Actas ROBOT 2011. 28-29 de Noviembre de 2011. Sevilla (España)

Robot Companions for Guiding People in Urban Areas

Anaís Garrell, Andreu Corominas Murtra and Alberto Sanfeliu

***Abstract*— In this paper we explain some developments on robot guidance of people and how we manage the robust navigation in urban areas. This work is part of the research of ongoing EU and national research projects (URUS, Rob-TaskCoop, CONET). We describe the Discrete Time Model for people guidance by robots, how we optimize the tasks of the robots for doing the guidance mission and how we have verified the model comparing the model results against the ground truth of people and robot motions. In this verification we have assumed that people are not afraid by the robot motions. We also describe the robust navigation method that is used by the robots that perform the guiding mission showing some examples of navigation in the Barcelona Robot Lab, an outdoor lab of 10.000m *sq m*.**

I. INTRODUCTION

Robot Companions is a new concept which can be seen as an ecology of sentient machines that will help and assist humans in the broadest possible sense to support and sustain our welfare. Robot Companions must be cognizant and aware of their physical and social world and respond accordingly. One of the areas where they can assist people is in urban sites, where a great number of scientific, technological, security, ethical, etc. issues must be solved. From the technological point of view, reducing the areas of free circulation of cars in the cities will imply a revolution in the planning of urban settings, imposing new challenges for transportation of goods to the stores, security issues, human assistance, etc. Several research projects, the Network Robot project in Japan, the URUS project in Europe [17] and the DustBot project also in Europe [14] have started to search for answers to these challenges from the technological point of view. They have analyzed the issue with the idea of incorporating robots, sensors and ubiquitous communication devices in the urban areas.

In one of these projects, the URUS project [17], [18], the general objective was to analyze and develop techniques to enable a networked robot to cooperatively interact with humans and the environment on guidance and assistance tasks, transportation of goods, and surveillance in urban areas. Specifically, we designed and developed a cognitive networked robot architecture that integrates cooperating urban robots, intelligent sensors (video cameras, acoustic sensors, etc.), intelligent devices (PDA, mobile telephones, etc.) and a communications network. Some technological challenges addressed included: design and development of the cognitive networked robot architecture; navigation and motion coordination among robots; cooperative environment perception; cooperative map building and updating; task scheduling and negotiation within cooperative systems; human robot interaction; and wireless communication strategies between users (mobile phones, PDAs), the environment (cameras, acoustic sensors, etc.), and the robots. The project was tested in the Barcelona Robot Lab, a 10,000 *sq m* area located in the Campus Nord of the Universitat Politècnica de Catalunya, equipped with video cameras and Gigabit Ethernet connection, and in the Passeig de Sant Joant street in Barcelona, transporting and guiding people from one place to another.

In this article we touch upon two technical issues which were especially relevant during these outdoor urban mobile robot experiments: robot services for people guidance and robust navigation. People guidance by robots is a key issue in the relation between humans and robots, which is commonly performed when people is walking with other people or for typical tourism of business tours. Robust navigation is of paramount importance for robots that share the sidewalks with pedestrians. Safety is the number one concern, and our methods for navigation and path planning have as number one priorities like human safety and robot reliability. We discuss in this paper the techniques used to model the guidance of people by robots denominated the Discrete Time Model, how we optimize the tasks of the robots for doing the guidance mission and how we have verified the model comparing the model results against the ground truth of people and robot motions. In this verification we have assumed that the people are not afraid by the robot motions. We also describe the robust and safety navigation method that is used by the robots that perform the guiding mission showing some examples of navigation in the Barcelona Robot Lab.

II. REPRESENTATION OF THE GUIDANCE MODEL

In the following section, we will present the modelization we use in the guidance mission using several robots and working with a group of people [6]. We use a group of robots working in a cooperative way, one as a tour guide (the leader robot) and the other one, as a shepherd robot. The mission of the leader robot is to guide a group of people from an origin to a destination. The other robot is used as an assistant

The authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, 08028 Barcelona, Spain. {agarrell, acorominas, sanfeliu} @iri.upc.edu.

This work was partially supported by CICYT projects RobTaskCoop (DPI2010-17112) and MIPRCV Consolider Ingenio (CSD2007-018); and by the projects Cooperating Objects Network of Excellence FP7-ICT-2007-2-224053 and CEEDS (FP7-ICT-2009-5-95682) of the European Community

based on shepherd dog theory [4], [13] and its objective is to regroup people who move away from the the crowd formation. The strategy followed is, firstly, the computation of the estimate people's velocity with a particle filter [1], [2], and secondly, the computation of the optimal path from the shepherd robot to the estimated position of people that are moving away.

