2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Appearance-based Motion Strategies for Object Detection

Israel Becerra†, Luis M. Valentín-Coronado†, Rafael Murrieta-Cid† and Jean-Claude Latombe◦
† Centro de Investigación en Matemáticas, CIMAT
Guanajuato, México
{israelb,luismvc,murrieta}@cimat.mx
◦ Stanford University, Stanford, CA, USA
latombe@cs.stanford.edu

***Abstract*— This paper investigates an object detection problem using a mobile robot equipped with a vision sensor. The main novelty of this work is an approach that combines localization of the robot relative to an object believed to be the target and confirmation of this object's identity. Since the position of the robot relative to the candidate target is never exactly known, we model this position by a probability distribution over a set of cells forming a decomposition of the workspace around the candidate target. By performing a series of moves the robot acquires several images and runs a target detector module on each image. Its goal is not only to reach a position where the target detector can confirm the target with high confidence (as this approach would be prone to false positives). It is also to reach a position where, with high probability, the target detector will confirm with high confidence that the candidate target is actually the target. This twofold goal reduces drastically the likelihood of false positives. The target confirmation problem is modeled as a Partially-Observable Markov Decision Process (POMDP), which is solved using Stochastic Dynamic Programming (SDP).**

I. INTRODUCTION

In this paper we investigate an object detection problem using a mobile robot equipped with a vision sensor. More precisely, we consider the following scenario: the robot is instructed to find a certain object T (the target) in its environment; at some point of the search process (e.g., by using a coverage software module such as the one described in [1]), the robot believes that it has encountered a candidate C for T, but it is not sure yet that C is T; taking advantage of its mobility, the robot tries to achieve adequate viewpoints to confirm (or infirm) that C is actually T. The robot plans its motion using an appearance-based model of T constructed during a preliminary training phase.

The main novelty of this work is an approach that combines localization of the robot relative to the object C (the candidate target) and confirmation of the object identity as the target. Since the position of the robot relative to the candidate target C is never exactly known, we model this position by a probability distribution over a set of cells forming a decomposition of the workspace around C. By performing a series of moves the robot can acquire several images of C and run a target detector module on each image. The goal of the robot is not just to reach a position where the target detector can confirm with high confidence that C is actually the target T (as this approach would be prone to false positives). It is also to reach a position where with high probability the target detector will confirm with high confidence that C is the target T. This twofold goal reduces drastically the likelihood of false positives.

We believe that the ability of a mobile robot to reliably confirm the presence of a target object in an environment has a wide range of applications, specially as an end module for search problems. There has been much research where a robot or a team of robots must search an environment to find an object or a moving agent (adversarial or not) [2], [4], [5]. Usually in this kind of research the target is considered found when a robot has established a line of sight with it. This detection condition is based on a strong assumption as it requires that the robot will be able to identify the target from any orientation at any distance. Several research works try to fulfill this detection condition by creating powerful detectors [6], [7], [8]. Instead, we believe here that the motion capabilities of the robot should be exploited to confirm the presence of the target.

Section II discusses related work. Section III defines the observation model of a target T. Section IV proposes a probabilistic model of the target confirmation problem. Section V describes the probabilistic model underlying the motion commands available to the robot. Section VI presents the planning method used to solve the confirmation problem. Secton VII reports on three experiments carried out in simulation. Section VIII concludes the paper and suggests directions for future research.

II. RELATED WORK

There has been a considerable amount of research on search problems in robotics. One such problem is a pursuit-evasion problem where a team of robots (the pursuers) must find an evasive mobile target (the evader). The planner must compute a path of the pursuers that guarantees that the evader will eventually be detected, even if it tries to sneak in areas of the environment that have previously been visited by pursuers [2], [3], [4]. One possible additional constraint is, for a given geometry of the environment, to minimize the number of pursuers. A common assumption made in this line of work is that the evader is detected as soon as a line of sight exists between one pursuer and the evader. This assumption is not very realistic in most applications.

The problem of searching a static object has often been treated in previous research as an active vision problem [10],

[11], where the mobility of the robot is exploited to take several images and the planner selects the successive viewpoints. This problem is related to the Next-Best-View problem in map building and object modeling [12], [13], [14]. A typical active vision method consists in minimizing the entropy of an estimator based on the motion of the visual sensor [15]. In this line of work, a method is presented in [9] that first learns viewpoint detection models for object categories. These models are then used in a sequential recognition process, where the robot incrementally infers and updates the likelihood that an object of a certain category is present in a scene, as new detector responses are received. The viewpoint planner uses a greedy approach that selects the next viewpoint that minimizes the entropy over the posterior belief of the presence of an object. Like in [9], our work is also based on the idea that the ability to detect a target depends on the viewpoints from which it is observed. However, our approach is not based on entropy minimization. Instead, it combines robot localization relative to a candidate target and confirmation of this candidates identity. Moreover, our planner uses a probabilistic model of robot motion. A POMDP-based approach for active perception planning has also been reported in [21].

