.

In this paper, we investigated a single possible proactive behavior: giving information. In this case, if we detect that the user has stopped following because he is looking at a touristic sight, or at an information screen, the robot can try to engage him and offer related information. At the moment, we just propose a simple routine-based framework for this behavior, and plan to further study it in the future. We believe that the solution of this problem could be rich, and that the robot should estimate the reaction of the user during the execution of its proactive behavior, in order to be able to interrupt if he doesn't want to be helped or to resume the original task if he is satisfied by the robot's actions.

We don't want the robot to be stuck for a long time waiting for a person. If there is a small amount of time to reach the destination, or the user is engaged in the activity for a longer period of time than the one predicted, the Suspend Model can issue a warning action, and eventually abandon the task if the person doesn't start following it again. Sometimes users will stop following without an apparent reason, perhaps outside any activity area. In this case the robot will still allow them some time before issuing a warning and eventually abandoning the task.

4 Motion Planning

A guiding robot needs to plan safe and socially acceptable motion. This requires continual integration of high-level social constraints with the low-level constraints of the robot vehicle.

We use the architecture proposed by the well-established `move_base` package of `ROS` middle-ware [14, 8] for navigation, replacing the global planner, i.e. a cost-grid base path planning module, as suggested in [21]. This module adds proxemics based costs in the grid-map around the detected humans that are static in the environment. The local planner, i.e. the module responsible for generating motor commands, is a *ContextCost* based algorithm suggested in [12]. This module continuously calculates the *compatibility* of the robot path by predicting and avoiding future collisions with moving persons and simultaneously keeping the robot as close as possible on the planned global path.

It should be noted that in our guiding experiments, the humans are mostly moving behind the robot and therefore the situation remains compatible for the local planner most of the time. During compatible situations the robot simply follows the way-points on the planned global path. The nominal velocity of the robot is set by the supervision system to achieve the desired behavior of slowing-down or speeding-up, as required by the situation.

5 Experiments and Analysis

We performed a first set of experiments with a person following a robot on a predefined path, in order to test the behaviors of the robot. Data from these experiments are shown in Table 1 [2]. We start by showing speed adaptation tests:

- adapting slow and fast: in these two tests (Figure 2) we used our system to guide respectively a user that would like to move at a slow pace, and a user that would like to move at a fast speed.
- no adaptation: in this experiments the robot won't adapt to the speed of the user, setting its own pace and stopping if it is too far.

[2] Videos from our experiments can be seen at http://homepages.laas.fr/mfiore/icsr2015.html

Looking at the data we can see that our system shows lower values for the variance of speed and distance, which means that after a certain time it's able to find a condition of equilibrium with the human follower. The 'no adaptation' system shows a significantly higher variance for both values, since the robot stopped several times to wait for the user. We will now show some tests regarding the proactive behaviors of the robot:

- proactive slow and fast: during the task, the robot proactively chooses to change pace, in the first case by slowing down and in the second by accelerating. In our tests the user adapted after some seconds to the robot's pace, but this behaviors should be studied in-depth in user studies.
- suspend with screen and with no reason: in these tests we asked a user to stop during the task. In the first case the user stopped near an information screen. After detecting this event, the robot approached the user to offer information, which lead to the resumption of the task. In the second case the user stopped at a different point of the path. The robot wasn't able to detect the reason for the suspension of the task and so simply issued a warning to the user and abandoned the task after some seconds.

Table 1. Experiment results: d is the distance between the robot and the user, s_r is the robot's speed, s_h is the human's speed, μ is the average and Δ is the variation of the quantity over the whole test. Distances are expressed in meters, velocities in meters for seconds.