The key element is the "Discrete Time Motion" (DTM), whose goal is to estimate at each time instance the position and velocity of every person, as well as to predict their future states. The DTM evaluates these data in discrete time instances, every N units of time (seconds or milliseconds).

The DTM model has two components: The Discrete Time component and the Discrete Motion component. The former estimates position, orientation and velocity of the robots and persons, and the position of the obstacles at a time instance k. It will be used to estimate the intersection of the people with the obstacles and detect if someone is leaving the group. The Discrete Motion component estimates the change of position, orientation and velocity of people and robots between two time instances k and $k+p$. It will be used to compute the robots' trajectory to reach the goal while preventing people leaving the group.

A. The Discrete Time Component

The first task of the Discrete Time component is to estimate position, orientation and velocity of the robots and people who are being guided. This is done with a standard particle filter formulation. Then, the Discrete Time component aims to represent the areas where the robots will be allowed to move, by means of potential fields.

In order to decide the trajectories the robots will follow we will define a potential field over the working area, and perform path planning in it [12]. To this end we will define a set of attractive and repulsive forces. In particular, the goal the robots try to reach will generate an attractive force pulling the robots towards it. On the other hand, the obstacles will generate a repulsive potential pushing a given robot away. The rest of robots and persons will generate similar repulsive forces, although with less intensity than the obstacle's forces.

We parameterized all these attractive and repulsive forces by Gaussian functions. For instance, the repulsive forces for people will be:

$$T_p(\mu_p, \Sigma_p)(x) = \frac{1}{|\Sigma_p|^{1/2}(2\pi)^{n/2}} e^{-\frac{1}{2}(x-\mu_p)^T \Sigma_p^{-1}(x-\mu_p)} \quad (1)$$

where $\mu_p = (\mu_{p_x}, \mu_{p_y})$ is the center of gravity of the person, and Σ_p is a covariance matrix whose principal axes (σ_x, σ_y) represent the size of an ellipse surrounding the person which is used as a security area. A similar expression defines the potential map associated to each robot.

These repulsive forces may be interpreted as continuous probability functions over the entire space. Once they are defined, the tensions at each point of the space may be computed as the intersection of these Gaussians.

We can then define people and robots by the set $\{(\mu_x, \mu_y), (\sigma_x, \sigma_y), v, \theta, T\}$, where v and θ are the velocity and orientation computed by the particle filter and T is the associated tension. As we said, the variances (σ_x, σ_y) represent the security area around each individual. This could be set to a constant value. However, for practical issues one may need larger security areas when the robots or persons move faster. As a consequence, we changed appropriately the values of the variances σ_x and σ_y depending on the velocity parameter v.

In the case of the obstacles, we define their tension as a set of Gaussian functions collocated at regular intervals around their boundaries. Let us denote by $X = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ the set of points evenly spaced around the boundary. Then this boundary will be defined by: $\{(x_i, y_i), (\sigma_{x_i}, \sigma_{y_i}), T_i\}$ for $i = 1, \ldots, n$, where T_i follows Equation 1.

After having defined the tensions for each of the components of the environment –i.e. robots, persons and obstacles– we are ready to define the potential field. This is easily computed as the intersection of all the Gaussian functions for a given variances.

Once the potential field is known, we will define the trajectories of the robots, based on the position of the persons and the goal and following the paths with minimum energy in the potential field. This will be explained in the following section.

B. Discrete Motion Component

The Discrete Motion Component will decide the motion strategies to be followed by the robots in order to achieve their goals, which are following a path to reach a specific position while preventing people from leaving the group. Therefore, we will consider two different motion strategies: (i) path planning till the goal, and (ii) shepherding strategies for avoiding people leaving the group.

In the first case the robot motion is computed using a simple path planning algorithm [11]. We first compute all the possible paths to reach the goal, i.e, the *roadmap*. Among all these paths, we then select the shortest one, and each node of this path will be considered as a subgoal. The robots will then move between consecutive subgoals avoiding people leaving the group. This path planning is only performed by the *leader* robot who transmits the computed path to the rest of the robots.

The second case is the study of shepherding algorithms [3], which are inspired in the shepherd dogs. The shepherding task is performed by all the robots except the *leader*, that only carries out the function of a guide. The method that we have developped to decide which is the behavior of the robots when one person o several people move away from the formation will be explained in the following section.

Figure 1 show a simulation in which a group of people is guided throught the Barcelona Robot Lab.

