Aquatic and Land-based Robotic Telemetry for Tracking Invasive Fish

Elliot Branson, Volkan Isler, Pratap Tokekar and Josh Vander Hook

***Abstract*— Carp is a highly invasive, bottom-feeding fish which pollutes and dominates lakes by releasing harmful nutrients. Recently, environmental scientists started studying carp behavior by tagging the fish with radio-emitters. The radio-tagged fish are tracked manually using GPS and a directional antenna. We have been working on developing a novel robotic sensor system in which the human effort is replaced by autonomous robots to find and track carp. During the summer months, we use robotic boats whereas in the winter, mobile robots track the fish on frozen lakes.**

In this extended abstract, we report the current state of our system including system architecture, coverage and active tracking algorithms. We also present results from field experiments including coverage experiments in which our boat travels 2.4 km in Lake Keller in Minnesota.

I. INTRODUCTION

Invasive fish such as the common carp pose a major threat to the ecological integrity freshwater ecosystems around the world. Presently, the only way to control these fish is through the use of non-specific toxins which are expensive, ecologically damaging, and impractical in large rivers and lakes. Recent studies in small lakes have established that some of these fishes aggregate densely at certain times and places and can be controlled by targeting these aggregations using netting. Therefore, biologists started using a new technology based on tracking radio-tagged carp to accurately predict the presence of large carp populations.

Unfortunately, carp aggregations are unpredictable. Manually locating tagged fish in large, turbid bodies of water remains a difficult task. Our goal is to replace this manual effort with robots. Toward this goal, we developed an autonomous robotic boat (Figure 1(a)) capable of localizing tagged fish in lakes and a field robot (Figure 1(b)) performing the same task on frozen lakes.

In this work, we build on our previous system [1] and present the following improvements: (1) **Coverage:** We have recently developed a new coverage algorithm to detect the presence of fish. The algorithm takes regions that are likely to contain fish as input and computes a path to cover these regions. This allows for incorporating scientists' domain knowledge. (2) **Active Localization:** After detecting the fish, the goal is to accurately estimate its location. We use multiple measurements taken at various locations for estimation. The problem we address is how to choose measurement locations in an online fashion so as to accurately localize the fish with a small number of measurements. We report recent results on active localization.

We also report results from field experiments for both the problems. We begin with the coverage problem.

(a) Robotic boat at during coverage experiments at Lake Keller, MN.

(b) Robot (with tracking equipment and antenna) on frozen Lake Casey, MN.

Fig. 1. Robotic system for monitoring radio-tagged carp during field trials.

II. SEARCH AND COVERAGE

In this section, we present our coverage algorithm for finding fish. We say that location x on the lake is *covered* if the boat moves to a location y from where the tag on the fish when located at x can be heard. While the fish move significantly throughout the day, they are expected to remain within a certain area for shorter periods of time. If we assume that a fish is approximately stationary during the search phase, the searching task reduces to a coverage task: find the shortest trajectory which ensures that all possible locations of the fish are covered.

We can speed-up the coverage task by incorporating domain knowledge. Suppose we are given a set of regions which are likely to contain the fish. For example, these can be areas rich in vegetation. We assume that these regions are connected in the sense that there is a path between any two points. We model the *regional contiguity* property as follows:

When the robot visits a region, it must cover it completely before visiting another region.

With the regional contiguity requirement, the coverage problem can be defined as follows: Given a set of connected regions $R = \{R_1, R_2, \dots, R_n\}$, find a minimum length tour with the regional contiguity property which covers every point in each region $R_i \in R$.

We propose an approach composed of two steps: First, we compute an α approximation tour τ_R that visits all the regions in R. We say that region R_i is *visited* if *any* point in R_i is visited by the tour. The tour, τ_R, imposes an ordering on the regions. Next, we compute a β approximation coverage

The work presented in this extended abstract is being considered for publication in IROS 2011. This work is supported by NSF Awards #0916209, #0917676, #0936710 and a fellowship from the Institute on the Environment at the University of Minnesota. The authors are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA. {elliot,isler,tokekar,jvander}@cs.umn.edu

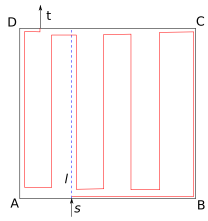


Fig. 2. Covering a rectangle with given entry and exit points.

tour C_{R_i} for each region $R_i \in R$ independently. The final tour τ is constructed by adding the coverage tours of each region to τ_R. We prove that imposing regional contiguity costs at most a factor $(\alpha+\beta)$ deviation from the unrestricted optimal solution.

