Finding and Tracking Targets in the Wild: Algorithms and Field Deployments

Volkan Isler, Narges Noori, Patrick Plonski, Alessandro Renzaglia, Pratap Tokekar and Josh Vander Hook

***Abstract*— We describe our efforts on building a robotic system for detecting and tracking radio-tagged invasive fish using teams of autonomous ground and surface vehicles. In addition to system building and field experiments, our efforts clustered around three fundamental problems: (1) Search: how to find the target as quickly as possible, (2) Active localization: how to actively choose measurement locations to accurately estimate target locations, and (3) Long-term autonomy through energy-efficiency and harvesting. We present specific problem formulations and a summary of our results so far. We conclude the paper with a discussion on our progress and next steps.**

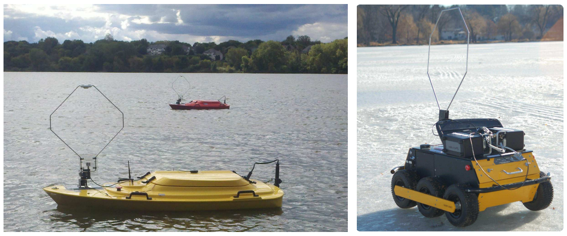


Fig. 1. (Left) Two robotic boats tracking invasive fish in Minnesota lakes. (Right) Rovers used in winter experiments when the lakes freeze over.

I. Introduction

Habitat monitoring is an application domain where robotics can have a huge positive impact. Robots equipped with a wide variety of sensors can monitor large areas for extended periods of time, and provide valuable data for scientists, policy makers and other stake-holders. At the same time, it is an exciting research frontier for robotics. Numerous fundamental robotics problems such as coverage, search and tracking arise as components of habitat monitoring tasks. In order for robots to deliver the promised impact, these tasks must be executed in unstructured and often dynamic settings. For long term operation, issues of robustness and energy management must be addressed.

Since 2010, we have been working on building a robotic sensor network to monitor Common Carp, an invasive fish infesting lakes across the Midwest. Biologists surgically insert radio tags in the fish and manually track them to study their behavior and find aggregations. The tracking process is performed throughout the year by searching for signals emitted from the tags using directional antennas [1]. This is a difficult and labor intensive job, especially in the winter. The primary goal of our project is to build a autonomous system which can track radio-tagged fish. We use autonomous surface vehicles in the summer and switch to ground vehicles in the winter when lakes freeze.

From a robotics perspective, we focus on three sets of problems. The first task is to find the fish. Radio tags have limited ranges which can drop to 10-20 meters depending on the depth of the fish and the strength of the tags' batteries. Therefore, when the robots are launched, they may not hear the signal. How should they move so that they establish contact with the signal as quickly as possible? This is the *search* task. Once the fish are found, they can be localized accurately. In order to carry out this task, the robots rotate their antennas and obtain bearing measurements which are then used to estimate fish locations. The uncertainty in the estimates is a function of the sensor/target geometry. The second task is to choose measurement locations so as to obtain high quality estimates. This *active-localization* task is challenging because the measurement locations must be chosen in an online fashion as new information becomes available. Finally, in the third task, we focus on *energy efficiency*. As a building block for other tasks, we study the problem of moving from point a to point b in an energy efficient fashion. In addition, our robots carry solar panels so the notion of efficiency involves not only energy consumption but also potential gains attained from harvesting.

In this paper, we present an overview of our progress so far. In addition to tackling underlying algorithmic problems, this project challenged us to implement our algorithms and test them in field conditions. We present our main results and report findings from field experiments. We conclude the paper with a reflection of where we are and a discussion on the challenges and opportunities for robotics research in habitat monitoring.

II. System Overview

We execute our carp tracking algorithms on robot boats in the summer and robot rovers in the winter which drive along the frozen surface of the lake. In this section we will present the details of these systems.

Our boats are built on the OceanScience Q-Boat 1800D[1]. The Q-Boat is 1.8m long, with a cruising speed of about 1 m/s and a turning radius of 5m. Our rovers are built on the Clearpath Husky A100 and A200[2], which are both rugged and heavy skid-steer vehicles. The A100 is 0.860m long by 0.605m wide with a datasheet mass of 35kg and a maximum payload of 40kg. The A200 is 0.990m long by 0.670m wide

The authors were at the University of Minnesota while this work was performed. 200 Union St. SE KHKH 4-192, Minneapolis, MN, 55455 USA, Emails: {isler, noori, plonski, arenzagl, tokekar, jvander}@cs.umn.edu

[1] www.oceanscience.com

[2] www.clearpathrobotics.com

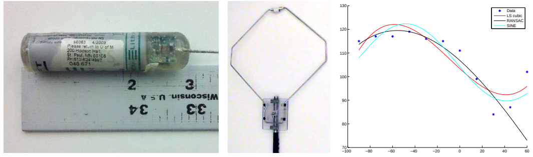


Fig. 2. (Left) The tag which is implanted in fish. The tag is approximately 5 cm in length, with a 30cm antenna trailing off (2 inches, 12 inches). (Middle) The antenna is approximately 61 cm in diameter (24 inches). The tags transmit an uncoded radio pulse once per second in the 47.9 MHz range. The antenna is direction-sensitive, meaning the perceived strength of the signal is dependent on the orientation of the antenna relative to the direction to the tag. In this way, it is possible to estimate the bearing to the transmitting tag. (Right) A coarse sampling of the signal strength. The horizontal axis is the pan angle and the vertical is the measured signal strength. Least squares estimation of a sine curve typically provides the best estimates.

with a datasheet mass of 50kg and a maximum payload of 75kg. Both rovers can travel at 1 m/s.

