Noname manuscript No.
(will be inserted by the editor)

Active Object Tracking using Context Estimation: Handling Occlusions and Detecting Missing Targets

Minkyu Kim · **Luis Sentis**

arXiv:1912.06754v1 [cs.RO] 14 Dec 2019

Received: date / Accepted: date

Abstract When performing visual servoing or object tracking tasks, active sensor planning is essential to keep targets in sight or to relocate them when missing. In particular, when dealing with a known target missing from the sensor's field of view, we propose using prior knowledge related to contextual information to estimate its possible location. To this end, this study proposes a Dynamic Bayesian Network that uses contextual information to effectively search for targets. Monte Carlo particle filtering is employed to approximate the posterior probability of the target's state, from which uncertainty is defined. We define the robot's utility function via information theoretic formalism as seeking the optimal action which reduces uncertainty of a task, prompting robot agents to investigate the location where the target most likely might exist. Using a context state model, we design the agent's high-level decision framework using a Partially-Observable Markov Decision Process. Based on the estimated belief state of the context via sequential observations, the robot's navigation actions are determined to conduct exploratory and detection tasks. By using this multimodal context model, our agent can effectively handle basic dynamic events, such as obstruction of targets or their absence from the field of view. We implement and demonstrate these capabilities on a mobile robot in real-time.

Keywords Active perception · Context estimation · Object tracking · POMDP

Minkyu Kim
2617 Wichita St, Austin, TX 78712
Tel.: +1-714-222-2011
E-mail: steveminq@utexas.edu

Luis Sentis
2617 Wichita St, Austin, TX 78712
E-mail: lsentis@austin.utexas.edu

1 Introduction

This paper addresses visual-based active object tracking using mobile sensor platforms. Recently, low cost vision sensors and object detection algorithms have been broadly available Redmon et al. (2016), upping the development of new object tracking capabilities. Visual information of objects can easily incorporate semantic information of known targets to resolve ambiguity related to data association in cluttered environments Makris et al. (2011). However, vision-based object tracking suffers from occlusion from overlapping objects, or missing targets from the field of view (FOV) Xiang et al. (2015) Yun et al. (2017). Although many researchers have investigated approaches to change camera states in order to keep tracking targets in sight, there is little work to solve the problem of long-term occlusions or missing targets in crowded scenes. Therefore, the focus of previous research has shifted from passive to active perception approaches (sensing) to handle more dynamic environments Kaelbling and Lozano-Pérez (2012) Eidenberger et al. (2009b).

Bajcsy firstly described active perception as "a problem of controlling strategies applied to the data acquisition process which depends on the current state of the data interpretation and the task of the process" Bajcsy (1988). Similarly, this paper proposes an active perception framework that seeks optimal control inputs that reduce target uncertainty based on information theoretic costs Eidenberger et al. (2009a). Information theoretic approaches have been widely used in state-estimation and control of mobile sensor systems such as mapping Charrow et al. (2015) Julian et al. (2014), Simultaneous localization and mapping (SLAM) Valencia and Andrade-Cetto (2018), or object pose estimation Wu et al. (2015). Related to our work, Ryan and Hedrick (2010) uses Bayesian estimation and information theoretic cost to reduce uncertainty of target locations.

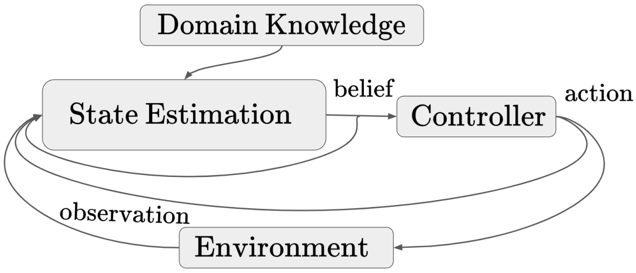


Fig. 1 The proposed Markov Decision Process: incorporating domain knowledge (context model).

Table 1 Nomenclature

Symbol	description
x	target states
q	robot (sensor) state configurations
u	robot inputs
c	context states
z	measurement of target states
Θ	measurement of context states

Object tracking with Bayesian estimation lacks support for dealing with the absence of targets since direct measurements might not be available. In such situations, we convert the tracking problem into an object search problem. Object search can be solved by finding the most informative actions to re-locate lost targets. Since there is no direct observation of an object, we aim to obtain an optimal action policy based on the probabilistic belief of possible target locations.

Bourgault et al. (2003) investigated optimal search strategies to minimize the expected time to find lost targets within probabilistic frameworks. Other probabilistic formulations for object search have been considered such as Bertuccelli and How (2006), Lau et al. (2006), and Chung and Burdick (2012). Most of these methods reduce the search space using a discrete-grid world or graphical structure, since computing the optimal search actions with uncertainty is known to be an NP-hard problem Tseng and Mettler (2017). Reformulating optimal search by reducing the search space mitigates the computational burden. However, these methods do not deal with dynamic situations such as occlusions or missing targets. Other works have shown progress recovering missing targets such as Radmard and Croft (2017) or Radmard et al. (2018). In addition, an occlusion-aware planning strategy for multiple robots has been studied using an information theoretic approach Hausman et al. (2016).

Inferring the possible location of targets draws inspiration from human perceptual capabilities Aydemir et al. (2013). Our research proposes estimating possible target locations based on predicted context information. By using prior knowledge of the target's surrounding context or past experience of the sensing process, the quality of a target's prediction can be improved significantly. From this perspective, providing context information to a state estimator is strongly desirable Denzler and Brown (2002) Kaelbling and Lozano-Pérez (2013). The main challenge with this concept is that this estimation framework needs to be formulated so that it can be used simultaneously to address tracking and searching of objects. To resolve this difficulty, this study proposes to bridge context (domain knowledge) and target estimation using tools such as Dynamic Bayesian Networks (DBN) and particle filters.

We estimate context states using probabilistic beliefs. Computing an optimal action over the belief space can be formulated as a Partially-Observable Markov Decision Process (POMDP). There exist approximate solutions for low dimensional systems Porta et al. (2006) and locally optimal solutions for continuous state and action spaces Van Den Berg et al. (2012) Bai et al. (2014). However, similarly to Sridharan et al. (2010) Li et al. (2016) we limit the use of POMDP to high-level planning in order to achieve real-time performance.