In [1] we have addressed the problem of covering an environment to find a target candidate. The new work presented here complements this previous work as follows. The method in [1] is intended to quickly find a candidate C of the target object T over a *large* environment by using simple visual clues (e.g., C has the same color as T); but it is not reliable enough for most applications. The new method described below is intended to confirm that C is, or is not, T with high probability by making a more thorough use of successive images of C taken from a *small* region around C.

III. OBSERVATION MODEL

We assume that the robot moves on a planar horizontal surface and that it is able to rotate in place (e.g., like a differential-drive robot). We also assume that during the target confirmation process, the candidate target will not be significantly hidden by any obstacle and that the robot will be able to move freely around it. The position of the robot is defined as the position of the camera in the horizontal plane. We assume that the camera is mounted on a turret, so that it can always point toward the candidate target.

Before it can confirm the presence of a target T, the robot must be equipped with a software module D_T, hereafter called the *detector* of T, capable of identifying T, at least from some viewpoints. Given an arbitrary image taken by the robot's camera, D_T returns a discrete detection score $y \in \{o_1, o_2, .., o_n\}$, where $o_1 < o_2 < ... < o_n$, measuring how well the image matches the appearance of T, hence the confidence of the identification. The response o_n indicates very high confidence, whereas o_1 means very poor confidence. In addition, we expect that D_T can sometimes make false positive mistakes, e.g., because the image contains an object similar to T or because it is noisy.

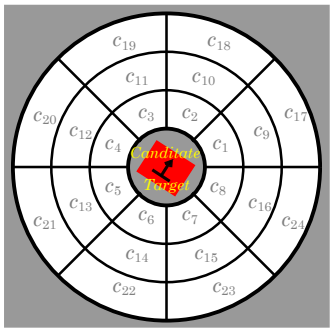


Fig. 1: Space decomposition around candidate target C

At this stage, the method used to create D_T is not important (any method could potentially be used). In Section VII we will present the specific method used in our experimental system.

The observation model is a trained probabilistic model of the response of D_T as a function of the location of the robot relative to T. To train this model, we first attach a horizontal two-dimensional coordinate frame to T and we decompose the space around T into non-overlapping cells $c_1, ..., c_m$. Fig. 1 shows a particular decomposition made of $m = 24$ cells created by drawing concentric circles centered at the origin of the coordinate frame (roughly the center of T) and erecting radial rays from this origin. Although many other decompositions could be used (some perhaps better), we will use this decomposition throughout the rest of this paper. We will discuss the potential impact of the resolution of the decomposition on the efficiency of the detection process in Section VII.

Given D_T and the space decomposition around T, the observation model of T is then created in the form of a probability distribution $P(o_j|c_i)$, $i \in \{1, ..., m\}$ and $j \in \{1, ..., n\}$, where $P(o_j|c_i)$ is the probability that the response of D_T is o_j given that the input image has been taken from within cell c_i. For each cell c_i, a collection of images of T is taken at random positions of the robot in c_i. The normalized frequency of each response o_j of D_T for all these images gives $P(o_j|c_i)$.

IV. CONFIRMATION PROCESS

Consider now the situation where, using the coverage module presented in [1], the robot has detected in the current image the presence of an object C that might be the target T (hence, C is a candidate target). The goal of the confirmation process is for the robot to plan and perform a series of moves that will eventually confirm, or infirm, that C is actually T.

To plan these moves, the robot assumes around C the same space decomposition $X = \{c_1, ..., c_m\}$ that was created around T to create the observation model of T (see Section III).

It also assumes that it was brought by the coverage module into one cell of this decomposition, but the robot does not know which one. In other words, the robot does not know its position inside the decomposition.

The confirmation process consists for the robot to perform a series of motion commands u_t at times $t = 0, 1, 2, ...$, until

the presence of T is confirmed or infirmed. The motion model underlying u_t, which depends on the decomposition X, will be described in Section V. At each time t, the location x_t of the robot will be modeled at the cell resolution by a probability distribution over the m cells of X.

Initially (i.e., just before time $t=0$), the unknown location of the robot is assumed to be uniformly distributed over the m cells. At any time $t \geq 0$ the probability distribution of the robot's position over X, $P(x_t|I_t)$, can be inferred as follows from the observation model of T and the history $I_t = \{y_t, u_{t-1}, I_{t-1}\}$ of applied motion commands and successive scores y_t returned by the detector D_T [16], [17]:

$$P(x_t|I_t) = \frac{\sum_{x_{t-1}} P(x_{t-1}|I_{t-1})P(x_t|x_{t-1},u_{t-1})P(y_t|x_t)}{\sum_{x_t}\sum_{x_{t-1}} P(x_{t-1}|I_{t-1})P(x_t|x_{t-1},u_{t-1})P(y_t|x_t)} \quad (1)$$

where x_t and x_{t-1} range over $\{c_1,...,c_m\}$.

