2013 IEEE International Conference on Automation Science and Engineering (CASE)

Cooperative and Active Sensing in Mobile Sensor Networks for Scalar Field Mapping

Hung Manh La, Weihua Sheng and Jiming Chen

***Abstract*—Scalar field mapping has many applications including environmental monitoring, search and rescue, etc. In such applications there is a need to achieve a certain level of confidence regarding the estimates at each location. In this paper, a cooperative and active sensing framework is developed to enable scalar field mapping using multiple mobile sensor nodes. The cooperative and active controller is designed via the real-time feedback of the sensing performance to steer the mobile sensors to new locations in order to improve the sensing quality. During the movement of the mobile sensors, the measurements from each sensor node and its neighbors are taken and fused with the corresponding confidences using distributed consensus filters. As a result an online map of the scalar field is built with a certain level of confidence of the estimates. We conducted computer simulations to validate and evaluate our proposed algorithms**

Keyword: Active sensing, Sensor fusion, Sensor networks.

I. Introduction

Active sensing in MSNs has recently attracted much research interest, especially in control engineering [1]. Some active sensing algorithms for source seeking and radiation mapping have been developed [2]–[5]. The problem of source seeking is first addressed in [2], and then it is thoroughly studied in [3], [4] for the case when direct gradient information of the measured quantity is unavailable. Also, the problem of chemical plume source localization is addressed by constructing a source likelihood map based on Bayesian inference methods [3]. Moreover, localization of a radiation source using only radiation intensity measurements has been done using a hybrid control strategy [4]. Additionally, active sensing for radiation mapping is developed in [5]. The control algorithm takes into account sensing performance and dynamics of the observed process therefore it can steer mobile sensors to locations where they maximize the information content of the measurement data.

In our previous work [6], [7], we have developed a cooperative sensing algorithm for an MSN to build the map of the scalar field. Based on our algorithm, all mobile sensors can form a quasi lattice formation and collaborate together to estimate the value at each cell of the field associated with

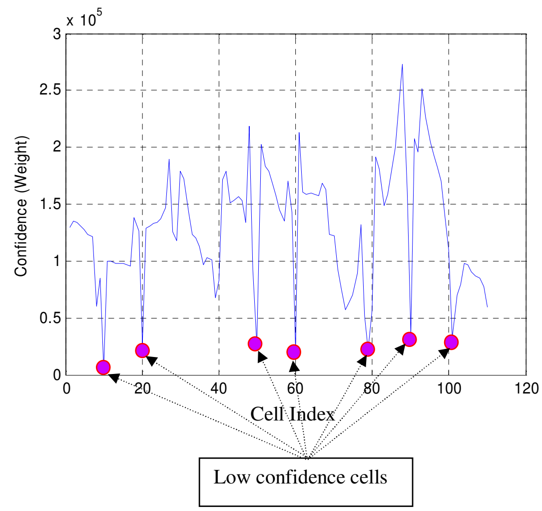

Fig. 1. The confidence at each cell of the scalar field.

its own confidence of estimation. However the cooperative controller does not include online feedback of the estimate confidence. Hence, the sensing performance or the confidence does not satisfy the desired one. This could affect some scalar field mapping applications such as temperature field mapping, search and rescue, where a need exists to achieve a certain level of confidence regarding the estimates at each location. As we can see from Figure 1, using the normal cooperative sensing algorithm developed in our previous papers [6], [7], we find that some cells have very low confidence. This means that we may miss important information at these locations (cells). For example, in search and rescue operation the MSN may miss the objects at the locations where the confidence of the estimate is not sufficient.

This motivates us to develop a cooperative and active sensing algorithm for MSNs so that each sensor only interacts with its neighbors and uses the local observation to automatically adjust the configuration of the MSNs such as relative location among sensors, orientation and focal length of the sensors (camera), etc. to adapt to the environments and improve the sensing performance. To achieve this goal the controller should be designed via the real-time feedback of the sensing performance. By this way the controller can steer the mobile sensor to move to the right locations of the field in order to improve the sensing quality. For simplicity, in this work we only focus on adjusting the relative location among sensors. Specifically, our problem focuses on how to control the movement of the mobile sensors to ensure quasi

This project is supported by the Department of Defense under DoD ARO DURIP grant 55628-CS-RIP, USA; and the Vietnamese Government under the MOET (Ministry of Education and Training) 322 program.

Hung M. La is with the Center for Advanced Infrastructure and Transportation (CAIT), Rutgers University, Piscataway, NJ 08854, USA, Email: hung.la11@rutgers.edu.

