

Discrete Time Motion Model for Guiding People in Urban Areas using Multiple Robots

Anaís Garrell, Alberto Sanfeliu, and Francesc Moreno-Noguer
Institut de Robòtica i Informàtica Industrial (CSIC-UPC)
08028 Barcelona, Spain

Abstract— We present a new model for people guidance in urban settings using several mobile robots, that overcomes the limitations of existing approaches, which are either tailored to tightly bounded environments, or based on unrealistic human behaviors. Although the robots motion is controlled by means of a standard particle filter formulation, the novelty of our approach resides in how the environment and human and robot motions are modeled. In particular we define a “Discrete-Time-Motion” model, which from one side represents the environment by means of a potential field, that makes it appropriate to deal with open areas, and on the other hand the motion models for people and robots respond to realistic situations, and for instance human behaviors such as “leaving the group” are considered.

I. INTRODUCTION

The interest on developing social and cooperative robots has significantly increased throughout the recent years. The applications of this field are very diverse, from developing automatic exploration sites [18], to building robot formations for transporting and evacuating people during emergency situations [5], [8].

Within the area of social and cooperative robots, arises one important application which is that of using one or several robots to guide a group of people. Similar applications have been previously developed for guiding flocks of animals [16], although these approaches are constrained to closed areas and only consider one robot.

We present a new approach for guiding people in open areas of urban settings using multiple robots acting in a cooperative way. One of the robots is the *leader*, as a human tour-guide. It is placed at the front of the group and its role is to estimate the trajectory of both the people and the rest of robots. The other robots, called *shepherds*, are responsible for guiding the people, preventing any person leaving the group, and following the path given by the leader. An schematic of the situations we will consider is shown in Fig. 1.

At the core of our approach lies a “Discrete Time Motion” (DTM) model which is used to represent people’s and robot’s motions. The DTM needs to predict people’s movements in order to give the motion instructions to the robots. This is

This research was conducted at the Institut de Robòtica i Informàtica Industrial (CSIC-UPC). It was partially supported by CICYT projects DPI2007-61452 and Ingenio Consolider CSD2007-018, by CSIC project 2008501055 and by IST-045062 of European Community. The first author acknowledges Spanish FPU grant ref. AP2006-00825.

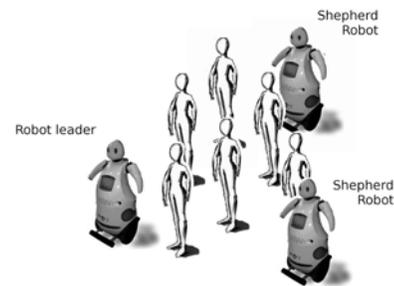


Fig. 1. Guiding people using a group of cooperative robots.

done by means of a Particle Filter formulation [1], with the particularity that it uses realistic human motion models.

The interaction with the obstacles of the environment, such as buildings or benches, is considered through a potential field, where the positions of people and robots are represented by continuous and derivable functions. Since the obstacles are assumed to be static, their positions are represented by constant functions. Using these parameterizations each point in the space will have assigned a potential value, which will be used to control the motion of all the robots.

In the remainder of the paper we start by discussing related work. Section III shortly describes the people’s motion model. Section IV describes the representation of the whole environment by means of a potential field, and how this is used to guide the robot’s motion. Section V describes the particle filter formulation we use to estimate position and velocity of people and robots. Results and conclusions are presented in sections VI and VII, respectively.

II. RELATED WORK

Developing social and cooperative robots is a quite novel field within robotics. As a consequence, the number of related references is not very large, especially if we refer to the challenge of guiding a group of people in urban areas. There has been some research in using a single robot for guiding people in exhibitions and museums [4], in hospitals or as an assistant [7]. Nevertheless, the main purpose of these robots was simply educational or to entertain, instead of guiding people. Similar applications have been developed for evacuating emergency areas, detecting hazardous materials, or offering task assistance to humans. For instance, the first known use of mobile robots for urban search and rescue, was

during the World Trade Center disaster [15], these robots have the unique capability to collect useful data, and they were not specifically designed to guide people, and they did not behave in a cooperative way. Animal flocks were automatically controlled using a single robot in [16]. Again, the cooperative behavior of our approach is not exploited in these methods, and the environment where the systems are shown to work are highly controlled, do not include obstacles and are tightly closed.