By estimating their context, our robots can take optimal decisions based on increasing the information gain. For example, when a target object is present, our robot will attempt to keep targets in its FOV by controlling its head or mobile base. When an occlusion occurs, our robot will move to the optimal sensor configuration obtained by solving the alluded information theoretical control problem. In addition, when targets suddenly disappear, our robot will search for them based on context information. For our study, we assume that objects can not move by themselves but only by nearby people. With this assumption, our robot's logical action is to find targets near observed people. When no person is present, our robot will start exploring the area around it to locate people.

In summary, the main contributions of this work are 1) devising a probabilistic framework for leveraging context information with DBN for active object tracking, 2) using POMDP to achieve high-level decision-making using information-theoretic costs, which we approximate using particle filters, and 3) integrating a realistic demonstration with a mobile robot for validating the newly defined capabilities.

2 State Estimation

2.1 Bayesian Filtering: Active Object Tracking

Bayesian filtering is a great tool to estimate the state of dynamic systems recursively in a probabilistic manner. This approach attempts to construct the posterior probability of a state based on a sequence of observations. In general, an active object tracking problem (object tracking with active sensing) can be formulated with a set of models: motion transition model $p(x_k|x_{k-1})$, sensor (robot) motion model $p(q_k|q_{k-1}, u_{k-1})$, and measurement model $p(z_k|x_k, q_k)$, where $x_k \in \Re^3$, q_k, u_k, and z_k are target states, sensor (robot) states, control inputs, and observations at time step k, respectively. In this study, a target state is described as the 3D position of an object, and q_k denotes the robot's base and its camera's configuration, namely, $q_k = (x, y, \theta, q_{tilt}, q_{pan})$. Here, a sensor motion model is the probabilistic form of the robot's forward dynamics, which assumes that q_{k+1} is observable. In addition, z_k is the detection output through an object recognition algorithm and point cloud processing, encoding the 3D position of the observed targets.

The goal of filtering is to estimate the target state x at time k, namely, the posterior distribution, using the a priori estimate (prediction step) and the current measurement of the sensor (update step). Assuming that the prior probability $p(x_{k-1}|z_{1:k-1})$ is available at time $k-1$, the prediction step attempts to estimate $P(x_k|z_{1:k-1})$ from previous observations as follows:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1}, \quad (1)$$

where $p(x_k|x_{k-1})$ is the target's motion model based on a first order Markov process. Then, when the measurement z_k is available, given sensor configuration q_k at time k, the estimated state can be updated as

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k, q_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1}, q_k)}, \quad (2)$$

where $p(z_k|z_{1:k-1}, q_k) = \int p(z_k|x_k, q_k)p(x_k|z_{k-1})dx_k$. For the update step, the measurement z_k is used to modify the prior estimate, leading to obtaining the posterior distribution of the current state. Finally, after calculating the target's belief state $p(x_k|z_k)$, the goal of active sensing is to provide optimal sensor control inputs to track the target or to find it if its lost.

2.2 Context Modeling

In general, for object tracking, a transition model is assumed to be known based on a constant velocity model hypothesis with white noise, i.e. $P(x_t|x_{k-1}) = x_{k-1} + v_{k-1}\Delta t + \nu$ where ν represents white noise. However, this approach is inefficient in active object tracking since targets might be frequently missing, meaning that v_k is unavailable. Thus, inspired by how humans seem to track objects, it would be beneficial to understand the current situational context in order to effectively estimate possible locations of lost targets. To leverage this abstract knowledge or context, we employ a Dynamic Bayesian Network model to infer the possible state of targets. Consequently, when an agent begins looking for a missing target it will benefit by knowing the belief state of the context and trying to find it based on such information.

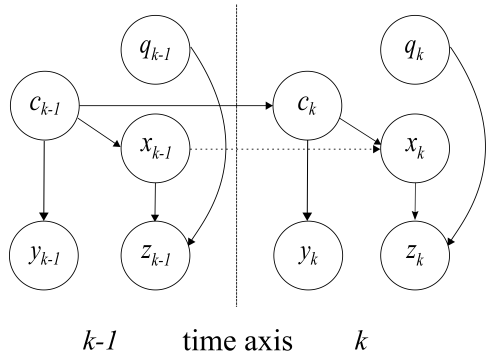


Fig. 2 Dynamic Bayesian Network with context states (c_k), target states (x_k), and robot configurations (q_k). y_k and z_k are measurements for context and targets, respectively. The arrows indicate the conditional dependencies between the random variables. The dotted line indicates that when an object is missing, it's target location cannot be directly estimated. In our case, it is inferred from context information.

2.2.1 Dynamic Bayesian Networks

DBN is an extension of Bayesian Networks, a graph model for representing causal relations and conditional dependencies, for temporal processes, which can evolve over time. The proposed model is shown in Fig. 2, where c_k denotes the context state at time step k, and y_k is the observation of the context state. Since target states can depend on context states, this model can be directly applied to Bayesian filtering as a transition model for prediction. In other words, instead of using $p(x_k|x_{k-1})$, we propose to use $p(x_k|c_{k-1})$ as the motion model, i.e.

$$p(x_k|z_{1:k-1}) \approx \int p(x_k|c_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1}, \quad (3)$$

In this sense, target states can be predicted using context states, which in turn can be estimated using context transition models and context measurements.

2.2.2 Hybrid Motion Model

More precisely, the target transition model can be reformulated using context states as

$$p(x_k|c_{k-1}) = \int p(x_k|c_k)p(c_k|c_{k-1})dx_k. \quad (4)$$

Here, we assume that $p(c_k|c_{k-1})$ is known (this will be explained in section 3.4.3). Furthermore, if the set of possible context states is finite, the posterior distribution of x_k can be calculated as

$$p(x_k|c_{k-1}) = \sum_{i=1}^{N} p(c_k^i)p(x_k|c_k^i), \quad (5)$$

where c_k^i denotes the i-th context state at time k and N is the cardinality of the set of context states. If each context

c_k is described using a Gaussian model, the posterior distribution becomes a Gaussian Mixture Model (GMM). As such, the probability density of the GMM is equivalent to the weighted sum of all components, i.e.

$$p(x_k|c_{k-1}) = \sum_{i=1}^{N} p(c_k^i)\mathcal{N}(x_k;\mu_i,\Sigma_i), \tag{6}$$

where $\mathcal{N}$ is the multivariate Gaussian model with mean vector μ and co-variance matrix Σ. Here, each μ_i and Σ_i represent the mean target location given the i-th context and its variance.