Weihua Sheng is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA, Email: weihua.sheng@okstate.edu.

Jiming Chen is with the State Key Lab. of Industrial Control Technology, Department of Control, Zhejiang University, Hangzhou, P.R.China, 310027,Email: jmchen@ieee.org.

uniform confidence on the estimates. Here by quasi uniform confidence we mean that the confidence is bounded by a lower and upper bound.

II. Scalar Field and Measurement Modeling

In this section we present the model of the scalar field and the measurement model of mobile sensors.

A. Model of the Scalar Field

We model the scalar field of interest as

$$F = \Theta\Phi^T, \tag{1}$$

here $\Theta = [\theta_1, \theta_2, ..., \theta_K]$, and $\Phi = [\phi_1, \phi_2, ..., \phi_K]$. ϕ_j, $j = 1, ..., K$, is a function representing a density distribution, and θ_j is the weight of the density distribution of the function ϕ_j.

We can model the function ϕ_j as a bi-variate Gaussian distribution

$$\phi_j = \frac{1}{\sqrt{det(C_j)(2\pi)^2}} e^{-\frac{1}{2}(x-\mu_x^j)C_j^{-1}(y-\mu_y^j)^T}, j \in [1, 2, ..., K].$$

here $[\mu_x^j \ \ \mu_y^j]$ is the mean of the distribution of function ϕ_j, and C_j is covariance matrix (positive definite) and it is represented by:

$$C_j = \begin{bmatrix} (\sigma_x^j)^2 & c_j^o\sigma_{xy}^j \\ c_j^0\sigma_{xy}^j & (\sigma_y^j)^2 \end{bmatrix},$$

where c_j^0 is a correlation factor.

B. Measurement Model

We partition the scalar field F into a grid of C cells. Each sensor i makes an observation (measurement) of the scalar field at cell k ($k \in \{1, 2, ..., C\}$) at time step t based on the following equation

$$m_i^k(t) = O_i^k(t)[F^k(t) + n_i^k(t)], \tag{2}$$

here $n_i^k(t)$ is the Gaussian noise with zero mean and variance $V_i^k(t)$ at time step t. We assume that n_i^k is uncorrelated noise which satisfies

$$Cov(n_i^k(s), n_i^k(t)) = \begin{cases} V_i^k, & if \quad s = t \\ 0, & \quad\quad otherwise, \end{cases}$$

here Cov is the covariance. $O_i^k(t)$ is the observability of sensor node i at cell k at time step t, and it is defined as

$$O_i^k(t) = \begin{cases} 1, & if \quad \|q_i(t) - q_c^k\| \leq r_i^s \\ 0, & \quad\quad otherwise, \end{cases} \tag{3}$$

here $q_i \in R^2$ is the position of sensor node i; $q_c^k \in R^2$ is the location of cell k at its center; and r_i^s is the sensing range of sensor node i. This definition tells us that if cell k is inside the sensing range, r_i^s, of sensor node i then cell k can be measured or observed. Otherwise the observability is zero.

Each mobile sensor node makes an measurement at cell k corresponding to its position. We assume that the variance $V_i^k(t)$ is related to the distance between the sensor node i and the location of the measurement according to:

$$V_i^k(t) = \begin{cases} \frac{\|q_i(t)-q_c^k\|^2+c_v}{(r_i^s)^2}, & if \quad \|q_i(t) - q_c^k\| \leq r_i^s \\ 0, & \quad\quad otherwise, \end{cases} \tag{4}$$

here c_v is the small positive constant between 0 and 1. The reason of introducing c_v is to avoid the variance $V_i^k(t)$ being zero when the distance $\|q_i(t) - q_c^k\|$ equals to zero.

III. Distributed Sensor Fusion Algorithm

A. Overall Approach

In this section we present a distributed sensor fusion algorithm to allow each sensor node to find out an estimate of the value at each cell of the scalar field based on its own measurement and its neighbor's measurements. Our algorithm has two phases. First, at time step t, each sensor node finds an estimate of the value of the scalar field F . Second, each sensor node finds a final estimate of the value of the scalar field F at each cell during its movement. To achieve it, we use two consensus filters. The consensus filter 1 is to find out an estimate of the value of the field F at each cell at time step t. Since each mobile sensor node makes its own measurement at each cell with its own weight (confidence), the consensus filter 2 is used to find out an agreement among these confidences.