All the methods mentioned above consider either single robots, or multiple robots moving independently from the rest. To our knowledge, only a few works deal with multiple robots behaving in a cooperative mode. For instance, [9] performs a qualitative analysis of the movements of different entities –such as humans or animals– and uses it to build an architecture of three robots to guide them. However, realistic situations, such as obstacles or dealing with people leaving the group are not considered. In [14] several types of robot formations, and different strategies for approaching the robots to the 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. Furthermore, in order to achieve the desired guiding results, robot motions with almost infinite accelerations are required.

In contrast to the previous approaches, the “Discrete-Time-Motion” model we propose, offers a framework to tackle more realistic situations and without the need of using such a large number of heuristics.

III. MODELING PEOPLE’S MOTION

In order to model people’s motion we will use the concepts introduced by the works of Helbing et al. [10], [11], that study the dynamics of pedestrian crowds from the “social” point of view. More specifically, they describe the motion of pedestrians based on social forces which are the result of the internal motivations of the individuals to perform certain motions. These forces, for some simple situations, can be described through probabilistic models. The three situations considered in the previous approaches are the following: (i) The pedestrian wants to reach a certain destination as comfortable as possible, (ii) the motion of a pedestrian is influenced by other pedestrians.

Let us now explain how these situations are mathematically represented. For the first situation, people usually take the shortest path, which may be formally represented as the shape of a polygon with edges $\vec{r}_\alpha^1 \dots \vec{r}_\alpha^n := \vec{r}_\alpha^0$, where α refers to a given person and \vec{r}_α^0 the destination he/she wants to reach.

The desired motion direction $\vec{e}_\alpha(t)$ of a pedestrian α will then be:

$\vec{e}_\alpha(t) := \frac{\vec{r}_\alpha^k - \vec{r}_\alpha(t)}{\|\vec{r}_\alpha^k - \vec{r}_\alpha(t)\|}$ where $\vec{r}_\alpha(t)$ is the *current position* and \vec{r}_α^k is the subsequent edge of the polygon that will be reached. A deviation of the desired speed, v_α^0 , from the current velocity, $\vec{v}_\alpha^0(t) := v_\alpha^0 \vec{e}_\alpha(t)$, may also exist due to deceleration or obstacle avoidance processes:

$$\vec{F}_\alpha^0(\vec{v}_\alpha, v_\alpha^0 \vec{e}_\alpha) := \frac{1}{\tau_\alpha} (v_\alpha^0 \vec{e}_\alpha - \vec{v}_\alpha) \quad (1)$$

where τ_α is a relaxation term. In practice we set the term τ to 0.5 for all the pedestrians.

Let us now consider the situation 2), when the pedestrian motion is influenced by other pedestrians from the group. This situation responds to the fact that each individual tries to maintain an empty security volume surrounding him [19]. This is in fact a repulsive effect which we model through the following vectorial quantity:

$$\vec{F}_{\alpha\beta}(\vec{r}_{\alpha\beta}) = -\nabla_{\vec{r}_{\alpha\beta}} V_{\alpha\beta} [b(\vec{r}_{\alpha\beta})] \quad (2)$$

where $V_{\alpha\beta}(b)$ is a repulsive potential which is assumed to be a monotonic decreasing function of b with equipotential lines having an elliptical shape. $\vec{r}_{\alpha\beta} = \vec{r}_\alpha - \vec{r}_\beta$. The parameter b denotes the semi-minor axis of the ellipse and is given by:

$$b = \frac{\sqrt{(\|\vec{r}_{\alpha\beta}\| + \|\vec{r}_{\alpha\beta} - v_\beta \Delta t \vec{e}_\beta\|)^2 - (v_\beta \Delta t)^2}}{2} \quad (3)$$

where, $v_\beta \Delta t$ is an approximation to the step size of a pedestrian β .

