Auton Robot (2016) 40:1123–1137
DOI 10.1007/s10514-015-9502-8

Human–robot planning and learning for marine data collection

Thane Somers[1] · Geoffrey A. Hollinger[1]

Received: 8 December 2014 / Accepted: 15 September 2015 / Published online: 3 October 2015

Abstract We propose an integrated learning and planning framework that leverages knowledge from a human user along with prior information about the environment to generate trajectories for scientific data collection in marine environments. The proposed framework combines principles from probabilistic planning with nonparametric uncertainty modeling to refine trajectories for execution by autonomous vehicles. These trajectories are informed by a utility function learned from the human operator's implicit preferences using a modified coactive learning algorithm. The resulting techniques allow for user-specified trajectories to be modified for reduced risk of collision and increased reliability. We test our approach in two marine monitoring domains and show that the proposed framework mimics human-planned trajectories while also reducing the risk of operation. This work provides insight into the tools necessary for combining human input with vehicle navigation to provide persistent autonomy.

Keywords Marine robotics · Adaptive planning · Informative path planning · Coactive learning

1 Introduction

We envision the future of scientific data collection as a collaborative endeavor between human scientists and autonomous robotic systems. High-impact examples include autonomous underwater vehicles assisting oceanographers to track biological phenomena (Smith et al. 2010), aerial vehicles providing imagery of changing ecosystems (Bryson et al. 2010), and ground vehicles monitoring volcanic activity (Nooner and Chadwick 2009). In these example domains, a key challenge lies in combining the expert knowledge of the scientist with the optimization capabilities of the autonomous system. The scientist brings specialized knowledge and experience to the table, while the autonomous system is capable of processing and evaluating large quantities of data. Leveraging these complementary strengths requires the development of collaborative systems capable of guiding long-term scientific data collection.

When robotic vehicles collaborate with humans, true autonomy relies on the robot having a clear understanding of the goals it uses and the tradeoffs it faces when making decisions. When a robot is assisting a human, the robot's goals must often mimic those of the human. This is particularly true in planning trajectories for underwater robots performing scientific monitoring. The robot must autonomously navigate the environment while maintaining the same goals as a human scientist. While propeller- and buoyancy-driven autonomous underwater vehicles (AUVs) are able to operate in aquatic environments for long time scales, true persistent autonomy requires them to be able to plan and replan their trajectories to fulfill the mission needs of the scientist without human intervention.

When planning trajectories for marine robotic missions, a scientist implicitly balances several environmental variables such as the risk of collision, the uncertainty in ocean currents, and the locations of points of interest. While existing planning algorithms can account for all of these variables, it is difficult to learn the correct tradeoffs among them (Silver et al. 2010). In this work, we create a framework that allows

✉ Thane Somers
somersth@oregonstate.edu

Geoffrey A. Hollinger
geoff.hollinger@oregonstate.edu

[1] Robotics Program, School of Mechanical, Industrial & Manufacturing Engineering, College of Engineering, Oregon State University, Corvallis, OR 97331, USA

an AUV to learn a human path planner's weighting of the variables involved in choosing a trajectory. The AUV then uses that weighting to plan paths mimicking those planned by the human. In this way, we can create an autonomous system that generalizes to different problems while still capturing the scientist's expert knowledge and experience.

The main novelty of this paper lies in its unified planning and learning framework that generates reliable and safe trajectories for autonomous marine vehicles based on human input. To our knowledge, this is the first work to combine Bayesian learning, probabilistic path planning, and user-generated trajectories in a unified approach. We present a novel algorithm that accounts for the variability in the quality of input provided by the human. In addition to simulations in ocean monitoring domains, we perform field trials on an EcoMapper AUV operating in an inland lake to demonstrate that our framework is robust and efficient enough for a human expert to easily use in real time.

These results demonstrate the advantage of combining uncertainty models with human preference learning for long-term marine monitoring missions. Our methods allow the robot to autonomously plan safe trajectories that meet a human operator's personal goals without further human intervention. With these techniques, a robot and scientist are able to function as a team through a shared autonomy framework. Using shared autonomy reduces the burden on the human operators, allowing complex, long-term monitoring missions without the need for continuous human involvement. We believe that solving this challenge is key in making long-term autonomy achievable.

The remainder of this paper is organized as follows. We first discuss related work in motion planning, learning, and human–robot interaction (Sect. 2). We then describe the proposed human–robot planning architecture (Sect. 3) and detail our trajectory refinement (Sect. 3.1) and coactive learning (Sect. 3.2) algorithms. In the next section, we present a number of data-driven simulations to show the benefit of the proposed approach (Sect. 4). Next, we present the results of field trials done on an AUV using our algorithm (Sect. 5). Finally, we draw conclusions and discuss avenues for future work (Sect. 6).

2 Related work

This paper draws on a large body of prior literature in robotic motion planning, learning, and human–robot interaction. We will now discuss related work in these three subareas and highlight the need for a unified architecture.

Motion and path planning are fundamental problems in robotics and have been studied extensively in the past two decades (Latombe 1991; LaValle 2006). Increasing the robot's degrees of freedom or the dimensionality of the environment typically causes an exponential increase in the computation required to solve the planning problem optimally. Thus, motion and path planning problems are generally computationally hard (NP-hard or PSPACE-hard) (Reif 1979). Modern planning methods have focused on the generation of approximate plans with limited computation [e.g., RRT* algorithms (Karaman and Frazzoli 2011)]. Our work extends these ideas to domains where human–robot collaboration is beneficial for the generation of high-quality plans.

A key component of our work is the generation of confidence measures for the reliability of the robot's trajectory. Prior work has examined the development of such measures for various environmental processes using straightforward statistical machine learning tools (Willmott et al. 1985) as well as more sophisticated Bayesian models (Lermusiaux 2006). To provide confidence measures on the prediction of uncertainty in scientific data collection domains, we propose utilizing Gaussian Process (GP) regression (Rasmussen and Williams 2006) augmented with an alternative measure of uncertainty based on the interpolation variance (Yamamoto 2000). Such approaches have been used successfully to improve the accuracy of such tasks as underwater navigation (Hollinger et al. 2013, 2012) and aerial vehicle surveillance (Kim et al. 2013). However, these prior approaches have not integrated human input into the learning and planning frameworks.

Much of the previous work on solving the problem of enabling robots to learn from human demonstration has focused on finding ways for the robot to effectively mimic the human. Researchers have studied a variety of problems such as planning driving trajectories (Ratliff 2009) and autonomous helicopter flight (Abbeel et al. 2010). However, most learning from demonstration problems assume that the expert is providing optimal feedback, which is often impossible to achieve. For example, when solving informative path planning problems (Hollinger and Sukhatme 2014; Krause and Guestrin 2011), humans cannot easily find the optimal path, but they can quickly choose which paths they prefer. In our work, we account for this limitation on human performance by allowing the human to merely present a preference for a solution. The optimal solution is never needed.

Several forms of coactive learning algorithms with theoretical bounds have been studied. These include regret bounds on the perceptron coactive learning algorithm (Shivaswamy and Joachims 2012) and cost bounds on the cost-sensitive perceptron and passive–aggressive algorithms (Goetschalckx et al. 2014). However, these bounds still assume optimal or locally optimal feedback and have not been tested directly with human experts.

Much of the work done on coactive learning algorithms has studied problems where both the expert and the learner are computer programs which solve and improve the solution using different methods (Goetschalckx et al. 2014; Shivaswamy and Joachims 2012). A few studies of using the coactive learning algorithm online with humans have been made, most notably in learning trajectories for robotic arms (Jain et al. 2013). They show that the robot can successfully learn from the human's iterative, suboptimal improvements and that the coactive algorithm performs better than other learning algorithms. Our algorithm builds upon these ideas by providing increased resistance to human error and shorter learning times.

Our work also builds on prior studies in imitation learning that construct cost maps from human-operator example paths (Ratliff 2009; Silver et al. 2010). These prior studies utilize maximum margin planning techniques to learn cost maps from user input. Prior work has focused on surveillance problems where the goal is to generate a safe path through a hazardous environment. Our proposed framework expands on these ideas by incorporating the human scientist into the planning loop.

This work also relates to work in adaptive sampling algorithms. These algorithms attempt to choose path goals that maximize information gain, minimize prediction uncertainty, or minimize risk (Thompson et al. 2010; Low et al. 2009 while minimizing the cost of performing the tour. Methods for doing this adaptively include the maximum entropy and maximum mutual information measures, used for planning in environments modeled as Gaussian Processes (Cao et al. 2013). Hoang et al. (2014) propose a non-myopic active learning framework using Gaussian Processes that jointly optimizes the exploration–exploitation tradeoff. In these algorithms, there is an implicit tradeoff between the various goals. Our work allows the robot to easily learn those tradeoffs from human users, particularly with respect to the importance of each environmental feature. These adaptive algorithms then closely match the performance expected by the human operator while operating autonomously in the field.

There has been a recent push in environmental monitoring towards the development of Decision Support Systems that allow the human operator to seamlessly track the progress of autonomous vehicles and to issue commands on the fly (Das et al. 2011; Li et al. 2006; Sattar and Dudek 2011). Such systems are capable of monitoring the progress of autonomous vehicles operating in the ocean and in other unstructured environments by providing data to scientists in real time. Our work complements these systems by generating suggestions for alternative paths in addition to useful passive data.

