Real-Time Hierarchical POMDPs for Autonomous Robot Navigation

Amalia Foka Panos Trahanias

Institute of Computer Science
Foundation for Research and Technology – Hellas (FORTH)
P.O.Box 1385, Heraklion, 711 10 Crete, Greece
and
Department of Computer Science, University of Crete
P.O.Box 1470, Heraklion, 714 09 Crete, Greece
e-mail:{foka,trahania}@ics.forth.gr

Abstract

This paper proposes a novel hierarchical representation of POMDPs that for the first time is amenable to real-time solution. It will be referred to in this paper as the Robot Navigation - Hierarchical POMDP (RN-HPOMDP). The RN-HPOMDP is utilized as a unified framework for autonomous robot navigation in dynamic environments. As such, it is used for localization, planning and local obstacle avoidance. Hence, the RN-HPOMDP decides at each time step the actions the robot should execute, without the intervention of any other external module. Our approach employs state space and action space hierarchy, and can effectively model large environments at a fine resolution. Finally, the notion of the *reference* POMDP, that holds all the information regarding motion and sensor uncertainty is introduced, which makes our hierarchical structure memory efficient and enables fast learning. The RN-HPOMDP has been tested extensively in a real-world environment.

1 Introduction

The autonomous robot navigation problem has been studied thoroughly by the robotics research community over the last years. The navigation problem involves the three main tasks of mapping, localization and path planning. Incorporating uncertainty in methods for navigation is crucial to their performance due to the the robot motion uncertainty and sensor uncertainty. Hence, probabilistic methods dominate the proposed approaches present in the literature. However, probabilistic methods, that integrate uncertainty, for motion planning have not been well studied until now in contrast to probabilistic methods for mapping and localization. Contemporary methods for robot motion planning [6] do not take into account the involved uncertainty. The probabilistic path planning methods present in the literature so far are dominated by methods based on Partially Observable Markov Decision Processes (POMDPs) [8, 13, 16, 15, 10, 9, 14] but they are mainly utilized only as high level path planners due to the computational complexity involved and require a lower level path planner, that most commonly is not probabilistic, to drive the robot between intermediate points and also perform obstacle avoidance. In this paper a Hierarchical POMDP (HPOMDP) is employed that facilitates probabilistic navigation where the probabilistic nature of POMDPs is exploited in all aspects of navigation tasks. The proposed HPOMDP solves in a unified manner the navigation tasks of localization, path planning and obstacle avoidance.

POMDPs provide the mathematical framework for probabilistic planning. POMDPs model the hidden state of the robot that is not completely observable and maintain a belief distribution of the robot's state. Planning with POMDPs is performed according to the belief distribution. Therefore, actions dictated by a POMDP drive the robot to its goal but also implicitly reduce the uncertainty of its belief.

Although POMDPs successfully meet their purpose of use, they are intractable to solve with exact methods when applied to real-world environments modelled at a fine resolution. Many approximation methods for solving POMDPs are present in the literature that have also been applied to robotics problems [1, 12, 4, 16, 9, 14]. The approximation methods presented in the literature so far can only deal with problems where the size of the state space is limited to at most a few thousands states. As a result, approximation methods known so far cannot model large real world environments at a fine resolution and hence POMDPs are used as high level mission planners. Furthermore, even when POMDPs are able to model large environments [12] they have to be amenable to real time solution to be applied as unified navigation model that can perform the navigation tasks of localization, path planning and obstacle avoidance since the POMDP will have to be solved at each time step.

In this paper, we propose a hierarchical representation of POMDPs for autonomous robot navigation, termed as the Robot Navigation-HPOMDP (RN-HPOMDP) that can efficiently model large real world environments at a fine resolution. The RN-HPOMDP provides a representation that for the first time enables real-time POMDP solution even when the state space size is extremely large. In effect, the RN-HPOMDP is solved on-line at each time step and decides the actual actions the robot performs, without the intervention of any other external modules. Hence, the RN-HPOMDP is utilized as a unified framework for the autonomous robot navigation problem, that integrates the modules for localization, planning and local obstacle avoidance.

