2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Modelling Search With a Binary Sensor Utilizing Self-Conjugacy of the Exponential Family

Devin Bonnie, Salvatore Candido, Timothy Bretl, and Seth Hutchinson

Abstract— **In this paper, we consider the problem of an autonomous robot searching for a target object whose position is characterized by a prior probability distribution over the workspace (the object prior). We consider the case of a continuous search domain, and a robot equipped with a single binary sensor whose ability to recognize the target object varies probabilistically as a function of the distance from the robot to the target (the sensor model). We show that when the object prior and sensor model are taken from the exponential family of distributions, the searcher's posterior probability map for the object location belongs to a finitely parameterizable class of functions, admitting an exact representation of the searcher's evolving belief. Unfortunately, the cost of the representation grows exponentially with the number of stages in the search. For this reason, we develop an approximation scheme that exploits regularized particle filtering methods. We present simulation studies for several scenarios to demonstrate the effectiveness of our approach using a simple, greedy search strategy.**

I. INTRODUCTION

In this paper, we consider the problem of a robot trying to localize an object in a continuous domain, using only binary measurements (i.e., the object is either detected, or not, at each stage of the search). This is an example of the probabilistic search problem. We restrict our attention to the case in which both the prior probability map for the object location and the sensor model belong to the exponential family of distributions. This assumption does not impose severe limitations, since the exponential family contains many typically used pdfs, e.g., the exponential and Gaussian pdfs, and can be used to approximate many other distributions. Due to the self-conjugacy properties of this family, we can demonstrate that the searcher's posterior probability density function for the object location will belong to a finitely parameterizable class of functions. For short-term searches, this feature admits the possibility of maintaining an exact representation of the searcher's belief regarding the object location. For long-term searches, it enables a computationally effective solution based on regularized particle filters.

Search problems have been studied extensively in the literature [1]–[3], and several papers have addressed Bayesian formulations and various strategies of search [4]–[6]. Many of these approaches deal with discrete domains or estimation methods, often using sampling methods or model reduction, to represent the posterior of a target's position. Such approaches lead to a discrete optimization over a discretized approximation of the search domain. In contrast, our method preserves a continuous representation, while providing a suitable structure for optimization.

A. Contributions

This paper is primarily concerned with identification of a probabilistic search model that has an exact, finitely parameterizable representation, and which readily yields a continuous approximation of the object location pdf using Bayesian inference. Thus, our primary contributions are:

1) We demonstrate that, using the self-conjugacy properties of the exponential family for a specific class of sensor models and initial conditions, we can represent the belief exactly using a finite representation.
2) We derive the filtering (or inference) equations that model the posterior likelihood of object location conditioned on robot motions and a sequence of noisy binary measurements.
3) We develop a practical and effective approximation approach based on a regularized particle filter.
4) We illustrate the feasibility of our model using a simple gradient ascent strategy, which exploits our preservation of gradient information.
5) We demonstrate our method for both single-agent and multiple-agent search for a stationary target.

These filtering equations can be used as a basis for object localization or other variants of search for these objectives. We discuss possible target applications in Section IV.

B. Related Research

Probabilistic search is well-studied in the literature. Much of the early work on search is rooted in operations research [1], [7], and later techniques are based on this formalization. Various search problems, surveyed in [2], [3], discuss planning techniques for the one-sided search problem where the target does not respond to the actions of the searcher, multiple targets and sensor models, and target evasion games. Similarly, the work in search-related problems spans a diverse list of topics such as exploration and mapping [8] and coverage path planning [9]. Their unifying factor is employing observations to improve planning trajectories in an online fashion while integrating sensor data with the problem-specific model. Such planning can be accomplished by utilizing a probabilistic map to generate the optimal path for maximizing object detection [10], acting as an extension to classical motion planning methods.

Probabilistic approaches to search and coverage are discussed in [4]–[6], [11]. A Bayesian approach to search for

This material is based in part upon work supported by the National Science Foundation under award CNS 0931871.

D. Bonnie, S. Candido, and S. Hutchinson are in the Department of Electrical and Computer Engineering and T. Bretl is with the Department of Aerospace Engineering at the University of Illinois. {bonnie1,candido,seth,tbretl}@illinois.edu

an object that is either stationary or mobile (but not evading) is conducted in [5]. The approach yields target beliefs in the form of probability density functions; however, unlike the previous work in [4] their practice numerically discretizes these beliefs over the seach environment. Similar to [10], the focus is placed on generating a search trajectory to maximize the detection of the target.

In [6], [11], [12] search is modelled as a hypothesis test given a search area. Similar to our work, Bayesian inference equations are developed (though in a discrete environment) for the case of a binary sensor, where false positives and negatives are included, to determine the probability of target detection. The discrete decomposition of the environment allows the target area to be represented as a graph, which in turn yields a representation of agent motion constraints. Multiple search strategies are considered in [11] and are compared by the expected time to reach a hypothesis. The work in [6] furthers the hypothesis testing by analyzing the necessary thresholds to determine a hypothesis and provides closed form target belief update expressions for specific cases.

C. Outline

In the next section, we develop the filtering (Bayesian inference) equations based upon our problem model. These equations demonstrate that we represent the exact condition of our search problem with a finite representation and result in straightforward evolution update rules. We then derive a search strategy using the well-known greedy method, namely gradient ascent over the posterior. For searches with many stages, we obtain a continuous approximation using a regularized sampling importance resampling particle filter. We then present both single and multi-agent searches employing this strategy, as well as our choice of approximation. Finally, we summarize our work and discuss future directions.