Finally, we will consider the repulsive effect produced by the distance that people try to keep from the obstacles of the environment. The nature of this force is the same we just described between individuals, with the difference that the obstacles do not move. If we denote by B the border of an obstacle the repulsive effect it creates will be described by:

$$\vec{F}_{\alpha B}(\vec{r}_{\alpha B}) = -\nabla_{\vec{r}_{\alpha B}} U_{\alpha B}(\|\vec{r}_{\alpha B}\|) \quad (4)$$

where $U_{\alpha B}(\|\vec{r}_{\alpha B}\|)$ is the repulsive and monotonically decreasing potential function, $\vec{r}_{\alpha B} = \vec{r}_\alpha - \vec{r}_B$, and B denotes the position of the border B that is closest to the pedestrian α .

IV. MODELING THE MOTION SPACE

In this section, we will discuss the modelization we use to represent the whole environment, made of an open and not bounded area with obstacles, and how the elements of this environment are related with the group of robots and persons. The key element to represent these relations 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), and the k total number of time instances is T .

The DTM model has two components (Fig.4): 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.

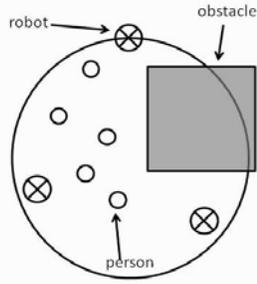


Fig. 2. Working area at a given time instance. The dimension and position of the working area changes over time.

A. The Discrete Time Component

The first task of the Discrete Time component is to estimate position, orientation and velocity of the robots and persons. This is done with a standard particle filter formulation, and we postpone the details of this filter until Section V.

Then, the Discrete Time component aims to represent the areas where the robots will be allowed to move, by means of potential fields. To this end, we define a set of functions that describe the tension produced by the obstacles, people and robots over the working area. These tensions are computed based on the area defined by a security region surrounding each one of the persons, robots and obstacles.

More specifically, we first define the position and dimension of the *working area*. This will be a circle big enough to include all the robots and persons, and placed in such a way that its perimeter passes through the *leader* robot. The obstacles placed within this area are also represented (Fig. 2). Note that this working area changes over time, and consequently the whole dimension of the environment is not strictly limited.

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 [13]. 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:

$$T_p(\mu_p, \Sigma_p)(x) = \frac{1}{|\Sigma_p|^{1/2} (2\pi)^{n/2}} e^{-\frac{1}{2}(x-\mu_p)^T \Sigma_p^{-1} (x-\mu_p)} \quad (5)$$

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

Algorithm 1 General strategy for guiding people

- 1: Obtain the start point and the goal point.
 - 2: Compute the *roadmap* with the path planning.
 - 3: Search the shortest path of the *roadmap*.
 - 4: Mark every node of the shortest path as a subgoal.
 - 5: **for** Every subgoal **do**
 - 6: Act upon the situation (open path, narrow passages...) carrying out the priorities.
 - 7: Move to the next subgoal
 - 8: **end for**
-

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 $X = \{(x_1, y_1), \dots, (x_n, y_n)\}$ 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 $i = 1, \dots, n$, where T_i follows Equation 5.

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 [12]. 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*

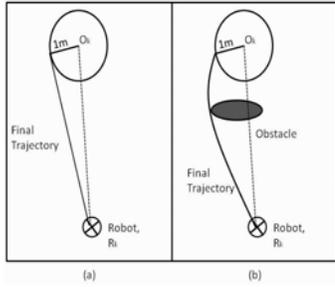


Fig. 3. (a) Rectilinear trajectory. (b) Trajectory with an obstacle.

robot who transmits the computed path to the rest of the robots.