We now present algorithms for the two components of the algorithm: Computing a tour that visits the regions and covering the regions.

A. Visiting the regions: TSPN and the Zookeeper Problems

The computation of the tour τ_R depends on the geometric properties of the regions. If the regions are convex polygons touching the boundary of a (simply-connected) lake then the tour can be computed optimally by computing the so-called zookeeper's route [2]. In this case $\alpha = 1$. If the regions are arbitrarily placed, we can use algorithms for TSP with neighborhoods (TSPN) such as [3].

Most geometric instances of the TSPN problem are NP-Hard. In our application, it is reasonable to model the lake as a simply-connected region. Further, areas of interest where the fish may lie are usually close to the shore because of vegetation and oxygen levels. This special instance of TSPN known as the zoo-keeper problem can be solved in polynomial time due to the following lemma.

Lemma 1 ([2]): Let $R = \{R_1, R_2, \ldots, R_i, \ldots, R_n\}$ be a set of convex regions located along the perimeter of a simply connected polygon P. There exists an optimal solution for visiting the regions in R which visits them in the order they appear along the boundary of P.

Once the ordering of the regions is known, the shortest tour visiting all regions can be calculated which yields entry and exit points for each region. To turn these tours into coverage paths, we need a way to cover a region with given entry and exit points. The computation of these paths is presented next.

B. Coverage

In the second step of our algorithm, a coverage tour for each region is computed. In our application, we represent regions with rectangles with arbitrary orientations since they are easy to specify on one hand and general enough for practical purposes on the other.

The algorithm presented in Section II-A generates an entry and exit point for each region. Our coverage problem is to travel through every vertex in a given rectangular graph with a given starting and ending point. The following lemma shows that we can cover the entire rectangle efficiently even with this constraint.

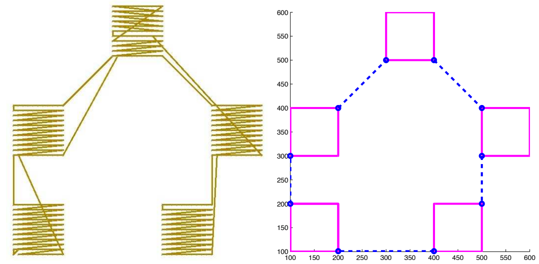


Fig. 3. Complete TSP path. Using the same regions the TSPN graphs show the entry and exit points for the region along with the the paths between the regions. The coverage paths are not shown for clarity.

	TSP	TPSN	Coverage	Total
Environment 1	8,585	500	4,970	5,470
Environment 2	11,361	600	5,940	6,540
Environment 3	2,522.8	302.2	1,450	1,752.2
Environment 4	3,278.1	371.44	1,825	2,196.4
Environment 5 (Fig. 3)	11,856	682.84	4,950	5,632.8

TABLE I

A COMPARISON OF THE TSP PATH VS. COMBINED TSPN/COVERAGE PATH FOR DIFFERENT INPUT SETUPS.

Lemma 2: Let R be a rectangle with a grid imposed on top. Let s and t be two grid points on the boundary specified as entry and exit points. There exists a tour T which starts at s, visits every grid point and exits at t such that the length of T is at most twice the optimal tour which visits every point.

Proof: Given start and end points s and t respectively, we construct a coverage path as shown in Figure 2. This path consists of three parts: an optimal part that covers the rectangle with length equal to that of the OPT, and parts that connect s and t to the start and end points of this optimal part. We can prove that the two connecting parts are non-overlapping and giving us a 2-approximation. The details of the proof are deferred for the full paper. ■

We also show that this analysis is tight: there are instances where we cover the region twice when s and t are fixed.

We now evaluate our proposed algorithms through simulations and field experiments for covering the lake.

C. Simulations and Field Experiments

We first compare the performance of our algorithms with the standard TSP solution. The TSP solution uses all grid points to be covered independent of the regions. For computing the TSP solution, we use the heuristic by Christofides [4] which yields a 3/2-approximation. We ran the two algorithms for the environments given in Figure 3. The results are reported in Table I, whose first column is the length of the TSP tour and the last column is the length of our solution.

As these results show, in addition to enforcing regional contiguity, our algorithm is more efficient than the Christofides heuristic in these instances. It seems that the matching component of the Christofides heuristic sometimes yields long tours. For example, in Figure 3, the TSP path is almost twice as long as our solution.