One goal of the robot during the confirmation process is to reach at some time t a position where D_T returns a high score $y_t \geq \hat{o}$, where $\hat{o}$ is a given threshold (in our experiment we usually set $\hat{o}$ to o_n). But this goal is not enough, since it could be achieved by chance (false positive). In addition, we require that the robot must have reached a position at t from which it expects with high probability (i.e., a probability greater than some given threshold λ) that D_T will return a score $y \geq \hat{o}$. The resulting twofold goal does not completely eliminate the risk of a false positive, but drastically reduces the chances that it happens. Hence, planning a sequence of moves to confirm the presence of T mixes robot localization relatively to the candidate target and target identification.

As modeled above, the problem of computing a motion strategy for the confirmation process can be regarded as a POMDP [19], [20]. In our implementation, we use Stochastic Dynamic Programming (SDP) with imperfect state information as described in [18] to compute such a strategy. But, before describing this computation, let us introduce the motion model underlying the commands u_t, $t = 0,1,2,...$

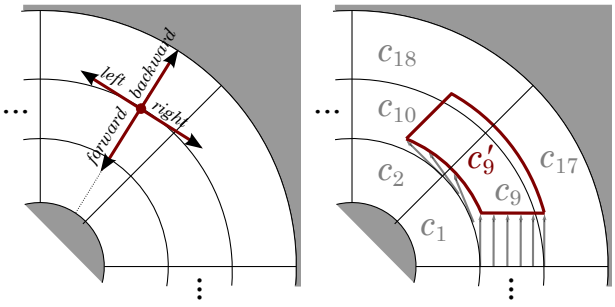
V. MOTION MODEL

The purpose of this section is to define the probability distribution $P(x_t|x_{t-1},u_{t-1})$.

At each step t, we allow the robot to choose among 4 motion commands (Fig. 2a): $U = \{forward, backward, left, right\}$.

The command *forward* is intended to make the robot move in the direction of the estimated origin of the candidate target's frame. The command *backward* is intended to make the robot move in the opposite direction. The commands *left* and *right* are intended to make the robot move tangentially around the estimated origin, respectively in the clockwise and counter-clockwise directions.

We estimate the position of the robot up to the resolution of the cells. This means that when the robot is located within a cell c_i, it can be anywhere in c_i with a uniform distribution over c_i. So, when a command u_{t-1} is executed from within a cell $x_{t-1} = c_i$, the transition probability $P(x_t = c_j|x_{t-1} = c_i, u_{t-1})$ is calculated as the ratio of the area of the region $R(x_{t-1} = c_i, u_{t-1})$ that is contained in cell $x_t = c_j$, where $R(x_{t-1} = c_i, u_{t-1})$ is defined as the set of all the robot positions resulting from the execution of u_{t-1} from each one of the positions within $x_{t-1} = c_i$. See Fig. 2b, where the command *left* was applied from within c_9 resulting in the region $c'_9 = R(x_{t-1} = c_9, u_{t-1} = left)$. (Note that here c'_9 is not a cell but a region that overlaps several cells.) For simplicity, we assume that motion control is perfect (i.e., the robot moves exactly as expected). This simplification does not eliminate the fact that the robot's position is uncertain (due to both its unknown initial position and space decomposition). Adding uncertainty in control could be done easily by enlarging the regions $R(x_{t-1} = c_i, u_{t-1})$.



(a) Motion commands (b) $c'_9 = R(x_{t-1} = c_9, u_{t-1} = left)$

Fig. 2: Motion model

The transition probabilities $P(x_t = c_j|x_{t-1} = c_i, u_{t-1})$ can be calculated geometrically for each one of the four commands. An advantage of the concentric cell decomposition of Fig. 1 is to benefit from symmetries that reduce computation.

VI. COMPUTATION OF MOTION STRATEGY

We compute a motion strategy up to a finite *planning horizon* N in the form of a policy $\pi(t, I_t)$ that gives the motion command to be applied at every time $t = 0,1,2,...,N-1$ for every possible history I_t. For this computation, we use Stochastic Dynamic Programming with Eqs. (2), (3), (4), (5), and (6) (see [18]), where $\tilde{g}(I_t, u_t)$ is the gain function.

$$J_{N-1}(I_{N-1}) = \max_{u_{N-1}\in U_{N-1}} \Big[\tilde{g}(I_{N-1},u_{N-1}) + \underset{x_{N-1}}{E}\Big\{\underset{x_N}{E}\{g_F(x_N)|x_{N-1},u_{N-1}\}|I_{N-1},u_{N-1}\Big\}\Big] \quad (2)$$

$$\pi(N-1,I_{N-1}) = \arg\max_{u_{N-1}\in U_{N-1}} \Big[\tilde{g}(I_{N-1},u_{N-1}) + \underset{x_{N-1}}{E}\Big\{\underset{x_N}{E}\{g_F(x_N)|x_{N-1},u_{N-1}\}|I_{N-1},u_{N-1}\Big\}\Big] \quad (3)$$

and for $t < N-1$

$$J_t(I_t) = \max_{u_t\in U_t}\Big[\tilde{g}(I_t,u_t) + \underset{y_{t+1}}{E}\{J_{t+1}(I_t,y_{t+1},u_t)|I_t,u_t\}\Big] \quad (4)$$

$$\pi(t,I_t) = \arg\max_{u_t\in U_t}\Big[\tilde{g}(I_t,u_t) + \underset{y_{t+1}}{E}\{J_{t+1}(I_t,y_{t+1},u_t)|I_t,u_t\}\Big] \quad (5)$$

