Article

The International Journal of Robotics Research
1–14

DOI: 10.1177/0278364917709507
journals.sagepub.com/home/ijr

Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots

Philip Dames[1], Pratap Tokekar[2] and Vijay Kumar[3]

Abstract

Target tracking is a fundamental problem in robotics research and has been the subject of detailed studies over the years. In this paper, we generate a data-driven target model from a real-world dataset of taxi motions. This model includes target motion, appearance, and disappearance from the search area. Using this target model, we introduce a new formulation of the mobile target tracking problem based on the mathematical concept of random finite sets. This formulation allows for tracking an unknown and dynamic number of mobile targets with a team of robots. We show how to employ the probability hypothesis density filter to simultaneously estimate the number of targets and their positions. Next, we present a greedy algorithm for assigning trajectories to the robots to allow them to actively track the targets. We prove that the greedy algorithm is a two-approximation for maximizing submodular tracking objective functions. We examine two such functions: the mutual information between the estimated target positions and future measurements from the robots and a new objective that maximizes the expected number of targets detected by the robot team. We provide extensive simulation evaluations to validate the performance of our data-driven motion model and to compare the behavior and tracking performance of robots using our objective functions.

Keywords

Autonomous agents, path planning for multiple mobile robot systems, sensor fusion, surveillance systems

1. Introduction

There are many applications that require target detection, localization, and tracking, such as search-and-rescue (Furukawa et al., 2006), wildlife tracking (Tokekar et al., 2013), surveillance (Grocholsky et al., 2006), and building smart cities (Li et al., 2011). Consequently, such problems have long been a subject of study in the robotics community. Active target tracking typically refers to two types of tasks: estimating the trajectories of the targets from the sensor data, and actively controlling the motion of the sensors to gather the data. Both problems have been studied in the literature under various settings. Solutions have been presented for radio-based sensors (Hollinger et al., 2012), range-only sensors (Zhou and Roumeliotis, 2008), bearing-only sensors (Logothetis et al., 1997), and range and/or bearing sensors (Zhou and Roumeliotis, 2011) in both centralized and decentralized settings.

Frew and Rock (2003) designed optimal trajectories for a single robot to track a single moving target using monocular vision. The problem of keeping targets in a robot's field of view can be formulated as a visual servoing problem. Gans et al. (2011) designed a controller which guarantees stability while keeping three or fewer targets in the field of view of a single mobile robot. Tracking multiple targets with multiple robots requires explicit or implicit assignment of targets to robots. Spletzer and Taylor (2003) presented a general solution for the multi-robot, multi-target case using a particle filter formulation. Xu et al. (2013) presented a mixed non-linear integer programming formulation for assigning robots to targets as well as for determining optimal robot positioning. Such a formulation is not directly applicable in our case since the number of targets itself is unknown, and thus explicit assignment is not possible.

Recently, there has been some work on actively detecting and/or localizing an unknown number of stationary targets using radio sensors (Song et al., 2012; Kim et al., 2014), range-only sensors (Charrow et al., 2015), and arbitrary sensor models (Dames and Kumar, 2015). Unlike most existing work, this paper addresses the case of tracking an *unknown* and *varying* number of *indistinguishable*

[1]Department of Mechanical Engineering, Temple University, USA
[2]Department of Electrical and Computer Engineering, Virginia Tech, USA
[3]Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, USA

Corresponding author:
Philip Dames, Temple University, 1947 North 12th Street, Philadelphia, PA 19122 USA.
Email: pdames@temple.edu

targets. This introduces a number of challenges. First, we cannot maintain a separate estimator for each target, since the required number of estimators itself is unknown. Second, we must account for the fact that targets appear and disappear from the environment over time. Third, we cannot maintain a history of the target positions because we cannot uniquely identify individual targets, making prediction difficult. Finally, the system must be capable of handling false positive and false negative detections and unknown data association in addition to sensor noise. Despite these challenges, we present positive results towards solving the problem.

To solve the estimation problem we turn to random finite sets (RFSs, Mahler, 2007). RFSs are random variables whose realizations are finite sets. Distributions over RFSs have both a distribution over the cardinality of the set (i.e. number of targets) and a distribution over the elements of the set (i.e. positions of the targets). RFS-based methods have one key advantage for multi-target tracking over standard tracking algorithms such as the Kalman filter or particle filter: they do not require the data association problem to be explicitly solved. Solutions based on the Kalman filter or particle filter require solving the data association problem, i.e. matching measurements to targets, prior to applying the filter updates while RFS-based methods simultaneously perform data association and tracking. Also, traditional data association solvers do not scale well with the number of targets since the problem is combinatorial.

The probability hypothesis density (PHD) filter (Mahler, 2003a) is the most common estimation strategy based on RFSs. The PHD filter and RFSs have recently been used for robot localization (Atanasov et al., 2016), simultaneous localization and mapping (Leung et al., 2014), localizing static targets (Ristic et al., 2011; Dames and Kumar, 2015), and more (Adams et al., 2014). Unlike all these works, we do not assume that the targets remain stationary.

An important consideration for target tracking is the motion model for the targets. A number of parametric motion models have been proposed in the literature (see the work by Li and Jilkov (2003) for a detailed survey). In general, most motion models can directly be used in the PHD filter. We employ a data-driven technique to extract the motion model, instead of assuming any parametric form. Specifically, we use a Gaussian process (GP) regression to learn a map of velocity vectors for the targets, as Joseph et al. (2011) have also done. GPs offer a highly flexible method for learning motion models. They can capture state-dependent behavior, which we will take advantage of to learn the traffic flow patterns in our experiments. GPs can be used in any situation with a prior dataset to characterize the target motion. Additionally, we show how to empirically model the appearance and disappearance of targets within an environment.

Next, we present a control policy to assign trajectories for all robots in order to maximize the objective function over a receding horizon. We study two objective functions using the PHD filter. The first is mutual information (MI), which is a common objective function in active perception tasks. The second is the expected number of detections (END) by the robot team and is a new objective that explicitly considers the possibility of missed detections. We show that both objective functions are submodular and use a result from the work by Tokekar et al. (2014) to prove that our greedy control policy is a two-approximation. Tokekar et al. (2014) used this greedy algorithm to track a *known* and *fixed* number of mobile targets using a team of aerial robots. In this paper, we build on this work to allow for the case of an *unknown* and *changing* number of targets.

In addition to the theoretical analysis we offer, we evaluate our algorithm using simulated experiments. While our framework may be applied to a number of robot and sensor models, for the purposes of testing we restrict our attention to fixed-wing aerial robots with downward-facing cameras. We use a real-world taxi motion dataset from the work by Piorkowski et al. (2009) to learn the target motion model and to drive the target motion in the experiments. This paper builds upon our previous work (Dames et al., 2015) and offers an additional discussion of the system, new simulation results, and more detailed analysis. In particular, we examine the performance of our system as a function of the relative speed of the targets and the robots. We also compare the performance of teams of robots using our proposed objective functions to that of teams of robots following a coverage-based controller. The simulation results reveal that robot teams using the information-based control objective track a smaller number of targets with higher precision compared to teams that maximize the expected number of detections. The coverage-based controller tracks a comparable number of targets to the other methods but with a higher uncertainty in the target positions.

2. Problem formulation

We address the problem of a team of R robots monitoring an area in order to detect, localize, and track an unknown number of moving targets using an inexpensive camera. The robots are able to localize themselves within the environment (e.g. using GPS). We denote the pose of robot r at time t by q_t^r.

The number of targets, n_t, is unknown and varies over time, since individual targets may enter and leave the area of interest. We use RFSs to represent the number and state of targets at any time. In the target tracking scenario, RFSs may represent either the measurements or the states of the targets. Let $X_t = \{x_{1,t}, x_{2,t}, \ldots, x_{n_t,t}\}$ denote a realization of a RFS of target states at time t, where $x_{i,t}$ is the state of target i at time t. Note that the set X_t contains information about both the number of targets (i.e. the size of the set) and the target locations (i.e. the elements of the set). A probability distribution of a RFS is characterized by a discrete distribution over the cardinality of the set and a family of densities

for the elements of the set conditioned on the size, i.e.

$$p(X = \{x_1, \ldots, x_n\}) = p(|X| = n)\; p(\{x_1, \ldots, x_n\} \mid |X| = n) \quad (1)$$

The first statistical moment of a distribution over a RFS is called the probability hypothesis density (PHD). The PHD, $v(x)$, is a density function over the state space of an individual target with the property that the integral over any region S is the expected number of targets in that region, i.e.

$$\int_S v(x)\; dx = E[|X \cap S|] \quad (2)$$

The PHD filter makes the assumption that targets are independent and identically distributed and that the cardinality of the target set is characterized by a Poisson distribution. The likelihood of such an RFS is

$$p(X) = \exp\left(- \int v(x)\; dx \right) \prod_{x \in X} v(x) \quad (3)$$

which is fully characterized by the PHD.

