Int J Soc Robot (2010) 2: 79–94
DOI 10.1007/s12369-009-0037-z

# Probabilistic Autonomous Robot Navigation in Dynamic Environments with Human Motion Prediction

**Amalia F. Foka · Panos E. Trahanias**

Accepted: 16 December 2009 / Published online: 6 January 2010


**Abstract** This paper considers the problem of autonomous robot navigation in dynamic and congested environments. The predictive navigation paradigm is proposed where probabilistic planning is integrated with obstacle avoidance along with future motion prediction of humans and/or other obstacles. Predictive navigation is performed in a global manner with the use of a hierarchical Partially Observable Markov Decision Process (POMDP) that can be solved online at each time step and provides the actual actions the robot performs. Obstacle avoidance is performed within the predictive navigation model with a novel approach by deciding paths to the goal position that are not obstructed by other moving objects movement with the use of future motion prediction and by enabling the robot to increase or decrease its speed of movement or by performing detours. The robot is able to decide which obstacle avoidance behavior is optimal in each case within the unified navigation model employed.

**Keywords** Navigation · Obstacle avoidance · Motion prediction · Path planning · POMDPs

## 1 Introduction

For humans, the ability to navigate intentionally is eminent. For mobile robotic systems, however, navigation in dynamic real-world environments is an extremely complex and challenging task. Such environments are characterized by their complex structure and the movement of humans and objects in them. The robot has to avoid collision with obstacles and also reach its goal position in a fast and optimal manner.

The problem of a robot navigating in a crowded environment is treated so far mainly by incorporating separate cooperating modules for global path planning, local path planning (obstacle avoidance) and localization. This paper utilizes a unified model that incorporates the modules for localization, global and local motion planning. The employed model is a hierarchical POMDP specifically designed for autonomous robot navigation, termed as the Robot Navigation-HPOMDP (RN-HPOMDP), originally presented in [9]. The RN-HPOMDP enables us to perform all aspects of navigation in a probabilistic and unified manner since there is no intervention of any other external module. The RN-HPOMDP offers great advantages since probabilistic approaches for localization and mapping have been widely employed but this is not the case for robot motion planning. The modelling of the RN-HPOMDP is presented in [9] and in this paper its application under the predictive navigation paradigm is introduced.

Probabilistic planning is performed under the predictive navigation paradigm in order to be able to simulate as best as possible the human behavior for obstacle avoidance. Future motion prediction is an intrinsic behavior of humans. When humans walk in an environment they perceive through their vision the movement of other humans or other moving objects. Humans use this information and attempt to estimate the easiest, i.e. unblocked, and shortest path they should follow to reach their destination. This is in effect a predictive behavior. It would be desirable for an autonomous robot to develop a similar behavior. Future motion prediction allows to obtain paths that are not only optimal in a time or distance

A.F. Foka (✉)
Department of Computer Engineering & Informatics, University of Patras, 26500 Patras, Greece
e-mail: foka@ceid.upatras.gr

P.E. Trahanias
Institute of Computer Science, Foundation for Research and Technology—Hellas (FORTH), P.O. Box 1385, Heraklion, 711 10 Crete, Greece
e-mail: trahania@ics.forth.gr

traveled sense but it can also foresee situations where the robot might get blocked and chooses new paths that avoid such situations.

The predictive navigation paradigm presented in this paper enables the robot to decide the best behavior for obstacle avoidance in each case. The robot can decide at any time step to either change completely its path to its goal position in order to follow an unblocked path, or perform a detour. The detour will be executed well before the robot gets too close to the obstacles and in addition the robot has foreseen that after performing this detour it will be able to complete its path to the goal position without any other obstructions. Furthermore, the robot can also decide to change its speed of movement to a lower one to allow an obstacle to move away from its motion path or bypass the obstacle by increasing its speed.

Obstacle avoidance in the proposed framework offers great advantages over standard methods present in [2, 3, 10, 13, 18, 21, 23]. The main disadvantage of all these methods is that they treat the problem locally. The local treatment of the problem directs the robot into executing globally suboptimal paths to its destination point. This is due to the fact that they stick on the initial planned path to the destination point obtained by the global path planning module and do not replan no matter the changes that occur in the environment.

Furthermore, obstacle avoidance in the proposed approach is performed within the predictive navigation framework, i.e. it utilizes future motion prediction of humans and/or other objects to decide what is the best approach the robot should employ. In the proposed approach two kinds of prediction are utilized: short-term and long-term. The short-term prediction refers to the one-step ahead prediction and the long-term to the prediction of the final destination point of the obstacle's movement. One-step ahead prediction has been previously utilized for obstacle avoidance in [5, 6, 12, 17, 19, 20, 24, 25, 27, 28]. However, knowing an obstacle's current position and its immediate next time-step position is not indicative about the general obstacle behavior. In result, one-step ahead prediction can potentially assist the obstacle avoidance module locally but cannot be that effective when deciding the final complete path the robot will execute to reach its destination position. For this reason, long-term prediction is employed that estimates the final destination point of an obstacle's motion trajectory. Recent methodologies for predicting the whole path an obstacle is following have been proposed in [1, 4, 16, 22, 26].

Finally, the robot can also avoid obstacles either by increasing its speed to bypass the obstacle or decreasing its speed to let the obstacle move away from the robot. Consequently, future motion prediction is also exploited to enable the robot to decide if it should increase or decrease its speed to avoid an obstacle more effectively. Once more, this feature is integrated into the global navigation model, the RN-HPOMDP.

Experimental results have shown the applicability of the proposed approach for predictive autonomous robot navigation in dynamic environments where humans and moving objects are avoided efficiently and the robot follows optimal paths to reach its destination.

The remaining of this paper describes how predictive navigation is achieved with the use of the RN-HPOMDP that is outlined in Sect. 2. First, it is described how future motion prediction is obtained in Sect. 3 and how it is integrated to the global navigation model in Sect. 4. Following, the methodology used to enable the robot to avoid obstacles by changing its speed of movement is presented in Sect. 5. Finally, the results and the conclusions of this work are presented in Sects. 6, 7 and 8, respectively.

## 2 Partially Observable Markov Decision Processes

In this paper, we utilize a hierarchical representation of POMDPs specifically designed for autonomous robot navigation, the RN-HPOMDP as detailed in [9]. Here, the formulation of the RN-HPOMDP is presented for clarity reasons and the rest of this paper is concerned with its application to the predictive navigation problem. The RN-HPOMDP can efficiently model large real world environments at a fine resolution and can be solved in real time. It is utilized as a unified framework for the autonomous robot navigation problem, where no other external modules are used to drive the robot. In other words, the RN-HPOMDP integrates the modules for localization, planning and obstacle avoidance. The RN-HPOMDP is solved on-line at each time step and decides the actual actions the robot performs.

### 2.1 POMDP Formulation for Robot Navigation

In the following we present a formulation of POMDPs for autonomous robot navigation in a unified framework. The POMDP decides the actions the robot should perform to reach its goal and also robustly tracks the robot's location in a probabilistic manner. In the problem considered in this paper, we are interested in dynamic environments and hence the POMDP also performs obstacle avoidance. All three functionalities are carried out without the intervention of any other external module.

In our implementation the robot perceives the environment by taking horizontal laser scans. In addition, an occupancy grid map (OGM) of the environment obtained at the desired discretization is provided. The OGM is used to determine the set of possible states the robot might occupy. Laser measurements are used to obtain observations. In the following, the elements of a POMDP, $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \mathcal{O} \rangle$, are instantiated for robot navigation as:

Set of states, $\mathcal{S}$: Each state in $\mathcal{S}$ corresponds to a discrete entry cell in the environment's occupancy grid map (OGM) and an orientation angle of the robot with respect to a global reference system, i.e. each state $s$ is a triplet $(x, y, \theta)$.

Set of actions, $\mathcal{A}$: It consists of all possible rotation actions from 0° to 360° termed as "action angles".

Set of observations, $\mathcal{Z}$: The observation set is the element of the POMDP that assists in the localization of the robot, that is the belief update after an action has been executed. The set of observations is instantiated as the output of the *iterative dual correspondence* (IDC) algorithm of Lu and Milios [15] for scan matching.

Reward function, $\mathcal{R}$: Since the proposed POMDP is used as a unified framework for robot navigation that will provide the actual actions the robot will perform and also carry out local obstacle avoidance for moving objects, the reward function is updated at each time step. The reward function is built and updated at each time step, according to two *reward grid maps* (RGMs): a *static* and a *dynamic* as in [7]. The RGM is defined as a grid map of the environment in analogy with the OGM. Each of the RGM cells corresponds to a specific area of the environment with the same discretization of the OGM, only that the value associated with each cell in the RGM represents the reward that will be assigned to the robot for ending up in a specific cell. The static RGM is built once by calculating the distance of each cell to the goal position and by incorporating information about cells belonging to static obstacles. The dynamic RGM is responsible for incorporating into the model information about the current state of the environment, i.e. whether there are objects moving within it or other unmapped objects. The detailed procedure for building and updating the RGMs is given in Sect. 4.

Transition and observation functions, $\mathcal{T}$ and $\mathcal{O}$: They are initially defined according to the motion model of the robot and then they are learned as detailed in [9].

