

CoTrack: A Framework for Tracking Dynamic Features with Static and Mobile Sensors

Thanh Dang*, Nirupama Bulusu*, Wu-chi Feng*, Sergey Frolov†, and António Baptista‡

*Portland State University, OR, 97201, USA. (dangtx,nbulusu,wuchi)@cs.pdx.edu

†Monterey Bay Aquarium Research Institute, CA, 95039, USA. frolovs@mbari.org

‡Oregon Health & Science University, OR, 97006, USA. baptista@stccmop.org

Abstract—Current feature tracking frameworks in sensor networks exploit advantages of either mobility, where mobile sensors can provide micro scale information of a small sensing area or of numerical models that can provide macro scale information of the environment but not both. With the continual development of sensor networks, mobility becomes an important feature to integrate next generation sensing systems. In addition, recent advances in environmental modeling also allow us to better understand basic behavior of the environment. In order to further improve existing sensing systems, we need a new framework that can take advantage of existing fixed sensor networks, mobile sensors and numerical models. We develop *CoTrack*, a *Collaborative Tracking* framework, that allows mobile sensors to cooperate with fixed sensors and numerical models to accurately track dynamic features in an environment. The key innovation in *CoTrack* is the incorporation of numerical models at different scales along with sensor measurements to guide mobile sensors for tracking. The framework includes three components: a macro model for large-scale estimation, a micro model for locale estimation of specific features based on sensor measurements, and an adaptive sampling scheme that guides mobile sensors to accurately track dynamic features. We apply our framework to track salinity intrusion in the Columbia River estuary in Oregon, United States. Our framework is fast and can reduce tracking error by more than 50% compared to existing data assimilation frameworks and state-of-the-art numerical models.

I. INTRODUCTION

As sensor networking technologies evolve, the ability to provide larger scale, better reliability, and longer-term service continues to improve. With all these advances, however, there is still a gap between the needs of some scientists and the ability of the sensor system to provide the data that they need. One such example includes environmental scientists who want to understand both small scale (< 1km) and large scale (> tens of km) environmental factors. The key problem they face is that they would like to understand particular dynamic physical phenomena that, while isolated to a relatively small area at a particular time, may need to be studied over a very large geographic region.

While deploying a relatively dense sensor network throughout the system being studied would be nice, the sheer area to be covered makes it impractical and cost prohibitive. As a result, environmental scientists have implemented and deployed very, very sparse and static sensor networks in order to provide some basic coverage of the area being studied and used complex data assimilation models with the resulting sensor data [1]. While this is good for understanding macro-scale systems (eg. average water volume or velocity), they provide limited information about small-scale phenomena (eg. exact salinity intrusion front location) that can impact the macro-scale system. To help improve this situation, one could deploy a number of mobile sensing systems such as unmanned underwater vehicles to capture data in selected areas to improve sensing within a region. Given the

dynamic nature of large-scale environmental systems, figuring out where to place such vehicles becomes hugely problematic.

To provide the best sensing data to the application in such scenarios, we believe that the best solution might be the careful integration of (i) sparse and static sensor nodes to provide basic macro-scale data, (ii) data models that assimilate the sparse sensor data into a basic understanding of where phenomena of interest may be occurring, (iii) mobile sensing platforms that can move to areas of interest, and (iv) feedback driven data assimilation to refine the movement of the sensors. In such a way, the data assimilation process can provide the basic macro-scale parameters which it is better at doing, and the mobile sensor can provide specific point data, which it is better at doing.

In this paper, we develop *CoTrack*, a Collaborative Tracking framework, for estimating and tracking dynamic features in an underwater environment using fixed and mobile sensors, and numerical models. The key innovation in *CoTrack* is the incorporation of numerical models at different scales with granularity of location ranging from 100 meters to 0.5 meters, and sensor measurements to guide mobile sensors for tracking. We apply our framework to track salinity intrusion in the Columbia River Estuary in Oregon. In short, the contributions of this paper are the following:

- A framework for building macro models for specific features (Section III-A). Unlike previous approaches, we address the case where general models for the environment are available. From these models and empirical data, we develop specific models for features of interest. Our model reduces the estimation error by 27% and the framework for building model can be adapted for other features.
- A framework for building micro models estimating distances to a feature from sensor measurements (Section III-B). The model estimates the true feature location from mobile sensor measurements. This model can reduce the estimation error of the macro model by 37%.
- An adaptive sampling scheme to improve tracking performance (Section III-C). Based on the two models above, we develop an iterative adaptive sampling scheme for mobile sensors. The idea is to use macro models to guide a mobile sensor where to take its first measurement and use micro models to iteratively refine the location of the feature. By combining these two models, we can reduce the total error by more than 50% compared to existing frameworks.

