

Biologically-inspired Self-deployable Heterogeneous Mobile Sensor Networks

Lidan Miao and Hairong Qi

Department of Electrical and Computer Engineering
University of Tennessee
Knoxville, TN 37996, USA
{lmiao1, hqi}@utk.edu

Feiyi Wang

Advanced Network Research
MCNC Research & Development Institute
Research Triangle Park, NC 27709, USA
fwang2@anr.mcnc.org

Abstract—This paper studies the problem of self-deployment of heterogeneous mobile sensors using biologically-inspired principles and methodologies. The initial sensor deployment is assumed to be random, based on which two interrelated issues are investigated: the design of an optimal placement pattern of heterogeneous sensor platforms and the self configuration from the initial random state to the optimal state through intelligent sensor movement. We first develop an optimal placement algorithm based on the mosaic technique inspired by the retina mosaic pattern widely observed in both human and many animal visual systems. Different types of mobile sensors are organized into a mosaic pattern for both maximizing network coverage and reducing network cost. Secondly, in order to converge to the optimal state, we investigate the swarm intelligence (SI)-based sensor movement strategy with the assistance of local communications, through which the randomly deployed sensors can self-organize themselves to reach the optimal placement state. The proposed algorithm is compared with the random movement and the SI-based method without direct communication using performance metrics such as network coverage, redundancy, convergence time, and energy consumption. Simulation results are presented to demonstrate the effectiveness of the mosaic placement and the SI-based movement with local communication.

Index Terms—Heterogeneous sensor network, Coverage, Mosaic pattern, Swarm intelligence

I. INTRODUCTION

In order to monitor activities within a region of interest (ROI), a number of sensors with capabilities of sensing, processing, communication and even mobilization can be deployed to perform collaborative tasks. The problem of sensor deployment then becomes an important and fundamental issue in sensor networks as the location of individual sensor node can largely affect the effectiveness of collaboration. An optimal deployment algorithm is desired that aims at minimizing energy consumption, reducing convergence time, maximizing network coverage, while using as few sensors as possible to minimize cost.

Sensor deployment algorithms vary with different application scenarios. The simplest case is, when the environment is sufficiently known and accessible. The sensor locations can then be predetermined and the deployment is performed manually so that the precise placement of sensors is possible. To this end, many investigators have concentrated on the design of spatial arrangement of sensor nodes to achieve

the desired detection performance [1], maximize the system lifetime [2], and ensure connected coverage [3]. Uncertainty-aware sensor deployment is discussed in [4], which takes account of inherent uncertainties associated with the predetermined sensor locations.

When the environment is unknown or inaccessible, e.g. some hazardous regions or adversarial environment, it is not practical to manually deploy sensors, let alone sensor networks of a large scale. Hence, in order to quickly form a sensor network, the most efficient deployment method might be airdrop. If we refer to the placement of sensor nodes after airdrop as the “initial state” of the network, then this initial state is almost certain not the “optimal state”. In order to reach a reliable coverage provided at the optimal state, we have to rely on mobile sensors to self-organize themselves and eventually converge to the optimal placement. This problem has been considered by a number of researchers, and various techniques like the potential fields [5], [6], the virtual force [7], and the Voronoi-based [8] deployment schemes have been thoroughly investigated. However, these algorithms either are based on the centralized control [5], [7], or require the sensors to stay within a certain distance with one another so that they can use the neighboring node information for further deployment of other sensors [5] or for constructing the Voronoi polygon [8]. The centralized control is not energy-efficient due to full communication between the sensor nodes and the cluster head or the base station. The failure of the base station would lead to the failure of the whole system. Furthermore, the centralized approach is not feasible for large scale networks. On the other hand, the dependence on the neighboring sensors, which is based on the assumption that there is no “isolated” sensor after the initial placement, is not realistic in real world applications.

In most existing deployment strategies, a common assumption is that the sensor platform is *homogeneous*, that is, all the nodes in a sensor network are exactly the same. In real-world applications, a sensor network usually consists of a large number of sensor nodes with various sensing and processing capabilities. In the context of a surveillance network, we define a *heterogeneous sensor network* as one that consists of sensor nodes that carry different types of or different combinations of sensing modalities. For example, a

sensor node may carry a microphone for acoustic sensing, a geophone for seismic sensing, a magnetic sensor, a sonar, or a combination of different modalities.

In this paper, a *heterogeneous* sensor network with *self-deployable* capability is investigated. We present a distributed algorithm enabling mobile sensors to self-organize themselves to achieve efficient coverage. The initial deployment is assumed to be random which is the case in many practical applications. The problem of optimal sensor deployment has to consider two interrelated issues, the derivation of an optimal state for *sensor placement*, and the convergence from the “initial state” to the “optimal state” by *sensor movement*.