The rovers are directly controlled with a serial port connection from a laptop computer. The boats are controlled with Arduino Mega2560 microcontroller boards that communicate with the motor controllers, servos, and sensors, and are in turn commanded by laptop computers over direct Ethernet using UDP. Self-localization is achieved through the use of a GPS unit and a compass, filtered with an Extended Kalman Filter. On the rovers, wheel odometry is used as well. All software is compatible with the Robot Operating System.

Our boats can be configured with a forward-looking, single-beam sonar unit that is used for obstacle avoidance. The sonar unit is an Imagenex 852 Digital Echo Sounder[3], with 10 degree beam width and maximum range of 50m. Navigating to a destination while avoiding obstacles is nontrivial with our sensor, because each sonar reading provides only a small glimpse of the map of the environment. In [2], we addressed this problem in an online competitive framework. The competitive ratio between the online strategy and the optimal offline strategy is shown to be constant when the obstacle is a rectangle perpendicular to the desired direction of travel, or logarithmic in the width of the obstacle if the shape is unknown. The algorithm is validated with real experiments in which the boat successfully navigates around a large obstacle.

The boats and rovers can also be equipped with solar panels to increase their system lifetime. The resulting system must have an ongoing estimate of expected solar power over the working environment, and include this information as part of a global path planner, as shown in Section V.

III. Search

The first phase of the carp monitoring task is to detect the fish as quickly as possible. A searcher (either the boat or a rover) detects a target if the target is within the detection range of the antenna. The detection range varies from $10m$ to $100m$ depending on the depth of the fish and the orientation of the antenna with respect to the fish. In order to formalize the search task, a model of target's motion is needed. In our work, we studied three models: stationary, stochastic and adversarial. In the stationary model, the target does not move and thus the search problem can be reduced to a coverage problem. In the stochastic model, the target moves based on a probability distribution. Thus, the motion is independent of the searcher's strategy. Finally, in the adversarial model, the target actively avoids capture by moving in the best possible way against any search strategy. Given the above considerations, we define the search problem as follows.

Problem 1 (Search)**.** *Given the environment and the target motion models, design a search strategy such that the target is detected in minimum time.*

In the following, we present an overview of our results in finding the common carp for each of the three motion models.

Approach 1: Stationary Target

It is possible that the fish do not move much compared with the detection range and robot speed. This happens, for example, when the fish aggregate in the winter [3]. In this case, we can reduce the search problem to that of finding a stationary object. In the worst case, the robot must cover the entire environment. To reduce the search time, we incorporate domain knowledge that allows us to restrict the search to only those regions within the lake which are likely to contain the fish (Figure 3(a)). We assume that there is a path between any two points in any pair of regions.

Our general approach is composed of two steps [1]. First, we compute a tour, τ, visiting all regions exactly once. The tour τ defines the order in which the regions will be visited. The (possibly same) points at which τ intersects a region for the first and the last time are termed as the entry and exit points for a region. In the case where the regions appear along the perimeter of a simply-connected lake, the optimal order in τ follows the perimeter (Lemma 1 in [1]). We can compute the entry and exit points by discretizing the region boundaries and using dynamic programming. This can be done in polynomial time because the order is fixed.

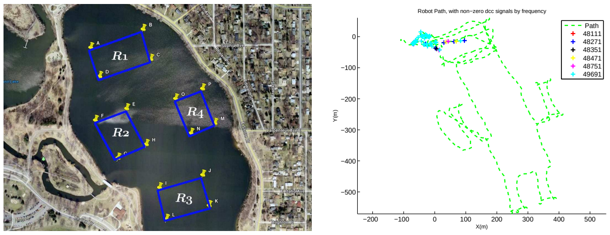
In the second step, we compute a separate coverage tour consisting of boustrophedon paths within each region. The final tour is constructed by augmenting the coverage tours of each region to the corresponding entry and exit points in τ. This algorithm computes a tour whose length is no more than a constant factor of an optimal tour [1].

We evaluated this search algorithm through field trials at Lake Phalen, MN, USA. The input regions for one such trial are shown in Figure 3(a). A sensing range of 50m was used to generate the boustrophedon paths. The actual trajectory executed by the robot is shown in Figure 3(b) which covered 5.6km in about 87min.

Approach 2: Adversarial Target

Next we address the motion of the target. One possible motion model is to treat the target as an adversary which actively tries to avoid the searcher. An adversarial target is aware of the search strategy and it can plan the best possible strategy to escape. Even though a perfectly adversarial target is unlikely in many real-world applications, this analysis

[3] www.imagenex.com



(a) Coverage regions

(b) Actual path executed by the robot.