Each robot receives a set of measurements $Z_t^r = \{z_{1,t}^r, z_{2,t}^r, \ldots, z_{m,t}^r\}$ to targets that it detects within the field of view (FoV) of its sensor. The number of measurements, m_t, varies over time due to false negative and false positive detections and the motion of the robots and the targets. Let $p_d(x \mid q)$ denote the probability of a robot at q detecting a target with state x. $p_d(x \mid q) = 0$ for targets outside of the FoV of the sensor and where $p_d(x \mid q) < 1$ indicates the possibility of a false negative, i.e., a missed detection. When a target is successfully detected, the sensor returns a measurement $z \sim g(\cdot \mid x, q)$. The sensor can also return measurements to clutter objects, causing false positive detections. Let $c(z \mid q)$ denote the PHD of clutter measurements.

The PHD filter is somewhat analogous to the Kalman filter, recursively updating the statistical moments necessary to fully characterize a distribution over the target states. Like the Kalman filter, there are two equations: the prediction and the update

$$\bar{v}_t(x) = b(x) + \int p_s(\xi) f(x \mid \xi)\, v(\xi)\; d\xi \quad (4)$$

$$v_t(x) = \big(1 - p_d(x \mid q)\big)\bar{v}_t(x) + \sum_{z \in Z_t} \frac{p_d(x \mid q)\, g(z \mid x, q)\, \bar{v}_t(x)}{c(z, q) + \int p_d(\xi \mid q)\, g(z \mid \xi, q)\, \bar{v}_t(\xi)\; d\xi} \quad (5)$$

Here, $\bar{v}_t(\cdot)$ is the predicted target PHD; $b(\cdot)$ is the PHD of target births, which accounts for new targets entering the area; $p_s(\cdot)$ is the target survival probability, which accounts for targets leaving the area; and $f(\cdot \mid \xi)$ is the target motion model. In the following section, we show how these parameters are learned from a real-world dataset.

Note that the representation in the PHD filter is inherently different from more traditional target trackers. With the PHD, there is no notion of target labels or of individual target tracks. Instead, the PHD filter tracks the density of targets over time, yielding information about the bulk motion rather than about the motion of individual targets. Future work will examine the recent labeled MeMBer filter (Reuter et al., 2014), which is also based on RFSs but uses a different representation such that it is able to output labeled target tracks.

3. Target tracking framework

The representative problem that we consider is of a team of fixed-wing aerial robots equipped with downward-facing cameras, tracking vehicles driving on the ground. However, the same methodology could be extended to work with robots with other mobility constraints (e.g. ground vehicles or quadrotor platforms) and other sensor modalities (e.g. lidars or 3D depth cameras).

3.1. Sensor parameterization

The problem of detecting vehicles using aerial imagery has been well studied (Zhao and Nevatia, 2003; Grabner et al., 2008). We use such studies to inform our selection of the sensor detection, measurement, and clutter models. The approaches presented by Grabner et al. (2008) and Zhao and Nevatia (2003) are similar, searching for image features over a range of scales in order to detect cars of different sizes or to detect cars from different elevations or with different image resolutions. In general, the system is able to have a higher detection rate if we accept a larger number of false positive detections (Grabner et al., 2008, Figure 8; Zhao and Nevatia, 2003, Figure 12). The detection rate may also vary with the number of pixels per target, which may be computed using the robot pose, the approximate length scale of a target, and the image resolution, to be

$$\#\text{ pixels per car} = \text{pixels per radian} \times \arctan \frac{\text{length of target}}{\text{distance from camera to target}} \quad (6)$$

We assume a logistic relationship between the number of pixels per target, $n_{\text{px}}(x, q)$, and the detection rate

$$p_d(x \mid q) = p_0 + \frac{p_{d,\max} - p_{d,0}}{1 + \exp\big(- k(n_{\text{px}}(x, q) - n_{px,0})\big)} \quad (7)$$

where $p_{d,0}$, $p_{d,\max}$, k, and $n_{px,0}$ are design parameters.

The camera returns pixel (i.e. bearing) measurements to the cars detected within the image. Using the pose of the robot, we can project measurements onto the ground plane to localize the targets. The measurement model is

$$g(z \mid x, q) = \mathcal{N}\left(z; [r_x, c_x]^T, \sigma^2 I\right) \quad (8)$$

where r_x and c_x are the pixel row and column values in an image taken at q, of a target at x, σ is the standard deviation in pixels, and I is a 2×2 identity matrix.

Like the targets, the clutter is modeled as a Poisson RFS, which is completely characterized by the PHD. Without a priori knowledge of locations that are likely to have clutter, the best choice is to use a uniform distribution over the measurement space. For most computer vision-based detection algorithms, the expected number of clutter detections depends upon the detection model, with a high detection likelihood resulting in a higher detection rate (Zhao and Nevatia, 2003; Grabner et al., 2008).

3.2. Target parameterization

In order to predict how the target set evolves, we need models for the motion of individual targets as well as the birth/death processes of the targets. A number of motion models have been proposed in the literature, ranging from random (Li and Jilkov, 2003) to adversarial (Chung et al., 2011). Often, a mixture of parametric motion models is used (Li and Jilkov, 2005). We take a data-driven approach to model the targets' motion, utilizing real-world datasets that are available (Kotz and Henderson, 2005). In particular, we use a GP regression (Rasmussen and Williams, 2006) to learn the function that maps the position coordinates of the targets to velocity vectors, as shown by Joseph et al. (2011). Unlike Joseph et al. (2011), we use a single GP rather than a mixture of GPs.

A GP regression is a Bayesian approximation technique to learn some function $f(X)$ given measurements $y = f(x)+\epsilon$ corrupted by Gaussian noise, $\epsilon \sim \mathcal{N}(0,\sigma^2)$. Here, $x = [x_1,x_2]^T$ is the position of the target and $y = [y_1,y_2]^T$ is the velocity of the target. We learn two independent functions, f_1 and f_2, one for each axis of the ground plane, assuming that the velocities along the two axes are independent. Instead of assuming a parametric model for f_i, a GP regression assumes that the joint distribution of $f_i(X)$ defined over any collection of positions, $X = \{x^1,\ldots,x^k\}$, is always Gaussian. Thus, each $f_i(X)$ is completely specified by its mean function, $\mu_i(X) = E[f_i(X)]$ and covariance function, $K_i(X,X') = E[(f_i(X)-\mu_i(X))(f_i(X')-\mu_i(X'))]$.

Given observed velocity vectors Y_1 and Y_2 taken at some subset of positions, X, a GP regression predicts the velocity vectors at some other set of positions, X^*, as a Gaussian distribution with conditional mean and variance values (Rasmussen and Williams, 2006)

$$\mu_i(X^*|X) = \mu_i(X^*) + K_i(X^*,X)\,[K_i(X,X)+\sigma^2 I]^{-1}(Y_i - \mu_i(X))$$

$$\sigma_i^2(X^*|X) = K_i(X^*,X^*) - K_i(X^*,X)\,[K_i(X,X)+\sigma^2 I]^{-1}K_i(X,X^*)$$

where $K_i(X,X')$ is a matrix whose $(m,n)^{th}$ entry is given by the covariance between $x^m \in X$ and $x^n \in X'$. We take the prior function, $\mu_i(X)$, to be a zero-mean distribution. Thus, if the covariance function is known, the above equations can fully predict the velocity values at arbitrary positions.

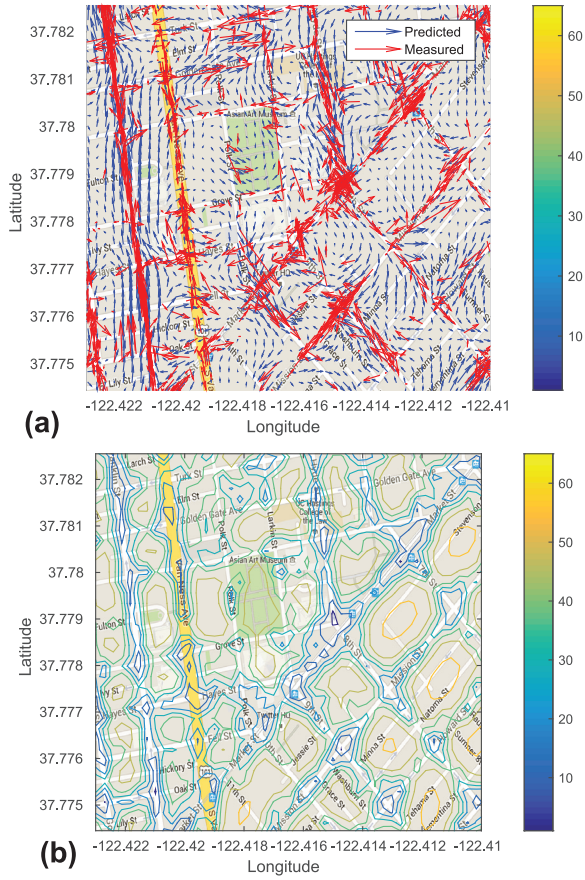

Fig. 1. The (a) mean and (b) standard deviation of the Gaussian Process regression motion model overlaid on the map. We only show a small patch of the environment to show the detail. The measured velocity vectors are shown in red, and the velocity vectors predicted over a grid are given in blue. The units of the velocity are m/s.

