9

Energy Efficient Data Acquistion in Wireless Sensor Network

Ken C. K. Lee[1], Mao Ye[2] and Wang-Chien Lee[2]
[1]Department of Computer and Information Science, University of Massachusetts Dartmouth, North Dartmouth,
[2]Department of Computer Science and Engineering, The Pennsylvania State University, University Park, USA

1. Introduction

Wireless sensor network (or sensor network, for brevity in the following) comes into practice, thanks to the recent technological advancement of embedded systems, sensing devices and wireless communication. A typical sensor network is composed of a number of wirelessly connected sensor nodes distributed in a sensed area. In the network, sensor nodes sense their surroundings and record sensed readings. The sensed readings of individual sensor nodes are then collected to present the measurement of an entire sensed area. In many fields including but not limit to, military, science, remote sensing Vasilescu et al. (2005), industry, commerce, transportation Li et al. (2011), public security Faulkner et al. (2011), healthcare and so on, sensor networks are recognized as important sensing, monitoring and actuation instruments. In addition, many off-the-shelf sensor node products Zurich (n.d.) and supporting software such as TinyOS Group (n.d.) are available in the market. Now sensor network application development is much facilitated. Many sensor networks are anticipated to be deployed soon.

Over those years, the computational capability and storage capacity of sensor nodes have been considerably improving. Yet, the improvement of battery energy is relatively small. Since battery replacement for numerous deployed sensor nodes is extremely costly and even impossible in hostile environments, battery energy conservation is a critical issue to sensor networks and their applications. Accordingly, how to effectively save battery energy is a challenge to researchers from academia, government agencies and industries. One common practice is to keep sensor nodes in sleep mode whenever they are not in use. During sleep mode, some hardware components of sensor nodes are turned off to minimize energy consumption. For instance, MICAz needs only 1μA when wireless interface is off and less than 15μA for processor in sleep mode Musaloiu-Elefteri et al. (2008). Besides, wireless communication is very energy consuming. For instance, MICAz consumes 17.4mA and 19.7mA in data sending and receiving, respectively, whereas it only needs 8mA for computation when its wireless interface and processor are on. Thus, reducing the amount of data transmitted between sensor nodes is another important means to save battery energy.

In many sensor network applications, data acquisition that collects sensed readings from remote sensor nodes is an essential activity. A primitive approach for data acquisition

can be collecting all raw sensed readings and maintaining them in a data repository for centralized processing. Alternatively, a large volume of raw sensed readings are streamed to a processing site where analysis and data processing are directly applied on streamed sensor readings Madden & Franklin (2002). However, costly wireless communication can quickly use up sensor nodes' battery energy. In other words, such a centralized approach is not energy efficient and thus undesirable in practice. As in the literature, a lot of original ideas and important research results have been developed for energy efficient data acquisition. Among those, many new techniques have been developed based on the idea of in-network query processing. Through in-network query processing, queries are delivered into sensor networks and sensor nodes evaluate the queries locally. By doing so, (partial) query results are transmitted instead of raw sensed readings. Since (partial) query results are in smaller size than raw sensed readings, energy cost can be effectively saved. Subject to the types of queries and potential optimization opportunities, various in-network query processing techniques have been developed and reported in the literature.

In this chapter, we review the main concepts and ideas of many representative research results on in-network query processing, which include some of our recent works such as itinerary-based data aggregation Xu et al. (2006), materialized in-network view Lee et al. (2007), contour mapping engine Xu et al. (2008) and in-network probabilistic minimum value search Ye, Lee, Lee, Liu & Chen (to appear). As briefly described, itinerary-based data aggregation is a new access method that navigates query messages among sensor nodes to collect/aggregate their sensed readings. Materialized in-network view is a novel data caching scheme that maintains (partial) query results in queried sensor nodes. Then, subsequent queries issued by different base stations can access cached results instead of traversing query regions from scratch to determine query results. Contour mapping engine derives fairly accurate contour line segments using data mining techniques. Besides, only the coefficients of equations representing contour line segments, which are very compact, are transmit. Finally, probabilitistic minimum value search is one of recent efforts in probabilistic sensed data aggregation. It finds the possible smallest sensed reading values in a sensor network.