At each time step t each mobile sensor node needs to find an estimate of the value of each cell based on consensus filter 1, and find an overall confidence of this estimate based on consensus filter 2. This process can be called the *spatial estimation phase*. Then, during the movement of each sensor node, it will have multiple spatial estimates of each cell associated with their own confidences. Hence, these spatial estimates are fused iteratively through the weighted average protocol, and this process can be called the *temporal estimation phase*.

B. Spatial Estimation Phase

In the spatial estimation phase, the measurements of each sensor node and its neighbors at cell k at time step t are inputs of the consensus filter 1. Then, the output of this consensus is the estimate of the value of the scalar field F at cell k at time step t.

Also in this phase, the confidences (weights) of the measurements of each sensor node and its neighbors at cell k at time step t are inputs of the consensus filter 2. Then, the output of this consensus is the estimate of the confidence of the measurement of the scalar field at cell k at time step t.

1) Consensus Filter 1: Distributed consensus [7], [8] is an important computational tool to achieve cooperative sensing. We consider distributed linear iterations of the following form

$$x_i^k(l+1) = w_{ii}^k(l)x_i^k(l) + \sum_{j \in N_i(t)} w_{ij}^k(l)x_j^k(l), \tag{5}$$

here l is iteration index. The initial condition for the state is given as $x_i^k(l = 0) = m_i^k(t)$. The weight, $w_{ii}^k(l)$, is the self weight or vertex weight of each sensor to cell k, and $w_{ij}^k(l)$ is the edge weight between sensor i and sensor j.

Our problem here is to estimate the value of the field F at each cell k at each time step t. Since each sensor node makes the observation at cell k at time step t based on its own confidence (weight), the consensus should converge to the weighted average of all observations (measurements) at cell k from all sensor nodes in the network. This weighted average is the estimate of the value at cell k at time step t, and it is computed as:

$$E^k(t) = \frac{\sum_{i=1}^n w_{ii}(t) m_i(t)}{\sum_{i=1}^n w_{ii}(t)}. \quad (6)$$

In order to make the consensus (5) converge to E^k in (6) we need to ensure that the sum of all weights including the vertex and edge weights at each node satisfies the following condition [6], [7]:

$$w_{ii}^k(l) + \sum_{j \in N_i(t)} w_{ij}^k(l) = 1. \quad (7)$$

From Equation (7) by assigning the same value to all edge weights we obtain:

$$w_{ij}^k(l) = \frac{1 - w_{ii}^k(l)}{|N_i(t)|}. \quad (8)$$

here, $w_{ii}^k(l)$ is defined as

$$w_{ii}^k(l) = \frac{c_1^w}{V_i^k(t)}, \quad (9)$$

where c_1^w is a constant. If sensor node i does not observe cell k ($O_i^k(t) = 0$) then the vertex weight $w_{ii}^k(l)$ is set to zero.

Therefore we have the following weight design

$$w_{ij}^k(l) = \begin{cases} \frac{c_1^w}{V_i^k(t)}, & if \quad i = j, \\ \frac{1 - w_{ii}^k(l)}{|N_i(t)|}, & if \quad i \neq j, j \in N_i(t), \\ 0, & otherwise. \end{cases} \quad (10)$$

In order to satisfy Equation (7) we need the following condition for c_1^w:

$$0 < \frac{c_1^w}{V_i^k(t)} < 1. \quad (11)$$

Since $min(V_i^k(t)) = \frac{c_v}{(r_i^s)^2}$ when $\|q_i(t) - q_c^k\| = 0$, we have:

$$0 < \frac{c_1^w}{\frac{c_v}{(r_i^s)^2}} < 1 \Rightarrow 0 < c_1^w < \frac{c_v}{(r_i^s)^2}. \quad (12)$$

2) Consensus Filter 2: Since each sensor node has its own confidence of the measurement of the value of the scalar field at each cell at each time step t we need to find an agreement among the confidences of sensor nodes. The consensus algorithm 2 is introduced to find the overall confidence from each time step t. This overall confidence is the estimated weight, $W_i^k(t)$, of the weighted average protocol as shown in Equation (17).