II. Search Model

We consider the problem of modelling search for a stationary target located within a given workspace, $\mathcal{W}$. The prior information about the object is in the form of a finitely parameterized probability density function over $\mathcal{W}$, where this prior belongs to the class of the exponential family of distributions. Specifically the location of the object, denoted $\boldsymbol{x}^o \in \mathcal{W}$, is a random vector distributed according to $\mathrm{f}_{\boldsymbol{x}^o}$. Furthermore, the binary sensor model considered is also a member of the exponential family class. Given these conditions, we desire to represent the object distribution in a finitely parameterizable representation through Bayesian filtering. The discussion in this paper is limited to workspace $\mathcal{W}$ equal to $\mathbb{R}^1$ or $\mathbb{R}^2$, but the approach generalizes to $\mathbb{R}^N$.

In this section we first give the specifics of our problem model. Using this model, we provide appropriate Bayesian inference equations. By applying these equations to the problem model we explicitly show, through our initial conditions, that this model leads to self-conjugacy under the operation of the Bayesian filter. This provides a finitely parameterizable representation of the object map, which is given specifically in the form of update rules.

A. Problem Model

The search agent considered is a mobile robot with state $x^r \in \mathcal{W}$ and the following discretized, deterministic[1] kinematic motion model

$$x^r_{t+1} = x^r_t + u_t, \qquad ||u_t|| \leq \bar{u} \tag{1}$$

trying to find an object with unknown (random) location, where the quantity u_t is a control input to the system. The robot is equipped with a noisy binary sensor where the observation at time t is denoted $\boldsymbol{y}_t$, with the bold symbol indicating a random object. The observation $\boldsymbol{y}_t = 1$, which corresponds to object detection, becomes more probable as the robot approaches the object. We assume that the probability of object detection can be expressed by a sensor model of the form

$$\mathrm{p}_{\boldsymbol{y}_t|x^r_t,\boldsymbol{x}^o}\left(\boldsymbol{y}_t = 1|x^r_t, x^o\right) = e^{\langle\theta_s, T(x^o)\rangle - F(\theta_s)} \tag{2}$$

where the sensor parameter θ_s is a function of x^r_t. Specifically, the parameters in the exponential function represent a probability distribution in the canonical form of the exponential family class. That is, θ_s is a vector containing the *natural parameters*, $T(x)$ is the *sufficient statistic*, and $F(\theta)$ is the *log normalizer* [13], [14].

Here we note that the current framework can be easily extended to employ multiple search agents. This extension stems from the intuition that multiple agents share a common belief of the object. As an agent independently makes an observation at some time t, each observation is separately used to generate a new, shared posterior. We demonstrate the use of multiple agents in Section III-C.

B. Bayesian Inference

In this section, we derive the sequential inference equations using standard Bayesian development. Let the information state be

$$I_t = \{x^r_0, \mathrm{f}_{\boldsymbol{x}^o}, u_1, \cdots, u_{t-1}, y_1, \cdots, y_t\}$$

where x^r_0 is the initial robot position and f_{x^o} is the initial belief of object location. As the robot's motion is deterministic (Equation 1) and its initial position is known, the distribution we wish to model solely represents the state of the object. Thus, for convenience we will define what we will refer to as the object map

$$m_t(x^o) := \mathrm{f}_{\boldsymbol{x}^o|\boldsymbol{I}_t}\left(x^o|I_t\right)$$

which is the posterior pdf of the object location conditioned on the information vector.

[1] For this problem, it is reasonable to assume a deterministic motion model, since uncertainty in robot motion will be insignificant compared to the uncertainty in the observation process. When this is not the case, POMDP models are more appropriate, but this adds significant complexity to the problem, and in many cases the added complexity does not bring significant performance gains.

Before deriving the generic update law for the object map, we note that the information vector can be equivalently represented with the robot position rather than a set of controls. Thus, let

$$\tilde{I}_t = \{\mathrm{f}_{\boldsymbol{x}^o}, x_0^r, x_1^r, \cdots, x_t^r, y_1, \cdots, y_t\}$$

be a modified information vector, which is completely equivalent to the canonical information vector.

We construct the Bayesian update equation for the object map utilizing the usual Bayesian development.

$$\begin{aligned} m_t(x^o) &:= \mathrm{f}_{\boldsymbol{x}^o|\boldsymbol{I}_t}\left(x^o|I_t\right) \\ &= \mathrm{f}_{\boldsymbol{x}^o|\tilde{\boldsymbol{I}}_t}\left(x^o|\tilde{I}_t\right) \\ &= \mathrm{f}_{\boldsymbol{x}^o|\boldsymbol{y}_t,x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(x^o|y_t,x_t^r,\tilde{I}_{t-1}\right) \end{aligned}$$

These steps are performed solely by manipulating equivalent representations of identical sets of events. Applying Bayes rule, we find

$$m_t(x^o) = \frac{\mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x}^o,x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(y_t|x^o,x_t^r,\tilde{I}_{t-1}\right)\mathrm{f}_{\boldsymbol{x}^o|x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(x^o|x_t^r,\tilde{I}_{t-1}\right)}{\mathrm{p}_{\boldsymbol{y}_t|x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(y_t|x_t^r,\tilde{I}_{t-1}\right)}$$

