Planning to Perceive: Exploiting Mobility for Robust Object Detection

Javier Velez[†] **Garrett Hemann**[†] **Albert S. Huang**[†] **Ingmar Posner**[‡] **Nicholas Roy**[†]

[†]MIT Computer Science and
Artificial Intelligence Laboratory
Cambridge, MA, USA

[‡]Mobile Robotics Group
Dept. of Engineering Science, Oxford University
Oxford, UK

Abstract

Consider the task of a mobile robot autonomously navigating through an environment while detecting and mapping objects of interest using a noisy object detector. The robot must reach its destination in a timely manner, but is rewarded for correctly detecting recognizable objects to be added to the map, and penalized for false alarms. However, detector performance typically varies with vantage point, so the robot benefits from planning trajectories which maximize the efficacy of the recognition system.

This work describes an online, any-time planning framework enabling the active exploration of possible detections provided by an off-the-shelf object detector. We present a probabilistic approach where vantage points are identified which provide a more informative view of a potential object. The agent then weighs the benefit of increasing its confidence against the cost of taking a detour to reach each identified vantage point. The system is demonstrated to significantly improve detection and trajectory length in both simulated and real robot experiments.

INTRODUCTION

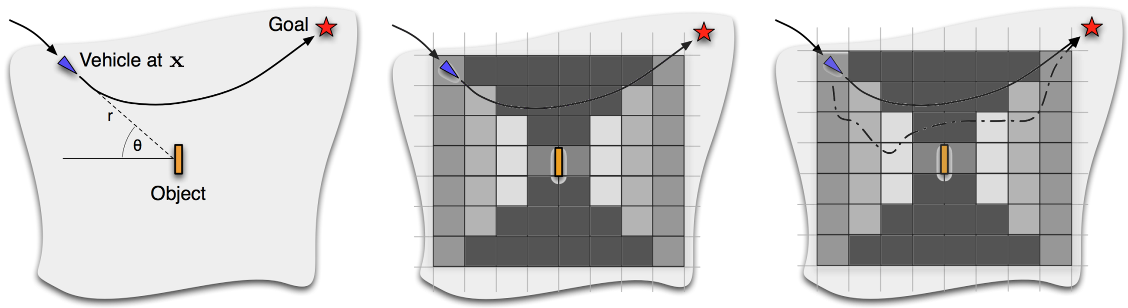
Years of steady progress in robotic mapping and navigation techniques have made it possible for robots to construct accurate maps of relatively complex environments and to robustly navigate within them (see, for example Newman et al., 2009). Increasingly, advances in vision- and laser-based object recognition are being leveraged to enrich the maps with higher-order, semantic information (e.g., Posner, Cummins, and Newman, 2009; Douillard, Fox, and Ramos, 2008; Mozos, Stachniss, and Burgard, 2005; Anguelov et al., 2004) and to thus enable more sophisticated interactions between an agent and its workspace. Commonly, these approaches to semantic mapping use standard object detection frameworks whose results are accepted *prima facie*, irrespective of sensor noise. As a consequence, shortcomings of the object detectors directly affect the quality of the map built. Rarely, however, are the capabilities of a mobile robot exploited to improve the robustness of the detection process by specifically counteracting known detector issues. Vision-based object detection, for example, is oftentimes plagued by significant performance degradation caused by a variety of factors including a change of aspect compared to that encountered in the training data and, of course, occlusion (e.g., Coates and Ng, 2010; Mittal and Davis, 2008). Both of these factors can be addressed by changing the current vantage point. Therefore, rather than placing the burden of providing perfect detections with the detector itself we argue that robots can *act* to improve their perception. Mobile robots can *plan to perceive*.

In this work, we explore how a robot's ability to selectively gather additional information about a possible object by moving to a specified location — a key advantage over more conventional, static deployment scenarios for object detectors — can improve the precision of object detections included in the map. At each step, the agent weighs the potential benefit of increasing its confidence about the detection against the cost of taking a detour to a more suitable vantage point. We make two primary contributions in this paper. Firstly, we explicitly incorporate the costs of motion when planning sensing trajectories. Previous work has largely ignored motion costs, leading to models that do not perform substantially better than greedy strategies. Secondly, we give an approximate sensor model that captures correlations between subsequent observations. Previous work typically assumes that observations are conditionally independent given the sensor position, but observations are in fact correlated by the unknown features of the environment (i.e., the partially observable objects). Inspired by recent progress in forward search for planning under uncertainty, we show that motion costs can be incorporated and correlations between measurements can be modeled, allowing us to efficiently find robust observation plans.

We begin with the problem formulation of planning trajectories to improve object detection. We describe the specific sensor model, beginning with an exact formulation and then an approximation required for efficient planning. We describe the planning algorithm and how the sensor model is incorporated. We present results on a simulated domain and a set of real-world robot experiments, demonstrating that our technique significantly improves the ability of a real robot to detect objects while reducing detours. We conclude with a discussion of related approaches and future directions.

PROBLEM FORMULATION

Consider a robot following a particular trajectory towards a goal. Assume that there are objects of interest at unknown locations, for example, a rescue robot looking for people in a first-responder scenario. Traditionally, an object detector can be used at waypoints along the trajectory where a detection is either accepted into the map or rejected based on simple detector thresholds. However, the lack of introspection



(a) (b) (c)

Figure 1: A conceptual illustration of (a) the robot at viewpoint $\mathbf{x}$ while following the original trajectory (bold line) towards the goal (red star), (b) the perception field for a particular object detector centered around an object hypothesis, and (c) an alternative path (bold dash-dotted line) along a more informative route. Cell shadings indicate the value of the conditional entropy, $H(Y|Z;\mathbf{x})$. Lighter values indicate a lower conditional entropy and therefore desirable vantage points.

of this approach regarding both the confidence of the object detector and the quality of the data gathered can lead to an unnecessary acceptance of spurious detections. Most systems simply discard lower confidence detections, and have no way to improve the estimate with further, targeted measurements. In contrast, we would like the robot to modify its motion to both minimize total travel cost and the cost of errors when deciding whether or not to add newly observed objects to the map.