II. PROBLEM STATEMENT

Let G be a finite set of points in space modeling the environment. For each point $\omega \in G$, let $p_{\omega,i} = (p_{\omega,i}^1, p_{\omega,i}^2, \dots, p_{\omega,i}^j, \dots, p_{\omega,i}^k)$ be a tuple of k physical parameters such as temperature, water velocity, or salinity level associated with ω at time step i . j is the parameter index.

Definition 1: A wedge of parameter p^j at threshold u at time i is a set $\Omega = \{\omega \in G | p_{\omega,i}^j = u\}$.

For example, a salinity wedge at 15 practical salinity unit (psu) is the set of points that have salinity level equal to 15 psu. We can track a wedge by estimating its relative location or *distance* to a landmark with known location. For many real scenarios, the wedge shape and location change over time. Hence, it is difficult to quantify the distance between two wedges or the distance from a wedge to a landmark. To simplify the tracking problem, we often estimate the wedge along a transect, whose location is also known. For example, Figure 1 shows the Columbia river estuary and a transect illustrated by the dashed line. Figure 2 shows the salinity level along the transect at different times of day. The salinity level ranges from 0 psu (fresh water) to 32 psu (sea water). The salinity wedge at 15 psu is about 15km, 18km, and 20km from the mouth of the estuary at 1:00am, 6:00am, and 12:00pm respectively.

Let d be the distance from the intersect of the true wedge and the transect to a landmark and \hat{d} be the estimation of d . The subscript i in d_i , \hat{d}_i and other notations is to indicate the parameter at the corresponding time step i . Let n be the total number of time steps and $M = \{m_1, m_2, \dots, m_n\}$ is the set of known physical parameters of the environment such as atmospheric pressure, wind velocity, and time of day and sensor measurements. The tracking wedge problem can be stated as

$$\text{Problem 1: Given } M, \min_{\hat{d}} RMSE = \sqrt{\frac{\sum_{i=1}^n (d_i - \hat{d}_i)^2}{n}}$$

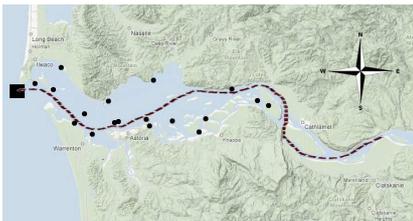


Fig. 1. Transect in the Columbia river estuary, Oregon, U.S. The model overlays on top of the physical map of the river estuary. The dashed line is the transect. The dots are existing fixed sensor stations. The square indicates the beginning of the transect.

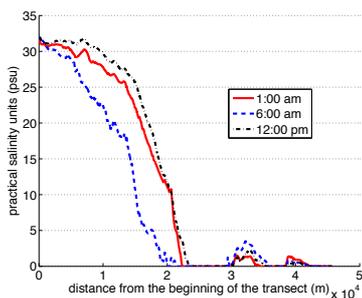


Fig. 2. Salinity levels along the transect at different times of day.

Table I lists the notation that we use through out the paper. In addition, we make the following assumptions:

- At every time step, the physical parameters of the environment, also called *forcings*, are available from global ocean models and weather models. In addition, the measurements from fixed sensor stations are also available via some means of communication.
- Mobile sensors can communicate with a base station and with each other via a real-time telemetry network. Each mobile sensor has certain computation power. In our case study, computation power equivalent to a laptop is reasonable.

| NOTATION | MEANING | REFERENCE |
|-------------------------|---|----------------|
| d | distance from a point on the true wedge to a landmark | definition 1 |
| \hat{d} | estimation of d | problem 1 |
| e | error defined as $d - \hat{d}$ | equation 3 |
| i | time step | definition 1 |
| n | total number of time steps | definition 1 |
| u | parameter threshold | definition 1 |
| u^r | real measurement | section III-B2 |
| δu | difference between real measurement and the threshold | equation 4 |
| $\mu, \delta\mu$ | first regression coefficients | equation 1, 4 |
| β, α, γ | vector of regression coefficients | equation 1, 4 |
| m | vector of inputs | equation 1, 4 |
| ϵ, ω | random variables | equation 1, 4 |

TABLE I
NOTATION

- Mobile sensors can travel to a defined location accurately. Current cruise ships are equipped with GPS and can estimate their location within meters of accuracy.