From (2) we know that $\boldsymbol{y}_t$ is conditionally independent of $\tilde{I}_{t-1}$ when conditioned on $\boldsymbol{x}^o$ and x_t^r. Furthermore, we know that the robot's current position (before taking a measurement at that position) does not affect our belief of object location. Thus, we can simplify our update rule to

$$m_t(x^o) = \frac{\mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x}^o,x_t^r}\left(y_t|x^o,x_t^r\right)\mathrm{f}_{\boldsymbol{x}^o|\tilde{\boldsymbol{I}}_{t-1}}\left(x^o|\tilde{I}_{t-1}\right)}{\mathrm{p}_{\boldsymbol{y}_t|x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(y_t|x_t^r,\tilde{I}_{t-1}\right)}$$

The inverse of the denominator in this equation is the standard normalization constant η_t, which can be computed using the total law of probability (conditioning on x^o). Also note that the second term of the numerator is exactly our definition of the object map, so by definition

$$m_t(x^o) = \eta_t \; \mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x}^o,x_t^r}\left(y_t|x^o,x_t^r\right) \; m_{t-1}(x^o) \tag{3}$$

with

$$\eta_t := \left[\int \mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x}^o,x_t^r}\left(y_t|x^o,x_t^r\right) \; m_{t-1}(x^o)dx^o\right]^{-1}$$

Thus, we have generated a generic update law for this model of search. In the next section, we show that weighted sums of distributions from an exponential family are self-conjugate under this update law. Furthermore, we explicitly provide the update rules for each sensor observation case.

C. Update Law for an Exponential Family

In this section we give an outline of our derivation, fully presented in Appendix I, using the generic Bayesian update law that allows an exact, finite parameterization of the object map belonging to the exponential family class of distributions. Furthermore, our derivation yields straightforward update rules to model the evolution of the object map. This is done inductively by using the well known self-conjugacy property [15] of the exponential family with an atypical mixture model [16], i.e., each mixture component's weight is not constrained to the set of positive reals. We define the form of the atypical mixture as

$$m_{t-1}(x^o) := \sum_{i=1}^{n} v_i \exp\left[\langle\theta_i, T(x^o)\rangle - F(\theta_i)\right] \tag{4}$$

for the stage $t-1$, where v_i is the i$^{\text{th}}$ component's non-standard weight.

Substituting the sensor model (Equation 2) and the non-standard mixture model (Equation 4) into the generic Bayesian update law (Equation 3) yields

$$m_t(x^o) = \eta_t \exp\left[\langle\theta_s, T(x^o)\rangle - F(\theta_s)\right] \; m_{t-1}(x^o)$$

at time t. Considering the case of a positive sensor observation, i.e. $y_t = 1$, results in the update law

$$m_t(x^o) = \sum_{i=1}^{n} v_i^+ \exp\left[\langle\theta_i^+, T(x^o)\rangle - F(\theta_i^+)\right] \tag{5}$$

where the "+" superscript denotes positive update parameters. Likewise, the update law for a negative sensor observation ($y_t = 0$) yields

$$\begin{aligned} m_t(x^o) = &\sum_{i=1}^{n} v_i^+ e^{\langle\theta_i^+, T(x^o)\rangle - F(\theta_i^+)} \\ &+ \sum_{i=1}^{n} v_i^- e^{\langle\theta_i^-, T(x^o)\rangle - F(\theta_i^-)} \end{aligned} \tag{6}$$

where the "-" superscript denotes a component with a negative weighting coefficient.

From our derivation, we present the resulting update rules in the following propositions.

Proposition 1: For an observation of $y_t = 1$, given the parameters (v_i, θ_i) for $m_{t-1}(x^o)$ and θ_s, the parameters for $m_t(x^o)$ are

$$v_i^+ = \eta_t v_i e^{\alpha_i} \qquad \theta_i^+ = \theta_i + \theta_s$$

where

$$\alpha_i = F(\theta_i + \theta_s) - F(\theta_i) - F(\theta_s) \tag{7}$$

The interpretation of Proposition 1 is that positive observations result in support of the previous object map that is close to the robot position being scaled up, meaning that the sensor reading is likely if the object is close. The support further from the robot will thus be scaled down.

Proposition 2: For an observation of $y_t = 0$, given the parameters (v_i, θ_i) for $m_{t-1}(x^o)$ and θ_s, the parameters for $m_t(x^o)$ are

$$\begin{aligned} v_i^+ &= \eta_t v_i & \theta_i^+ &= \theta_i \\ v_i^- &= -\eta_t v_i e^{\alpha_i} & \theta_i^- &= \theta_i + \theta_s \end{aligned}$$

where α_i is computed as previously defined in Equation 7.

Essentially, in this case the number of components in the mixture will double. Each component of the original mixture will produce a scaled copy of itself (represented by parameters v_i^+, θ_i^+) and a component whose coefficient has opposite sign (represented by parameters v_i^- and θ_i^-). Such growth reveals the necessity of the atypical mixture model representation for the object map. Fig. 1 illustrates this behavior, with an initial object map comprised of only positive components, for $\mathcal{W} = \mathbb{R}^1$.

The rules given by both propositions provide an intuitive manner for updating the object map given the current sensor measurement, the object map, and the sensor parameter θ_s. This implementation is described in Algorithm 1 for a single time-step. Here we see that it is sufficient to separate positive and negative components for bookkeeping, since the object map at each time-step may contain both types of components.