We assume that the covariance function belongs to the Matérn class with the parameter $\nu = 3/2$ (Rasmussen and Williams, 2006), since this choice of covariance function yields a better fit compared to the standard squared-exponential function used by Joseph et al. (2011). The length hyperparameter of the Matérn covariance is learned using training data from the Cabspotting taxi dataset from the work by Piorkowski et al. (2009). The training data consists of 436 taxi traces over a 5 h period that includes the evening rush hour in order to ensure high taxi usage. We reserve the remaining 100 taxi traces to use as test data. Figure 1 shows the predicted mean and variance values given by the GP regression using the learned hyperparameter values.

We use an empirical approach to learn the target survival ($p_s(\mathbf{x})$) and birth processes ($b(\mathbf{x})$). We overlay a uniform grid (1 m resolution) over the environment. Whenever a

target appears in a cell, we add one to the survival count if the target was previously in another cell, add one to the birth count if the target was previously outside the environment, and add one to the death count if at the next time step the target leaves the environment. The birth count for each cell is initialized to 10, so that the distribution of the birth locations is uniform if there is no data. Similarly, the survival and death count for each cell are initialized to nine and one, respectively. The survival probability in a cell is given by the ratio of the survival count to the total survival and death counts in that cell. In the absence of data, this yields a uniform probability of survival of 0.9.

Figure 2(a) shows the environments used in the simulations, with the target survival probability in Figure 2(b) and birth PHD in Figure 3. As Figure 2(b) shows, the targets survive with high probability in the majority of the environment. The probability decreases near the western and southern edges of the environment, where there are roads along the edge of the environment. These same areas also have the highest rates of target births, as Figure 3(b) shows. One may also clearly see the highways in the southeast and the bridge in the northeast, which have the highest rates of traffic, and thus of target births and deaths. The target birth rate per minute, when considering all 536 taxis in the dataset, is 4.548 targets per minute of real time. The actual and fit birth rates are shown in Figure 3(a), with the Poisson approximation fitting the data well.

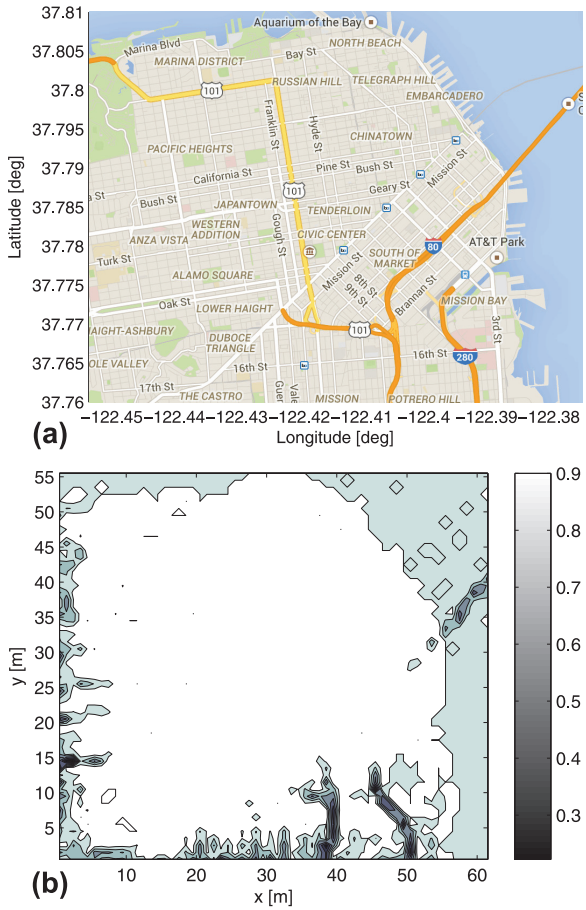

Fig. 2. (a) The area of interest, a roughly 6.15 × 5.56 km region surrounding downtown San Francisco. (b) The probability of target survival as a function of position.

3.3. PHD Filter

The motion model in the PHD filter, $f(x \mid \xi)$ from equation (4), is fully characterized by the GP learned in the previous section. Let $\mathcal{N}(\mathbf{x} \mid \mu, \sigma^2)$ be a Gaussian distribution with mean μ and covariance σ^2 evaluated at $\mathbf{x}$. Then $f(\mathbf{x} \mid \xi) = \mathcal{N}((\mathbf{x} - \xi)\,\Delta_t^{-1} \mid \mu(\xi), \sigma^2(\xi))$, where Δ_t is the time step between measurements and μ and σ are the mean and covariance functions of the GP.

We utilize the sequential Monte Carlo (SMC) PHD filter from the work by Vo et al. (2005). This approximates the PHD using a set of weighted particles, $v(x) \approx \sum_{i=1}^{P_t} w_i\, \delta(\mathbf{x} - \mathbf{x}_i)$. The SMC PHD filter allows for arbitrary, non-linear sensor and motion models, including a finite FoV for the sensor. New particles are added to the PHD using the birth PHD described above as well as using the most recent measurement set (Ristic et al., 2010). A fixed number of particles, P_b, are drawn from the birth PHD and an additional P_m particles are drawn from the inverse measurement model for each measurement in the most recent set, Z_t. The weight of each of these particles is

$$w = \frac{\int c(z)\, dz}{P_b + |Z_t| P_m}$$

where $|Z_t|$ is the cardinality of the measurement set. We resample the particles every five time steps using the low variance resampling method from Thrun et al. (2005) as it has low-complexity and offers a more systematic coverage of the sample space. We use a maximum of 20,000 particles, which is over 200 particles per true target, and set $P_b = 500$ and $P_m = 100$.

3.4. Control policy

In this section, we first present the method we use to generate candidate trajectories for each robot. We then present our control policy for assigning trajectories for the robots and two objective functions for the control policy.

3.4.1. Trajectory generation and assignment. As previously mentioned, in this paper we are considering fixed-wing aircraft. We use a simple model for the aircraft with three basic control inputs: forward velocity, yaw rate, and pitch rate. For each control input we select a range of possible values. For each possible set of control inputs we integrate the position, yaw, and pitch forward in time using a one-step Euler integration scheme for TD seconds, where T is the number of steps and $D = 5$ s is the duration of each step. Note that while Euler integration does accumulate error, the trajectories are over a relatively short time scale, and the integration time step is very small (10 ms).

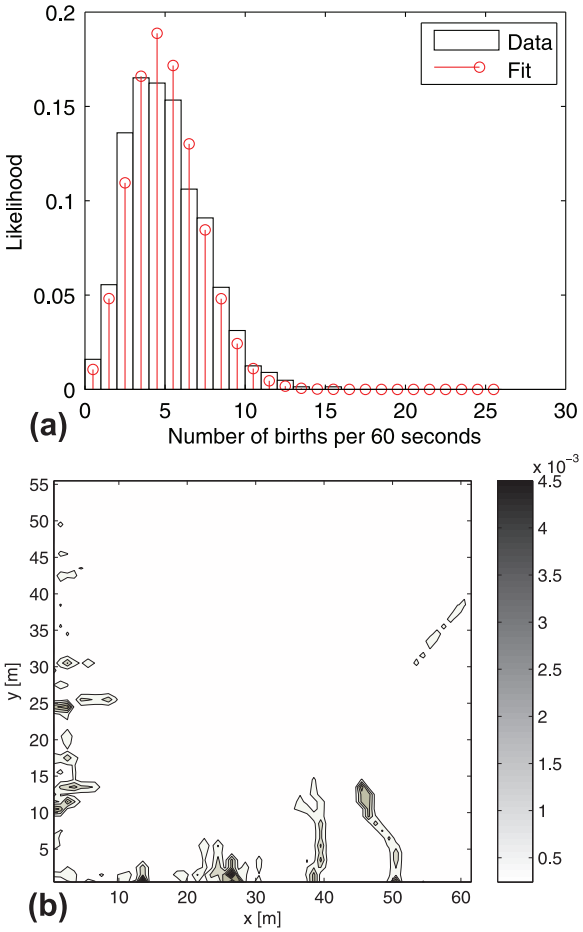

Fig. 3. The empirical target birth PHD. (a) Birth rate and (b) Birth PDF.

Additionally, the robots replan before completing any given trajectory and are assumed to have perfect self-localization. Any trajectories that bring the robots above or below the elevation limits are discarded as invalid, as are any that result in collision. The valid trajectories are interpolated to yield the sequence of N_s poses at which each robot will take a measurement. Each such sequence is an action q^r and let Q^r be the set of all candidate actions for robot r. Figure 4 shows an example set of trajectories for a single robot.

