The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Decision-theoretic Robot Guidance for Active Cooperative Perception

Abdolkarim Pahliani Matthijs T. J. Spaan Pedro U. Lima

Abstract— **We consider the problem of sensor-aware path planning for a robot in a Networked Robot System, in particular in urban environments equipped with a network of surveillance cameras. A robot can use observations from the camera network to improve its own localization performance, but also needs to take into account the specifics of its local sensors. We model our problem in the Markov Decision Process framework, which forms a natural way to express concurrent and possibly conflicting objectives – such as reaching a goal quickly, keeping the robot localized, keeping the target in sight – each with their own priority. We show how we can successfully prioritize the different objectives in a flexible way by changing the reward function, based on the sensory needs of the system.**

I. Introduction

Robots are leaving the research labs and operating more often in human-inhabited environments, such as urban pedestrian areas. The main idea of the URUS (Ubiquitous Networking Robotics In Urban Settings) Project [1], [2] is to incorporate a network of intelligent components, e.g., robots, sensors, devices and communications in order to improve quality of life in urban areas. The scenario we consider in our work is a group of robots assisting humans in a car-free area, a so-called Networked Robot System (NRS). The pedestrian area in which the robots operate is equipped with surveillance cameras providing the robot with more information. Implementing such a system requires addressing many scientific and technological challenges such as cooperative localization and navigation, map building, human-robot interaction, and wireless networking, to name but a few. In this paper, we focus on one particular problem, namely how to plan paths for robots taking into account the coverage of the camera network as well as the robots' own sensors.

In many NRS, surveillance cameras will run a set of event detection algorithms, for instance observing events such as people waving, people lying on the floor, fires, or other emergencies, each with a different priority. However, the network of cameras will have a limited coverage and accuracy. In particular, the environment might contain blind spots that are not observed by any fixed camera. As such, though the camera network is supposed to cover the scene, employing mobile robots for visual coverage is a need. A camera network might cover a lab environment, but providing full coverage for urban environments is a difficult task. There are often obstacles both natural and man-made in the environment which make parts of the environment hidden from the camera network. Even if we could employ a large number of cameras to have an environment fully in view, dynamic obstacles still can create new hidden patches.

Furthermore, other areas might be observed by a camera, but not with sufficient resolution for accurate event detections. In this case, we send mobile robots to positions where higher resolution images are required. In NRS the interaction between the system and humans will largely be achieved through human-robot interaction, which in general requires a robot to be close to a human subject. In this work, we consider the problem of a robot planning a path to reach a target location. For instance, consider a situation where a robot needs to reach a human for interaction purposes. The robot should take into account available sensory capabilities provided by a robot's mounted sensors as well as by the network of surveillance cameras. In particular, a robot can use observations from the camera network for its own localization, or take into account the specifics of its mounted sensors to plan an approach to a target location that maximizes the information its sensors will give it about the target.

We use a Markov Decision Process (MDP) framework to address our sensor-aware path planning problem [3], [4]. A decision-theoretic framework such as the MDP forms a natural way to express concurrent and possibly conflicting objectives such as reaching the goal quickly, keeping the robot localized, keeping the target in sight, each with their own priority. Given the partially observable nature of the problem, modeling it as a partially observable MDP (POMDP) would be appropriate. However, given the scale and level of detail of the problems we are targeting, with many states, and, more importantly, a large number of possible observations and a high planning horizon, this is beyond current state-of-the-art approximate POMDP planners.

II. Related Work

In related work, the Coastal Navigation algorithm models the problem of navigating a robot while keeping localization uncertainty low as a POMDP [5]. It converts the POMDP into an augmented MDP, which has an extended state space composed of robot locations and discretized entropy levels. The entropy is used as a measure for the uncertainty of the robot's localization. In our case, we keep the size of the state space constant, focusing on modifying the reward function instead. This is a flexible way of incorporating different objectives, beyond only caring about the robot's localization certainty: we also consider the visibility of the target by the robot. Keeping a constant state space allows for quick

This work is supported by the European Project FP6-2005-IST-6-045062-URUS and by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the POS_Conhecimento Program that includes FEDER funds, and by grant SFRH/BD/23394/2005.