Preliminary versions of our algorithms were presented in our prior workshop and conference papers (Somers and Hollinger 2014, 2015; Hollinger and Sukhatme 2014). This paper combines coactive learning with trajectory optimization in a unified framework and provides additional simulations and field experiments.

3 Human–robot architecture

The workflow of the proposed architecture is as follows: (1) a human operator specifies a series of waypoints for a vehicle to gather scientific data, (2) the waypoints are refined by the system to suggest alternative trajectories that have lower risk of collision, and (3) the human operator chooses the desired path. Figure 1 gives a visualization of the proposed architecture.

The trajectory refinements that the system suggests are based on an estimate of the operator's importance weighting of the environmental features that are used to determine the operator's planned trajectory. This importance weighting allows the system to suggest trajectories that match the operator's desired goals more closely. Using a coactive learning algorithm, our system learns these operator preferences based on a set of improvements to simulated environmental maps. These learned preferences are then used to inform the trajectory refinement algorithm of the type of trajectories desired.

The problems that we will examine under this framework comprise of the following components: a trajectory of scientist-provided waypoints that indirectly specify the quality of information gathered, a "risk" map that gives the expected safety of operating in a particular area, and a model of the environment that determines how reliably the autonomous vehicle can move between points. Figure 2 gives an example of the necessary maps in an oceanographic monitoring domain where an autonomous underwater vehicle is surveying a number of ecological hotspots [e.g., harmful algal blooms (Das et al. 2010)]. The waypoints specified by

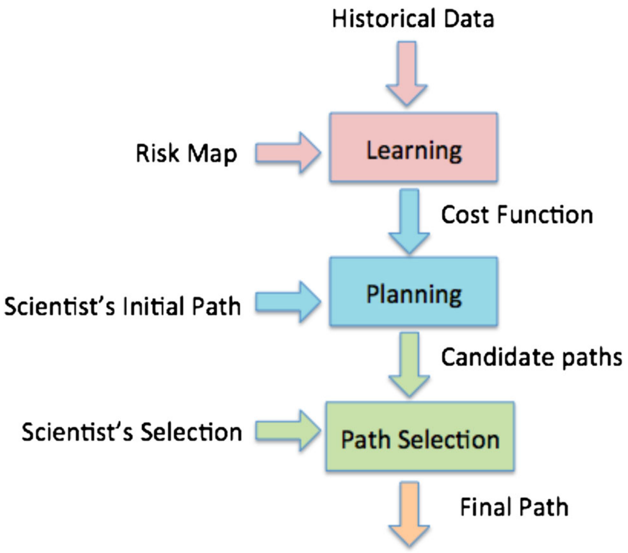

Fig. 1 Proposed framework for human–robot collaboration to generate safe and informative paths for autonomous vehicles to gather scientific data

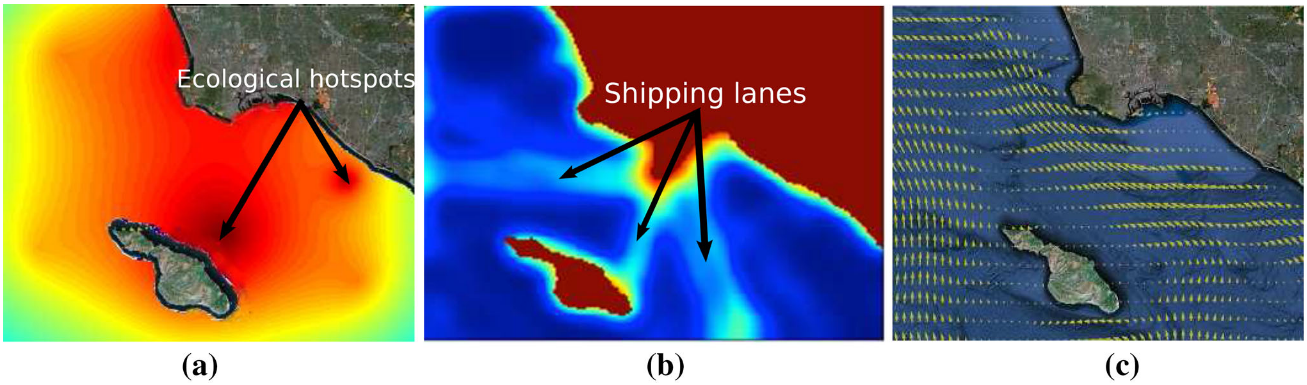

Fig. 2 Three maps that must be combined to perform efficient information gathering in an ocean monitoring scenario where an AUV is tracking a harmful algal bloom. The ocean currents affect the planned path of the vehicle, the risk map determines the safety of operation, and the pre-specified waypoints that the scientist provides gives the benefit of the information gathered. Our proposed system integrates these maps to improve scientific data collection. **a** The scientist provides waypoints that specify ecological hotspots. **b** Collision probability with shipping lanes and land determines risk of operation. **c** Strength of ocean currents provides reliability of operation

the scientist provide areas of high algal bloom density, the risk map provides the probability of colliding with a ship or running into land, and the prevailing ocean currents provide the reliability of operation. We note that all of these maps are uncertain in the sense that the quantities are not known exactly at the time of planning. For reliable operation, it is necessary to predict the values of the information, risk, and reliability maps, as well as the uncertainty in those values.

3.1 Trajectory refinement algorithms

We first discuss our proposed approach for incorporating input from a human scientist into an optimization framework. We will build on ideas originally presented in prior work to learn cost maps that guide autonomous ground reconnaissance vehicles (Ratliff 2009). Unlike prior work, our methods will incorporate a measure of risk into the planning and learning framework. We will also incorporate measures of uncertainty into our predictions.

The formal problem is to plan a path ξ that is a solution to the following optimization problem:

$$\xi^* = \underset{\xi \in \Psi}{\operatorname{argmin}} \; R(\xi) + \alpha D(\xi, \xi_0), \tag{1}$$

where $D(\xi, \xi_0)$ is the deviation from the scientist's initial trajectory of waypoints ξ_0, $R(\xi)$ is an expected risk of executing ξ, Ψ the space of all possible paths, and α is a weighting parameter. We assume that we are given an example trajectory of waypoints ξ_0 from the human operator and that an explicit cost function is not provided as part of the trajectory. The α parameter that matches the operator's preferences is determined using the coactive learning procedure described in Sect. 3.2. After finding approximate solutions to the above optimization problem, the resulting trajectory is presented to the operator for final evaluation. The operator then has the option to adjust the values of α to make the trajectories deviate more or less from the initial plan.

There are several properties of the above problem that make it difficult to solve optimally. If the risk function is non-convex, optimizing it will typically be NP-hard for any rich space of paths (LaValle 2006). In addition to the inherent complexity of the path planning problem, the functions D and R may be computationally intensive to compute. Furthermore, the deviation and risk functions may not be definitively known in advance (e.g., risk is only known with some certainty), and it may become necessary to estimate their expected values based on a distribution of possibilities. Similarly, for a given path, it may not be certain that the vehicle can execute the path exactly, which adds an additional level of uncertainty to the optimization.

Successfully addressing these challenges and optimizing the vehicle's paths requires the development of both uncertainty modeling and planning solutions. We will now describe how the proposed architecture addresses each of these subproblems.

3.1.1 Modeling uncertainty

A key component of our proposed work is to provide a principled estimate of uncertainty for predictions of the vehicle's actions. These uncertainty estimates will be incorporated through probabilistic planning to provide the final suggested paths for the vehicle. We propose using non-parametric Bayesian Regression in the form of Gaussian Processes (GPs) to provide measures of uncertainty (Rasmussen and Williams 2006). We will now discuss background in GPs and show how we can use similar ideas to develop novel representations of uncertainty. This formulation closely follows our prior work in uncertainty modeling for ocean currents (Hollinger et al.

2013). For this uncertainty modeling approach, we assume that estimates of the disturbances (such as ocean currents) are available [e.g., through regional ocean modeling systems and satellite data (Shchepetkin and McWilliams 2005)].

The disturbances at a given latitude lat, longitude lon, and time t can be written as a tuple $\mathbf{c}(lat, lon, t) = (u, v)$, where u and v are scalar values representing components of the disturbance vector along the cardinal axes. At a given time T, we assume we have access to some historical data of the disturbances for times $t = \{T - 1, T - 2, \ldots\}$. Given these data, we want to provide predictions for future points of time as well as confidence bounds for these predictions.

A GP models a noisy process $z_i = f(\mathbf{x}_i) + \varepsilon$, where $z_i \in \mathbb{R}$, $\mathbf{x}_i \in \mathbb{R}^d$, and ε is Gaussian noise. Since the standard GP models a one-dimensional value z_i, we can model the full 2D or 3D space using separate GPs or as a coupled process (e.g., using the techniques in Alvarez and Lawrence (2011)).

We are given some data of the form

$$D = [(\mathbf{x}_1, z_1), (\mathbf{x}_2, z_2), \ldots, (\mathbf{x}_n, z_n)],$$

where $\mathbf{x}_i$ represents a vector of latitude, longitude, and time values for a data point i, and z_i represents a component of the disturbance vector at that point and time. We refer to the $d \times n$ matrix of $\mathbf{x}_i$ vectors as $\mathbf{X}$ and the vector of z_i values as $\mathbf{z}$.