Let us represent the robot as a point $\mathbf{x} \in \mathbb{R}^2$; without loss of generality, we can express a robot trajectory as a set of waypoints $\mathbf{x}^{0:K}$, with an associated motion cost c_{mot} for the travel between $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$. If the robot has a prior map of the environment and is planning a path to some pre-specified goal, then computing a minimum cost path $\mathbf{x}^{0:K}$ is a well-known and understood motion planning problem.

As the robot moves, it receives output from its object detector that gives rise to a belief over whether a detected object truly exists at the location indicated. Let the presence of an object at some location (x, y) be captured by the random variable $Y \in \{\text{object, no-object}\}$[1]. Let us also define a *decision* action $a \in \{\text{accept, reject}\}$, where the detected object is either accepted into the map (the detection is determined to correspond to a real object) or rejected (the detection is determined to be spurious). Additionally, we have an explicit cost $\xi_{dec} : \{\text{accept, reject}\} \times \{\text{object, no-object}\} \mapsto \mathbb{R}$ for a correct or incorrect accept or reject decision. We cannot know the true costs of the decisions because we ultimately do not know the true state of objects in the environment. But, we can use a probabilistic sensor model for object detections to minimize the expected cost c_{dec} of individual decision actions ξ_{dec} given the prior over objects.

We therefore formulate the planning problem as choosing a sequence of waypoints to minimize the total travel cost along the trajectory and the expected costs of the decision actions at the end of the trajectory such that the plan π is a sequence of waypoints and decision actions, $\pi \mapsto \{\mathbf{x}^{0:K} \times a\}$ for a path of length K.

[1]We assume the robot knows its own location, and has a sufficiently well-calibrated camera to assign the location of the object as well as measure the distance from the robot to the object. In this work, the uncertainty is whether or not an object is present at the given location (x, y).

A SENSOR MODEL FOR OBJECT DETECTION

In order to compute the expected cost of decision actions, we must estimate the probability of objects existing in the world, and therefore require a probabilistic model of the object detector. The key idea is that we model the object detector as a spatially varying process; around each potential object, we characterize every location with respect to how likely it is to give rise to useful information.

A measurement, z, at a particular viewpoint consists of the output of the object detector, assumed to be a real number indicating the confidence of the detector that an object exists. The distribution over the range of confidence measurements is captured by the random variable Z defined over a range $\mathcal{Z}$ of discretized states (bins). At every location $\mathbf{x}$ the posterior distribution over Y can be expressed as

$$p(y|z,\mathbf{x}) = \frac{p(z|y,\mathbf{x})p(y)}{\sum_{y \in \mathcal{Y}} p(z|y,\mathbf{x})p(y)}, \tag{1}$$

where $p(z|y,\mathbf{x})$ denotes the likelihood, for every possible state of Y, of observing a particular detector confidence at $\mathbf{x}$. This likelihood can be obtained empirically. (The expression would seem to require $p(y|\mathbf{x})$, but y is independent of the viewpoint until measurement z is received.)

For a single observation, the likelihood of any given measurement is purely a function of the underlying state as described by Equ. 1. Furthermore, for observations that are directly produced by a physical device such as a camera, we often represent the observation as conditionally independent given the state of the robot (see Fig. 2(a)). However, the observations are *not* independent given the current state, but in fact are correlated via the unknown state variable y and environment Ψ as shown in Fig. 2(b). This can be seen clearly by noting that if the robot were stationary, aimed at a static scene, we would not expect the response of the object detector on successive images to be independent. We expect observations from the object detector to be extremely correlated, with the expectation that *no* new information would be gained after more than a handful of images.

To correct our observation model we can maintain a history of observations. As more viewpoints are visited, knowledge regarding object o can be integrated recursively. Let $\mathcal{T}^K$ denote a trajectory of K locations traversed in sequence.

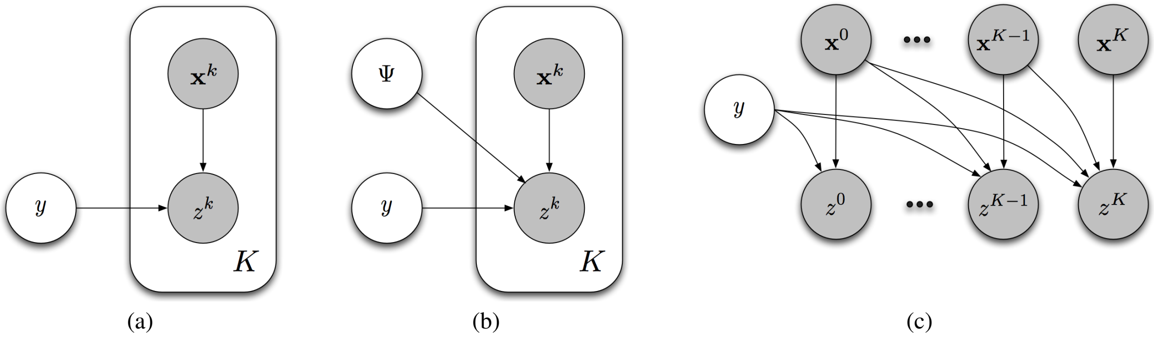


Figure 2: Different graphical models representing the observation function. (a) A naive Bayes approximation, that assumes that every observation is conditionally independent given knowledge of the object. (b) The true model that assumes that observations are independent given knowledge of the environment and the object. (c) The model employed here, in which the correlations are approximated by way of a mixture model parameterized by α as per Equ. 3.