The optimal placement of heterogeneous sensor nodes in a sensing field resembles the phenomena of different photoreceptors tiled on the retina of both human and many animal visual systems. In the human visual system, three types of cones with absorbance maxima in the long-, middle-, and short-wavelength (L, M, and S cones) region, respectively, are organized into mosaics on the retina [9], [10]. Only a single type of photoreceptor samples the image at any given location, while the reconstructed image presents full color information at each pixel. Inspired by this biological phenomena, we extend the mosaic technique for use in sensor placement, in which different types of sensor nodes form a mosaic pattern to provide reliable coverage with minimum cost. Following how the insects self-organize themselves into a harmonious society [11], [12], we present a swarm intelligence (SI)-based movement algorithm. Each sensor can move freely within the sensing field, guided by a simple rule carried by the sensor. The individual behavior of different sensors leads the whole sensor network to converge to the optimal state. To speed the convergence, local communication is considered to conserve sensor energy. Each sensor behaves independently according to its own knowledge base. The effectiveness of the deployment algorithm is measured in terms of network coverage, redundancy, convergence time, and energy consumption.

In the following, we first discuss the design of optimal placement in Sec. II. We then present the SI-based movement strategy in Sec. III. Sec. IV discusses the performance metrics and Sec. V describes a series of experimental results. The paper is concluded in Sec. VI.

II. OPTIMAL PLACEMENT OF HETEROGENEOUS SENSOR PLATFORMS

In order to solve the problem of optimal placement of heterogeneous sensors using the mosaicked technology, three questions have to be answered, (1) what types of different sensors should be deployed for a given task? (2) how many sensors of each type should be deployed? and (3) how to distribute each type of sensors spatially to achieve maximum coverage using minimum number of sensor nodes?

The answer to the first question is mission-dependent. In a sensor network, different types of sensors are specialized in detecting different phenomena. Hence for a specific mission, the configuration of sensor platforms should be different. For example, if the mission is to detect civilian vehicles, then

the most useful sensing modalities might be microphone, geophone and passive infrared (pir). In this paper, we assume that the different types of sensor nodes have been specified based on different applications.

The answer to the second question is referred to as the problem of determining the *probability of appearance* (POA), which has a major role in generating an efficient placement pattern. Take the same civilian target recognition application as an example, among the three different sensing modalities, microphones are more effective in detecting civilian vehicles than geophones or pir sensors. Therefore, we might want to deploy more nodes that carry microphones than nodes that carry geophones or pir sensors. We use r to represent the ratio of the amount of different sensor nodes. For example, $r = 1/2 : 1/4 : 1/4$ indicates that the number of microphones is twice as many as the number of geophones and pir sensors. In this paper, we use this ratio as an important factor to determine the optimal pattern. However, we do not consider how this ratio can be obtained as it is also highly mission-oriented.

The answer to the third question concerns the spatial distribution of the sensor nodes. In the animal society, the spatial arrangement of cone mosaic has been examined in a variety of species, and it has been found that the mosaic array of most vertebrates is regular. Those animals who need high acuity and rely heavily on vision possess a very regular mosaic array, such as fish [13], [14] and mouse [15], [16]. Similarly, in sensor networks, a promising and intuitive solution of sensor deployment is to follow the *uniform distribution*. If a certain type of sensor nodes distribute densely in some regions while sparsely in others, the quality or coverage of the surveillance will be affected, resulting in certain sparsely deployed regions not being covered. On the other hand, the redundant data reported from densely deployed regions would also increase the network load. Therefore, the distribution of each type of sensor nodes should be as uniform as possible.

Based on these design concerns, we develop a generic algorithm to generate optimal mosaic placement given the different sensor nodes and their corresponding POAs. We represent a large surveillance area covered by a sensor network as a rectangular grid, and each grid point can then be interpreted as a site for the mobile sensors. The distance between two adjacent grid points corresponds to the unit sampling distance of the sensing field. A sensor placement example with three different types of sensors (denoted by different colors) on an 8×8 grid is shown in the right figure.

We choose the checkerboard pattern as the starting point because of its symmetry, uniform distribution, and the same sampling frequency along the two orthogonal directions (i.e. horizontal and vertical directions). Given the number of different types of sensor nodes and the POA of each type, the generic algorithm uses a combination of checkerboard decomposition and subsampling to generate the optimal pattern that satisfies the pre-defined POAs. This process can be

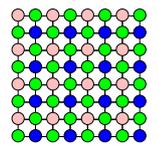


Fig. 1. A 3-type network

described using a binary tree, with the checkerboard pattern as the root of the tree and the leaf nodes as the different types of sensor nodes. The POA of each leaf node is given by $1/2^{level}$, where $level$ refers to the level of the binary tree.