Fig. 3. Coverage experiment conducted at Lake Phalen, MN, USA. **(a)** The input regions, **(b)** the actual path followed by the robot. The robot traveled a distance of 5.6km in 87min. Locations where signals from radio-tags were detected are marked, along with their frequencies. From [1].

remains important because it can represent a worst case scenario. A capture strategy against an adversarial target (if it exists) guarantees finite capture time for all possible trajectories of the target.

The search problem corresponding to the adversarial target is known as the Princess and Monster Game. In this game, the players are in a dark room, i.e., they cannot observe each other unless the monster captures the princess. This model is applicable because our sensing radius is likely to be bigger than the fish's. This game was proposed by Isaacs [4] and solved by Gal [5] who presented a randomized strategy to find the evader in time proportional to the area of the environment. In [6], we adapt this strategy for a square environment Ω of side L, which yields a simplified analysis.

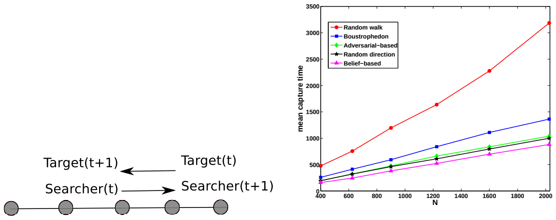
The search strategy is divided into rounds. Each round has two parts which take exactly L steps. The searcher starts on the left boundary of the environment. At the beginning of each round, a row is randomly selected with a uniform distribution and the searcher travels to this row and waits until L total steps have passed. In the second part, it sweeps the entire selected row. The searcher repeats this strategy until a capture occurs. We show that the capture time of this strategy is $O(A)$ where A is the area of Ω. It is worth mentioning that we do not require any assumptions on the players' velocity, and the same result holds for a target which can move faster than the searcher as long as the searcher's detection range is greater than the target's.

Approach 3: Random Walking Target Modeling the fish as an adversarial target can be very conservative. More efficient strategies with less expected capture time can be designed by using probabilistic models for the target's motion. In the absence of any other information, a natural movement model is the simple random walk where the target moves to its neighboring regions with equal probability. We studied two scenarios: When the random walker moves on a line segment between a set of discrete nodes, and when it moves in a square grid. The searcher's task is to find the target by moving onto its node. The players do not observe each other until this happens. How should the searcher move? Surprisingly this problem is open. In the following, we summarize our results for both the one-dimensional and the two-dimensional random walker.

On the line segment we envision two detection criteria that we refer to as the *no-crossing* and the *crossing* conditions. In the no-crossing model, the searcher can detect the target continuously as it is moving along the edges. Since the energy and the time budgets are limited, the best strategy is to move to the right for an optimum number of steps, namely N_R, and then spend the rest of the time budget waiting at $x = N_R$. In [7], we computed the value of N_R that maximizes the total capture probability.

A more challenging detection criterion is the crossing model where detection occurs only at the discrete set of nodes. If the players cross each other by taking the same edge in opposite directions, the target is not detected (Figure 4(a)). This condition takes into account the observation that as the boat is moving, the measurements are not reliable due to the noise interference from the motors. We show that the best strategy is of the form $(R^k S)^m$ – move right for k steps and stop for one step then repeat. The parameter k depends on the time and the size of the environment. In most cases, the best number of right actions is two [8].

The problem becomes much more complex in a square grid because of the larger state space. In [6], we proposed search strategies and compared their mean capture probabilities. See Figure 4. In this figure, in the random walk strategy the searcher moves in the same way as the target. The adversarial-based path is the strategy described in Section III. In the random direction strategy, the searcher picks a random, uniformly-distributed point on the boundary to move toward. When the searcher reaches the point it repeats this process. In the belief-based strategy, at each iteration the maximum of the belief is identified and the searcher moves toward its direction. Among these strategies the best strategy is the belief-based strategy. Interestingly, the belief based strategy performs better than the game theoretic solution assuming that the target is adversarial (Section III). Finding the structure of the optimal strategy by approaching the problem as a Partially Observable Markov Decision Process formulation is our ongoing work.



(a)

(b)

Fig. 4. **(a)** A crossing event is illustrated: at time t, the searcher is at node i, the target is at node $i + 1$, and they move toward each other by taking the same edge. From [8]. **(b)** Behavior of the mean capture time as a function of the number of nodes N for five different searcher strategies: moving as a random walk, boustrophedon path, the game-theoretic strategy and the adaptive belief-based strategy based on the belief function which moves toward the max of the belief. From [6].

IV. Tracking

The previous section discussed methods for detecting the fish. Since radio tags can sometimes be detected from a long distance, detection alone does not provide a good estimate of the fish's location. To provide a precise estimate, the robot can use bearing measurements. The robot can rotate the antenna and sample the signal strength at each orientation. With enough samples, we estimate the orientation with the maximum recieved signal strength, which is roughly the bearing to the radio tag. See Figure 2. Since the fish tend to aggregate in large, stationary groups for long periods, an individual target is considered static during localization. In practice, the slow motions of the target are confined to a small area, and are considered part of the measurement noise. This brings us to our second problem:

Problem 2 (Active Localization)**.** *As quickly as possible, construct an initial estimate of the fish's location and use bearing measurements to locate its position to within a given uncertainty bound.*

As a first step, we establish an initial estimate. In [9], we showed how a Gaussian prior for each target can be constructed by searching for the boundary of the region of detection: a roughly circular area around each tag from which the transmissions can be reliably detected. The centers of these circles are used as the initial target location $\hat{x}$, and each circle becomes the three-sigma boundary of the covariance.