At each location a measurement is obtained, giving a possible detection and the corresponding confidence. The trajectory is thus described by a set of K location-observation pairs, $\mathcal{T}^K = \{\{\mathbf{x}^1, z^1\}, \{\mathbf{x}^2, z^2\}, \ldots, \{\mathbf{x}^K, z^K\}\}$. Knowledge gained at each step along the trajectory can be integrated into the posterior distribution over Y such that

$$p(y|\mathcal{T}^K) = \frac{p(z^K|y, \mathbf{x}^K, \mathcal{T}^{K-1})p(y|\mathcal{T}^{K-1})}{p(z^K|\mathbf{x}^K, \mathcal{T}^{K-1})}, \qquad (2)$$

where z^K is the K^{th} observation, which depends not only on the current viewpoint but also on the history of measurements $\mathcal{T}^{K-1}$. The denominator in Equ. 2 serves to moderate the influence of the measurement likelihood on the posterior based on any correlations existing between observations taken along the trajectory. In principle, K can be arbitrarily long, so we would need a model that predicted observations given an infinite history of observations. In practice, realistic models will be difficult to come by due to the amount of data required growing exponentially with trajectory length.

We note that the images themselves are not in reality conditionally independent but correlated through the environment Ψ. If the robot position and the scene are stationary, then the probability of individual pixel values in successive images will be strongly correlated by the shared environmental representation and robot position, varying only by sensor noise. Subsequently, the object detector responses will also be strongly correlated. However, correctly representing observations in this way requires an environmental model sufficient to capture the image generation process, an intractable computational and modeling burden.

To overcome this difficulty, we approximate the real process of object detection with a simplistic model of how the images are correlated. We replace the influence of environment Ψ on correlations between observations with a convex combination of a fully uncorrelated and a fully correlated model such that the new posterior belief over the state of the world is computed as

$$p(y|\mathcal{T}^K) = \left((1-\alpha)\ \frac{p(z^K|y, \mathbf{x}^K)}{p(z^K|\mathbf{x}^K)} + \alpha\right) p(y|\mathcal{T}^{K-1}) \quad (3)$$

This captures the intuition that repeated observations from the same viewpoint add little to the robot's knowledge about the state of the world. Observations from further afield, however, become increasingly independent; Ψ has less of a correlating effect. The mixing parameter, α, is computed as a truncated linear function of the Euclidean distance, d, between two viewpoints and is normalized with respect to a maximum distance d_{max}, beyond which observations are treated as fully independent. Thus,

$$\alpha = \begin{cases} 1 - \frac{d}{d_{max}} & \Leftrightarrow d < d_{max} \\ 0 & \Leftrightarrow d \geq d_{max} \end{cases} \qquad (4)$$

In other words, no information is gained by taking additional measurements at the same location and the information content of observations increases linearly with distance from previous ones. As shown in Fig. 2(c), we remove Ψ and add a dependency between previous viewpoints and the current observation z^K. We now show how this detector model provides the information required to plan trajectories which increase the robustness of the object detections.

PLANNING TO PERCEIVE

Given the observation model described in the previous section, we now describe a planning algorithm that trades off the necessity of gaining additional information about an object hypothesis against the operational cost of obtaining this information. In particular, when an object is first detected, a new path to the original goal is planned based on the total cost function $c_t(\mathbf{x}^{0:K}, a)$, which includes both the motion cost c_{mot} along the path and the value of measurements from locations along the path expressed as a reduction in the expected cost of decision actions. The cost function consists of two terms: the motion cost $c_{mot}(\mathbf{x}^{0:K})$ and the decision cost $c_{dec}(\mathbf{x}^{0:K}, a)$, such that the optimal plan π^* is given by

$$\pi^* = \underset{\mathbf{x}^{0:K}, a}{\arg\min} \left(c_{mot}(\mathbf{x}^{0:K}) + c_{dec}(\mathbf{x}^{0:K}, a)\right), \qquad (5)$$

$$c_{dec}(\mathbf{x}^{0:K}, a) = E_{y|\mathbf{x}^{0:K}}[\xi_{dec}(a, y)], \qquad (6)$$

where $E_{y|\mathbf{x}^{0:K}}[\cdot]$ denotes the expectation with respect to the robot's knowledge regarding the object, after having executed path $\mathbf{x}^{0:K}$.

Motion cost The path cost, $c_{mot}(\mathbf{x}^{0:K})$, encompasses operational considerations such as power expended and time taken when moving along a particular trajectory and is typically proportional to the length of the trajectory.

Decision Cost The decision cost, $c_{dec}(\mathbf{x}^{0:K}, a)$, not only captures the expected cost of accepting (or rejecting) a potential object detection, but it also captures the expected yield in information from observations along path $\mathbf{x}^{0:K}$. The trajectory affects the cost of the decision actions in terms of changing the expectation, rather than the decision actions themselves, in effect allowing the algorithm to decide if more observations are needed.

