Localization Using Boundary Sensors: An Analysis Based on Graph Theory

YUNHUI ZHENG, DAVID J. BRADY, and PANKAJ K. AGARWAL
Duke University

We consider sensors, such as fibers, lasers, and pyroelectric motion detectors, that fire when objects cross a boundary. A moving object can be localized by analyzing sequences of boundary crossings. We consider the number of distinct sequences and object positions that can be achieved using boundary sensors in one- and two-dimensional spaces. For 1D systems we use representations of sensor sequences on graphs to derive limits on the number of object locations that can be monitored by a given sensor population and sequence length. For 2D systems we show that in certain circumstances the ratio of the number of unique sensor sequences to the number of unique object paths is exponential in the sequence length and we argue that the probability of unique identification is high for sufficiently large sequences. We also prove the triangle grid can track an object with error limited to a small neighborhood.

Categories and Subject Descriptors: I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis

General Terms: Algorithm, Design, Theory

Additional Key Words and Phrases: Boundary sensor, deployment sequence, deployment graph, sensor sequence, sequence graph

ACM Reference Format:

Zheng, Y., Brady, D. J., and Agarwal, P. K. 2007. Localization using boundary sensors: an analysis based on graph theory. ACM Trans. Sens. Netw. 4, 3, Article 21 (October 2007), 19 pages. DOI = 10.1145/1281492.1281496 http://doi.acm.org/10.1145/1281492.1281496

1. INTRODUCTION

The concept of distributed sensors that fire when a person or object crosses a geometric structure, as defined for example by laser beam, is familiar to fans of spy and crime cinema. Surprisingly little academic analysis has appeared

This work is supported by the Army Research Office through grant 313-2082.
Authors' addresses: Y. Zheng and D. Brady, Fitzpatrick Center for Photonics and Communication Systems, Department of Electrical and Computer Engineering, Duke University, Durham, NC 27707; email: yzheng@ee.duke.edu, dbrady@duke.edu; P. Agarwal, Department of Computer Science, Levine Science Research Center D214A, Box 90128, Duke University, Durham, NC 27708-0129, email: pankaj@cs.duke.edu.

as to the optimal deployment and capacity of this class of sensor networks, perhaps accounting for the universal ability of Hollywood spies to penetrate these systems. While we do not resolve this problem here, we do describe the object localization capacity of a particular class of geometric sensors. We hope by this incremental analysis to seed continuing scholarly interest in the design and analysis of such sensors.

In general, design of a sensor network for localization and tracking involves both spatial sensor deployment and receiver pattern modulation. Deployment strategies were previously considered [Tian and Georganas 2002; Ye et al. 2003; Huang and Tseng 2003] assuming circular receiver pattern. Related work on sensor deployment and path identification used path exposure as a merit function to determine the quality of sensor deployment for target detection [Clouqueur et al. 2003; Meguerdichian et al. 2001a; Meguerdichian et al. 2001b; Meguerdichian et al. 2001c]. Our group previously looked at receiver pattern design in Potuluri et al. [2003], Gopinathan et al. [2003], and Zheng et al. [2005, 2006]. These works employ a static sensing mode that tries to localize the object based on an instantaneous detected signal vector/codeword, not a temporal sequence. Every point in the object space is thus encoded with a binary codeword by proper sensor deployment.

The physical interface of a sensor system is determined mappings between a target configuration and a physical distribution in the target embedding space and between the target distribution and the sensor state. In the case of photography, for example, the target is an object in a 3D embedding space and the mapping from the target to a physical distribution occurs when the target radiates or scatters light. The mapping from the light to the sensor state is mediated by lenses such that the sensor state is proportional to the light density on the target.

With interest in computational distributed sensor networks, novel conceptions of the target-field and field-sensor mappings have arisen. Using various approaches to sensor deployment, optical and electronic interfaces, and object radiation, it is possible to design sensors that are sensitive to objects occupying specific points, average values of objects covering a volume, or objects crossing a curve or a surface. Expanding the concept of the object embedding space to include hypercubes of spatial, spectral, coherence, polarization, and even chemical information, it is possible to imagine many interesting relationships between the sensor state and the target state.

The sensor receiver pattern is a general approach to describe the object field-sensor data mapping. The receiver pattern describes the points in the object space to which a given sensor is sensitive. For example, a pixel on a camera may be sensitive to all points along a specific ray extending from the camera or a motion sensor may be sensitive to all objects moving in a specific volume.

The goals of sensor system design are to maximize the capacity of the system relative to operational metrics (such as probability of detection, localization, and classification) while minimizing system complexity, computational complexity, and latency, deployment and operations costs and power consumption. Many different classes of sensors and algorithms may be applied to a sensing problem; ultimately sensor design must decide among these approaches. The sensor

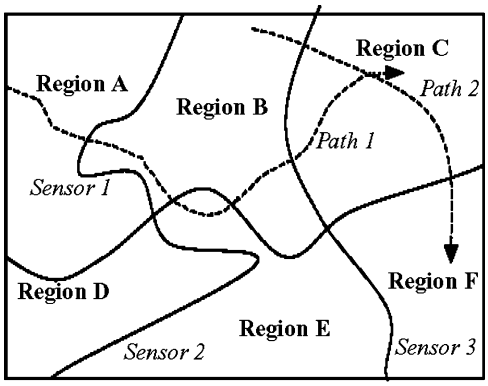

Fig. 1. An object space monitored by the boundary sensor network. An object entering the object space along path 1 induces the sensor sequence (2113), while path 2 induces the sensor sequence (31).

receiver pattern is the most fundamental determinant of the capabilities of a particular sensing approach.

As an example of a novel approach to sensor system design and analysis, this article considers boundary sensors. While the concept of a boundary sensor is by no means novel, detailed analysis of the tracking and localization capacity of networks of boundary sensors has not, to our knowledge, been previously presented. A boundary sensor in $\Re^N$ is triggered when an object crosses an $N-1$-dimensional structure. For example, a boundary is determined by a discrete set of points in $\Re^1$, by a set of curves in $\Re^2$, and by a set of surfaces in $\Re^3$. The detection of boundary crossing can be realized by video cameras [Zahnd et al. 2003; Bellutta et al. 1991], pyroelectric sensors [Dougherty et al. 2004; Gopinathan et al. 2003], laser beams, and optical fibers. When the object moves in the object space continuously, it triggers a series of boundary sensors. The object can then be localized based on detected sequence of activated boundary sensors, which will be called a sensor sequence. An example is shown in Figure 1

In contrast with staring sensor networks, the bounary sensor network uses temporal sequences of sensor events to encode spatial information. A spatial location can be reached by multiple paths generating distinct sequences, but ideally each sequence corresponds uniquely to the current object location. In order to localize a moving object based on the detected sequence, each distinct path must be associated with a unique sensor sequence. (We consider all paths through the same sequence of boundary defined regions to be identical.) A one-to-one mapping from sequences to paths requires proper deployment of the boundary sensors. In general, it is also desired in that deployment scheme to have multiplex sensors so that we can track the object in as many regions as possible with as few sensors as possible. At the same time, certain system robustness should be introduced to combat noise and system error.