Let $y_i^k(l = 0)$ be the confidence of the measurement of the value of the scalar field at cell k at each time step t for sensor node i, or $y_i^k(l = 0) = w_{ii}^k(t)$. Let $y_j^k(l = 0)$ be the confidence of the measurement of the value of the scalar field at cell k at each time step t for sensor node j with $j \in N_i(t)$, or $y_j^k(l = 0) = w_{jj}^k(t)$. Then, we have the following consensus filter

$$y_i^k(l+1) = w_{ii}^k(l) y_i^k(l) + \sum_{j \in N_i(t)} w_{ij}^k(l) y_j^k(l), \quad (13)$$

In this consensus filter, we use the Metropolis weight [8] as

$$w_{ij}^k(l) = \begin{cases} \frac{1}{1 + max(|N_i(t)|, |N_j(t)|)}, & if \quad i \neq j, j \in N_i(t), \\ 1 - \sum_{j \in N_i(t)} w_{ij}^k(l), & if \quad i = j, \\ 0, & otherwise. \end{cases} \quad (14)$$

C. Temporal Estimation Phase

From the consensus filters 1 and 2, to allow each sensor node to on-line estimate the value of the scalar field at each cell based on its own measurement and its neighbor's measurements, the temporal estimate phase is used. During the movement of sensor nodes, each sensor obtain several spatial estimates (from the *spatial estimation phase*) of the value at cell k associated with its own confidence, hence the final estimate is iteratively updated based on these spatial estimates via the weighted average protocol.

For details, first let l_c be the iteration that both consensus filter 1 and 2 converge, then we have the estimates of cell k: $E_i^k(t) = x_i^k(l_c)$; and $W_i^k(t) = y_i^k(l_c)$. We can find the final estimate of the value of the scalar field at cell k based on following equations:

- Update weight (confidence):

$$\overline{W}_i^k(t) = W_i^k(t-1) + W_i^k(t-2) + ... + W_i^k(0) \quad (15)$$

- Update the final estimate (*weighted average protocol*):

$$\overline{E}_i^k(t = 0) = E_i^k(t = 0) = x_i^k(l_c) \quad (16)$$

$$\overline{E}_i^k(t) = \frac{\overline{W}_i^k(t-1)\overline{E}_i^k(t-1) + W_i^k(t) E_i^k(t)}{\overline{W}_i^k(t-1) + W_i^k(t)} \quad (17)$$

IV. Potential Controller Design for Active Sensing

In this section we aim to develop a potential controller for cooperative and active sensing, and the main idea of our approach is shown in Figure 2. Our purpose is using the feedback of the confidence of the estimate to adjust the movement of the sensors to adapt to the environments so that they can improve the sensing performance in a distributed fashion. Specifically, the potential controller is designed to steer the mobile sensors to the expected locations in order to achieve the quasi uniformity of the confidence. First, we describe the flocking control algorithm [9], [10] which is used to control mobile sensors to move together without collision.

A. Flocking Control

We consider n mobile sensor nodes moving in two dimensional Euclidean space. The dynamic equations of each sensor node are described as:

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i, \quad i = 1, 2, ..., n. \end{cases} \quad (18)$$

here $(q_i, \ p_i) \in R^2$ are the position and velocity of sensor node i, respectively, and u_i is the control input of sensor node i.

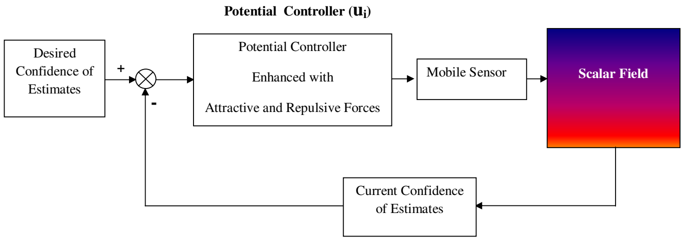

Fig. 2. Diagram of cooperative and active sensing based on the flocking controller enhanced with attractive and repulsive forces via confidence feedback.

The geometry of the MSN is modeled by an α-lattice [9] that meets the following condition:

$$\|q_j - q_i\| = d, j \in N_i(t), \tag{19}$$

here d is a positive constant indicating the distance between sensor node i and its neighbor j. However, at singular configuration ($q_i = q_j$) the collective potential used to construct the geometry of flocks is not differentiable. Therefore, the set of algebraic constraints in (19) is rewritten in term of σ - norm [9] as follows:

$$\|q_j - q_i\|_\sigma = d^\alpha, j \in N_i(t),$$

here the constraint $d^\alpha = \|d\|_\sigma$ with $d = r/k_c$, where k_c is the scaling factor. The σ - norm, $\|.\|_\sigma$, of a vector is a map $R^m \implies R_+$ defined as $\|z\|_\sigma = \frac{1}{\epsilon}[\sqrt{1+\epsilon\|z\|^2} - 1]$ with $\epsilon > 0$. Unlike the Euclidean norm $\|z\|$, which is not differentiable at $z = 0$, the σ - norm $\|z\|_\sigma$, is differentiable every where.