To fully define a GP, we must choose a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ that relates the points in $\mathbf{X}$ to each other. As in our prior work, we utilize a space/time squared exponential kernel to model correlations among the data (Hollinger et al. 2013). Having defined the kernel, combining the covariance values for all points into an $n \times n$ matrix $\mathbf{K}$ and adding a Gaussian observation noise hyperparameter σ_n^2 yields $\mathbf{K}_z = \mathbf{K} + \sigma_n^2\mathbf{I}$. The following equation predicts the mean function value (e.g., a disturbance value along the predicted trajectory) $\mu(\mathbf{x}_*)$ and variance $\mathbb{V}_{gp}(\mathbf{x}_*)$ at a selected point $\mathbf{x}_*$ given the historical and prediction training data:

$$\mu(\mathbf{x}_*) = \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{z}, \tag{2}$$

$$\mathbb{V}_{gp}(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{k}_*, \tag{3}$$

where $\mathbf{k}_*$ is the covariance vector between the selected point $\mathbf{x}_*$ and the training inputs $\mathbf{X}$. This model gives a mean and variance for a particular latitude, longitude, and future time point.

The Gaussian Process variance described above gives an estimate of the uncertainty of a prediction based on the estimated hyperparameters and the sparsity of the data around that point. While the GP variance provides some useful insight into the uncertainty in predictions, it has been shown in prior work that it fails to correlate with the error in complex disturbances (e.g., ocean currents) (Kim et al. 2013; Yamamoto and Monteiro 2008). Based on this work, we instead utilize a method based on the interpolation variance, providing a more informed uncertainty measure. Once a GP has been learned, the interpolation variance can be estimated as

$$\mathbb{V}_{iv}(\mathbf{x}_*) = \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}(\mathbf{z} - \boldsymbol{\mu})^T(\mathbf{z} - \boldsymbol{\mu}), \tag{4}$$

where $\boldsymbol{\mu}$ is a vector of all $\mu(\mathbf{x_i})$ values.

This measure of variance provides a richer representation that accounts for both data sparsity and data variability and while providing improved prediction for the trajectories of AUVs (Hollinger et al. 2013).

3.1.2 Probabilistic planning

The learned uncertainty predictions described above can be incorporated into probabilistic path planners to refine human-provided trajectories. We propose utilizing Monte Carlo Sampling methods to estimate the transition function in a probabilistic model. The planner assumes that the stochasticity in the predictions uses the spatio-temporal variance estimates from the Gaussian Process (either the GP variance or the interpolation variance). These variances are used to generate a distribution of surfacing locations from a set of prior simulations.

This distribution of surfacing locations is obtained by performing a set of Monte Carlo simulations of a glider traveling through the ocean. For each simulation, starting at an initial state s, we choose a waypoint to move towards, which represents taking action a. The ocean currents for each point $\mathbf{x}_*$ are then drawn from the normal distribution centered at $\mu(\mathbf{x}_*)$ with variance $\mathbb{V}_{iv}(\mathbf{x}_*)$ or $\mathbb{V}_{gp}(\mathbf{x}_*)$. The simulation then determines the surfacing location s' based on these ocean current values. Aggregating these trials together, let $M_{s',s,a}$ be the number of samples ending at s', starting at s, and taking action a. Also let $M_{s,a}$ be the total number of samples starting at s, taking action a, and ending in any state. We can generate an estimate of the transition function as $T(s'|s, a) = M_{s',s,a}/M_{s,a}$, which describes the probability of moving to state s' given the choice of taking action a from state s.

The proposed algorithm uses the transition function described above to evaluate a number of candidate plans. The costs of the plans are calculated using a weighting of the risk obtained from the risk map and the deviation from the operator's initial trajectory of waypoints ξ_0 as described in Eq. 1. The algorithm sequentially examines each operator-provided waypoint, then checks all possible alternative waypoints. From each initial waypoint s, the cost values C are calculated for each possible action a that could be taken using the following rule:

$$C(s, a) \leftarrow \sum_{s'} T(s'|s, a)(\Delta D(s', \xi_0) + \alpha \Delta R(s')), \tag{5}$$

where $\Delta D(s', \xi_0)$ is the distance deviation between waypoints caused by adding state s' to the trajectory and $\Delta R(s')$ is the difference in risk incurred by adding state s' to the trajectory. In the domains of interest, the actions in the above equation represent target waypoints. Note that the actual waypoint reached will be different from the target waypoint due to the modeled disturbances. We discretize the possible target waypoints in the environment and then select the action with the lowest expected cost value. This process is then repeated for the remaining waypoints until the entire trajectory has been modified. This modified trajectory is spatially similar to the input trajectory, but it is optimized based on the weighting between the expected distance deviation and risk reduction in the surfacing locations.

Given the appropriate uncertainty measures and the planning methods described above, we now have a framework to modify the waypoints that are provided by the user. This combines the user's intuition with the computer's ability to optimize over large datasets. We would expect the transition models and risk maps to provide improvements in the reliability and safety of the resulting plan. The data-driven simulations in Sect. 4 will confirm this trend. However, in order for the computer to provide these modified trajectories, it must know α, the user's balance between risk and reward.

3.2 Coactive learning algorithm

The trajectory refinement algorithm presented in the preceding section balances risk and deviation from a provided trajectory using a weighting parameter α. This weighting represents the human operator's willingness to trade between the risks an AUV faces and the value of the information it gathers during a mission. In order to present relevant trajectories to the operator, our framework must have an estimate of the operator's implicit weighting. To provide this, we propose an algorithm where the human iteratively refines trajectories given by a computer, which allows us to learn a generalized weighting of risk and reward.

Measuring deviation requires the operator to supply an initial trajectory. In this section, we relax this assumption and consider a more general "reward" map to model the human's intent. We adapt the coactive learning algorithm (Shivaswamy and Joachims 2012) to learning a human expert's preferences when planning paths for underwater scientific data collection. The algorithm attempts to learn the expert's judgment of the utility of a set of paths. This learned utility function, described by α, can then be used to create a refined trajectory mimicking that which the human would have planned.

Unlike the trajectory refinement algorithm, which only considers user preferences based on a single human-provided path, the coactive learning algorithm utilizes multiple, incremental updates to a set of paths. This allows the algorithm to learn a much more generally applicable representation of human preferences based on risk and reward maps. In addition, it does not require the user to input a new trajectory whenever the algorithm runs.

We first present the basic perceptron coactive learning algorithm from prior work. We build upon previous work by adapting the coactive algorithm to noisy environments (Raman et al. 2013) and present a novel approach to dealing with suboptimal updates made by the human expert.

3.2.1 Perceptron coactive learning algorithm

The perceptron coactive learning algorithm attempts to learn an expert's utility function, $U(\langle x, y\rangle) \rightarrow \mathbb{R}$, for judging a candidate solution y for a given problem x (as in Goetschalckx et al. 2014). We assume that the expert's utility function can be approximated as a weighted linear function of features of the candidate solution: $\hat{U}(\langle x, y\rangle) = \mathbf{w}^\top \boldsymbol{\phi}(\langle x, y\rangle)$. These features are simple numerical descriptions of a solution to the task, either concrete or abstracted (e.g., path length, distance to obstacles, probability of failure). However, the set of features used must be rich enough to describe the utility function the human uses to evaluate the task (Abbeel and Ng 2004).

The ultimate goal of the algorithm is to learn the parameters $\mathbf{w}$ that match the expert's method for judging the utility of a solution. This is equivalent to learning the operator's preference weighing, α, from the previous section, as $\alpha = w_1/w_2$, where w_1 and w_2 are the weights of the path deviation and risk features, respectively.

On each update of the coactive learning algorithm, the algorithm creates a candidate solution y_t that maximizes $\mathbf{w}_t^\top \boldsymbol{\phi}(\langle x_t, y_t\rangle)$, based on its current estimate $\hat{U}$ of the expert's utility function. This solution is presented to the expert. The expert has a set of operators, $\mathbb{O}$, that can be applied to the solution to improve it: $\mathbb{O}_i \in \mathbb{O} : \langle x, y\rangle \rightarrow \langle x, y'\rangle$. These operators are specific to the problem domain. In path planning, for instance, these operators could involve altering the trajectory. The cost for the update C_t is equal to the number of operators the expert applies to improve the solution. The learning algorithm then adjusts $\hat{U}$ based on the difference in parameters between y_t and y'.

Algorithm 1 shows how the weights w are updated. If the expert has modified the proposed solution, the difference in parameters $\boldsymbol{\delta}$ between the proposed and modified solution is calculated. This difference is then scaled by the learning rate and added to the previous estimated weights to find the new estimated weights.

In this work, as in most previous research into coactive learning, C_t is simply the number of operators applied. However, there are several ways to make the cost more expressive, particularly for complex domains. For instance, different operators can have different costs, or the cost could vary based on the how the operator is used (e.g., proportional to the

Algorithm 1: CoactiveLearningUpdate (problem x_t, learning algorithm's solution y_t, improved solution y', cost C_t)

```
if $C_t > 0$ then
    $\boldsymbol{\delta}_t := \boldsymbol{\phi}(\langle x_t, y' \rangle) - \boldsymbol{\phi}(\langle x_t, y_t \rangle)$
    $\mathbf{w}_{t+1}^{\top} = \mathbf{w}_t^{\top} + \lambda_t * \boldsymbol{\delta}_t$
end
```

distance a point is moved). Ultimately, the cost should reflect how much effort the human spends in improving a candidate solution, measuring how close the proposed solution is to the humans ideal solution. In our work, this effort decreases as the proposed solutions converge on the target ratios.