The second case is the study of shepherding algorithms [3], [6], 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 rest of robots follow the strategy depicted in Algorithm 1. Note that this algorithm does not explicitly consider safety conditions for the persons; i.e., when the robots are working it is necessary to satisfy a set of priorities for the safety of the people, such as avoiding collisions. However this was already taken into account when we defined the security areas in the Gaussian functions parameterizing the tensions.

Algorithm 1 just presents the general shepherding strategy. However, we may find several particular situations which should be considered, such as guiding in open roads with no obstacles, guiding in narrow corridors, moving closer the group of persons or rescuing a group member that left the group. In these particular situations one may have to design specific algorithms.

One important situation we must carefully consider is the case when people escape from the group. We are not aware of any approach tackling this problem. The solution we adopt for this situation is to choose one of the robots—the one closer to the individual who left—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. 3(a): 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.

As shown in Fig. 3(b), if the rectilinear trajectory contains an obstacle, the path that minimizes tensions and is closer to the subgoal will be chosen.

V. ESTIMATING POSITIONS AND VELOCITIES OF PEOPLE AND ROBOTS

As we previously mentioned, in order to model the motion space we need to estimate people positions and velocities. This is done using a Particle Filter, whose particular implementation will be explained below.

Our main concern was to prevent humans and robots enter in collision with each other and with the obstacles. This was achieved by a modification of the measure update part of the algorithm, as was previously suggested in [1] and [2]. Let us see how we proceed:

The problem consists in estimating the dynamic state of a nonlinear stochastic system, based on a set of noisy observations. Our model can be written as follows:

$$\begin{aligned} x_n &= f(x_{n-1}, u_n) && \text{(process equation)} \\ y_n &= h(x_n, v_n) && \text{(observation equation)} \end{aligned} \quad (6)$$

where $f(\cdot)$ and $h(\cdot)$ are some known nonlinear functions, x_n the state vector, y_n the observation, and u_n and v_n are random noise components of given distributions. We denote by $x_{0:n}$ and $y_{0:n}$ the signal and observation up to time n , respectively, i.e., $x_{0:n} := \{x_0, \dots, x_n\}$ and $y_{0:n} := \{y_0, \dots, y_n\}$. Our aim is to recursively estimate the posterior distribution $p(x_n|y_{0:n})$, and the predictive distribution $p(x_{n+1}|y_{0:n})$. We can write them following the standard Bayesian filter equations:

$$\begin{aligned} p(x_n|y_{0:n}) &= C_n p(x_n|y_{0:n-1}) p(y_n|x_n) && (7) \\ C_n^{-1} &= \int p(x_n|y_{0:n-1}) p(y_n|x_n) dx_n \\ p(x_{n+1}|y_{0:n}) &= \int p(x_{n+1}|x_n) p(x_n|y_{0:n}) dx_n && (8) \end{aligned}$$

The Particle Filter approximates the distributions of Eq. 7 and 8 by a set of weighted particles. The steps are summarized in Algorithms 2 and 3. Initially, a set of M particles $X = \{x_n^{(1)}, \dots, x_n^{(M)}\}$ from a so-called importance sampling distribution $\pi(x_n)$ are generated. A weight $w^{(j)} = p(x_n^{(j)})/\pi(x_n^{(j)})$ is then assigned to each one of the particles. If we write $W = \{w^{(1)}, \dots, w^{(M)}\}$, the set $\{X, W\}$ will represent samples that approximate the posterior distribution $p(x_n|y_{0:n})$.

