

Environmental Boundary Tracking and Estimation Using Multiple Autonomous Vehicles

Zhipu Jin and Andrea L. Bertozzi

Abstract—In this paper, we develop a framework for environmental boundary tracking and estimation by considering the boundary as a hidden Markov model (HMM) with separated observations collected from multiple sensing vehicles. For each vehicle, a tracking algorithm is developed based on Page’s cumulative sum algorithm (CUSUM), a method for change-point detection, so that individual vehicles can autonomously track the boundary in a density field with measurement noise. Based on the data collected from sensing vehicles and prior knowledge of the dynamic model of boundary evolution, we estimate the boundary by solving an optimization problem, in which prediction and current observation are considered in the cost function. Examples and simulation results are presented to verify the efficiency of this approach.

Index Terms—Boundary tracking and estimation, hidden Markov model, change-point detection, CUSUM, optimization.

I. INTRODUCTION

Monitoring environmental boundaries has been an interesting topic for many years due to scientific and public safety applications. Examples include monitoring poisonous oil spills, harmful algae blooms, wild fire spreading, temperature and salinity distribution in the ocean, and hazardous weather conditions such as hurricanes and tropical storms.

There are many cases in which it is difficult to get global images of the boundary we are interested in from remote sensing technology. Two examples are transparent chemical spills and underwater “dead-zones” generated by algae blooms. In recent years, a considerable amount of work has been reported on using mobile sensing vehicles for environmental monitoring. Barat and Rendas [1] use single autonomous underwater vehicle (AUV) with a profiler sonar to detect the boundaries between distinct benthic regions. Kemp *et al.* [2] propose a simple algorithm for multiple AUVs surveillance that only requires concentration measurements. This algorithm has been tested on Caltech’s multi-vehicle wireless testbed [3], [4]. Bertozzi *et al* [5] design a centralized collective motion algorithm based on the “snake algorithm” in image processing to detect and track algae blooms, where each agent needs to measure the concentration gradient.

Coordination among multiple sensing vehicles is also a related research area. Clark and Fierro [6] try to detect and

surround a dynamic perimeter using a group of nonholonomic robots equipped with collision avoidance controllers. Zhang and Leonard [7] use four robots to compose a formation so that the gradient at the formation center can be measured in a density field. Susca *et al.* [8] propose a distributed coordination algorithm in which each vehicle tracks the boundary individually and communicates with its nearby neighbors. Vertices are generated uniformly along the boundary based on certain metric and a polygon is generated to approximate the boundary.

In this paper, we treat those mobile vehicles as an “observer” in which each vehicle can autonomously track the boundary with or without coordinating with others. The outputs of this “observer” are separated and noisy observations collected from vehicles. Assuming that there exists certain “hidden” relationship behind those separated data, which governs the dynamics of the boundary evolution, and we need to reveal this relationship so that the boundary can be accurately tracked and estimated. This idea is inspired by hidden Markov model (HMM) in [9] and interactive Bayesian filters in [10].

Our contribution includes two parts: First, we propose a revised tracking algorithm for a single vehicle based on [2], where CUSUM filters are embedded so that each vehicle can track the boundary even with noisy measurements. CUSUM is an efficient change-point detection method which can quickly detect small drifts in the parameters for random process [11], [12], [13], [14]. Second, assuming that the boundary can be approximated with an ellipse, we formulate the boundary estimation problem as an optimization problem. Finding the optimal ellipse parameters can be done by combining a priori prediction and measurements from sensing vehicles with additional noise. This is a first step towards a general framework using spatio-temporal nonlinear filtering for pattern recognition in environment monitoring.

The remainder of this paper is organized as follows: In Section II, we develop a tracking algorithm for a single vehicle by using CUSUM filters to process noisy measurements. CUSUM filters generate a binary signal, which is used as a navigation signal in each sensing vehicle. We then propose a framework to estimate the boundary based on separated observations in Section III. Assuming the boundary can be approximated by an ellipse and its parameters evolve according to Markov models, we show how to estimate those parameters by solving an optimization problem. Section IV is devoted to examples and simulation results that verify the efficiency of this approach. Finally, conclusions and future work are summarized in Section V.