Several variations of the coactive learning algorithm have been proposed. Goetschalckx and Tadepalli (2014) examine adjusting the learning rate λ. In addition to the above perceptron (PER) algorithm with with a constant learning rate, they also study a passive-aggressive (PA) algorithm, which adjusts lambda to ensure the solver's most recent mistake is corrected and a cost-sensitive perceptron (CSPER) algorithm where the learning rate is proportional to the number of operators applied.

Assuming that the expert provides a locally optimal solution, they prove an upper bound on the effort required by the expert. With T being the number of update steps, the upper bound is $O(1/\sqrt{T})$ for the PER and PA algorithms and a bound of $O(1/T)$ for the CSPER algorithm. In Shivaswamy and Joachims (2012), a lower bound of $O(1/\sqrt{T})$ on the algorithm's regret is shown, assuming the expert provides an optimal solution.

However, using the algorithm with a human expert breaks these assumptions. The solutions provided by the human are unlikely to be locally optimal and could even be unintentionally misleading. Furthermore, only the human's incorrect updates to a near-optimal solution change the learned weights. This causes the weights to oscillate as update steps are performed. One way to mitigate this issue is to present suboptimal candidate solutions half of the time, allowing the learned weights to be reinforced (Raman et al. 2013). We incorporate the essence of this solution, as our learning algorithm does not create perfect solutions. We further extend it to specifically remove the effect of incorrect or erroneous updates.

3.2.2 Proposed histogram algorithm

The baseline perceptron algorithm is sensitive to suboptimal updates made by the human expert. The coactive learning algorithm assumes that all changes made to the candidate solutions are improvements in the eyes of the human. However, particularly in complex and noisy domains, it is easy for the human to make sub-optimal updates that they believe are improvements. The perceptron algorithm weights all updates equally, and does not attempt to distinguish good updates from poor ones. To overcome this limitation, we developed an alternate algorithm to identify and reduce the effect of these suboptimal updates.

Our algorithm takes all previous improved weights into account when determining the new estimated weights. A histogram of the new and previously improved weights, $\mathbf{w}_t$, for each feature is created. A normal distribution is fitted to the histogram. The center of the normal distribution is taken as the new estimated weight for each respective feature. This histogram identifies the weighting that is mostly likely to be correct based on the distribution of previous weights. Weights that are significantly different than the majority are averaged out, greatly reducing their effect. Thus, this method excludes outliers and prevents new updates from completely changing the estimated weights. In this way, the algorithm is able to continuously converge on the human experts weightings, even when a number of incorrect estimated weights are included.

For a small number of updates it can be difficult to reliably fit a reasonable distribution to the data. In this case the $median(\mathbf{w}^{\top})$ provides a good estimate of the new weights.

4 Simulations and results

We evaluated the components of our learning and planning framework in the context of several different data-driven simulations. We begin by comparing the performance of our histogram-based coactive learning algorithm in estimating a human's planning preferences to the performance of the baseline coactive learning algorithm. This demonstrates the algorithm's success in learning a human's weighting of several environmental variables. We then apply our trajectory optimization framework to a simulated Slocum glider AUV collecting environmental data in the Southern California Bight. The human's preference weighting is used to inform the type of modified trajectories presented to the user by the framework. Finally, we present the results of field trials using the framework on a propeller-driven EcoMapper AUV for monitoring lake ecology.

4.1 Comparative learning algorithm simulations

First, we evaluate the ability of our coactive learning algorithm to estimate a human operator's trajectory planning preferences. The problem we examine consists of several components: a planned trajectory of waypoints, a target vector $\mathbf{w}^{\top}$ of feature weights for the learning algorithm to learn, and maps representing the value of those features in a region. These feature maps could represent real world variables, such as temperature or pH, or abstract features, such as a "risk"

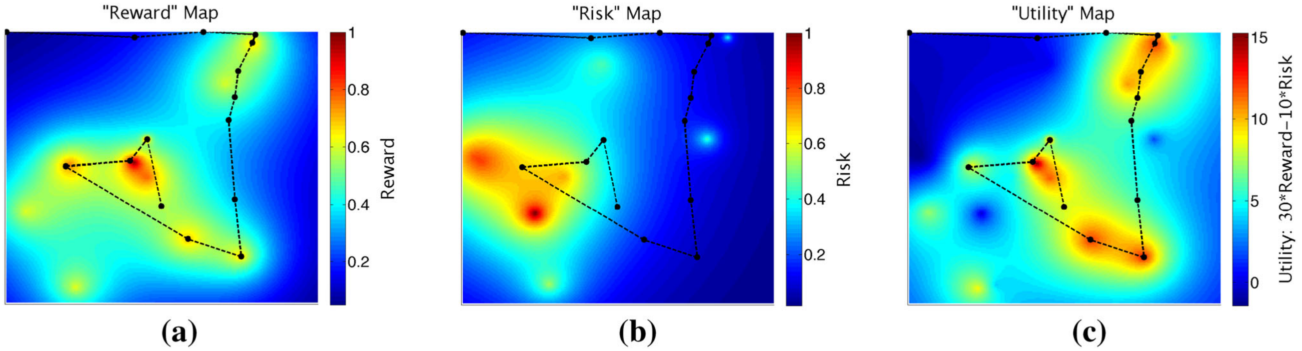

Fig. 3 An example path and utility field generated using the coactive learning algorithm simulations. Only the utility map shown in **c** is presented to the expert during trials. The *black line* represents the robot's path through the environment. Here, risk and reward integrated along the path are the features used in the utility function. The computer proposes a trajectory to the expert, who then improves it. The framework learns the expert's underlying weighting between gathering information and risk using coactive learning. **a** Reward map representing the value of traveling in a particular area. **b** Risk map showing the risk of traveling in a region. **c** Utility map generated from a weighted sum of the risk and reward maps. Here, the target weights of risk and reward are −10 and 30, respectively (Color figure online)

feature representing the cost of traveling in a given region or a "reward" feature that represents the quality and value of information gained by traveling in a given area (Singh et al. 2007, 2009).

For our simulation, as in the trajectory optimization algorithm, we assume that the expert's utility function is linearly composed of two features: the risk the robot incurs and the information it gains during its tour. The total risk and total information for a path are found calculating the line integral of each respective feature map along the path.

To test the algorithm's ability to learn a human expert's weighting, we have a human plan paths over a map created using a predetermined utility function. The expert is presented with a path overlaid on a map of the utility at each location in a region, as shown in Fig. 3c. Maps of risk and reward are generated as a random sum of Gaussians, shown in Fig. 3a and b. The utility map is generated by weighting these risk and reward maps by their respective target weights and summing them. For all tests, we use target weights of −10 and 30, for risk and reward respectively. This represents a stronger preference for gathering information than avoiding risk. The human is shown only the utility map. Since they are optimizing the path based solely on a map of utility calculated using the predetermined target weights, we can test how effectively the coactive algorithm learns these weights without changing the human's propensity for misjudging the utility of a given path.

In our tests, the algorithm used a simple greedy information-gathering path planner to generate candidate paths. First, it finds the peaks of a utility map made from the feature maps weighted by the learned weights. Then, using a locally optimal traveling salesman problem solver (Applegate et al. 2006), it connects the peaks using a path that minimizes the inverse of the utility along the path. Thus, the planner finds a short path while still maximizing the utility of that path.

At each update, the expert improves the path by moving one of the points of the path. Shown only the utility map, they modify the path, attempting to maximize the line integral of the utility along the path. Thus, the expert's utility function in planning matches the target utility function. After each modification, the change in information and risk are calculated from the hidden risk and information maps, and used in the coactive learning update to update the learning algorithm's estimate of the expert's utility function. A new map and path are generated for each coactive update.

We conducted 20 trials each for the baseline perceptron and histogram algorithms. Each trial consisted of performing 16 updates based solely on the provided utility maps. Each update used a different map, with the expert moving one point on each map.

As shown in Fig. 4, the histogram algorithm accumulates regret more slowly than the perceptron coactive learning

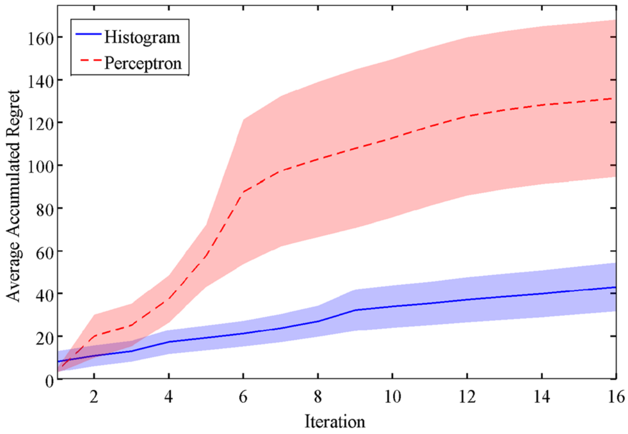

Fig. 4 The regret (sum of deviation between target weight and estimated weight) for each algorithm averaged over 20 trials. The standard error of the mean for the averages is shown. The histogram coactive learning algorithm accumulates regret more slowly (Color figure online)