## 3 Motion Prediction

Motion prediction is utilized to obtain a more effective behavior for obstacle avoidance. Two kinds of prediction are used: short-term and long-term prediction. Short-term prediction refers to the one-step ahead prediction of a human's or an other object's future position. Short-term prediction is obtained by a Polynomial Neural Network (PNN) that is trained with an evolutionary method as originally presented in [7]. The PNN is trained once offline with motion data obtained and thus there is no computational overhead for short-term prediction. Furthermore, the results obtained from the PNN are significantly more accurate than any other simpler prediction model. Long-term prediction refers to the prediction of the final destination point of a human's or an other object's motion trajectory.

### 3.1 Long-Term Prediction

Prediction methods known so far give satisfactory results for one-step ahead prediction. For the robot navigation task it would be more useful to have many-steps ahead prediction. This would give the robot sufficient time and space to perform the necessary manoeuvres to avoid obstacles and more importantly change its route towards its destination position. It is desirable for the robot to develop a behavior that will prefer routes that are not crowded and thus avoid ever getting stuck.

It is unlikely that any of the available standard prediction methods would give satisfactory results for many-steps ahead prediction, given the complexity of the movement behavior. To achieve this, it is proposed to employ a long-term prediction mechanism. The long-term prediction refers to the prediction of a human's final destination position. It is plausible to assume that humans mostly do not just move around but instead move purposively with the intention of reaching a specific location. Our approach for performing long-term prediction is based on the definition of the so-called "hot" points (HPs) in the environment, i.e. the points where people would have interest in visiting them. For example, in an office environment desks, doors and chairs are objects that people have interest in reaching them and could be defined as hot points of interest. In a museum, the points of interest can be defined as the various exhibits that are present. Moreover, other features of the environment such as the entry points, passages, etc., can be defined as points of interest. Evidently, hot points of interest convey semantic information about a workspace and hence can only be defined with respect to the particular environment.

The HPs in an environment can be defined either manually or through an automated procedure. In Sect. 3.3, an automated procedure for obtaining a map of HPs is presented. The methodology for obtaining the long-term prediction is given for both cases where the HPs are defined manually and when they are defined through a learned map. A primitive version of long term prediction with manually defined HPs was presented in Foka and Trahanias [7] evaluated only in simple simulated environments. The manually defined hot points for the FORTH main entrance hall where the experiments were conducted are shown in Fig. 1. First, the methodology for obtaining the estimated destination position of a moving object with manually defined HPs is presented and then this approach is extended to be used with a learned map of HPs.

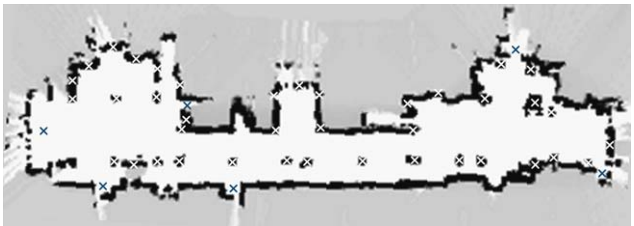


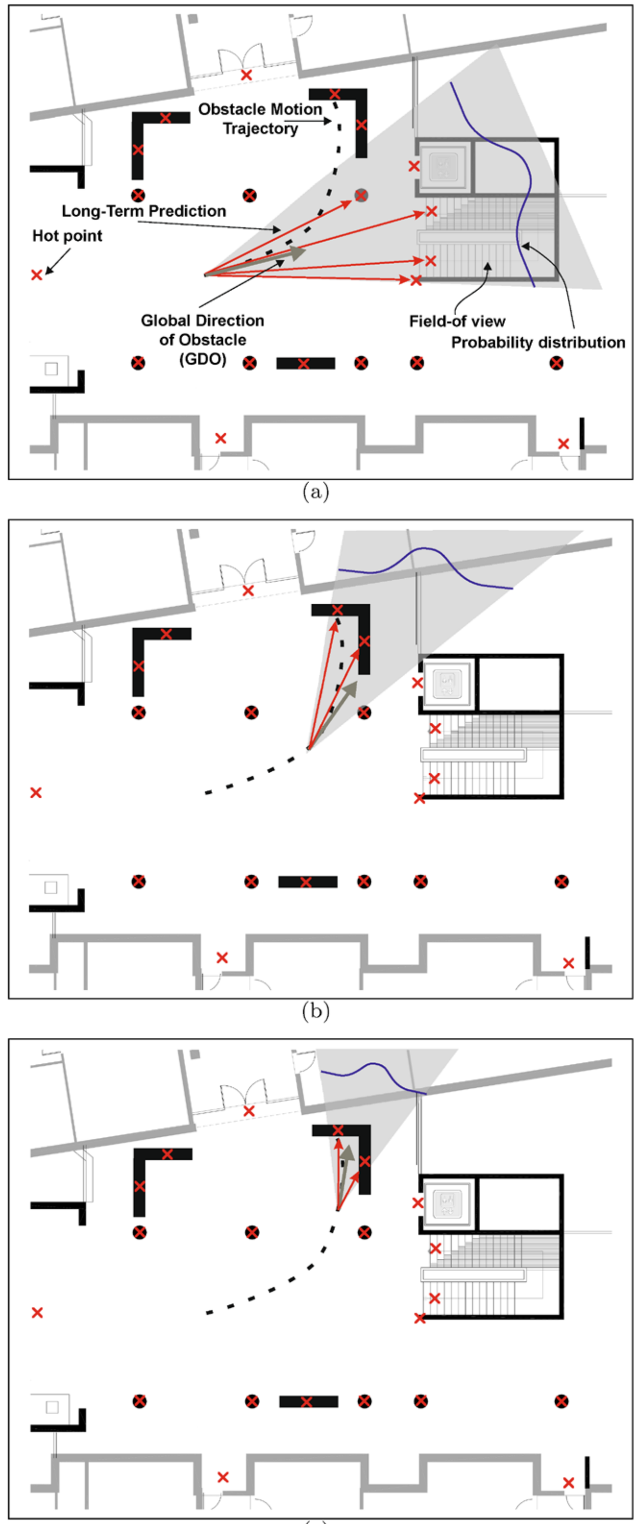
**Fig. 1** The "hot" points defined for the FORTH main entrance hall, marked with "x"

### 3.2 Estimation of a Moving Object's Destination Position

Once the points of interest of an environment are defined, then the long-term prediction refers to the prediction of which Hot Point (HP) a moving obstacle is going to approach. At each time step $t$, the tangent vector of the obstacle's positions at times $t-1, t$ and the predicted position at time $t+1$ is taken. This tangent vector essentially determines the global direction of the obstacle's motion trajectory, termed as Global Direction of Obstacle (GDO). This direction is employed to determine which HP a moving obstacle is going to approach. A HP is a candidate final destination point if it lies roughly in the direction of the evaluated tangent vector, the GDO. In order to find such HPs, we establish a field-of-view, that is an angular area centered at the GDO. HPs present in the field-of-view are possible points to be reached, with a probability $w_i$, according to a probability model. The latter is defined as a Gaussian probability distribution centered at the GDO with a standard deviation in proportion to the angular extent of the field-of-view. Thus, points of interest present in the center of the field-of-view are assigned a high probability, and points of interest present in the periphery of the field-of-view are assigned a lower probability.

With this approach, at the beginning of the obstacle's movement a multiple number of points of interest will be present in its field-of-view but as it continues its movement the number of such points is decreased and finally it converges to a single point of interest.

In Fig. 2 an example of the procedure for long-term prediction with manually defined HPs is shown. At the beginning of the obstacle's movement the long-term prediction obtained is shown in Fig. 2(a). It can be observed that at this point there are multiple candidate destination points for the obstacle's movement. However, the destination point that infers that the moving objects is going to walk down the stairs is estimated as the most probable destination point of the obstacle's movement as directed by the GDO. If the whole obstacle motion trajectory, shown in the same figure, is observed it is obvious that the estimated destination point dictated by the long-term prediction methodology is not the correct one. However, the long-term prediction obtained is utilized partially. Therefore, when a long-term prediction




(a)

(b)

(c)

**Fig. 2** An example of making long-term prediction for an object's movement

is obtained, it is utilized partially only for a short interval that is close to the obstacle's current position. Hence, the long-term prediction when utilized only partially it provides a good estimate of the future motion of the obstacle's movement. The partial utilization of the obtained long-term prediction is fully detailed in Sect. 4, where its integration to the navigation model is explained. Additionally, the long-term prediction is updated at each time step, that is a short time interval, and hence bad estimates can be corrected quickly. Consequently, as the object has advanced through its motion trajectory the long-term prediction estimate is closer to the actual destination point as shown in Fig. 2(b).

The example shown in Fig. 2 has been chosen such that to demonstrate the weakness of using manually defined HPs and hence necessitate the use of the map of HPs. When the obstacle is close to the end of its motion trajectory, as shown in Fig. 2(c), there is no HP defined in the direction dictated by the GDO. Instead, there are two HPs in the periphery of the GDO, and one of them is the actual destination point of the obstacle. However, the actual destination point will be assigned a low probability since it is located in the periphery of the GDO. The same holds for the other HP located within the field-of-view. In this scenario, if there was a map of HPs available there would be defined a whole area of HPs instead of two unique points. Specifically, in the area of the map under discussion there are two sofas where people often go there and sit. When the map of HPs is constructed the whole area covered by the sofas is determined as an interesting point instead of the two unique points in the center of the sofas that have been manually defined. Hence, we would still have obtained a long-term prediction with high probability. Additionally, the probability of each estimated destination point is not determined only according to each location within the field-of-view but also to the popularity of this point as determined through the learning procedure.