Algorithm 1 Object Map Update for a Single Time Stage

```
Input: $y_t, \theta_s, m_{t-1} = \{(v_i, \theta_i)\}_{i=1}^n$
  $n_t = \left(\sum_{i=1}^n v_i e^{\alpha_i}\right)^{-1}$ where $\alpha_i$ is given by (7)
  if $y_t = 1$ then
    for each $(v_i, \theta_i) \in m_{t-1}$ do
      $\theta_i^+ = \theta_i + \theta_s$
      $\alpha_i = F(\theta_i^+) - F(\theta_i) - F(\theta_s)$
      $v_i^+ = \eta_t v_i e^{\alpha_i}$
      $m_t = m_t \cup \{(v_i^+, \theta_i^+)\}$
    end for
  else
    $n_t = 1 - n_t$ (take the complement)
    for each $(v_i, \theta_i) \in m_{t-1}$ do
      $\theta_i^+ = \theta_i$
      $v_i^+ = \eta_t v_i$
      $\theta_i^- = \theta_i + \theta_s$
      $\alpha_i = F(\theta_i^-) - F(\theta_i) - F(\theta_s)$
      $v_i^- = -\eta_t v_i e^{\alpha_i}$
      $m_t = m_t \cup \{(v_i^+, \theta_i^+, v_i^-, \theta_i^-)\}$
    end for
  end if
Output: $m_t$
```

III. Probabilistic Search

A. Search Strategy

Given a continuous representation of the object map at time t, a greedy search strategy follows the direction of the object map's positive gradient. That is, the desired robot state $\tilde{x}_{t+1}^r$ at time $t+1$ is

$$\tilde{x}_{t+1}^r = x_t^r + \gamma \, \nabla m_t(x^o) \tag{8}$$

for a sufficiently small step-size γ. The robot's control input u_t is simply the difference of $\tilde{x}_{t+1}^r$ and the robot's current position, such that the control input satisfies the constraint given in Equation 1. Our exact representation produces a direct means for computing the gradient of the object map. By assuming that the object map at time t consists of both positive and negative components, the gradient of Equation 6 yields

$$\nabla m_t(x^o) = \sum_{i=1}^n v_i^+ \, \nabla \left(e^{\langle \theta_i^+, T(x^o) \rangle - F(\theta_i^+)} \right) + \sum_{i=1}^n v_i^- \, \nabla \left(e^{\langle \theta_i^-, T(x^o) \rangle - F(\theta_i^-)} \right) \tag{9}$$

which holds by the linearity of the gradient operator.

Additionally, we explicitly show the gradient of an atypical mixture of Gaussians specifically for $\mathcal{W} = \mathbb{R}^2$. Once this mixture is obtained from the modified update rules, presented in Appendix II, the object map's form is

$$m_t(x^o) = \sum_{i=1}^n v_i^+ \mathcal{N}(u_i^+, \Sigma_i^+) + \sum_{i=1}^n v_i^- \mathcal{N}(u_i^-, \Sigma_i^-) \tag{10}$$

where a component's mean μ_i is a two-dimensional column vector and Σ_i is a two-dimensional covariance matrix. We obtain the gradient of the Gaussian mixture (Equation 10), which is

$$\nabla m_t(x^o) = \sum_{i=1}^n v_i^+ \nabla \mathcal{N}(\mu_i^+, \Sigma_i^+) + \sum_{i=1}^n v_i^- \nabla \mathcal{N}(\mu_i^-, \Sigma_i^-) \tag{11}$$

Furthermore, the gradient of a Gaussian component is given as a two-dimensional column vector where it is evaluated at the current position of the robot at time t (where the time subscript has been temporarily replaced by the vector element). Thus, the gradient of the object map is the sum of partial derivatives

$$\nabla \, m_t(x^o) = \sum_{i=1}^n v_i^+ \left[\frac{\partial \mathcal{N}}{\partial x_1^r}, \frac{\partial \mathcal{N}}{\partial x_2} \right]^T + \sum_{i=1}^n v_i^- \left[\frac{\partial \mathcal{N}}{\partial x_1^r}, \frac{\partial \mathcal{N}}{\partial x_2} \right]^T \tag{12}$$

The partials of these components take a more direct form arising from the structure of the object map component covariance matrices. One such condition is that the object map is comprised of circular Gaussian components, each with a covariance matrix of the form

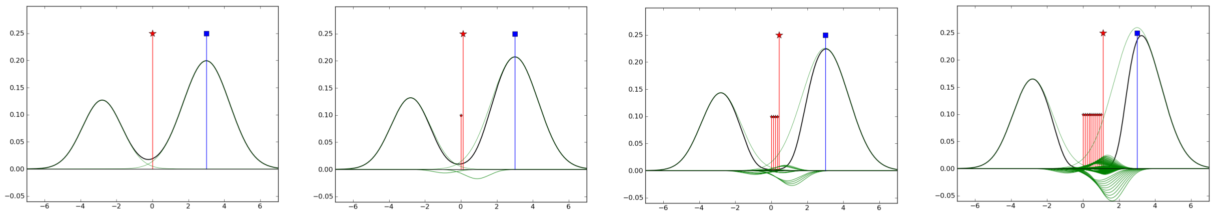
$$\Sigma_i = \sigma_i^2 \cdot I \tag{13}$$

where σ_i^2 is the variance and I denotes the two-dimensional identity matrix. Therefore, a positive component partial is of the form

$$v_i^+ \left[\frac{\partial \mathcal{N}}{\partial x_1}, \frac{\partial \mathcal{N}}{\partial x_2} \right]^T = - \frac{v_i^+ \mathcal{N}(\mu_i^+, \Sigma_i^+)}{(\sigma_i^+)^4} \cdot \begin{bmatrix} (x_1 - \mu_{i,1}^+) \\ (x_2 - \mu_{i,2}^+) \end{bmatrix}$$

where this component acts as an attractor, pulling the robot toward its mean, proportional to its squared variance. The partial of a negative component is of the same form, but has the opposite effect due to its negative weight. Thus, the computed component gradients above can be substituted into Equation 8 to produce the control input.