III. CoTRACK FRAMEWORK OVERVIEW

Instead of developing a specific physical model for the wedge, we decide to develop a statistical model for several reasons. First, the detailed physical relationship between external parameters and the physical process is complex and still not well understood. Second, statistical modeling does not require expert knowledge of a specific field. Hence, we can apply our framework for tracking to a broad range of applications. Finally, in many cases [2], a statistical model performs better than a physical model in terms of estimation error because physical modeling often seek to understand the relationship between physical parameters and how the process works rather than minimizing estimation error.

Based on the above advantages, we develop the CoTrack framework with three main components (Figure 3). The first component is a *macro model* that estimates the wedge location at a large scale from existing information such as time, environmental forcings, and fixed sensor measurements. This model might have a high estimation error but it helps us to quickly zoom into the area where the wedge location can be. The second component is a *micro model* that estimates the distance to the true wedge from mobile locations where the measurements are taken. This model estimates wedge location at a much smaller scale compared to the macro model because it assumes that the measurements are taken at locations estimated from the macro model and hence are nearby the true wedge location. The final component is an *adaptive sampling* scheme that incorporates both macro and micro models to guide mobile sensors. At the beginning, mobile sensors go to the locations estimated by the macro model and take measurements. The mobile sensors should form a non-collinear shape to increase tracking accuracy. Priyantha *et al.* [3] address the problem of finding the best formation of mobile sensors to improve tracking performance. The mobile sensors then use the micro model to calculate the offset between their current locations and the true wedge location. They can exchange their estimations to find the best matched location. The process can be iterative until the estimation is stable or the difference between the sensor measurements and the expected threshold is negligible.

In the following sections, we will describe the three components; macro model, micro model, and adaptive sampling.

A. Macro Model for Tracking Wedge

1) *Regression Model*: Let d be the distance from the intersect between the wedge at a certain threshold (eg. 15 psu) and the transect

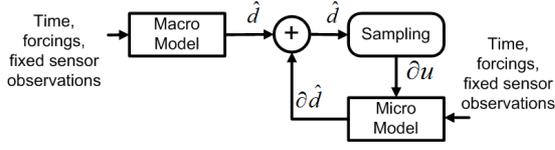


Fig. 3. CoTrack Overview. *Macro model* quickly zooms into the area where the wedge is. *Micro model* refines the estimation at a higher accuracy. *Adaptive sampling* iteratively guides mobile sensors based on estimations from macro model and micro model to track the wedge.

to the beginning of the transect (marked as the square in Figure 1) and m is a length p vector of inputs including time of day, day of month, forcings and observations and their combinations. We can describe the relationship between d and m by using the following linear model

Macro model

$$d = \mu + m \times \beta + \epsilon \quad (1)$$

where

- μ is the first regression coefficient, which is usually estimated as the average of d ,
- β is a vector of regression coefficients, and
- ϵ is a zero mean random variable representing error in the model.

m is usually called the predictor variable and d is called the response. If we observe n pairs of responses d_i and predictor variables m_i , the model can be written as

$$d_i = \mu + m_i \times \beta + \epsilon_i. \quad (2)$$

The goal of regression is to find μ and β that minimize the overall error ϵ . Although the model is linear in the coefficients, the predictor variables can be nonlinear such as square or cube or combination of variables. We actually consider predictor variables up to the 6th polynomial order and their combinations. Once we find μ and β , we can fit a distance \hat{d}_i from inputs m_i as

$$\hat{d}_i = \mu + m_i \times \beta. \quad (3)$$

The difference between the true distance d_i and fitted distance \hat{d}_i , $e_i = d_i - \hat{d}_i$, is the residual.