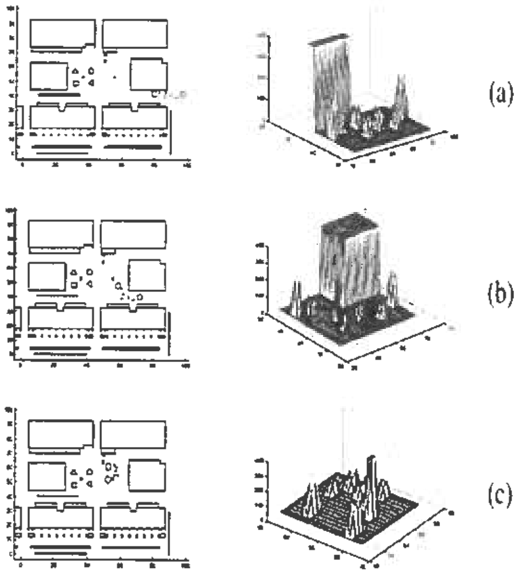

Fig. 1. Left: simulation of cooperative people guidance on the Barcelona Robot Lab. Right side: Values of the tensions values in specific instants of time.

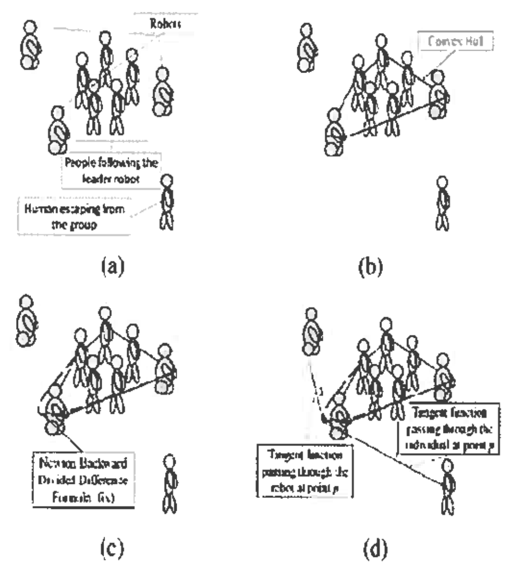

Fig. 2. (a) Environment representation with people and robots. (b) Computation of the convex hull. (c) Interpolation of the convex hull with Newton Backward Divided Difference Formula. (d) Computation of the trajectory for rescuing the individual, this trajectory is composed by two tangents of the function f(x) at point p: (1) passing through the shepherd robot (2) passing through the individual is escaping.

III. Local Optimization Function of Cooperative Robot Movements for Guiding Mission

One important situation we must carefully consider is the case when people move away from the group [5]. The main goal of the robots that work in the especific task of guiding a group of people is to accompany everyone to the goal. So, the possibility of loose people is not considered. We are not aware of any approach tackling this problem. The solution we adopt for this situation is to choose one of the robots and bring him back to the formation. For computing the trajectory that will be considered for intercepting the person who escaped from the group, we first used a Particle Filter to estimate the position and velocity of the person and compute the interception point. Once the interception point O_k at time k is obtained, it will become the next subgoal for the closer robot, called R_k. Then, the trajectory the robot must follow is computed as shown in Fig. 2: This trajectory is simply the line passing through the robot R_k and tangent to a 2 meters diameter circle centered at the interception point.

The main purpose of our work is to analyze which is the best strategy in the following situation: "Given a fixed number of robots (usually 2 or 3), assign robots' tasks that will minimize the work required by them, and, also, will produce the minimum displacement problems for guiding people". The cost function, described below, speaks in Work terms, and it can be divided into two blocks: *(i)* Robot work motion, and *(ii)* Human work motion.

Nonetheless, robots must be able to solve this task while they are navigating and avoiding obstacles and do not infer in people's living space. Furthermore, there are other situations that can happen, however they have not been considered in the present work, for instance, one robot is used as a barrier in a corner, in order that people do not miss the way.

The Robot tasks that we are considering are:

- *Leader task*: Leader robot computes a path planning and moves to the next point. We also assume that there exists a *drag force* that will attract people behind the robot. In case that a robot, that is not the leader, takes its role, this robot will have first to move still leader's present position and then carry out this task.
- *Looking for a person that goes away task*: The robot moves to the estimated position of the individual who goes away from the crowd formation. In this case, the robot has to compute all possible paths to reach the estimate position and then, take the one which minimize the itinerary.
- *Pushing task*: The robot pushes a person that has gone away in order to reach the crowd formation. This task can be also applied when a robot pushes a person (or a group of people) who is (are) going behind the crowd formation in order to regroup people when the formation

is broken down. We assume that there exists a repulsion force that pushes the person to follow the direction of the robot.

- *Crowd traversing task*: The robot has to move through the formation to achieve the estimated position of the person that goes away from the crowd formation. This task implies that the robot has to push people away from their path, which creates a set of repulsion forces from the robot to people. We do not consider this task due to security reasons.