Note that the decision actions can be treated independently of each other and also independently of the robot motion, which allows us to compute the expected decision costs very efficiently. We take advantage of this efficiency to move the minimization over decision actions directly inside the cost function. Abusing notation for c_{dec}, we have

$$c_{dec}(\mathbf{x}^{0:K}) = \arg\min_{a} c_{dec}(\mathbf{x}^{0:K}, a) \tag{7}$$

$$= \arg\min_{a} E_{y|\mathbf{x}^{0:K}}[\xi_{dec}(a, y)] \tag{8}$$

Next, we can write the plan in terms of $\mathbf{x}^{0:K}$.

$$\pi^* = \arg\min_{\mathbf{x}^{0:K}} \left(c_{mot}(\mathbf{x}^{0:K}) + c_{dec}(\mathbf{x}^{0:K})\right) \tag{9}$$

$\xi_{dec}(\text{accept}, \cdot)$ and $\xi_{dec}(\text{reject}, \cdot)$ are the costs associated with declaring that the object exists or not, respectively, after measuring z at $\mathbf{x}$ following traversal of trajectory $\mathcal{T}^{K-1}$. These costs include the penalties imposed when accepting a true positive detection and when accepting a false positive detection, respectively.

The expectation inside Equ. 8 relies on a model of y conditioned on the trajectory $\mathbf{x}^{0:K}$; as can be seen in Fig. 2(c), y and $\mathbf{x}^{0:K}$ are correlated through z. During planning, the actual z that will be received cannot be known ahead of time, so the expectation must be taken with respect to both the object state y and the received observations, as in

$$E_{y|\mathbf{x}^{0:K}}[\xi_{dec}(a, y)] = \sum_{z\in\mathcal{Z}} \left(p(z|\mathbf{x}^K, \mathcal{T}^{K-1}) E_{y|z,x^{0:K-1}}[\xi_{dec}(a, y)]\right), \tag{10}$$

where $p(z|\mathbf{x}^K, \mathcal{T}^{K-1})$ denotes the probability of obtaining a particular detector confidence value when observing the object from $\mathbf{x}$ given a previous trajectory $\mathcal{T}^{K-1}$, and is computed akin to the posterior in Equ. 2.

The planning process therefore proceeds by searching over sequences of $\mathbf{x}^{0:K}$, evaluating paths by computing expectations with respect to both the observation sequences and the object state. The paths with the lowest decision cost will tend to be those leading to the lowest posterior entropy, avoiding the large penalty for false positives or negatives.

Multi-step Planning

The algorithm scales approximately exponentially in the length of the planning horizon K and thus rapidly becomes intractable as more observations are considered. We therefore adopt a roadmap scheme in which a fixed number of locations are sampled every time a new viewpoint is to be added to the current trajectory. A graph is built between the sampled poses, with straight-line edges between samples.

The sampling scheme is biased towards locations more likely to lead to a useful observation — that is, an observation which, on average, reduces the current uncertainty in Y. Given a viewpoint $\mathbf{x}^K$ and the trajectory $\mathcal{T}^{K-1}$ visited thus far, the reduction in uncertainty is captured by the mutual information between the object state Y and the observation Z received at $\mathbf{x}^K$ such that

$$I(Y, Z; \mathbf{x}^K, \mathcal{T}^{K-1}) = H(Y; \mathcal{T}^{K-1}) - H(Y|Z; \mathbf{x}^K, \mathcal{T}^{K-1}), \tag{11}$$

where $H(Y; \mathcal{T}^{K-1})$ and $H(Y|Z; \mathbf{x}^K, \mathcal{T}^{K-1})$ denote the entropy and the conditional entropy, respectively (we drop the $\mathbf{x}^K$ from the entropy since the distribution of Y is independent of the robot being at $\mathbf{x}^K$ without the corresponding observation Z). Thus, $H(Y; \mathcal{T}^{K-1})$ expresses the certainty of the current belief over whether the object exists or not given the trajectory thus far, unswayed by any new measurements. At every time step, this term is constant for every viewpoint considered and is therefore disregarded. The conditional entropy in Equ. 11 can be expanded in terms of the posterior over the state of the hidden variable Y given the previous trajectory $\mathcal{T}^{K-1}$ and an additional measurement taken at $\mathbf{x}^K$, $p(y|z, \mathbf{x}^K, \mathcal{T}^{K-1})$ (c.f. Equs. 2 and 3), and the likelihood of Z taking a particular value conditioned on the trajectory thus far and whether an object viewed from $\mathbf{x}^K$ is present or not, $p(z|\mathbf{x}^K, \mathcal{T}^{K-1})$,

$$H(Y|Z; \mathbf{x}^K, \mathcal{T}^{K-1}) = -\sum_{z\in\mathcal{Z}} \left[p(z|\mathbf{x}^K, \mathcal{T}^{K-1}) H(Y|z, \mathbf{x}^K, \mathcal{T}^{K-1})\right] \tag{12}$$

Given the sensor model described previously, and Equ. 3 in particular, the conditional entropy can be readily evaluated using empirically determined quantities.

Conditional entropy values for all viewpoints in the robot's workspace form the *perception field* for a particular object hypothesis (see Fig. 1(b)). This field is used to bias sampling towards locations with high expected information gain. Due to the correlations between individual observations made over a trajectory of viewpoints, the perception field changes as new observations are added. In particular, the naïve correlation model imposed in this work forces $I(Y, Z; \mathbf{x}^K, \mathcal{T}^{K-1}) = 0$ when considering measurements from viewpoints already visited. In other words, the robot will prefer to observe the putative object from different viewpoints over taking repeated measurements at the same place.

Algorithm `replan_on_new_detection` (Fig. 3) summarizes the planned-viewpoints approach of sampling and evaluating trajectories to balance increased confidence with motion costs.