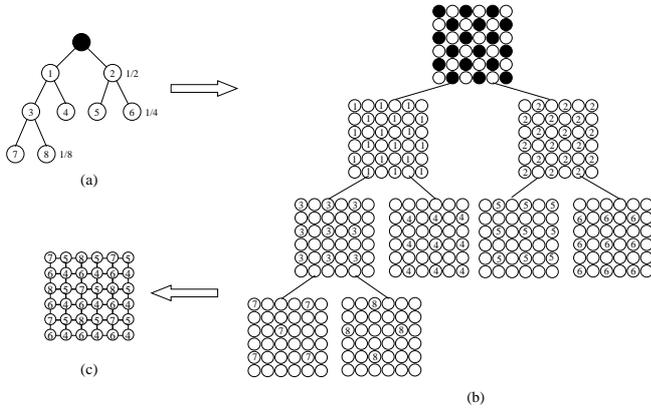


Fig. 2. Illustration of the generation process of the optimal placement pattern. (a) Generated binary tree. (b) Checkerboard separation. (c) A 5-type optimal sensor placement pattern generated by combining all the leaf patterns in (b).

Figure 2 illustrates the generation of a mosaic pattern of five different types of sensor nodes using a binary tree with five leaves (Figure 2(a)). Based on this binary tree, we separate the checkerboard pattern as shown in Fig. 2(b). The decomposition process is applied at the even levels of the binary tree (including level zero, i.e. the root), and the subsampling process the odd levels. The function of decomposition is to divide the black and white blocks in the original pattern into two sub-patterns such as the “label-1” and “label-2” pattern illustrated in Fig. 2(b). The decomposition of intermediate nodes needs some extra steps as exemplified in Fig. 3, where node “3” at level two of the binary tree is decomposed into node “7” and node “8”. The first step is to extract all the labeled pixels to form a new smaller pattern (S_1), then a checkerboard based on the resulted smaller pattern is constructed (S_2), and finally the black and white blocks are extracted separately and assigned to their original locations to form new patterns as illustrated by S_3 and S_4 in Fig. 3.

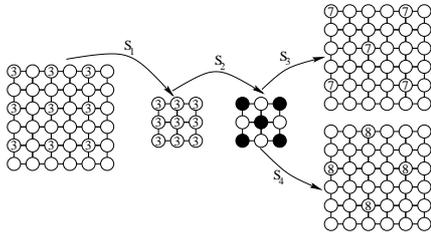


Fig. 3. Illustration of decomposing an intermediate node.

The subsampling process is to sample the pattern by $2^{\frac{level+1}{2}}$ along the horizontal and vertical directions. For example, the label “3” and label “4” patterns are obtained by

subsampling pattern “1”, and label “5” and label “6” patterns are the results of subsampling pattern “2” by 2.

Process the checkerboard until it has the same structure as the binary tree. The next step is to combine all the leaves to generate a mosaic pattern, as shown in Fig. 2(c), in which the pattern is obtained by combining all the leaf patterns in Fig. 2(b). If we replace the pixels labeled with different numbers by different types of sensor nodes, we actually generate a 5-type sensor placement pattern.

The binary-tree-driven pattern generation process would guarantee the uniform distribution of each type of sensor node across the whole sensing field. What is more important, given the POA, a binary tree is uniquely generated, based on which the location of different types of sensor nodes can be derived with some general rules. The only restraint is that the ratios of different sensor nodes are limited to power of two, which may not be the case in real world application. However, we can always choose the closest approximation to substitute the real ratio.

III. SWARM INTELLIGENCE (SI) BASED MOVEMENT

In order to converge to the optimal placement presented in Sec. II, we propose an SI-based movement mechanism, where the individual behavior of different sensors results in the collaborative efforts of the whole system. The movement strategy attempts to enable the network converge to the optimal topology with the least convergence time, movement distance, and energy consumption.

In recent years, development in SI has motivated researchers to find biologically emulated strategies to solve complex system problems. The research on SI is based on the idea that simple rules carried by individual entity can lead to a complex behavior of the whole system [11]. The behavior of social insects has inspired the development of different algorithms [12] applied to different problem domains, such as the travelling salesman problem [17], [18], the sequential ordering problem [19], the quadratic assignment problem [20], and the network routing problems [21], [22].

In this paper, we apply SI to the mobile sensor deployment problem. Each sensor node is considered as an individual entity that carries a simple rule. The limited capability of each sensor leads to an optimal configuration of the whole sensor network. As we will show in Sec. IV, the SI-based method is able to converge to the optimal state with rapid convergence and efficient energy conservation.

It is known from the sensor placement design that the distributions of different sensor platforms present regular pattern. If we set up an orthogonal coordinate system with the origin at the top-left corner of the grid and the distance between two grid points being unit, we can specify the positioning criteria (stop condition) according to the coordinates of sensors, which satisfy a set of unique constraints for different sensor platforms. Figure 4 shows an example of a sensor network with seven different sensor nodes, in which the x and y are the coordinates of a sensor. Each sensor carries the stop condition of its own sensor type. At each grid point, the sensor checks

the stop condition and determines whether it should position itself or go somewhere else. We assume each sensor should know whether the current position has been occupied or not.

