Journal of Bionic Engineering, Volume 13, Issue 4, October 2016, Pages 679–689.
DOI: 10.1016/S1672-6529(16)60339-6

# Tracking and sensor coverage of spatio-temporal quantities using a swarm of artificial foraging agents

John Oluwagbemiga Oyekan[a], Dongbing Gu[b], Huosheng Hu[b]
[a]School of Aerospace, Transport and Manufacturing,
Cranfield University, Cranfield, Bedfordshire, MK45 0AL.
[b] School of Computer Science and Electronic Engineering,
University of Essex, Wivenhoe Park, Colchester CO4 3SQ.
*Email: j.oyekan@cranfield.ac.uk; oyekanjohn@gmail.com*

***Abstract*—Using a network of mobile sensors to track and map dynamic spatio-temporal processes in the environment is one of the current challenges in multi-agent systems. In this work, a distributed probabilistic multi-agent algorithm inspired by the bacterium foraging behaviour is presented. The algorithm is capable of tracking as well as providing coverage to a dynamic spatio-temporal without the need of machine learning or future planning. This is unlike most current techniques that rely heavily on machine learning to estimate the distribution as well as the profile of spatio-temporal quantities. Our algorithm is computationally tractable for simple agents, shown to adapt to its environment and can deal successfully with noise in sensor readings as well as in robot dynamics.**

***Index Terms*—spatio-temporal, Pollution, Visualization, Optimal Coverage, Environmental Monitoring, Bacteria.**

## I. Introduction

In this paper, a distributed probabilistic algorithm capable of making a swarm of mobile sensors track and provide an optimal coverage of a spatio-temporal function (e.g temperature, pollution etc) in a region of interest is considered is proposed and developed. Our goal is to enable agents respond dynamically to a changing spatio-temporal profile so that (i) they are optimally placed for collecting data as well as (ii) provide a representative visual map reference that can be used by humans to understand the spatio-temporal quantity. This map could be used to gain insight into the dynamic properties of the spatio-temporal quantity from a distance and could be particularly useful if the quantity were hazardous. In order to achieve this goal however, various challenges need to be solved by the algorithm. The developed algorithm must enable the mobile agents: (i) to co-operate and maintain coordination with each other; (ii) to provide adequate coverage of the pollutant; (iii) to quickly adapt to changes in the distribution of the spatio-temporal quantity caused by wind changes, temperature induced conventional currents and so on; (iv) to explore the environment to find the pollutant after release and for reacquiring the pollutant when it changes its distribution and (v) to deal with noise in their sensors and their dynamics.

The challenges mentioned above are what organisms deal with in their environment everyday. For example, due to their need to forage, organisms need to keep track of the continuously changing spatio-temporal property of food concentration in their environment whilst dealing with noise in their sensors, environment and dynamics. Cemetery construction in ants is another example in which organisms adapt to changes in the spatio-temporal properties in the environment. It has been observed that ants adjust the location and shape of ant cemetery clusters by keeping track of wind flow direction in the environment [4][5]. Taking inspiration from these examples and others, we develop a probabilistic coverage controller that is capable of addressing the challenges stated above. We make use of a bacterium's: (i) exploration ability in finding food sources in a space much larger than its size, (ii) exploitation ability in tracking down high concentrations of food once found, (iii) self organizing property to form spatio-temporal distributions on food sources in the environment, (iv) feedback mechanism in terms of keeping a relative distance to another individual in a swarm and (v) ability to filter out noise in its sensors. This is unlike our previous work in [11] where we focused on spatial quantities.

There have been attempts in literature to address the above stated challenges in one form or another but mostly for spatial quantities in the environment. Attempts include using deterministic annealing [7], virtual springs [18] and Voronoi partition method [3]. Deterministic Annealing is based upon Simulated Annealing and utilizes the cooling cycle of Simulated Annealing to drive agents to minimal energy configurations. This method achieves exploration by using high temperatures to cause an increase in the energy of the system before slowly cooling it down. However, it relies on a high communication through put between the agents in order to synchronize the entire system. The method

Published by Elsevier. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial No Derivatives License (CC:BY:NC:ND: 4.0).
The final published version (version of record) is available online at 10.1016/S1672-6529(16)60339-6. Please refer to any applicable publisher terms of use.

of Virtual Springs developed by Shucker et al [18] uses mathematical equations that describe the length of a spring in order to control the spread of agents that are covering a spatio-temporal function. By tracking target points in the spatio-temporal function and then adjusting the length of the springs between agents, it was possible to form a visual representation of a diffusing target. However, this work relies on long distance communication and might not be possible on simplistic throwaway robotic agents that need to be cheaply designed and manufactured for maybe one time use.

The use of Voronoi partition method for optimal sensor placement in the environment is perhaps the most common and was pioneered by J.Cortes [3].It has been extended in various ways as in [15][8][10]. However, the fundamental problem of the Voronoi partition method is that it requires machine learning algorithms as well as an assumption of a sensing radius around each agent in order to estimate the distribution of the pollutant in the environment. This is due to its dependence on using the value of the mass density of the spatial function in each Voronoi cell that was generated by the robots positions [3][15]. From the mass density, the centroid position of each Voronoi cell is computed towards which the agents are moved to. Due to these requirements, a near accurate knowledge of the agent positions in the environment is needed as well as a communication network over which spatial sensor data and position of each agent can be broadcast to others in order to reach a consensus on the spatial function's distribution [17]. As a result, the Voronoi partition method could become intractable for very computationally simple agents. Furthermore, the requirements of the Voronoi approach could slow it down in real life when tracking a highly dynamic spatio-temporal profile and also lead to inaccurate sensor placements due to (i) errors in the estimation of the spatial function and (ii) delays in computation resulting in lateness to coming to an estimate of the spatial function. Moreover, the Voronoi approach lacks an exploration ability and must be combined with other schemes to provide this capability as in [15].