We utilize a greedy approximation strategy, outlined in Algorithm 1, to assign trajectories to each robot. This is similar to the approach used by Tokekar et al. (2014). Initially, each robot computes the utility of each action $q^r \in Q^r$ according to some desired objective function. The robot and action with the highest utility are selected. The remainder of the team then recomputes the utility of each action conditioned on the action of the first robot, discarding any trajectories that lead to collisions, and the robot and action with the highest utility are again selected. This process repeats until all robots have been assigned an action. Let $Q = \{q^1, q^2, \ldots, q^R\}$ be a candidate set of trajectories for the team of R robots. For any objective $f(Q)$ that is a submodular set function of Q, we can show that this greedy assignment policy is a two-approximation.

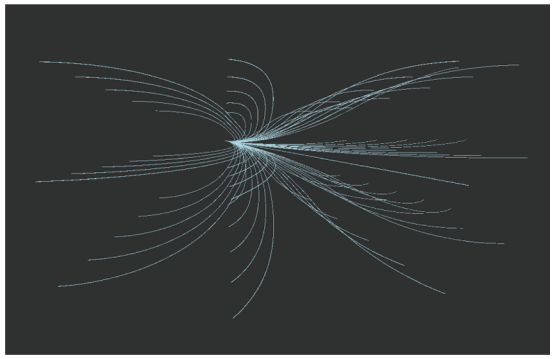

Fig. 4. A sample set of unmanned aerial vehicle (UAV) trajectories.

Algorithm 1 Greedy trajectory assignment.

Inputs: $f(\cdot)$ ▷ Objective function
$Q^1, \ldots, Q^R$ ▷ Candidate action sets
$Q^G = \emptyset$ ▷ Greedy assignments
$A = \{1, \ldots, R\}$ ▷ Agents left to plan for
while $A \neq \emptyset$ **do**
  $Q = \cup_{r \in A} Q^r$
  $q^*, r^* = \text{argmax}_{q^r \in Q} f(Q^G \cup \{q^r\})$
  $Q^G \leftarrow Q^G \cup \{q^*\}$ ▷ Assign best action q^* to r^*
  $A \leftarrow A \setminus \{r^*\}$ ▷ Remove robot r^* from list
end while
return Q^G

Theorem 1 *Let Q^G be the robot poses selected by the greedy assignment policy in Algorithm 1 and Q^* be the robot actions selected by the full, joint evaluation of*

$$Q^* = \text{argmax}_{Q \in Q^{1:R}} f(Q) \tag{9}$$

where $Q^{1:R} = Q^1 \times \cdots \times Q^R$. Then greedy is a two-approximation, i.e. $f(Q^G) \geq \frac{1}{2} f(Q^*)$.

Proof. It is known that the greedy algorithm yields a two-approximation for maximizing a monotone, submodular function subjected to a partition matroid constraint (Calinescu et al., 2007). We can create a set system using the candidate robot actions, as shown in the work by Tokekar et al. (2014). This set system defines a partition matroid, which along with the previous lemma proves the desired result. □

Atanasov et al. (2015) recently proved the same bound holds for the centralized and the decentralized case. This result significantly speeds up the control computations as the complexity of Algorithm 1 scales linearly with the number of robots while equation (9) scales exponentially.

3.4.2. Mutual information objective. Mutual information (MI) is a way of quantifying the dependence between two random variables (Cover and Thomas, 2012), and can be

defined in multiple ways

$$I[\mathcal{X}, \mathcal{Z}] = \int p(X,Z) \log \frac{p(X,Z)}{p(X)p(Z)}\, dXdZ \tag{10}$$

$$= \int \mathrm{KL}\big[p(X \mid Z) \,||p(X)\big]\, p(Z)\, dZ \tag{11}$$

The last term above states that mutual information can be interpreted as the expected Kullback–Leibler divergence between the prior and the posterior, given the unknown future measurements. Thus, maximizing the mutual information between the target set and the future measurements of the robots will cause the robots to take measurements that will change their belief quickly.

We utilize binary measurements, rather than the full measurement sets, in order to decrease the computational complexity of the control policy. This allows us to derive a closed-form expression, and we have previously shown that this approach effectively drives a team of robots to detect and localize static targets (Dames and Kumar, 2015). The binary measurements are defined to be $y = \mathbf{1}(Z \neq \emptyset)$, where $\mathbf{1}(\cdot)$ is the indicator function. Here, $y = 0$ is the event that the robot receives no measurements to any (true or clutter) objects, while $y = 1$ is the complement of this, i.e. the robot receives at least one measurement. Kreucher et al. (2005) take a similar approach, using a binary sensor model and an information-based objective function to schedule sensors to track an unknown number of targets.

Theorem 2 *The mutual information between the target set and the binary measurement model is a lower bound on the mutual information between the target set and the full measurement set, i.e.* $I[\mathcal{X}; \mathcal{Y} \mid Q] \leq I[\mathcal{X}; \mathcal{Z} \mid Q]$.

Proof. Note that y is deterministically related to Z, $y = \mathbf{1}(Z \neq \emptyset)$. This allows us to apply the data processing inequality (Cover and Thomas, 2012, Theorem 2.8.1), which states that functions of the data cannot increase the amount of information. □

The optimal strategy is then to choose robot trajectories that maximize the mutual information between the target set and its future measurements

$$Q_\tau^* = \mathrm{argmax}_{Q_\tau \in Q_\tau^{1:R}} I[\mathcal{X}_{t+N_s}; \mathcal{Y}_\tau^{1:R} \mid Q_\tau] \tag{12}$$

where $\tau = \{t+1, \ldots, t+T\}$ is the time horizon, $\mathcal{X}_{t+T}$ is the predicted location of the targets at time $t + N_s$, $\mathcal{Y}_\tau^{1:R}$ is the collection of binary measurements for robots 1 to R from time steps $t+1$ to $t+T$, and Q_τ are the future poses of the robots. These measurements depend on the future locations of the robots $Q_\tau = \{q_{t+1}^1, \ldots, q_{t+T}^1, \ldots, q_{t+T}^R\}$.

We can can show that the objective function in equation (12) is a submodular set function of the robot poses, which allows us to use the greedy algorithm to select actions that approximately maximize the mutual information.

Lemma 1 $I[\mathcal{X}; \mathcal{Y} \mid Q]$ *is a submodular set function of* Q.

Proof. See the work by Krause and Guestrin (2005, Proposition 2). □

3.4.3. Expected number of detections objective. We introduce the expected number of detections (END) objective function to be

$$N[X \mid Q] = \int \left(1 - \prod_{q \in Q} \big(1 - p_d(x \mid q)\big)\right) v(x)\, dx \tag{13}$$

This objective is similar to the posterior expected number of targets (PENT) objective from the work by Mahler (2003b), but is simpler to compute as it does not involve propagating the target state forward in time. The END objective gives the expected number of targets detected by at least one robot and will increase as the robots' sensor footprints cover high target-density regions and as the robots move towards areas where they are more likely to detect targets. We can show that the END objective is a submodular set function of Q so the greedy assignment algorithm will be a two-approximation, similar to the previous theorem.

Lemma 2 *The END objective function,* $N[X \mid Q]$*, is a submodular function of* Q.

Proof. The difference in the objective when adding a single robot is

$$N[X; Q \cup \{q'\}] - N[X; Q] = \int p_d(x \mid q') \prod_{q \in Q} (1 - p_d(x \mid q))\, v(x)\, dx$$

For any $R \subseteq Q$, the product $\prod_{r \in R}(1 - p_d(x \mid r)) \geq \prod_{q \in Q}(1 - p_d(x \mid q))$ since $p_d(x \mid q) \in [0,1], \forall x, q$. Thus $N[X; R \cup \{q'\}] - N[X; R] \geq N[X; Q \cup \{q'\}] - N[X; Q]$, so by definition $N[X, Q]$ is submodular. □

3.4.4. Coverage based control. The last control method that we consider in this work is a coverage-based controller where the robots follow a pre-determined pattern. We use a simple lawnmower pattern at a fixed elevation to ensure that each part of the environment receives equal coverage as the robots explore. When there is more than one robot, the robots are evenly spaced along the trajectory. This will provide a baseline to which we can compare the other two objective functions which we will refer to as the lawnmower (LM) controller.

4. Results

To validate the performance of our system we ran a series of simulated experiments, varying the motion model of the targets and the speed of the targets relative to the speed of the robots. We tested each configuration with different team sizes ($R = 2, 4, 6$), planning horizons, and control objectives. For the planning horizon we examined two scenarios, one with a fixed number of steps ($T = 1, 2$) and one where the total number of measurements per plan was held constant ($RT = 12$). We performed five trials with

each configuration, randomly selecting a subset of 80 targets to use for the ground truth target motion from the 100 taxi traces originally reserved for testing. The true number of targets in the area of interest varies over time as targets enter and leave. The robots monitor the area from Figure 2(a), scaled down by a factor of 100. We also sped up the data by a factor of 30 in order to speed up the simulations, so 1 s in simulation represents 30 s of real time. The data is taken from 4–9 pm on 18 May 2008, a time of day where there was plenty of taxi traffic.