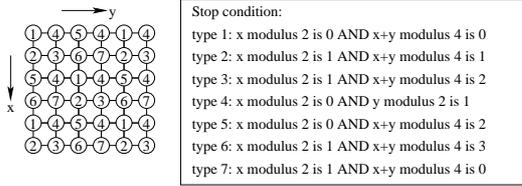


Fig. 4. Positioning criterion of a sensor network with seven different platforms.

For a team of sensors, an important issue is to determine which level of intelligence each sensor should have in order to present swarm intelligence. Low level of intelligence of each sensor might not be enough to manifest *swarm* intelligence, or at the cost of consuming more system sources. On the other hand, high intelligence is normally infeasible for a simple sensor node. In the context of this paper, we consider low intelligence as that each sensor only carries its stop condition. At each location, the sensors randomly choose the next target position. We expect the convergence time to be very long due to the completely random movement. We name this kind of movement the *random movement*.

If we add another level of intelligence, that is, at a certain sensor site, the sensor knows which neighboring sensor site is or closer to its destined sensor location, then it can go directly to that direction without random walk. By confining the moving direction, we expect to reduce the deployment time as well as the energy consumption during movement compared to the random movement strategy. We call this movement strategy the *SI-based movement* due to the increased intelligence level. An 8-direction movement is considered: north (N), south (S), east (E), west (W), north-east (NE), north-west (NW), south-east (SE) and south-west (SW). Each sensor carries a simple rule, based on which, the sensor should know which direction to go. Algorithms 1-3 demonstrate the rules that should be carried by different sensors in a 3-type sensor network, as shown in Fig. 1 ($r = 1/2 : 1/4 : 1/4$), where the red, green, and blue colors represent the type 1, 2 and 3 sensors, respectively.

In the random and the SI-based movement strategies, no direct communication is needed. Before a sensor reaches its target location, it has no information about whether the target location has been occupied or not. If the target position has already been occupied, the sensor has to move to other locations, which consumes extra energy. To alleviate the unnecessary energy consumption, we allow localized communication between sensors. All the positioned sensors (sensor that find their final locations) broadcast their coordinate to inform that the corresponding positions have been occupied. At each grid point, the unpositioned sensor first checks the occupation of its neighbors. By doing so, the sensor can then move to a vacant sensor site. If all the neighboring sensor locations

Algorithm 1: Rules carried by type 1 sensors.

```

Data : Node type 1 (R), current position  $(x, y)$ 
while  $x \bmod 2 \neq 0$  AND  $y \bmod 2 \neq 0$  do
  if  $(x + y) \bmod 2 \neq 0$  then
    if  $x \bmod 2 == 1$  then
      random walk N,S;
    end
  else
    random walk W,E;
  end
end
else
  random walk NW,NE,SW,SE;
end
end

```

Algorithm 2: Rules carried by type 2 sensors.

```

Data : Node type 2 (G), current position  $(x, y)$ 
while  $(x + y) \bmod 2 == 0$  do
  random walk N,S,W,E;
end

```

are occupied, the sensor randomly chooses one direction and moves out of the local eight-neighborhood. Through communication, the sensor movement can be reduced to a large extent and thereby reducing the energy consumption on mechanical movement. However, the energy consumption on communication is introduced.

IV. PERFORMANCE METRICS

In order to evaluate the performance of the placement strategy as well as different movement mechanisms, we design three metrics, including the *network coverage and redundancy*, the *convergence time*, and the *energy consumption*.

Network Coverage and Redundancy: Network coverage is a commonly referenced metric for evaluation of sensor deployment algorithms in a number of literature [4], [5], [6], [7], [8]. It is defined as the sensing capability of a sensor network, that is, how well the network can observe or monitor the given sensing field.

Algorithm 3: Rules carried by type 3 sensors.

```

Data : Node type 3 (B), current position  $(x, y)$ 
while  $x \bmod 2 \neq 1$  AND  $y \bmod 2 \neq 1$  do
  if  $(x + y) \bmod 2 \neq 0$  then
    if  $x \bmod 2 == 0$  then
      random walk N,S;
    end
  else
    random walk W,E;
  end
end
else
  random walk NW,NE,SW,SE;
end
end

```

The sensing range is modelled as a circle whose center denotes the sensor. In this paper, we consider a probabilistic model [4], where the sensing sensitivity depends on the geometrical distance from the sensor. Let ss_p denote the sensing sensitivity of a sensor, then $ss_p = e^{-\alpha d}u(c - d)$, where α represents the sensitivity coefficient, which determines how fast the sensitivity decreases as the target goes further away from the sensor. $u(\cdot)$ is the step function, and d denotes the Euclidean distance between the sensor and the point investigated. The constant c is the sensing range of a sensor, above which the sensing capability is zero.