In order to compute the dragging, pushing and crowd traversing forces, we use the equations defined in previous works on human behavior with other individuals [8], [9], [10]. People movements are determined by their desired speed and the goal they wish to reach. In our case, the direction of the person movement $\vec{e}_i(t)$ is given by:

$$\vec{e}_i(t) = \vec{e}_{robot}(t) + \vec{u}(t) \quad (2)$$

where $\vec{u}$ is the noise. Usually, people do not have a concrete goal and should follow the leader robot, thus, its direction is determined by the robot's movement or the individual that they have in front, if the robot is not in their visual field.

In following sections we will describe the different forces for the computation of the cost function.

A. Robot Work Motion

Working with autonomous mobile robots, the robot i work motion is expressed by:

$$f_i^{mot} = m_i a_i \quad (3)$$
$$W_i^{mot} = f_i^{mot} \Delta x_i \quad (4)$$

where m_i is the mass of the i-th robot, a_i its acceleration and Δx_i the space traversed by the robot to achieve its goal.

B. Human Work Motion

In Human Robot Interaction, it is necessary to consider the *dragging*, *pushing* and *crowd intrusion forces* that robot's motion produces and that can affect on people's behavior. This component is called *Human Work Motion*, and it is the expense of people's movements as a result of robot's motions. As it has been mentioned several times in this paper, the group follows the robot guide/leader, and there is a set of robots that help to achieve their goal. The effect of robots on people as forces works as follows: *(i)* leader robot: attractive (dragging) force, it is inversely proportional to the distance, until a certain distance. *(ii)* shepherding robot: Repulsive (pushing,traversing) force, has a repulsive effect inside people's living space.

1) Dragging Work: The dragging force is necessary when the leader robot guides the group of people from one place to another. It acts as an attractive force, hence the force applied by robot leader i to each person j is:

$$f_{ij}^{drag}(t) = -C_{ij}\vec{n_{ij}}(t) = -C_{ij}\frac{x_i(t)-x_j(t)}{d_{ij}(t)} \quad (5)$$
$$d_{ij}(t) = \|x_i(t)-x_j(t)\| \quad (6)$$

where $d_{ij}(t)$ is the normalizated vector pointing from person j to robot i at instant t. See [7] for more information about the parameter C_{ij} , which reflects the attraction coefficient over the individual j, and it depends on the distance between the robot leader and person j.

Thus, the dragging work that robot leader applied to each individual is defined by:

$$W_{drag} = \sum_{\forall \text{ person j}} f_{ij}^{drag} \Delta s_j \quad (7)$$

Where Δs_j is the distance traveled by the person j.

2) Pushing Work: The *Pushing force* is given by the repulsive effect developed by shepherding robot on the group of people, for regrouping a person (or the broken crowd) in the main crowd formation. This repulsive force is due by the intrusion of the robot in the people's living space, which is five feet around humans. The territorial effect may be described as a repulsive social force:

$$f_{ij}^{push} = A_i exp^{(r_{ij}-d_{ij})/B_i}\vec{n_{ij}}\left(\lambda_i + (1+\lambda_i)\frac{1+cos(\varphi_{ij})}{2}\right) \quad (8)$$

Where A_i is the interaction strength, $r_{ij} = r_i + r_j$ the sum of the radiis of robot i and person j, usually people has radii of one meter, and robots 1.5 m, B_i parameter of repulsive interaction, $d_{ij}(t) = \|x_i(t)-x_j(t)\|$ is the distance of the mass center of robot i and person j. Finally, with the choice $\lambda < 1$, the parameter reflects the situation in front of a pedestrian has a larger impact on his behavior than things happening behind. The angle $\varphi_{ij}(t)$ denotes the angle between the direction $\vec{e}_i(t)$ of motion and the direction $-\vec{n_{ij}}(t)$ of the object exerting the repulsive force. See [7].

So we can write pushing work by:

$$W_{push} = \sum_{\forall \text{ person in } \Omega_i} f_{ij}^{push}(t) \Delta s_j \quad (9)$$

Where Ω_i is the set of people in which one of the helper robots have reached the living space, if an individual is at certain distance from the robot, more than two meters, it is considered that the robot does not penetrate in his living space, and therefore is not affected by the drag force.

3) Traversing Work: And last but not least, the *Traversing force* is determined by the forces applied by the robot when is traversing the crowd. For security reasons, we have considered in this research that the value of this force is infinity, so we will ensure that a robot will not cross the crowd in order to avoid any damage on people.

C. Total Cost for One Robot

The cost function for robot i, given a specific task, is the following one:

$$W_i = \delta_{mot} W_i^{mot} + \delta_{drag} W_i^{drag} + \delta_{push} W_i^{push} + \delta_{trav} W_i^{trav} \quad (10)$$

$$\text{where } \delta_k = \begin{cases} 1 \text{ if this task is assigned} \\ 0 \text{ if this task is not assigned} \end{cases}$$

Where k could be *pushing, dragging, traversing* or *motion*. For each period of time, the leader and shepherded robots will be given a task in the guiding mission, which will imply one or several robot motion works and human robot works.