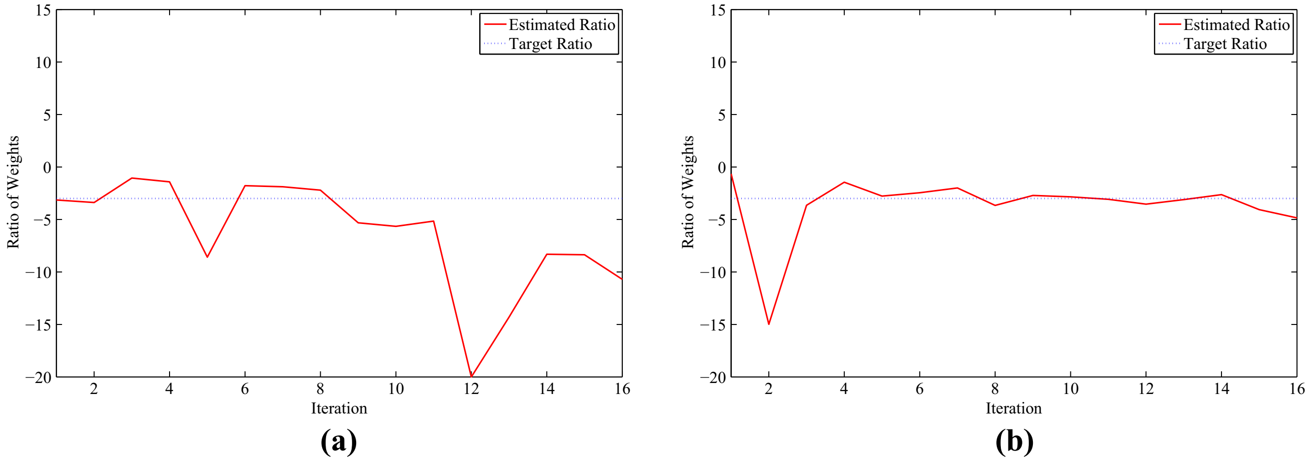

Fig. 5 Example plots of how the estimated weight changes over a trial in simulations of the perceptron and histogram learning algorithms. Note that the perceptron algorithm initially tracks the target weights relatively well until a suboptimal update from the human expert throws it off. Comparatively, the histogram method converges on the target ratio, discounting an initial suboptimal update. **a** An example of the estimated ratio of weights over a trial of the perceptron coactive learning algorithm. **b** An example of the estimated ratio of weights over a trial of the histogram coactive learning algorithm (Color figure online)

algorithm. Additionally, it also smoothly converged towards a set of estimated weights, as each update shifts the histogram only slightly. As seen in Fig. 5, the perceptron algorithm is still highly susceptible to suboptimal updates made by the human, even after many iterations. This is because each update is valued equally and the algorithm cannot compare the current update to previous feedback. However, the histogram algorithm learns what the optimal weighting is and is able to ignore or reduce the effects of suboptimal updates.

In our tests, we noted that, while the operator generally made good modifications to the path, suboptimal updates were still common and the resulting ratios of risk and reward were noisy, as seen in Fig. 5a. While further study is necessary, we hypothesize that this is because humans concentrate on the features at the vertices of the trajectory, instead of examining the complete path.

The coactive learning algorithm provides a fast and effective method for learning an expert's weighting, α, between information gathered and risk incurred. There is no need for the human to tune α through trial and error. This weighting is then used as a parameter in the trajectory refinement algorithm to provide path suggestions that are relevant to the expert's goals.

4.2 Data-driven trajectory refinement simulations

We now present a validation of the proposed trajectory refinement framework in the underwater monitoring domain in the Southern California Bight region where an autonomous underwater vehicle (AUV) is monitoring an oceanographic phenomenon with the help of a scientist. The simulations model a Slocum Glider (Pereira et al. 2013), which is a buoyancy-controlled AUV that moves at a speed of approximately 0.3 m/s. The scientist provides the glider with a series of waypoints, and the vehicle dives between the waypoints while using dead reckoning to determine when to surface. Due to its slow speed, the glider is highly susceptible to ocean currents, and it is often difficult to predict where exactly the glider will surface relative to the specified waypoint. The glider is in danger of running aground if it comes too close to land, and in addition, if the glider surfaces within a shipping lane, it becomes susceptible to collision with passing boat traffic.

The goal of these simulations is to determine the extent to which we can improve the safety of operation using the proposed learning and planning methods. In this domain, safety is measured by the probability of the underwater vehicle successfully completing its mission without coming too close to land or encountering a passing ship. The scientist provides the initial series of waypoints, and the proposed framework is then used to modify these waypoints to decrease the probability of collision while also minimizing the necessary deviation from the initial plan.

The simulations were performed on a single desktop PC with a 3.2 GHz Intel i7 processor and 9 GB of RAM. The simulations incorporate data from ocean currents provided by the JPL Regional Ocean Modeling System (ROMS) (Shchepetkin and McWilliams 2005). The JPL ROMS system provides estimates of the ocean currents but not uncertainty in those estimates. The uncertainty in the ocean currents was determined using the interpolation variance as discussed in Sect. 3.1. The uncertainty learning portion of the proposed method took approximately 5 min to complete, and the plan-

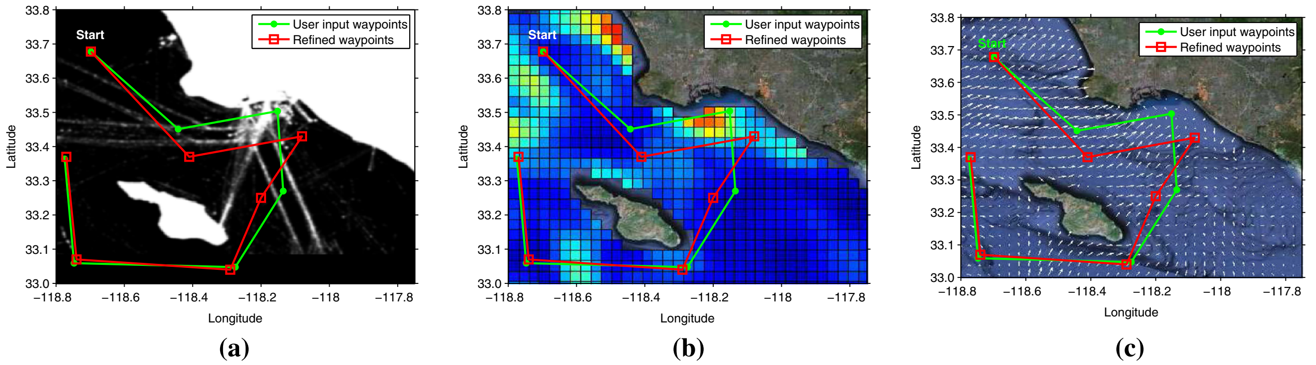

Fig. 6 Example of improving an initial trajectory of waypoints using the proposed learning and planning framework for data from April 24, 2013. The refined path avoids the riskier (*lighter*) areas and also remains in areas where the uncertainty of the ocean currents is low. The resulting path is safer and more reliable without deviating much from the initial waypoints given by the scientist. **a** Initial and refined waypoints overlaid on risk map. *Lighter areas* have higher risk of collision with land or passing ships, **b** Initial and refined waypoints overlaid on ocean current uncertainty map. *Redder areas* denote higher areas of normalized uncertainty. **c** Initial and final trajectory overlaid on ocean current predictions. Vectors denote direction and magnitude of ocean currents (Color figure online)

ning portion completed in less than a second using a 40 × 40 discretized grid of possible waypoint locations. We note that the uncertainty learning portion only needs to be run once per day, and many trajectories can then be refined using those uncertainty estimates.

Risk maps were generated for the simulation using historical Automatic Identification Systems (AIS) shipping data. AIS is a tracking system that mandates a large number of vessels in the United States (and other countries) to broadcast their location information via VHF transceivers (see Pereira et al. 2013 for more details). We used historical AIS data collected over a period of 5 months (between January and May, 2010) in the region 33.25° N–34.13° N and 117.7° and 118.8° W. Using these data, we calculated an aggregate risk value, $R(s)$, at all possible discretized waypoints, s, in the region, which correlates with the chance of hitting a passing ship.

Figure 6 shows an example of how an initial trajectory was modified to reduce risk in the ocean monitoring domain. In this example, the human operator chooses to move the AUV into a risky harbor region to gather data. Using the previously learned α, the algorithm then modifies this trajectory. The incorporation of the learned weighting allows the refined trajectories to balance the operator's level of risk aversion with their desire of gathering data in the harbor. Figure 7 gives a visualization of how changing the weighting value (α) affects the resulting trajectories.

These modified trajectories collect data in the same general areas while simultaneously avoiding the high-risk shipping lanes and the most dangerous areas of the harbor. We also see that the estimates of the ocean currents are more certain in the areas traversed by the modified trajectories (i.e., the modified trajectories move away from the red areas in Fig. 6(b)). Effectively combining these two types of information would require an expert operator capable of processing data in real-time. With the proposed system, the operator can choose the desired trajectory without any underlying knowledge of the risk and then have the system refine it for increased safety and reliability.