The authors are with the Institute for Systems and Robotics at Instituto Superior Técnico, Technical University of Lisbon, Portugal. {apahliani,mtjspaan,pal}@isr.ist.utl.pt

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

solving of the MDPs. The environment, the costs and the rewards can be modeled in advance and the optimal path can be determined for all destinations. Moreover, if there are changes in the environment, updating the MDP model even with a large number of states is quite fast. We have a good initial estimate of the value functions which causes the algorithm to converge quickly.

Some researchers studied this problem under a path planning framework. Choi et al [6] used Q-learning to find the path which can maintain good kinematic isotropic property while avoiding obstacles. Singh [7] et al introduce a greedy search approach for motion planning in order to maximize the amount of information collected while placing bounds on their resources. Since the original algorithm, called *recursive greedy*, is computationally expensive, an approximate algorithm is used which decomposes the state space in a uniform grid in order to reduce the computational complexity. The algorithm is suboptimal and is still expensive to apply to real-time applications. In [8], a gradient-search-based algorithm is used to provide a suboptimal solution for sensor position selection to realize the best observation of a moving target in an environment with no obstacles. Comparing to our work, the authors only considered the localization certainty as a parameter that affects the robot path. Moreover, the algorithm only considers one step ahead rewards based on the other robots' position prediction. Park [9] proposes a real-time path planning by combining probabilistic roadmap and reinforcement learning to deal with uncertain dynamic environments and similar environments. To avoid obstacles, the Q values in the states occupied by the obstacles are set to zero. This is one shortcoming of this work because the planned path might not be optimal anymore, specially if the environment is highly dynamic.

III. Background on Markov Decision Processes

We will briefly introduce the Markov Decision Process (MDP) framework [3], [4]. MDPs provide strong mathematical tools for decision making under uncertainty, in case the state of the environment is observable to the robot. It is formally specified by a four tuple (S, A, T, R) where S is a (finite) set of states, A is a (finite) set of atomic actions, T is the transition model and R is a reward function. Each element of S describes the state of the system at a given time instant. Each action element $a \in A$ represents the action that agent takes, at any time step. A value function defined as $V : S \rightarrow \mathbb{R}$ determines the sum of total expected future reward from being in a state s: $V(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R_t(s, a)\right]$, where $0 \leq \gamma \leq 1$ is a discount factor. A policy is a function $\pi : S \rightarrow A$ which maps states to actions. $\pi(s)$ states the action that should be taken in state s and the value of the policy $V_\pi(s)$ is the expected cumulative discounted future reward that the agent gets if it executes π. The optimal policy π^* tells us which action to take at each state in order to maximize the expected reward, and can be implemented using the optimal value function V^*. It is known that V^* verifies

$$V^*(s) = max_{a \in A} \left\{ R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s') \right\}. \tag{1}$$

In order to compute V^*, dynamic programing techniques such as value iteration can be used [3], [4].

IV. Costs and Rewards for Active Cooperative Perception

We will implement our ideas on decision-theoretic robot guidance by defining the MDP's reward function. This is a flexible way for the user of the system to specify the relative importance of the considered factors. In particular, the idea of taking the best path is directly related to costs and rewards. By rewards, we mean what the agents receive along the path or at destination. The costs are defined as the amount of resource consumption, effort, loss necessary to achieve the goal or the risk, e.g., risk of bumping into an obstacle due to taking a narrow path. In our scenario, localization certainty, visibility of the target location, as well as reaching the destination are considered as the rewards. Maneuvering risk and traveling are considered as costs, i.e., as negative rewards. Each of them are explained below in detail.

Before going into details, it is necessary to mention that the world is discretized in a number of states. Each state is specified by its position and its orientation. The orientation space is divided into eight equal sectors and the first starts at zero radian. There are three atomic actions possible in each state: stay in the same state but change the orientation $\pm\frac{\pi}{4}$, or move forward.

A. Goal Reward

The goal reward ρ_G is defined as the reward the agent receives when it reaches the goal state. This reward may vary based on the degree of our interest in the goal and the situation. For example, if the camera network detects a fire and the system deploys the robot to provide more details, considering the urgency of the case, the system only considers the rewards which result in generating the fastest path to the goal and ignores other possible rewards.