Two other HPOMDP approaches are currently present in the literature that employ either state space hierarchy [15], applied as a high level mission planner, or action and state space hierarchy [11], applied for high level robot control and dialogue management. Independently and concurrently with these works we have come up with the RN-HPOMDP[1] that applies both state space and action space hierarchy. It is specifically designed for the autonomous robot navigation problem and offers specific advantages over the two approaches mentioned above. A comparison between the RN-HPOMDP and the mentioned approaches can be found in Section 4.

Experimental results have shown the applicability of the RN-HPOMDP for autonomous robot navigation in large real world and dynamic environments where humans and moving objects are effectively avoided and the robot follows optimal paths to reach its destination.

2 Partially Observable Markov Decision Processes (POMDPs)

POMDPs are a model for planning under uncertainty [5]. A POMDP is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \mathcal{O} \rangle$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$, is a finite set of actions, $\mathcal{T}$ is the *state transition function*, $\mathcal{Z}$, is a finite set of observations, $\mathcal{O}$ is the *observation function* and $\mathcal{R}$ is the *reward function*, giving the expected immediate reward gained by the agent for taking each action in each state. The robot maintains a belief distribution at all times, b_t, over the set of environment states, $\mathcal{S}$.

Each state s represents the location (x, y) of the robot and its orientation θ, termed as the *orientation angle*. The set of actions is composed of all possible rotation actions a from 0° to 360° that are termed as *action angles*. The set of observations is instantiated in our approach as the output of the *iterative dual correspondence* (IDC) [7] algorithm for scan matching. Therefore, the output of the IDC algorithm, that is the dx, dy and $d\theta$ from the estimated location provided, is discretized and the observation set if formed.

The RN-HPOMDP provides the actual actions that are executed by the robot and also carries out obstacle avoidance for moving objects. Therefore, the reward function is built and updated at each time step according to two reward grid maps (RGMs): a *static* and a *dynamic*. 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 the 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 on whether there are objects moving within it or other unmapped objects. Superimposing the static and dynamic RGMs provides the reward function that is updated at each time step. The use of the static and dynamic RGM alleviates the need for modelling moving objects as observations.

3 The RN-HPOMDP

The RN-HPOMDP is built through an automated procedure using as input a map of the environment and the desired discretization of the state and action space. The map of the environment can be either a probabilistic grid map obtained at the desired discretization or a CAD map.

The RN-HPOMDP structure is built by decomposing a POMDP with large state and action space into multiple POMDPs with significantly smaller state and action spaces. The process of building the hierarchical structure is performed in a top-down approach. The number of levels of the hierarchical structure is determined by the desired discretization of the action angles or the orientation angles, since their discretization is the same in the RN-HPOMDP. Thus, if the desired discretization of the action angles or the orientation angles is ϕ, the number of levels of the RN-HPOMDP, L, will be $L = \log_2(90^\circ/\phi) + 1$.

The number of levels of the RN-HPOMDP in conjunction with the desired discretization of the state space affects the size of the top-level POMDP and in effect the performance of the RN-HPOMDP regarding the time complexity of solving it.

[1] Preliminary versions of the RN-HPOMDP are presented in [2, 3]

Table 1: Properties of the RN-HPOMDP with L levels.

	Top Level	**Intermediate Level l**
No of POMDPs	1	$\lvert\mathcal{A}^{l-1}\rvert \times \lvert\mathcal{S}^{l-1}\rvert$
$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{S}^0\rvert/2^{2(L-1)}$	20 except when $l = L$ where $\lvert\mathcal{S}\rvert = 5 \times (2+r)^2$
θ and a range	$[0°, 360°]$	$(\theta_p, \alpha_p) \pm (90°/2^{l-1})$
θ and a resolution	$90°$	$90°/2^{l-1}$
$\lvert\mathcal{A}\rvert$	4	5

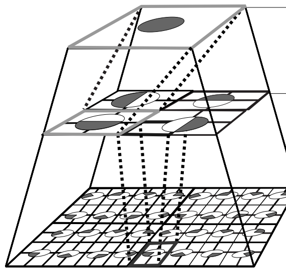


Figure 1: State space hierarchy decomposition. Orientation angle range is denoted by the shaded region of the circles for each POMDP.