Zhipu Jin is with the Department of Mathematics, University of California, Los Angeles, USA. He also holds a position as visiting scholar in the Department of Control and Dynamical Systems, California Institute of Technology. zhipu@math.ucla.edu

Andrea L. Bertozzi is with the Faculty of the Department of Mathematics, University of California, Los Angeles, USA. bertozzi@math.ucla.edu

II. TRACKING ALGORITHM WITH SINGLE VEHICLE

Assume an environmental boundary can be described by a density field in 2-D Euclidian space. The density function is a mapping

$$d(x) : \mathbb{R}^2 \rightarrow \mathbb{R} \quad (1)$$

where x is the location and $d(x)$ is the density value. The boundary is defined by a level set

$$\Omega = \{x \in \mathbb{R}^2 \mid d(x) = B\} \quad (2)$$

where B is the density threshold. In order to simplify the problem, we assume that $d(x) \in C^1$ and all level sets are smooth, i.e., Ω is continuous. Also, we assume that $d(x) > B$ if x is inside Ω and $d(x) < B$ if x is outside. Suppose each vehicle has a sensor which can measure the density field as

$$z(k) = d(x(k)) + v(k) \quad (3)$$

where k is the time, $v(k)$ is the sensor noise with zero mean, and $z(k)$ is the density measurement. Moreover, we assume that the speed of the vehicle is a constant, V , and we can control its orientation, θ , by

$$\theta(k+1) = \theta(k) + u(k). \quad (4)$$

Perhaps the simplest tracking algorithm for a single vehicle is the one used in [2], where the vehicle keeps turning in one direction when it is inside the boundary and in another direction when it is outside, i.e.,

$$u(k) = \begin{cases} +\omega & \text{when } z(k) > B \\ 0 & \text{when } z(k) = B \\ -\omega & \text{when } z(k) < B. \end{cases} \quad (5)$$

where $\omega > 0$. It is similar to the bang-bang control strategy except that ω is not necessarily the maximum value for the control law. Figure 1 shows a simple diagram of the vehicle's trajectory between two consecutive crossing points. This algorithm works well except for a few drawbacks.

- With large crossing angle θ_c , the tracking becomes very inefficient. The plot at the top of Figure 2 shows an example.
- When the noise $v(k)$ is large, it may turn the wrong way and fail to track the boundary.
- It may fail to cover the whole boundary when the boundary has narrow bottlenecks [2].

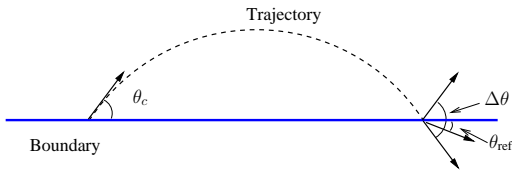


Fig. 1. Geometry analysis between two consecutive crossing points.

In this section, we propose an control strategy with non-linear filters to overcome the first two issues. The crossing angle θ_c may change a lot when a sensing vehicle tracks a boundary. Suppose the vehicle crosses the boundary at

time t_1 and t_2 , consecutively. Based on the time difference $\tilde{t} = t_2 - t_1$, we know the angle change $\Delta\theta = \tilde{t}\omega$ and define the control law at t_2 to be

$$u(t_2) = \begin{cases} (\tilde{t} \cdot \omega - 2\theta_{\text{ref}})/2 & \text{when } z(k) > B \\ -(\tilde{t} \cdot \omega - 2\theta_{\text{ref}})/2 & \text{when } z(k) < B \end{cases} \quad (6)$$

where θ_{ref} is a pre-set reference. Figure 2 shows that the efficiency can be improved, although there are still difficulties where the boundary has sharp turns. Obviously, Equation (6) depends on an accurate record of crossing times. When noise $v(k)$ is not negligible, false records may be taken near the boundary and make Equation (6) useless. Thus, we need filters to attenuate the noise and record the crossing moments.

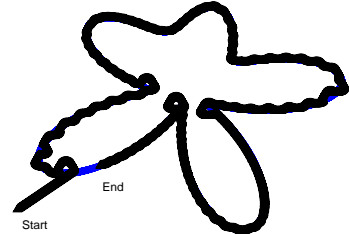
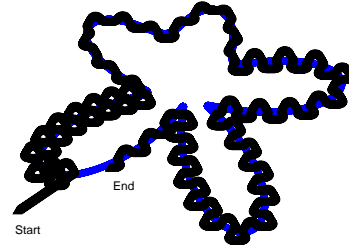


Fig. 2. Tracking a boundary. The blue curve is the boundary and the black curve is the trajectory of a sensing vehicle. Top: without angle correction. Bottom: with angle correction.