The robot can now begin the process of choosing a sequence of sensor locations $S = \{s_1, \cdots, s_n\}$ from which it will collect bearing measurements and localize the targets. A target is considered localized if the posterior covariance is sufficiently small. Thus, the objective of the localization step is to take measurements such that the eigenvalues of the covariance are reduced to below a threshold.

As an example, to locate a target to within typical handheld GPS precision (approximately 5 meter expected error), we would constrain the covariance to have the square root of both eigenvalues smaller than 5.

Note that the optimal number of measurements for each robot as well as the positions of these measurements is unknown. Additionally, the optimal sequence of measurements must consider the cost per measurement. If the cost is high relative to the cost to relocate, then we expect the optimal algorithm to travel to very informative measurement locations. But, if the cost to relocate the robots is high, we expect the robots to move very little and take many measurements. Communication is a further complication because any two robots must be nearby to share their measurement values, form a consistent estimate of the target location, and plan the next measurement step(s).

We can concisely state this multi-robot active localization problem as follows.

Problem 3 (Active Bearing-Based Localization)**.** *N robots travel with velocity v, take time t_m to take noisy bearing measurements of the target $x^\star$, and have Gaussian sensor noise with variance σ_s^2. Furthermore, two robots must meet to exchange measurements by moving to within a distance r to communicate the results of their measurements. What is the optimal sequence of measurement and communication locations for each robot u, $S_u = \{s_{u,1}, \cdots, s_{u,n_u}\}$ such that all robots have a consistent final target estimate and the final target estimate has bounded expected error?*

We consider two ways of solving the problem: Offline and Online.

Approach 1: Offline Analysis

The offline formulation assumes that the true target location is known. This assumption is not valid in practice. However, we use the solution to analyze the performance of online strategies. In the offline setting, the goal is to bound the expected error of the target estimate, given the known sensor noise, sensor locations, and target location. The metric commonly used in this setting is the Fisher Information Matrix (FIM). The inverse of the FIM is called the Cramer Rao Lower Bound, and gives the expected error of the target estimate around the true target location (the covariance).

In [10], we prove the geometric structure for the optimal offline solution for two cooperating robots. The optimal solution can then found using a simple gradient descent search. Subsequently, the two-robot solution is extended to include communication constraints so that both robots can rendezvous and form a joint estimate of the target location. Finally, the same solution is extended to include an arbitrary number of robots without increasing the computational complexity. These results are summarized in the following theorem.

Theorem 1 (Optimal Offline Active Localization)**.** *There exists an algorithm $\mathcal{A}$ which produces a sequence of measurement locations for N robots with respect to a known target location such that when all measurements are combined, the following are satisfied. Let the final covariance around the true target location be Σ.*

- *The maximum eigenvalue of Σ is less than the desired amount.*
- *When $N = 2^i$ for a non-negative integer i, the cost of $\mathcal{A}$ is optimal.*
- *When $N = 2^i + 1$ for a positive integer i, the cost of $\mathcal{A}$ is less than 2 times optimal.*
- *When the robots have distance-based constraints on their communication, the cost of $\mathcal{A}$ is less than twice the optimal cost.*
- *The computational complexity of computing the N robots' measurement and communication locations is independent of N.*

The algorithm is presented in [10]. Critically, the cost bounds in the previous results incorporates the velocity of the robots, communication constraints between them, measurement time, and measurement uncertainty to produce a baseline comparison for the cost of any bearing-only active localization algorithm.

Approach 2: Online Analysis

In the online setting, the measurements collected at each

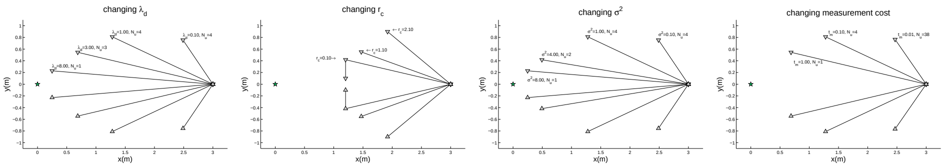


Fig. 5. Optimal two-robot trajectories for various system parameters. In all figures the robots started at location $(3, 0)$ and the true target was at $(0, 0)$. Communication constraints were only considered in the top right figure and all other parameters were held fixed. Top-left: $\lambda_d \in \{.1, 1, 3, 8\}$. Top-right: $r_c \in \{2.1, 1.1, .1\}$. When $r_c = .1$, the robots rendezvous after measuring. When $r_c = 2.1$ the output is the same as the result from unbounded r_c. Bottom-right: $t_m \in \{.01, .1, 1\}$ Note that as t_m increases, the optimal algorithm travels to more informative locations so that fewer measurements are required. Bottom-left: $\sigma \in \{.1, 1, 4, 8\}$. Note that changing the sensor noise produces the same effect as requiring more information (compare left two figures). From [10].

location are used to update the target estimate. Then, subsequent measurement locations are planned with respect to the new estimate. In [11], a single robot algorithm is provided which can localize a single target using only a prior estimate of its location. The cost of the algorithm was bounded, and compared to the optimal offline algorithm. An example of the trajectories and measurement locations for two robots is shown in Figure 5.