Since we want the robot to achieve a position where the condition $P(y_{t+1} \geq \hat{o}|I_t,u_t) > \lambda$ is satisfied, we set the gain

function $\tilde{g}(I_t, u_t)$ to $P(y_{t+1} \geq \hat{o} | I_t, u_t)$ as given by Eq. (6). This equation makes use of the observation model, the motion model, and the current belief on the robot's location:

$$P(y_{t+1} \geq \hat{o} | I_t, u_t) = \sum_{x_{t+1}} P(y_{t+1} \geq \hat{o} | x_{t+1}) \sum_{x_t} P(x_{t+1} | x_t, u_t) P(x_t | I_t) \quad (6)$$

We also set an *execution horizon* N_e that is usually much larger than the planning horizon N. The planner first plans a motion strategy up to horizon N. This strategy is then executed. If the goal is achieved–i.e., if the robot reaches at time t a position where the condition $P(y_{t+1} \geq \hat{o} | I_t, u_t) > \lambda$ is satisfied *and* if the detector actually returns a confidence score greater than $\hat{o}$ at time $t+1$–then the confirmation process ends with success. Otherwise, a new strategy is computed (from the current robot position), again with horizon N. This iterative process ends whenever the goal is achieved or when $p \times N \geq N_e$, where p is the number of times the planner has been invoked. In the later case, the robot considers that the object C is not the target T. So, for instance, if $N = 4$ and $N_e = 100$, the planner may be called up to 25 times, if the presence of the target is not confirmed sooner.

This approach differs from entropy minimization approaches commonly used in pure localization problems [16]. Here, minimizing entropy over the state belief could lead the robot to knowing its position well, but at a location where D_T may return poor scores. Our approach does not minimize entropy by concentrating the probability mass on a particular object (the target) among several other objects as was proposed in [9], [21]. This allows us to only focus on the target's appearance and avoids the need to generate observation models for many objects that are not potential targets.

However, our approach may get trapped into local maxima in the function J_t if the horizon N is too small. We will discuss this issue in Subsection VII-D.

VII. SIMULATION RESULTS

A. General setting

To test our approach and simultaneously have control over the experimental scene, we built the textured virtual environment shown in Fig. 3, in which we can obtain synthetic pictures with realistic appearance while the robot moves to successive viewpoints.

For all the experiments reported below the planner uses the 24-cell decomposition shown in Fig. 1. For each target T, the detector D_T uses a deformable part model algorithm [6] trained on a set of images taken from a single cell c_g of the decomposition. Such algorithm has been tested in large image databases containing highly variable conditions [6], e.g., scale and light conditions. Cell c_g is selected so that distinctive features of the target are visible. Therefore the detector returns high confidence scores when the robot is in c_g. The scores of the detector tend to decrease when the viewpoint moves away from c_g, except in cases where the target has similar appearance under very different viewpoints. The scores range over 6 values, i.e., $n = 6$.

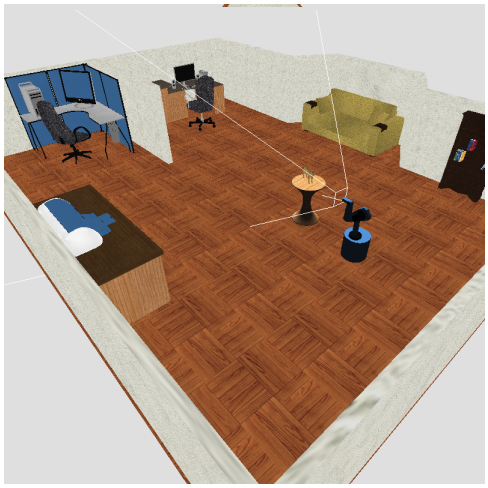
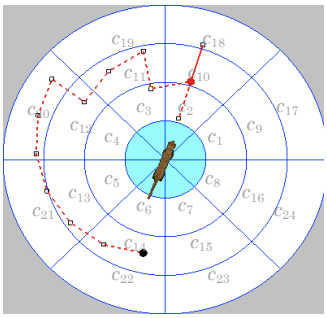


Fig. 3: Virtual environment.