D. Optimal Robot Task Assignment

Finally, the task assignment for the robots will be the one which minimizes the minimum assigned work cost required to do the global task. It is computed by the following way:

$$C = argmin\{W_{total}(c)\}, \; \forall \text{ configuration c} \quad (11)$$

where the *Configurations* mean how the tasks are distributed among the robots, for each configuration c robots compute W_{total} which is the addition of all W_i for all robots i that are working cooperatively.

Once we have this cost function, we can determine which are the optimal trajectories the robots must follow to achieve their goal, and which are the roles for each robot.

In Fig.3 we show the evolution of the work performed by the group of robots while they guide a group of people. The two local maxima represents the instant of time when one person tries to move away from the formation.

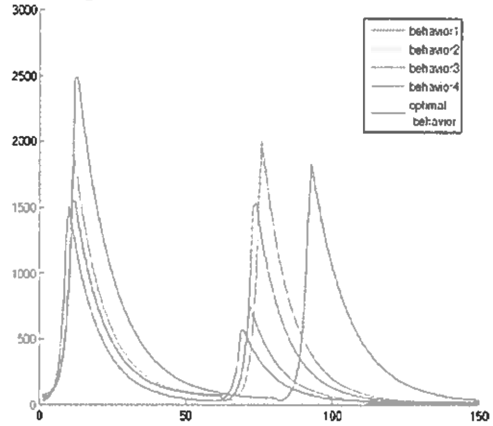

Fig. 3. Evolution of the cost function along time of different behaviors of robots when people are escaping in two different instants of time. Behavior 1: Robot Leader looks for people who are moving away. Behavior 2: Shepherd Robot look for people who are escaping without choosing the shortest way. Behavior 3: Shepherd Robots interchange their positions before looking for people who are escaping. Behavior 4: Shepherd robot which is nearest of people who are escaping is the responsible for resolving this mission without considering the forces presented before. Behavior 5: Robots choose the configuration which minimizes the cost function.

IV. Verification of the Guidance Model

Here, we present the validation of the DTM model described previously. The articles presented in the literature show their contributions for groups of robots that interact with people using simulations. However, realistic situations, such as the existence of obstacles or dealing with people leaving the group are not considered, and their models were not validated. In [13] several types of robot formations and different strategies for approaching the robots to people are considered. However, all these issues and the general movements of the robots are ruled by a large number of heuristics which makes the system impractical.

Before the experimentation of the DTM with real robots, it is necessary to validate the functionality of the model throught simulations. For the validation of this research, we used the network of cameras installed in the Barcelona Robot Lab, in the Universitat Politècnica de Catalunya (UPC), an area of 10.000 m^2 in the north Campus. We performed a set of experiments where a group of people were guided by three people playing the role of the guide robots. They have performed 9 experiments following the same path, but there were some differences in each experiment. The group made the same trajectory nine times, and in each one the group presented different behaviors, for instance, a person escaped from the group and must be regrouped, or some people escaped in different directions. These human behaviors were simulated and we studied how the behavior of the robots should be for achieving their objective, this is, lead the group towards the goal. In the present paper, we validate the DTM model by analyzing the position and velocity error differences of people and robots.

Model validation is used for verifying that the model, within its domain of applicability, behaves consistently with the objectives. Model validation deals with building the right model. In model testing, the model is subjected to test data or tests cases to determine if it functions properly. Test failed implies the failure of the model, not the test.

For data collection we use the camera network mounted on the Barcelona Robot Lab, composed by 21 interconnected cameras, see Fig. 4, which took a set of video sequences of the group of people being guided by three other people (who play the role of guide robots). The tour, was performed nine times and in each one of the trials, some people of the group behaved in different ways.

For the model validation process, the trajectory of every person has to be taken into account in the complete path, as well as the trajectory of all the group. This trajectories are then compared against the estimation obtained by the simulations using the DTM model. The comparison is based on the quadratic error of the real (from the video sequence) and estimated (from the simulation) positions.

For robots, several cases were analyzed. For the leader

Fig. 4. Images fo the set of cameras, a group of people is being guide in the Barcelona Robot Lab.