In order to choose the sampling function $\pi(\cdot)$ we considered $p(x_n|y_{0:n-1}) = N(x_n, \mu_n, \Sigma_n)$. What is new in our particular implementation of the algorithm is how we compute the weights based on the probabilities of our motion space. To each particle we associate a position in the *working area* and we recompute the weight using the probability in that position based on *potential field* previously estimated. As written in Equation 10, this reweighting is done using a new function $g(x_n^{(j)})$, which re-adjusts the weights of all the set of particles $X = \{x_n^{(1)}, \dots, x_n^{(M)}\}$.

VI. IMPLEMENTATION AND RESULTS

The current work is done within the framework of the European Project URUS [17], and the scenario where the experiments will be performed corresponds to an urban area of about 10.000 m^2 within the North Campus of the Technical University of Catalonia (UPC). The area contains different obstacles, such as buildings, benches and trash cans. An schematic of the testing area is depicted in Fig. 7, left side.

The results we will present correspond to different synthetic experiments, some of them within the previous map.

Algorithm 2 Measurement update algorithm

- 1: Draw samples from the function $\pi(x_n|y_{0:n}) \rightarrow \{x_n^{(j)}\}_{j=1}^M$
- 2: Compute the respective weights by:

$$\bar{w}_n^{(j)} = \frac{p(y_n|x_n^{(j)})N(x_n = x_n^{(j)}, \bar{\mu}_n, \bar{\Sigma}_n)}{\pi(x_n^{(j)}|y_{0:n})} g(x_n^{(j)}) \quad (9)$$

$$g(x_n^{(j)}) = \begin{cases} \alpha & \text{if } T(g(x_n^{(j)})) = 0, \alpha \gg 1 \\ \frac{1}{T(g(x_n^{(j)}))} & \text{otherwise} \end{cases} \quad (10)$$

- 3: Normalize the weight as

$$w_n^{(j)} = \frac{\bar{w}_n^{(j)}}{\sum_{j=1}^M \bar{w}_n^{(j)}} \quad (11)$$

- 4: Estimate the mean and covariance by

$$\begin{aligned} \mu_n &= \sum_{j=1}^M w_n^{(j)} x_n^{(j)} \\ \Sigma_n &= \sum_{j=1}^M w_n^{(j)} (\mu_n - x_n^{(j)}) (\mu_n - x_n^{(j)})^T \end{aligned} \quad (12)$$

Algorithm 3 Time update algorithm

- 1: Draw samples from $N(x_n, \mu_n, \Sigma_n) \rightarrow \{x_n^{(j)}\}_{j=1}^M$
- 2: **for** $j=1, \dots, M$ **do**
- 3: sample from $p(x_{n+1}|x_n = x_n^{(j)})$ to obtain $\{x_{n+1}^{(j)}\}_{j=1}^M$
- 4: **end for**
- 5: Compute the mean $\bar{\mu}_{n+1}$ and the covariance by $\bar{\Sigma}_{n+1}$

$$\begin{aligned} \bar{\mu}_{n+1} &= \frac{1}{M} \sum_{j=1}^M x_{n+1}^{(j)} \\ \bar{\Sigma}_{n+1} &= \frac{1}{M} \sum_{j=1}^M (\bar{\mu}_{n+1} - x_{n+1}^{(j)}) (\bar{\mu}_{n+1} - x_{n+1}^{(j)})^T \end{aligned} \quad (13)$$

In these experiments, the dynamical models of the persons –we considered a group of 5 persons– will follow the models described in Section III. We will assume a group of two robots, that will move according to the motion model computed in Section IV.B, after estimating the potential field. The observations are simulated by adding Gaussian noise to the real positions.

In order to make the problem tractable we discretized the working area by means of a rectangular mesh. The resolution and size of the mesh will depend on the number and density of people (Fig. 4). Typical sizes are 15×15 meshes with an internode distance of 25 cm. Note that this is only a local discretization, and the mesh will move along the whole environment of $10.000 m^2$. The tension values and potential

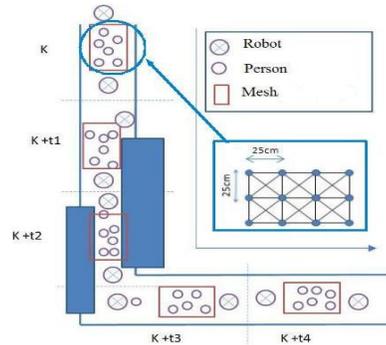


Fig. 4. Discretization of the working area using a local mesh.

field will be computed on the nodes of the mesh.