Given a sensing field with Q grid points, suppose a group of sensors with T different types are deployed. We define a *probabilistic coverage* model as follows. For a certain grid point $g_n, n = 1, 2, \dots, Q$, we search for the nearest sensor $s_t, t = 1, 2, \dots, T$ of each type. If the Euclidean distance d_t between the sensor s_t and the grid point g_n is less than the corresponding sensing range c_t of sensor type t , we say this grid point is covered by this kind of sensor at the sensitivity of $e^{-\alpha_t d_t}$, where α_t is the sensitivity coefficient of sensor type t . The coverage of grid point n can then be formulated as

$$cover_n = \frac{1}{T} \sum_{t=1}^T e^{-\alpha_t d_t} u(c_t - d_t) \quad (1)$$

And the coverage of a sensor network is defined as the average coverage of all grid points.

The network coverage can be increased by deploying more sensor nodes especially in the sparsely deployed region. However, as more sensors are deployed in a given sensing field, it is very likely that more than one sensor of the same type would cover a single point. This can be measured by *network redundancy*. For the grid point g_n , the total number of sensors that cover this point is $N(N > 0)$, then the redundancy of this grid point is defined as

$$redundancy_n = \frac{\sum_{i=1}^N e^{-\alpha_i d_i} u(c_i - d_i) - T cover_n}{\sum_{i=1}^N e^{-\alpha_i d_i} u(c_i - d_i)} \quad (2)$$

The grid point that is not covered by any sensor is not considered when calculating the redundancy. The redundancy of a sensor network is the average redundancy of all grid points that are covered.

Convergence Time: Convergence time refers to the time needed for a sensor network to converge from the initial state to the final configuration. As mentioned before, we define the convergence time as the number of deployment epoch of the last positioned sensor. The deployment epoch refers to the time for an unpositioned sensor to move from one grid point to an adjacent grid point. To simplify the discussion without loss of generality, we assume the movement from one grid point to another takes the same amount of time irrespective of the actual distance. And hence, the larger the number of deployment epochs, the longer the convergence time, and the worse the system performance.

Energy Consumption: In sensor networks, the key challenge is to conserve system energy, thereby maximizing

the lifetime and improving the performance of the network. Energy consumption is the most relevant metric in a sensor network, since sensor nodes only carry limited power source, and in most applications, it is impossible to replace sensor battery. The two major energy consumption sources in mobile sensor networks are due to mechanical movement and communication. In the random and the SI-based method, only mechanical movement consumes energy. While for the SI-based movement with local communication, besides the mechanical movement, the energy consumption in transmission and receiving also needs to be considered.

We use a linear model to calculate the energy consumption in mechanical movement based on the assumption that the energy consumed in movement is proportional to the moving distance, that is $E_{cons} = d \times e_{move}$ where d is the moving distance, and e_{move} represents the unit distance energy consumption.

The energy model for electronic communication is based on the first order radio model [23]. Two parts of energy consumption in communication are considered: electric circuitry e_{elec} and transmitter amplifier e_{amp} , where we assume $e_{elec} = 500e_{amp}$. The energy consumed by a sensor when receiving a data packet is given by $E_{rcv} = e_{elec}B$, while the energy consumed in transmitting a data packet is given by $E_{tran} = e_{elec}B + e_{amp}d^2B$, where d is the maximum communication distance, and B is the number of bits in the data packet. To be able to compare energy consumption in different movement strategies, we need to investigate the relationship between energy consumption in movement and communication. We assume $e_{move} = \beta e_{elec}$ and by choosing different ratios β , we study the effect of energy consumption in movement and communication.

V. EXPERIMENTS

In all the simulations, we assume each sensor is equipped with a positioning system (such as GPS) that allows the sensor to find its location and orientation during the movement; and all the sensors can move freely within the sensing field. For example, we assume there is no collision between different sensors and between sensors and obstacles. Each sensor can detect the boundary of the network so that it will not move out of the sensing field.

The sensing field is chosen to be a 50m×50m 2-D region. There are five types of sensing platforms with the ratio equal to $r = \frac{1}{4} : \frac{1}{4} : \frac{1}{4} : \frac{1}{8} : \frac{1}{8}$ to be deployed. We assume the same sensing range of 10m and the same sensing sensitivity coefficient of $\alpha = 0.035$ for different modalities. The sensing sensitivity coefficient is selected such that at the distance of sensing range c , the sensor still has 70% sensing sensitivity. We first compare the proposed optimal mosaic placement design with the random placement in the sense of network coverage and redundancy, then we evaluate the performance of the SI-based movement strategy.

A. Random placement vs. Optimal mosaic placement

Figure 5 shows an initial and final network configurations. Different colors and marks are used to represent different types of sensor nodes. It can be seen that in the initial stage, the sensors clump in some regions, while in the final stage, all the sensors distribute uniformly across the whole sensing field. We expect the even spacing would improve the network coverage.