To extend to multiple robots, the optimal offline algorithm was shown to be adaptable to an online setting using results also presented in [10]. We show that any online algorithm satisfying mild requirements about measurement positioning with respect to the target estimate can be used to build a near-optimal online algorithm. In fact, the resulting online algorithm is shown to be within a logarithmic factor of the optimal cost. Thus, the previous result can be extended to an online strategy which is proven to locate the target up to a desired uncertainty level at near-optimal cost:

Theorem 2 (Online Adaptation)**.** *Let $\mathcal{A}$ be an offline algorithm which produces a sequence of measurement locations for N robots with respect to a known target location. Then $O(\log(\lambda))$ calls to $\mathcal{A}$ are required to localize a target in an unknown location, when the final covariance, Σ must have both eigenvalues less than $\frac{1}{\lambda}$. Let the cost of $\mathcal{A}$ be $O(C^\star)$, where $C^\star$ is the cost of the optimal algorithm. Then the cost to localize a target in this way is a logarithmic factor worse than optimal. Or, $O(C^\star \log(\lambda))$.*

The online algorithm was tested in field trials on Lakes Staring and Gervais and on the University of Minnesota campus. One such trial is shown in Figure 6.

V. Energy

When mobile sensors must operate outdoors, their system lifetimes are often severely constrained by the maximum energy of their batteries. Therefore, energy awareness and optimization is critical for these settings. In this section we present methods for planning *how* a mobile robot should make its way to a sensing position while minimizing its energy cost. This is a task with two parts: First we will examine how a mobile robot should navigate to a position in a way that minimizes the energy expended by its motors. Then we will examine how the robot should navigate to maximize its battery life in situations when the robot is equipped with solar photovoltaic panels which can harvest energy from the environment.

A. Power to Drive

First, to understand the structure of optimal trajectories, consider the problem when the robot is car-like (with minimum turning radius but no turning cost), the path is fixed, such as a road or a track, and we only need to find the optimal velocity profile. There is a constraint, however, that different portions might have different maximum speeds, usually from traction constraints while turning at a particular radius on a particular surface. We can formally phrase this problem as follows:

Problem 4 (Velocity Optimization)**.** *Let the path that a car-like robot must traverse be τ. τ consists of N segments which are straight lines or curves. There is a separate velocity bound for each segment. Find the optimal velocity profile to minimize the power consumed by the robot while traversing τ.*

In [13] this problem is solved in closed form given a detailed electric motor model and a single segment. When there are multiple segments, the optimal profile can by found through dynamic programming by discretizing the velocity at the transition points between segments. An example optimal velocity profile is shown in Figure 7(a). The strategy is experimentally calibrated to the motor of a real robot car, and optimal velocity profiles are planned and executed.

Now consider the more general case, where the robot only needs to reach a particular destination and it is free to chart its own path:

Problem 5 (Trajectory Optimization)**.** *Given start and goal poses, compute a path τ and a velocity profile along this path for a car-like robot traverse from the start to the goal while minimizing its power consumption. The velocity at all times must obey a traction constraint which restricts the maximum speed as function of the turning radius.*

In [13] this problem is solved for the same motor model through dynamic programming over discrete positions and velocities. The solution is optimal, subject to the resolution of discretization by fixed-curvature paths. The solution is often quite different from selecting first the shortest feasible path and then optimizing the velocity; frequently a longer path can result in lower overall energy cost, by allowing the

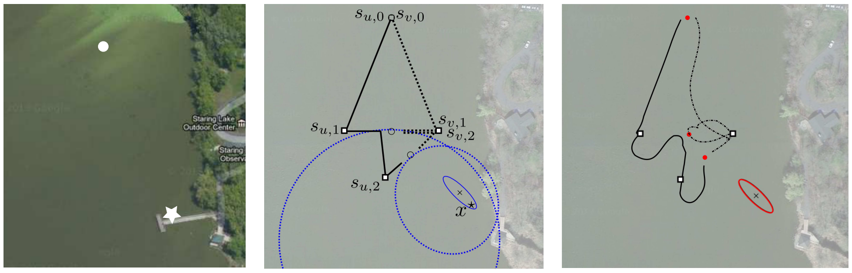
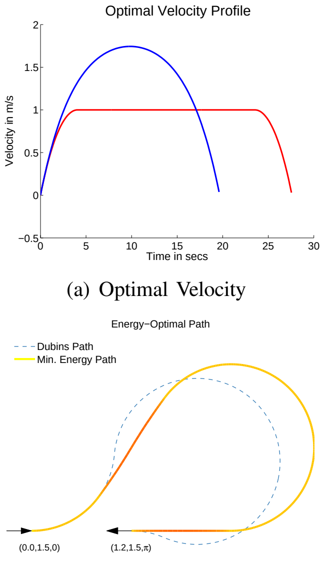


Fig. 6. Experiment results from Lake Staring, MN, USA using two robots and the algorithm described in [10]. **(a)**: The experiment area. The true target (and camera in [12]) were placed on the docks near the bottom right corner (labelled with a star). The robots began near the top-middle (circle). **(b)**: The two measurement steps produced the dark paths shown, with measurements taken at the white squares. The final actual uncertainty was the inner blue ellipse. **(c)**: More execution details. The solid red circles are the points where the robots exchanged information and the measurement locations are again given as white squares. The figure covers an area approximately 200m vertically by 150m horizontally. From [10].