The flocking control algorithm which consists of the formation control term and the leader tracking control term is presented as

$$u_i = f_i^\alpha + f_i^t.$$

The formation controller [9] is used to control the network to form a quasi lattice formation, and it is designed based on a pairwise attractive/repulsive force.

$$f_i^\alpha = c_1^\alpha \sum_{j \in N_i} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + c_2^\alpha \sum_{j \in N_i} a_{ij}(q)(p_j - p_i), \tag{20}$$

where c_1^α and c_2^α are positive constants. More details of how to compute f_i^α please see [9].

The leader tracking controller is used to control each mobile sensor to track the virtual leader. The trajectory of the virtual leader is planned so that the MSN can cover the entire scalar field. This controller is presented as

$$f_i^t = -c_1^t(q_i - q_t) - c_2^t(p_i - p_t) \tag{21}$$

here c_1^t and c_2^t are positive constant, and q_t and p_t are position and velocity of the virtual leader, respectively.

B. Design of Attractive Force

In this subsection, we introduce the attractive force term to increase the confidence level. The attractive force will steer the mobile sensors to the cells which have low confidence. In order to do this, first let q_c^k be the location of the cell that has confidence lower than the lower bound, or $k \in O_i^L(t)$, here $O_i^L(t)$ is the subset of cells covered by mobile sensor i at time t, which have confidence lower than the lower bound. $O_i^L(t) \subset O_i^c(t)$, here $O_i^c(t)$ is the set of cells covered by mobile sensor i at time t, and it is defined as

$$O_i^c(t) = \left\{k \in \vartheta_O : \|q_c^k - q_i\| \le r_i^s, \vartheta_O = \{1, 2, ..., k\}\right\}. \tag{22}$$

At each time t, the mobile sensor i may have several cells which have confidence lower than the desired one. In order to steer the mobile sensor to go to these low confidence cells, the virtual attractive force are generated at these cells. If the cell has lower confidence the bigger attractive force is generated. To express the details of the attractive force design, first let $\overline{W}_d^L$ be a lower bound of the desired confidence of the estimates of all cells in the scalar field, and $\overline{W}_d^L$ is a vector of $1 \times C$ dimension. Let $\Delta_W^L(t) = \overline{W}_d^L - \overline{W}(t)$ be the difference between the current confidence and the lower bound (see Figure 3), $\Delta_W^L(t) = [\Delta_W^1(t), \Delta_W^2(t), ..., \Delta_W^C(t)]$. Based on this feedback, $\Delta_W^L(t)$, we can design a attractive force as shown in Equation (23).

$$f_i^{att} = - \sum_{k \in O_i^L(t)} C_k^{att} \phi_{att}(\|q_c^k - q_i\|_\sigma) n_{i,k}^{att} \tag{23}$$

here, $C_k^{att} = c_a \frac{\Delta_W^k(t)}{\sqrt{1+(\Delta_W^k(t))^2}}, \Delta_W^k(t) \in \Delta_W^L(t)$, and c_a is a positive constant. C_k^{att} is used to control the amplitude of the attractive force. Namely, if cell k has low confidence or $\Delta_W^k(t)$ is large, the the amplitude of the attractive force is big in order to attract the mobile sensor to go to closer this cell. The attractive force function $\phi_{att}(\|q_c^k - q_i\|_\sigma)$ is designed as:

$$\phi_{att}(\|q_c^k - q_i\|_\sigma) = \rho_h(\frac{\|q_c^k - q_i\|_\sigma}{r_\alpha^s}) \frac{\|q_c^k - q_i\|_\sigma}{\sqrt{1 + \|q_c^k - q_i\|_\sigma^2}} \quad k \in O_i^L(t).$$

here, $r_\alpha^s = \|r^s\|_\sigma$ (r^s is sensing range as defined before). Similar to [9], the bump function $\rho_h(\frac{\|q_c^k - q_i\|_\sigma}{r_\alpha^s})$ with $h \in (0, 1)$ is defined as

$$\rho_h(\frac{\|q_c^k - q_i\|_\sigma}{r_\alpha^s}) = \begin{cases} 1, if \frac{\|q_c^k - q_i\|_\sigma}{r_\alpha^s} \in [0, h) \\ 0.5[1 + cos(\pi(\frac{\frac{\|q_c^k - q_i\|_\sigma}{r_\alpha^s} - h}{1-h}))] \\ if \frac{\|q_c^k - q_i\|_\sigma}{r_\alpha^s} \in [h, 1] \\ 0, otherwise. \end{cases} \tag{24}$$