Standard techniques [2] such as multicollinearity handling, model selection, model checking and evaluation, and cross validation are applied to build the model and examine if the model is good.

We have described the model for predicting the wedge from fixed sensor measurements and other physical parameters. A mobile sensor can travel to the predicted location and take measurements to improve the next prediction. In the next section, we will describe the micro model that incorporates new mobile sensor measurements to refine where the wedge is.

B. Micro Model For Tracking Wedge From Measurements

1) *Measurement Types*: Several possible measurement types are used in practice. *Binary measurements* only indicate whether an event is present or not. In our case, they can be whether the current location is the salinity wedge or not. *Location measurements* contain location information of an event. In our case, they can be the locations of the salinity wedges. However, the location often cannot be measured directly and can only be inferred from other information. *Scalar measurements* contain values of some physical parameters. In our case, they can be the salinity, temperature or water velocity. These measurements can be obtained directly from sensors. *Combined measurements* contain a combination of some physical parameters. For example, Sanford et al. [4] have developed a sensor device that can measure integrated salinity across the entire water column.

We choose to build the model using scalar measurements for several reasons. Binary measurements can be inferred directly from scalar measurements and often contain less information. Location measurements are unavailable and are actually what we would like to predict. Combined measurements are promising, but new sensors have not been deployed widely yet. In particular, we develop the model that predicts a nearby wedge from measurements recorded from mobile sensors.

2) *Regression Model*: Let u be the threshold of the wedge y and u^r be the real measurement from a mobile sensor. Let $\delta u = u - u^r$ be the difference between the threshold or the expected value and the real measurement. Let δd be the distance from the mobile sensor to the true wedge. We can describe the relationship between δd and δu using the following model

$$\delta d = \delta \mu + \delta u \times \alpha + m \times \gamma + \omega \quad (4)$$

where

- m is the vector of inputs (predictor variables) including time of day, day of month, forcings, and observations
- α and γ are regression coefficients, and
- ω is a zero mean random vector representing error.

The model is for δu within a vicinity of 1000 m since the *macro model* has already predicted the wedge within a reasonable accuracy. As in section III-B, we also remove multicollinearity, use stepwise model selection and check the model [2].

C. Adaptive Sampling

Based on the two models, we develop an adaptive sampling algorithm to track wedges. First, we use the macro model to predict roughly where the wedge is and let the mobile sensor go there and take a measurement. Second, we use the micro model to refine where the true wedge is from the mobile sensor measurements. The process can be iterative by simultaneously taking measurements and using the micro model to further refine the wedge location. Table II describes the algorithm.

```

Predict Location Use macro model to calculate  $\hat{d}_i$ 
Take a measurement  $u_i^r$  at  $\hat{d}_i$ 
for  $i = 1$  to  $MAX\_ITERATIONS$  do
  Predict Offset Use micro model to calculate  $\delta\hat{d}_i$ 
  Update  $\hat{d}_i = \hat{d}_i + \delta\hat{d}_i$ 
  if  $\delta u_i \leq u_{threshold}$  or  $\delta\hat{d}_i < \delta d_{threshold}$  then
    exit
  else
    Take a measurement  $u_i^r$  at  $\hat{d}_i$ 
     $\delta u_i = u_i^r - u_i$ 
  end if
end for

```

TABLE II

COTRACK ADAPTIVE SAMPLING ALGORITHM

IV. CASE STUDY: TRACKING SALINITY INTRUSION IN THE COLUMBIA RIVER ESTUARY

CORIE is a pilot environmental observation and forecasting system (EOFS) for the Columbia River estuary and the Eastern North Pacific ocean. CORIE integrates a real-time sensor network, a data management system and advanced numerical models. The goal of CORIE is to characterize and predict complex circulation and mixing processes in a system encompassing the lower river, the estuary and the near-ocean. The CORIE observation network includes an extensive array of 24 stations in the Columbia River estuary and the nearby coastal ocean. At each station, in-situ sensors measure physical properties of water or atmosphere. Data assimilation combines observational data with numerical data models to produce an estimated system state for the physical process. Mobile sensors such as research vessels are used to dynamically collect interesting data and track features, such as water flow, temperature map, and salinity wedges.