2.3 Particle Filter

We approximate the posterior distribution $p(x_k|z_{1:k})$ by a set of N number of particles $\{x_k, w_k\}^N$, where w_k is the corresponding weight of the particle x_k at time step k. Generally, the weighted approximation is calculated as

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i), \tag{7}$$

where particles x_k^i are drawn from a proposal density $q(x_{0:k}|z_{1:k})$ and the weight of the particles is updated using importance sampling. The normalized weight of the i-th particle can be defined as

$$w_k^i = \frac{p(x_{0:k}^i|z_{1:k})}{q(x_{0:k}^i|z_{1:k})}. \tag{8}$$

Assuming the Markov property, the proposal density can be decomposed into the prior proposal density and the propagated density as

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{k-1},z_{1:k})q(x_{0:k-1}|z_{1:k-1}). \tag{9}$$

Thus, this equality yields

$$w_k^i = \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1})p(x_{0:k-1}^i|z_{1:k-1})}{q(x_k^i|x_{k-1}^i,z_{1:k})q(x_{0:k-1}^i|z_{1:k-1})} \tag{10}$$

$$= \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1})}{q(x_k^i|x_{k-1}^i,z_{1:k})} w_{k-1}^i \tag{11}$$

Here, a common choice for $q(x_k^i|x_{0:k-1}^i,z_{1:k})$ is the motion model $p(x_k|x_{k-1})$, which minimizes the variance of $p(x|z)$. Also, as mentioned before, in the above equations we use Eq. (5) to estimate the process $p(x_k^i|x_{k-1})$. Based on Bayesian filtering, at every time step k, these weights can be recursively updated from the time step $k-1$,

$$w_k^i = w_{k-1}^i \frac{p(z_k|x_k^i,q_k)}{\sum_{j=1}^{N} p(z_k|x_k^j,q_k)}. \tag{12}$$

2.4 Sensor Model

The sensor model, $p(x|z,q)$, has the form of a joint distribution for z, x, and q. This model must consider the case of an empty measurement set (i.e. when the target is missing). In addition, the sensor model depends on whether the target is within the FOV or not. The FOV can be formulated as a function of q, $f(q)$, given the robot's configuration Freda et al. (2008). For an RGB-D camera, $f(q)$ has a cone shape with center $c(q)$, opening angle α, and radius R. These parameters are later determined based on real-world sensor specifications. In this study, the sensor model is defined as

$$\begin{aligned} p(z=\emptyset|x,q) &= 1-p_e && \text{if } x \not\subset f(q) \\ p(z=\emptyset|x,q) &= 1-p_d && \text{if } x \subset f(q) \\ p(z\neq\emptyset|x,q) &= p_e && \text{if } x \not\subset f(q) \\ p(z\neq\emptyset|x,q) &= p_d\mathcal{N}(z;x,\Sigma^2) && \text{if } x \subset f(q) \end{aligned} \tag{13}$$

where p_d is a user-defined true positive probability of the detection algorithm and p_e is the false negative probability (e stands for error).

2.5 Importance Sampling

Typically, sequential sensor observations are used to update weights of particles and compute the resulting probability distributions. However, for our active object tracking problem, continuous sensor readings are not always available due to frequently missing targets from occlusion or disappearance. We use another method to update the importance of particles in the FOV as a result of missing the target. The weight of particles in the field of view ($S_2(q)$ in Fig. 3) should be lowered. However, it is impossible to update the precise weights of the particles outside the FOV ($S_1(q)$) without additional information about the target when it is missing. Instead we use the value 1-p_e for updating the weights of the particles in $S_1(q)$.

However, due to importance sampling, the number of particles in $S_2(q)$ reduces significantly due to having negligible weights. This phenomenon is called the degeneracy problem. To avoid it, we use a re-sampling method in which new particles are generated in $S_1(q)$ corresponding to the ones lost in $S_2(q)$. In this case, the weights are set to $w_{k-1}^i = \frac{1}{N_s}$, and after updating, they are set to the sensor likelihood, $w_k^i \propto p(z_k|x_k^i,q_k)$.

3 Methods

3.1 Entropy to Reduce Uncertainty

The goal of active perception is to gather as much information as possible, or equivalently to reduce uncertainty.

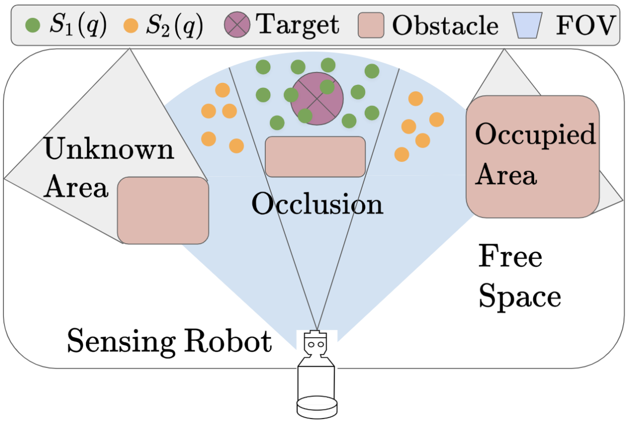


Fig. 3 An illustration of our problem, including the robot's FOV, a target object, and its environment. The target (purple circle) is occluded by the pink object so that it cannot be detected by the robot. We separate particles into two sets, $S_1(q)$ and $S_2(q)$, based on their visibility by the robot.