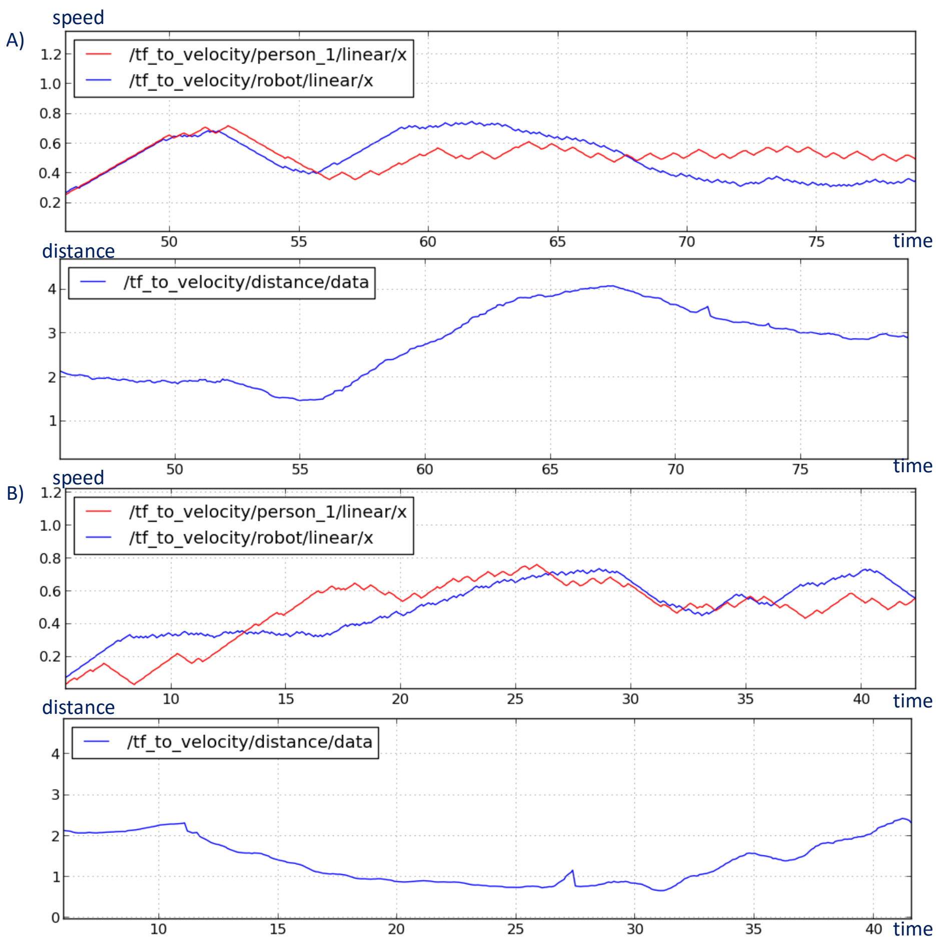
test name	μ $distance$	μ $speed\ difference$	Δ $distance$	Δ $speed\ difference$
adapting slow	2.82	-0.03	0.64	0.02
adapting fast	1.38	0.00	0.29	0.01
no adaptation	3.08	-0.09	1.04	0.07
proactive slow	1.45	-0.06	0.04	0.10
proactive fast	2.66	-0.11	0.63	0.01

6 Conclusions

In this paper we introduced a robotic system able to guide, in a human-aware manner, an user to a destination. Our system is able to estimate, using Situation Assessment and a set of planners based on hierarchical MOMDPs, if the human user is currently engaged in the task, to adapt its actions to his behaviors, and to proactively help him. Through a set of experiments we showed that the robot is able to adapt its speed to its follower, in order to provide a socially acceptable behavior. We also began to study how the system can influence its user, by proposing a new speed, based on environmental and task related stimulus, and by proactively interacting with him when he stop following.

Though not shown in this paper, our system is also able to represent and guide groups of users, by reasoning both on the group as a single entity (represented through its spatial centroid) and on its single members. We plan, in the future, to perform user studies on groups, to understand how they react to the robot's behaviors, and eventually modify the system.

Fig. 2. Experiments: a) Adapting robot speed to a slow user. The first figure shows the speed of the user $(tf\_to\_velocity/person\_1/linear/x)$ and of the robot $(tf\_to\_velocity/robot/linear/x)$, and the second their distance. The robot starts slowing down at $t = 60$, when the distance from the user is growing, until it finds an equilibrium with the user's speed. Notice that there is a turn in the path, at $T = 50$, that causes the robot and the user to slow down. Distances are expressed in meters, velocities in meters for seconds. b) Adapting robot speed to a fast user. As before, the figures show the robot and user's speed and their distance. The robot starts accelerating at $t = 15$ when the distance from the user becomes small.



We would also like to study learning techniques, both in motion planning and in supervision, and integrate them in the system, to adapt even more the robot's movement and its decision on specific followers.