The vector along the line connecting q_c^k ($k \in O_i^L(t)$) and q_i is defined as:

$$n_{ik}^{att} = (q_c^k - q_i)/\sqrt{1 + \epsilon\|q_c^k - q_i\|^2}, k \in O_i^L(t). \tag{25}$$

here, ϵ is small positive constant.

C. Design of Repulsive Force

Based on the attractive force design in the previous subsection, the confidence level can be increased, however some cells may have too high confidence. This is unnecessary since this needs more measurements, and causes more energy consumption. Therefore, it is desirable if we can maintain

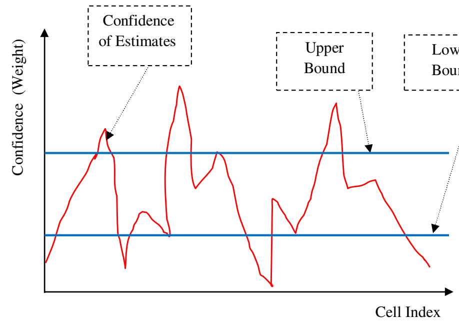

Fig. 3. Illustration of confidence feedback for quasi uniformity of the confidence. The upper bound and lower bound are used to create a quasi uniform of the confidence.

both lower and upper bound of the confidence performance, or a quasi uniform confidence (see Figure 3). Hence, we introduce a repulsive force term to the Potential Controller in order to steer the mobile sensors to move away from the cells which have too high confidence.

Let q_c^k be the location of the cell that has confidence higher than the upper bound (see Figure 3). For these cells we will create the virtual repulsive force to steer the mobile sensors to move away.

To express the details of the repulsive force design, first let $\overline{W}_d^H$ be a upper bound of the desired confidence of the estimates of all cells in the scalar field, and $\overline{W}_d^H$ is a vector of $1 \times C$ dimension. Let $\Delta_W^H(t) = \overline{W}_d^H - \overline{W}(t)$ be the difference between the current confidence and the upper bound (see Figure 3), $\Delta_W^H(t) = [\Delta_W^1(t), \Delta_W^2(t), ..., \Delta_W^C(t)]$. Based on this feedback, $\Delta_W^H(t)$, we can design a repulsive force as shown in Equation 26.

$$f_i^{rep} = \sum_{k \in O_i^H(t)} C_k^{rep} \phi_{rep}(\|q_c^k - q_i\|_\sigma) n_{i,k}^{rep} \tag{26}$$

here, $C_k^{rep} = c_r \frac{|\Delta_W^k(t)|}{\sqrt{1+(\Delta_W^k(t))^2}}$, $\Delta_W^k(t) \in \Delta_W^H(t)$, here c_r is a positive constance. $O_i^H(t)$ is the subset of cells covered by mobile sensor i at time t, which have confidence higher than the upper bound. Obviously, $O_i^H(t) \subset O_i^c(t)$. C_k^{rep} is used to control the amplitude of the repulsive force. Namely, if cell k has high confidence, or $\Delta_W^k(t)$ is large, the the amplitude of the repulsive force is big in order to push the mobile sensor to move away from this cell further. The repulsive force function $\phi_{rep}(\|q_c^k - q_i\|_\sigma)$ is designed as:

$$\begin{aligned}\phi_{rep}(\|q_c^k - q_i\|_\sigma) &= \rho_h(\frac{\|q_c^k - q_i\|_\sigma}{r_\alpha^s}) \\ &\times(\frac{\|q_c^k - q_i\|_\sigma - r_\alpha^s}{\sqrt{1+(\|q_c^k - q_i\|_\sigma - r_\alpha^s)^2}} - 1) \\ &, k \in O_i^H(t).\end{aligned}$$

The bump function $\rho_h(\frac{\|q_c^k - q_i\|_\sigma}{r_\alpha^s})$ is defined as (24), but it is now applied for the high confidence cells or $k \in O_i^H(t)$. The vector along the line connecting q_c^k ($k \in O_i^H(t)$) and q_i is defined as:

$$n_{ik}^{rep} = (q_c^k - q_i)/\sqrt{1+\epsilon\|q_c^k - q_i\|^2}, k \in O_i^H(t). \tag{27}$$

Finally, the potential controller for the cooperative and active sensing is presented as follows:

$$u_i = f_i^{rep} + f_i^{att} + f_i^\alpha + f_i^t. \tag{28}$$

V. Simulation Results

In this section, we test our cooperative and active sensing algorithm and compare it with the normal cooperative sensing algorithm [6], [7] in terms of the sensing performance.