Shannon's entropy is defined as measuring the uncertainty of a random variable. For our problem, the desired control actions (i.e. moving the robot to a new configuration) correspond to policies that reduce uncertainty the most. Using the target's posterior distribution, $p(z|x)$, its entropy is represented as

$$H(p(x|z,q)) = \int_{x\in X} -p(x|z,q) log p(x|z,q)dx. \tag{14}$$

Using eq. (7), the entropy can be approximated by the particle weights as

$$H(p(x|z,q)) = -\sum_{i}^{N} w_i \log w_i \tag{15}$$

The full set of particle weights Ω can be divided into two sets ($S_1(q_k)$, $S_2(q_k)$) based on the FOV of the sensor $f(q_k)$. Here S_1 denotes the set of particles within the FOV, while S_2 is the set of particles that are not included in the FOV. Note that the region defined by $S_1(q)$ does not belong to $f(q_k)$ since it is occluded. Mathematically, $S_1 = \{w^i \in \Omega : x^i \notin f(q_k)\}$ and is disjointed from S_2, i.e. $S_2 = \Omega \setminus S_1$). As a result, the entropy equation can be expressed as

$$H(p(x_k|z_k,q_k)) \approx -\sum_{i\in S_1} w_i \log w_i - \sum_{j\in S_2} w_j \log w_j. \tag{16}$$

The expected information gain (IG) only depends on the first term of the above equation, and thus can be written as

$$\begin{aligned} IG(q_k) &= -E[H(x_{k-1}|z_{k-1},q_{k-1}) - H(p(x_{k-1}|z_{k-1},q_k))] \\ &\approx \sum_{i\in S_1(q_k)} w_i \log w_i. \end{aligned} \tag{17}$$

3.2 Utility Function For Target Detection

The first term of our utility function is the information gain, which involves the amount of information that can be obtained given sampled configurations. The second term penalizes traveling cost. Lastly, we define a perception gain that assists the robot to approach targets to increase the quality of the sensing process. Mathematically, the cost function is described as

$$J(x_k, q_{k-1}, q_k) = IG - \zeta_{travel} - \zeta_{perception} \tag{18}$$

$$\zeta_{travel} = \beta(q_k - q_{k-1})^T Q(q_k - q_{k-1}) \tag{19}$$

$$\zeta_{perception} = \gamma(x_k - S_d\, q_k)^T (x_k - S_d\, q_k). \tag{20}$$

Here x_k is the estimated position of the target at time k, q_{k-1} is the current robot configuration, q_k are sampled robot configurations, β, γ are cost weighting factors, and S_d is a mapping that extracts the Cartesian position of the robot from its position-rotation configuration. Solving this cost yields,

$$q_k^* = \arg\max J(x_k, q_{k-1}, q_k) \tag{21}$$

To solve this problem we use a sampling based approach and greedy search to obtain the final result, q_k^*.

3.3 POMDP High-level Planner

For our decision framework, we will estimate the current context and based on it choose actions to increase rewards. Since context cannot be directly measured, our planner is formulated as a Partially-Observable Markov Decision Process (POMDP). A POMDP can be described by the tuple (S,I,A,Z,T,R), a finite set of states $S = \{s_1, \cdots s_{|S|}\}$, an initial probability distribution over these states I, a finite set of actions $A = \{a_1, \cdots a_{|A|}\}$, a finite set of observations $Z = \{o_1, \cdots o_{|Z|}\}$, and a transition function $T(s,a,s') = P(s'|s,a)$ that maps $S \times A$ into discrete probability distributions over S. In detail, a transition model $T(s,a,s')$ specifies the conditional probability distribution of shifting from state s to s' by applying action a. $Z(s',a,o) = P(o|s,a)$ is the observation mapping that computes the probability of observing o in state s' when executing action a. $R = r(s',a,s)$ is the reward function.

A POMDP is equivalent to a continuous-state Markov Decision Process where the states are believes, also called a belief MDP Krishnamurthy (2016). Thus a state can be rewritten using belief states $b(s) = p(s)$, defined as the posterior distribution over all possible states given the history of actions and observations. For our purposes, the states will be the context situations. In the vein of Eqs. (1) and (2), using

Bayesian formulation belief states can be represented recursively as follows:

$$b_k(s|a_{k-1}) = \sum_{s' \in S} T(s, a_{k-1}, s') b_{k-1}(s) \tag{22}$$

$$b_k(s|a_{k-1}, o_k) = \eta Z(s, a_{k-1}, o_k) b_k(s|a_{k-1}), \tag{23}$$

where η is a normalizing constant. The goal of the POMDP is to select a sequence of actions over time to maximize the expected cumulative reward. Value iteration algorithms are used for optimally solving POMDPs. This strategy can be defined as a policy π^*, which maps a belief b to actions. Given policy π, a belief $b(s) \in B$, the value function can be computed via the Bellman equation as

$$V(b, \pi) = \rho(b, \pi(b)) + \gamma \sum_{b' \in B} \tau(b, \pi, b') V(b', \pi), \tag{24}$$

where γ is a discount factor, ρ is the expected reward, and τ is the transition probability to b' from b under π, which can be computed as Ross et al. (2008)

$$\tau(b, \pi, b') = \sum_{o \in Z} p(b'|b, \pi, o)\ p(o|b, \pi). \tag{25}$$

The best policy π^* is obtained by solving the optimization problem:

$$\pi^*(b) = \arg\max V(b_k, \pi_k). \tag{26}$$

In this paper, we utilize an on-line POMDP solver, the Determinized Sparse Partially Observable Tree (DESPOT) Somani et al. (2013), to obtain the optimal action policy based on the current belief of context states.

3.4 POMDP Problem Formulation

3.4.1 States

We define context states as one of the following semantically grounded symbols $\{$*Visible, Occluded, Disappearance, Irrecoverable*$\}$, some of which cannot be measured directly by sensors. *Visible* is the state in which the target can be directly measured by the robot's sensors. *Occluded* is the state when the target has been occluded by another object but is believed to be behind it. *Disappearance* is the state in which the target is not directly visible by the robot and is not believed to be occluded by another object in the robot's field of view. In this state, the target is believed to have been moved away by a person. Finally, *Irrecoverable* corresponds to the case when the robot does not have information about the whereabouts of the target nor it can use context information to find it.