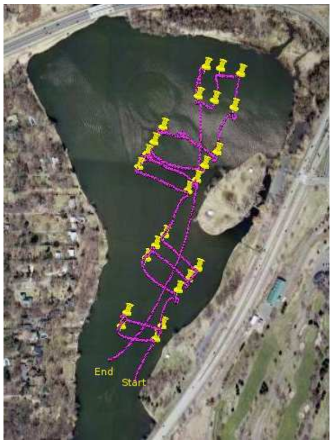


Fig. 4. The GPS trace of the path taken by the boat during the experiment. The trails shows that the boat covered all four regions by visiting all the waypoints robustly. Also the trace suggest that the navigation algorithm negotiated well with the drift caused by wind. The boat traveled approximately 2.5 km in 36 minutes of the run.

We conducted field experiments in Lake Keller, Maplewood, MN to test the coverage algorithm and the navigation performance of the system. The size of the lake is approximately $900m \times 350m$.

We fixed four regions of interest in the lake. The dimension of the four regions are approximately $71m \times 23m$, $100m \times 70m$, $118m \times 100m$ and $93m \times 83m$ with a total area of $28,150m^2$. During the experiment the boat traveled approximately 2.5 kilometers in 36 minutes until all the waypoints were covered Figure 4 shows the boat's path.

From the experiment we conclude that the coverage algorithm proposed in this paper is useful for real applications. The experiment also demonstrates that our robotic system is capable of robustly navigating using waypoints for long periods of time.

Using the above algorithm, we can efficiently search the lake for stationary fish. However, the radio antenna has a large range (about $30m$) and we get only coarse estimate for the fish. To accurately localize the fish, we must combine multiple measurements. The following section proposes three strategies to obtain these measurement locations so that the resulting uncertainty in fish is minimized.

III. Active Localization

We first describe how we obtain bearing measurements from the radio antenna. Then, we propose three active localization strategies followed by their evaluation through simulations and field experiments.

A. Measurement Model

The radio antenna used to detect the tag is direction sensitive: the signal strength output from the antenna depends on the relative angle between antenna and the tag. Hence, we take a coarse sampling of signal strength by rotating the antenna in steps of $15°$. We can then fit sine waves and third-degree polynomials using least squares, estimate the maxima accordingly (Figure 5).

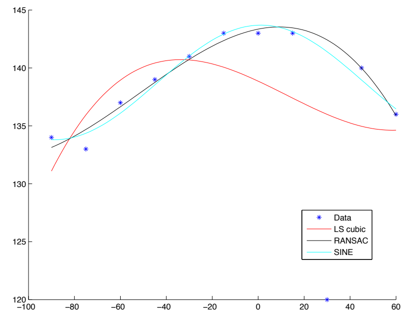


Fig. 5. A coarse sampling (signal strength versus bearing) and various least-squares fitting. RANSAC estimation of a cubic polynomial typically provided the best estimates. The true bearing is $15°$.

This maxima gives us a bearing measurement towards the target. Our objective is to estimate the location (x_f, y_f) of the fish using these bearing measurements. We use an Extended Kalman Filter with the combined robot and target state $X(t) = (x_r, y_r, \theta_r, x_f, y_f)$ to be estimated. The onboard GPS and compass measurements are used to perform EKF updates for the robot state, while bearing measurements are used to update the entire state. Since the bearing measurement function is a non-linear equation, we linearize the measurement about the current state estimate. The resulting uncertainty (as show by the determinant of state covariance) depends on the locations from where the measurements were obtained. Hence, we can optimize these measurement locations to minimize the final uncertainty.

The underlying telemetry technology used by the fisheries researchers introduces another constraint: each tag emits a signal at a dedicated frequency once every second. Since each bearing measurement requires sampling the antenna in multiple directions, we restrict the total measurements to k discrete locations as opposed to obtaining continuous measurements.

For the discussion that follows next, we assume that the initial fish location and covariance estimates are known, propose three strategies for optimization and evaluate them with simulations and field experiments, then discuss the initialization procedure at the end of the section.