The top level of the hierarchical structure is composed of a single POMDP with very coarse resolution so it can represent the whole environment in a small number of states. The grid resolution of the top level states is equal to $d \times 2^{L-1}$, where d is the desired discretization of the corresponding flat POMDP. The orientation angle of the robot and the action angles are also discretized in a very coarse resolution of $90°$ and thus represent the basic four directions $[0°, 90°, 180°, 270°]$.

To summarize, the top-level is always composed of a single POMDP with predefined discretization of the orientation and action angles at $90°$. The state space size of the top-level POMDP is variable and dependent on the discretization of the corresponding flat POMDP and the number of levels of the hierarchical structure. Hence, the number of levels of the RN-HPOMDP, L, should be such that it ensures that the size of the top-level POMDP remains small.

Subsequent levels of the RN-HPOMDP are composed of multiple POMDPs, each one representing a small area of the environment and a specific range of orientation angles. The actions of an intermediate level POMDP are also a subset of the actions of the corresponding flat POMDP.

In detail, each state of the top level POMDP corresponds to a POMDP at the immediate next level, as we go down the hierarchical structure. A POMDP at an intermediate level l, has states that represent grid locations of the environment at a resolution of $d \times 2^{(L-l)}$. Thus, by going down the hierarchical structure the grid resolution of a level's POMDPs is always twice the resolution of the previous level. Therefore, when a top level state, that corresponds to a specific grid location, is decomposed it will be represented in the immediate next level POMDP by an area of 2×2 cells with double grid resolution than the top level's grid resolution.

Going down the hierarchical structure, the resolution of the orientation angle is also doubled. Since the resolution of the orientation angle is increased as we go down the hierarchical structure, the whole range of possible orientation angles, $[0°, 360°]$, cannot be represented in every intermediate level POMDPs. This would dramatically increase the size of the state space and therefore we choose to have many POMDPs that represent the same grid location but with a different range of orientation angles.

The range of orientation angles that is represented within each intermediate level POMDP is expressed in terms of the orientation angle, θ_p, of the previous level state that is decomposed, and is equal to $\left[\theta_p \pm \left(90°/2^{l-2}\right)\right]$, where l is the current intermediate level. By the above expression of the range of orientation angles, every intermediate level POMDP will always have five distinct orientation angles. For example, if the state of the top level POMDP, $l = 1$, has orientation angle $\theta_p = 90°$, the range of orientation angles at the next level, $l = 2$, will be equal to $[0°, 180°]$. As mentioned earlier the angle resolution of the top level is always equal to $90°$ and the next level will have double resolution, i.e. $45°$. Therefore, the range of orientation angles $[0°, 180°]$ will be represented by five distinct orientation angles. Consequently, the size of the state space for every intermediate level POMDP is constant and equal to 20, since it always has five possible orientation angles and it represents a 2×2 area of grid locations.

Action angles are decomposed from the top level POMDP to the next intermediate level in the same manner as with the orientation angles. The resolution of the action angles at each level is the same as the resolution of the orientation angle. Hence, it is equal to $90°/2^{l-1}$. As a result, a top level state is also decomposed into multiple POMDPs, each one with a different range of orientation angles but also with a different range of action angles. The range of an action set is equal to $\left[a_p \pm \left(90°/2^{l-2}\right)\right]$, where a_p is the previous level action and l is the current intermediate level. The action angles set is also always composed of five distinct actions according to the above expression.