We model the environment (scalar field F) as multiple variate Gaussian distributions. The scalar vector Θ can be arbitrarily selected, for example $\Theta = [20\ 50\ 35\ 40]$, corresponding to four multiple variate Gaussian distributions ($K = 4$), and each one is represented as:

$$\phi_1 = \frac{1}{\sqrt{det(C_1)(2\pi)^2}} e^{-\frac{1}{2}(x-2)C_1^{-1}(y-2)^T}.$$

here we can select: $C_1 = \begin{bmatrix} 2.25 & 0.2999 \\ 0.2999 & 2.25 \end{bmatrix}$, with the correlation factor $c_1^0 = 0.1333$.

For the functions ϕ_2, ϕ_3, ϕ_4: the means (μ_x, μ_y) are (1, 0.5), (4.3, 3.5), (3, -3), respectively; the matrix $C_2 = C_3 = C_4 = \begin{bmatrix} 1.25 & 0.1666 \\ 0.1666 & 1.25 \end{bmatrix}$; the correlation factor $c_4^0 = c_3^0 = c_2^0 = c_1^0$.

We set the lower bound of the confidence level is 0.5×10^5, and the higher bound of the confidence level is 1.9×10^5. The field F has a size of 10×9, and it is partitioned into 110 cells. The snapshots of multiple sensor nodes forming a flock and building the map of the unknown scalar field are shown in Figure 4.

The final confidence of the estimate in one dimension at each cell of the field F is shown in Figure 5. In this figure we compared three methods together. Namely, Figure 5 (a) shows the confidence of normal cooperative sensing (Potential Controller without attractive and repulsive forces). Figure 5 (b) shows the confidence of active sensing with the Potential Controller with attractive force only. Figure 5 (c) shows the confidence of active sensing with Potential Controller with both attractive and repulsive forces. From these results, we can see that by using both attractive and repulsive force controllers we have better uniformity of the confidence performance. This indicates that all the cells of the scalar field are observed with a certain level of confidence.

To see the advantages of the active sensing we compare it with the normal sensing in term of mapping error as shown in Figure 6. We can see that the error between the original map and the built map over cells is small when applying the active sensing (see Figure 6 (b)), but it is bigger when applying the normal sensing (see Figure 6 (a)).

VI. Conclusion

This paper presented cooperative and active sensing algorithms for mobile sensor networks to build the map of an unknown scalar field. The proposed distributed sensor fusion algorithm consists of two different distributed consensus

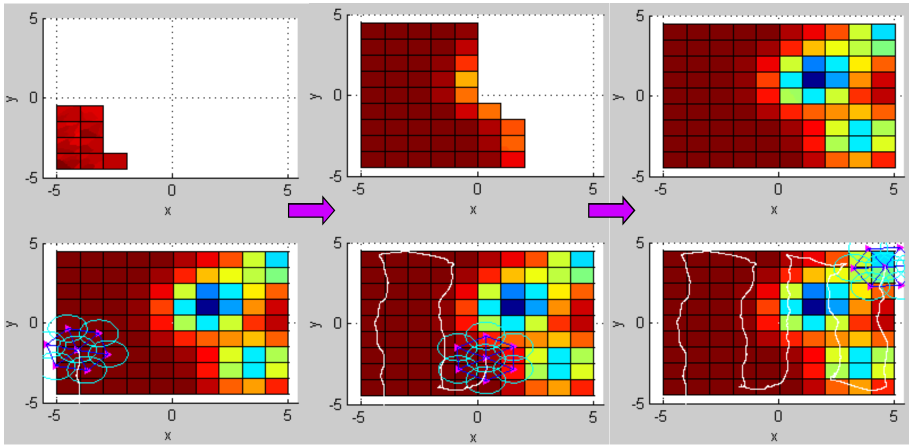

Fig. 4. Snapshots of building the map of the scalar field F using the distributed fusion algorithm and the cooperative and active sensing algorithm (28).