It is worth noting that a number of competing multi-target tracking methods exist (Stone et al., 2013), most notably the multiple hypothesis tracker (MHT) and joint probabilistic data association (JPDA). However, to the best of our knowledge, no active multi-robot control policies exist based on these estimation algorithms. This makes comparisons to these methods beyond the scope of this paper.

One common measure of uncertainty in the target estimate is entropy (Cover and Thomas, 2012), defined as

$$H[X] = \int p(X) \log p(X)\ \delta X \tag{14}$$

$$= \lambda + \lambda\big(\underbrace{H[p(x)]}_{\text{entropy per target}} - \log\lambda\big) \tag{15}$$

where $\lambda = \int v(x)\ dx$ is the expected number of targets in the environment and $p(x) = \lambda^{-1} v(x)$ is the probability distribution for individual targets. Note that equation (15) results from plugging equation (3) into the definition of entropy given by equation (14). See the work by Dames and Kumar (2015) for a derivation of this expression. We will use entropy to measure uncertainty, and the term in parentheses in equation (15) is the entropy per target.

4.1. Motion models

The two target motion models that we consider are the GP described in the previous section and a Gaussian random walk (GRW) model. The GRW models the targets as performing a random walk, with a velocity drawn at random from a Gaussian distribution.

Note that these models are used only to update the PHD; the actual trajectories of the targets are given by the taxi dataset. In both cases we use the survival and birth processes described in the previous section, with the birth rate set to 0.6788 to account for the reduced number of data files used (80 for testing versus 436 for training). The average velocity of the robots and targets is approximately equal.

Figure 5 shows the ratio of the expected number of targets to the true number of targets, and Figure 6 shows the average entropy per target. In general, the GP model tracks more targets with less uncertainty than the GRW model. The GRW model tracks approximately the same number of targets with the same amount of uncertainty per target regardless of the number of robots or the planning horizon. The GP model tracking improves slightly with the number of robots and the planning horizon, with the largest effect being the increase from a myopic ($T = 1$) controller to a receding horizon ($T > 1$) controller. This validates the effectiveness of our data-driven motion model, which we use exclusively for the remainder of the paper.

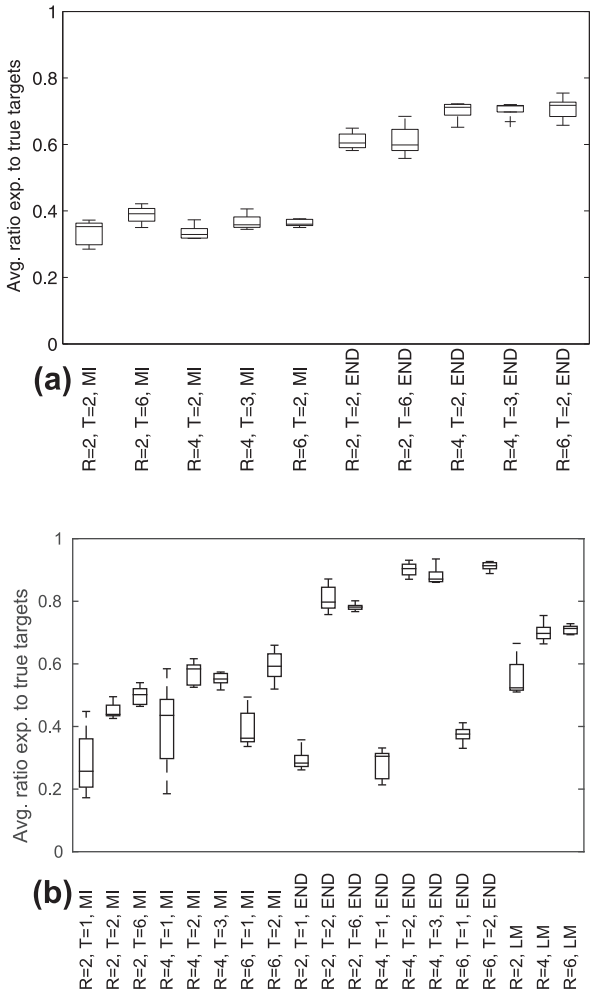

Fig. 5. The average ratio of the expected number to the true number of targets over a single run. (a) Gaussian random walk (GRW) motion model and (b) Gaussian process (GP) motion model.

4.2. Objective functions

As we see in Figures 5(b) and 6(b), the END objective seems to have the best performance out of our three objectives. Robots using END track the highest fraction of targets with the lowest uncertainty per target. Robots using MI track the fewest targets while robots using LM have the highest uncertainty per target.

We also see that robots using either MI or END benefit greatly from increasing the planning horizon from $T = 1$ to $T = 2$. This effect is particularly strong with the END objective, where the robots' performance goes from being slightly worse than that of MI to significantly better. The addition of more robots has the next largest effect on the

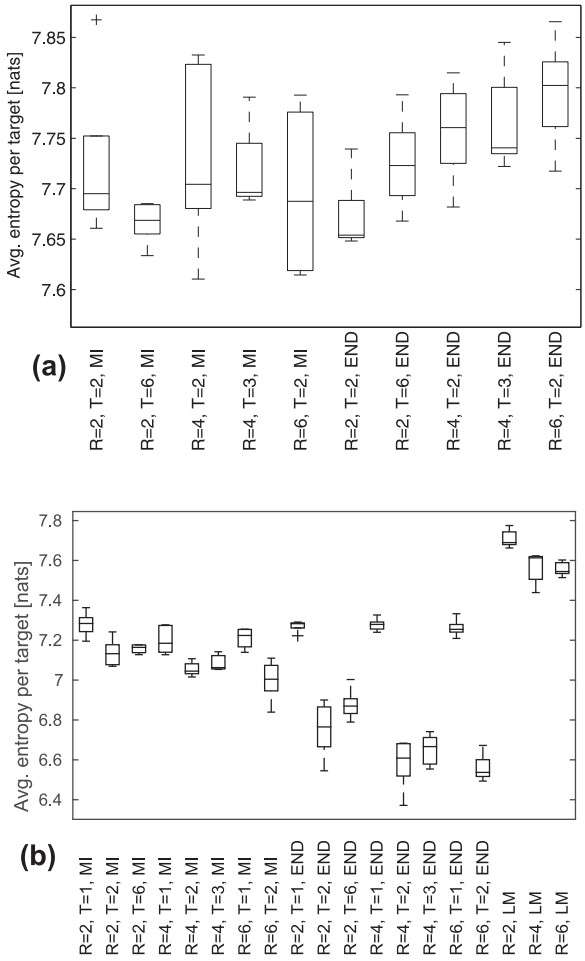

Fig. 6. The average entropy per target over a single run. (a) Gaussian random walk (GRW) motion model and (b) Gaussian process (GP) motion model.

team's tracking performance. Increasing the planning horizon beyond two makes the performance more consistent across trials but only leads to modest gains in tracking quality.

Figures 7 to 9 show how the ratio of the expected number of targets to true targets, the robot elevation, and the target set entropy evolve during the course of a single run. Extension 1 shows a video of the full simulation runs of these three trials. These are representative trials of a team of $R = 2$ robots with a planning horizon of $T = 6$ time steps. Overall robots using the END objective function seem to better estimate the number of targets, quickly reaching the correct number and clearly following the trend. However, the END objective has larger jumps in the target count estimate and the target entropy values than MI or LM. MI maintains a nearly constant value of entropy per target across the experiment, which makes sense as this is the only objective function to explicitly consider entropy. The LM objective is closer to the MI objective in terms of performance, though it does not follow the trend in the number of targets as well as MI and has a higher entropy per target.