The map of HPs can also take care of the worst case scenario where there are not manually defined HPs present in the field-of-view marginally and hence there would be no estimate available.

### 3.3 Map of Hot Points

An alternative approach to defining manually the hot points of the environment is to obtain a map of hot points of the environment through an automated procedure.

The map of hot points is a probabilistic map that gives for each point of the environment the probability that this point is a hot point. This map is built off-line by a learning procedure that uses motion traces of humans operating in the environment. At every step of each collected motion trace the probabilities of the map of hot points are updated in a similar manner to this used when building an occupancy grid map of an environment from sensor readings.

Having obtained the Global Direction of Obstacle (GDO), defined in Sect. 3.2, the field-of-view is determined. The field-of-view now defines the area of cells that their probabilities of being a hot point is going to be updated. Lines

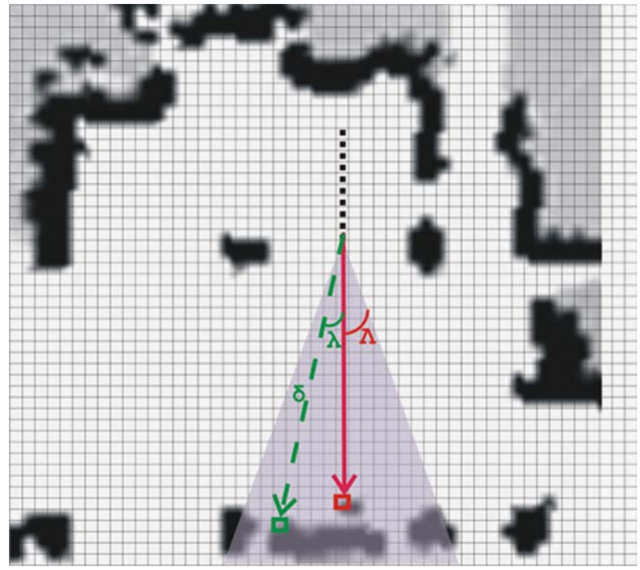



**Fig. 3** The probability assignment for possible hot points is dependent on the angular distance of the considered cell and the GDO and its distance from the obstacle's current position

from the obstacle's current position at all possible angles within the field-of-view are examined whether they intersect with a feature of the environment. This intersection point is considered as a possible hot point. The probability assignment is performed by obeying two rules:

– the smaller the angular distance, $\lambda$, of the candidate cell and the GDO the higher the probability;
– the smaller the distance, $\delta$, of the candidate cell and the obstacle's current position the higher the probability.

The probability of a cell $(i, j)$ being a hot point given an obtained GDO at time $t$, $P(H_{i,j})$, is given by:

$$P(H_{i,j}) = \frac{1}{2}\left(\frac{D-\delta}{D} + \frac{\Lambda-\lambda}{\Lambda}\right).$$

As shown in Fig. 3, the maximum angular distance from the GDO is $\Lambda$ and is defined by the size of the field-of-view. $D$ is a constant that determines the maximum distance from the object's current position allowed for a cell to be considered as a hot point. The probabilities are updated according to Bayes rule every time a certain cell is considered to be a hot point according to the motion traces.

Having obtained the probability map of hot points long-term prediction is now performed using this probability map by considering all lines within the field-of-view that intersect with a feature of the environment according to the probability assigned at the map of hot points.

The map of HPs obtained for the FORTH main entrance hall is shown in Fig. 4. The map of HPs was constructed by taking laser measurements at various times of the activity in the FORTH main entrance hall in order to obtain real motion

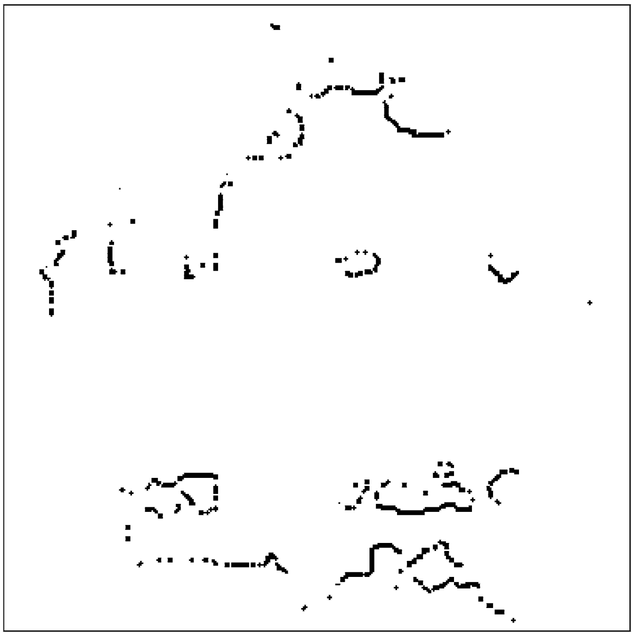


**Fig. 4** The map of "hot" points obtained for the FORTH main entrance hall. The same area of the environment in a CAD map is shown in Fig. 2

paths that will reveal the true points of interest in the environment at any time of a day. Each hot point present in the map has a probability associated with it that infers how often it is visited by humans. The same area of the environment is shown in CAD map in Fig. 2, to reveal the point correspondence to actual features. This probability is used when obtaining multiple HPs as estimated destination positions to prune away bad estimates as dictated by the procedure explained in Sect. 4 that details how the result of long-term prediction is included in the HPOMDP.

### 3.4 Motion Tracking

The methodology used for object tracking in this paper is a modification of the commonly used Kalman tracker. In the Kalman tracker, a Kalman filter is used for predicting the position of a previously detected object and hence decide if the object actually moved to its predicted position. In our approach, the Kalman filter is substituted by the short-term and long-term prediction obtained as described in the previous sections. The data association is performed by a nearest neighbor filter that is validated by the long-term prediction module.

Initially, all range measurements not belonging to objects present in the static map of the environment are regarded as moving objects. The position of each currently moving object has to be decided if it belongs to the trajectory of a previously detected object or if it is a newly detected object. Matching the positions of the currently detected moving objects with previously detected objects is performed by utilizing the short-term and long-term prediction. For each previously detected object a short-term and long-term prediction is obtained. The distance of a currently detected moving object position with the short-term prediction of a previously detected object is evaluated. The minimum evaluated distance of all previously detected objects for a specific current position is regarded to belong to this object if it is smaller than a certain threshold.

However, data association with the nearest neighbor filter is verified by the long-term prediction for the motion trajectory of the object indicated to be matched with. If the newly detected object's position belongs to the trajectory indicated by the long-term prediction with a probability higher than a certain threshold then matching is achieved. Otherwise, matching is verified with the immediate next object in distance measures as dictated by the nearest neighbor filter.

Remaining positions of currently detected objects that were not matched by the short-term or long-term prediction are regarded as new objects.

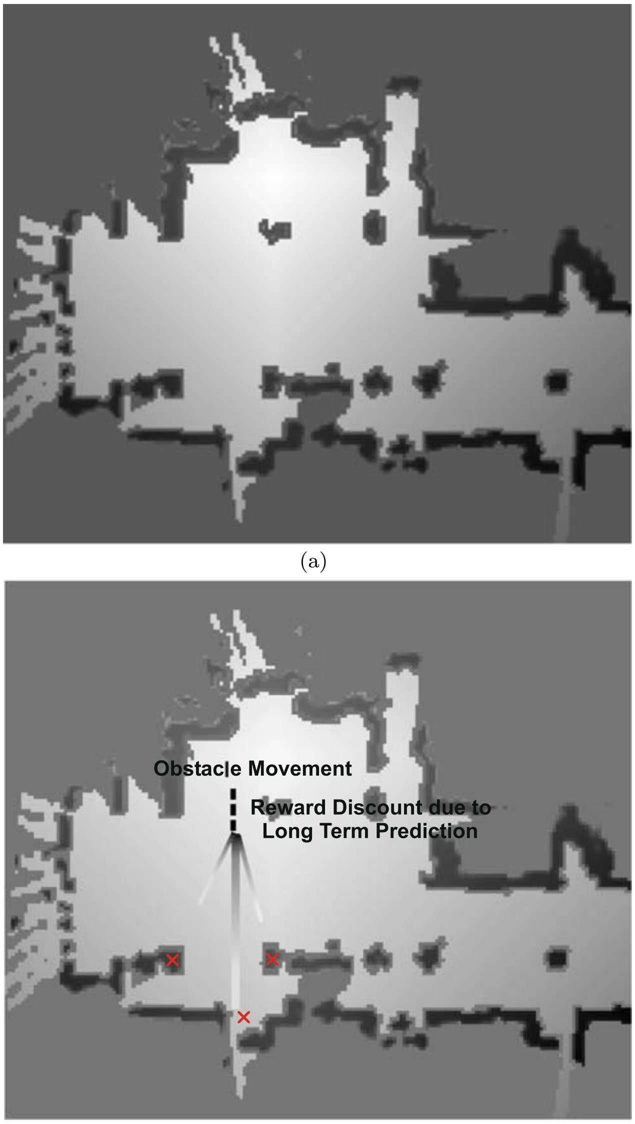
## 4 Prediction Integration into the Model

The short-term and long-term prediction are integrated in the global model by including them in the reward function of the POMDP. The reward function is built and updated at each time step, according to two *reward grid maps* (RGMs): a *static* and a *dynamic* as originally defined in [7]. The RGM is defined as a grid map of the environment in analogy with the Occupancy Grid Map (OGM). Each of the RGM cells corresponds to a specific area of the environment with the same discretization of the OGM, only that the value associated with each cell in the RGM represents the reward that the robot will receive if it ends up in a specific cell. Thus, it would be desirable that this value gives a description of the state of this square area in the environment as

– how far it is from the goal position,
– whether it is occupied by a static obstacle,
– whether it is occupied by a moving obstacle, i.e. a human or another object,
– whether it will be occupied and how soon by a moving obstacle.