(a) Optimal Velocity

(b) Optimal Trajectory

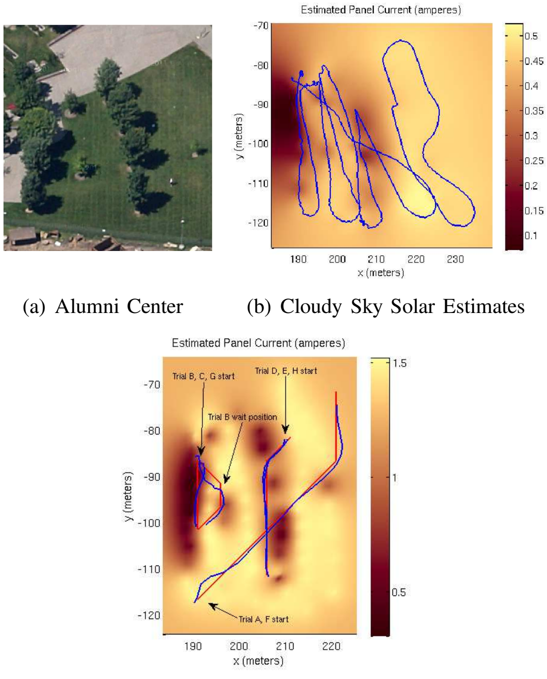
Fig. 7. Optimal unlimited velocity profile compared with optimal profile limited to less than 1 m/s, and optimal trajectory compared with shortest Dubin's path. When the robot turns less sharply, it can travel at a higher speed, and reach its destination with less energy consumption. From [13].

robot to traverse it at a more efficient speed. An example is shown in Figure 7(b).

B. Solar Power Estimation

Now, consider energy that can be collected from the sun. If the field of solar power is stationary across the state space of the robot, minimizing the motor current is sufficient to also maximize the battery of the robot after the move. However, if the field is complex, different paths to the destination might collect different amounts of solar power. In some cases it is even worthwhile for the robot to deviate far from the shortest path and wait in the sun if there is sufficient time.

For the robot to plan the energy-optimal path, it requires an accurate map of the solar field. Often this field is estimated for clear conditions by first estimating the 3D geometry of the environment in high detail then performing raytracing from the computed position of the sun at any time. However, when the geometry is not known a priori, it can be challenging to reconstruct without the use of sophisticated sensors. An interesting problem, therefore, is the following: Can a robot with minimal sensors estimate the solar power it will receive if it travels along a candidate path? A natural minimally-sensing robot is one which can measure only its own position, and how much solar power it receives as it moves. In this setup we estimate future measurements of solar power at positions from past measurements of solar power at positions. The problem can be formally stated as follows:



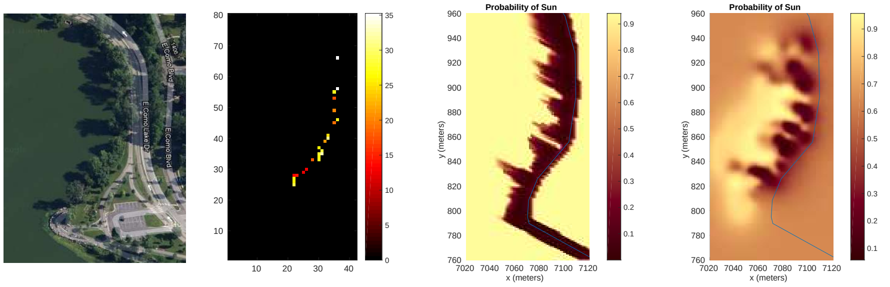
(a) Alumni Center

(b) Cloudy Sky Solar Estimates

(c) Clear Sky Solar Trials

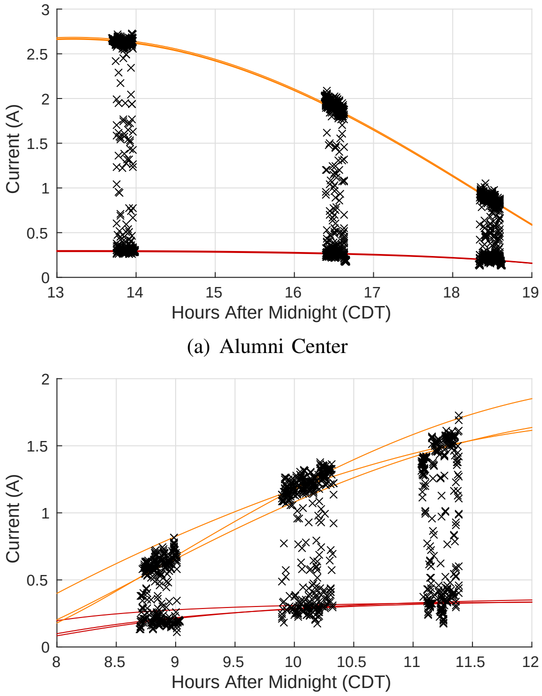
Fig. 8. The McNamara Alumni Center testing environment, and Gaussian Process Regression estimates of the solar intensity across the environment on a cloudy and clear day. The cloudy estimate includes the training path that the robot drove along to form its estimate. The clear estimate includes planned and executed trials where the robot attempted to minimize its energy cost to reach the destination in the specified amount of time, given the solar map. From [14].

