Adaptive Event Coverage Using High Power Mobiles Over a Sensor Field

Jagadeesh Balam and Jerry D. Gibson
Department of Electrical and Computer Engineering
University Of California, Santa Barbara, California 93106-9560
Email: balam@engr.ucsb.edu and gibson@mat.ucsb.edu
Telephone: (805) 893–4404, Fax: (805) 805–2930

***Abstract*— We consider a high density network of low power, stationary sensor nodes (motes) with an overlay of fewer, higher power mobiles that respond to events sensed by the motes. The motes sense the event(s) in progress and guide the much fewer, more expensive mobiles to move to locations which provide best coverage of the event. The mobiles use the distributed Lloyd algorithm to optimally distribute the entire set of mobiles across the event area. In the distributed Lloyd algorithm, mobiles communicate only with their neighbor mobiles and do not have available the locations of the remaining mobiles. We study the effect of constraining the distance within which the motes can communicate with the mobiles. Then we suggest an energy-efficient model for communication between the mobiles and the motes. We also investigate the effect of varying the mote density on the coverage provided by the mobiles.**

I. Introduction

We consider a scenario where a large number *n* of low power stationary sensor nodes (henceforth referred to as motes) are spread over a field for monitoring an event, say a fire or a gas leak. There are *m(<<n)* mobile sensor nodes in the field which can communicate with each other and also collect data from the motes in the field. Each mobile node uses the information provided by the motes, to move to a location which gives better *coverage* of the event.

One of the significant features of low power stationary motes is their low cost, which makes it feasible to have a network with motes on the order of thousands. A high density of motes guarantees high spatial resolution and fault tolerance, but addition of more functionality to these motes might not be feasible due to the low cost and low power requirements. Such an added functionality could be provided using high power mobile sensor nodes. However, the high cost of mobile sensor nodes constrains the number of mobile sensors that can be deployed in a sensor network. The quality-of-service / sensor-coverage at a particular area in the field can be improved by increasing the density of motes in that area, but when the area of significance is changing with time, increasing the number of motes is not a feasible solution. High power mobile sensors can provide the flexibility to adapt to a changing environment and provide better quality-of-service at the critical area. We call this *Adaptive Event Coverage*.

Mobile robots equipped with sensors for physical phenomena have exciting applications in wireless sensor networks. Such robots in conjunction with the low-power motes could make both detection and correction of a problem autonomous. The mobiles could use the data collected from the motes to track down a target or to determine the source of a problem. In situations where accurate data is required from a specific area in the field, mobile robots fitted with better sensors and processors when compared with the motes could move to the critical area for collection and also analysis of data. Mobile robots fitted with cameras could also be used for visual monitoring of a critical area.

In [1], it is proposed to use mobile robots as a gateway to the low power sensor network. The robots can collect the data, interpret and also take the necessary action instead of waiting for the base station to do the processing. Not only does such a setup decrease communication costs for the low-power motes but additional tasks like repair or recalibration of out-of-tolerance sensors can be carried out by the robots. In [2], it is suggested to use powerful mobile agents to perform critical network operations rather than burdening the low power sensors with sophisticated processing.

In [3], the problem of optimizing the quality of service/coverage provided by an adaptive sensor network in a dynamic environment was introduced. Coverage, here, is not the coverage of area in the sensor field, but coverage of an event in the sensor field. Each point in the sensor field is weighted on the basis of the event distribution. The weight could be a measure of the probability of the event at the point or a measure of changes in physical phenomena due to the event. Different kinds of coverage problems have been addressed hitherto in the sensor network literature. In [4] algorithms to reduce Worst case coverage (Maximal Breach Path) and best case coverage(Maximal support path) are presented. In [5], the authors talk about a network providing various degrees of coverage while keeping the nodes connected. To distinguish our problem from the other coverage problems, we call it an *Event-Coverage* problem.

A. Adaptive Event Coverage

The goal of the *Adaptive Event Coverage* problem is to provide coverage to events in a dynamic environment. The mobile sensors (mobiles) move to locations which give best coverage of the event. This problem was introduced by Cortés et al. in [3]. They call this the *optimal sensor placement problem*. The sensing performance of a node at a point q in the field degrades with an increase in the distance between the point q and the node p_i. If this degradation is represented by a non-decreasing function $f(||q-p_i||)$ and the event density function is ϕ, then, for m mobile nodes, the coverage problem can be expressed as the minimization of the following locational optimization function,

$$J(P,W)=\sum_{i=1}^{m}\int_{W_i} f(||q-p_i||)d\phi(q) \qquad (1)$$

where the optimization function J depends on the positions p_i of the mobile nodes and the dominance region W_i assigned to each node. The dominance region of a node i is the region where the ith node's sensing performance is better than that of all other nodes. The optimization function J in Eq. (1) is similar to average distortion D minimized in the classic problem of the design of minimum-distortion vector quantizers [6].