The static RGM is built once by calculating the distance of each cell to the goal position and by incorporating information about cells belonging to static obstacles. Hence, it includes the first two sources of information concerning the goal position and static obstacles.

Information provided from the short-term and long-term prediction modules is included in the dynamic RGM. The inclusion of the short-term prediction is trivial and it involves zeroing the reward associated with the grid cell that is the predicted next-time position of the obstacle.




(a)

(b)

**Fig. 5** (**a**) The static and (**b**) dynamic RGM. Reward discount is performed according to the obtained long-term prediction. Long-term predictions for hot points present in the periphery of the field-of-view have low probability, $w_i$, and thus the reward discount is smaller

The long-term prediction refers to the prediction of the destination position of the obstacle's movement. Thus, the reward value of the grid cells that are in the trajectory from the obstacle's current position and its predicted destination position is discounted. Hence, the value of a cell, $p$, in the dynamic grid map, *DGM*, is given by a function

$$DGM(p) = w_i \cdot extent^{\gamma},$$

where $w_i$ is the probability that the predicted destination point will be reached, *extent* is a constant that controls how far the robot should stay from obstacles and $\gamma$ is the factor that determines how much the value of *extent* should be discounted.

The probability that the predicted destination point will be reached, $w_i$, is defined in according to the position of the predicted point within the considered obstacle's movement field-of-view as shown in Fig. 5. Thus, $w_i$ will be higher when the predicted destination point lies in the center of the field-of-view, and lower when it is at the periphery. The value of $\gamma$ depends on the estimate of how far in the future this cell will be occupied, and it takes values in the range of $[0, 1]$. For example, if a cell $p$ is to be occupied shortly in the future, $\gamma$ will be close to 0, and thus the reward assigned to cell $p$ will be small. On the other hand, if cell $p$ is to be occupied far in the future, $\gamma$ will be close to 1 and thus the reward assigned to this cell will not be significantly discounted. The $\gamma$ factor is defined such that the long-term prediction is utilized only partially for predictions that are not too far in the future. Furthermore, the $\gamma$ factor is dependent on the distance between the robot's current position and the position of predicted object position to be discounted. In this way, the movement of objects and humans that are far from the area that the robot operates currently do not affect the reward values significantly.

Superimposing the static and dynamic RGMs provides the reward function that is updated at each time step. In Fig. 5 an example of discounting the reward values for an obtained long-term prediction is shown.

## 5 POMDP Solution for Controlling the Robot's Speed

In this section, we develop a methodology of solving POMDPs that allows the robot to decide if it should increase or decrease its speed to avoid an obstacle more effectively, originally presented in [8]. Without control of its speed, the robot has to make detours or follow a suboptimal path to reach its goal in order to avoid collision with obstacles. In many cases, the robot can avoid making these suboptimal actions if it can either increase its speed to bypass the obstacle or decrease its speed to let the obstacle move away from the robot.

The speed of the robot is chosen not to be included as a characteristic of its state, as with its location and orientation. Such a choice would further increase the state space or action space. Instead, the solution of the POMDP is modified to account for the speed decision that the robot has to make.

The robot is allowed to move at three different speeds: *normal*, *fast* and *slow*.

### 5.1 Exact Solution

To solve a POMDP exactly it is required to evaluate the $\alpha$-vectors that form a policy tree as shown in Fig. 6 by the following equation as in [14]:

$$V_t^*(b) = \max_{p \in P} b \cdot \alpha_p^t$$

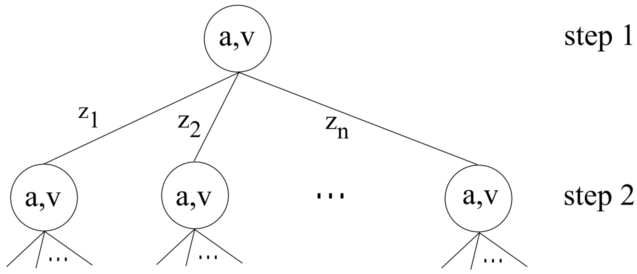



**Fig. 6** An example policy tree of a POMDP with pairs of actions and speeds

that searches all possible policy trees $P$, to maximize this value function for the belief $b$. Recall that policy trees are defined by having as nodes actions, $a$, that are connected with each possible observation, $z_i$. To decide the action to be executed as well the speed $v$ of the robot, nodes are now defined by pairs of actions and speeds.

Since policy tree nodes are composed of action-speed pairs the value function of a policy tree has now to be evaluated by considering the choice of the action as well as the speed. The value function of a policy tree is evaluated by the following equation, that has incorporated the speed decision:

$$V_t^p(b) = \sum_{s \in S} b(s) \Bigg[ R_m(s,a,v) + \gamma \sum_{z \in \mathcal{Z}} \sum_{s' \in \mathcal{S}} \mathcal{O}_m(z,s',a,v) \mathcal{T}_m(s,a,v,s') V_{t-1}^{z_p}(s') \Bigg].$$

The above equation dictates that it is required to a have a modified version of the reward, transition and observation function. However, instead of defining new POMDP functions, the notion of the *projected state* is defined that allows to use the original POMDP functions.

## 5.2 The Projected State

When the robot is in a state $s$ and performs an action $a$ with a velocity $v$, other than the *normal* velocity, it is expected to end up with a certain probability in a state $s'$. Then, the *projected* state is defined as the state $s_p$, where if the robot executes the same action $a$, from state $s_p$, with the *normal* velocity it will end up with the same probability to state $s'$. Of course, if $v$ is the *normal* velocity of the robot, then $s_p$ is $s$.

In Fig. 7 an example of determining the projected state when the robot moves with the *fast* and the *slow* speed is illustrated. For clarity reasons, in this example it is assumed that the *fast* speed is twice the *normal* speed and the *slow* speed is half the *normal* speed. In the case the robot moves at *fast* speed it forwards two grid cells since with the *normal* speed it would forward one grid cell. On the other hand, in the case the robot moves at *slow* speed it will remain in the grid cell.

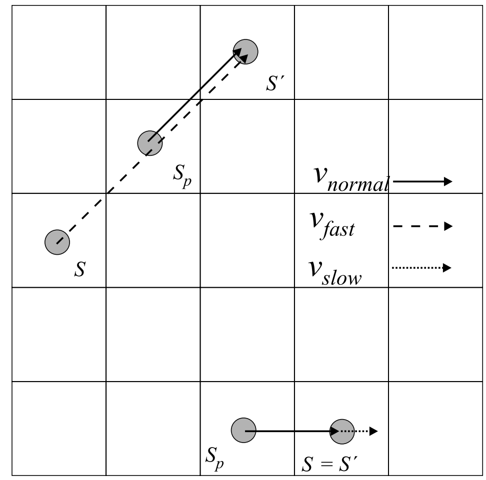



**Fig. 7** Definition of the projected state $s_p$

The projected state $s_p$ is determined geometrically by triangulation in the continuous space. The initial state $s$ is transferred to the continuous space regarding that the robot is in the center location of the grid cell represented by state $s$. Following, the vector of the action angle is constructed in analogy with the vector of the action angle for *normal* velocity. When the POMDP is built a certain *normal* velocity is considered along with a certain duration of movement of each action cycle. Therefore, a certain *normal* speed motion vector is always assumed in the initial transition probabilities. This vector is used to determine the *fast* speed motion vector or *slow* speed motion vector as the *fast* and *slow* speeds are defined as a fraction of the *normal* speed. Hence, by triangulation the resulting state $s'$ is determined. Next, the inverse procedure is used to determine the projected state $s_p$. Knowing the state $s'$ and the *normal* speed motion vector, triangulation is performed to determine the location of the projected state $s_p$. Finally, the location of the projected state $s_p$ is transferred from the continuous space to a discrete grid cell.

The determination of the projected state $s_p$ is an approximation according to the motion model of the robot used to determine the transitions between states.

## 5.3 The Modified POMDP Functions

As mentioned above, the POMDP parameters are not altered to include information about the robot velocity. Therefore, a formulation of $\mathcal{R}_m$ and $\mathcal{T}_m$ that relates them to the original $\mathcal{R}$ and $\mathcal{T}$ functions that were built considering the *normal* velocity of the robot only is required. This is achieved by utilizing the *projected* state of the robot.

Having defined the projected state, the relation of $\mathcal{R}_m$ and $\mathcal{T}_m$ to the original $\mathcal{R}$ and $\mathcal{T}$, respectively, can now be defined.