The details of those works will be discussed in the following sections. First of all, we present a system model that our reviewed research results are based upon. Then, we discuss research results in in-network data aggregation and in-network data caching as well as in-network contour map computation. We further discuss recent results on in-network probabilistic data aggregation. Last but not least, we summarize this chapter and discuss some future research directions.

2. System model

Without loss of generality, a sensor network is composed of a number of battery powered stationary sensor nodes deployed over a sensed area. The spatial deployment of sensor nodes in a target sensed area is one of the research problems in sensor networks; and many research works (e.g. Bojkovic & Bakmaz (2008)) were proposed to maximize the area coverage by a given quantity of sensor nodes while providing required network connectivity among sensor nodes. The issue of sensor node deployment is usually considered to be independent from others. As will be discussed in the following, research works on data acquisition mostly assume that sensor networks are already set up and all sensor nodes are with identical hardware configurations.

In a typical sensor network, some senor nodes in the sensor network are directly connected to computer terminals; and they are called *base stations*. Through base stations, computer terminals can issue commands to administer sensor nodes and collect their sensed readings. Besides, all sensor nodes are wirelessly connected, e.g., MICAz uses 2.4GHz IEEE 802.15.4 radio. That means messages are all sent through wireless broadcast. When a node delivers a message, other sensor nodes within its radio coverage range can receive the message. Messages can be conveyed transitively from a sender sensor node to a distant target receiver node Xu et al. (2007). On the other hand, because of shared radio frequencies, simultaneous messages from closely located sensor nodes may lead to signal interference. Moreover, due to ad hoc connectivity and sensor node failure, which is common in practice, connections among sensor nodes are mostly transient and unreliable. Thus, other than regular data messages, every sensor node periodically broadcasts a special message called *beacon* to indicate its liveness to its neighboring sensor nodes. Also, data messages are sent through multiple paths from a sender sensor node towards a destination to deal with possible message loss Xu et al. (2007). As a result, those extra messages incur additional energy costs.

To save battery energy, sensor nodes stay in sleep mode for most of the time; and each of them periodically wakes up to sense its surrounding and record its measurements as sensed readings. For data acquisition, an entire sensor network (i.e., a set of sensor nodes N) presents a set of sensed reading values V, notationally, $V = \{v_n \mid n \in N\}$ where v_n is a sensed reading value provided by a sensor node n. Based on V, data analysis is conducted to understand the entire sensed area. As already discussed, it is very costly to collect V from all sensor nodes. Accordingly, some research results were reported in the literature exploring techniques to collect a subset of sensed readings $V'(\subset V)$ from a subset of sensor nodes N' $(\subset N)$, while collected readings may only provide approximate analytical results. The following are two sorts of techniques. Sampling is the first technique that sensed readings are only collected from some (randomly) selected sensor nodes Biswas et al. (2004); Doherty & Pister (2004); Huang et al. (2011). Those unselected sensor nodes do not need to provide their sensed readings. The sampling rate is adjustable according to the energy budget. The second technique is based on a certain prediction model Silberstein et al. (2006) that, some sensed readings can be omitted from being sent as long as they can be (approximately) predicted according to other sensed readings, which can be from some neighboring sensor nodes, or from the previous sensed reading values of the same sensor nodes. Meanwhile, another important research direction for energy efficient data acquisition based on in-network query processing Hellerstein et al. (2003) has been extensively studied; and we shall review some of the representative works in the coming four sections.