Problem 6 (Solar Estimation)**.** *Let the prior points visited by the robot be* $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$*, where each* $\mathbf{x}_i$ *is*



(a) Lake Como (b) $\mathcal{H}^\ell$ in meters (c) Estimate for 09:30 (d) Estimate for 09:30

Fig. 9. The Lake Como testing environment, the estimated lower bound heightmap constructed from observations of solar power at positions, an example solar map estimated for 09:30 CDT from the constructed heightmaps, and by using GP regression. From [15].



(a) Alumni Center

(b) Lake Como

Fig. 10. The measured solar current for our trials plotted against time of day, with best fit curves estimating the current that can be collected in the sun and in the shade. There are three curves showing the estimate for leaving out each of the data sets. At the Alumni Center the three curves agree almost perfectly, but this is not the case at Lake Como. The average cross validation error at the Alumni Center was 0.0660 amperes, and at Lake Como it was 0.1301 amperes. For planning purposes, the estimated magnitude is close to the true magnitude. From [15].

a vector containing the position and time of the visit. The robot records the solar power y_i at each point to form the set of measurements $\mathcal{Y}$. Estimate solar power $\mathcal{Y}^*$ that will be received at points $\mathcal{X}^*$.

Approach 1: Gaussian Process Regression

A first approach to estimate the solar field is by approximating it as a continuous spatial Gaussian Process, as in [14]. This approach has the advantage that it is straightforward to implement, and it works equally well for cloudy and clear days. However, it is most useful over short time scales, because it ignores the change in solar intensity and sun position during the day.

Approach 2: Gaussian Process Classification

On a sunny day, almost all positions can be classified as `sunny` or `shaded`. Then both the shadows and the solar intensity can be adjusted during the day for the position of the sun, treating the shadow map as a spatiotemporal Gaussian Process. This approach is detailed in [15]. It has the advantage that it can provide good estimates of shadow maps with very little information about the full shapes of the shadows that have been partially observed.

Approach 3: Environment Reconstruction

If there is sufficient information about the shadows that have been cast, it is possible to reconstruct the environmental geometry of the objects that cast the shadows. In [15] this is done by modeling the shadow casting objects as a terrain, with an upper bound and a lower bound on the height of the terrain at each discrete square. The accuracy of the estimated environment can be improved with a prior estimate of the position of the objects, such as a satellite image of the boundary of a lake.

All three approaches to solar estimation were attempted at the McNamara Alumni Center, a field at the University of Minnesota in Minneapolis that contains scattered trees. In addition, Gaussian Process Classification and Environment Reconstruction were attempted at Lake Como, a lake in St Paul, Minnesota. At the lake it was possible to eliminate candidate positions of shadow casting objects when using the reconstruction method. All three methods produced accurate solar maps when there was sufficient coverage. Environment reconstruction accurately placed shadow-casting objects, as shown in Figure 9. Quantitatively, GP regression was successfully used to plan solar-aware paths, as shown in Figure 7. Estimates of shadow maps from GP classification and reconstruction have good cross-validated performance, as do the estimates of solar power in the sun and shade across the day (shown in Figure 10).

C. *Planning Solar-Aware Paths*

Finally, there remains the problem of how to find the lowest energy cost path given a power-to-drive model and a solar map. The problem setup often requires a time limit to reach the destination, otherwise a solar-powered robot might have unbounded energy gains. This problem was addressed in [14], using a simplified power-to-drive model for a differential drive rover. The path is found through spatiotemporal dynamic programming. The power gained by the rover was compared with the power gained by a rover without a panel that traveled straight to the destination. The results of the experimental trials are shown in Table I. These results demonstrate that it is possible for a solar-powered robot with minimal sensors to learn enough about its environment to plan efficient paths.

VI. Conclusion

We presented our efforts toward building an autonomous multi-robot system for tracking mobile targets – in this case, radio-tagged invasive fish. We presented an overview of our results on three motion planning problems where the goals are (1) to find the target, (2) to precisely localize the target, and (3) to maximize energy efficiency both by minimizing energy consumption and by maximizing harvested solar energy. We reported our main theoretical results and results from field experiments. For further details, the reader is referred to corresponding papers.

In order to make the system fully autonomous, other challenges must be overcome. One interesting question is regarding communication: if the robots lose each other, how can they re-establish communication? This can be formulated as a cooperative search problem. Another issue which proved critical is fault detection. For extended missions, the robots must be able to detect e.g. when a sensor fails and find ways to mitigate this failure perhaps by reporting to a mission center. Finally scaling the system to many robots is an exciting research frontier.