### 5.3.1 The Modified Transition Function $\mathcal{T}_m$

By the definition of the projected state $s_p$, the relation of $\mathcal{T}_m$ to $\mathcal{T}$ is straightforward, and is written as:

$$\mathcal{T}_m(s, a, v, s') = \mathcal{T}(s_p, a, s').$$

The above equation assumes that the transition probabilities for executing a certain rotation action $a$ at *normal* speed are preserved when executing the same rotation action at the *fast* or *slow* speed. This is a safe approximation since in the context in which the change of speed is used, the robot will move at a speed other than the *normal* speed for very short intervals only, i.e. only when the robot has to bypass an obstacle with the *fast* speed or allow it move way by slowing down. Furthermore, the *fast* and *slow* speed are defined as fractions of the *normal* speed that are rather small and therefore the motion behavior of the robot does not change dramatically.

### 5.3.2 The Modified Reward Function $\mathcal{R}_m$

The definition of $\mathcal{R}_m$ is not as straightforward as for $\mathcal{T}_m$. If $\mathcal{R}_m$ is simply defined as $\mathcal{R}_m(s, a, v) = \mathcal{R}(s_p, a)$, then the robot would always choose to move with the *fast* speed. This is because the *fast* speed will always get the robot closer to the goal and thus the reward that it will receive will be bigger. Instead, it is desirable that the robot moves at a different speed from its *normal* speed only if it has to avoid an obstacle. For that reason change of speed is penalized.

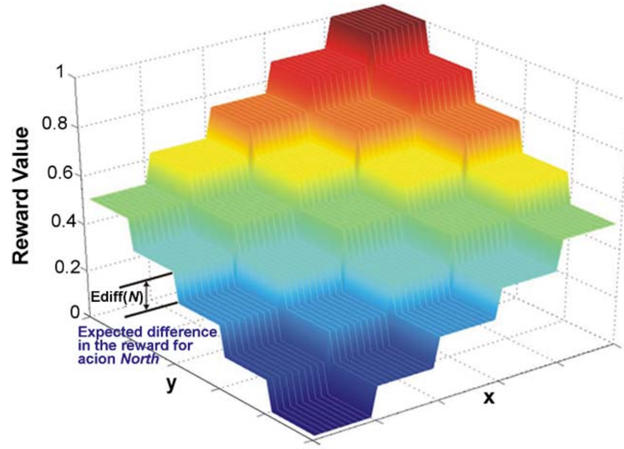
$$\mathcal{R}_m(s, a, v) = \mathcal{R}(s_p, a) - \mathit{penalty}.$$

The penalty factor for change of speed is defined in relation to the reward function to ensure that the robot has the desirable behavior. The reward function is built by calculating the distance of each grid cell to the goal position. The value assigned to each grid cell is the distance to the goal position, inverted and normalized. Therefore, the reward value of neighboring cells will always differ by a certain amount, as it can be seen in Fig. 8(a). When the static OGM is built, the average expected difference of the reward value between adjacent grid cells for every rotation action $a$, $\mathit{Ediff}(a)$, can be evaluated.

Thus, we define $\mathcal{R}_m$ as:

$$\mathcal{R}_m(s, a, v) = \mathcal{R}(s_p, a) - \alpha \cdot \frac{|v - v_n|}{v_n} \cdot \mathit{Ediff}(a),$$

where $v_n$ is the *normal* velocity of the robot and $\alpha$ is a constant that controls how preferable are the velocity changes.



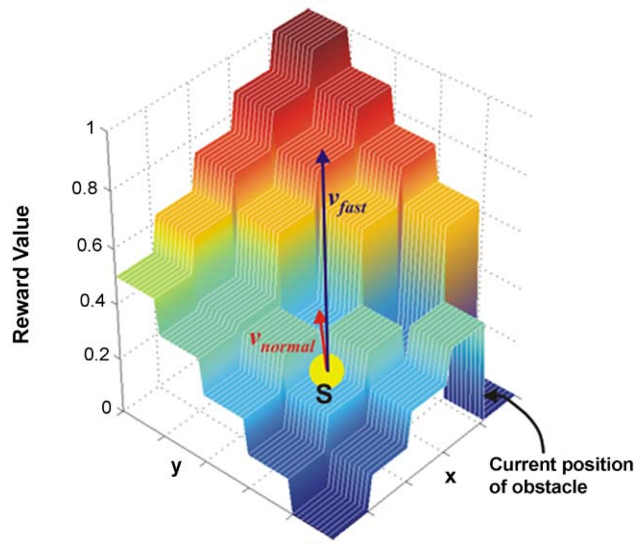




**Fig. 8** (**a**) The static OGM and (**b**) an example of the robot choosing to move with the *fast* velocity

The bigger the value of $\alpha$, the less preferable the velocity changes are. The $\frac{|v-v_n|}{v_n}$ factor, ensures that when $v$ is the *normal* velocity, the reward the robot will receive will not be penalized. It also accommodates for the effect of the difference between the *fast* or *slow* velocity with the *normal* velocity on the reward the robot receives by $\mathcal{R}(s_p, a)$. For example, when the *fast* velocity is double the *normal* velocity, then we expect that the reward the robot will receive in these two cases will differ by $\mathit{Ediff}(a)$. In the case that the *fast* velocity is triple the *normal* velocity, then $\frac{|v-v_n|}{v_n}$ will be equal to 2, as we expect that the reward the robot will receive in these two cases will differ by $2 \times \mathit{Ediff}(a)$.

When there is no obstacle in the route of the robot to its goal, the reward values will be unaltered and dependent only on the distance to the goal. Hence, in the case of the *fast* velocity, the reward the robot will receive after being penalized, will be the same with the reward the robot will receive for choosing the *normal* velocity, for $\alpha$ equal to 1.

In the case that there is an obstacle moving in the route of the robot to its goal, then the reward values of the cells that are predicted to be occupied by the obstacle in the future will be discounted. Then, the reward the robot will receive for *fast* velocity will be bigger than the reward for *normal* velocity even after it is penalized for changing speed.

An example for this case is illustrated in Fig. 8(b). It is assumed that there is an obstacle moving in the environment. For clarity reasons, in this example the obstacle is assumed to occupy a single cell. The reward value of the cell that corresponds to the obstacle's current position is set to zero. The reward value of the cells that are in the trajectory from the obstacle's current position to its predicted destination position is discounted. The original reward value of these cells can be seen in Fig. 8(a). The discount of the reward values is not the same for all cells, since it depends on the estimate of how far in the future each cell will be occupied. The robot is currently at the cell denoted with $s$. If the robot moves with the *normal* velocity, it will maximize its reward when executing one of the suboptimal actions to reach the goal since the reward for executing action *North-East* has been discounted due to the long-term prediction. When the reward values for the *fast* velocity are evaluated, it can be seen that the reward for executing action *North-East* will be the maximum. That is because the robot will end up in a state where its reward has not been discounted due to long-term prediction and even when the expected difference in the reward values for action *North-East* is deducted it will still remain bigger than all other rewards. Hence, the robot will move with the *fast* velocity and bypass the moving obstacle.

In the case of the *slow* speed, the reward the robot will receive for executing any action from the projected state will always be smaller than the reward the robot will receive for choosing the *normal* speed, when there is no obstacle. This reward will be further decreased by the penalty factor. Therefore, for the robot to choose the *slow* speed, the reward it receives for the *normal* and *fast* speed has to be smaller. This will be the case when there is an obstacle very close to the robot and did not have a long-term prediction to be able to avoid it by increasing its speed.

#### 5.3.3 *The Modified Observation Function* $\mathcal{O}_m$

The relation of $\mathcal{O}_m$ to the original observation function $\mathcal{O}$, using the projected state $s_p$, is straightforward and is written as:

$$\mathcal{O}_m(s, a, v, z) = \mathcal{O}(s_p, a, z).$$

The above definition holds due to the way the observation set has been defined in Sect. 2.1. An observation is actually the distance the robot travelled when it has executed a certain action $a$. As a result, the relation of $\mathcal{O}_m$ to the original observation function $\mathcal{O}$ is in analogy with the relation to the definition for the transition function.

### 5.4 Approximation Methods

The described methodology for controlling the robot's speed can also be applied with any of the approximation methods reviewed in [11] with the use of the modified POMDP functions.

In the case that the MLS heuristic is used the optimal value function is computed as:

$$V_t^*(s) = \max_{a\in\mathcal{A}, v\in\mathcal{V}} Q_m\left(\arg\max_{s\in\mathcal{S}}(b(s)), a, v\right)$$

where the modified $Q$-function, $Q_m$, is now defined as:

$$Q_m^t(s, a, v) = \mathcal{R}_m(s, a, v) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{T}_m(s, a, v, s')V_{t-1}(s').$$

In the case that the *voting* heuristic is used the optimal value function is given by:

$$V_t^*(s) = \max_{a\in\mathcal{A}, v\in\mathcal{V}} \sum_{s\in\mathcal{S}} b(s)\delta(\pi_{MDP(s)}, a, v)$$

where

$$\pi_{MDP(s)} = \arg \max_{a\in\mathcal{A}, v\in\mathcal{V}} Q_m(s, a, v)$$

and

$$\delta(a_i, v_i, a_j, v_j) = \begin{cases} 1, & \text{if } a_i = a_j \text{ and } v_i = v_j, \\ 0, & \text{if } a_i \neq a_j \text{ or } v_i \neq v_j. \end{cases}$$

In the same manner the modified POMDP functions can be applied to other approximation methods present in the literature.

## 6 Results

This section presents experimental results that validate the proposed approach for autonomous robot navigation. Initially the experimental configuration for the real-world environment as well as the simulated one is presented. Finally, results that demonstrate the behavior in general of the predictive navigation framework are illustrated.