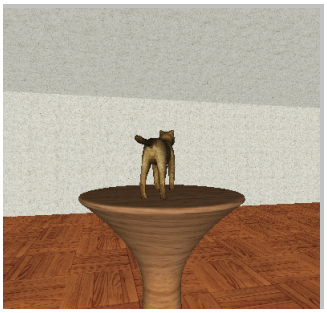
For simplicity, in the motion model we consider deterministic controls. As already mentioned, this does not eliminate the uncertainty regarding the exact robot's position since we only estimate it up to the cell resolution and the robot moves with a constant step S in the 4 predefined directions (see Section V). We also emulate a range sensor that prevents the robot from going too close or too far away from the candidate object C.

B. Experiment #1

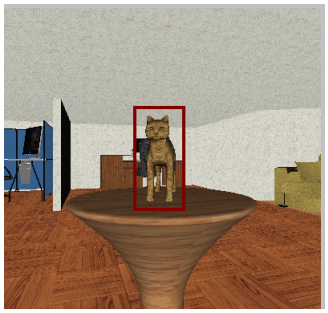
(a) Cat target



(b) Path generated with $N = 3$ and $\lambda = 0.55$



(c) Initial view



(d) View on detection

Fig. 4: Experiment #1

Here the target is the cat shown in Fig. 4a. To train the detector we chose cell c_{10} to be c_g. The training set consisted

	o_1	o_2	o_3	o_4	o_5	o_6
c_1	0.236	0.324	0.220	0.140	0.068	0.012
c_2	0.000	0.000	0.000	0.132	0.644	0.224
c_3	0.196	0.376	0.224	0.160	0.044	0.000
c_4	0.916	0.084	0.000	0.000	0.000	0.000
c_5	0.312	0.560	0.128	0.000	0.000	0.000
c_6	0.000	0.876	0.124	0.000	0.000	0.000
c_7	0.144	0.688	0.168	0.000	0.000	0.000
c_8	1.000	0.000	0.000	0.000	0.000	0.000
c_9	0.152	0.412	0.156	0.180	0.092	0.008
c_{10}	0.000	0.000	0.000	0.000	0.428	0.572
c_{11}	0.112	0.360	0.268	0.096	0.144	0.020
c_{12}	0.896	0.104	0.000	0.000	0.000	0.000
c_{13}	0.276	0.616	0.108	0.000	0.000	0.000
c_{14}	0.000	0.420	0.580	0.000	0.000	0.000
c_{15}	0.200	0.540	0.260	0.000	0.000	0.000
c_{16}	1.000	0.000	0.000	0.000	0.000	0.000
c_{17}	0.596	0.204	0.124	0.068	0.008	0.000
c_{18}	0.000	0.040	0.320	0.372	0.260	0.008
c_{19}	0.440	0.244	0.232	0.064	0.020	0.000
c_{20}	1.000	0.010	0.290	0.390	0.290	0.010
c_{21}	0.804	0.196	0.000	0.000	0.000	0.000
c_{22}	0.068	0.880	0.052	0.000	0.000	0.000
c_{23}	0.740	0.260	0.000	0.000	0.000	0.000
c_{24}	1.000	0.000	0.000	0.000	0.000	0.000

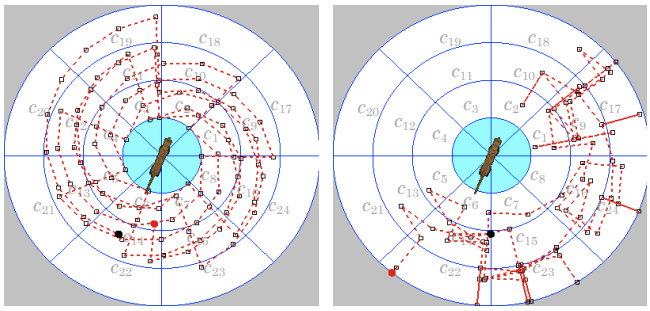
TABLE I: Cat observation model $P(y = o_j|x = c_i)$ (Experiment #1)

of 100 sample images from randomly selected viewpoints within c_{10}. Then, we generated the detector's observation model using 200 images sampled at random over each of the 24 cells (shown in Table I). As expected, with high probability, the detector returns high scores (o_5 or o_6) in c_{10}. Conversely, the probability that the detector returns the lowest scores o_1 and o_2 is greater in cells distant to c_{10}.

Using the method of Section VI, we generated motion strategies with 4 planning horizons $N = 1, 2, 3$ and 4. The robot was initially in cell c_{14} (located far away from c_{10}). The execution horizon was set to $N_e = 100$. The threshold $\hat{o}$ was set to o_6 (the highest confidence score of the detector). Runs were performed with 3 values of the probability threshold λ: 0.45, 0.50, and 0.55. Note that, together, the observation model of Table I and Eq. (6) determine the upper-bound $\max_{c_i}[P(y \geq \hat{o}|c_i)]$ for λ, beyond which it is impossible to confirm the target presence. Here this upper-bound is 0.572. It depends on the shape and resolution of the decomposition. In general, it is higher for finer decomposition.