Multiple Objects

We formally define a *vantage point* relative to an object o, $\mathbf{v}_o \in \mathbb{R}^M$, as a vector in an M-dimensional feature space describing the configuration of the robot relative to the potential object. We also define a mapping $F : \mathbb{R}^2 \times O \mapsto \mathbb{R}^M$ between a robot viewpoint $\mathbf{x}$ and its corresponding vantage point $\mathbf{v}_o = F(\mathbf{x}, o)$. In principle, a vantage point need not be restricted to spatial coordinates but may incorporate additional information such as, for example, the degree of occlusion experienced or image contrast (for an appearance based detector). In this work, however, only the range, r, and aspect, θ, relative to the object are considered such that

Algorithm `replan_on_new_detection`

Input: an object detection z

update perception field P with z
update object belief with z (Equ. 3)
while planning time remains **do**
    $T_i \sim P$ *// Sample from perception field*
    $\mathbf{T} \leftarrow \mathbf{T} \cup T_i$
$T^* \leftarrow \underset{T_i \in \mathbf{T}}{\arg\min} \, (c_t(T_i))$
execute trajectory T^*

Figure 3: The viewpoint planning algorithm samples trajectories using the perception field, then chooses the trajectories which balance increasing the robot's confidence about an object with minimizing trajectory costs (Equ. 5).

$\mathbf{v}_o \in \mathbb{R} \times \mathbb{S}^1$ (see Fig. 1(a)). The planning approach described so far can be extended to planning in an environment with M object hypotheses $\mathcal{O}^M = \{o^1, o^2, \ldots, o^M\}$ by considering a modified cost function which simply adds the cost for each object. In this work we consider an object's existence to be independent of other objects hence the individual object perception fields are additive for a particular viewpoint $\mathbf{x}$. Given *no* prior information about object locations, we do not hypothesize how many objects there are in the world. We initially let $M = 0$ and run the object detector during the robot motion. After each image is processed by the object detector, if the detector determines that the probability of an object at some new location is above a threshold, the number of object hypotheses M is increased and the robot replans.

EXPERIMENTS

We tested our approach on both a simulated robot with an empirically derived model of an object detector and on an actual autonomous wheelchair (Fig. 4) using a vision-based object detector (Felzenszwalb, McAllester, and Ramanan, 2008). In both the simulation and physical experiments, the robot was tasked with reaching a manually specified destination. The robot was rewarded for correctly detecting and reporting the location of doors in the environment, penalized for false alarms, and incurred a cost proportional to the length of its total trajectory. We chose doors as objects of interest due to their abundance in indoor environments and their utility to mobile robots – identifying doorways is often a component of higher level tasks.

Our autonomous wheelchair (Fig. 4) is equipped with onboard laser range scanners, primarily used for obstacle sensing and navigation, and a Point Grey Bumblebee2 color stereo camera. The simulation environment is based on empirically constructed models of the physical robot and object detector. All experiments were run with d_{max} empirically chosen to be 3 meters. We set $\xi_{dec}(\text{reject}, \cdot)$ to zero, indicating no penalty for missed objects.

Object detector We used the object detector due to Felzenszwalb, McAllester, and Ramanan (2008). The detector was trained on approximately 1400 positive and 2000 negative examples from manually labeled images collected from a large range of indoor areas excluding our testing environment. The precision-recall curve for the detector on a set of images from the testing environment is shown in Fig. 5. Performance on images from the testing environment was low due to false positives triggered by visual structures not present in the training images. The detector could be re-trained to improve performance, but the problem recurs when new environments are encountered.

Figure 4: Robotic wheelchair platform

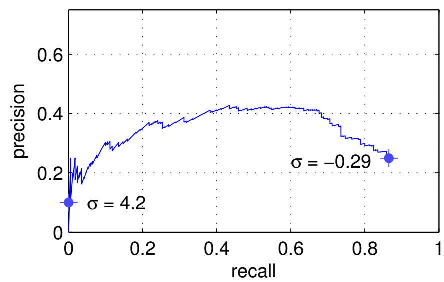


Figure 5: A precision/recall graph for the object detector, when evaluated on images from the test environment. Detector score thresholds (σ) for two data points are shown.

The object detector produces image-space bounding-boxes for each detected object, along with a scalar confidence value. Stereo disparity was used to estimate object distance, and a plane-fitting procedure provided estimates of object orientation. An observation likelihood model (Equ. 1) was constructed by binning the results of the object detector on the 3400 training examples.

Fig. 6 shows the perception field for the detector model learned from this data, with each cell indicating the conditional entropy of the posterior distribution over the true state of an object given an observation from this viewpoint, $p(y|z, \mathbf{x})$ and a uniform initial belief. Each point represents the conditional entropy for a specific robot pose where the robot is pointing at the object (brighter regions correspond to lower conditional entropy). Intuitively, this suggests that the viewpoint from which a single observation is most likely to result in a low-entropy posterior belief is approximately 9 m directly in front of the door. Viewpoints too close to or far from the door, and those from oblique angles are less likely to result in high-confidence posterior beliefs.

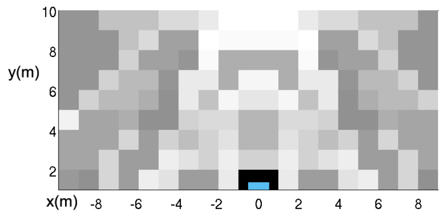


Figure 6: Learned perception field for a possible door. The unknown object is centered at the origin (blue). Brighter regions correspond to viewpoints more likely to result in higher confidence posterior beliefs.