Assume $v(k)$ is given by Gaussian white noise with zero mean and covariance $R > 0$. Thus $z(k)$ is a random process with the same covariance and the mean value is $E[z(k)] = d(x(k))$. Suppose the vehicle crosses the boundary at time t_1 and $E[z(k)]$ drifts from below B to above B . The filter takes $z(k)$ as the input and outputs a time $\tilde{t} > t_1$, which is the time that the filter believes the crossing happened. The detection delay is defined as $\Delta = \tilde{t} - t_1$. Another performance metric is the probability of making a false record. A good filter should have a small Δ while keeping the false record probability low. This kind of problem is called a change-point detection problem in statistics analysis society. CUSUM is one of the most powerful methods for change-point detection and is especially good for small or linear drifts [11], [13].

For each vehicle, we employ two independent CUSUM filters to detect crossing times from outside to inside and

form inside to outside, respectively. Equation (7) describes the main part of the first filter, called the “high-side filter”. When the mean value increases above B , $U(k)$ quickly increases due to the accumulation of the measurements. When $U(k) > \bar{U}$, an accumulation threshold we can define, the filter believes that $E[z(k)] > B$ and outputs the time k . For discrete-time, it is still an open problem to analytically identify the tradeoff between the delay and the probability of a false record even with a Gaussian distribution and linear drifts. There are two parameters: “dead-zone” c_u and threshold \bar{U} . Simply speaking, the value of c_u determines the speed of the accumulation. A smaller c_u means the accumulation increases faster but false records are more likely. The value of \bar{U} affects the delay. A larger \bar{U} means false records are less likely but delays are larger. The same property holds for the second filter, called the “low-side filter”, which is described in Equation (8).

$$U(k) = \begin{cases} 0 & k = 0 \\ \max(0, z(k) - B - c_u + U(k-1)) & k > 0 \end{cases} \quad (7)$$

$$L(k) = \begin{cases} 0 & k = 0 \\ \min(0, z(k) - B + c_l + L(k-1)) & k > 0 \end{cases} \quad (8)$$

Combining the outputs of those two filters, we have a record of crossing times $\{t_1, t_2, \dots\}$ and we can easily generate a binary signal $b(k)$ if a correct initial condition can be obtained. Let $b(k) = 1$ when the vehicle is inside the boundary and $b(k) = 0$ when the vehicle is outside. Then the conditions on $z(k)$ in Equation (5) and (6) can be replaced by conditions on $b(k)$.

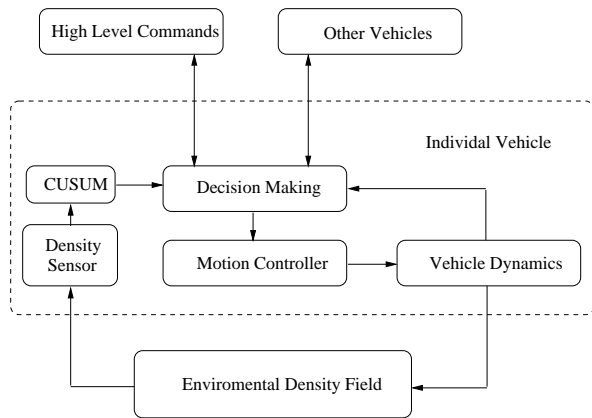


Fig. 3. Diagram for single vehicle tracking with CUSUM filters.

Figure 3 shows the feedback control diagram for a single vehicle in which the local decision is made based on CUSUM filters, high level commands, and possible coordination with other vehicles if possible. (In this paper, we will not discuss the coordination issue.) Figure 4 shows an example in which a vehicle tracks a straight line with noisy density measurements. The parameters for vehicle include the speed $V = 2$, updating rate is $10KHz$, and turning rate $\omega = 2 \times 10^{-4}$. For

CUSUM filters, we choose $c_u = 0.1$, $c_l = 0.1$, $\bar{U} = 500$, and $\bar{L} = -500$. The density field linearly increases along Y axis. The boundary that the vehicle tries to follow is $Y = -1$. Clearly, the raising edges of the high-side filter and falling edges of the low-side filter are good indicators for boundary crossings. In this particular case, we set upper limits on the absolute values of $U(k)$ and $L(k)$ as \bar{U} and \bar{L} , respectively. This cutoff explains the square wave appearance of the CUSUM outputs.