Fig. 4b plots the path followed by the robot in a run with planning horizon $N = 3$ and probability threshold $\lambda = 0.55$. The initial position of the robot is shown with a black circle, its final configuration as a red (gray) circle, and the intermediate sensing locations as squares. The initial appearance of the cat (then the candidate target) is shown in Fig. 4c. The detector returns a low score, so that the robot needs to move to a better viewpoint. Fig. 4d shows the target's final appearance when the robot confirms the presence of the cat. In this case, the confirmation process required 15 moves (some sensing locations are repeated as the robot moves back and forth to refine its localization and eventually confirm the detection).

For comparison, we also ran separately a fixed motion strategy that consists for the robot to move around the potential target in clockwise direction. We also ran a random



(a) Fixed strategy path (b) Random strategy path

Fig. 5: Sample Trials with $\lambda = 0.55$

strategy that consists for the robot to perform a move picked uniformly at random at each step. In both cases, the maximum number of steps was set to $N_e = 100$. Fig. 5 plots the path of the robot produced by a run of the fixed strategy and the path produced by a run of the random strategy. In either cases the robot was unable to confirm the presence of the target in 100 steps.

Planning horizon	λ	# of sensing locations	Path length	Planning time (ms)	% of confirmation
1	0.45	33.698	32.580	9.854	93
	0.50	42.994	41.837	12.693	88
	0.55	42.493	41.385	12.513	88
2	0.45	12.915	11.816	37.428	100
	0.50	13.150	12.068	38.004	100
	0.55	14.385	13.305	42.443	100
3	0.45	12.837	11.471	405.655	100
	0.50	13.120	11.715	415.278	100
	0.55	13.875	12.385	440.959	100
4	0.45	12.230	11.028	35485.734	100
	0.50	12.587	11.402	37040.285	100
	0.55	13.655	12.431	40236.170	100
Random	0.45	55.750	43.344	-	22
	0.50	62.962	46.266	-	26.5
	0.55	61.354	47.201	-	15.5

TABLE II: Statistics for Experiment #1 (cat)

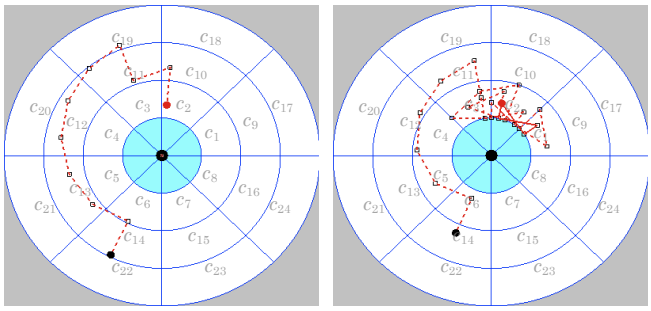
Table II shows some statistics (averages of number of sensing locations, path length, and planning time), as well as the percentage of target confirmation. Each line of the table was obtained over 200 runs with different initial positions of the robot within cell c_{14} (more runs did not produce significantly different results). Statistics for the fixed strategy are not included in the table because all 200 runs failed to confirm the target's presence. With planning horizons set to 2, 3, and 4, the presence of the target is confirmed by our method in all runs for all values of λ. When the horizon is set to 1, the percentage of confirmation drops to about 90%, but it is still much better than for the random strategy. For all values of λ, the smallest number of sensing locations and the shortest path lengths are obtained with a planning horizon of 4. The number of sensing locations and the path lengths are the largest for the random strategy.

So, as expected, we get better results when the planning horizon increases. However, the time spent in planning also increases quickly with N (and also with λ). Here, planning is

approximately 90 times longer when $N = 4$ than when $N = 3$, for approximately the same result quality. In general, the best tradeoff will depend on the specifics of the application.

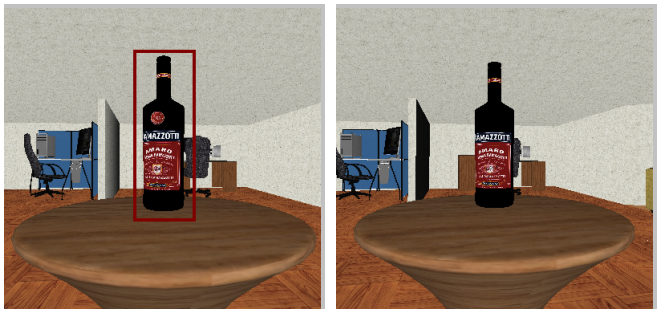
C. Experiment #2

(a) True bottle (b) False bottle



(c) Path generated with $N = 3$ and $\lambda = 0.8$ (true bottle)

(d) Path generated with $N = 3$ and $\lambda = 0.8$ (false bottle)



(e) Final view (true bottle) (f) Final view (false bottle)

Fig. 6: Experiment #2

Here we have two similar bottles, the "true" bottle (Fig. 6a) and the "false" bottle (Fig. 6b). The true bottle has a circular label that the false bottle does not have. Apart from this label, the two bottles are identical. The target is the true bottle. The purpose of this experiment is to evaluate the robot's ability to discriminate between two similar objects.