Nevertheless, Schwager et al were able to realize the use of Voronoi partition method on physical agents in [16] using distributed efficient computation of integrals for the Voronoi cell estimation as well as a multi-robot localisation system achieved over a proprietary communication network. Also in [8], the authors were able to apply the Voronoi partition approach to a network of mobile sensors connected via bluetooth to a centralised computer towards tracking a simulated dynamic spatio-temporal function.

In this paper, a dynamic spatio-temporal function is tracked and mapped using a distributed probabilistic framework inspired by the bacterium's self-organising behaviour on food sources. We claim the following contributions: **(i)** A distributed noise robust probabilistic algorithm that is computationally tractable and does not rely on future planning. **(ii)** The algorithm does not rely on kernel-based learning. It can however adapt its strategy based on observations from its environment. This makes it maximize its chances of providing coverage to a spatio-temporal function in the environment. **(iii)** In combination with a multi-agent controller, the algorithm is capable of achieving optimal coverage of a dynamic spatio-temporal function in the environment. This is achieved without the need to communicate spatial quantity readings to other agents in the network for reaching a consensus.

The rest of this manuscript is organised as follows: Section II provides the problem statement that we are aiming to solve, while in Section III, we briefly present our distributed probabilistic coverage controller inspired by the bacterium foraging action and show through mathematical analyses how it is capable of tracking a spatio-temporal function. This is followed by Sections IV and V in which we use physical experiments and simulations respectively to support the mathematical analyses carried out in Section III. We conclude in Section VI with a brief discussion and planned future work.

## II. Problem formulation

A spatio-temporal function can be modelled by using a dynamical spatio-temporal model [19] as shown in Equation 1.

$$C(X,t) = \begin{cases} M_i C(X,t-1) + n_i(t) & if f_i(\epsilon(t)) \in \gamma_i \\ : & \\ : & \\ M_k C(X,t-1) + n_k(t) & if f_k(\epsilon(t)) \in \gamma_k \end{cases} \tag{1}$$

Where $M$ is the propagation matrix whose value depends on the spatio-temporal process at the previous time $(t-1)$ as well as time-varying parameters [19]. $X \in D \subset D^2$, $C$ is the concentration of the pollution, $n$ is a noise term that is spatially dependent, $f_i(\epsilon(t))$ is some function of a time varying variable $\epsilon(t)$ and $f_k(\epsilon(t) \in \gamma_k$ gives the condition under which the $i$th Equation is appropriate. In summary, Equation 1 shows that the spatio-temporal function can take on any shape depending on the properties of the spatio-temporal function and environmental prevailing properties (such as wind, temperature, humidity and so on). An efficient coverage algorithm should be able to adapt to these changes in shape and move the agents appropriately so that the cost function $H_V$ in Equation 2 is minimized where $Z(X)$ is the population of agents and $C(X)$ is the spatio-temporal function.

$$H_V = \int_X \|Z(X) - C(X)\|^2 dx \tag{2}$$

## III. The Unicellular Bio-inspired Coverage Controller

### *A. Probabilistic controller based on Berg and Brown bacterium model*

A single bacterium motion is composed of two phases; a tumble and a run phase. The tumble phase involves changing its heading towards a completely new randomly chosen heading through the counter clockwise rotation of its flagellum while in the absence of disturbances, the run phase can be viewed as a straight line motion for simplicity. The bacterium finds its food by navigating up food chemical gradients. When the bacterium is making progress towards the food, it tumbles less thereby increasing the length of the run phase towards the source of the food. If moving in the wrong direction, it tumbles more often in order to reorient itself towards the food source. This was observed by Berg and Brown through experiments in [2]. By fitting the results obtained to a best fit equation in [1], a model was developed. The model describes the relationship between the tumble and run behavior of the bacterium during foraging and is depicted in Equations 3 to 5.

$$\tau = \tau_o exp\left(\alpha \frac{d\overline{P_b}}{dt}\right) \tag{3}$$

$$\frac{d\overline{P_b}}{dt} = \frac{1}{\tau_m}\int_{-\infty}^{t} \frac{dP_b}{dt^{'}} exp\left(\frac{t^{'}-t}{\tau_m}\right) dt^{'}, \tag{4}$$

$$\frac{dP_b}{dt} = \frac{k_d}{(k_d + C(x,t))^2} C(\Delta x, \Delta t) \tag{5}$$

where $\tau$ is the mean run time, $\tau_o$ is the mean run time in the absence of concentration gradients and $\alpha$ is an amplification constant of the bacterial system.

Assuming a spatial function $C$ in the environment $X$, $P_b$ is the fraction of the receptor bound of the agent when measuring concentration $C(x,t)$ at position $x$. In this work, $C(x,t)$ was the present reading taken by the robotic agent at position $x$. $k_d$ is the dissociation constant of the bacterial chemoreceptor and controls the chemical sensitivity of the bacteria. $\frac{dP_b}{dt}$ is the rate of change of $P_b$ and $\frac{d\overline{P_b}}{dt}$ is the weighted rate of change of $P_b$, while $\tau_m$ is the time constant of the bacterial system. $\frac{d\overline{P_b}}{dt}$ resembles an output from an exponential moving average low pass filter. This is especially useful in a bacterium system because it encounters noise in its dynamics, chemoreceptors, and environmental turbulence. The bacteria remembers the past $4$ seconds chemoreceptor readings [14] and is able to filter out high frequency components while still responding to immediate changes in its environment. $C(\Delta x, \Delta t)$ is obtained through the motion of the agent as it explores its environment. It is the difference between the measured concentration at $(x-1, t-1)$ and $(x,t)$.