B. Active Localization

1) Cramer-Rao Lower Bound: The Cramer-Rao lower bound for an unbiased estimator $\hat{X}$ of state X is a lower bound on the estimation error covariance matrix P_k and is given as the inverse of the Fisher Information Matrix (FIM) I. For k bearing measurements with zero-mean Gaussian noise, determinant of I is inversely proportional to the square of the area of the 1-σ uncertainty ellipse and can be expressed as,

$$|I| = \frac{1}{\sigma^4} \sum_{i=1}^{k} \sum_{j=1}^{k} \left[\frac{\sin(\theta_i - \theta_j)}{d_i d_j} \right]^2 . \quad (1)$$

where $\Delta x_i = (x_r(i) - x_t)$, $\Delta y_i = (y_r(i) - y_t)$, and $d_i^2 = \Delta x_i^2 + \Delta y_i^2$. Here, $(x_r(i), y_r(i))$ is the location of the robot for the i^{th} measurement, and (x_t, y_t) is the true target location.

To compute the k locations, we impose a grid about the current position of the robot of size $n \times n$. The total number of candidate points for measurement locations are n^2. Hence, to compute the k measurement locations, we consider each of the $C(n^2, k)$ combinations as a candidate trajectory and compute the FIM given by 1.

2) Greedy: Instead of computing a fixed path for the k measurements, we can instead use a greedy strategy which picks the next measurement location based on the current estimate and uncertainty of the target. Given the current robot position and target position, Greedy looks at all neighboring locations of the robot. At every location, we simulate all candidate measurements (e.g. by uniformly picking s samples between 0 to 360°). Using the current state and covariance, we can estimate the posterior covariance by simulating an EKF update using each of these candidate measurements. Thus, for every neighboring location, we will have s posterior covariances. Greedy then picks the candidate location where the maximum determinant of the s posteriors is minimum. This ensures best worst-case uncertainty for the target's position in a greedy fashion. Instead of the best worst-case uncertainty, we can choose some other heuristic for the greedy.

3) Enumeration tree: We extend the objective function of Greedy here, to minimize the worst-case uncertainty obtained by the EKF after k measurements. We use a min-max tree to achieve this objective.

The tree is built by assigning each adjacent measurement location to an action node, and the corresponding measurements to bearing nodes. We recursively define the uncertainty of the actions and bearings and build the tree to depth $2k$, which would correspond to the k desired measurement locations and k measurements. Each bearing node holds a worst-case estimate of the measurement uncertainty, as calculated by the EKF propagation.

Since we use discrete measurement samples while building the tree, we need to find that child node which is closest to the current measurement. As there is some uncertainty associated with the position of the robot itself, we instead use the *Bhattacharya Distance* to find that child node, whose posterior covariance is closest to the current robot covariance (after the measurement update). The robot then repeats the above steps until it reaches the leaf nodes (corresponding to the k^{th} measurement location).

In each of the three strategies proposed above, we assume initial estimates for the fish position and covariance were known. We use the following lemma to pick the first two bearing measurement locations before beginning the strategies.

Lemma 3: Let r_{min} and r_{max} be the minimum and maximum sensing range of the sensor. Assume w.l.o.g the first measurement taken from the origin is along the X-axis. Then, if the second measurement is taken from $(\frac{r_{max}+r_{min}}{2}, \pm\frac{r_{max}-r_{min}}{2})$, the worst-case uncertainty in the target's position after two measurements is minimized.

C. Simulations and Experiments

We ran 100 random trials for each strategy using the same initial conditions, target locations, and random seed to generate measurement noise. The result of the simulations is presented in Table II, and the corresponding histograms of final error and determinant of the final covariance matrix are shown in Figures 6(a) and 6(b) respectively. The outliers with large error resulted from poor initial estimates.

TABLE II

SIMULATION RESULTS FOR 100 TRIALS

Method	Mean final error	Mean final uncertainty
Enumeration tree	$5.7275m$	48.36
Greedy	$5.9809m$	40.59
FIM	$6.2975m$	54.81

The two best closed-loop (online) strategies, Enumeration tree and Greedy were then evaluated in field experiments using the Husky and tracking equipment (Figure 1(b)). Two results are shown in Figures 6(c) and 6(d). In both plots the robot's mean estimated positions are labeled by green circles, while estimates of fish locations are blue marks. The rest of the results are presented in Table III. Similar to the simulation results, we can see that the Enumeration tree performs better than the Greedy strategy.