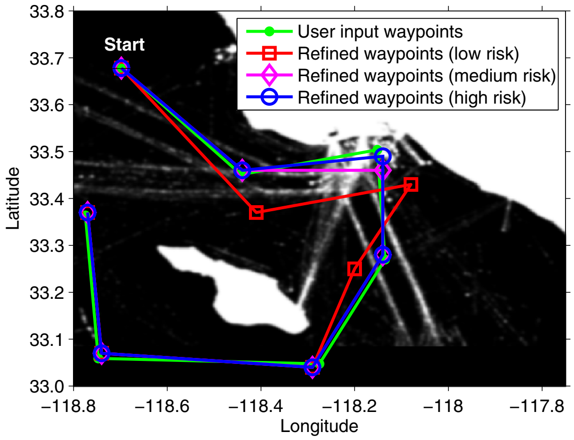

Fig. 7 Initial trajectory and set of suggested trajectories using different weightings between deviation and risk. Higher weighting of risk leads to safer paths at the cost of deviating from the initially specified trajectory (Color figure online)

Finally, we examine the effect changing the magnitude of the learned α parameter has on the modified trajectory. We provide quantitative evaluations of the deviation and risk tradeoff between a high ($\alpha = 1000$) and low ($\alpha = 100$) value of the weighting parameter. In Fig. 8, we see that the proposed method provides trajectories that range from closely tracking the initial trajectory with high risk to loosely following the initial trajectory with lower risk. In some cases we are able to achieve up to a 51 % reduction in risk while deviating from the path by less than 0.8 km.

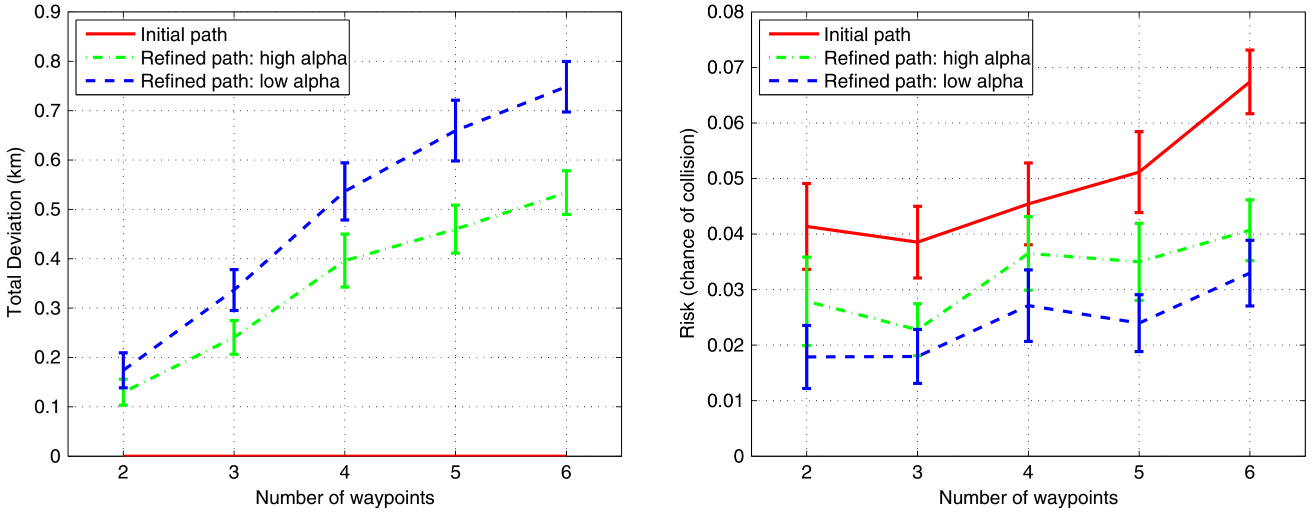

Fig. 8 Deviation and risk for varying weighting parameters for trials of the trajectory optimizer using data from April 24, 2013. The proposed method allows the scientist to trade between the initially selected waypoints and safer paths that deviate from them. Each data point is averaged over 20 user-input trajectories, and *error bars* are one SEM (Color figure online)

5 Field trials

Finally, we demonstrate the framework's capabilities in a lake monitoring environment. While our simulations show that the the framework is able to learn and converge on a target weight and that it can refine trajectories to improve their safety, the goal in these experiments is to demonstrate that the framework was robust enough for use in an integrated field environment. They also show that the resulting feature weights of paths planned by the human and those by the framework were the same in a real world scenario. This demonstrates that the coactive learning algorithm is able effectively learn the human's planning preferences and that these preferences can be incorporated into a path planner to effectively improve upon a human's planning abilities. By combining the human's preferences and goals with a computer's ability to quickly analyze data and plan trajectories, safer, more informative missions can be planned and run with less effort expended by the human operator.

We performed a series of trials with our framework using a YSI EcoMapper autonomous underwater vehicle in a lake ecology monitoring scenario. These propeller-driven AUVs are able to maintain speeds of 2 m/s for up to 10 h. As such, they are often used in ecological monitoring and oceanographic research missions (Ellison and Cook 2009). They have a wide range of sensors. These include water conductivity, temperature, and depth sensors for ecological monitoring and a Doppler Velocity Log and GPS unit for vehicle localization. Missions of waypoints for the AUV to follow are uploaded wirelessly using the standard 802.11 wireless protocol. Our field trials were conducted in an inlet of Puddingstone Reservoir in San Dimas, California (Lat. 34.08, Lon. −117.81).

We trained our algorithm to plan paths based on the water temperatures and water depths along the path. These act as an analog to the risk and information maps used in the simulations. These also closely match a true ecological monitoring mission, where an oceanographer might target a certain combination of environmental features, such as temperature, depth, conductivity, and salinity in order to study a certain organism or ecological phenomenon. For each trial, we began by teaching our preferences to an information-gathering planner using our coactive learning algorithm. Unlike in the simulations, the human was shown both temperature and depth maps. Their true preference weighting between the features was learned. As in Sect. 3.2.1, this preference was represented as a linear utility function comprised of weighted temperature and depth features.

We then ran a dense lawnmower pattern over the inlet to establish a map of water temperature and lake depth for the planning framework to use. Using the learned utility function, the planner than ran the AUV on a path attempting to maximize the utility of the sensed information. Due to the depth of the inlet and to simplify the experiments, the AUV was used on the surface using 2D trajectories.

One limitation of these experiments is that depth and temperature are not independent features. Deeper locations are often colder because it takes longer for solar heating to warm them. As such, it is not always possible to find a path matching a given ratio of depth and temperature. For example, if maximizing depth is weighted highly with minimizing temperature having a smaller weight, it is likely the planned path will appear to satisfy both features equally. Additionally, since historical data for the lake were unavailable, the uncertainty estimation portion of the framework was not used. Even with these limitations, these trials provide a demonstra-

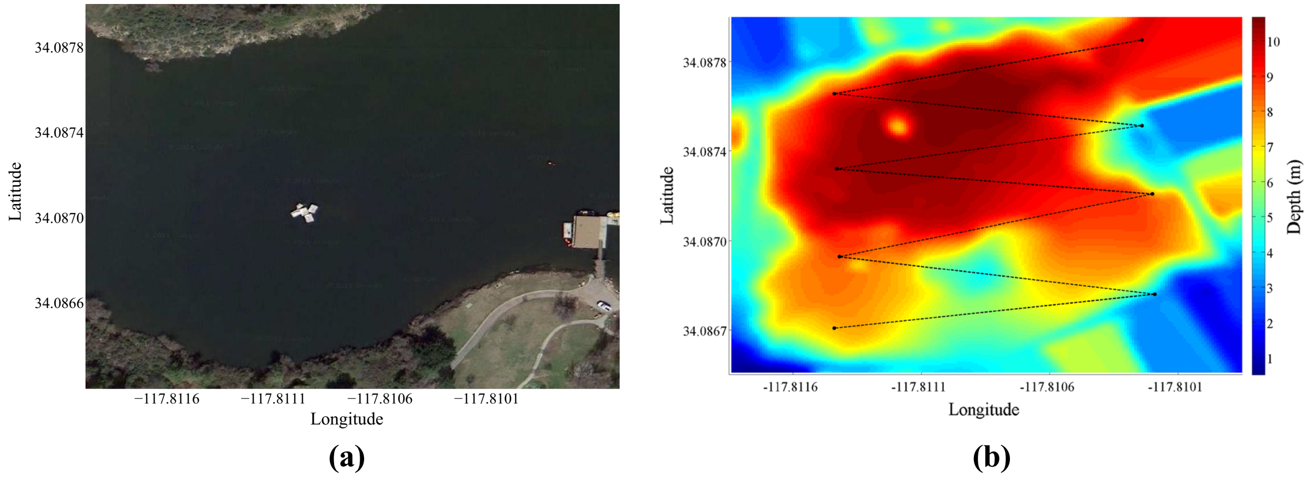

Fig. 9 An aerial view of the test area at Puddingstone Reservoir (*left*) with the corresponding depth map (*right*). A simple baseline lawnmower path is shown over the depth map. The lawnmower path is a commonly used path for guiding AUVs in ecological monitoring missions (Mora et al. 2013). The depth map was created by interpolating between depth points measured on a dense survey of the area (Color figure online)