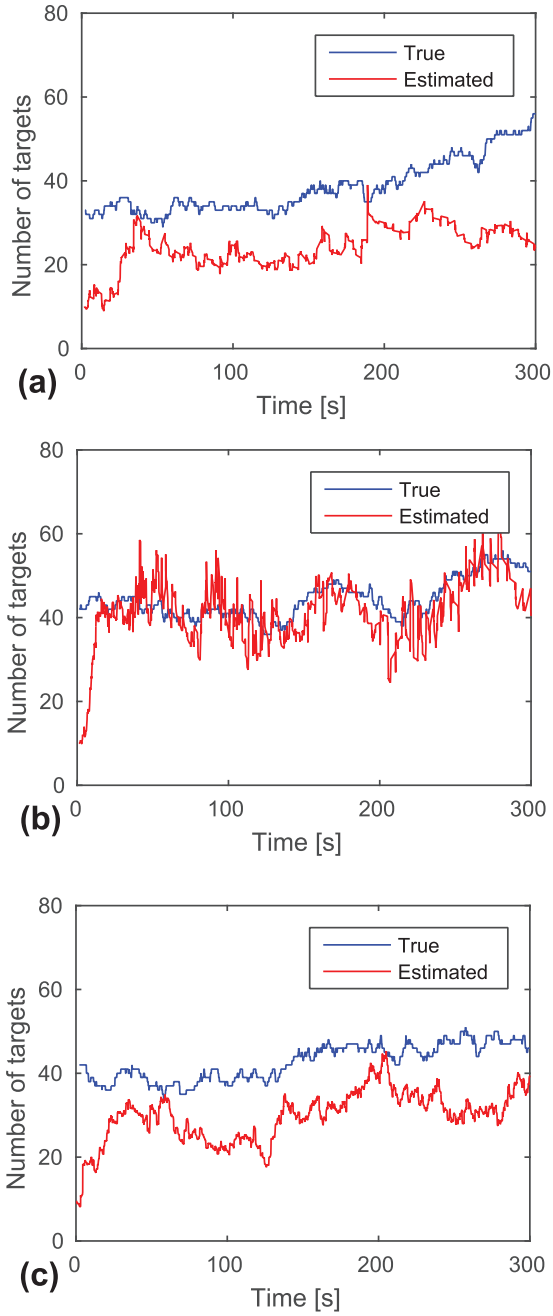

Fig. 7. The ratio of the expected number to the true number of targets over a single run for $R = 2$ and $T = 6$. (a) GP motion model with MI objective, (b) GP motion model with END objective, and (c) GP motion model with LM objective.

The emergent behavior of the different control objectives is also quite different. Robots using MI tend to stay closer to the ground in order to decrease uncertainty in the location of individual targets. On the other hand, robots using END fly at a higher altitude, though robots never reach the maximum altitude limit of 50 m. Note that increasing the altitude decreases the probability of detection in equation (6), while increasing the size of the sensor FoV. These results indicate that MI and END make different tradeoffs in terms of the

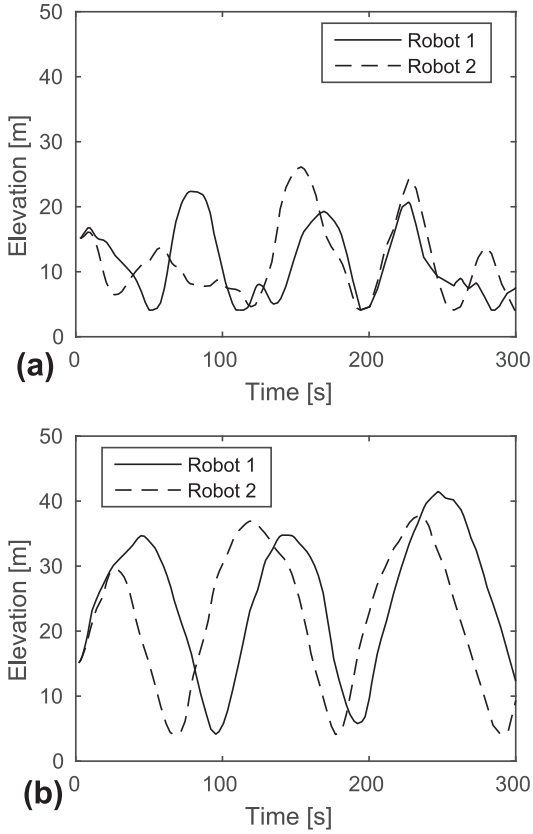

Fig. 8. The elevation of the robots over a single run for $R = 2$ and $T = 6$. (a) GP motion model with MI objective and (b) GP motion model with END objective.

size of the FoV versus the probability of detection. These differences are consistent across team sizes and planning horizons, as Figure 10 shows, with robots using END having a higher average altitude. This trend also explains why robots using END keep a larger fraction of the targets within their collective FoV, as Figure 11 shows. We selected the constant elevation for LM to be 15 m to be consistent with the elevation for MI and the myopic END situations.

4.3. Target speed

In all of the previous trials, the robot and target speeds were approximately equal, with both having an average speed of 1.5 m/s in the scaled down simulation environment. To test the performance of our system as the speeds varied we ran a series of simulations where the targets moved at 10%, 25%, and 50% of the robots' speed.

Figure 12 shows the average ratio of tracked targets and Figure 13 shows the average entropy per target for each configuration tested. We again see that the END objective tracks a higher fraction of the targets with lower uncertainty, though the gain over MI decreases as the target speed decreases. The performance of robots using LM is fairly consistent across the robot speeds. This is natural, as the controller is not reactive to the current environment and has

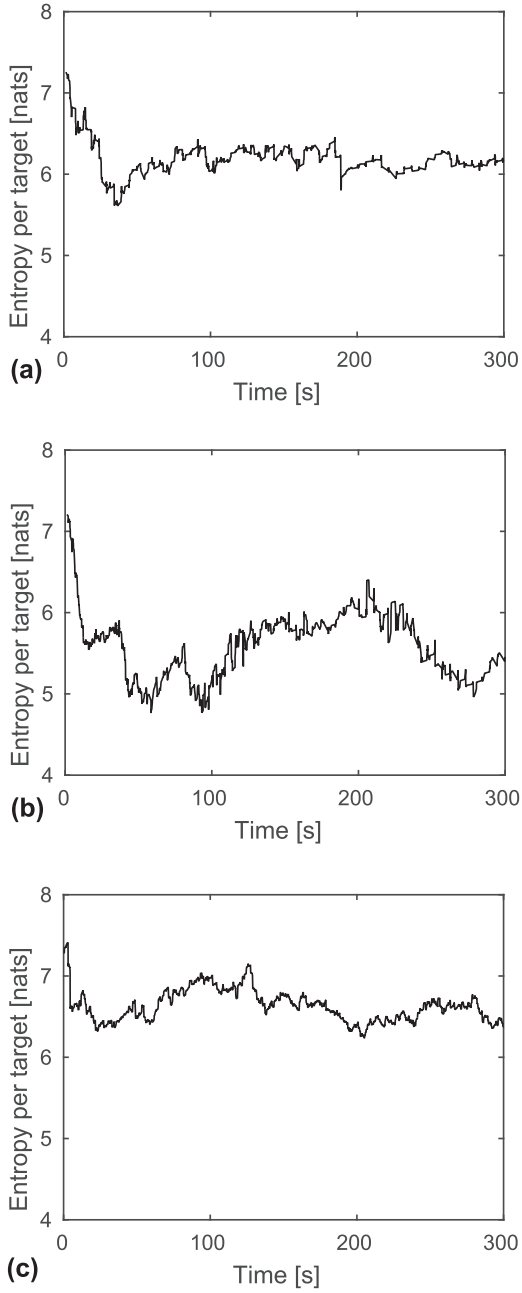

Fig. 9. The entropy per target of the target set over a single run for $R = 2$ and $T = 6$. (a) GP motion model with MI objective, (b) GP motion model with END objective, and (c) GP motion model with LM objective.

no model of the target motion or sensor. The only exception to this is the decrease in entropy per target from around 8.5 nats at 25% speed to 7.2 nats at 10% speed.

Finally, we test the performance of our framework with static targets using a team of $R = 4$ robots with a planning horizon of $T = 3$. The simulation parameters are identical except that we replace the 80 taxi data traces with 80 randomly drawn static target locations, and the target motion model is the identity map. The resulting final estimated number of targets and target entropies are shown in

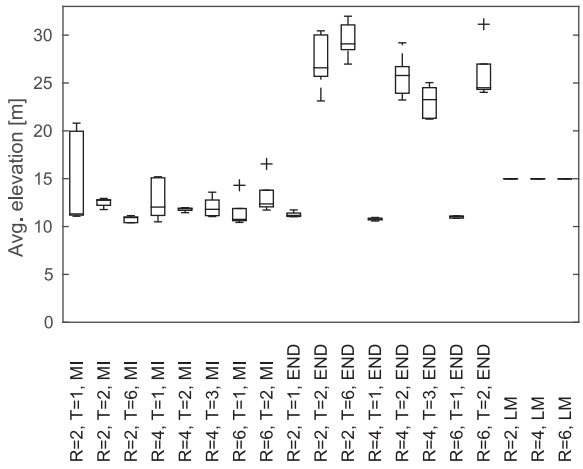

Fig. 10. The average elevation of the team over a single run.

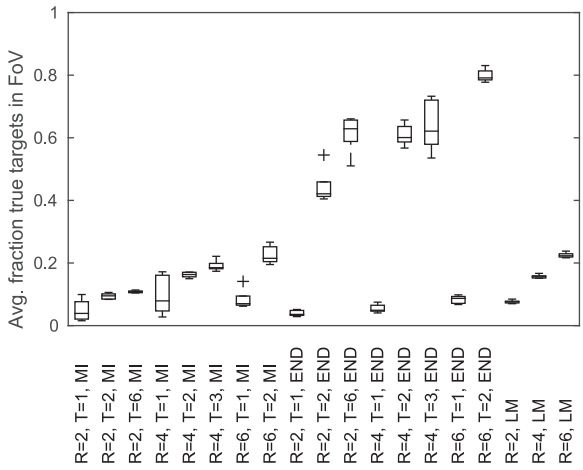

Fig. 11. The average fraction of the number of targets within the team's FoV over a single run.