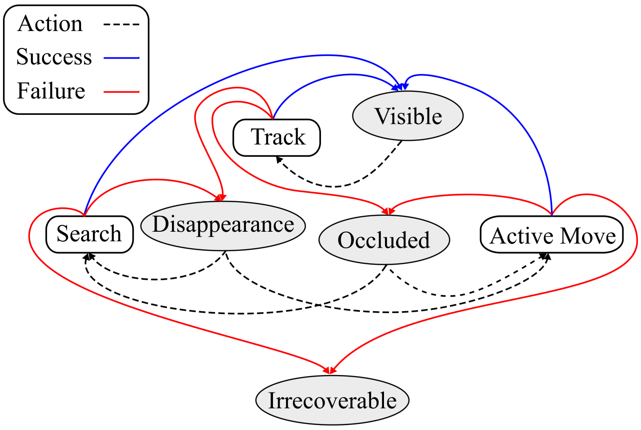


Fig. 4 POMDP problem formulation for object tracking. The actions are Search, Track, and Active Move. The states are shaded in gray. When actions don't result on finding the target, i.e. Visible state, we consider that action to Fail. Otherwise it's a Success.

3.4.2 Actions

Our action set is defined as $A = \{$ *Track, Search, Active Move*$\}$. The *Track* action commands the robot to track targets by panning its head. *Active Move* prompts the robot to change its location to better track a visible target or to overcome Occlusion situations where the taregs is believed to be behind another object. Lastly *Search* prompts the robot to explore its environment in search of humans by turning its head. This action corresponds to the hypothesis that when an object is missing it might be around a human therefore finding humans might reveal the target. In addition, we search for humans instead of directly searching for the target, since people are larger and easier to spot from afar.

3.4.3 Transitions

We define a transition model to estimate the next context state based on the available actions, i.e. $p(c_k|c_{k-1})$ as defined in Equation (23). We remark that context updates are performed at asynchronous time frames since actions take more than one time step to complete. Accordingly, we define transitions as:

$$p(c_l|c_{l-1}) = \begin{cases} I & \text{if } a \text{ is active} \\ b_{l-1}(s|a_{l-1}) & \text{if } a \text{ is complete,} \end{cases} \tag{27}$$

where l is a distinct time frame from k since transitions among context states are determined based on the execution time of high-level actions. Transitions between context states are described in Fig. 4.

3.4.4 Observations

To estimate hidden context states, we rely on four features. The most useful feature is whether a target observation exists or not. If z_k is non-empty, the context state might be *visible* with high probability. Thus, the first feature variable is defined as $\theta_{Target} = 1$ if z_k exists, otherwise, $\theta_{Target} = 0$.

The second feature we consider reflects the probability of occlusions to occur, and can be computed by using existing object recognition algorithms. Once an occlusion occurs, the object being tracked will be occluded by another object (both of them represented by bounding boxes traced by the object recognition software). In the case that objects causing occlusions are semantically identifiable, an overlap ratio is defined based on the two bounding boxes (the object being tracked, i, and the newly occluding object, j) as

$$\mathrm{OR}_k^{ij} = \frac{BB_k^j \cap BB_{k-1}^i}{BB_{k-1}^i}, \tag{28}$$

where BB_k^i denotes the bounding box for object i at time index k. We define a feature $\theta_{OR}^i \in \{0,1\}$ as the rule $(OR^{ij} > \lambda_{OR}) \cap (z = \emptyset)$ where λ_{OR} is a predefined threshold and $z = \emptyset$ denotes the absence of a target. In the case that the occluding object is unidentifiable, the depth variance can also be used to detect the occlusion. Note that the depth within the bounding box will become smaller when a new object starts occluding the previous one. Therefore, features affected by depth variation can be formulated as

$$\Delta Z_k^i = \bar{Z}_{k-\Delta T:k}^i - Z_k^j, \tag{29}$$

where $\bar{Z}_{k-\Delta T:k}^i = \frac{1}{\Delta T} \sum_{k-\Delta T}^{k} Z^i$ denotes the average depth for time period ΔT within BB^i, and Z_k^j stands for the newly detected depth value (i.e. the occluding object) for the same region. Thus, we define a feature variable for depth information as follows

$$\theta_{Depth}^i = \begin{cases} 1 & \left(\Delta Z_t^i > \lambda_{depth} \cap z = \emptyset\right) \\ 0 & \text{otherwise.} \end{cases} \tag{30}$$

Lastly, the loss of a target can also be inferred based on the premise that the main cause of its disappearance is a human taken it away. This assumption seemingly makes sense since most common objects cannot change their location by themselves. Accordingly, we define a new feature associate with the presence of a human as, $\theta_{Human} = 1$ if a human is present. In practice, we detect humans by using the object recognition algorithm Redmon et al. (2016).

Consequently, the observation model is expressed using the feature vector

$$\Theta(a) := (\theta_{Target}(a), \theta_{OR}(a), \theta_{Depth}(a), \theta_{Human}(a)). \tag{31}$$

By detecting features, we estimate the context state using the likelihood distribution $p(o|s,a) \approx p(\Theta|s) = \Pi_{i=1}^4 p(\theta_i|s)$ which relies on the assumption that each feature is conditionally independent.

3.4.5 Rewards

We assign positive rewards only when the context state is *visible*. Otherwise we don't give any reward. As such, we compute the expected reward $\rho(\cdot)$ using belief the belief state $b(s)$ and the reward function $r(s,a)$ as

$$\rho(b(s),a) = \sum_s b(s) r(s,a) \tag{32}$$

$$r(s,a) = \left\{ \begin{array}{ll} 10 & \text{if } s \text{ } is \text{ Visible} \\ 0 & \text{Oterwise} \end{array} \right\}. \tag{33}$$

3.5 Context Models

As indicated in Equation (6), each context state has its own prediction model regarding the target's location given prior knowledge.

3.5.1 Visible Context

In this case, we predict the target's position from the last target state x_{k-1} using a Gaussian distribution, i.e.

$$p(x_k|c = visible) \approx \mathcal{N}(x_k; x_{k-1} + v_{k-1}\Delta t, \sigma_x^2) \tag{34}$$

where σ_x^2 is the sensor noise.