We made three different experiments. In the first one two robots guided a group of 5 people in an open area without obstacles. Fig. 5 shows different time instances of the simulation process. The left hand images represent a top view of the environment map, in this case without obstacles. The position of the two robots is plotted with circles and the five persons are represented by asterisks. The right hand images plot the corresponding potential field. Fig.5(a) shows the initial configuration with the robots surrounding the group of persons. In Fig.5(b) one of the individuals just left group, and immediately after one of the robots followed him (Fig.5(c)). In Fig.5(d) we plot the final configuration, where all the persons reached the goal.

In the second experiment we introduced one obstacle between the initial position of the group and the goal. Fig.6 shows different time instances, again assuming that one robot needs to follow one of the individuals who left the group.

Finally, in the third experiment we show the performance of our algorithm when the group of 5 people is moving across the Campus area (Fig.7). Note that in this case the task of the robots is made easier because the large number of obstacles –buildings, walls, stairs– highly constrain the movement of the persons.

VII. CONCLUSION

We have presented a new model to guide people in urban areas with a set of mobile robots working in a cooperative manner. In contrast to existing approaches, our method can tackle more realistic situations, such as dealing with large environments with obstacles, or regrouping people who left the group. For that reason, this work can be applied in some real robots applications, for instance, guiding people in emergency areas, or acting as a robot companion.

We presented various results in different situations: guiding in open areas, in areas with a single obstacle, and urban areas with a large number of obstacles. In all of these experiments we showed that the robots can act early enough to satisfactorily guide group of people through a path calculated previously.

We are currently working in improving the strategies with several robots. For instance, we plan to study which is the optimal number of robots depending on the number of people

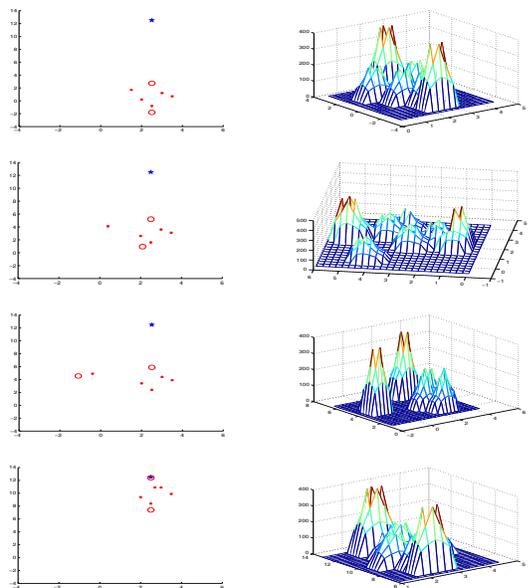


Fig. 5. Experiment 1: Guiding people in an open area with no obstacles.