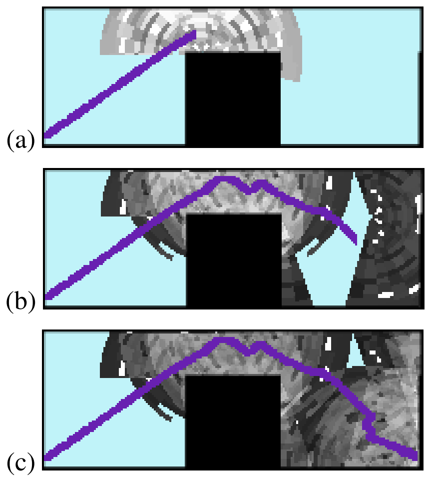


Figure 7: Evolution of the perception field during a simulation run. As more detections are received, regions that were initially rich in information (a) grow dimmer (c), indicating fewer useful observations can be made from those regions.

Simulation Results

We first assessed our planning approach using the learned model in a simulated environment. Our simulation environment consisted of a robot navigating through an occupancy map, with object detections triggered according to the learned object detection model and correlation model α. We also simulated false positives by placing non-door objects that probabilistically triggered object detections using the learned model for false-alarms. The processing delay incurred by the actual object detector was also simulated (the object detector requires approximately 2 seconds to process a spatially decimated 512x384 pixel image).

Comparison Algorithms For the simulation trials we compared our algorithm against three other algorithms. The Random$_\beta$ algorithm repeatedly obtained observations from randomly selected viewpoints near detected objects until the belief of each object exceeded a threshold β, and then continued on to the original destination. The Greedy$_\beta$ algorithm selected the *best* viewpoint according to our perception field for each potential object until the belief of each object exceeded a threshold β. Lastly, we compared our algorithm against the RTBSS online POMDP algorithm (Paquet, Tobin, 2005). We chose a maximum depth of 2 and modeled the world using a resolution comparable to d_{max} of 2.5 meters for the RTBSS algorithm.

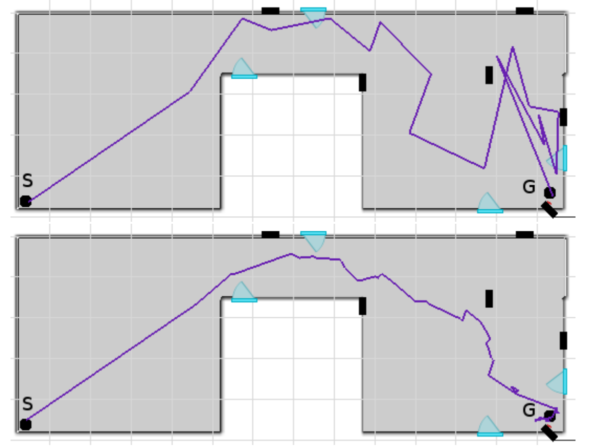


Figure 8: Simulation scenario with multiple doors (cyan triangles) and non-doors (black bars). The robot starts at the bottom-left (S) with a goal near the bottom-right (G). The robot's object detector responds to both doors and non-doors. Top: example trajectory (purple) executed using random viewpoint selection. Bottom: example trajectory (purple) executed using planned viewpoints.

Single Door Simulation First, we tested our planning algorithm on a small simulation environment with one true door and two non-doors. Table 1 shows the results of 50 trials. Overall, explicitly planning viewpoints resulted in significantly higher performance. The planned viewpoints algorithm performed better than RTBSS in terms of precision and recall, most likely because our algorithm sampled continuous-space viewpoints and the RTBSS algorithm had a fixed discrete representation, while RTBSS paths were shorter.

Multi Door Simulation Next, we evaluated our algorithm in a larger, more complex scenario containing four doors and six non-door objects. Figure 7 shows the evolution of the perception field over time for a particular simulation run. Figure 8 shows the multiple door simulation environment and example trajectories planned and executed by the planned and Random$_\beta$ algorithms.

Table 2 shows the simulation results for the multi-door scenario. Our planned viewpoints algorithm resulted in the second shortest paths after Greedy$_{\beta=0.6}$ but with superior detection performance. Planned viewpoints also resulted in significantly shorter paths than RTBSS given the same operating point on the ROC curve. Investigating the relationship between our algorithm and RTBSS is a potential avenue for future research.

Physical Wheelchair Trials

We conducted a small experiment comparing our planned viewpoints algorithm and the Greedy$_{\beta=0.8}$ on a robot wheelchair platform (Fig. 4). The robot was given a goal position such that a nominal trajectory would bring it past one true door, and near several windows that trigger object detections.

Figure 9 illustrates the trajectory executed during a single trial of the planned viewpoints algorithm, and Table 3

Average	Random$_{\beta=0.8}$	Random$_{\beta=0.6}$	Greedy$_{\beta=0.8}$	Greedy$_{\beta=0.6}$	Planned	RTBSS
Precision	0.27 ±0.03	0.26 ±0.04	0.31 ±0.06	0.60 ±0.07	0.75 ±0.06	0.45 ±0.06
Recall	0.72 ±0.06	0.60 ±0.07	0.44 ±0.07	0.62 ±0.07	0.80 ±0.06	0.58 ±0.07
Path Length (m)	62.63 ±0.02	62.03 ±0.67	67.08 ±2.23	41.95 ±0.88	54.98 ±3.04	47.57 ±0.19
Total Trials	50	50	50	50	50	50

Table 1: Simulation performance on single door scenario, with standard error values.