Acknowledgement

This work is funded by the National Science Foundation Project # 1111638: *A Robotic Network for Locating and Removing Invasive Carp from Inland Lakes*, #0916209, #0917676 and #0936710. The authors would like to thank RSN Lab members who contributed to the project throughout the years. The authors would also like to thank Brett Miller and Mary Haedrick for their involvement in gathering telemetry data.

References

[1] P. Tokekar, E. Branson, J. Vander Hook, and V. Isler, "Tracking aquatic invaders: Autonomous robots for monitoring invasive fish," *Robotics Automation Magazine, IEEE*, vol. 20, pp. 33–41, Sept 2013.

[2] P. A. Plonski, J. Vander Hook, C. Peng, N. Noori, and V. Isler, "Navigation around an unknown obstacle for autonomous surface vehicles using a forward-facing sonar," Tech. Rep. 15-005, Department of Computer Science, University of Minnesota, 04 2015.

[3] P. G. Bajer, H. Lim, M. J. Travaline, B. D. Miller, and P. W. Sorensen, "Cognitive aspects of food searching behavior in free-ranging wild common carp," *Environmental Biology of Fishes*, vol. 88, no. 3, pp. 295–300, 2010.

[4] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Welfare and Pursuit, Control and Optimization*. John Wiley and Sons, 1965.

[5] S. Gal, "Search games with mobile and immobile hider," *SIAM Journal on Control and Optimization*, vol. 17, no. 1, pp. 99–122, 1979.

[6] A. Renzaglia, N. Noori, and V. Isler, "The role of target modeling in designing search strategies," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 4260–4265, Sept 2014.

[7] N. Noori, A. Renzaglia, J. Vander Hook, and V. Isler, "Constrained probabilistic search for a one-dimensional random walk," *Under review, IEEE Transactions on Robotics*, 2013.

[8] N. Noori, A. Renzaglia, and V. Isler, "Searching for a one-dimensional random walker: Deterministic strategies with a time budget when crossing is allowed," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 4811–4816, Nov 2013.

[9] J. Vander Hook, P. Tokekar, E. Branson, P. G. Bajer, P. W. Sorensen, and V. Isler, "Local-search strategy for active localization of multiple invasive fish," in *Experimental Robotics* (B. Siciliano and O. Khatib, eds.), vol. 88, pp. 859–873, Springer Tracs in Advanced Robotics, 2013.

[10] J. Vander Hook, P. Tokekar, and V. Isler, "Algorithms for cooperative active localization of static targets with mobile bearing sensors under communication constraints," *IEEE Transactions on Robotics*, 2015. Accepted. Available as TR 13-012 from the University of Minnesota.

[11] J. Vander Hook, P. Tokekar, and V. Isler, "Cautious greedy strategy for bearing-only active localization: Analysis and field experiments," *Journal of Field Robotics*, vol. 31, pp. 296–318, April 2014.

[12] "Two-robot field experiment `https://www.youtube.com/watch?v=jZWTUxqTskY`," 2013.

[13] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal trajectory planning for car-like robots," *Autonomous Robots*, vol. 37, no. 3, pp. 279–300, 2014.

[14] P. A. Plonski, P. Tokekar, and V. Isler, "Energy-efficient path planning for solar-powered mobile robots," *Journal of Field Robotics*, vol. 30, no. 4, pp. 583–601, 2013.

[15] P. A. Plonski, J. Vander Hook, and V. Isler, "Environment and solar map construction for solar-powered mobile systems," Tech. Rep. 15-004, Department of Computer Science, University of Minnesota, 04 2015.

Solar Trial	**Duration**	**Expected Solar**	**Actual Solar**	**Expected Net Cost**	**Actual Net Cost**	**Control Trial**	**Duration**	**Actual Net Cost**
A	401 s	7, 025.5 J	6, 974.1 J	577.16 J	744.6 J	F	45 s	6, 295.4 J
B	400 s	6, 606.6 J	6, 828.6 J	−3, 265.9 J	−3, 256.7 J	G	19.1 s	2, 888.1 J
C	104 s	1, 148.3 J	611.26 J	879.99 J	2, 253, 4 J			
D	104 s	1, 600.9 J	1, 297.6 J	2, 907.3 J	3, 480.3 J	H	30.4 s	3, 530.4 J
E	104 s	1, 600.9 J	1, 156.2 J	2, 907.3 J	2, 822.5 J			

TABLE I

PATH EXECUTION RESULTS. ON THE LEFT SIDE OF THE TABLE ARE THE FIVE PLANNED AND EXECUTED SOLAR-AWARE PATHS, SORTED BY START POSITION AND END POSITION. ON THE RIGHT SIDE OF THE TABLE ARE THE THREE SHORTEST PATHS EXECUTED WITH NO SOLAR PANEL, FROM THE SAME START AND END POSITIONS AS THE TRIALS DIRECTLY TO THEIR LEFT. OBSERVE THAT, AS EXPECTED, THE SOLAR POWERED ROBOT PERFORMS BEST WHEN IT IS ALLOWED TIME TO DEVIATE FROM THE SHORTEST PATH AND CHARGE ITS BATTERY IN THE SUN. FROM [14].