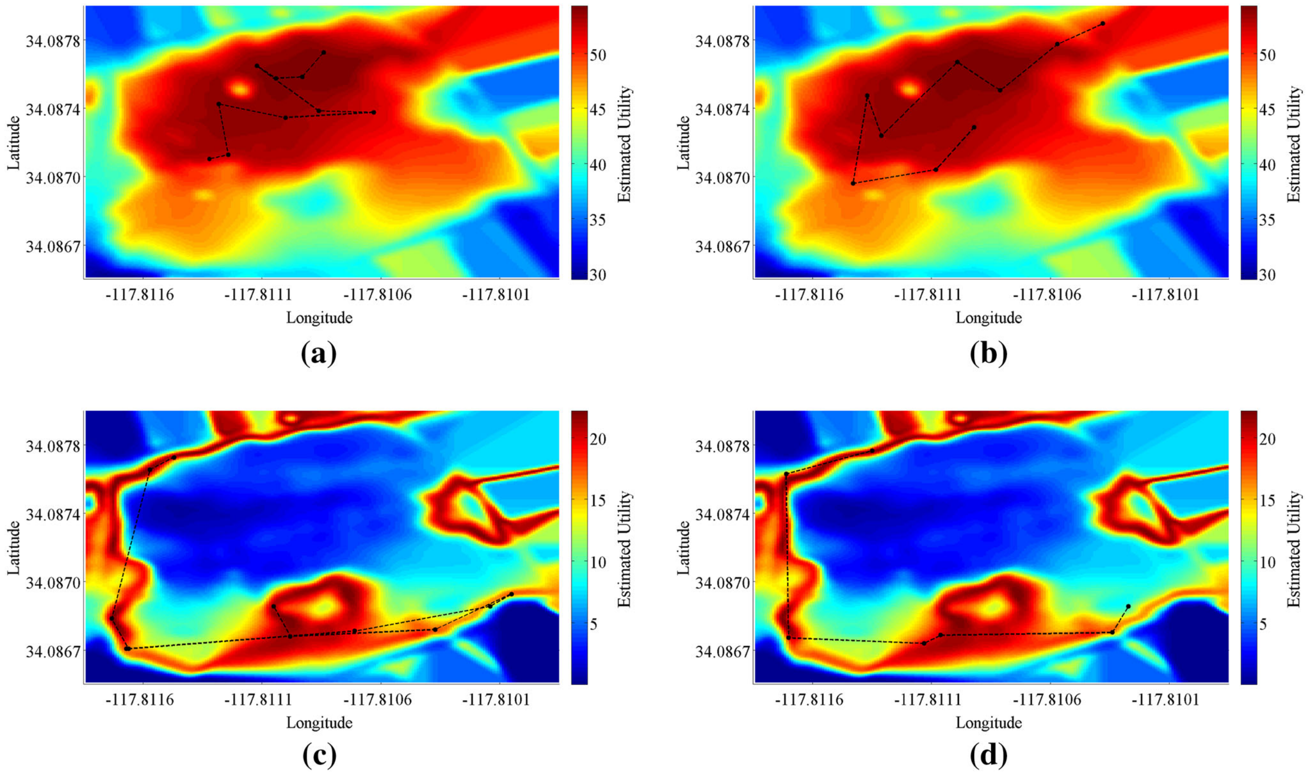

Fig. 10 **a** A trajectory planned by our proposed algorithm maximizing the depth while minimizing the measured temperature. **b** A human-planned trajectory maximizing the depth while minimizing the measured temperature. **c** A trajectory planned by our proposed algorithm targeting a depth of 6 m and a temperature of 27 °C. **d** A human-planned trajectory targeting a depth of 6 m and a temperature of 27 °C (Color figure online)

tion of the value of incorporating preference learning into a planning framework.

We began by running a loose lawnmower pattern, as shown in Fig. 9. These types of trajectories are often used on ecological monitoring missions as they are easy to set up, so they formed a relevant baseline for our tests. The resulting ratio of depth to temperature integrated along the path was 1.157.

We then taught the algorithm to strongly maximize the depth while minimizing the temperature and ran the mission shown in Fig. 10a. The measured ratio of 2.48 closely matched the learned ratio of 2.47. It also closely matched the measured ratio from the human-planned mission in Fig. 10b of 2.45.

We also taught the algorithm to target specific depths and temperatures. We targeted lake depths close to 6 m and temperatures very close to 27 °C, weighting the depth more strongly. We again ran a computer-planned and a human-planned mission, shown in Fig. 10c and d. The algorithm learned a weight ratio of 2.56. The measured utility ratios were 1.564 for the computer's path and 1.495 for the human's path. While these are not exactly the same, they are still quite close. Additionally, the achievable ratio was limited by the correlation of the temperature and depth in the lake.

Finally, we tried to train the algorithm to follow the 6 m depth contour by strongly preferring paths at that depth while ignoring all other features. The algorithm learned a weight of 19.34 for the utility of sampling a 6 m depth and a weight of only 1.97 for the utility gained from sampling a 27.7 °C temperature, giving a learned ratio of 9.81. The measured ratio of depth to temperature was 1.078, similar to the human-planned ratio of 1.205. Again, the learned ratio was not achievable due to the correlation of depth and temperature. Due to the high importance weight placed on measuring points at a 6 m depth, the algorithm chose points with a depth of 6 m, successfully planning a route that closely follows the 6 m depth contour line (Fig. 11).

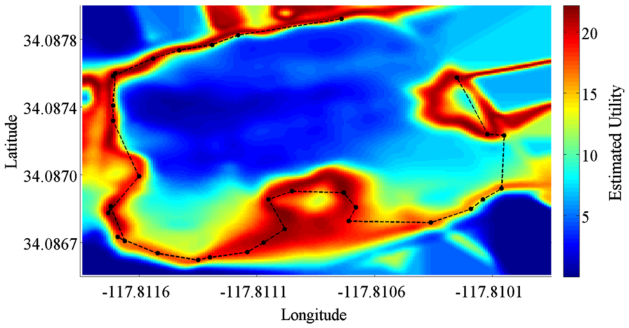

Fig. 11 A contour-following path generated using our coactive learning algorithm. The algorithm was trained by consistently preferring paths around 6 m of depth. It learned a weight of 19.34 for depths around 6 m while ignoring other depths and temperature features with weights of less than 2 (Color figure online)

For each trial, we compared the ratio of the temperature and depth features sensed along the path for the human- and framework-planned paths to the learned weights. The results are summarized in Table 1. While there was a small amount of variability due to inaccuracies in following the planned path and changing water temperatures, the ratios matched well. This shows that the framework is able to autonomously plan paths that follow the same preferences as a human's.

These results show the framework's benefit in marine data gathering scenarios. The robot is able to quickly learn a human operator's goals and preferences, then autonomously plan trajectories that match these goals and preferences without further human intervention.

6 Conclusion and future directions

The results in this paper have shown that it is possible to combine waypoints provided by a human operator with historical data to improve the operation of autonomous vehicles in scientific monitoring scenarios. We have proposed Bayesian learning techniques that allow for uncertainty in predictions to be incorporated into the final trajectory, and we have integrated these uncertainty estimates into a probabilistic planning framework. We also successfully integrated coactive learning algorithms into the trajectory optimization framework, allowing it to learn and mimic a human expert's priorities. Using probabilistic techniques, our modified coactive learning algorithm gracefully handles imperfect updates made by the human.

The resulting framework allows for reduced risk of collision for an autonomous glider performing an ocean monitoring task with input from a human operator. By integrating feedback from the user into an algorithmic planning framework, we have effectively improved the safety and reliability of autonomous vehicle operation. The effectiveness of the framework has been shown in two simulations. In the first, we found that the algorithm's estimated weights converge on a set of target weights in a reasonable amount of time for use with a human expert. In field trials, we showed

Table 1 Results of the lake ecology monitoring field trials

Experimental trial	Learned ratio	Actual ratio (human-planned)	Actual ratio (framework-planned)
Maximize depth, minimize temp	2.47	2.45	2.48
Target 6 m depth and 27 °C	2.56	1.495	1.564
Target 6 m depth contour	9.81	1.205	1.078

For each trial, a human operator taught our framework a preference between temperature and depth and planned a path based on that preference. The resulting measured depth to temperature ratios for the human-planned and framework-planned trajectories are shown. In each trial, the ratios measured match closely that the framework is able to use these learned weights to plan paths matching the performance of those planned by a human operator. The second data-driven simulation showed the effectiveness of the framework in making slight modifications to a trajectory to greatly increase the safety of the planned route.

The framework presented in this paper opens up a number of avenues for future work at the intersection of human–robot interaction and autonomous path planning. Further work includes testing the algorithm on a range of human experts to comprehensively evaluate the use of coactive learning for learning human preferences. Other path parameters should be added in order to more closely match the human's intentions and account for possible path parameters. We hope to be able to learn a human's preferences in trajectory planning without complete knowledge of the underlying parameters used.

An area for future work lies in combining the preference learning and risk modeling techniques presented in this paper with modern adaptive sampling methods. Our learning and refinement methods allow the human and robot to work together to meet a common goal through shared autonomy. Combining this capability with adaptive sampling and planning methods would allow marine vehicles to continuously monitor and adapt to their environments in a long-term mission while pursuing the same goals as a human. Thus, marine robots would have a level of persistent autonomy, allowing them to safely complete long, complex marine data collection missions.

One promising avenue is the development of lifelong learning approaches that allow trajectories specified by the human operator to be stored to improve trajectory generation for future plans. In addition, it may be beneficial to change the ordering of the waypoints to improve the safety of the trajectory. The incorporation of re-ordering into our framework is fairly straightforward; however, it would require additional metrics to determine the deviation penalty from the scientist's original path. Ultimately, we believe that techniques like the one proposed here will improve the efficiency of scientific data collection and allow human–robot teams to gather data safely and persistently in challenging environments.