The above equations determine the time between tumbles and hence the length of runs between tumbles. During the tumble phase, the agent can randomly choose an angle in the uniform distribution set $\sigma \in [0, ..., 360]$.

The above model can be cast into a probabilistic framework with the control policy $\pi$ determined by the run phase $\kappa$ and the tumble phase $\zeta$ as in Equation 6.

$$\pi : c(x)_{1:t-1}, u_{1:t-1} \longrightarrow u_t \in [\kappa, \zeta] \tag{6}$$

The expected cumulative payoff is the progress it makes towards a food source as depicted by increasing food concentrations and can be given by:

$$R_T(x_t) = E\left[\sum_{\lambda=1}^{T} \gamma^{\lambda} r_{t+\lambda} | x_t\right] \tag{7}$$

Where $E[]$ is the expectation of the model over future payoffs between the present time $t$ and future time $t+\lambda$. $r$ is the reward term and does not include the cost of the agent's motion. From the model in Equations 3 to 5. It is assumed that the discount factor $\gamma$ is give by Equation 4 so that

$$\gamma = \frac{1}{\tau_m}\int_{-\infty}^{t} \frac{dP_b}{dt^{'}} exp\left(\frac{t^{'}-t}{\tau_m}\right) dt^{'}, \tag{8}$$

Equation 8 ensures that the discount factor adapts with time depending on the sensory readings from the environment as given by Equation 5. As a result, this enables the agent to learn from it's environment and then uses the outcome to decide it's control policy appropriately. Furthermore, the nature of the discount factor ensures that it stays in the range of $[0:1]$. The $k_d$ term in Equation 5 is user defined and when the value is high, the discount factor is low making the agent focus on immediate payoffs. When $C(x,t)$ is high, the immediate payouts are more important making the agent stay in areas of high concentrations. However, this also makes it greedy so that there is a possibility that it can be trapped in local maximums.

Nevertheless, this predicament is partly solved by using past measurements up to a time horizon $T = t-4$ as shown in Equation 9.

$$R_T(x_t) = \sum_{t'=t-4}^{t} \gamma^{t'} r_{t'} | x_t \tag{9}$$

Due to the nature of Equation 9, the probabilistic scheme only needs to store the past four measurements and no future planning algorithm is needed. This makes the result algorithm tractable for simplistic agents and can be completed in polynomial time. Furthermore, Equation 9 results in delayed gratification and prevents the agent from being too greedy to make payoffs on immediate high sensory readings.

We define a value function of our scheme using Equation 10:

$$V_t(x) = \tau_o exp(R_T(x_t)) \tag{10}$$

In order to select a motion action from the control policy $\pi_{u_t} \in [\kappa, \zeta]$, we introduce an iterative cost term $\psi$ given by:

$$\psi_t = \psi_{t-1} + \Delta t \tag{11}$$

The cost term increases by time interval $\Delta$. The motion action to select is driven by the cost term according to Equation 13 which is closely linked to the spatial function's distribution.

$$\psi_t = \psi_{t-1} + \Delta t \tag{12}$$

$$motion = \begin{cases} tumble & \zeta \quad if V_t(x) < \psi_t \\ run & \kappa \quad v_o^i \end{cases} \tag{13}$$

### *B. Achieving Multi-agent coverage*

In this work, we will be focusing on a two dimensional scenario where $X \in D \subset D^2$. Using conditional independence, the organism's motion with respect to a spatial function's distribution can be described as follows:

$$[Z(X, t \to \infty), C(X)] = [Z(X, t \to \infty)|C(X)] \tag{14}$$

Where $Z(X, t \to \infty)$ is the motion of the bacterium in $X$ as time goes to $\infty$. Equation 14 is true for an organism's motion in a spatial function and can also be extended for its motion in a spatio-temporal function as in Equation 15 below where $C(X)$ changes distribution from time $t-1$ to $t$ after being released from $C(0)$ at time $t = 0$.

$$\begin{aligned}[Z(X,t), C(X,t)] = \prod [Z(X,t)|C(X,t)][C(X,t)| \\ C(X,t-1)][C(0)]\end{aligned} \tag{15}$$

Since a population of agents is required to provide coverage to a spatio-temporal function, Equation 15 becomes 16. Where $[Z(x_1,t), Z(x_2,t), Z(x_3,t).....Z(x_N,t)]$ is the population of agents and each agent performs a sensor measurement of the spatio-temporal function $C(X,t)$ as $[C(x_1,t), C(x_2,t), C(x_3,t).....C(x_N,t)]$.

$$\begin{aligned}&[Z(x_1,t), Z(x_2,t), Z(x_3,t).....Z(x,N,t), \\ &C(x_1,t), C(x_2,t), C(x_3,t).....C(x_N,t)] \\ &= \prod [Z(x_1,t), Z(x_2,t), Z(x_3,t).....Z(x,N,t)| \\ &C(x_1,t), C(x_2,t), C(x_3,t).....C(x_N,t)] \\ &[C(X,t)|C(X,t-1)][C(0)] \\ &= \prod_{i=1}^{N} [Z(x_N,t)|C(x_N,t)][C(t)|C(t-1)][C(0)]\end{aligned} \tag{16}$$