Figure 14. The final ratio of the estimated number of targets to the true number of targets is very close to 1 using both objective functions, indicating that the system is able to correctly and consistently determine the number of targets. The entropy is also orders of magnitude lower than in the case of moving targets, with MI reaching a lower final entropy than END.

5. Conclusions

In this paper, we describe a framework for detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots. We create a data-driven motion model for the targets using GP regression and also create position-dependent models of the target appearance and disappearance statistics. The robot team uses the PHD filter to simultaneously estimate the number of targets and the states of the targets. The PHD filter is robust to false negative and false positive detections and sensor noise and does not require any explicit data association. Using the estimate of the target set from the PHD filter, the robots

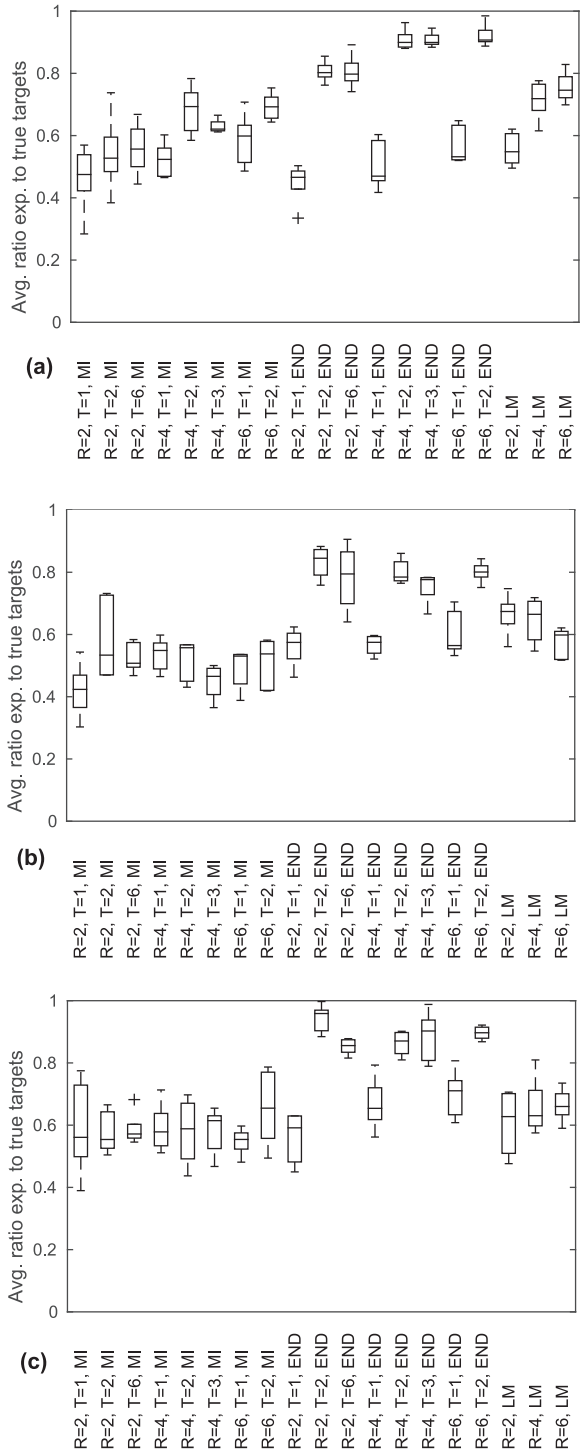

Fig. 12. The average ratio of the expected number to the true number of targets over a single run. (a) Targets move at 50% of robot speed, (b) targets move at 25% of robot speed, and (c) targets move at 10% of robot speed.

greedily select actions that maximize submodular control objectives. The three control objectives that we consider in this paper are the expected number of detections (END) targets by the team, the mutual information (MI) between the predicted targets and the future detections of the robots,

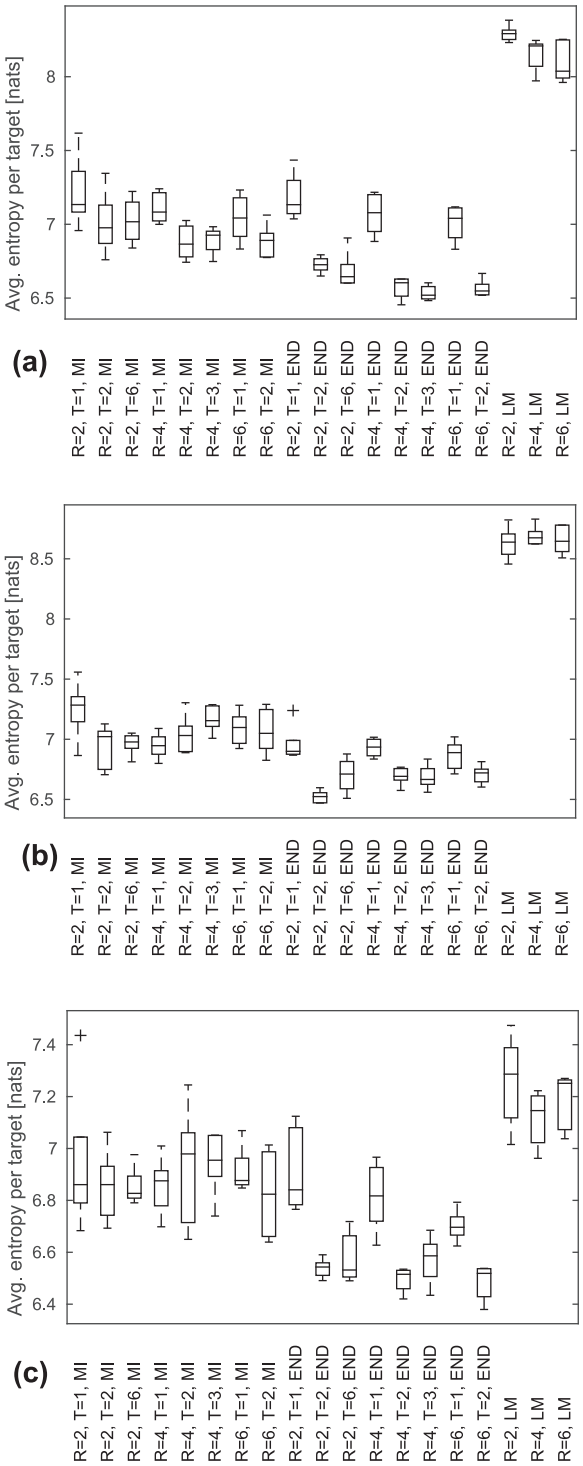

Fig. 13. The average entropy per target over a single run. (a) Targets move at 50% of robot speed, (b) targets move at 25% of robot speed, and (c) targets move at 10% of robot speed.

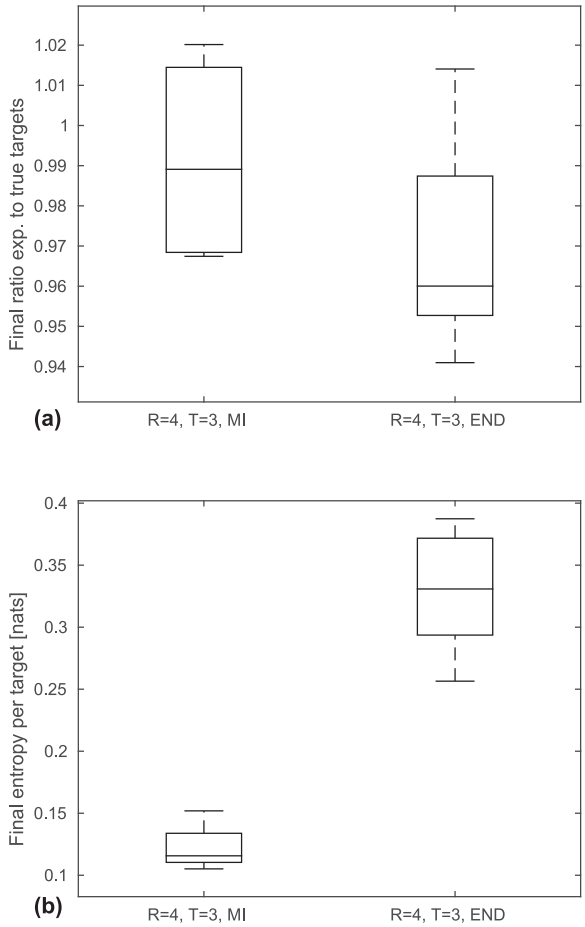

Fig. 14. The performance of our framework with static targets. (a) Final ratio of the expected number of targets to the true number of targets and (b) final entropy per target of the estimated target set.

and a coverage-based controller where the robots follow a lawnmower pattern (LM).