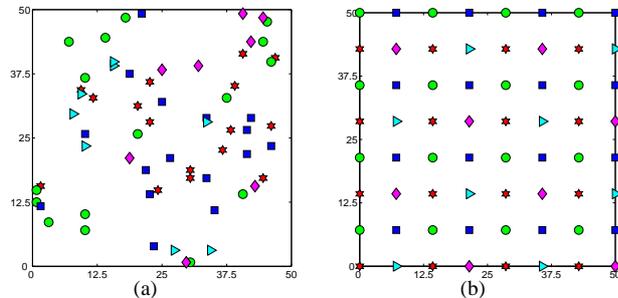


Fig. 5. Network configuration: (a) Initial state (b) Final state after 1965 epochs using the random movement.

For the given sensing field, we deploy different numbers of sensors using both the random placement and the proposed mosaic algorithm. To calculate the network coverage and redundancy, we assume the sensing field is resampled into a 100-by-100 rectangular grid. These grid points are not the real sensor sites. By doing so, we can study the coverage and redundancy of the sensing area at a higher resolution of 0.5m. Moreover, the uncertainties associated with the precise sensor locations are considered. Currently, after error corrections, some GPS systems can provide $< 1m$ position accuracy [24]. As a consequence, we add $0 \sim 1m$ random error to the final mosaic sensor locations to simulate the position error.

In Fig. 6(a), we evaluate the coverage metric versus sensor density with the number of sensors deployed varied among 25, 36, \dots , 256. We observe that to reach 80% coverage, about 160 sensors must be deployed using the random placement, while only 90 sensors are needed using the proposed placement algorithm. From the simulation results, it is quite obvious that the mosaic placement improves the sensing field coverage compared to the random sensor placement. Figure 6(b) illustrates network redundancy against the network coverage. We again observe that for the same coverage level, the network redundancy is much higher using the random placement. As the number of sensors increases, the redundancy increases and the difference between the random placement and the mosaic placement decreases.

B. Random Movement vs. SI-based Movement

In this simulation, we compare the random movement strategy with the SI-based movement with and without communication from the perspectives of convergence time, network coverage, and energy consumption. In the following experiments, we fix the number of sensors to be 64 except

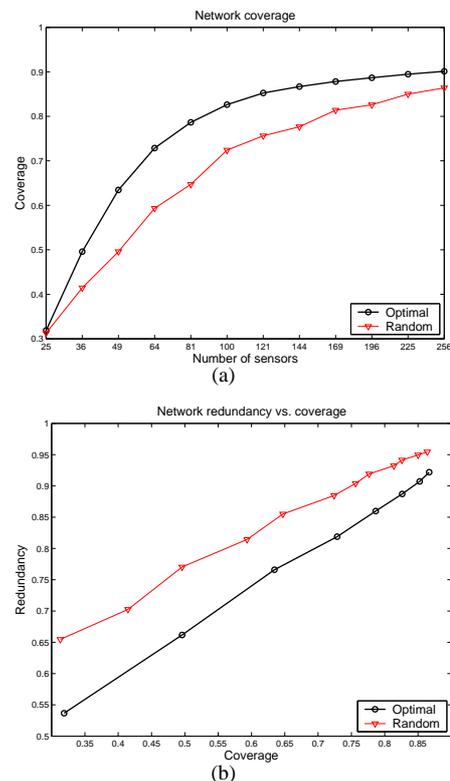


Fig. 6. The network coverage and redundancy using both random deployment and optimal placement. (a) Network coverage (b) Network redundancy

when we investigate the scalability of the SI-based methods with and without communication.

To investigate the convergence time, different movement methods are performed twenty times and the deployment epochs are recorded. After deleting the two maximum and two minimum values, the average deployment epochs are about 5000, 180, and 50 for the random movement, the SI-based method without and with local communication, respectively. It can be seen that the SI-based method with communication converges the fastest, and the random movement the lowest. In Fig. 7, we plot the percentage of positioned sensors as a function of deployment epochs. It can be seen that the SI-based method with communication needs shorter time to deploy all the 64 sensors. We also observe that around 90% sensors are positioned within the first 15 epochs using the SI-based method with communication. As the number of positioned sensors increases, it takes much longer time to converge since most of the sensor sites have been occupied.

Figure 8 shows the measured network coverage against the number of deployment epochs using the three movement methods. Due to the randomness, the plots present different degrees of fluctuation. Note that in general, the network coverage increases with the deployment epochs, independent of the movement methods used. Given a specified coverage level, the SI-based movement with communication converges faster than the other two methods. At a certain deployment time, the SI-based method with communication presents

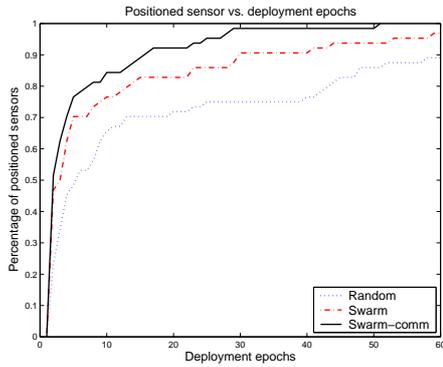


Fig. 7. The percentage of positioned sensors vs. deployment epochs.

higher network coverage level. Another observation is that the coverage increases very quickly at the early stage of the sensor movement. After a certain time, the profiles become stable. At this stage, even though the network has not yet reached its optimal placement state, the network coverage is already very high. This is because at the early stage of deployment, most sensor sites are vacant, and it is very easy for an unpositioned sensor to find its final sensor site. As the area covered by the positioned sensor nodes increases, the other nodes will have to travel further to reach their final locations.