However, in real world applications, sensor measurements are often corrupted by noise. As a result, the sensor measurements $C(x_i,t)$ of each agent $i$ in the population can be described as Equation 17 where $H(x_i,t)$ is the observation matrix, $\epsilon(t)$ is the additive Gaussian noise and $d(t)$ is a bias term of the sensor.

$$S(x_i,t) = H(x_i,t)C(x_i,t) + \epsilon(t) + d(t) \tag{17}$$

The bacterium uses its exponentially weighted short term memory in Equation 4 to filter out noise. As a result, in the best scenario, the final distribution of the agents $[Z(X, t \to \infty), C(X)]$ will be made up of the real signal $C(x_i,t)$ and the bias term $d(t)$ while in a worst case scenario, the final distribution would have component related to the noise component $\epsilon(t)$ as well. Nevertheless, we will show later in experiments that with an off the shelf sensor, our proposed method works in addition to dealing with errors in dynamics of the agent caused by the occasional wheel slippage.

The term $[C(X,t)|C(X,t-1)]$ in Equation 16 captures the time varying property of the spatio-temporal function. As shown in Equation 16, the spatio-temporal function and the agents distribution are closely linked and as a result desirable for achieving dynamic coverage of a spatio-temporal function.

### *C. Preventing agent collisions by introducing a Flocking Controller*

In deployment of our proposed controller on a swarm of robotic agents, collision is very likely especially when agents move into favorable highly concentrated areas. In order to prevent undesirable collisions, a flocking controller is introduced. The flocking behaviour was necessary to prevent collisions with agents in the flock. The flocking controller was combined with the bacterium controller in a behaviour based architecture as shown in Figure 1 below.

Ignoring the tumble phase for simplicity, the dynamics of the agent using the bacteria behavior can be described using Equation 18.

$$X_b^i(t + \Delta t) = x_b^i(t) + v_b^i(t) \tag{18}$$

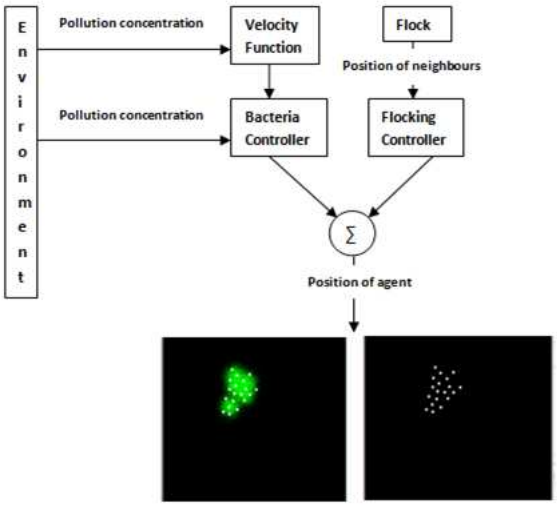



Fig. 1. Architecture of our bacterium inspired coverage controller.

$v_b^i(t)$ is given by Equation 19, where $v_o^i$ is the velocity when concentration $C(X) = 0$.

$$v_b^i(t) = \begin{cases} \frac{v_o^i}{C(x,t)} & if C(x,t) > 0 \\ v_o^i \end{cases} \tag{19}$$

Using Equation 19 preserves the ability of our proposed method to achieve Equation 16.

Nevertheless, the use of a velocity function such as Equation 19 affects the level of coverage produced by the agents [13] and could cause a problem especially when $C(x,t)$ is high enough to cause an agent stall. This is one of the major weaknesses of our approach and requires a bit of domain knowledge before deployment. Nevertheless, one of the ways this problem could be mitigated against is by using the property of the pollution detection sensor. All physical sensors have an upper limit (e.g 255 for a digital sensor that has a resolution of 8 bit) beyond which they can not read. This upper limit could be used as the velocity constant $v_o^i$ value whilst taking into account the maximum velocity of the agent. Furthermore, the run and tumble motions of the agent result in a rich exploration ability. Nevertheless, the agents' movements are often random and might be viewed as erratic. As a result, it could be said that the bacterium controller is more suited for agents which are unbounded in any direction (such as a UAV) compared to wheel robots. Nevertheless, we have been able to adapt it to wheel robots in our previous works of [9][11].

For the flocking behavior, the morse potential is used and this is given by Equation 20.

$$v_f^i(t + \Delta t) = G_G[G_A - G_R]exp\left(-\frac{r_{ij}}{k}\right) \tag{20}$$

where $G_A$ is the attraction gain, $G_R$ is the repulsion gain, $G_G$ is the gain that controls the amount of difference between both gains, $r_{ij}$ is the distance of the closest neighbor $j$ to an agent $i$ and $k$ is a constant that can be used for tuning how close the neighbor $j$ gets to agent $i$.

Combining both the velocity from the bacteria behavior and the flocking behavior results in Equation 21.

$$X^i(t + \Delta t) = X^i(t) + \gamma v_b^i(t) + \zeta v_f^i(t) \tag{21}$$

where $\gamma$ is the gain of the bacteria behavior and $\zeta$ is the gain of the flocking behavior. In experiments conducted, the values of $\gamma$ and $\zeta$ were set to 0.8 each after trial and error.