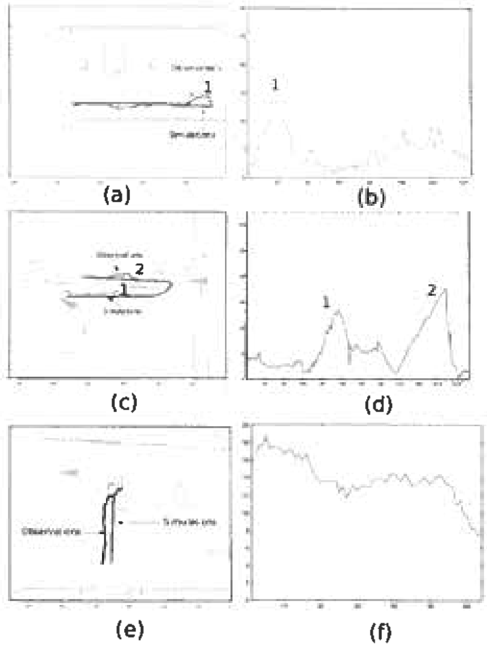

Fig. 5. Leader Robot behavior in three different scenes are shown, on left hand the comparison error between the simulated behavior and the observation data are represented. In (a) and (b) it must be considered that point 1 shows a great error due to the observation trajectory performs a curvilinear path while the simulated trajectory performs a rectilinear one, both trajectories are correct. In (c) and (d) occurs a similiar behavior, at points 1 and 2 different path produce a higher error, again, both trajectories can be considered right.

we examine the motion trajectory that it has followed for guiding the group and the velocity at each position. For the shepherd we compute the motion trajectory for following the group and the motion trajectory to look for a person that escapes from the formation and he has to be regrouped. For the guided people we compute the estimated trajectory using the particle filter against the observations obtained from the camera network.

All this comparison are done in the areas where the camera network have information of the position of people and robots. There are some areas where the cameras are not able to see people's trajectories due to environment constraints. In the areas where there are not real data, we only can estimate their positions, and the error will be computed at the next position obtained from the camera network. See Figure 5 to see the comparation of the trajectories performed by the simulated Leader and the real Leader.

V. Robust Navigation in Urban Areas

All robotic mobility services, be these people guidance, surveillance or transportation of goods, share a common basic module, that of robust autonomous navigation. This module is in charge of executing *go to* requests coming for task driven higher-level modules, or directly from an operator. The requests come typically in the form of 2D point coordinates with respect to a given reference map. The map comes either from a previously executed mapping session, or from a CAD model provided by city authorities. Either way, these maps are usually not complete models and robust navigation must deal with such lack of detail, as well as with other contingencies such as scene dynamics, non-modeled obstacles, or unreliable or imprecise GPS coverage.

We have built a map-based robust autonomous navigation system able to cope with these requirements for our two robots Tibi and Dabo. These robots are two-wheel self-balancing units with near holonomic motion capabilities. Dexterous mobility of self-balancing platforms comes at the cost of difficult terrain perception and control issues [19]. Their primary sensors for navigation are three laser scanners and the platform embedded wheel encoders, gyroscopes and inclinometers. Two laser scanners are mounted in the front and rear to scan a 360 deg fov horizontal plane at a height of 40 *cm* from the floor, and the third laser scanner is mounted at a height of 80 *cm* to scan a vertical plane in front of the robot.

Figure 6 shows a diagram with all processes involved in the autonomous navigation module. The approach is based on two loops that iterate concurrently at different rates: a reactive loop and a deliberative one. The reactive loop runs at 10 *Hz* and is in charge of safe local navigation. The process receives point goals expressed as local robot coordinates and computes platform velocity commands to direct the robot towards such goals while avoiding obstacles perceived by front horizontal and vertical lasers. The deliberative loop runs at a lower speed of 3 to 5 *Hz* and is in charge of

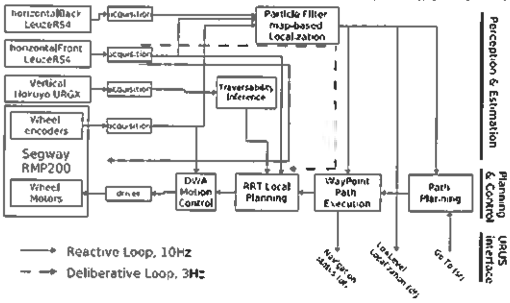

Fig. 6. Map-based autonomous navigation module. Reactive and deliberative loops run concurrently in real-time and at different rates.

global path planning on the map, and of translating global way points into local robot coordinates. This coordinate transformation is possible only thanks to a robust map-based Montecarlo localization process, that continuously updates the robot position estimate integrating all data available from laser scanners and platform embedded sensors.

Our first version of the full autonomous navigation module was based on a 2D model of the environment producing 3 degrees of freedom localization output estimates (x,y,θ). Using this approach, a series of experimental sessions were accomplished with more than $3Km$ of autonomous navigation in real outdoor pedestrian environments, resulting in a final success rate of about 79% navigation tasks completed [15]. Main failures can be attributed to localization failures because the 2D map was not rich enough in areas with high presence of 3D elements such as ramps, holes or steps. From these results, a second version of the navigation approach was developed, using a 3D geometric model of the environment and with a localization output estimate of the full 6 DOF robot pose (x,y,z,θ,ϕ,ψ). This new version allowed us to run a new series of experimental sessions in two outdoor pedestrian scenarios, accomplishing more than $6Km$ of autonomous navigation, with a success ratio of about 99% completed *go to* requests [19].