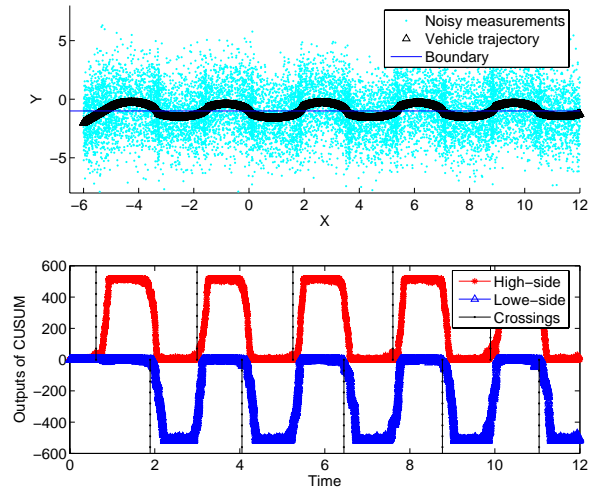


Fig. 4. Tracking a straight line with CUSUM filters.

Considering the stability of the tracking algorithm (5) and (6) with respect to detection delay Δ , it can be proved easily that, except the first detection delay, the algorithm can keep tracking a straight line if the detection delay

$$\Delta < (\pi - \theta_{ref})/\omega.$$

III. BOUNDARY ESTIMATION USING MULTIPLE VEHICLES

Suppose we have N vehicles tracking a boundary at the same time and there is no coordination among those vehicles. Assume we can observe the vehicles’ positions as $\{y_1(k), \dots, y_N(k)\}$ at time k .

$$y_i(k) = x_i(k) + w_i(k) \quad (9)$$

where $x_i(k)$ is the real position of vehicle i and $w_i(k)$ is the observation noise, which is Gaussian with zero mean and covariance q_i . Those positions are separated. If we do not have any dynamic model for the boundary evolvment, the best estimate we have for the boundary is a certain geometry shape based on those position data, for example, the inscribed polygon in [8].

However, for most environmental boundaries, we do have certain model parameters $s(k)$ to describe the dynamics of the density field. We assume that the timescale of the motion of sensing vehicles is much faster than that of the density field so that the vehicles can track the boundary.

An approximate boundary $\hat{\Omega}$ is generated based on $s(k)$. For simplicity, we assume that $s(k)$ evolves according to a Markov chain model and we formulate the boundary estimation problem as a hidden Markov model (HMM).

$$\begin{cases} s(k) &= \mathcal{Q} \cdot s(k-1) \\ \hat{\Omega}(k) &= g(s(k)) \end{cases} \quad (10)$$

where \mathcal{Q} is the transition matrix and $g(\cdot)$ is a function that maps $s(k)$ to the corresponding boundary $\hat{\Omega}$. The observations are defined by

$$Y(k) = X(k) + W(k) \quad (11)$$

where $Y = [y_1, \dots, y_N]'$, $X = [x_1, \dots, x_N]'$, and $W = [w_1, \dots, w_N]$. The goal is to find the best estimate of $s(k)$ given $Y(k)$. But the challenge is how to find a direct mapping between $\hat{\Omega}$ and $Y(k)$ given that each vehicle tracks the boundary autonomously.

One of the conventional approaches for state estimation in HMM is recursive Bayesian method, in which the distribution $p(s(k)|Y(k))$ is first predicted from $p(s(k-1)|Y(k-1))$ and $p(s(k)|s(k-1))$, and then corrected by the measurement likelihood $p(Y(k)|s(k))$. In our case, it is hard to determine $p(Y(k)|s(k))$ because we do not have a deterministic relationship between $s(k)$ and $X(k)$. Instead of the probabilistic approach, we recast the estimation problem as an optimization problem. The optimization problem is

$$\min_{\bar{s}(k)} \left(\bar{s}(k) - \tilde{s}(k) \right) \cdot \Pi^{-1} \cdot \left(\bar{s}(k) - \tilde{s}(k) \right)^T + \sum_{i=0}^N \|h(\hat{\Omega}(\bar{s}(k)), y_i(k))\|^2 / q_i \quad (12)$$

where $\bar{s}(k)$ is the estimate of $s(k)$, $\tilde{s}(k)$ is the one-step prediction based on $\bar{s}(k-1)$, function $h(\cdot, \cdot)$ calculates the distance from $\hat{\Omega}$ to $y_i(k)$, and matrix Π is a weight matrix that balances the predictions and the measurements.