In vector quantization, vectors from k-dimensional Euclidean space are mapped onto a finite set of code vectors. The design problem is to find the set of code vectors and an encoding rule that minimizes a chosen distortion measure. The encoding rule specifies the partition of the k-dimensional space that is assigned onto a code vector or the set of vectors that are reproduced by the code vector. The code vectors in the vector quantization design problem are analogous to the mobiles in the coverage problem. In fact the locational optimization problem addressed here is similar to a two dimensional vector quantization problem. The problem of finding the locations at which the mobiles give the best coverage of the area is analogous to finding the points in a 2D space which represent the other points in the space with minimum average distortion.

In the next section, we present an overview of the distributed Lloyd algorithm presented in [3] in a scenario with motes. In Section III, we discuss the effect of the number of motes on the performance of the Lloyd algorithm. Section IV deals with developing a communication model for the Lloyd algorithm, followed by conclusions in Section V.

II. Distributed Lloyd Algorithm for Adaptive Event Coverage

The most popular algorithm for the design of vector quantizers is the Lloyd algorithm [7]. This algorithm successively applies the two necessary optimality conditions on the partitions (Nearest Neighbor condition) and the positions (Centroid condition) of the code vectors to monotonically reduce the average distortion in successive iterations . For squared error distortion, it is observed that the optimal partitions are Voronoi partitions of the code vectors , i.e., all the vectors nearest, in a squared distance sense, to a code vector are assigned to it.

The Lloyd algorithm can easily be applied to find the optimal positions for mobiles at which the mobiles provide best coverage of an event. Cortés et al [3] present a distributed Lloyd algorithm for coverage control of mobile sensing networks. The algorithm is designed to be *adaptive* to changing environments and sensing tasks, *distributed*, i.e., each mobile needs to know only its neighbors' locations, and *asynchronous*, so global synchronization is not required. The algorithm is also guaranteed to converge to centroidal Voronoi configurations like the generalized Lloyd algorithm. Each mobile first finds its Voronoi partition with only the knowledge of the locations of neighbor mobiles. In the next step each mobile moves towards the centroid of its Voronoi partition calculated using the density function ϕ.

The problem here is that we do not usually have the knowledge of the distribution of the event, a priori. Also, we want the mobiles to track transient events in real time. Hence, we use the suggested setup, wherein a large number of motes are distributed over the sensor field to monitor for events. The low-cost motes spread over the field send samples of some measure of the event, like temperature when there is a fire, to the mobiles.The mobiles use this data to calculate the centroid of their respective Voronoi cell. For example, for an event like a fire, the mobiles need to move in the direction of increasing temperature gradient. Suppose the motes provide the temperature at their location to the mobiles. The mobiles can use the temperature readings of the motes within their Voronoi partition to calculate the centroid. If t_i is the temperature recorded by mote *i* located at (x_i,y_i) then

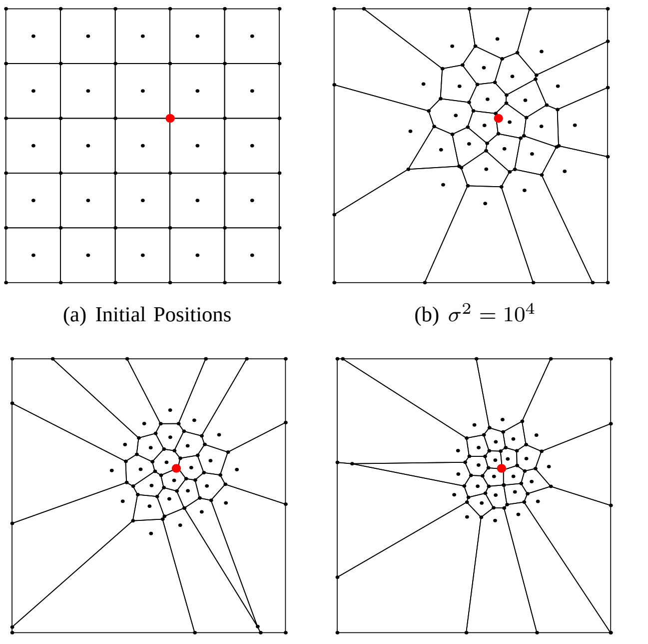
$$Centroid(k)=\frac{\sum(t_i*(x_i,y_i))}{\sum(t_i)}\forall(x_i,y_i)\in V_k$$

where V_k is the Voronoi partition of mobile *k*. The key steps in the distributed Lloyd algorithm are summarized in Fig. 1.