The procedure described is used to built all intermediate levels of the hierarchical structure until the bottom level is reached. Bottom level POMDPs' state and action space is discretized at the desired resolution as a flat POMDP would be discretized. The bottom level is composed of multiple POMDPs having the same properties as all other intermediate levels' POMDPs, only that

```
while not reached the goal state
    compressTopBelief(top level)
    a_p = solveTopLevel(top level)
    for l = 2 to L
        whichPOMDP = selectPOMDP(l, a_p)
        compressBelief(l, whichPOMDP)
        a_p = solveLevel(l, whichPOMDP)
    end
    executeAction(a_p)
    z = getObservation()
    belief_{l,whichPOMDP} = updateBelief(l, whichPOMDP, a_p, z)
    updateFullBelief(belief_{l,whichPOMDP}, l, whichPOMDP)
end
```

Table 2: RN-HPOMDP planning.

the grid location the bottom level POMDPs represent is overlapping by a region r. Overlapping regions are required to be able to solve the bottom level POMDPs for border location states.

The properties of the RN-HPOMDP are summarized in Table 1.

3.1 Planning with the RN-HPOMDP

Solving the RN-HPOMDP to obtain the action the robot should perform, involves solving a POMDP at each level. The intuition of the RN-HPOMDP solution is to obtain at first a coarse path that the robot should follow to reach a goal position, and then refine this path at each subsequent level in the area that the robot's current position lies. In Table 2 the algorithm for the RN-HPOMDP planning procedure is detailed.

During the RN-HPOMDP planning procedure the belief distribution of the corresponding flat POMDP is maintained at all times. This distribution will be denoted as the *full belief*. Before solving any POMDP at a level, the *full belief* is compressed, by the functions `compressTopBelief()` and `compressBelief()`, to obtain the belief distribution of the POMDP to be solved. Belief compression is performed according to the state abstraction present at each level of the RN-HPOMDP structure. Therefore, the belief assigned to an abstract state will correspond to the average belief of all the corresponding flat POMDP states that has integrated. The belief distribution obtained for any POMDP is normalized before solving it.

The top level POMDP is solved, by the function `solveTopLevel()`, at an infinite horizon, until the goal state is reached. The immediate abstract action to be executed, a_p, as dictated by the top level POMDP solution determines which POMDP at the immediate next level of the hierarchical structure will be solved to obtain a new refined abstract action.

The POMDP to be solved at the next level is determined by the function `selectPOMDP()`. This function searches a level l for the POMDP that satisfies the following two criteria:

- The zero moment of the full belief distribution over the area that is defined by the candidate POMDP states is maximum.
- The set of actions of the candidate POMDP contains an action that has minimum distance from the the previous level solution's action, a_p.

The structure of the RN-HPOMDP, as described in Section 3, ensures that when solving an intermediate level POMDP the action obtained from the previous level will be refined to a new action since the action subset range is equal to $\left[a_p \pm \left(90^\circ/2^{l-2}\right)\right]$. Therefore the solution of an intermediate level POMDP is bounded according to the previous level solution.

The described procedure continues until the bottom level is reached where an abstract action will be refined to an actual action, that is the action the robot will perform.

When the robot executes the action obtained by the bottom level POMDP solution, an observation, z, is obtained and the belief distribution of this bottom level POMDP is updated by `updateBelief()`. Bottom level POMDPs are composed of actual states and actions, i.e. subsets of states and actions that compose the corresponding flat POMDP. Hence, updating the belief of a bottom level POMDP amounts to updating a specific region of the *full belief*. Therefore, the belief distribution of the bottom level POMDP that was solved is transferred to the *full belief* by the function `updateFullBelief()`.

All POMDPs at all levels are solved in our current implementation using the Voting heuristic, that is an MDP-based approximation method. However, this is not an inherent feature of the RN-HPOMDP structure, as any other POMDP solution method can be used. Furthermore, the POMDP solution method used can also be different for each level of the hierarchical structure.

3.2 Complexity Analysis of the RN-HPOMDP Solution

In the complexity analysis that follows, computation time complexities are evaluated for the RN-HPOMDP solution using exact methods and heuristics.