### D. Simulation assumptions

In our simulations, we made the following assumptions:
**(i)** Agents have a point mass dynamics given by Equation 22. The idea is that the algorithms developed in this paper would sit upon an underlying controller such as the one developed in our previous work of [12] for controlling Unmanned Aerial Vehicles towards providing aerial 3D visual representation of an hazardous pollutant.

$$\begin{aligned} \dot{x} &= (v_b + v_f)cos(\sigma) \\ \dot{y} &= (v_b + v_f)sin(\sigma) \end{aligned} \tag{22}$$

**(ii)** At this stage of our work, agents' motions or turbulence from their propellers do not affect the distribution of the pollutant as they move within it.
**(iii)** Agents know their location precisely. Presently, this can be achieved by using an accurate GPS. Nevertheless, our algorithm does not rely heavily on this assumption. This assumption is used only to enable the flocking controller to achieve cohesion and repulsion. In order words, if there was a range sensor such as a radar, vision, or laser, then the algorithm would not need GPS. It will only need to know the proximity of agents in its immediate vicinity.
**(iv)** Communication is possible with only immediate neighbours closer than a radius $r$ of fifty pixels **(v)** Each agent has a flocking memory limited to five elements. This is used to store the received positions of agents for flocking computations. It also has a four element memory dedicated to storing last four spatial measurements.

## IV. Experiments with Bacterium Component

In this section, results of investigation carried out on a robotic physical platform in an arena as shown in Figure 2 are presented.

The arena had dimensions of 1200mm by 1400mm. This investigation was conducted in order to ascertain if the probabilistic coverage controller would converge to the distribution of a simulated spatial function according to Equation 14. The spatial function was simulated by printing a gradient of black color on paper as shown in Figure 2 with the source placed

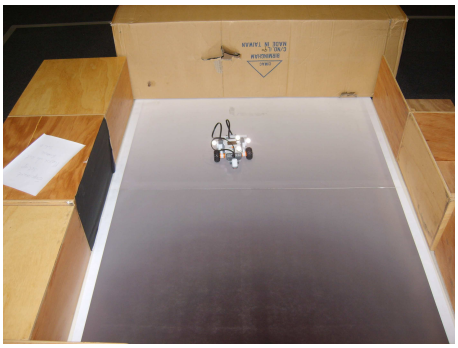
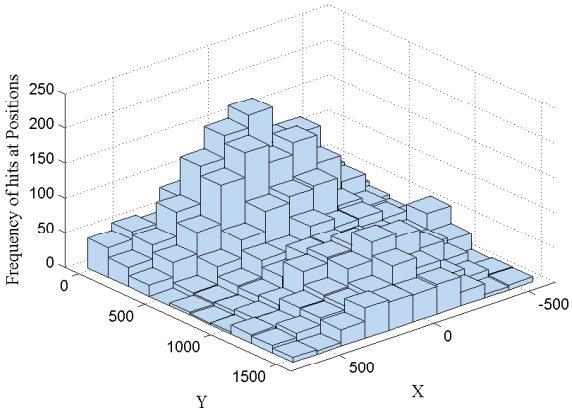
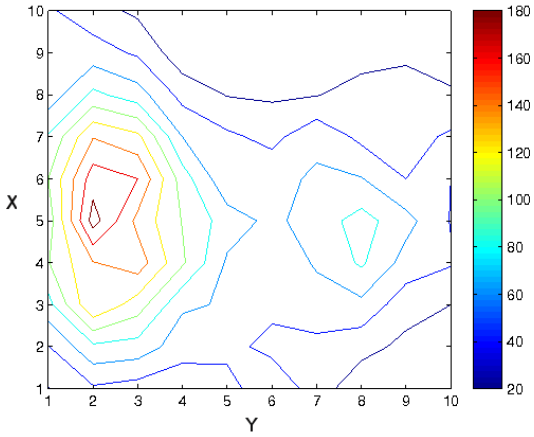


Fig. 2. Lego mindstorm platform in the arena with little background light. The paper in the arena is the simulated pollutant obtained by printing a gradient of black ink on paper. We use the black ink's gradient to simulate a pollutant.




(a) Histogram showing frequency of positions covered by agent when $\tau_o$ = 2. The units of the axis are in millimetres.




(b) Normalised Contour Plot of positions covered by agent when $\tau_o$ = 2. The red side of the scale of the isohype curve indicate higher frequency of coverage while the blue end of the scale indicate lesser frequency.

Fig. 3. Spatial coverage of the simulated pollutant by the robotic agent using $\tau_o$ parameter value of 2,$\alpha = 1000$ and $k_d = 2$.

at a position of $(0, 0)$ in the arena. An infra red light sensor was used to read values from the paper and the readings used accordingly. The robot had a differential drive system with a velocity of 18cm/sec. The effects of $k_d$ and $\tau_o$ parameters were tested in this section in order to investigate their effect on the coverage provided by the controller. For each change in parameter, 20 readings were collected. Each experiment was allowed to run for 120 seconds each.

### A. The $\tau_o$ parameter

Equation 10 and 13 show that a large $\tau_o$ value would result in an agent covering more distance during a run phase before changing direction. This would result in more exploration of the environment. In order to test this hypothesis, experiments were carried out using $\tau_o$ values of 2 and 20 with $k_d$ = 2 and $\alpha$ = 1000. Results shown in Figures 3 and 4 show that as $\tau_o$ increases, the coverage provided by the agent increases thereby confirming the raised hypothesis.