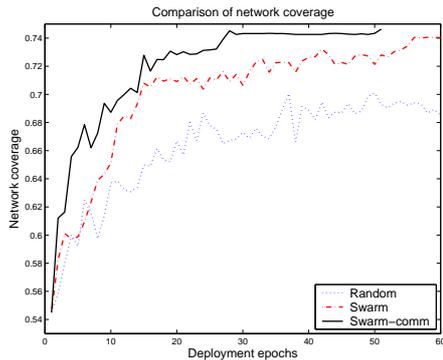


Fig. 8. Network coverage vs. deployment epochs.

Both the results in studying the network coverage and the convergence time reveal that there is a tradeoff in between, which is highly related to the energy consumption. In some applications, the coverage requirement might not be that critical, and it is not cost-effective to move sensors for a long distance to get a very small coverage improvement. In this case, we would like to terminate the deployment process before the maximum coverage is reached to save energy and reduce the deployment time. To terminate the deployment procedure earlier, we can set a threshold, defined as the minimum energy left in the sensor, below which the sensor will not move.

We next study the energy consumption during deployment. The communication range is assumed to be the same as the sensing range, which is $10m$. The coordinate information is

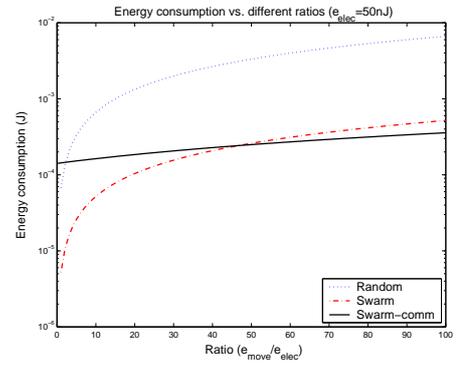


Fig. 9. Comparison of energy consumption using different movement methods.

coded using $32bit$ data. We study the energy consumption in communication and movement by choosing different ratios of $\beta = e_{move}/e_{elec}$ given fixed e_{elec} ($50nJ/bit$ [23]). The experimental results are illustrated in Fig. 9. It can be seen that if the ratio is less than 3, the SI-based method with communication performs the worst. When the ratio is larger than 3 and less than 47, the SI-based method with communication outperforms the random movement, but it is still worse than the SI-based method without communication. If the ratio is greater than 47, the SI-based method with communication presents the best performance. Note that the SI-based method without communication always outperforms the random movement.

Figure 10 illustrates the performance of the SI-based method with and without communication when scaled to different sizes of networks, where we deploy 25 and 144 sensors in the same sensing field as stated at the beginning of this section, respectively. The solid line denotes the result of the SI-based method with communication, and the dotted line the SI-based method without communication. The symbol ∇ represents the results of 144 sensors and the symbol \times corresponds to the results of 25 sensors. We observe that for the given sensing field, the change of energy consumption in sensor movement is smaller than that in communication. Since the total area of the sensing field is fixed, as the number of deployed sensors increases, the distance between two adjacent grid points would decrease such that the energy consumed when a sensor moves from one grid point to another is reduced. Even though the total number of sensors is large, the total energy consumption in sensor movement does not change too much.

From the simulation results, we see that the proposed sensor placement design can achieve reliable coverage level with lower redundancy, and the SI-based movement outperforms the random movement. The comparison between the SI-based movement with and without communication depends on the ratio β . The sensor network is able to converge to the optimal state with less convergence time and lower energy consumption using the SI-based methods.

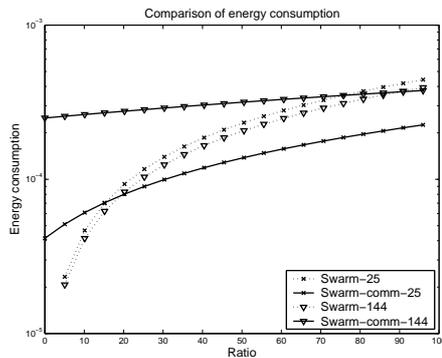


Fig. 10. Comparison of energy consumption with different network sizes.