The primary results presented in this paper are limits on the localization and sensor sequence generation capacity of boundary sensor networks. We define m to be the number of boundary sensors in the network and n to be the length

of the sequence used to track the object state. We assume that

(a) There is at most one object moving in the object space.
(b) At any instant the object can only trigger up to one boundary sensor.
(c) The boundary sensor will not miss any boundary crossing event.
(d) A given boundary sensor defines the boundary between a given object location and at most one other object location. (The object cannot change location by triggering the same sensor twice in immediate succesion. Events in which the same sensor is triggered twice in a row drops that sensor from the current sequence.)

We consider one dimensional boundary sensor systems in Section 2 under these assumptions. In 1D we obtain, in Theorem 2.7, expressions for the number of distinct target locations that are determined by m sensors from sequences of length n. We do not obtain similar definitive results for 2D boundary sensor networks, but we are able in Section 3 to bound the ratio of the number of distinct target paths to the number of sensor sequences for two dimensional systems.

2. ONE-DIMENSIONAL BOUNDARY SENSOR SYSTEMS

In one dimension, a boundary sensor triggers when a target passes a point. Our model allows a single sensor point to be assigned detection points along $\Re^1$. The model is motivated in part by an understanding of the 1D boundary segmentation as a projection of a 2D map. In $\Re^2$ we will allow the boundary to curve arbitrarily. In crossing the projections of a 2D segmentation into $\Re^1$, we could cross the same curve multiple times, thus creating the possibility of firing the same sensor at multiple points. In a particularly simple case, one can imagine the use of crossed 1D boundary structures for localization on a 2D space. This approach is illustrated in Figure 2. The x and y axes are each segmented by arrays of line sensors. By connecting these line sensors using curves outside the object space, we create 1D boundary sensors with repetitive coding for each sensor. Allowing a single sensor to curve through the space multiple times allows us to minimize the number of sensors used. The discussion of the 1D case thus can be generalized to an important class of 2D solutions, which will be more useful and applicable in real world.

Whether or not it is actually deployed in 2D as illustrated in Figure 2, a 1D boundary sensor may be represented graphically by a segmentation of the number line. A line and a number of sensors denoted as points spread over the line are shown in Figure 3. Each point stands for a "probe" from a particular sensor, and a sensor can have multiple probes. The key requirement is to arrange these points so that given m sensors and a desired sensor sequence of length n, every consecutive point sequence (number sequence) of length n is unique among all such sequences deployed along this line irrespective of location. For example, in Figure 3, given $m = 5$, $n = 3$, one can list all the consecutive point sequences of length 3: 123, 231, 314, 145, 454, 542. We will discuss bidirectional sequence (bidirectional sequence means, for example, 123 can be read as either 123 or 321) later. We find all such point sequences are unique. Thus the deployment

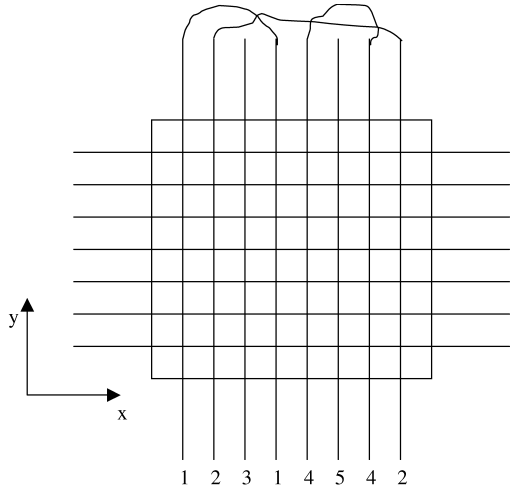

Fig. 2. An example of the separable Cartesian gird design. The parallel lines are the boundary sensors. Lines are connected at one end if they are the same sensor. The index of the sensors are marked at the other end. The sensors parallel to x axis have the identical numbering but are not marked in this figure. During operation, the x, y coordinates of the object are identified separately.

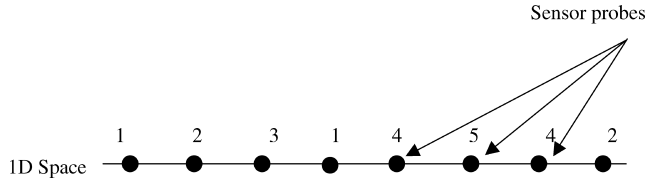

Fig. 3. The line model of 1D boundary sensor deployment. The line indicates the 1D space. The points stand for the probe of a particular sensor. Each sensor can have multiple probes.

sequence $D = (12314542)$ satisfies our requirement, where D consists of all the sensor probes along the line from left to right and $D \in [1, 2 \ldots, m]^l$, l is the number of sensor probes. However, this deployment is clearly not the longest one. We can add another 5 to the right end deployment sequence to form a longer one that satisfies our requirement. A longer deployment sequence means one can use the same amount of sensor resources to monitor a larger space. Our design goal is to find the longest possible sequence for a given length n. We will start from the definition of several useful concepts.

Definition 2.1. A *deployment sequence* is an arrangement of sensor points in 1D space described by a number sequence wherein each number stands for a sensor.

Definition 2.2. A *sensor sequence* of length n is an n-symbol subsequence of a deployment sequence, corresponding to the sensor firings that would be detected as an object moved between corresponding locations on $\Re^1$.

As an object moves along $\Re^1$, sensors in the deployment sequence fire and segments of the deployment sequence are recorded by the sensor system. If the system must determine the location of the object based on a single firing,

each sensor point must be assigned to an independent sensor. As we consider longer sequences of sensor firings, however, it becomes possible to localize the object even when single sensors correspond to multiple edges between object locations. The localization capacity of the 1D sensor system (i.e., the maximum number of unique object locations that can be achieve along $\Re^1$) is characterized by the maximum length of a deployment sequence under the constraint that every subsequence of length n is unique.

The following discussion will involve a number of concepts in graph theory. We will start by reviewing their definition here as a handy reference.

Definition 2.3. A *graph* is a collection of points and lines connecting some subsets of the points.