Figure 1 shows the physical layout of the Columbia river estuary. Due to tidal forces, the sea water intrudes into the river estuary daily. Understanding the intrusion behavior will help us better monitor the impact of natural and human activities on the estuary ecosystem. Our goal is to track the salinity intrusion wedges at various levels along the transect. The range of salinity measurement in *practical salinity unit* (psu) is from 32 psu (sea water) to 0 psu (fresh water). In this case study, d is the distance from the beginning of the transect to the intersect between the salinity wedge and the transect.

A. Experimental Design and Evaluation

1) *Ground Truth*: One of the challenges in designing the experiments is lack of ground truth due to the large scale environment in the Columbia River estuary. To create the ground truth for evaluation, we use an existing numerical model [5] and the state of the art data assimilation framework [1] to estimate the system state and consider it as the truth. The system state is for the whole Columbia River estuary from 15-May-2004 to 29-Jul-2004 [6]. This system state is then used to generate measurements based on real sensor models. The generated measurements and the real forcings are used to develop the macro and micro models.

2) *Experiment Setup*: Figure 4 shows the main components of the experiment setup.

- A physical model ELCIRC [7] was used to simulate the *ground truth* including true wedge distance d and measurements m as described in section IV-A1.
- A physical model SELFE [5](with different formulation of the model numerics) was used to generate the first guess d^- of the wedge location. In practice, the model was approximated using a nonlinear model surrogate.
- Existing data assimilation framework [1] estimates the wedge d^{DA} from the measurements m and the first SELFE guess d^- .
- CoTrack independently estimates the wedge $d^{CoTrack}$ from the measurements m .
- The two estimates are then compared with the true wedge location d using RMSE and MAE metrics.

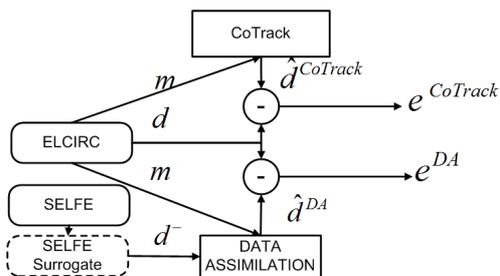


Fig. 4. Experiment setup

For the macro model, we have 74 days of hindcast data with observations from fixed sensor stations every 0.5 hour. We randomly select one week - 7 days - and leave it out for evaluation. The rest of the data are used for regression. We first consider all the predictor variables up to the 6th polynomial order and their interactions. We realize that there are many redundant predictor variables that are unnecessary and also cause multicollinearity, which may make the model unstable numerically. We decide to apply stepwise regression for selecting a model. Stepwise regression starts with a single most significant predictor variable and adds more variables if they significantly improve the model or remove variables otherwise. The regression procedure is performed on Matlab with the Matlab statistics toolbox.

Similarly for the micro model, from each salt wedge location, we consider 200 nearby points and use the ground truth to generate the measurements and the relative distance from the salt wedge. We

choose only 200 points because we assume that a measurement is taken within a vicinity of maximum 1000m the the wedge as the previous model has estimated the salt wedge with a RMSE of 400m. Due to a large number of points, we independently and randomly select 5000 entries for regression and 5000 entries for evaluation. We also decide to apply stepwise regression for selecting a model. Stepwise regression starts with a single most significant predictor variable and adds more variables if they significantly improve the model or remove variables otherwise. The regression procedure is performed on Matlab with the Matlab statistics toolbox.

To ensure that the model is adequate, we examine the normality and independence properties of the residual error. If the residual error is independent Gaussian noise, we find a good model because we cannot extract more information from the noise.

We compare the performance of existing numerical models [5], the state-of-the-art data assimilation framework [1], and CoTrack in estimating the salt wedge at various thresholds from 32 psu (sea water) to 0 psu (fresh water). Both frameworks use one mobile sensor. In addition, we also compare the performance of the adaptive sampling scheme using the macro model and multiple refinement iteration using the micro model. All implementation is done in Matlab. A preliminary version of the code is available at

<http://sys.cs.pdx.edu/home/projects/cotrack>.