### 6.1 Experimental Configurations

For testing the performance of the proposed framework, we have performed extensive tests with both real and simulated data. All real data have been assessed on Lefkos, an iRobot B21r robotic platform of our lab, while acting in various indoor areas of FORTH.

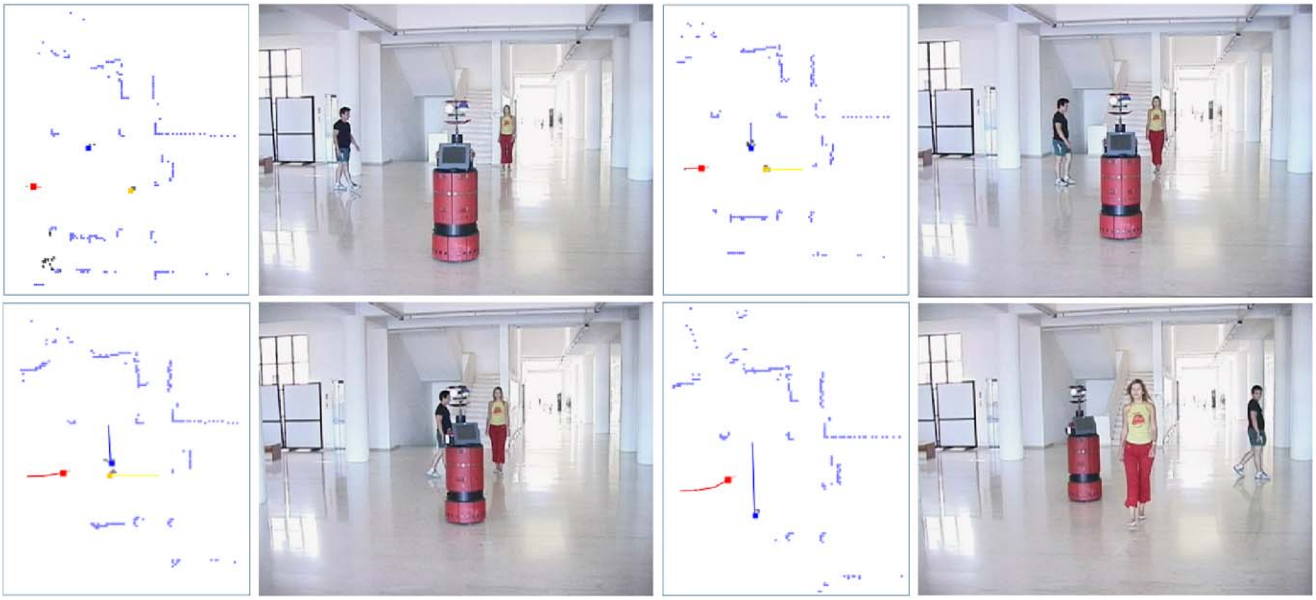


**Fig. 9** Avoiding two moving objects with a detour. The robot detects early in its movement that its path to the goal point, the stairs, will be completely blocked by the two persons moving. Hence, it starts making a detour well before it faces any of the two persons

### 6.2 Real Environment Experiments

In this section a representative set of result of the robot operating in the FORTH main entrance hall is shown. The robot was set to operate for more than 70 hours. The environment was modelled with a RN-HPOMDP with size of the set states, actions and observations being respectively $|S| = 18,411,520$, $|A| = 256$ and $|Z| = 24$. This results to grid cells of actual size 10 cm$^2$. Experiments were performed in a dynamic environment where people were moving within it. In all cases the proposed navigation model has shown a robust behavior in reaching the assigned goal points and avoiding humans or other objects. Following, sample paths the robot followed to reach its goal position by demonstrating the four main behaviors it uses to avoid obstacle are presented.

#### *6.2.1 Avoiding Obstacles with a Detour*

In this experiment we demonstrate how the robot avoids two humans moving in the environment in such a manner that they block its route to the goal position. If there were no humans or other objects moving, the robot would follow a straight path to its goal, defined for our experiments as shown in Fig. 9. In our experiment two humans are moving in the environment. One of them is moving towards the straight path that the robot would follow to reach its goal and the other one is moving in a straight direction vertical to the one the robot would follow. As shown in the figures the robot detects the moving humans and obtains the long-term prediction of their movement and hence decides to make a detour by turning. The decision the robot makes about the detour is long before the robot actually faces the moving humans and where a local obstacle avoidance method would decide to make a detour.

#### *6.2.2 Avoiding Obstacles by Following a Replanned Path*

In this experiment we show that the robot can decide to follow a completely different path from the one it would follow in a static environment in order to avoid humans moving. It is obvious from the images shown in Fig. 10, that the optimal path to reach the goal position if the robot was operating in a static environment it would be to follow a straight trajectory. In our experiment, a human was moving to block this static optimal path and the robot decided to follow a completely different path, i.e. follow a trajectory that goes behind the building's column to reach the goal position.

#### *6.2.3 Avoiding Obstacles by Increasing the Robot's Speed*

In the experiment shown in Fig. 11 the robot increases its speed to bypass the human's movement in order to reach its goal position without the need of making a detour. The human's movement is perceived by the robot from the beginning and hence it obtains the long-term prediction early enough to decide to increase its speed to bypass the human's predicted motion trajectory. When the robot has passed the human's motion trajectory its decreases its speed to normal to continue its movement.

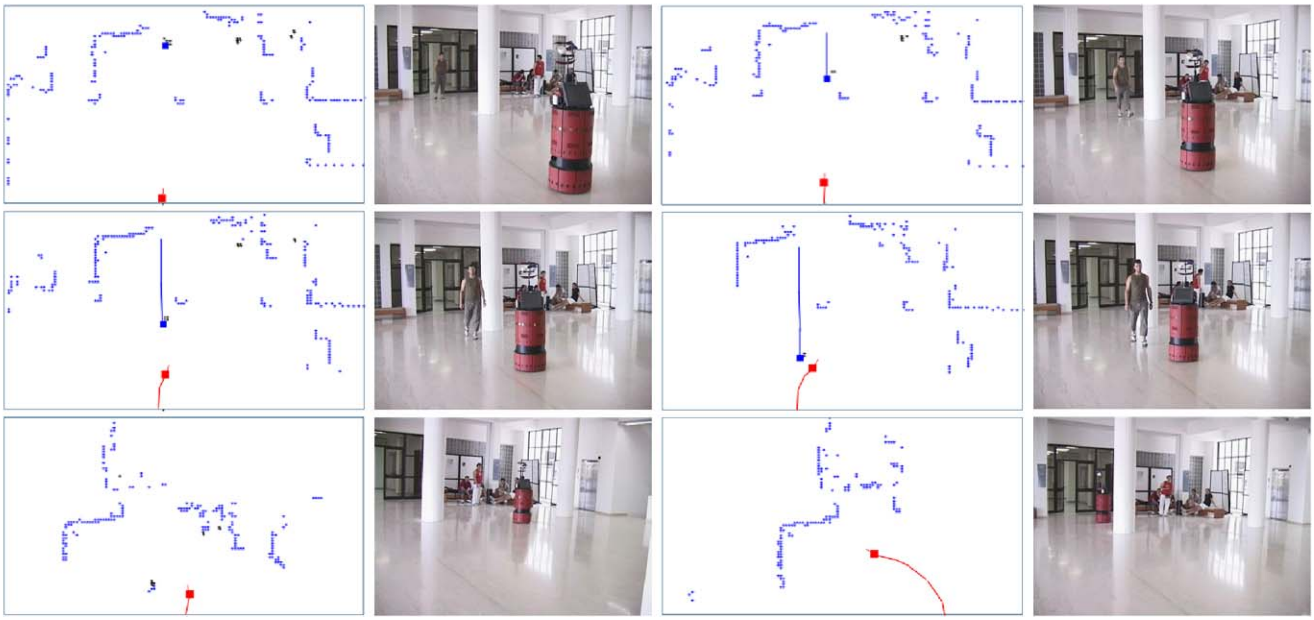


**Fig. 10** Deciding to follow a completely different path. The robot has to follow a simple straight path to reach its goal position that is the starting point of movement of the person present in the environment. However, the robot detects early enough that the person moves towards its and decides to change its path to the goal by going around the building's column