Definition 2.4. A *walk* is an alternating sequence of vertices and edges, with each edge being incident to the vertices immediately preceeding and succeeding it in the sequence.

Definition 2.5. A *trail* is a walk with no repeated edges.

Definition 2.6. An *Euler trail* is a walk on the graph edges which uses each graph edge in the original graph exactly once.

The result of the 1D case is now summarized by Theorem 2.7.

THEOREM 2.7.

(1) *The maximum achievable number of locations identified by a boundary sensor system with m sensors using sensor sequences of length n such that each sensor sequence is unique in the deployment sequence along either positive or negative direction is*

$$P(m,n) = \begin{cases} m, n = 1 \\ m(m-1)^{n-1}, n \geq 2. \end{cases} \tag{1}$$

The number of distinct deployment sequences derived from the sequence graph for m sensors and sensor sequence of length n is

$$D = (m-1)!^{m(m-1)^{n-2}} det(K), \tag{2}$$

where K is the principal submatrix obtained from the Laplacian [Chung 1997] of the sequence graph by deleting any row and the corresponding column.

(2) *From* (1)*, if there is an additional requirement that no sensor appears with less than two intervening distinct sensors, the maximum number of location becomes*

$$P(m,n) = \begin{cases} m, n = 1 \\ m(m-1)(m-2)^{n-2}, n \geq 2, m \geq 4. \end{cases} \tag{3}$$

The number of distinct deployment sequences derived from the enhanced sequence graph for m sensors and sensor sequence of length n is

$$D = (m-2)!^{m(m-1)(m-2)^{n-3}} det(K). \tag{4}$$

(3) *From* (2)*, if the sensor sequence is globally unique (unique along both positive and negative directions) in the deployment sequence, this maximum number of location becomes*

$$P_u(m,n) = \begin{cases} m, n = 1 \\ \frac{m^2-m}{2}, modd, n = 2 \\ \frac{m^2-2m+2}{2}, meven, n = 2 \\ \frac{m(m-1)(m-2)^{n-2}}{2}, n \geq 3, m \geq 4. \end{cases} \quad (5)$$

Theorem 2.7 has 3 parts. Each part corresponds to a specific scenario/ requirement. Since a number of lemmas are required to prove each part, we will prove them one by one.

Definition 2.8. A *sequence graph* for a system with m sensors based on sensor sequences of length n is a directed graph consisting of

—A vertex corresponding to every possible sensor sequence of length $n-1$.
—Directed edges between all vertices such that the last $n-2$ numbers of the sensor sequence of the originating vertex are the same as the first $n-2$ numbers of the sensor sequence of the terminating vertex.

Under these conditions each edge corresponds to a sensor sequence with length n and a trail in the sequence graph corresponds to a deployment sequence.

LEMMA 2.9. *A sequence graph has $m(m-1)^{n-2}$ vertices. The in-degree and out-degree for each vertex are both $m-1$. The total number of edges is $m(m-1)^{n-1}$.*

PROOF. According to assumption (d), no consecutive numbers within a sensor sequence should be the same; thus, the number of vertices equals the number of sensor sequences with length $n-1$. After the first sensor in a sensor sequence is determined from the m candidates, there are $m-1$ candidates for each successive sensor in the sequence. The total number of allowable sequences of length n is calculated as $m \cdot \underbrace{(m-1)(m-1)\cdots(m-1)}_{n-2} = m(m-1)^{n-2}$.

Since outgoing and incoming edges match subsequences of length $n-2$, the number of such matches is determined by the number of different first or last number in the length $n-1$ vertex sequences. Since we have assumption (d), only $(m-1)$ choices are available. The total number of edges is calculated as $\frac{2m(m-1)^{n-2}\cdot(m-1)}{2} = m(m-1)^{n-1}$ □

LEMMA 2.10. *A sequence graph is strongly connected.*

PROOF. We need to prove there is a directed path from any vertex to another vertex.

Suppose the last sensor of the starting vertex is A and the first sensor of the ending vertex is B. We will find a directed path from the starting vertex to the ending vertex based on the rules mentioned before for two different situations. Note that once we reach a vertex whose sensor sequence ends at B, it is trivial to find the rest of path to reach the ending vertex whose sensor sequence begins

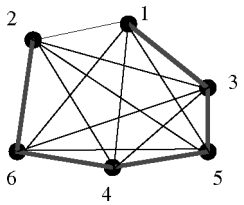

Fig. 4. A graph with vertices for 6 sensors and edges for potential sensor boundaries for $n = 1$. The darkened path is one of the Hamiltonian paths of this graph.

with B by simply shifting B to the left and adding a vertex to its right by sticking to the definition of sensor sequence.

(1) when $A = B$, the corresponding path is $(\ldots A) \Rightarrow (\ldots B) \Rightarrow \cdots (B \ldots)$
(2) when $A \neq B$, the corresponding path is $(\ldots A) \Rightarrow (\ldots AB) \Rightarrow (\ldots B) \cdots (B \ldots)$ □

LEMMA 2.11. *A directed Euler circuit exists for a directed graph iff the graph is strongly connected and each vertex has equal number of in-degree and out-degree [Bollobas* 1979*]*

Now we are ready to prove part 1 of Theorem 2.7

PROOF. When $n = 1$, we will construct a graph with the vertices corresponding to the m sensors and edges connecting the sensors. All the vertices are connected by undirected edges since any two different sensors can be adjacent to each other. A deployment sequence that maximizes sensing efficiency corresponds to a path that transverses all the vertices. Such path is a Hamiltonian path [Bollobas 1979]. It is trivial to find a Hamiltonian path in a complete graph. For example, the darkened edges in Figure 4 are a Hamiltonian path corresponding to the deployment sequence (135462).

When $n > 1$, from Lemmas 2.9, 2.10, and 2.11, it is found that a directed Euler circuit exists for the sequence graph. The Euler circuit is a special kind of Euler trail with the same starting and ending vertex. Thus we can generate an optimal deployment sequence utilizing all the $m(m-1)^{n-1}$ sensor sequences based on the Euler circuit.

To determine the number of distinct deployment sequence, equivalently, we need to calculate the number of distinct Euler circuits. Note that in the deployment graph an Euler trail is also an Euler circuit. From the BEST formula (named after de Bruijn, van Aardenne-Ehrenfes, Smith, and Tutte) [de Bruijn and van Aardenne-Ehrenfest 1951; Smith and Tutte. 1941], the number of Euler trails in a digraph is given by $\underbrace{(d_1-1)(d_2-1)\ldots(d_i-1)}_{number of vertices} det(K)$, where d_i is the number of outdegree for ith vertex. Inserting the values presented in lemma 2.9 to the BEST formula, we obtain Equation(2). □

Figure 5 shows an example when $n = 4, m = 3$. Fleury's algorithm is employed to find an Euler circuit in the graph. [Lucas 1891] Note that the Euler circuit derived is not unique. One can start from another vertex or take another edge during the trail to form another deployment sequence.