The flat POMDP solution has time complexity, for a single step, $\mathbf{O}\left(|\mathcal{S}|^2|\mathcal{A}|\right)$ when solved with the MLS or Voting heuristic. Referring to Table 1, where the properties of the RN-HPOMDP structure are detailed, the solution of the top level POMDP requires $\mathbf{O}\left(\left(|\mathcal{S}|/2^{2(L-1)}\right)^2\right)$ time, where L is the number of levels of the hierarchical structure.

The solution of all intermediate levels POMDPs requires $\mathbf{O}(C_1)$ time, since the size of the state space and action space is constant and predefined. The bottom level POMDP solution is $\mathbf{O}(C_2)$, since the state space and action space is again constant and predefined. Therefore, the total time required to solve the RN-HPOMDP reduces to actually the complexity of the top level POMDP. The top-level POMDP state and action space size can remain small regardless of the size of the whole environment by increasing the number of levels, L, of the hierarchical structure.

When solving a flat POMDP exactly for a single step in time t, the time complexity is $\mathbf{O}\left(|\mathcal{S}|^2|\mathcal{A}||\Gamma_{t-1}|^{|\mathcal{Z}|}\right)$, where $|\Gamma_{t-1}|$ is the number of linear components required to represent the value function at time $t-1$. The size of Γ at any time t is equal to $|\Gamma_t| = |\mathcal{A}||\Gamma_{t-1}|^{|\mathcal{Z}|}$.

The time complexity and size of the RN-HPOMDP when solved exactly is $\mathbf{O}\left(\left(\left(|\mathcal{S}|/2^{2(L-1)}\right)\right)^2|\Gamma_{t-1}|^{|\mathcal{Z}|}\right)$ and $|\Gamma_t| = |\Gamma_{t-1}|^{|\mathcal{Z}|}$, respectively.

Apart from the notable reduction in computation time due to the reduced size of the state and action space, it should be noted that the above mentioned times are for a single time step. The infinite horizon solution of a flat POMDP would require these computations to be repeated for a number N of time steps until the goal point is reached, that is dependent on the number of states of the flat POMDP, $|\mathcal{S}|$. In the RN-HPOMDP case, only the top level POMDP is solved at an infinite horizon, and the number of time steps N' until the goal point is reached, is now dependent on the number of states of the top level POMDP, $(|\mathcal{S}|/2^{2(L-1)})$.

From this short complexity analysis, we may conclude that the particular POMDP formulation in our approach takes care of the "curse of dimensionality" [5] and also the "curse of history" [9].

3.3 The Reference POMDP

The RN-HPOMDP described in the previous section, can cope with the computational time requirements but cannot address the memory requirements. A flat POMDP would require to hold a transition matrix of size $(|\mathcal{S}|^2 \times |\mathcal{A}|)$ and an observation matrix of size $(|\mathcal{S}| \times |\mathcal{A}| \times |Z|)$.

The RN-HPOMDP structure requires to hold the transition and observation matrices for all the POMDPs at all levels. As it can be seen in Table 1 the number of POMDPs at each level is large and dependent on the size of action space and state space. Consequently, even thought each POMDP's observation and translation matrix is small the total memory requirements would be extremely large. The RN-HPOMDP has larger memory requirements than the flat POMDP, although the flat POMDP memory requirements are already very hard to manage for large environments. For this reason, the notion of the *reference* POMDP (rPOMDP) is introduced.

The transition and observation matrices hold probabilities that carry information regarding the motion and sensor uncertainty. In the formulation of the autonomous robot navigation problem with POMDPs, as described in Section 2, transition and observation probabilities for a given action, a, and an observation, z, depend actually only on the relative position and orientation of the robot. This is due to the design choice to model the environment structure and state in the reward function instead of the transition and observation matrices as commonly used in the POMDP literature. Therefore, the transition and observation probabilities are dependent only on the robot motion model.

The transition probability of a robot from a state s to a new state s', when it has performed an action a is only dependent on the action a. Therefore when the robot is executing an action a, the transition probability will be the same for any state s when the resulting state s' is defined relatively to the initial state s.