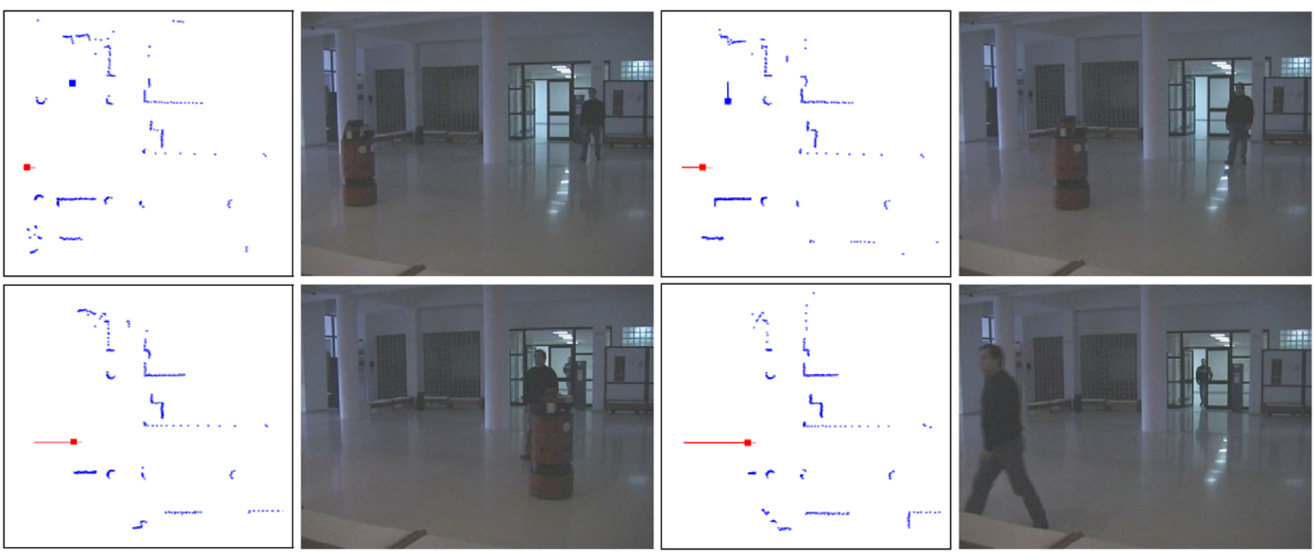


**Fig. 11** Avoiding obstacles by increasing the robot's speed. The robot has to reach the stairs by following a straight path. It detects that a person is coming towards it and decides to increase its speed since it cannot perform a detour or change its path to the goal

#### *6.2.4 Avoiding Obstacles by Decreasing the Robot's Speed*

In the experiment shown in Fig. 12 the robot cannot perceive the movement of both obstacles all the time. The person's movement denoted in the figure with the yellow square is occluded at the beginning and the robot can see it only after it has passed the building's column. However, at that point the robot cannot increase its speed to bypass this person or make a detour since there not enough space at that point. Hence, the robot decides to decrease its speed until the person blocking its way to the goal has passed away. After this point the robot reverts to its normal speed and continues its movement until it has reached its goal position.

## 7 Comparative Results

To further evaluate the appropriateness of the proposed approach a set of comparative experiments have been per-

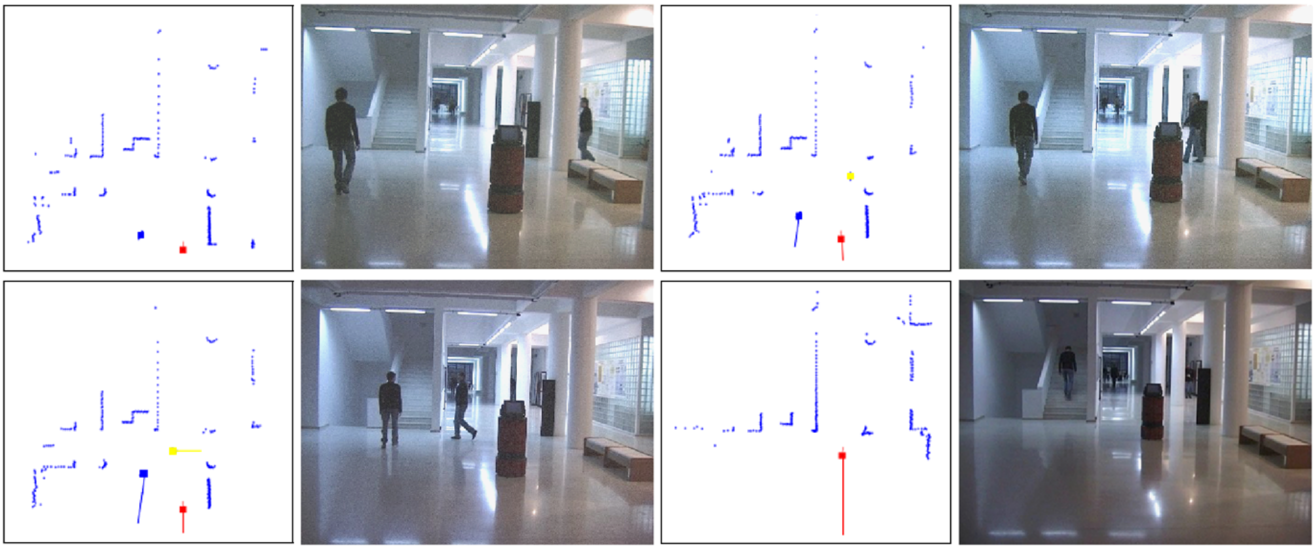


**Fig. 12** Avoiding obstacles by decreasing the robot's speed. The robot has to follow a straight path to reach a distant goal. There is a person moving in parallel with the robot that wants to reach the stairs. Another person's movement that was initially blocked by the building columns is detected that it will block its way. The robot decides to decrease its speed to allow the second person to continue its movement since it cannot perform any detours

formed. With these experiments it is aimed to provide quantitative measures of how well the proposed approach performs when it is applied in dynamic environments where the robot's movement is obstructed by humans.

The experiments were performed in the simulated environment of the FORTH main entrance hall. The robot was set to reach various goal points and each goal point was reached in the environment in the case where it is static, i.e. there is no human movement, and in the case where it is dynamic.

In the case of operating in a dynamic environment, the same experiment was performed by having from one up to five humans moving within the goal area the robot has to reach. Furthermore, in the case of having four and five humans moving within the environment, experiments were setup such us that all humans move in the goal area that the robot approaches and they were also performed with a setup where humans were moving within the goal area and also within the area that the robot would perform manoeuvres to avoid humans. In Fig. 13 an example is shown of the human motion areas defined for the comparative experiments performed. In this figure a sample configuration for a start and a goal point is shown along with a shaded area that denotes the area within the robot would choose to move to reach the goal point in a static environment. Hence, in the performed experiments humans are set up to move within this shaded area so that the robot would have to employ obstacle avoidance techniques. In the case of the experiments where there are four or five humans moving in the environment there have been used two setups. In one of them all humans move within the shaded area and in the other setup humans move

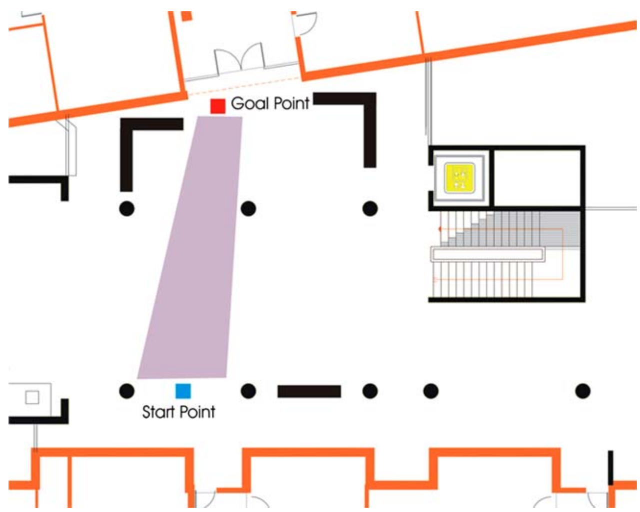



**Fig. 13** An example of how the human motion areas are defined for the comparative experiments performed

also outside this area. The second setup enables us to further evaluate the performance of the proposed approach since alternatives routes to a goal point will be also obstructed by human motion. In addition, the dynamic environment experiments were performed with and without the use of the prediction module. The experiments were performed using 50 different configurations of start and goal points of the robot.

The time required to execute the path obtained in the case where the environment is static is taken as the optimal time required to perform each experiment. This time is utilized as a reference time to compare against the time required to reach the goal point in all other types of experiments. Thus, we assume that the desirable behavior of the proposed approach is to be able to reach the goal point in a time that is as close as possible to the time the robot has taken when it operated in a static environment. This measure provides us with an insight of how efficiently and effectively the robot can avoid moving obstacles.

In Table 1, the outcome of the performed experiments is presented. In this table, a number that ranges from zero to one has been evaluated for each type of experiment, that denotes how close the performance of the proposed approach was to the optimal one, i.e. when the robot performed the experiments in a static environment. Since all the experiments were performed in a simulated environment the time taken to complete a motion path is actually the number of time steps that have been executed. The overall performance for each type of experiment has been evaluated as the average of the performance for each of the 50 distinct configurations used, as:

$$C_i = \sum_{k=1}^{50} c_{i,k}/50,$$

where $C_i$ is the overall performance of each type of experiment $i$ as shown in Table 1 and $c_i$ is the performance of each type of experiment $i$ when executing a specific configuration $k$, that is evaluated as:

$$c_{i,k} = \frac{no.\ of\ time\ steps\ in\ static\ environment\ at\ config.\ k}{no.\ of\ time\ steps\ in\ experiment\ i\ at\ config.\ k}.$$

As it can be observed in Table 1, the proposed approach when utilized with motion prediction of the human movement is superior to that when used without prediction. Furthermore, as the number of humans increases the difference in performance of the proposed approach with prediction and without prediction is more apparent. This is due to the fact that when prediction is available the robot can decide to follow a completely different path that is free instead of getting close to humans and making manoeuvres to avoid them, where as the number of humans increases it is more difficult to perform such manoeuvres. This becomes clear when the performance of the proposed approach is observed in the case of the experiments performed with four or five humans and the effect of having them moving in one area or two. When the proposed approach is utilized with prediction it can be seen that the performance is not affected dramatically since the robot can decide early enough to follow a path that will not get congested. On the other hand, when there is no prediction utilized the robot has to avoid many humans and in many cases it does not have the space to perform appropriate manoeuvres. Consequently, the performance is greatly affected when all humans move within