### B. The $k_d$ parameter

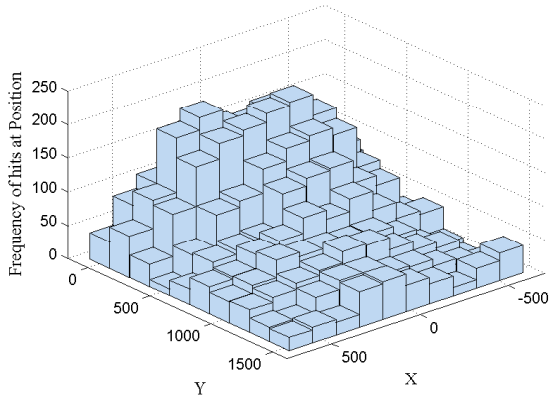
Equation 5 to 9 show that a high $k_d$ would result in a low discount factor $\gamma$ This would cause the agent to make immediate payoffs on detected spatial quantities. As a result, a high $k_d$ would cause the agent to "stick" to areas of higher spatial concentrations. To test this hypothesis, $k_d$ values of 10 and 30 were used with $\alpha$ = 1000 and $\tau_o$ = 2. As seen when comparing Figures 5 and 6, this is the case. A higher $k_d$ causes the agent to ignore lower values of $C(X)$ and cluster around the "source" of the chemical field.

## V. Mapping Spatio-Temporal Functions

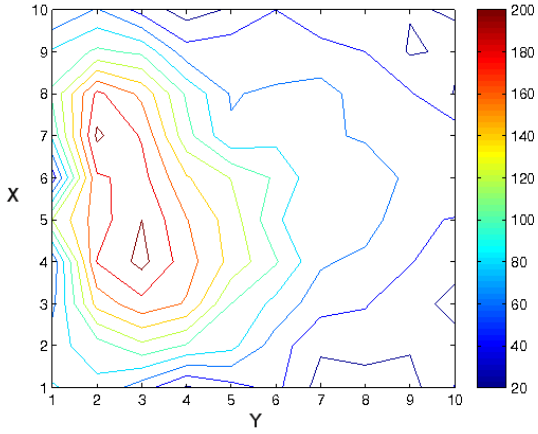
In the previous section, it was shown that a single agent can be used to form a spatial function distribution while in [11], we show that we can use a swarm of agents to form a spatial function represented by a light source. In both cases, the spatial function did not have spatio-temporal properties. In this section, we show that our proposed approach is capable of dynamically tracking a spatio-temporal function in the environment. This would show that Equation 16 holds true.

In order to test whether this is the case, a simulated spatio-temporal function was used. The spatio-temporal function was based upon the video of a black ink dispersed into water as shown in Figure 7.

Frames were taken from the video and the RGB value of each pixel extracted. This value was then compared to a threshold. When the RGB value was below the threshold, 0 was assigned to that location or one otherwise. The amount of spatio-temporal quantity at a location was measured by reading the number of particles at a location and this value was then used by our algorithm. The spatio-temporal distribution $C(X, t)$ evolved every 100 iterations while the speed of the agents were constrained to a value of 10 pixels per

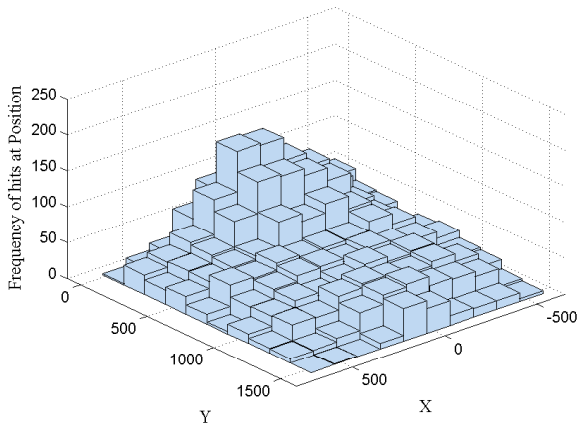



(a) Histogram showing frequency of positions covered by agent when $\tau_o$ = 20. The units of the axis are in millimetres.

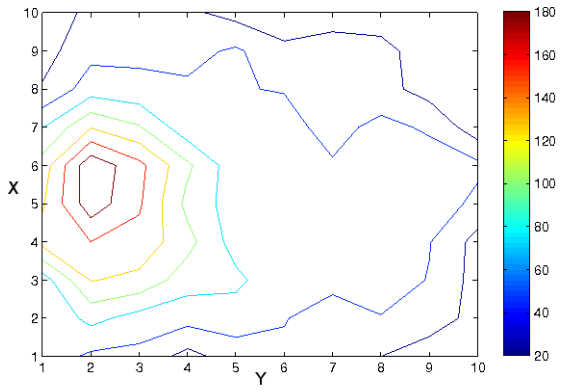



(b) Normalised Contour Plot of positions covered by agent when $\tau_o$ = 20. The red side of the scale of the isohype curve indicate higher frequency of coverage while the blue end of the scale indicate lesser frequency.

Fig. 4. Spatial coverage of the simulated pollutant by the robotic agent using $\tau_o$ parameter value of 20,$\alpha = 1000$ and $k_d = 2$.




(a) Histogram showing frequency of positions covered by agent when $k_d$ = 10. The units of the axis are in millimetres.




(b) Normalised Contour Plot of positions covered by agent when $k_d$ = 10. The red side of the scale of the isohype curve indicate higher frequency of coverage while the blue end of the scale indicate lesser frequency.

Fig. 5. Spatial coverage of the simulated pollutant by the robotic agent using $k_d$ parameter value of 10, $\alpha = 1000$ and ,$\tau_o = 2$.

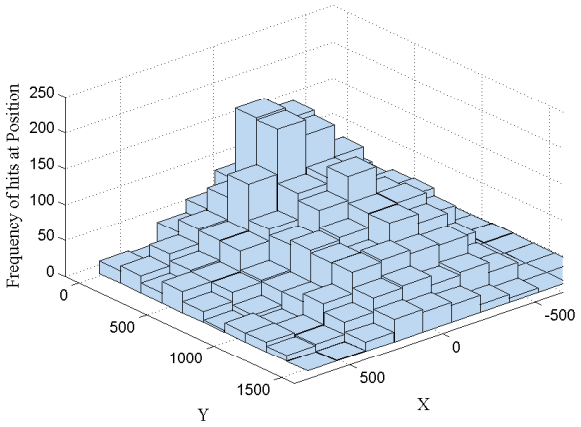
iteration.