3.5.2 Occluded Context

In this case, we estimate the target's position from the fact that is behind the occluding object's position, x_{occ}, i.e.

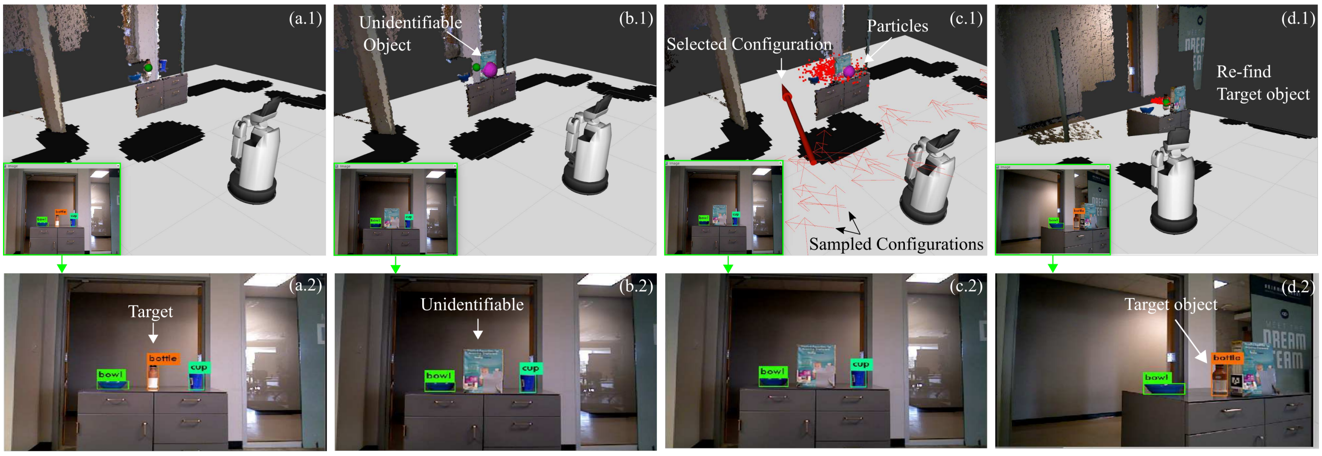
$$p(x_k|c = Occluded) \approx \mathcal{N}(x_k; x_{occ} + \delta_{\text{offset}}, \sigma_{occ}^2), \tag{35}$$

where δ_{offset} is an approximate distance value such as the length of the bounding box describing the occluding object.

3.5.3 Disappearance Context

In this state, the position of the target is inferred from knowledge about nearby people. If no one is visible in the FOV, the robot will begin to look for nearby people by rotating its base and head using the *search* action. Once a person has been detected, a prediction model will generate particles based on the following model,

$$p(x_k|c = Disappearance) \approx \mathcal{N}(x_k; x_{human}, \sigma_{human}^2). \tag{36}$$



(b) (c) (d)

Fig. 5 Example resolving an occlusion event and leading to re-locating the target. (a) The robot is tracking a target object (a bottle) using its cameras. (b) an occlusion suddenly occurs (e.g. a person has placed a box in front of the bottle); an occluding marker (a purple ball in the figure) is placed in the computer visualization window in front of the target. (c) Using our proposed active sensing method, the robot chooses its next configuration (shown as a big red arrow) in order to maximize its information gain, which is approximated by particles. (d) For this example, the robot finally succeeds to re-locate the target behind the occluding object.

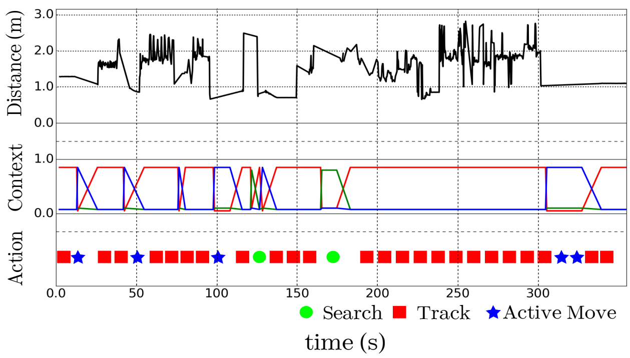


Fig. 6 Time analysis for an example scenario of active object tracking. The first row shows the estimated distance between the robot and the target object. The second row shows the estimated context state (red line: *visible*, blue line: *Occluded*, and green line: *Disappearance*). The third row provides the chosen actions at each time frame.

4 Experimental Results

4.1 Robot Description

To validate our approach experimentally, we use the Toyota Human Support Robot (HSR), a mobile manipulator equipped with an omnidirectional mobile base and a panning and tilting head. A depth camera (Xtion, Asus) is located on top of the robot's head to obtain an RGB-D stream. A laser range scanner, Hokuyo, is installed at the front bottom of the base in order to detect static and dynamic obstacles. The HSR uses two different computers, one Intel Core i7, 4th Gen with 16GB RAM is used for basic navigation functions of the robot and another one, an Alienware Intel Core i7-7820HK, GTX 1080 laptop, is used for running the object and human detection algorithm, named YOLO Redmon et al. (2016). Interprocess communications are handled with ROS. The testing facility is the UT Austin's Human Centered Robotics Lab.

4.2 Scenarios

We deal with three possible situations: 1) occlusions occur due to the interference of another object, 2) objects disappear when they are taken away by people, and 3) objects temporarily move outside of the FOV but can be quickly found by employing active tracking. Fig. 5 shows that the robot is able to track and resolve occlusion situations through searching for new configurations and successfully re-locating the target. In addition, the examples demonstrating the second and third cases above are shown in Fig. 7. Details can be found in Fig. 6. This figure provides the evolution of the estimated context states and the corresponding high-level actions taken over time based on context states.

4.3 Performance Evaluation

We propose four criteria to evaluate performance: Success Ratio (SR), Tracking Ratio (TR), Average Restoring Time (ART), and Failure Time (FT). Each criterion is evaluated over 20 trials until the robot fails to track the target. The statistical results are shown in Table 2.