For simplify, we assume that a 2-D boundary is approximated by an ellipse with parameters $s = (a, b, c_1, c_2)$ where (c_1, c_2) is the center and (a, b) are semi-major and semi-minor axes. For any point $x = (x_1, x_2) \in \hat{\Omega}(a, b, c_x, c_y)$, we have

$$\frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} = 1. \quad (13)$$

Parameters (c_1, c_2) evolve according to a random walk as

$$(c_1(k), c_2(k)) = \begin{cases} (c_1(k-1) + \delta_{c_1}, c_2(k-1)) & \text{with probability } \lambda_1 \\ (c_1(k-1) - \delta_{c_1}, c_2(k-1)) & \text{with probability } \lambda_2 \\ (c_1(k), c_2(k) + \delta_{c_2}) & \text{with probability } \lambda_3 \\ (c_1(k), c_2(k) - \delta_{c_2}) & \text{with probability } \lambda_4 \\ (c_1(k), c_2(k)) & \text{with probability } \lambda_5 \end{cases} \quad (14)$$

where $\sum_{i=1}^5 \lambda_i = 1$ and δ_{c_1} and δ_{c_2} are the step sizes in different directions.

The semi-major axis a has three changing patterns: expansion, shrinkage, and sustenance. So let a evolve according to

$$a(k) = a(k-1) + \eta_1 \cdot \delta_1 \quad (15)$$

where $\eta_1 \in \{-1, 0, 1\}$ and δ_1 is another step size. After every M steps, η_1 changes its value by randomly choosing

one of three possible values with equal probability. The same random process holds for semi-minor axis b as well with η_2 and δ_2 .

Thus, the prediction

$$\bar{s}(k) = (\bar{a}(k), \bar{b}(k), \bar{c}_1(k), \bar{c}_2(k)) \quad (16)$$

is calculated according to

$$\begin{cases} \bar{a}(k) &= \bar{a}(k-1) + \bar{\eta}_1 \delta_1 \\ \bar{b}(k) &= \bar{b}(k-1) + \bar{\eta}_2 \delta_2 \\ \bar{c}_1(k) &= \bar{c}_1(k-1) + \frac{\lambda_1 - \lambda_2}{\sum \lambda_i} \cdot \delta_{c_1} \\ \bar{c}_2(k) &= \bar{c}_2(k-1) + \frac{\lambda_3 - \lambda_4}{\sum \lambda_i} \cdot \delta_{c_2} \end{cases} \quad (17)$$

where $\bar{\eta}_1$ and $\bar{\eta}_2$ are updated every M steps based on the differences between $\bar{a}(k)$ and $\bar{a}(k-1)$, and $\bar{b}(k)$ and $\bar{b}(k-1)$, respectively.

The function $h(\cdot, \cdot)$ is defined as the distance from any point x to the ellipse $\hat{\Omega}$, i.e.,

$$h(s, x) = \min \|x - z\|_2 \quad (18)$$

for all $z \in \hat{\Omega}(s)$.

For N vehicles, some of the vehicles may be ‘‘lost’’, i.e., they have not detected any boundary crossings after turning in a complete circle, and we set $q_i = \infty$ so that y_i does not affect the cost function. Moreover, we add an additional constraint for the optimization problem (12) based on $b(k)$. If $b(k) = 1$, y_i should be inside $\hat{\Omega}$, i.e.,

$$\left(y_i - \begin{bmatrix} \bar{c}_1 \\ \bar{c}_2 \end{bmatrix} \right) \cdot \begin{bmatrix} \bar{a} & \\ & \bar{b} \end{bmatrix}^{-1} \cdot \left(y_i - \begin{bmatrix} \bar{c}_1 \\ \bar{c}_2 \end{bmatrix} \right)^T < 1. \quad (19)$$

Otherwise, y_i is outside and the norm is bigger than 1.