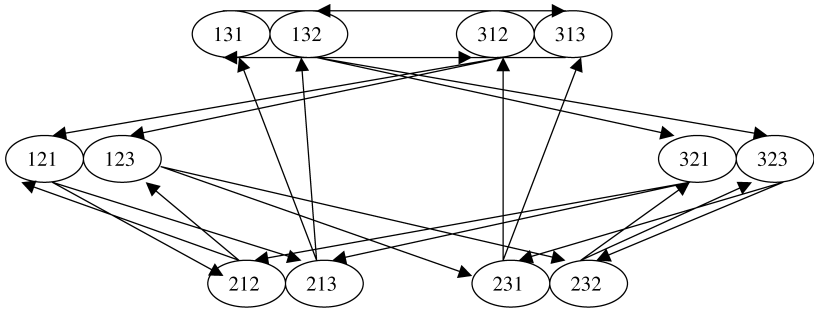

Fig. 5. The sequence graph for $m = 3, n = 4$. The total number of edges is $3 * (3-1)^{4-1} = 24$. An Euler trail can be found within this graph that corresponds to a deployment sequence to track an object in 24 positions with 3 sensors and at least 4 consecutive measurements. One of the deployment sequence we get from this Euler circuit is (121232312313121321313232121).

In part (1) of Theorem 2.7 we have derived the number of locations and deployment sequences we can achieve in 1D with m sensors and n detection events. It is useful to extend these results to more constrained deployment sequences that are robust against *false crossings*. A false crossing occurs when an object triggers a boundary sensor but does not finish the crossing. The object could touch the boundary and trigger the sensor but remain in the same region as it was before triggering the sensor, or it could cross in a complicated multiple trigger manner as induced by the walking pattern of legged objects. In a robust system we want to detect false crossings by sacrificing a little sensing efficiency. We will need to apply another constraint on the sensor sequence arrangement, which is stated as assumption (*e*)

(e) No alternating sensor in a sensor sequence can be the same.

This scenario correspond to part (2) of Theorem 2.7. To prove it, We first start with a new definition: *enhanced sequence graph*. Then we will prove several supporting lemmas.

Definition 2.12. An *enhanced sequence graph* is a sequence graph excluding deployment sequences in which sensors repeat in less than two steps.

LEMMA 2.13. *A enhanced sequence graph has* $m(m-1)(m-2)^{n-3}$ *vertices. The in-degree and out-degree for each vertex are both* $m-2$*. The total number of edges is* $m(m-1)(m-2)^{n-2}$.

PROOF. For the first sensor in the sensor sequence, we still have m candidates. After the first sensor is chosen, the second should not be the same with first one according to assumption (f), so we have $(m-1)$ choices. For the ith sensors in the sensor sequence, it can neither be the same as the $(i-1)$th sensor nor the $(i-2)$th sensor according to assumption (g), so it only has $m-2$ choices. That is the same as the in-degree or out-degree for any vertex in the enhanced sequence graph. Since each vertex corresponds to a sensor sequence of length $n-1$, the number of such sensor sequence (same as the number of vertices) now is $m \cdot (m-1) \cdot \underbrace{(m-2)(m-2)\cdots(m-2)}_{n-3} = m(m-1)(m-2)^{n-3}$. □

LEMMA 2.14. *An enhanced sequence graph is strongly connected when* $n = 2$ *or* $n \geq 3$ & $m \geq 4$

PROOF. The proof is trivial when $n = 2$.

When $n \geq 3$ & $m \geq 4$, suppose the last two sensors in the starting vertex are A_1A_2 and the first two sensors of the ending vertex are B_1B_2. We will find a directed path from the starting vertex to the ending vertex based on the rules mentioned before for six different situations. Note that once we reach a vertex whose sensor sequence ends at B_1B_2, it is trivial to find the rest of path to reach the ending vertex whose sensor sequence begins with B_1B_2.

(1) When $A_1 \neq B_1, B_2, A_2 \neq B_1, B_2$, the corresponding path is $(\ldots A_1A_2) \Rightarrow (\ldots A_1A_2B_1) \Rightarrow (\ldots A_2B_1B_2) \Rightarrow \cdots \Rightarrow (B_1B_2\ldots)$.

(2) When $A_1 = B_1, A_2 \neq B_1, B_2$, suppose there is A_3 where $A_3 \neq B_1, B_2, A_2$. Then the corresponding path is $(\ldots B_1A_2) \Rightarrow (\ldots B_1A_2A_3) \Rightarrow (\ldots A_2A_3B_1) \Rightarrow (\ldots A_3B_1B_2) \cdots \Rightarrow (B_1B_2\ldots)$.

(3) When $A_1 = B_2, A_2 \neq B_1, B_2$, suppose there is A_3 where $A_3 \neq B_1, B_2, A_2$. Then the corresponding path is

(4) $(\ldots B_2A_2) \Rightarrow (\ldots B_2A_2A_3) \Rightarrow (\ldots A_2A_3B_1) \Rightarrow (\ldots A_3B_1B_2) \cdots \Rightarrow (B_1B_2\ldots)$.

(5) When $A_2 = B_1, A_1 \neq B_1, B_2$, then the corresponding path is $(\ldots A_1B_1) \Rightarrow (\ldots A_1B_1B_2) \Rightarrow \cdots \Rightarrow (B_1B_2\ldots)$.

(6) When $A_2 = B_2, A_1 \neq B_1, B_2$, suppose there is A_3 where $A_3 \neq B_1, B_2, A_1$.

Then the corresponding path is $(\ldots A_1B_2) \Rightarrow (\ldots A_1B_2A_3) \Rightarrow (\ldots B_2A_3B_1) \Rightarrow (\ldots A_3B_1B_2) \Rightarrow \cdots \Rightarrow (B_1B_2\ldots)$.

Based on the previous discussion, we find if $m \geq 4$, there is always a directed path from any vertex to another vertex. □

Now we are ready to prove part (2) of Theorem 2.7

PROOF. When $n = 1$ the proof is same as in Theorem 2.7. When $n \geq 2, m \geq 4$, we will construct an enhanced sequence graph. Then from Lemma 2.13 and 2.14, an Euler circuit exists for an enhanced sequence graph when $m \geq 4$, which corresponds to a deployment sequence monitoring $m(m-1)(m-2)^{n-2}$ regions.