B. Localization Certainty Reward

Often, the pose of a target, e.g., a robot, a person, etc. is an important piece of information we need to know. For example, when the robot should approach a person to let the person to interact with the robot, in order to prevent collision, having an accurate relative localization of robot and the person is very important. In another word, if the person is localized but with a large uncertainty, the robot can use its sensors in order to help the camera network to better localize the person. Therefore, if the localization uncertainty of the robot is not good enough, while it is traveling toward the person, it has to give more priority to paths with larger Certainty Reward than to other paths, e.g., shorter paths but with a large localization uncertainty.

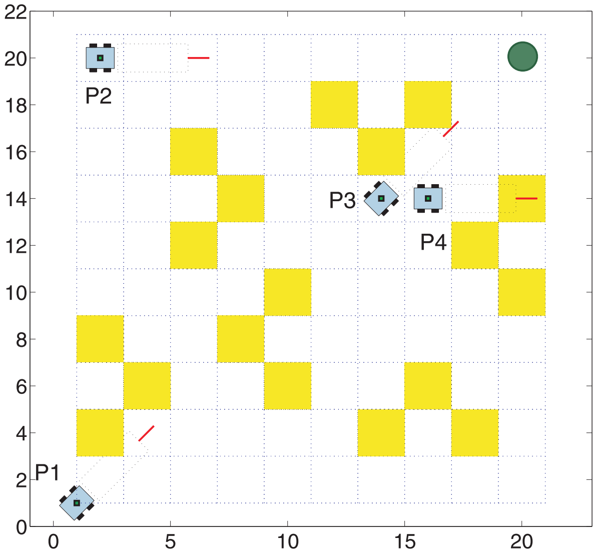


Fig. 1. The figure shows a robot in several positions. The reward in Positions P1 and P4 is zero but in P2 and P3 is not zero. The reward value depends on the relative distance and the angle receives.

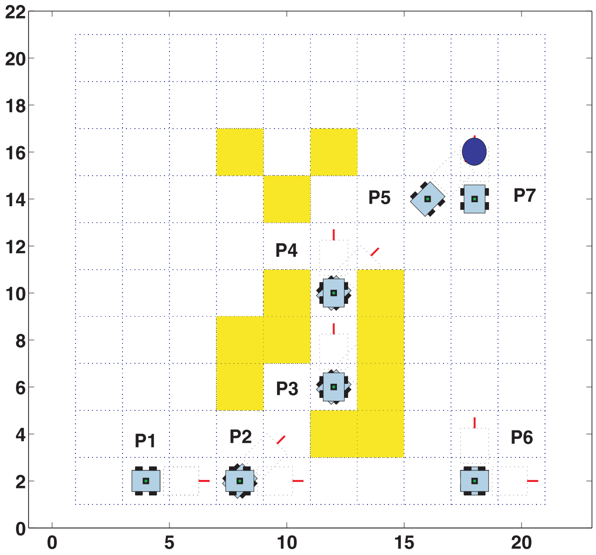


Fig. 2. Illustration of the maneuvering cost function. Giving more attention to maneuvering cost than the traveling cost, in absence of other costs and rewards, forces the robot to take path P1-P2-P6-P7 instead of P1-P2-P3-P4-P5, although P1-P2-P3-P4-P5 is shorter.

The observation model of the surveillance cameras is assumed Gaussian, with the mean centered at the real value. The covariance increases proportionally with the relative distance between the camera and object of interest. To each state in state space we assign a real number ρ_{L} which is called Localization Certainty and is defined as:

$$\rho_{\mathrm{L}} = \frac{1}{1+\sigma_i} \tag{2}$$

where σ_i is defined as:

$$\sigma_i = \frac{1}{e^T \Sigma^{-1} e} \tag{3}$$

where $\mathbf{e} = [1, 1, ..., 1]^T$ is a $1 \times N$ vector, N is the number of cameras that can observe the state and Σ is the covariance matrix of cameras which cover the state.

C. Visibility Reward

One important issue in our scenario is the visibility issue. The visibility is defined as feasibility of observing the object of interest at a specific position and angle. We explain this concept by providing an example which is drawn in Fig. 1. In this example, a robot with an on-board camera in several positions is shown. The robot in P1 is not able to see the object of interest which is depicted by a circle because its line of sight is blocked by the obstacle on the way. However, in P2, the robot is potentially able to view the object. It means that although the object may not be in robot camera view field, there is no obstacle that blocks the robot line of sight. The robot in P3 can see the target and we give a higher visibility reward compared to P2 because as it is closer to the object and the visibility is less sensitive to change in the orientation. Moreover, in P4, a zero visibility reward is assigned. Although it is closer to the object than the robot in P2, considering its orientation, the target is not in the robot's line of sight.