We validate our framework through extensive simulations using a real-world dataset for target motion. We find that robots using our data-driven motion model track more targets and have less uncertainty than robots using a GRW model. We also find that robot teams using our new END objective track a higher fraction of the targets with higher precision than teams using mutual information or a coverage-based controller. This trend persists as we vary the speed of the targets and consider different target motion models. The END and MI objectives both make tradeoffs between the size of the sensor FoV and the probability of detection, with END favoring a larger FoV and MI favoring a higher probability of detection.

Future work will aim to replicate these results on a team of real-world robots. We will also explore the effects of changing the robot and target kinematics. We hypothesize that a team of multi-rotor robots would be able to track a larger number of targets due to the fact that they can hover, unlike the fixed-wing aircraft model used in this paper, so long as the targets do not move significantly faster than the robots. We also believe that a team of cars would track significantly fewer robots with less uncertainty due to the much smaller sensor footprint. Further work will also consider using a heterogeneous team of robots to take advantage of each type of robot motion while mitigating the drawbacks.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was funded by ONR (MURI grant numbers N00014-07-1-0829, N00014-09-1-1051, and N00014-09-1-1031), ARO (grant number W911NF-13-1-0350), NSF (grant number IIS-1426840), and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. P Tokekar was supported in part by NSF (grant number IIS-1566247).

References

Adams M, Vo BN and Mahler R (eds) (2014) Advances in probabilistic modeling: Applications of stochastic geometry. *IEEE Robotics and Automation Magazine* 21(2).

Atanasov N, Le Ny J, Daniilidis K, et al. (2015) Decentralized active information acquisition: Theory and application to multi-robot slam. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Seattle, WA, USA, 26–30 May 2015.

Atanasov N, Zhu M, Daniilidis K, et al. (2016) Localization from semantic observations via the matrix permanent. *International Journal of Robotics Research* 35(1–3): 73–99.

Calinescu G, Chekuri C, Pál M, et al. (2007) Maximizing a submodular set function subject to a matroid constraint. In: *Integer Programming and Combinatorial Optimization*. Berlin, Heidelberg: Springer, pp.182–196.

Charrow B, Michael N and Kumar V (2015) Active control strategies for discovering and localizing devices with range-only sensors. In: *Algorithmic foundation of robotics XI*, pp. 55–71. Springer International Publishing.

Chung TH, Hollinger GA and Isler V (2011) Search and pursuit-evasion in mobile robotics. *Autonomous Robots* 31(4): 299–316.

Cover T and Thomas J (2012) *Elements of Information Theory*. Hoboken, NJ: John Wiley & Sons.

Dames P and Kumar V (2015) Autonomous localization of an unknown number of targets without data association using teams of mobile sensors. *IEEE Transactions on Automation Science and Engineering* 12(3): 850–864.

Dames P, Tokekar P and Kumar V (2015) Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots. In: *Proceedings of the international symposium on robotics research*, Sestri Levante, Italy, 12–15 September 2015. Available at: https://www.springer.com/us/book/9783319515311.

Frew EW and Rock SM (2003) Trajectory generation for constant velocity target motion estimation using monocular vision. In: *Proceedings of the IEEE international conference on robotics and automation*, Taipei, Taiwan, 14–19 September 2003, pp.3479–3484.

Furukawa T, Bourgault F, Lavis B, et al. (2006) Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets. In: *Proceedings of the IEEE international conference on robotics and automation*, Orlando, FL, USA, 15–19 May 2006, pp.2521–2526.

Gans NR, Hu G, Nagarajan K, et al. (2011) Keeping multiple moving targets in the field of view of a mobile camera. *IEEE Transactions on Robotics* 27(4): 822–828.

Grabner H, Nguyen TT, Gruber B, et al. (2008) On-line boosting-based car detection from aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing* 63(3): 382–396.

Grocholsky B, Keller J, Kumar V, et al. (2006) Cooperative air and ground surveillance. *IEEE Robotics and Automation Magazine* 13(3): 16–25.

Hollinger GA, Djugash J and Singh S (2012) Target tracking without line of sight using range from radio. *Autonomous Robots* 32(1): 1–14.

Joseph J, Doshi-Velez F, Huang AS, et al. (2011) A Bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots* 31(4): 383–400.

Kim CY, Song D, Xu Y, et al. (2014) Cooperative search of multiple unknown transient radio sources using multiple paired mobile robots. *IEEE Transactions on Robotics* 30(5): 1161–1173.

Kotz D and Henderson T (2005) CRAWDAD: A community resource for archiving wireless data at Dartmouth. *IEEE Pervasive Computing* 4(4): 12–14.

Krause A and Guestrin CE (2005) Near-optimal sensor placements in Gaussian processes. In: *Proceedings of the international conference on machine learning*, Bonn, Germany, 7–11 August 2005.

Kreucher C, Kastella K and Hero III AO (2005) Sensor management using an active sensing approach. *Signal Processing* 85(3): 607–624.

Leung K, Inostroza F and Adams M (2014) Evaluating set measurement likelihoods in random-finite-set SLAM. In: *IEEE international conference on information fusion*, Salamanca, Spain, 7–10 July 2014, pp.1–8.

Li X, Lu R, Liang X, et al. (2011) Smart community: An internet of things application. *IEEE Communications Magazine* 49(11): 68–75.

Li XR and Jilkov VP (2003) Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems* 39(4): 1333–1364.

Li XR and Jilkov VP (2005) Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems* 41(4): 1255–1321.

Logothetis A, Isaksson A and Evans RJ (1997) An information theoretic approach to observer path design for bearings-only tracking. In: *IEEE international conference on decision and control*, San Diego, CA, USA, 10–12 December 1997, vol.4, pp.3132–3137.

Mahler R (2003a) Multitarget bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems* 39(4): 1152–1178.

Mahler R (2003b) Objective functions for Bayesian control-theoretic sensor management, 1: Multitarget first-moment approximation. In: *IEEE aerospace conference proceedings*, vol.4. Big Sky, MT, USA, 8–15 March 2003.

Mahler R (2007) *Statistical Multisource–Multitarget Information Fusion*, Boston: Artech House.

Piorkowski M, Sarafijanovic-Djukic N and Grossglauser M (2009) CRAWDAD data set epfl/mobility (v. 2009-02-24). Downloaded from http://crawdad.org/epfl/mobility/.

Rasmussen C and Williams C (2006) *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press.

Reuter S, Vo BT, Vo BN, et al. (2014) The labeled multi-Bernoulli filter. *IEEE Transactions on Signal Processing* 62(12): 3246–3260.

Ristic B, Clark D and Vo BN (2010) Improved SMC implementation of the PHD filter. In: *IEEE International Conference on Information Fusion*. IEEE, pp.1–8.

Ristic B, Vo BN and Clark D (2011) A note on the reward function for PHD filters with sensor control. *IEEE Transactions on Aerospace and Electronic Systems* 47(2): 1521–1529.

Song D, Kim CY and Yi J (2012) Simultaneous localization of multiple unknown and transient radio sources using a mobile robot. *IEEE Transactions on Robotics* 28(3): 668–680.

Spletzer JR and Taylor CJ (2003) Dynamic sensor planning and control for optimally tracking targets. *International Journal of Robotics Research* 22(1): 7–20.

Stone LD, Streit RL, Corwin TL, et al. (2013) *Bayesian Multiple Target Tracking*. Boston, MA, USA: Artech House.

Thrun S, Burgard W and Fox D (2005) *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press.

Tokekar P, Branson E, Vander Hook J, et al. (2013) Tracking aquatic invaders: Autonomous robots for monitoring invasive fish. *IEEE Robotics and Automation Magazine* 20(3): 33–41.

Tokekar P, Isler V and Franchi A (2014) Multi-target visual tracking with aerial robots. In: *IEEE/RSJ international conference on intelligent robots and systems*, Chicago, IL, USA, 14–18 September 2014, pp.3067–3072.

Vo BN, Singh S and Doucet A (2005) Sequential Monte Carlo methods for multi-target filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems* 41(4): 1224–1245.

Xu Z, Fitch R, Underwood J, et al. (2013) Decentralized coordinated tracking with mixed discrete–continuous decisions. *Journal of Field Robotics* 30(5): 717–740.

Zhao T and Nevatia R (2003) Car detection in low resolution aerial images. *Image and Vision Computing* 21(8): 693–703.

Zhou K and Roumeliotis SI (2008) Optimal motion strategies for range-only constrained multisensor target tracking. *IEEE Transactions on Robotics* 24(5): 1168–1185.

Zhou K and Roumeliotis SI (2011) Multirobot active target tracking with combinations of relative observations. *IEEE Transactions on Robotics* 27(4): 678–695.

Appendix

Index to multimedia extension

Archives of IJRR multimedia extensions published prior to 2014 can be found at http://www.ijrr.org, after 2014 all videos are available on the IJRR YouTube channel at http://www.youtube.com/user/ijrrmultimedia

Table of multimedia extension.

Extension	Media type	Description
1	Video	Summarizing the simulation trials from Figures 7 to 9