When employing gradient ascent it is imperative that care be taken for the existence of local maxima. It is possible that the robot will reach a steady-state at some x_t^r, following the presented gradient ascent strategy, given successive positive detections and will not find the object. To overcome this



(a) $t = 0, y_0 = 0$

(b) $t = 1, y_1 = 0$

(c) $t = 4, y_4 = 0$

(d) $t = 10, y_{10} = 0$

Fig. 1. Exact evolution of $m(x^o)$ for a one-dimensional problem. For this series only negative detections were observed. The large red star, at the tip of the red impulse, denotes the robot's current position (the smaller stars are the previous positions), and the object's position is given by the blue square.

drawback a method used in gradient following problems will be employed, namely a random walk. Such a method can also be used to escape a region where mixture components equally "pull" or "push" the robot, rendering the robot stationary.

B. Approximation with a Regularized Particle Filter

Although we demonstrate that the object map can be represented with a finite representation, the representation size is not fixed. Though the number of natural parameters θ_i of each distribution will not increase, we may require parameters for up to $N2^t$ components at stage t, where N is the number of components present in m_0. This exponential growth of the representation may cause serious computational issues where a large number of search stages are needed. However, as the representation and its evolving form is known, we can easily produce a model approximation with a regularized particle filter.

The use of a regularized sampling importance resampling (SIR) particle filter [8], [17], [18] allows us to maintain a continuous representation of object location, given our knowledge of the evolving object map form. Such a filter provides a continuous approximation through the careful choice of a regularization kernel. Since the object map's parameterization remains in the exponential family class we can readily choose an appropriate regularization kernel, but such a choice is also dependent on the initial map form. That is, given an initial map consisting of a specific density (or mixture), it is intuitive to use this density as the kernel due to our knowledge of the evolving map's form.

A regularized SIR particle filter process closely resembles the standard SIR filter. First, N samples are drawn from m_t resulting in N particles (x^i, w^i) with a state and associated weight. The weights of each particle are updated by computing the probability of detection (Equation 2), given the appropriate sensor observation y_t. However, rather than resampling from a discrete density (typically the Dirac delta), samples are drawn from the kernel distribution. This yields the approximation in the form

$$\hat{m}_{t+1}(x^o) = \sum_{i=1}^{N} w_i \, K(x_i; x^o)$$

where K is the regularization kernel with N particles.

For the results presented in Section III-C, we employ the regularized SIR particle filter to approximate the object map, which is given as a mixture of Gaussians. An obvious choice for our regularization kernel is a Gaussian distribution, where the map approximation

$$\hat{m}_{t+1}(x^o) = \sum_{i=1}^{N} w_i \, \mathcal{N}(x_i, \Sigma; x^o)$$

is simply a mixture of Gaussians, each with a covariance matrix Σ. Though there exist principled methods for selecting Σ, we choose this parameter experimentally. Note that this continuous, finite approximation of our object map is still in exponential family form, though solely comprised of positive mixture components. Thus, our general update rules are still applicable for the evolution of the object map. Furthermore, the gradient ascent strategy is still valid, though only positive components are present.

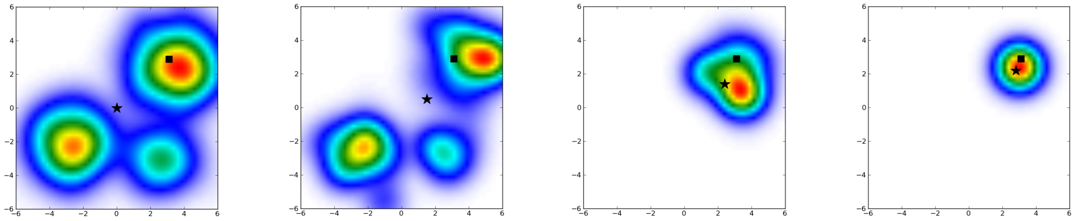
C. Simulations

We present both single and multi-agent search simulations for $\mathcal{W} = \mathbb{R}^2$ using the regularized SIR particle filter and the gradient ascent search strategy. An object is considered "found" if it lies within a radius of one-half the gradient ascent step size (0.1) from the agent. The initial object map is given as a mixture of circular Gaussians (subfigure 2a). The equipped binary sensor's covariance matrix, for each agent, is $\Sigma_s = 0.4 \cdot I$. The number of particles used for the regularized filter is $N = 600$ and our regularization kernel is a Gaussian distribution.

For the initial map 400 trials were conducted, each using the same initial robot state, sensor variance, and the true object position. For these 400 trials the average time to find the object, with a single agent, was ≈ 1193 time-steps. However, if the object's position was exactly known, for this scenario, the robot would need no more than 16 time-steps to reach it. This disparity illustrates the amount of uncertainty of searching solely with a binary sensor. Fig. 2 shows a trial where the object was found in 57 time-steps.