Formally, ρ_{V} is defined as:

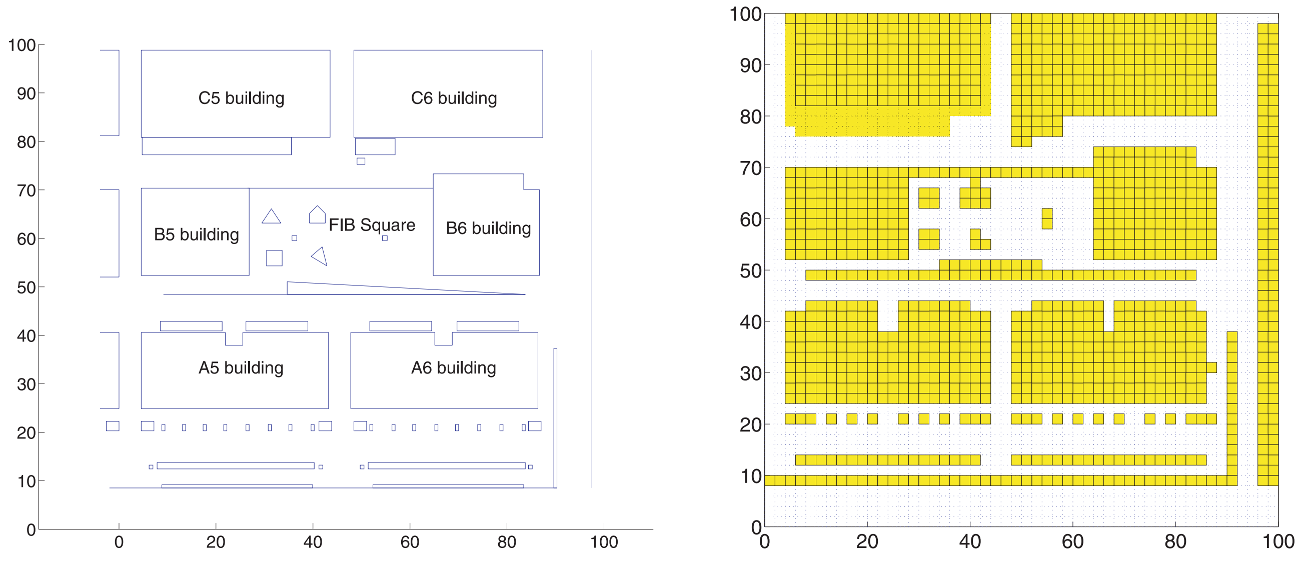
$$\rho_{\mathrm{V}} = \frac{\alpha_{vi}}{\Delta_p} \tag{4}$$

$\alpha_{vi} = 0$ if $\Delta_p > \eta$, $|\Delta_\theta| > \xi$, line of the sight is blocked by an obstacle placed between the state and the goal or an obstacle is in the state; Otherwise $\alpha_{vi} = 1$. Δ_p is defined as the Euclidean distance between the goal and the state and η is a positive number representing the maximum visibility radius. Δ_θ is the relative angle between the robot's orientation and the line of the sight to the goal and ξ representing the maximum visibility angle. The visibility and the robot sensor range are related but visibility is a different concept, as it is affected by the robot orientation and, more importantly, the path characteristics. A path with many obstacles between the goal and the robot has a low visibility, even if the robot is equipped with a long-range sensor.

D. Maneuvering Cost

Often, a robot needs to change its orientation. To do so, it needs space. In larger spaces, the maneuvering risk is smaller. For a robot, it is less possible to bump into an obstacle when it has a larger free space to maneuver. The places closer to the obstacle are more risky for changing the orientation. Moreover, a narrow passage is more risky to take than a wider passage. Therefore, the maneuvering cost ρ_{M} of each state is defined as a function of two factors:

$$\rho_{\mathrm{M}} = \frac{\alpha_M}{\lambda} + (1-\alpha_M) * \vartheta \tag{5}$$



(a)

(b)

Fig. 3. The left figure shows the schematic diagram of the URUS test bed which is located in UPC Nord campus, Barcelona. The right figure presents the free spaces and obstacles. The places marked with yellow filled squares are the obstacles and the rest is the free space.

where $0 \leq \alpha_M \leq 1$ is used to balance the importance of passage width (first term) versus the number of surrounding obstacles (second term) and ϑ is the number of surrounding obstacles. λ is defined as:

$$\lambda = \frac{\lambda_i}{\lambda_{max}} \tag{6}$$

λ_i is the cell size of i^{th}cell and λ_{max} is the largest cell size. To make things clear, a scenario is explained in Fig. 2. Considering a higher cost for maneuvering, the robot is forced to take path P1-P2-P6-P7 instead of taking the path P1-P2-P3-P4-P5, even though the second path is shorter.