The probability that the robot observes a feature z, when it is in a state s and performs an action a, can also be defined in the same manner as with the transition probabilities, since the set of features $\mathcal{Z}$ has been defined in Section 2 to be the result of the scan matching algorithm when feeded with a reference laser scan and the actual scan the robot perceived. Therefore, perceived features are dependent on the motion of the robot, i.e. the action a it performed.

The rPOMDP is built by defining a very small state space, defined as an $R \times R$ square grid (in our implementation $R = 7$) representing a subset of possible locations of the robot and all the orientation angles of the robot that would be assigned in the flat POMDP. The size of the grid that defines the possible robot locations in the rPOMDP is established by determining the largest possible location transition when a single action a is executed. This is due to the fact that the rPOMDP conveys the transition and observation probabilities based only on the actual robot motion independently of the exact location of the robot in the environment. The center location of the state space represents the invariant state s_r of the robot. The action and observation spaces are defined in the same manner they would be defined for the original POMDP. This rPOMDP requires to hold transition and observation matrices of size $((R \times 2^{2+L})^2 \times |A|)$ and $((R \times 2^{2+L})^2 \times |A| \times |Z|)$, respectively. The size of the matrices is only dependent on the size of the set of actions and observations and the number of levels of hierarchy, L, since the number of levels defines the discretization of the robot's orientation angle.

Transition and observation probabilities for each POMDP in the hierarchical structure are obtained by translating and rotating the reference transition and observation probability distributions over the current POMDP state space. The transfer of probabilities is performed on-line while a POMDP is solved or the robot's belief is updated.

The transition probability for any POMDP of the hierarchical structure, $\mathcal{T}(s, s', a)$, is equivalent to the transition probability of the rPOMDP, $\mathcal{T}_r(s_r, s'_r, a_r)$. The reference result state, s'_r, is determined by the following equation:

$$\begin{bmatrix} x'_r \\ y'_r \\ f'_r \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \\ f_r \end{bmatrix} + \begin{bmatrix} x' - x \\ y' - y \\ f' - f \end{bmatrix},$$

where, the states s, s', s_r and s'_r are decomposed to the location and orientation triplets $(x, y, f), (x', y', f'), (x_r, y_r, f_r)$ and (x'_r, y'_r, f'_r), respectively. The reference action is determined by $a_r = a + f - f_r$.

In the same manner, the observation probability for any POMDP of the hierarchical structure, $\mathcal{O}(s, z, a)$, is equivalent to the observation probability of the rPOMDP, $\mathcal{O}_r(s_r, z_r, a_r)$. The reference observation, z'_r, is now determined as:

$$\begin{bmatrix} dx_r \\ dy_r \\ df_r \end{bmatrix} = \begin{bmatrix} d\cos(f_r + a_r) \\ d\sin(f_r + a_r) \\ df \end{bmatrix},$$

where the observations z and z_r are decomposed into (dx, dy, df) and (dx_r, dy_r, df_r), respectively, as observations are defined as the position and angle difference between laser scans, and d is the distance $d = \sqrt{dx^2 + dy^2}$.

4 Comparison with other HPOMDPs

4.1 Comparison with the Theocharous approach

The Theocharous [15] approach uses a topological map of the environment where state abstraction in high levels of the HPOMDP, has a physical meaning based on the environment. Thus, abstract states are manually defined such that they represent a corridor or a junction.

The Theocharous HPOMDP has been used as a high-level planner where the POMDP is solved once to obtain the shortest path to the goal position. As a result, the state space resolution is set to $2m^2$ and the action space is discritized at a resolution of $90°$.

The Theocharous approach, uses the MLS heuristic and has time complexity[2] between $\mathbf{O}(|\mathcal{S}|^{\frac{2}{d}} N |\mathcal{A}|)$ and $\mathbf{O}(|\mathcal{S}|^2 |\mathcal{A}|)$, based on how well the HPOMDP was constructed. The time required to solve the proposed RN-HPOMDP, with the MLS heuristic, is $\mathbf{O}((|\mathcal{S}|/2^{2(L-1)})^2)$, hence the complexity reduction of our approach is significantly greater and also is not dependent on any quality measure of the hierarchical structure.