By adding the prediction term in the cost function, we actually implement the ‘‘correction’’ step by solving this optimization problem. The only problem left is how to find the matrix Π . In this paper, we assume that it is constant and is defined by

$$\Pi = \begin{bmatrix} q_a & & & \\ & q_b & & \\ & & q_{c_1} & \\ & & & q_{c_2} \end{bmatrix} \quad (20)$$

where q_a is the average error covariance on the boundary generated by the randomness of a . The other covariances, q_b , q_{c_1} , and q_{c_2} , are defined in the same way. Of course this is not accurate since even though those parameters may change independently, their contributions to the position error on the boundary may be correlated. Note this form is just an approximation. The value of q_a can be calculated as

$$q_a = \frac{1}{2\pi} \int_0^{2\pi} \frac{2}{9} (\epsilon_1^2 + \epsilon_2^2 - \epsilon_1 \epsilon_2) d\tau \quad (21)$$

where

$$\epsilon_1 = \left\| \begin{bmatrix} c_1 + (a + \delta_1) \cos(\tau) \\ c_2 + b \sin(\tau) \end{bmatrix} - \begin{bmatrix} c_1 + a \cos(\tau) \\ c_2 + b \sin(\tau) \end{bmatrix} \right\|_2 \quad (22)$$

and

$$\epsilon_2 = \left\| \begin{bmatrix} c_1 + (a - \delta_1) \cos(\tau) \\ c_2 + b \sin(\tau) \end{bmatrix} - \begin{bmatrix} c_1 + a \cos(\tau) \\ c_2 + b \sin(\tau) \end{bmatrix} \right\|_2. \quad (23)$$

The value of q_b can be calculated in the same way. For q_{c_1} and q_{c_2} , the distributions of errors are shown in Table I and the covariance can be calculated easily.

TABLE I
DISTRIBUTION OF ERRORS GENERATED BY c_1 AND c_2

Error distribution for c_1			
Error	δ_{c_1}	0	$-\delta_{c_1}$
Probability	λ_1	$\lambda_3 + \lambda_4 + \lambda_5$	λ_2
Error distribution for c_2			
Error	δ_{c_2}	0	$-\delta_{c_2}$
Probability	λ_3	$\lambda_1 + \lambda_2 + \lambda_5$	λ_4

IV. EXAMPLES AND SIMULATION RESULTS

In this section, we present some simulation results. Suppose we have an dynamic ellipse. Parameters c_1 and c_2 evolve according to Equation (14) with $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 1/5$ and $\sigma_{c_1} = \sigma_{c_2} = 0.004$. For a and b , the steps are $\sigma_1 = 0.01$, $\sigma_2 = 0.015$, and $M = 100$. We use five sensing vehicles to track the ellipse. The motion parameters for vehicles are $V = 0.1$ per step and $\omega = 0.4$. Each vehicle starts from a random initial position outside the ellipse. Its initial orientation is set towards the initial center of the ellipse. Each vehicle runs the tracking algorithm discussed in Section II right after it crosses the boundary for the first time.

Assume we have synchronized observation $Y(k)$ at each time k . The position noise w_i associated with each vehicle has unit covariance $q_i = 1$. We estimate the ellipse by solving the optimization problem listed in Equation (12). Figure 5 shows how the estimates evolve. The estimates of semi-major and semi-minor axes are consistent to the real values quiet well. Figure 6 shows the result in which, if we omit the first term in Equation (12), i.e., estimating the ellipse only based on the current observations. It appears to be random due to the noise. Also, Figure 7 and 8 show some snapshots for the estimation where the blue curve is the real ellipse, the red dashed curve is the estimate, the black dots shows the positions of the sensing vehicles in the last 40 steps (from black decay to grey), and the red stars are the observations at current time. In order to make those pictures clean, we do not show the sensing data for the density field. The CUSUM filters eliminate the impacts of density sensing noises so that each vehicle can track the ellipse very well.

Since we do not update Π , the result is sensitive to the initial value of \bar{s} . There are two issues we have not answer yet. First, the solution of the optimization problem may be not unique if the number of sensing vehicles is small. Second, we have not discuss the stability issue, which needs a better way to determine Π . However, even with this simple prediction strategy, the estimation result is still improved.