TABLE III

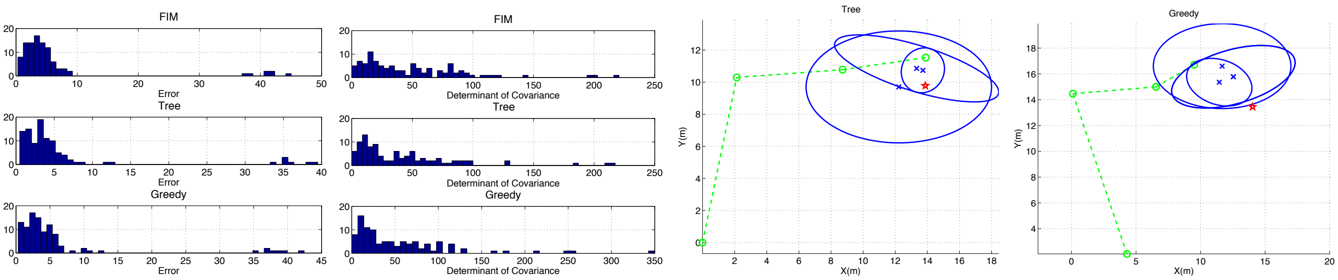
EXPERIMENTAL RESULTS WITH DEPTH 2

Method	Final error	Final uncertainty
Enumeration Tree	0.97	3.53
	3.32	8.57
	5.35	6.04
Greedy	3.21	20.52
	3.29	11.93
	8.65	11.34

From the results we observe that both the mean final error and final uncertainty (determinant) is better for the Enumeration Tree, where as the FIM strategy performs the worst of the three. This result is not surprising, for two main reasons: (1) Since the true target location is unknown, we compute the FIM using the initial estimate of the target's location. (2) The FIM strategy computes locations which minimize the lower bound on the final uncertainty of an "efficient estimator". Since the Extended Kalman Filter is not an efficient filter, there is no guarantee that it would achieve this lower bound. On the other hand, the Enumeration tree and the Greedy actually compute the covariance of the EKF estimator and pick the location which would minimize its determinant.

IV. CONCLUSION

In this paper, we focused on a novel application in which a robotic boat equipped with a directional antenna searches for radio-tagged invasive fish and picks measurement locations to precisely localize the fish. We presented a new coverage



(a) Final errors.

(b) Determinant of the final covariance.

(c) Enumeration tree experiment.

(d) Greedy experiment.

Fig. 6. **Simulations (a) & (b):** We conducted 100 trials with $k = 3$ for each strategy. The mean final error for FIM, Enumeration tree and Greedy was $6.30m$, $5.73m$ and $5.98m$ respectively, and determinant of final covariance was 48.36, 40.59 and 54.81 respectively. **Experiments (c) & (d):** .The true location of the tag is marked by a star. The initial estimate along with estimates after first and second measurement is shown with 1-σ bounds in blue. The measurement locations are shown in green.

algorithm with regional contiguity properties. After presenting theoretical results on the performance of the algorithm, we compared it with a standard TSP solution. In field experiments, we showed that the boat can cover large areas efficiently using our algorithm. For the localization problem, we proposed three strategies, compared them in simulations and reported results from field experiments which show that our system is capable of localizing the target within a meter of the true location.

There are a number of directions we have identified for future work. Since the fish move very little for long periods of time in the winter, in our algorithms we make the assumption that the target is stationary. To handle violations of this assumption, we are working on strategies to localize moving fish. We are also planning to acquire additional boats. Having multiple robots is useful because robots can localize the fish as well as each other more accurately. New algorithms for multi-robot coordination are being developed and will be tested on the field.

V. ACKNOWLEDGMENT

We thank the members of the Sorensen lab at the Department of Fisheries at the University of Minnesota for many useful discussions and sharing equipment. The Husky Robot is provided by ClearPath Robotics through their PartnerBot program.

REFERENCES

[1] P. Tokekar, D. Bhadauria, A. Studenski, and V. Isler, "A robotic system for monitoring carp in minnesota lakes," *Journal of Field Robotics*, vol. 27, no. 6, pp. 779–789, 2010.

[2] W. Chin and S. Ntafos, "The zookeeper route problem," *Information Sciences: an International Journal*, vol. 63, no. 3, pp. 245–259, 1992.

[3] J. S. Mitchell, "A constant-factor approximation algorithm for tsp with pairwise-disjoint connected neighborhoods in the plane," in *SoCG '10: Proceedings of the 2010 annual symposium on Computational geometry*. New York, NY, USA: ACM, 2010, pp. 183–191.

[4] N. Christofides, "Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem." CARNEGIE-MELLON UNIV PITTSBURGH PA MANAGEMENT SCIENCES RESEARCH GROUP, Tech. Rep., 1976.