The proof the number of distinct deployment sequences in the enhance sequence graph is similar to that in part (1) by substituting the values in Lemma 2.13. □

We note in the first two parts of Theorem 2.7, the sensor sequence is unique in the deployment sequence along either positive or negative direction, but not globally unique (unique along both directions). However, we find that the algorithm to find an Euler circuit in a graph can be modified to find a deployment sequence with globally unique sensor sequence in an enhanced sequence graph. We will travel in the enhanced sequence graph in such a way that, once a directed edge is traversed, the edge with mirror sensor sequence of the traversed one will be removed to prevent it from being included in the trail in the future. In such way, we can guarantee each sensor sequence is globally unique. The result is summarized in part (3) of Theorem 2.7. Again, we start from supporting lemmas.

LEMMA 2.15. *By developing a trail with the traveling and discarding method, exactly half of the edges in an enhanced sequence graph will be traversed when* $n \geq 3$ & $m \geq 4$.

PROOF. Here we present a constructive proof showing an algorithm how to find a trail by the *traveling and discarding* method.

Select a vertex v and begin to build a directed trail starting at v. For every edge added to this trail, remove the edge whose corresponding sensor sequence is the reversal of the one added to the trail to form a "mirror" trail. Proceed until you can go no further. Now one of the following must hold:

—We returned to v, forming a circuit C^*.
—We ended at a vertex w other than v.

If the latter condition holds, then w has different in- and out- degree, and one of the following conditions must hold:

(1) None or equal number of inward and outward edges of w has been removed. The algorithm halts with "proof" that G has no directed Euler circuit, which contradicts with our statement before.
(2) One more out edge of w was removed than the in edges. We assume the vertex right before w is u and that before u is t. In this case the sensor sequence of either u or t must be the reversal of the sensor sequence of w. Supposed the sensor sequence of u is $(A_1A_2 \dots A_{n-2})$ and that of t and w are $(B_1B_2 \dots B_{n-2})$ and $(C_1C_2 \dots C_{n-2})$. For the first case we have $(A_2 \dots A_{n-2}) = (C_1 \dots C_{n-3})$ and $(A_1A_2 \dots A_{n-2}) = (C_{n-2} \dots C_2C_1)$, which will lead to $A_i = A_{i+2}$ or $A_i = A_{i+1}, (3 \leq i \leq n-2)$, contradicting the rule that no consecutive or alternating sensor should be the same. For the latter case, we have $(B_1B_2 \dots B_{n-2}) = (C_{n-2} \dots C_2C_1)$, $(A_1 \dots A_{n-3}) = (B_2 \dots B_{n-2})$ and $(A_2 \dots A_{n-2}) = (C_1 \dots C_{n-3})$, which again leads to $A_i = A_{i+2}$ or $A_i = A_{i+1}, (3 \leq i \leq n-2)$, contradicting the rule that no consecutive or alternating sensor should be the same.

Thus the latter condition does not hold in any case, we will "grow" C by adding C^* to it.

Now, starting at v, proceed along C. One of the following must hold:

—We find a vertex u on C which still has edges leaving it.
—We find no such vertex, and C contains half of the edges of G; the other half was removed when we grow C.
—We find no such vertex, and C does not contain half of the edges of G.

In the third case, we have proved that G is disconnected: There is no directed path from any vertex of C to any vertex not on C, and so we can halt with a proof that G has no Euler circuit, contradicting Lemma 2.14 before. In the second case, C is the trail we want to find, so the algorithm halts in the happy state. In the first case, we let $v = u$ and return to Step 1, to continue to grow our circuit.

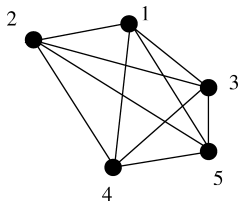

Fig. 6. The complete graph when $m = 5, n = 2$. An Euler trail can be found in such a graph to construct a deployment sequence in which the sensor sequence is globally unique along both directions. One example is (1234524135).

The final trail C will traverse exact half of the edges of the graph, corresponding to a deployment sequence containing $\frac{m(m-1)(m-2)^{n-2}}{2}$ sensor sequences. This is the maximum since adding any sensor sequence will make it not globally unique. □

LEMMA 2.16. *An Euler trail exists for a undirected graph iff there are exactly* 0 *or* 2 *vertices of odd degree. [Bollobas* 1979*]*

Now we are going to prove part (3) of Theorem 2.7

PROOF. When $n = 1$, the proof is trivial.

When $n = 2$, we can generate a sequence graph but with undirected edges. Now an edge corresponds to two sensor sequences that are mirrors of each other. In this graph an Euler trail corresponds to a deployment sequence among which every sensor sequence is globally unique because each undirected edge can be traversed only once. When m is odd, a complete graph with m vertices satisfying Lemma 2.16 has the maximum number of edges $\frac{m(m-1)}{2}$. When m is even, we need to remove $\frac{m-2}{2}$ edges from a complete graph to satisfy Lemma 2.16. Thus the total number of edges is $\frac{m^2-2m+2}{2}$. An example with $m = 5$ is shown in Figure 6.

When $n \geq 3$ & $m \geq 4$, we will travel in an enhanced sequence graph by traveling and discarding method to get the deployment sequence with length $\frac{m(m-1)(m-2)^{n-2}}{2}$. □

In Figure 7 we use the algorithm discussed before to travel in an enhanced sequence graph and find the deployment sequence containing both globally unique sensor sequence and one whose sensor sequence is unique only along positive or negative direction.

3. 2D DEPLOYMENT STRATEGY

In two dimensions, a boundary sensor is associated with a arbitrary 2D curve in the plane. As in the 1D case, an object moving continuously through the object space to form a path passing $n+1$ regions triggers n boundary sensors to form a sensor sequence of length n. If there is one-to-one mapping between the object path and the sensor sequence, we are able to localize the final location of an object without ambiguity.