The agents were blue when they did not detect any spatial reading and red when they detected the spatio-temporal function. This method of using colors to indicate various pollutants has been used in [6] and could be used to show humans which areas are safe and those that are not. In the Figures 8(a) to 8(c), the green object is the simulated spatio-temporal function obtained from the video frame, while the yellow objects are depicting an electronic map at a base station where the agent positions are used to generate randomly placed particles using the agents positions as the mean point. This gives a higher resolution mapping of the pollutant. The results shown in Figures 8(a) to 8(c) show that the algorithm is capable of following the distribution of the spatio-temporal function.

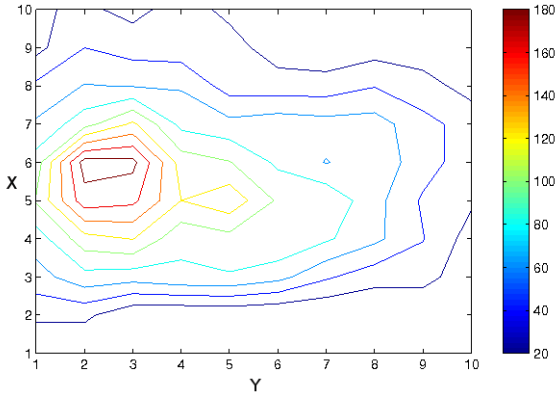
## VI. Conclusion

In this work, we have developed a distributed probabilistic algorithm that does not rely on future planning or machine learning to track and provide coverage to a dynamic spatio-temporal process. The nature of our algorithm ensures that it is computationally tractable and can be used on simplistic agents. Furthermore, we show that the algorithm is capable of adapting to its environment thereby maximising its chances of providing coverage to a spatio-temporal process in the environment. We were able to successfully deploy our algorithm on a physical agent in order to validate our approach.

In combination with a multi-agent controller, we showed through simulations that the probabilistic algorithm can be used to track and provide coverage of a dynamic spatio-temporal process without the need for the agents to each




(a) Histogram showing frequency of positions covered by agent when $k_d$ = 30. The units of the axis are in millimetres.




(b) Normalised Contour Plot of positions covered by agent when $k_d$ = 30. The red side of the scale of the isohype curve indicate higher frequency of coverage while the blue end of the scale indicate lesser frequency.

Fig. 6. Spatial coverage of the simulated pollutant by the robotic agent using $k_d$ parameter value of 30,$\alpha = 1000$ and ,$\tau_o = 2$.



(a) Initial plume. (b) Intermediate plume.



(c) Matured plume.

Fig. 7. Frames of a video showing a plume of black ink released into water.

a consensus on the spatio-temporal process' distribution. In the future, we plan to deploy the multi-agent distributed probabilistic algorithm on physical agents towards tracking and mapping of true physical spatio-temporal processes.

## REFERENCES


[1] Howard C Berg and Edward M Purcell, *Physics of chemoreception*, Biophys. J **20** (1977), 193–219.

[2] Douglas A Brown and Howard C Berg, *Temporal stimulation of chemotaxis in Escherichia coli*, Proc. Natil. Acad. Sci. U.SA **71** (1974), 1388.

[3] Jorges Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo, *Coverage control for mobile sensing networks*, IEEE Transactions on Robotics and Automation **20** (2004), no. 2, 243–255.

[4] Brian R Johnson, *Pattern formation on the combs of honeybees: increasing fitness by coupling self-organization with templates*, Proceedings of the Royal Society: Biological sciences **276** (2009), no. 1655, 255–261.

[5] Christian Jost, Julie Verret, Eric Casellas, Jacques Gautrais, Mélanie Challet, Jacques Lluc, Stéphane Blanco, Michael J Clifton, and Guy Theraulaz, *The interplay between a self-organized process and an environmental template: corpse clustering under the influence of air currents in ants.*, Journal of the Royal Society, Interface **4** (2007), no. 12, 107–16.

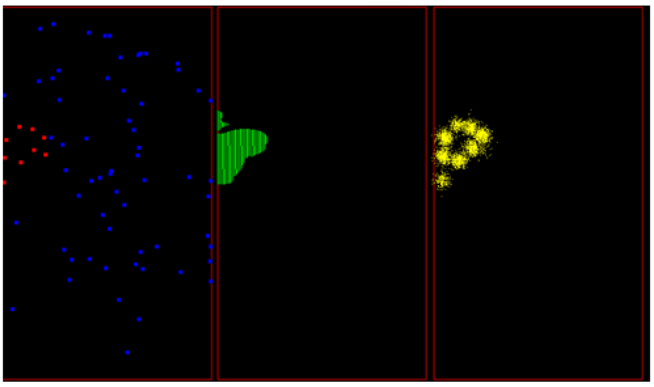
[6] Stacey Kuznetsov, George Noel Davis, Eric Paulos, Mark D. Gross, and Jian Chiu Cheung, *Red Balloon , Green Balloon , Sensors in the Sky*, UbiComp'11, Sep 17-Sep 21, Beijing, China, 2011.

[7] Andrew Kwok and Sonia Martinez, *A distributed deterministic annealing algorithm for limited-range sensor coverage*, Proc. of American Control Conference,St. Louis, Missouri, USA (2009), no. 1, 1448–1453.