Each mobile *i* at time t_i:
1) Identifies the neighbor mobiles, *2) Computes the Voronoi polygon* *3) Broadcasts a request message to all the motes within the Voronoi Polygon to send their data* *4) Calculates the centroid (weighted mean) of the polygon using the data d_k sent by the motes at (x_k,y_k) as* $Centroid(i)=\frac{\sum(d_k*(x_k,y_k))}{\sum(d_k)}\forall(x_k,y_k)\in V_i$ *5) Move with a velocity* u *that is proportional to the distance between the current location and the centroid, towards the centroid for a time* δt

Fig. 1. Distributed Lloyd Algorithm

We earlier mentioned an interesting application of mobile sensor nodes from [1], where the mobiles are used as a gateway to the low power sensor network. Here we consider sensor networks with a hierarchal networking architecture consisting of the stationary low power nodes (motes) and high power mobiles. Each mobile acts as a cluster node for all the motes within its Voronoi partition. Consider the problem of minimizing the energy spent by the motes in communicating information to the mobiles. If we assume that the energy spent in communication increases with the squared distance between the mote and the mobile, then we can use the distributed Lloyd algorithm to find the locations of the mobiles which minimize this energy. In [8], an energy efficient clustering algorithm is presented, wherein each non-clusterhead node joins its nearest cluster-head, forming a Voronoi tessellation. Each mobile finds the locations of motes within its Voronoi partition and moves towards the mean (all the motes have equal weights attached) of the locations. All the motes join the nearest mobile forming



(a) Initial Positions

(b) $\sigma^2 = 10^4$

(c) $\sigma^2 = 10^4/2$

(d) $\sigma^2 = 10^4/3$

Fig. 2. Distributed Lloyd algorithm applied on 25 mobiles over a sensor field 500 m wide with a Gaussian event distribution given by $\phi = exp(-((x - 300)^2 + (y - 300)^2)/\sigma^2)$

a Voronoi tessellation. Such a setup can also made adaptive to node (both mote and mobile) failures.

III. Density of the Sensor Field

One of the significant factors that would affect the performance of the Lloyd algorithm in the current scenario is how accurately the mobiles know the density function ϕ. The motes provide spatial samples of the continuous function ϕ to the mobiles. For ϕ to be accurately represented, the number of motes should tend towards infinity, pushing the spatial resolution to zero. An increase in the number of motes also increases the communication and computation overheads of the network. The time taken to converge would also increase with an increase in the number of motes.

We simulated a square sensor field 500 m per side with 25 mobiles distributed over the field. In our simulations, we investigated the effect of the number of motes on performance of the Lloyd algorithm. Our implementation of the Lloyd algorithm is distributed but not asynchronous. Each mobile calculates its Voronoi partition from the knowledge of the positions of only its neighbors but all the mobiles move towards the centroids at the same time. The setup was simulated using *Mathematica*. The initial positions of the mobiles also affect the local optimum reached. Each simulation was run with the mobiles at the same initial positions. The centers of square partitions of the field were chosen as the initial positions of the mobiles. Figure 2 shows initial positions of the mobiles (2(a)) and final positions (2(b) 2(c) 2(d)) of the mobiles for three different variances of the event distribution.