For the multi-agent simulations, 400 trials were again performed in the manner similar to the single-agent results. That is, each agent is separately governed by the gradient ascent strategy and must individually invoke a random walk when necessary. Furthermore, given the object map $m_{t-1}(x^o)$, each agent's current position, and the set of observations $\boldsymbol{Y}_t$ where

$$\boldsymbol{Y}_t = \{\boldsymbol{y}_t^1, \boldsymbol{y}_t^2, \cdots, \boldsymbol{y}_t^N\}$$



(a) $t = 0$ (b) $t = 15, y_{15} = 0$ (c) $t = 24, y_{24} = 1$ (d) $t = 52, y_{52} = 1$

Fig. 2. Evolution of $m_{(}x^o)$ for a two-dimensional problem employing the regularized particle filter. The robot (black star) follows a gradient ascent strategy dictated by the object's (black square) likelihood. Negative observations are made until $t = 24$, which scales the object likelihood close to the robot. Successive positive detections further scales this as the robot follows the gradient ascent strategy.

for N agents, the object map $m_t(x^o)$ is generated by the given update laws. The average time using $N = 10$ agents was ≈ 89 time-steps. This result is intuitive, as multiple agents provide much more information in the workspace, which is reflected in the evolution of their shared object map. Furthermore, this increased effectiveness illustrates the viability of employing multiple agents, equipped with binary sensors, though following a sub-optimal strategy.

IV. DISCUSSION

A. Summary

Using the self-conjugacy properties of the exponential family we demonstrated that, for a specific class of sensor models and initial conditions, we can represent the object pdf exactly using a finite representation. We derived the filtering (or inference) equations that model the posterior likelihood of object location conditioned on robot motion and a sequence of noisy binary measurements. Due to the possibility of exponential growth of mixture components, regularized particle filtering was employed for both single-agent and multi-agent models requiring a large number of stages. A number of target applications forms the basis for future exploration of this approach.

B. Future Work

As the viability of our Bayesian search model has been ascertained, there are various relevant extensions of this work. The non-stationary single target search can be applied to a moving target, which is simply the search-and-track problem. Logically, the non-stationary problem is readily extensible towards the search for multiple targets. Affirming that all targets are independent of each other allows their belief representation to be partitioned into separate pdfs, allowing the possibility for planning search trajectories. Furthermore, both the stationary and non-stationary problems can be modelled as a POMDP, which provides an amenable structure for optimization of the search problem.

A probabilistic localization problem entails using (solely) a binary sensor determine the location of an object. If the observation model is probabilistic, the object cannot be perfectly localized in finite time. Thus, the criterion to be minimized would be expected entropy of the object map. With a POMDP formulation, the policy optimization would be performed over the subset of belief-feedback policies that guarantee consistent estimation.

One advantage of using mixtures of distributions from an exponential family is that many good approximation techniques exist. For example, algorithms for soft [19] and hard [20] clustering based on Bregman divergences can significantly reduce the number of components required in the representation, while causing only a small perturbation (with respect to relative entropy) between the true and approximated distribution. Mixtures from the process model described in this paper tend to be good candidates for approximation using these methods, because the different components tend to naturally form clusters. This is due to the incremental updating of the map and robot position (Fig. 1). Experimental experience tends to indicate this is not an exceptional case.

APPENDIX I
DERIVATION OF UPDATE RULES

Here we explicitly derive the update rules, stemming from the general Bayesian inference for our model, that parameterizes the evolution of the posterior. This is accomplished by assuming both the initial distribution describing the object's location and the sensor model belong to the class of the exponential family of distributions. Furthermore, our derivation results in explicit update rules for the two possible sensor measurements.

First, we give the initial prior as a standard mixture of exponential family distributions to represent the probable object locations over $\mathcal{W}$, represented as the convex combination

$$m_0(x^o) = \sum_{i=1}^{n} w_i \exp\left[\langle \theta_i, T(x^o)\rangle - F(\theta_i)\right] \tag{14}$$

where $\sum_i^n w_i = 1$ and $w_i > 0$ for all i. However, we demonstrate the derivation inductively by defining an atypical mixture with non-standard weighting coefficients v_i,

$$m_0(x^o) := \sum_{i=1}^{n} v_i \exp\left[\langle \theta_i, T(x^o)\rangle - F(\theta_i)\right] \tag{15}$$

where $\sum_i^n v_i = 1$ but $v_i \in \mathbb{R}$, i.e., not constrained to the set of positive reals, and $m_t(x^o) \geq 0$ for all $x^o \in \mathcal{W}$. Essentially, this is a mixture of "positive" components summed with "negative" components. We denote the weighting coefficients of the components of the mixture with v_i to emphasize that this mixture is not the usual convex sum of component distributions. Furthermore, note that any distribution of the form of Equation 14 can be written in the form of Equation 15 without modification, so the initial condition of the induction is shown.

Our derivation continues by demonstrating the inductive step first for the case of $y_t = 1$. Substituting the sensor model

$$\mathrm{p}_{\boldsymbol{y}_t|x^r_t,\boldsymbol{x}^o}\left(\boldsymbol{y}_t = 1|x^r_t, x^o\right) = e^{\langle\theta_s, T(x^o)\rangle - F(\theta_s)}$$

into the generic Bayesian update law

$$m_t(x^o) = \eta_t \; \mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x}^o,x^r_t}\left(y_t|x^o, x^r_t\right) \; m_{t-1}(x^o) \tag{16}$$

yields

$$m_t(x^o) = \eta_t \; \exp\left[\langle\theta_s, T(x^o)\rangle - F(\theta_s)\right] \; m_{t-1}(x^o)$$