**Table 1** Overall performance of the proposed approach for each type of experiment performed

| Experiment | No. of humans | With prediction | Without prediction | Human areas |
|---|---|---|---|---|
| *dyn*1 | 1 | 0.947 | 0.938 | 1 |
| *dyn*2 | 2 | 0.922 | 0.873 | 1 |
| *dyn*3 | 3 | 0.859 | 0.786 | 1 |
| *dyn*4 | 4 | 0.795 | 0.691 | 1 or 2 |
| *dyn*4 | 4 | 0.823 | 0.584 | 1 |
| *dyn*4 | 4 | 0.767 | 0.798 | 2 |
| *dyn*5 | 5 | 0.788 | 0.609 | 1 or 2 |
| *dyn*5 | 5 | 0.818 | 0.590 | 1 |
| *dyn*5 | 5 | 0.758 | 0.628 | 2 |

the same area. The performance in the case of the experiments performed without prediction and with humans moving within two areas is actually equivalent to that of the experiments *dyn*2 and *dyn*3 since the robot does not change its path to the goal. Finally, overall the performance of the proposed approach with prediction utilized remains good as the number of humans increases and the deviation of the optimal time required when operating in a static environment remains small. As a result, the proposed predictive navigation approach has shown a stable performance as the number of humans present in the environment increases and is able to produce paths to reach a goal point that are not too far in time measures of the corresponding paths that the robot would execute in a static environment.

These results provide a good insight of how the proposed approach performs since there can be no direct comparison with other human motion prediction and obstacle avoidance methodologies.

## 8 Conclusions

In this paper, we have proposed a novel predictive approach to the autonomous robot navigation problem. The proposed approach is based on a Partially Observed Markov Decision Process (POMDP), that is the navigation problem is treated in a probabilistic manner. Furthermore, the POMDP model is utilized as a unified model for navigation that does not require any other external modules to perform the tasks of localization, planning and obstacle avoidance.

Probabilistic planning is performed under the predictive navigation framework where future motion prediction is employed for effective obstacle avoidance. Motion prediction refers to the estimation of the final destination of a human's or an other object's motion trajectory. This kind of prediction provides information that is utilized for effective obstacle avoidance since the robot is able to plan in awareness of predicted changes in the environment. The predictive navigation framework provides the robot the option to choose the suitable behavior among the following four for obstacle avoidance:

– execute a detour
– change completely the planned path to the goal position
– increase its speed to bypass the obstacles
– decrease its speed to let the obstacles move away.

The proposed approach is capable of performing obstacle avoidance in a unique manner as compared to standard methods that perform manoeuvres to avoid obstacles locally only when the robot gets close to them. The performance of the predictive navigation approach has been experimentally validated and the results have shown that it can provide optimal paths that are affected the least possible by other moving obstacle's movement.

## References


1. Bennewitz M, Burgard W, Cielniak G, Thrun S (2005) Learning motion patterns of people for compliant robot motion. Int J Robot Res 24(1)
2. Borenstein J, Koren Y (1991) The Vector Field Histogram—fast obstacle avoidance for mobile robots. IEEE Trans Robot Autom 7(3):278–288
3. Brock O, Khatib O (1999) High-speed navigation using the global dynamic window approach. In: Proceedings of the IEEE international conference on robotics & automation (ICRA)
4. Bruce A, Gordon G (2004) Better motion prediction for people-tracking. In: Proceedings of the IEEE international conference on robotics & automation (ICRA)
5. Chang CC, Song KT (1996) Dynamic motion planning based on real-time obstacle prediction. In: Proceedings of the IEEE international conference on robotics & automation (ICRA), vol 3, pp 2402–2407
6. Elganar A, Gupta K (1998) Motion prediction of moving objects based on autoregressive model. IEEE Trans Syst Man Cybern Part A 28(6):803–810
7. Foka A, Trahanias P (2002) Predictive autonomous robot navigation. In: Proceedings of the IEEE/RSJ international conference on intelligent robots & systems (IROS)
8. Foka A, Trahanias P (2003) Predictive control of robot velocity to avoid obstacles in dynamic environments. In: Proceedings of the IEEE/RSJ international conference on intelligent robots & systems (IROS)
9. Foka AF, Trahanias PE (2007) Real-time hierarchical POMDPS for autonomous robot navigation. Robot Auton Syst 55(7):561–571
10. Fox D, Burgard W, Thrun S (1997) The dynamic window approach to collision avoidance. IEEE Robot Autom Mag 4(1):23–33
11. Hauskrecht M (2000) Value function approximations for Partially Observable Markov Decision Processes. J Artif Intell Res 13:33–95
12. Kehtarnavaz N, Li S (1988) A collision-free navigation scheme in the presence of moving obstacles. In: CVPR'88 (IEEE computer society conference on computer vision and pattern recognition, Ann Arbor, MI, 5–9 June 1988). Computer Society Press, Washington, pp 808–813
13. Khatib O (1986) Real-time obstacle avoidance for robot manipulator and mobile robots. Int J Robot Res 5(1):90–98
14. Littman ML, Goldsmith J, Mundhenk M (1998) The computational complexity of probabilistic planning. J Artif Intell Res 9:1–36
15. Lu F, Milios E (1998) Robot pose estimation in unknown environments by matching 2d range scans. J Intell Robot Syst 18:249–275
16. Müller J, Stachniss C, Arras K, Burgard W (2009) Socially inspired motion planning for mobile robots in populated environments. In: International Conference on Cognitive Systems (CogSys), Karlsruhe, Germany, 2008
17. Nam YS, Lee BH, Kim MS (1996) View-time based moving obstacle avoidance using stochastic prediction of obstacle motion. In: Proceedings of the 1996 IEEE international conference on robotics and automation, pp 1081–1086
18. Ogren P, Leonard N (2005) A convergent dynamic window approach to obstacle avoidance. IEEE Trans Robot 21(2):188–195
19. Oliver S, Saptharishi M, Dolan J, Trebi-Ollennu A, Khosla P (2000) Multi-robot path planning by predicting structure in a dynamic environment. In: Proceedings of the first IFAC conference on mechatronic systems, vol II, pp 593–598
20. Ortega JG, Camacho EF (1996) Mobile robot navigation in a partially structured static environment, using neural predictive control. Control Eng Pract 4(12):1669–1679
21. Petti S, Fraichard T (2005) Safe motion planning in dynamic environments. In: Proceedings of the IEEE/RSJ international conference on intelligent robots & systems (IROS), pp 2210– 2215
22. Rohrmuller F, Althoff M, Wollherr D, Buss M (2008) Probabilistic mapping of dynamic obstacles using Markov chains for replanning in dynamic environments. In: IROS, pp 2504–2510
23. Stachniss C, Burgard W (2002) An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In: Proceedings of the IEEE/RSJ international conference on intelligent robots & systems (IROS)
24. Tadokoro S, Ishikawa Y, Takebe T, Takamori T (1993) Stochastic prediction of human motion and control of robots in the service of human. In: Proceedings of the 1993 IEEE international conference on systems, man and cybernetics, vol 1, pp 503–508
25. Tadokoro S, Hayashi M, Manabe Y (1995) On motion planning of mobile robots which coexist and cooperate with human. In: Proceedings of the 1995 IEEE/RSJ international conference on intelligent robots and systems, pp 518–523
26. Vasquez D, Fraichard T (2004) Motion prediction for moving objects: a statistical approach. In: Proceedings of the IEEE international conference on robotics & automation (ICRA)
27. Yung NHC, Ye C (1998) Avoidance of moving obstacles through behavior fusion and motion prediction. In: IEEE international conference on systems, man and cybernetics, pp 3424–3429
28. Zhu Q (1991) Hidden Markov Model for dynamic obstacle avoidance of mobile robot navigation. IEEE Trans Robot Autom 7(3):390–397


**Amalia F. Foka** was born in Patras, Greece in 1976. She received the B.Eng. degree in Computer Systems Engineering and the M.Sc. degree in Advanced Control from UMIST, UK in 1998 and 1999 respectively. She received the Ph.D. degree in 2005 from the Department of Computer Science, University of Crete, Greece. She currently is a visiting lecturer at the Department of Computer Engineering and Informatics, University of Patras, Greece. Her research interests include robotics and artificial intelligence.

**Panos E. Trahanias** received his Ph.D. in Computer Science from the National Technical University of Athens, Greece. Currently he is a Professor with the University of Crete, Greece and the Foundation for Research and Technology—Hellas (FORTH). From 1991 to 1993 he was

with the Department of Electrical and Computer Engineering, University of Toronto, Canada, as a Research Associate. He has participated in many RTD programs in image analysis at the University of Toronto and has been a consultant to SPAR Aerospace Ltd., Toronto. Since 1993 he is with the University of Crete and FORTH; currently, he is the Director of Graduate Studies, Department of Computer Science, University of Crete, and the Head of the Computational Vision and Robotics Laboratory at FORTH, where he is engaged in research and RTD projects in autonomous mobile platforms, sensory technologies, computational vision, mixed realities and human-robot interaction. Professor Trahanias has extensive experience in the execution and co-ordination of large research projects. Moreover, his work has been published extensively in scientific journals and conferences. He has participated in the Programme Committees of numerous International Conferences; he has been the General Chair of Computer Graphics International 2004, and will be General Co-Chair of Eurographics 2008 and the European Conference on Computer Vision 2010.