The performance of the algorithm was evaluated using the value of the optimization function J. Continuous integrals

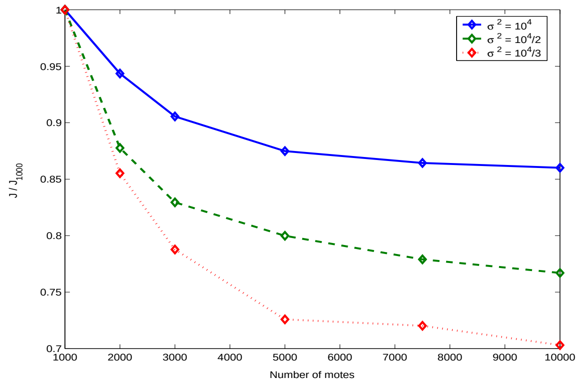


Fig. 3. J/J_{1000} vs Number of motes. As the density of motes increases the value of J decreases

of the density function ϕ over the Voronoi partitions of the mobiles ($\int_{W_i} f(||q - p_i||)d\phi(q)$), at their final positions, were used in evaluating J. The event distribution is given by a Gaussian function with the mean inside the sensor field. The algorithm was run for different numbers of motes and different variances of the event distribution till the mobiles converged to a local minimum. Figure 3 shows that the value of J decreases, i.e., the coverage provided by the mobiles improves with an increase in the number of motes. To make the results for different variances comparable we divided J obtained by the value of J obtained for 1000 motes in each case. J was averaged over 10 runs for each mote density. Observe that there is no significant change in J for an increase in the number of motes over 7500 when $\sigma^2 = 10^4/2$ and there is no significant improvement in coverage for more when the number of motes are increased above 5000 for a variance of 10^4.

As expected the coverage provided by the mobiles improves with an increase in the number of motes. Since the increased density is only helpful to guide the mobiles when an event is detected, when there is no event we can use a mechanism where only a fraction of the motes in the field are kept awake. A scheduling scheme such as the one presented in [9] can be used for the motes. In this scheme redundant motes are switched off keeping the coverage provided by the motes constant. Here *coverage* is the area covered by the motes in the sensor field.

A. The problem of local optima

The Lloyd algorithm can get stuck in a local optimum rather than reaching a global optimum. Here we investigate the algorithm performance when compared to a global optimum. Gray and Karnin [10] demonstrate the existence of multiple local optima. In [11], Zeger et. al. present techniques to find a globally optimal vector quantizer. The idea is to introduce randomness into an otherwise deterministic Lloyd algorithm. We compare the performance of the Lloyd algorithm in the current scenario with a solution found using one of the techniques suggested in [11]. We chose the reduced complexity SR algorithm for decoder perturbation (SR-D algorithm) to find a global optimum because it can be implemented as a distributed algorithm. In this algorithm a random noise is

TABLE I

COMPARISON OF LOCAL OPTIMA AND GLOBAL OPTIMA FOR DIFFERENT MOTE DISTRIBUTIONS

Seed	J		% diff.
	Lloyd alg.	SR-D alg.	
1	2278.15	2236.43	1.831
2	2284.35	2273.40	0.48
3	2421.21	2252.46	6.9

added to each mobile's location at the end of each iteration. The mobiles calculate their Voronoi polygons and then find the centroid of their partition. Then a random noise of decreasing variance is added to each of the centroids. This algorithm can be adapted to a distributed case by simply allowing the mobiles to add a random noise to their centroid calculations starting from the time when the event was first detected. Each mobile needs to add a noise of decreasing variance to the calculated centroid for a certain number of steps and then follow the Lloyd algorithm without adding the noise till they converge.

We choose σ_x as 100 and the number of iterations for which SR is applied, I, is 100. The value of p was chosen to be 3 as suggested in [11]. The number of motes was fixed at 3000 and the event distribution is Gaussian in a square field of width 500 m. The variance of Gaussian random noise added to the centroids at the mth iteration is given by

$$T_m = \sigma_x^2 (1 - \frac{m}{I})^p \tag{2}$$

The experiment was run for different mote locations and the value of J achieved by the Lloyd algorithm was compared to the value of J at the global optimum found using the SR-D algorithm. Observe from Table I the difference between the local optimum and the global optimum. The largest difference seen here is 6.9%. It is difficult to comment on how significant this difference is in a real scenario but the overhead due to the SR-D algorithm is significantly larger. On an average the Lloyd algorithm took 45 iterations to converge whereas the SR-D algorithm took 100 iterations.

IV. COMMUNICATION MODEL FOR THE DISTRIBUTED LLOYD ALGORITHM

In the present scenario, motes need to communicate with the mobile at each iteration of the Lloyd algorithm. Communication is a costly affair for the low-power motes. We try to find a suitable communication model for the motes so as to minimize the energy spent in communication with the mobiles. We assume that the mobiles have dual communication capabilities, i.e., they use a different mechanism like 802.11 for communication with the other mobiles and radio waves for communication with the motes. We assume that all the mobiles and the stationary motes have knowledge of their global positions. We are not concerned about energy spent by the mobiles since we assume them to be high-power nodes.