B. Results

1) *Macro Model*: There are 660 predictor variables included in the model. It has an RMSE = 500m. The R-square is 98%, meaning the model can explain 98% of the variations in the responses based on the inputs. d in this case study is the distance from the salinity wedge to the beginning of the transect. The model can be rewritten as

$$d = 2.36 \times 10^{12} + m \times \beta + \epsilon \quad (5)$$

where the first coefficient 2.36×10^{12} and β are found during regression, and

- m is the vector of 660 predictor variables including time of day, forcings, sensor observations and their combinations. These variables are available from weather stations.
- β is a vector of regression coefficients, and
- ϵ is a random variable with the distribution is approximately $N(0, 500^2)$.

To evaluate if the model we found was good, we inspected the residual plot, normality plot, and the autocorrelation of the residuals and found that the residuals had a normal distribution but there were dependencies among them. Unfortunately, we tried to fit an auto regressive model and found that we could not eliminate this dependency. Further inspection into the residuals may be able to build a better model where the residuals are completely independent.

2) *Micro Model*: Although the total number of predictor variables is 120, only 24 variables (time of day and some forcing parameters) are included in the model. It has an RMSE = 200m. The R-square is 80%, meaning the model can explain 80% of the variations in the responses based on the inputs. The model can be rewritten as

$$\delta d = 19.7 - \delta u \times 354.8 + m \times \gamma + \omega \quad (6)$$

where

- m is the vector of 24 predictor variables including time of day and forcings. These variables are available from weather stations.
- γ are regression coefficients which are found during regression, and
- ω is a random variable with a distribution approximately $N(0, 200^2)$.

To evaluate whether the micro model we found is good, we inspect the residual plot, normality plot, and the autocorrelation of

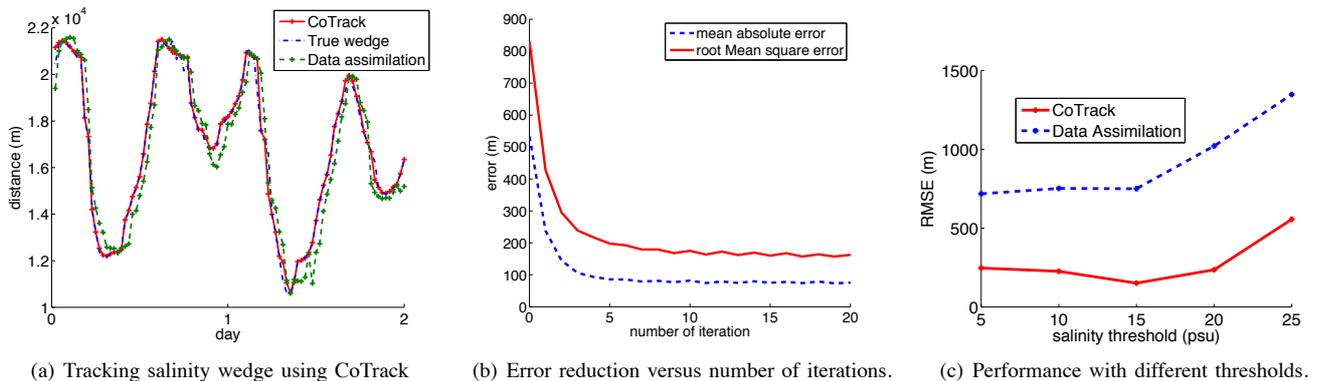


Fig. 5. a) Tracking salinity wedge using adaptive sampling. Using CoTrack, we can follow the salinity wedge most of the time. b) Error reduction versus number of iterations for tracking salinity wedge at 10 psu. The error degrades quickly and becomes stable after 6 iterations. c) Tracking performance of CoTrack and existing data assimilation framework at various thresholds. CoTrack can reduce the RMSE by half compared to the existing data assimilation framework.

the residuals. We find that the residuals are independent although they do not strictly follow a normal distribution.

3) CoTrack Adaptive Sampling:

Error reduction using CoTrack. CoTrack can improve tracking performance compared to existing models and data assimilation framework. Figure 5(a) plots the true salinity wedge at 15 psu and the estimates using data assimilation and CoTrack adaptive sampling for 7 days. CoTrack can estimate the salinity wedge well most of the time. The comparison among different frameworks is shown in Table II. In terms of RMSE, SELFE model has the worst RMSE of more than 6km. Existing data assimilation framework can estimate the salinity wedge with a RMSE = 550m. CoTrack can reduce the error to 200m.

| Metric | SELFE | Data Assimilation surrogate model | CoTrack |
|---------------------|-------|-----------------------------------|---------|
| RMSE (m) | 6362 | 550 | 200 |
| MAE (m) | 4553 | 450 | 155 |
| Processing Time (s) | 0.02 | 25 | 0.2 |

TABLE II
TRACKING PERFORMANCE COMPARISON.