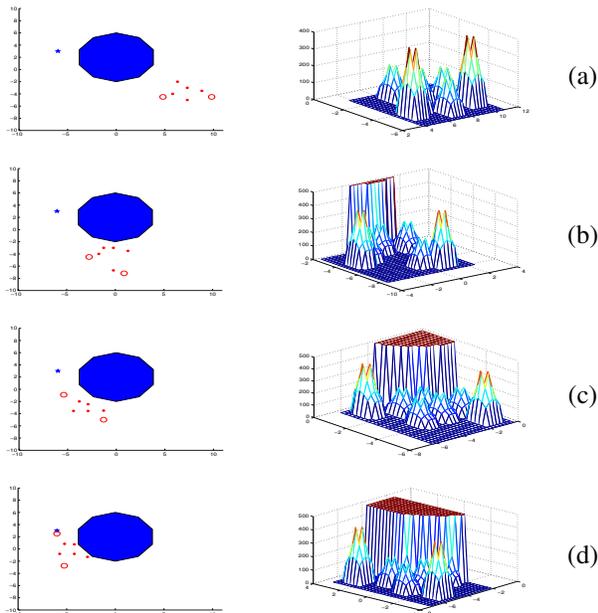


Fig. 6. Experiment 2: Guiding people in an area with one obstacle.

and the configuration of the environment. We also plan to add the path planning capabilities to the *shepherd* robots, and not only to the *leader*. Finally, use smoother trajectories, instead of piecewise rectilinear ones, is also part of the future work.

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-gaussian bayesian 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.

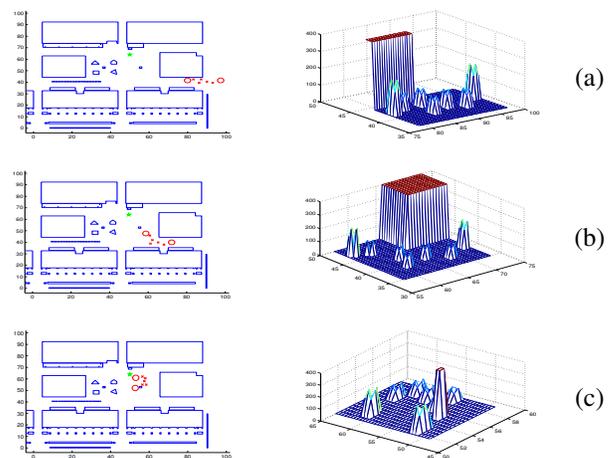


Fig. 7. Experiment 3: Guiding people across North Campus UPC.

- [4] W. Burgard, A. B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proc. of the Fifteenth National Conference on Artificial Intelligence, Madison, Wi*, 1998.
- [5] J. Casper and R. R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE transactions on systems, man, and cybernetics.*, 33(3):367–385, 2003.
- [6] 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*, page 231240, 2004.
- [7] K. Dautenhahn, S. Woods, C. Kaouri, M. L. Walters, K. L. Koay, and I. Werry. What is a robot companion friend, assistant or butler? *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*, 25:1488–1493, 2005.
- [8] J.P. Desai, J.P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *Robotics and Automation, IEEE Transactions on*, 17(6):905–908, 2001.
- [9] Martinez Garcia E.A., Akihisa O., and S. Yuta. Crowding and guiding groups of humans by teams of mobile robots. *Advanced Robotics and its Social Impacts, 2005. IEEE Workshop on*, pages 91–96, 2005.
- [10] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E* 51, pages 4282–4286, 1995.
- [11] 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.
- [12] 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.
- [13] Y. K. Hwang and N. Ahuja. A potential field approach to path planning. *IEEE Transactions on Robotics and Automation* 8(1), page 2332, 1992.
- [14] 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.
- [15] R.R. Murphy. Human-robot interaction in rescue robotics. *IEEE Trans. on Systems, Man and Cybernetics*, 34(2):138–153, 2004.
- [16] R.T. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron. Experiments in automatic flock control. *Journal Robot and Autonom. Sys.*, pages 191–209, 2000.
- [17] A. Sanfeliu and J. Andrade-Cetto. Ubiquitous networking robotics in urban settings workshop on network robot systems. *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*, 2006.
- [18] C. Trevai, Y. Fukazawa, J. Ota, H. Yuasa, T. Arai, and H. Asama. Cooperative exploration of mobile robots using reaction-diffusion equation on a graph. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2, 2003.
- [19] S. Worchel, J. Cooper, G. R. Goethals, and J. M. Olson. *Social psychology*. Thomson Learning Ibero, 2002.