Such results for 3D navigation in real time were only possible thanks to a key software module that was developed to compute expected laser range observations by means of rendering the 3D environment model in a rendering window resized from the laser device specifications [16]. These expected range measurements were compared to true laser measurements in the correction loop of the particle filter-based localization module. Clever optimizations in the code allowed us to run the filter with a sufficiently large number of particles without loosing real-time performance and with low latency on the localization estimate.

Figure 7 shows the graphical user interface used to monitor all the components of the real-time autonomous navigation experiments made in our urban scenarios. Starting from the top left and clockwise: (1) simulated 3D environment model view from the estimated robot position, (2) scene image of the experiment from a hand held camera, (3) projection of the vertical laser scanner data on the local xz plane, (4) projection of the horizontal back and front laser data on the local xy plane in green, and expected range data for the front laser in red, (5) institutional logos, (6) inputs and outputs of the local planner shown on the local xy plane: laser hit points are shown in red (front) and in gray (back), the local goal and RRT segments are also plotted; the green mesh represents slopped region of the local xy plane detected thanks to vertical laser processing, (7) Zoomed region of the global xy plane: in blue the particle set projected to that plane, and in red, way points of the global path, (8) Global xy plane with the 2D map representation with the whole planned path for the current *go to* request in red, the accumulated odometry path in green and robot localization estimates in blue.

Future Requirements

The URUS project allowed us to gain valuable experience during the field sessions of autonomous navigation in urban pedestrian areas. This experience points out a set of future requirements and research lines that should be addressed to improve autonomous navigation in such environments. We divide these either with respect to the local reactive loop or the global deliberative one.

In the local robot plane, full 3D perception and planning becomes mandatory for robust and reliable navigation in an urban pedestrian scenario. To this end, 3D sensors with reliable outdoor response, and real-time processing methods for local planning are both key modules to be implemented in the near future. Also in the reactive loop, on-board pedestrian detection and tracking, and execution of socially acceptable trajectories are two research topics to need to be addressed by urban pedestrian mobile robotics research.

In the deliberative loop, the key module was that of localization. In the case of densely populated outdoor environments, the localization approach should combine an improved odometry technique (such as visual odometry) with a map-based one, in order to cope with large periods where real and expected observations would differ due to the presence of pedestrians. Moreover, when a sensor network will be available in the environment, an extra global localization layer, running at low rate and fusing remote observations from the sensor network, should be envisaged to provide a final solution with a greater level of robustness.

VI. Conclusions

We have presented in this work recent research results of people guidance by robots in large urban sites and robust navigation. All the techniques have been validated with field experiments in the Barcelona Robot Lab, a 10.000 *sq m*, outdoor site. Results are promising, however, we have realized that one important issue that have to be solved before we can massively deploy service robots in urban pedestrian areas. Robust navigation requires good sensors that give 3D information in real time along with real time

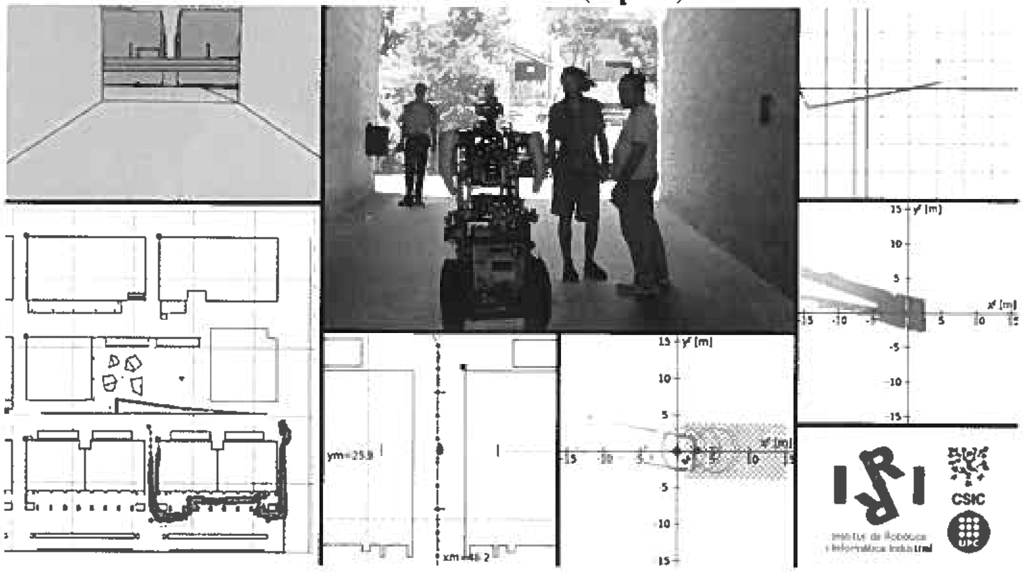

Fig. 7. Graphical user interface for real-time monitoring of autonomous navigation experiments in urban scenarios The figure shows a moment in which the robot was avoiding two pedestrians in a sloped area.