VI. CONCLUSION

This paper addressed the problem of self-deployment of heterogeneous mobile sensor networks using biologically-inspired principles and methodologies for cost-effective and reliable-coverage purposes. A mosaic sensor placement is developed and the simulation results demonstrate the optimality of the proposed placement design. To converge to such an optimal state, the SI-based self-organization mechanism is investigated. Performance evaluation is conducted from three perspectives, including network coverage, redundancy, convergence time, and energy consumption. The experiments described in this paper confidently demonstrate that the SI-based sensor movement with communication can be used to deploy mobile sensor networks with rapid convergence and low energy consumption.

In this paper, several assumptions were made during the simulation in order to simplify the problem and highlight baseline algorithm evaluation. Interesting future directions would include sensor movement for network reconfigurations adapting to the event distribution, more realistic models of the environment such as the introduction of obstacles, other sampling grids like hexagonal or triangular to relax the POA restraint.

REFERENCES

- [1] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. Saluja, "Sensor deployment strategy for target detection," in *First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.
- [2] K. Dasgupta, M. Kukreja, and K. Kalpakis, "Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks," in *IEEE International Symposium on Computers and Communications*, 2003.
- [3] K. Kar and S. Banerjee, "Node placement for connected coverage in sensor networks," Extended Abstract. Proceedings of WiOpt 2003, Sophia-Antipolis, France, March 2003.
- [4] Y. Zou and K. Chakrabarty, "Uncertainty-aware sensor deployment algorithms for surveillance applications," in *IEEE GLOBECOM*, 2003, pp. 2972–2976.
- [5] A. Howard, M. J. Mataric, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots, Special Issue on Intelligent Embedded Systems*, vol. 13, no. 2, pp. 113–126, Sept. 2002.

- [6] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential field: A distributed, scalable solution to the area coverage problem," in *In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, 2002, pp. 299–308.
- [7] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 1, pp. 61–91, February 2004.
- [8] G. Wang, G. Cao, and T. Porta, "Movement-assisted sensor deployment," in *IEEE Infocom*, 2004.
- [9] O. Packer and D. R. Williams, *The Science of Color*, Optical Society of America, 2003.
- [10] A. Roorda, A. B. Metha, P. Lennie, and D. R. Williams, "Packing arrangement of the three cone classes in primate retina," *Vision Research*, vol. 41, no. 10–11, pp. 1291–1306, 2001.
- [11] E. Bonabeau and G. Theraulaz, "Swarm smarts," *Scientific American*, pp. 73–79, 2000.
- [12] M. Dorigo and T. Stutzle, *Ant colony optimization*, The MIT Press, Cambridge, Massachusetts, London, England, 2004.
- [13] P. A. Raymond, L. K. Barthel, and G. A. Curran, "Developmental patterning of rod and cone photoreceptors in embryonic zebrafish," *J Comp Neurol*, vol. 359, pp. 537–550, 1995.
- [14] S. Thoya, A. Mochizuki, and Y. Iwasa, "Formation of cone mosaic of zebrafish retina," *J. theor. Biol.*, vol. 200, pp. 231–244, 1999.
- [15] Y. Fei, "Development of the cone photoreceptor mosaic in the mouse retina revealed by fluorescent cones in transgenic mice," *Molecular Vision*, vol. 9, pp. 31–42, February 2003.
- [16] M. A. Raven and B. E. Reese, "Mosaic regularity of horizontal cells in the mouse retina is independent of cone photoreceptor innervation," *Investigative Ophthalmology and Visual Science*, vol. 44, pp. 965–973, March 2003.
- [17] M. Dorigo and L.M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [18] T. Stutzle and M. Dorigo, "Aco algorithms for the traveling salesman problem," in *Evolutionary Algorithms in Engineering and Computer Science*, K. Miettinen, M. M. Makela, P. Neittaanmaki, and J. Periaux, Eds., Chichester, UK: John Wiley & Sons, 1999, pp. 163–183.
- [19] L. M. Gambardella and M. Dorigo, "Ant colony system hybridized with a new local search of the sequential ordering problem," *INFORMS Journal on Computing*, vol. 12, no. 3, pp. 237–255, 2000.
- [20] L. M. Gambardella, E. D. Taillard, and M. Dorigo, "Ant colonies for the quadratic assignment problem," *Journal of the Operational Research Society*, vol. 50, no. 2, pp. 167–176, 1999.
- [21] G. D. Caro and M. Dorigo, "Antnet: Distributed stigmergic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.
- [22] M. Heissenbuttel and T. Braun, "Ants-based routing in large-scale mobile ad-hoc networks," in *Proceedings of Kommunikation in verteilten Systemen*, K. Irmscher and R. P. Fahrnich, Eds., Berlin, VDE Verlag GmbH, 2003, pp. 91–99.
- [23] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International conference on System Sciences*, 2000, vol. 8, pp. 3005–3014.
- [24] "Introduction to the global positioning system for gis and traverse," <http://www.cmtinc.com/gpsbook/>.