Average	Random$_{\beta=0.8}$	Random$_{\beta=0.6}$	Greedy$_{\beta=0.8}$	Greedy$_{\beta=0.6}$	Planned	RTBSS
Precision	0.64 ±0.03	0.67 ±0.03	0.64 ±0.03	0.54 ±0.03	0.53 ±0.05	0.70 ±0.03
Recall	0.64 ±0.04	0.69 ±0.03	0.63 ±0.02	0.57 ±0.03	0.76 ±0.03	0.66 ±0.03
Path Length (m)	199.62 ±11.24	161.36 ±6.13	153.32 ±4.37	121.35 ±1.32	138.21 ±7.12	160.74 ±6.08
Total Trials	50	50	50	50	50	50

Table 2: Simulation performance on multiple door scenario, with standard error values.

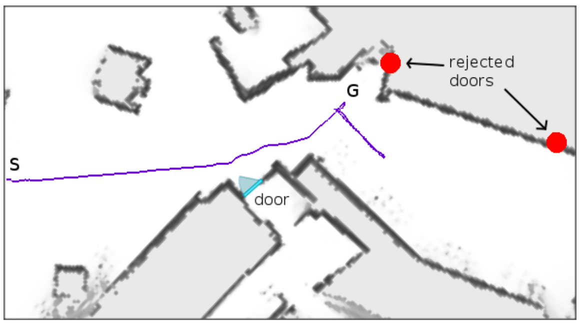


Figure 9: Trajectory executed on the actual robot wheelchair using planned viewpoints from 'S' to 'G' where the robot discovers one true door (cyan). Near the goal, it detects two more possible doors (red dots), detours to inspect them, and (correctly) decides that they are not doors.

Average	Greedy$_{\beta=0.8}$	Planned
Precision	0.53 ±0.14	0.7 ±0.15
Recall	0.60 ±0.14	0.7 ±0.15
Path Length (m)	153.86 ±33.34	91.68 ±15.56
Total Trials	10	10

Table 3: Results of real-world trials using robot wheelchair.

summarizes the results of all trials. Our planned viewpoints algorithm resulted in significantly shorter trajectories while maintaining comparable precision and recall. For doors detected with substantial uncertainty, our algorithm planned more advantageous viewpoints to increase its confidence and ignored far away detections because of high motion cost.

RELATED WORK

The problem of planning motion trajectories for a mobile sensor has been explored by a number of domains including planning, sensor placement, active vision and robot exploration. The most general formulation is the partially observable Markov decision process Sondik (1971). Exact solutions to POMDPs are computationally intractable, but recent progress has led to approximate solvers that can find good policies for many large, real-world problems (Pineau, Gordon, and Thrun, 2006; Smith and Simmons, 2005; Kurniawati, Hsu, and Lee, 2008; Kurniawati et al., 2010). However, the complexity of representing even an approximate POMDP solution has led to forward search strategies for solving POMDPs (Ross et al., 2008; Prentice and Roy, 2009; He, Brunskill, and Roy, 2010). The approach presented here is inspired by these forward search algorithms, but incorporates a more complex model that approximates the correlations between observations.

In contrast to POMDP models of active sensing, the controls community and the sensor placement community use information-theoretic models, where the goal is *only* to minimize a norm of the posterior belief, such as the entropy. This objective function does not depend on the motion costs of the vehicle, and is sub-modular (Krause and Guestrin, 2007). As a consequence, greedy strategies that choose the next-most valuable measurement can be shown to be boundedly close to the optimal, and the challenge is to generate a model that predicts this next-best measurement (Guestrin, Krause, and Singh, 2005; Krause et al., 2008). In terms of image processing and object recognition, Denzler and Brown (2002) and Sommerlade and Reid (2010) showed that information-theoretic planning could be used to tune camera parameters to improve object recognition performance and applied to multi-camera systems, although their use of exhaustive search over the camera parameters "rapidly becomes unwieldy." Lastly, Sridharan, Wyatt, and Dearden (2008) showed that by formulating an information-theoretic problem as a decision-theoretic POMDP, true multi-step policies did improve the performance of a computer vision system in terms of processing time. However, all of these previous algorithms use models for sequential decision making where the costs of the actions are independent (or negligible), leading to a submodular objective function and limited improvement over greedy strategies.

There has been considerable work in view point selection in active vision which is impossible to review comprehensively. A few relevant pieces of work include Arbel and Ferrie (1999) and more recently Laporte and Arbel (2006) who use a Bayesian approach to model detections that is related to ours, but only search for the next-best viewpoint, rather than computing a full plan. The work of Deinzer, Denzler, and Niemann (2003) is perhaps most similar to ours in that the viewpoint selection problem is framed using reinforcement learning, but again the authors "neglect costs for camera movement" and identify the absence of costs as a limitation of their work. Similarly, Mittal and Davis (2008) learn a model of object occlusion and use simulated annealing to solve for the optimal plan; their contribution is to learn a predictive model of good viewpoints.

In robot exploration, where the goal is to generate robot trajectories that learn the most accurate and complete map with minimum travel cost, the costs of motion must be incorporated. Bourgault et al. (2002) developed a full exploration planner that incorporated an explicit tradeoff between motion plans and map entropy. Stachniss, Grisetti, and Burgard (2005) described a planner that minimized total expected cost, but only performed search over the next-best action. To address the computational challenge, Kollar and Roy (2008) used reinforcement learning to both learn a model over the expected cost to the next viewpoint in the exploration, and minimize the total expected cost of a complete trajectory.

The contribution of our work over the existing work is primarily to describe a planning model that incorporates both action costs and detection errors, and specifically to give an approximate observation model that captures the correlations between successive measurements that still allows forward-search planning to operate, leading to an efficient multi-step search to improve object detection.