The mobiles communicate with their neighbors to construct the Voronoi diagram. The mobiles then broadcast a request signal to the motes to send the data. Mobiles need data from only the motes within their Voronoi cell. Motes could receive multiple requests from different mobiles but a mote has to send data to the mobile whose Voronoi cell it belongs to. We neglect this energy loss in the motes due to receipt of multiple request signals. Such a loss may be avoided if the mobiles broadcast the signals using directional antennas and limit the communication to the motes within their respective Voronoi polygons.

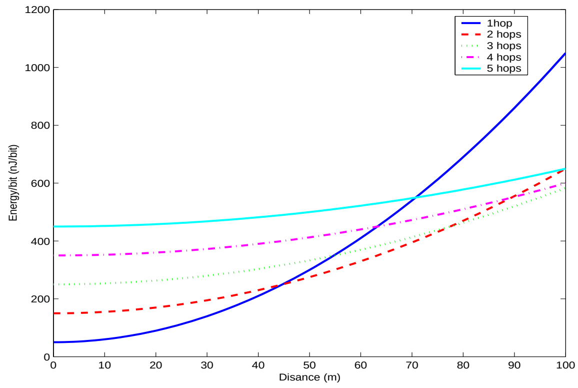


Fig. 4. Energy expended by the radio

To avoid communication from the motes outside the Voronoi polygon of the mobile, a mobile could broadcast the vertices of its Voronoi polygon. Each mote runs a simple check to find whether it lies inside the Voronoi polygon and responds only if it belongs to the Voronoi polygon. Algorithms exist to check whether a point lies inside a polygon. These algorithms are simpler and faster for convex polygons, which the Voronoi polygons are.

As the Lloyd algorithm converges, the mobiles are concentrated in the area where the event density is high. The Voronoi polygons of the mobiles that are away from the mean of the event distribution increase in area. At each iteration, the number of motes within the Voronoi polygons of these mobiles increases. Communication with all the motes within the polygon is not only time consuming but also energy draining. We limit communication to the motes lying within a distance from the mobile. We examine how the coverage provided changes when such a limitation is imposed.

We consider two models of communication for the motes. The motes can either use a multi-hop network to communicate with the mobile or directly communicate with the mobile. Min and Chandrakasan study energy-efficient communication for sensor networks in [12]. They state that for sensor networks with communication over small distances direct communication of the information consumes less energy than communication using multiple hops. According to the radio model suggested in [13], the transmitter or receiver circuitry consumes E_{elec}=50 nJ/bit and the transmit amplifier needs ϵ_{amp}=100 pJ/bit/m^2. The radio expends E$_{Tx}(k, d)$ to transmit a *k*-bit message over a distance *d* and E$_{Rx}(k)$ to receive a *k*-bit message, where

$$\begin{aligned} E_{Tx}(k,d) &= E_{Tx-elec} * k + \epsilon_{Tx-amp} * k * d^2 \\ E_{Rx} &= E_{elec} * k \end{aligned}$$

Figure 4 shows the energy expended per bit for varying d and h. Observe that for smaller distances (<50m) single-hop

TABLE II

AVERAGE ENERGY EXPENDED FOR COMMUNICATION

Communication Model	Energy(μJ/bit/mote)	J
Multi-hop (all motes)	41.65	2229.28
Multi-hop (motes within 50 m)	5.51	2480.15
Direct (motes within 50 m)	2.502	2480.15

or direct communication is least energy consuming. A similar evaluation has been done in [12] for an r^3 path loss model, where for the parameters assumed, direct communication (single hop) is observed to be energy-efficient for $d < 30$ meters. Also when overheads due to protocol and MAC layer are considered, multi-hop communication consumes significantly more energy than simple direct communication.