Processing time. We evaluate the three frameworks on a Lenovo X61 laptop with Intel(R) Core(TM)2 Duo 1.6 Ghz CPU, 2G RAM, and running Windows Vista Business under a normal load. Running SELFE model takes about 0.02 seconds, which is the fastest. While CoTrack takes only 0.2 seconds to process the inputs, the data assimilation takes about 25 seconds.

Error reduction versus number of iterations. Figure 5(b) shows the estimation error versus the number of iterations in CoTrack adaptive sampling. The RMSE degrades quickly and becomes stable after about 6 iterations. This results suggest that only about 6 iterations in adaptive sampling are able to provide a good estimation of the salinity wedge.

Performance at difference salinity thresholds. Figure 5(c) compares the existing data assimilation framework with CoTrack in tracking wedges at different thresholds. CoTrack can reduce the RMSE by half compared to data assimilation. Interestingly, CoTrack seems to work best with thresholds from 10 to 20 psu. The performance gets worse for salinity levels close to sea water or fresh water. For thresholds that are close to 32 psu and 0 psu, the physical process is much more dynamic than at 15 psu. In addition, the framework might also suffer from the boundary problem where we do not have adequate empirical data to regress a good model.

V. CONCLUSION AND FUTURE WORK

We have described CoTrack, a collaborative tracking framework that incorporates existing fixed sensor networks with mobile sensors and numerical models to track dynamic features in a large scale environment. The key idea is to use a macro model to quickly locate a feature's vicinity and a micro model to iteratively refine the feature location. CoTrack has three main components: a macro model that estimates features' locations at a large scale within 100 meters, a micro model that estimates the offset between mobile sensors' locations and the features' locations at a small scale within 0.5 meters, and an adaptive sampling scheme that uses the two models to iteratively refine the estimation. We apply CoTrack to track salinity intrusion in the Columbia river estuary in Oregon. CoTrack is fast and can reduce the error by more than 50% compared to the state-of-the-art data assimilation framework. This improvement promises a significant contribution in understanding and improving existing physical models as well as the impacts of natural and human activities in the river estuary ecosystem.

Acknowledgements The research described in this paper was supported by National Science Foundation grants NSF 0424602 and NSF 0747442.

REFERENCES

- [1] S. Frolov, A. M. Baptista, Y. Zhang, and C. Seaton, "Estimation of ecologically significant circulation features of the columbia river estuary and plume using a reduced-dimension kalman filter," *Continental Shelf Research*, vol. 9, no. 2, pp. 456–466, 2009.
- [2] B. Abraham and J. Ledolter, *Introduction to Regression Modeling*. Belmont, CA, USA: Thomson Brooks cole, 2006.
- [3] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-assisted localization in wireless sensor networks," in *Proceedings of the 24th Conference on Computer Communications 2005, INFOCOM 2005*, Miami, FL, March 2005, pp. 172–183.
- [4] A. M. Baptista. (2009, July) Remote sensing of salinity intrusions. *emoteSensingofSalinityIntrusions.pdf*. [Online]. Available: <http://www.stccmop.org/files/RemoteSensingofSalinityIntrusions.pdf>
- [5] Y. Zhang and A. M. Baptista, "Selfe: A semi-implicit eulerianlagrangian finite-element model for cross-scale ocean circulation," *Ocean Modelling*, vol. 21, no. 3-4, pp. 71–96, 2008.
- [6] A. M. Baptista. (2009, July) Center for coastal margin observation and prediction (cmop) data center. *baseframe.html*. [Online]. Available: <http://www.stccmop.org/datamart>
- [7] Y. Zhang, A. M. Baptista, and E. P. M. III, "A cross-scale model for 3d baroclinic circulation in estuaryplumes shelf systems: I. formulation and skill assessment," *Continental Shelf Research*, vol. 24, no. 1, pp. 2187–2214, 2004.