4.2 Comparison with the Pineau approach

In the Pineau HPOMDP approach [11], actions are grouped into abstract actions called subtasks. Subtasks are defined manually and according to them state abstraction is performed automatically. States that have the same reward value for executing any action that belongs to a predefined subtask are clustered. Observation abstraction is performed by eliminating the observations that have zero probability over all state clusters for that actions belonging to a specific subtask.

[2] d is the depth of the tree and N is the maximum number of entry states for an abstract state.

Planning with the Pineau HPOMDP involves solving the POMDP defined for each action subtask, that are solved using the exact POMDP solution method.

The HPOMDP proposed by Pineau does not have a guaranteed reduction of the action space and state space since it is dependent on the action abstraction that is defined manually. The authors have performed experiments (real and simulated) only for problems of high level behavior control. Hence it is not clear whether their approach of state abstraction could be applied to the problem of the autonomous robot navigation in the context that we have defined or more importantly if it would perform as well as the RN-HPOMDP does, since it has a guaranteed reduction of the state space that is equal to $\left(|\mathcal{S}|/2^{2(L-1)}\right)$. On the other hand, the authors in [11] do not state how well their approach performs in terms of state space abstraction.

4.3 Approximation methods for solving flat POMDPs

Reference [4] presents a review of approximation methods for solving POMDPs. The complexity of the methods reviewed there is in the best case polynomial to the POMDP size. Furthermore, one of the most recent methods for approximation is the Point Based Value Iteration (PBVI) [9] method, where its complexity is again polynomial to the size of the POMDP.

All the above mentioned methods have been applied to problems where in the best case the POMDP comprised of a few thousand states with an exception of the work in [12] where the POMDP is comprised of millions of states as with our approach but it cannot be solved in real time. The problem we consider consists of many orders of magnitude larger state space. As a result the reduction of the state space that the RN-HPOMDP offers and also the reduction of the action space is crucial to its performance. Furthermore, since the proposed RN-HPOMDP is not restricted to a specific method for solving the underlying POMDPs, a combination of an approximation method for solving a flat POMDP with the proposed hierarchical structure improves dramatically its performance.

5 Experimental Results

In Table 3, the CPU time required to solve the proposed HPOMDP structure using the Voting heuristic for varying grid size and number of levels is given, where as it can be observed with appropriate choice of the number of levels real time POMDP solution is possible. It is evident from the results presented in Table 3 that the RN-HPOMDP is amenable to real-time solution in problems with extremely large state and action spaces.

Table 3: Computation time required to solve the proposed HPOMDP.

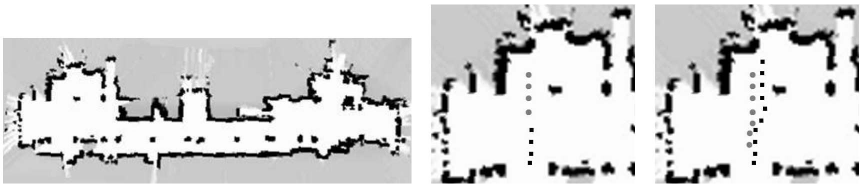
No. of Levels	**Grid size**	**POMDP size**		**time (sec)**
5	$5cm^2$	$\lvert S\rvert = 18,411,520$	$\lvert A\rvert = 64$	18.520
5	$10cm^2$	$\lvert S\rvert = 4,602,880$	$\lvert A\rvert = 64$	0.911
5	$15cm^2$	$\lvert S\rvert = 2,038,080$	$\lvert A\rvert = 64$	0.426
5	$20cm^2$	$\lvert S\rvert = 1,150,720$	$\lvert A\rvert = 64$	0.257
5	$25cm^2$	$\lvert S\rvert = 734,976$	$\lvert A\rvert = 64$	0.262
5	$30cm^2$	$\lvert S\rvert = 503,808$	$\lvert A\rvert = 64$	0.251