A significant amount of energy is expended in the startup of radio in a mote. Startup of the radio module usually takes hundreds(more than 400) of microseconds and is much larger than the time taken for transmitting small packets of data. The motes in our problem transmit their location and the corresponding sensor reading to the mobile. If we assume each value takes 4 bytes, then the mote transmits 96 bits (8 bytes for location, 4 bytes for the reading) for each request. At 1 Mbps rate, each transmission takes 96 microseconds. For multiple hops, nodes at each hop expend energy for startup of the radio. So more energy is spent in the startup of intermediate nodes than the energy spent for transmission of the data. Direct communication of information is more energy-efficient than multi-hop communication for transmitting small payloads. But as noted earlier, the Voronoi polygons of mobiles can be large and the motes which lie at the extremes may not be able to directly communicate with the mobile. We next investigate how the Lloyd algorithm performs when we limit communication to the motes located within a certain distance from the mobile.

A. Experiments and Results

We simulated a square sensor field 500 m wide with a high density of motes distributed over the field. We carried out our simulations for 5000 motes (1 per 50 m^2) uniformly distributed over the field to guide 25 mobiles in covering the event. The distribution of the event is represented by a Gaussian function ($\phi = exp(-((x-300)^2+(y-300)^2))$). We investigate the performance of the distributed Lloyd algorithm when the mobiles talk to the motes which are within a distance of 50 m from them. We also demonstrate the difference in energy expended when direct communication is used instead of a multi-hop network through these simulations.

The following are the assumptions made for the multi-hop communication. 1) The motes broadcast their message over a communication range *r* of 10 m. 2) We assume that a minimum-hop routing protocol exists and the message is transmitted with a minimum number hops. So the minimum number of hops required to communicate a message to the mobile that is at a distance *d* is given by $\lceil \frac{d}{r} \rceil$. 3) We neglect the MAC and network layer overheads. 4) The environment is error-free, so that the motes do not retransmit any data. For direct communication we assume an r^2 path loss.

The results averaged over five runs are presented in Table II. Observe that the loss in J is only about 11% when the mobiles communicate with only the motes within a 50 m radius and the energy consumed is reduced by 94%. Also observe that energy consumption is reduced by more than 50% when the mobiles directly communicate with the motes within 50 m from them instead of using a multi-hop network for communication.

V. CONCLUSIONS

We proposed a mechanism to provide adaptive event coverage in a sensor field using high power mobile sensor nodes and a large number of low power stationary sensor nodes. We use the distributed Lloyd algorithm presented in [3] to control the movement of the mobiles. We observed that there is no significant decrease in J when the mote density is increased above 1 per 50 m^2 when $\sigma = 100$. Observe that the gain achieved by increasing the number of motes is higher when the variance of the event distribution is lower. If we can estimate the lower bound on the variance of the event, then for a given number of mobiles we can find the number of motes that give close to optimum results for any event in the field. We studied the behavior of the algorithm when the mobiles communicate with only the motes within a certain distance from them and observed that the coverage provided is not affected significantly because of this limitation. Thus, we can use a direct communication model between the motes and the mobiles that saves a significant amount of energy when compared to multi-hop communication.

REFERENCES

[1] J. Butler. (2003) Robotics and microelectronics: Mobile robots as gateways into wireless sensor networks. [Online]. Available: http://www.intel.com/update/contents/it05031.htm

[2] L. Tong, Q. Zhao, and S. Adireddy, "Sensor networks with mobile agents," in *Proc. IEEE Military Comm. Conf.*, Boston, MA, Oct. 2003.

[3] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *IEEE Infocomm*, Mar. 2003.

[4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage problems in wireless ad-hoc sensor networ," in *IEEE INFOCOM01*, 2001.

[5] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Sensys*, Nov. 2003.

[6] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell,MA: Kluwer, 1992.

[7] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," in *IEEE Trans. Commun.*, vol. COM-28, Jan. 1980, pp. 84–95.

[8] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *IEEE INFOCOM03*, San Francisco, CA, 2003.

[9] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proc. ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, Oct. 2002.

[10] E. D. Karnin and R. M. Gray, "Multiple local optima in vector quantizers," in *IEEE Trans. Inform. Theory*, vol. 28(2), Mar. 1982, pp. 256–261.

[11] K. Zeger, J. Vaisey, and A. Gersho, "Globally optimal vector quantizer design by stochastic relaxation," in *IEEE Trans. Signal Proc.*, vol. 40(2), Feb. 1992, pp. 310–322.

[12] R. Min and A. Chandrakasan, "Energy-efficient communication for ad-hoc wireless sensor networks," in *Conf. Rec. of Thirty-Fifth Asilomar Conf. on Signals, Systems and Computers*, vol. 1, Boston, MA, Nov. 2001.

[13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *HICSS 00*, Jan. 2000.