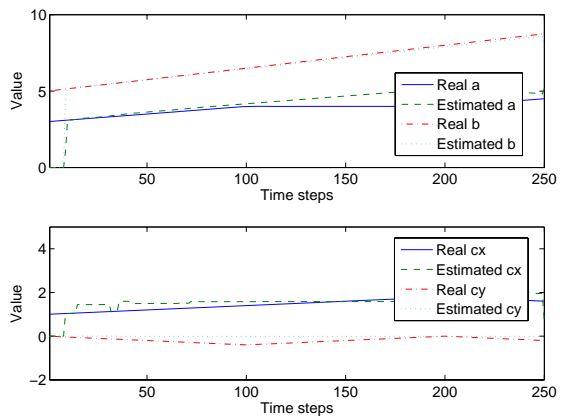


Fig. 5. Estimates on parameters with prediction.

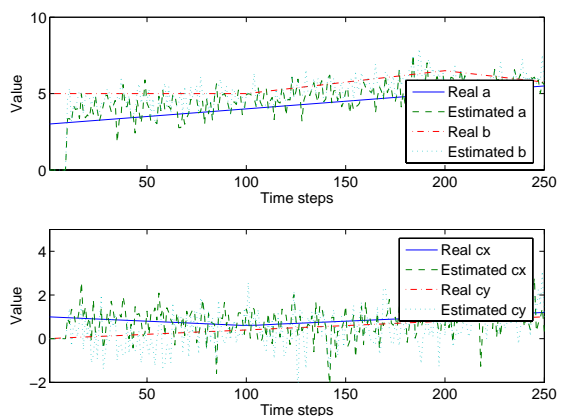


Fig. 6. Estimates on parameters without prediction.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we discuss environmental boundary tracking and estimation problem. We propose a tracking algorithm for individual vehicles where only the density measurements are needed. The measurement noise is processed by CUSUM filters and an orientation correction mechanism is employed to improve the performance. Moreover, as the first step towards a general framework on boundary estimation with separated measurements, we assume the boundary is defined by certain boundary parameters that we then estimate from observations by solving an optimization problem. Simulation results are presented to verify the efficiency of this approach.

There are several issues that should be addressed in future work. First, using more sophisticated models, such as spline theory, to approximate the boundary may be valuable. Second, better choices of the matrix Π should be investigated. An updating algorithm based on Bayesian theory, such as

$$\Pi(k) = \begin{bmatrix} q_a & & & \\ & q_b & & \\ & & q_{c_1} & \\ & & & q_{c_2} \end{bmatrix} + \Pi(k-1) \quad (24)$$

where Π represents how close $\bar{s}(k)$ is to the real parameters,

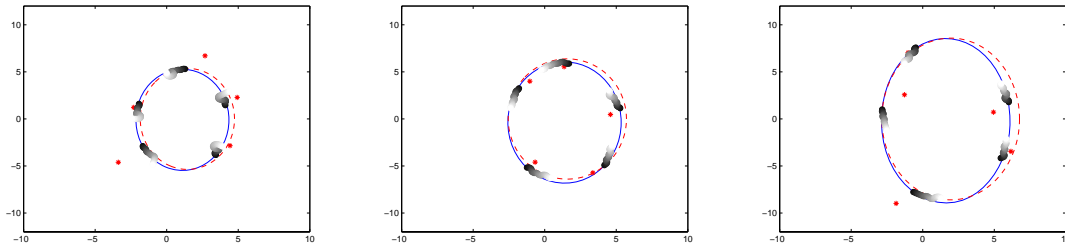


Fig. 7. Snapshots for estimating an ellipse with prediction.

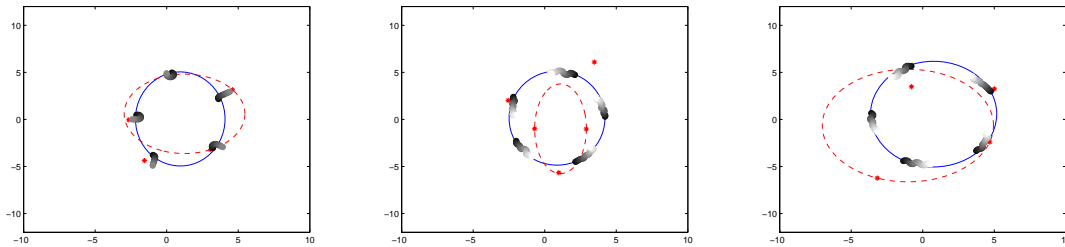


Fig. 8. Snapshots for estimating an ellipse without prediction.

is a good target for us to work towards. Third, the dynamics and motion patterns of individual vehicles definitely are crucial to determining the distribution $p(Y(k)|s(k))$. Also, the single vehicle tracking algorithms can be improved with more advanced interpolation and estimation methods. Last, but not least, coordination and cooperation among vehicles should help with boundary estimation by making the motion of vehicles more predictable.