CONCLUSIONS AND FUTURE WORK

Previous work in planned sensing has largely ignored motion costs of planned trajectories and used simplified sensor models with strong independence assumptions. In this paper, we presented a sensor model that approximates the correlation in observations made from similar vantage points, and an efficient planning algorithm that balances moving to highly informative vantage points with the motion cost of taking detours. The performance of our algorithm could be further improved by future work in both the sensor model and planning technique.

We currently approximate the true observation model with a hand-tuned mixture of fully correlated and fully uncorrelated models. In the future, we hope to learn correlation parameters from data, and incorporate occlusion into the observation model. Additionally, our planning algorithm uses a forward search approach guided by a perception field, but is limited in search depth. Techniques that trade off search depth for breadth, such as heuristic-guided macro-action forward search, may be useful for increasing our algorithm's planning horizon.

References

Anguelov, D.; Koller, D.; Parker, E.; and Thrun, S. 2004. Detecting and modeling doors with mobile robots. In *Proc. ICRA*.

Arbel, T., and Ferrie, F. P. 1999. Viewpoint selection by navigation through entropy maps. In *Proc. ICCV*.

Bourgault, F.; Makarenko, A. A.; Williams, S. B.; Grocholsky, B.; and Whyte, D. H. F. 2002. Information based adaptive robotic exploration. In *Proc. IROS*.

Coates, A., and Ng, A. Y. 2010. Multi-camera object detection for robotics. In *Proc. ICRA*.

Deinzer, F.; Denzler, J.; and Niemann, H. 2003. Viewpoint selection - planning optimal sequences of views for object recognition. In *Proc. ICCV*. Springer.

Denzler, J., and Brown, C. M. 2002. Information theoretic sensor data selection for active object recognition and state estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 24(2):145–157.

Douillard, B.; Fox, D.; and Ramos, F. 2008. Laser and Vision Based Outdoor Object Mapping. In *Proc. RSS*.

Felzenszwalb, P.; McAllester, D.; and Ramanan, D. 2008. A discriminatively trained, multiscale, deformable part model. In *Proc. CVPR*.

Guestrin, C.; Krause, A.; and Singh, A. 2005. Near-optimal sensor placements in gaussian processes. In *Proc. ICML*.

He, R.; Brunskill, E.; and Roy, N. 2010. Puma: Planning under uncertainty with macro-actions. In *Proc. AAAI*.

Kollar, T., and Roy, N. 2008. Trajectory optimization using reinforcement learning for map exploration. *International Journal of Robotics Research* 27(2):175–197.

Krause, A., and Guestrin, C. 2007. Near-optimal observation selection using submodular functions. In *Proc. AAAI*.

Krause, A.; Leskovec, J.; Guestrin, C.; VanBriesen, J.; and Faloutsos, C. 2008. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management* 134:516.

Kurniawati, H.; Du, Y.; Hsu, D.; and Lee, W. 2010. Motion planning under uncertainty for robotic tasks with long time horizons. *International Journal of Robotics Research*. To appear.

Kurniawati, H.; Hsu, D.; and Lee, W. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. RSS*.

Laporte, C., and Arbel, T. 2006. Efficient discriminant viewpoint selection for active bayesian recognition. *International Journal of Computer Vision* 68(3):267–287.

Martinez, O. M.; Stachniss, C.; and Burgard, W. 2005. Supervised Learning of Places from Range Data using Adaboost. In *Proc. ICRA*.

Mittal, A., and Davis, L. 2008. A general method for sensor planning in multi-sensor systems: Extens ion to random occlusion. *International Journal of Computer Vision* 76:31–52.

Newman, P.; Sibley, G.; Smith, M.; Cummins, M.; Harrison, A.; Mei, C.; Posner, I.; Shade, R.; Schroeter, D.; Murphy, L.; Churchill, W.; Cole, D.; and Reid, I. 2009. Navigating, recognizing and describing urban spaces with vision and laser. *International Journal of Robotics Research*.

Paquet, S. and Tobin, L 2005 Real-Time Decision Making for Large POMDPs. In *Proc. Canadian Conference on Artificial Intelligence* .

Pineau, J.; Gordon, G.; and Thrun, S. 2006. Anytime point-based approximations for large POMDPs. *Journal of Artificial Intelligence Research* 27:335–380.

Posner, I.; Cummins, M.; and Newman, P. 2009. A generative framework for fast urban labeling using spatial and temporal context. *Autonomous Robots*.

Prentice, S., and Roy, N. 2009. The belief roadmap: Efficient planning in belief space by factoring the covariance. *International Journal of Robotics Research* 8(11-12):1448–1465.

Ross, S.; Pineau, J.; Paquet, S.; and Chaib-draa, B. 2008. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research* 32(1):663–704.

Smith, T., and Simmons, R. 2005. Point-based POMDP algorithms: Improved analysis and implementation. In *Proc. UAI*.

Sommerlade, E., and Reid, I. 2010. Probabilistic surveillance with multiple active cameras. In *Proc. ICRA*.

Sondik, E. J. 1971. *The Optimal Control of Partially Observable Markov Processes*. Ph.D. Dissertation, Stanford University.

Sridharan, M.; Wyatt, J.; and Dearden, R. 2008. HiPPo: Hierarchical POMDPs for Planning Information Processing and Sensing Actions on a Robot. In *Proc. ICAPS*.

Stachniss, C.; Grisetti, G.; and Burgard, W. 2005. Information gain-based exploration using rao-blackwellized particle filters. In *Proc. RSS*.