E. Traveling Cost

The cost of traveling ρ_T has two components: the relative distance and rotation. It is considered as a linear combination of the two costs:

$$\rho_\mathrm{T} = \alpha_T * \Delta_p + (1 - \alpha_T) * \Delta_\theta, \tag{7}$$

where $0 \leq \alpha_T \leq 1$.

The first component is calculated based on the relative Euclidean distance Δ_p the robot needs to take to travel from one state to another. The second component is determined by calculating the absolute difference Δ_θ between the orientation of the two states. We usually give the higher relative importance to the second term as for our robots changing the orientation needs more resources in terms of energy and time.

V. Decision-theoretic Robot Guidance

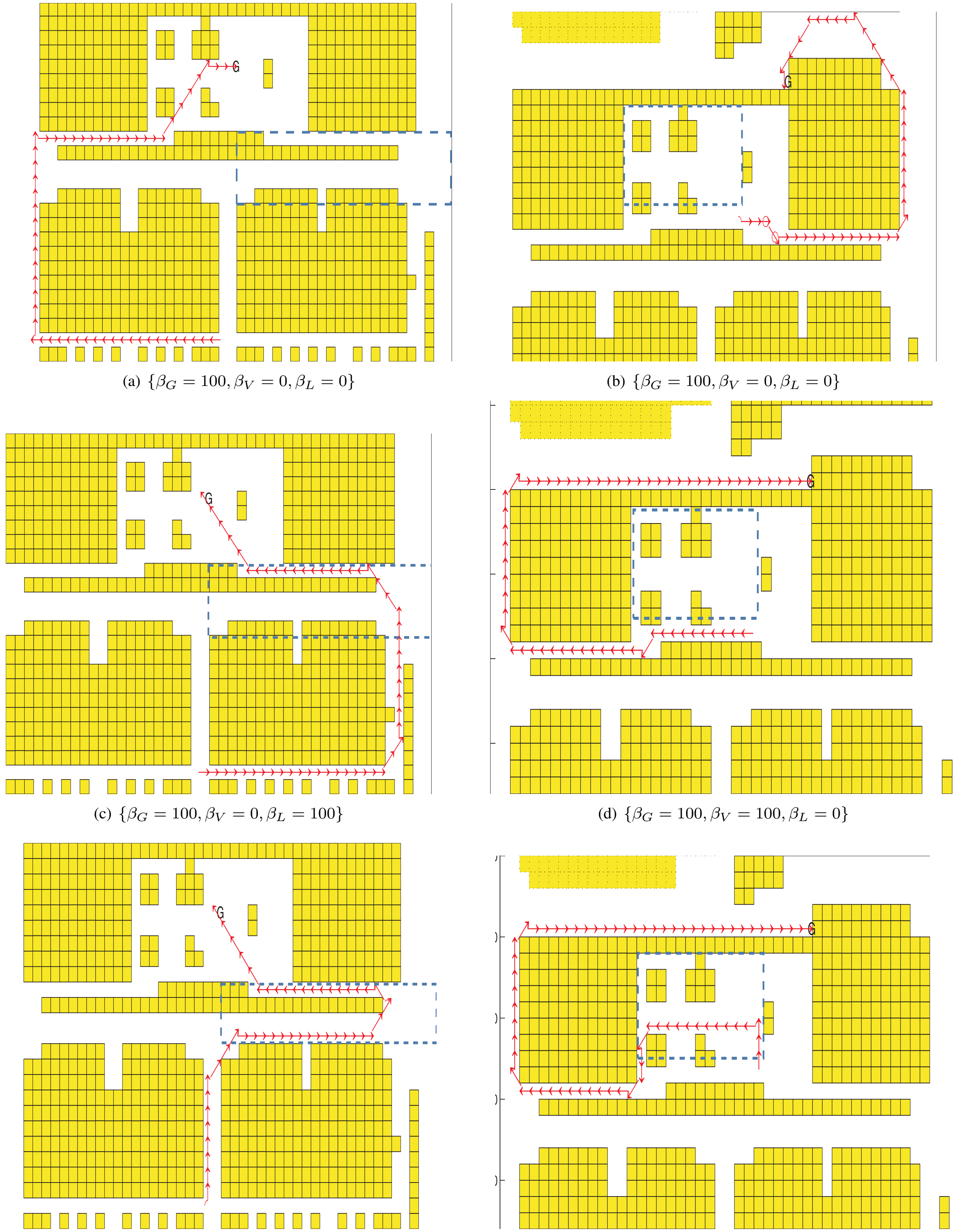
We use the concepts defined in the previous section to plan paths using value iteration (Section III). Value iteration considers all the quantitative rewards and computes the best path. We model the environment, cost and rewards and then, using a simulated environment, we determine the optimal path for all possible goal states off-line. One important issue with this method is the change in the environment. Since the environment is dynamic, we may experience changes in the environment, e.g., an unforeseen obstacle appears on the robot path and blocks it. In this case we recalculate the value function. As we have a good initial starting value for value functions, in a few iterations the algorithm might converges. Because of that we call this method *active* cost-reward based robot guidance since we can change the optimal path according to changes in the environment and also our needs. We define the rewards as:

$$\rho = \beta_G\rho_G + \beta_V\rho_V + \beta_L\rho_L + \beta_T\rho_T + \beta_M\rho_M \tag{8}$$

Choosing β is based on the robot mission. To limit the search space of ρ's, we normalize the cost and rewards, $\rho \in$ [0,1].

VI. Experiments

To verify the performance of the proposed method, we ran a series of simulations. Fig. 3(a) shows the schematic diagram of the URUS test bed which is located in UPC Nord campus, Barcelona. The area size is about 1 hectare, which we divided in equal size $2 \times 2\ m^2$ squares as depicted in Fig. 3(b). In each cell, we considered 8 different robot orientations. The first orientation is at 0 radian and the step is $\frac{\pi}{4}$. The total number of states is 20000. However, part of them are occupied by obstacles and we only deal with the free states. Fig. 3(b) presents the free spaces and obstacles. The places marked with yellow filled squares are the obstacles and the rest is the free space. A discrete MDP is used to model the path generation. The reward function is considered according to (8). The basic atomic actions are either to stay in the same cell and only change the orientations $\pm\frac{\pi}{4}$ or move



(a) $\{\beta_G = 100, \beta_V = 0, \beta_L = 0\}$

(b) $\{\beta_G = 100, \beta_V = 0, \beta_L = 0\}$

(c) $\{\beta_G = 100, \beta_V = 0, \beta_L = 100\}$

(d) $\{\beta_G = 100, \beta_V = 100, \beta_L = 0\}$

(e) $\{\beta_G = 100, \beta_V = 0, \beta_L = 300\}$

(f) $\{\beta_G = 100, \beta_V = 100, \beta_L = 100\}$

Fig. 4. The figures represents the generated path in different situations. In the figures, the dashed rectangle corresponds to a region covered by cameras, while the yellow squares are obstacles. To make the results better visible, only part of the scene shown in 3(b) in which the scenarios take place are shown. For all above cases, we set $\beta_T = -1$ and $\beta_M = 0$.

forward. Here we assume deterministic actions, however it is trivial to extend the work to noisy actions, as the same value iteration procedure can be applied. In Fig. 4, the goal position is specified by 'G' and the area under the camera coverage is shown by a rectangle with a dashed edge.

Consider the case when the network of cameras detects a fire. The robot should be deployed in such a way it gets to the place in the shortest possible time. Another situation is where the robot is asked to approach and provide a service for a person who is localized but with large uncertainty. To do so, robot has to know its own localization very well in order to find out the position of the person using the relative localization for further operation. This is the situation where robot has to take a path under camera coverage and with acceptable ρ_L. The aim of the experiments are to evaluate the effect of different parameters on the generated path. Fig. 4(a) shows a scenario where we have a camera which covers the area marked with the dashed rectangle. We check the behavior of the system, the generated optimal path, by changing the values of β_L and β_T while β_V is set to zero. First, we set β_L to zero. Naturally, the generated path is the path with the lowest traveling cost. In Fig. 4(a), the generated path is shown. In Fig. 4(c), we kept all β's the same but change β_L. Increasing β_L causes a different path to be considered for the robot. The generated path goes through the area covered by the camera. To see the further effect of increasing β_L, we use the same setting but increase β_L. This time the system changes the generated path in such a way it stays longer under the area covered by the camera. The result is shown in Fig. 4(e). It can be seen that even when we change the robot orientation, due to a large β_L, the system still guides the robot to the area covered by the camera.