Acknowledgments The authors thank Robby Goetschalckx, Alan Fern, and Prasad Tadepalli from Oregon State University for their insightful comments. Further thanks go to Gaurav Sukhatme from the University of Southern California for providing access to the Ecomapper vehicle in the field experiments. This work was supported in part by the following Grants: NSF IIS-1317815.

References

Abbeel, P., Coates, A., & Ng, A. Y. (2010). Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, *29*(13), 1608–1639.

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In: *Proceedings of the International Conference on Machine Learning*, pp. 1–8.

Alvarez, M., & Lawrence, N. (2011). Computationally efficient convolved multiple output gaussian processes. *The Journal of Machine Learning Research*, *12*, 1459–1500.

Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). *The traveling salesman problem: A computational study*. Princeton: Princeton University Press. doi:10.1109/ICICTA.2009.96.

Bryson, M., Reid, A., Ramos, F., & Sukkarieh, S. (2010). Airborne vision-based mapping and classification of large farmland environments. *Journal of Field Robotics*, *27*(5), 632–655.

Cao, N., Low, K. H., & Dolan, J. M. (2013). Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 7–14.

Das, J., Maughan, T., McCann, M., Godin, M., O'Reilly, T., Messie, M., Bahr, F., Gomes, K., Py, F., Bellingham, J. G., Sukhatme, G. S., & Rajan, K. (2011). Towards mixed-initiative, multi-robot field experiments: Design, deployment, and lessons learned. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3132–3139.

Das, J., Rajany, K., Frolovy, S., Pyy, F., Ryany, J., Caron, D. A., & Sukhatme, G. S. (2010). Towards marine bloom trajectory prediction for AUV mission planning. In *IEEE International Conference on Robotics and Automation*, pp. 4784–4790.

Ellison, R., & Cook, M. (2009). Cost-effective automated water quality monitoring systems providing high-resolution data in near realtime. In: *World Environmental and Water Resources Congress*, pp. 1–10.

Goetschalckx, R., Fern, A., & Tadepalli, P. (2014). Coactive learning for locally optimal problem solving. In *AAAI Conference on Artificial Intelligence*, pp. 1824–1830.

Hoang, T. N., Low, K. H., Jaillet, P., & Kankanhalli, M. (2014). Nonmyopic epsilon-Bayes-optimal active learning of Gaussian processes.

Hollinger, G. A., Englot, B., Hover, F. S., Mitra, U., & Sukhatme, G. S. (2012). Active planning for underwater inspection and the benefit of adaptivity. *The International Journal of Robotics Research*, *32*(1), 3–18.

Hollinger, G. A., Pereira, A. A., & Sukhatme, G. S. (2013). Learning uncertainty models for reliable operation of autonomous underwater vehicles. In *IEEE International Conference on Robotics and Automation*, pp. 5593–5599.

Hollinger, G. A., & Sukhatme, G. S. (2014). Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, *33*(9), 1271–1287.

Hollinger, G. A., & Sukhatme, G. S. (2014). Trajectory learning for human-robot scientific data collection. In *IEEE International Conference on Robotics and Automation*, pp. 6600–6605.

Jain, A., Wojcik, B., Joachims, T., & Saxena, A. (2013). Learning trajectory preferences for manipulators via iterative improvement. In *Advances in Neural Information Processing Systems*, pp. 575–583.

Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, *30*, 846–894.

Kim, Y. H., Shell, D. A., Ho, C., & Saripalli, S. (2013). Spatial Interpolation for robotic sampling: Uncertainty with two models of variance. *International Symposium on Experimental Robotics*, *88*, 759–774.

Krause, A., & Guestrin, C. (2011). Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology*, *2*(4), 32:1–32:20.

Latombe, J. C. (1991). *Robot motion planning*. Boston, ma: Kluwer Academic Publishers.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge: Cambridge University Press.

Lermusiaux, P. F. (2006). Uncertainty estimation and prediction for interdisciplinary ocean dynamics. *Journal of Computational Physics*, *217*(1), 176–199.

Li, P., Chao, Y., Vu, Q., Li, Z., Farrara, J., Zhang, H., & Wang, X. (2006). OurOcean—an integrated solution to Ocean monitoring and forecasting. In: *OCEANS*, pp. 1–6.

Low, K. H., Dolan, J. M., & Khosla, P. K. (2009). Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proceedings of the International Conference on Automated Planning and Scheduling*, pp. 233–240.

Mora, A., Ho, C., & Saripalli, S. (2013). Analysis of adaptive sampling techniques for underwater vehicles. *Autonomous Robots*, *35*(2), 111–122.

Nooner, S. L., & Chadwick, W. W. (2009). Volcanic inflation measured in the caldera of axial seamount: Implications for magma supply and future eruptions. *Geochemistry, Geophysics, Geosystems*, *10*(2), 1–14.

Pereira, A. A., Binney, J., Hollinger, G. A., & Sukhatme, G. S. (2013). Risk-aware path planning for autonomous underwater vehicles using predictive ocean models. *Journal of Field Robotics*, *30*(5), 741–762.

Raman, K., Joachims, T., Shivaswamy, P., & Schnabel, T. (2013). Stable coactive learning via perturbation. *Proceedings of the International Conference on Machine Learning*, *28*, 837–845.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge: The MIT Press.

Ratliff, N. D. (2009). Learning to search: Structured prediction techniques for imitation learning. Ph.D. Thesis, Carnegie Mellon University.

Reif, J. H. (1979). Complexity of the mover's problem and generalizations. In *Symposium on Foundations of Computer Science*, pp. 421–427.

Sattar, J., & Dudek, G. (2011). Towards quantitative modeling of task confirmations in human-robot dialog. In *IEEE International Conference on Robotics and Automation*, pp. 1957–1963.

Shchepetkin, A. F., & McWilliams, J. C. (2005). The regional oceanic modeling system (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, *9*(4), 347–404.

Shivaswamy, P., & Joachims, T. (2012). Online structured prediction via coactive learning. *Proceedings of the International Conference on Machine Learning* pp. 1–8.

Silver, D., Bagnell, J. A., & Stentz, A. (2010). Learning from demonstration for autonomous navigation in complex unstructured terrain. *The International Journal of Robotics Research*, *29*(12), 1565–1592.

Singh, A., Krause, A., Guestrin, C., Kaiser, W. J., & Batalin, M. A. (2007). Efficient planning of informative paths for multiple robots. *International Joint Conference on Artificial Intelligence*, *7*, 2204–2211.

Singh, A., Krause, A., & Kaiser, W. J. (2009). Nonmyopic adaptive informative path planning for multiple robots. *International Joint Conference on Artificial Intelligence* pp. 1843–1850.

Smith, R., Das, J., Heidarsson, H., Pereira, A., Arrichiello, F., Cetnic, I., et al. (2010). USC CINAPS builds bridges: Observing and monitoring the Southern California bight. *IEEE Robotics & Automation Magazine*, *17*(1), 20–30.

Somers, T., & Hollinger, G. A. (2014). Coactive learning with a human expert for robotic monitoring. *Workshop on Robotic Monitoring at Robotics Science and Systems*, *2014*, 1–2.

Somers, T., & Hollinger, G. A. (2015). Coactive learning with a human expert for robotic information gathering. In *IEEE International Conference on Robotics and Automation*, pp. 559–564.

Thompson, D. R., Chien, S., Yi Chao, Li, P., Cahill, B., Levin, J., Schofield, O., Balasuriya, A., Petillo, S., Arrott, M., & Meisinger, M. (2010). Spatiotemporal path planning in strong, dynamic, uncertain currents. In *IEEE International Conference on Robotics and Automation*, pp. 4778–4783.

Willmott, C. J., Ackleson, S. G., Davis, R. E., Feddema, J. J., Klink, K. M., Legates, D. R., et al. (1985). Statistics for the evaluation and comparison of models. *Journal of Geophysical Research*, *90*(C5), 8995–9005.

Yamamoto, J. K. (2000). An alternative measure of the reliability of ordinary Kriging estimates. *Mathematical Geology*, *32*(4), 489–509.

Yamamoto, J. K., & Monteiro, M. (2008). Properties and applications of the interpolation variance associated with ordinary kriging estimates. In: *International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, pp. 70–77.

Thane Somers is a graduate student currently working towards a Ph.D degree in Robotics at Oregon State University in Corvallis, Oregon. In 2013 he graduated with a Bachelor of Science degree in Mechanical Engineering degree from Washington University in St. Louis. His research interests include planning under uncertainty, underwater manipulation, and human–robot interaction for marine robotics.

Geoffrey A. Hollinger is an Assistant Professor in the School of Mechanical, Industrial & Manufacturing Engineering at Oregon State University. His current research interests are in planning, decision making, and learning for robotic systems in marine, air, and ground environments. He has served as PI and co-PI on research grants from ONR, NSF, USDA, and NASA. His past research includes networked marine robotics at the University of Southern California, multi-robot search at Carnegie Mellon University, personal robotics at Intel Research Pittsburgh, active estimation at the University of Pennsylvania's GRASP Laboratory, and miniature inspection robots for the Space Shuttle at NASA's Marshall Space Flight Center. He received his Ph.D. (2010) and M.S. (2007) in Robotics from Carnegie Mellon University and his B.S. in General Engineering along with his B.A. in Philosophy from Swarthmore College (2005). He has been a Guest Editor for the Autonomous Robots journal, Associate Editor for the ICRA and IROS conferences, and Area Chair for the RSS conference.