VI. ACKNOWLEDGEMENTS

The authors would like to thank Prof. Boris Rozovsky, Dr. Alexander Tartakovsky, and Ernie Esser for discussions and comments. This research is partly supported by ARO MURI grant 50363-MA-MURI and ONR grant N000140610059.

REFERENCES

- [1] C. Barat and M. J. Rendas, "Benthic boundary tracking using a profiler sonar," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Oct. 2003, pp. 830–835.
- [2] M. Kemp, A. L. Bertozzi, and D. Marthaler, "Multi-uuv perimeter surveillance," *Proceedings of 2004 IEEE/OES Workshop on Autonomous Underwater Vehicles*, pp. 102–107, 2004.
- [3] C. H. Hsieh, Z. Jin, D. Marthaler, B. Q. Nguyen, D. J. Tung, A. L. Bertozzi, and R. M. Murray, "Experimental validation of an algorithm for cooperative boundary tracking," *Proceedings of the American Control Conference 2005*, pp. 1078–1083, 2005.
- [4] B. Q. Nguyen, Y.-L. Chuang, D. J. Tung, C. H. Hsieh, Z. Jin, L. Shi, D. Marthaler, A. L. Bertozzi, and R. M. Murray, "Virtual attractive-repulsive potentials for cooperative control of second order dynamic vehicles on the caltech mvwt," *Proceedings of the American Control Conference 2005*, pp. 1084–1089, 2005.
- [5] A. L. Bertozzi, M. Kemp, and D. Marthaler, *Cooperative Control, Lecture Notes in Control and Information Systems*, 2004, vol. 309, ch. Determining Environmental Boundaries: Asynchronous communication and physical scales, pp. 25–42.
- [6] J. Clark and R. Fierro, "Cooperative hybrid control of robotic sensors for perimeter detection and tracking," *Proceedings of the American Control Conference 2005*, pp. 3500–3505, 2005.
- [7] F. Zhang and N. Leonard, "Coordinated patterns on smooth curves," *Proceedings of IEEE International Conference On Networking, Sensing and Control*, pp. 434–440, 2006.
- [8] S. Susca, S. Martinez, and F. Bullo, "Monitoring environmental boundaries with a robotic sensor network," *IEEE Transactions on Control Systems Technology*, 2007, to appear.
- [9] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*, ser. Prentice Hall Signal Processing Series, A. V. Oppenheim, Ed. Prentice Hall PTR, 1993.
- [10] B. L. Rozovskii, A. Petrov, and R. B. Blazek, "Interactive banks of bayesian matched filters," *SPIE Proceedings: Signal and Data Processing of Small Targets*, 2000.
- [11] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, Jun. 1954.
- [12] L. L. Gan, "Cusum control charts under linear drift," *The Statistician*, vol. 41, no. 1, pp. 71–84, 1992.
- [13] M. S. Srivastava and Y. Wu, "Comparison of ewma, cusum and shiryayev-roberts procedures for detecting a shift in the mean," *The Annals of Statistics*, vol. 21, no. 2, pp. 645–670, 1993.
- [14] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blazek, and H. Kim, "Detection of intrusions in information systems by sequential change-point methods," *Statistical Methodology*, vol. 3, pp. 252–293, 2006.
- [15] A. Savvides, J. Fang, and D. Lymberopoulos, "Using mobile sensing nodes for boundary estimation," in *Workshop on Applications of Mobile Embedded Systems*, Boston, MA, Jun. 2004.
- [16] P. Bhatta, E. Fiorelli, F. Lekien, N. E. Leonard, D. A. Paley, F. Zhang, R. Bachmayer, R. E. Davis, D. M. Fratantoni, and R. Sepulchre, "Coordination of an underwater glider fleet for adaptive sampling," *Proceedings of International Workshop on Underwater Robotics*, pp. 61–69, 2005.
- [17] S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli, "On synchronous robotic networks Part I: Models, tasks and complexity," *IEEE Trans. Automat. Contr.*, 2007, to appear.