Like in experiment #1, we created the observation model of the true bottle by training the detector over a set of 100 images taken from within cell c_{10}, after placing the true bottle so that its two labels are visible from c_{10}. This model is shown in Table III. Note that in this table, the probability of getting the score o_6 is higher in in cell c_2 than in cell c_{10} (because in c_2 the robot is closer to the labels).

	o_1	o_2	o_3	o_4	o_5	o_6
c_1	0.000	0.000	0.010	0.477	0.427	0.086
c_2	0.000	0.000	0.000	0.000	0.095	0.905
c_3	0.000	0.000	0.010	0.558	0.392	0.040
c_4	0.000	0.000	0.839	0.161	0.000	0.000
c_5	0.000	0.000	1.000	0.000	0.000	0.000
c_6	0.000	0.000	1.000	0.000	0.000	0.000
c_7	0.000	0.000	0.995	0.005	0.000	0.000
c_8	0.000	0.000	0.834	0.166	0.000	0.000
c_9	0.000	0.000	0.036	0.472	0.437	0.055
c_{10}	0.000	0.000	0.000	0.000	0.317	0.683
c_{11}	0.000	0.000	0.065	0.518	0.392	0.025
c_{12}	0.000	0.010	0.900	0.090	0.000	0.000
c_{13}	0.000	0.000	1.000	0.000	0.000	0.000
c_{14}	0.000	0.055	0.945	0.000	0.000	0.000
c_{15}	0.000	0.000	1.000	0.000	0.000	0.000
c_{16}	0.000	0.126	0.819	0.055	0.000	0.000
c_{17}	0.000	0.276	0.422	0.251	0.051	0.000
c_{18}	0.000	0.000	0.186	0.523	0.291	0.000
c_{19}	0.015	0.171	0.477	0.292	0.045	0.000
c_{20}	0.070	0.598	0.332	0.000	0.000	0.000
c_{21}	0.216	0.533	0.251	0.000	0.000	0.000
c_{22}	0.256	0.693	0.051	0.000	0.000	0.000
c_{23}	0.241	0.573	0.186	0.000	0.000	0.000
c_{24}	0.312	0.523	0.165	0.000	0.000	0.000

TABLE III: Bottle observation model $P(y = o_j | x = c_i)$

Scene object	λ	# of sensing locations	Path length	Planning time (ms)	% of confirmation
True Bottle	0.80	10.820	9.346	367.723	100
	0.85	10.825	9.122	361.993	100
	0.90	12.030	9.244	415.965	99.5
False Bottle	0.80	21.333	18.002	721.861	1.5
	0.85	17	14.561	621.074	0.5
	0.90	-	-	-	0

TABLE IV: Statistics for Experiment #2 (similar bottles)

For each of the two bottles we performed runs with the planning horizon N set to 3 and the execution horizon N_e set to 30. We set the probability threshold λ to 0.80, 0.85, and 0.90. (Note that the observation model allows us to select greater values of λ than in the cat example of Experiment #1.) In each case, we collected data over 200 runs. Table IV shows the results. The robot was able to differentiate quite well between the two bottles. When the true bottle was present, it confirmed this presence in 100% of the runs when λ was set to 0.80 and 0.85, and in 99.5% of the runs when λ was set to 0.90. On the other hand, when the false bottle is present, the robot incorrectly confirms the presence of the target (the true bottle) in 1.5% of the runs (3 runs out of 200) when λ was set to 0.80, 0.5% of the runs (1 out 200) when λ was set to 0.85, and 0% of the runs when λ was set to 0.90. Figs. 6c and 6d show two paths generated when λ was set to 0.80, one when the candidate object was the true bottle, the other when it was the false bottle. In Fig. 6c the confirmation process ends in cell c_2. In Fig. 6d the planner also guided the robot toward cell c_2, but the detector failed to return high scores and the confirmation process ended with no confirmation after 30 steps.

D. Experiment #3

Here, the target is a bottle that looks as in Fig. 6a on one side and as in Fig. 6b on the opposite side. The detector is trained in cell c_{10} after placing the bottle so that the face with the circular label is visible.

We performed 200 runs with planning horizon N set to 2 and 200 runs with N set to 3. In both cases, the execution horizon N_e was set to 30 and λ to 0.85. With $N = 2$ the robot got trapped into a local maximum in the function J_t (see Eqs. (2) and (4)) located in c_5. It confirmed the presence of the bottle only in 1% of the runs. With $N = 3$ the robot confirmed the presence of the bottle in all 200 runs. This is not surprising since larger planning horizons are less prone to get stuck at local maxima. In other words, SDP delivers the global maximum for a given planning horizon, but such maximum might correspond to a local maximum for a larger planning horizon.