The next scenario is designed to see the effect of β_V and β_T on the generated path while either β_L is fixed or changed. There are situations where sending the robot to the position where the object is in robot line of sight has the top priority e.g., the camera detects an intruder and has to send the robot to track. In other words, the priority is that the robot reaches to a point in which it can observe the person as quickly as possible. Using a robot equipped with a laser range finder, the system can then track the person. Fig. 4(b) shows the case where we set all β_V and β_L to zero. This is similar to Fig. 4(a) but the starting and goal locations are different. The generated path has the lowest traveling cost. In the second case we gave more priority to β_V over β_T. As we can see in Fig. 4(d), the robot is provided with a different path. Only considering the traveling cost, this path is more costly. However, taking this path, causes the robot to reach earlier a point at which the object is in its line of sight. We have an area under camera coverage close to starting state. For improving robot localization uncertainty, giving some weight to β_L, causes the path generated to become longer but pass through the area covered by the camera. This is depicted in Fig. 4(f).

VII. Conclusion and Future Work

In this paper we address the problem of generating an optimal path for a robot taking into account available sensory capabilities, both provided by a robot's own sensors and by a network of surveillance cameras. By changing some parameters we can guide the robot to the same position but taking different paths. The urban environments we target are highly dynamic environments in which demands change rapidly. Sometimes a robot should reach the goal as fast as possible, sometimes it should consider other factors such as its localization uncertainty and sometimes for an optimal path we should consider the positions of both the object of interest and the robot. We model the path planning problem as Markov Decision Process, which allows to prioritize the different objectives in a flexible way by changing the reward function. We also can solve the MDP in real time using value iteration. Since main focus of the NRS is to employ a network of cooperative robots in order to assist and provide services for human beings, extending this solution to the multi-robot and multi-goal active guidance is necessary. Since the number of robots is limited and we might have more demands for services at the same time than available resources, we have to prioritize our planning based on degree of our interest in the objects, the costs and rewards explained in this paper. In other words, the challenge will be to tell the system which robot should take which path and in which order.

References

[1] A. Sanfeliu and J. Andrade-Cetto, "Ubiquitous networking robotics in urban settings," in *Proceedings of the IEEE/RSJ IROS Workshop on Network Robot Systems*, 2006.

[2] M. Barbosa, A. Bernardino, D. Figueira, J. Gaspar, N. Gonçalves, P. U. Lima, P. Moreno, A. Pahliani, J. Santos-Victor, M. T. J. Spaan, and J. Sequeira, "ISRobotNet: A testbed for sensor and robot network systems," in *Proc. of International Conference on Intelligent Robots and Systems*, 2009, to appear.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2000.

[5] N. Roy and S. Thrun, "Coastal navigation with mobile robots," in *Advances in Neural Information Processing Systems 12*. MIT Press, 2000.

[6] M. Choi, W. Kim, and B.-J. Yi, "Trajectory planning in 6-degrees-of-freedom operational space for the 3-degrees-of-freedom mechanism configured by constraining the stewart platform structure," *Control, Automation and Systems, 2007. ICCAS '07. International Conference on*, pp. 1222–1227, Oct. 2007.

[7] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin, "Efficient planning of informative paths for multiple robots," in *International Joint Conference on Artificial Intelligence (IJCAI)*, January 2007.

[8] T. Chung, V. Gupta, J. Burdick, and R. Murray, "On a decentralized active sensing strategy using mobile sensor platforms in a network," *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 2, pp. 1914–1919 Vol.2, Dec. 2004.

[9] J.-J. Park, J.-H. Kim, and J.-B. Song, "Path planning for a robot manipulator based on probabilistic roadmap and reinforcement learning," *International Journal of Control, Automation, and Systems*, vol. 5, no. 6, pp. 674–680, 2007.