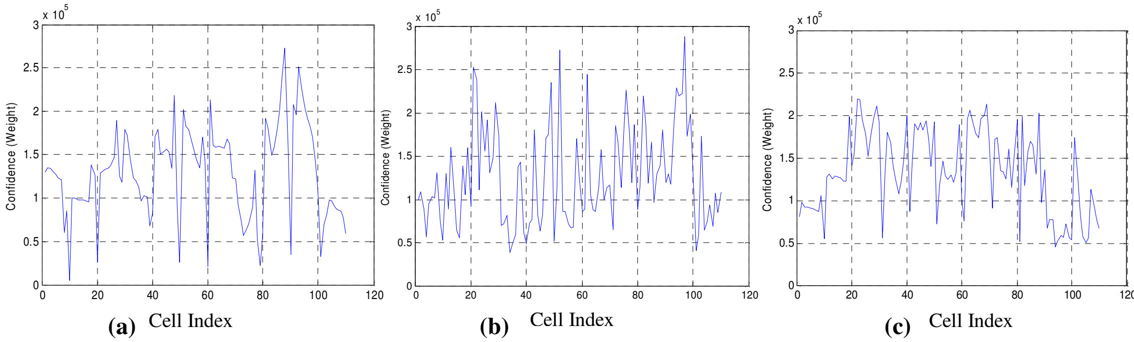

Fig. 5. Confidence over the cells in one dimension: (a) for normal cooperative sensing [6], [7]; (b) for active sensing with Potential Controller using only attractive force; (c) for active sensing with Potential Controller using both attractive and repulsive forces (28).

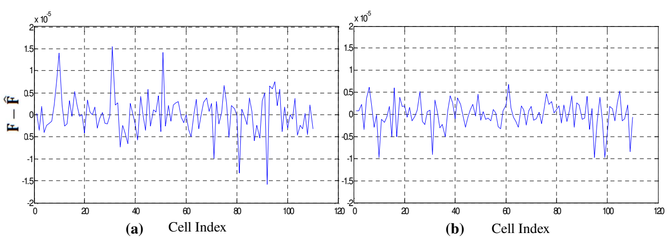

Fig. 6. Error between the original map and the built map in one dimension over cells: (a) for the normal sensing [6], [7]; (b) for the active sensing.

filters which can find an agreement on the estimates and an agreement on the confidence among sensor nodes. Each sensor node cooperates with neighboring sensors to estimate the value of the field at each cell. The final estimates of the values of the scalar field are updated on-line based on the weighted average protocol. More importantly, the mobile sensors can automatically adjust their movement to achieve quasi uniform confidence through a potential field based feedback control algorithm. Simulation results are collected to demonstrate the proposed algorithms.

REFERENCES

[1] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray. On a decentralized active sensing strategy using mobile sensor platforms in a network. *the 43th IEEE Conf. on Decision and Control*, pages 1914–1919, 2004.

[2] C. Zhang, D. Arnold, N. Ghods, A. Siranosian, and M. Krstic. Source seeking with nonholonomic unicycle without position measurment part 1: Tuning of forward velocity. *IEEE Conf. on Decision and Control*, pages 3040–3045, 2006.

[3] S. Pang and J. A. Farrell. Chemical plume source localization. *IEEE Trans. on Systems, Man, and Cybernetics – Part* B, 36(5):1068–1080, 2006.

[4] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel. Robust source-seeking hybrid controllers for autonomous vehicles. *American Control Conf.*, pages 1185–1190, 2007.

[5] H. G. Tanner R. A. Cortez and R. Lumia. Distributed robotic radiation mapping. *Experimental Robotics–The Eleventh Int.l Symposium, volume 54 of Springer tracts in advanced robotics, Springer*, pages 147–156, 2009.

[6] H. M. La and W. Sheng. Cooperative sensing in mobile sensor networks based on distributed consensus. *the Signal and Data Processing of Small Targets Conf., Proc. of* SPIE, 2011.

[7] H. M. La and W. Sheng. Distributed sensor fusion for scalar field mapping using mobile sensor networks. *IEEE Trans. on Cybernetics*, 43(2):766–778, Apr. 2013.

[8] L. Xiao, S. Boy, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. *International Conference on Information Processing in Sensor Networks*, pages 63–70, 2005.

[9] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[10] H. M. La and W. Sheng. Flocking control of a mobile sensor network to track and observe a moving target. *Proc. of the 2009 IEEE Int. Conf. on Robotics and Automation (*ICRA 2009*), Kobe, Japan*, pages 3129 – 3134, 2009.