4.3.1 Success Ratio (SR)

Success ratio given an action represents the number of successful target re-locations versus the total number of trials. In our experiments, we achieved an overall success rate of

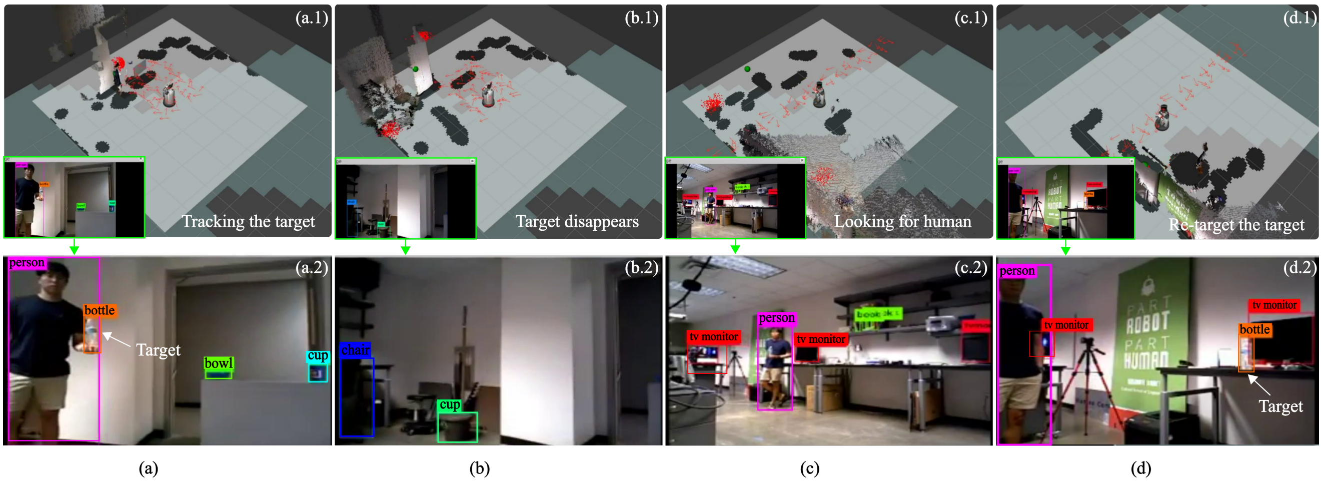


Fig. 7 Demo showcasing finding a target when it suddenly disappears from the robot's FOV and is not believed to be occluded by another object. (a) Initially, the robot tracks the target (a bottle) using its depth cameras. (b) The target suddenly disappears from its FOV; here it is believed the target has disappeared because of the information returned from the observation model described earlier. (c) The robot moves to find a person around its neighborhood. If a person is detected, the robot navigates to that person and attempts to locate the target nearby. (d) In our demo, the robot succeeds in re-locating the object since it was placed next to the person.

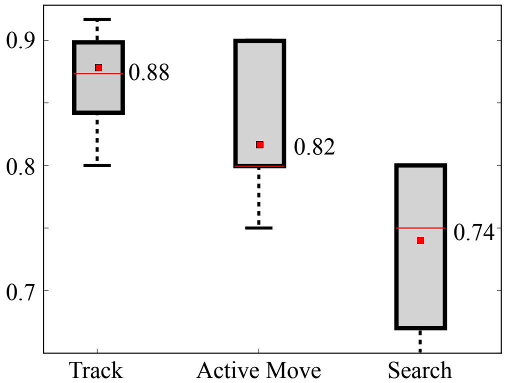


Fig. 8 Success ratio for various active sensing scenarios. The rates represent the ratio of success to track or find a target after action execution depending on believed context states. *Track* behaviors have the highest success rates, 0.88, followed by *Active Move*, while *search* has the highest variance and the lowest success rates, 0.74.

0.82 with standard deviation equal to 0.097 despite the conditions being highly dynamic (i.e. the target is moving or occluded or has suddenly disappear). The success ratio case is shown for each context and action being taken in Fig. 8.

4.3.2 Tracking Ratio (TR)

This ratio can be regarded as keeping the target in the FOV. It is calculated using the amount of time the target is believed to be in the visible state versus the total time an experiment lasts. We achieved an average TR value of 0.7 for the demos above, i.e. a mixture of experiments where we repeatedly occluded the target, or a person moved it away to nearby locations outside of the robot's FOV, or a person rapidly moved the target around the robot.

Table 2 Four Criteria results

Criterion	Mean	Standard Deviation
SR	0.82	0.097
TR	0.71	0.096
ART (s)	10.22	7.9
FT (s)	232	44.2

4.3.3 Average Restoring Time (ART)

This time is calculated as the differential of time between loosing sight of a target until it is re-located. It varies depending on the type of action. For example, the average restoring time for *Track* was 2 seconds while that for *Active-Move* was 12.15 seconds with a standard deviation of 3.95. Lastly, the ART for *Search* was 16.5 seconds with standard deviation of 7.8.

4.3.4 Failure Time (FT)

Lastly, we measure the amount of time it takes the robot to fail tracking or searching targets. If the target is not found within 1 minute, the robot will go into failure mode, i.e. the context state is irrecoverable. The experiment for this measurement is done based on an arbitrary mixture of context states performed by moving the target around or occluding it. The overall FT for our action sequence is 232 seconds with standard deviation of 44.2 seconds. This indicates that the proposed algorithm performs successful target tracking and searching tasks for approximately 4 minutes before becoming irrecoverable due to arbitrary user manipulations.

5 Concluding Remarks

This paper addresses active target tracking and searching capabilities using mobile robots. Experimental results are performed using multiple scene trials. To this point, we employ information-theoretic costs for active search. Integrating a DBN model, particle filtering and POMDP planning, we are able not only to infer target locations from context information in a probabilistic manner, but also to define cost functions effectively. Obtaining the desirable control inputs for gathering information allows our robot to have better tracking and search capabilities under several dynamic conditions, such as occlusions and the sudden disappearance of a target.