To address this local-maxima issue when small planning horizons are used, we added a second gain function g_F in Eq. (2), in addition to $\tilde{g}$. We defined this second gain function as a concave function with a maximum in cell c_g (here, c_{10}), so that it gives an incentive for the robot to reach cells in the neighborhood of c_g. With this additional gain function, all runs with $N = 2$, as well as with $N = 3$, confirmed the presence of the bottle.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a POMDP-based planning approach to confirm the detection of a candidate target with a mobile robot equipped with a vision sensor. Our main contributions are:

- A strategy combining robot localization and target confirmation: this strategy uses a cell-based probabilistic observation model of the target to iteratively refine a probabilistic estimation of the robot's location.
- A confirmation process that reduces the risk of false positives: this process requires that the robot reaches a position where it is highly probable that the target detector will return a high confidence score *and* where this detector actually returns such a high score.
- A mechanism to avoid local maxima of the objective function, by adding a term that gives an incentive to the robot to finish in, or close to, a given cell of the space decomposition.
- A set of experimental tests in simulation that demonstrate the ability of our approach to actually confirm the presence of selected targets and to distinguish targets from similar objects.

We are currently testing our method on a real mobile robot in an office environment. In our future research, we plan to experiment with various detectors, especially detectors trained over several cells of the space decomposition (instead of one, as is currently the case) and under several lighting conditions. We would also like to consider the presence of obstacles around the target and to add more degrees of freedom to the camera (e.g., by mounting a manipulator arm on our mobile robot and fixing the camera at the extremity of this arm). We hope that detectors trained over several cells combined with additional degrees of freedom for the camera will make it possible to deal with both visibility and collision constraints resulting from the presence of other objects around the candidate target.

REFERENCES

[1] J. Espinoza, A. Sarmiento, R. Murrieta-Cid and S. Hutchinson,"A Motion Planning Strategy for Finding an Object with a Mobile Manipulator in 3-D Environments", *Advanced Robotics*, vol. 25, no. 13-14, August 2011.

[2] L. Guibas, J. Latombe, S. LaValle, D. Lin and R. Motwani, "Visibility-Based Pursuit-Evasion in a Polygonal Environment", *Int. J. of Computational Geometry and Applications*, pp. 17-30, 1997.

[3] V. Isler, S. Kannan and S. Khanna, "Randomized Pursuit-Evasion in a Polygonal Environment", *IEEE Tr. on Robotics*, vol. 5, no. 21, pp. 864-875, 2005.

[4] T. Chung, G. Hollinger and V. Isler, "Search and Pursuit-evasion in Mobile Robotics", *Autonomous Robots*, 2011.

[5] B. Tovar and S.M. LaValle, "Visibility-Based Pursuit Evasion with Bounded Speed", *Int. J. of Robotics Research (IJRR)*, 2008.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models", *IEEE Tr. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, 2010.

[7] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *Int. J. of Computer Vision*, vol. 60, pp. 91-110, 2004.

[8] K. Grauman and T. Darrell, "The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features", *IEEE Int. Conf. on Computer Vision (ICCV)*, Beijing, China, October 2005.

[9] D. Meger, A. Gupta and J. Little, "Viewpoint Detection Models for Sequential Embodied Object Category Recognition", *IEEE Int. Conf. on Robotics and Automation*, 2010.

[10] Y. Ye and J. K. Tsotsos, "Sensor Planning for 3D Object Search", *Computer Vision and Image Understanding*, vol. 73, no. 2, 1999.

[11] J. K. Tsotsos and K. Shubina, "Attention and Visual Search: Active Robotic Vision Systems that Search", *Int. Conf. on Computer Vision Systems*, 2007.

[12] H.H. Gonzalez-Banos and J.C. Latombe, "Navigation Strategies for Exploring Indoor Environments", *Int. J. of Robotics Research (IJRR)*, vol. 21, no. 10-11, pp. 829-848, 2002.

[13] R. Pito, "A Solution to the Next Best View Problem for Automated Surface Acquisition", *IEEE Tr. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1016-1030, 1999.

[14] W. R. Scott, G. Roth and J.-F. Rivest, "View Planning for Automated Three-dimensional Object Reconstruction and Inspection", *ACM Comput. Surv.*, vol. 35, no. 1, pp. 64-96, 2003.

[15] C. Laporte and T. Arbel, "Efficient Discriminant Viewpoint Selection for Active Bayesian Recognition", *Int. J. of Computer Vision*, vol. 68, pp. 1405-1573, 2006.

[16] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*, MIT Press; 2005.

[17] S. M. LaValle, *Planning Algorithms*, Cambridge University Press; 2006.

[18] D. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientic; 2000.

[19] S. Candido and S. Hutchinson, "Minimum Uncertainty Robot Navigation using Information-guided POMDP Planning", *IEEE Int. Conf. on Robotics and Automation*, 2011.

[20] L.P. Kaelbling, M.L. Littman and A.R. Casandra, "Planning and Acting in Partially Observable Stochastic Domains", *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99-134, 1998.

[21] R. Eidenberger and J. Scharinger, "Active Perception and Scene Modeling by Planning with Probabilistic 6D Object Poses", *IEEE Int. Conf. on Intelligent Robots and Systems*, 2010.