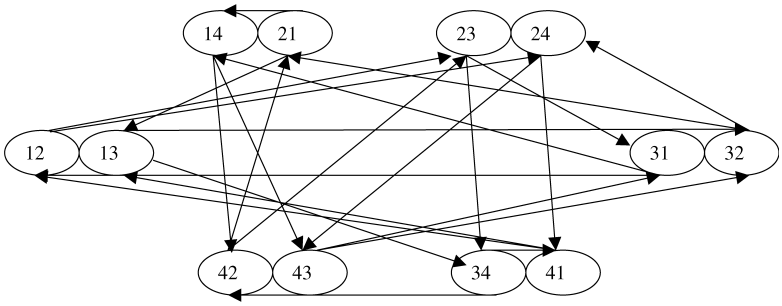

Fig. 7. The enhanced sequence graph for $m = 4, n = 3$. The total number of edge is $4*(4-1)*(4-2)^{4-1} = 24$. Again, an Euler trail can be found within this graph that corresponds to a deployment sequence to track an object in 24 positions with 4 sensors and at least 3 consecutive measurements. One of deployment sequence is (142314324124312341342132 14), whose length (number of sequences of length 3) is 24. The deployment sequence whose sensor sequences are globally unique based on the traveling and discarding algorithm is (14312342314214), whose length is $24/2 = 12$.

Design of a general efficient 2D deployment strategy for a boundary sensor array is a daunting problem. We begin discussion here by considering the performance of quasi-random 2D deployments. The unique mapping between a object path and a sensor sequence in a random deployment is not guaranteed. However, if the ratio of the total number of paths to the total number of available sensor sequences is much smaller than 1, we expect with high probability there is such a unique mapping. Moreover, we also hope this ratio will further decrease when we increase m and n so that less ambiguity can be achieved by a little more cost. That statement is actually true according to Theorems 3.11 and 3.13, which will be proved later. We start with the definition of deployment graph—a 2D counter part of deployment sequence which describes how the boundary sensor are deployed in $\Re^2$.

Definition 3.1. A deployment graph is a graph that describes the original layout of a boundary sensor network in 2D space. In the deployment graph, the object space boundary is converted to a special vertex. The intersections between boundary sensors are also also converted vertices. Two vertices are connected with an edge if there is a common sensor segment between them.

LEMMA 3.2. *A deployment graph is a finite, connected, planar graph.*

PROOF. This can be derived from the construction of deployment graph. □

LEMMA 3.3. *The dual graph [Harary* 1994*] of a deployment graph is a simple, finite, connected, and planar graph.*

PROOF. It is obvious that the dual graph is a finite, connected, and planar graph. However, in order to show it is a simple graph, we need to assume that the boundaries associated with 2D sensors do not end within the object space. By this assumption, two adjacent regions can share only one common boundary. The regions in the deployment graph correspond to vertices in the dual graph, and the boundaries correspond to edges in dual graph. Thus at most one edge

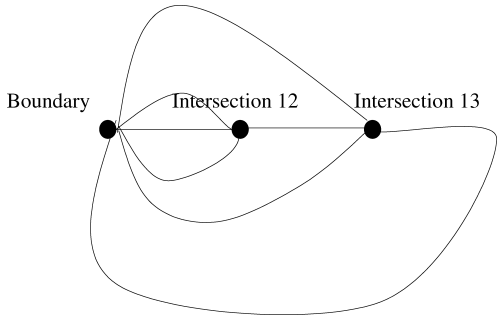

Fig. 8. The deployment graph of Figure 1.

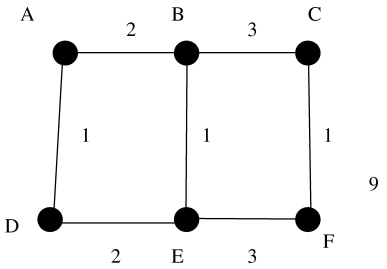

Fig. 9. The dual graph for the deployment graph shown in Figure 8. The numbers here stand for the sensors and the characters stand for the regions in Figure 1.

connects any two vertices in the dual graph, which means the dual graph is simple. □

For example , the dual graph of Figure 8 is shown in Figure 9.

LEMMA 3.4. *If G is a simple, finite planar graph with v vertices ($v \geq 3$), then G has at most $3v - 6$ edges [Harary 1994].*

LEMMA 3.5. *The upper bound of the average degree of a simple, finite, planar graph is 6.*

PROOF. Average degree $\leq \frac{2(3v-6)}{v} = 6 - \frac{12}{v} \leq 6$ □

LEMMA 3.6. *The total number of object paths passing $n + 1$ regions is $O(V(m)(\frac{5}{2})^n)$, where $V(m)$ is the number of vertices in the dual graph.*

PROOF. An object path from one region to another in the deployment graph corresponds to a path from and to the corresponding vertices in the dual graph. We have $V(m)$ choices to start. At each step we will try to extend the path from one of its end vertices. Thus we have $e_i - 1$ choices to extend the path, where e_i is the degree of that end vertex and $i = (1, 2, \ldots)$. Since we can extend the path at both end, the new path will be counted twice considering that we can count it from either ends. Now the total number of object paths passing $n + 1$ regions is given by $P_{curve}(m, n) = O(V(m)\underbrace{(\frac{e_1-1}{2})\cdots(\frac{e_n-1}{2})}_{n}) = (V(m)(\frac{1}{2})^n(\frac{\sum^n e_i}{n} - 1)^n)$. When

$n \to v$, $\frac{\sum^n e_i}{n} \to \frac{\sum^v e_i}{v} \leq \frac{2(3v-6)}{v} = 6 - \frac{12}{v} \leq 6$. Substitude this to the expression of $P_{curve}(m, n)$, we have $P_{curve}(m, n) = (V(m)(\frac{1}{2})^n(6-1)^n) = (V(m)(\frac{5}{2})^n)$. □

LEMMA 3.7. *The total number of sensor sequences for m sensors and a length of n is* $S_{curve}(m, n) = m(m-1)^{n-1}$.

PROOF. In a sensor sequence you can select the first sensor from any of the m choices. The second sensor can be any of the $m-1$ candidates as long as it is different from the first one. The ith sensor can be any of the $m-1$ candidates as long as it is different from the $(i-1)$th sensor. Thus in total we have $S_{curve}(m, n) = m(m-1)^{n-1}$ unique sensor sequences. □

Two assumptions are made to further estimate the order of $V(m)$:

(a) The deployment graph is a simple graph. This means there could be multiple edges between the vertex for object space boundary and other vertices, but the number of such edges is small compared with the total number of edges.
(b) The layout of each boundary sensor is assumed to be approximated with polynomial curve with order up to K. This is based on the fact that in real world the curve of a boundary sensor won't have infinite number of "twist" (This assumption is for twistable sensor such as fiber sensors, there could be other assumptions for other sensor forms.)