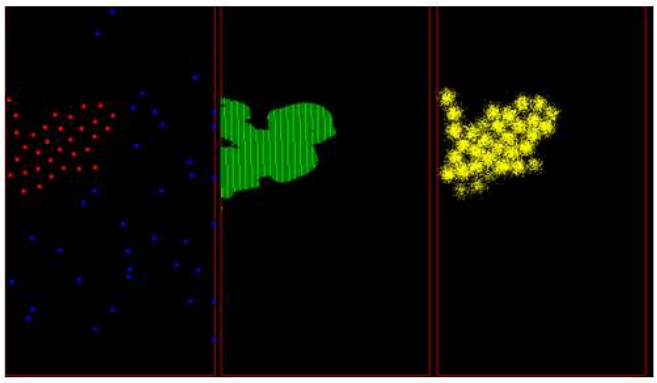
[8] Bowen Lu, Dongbing Gu, and Huosheng Hu, *Tracking and modeling of spatio-temporal fields with a mobile sensor network*, Intelligent Control and Automation (WCICA), 2014 11th World Congress on, IEEE, 2014, pp. 2711–2716.

[9] Bowen Lu, John Oyekan, Dongbing Gu, Huosheng Hu, and Hossein Farid Ghassem Nia, *Mobile sensor networks for modelling environmental pollutant distribution*, International Journal of Systems Science **42** (2011), no. 9, 1491–1505.

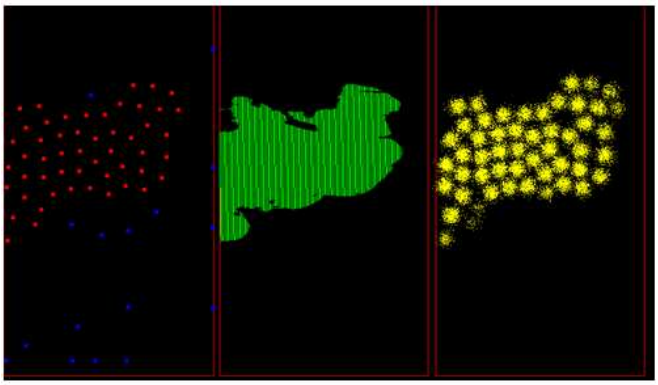
[10] Hamid Mahboubi, Amir G Aghdam, and Kamran Sayrafian-Pour, *Area*



(a) Agents distribution in initial black ink plume.



(b) Agents distribution in intermediate black ink plume.



(c) Agents distribution in matured black ink plume.

Fig. 8. Various stages in the coverage of a simulated spatio-temporal function $C(X, t)$ with velocity of agents = 10 pixels per iteration. In the left pane, red agents show the polluted areas and blue agents the safe areas. The central pane shows the simulated pollutant while the right pane shows a simulated electronic map at a base station made by using a Gaussian function to randomly distribute particles around each agent's position.


*coverage in a fixed-obstacle environment using mobile sensor networks*, Control and Systems Engineering, Springer, 2015, pp. 135–151.

[11] John Oyekan, Dongbing Gu, and Huosheng Hu, *Visual imaging of invisible hazardous substances using bacterial inspiration*, Systems, Man, and Cybernetics: Systems, IEEE Transactions on **43** (2013), no. 5, 1105–1115.

[12] John Oyekan and Huosheng Hu, *Towards autonomous petrol behaviours for uavs*, Proceedings of UK EPSRC Workshop on Human Adaptive Mechatronics, Staffordshire University, Stafford, UK, 2009, pp. 15–16.

[13] John Oyekan, Huosheng Hu, and Dongbing Gu, *Exploiting bacterial swarms for optimal coverage of dynamic pollutant profiles*, Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on, IEEE, 2010, pp. 1692–1697.

[14] Kevin M. Passino, *Biomimicry of bacterial foraging for distributed optimization and control*, IEEE Control Systems Magazine (2002), no. June, 52–67.

[15] Mac Schwager, Francesco Bullo, David Skelly, and Daniela Rus, *A ladybug exploration strategy for distributed adaptive coverage control*, Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, IEEE, 2008, pp. 2346–2353.

[16] Mac Schwager, James McLurkin, and Daniela Rus, *Distributed coverage control with sensory feedback for networked robots.*, robotics: science and systems, 2006.

[17] Mac Schwager, Jean-jacques Slotine, and Daniela Rus, *Decentralized, Adaptive Control for Coverage with Networked Robots*, IEEE Int. Conf. on Robotics and Automation (2007), 3289–3294.

[18] Brian Shucker, Todd Murphey, John K Bennett, and Senior Member, *Convergence Preserving Switching for Topology Dependent Decentralized Systems*, IEEE Transactions on Robotics **24** (2008), 1–11.

[19] Christopher K Wikle and Mevin B Hooten, *A general science-based framework for dynamical spatio-temporal models*, Test **19** (2010), no. 3, 417–451.



https://dspace.org/

# Tracking and sensor coverage of spatio-temporal quantities using a swarm of artificial foraging agents

**Oyekan, John; Gu, Dongbing; Hu, Huosheng**

**2016-11-09**
Elsevier
http://doi.org/10.1016/S1672-6529(16)60339-6
Oyekan J, Gu D, Hu H. (2016) Tracking and sensor coverage of spatio-temporal quantities using a swarm of artificial foraging agents. Journal of Bionic Engineering, Volume 13, Issue 4, October 2016, Pages 679–689

Downloaded from a DSpace repository. https://dspace.org/