No. of Levels	**Grid size**	**POMDP size**		**time (sec)**
3	$10cm^2$	$\lvert S\rvert = 1,150,720$	$\lvert A\rvert = 16$	201.210
4	$10cm^2$	$\lvert S\rvert = 2,301,440$	$\lvert A\rvert = 32$	16.986
5	$10cm^2$	$\lvert S\rvert = 4,602,880$	$\lvert A\rvert = 64$	0.911
6	$10cm^2$	$\lvert S\rvert = 9,205,760$	$\lvert A\rvert = 128$	0.460
7	$10cm^2$	$\lvert S\rvert = 18,411,520$	$\lvert A\rvert = 256$	0.411

The RN-HPOMDP has been tested extensively in a real world environment. The robot was set to operate for more than 70 hours in the FORTH main entrance hall shown in Figure 2. The environment was modeled with a RN-HPOMDP of size $|S| = 18,411,520$, $|A| = 256$ and $|Z| = 24$, built with 7 levels. Experiments were performed in a dynamic environment where people were moving within it. A sample path the robot followed to reach its goal and also performed local obstacle avoidance to avoid a human is shown in Figure 2.

6 Conclusions and Future Work

In this paper, a novel hierarchical representation of POMDPs for autonomous robot navigation has been proposed that can be solved for the first time in real-time when an extremely large state space is involved and is memory efficient. Hence, the RN-HPOMDP provides a unified framework for robot navigation that is able to provide the actual actions the robot executes without the intervention of any other external modules. Our proposed hierarchical structure employs state space and action space



(a) (b) (c)

Figure 2: The FORTH main entrance hall and avoiding a human to reach the goal position. The robot track is marked with the black dots (•) and the human track is marked with the grey dots (•).

hierarchy. Memory efficiency is achieved by introducing the *reference* POMDP that holds all the information regarding motion and sensor uncertainty. Our comparative experiments have indicated that our approach results in very efficient computation times and manageable memory requirements for realistic environments. The RN-HPOMDP provides an approximate solution suited for the robot navigation problem. Preliminary versions of the proposed RN-HPOMDP structure have already been applied for predictive obstacle avoidance [2] and robot velocity control in dynamic environments [3]. Future work involves applying the RN-HPOMDP into the multi-robot navigation problem.

References

[1] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[2] A. Foka and P. Trahanias. Predictive autonomous robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, 2002.

[3] A. Foka and P. Trahanias. Predictive control of robot velocity to avoid obstacles in dynamic environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, 2003.

[4] M. Hauskrecht. Value function approximations for Partially Observable Markov Decision Processes. *Journal of Artificial Intelligence Research*, 13:33–95, 2000.

[5] L. P. Kaebling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[6] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[7] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1998.

[8] I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish an office-navigating robot. *AI Magazine*, 16(2):53–60, 1995.

[9] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2003.

[10] J. Pineau, N. Roy, and S. Thrun. A hierarchical approach to POMDP planning and execution. *Workshop on Hierarchy and Memory in Reinforcement Learning (ICML)*, 2001.

[11] J. Pineau and S. Thrun. An integrated approach to hierarchy and abstraction for POMDPs. Technical Report CMU-RI-TR-02-21, Carnegie Mellon University, 2002.

[12] P. Poupart and C. Boutilier. VDCBPI: an approximate scalable algorithm for large scale POMDPs. In *Neural Information Systems (NIPS)*, 2004.

[13] R. Simmons, J. Fernandez, R. Goodwin, S. Koenig, and J. O'Sullivan. Xavier: An autonomous mobile robot on the web. *Robotics and Automation Magazine*, 1999.

[14] M. Spaan and N. Vlassis. A point-based POMDP algorithm for robot planning. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, 2004.

[15] G. Theocharous. *Hierarchical Learning and Planning in Partially Observable Markov Decision Processes*. PhD thesis, Michigan State University, 2002.

[16] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive musuem tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.