LEMMA 3.8. *If G is a connected planar graph with v vertices, r faces, and e edges, then its dual graph G* has r vertices, v faces, and e edges. [Aigner and Ziegler* 1998*]*

LEMMA 3.9. *In the deployment graph, the number of vertices is of the same order as the number of regions.*

PROOF. Assume the number of vertices, edges, and regions in the deployment graph are v, e, and r. According to Euler's formula [Aigner and Ziegler 1998], we have $v - e + r = 2$. From Lemma 3.4 and Lemma 3.8 we have $e \leq 3v - 6$ (derived from the deployment graph) and $e \leq 3r - 6$ (derived from the dual graph) Combining these three, we get $\frac{v}{2} + 2 \leq r \leq 2v - 4$. Thus we have $v = O(r)$ and $r = O(v)$. □

LEMMA 3.10. *Two polynomial curves with order up to K have at most K intersections.*

THEOREM 3.11. *The ratio of the total number of paths* $P_{curve}(m, n)$ *passing* $n+1$ *regions to the total number of sensor sequence* $S_{curve}(m, n)$ *of length n is*

$$\frac{P_{curve}(m, n)}{S_{curve}(m, n)} = O\left(\left(\frac{5}{2(m-1)}\right)^n V(m)\right), \tag{6}$$

where $V(m)$ *is the number of vertices in the dual graph or number of regions in the deployment graph. If we further assume the deployment graph is a simple graph and the deployment of each boundary sensor can be described by a*

polynomial curve with order up to K, Equation (6) *becomes*

$$\frac{P_{curve}(m,n)}{S_{curve}(m,n)} = O\left(\left(\frac{5}{2(m-1)}\right)^{n-2}\right). \tag{7}$$

PROOF. Equation (6) can be derived from lemma 3.6, 3.7 and note that $P_{curve}(m,n) = m(m-1)^{n-1} > (m-1)^{n-1}$.

From Lemma 3.10, we find $V(m) = r = O(v) = O(K(\binom{m}{2})) = O(m^2)$. Substituting $V(m)$ into Equation (6) yields Equation (7) □

Theorem 3.11 confirms our speculation stated earlier this section. When $m \geq 4$ and $n \geq 3$ (which is easy to be met in real world applications), we find that $P_{curve}(m,n)$ will be far smaller than $S_{curve}(m,n)$ when m or n increases. Since each path is always associated with a sensor sequence (but not vice versa), a pool of sensor sequences with far larger quantity than that of the paths means we have great possibility to make each path associate with a distinct sensor sequence.

When both the boundary sensor and the object path are straight lines, Theorem 3.11 can be further simplified.

LEMMA 3.12. *When both the boundary sensor and object path are limited to straight lines that extend to infinity,*

$$\frac{P_{curve}(m,n)}{S_{curve}(m,n)} = O\left(\frac{m^4}{(m-1)^n}\right). \tag{8}$$

PROOF. Let L be a set of m lines (boundary sensor). We dualize L to a set $L^*$of points. The duality transform preserves above-below relationship, that is, if a point p lies above (resp.\below, on) a line l, then the line p^*dual to p lies above (resp.\ below, on) the point l^* dual to l. Let H be the set of $O(m^2)$ lines connecting every pair of points in L^*; each line in H is dual to an intersection point of lines in L. The lines in H induce a planar subdivision, denoted by $A(H)$, whose vertices are intersection points of H, the edges are the maximal portions of lines in H that do not contain any intersection point, and faces are the maximal connect regions that do not intersect any line of H. Let p and q be two points that lie on the same edge or face of $A(H)$ Then the lines p^* and q^* intersect the lines of L in the same order because all intersection points of L lie on the same side of p^* and q^*. Note that p^* and q^* are the straight object paths and they should be regarded as the same path if they satisfy the above conditions. Hence, there are $O(m^4)$ different object paths. Combine this result with Lemma 3.7; then we get Equation (8) □

THEOREM 3.13. *When the boundary sensors are straight lines and object paths are straight lines passing* $n+1$ $(n \leq m)$ *regions, we have*

$$\frac{P_{curve}(m,n)}{S_{curve}(m,n)} = O\left((m-n+1)\frac{m^4}{(m-1)^n}\right). \tag{9}$$

PROOF. The number of straight-line path passing $n+1$ regions along a particular line is $m-n+1$, combined with Lemma 3.12 yields Equation (9) □

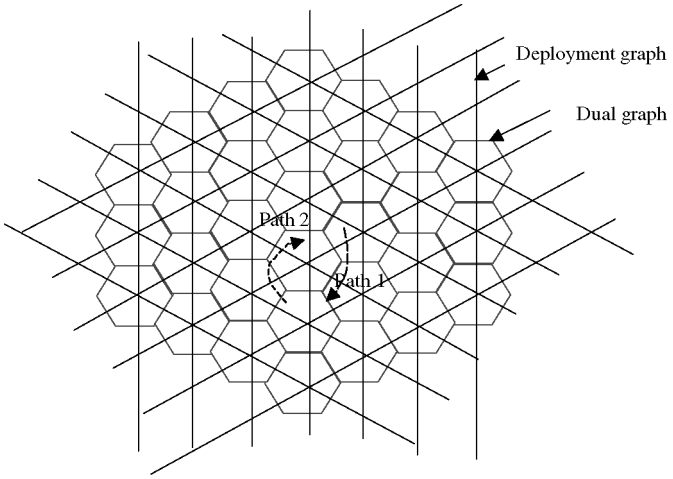

Fig. 10. The deployment graph (boundary not drawn) and dual graph for the triangle grid.

To conclude this section, we will also discuss a special nonrandom 2D deployment scheme that is a generalization of the 1D case in Section 2. As discussed before, 1D boundary sensor networks arranged orthogonally in the plane may be used to localize sources in 2D. This Cartesian deployment will fail to localize along one axis, however, if the object only moves along a line perpendicular to that axis. For example, if an object only makes moves parallel to the x axis, then we are not able to resolve its y coordinate with the Cartesian deployment. To solve this problem, a third parallel set of straight boundary sensors can be included so that they will coincide with the diagonals, as shown in Figure 10.

THEOREM 3.14. *In a deployment graph formed by a triangle grid where each straight line corresponds to a distinct sensor, object paths that share common sensor sequences of length n* ($n \geq 2$) *have to be the same or they must end at the two triangle regions which use the two vertical angles assoicated with a vertex. (See path* 1 *and path* 2 *of Figure* 10*)*

PROOF. We will first map the object path in the deployment graph into the path along edges in the dual graph. We number each of the three parallel sets of boundary sensors with independent of number set a, b, and c. ($a = 1, 2, \ldots, b = 1, 2, \ldots, c = 1, 2, \ldots$). Next in the dual graph we assign a 3-element coordinate (a, b, c) to each vertex, where the value of each element is determined by number of the corresponding edge. (The index of the edge is determined by the number of boundary sensor it corresponds to). Figure 11 shows the coordinate of the vertices of one of the hexagon in the dual graph. We notice that the vertices have the same value for 2 out of 3 elements in their coordinate can only be on the two ends on a diagonal.