References

1. Araya-López, M., Thomas, V., Buffet, O., Charpillet, F.: A closer look at momdps. In: Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on. vol. 2, pp. 197–204. IEEE (2010)
2. Bratman, Michael, E.: Shared agency. Philosophy of the social sciences: philosophical theory and scientific practice pp. 41–59 (2009)
3. Bratman, M.: Two faces of intention. Philosophical Review 93, 375 – 405 (1984)
4. Clodic, A., Fleury, S., Alami, R., Chatila, R., Bailly, G., Brethes, L., Cottret, M., Danes, P., Dollat, X., Elisei, F., et al.: Rackham: An interactive robot-guide. In: Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on. pp. 502–509. IEEE (2006)
5. Evers, V., Menezes, N., Merino, L., Gavrila, D., Nabais, F., Pantic, M., Alvito, P.: The development and real-world application of frog, the fun robotic outdoor guide. In: Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing. pp. 281–284. ACM (2014)
6. Fiore, M., Clodic, A., Alami, R.: On planning and task achievement modalities for human-robot collaboration. In: The 2014 International Symposium on Experimental Robotics (2014)
7. Fong, T.W., Kunz, C., Hiatt, L., Bugajska, M.: The Human-Robot Interaction Operating System. In: 2006 Human-Robot Interaction Conference. ACM (March 2006)
8. Garage, W.: The robot operating system (ros) is a set of software libraries and tools that help you build robot applications. http://www.ros.org/
9. Jensen, B., Tomatis, N., Mayor, L., Drygajlo, A., Siegwart, R.: Robots meet humans-interaction in public spaces. Industrial Electronics, IEEE Transactions on 52(6), 1530–1546 (Dec 2005)
10. Karami, A.B., Jeanpierre, L., Mouaddib, A.I.: Human-robot collaboration for a shared mission. In: Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on. pp. 155–156. IEEE (2010)
11. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. Robotics: Science and Systems, Berlin (2013)
12. Kruse, T., Basili, P., Glasauer, S., Kirsch, A.: Legible robot navigation in the proximity of moving humans. In: Advanced Robotics and its Social Impacts (ARSO), 2012 IEEE Workshop on. pp. 83–88 (May 2012)
13. Kruse, T., Pandey, A.K., Alami, R., Kirsch, A.: Human-aware robot navigation: A survey. Robotics and Autonomous Systems 61(12), 1726–1743 (2013)
14. Marder-Eppstein, E.: move_base, a ROS Package that lets you move a robot to desired positions using the navigation stack. http://wiki.ros.org/move_base
15. Milliez, G., Warnier, M., Clodic, A., Alami, R.: A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management. In: Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on. pp. 1103–1109 (Aug 2014)
16. Pineau, J., Roy, N., Thrun, S.: A hierarchical approach to pomdp planning and execution. In: Workshop on hierarchy and memory in reinforcement learning (ICML). vol. 65, p. 51 (2001)
17. Rashed, M.G., Suzuki, R., Lam, A., Kobayashi, Y., Kuno, Y.: Toward museum guide robots proactively initiating interaction with humans. In: Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts. pp. 1–2. ACM (2015)

18. Rios-Martinez, J., Spalanzani, A., Laugier, C.: From Proxemics Theory to Socially-Aware Navigation: A Survey. International Journal of Social Robotics 7(2), 137–153 (Sep 2014)
19. Samejima, I., Nihei, Y., Hatao, N., Kagami, S., Mizoguchi, H., Takemura, H., Osaki, A.: Building environmental maps of human activity for a mobile service robot at the miraikan museum. In: Field and Service Robotics. pp. 409–422. Springer (2015)
20. Sebanz, N., Bekkering, H., Knoblich, G.: Joint action: bodies and minds moving together. Trends in cognitive sciences 10(2), 70–76 (2006)
21. Sisbot, E., Marin-Urias, L., Alami, R., Simeon, T.: A human aware mobile robot motion planner. Robotics, IEEE Transactions on 23(5), 874–883 (Oct 2007)
22. Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Haehnel, D., Rosenberg, C., Roy, N., et al.: Probabilistic algorithms and the interactive museum tour-guide robot minerva. The International Journal of Robotics Research 19(11), 972–999 (2000)
23. Yousuf, M.A., Kobayashi, Y., Kuno, Y., Yamazaki, A., Yamazaki, K.: Development of a mobile museum guide robot that can configure spatial formation with visitors. In: Intelligent Computing Technology, pp. 423–432. Springer (2012)