information of the environment sensors. People guidance research results are quite good but still premature, since we have not considered the human robot interaction with unpredictable human behavior in urban spaces.

REFERENCES

[1] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S.A. Adelaide. A tutorial on particle filters for online nonlinear/non-gaussianbayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

[2] M. D. Bolic and P. M. S. Hong. Resampling algorithms and architectures for distributed particle filters. *Signal Processing, IEEE Transactions on*, 53(7):2442–2450, 2005.

[3] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. Lockerd, and D. Mulanda. Humanoid robots as cooperative partners for people. *Int.Journal of Humanoid Robots*, 1(2), 2004.

[4] L. Chaimowicz and V. Kumar. Aerial shepherds: Coordination among uavs and swarms of robots. *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.

[5] A. Garrell and A. Sanfeliu. Local optimization of cooperative robot movements for guiding and regrouping people in a guiding mission. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3294–3299, Taipei, 2010.

[6] A. Garrell, A. Sanfeliu, and F. Moreno. Discrete time motion model for guiding people in urban areas using multiple robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, 2009.

[7] D. Helbing. A mathematical model for the behavior of pedestrians. *Behavioral Science*, 36(4), 1991.

[8] D. Helbing, I. Farkas, P. Molnar, and T. Vicsek. Simulation of pedestrian crowds in normal and evacuation simulations. *M. Schreckenberg and S. Sharma (Eds.), Pedestrian and Evacuation Dynamics, Springer*, 2002.

[9] D. Helbing and P. Molnar. Social force model for pedestrian dynamics *Physical Review E 51*, pages 4282-4286, 1995.

[10] D. Helbing and P. Molnar. Self-organization phenomena in pedestrian crowds, in self-organization of complex structures: From individual to collective dynamics. *F. Schweitzer (Ed.) (Gordon and Breach, London)*, pages 569-577, 1997.

[11] C. Hofner and G. Schmidt. Path planning and guidance techniques for an autonomous mobilecleaning robot. *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*, 1, 1994.

[12] Y. K. Hwang and N. Ahuja. A potential field approach to path planning. *IEEE Transactions on Robotics and Automation 8(1)*, 1992.

[13] J.M. Lien, S. Rodriguez, J.P. Malric, and N.M. Amato. Shepherding behaviors with multiple shepherds. *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2005.

[14] B. Mazzolai, V. Mattoli, C. Laschi, P. Salvini, G. Ferri, G. Ciaravella, and P. Dario. Networked and cooperating robots for urban hygiene: the EU funded Dustbot project. In *International Conference on Ubiquitous Robots and Ambient Intelligence*, Seoul, 2008.

[15] A. Corominas Murtra, E. Trulls, O. Sandoval, J. Pérez-Ibarz, D. Vasquez, J.M. Mirats Tur, M. Ferrer, and A. Sanfeliu. Autonomous navigation for urban service mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei.

[16] A. Corominas Murtra, E. Trulls, J.M. Mirats Tur, and Sanfeliu. Efficient use of 3d environment models for mobile robot simulation and localization. In *International Conference on Simulation, Modeling and Programming for Autonomous Robots*, pages 461–472, Darmstadt, Vol. 6472 of Lecture Notes on Artificial Intelligence.

[17] A. Sanfeliu and J. Andrade-Cetto. Ubiquitous networking robotics in urban settings. In *IROS Workshop on Network Robot Systems*, Beijing, 2006.

[18] A. Sanfeliu, J. Andrade-Cetto, M. Barbosa, R. Bowden, J. Capitán, A. Corominas Murtra, A. Gilbert, J. Illingworth, L. Merino, Josep M. Mirats Tur, P. Moreno, A. Ollero, J. Sequeira, and M.T. Spaan. Decentralized sensor fusion for ubiquitous networking robotics in urban areas. *Sensors*, 10(3):2274–2314, March 2010.

[19] E. Trulls, A. Corominas Murtra, J. Pérez-Ibarz, G. Ferrer, D. Vasquez, J.M. Mirats Tur, and A. Sanfeliu. Autonomous navigation for mobile service robots in urban pedestrian environments. *Journal of Field Robotics*, 2011. To appear.