If two paths are different in the dual graph, at least at one particular step their vertices should be different. Assume they are v_1 at path 1 and v_2 at path 2. Since in the next step they will trigger the same sensor, at least one of their coordinate elements should be the same and this element will not change after

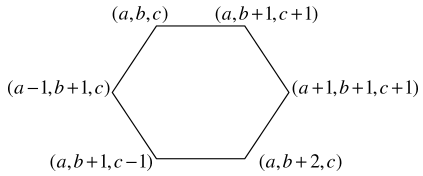

Fig. 11. The coordinate system for the dual graph of a triangle grid. If two vertices have 2 coordinate elements in common, they have to be on the two ends of the diagonal of a hexagon.

this step. Without losing generality, let us assume after this step, $v_1 = (a, b_1, c_1)$ and $v_2 = (a, b_2, c_2)$. Now starting from v_1, v_2, since they are going to trigger another sensor, another element in the coordinate should be the same. (Keep in mind in the path of the object is not allowed to directly turn back to trigger the same sensor again; otherwise a new tracking sequence will begin). Without losing generality, we assume it is the second element; thus, we can rewrite the coordinate of v_1, v_2 as $v_1 = (a, b, c_1)$ and $v_2 = (a, b, c_2)$. Now since v_1, v_2 share two common elements, they must be on the two ends of the diagonal of a hexagon. The same analysis can be applied to following vertices in the paths. We will find if two paths share the common sensor sequence, they have to be the same or part of the two paths have to be symmetric on a particular hexagon in the dual graph, which correspond to the two triangle regions in the deployment graph using the two vertical angles assoicated with a vertex.

Thus Theorem 3.14 guarantees that the tracking error for a triangle grid is localized within a small neighborhood. □

4. CONCLUSION

We propose a new paradigm for sensor network that tracks moving objects based on detected sensor signal sequences in the time domain. A complete analysis is given for the 1D case, based on graph theory; we find the most efficient boundary sensor design and give the construction method for several cases. This 1D boundary sensor has potential application area such as tracking automobiles along a street, machine position along an axis, or human position in a plane.

For random deployment of a boundary sensor array in 2D domain, we give the upper bound of the ratio of the number of total paths to the number of sensor sequence, when the path and sensor can be either arbitrary curves or limited to straight lines. We find in both cases, this ratio decreases rapidly when the number of sensors or the length of path increase, indicating great possibility to have each path associated with a unique sensor sequence. We also prove that a triangle grid can track an object with the error localized to a small neighborhood when the length of the path is no smaller than 2.

REFERENCES

AIGNER, M. AND ZIEGLER, G. M. 1998. *Proofs from the Book*. Springer-Verlag, Berlin, Germany.

BELLUTTA, P., ANCONA, N., AND POGGIO, T. 1991. A floor boundary sensor for autonomous robot navigation. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI)*.

BOLLOBAS, B. 1979. *Graph Theory: An Introductory Course*. Springer-Verlag, Berlin, Germany.
CHUNG, F. R. K. 1997. *Spectral Graph Theory*. American Mathematical Society, Providence, RI.
CLOUQUEUR, T., PHIPATANASUPHORN, V., RAMANATHAN, P., AND SALUJA, K. K. 2003. Sensor deployment strategy for detection of targets traversing a region. *Mob. Netw. Appl. 8*, 4, 453–461.
DE BRUIJN, N. AND VAN AARDENNE-EHRENFEST, T. 1951. Circuits and trees in oriented linear graphs. *Simon Stevin 28*, 203–217.
DOUGHERTY, B. C., BUNCH, R. M., LUKENS, R. E., NAGEL, J., AND WOLFE, W. L. 2004. Passive infrared device for detection of boundary crossings. United States Patent Application 20040129883.
GOPINATHAN, U., BRADY, D. J., AND PITSIANIS, N. P. 2003. Coded apertures for efficient pyroelectric motion tracking. *Optics Express 11,* 18, 2142–2152.
HARARY, F. 1994. *Graph Theory*. Addison-Wesley, 376–379.
HUANG, C.-F. AND TSENG, Y.-C. 2003. The coverage problem in a wireless sensor network. In *ACM International Workshop on Wireless Sensor Networks and Applications*.
LUCAS, E. 1891. *Recreations Mathematiques*. Gauthier-Villares, Paris, France.
MEGUERDICHIAN, S., KOUSHANFAR, F., POTKONJAK, M., AND SRIVASTAVA, M. 2001a. Coverage problems in wireless ad-hoc sensor networks. In *Proceedings of INFOCOM*. 1380–1387.
MEGUERDICHIAN, S., KOUSHANFAR, F., QU, G., AND POTKONJAK, M. 2001b. Exposure in wireless ad-hoc sensor networks. In *Proceedings of MOBICOM*. 139–150.
MEGUERDICHIAN, S., SLIJEPCEVIC, S., KARAYAN, V., AND POTKONJAK, M. 2001c. Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure. In *Proceedings of MOBIHOC*. 106–116.
POTULURI, P., GOPINATHAN, U., ADLEMAN, J. R., AND BRADY, D. J. 2003. Lensless sensor system using a reference structure. *Optics Express 11,* 8, 965–974.
SMITH, C. AND TUTTE, W. 1941. On unicursal paths in a network of degree 4. *Amer. Math. Month. 48*, 233–237.
TIAN, D. AND GEORGANAS, N. D. 2002. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *ACM International Workshop on Wireless Sensor Networks and Applications*.
YE, F., ZHONG, G., LU, S., AND ZHANG, L. 2003. A robust energy conserving protocol for long-lived sensor networks. In *International Conference on Distributed Computing Systems*.
ZAHND, S., LICHTSTEINER, P., AND DELBRUCK, T. 2003. Integrated vision sensor for detecting boundary crossings. In *Proceedings of the IEEE International Symposium on Circuits and Systems*. 376–379.
ZHENG, Y., BRADY, D. J., SULLIVAN, M. E., AND GUENTHER, B. 2005. Fiber optical localization by geometric space coding with 2d gray code. *Appl. Optics 44,* 20, 4306–4314.
ZHENG, Y., PITSIANIS, N. P., AND BRADY, D. J. 2006. A fiber sensor floor web for multi-person tracking. *IEEE Sensor J.* To appear.

Received August 2006; accepted April 2007