We highlight the following achievements. First, our methods are scalable and versatile via the proposed hybrid state estimator, i.e. the particle filter plus POMDP based on context states. To this point, the action sequence shown in Fig. 6 effectively demonstrates the robot's capability to actively find occluded or missing targets under various dynamic conditions. Second, as shown in Fig. 8, our active sensing algorithm, on average, re-locates targets with high success rates. Finally, we have verified robustness and efficiency of our methods using the proposed criteria, SR, TR, ART, and FT. In essence, we verified that the robot can perform practical tracking and search tasks while operating in dynamic environments.

There are limitations to our approach. Better probabilistic models are required to model realistic target search situations. For example, our current method assumes that direct transitions between occlusion and disappearance states do not happen. Furthermore, to deal with more realistic situations, a memory-based approach that uses historical data should be used to predict context. Recurrent Neural Networks might be a good solution for future work. Lastly, there exists a room for extending our proposed work to multi object tracking cases.

Overall, this work demonstrates a prototype of robust autonomous active target tracking performed using a mobile robot in a practical setup.

References

Aydemir A, Pronobis A, Göbelbecker M, Jensfelt P (2013) Active visual object search in unknown environments using uncertain semantics. IEEE Transactions on Robotics 29(4):986–1002

Bai H, Hsu D, Lee WS (2014) Integrated perception and planning in the continuous space: A pomdp approach. The International Journal of Robotics Research 33(9):1288–1302

Bajcsy R (1988) Active perception. Proceedings of the IEEE 76(8):966–1005

Bertuccelli LF, How JP (2006) Search for dynamic targets with uncertain probability maps. In: 2006 American Control Conference, IEEE, pp 6–pp

Bourgault F, Furukawa T, Durrant-Whyte HF (2003) Optimal search for a lost target in a bayesian world. In: Field and service robotics, Springer, pp 209–222

Charrow B, Liu S, Kumar V, Michael N (2015) Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 4791–4798

Chung TH, Burdick JW (2012) Analysis of search decision making using probabilistic search strategies. IEEE Transactions on Robotics 28(1):132–144

Denzler J, Brown CM (2002) Information theoretic sensor data selection for active object recognition and state estimation. IEEE Transactions on pattern analysis and machine intelligence 24(2):145–157

Eidenberger R, Grundmann T, Zoellner R (2009a) Probabilistic action planning for active scene modeling in continuous high-dimensional domains. In: 2009 IEEE International Conference on Robotics and Automation, IEEE, pp 2412–2417

Eidenberger R, Zoellner R, Scharinger J (2009b) An integrated active perception module for a distributed cognitive architecture. In: 2009 International Conference on Advanced Robotics, IEEE, pp 1–7

Freda L, Oriolo G, Vecchioli F (2008) Sensor-based exploration for general robotic systems. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp 2157–2164

Hausman K, Kahn G, Patil S, Müller J, Goldberg K, Abbeel P, Sukhatme GS (2016) Occlusion-aware multi-robot 3d tracking. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 1863–1870

Julian BJ, Karaman S, Rus D (2014) On mutual information-based control of range sensing robots for mapping applications. The International Journal of Robotics Research 33(10):1375–1392

Kaelbling LP, Lozano-Pérez T (2012) Unifying perception, estimation and action for mobile manipulation via belief space planning. In: 2012 IEEE International Conference on Robotics and Automation, IEEE, pp 2952–2959

Kaelbling LP, Lozano-Pérez T (2013) Integrated task and motion planning in belief space. The International Journal of Robotics Research 32(9-10):1194–1227

Krishnamurthy V (2016) Partially observed Markov decision processes. Cambridge University Press

Lau H, Huang S, Dissanayake G (2006) Probabilistic search for a moving target in an indoor environment. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp 3393–3398

Li JK, Hsu D, Lee WS (2016) Act to see and see to act: Pomdp planning for objects search in clutter. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 5701–5707

Makris A, Kosmopoulos D, Perantonis S, Theodoridis S (2011) A hierarchical feature fusion framework for adaptive visual tracking. Image and Vision Computing 29(9):594–606

Porta JM, Vlassis N, Spaan MT, Poupart P (2006) Point-based value iteration for continuous pomdps. Journal of Machine Learning Research 7(Nov):2329–2367

Radmard S, Croft EA (2017) Active target search for high dimensional robotic systems. Autonomous Robots 41(1):163–180

Radmard S, Meger D, Little JJ, Croft EA (2018) Resolving occlusion in active visual target search of high-dimensional robotic systems. IEEE Transactions on Robotics 34(3):616–629

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788

Ross S, Pineau J, Paquet S, Chaib-Draa B (2008) Online planning algorithms for pomdps. Journal of Artificial Intelligence Research 32:663–704

Ryan A, Hedrick JK (2010) Particle filter based information-theoretic active sensing. Robotics and Autonomous Systems 58(5):574–584

Somani A, Ye N, Hsu D, Lee WS (2013) Despot: Online pomdp planning with regularization. In: Advances in neural information processing systems, pp 1772–1780

Sridharan M, Wyatt J, Dearden R (2010) Planning to see: A hierarchical approach to planning visual actions on a robot using pomdps. Artificial Intelligence 174(11):704–725

Tseng KS, Mettler B (2017) Near-optimal probabilistic search via submodularity and sparse regression. Autonomous Robots 41(1):205–229

Valencia R, Andrade-Cetto J (2018) Active pose slam. In: Mapping, Planning and Exploration with Pose SLAM, Springer, pp 89–108

Van Den Berg J, Patil S, Alterovitz R (2012) Motion planning under uncertainty using iterative local optimization in belief space. The International Journal of Robotics Research 31(11):1263–1278

Wu K, Ranasinghe R, Dissanayake G (2015) Active recognition and pose estimation of household objects in clutter. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 4230–4237

Xiang Y, Alahi A, Savarese S (2015) Learning to track: Online multi-object tracking by decision making. In: Proceedings of the IEEE international conference on computer vision, pp 4705–4713

Yun S, Choi J, Yoo Y, Yun K, Young Choi J (2017) Action-decision networks for visual tracking with deep reinforcement learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2711–2720