Using the inductive hypothesis, by assuming m_{t-1} is of the form of Equation 15, allows the following where

$$\begin{aligned} m_t(x^o) &= \eta_t \; e^{\langle\theta_s, T(x^o)\rangle - F(\theta_s)} \; \sum_{i=1}^{n} v_i e^{\langle\theta_i, T(x^o)\rangle - F(\theta_i)} \\ &= \eta_t \; \sum_{i=1}^{n} v_i e^{\langle\theta_i+\theta_s, T(x^o)\rangle - F(\theta_i) - F(\theta_s)} \\ &= \eta_t \; \sum_{i=1}^{n} v_i e^{\alpha_i} \; e^{\langle\theta_i+\theta_s, T(x^o)\rangle - F(\theta_i+\theta_s)} \end{aligned}$$

such that $\alpha_i = F(\theta_i + \theta_s) - F(\theta_i) - F(\theta_s)$. Rather than computing η_t with

$$\eta_t := \left[\int \mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x}^o,x^r_t}\left(y_t|x^o, x^r_t\right) \; m_{t-1}(x^o) dx^o\right]^{-1}$$

it is computed based on its normalizing property alone, i.e.

$$\begin{aligned} \eta_t &= \left[\int \sum_{i=1}^{n} v_i e^{\alpha_i} \; e^{\langle\theta_i+\theta_s, T(x^o)\rangle - F(\theta_i+\theta_s)} dx^o\right]^{-1} \\ &= \left[\sum_{i=1}^{n} v_i e^{\alpha_i} \; \int e^{\langle\theta_i+\theta_s, T(x^o)\rangle - F(\theta_i+\theta_s)} dx^o\right]^{-1} \\ &= \left[\sum_{i=1}^{n} v_i e^{\alpha_i}\right]^{-1} \end{aligned} \tag{17}$$

The final step is performed by noting that each component is a probability distribution; they each integrate to one. Thus, the object map at time t can be parameterized as

$$m_t(x^o) = \sum_{i=1}^{n} v_i^+ \exp\left[\langle\theta_i^+, T(x^o)\rangle - F(\theta_i^+)\right] \tag{18}$$

with the corresponding parameter update

$$v_i^+ = \eta_t v_i e^{\alpha_i} \tag{19}$$

$$\theta_i^+ = \theta_i + \theta_s \tag{20}$$

The superscript '+' indicates the component in the stage t object map has a positive weighting coefficient. Hence, the parameters (v_i^+, θ_i^+) are generated (through the Bayesian filtering equation) by the component in the stage $t-1$ object map, with parameters (v_i, θ_i), given an observation of $y_t = 1$. This satisfies the first case of our derivation, because Equation 18 is in canonical exponential family form.

The second case of $y_t = 0$ is considered with the same inductive reasoning presented above. Substituting the sensor model complement into Equation 16 yields

$$m_t(x^o) = \eta_t \; \left(1 - e^{\langle\theta_s, T(x^o)\rangle - F(\theta_s)}\right) \; m_{t-1}(x^o)$$

Utilizing the inductive hypothesis and using the same steps as the previous case, manipulating

$$m_t(x^o) = \eta_t m_{t-1}(x^o) - \eta_t \sum_{i=1}^{n} v_i e^{\alpha_i} e^{\langle\theta_i+\theta_s, T(x^o)\rangle - F(\theta_i+\theta_s)}$$

results in the parameterization

$$\begin{aligned} m_t(x^o) = \sum_{i=1}^{n} & v_i^+ e^{\langle\theta_i^+, T(x^o)\rangle - F(\theta_i^+)} \\ &+ \sum_{i=1}^{n} v_i^- e^{\langle\theta_i^-, T(x^o)\rangle - F(\theta_i^-)} \end{aligned}$$

Finally, the parameter update for this step is

$$v_i^+ = \eta_t v_i \tag{21}$$

$$\theta_i^+ = \theta_i \tag{22}$$

$$v_i^- = -\eta_t v_i e^{\alpha_i} \tag{23}$$

$$\theta_i^- = \theta_i + \theta_s \tag{24}$$

The normalization constant η_t can be computed by simply taking the complement of Equation 17. The superscript '+' retains the meaning as before, and the superscript '$-$' similarly indicates a component with a negative weighting coefficient. For the case of either observation, the property $m_t(x^o) \geq 0$ holds as a consequence of Bayes Theorem. Thus, as the second observation case holds through the induction step, we are able to represent m_t with a finite number of parameters in exponential family form and provide explicit update rules for both observations.

Appendix II
Filtering with Gaussian Mixtures

This appendix demonstrates our method using the exponential family representation for mixture of Gaussian distributions. First, we illustrate the binary sensor model. We then present the update rules specifically for a mixture of Gaussians, using the derivations presented in Appendix I.

The binary sensor model for the case of $\mathcal{W} = \mathbb{R}^1$ is parameterized as

$$\mathrm{p}_{\boldsymbol{y}_t|x^r_t,\boldsymbol{x}^o}\left(\boldsymbol{y}_t = 1|x^r_t, x^o\right) = \kappa \cdot \mathcal{N}(x^r_t, \Sigma_s)$$

which can be expressed in the exponential family form with

$$\begin{aligned} \theta_s &= \left[\frac{x^r_t}{\sigma_s^2}, -\frac{1}{2\sigma_s^2}\right]^T \\ F(\theta_s) &= \frac{(x^r_t)^2}{2\sigma_s^2} + \frac{1}{2}\ln(2\pi\sigma_s^2) \\ T(x^o) &= \left[x^o, (x^o)^2\right]^T \end{aligned}$$

The importance of κ is given for the univariate case. If the sensor's standard deviation, σ_s, is chosen such that

$$\sigma_s < \frac{1}{\sqrt{2\pi}}$$

then the sensor model is not a valid pdf (does not integrate to one over $\mathcal{W}$). Thus κ is used to enforce this necessary condition.

The sensor model for $\mathcal{W} = \mathbb{R}^2$ is simply a bivariate Gaussian parameterized in exponential family form. This form can be seen in our filtering equations for an initial object map composed of Gaussian components. Likewise, the reader may consult [13], [14] for further information on the exponential family of distributions.

Here, we show our derived update rules given an object map comprised of Gaussian components. That is, the object map is of the form

$$m_{t-1}(x^o) = \sum_{i=1}^{n} v_i \ \mathcal{N}(\mu_i, \Sigma_i)$$

at time $t-1$. Given this multi-variate mixture of Gaussians, a component's mean vector μ_i and covariance matrix Σ_i can be represented by the natural parameters θ_i and sufficient statistic $T(x^o)$ in exponential family form with

$$\theta_i = [\Lambda_i, \Psi_i]^T = \left[\Sigma_i^{-1}\mu_i, \frac{1}{2}\Sigma_i^{-1}\right]^T \tag{25}$$
$$T(x^o) = \left[x^o, -x^o(x^o)^T\right]^T$$

The log normalizer can be expressed in terms of the natural parameters of the distribution.

$$F(\theta_i) = \frac{1}{4}\text{Trace}(\Psi_i^{-1}\Lambda_i\Lambda_i^T) - \frac{1}{2}\ln(\det(\Psi_i)) + \frac{1}{2}\ln(\pi)$$

One the initial distribution is "converted" from a mixture of Gaussians to a representation in the exponential family form, we can explicitly use our update rules to respond to the appropriate observation.

We can, of course, convert the mixture of exponentials representation back to a mixture of Gaussians representation using the mapping from natural parameters to mean and covariance.

$$\mu = \frac{1}{2}\Lambda^{-1}\Psi \tag{26}$$

$$\Sigma = \frac{1}{2}\Psi^{-1} \tag{27}$$

Using this mapping, we see that our update rule (in terms of mean and covariance) for $y_t = 0$ is

$$\begin{aligned}
v_i^+ &= \eta_t v_i & v_i^- &= -\eta_t \ \kappa v_i e^{-\alpha_i} \\
\mu_i^+ &= \mu_i & \mu_i^- &= (\Sigma_s^{-1} + \Sigma_i^{-1})^{-1}(\Sigma_s^{-1}\mu_s + \Sigma_i^{-1}\mu_i) \\
\Sigma_i^+ &= \Sigma_i & \Sigma_i^- &= (\Sigma_s^{-1} + \Sigma_i^{-1})^{-1}
\end{aligned}$$

This can be shown by substituting (25) and (26)-(27) into (21)-(24). Substituting (25) and (26)-(27) into (19)-(20) will give the corresponding rule for $y_t = 1$.

References

[1] B. O. Koopman, "Search and its optimization," *The American Mathematical Monthly*, vol. 86, no. 7, pp. 527–540, 1979.

[2] S. J. Benkoski, M. G. Monticino, and J. R. Weisinger, "A survey of the search theory literature," *Naval Research Logistics (NRL)*, vol. 38, no. 4, pp. 469–494, 1991.

[3] L. D. Stone, "What's happened in search theory since the 1975 Lanchester Prize?" *Operations Research*, vol. 37, no. 3, pp. pp. 501–506, 1989.

[4] E.-M. Wong, F. Bourgault, and T. Furukawa, "Multi-vehicle Bayesian search for multiple lost targets," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 3169–3174.

[5] F. Bourgault, T. Furukawa, and H. Durrant-Whyte, "Optimal search for a lost target in a Bayesian world," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, S. Yuta, H. Asama, E. Prassler, T. Tsubouchi, and S. Thrun, Eds. Berlin, Germany: Springer, 2006, vol. 24, pp. 209–222.

[6] T. Chung, "On probabilistic search decisions under searcher motion constraints," in *Algorithmic Foundation of Robotics VIII*, ser. Springer Tracts in Advanced Robotics. Berlin, Germany: Springer, 2009, vol. 57, pp. 501–516.

[7] L. Stone, *Theory of Optimal Search*, ser. Mathematics in Science and Engineering. New York, NY: Academic Press, 1975, vol. 118.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[9] H. Choset, "Coverage for robotics-A survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001.

[10] Y. Zhang, M. Schervish, and H. Choset, "Probabilistic hierarchical spatial model for mine locations and its application in robotic landmine search," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 681–689.

[11] T. Chung and J. Burdick, "A decision-making framework for control strategies in probabilistic search," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 4386–4393.

[12] J. Ma, T. H. Chung, and J. Burdick, "A probabilistic framework for object search with 6-dof pose estimation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1209–1228, 2011. [Online]. Available: http://ijr.sagepub.com/content/30/10/1209

[13] L. Brown, *Fundamentals of Statistical Exponential Families: With Applications in Statistical Decision Theory*. Hayward, CA: Institute of Mathematical Statistics, 1986.

[14] F. Nielsen and V. Garcia, "Statistical exponential families: a digest with flash cards," arXiv:0911.4863, Nov. 2009.

[15] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. New York, NY: John Wiley & Sons, 1994.

[16] B. Zhang and C. Zhang, "Finite mixture models with negative components," in *Machine Learning and Data Mining in Pattern Recognition*, ser. Lecture Notes in Computer Science, P. Perner and A. Imiya, Eds. Berlin, Germany: Springer, 2005, vol. 3587, pp. 31–41.

[17] A. Doucet, N. D. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer, 2001.

[18] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[19] V. Garcia and F. Nielsen, "Simplification and hierarchical representations of mixtures of exponential families," *Signal Processing*, vol. 90, no. 12, pp. 3197–3212, 2010.

[20] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *The Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.