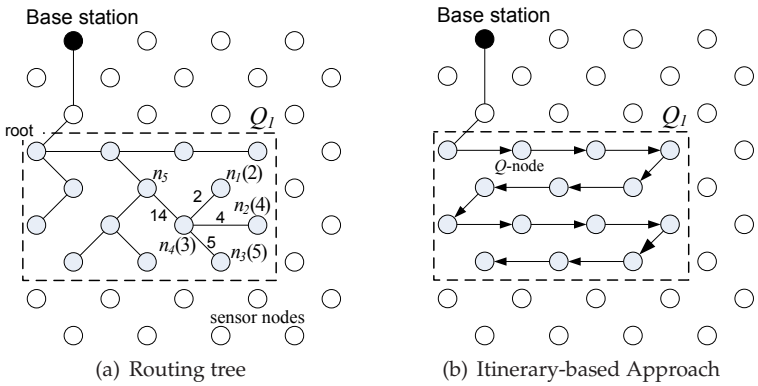
3. In-network data aggregation

Data aggregation is often used to summarize a large dataset. With respect to all sensed readings V from all sensor nodes N, an aggregate function f is applied on V to obtain a single aggregated value, i.e., $f(V)$. Some commonly used aggregate functions include SUM, COUNT, MEAN, VARIANCE, MAX and MIN etc. Aggregated data can provide a very small summary of sensed readings (e.g., the highest, average and lowest temperature) in a sense area. In many situations, it can be sufficient for scientists to know about a remote sensed area. Besides, aggregated data is usually small to transmit and data aggregation is not very computationally expensive for sensor nodes to perform so that in-network data aggregation

is very suitable to sensor networks. In the following, we discuss two major strategies, namely, *infrastructure-based approaches* and *itinerary-based approaches*, for in-network data aggregation.

3.1 Infrastructure-based data aggregation

As their name suggests, infrastructure-based approaches build certain routing structures among sensor nodes to perform in-network data aggregation. TAG Madden et al. (2002) and COUGAR Yao & Gehrke (2003) are two representative infrastructure-based approaches. They both form a routing tree to disseminate a query and to derive aggregated sensed readings in divide-and-conquer fashion. The rationale behind these approaches are two ideas. First, some aggregate functions f are decomposable so that $f(V)$ can be transformed to $f(f(V_1), f(V_2), \cdots f(V_x))$, where $V_1, V_2, \cdots V_x$ are sensed reading values from x disjointed subsets of sensor nodes and the union of all of them equals V, and f can be applied to readings from individual subsets of sensor nodes and to their aggregated readings. For example, $\mathsf{SUM}(V)$, where SUM adds all sensed reading values, can be performed as $\mathsf{SUM}(\mathsf{SUM}(V_1), \mathsf{SUM}(V_2), \cdots \mathsf{SUM}(V_x))$. Second, the connections among sensor nodes can be organized as a tree topology, in which the root of any subtree that covers a disjointed subset of some sensor nodes can carry out local aggregation on data from its descendant nodes. In other words, in-network data aggregation incrementally computes aggregated values at different levels in a routing tree.



(a) Routing tree (b) Itinerary-based Approach

Fig. 1. Strategies for in-network data aggregation

Figure 1(a) exemplifies a routing tree formed for data aggregation. In brief, upon receiving a SUM query for the total of sensed reading values from its connected computer terminal, a base station disseminates the query to sensor nodes within a specified queried region. The specified queried region can be a small area or an entire sensed area. With the queried region, sensor nodes join the routing tree when they receive the query. A node becomes the parent node of its neighboring nodes in a routing tree if those nodes receive the query from it. In a routing tree, the first queried node within the region serves as the root. Meanwhile, every non-root tree node should have another sensor node as its parent node, and non-leaf nodes are connected to some other nodes as their child nodes.

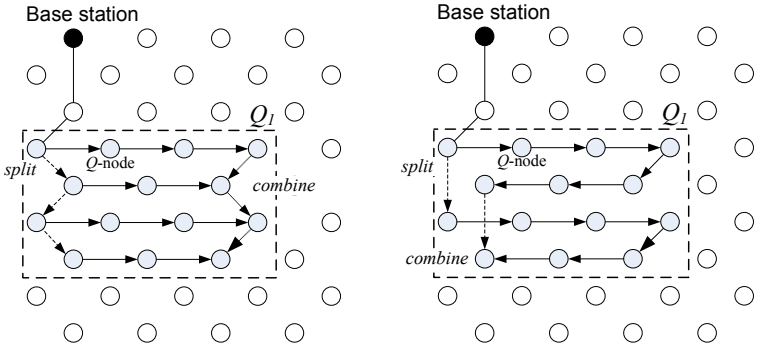
After the tree is built, data aggregation starts from leaf nodes. The leaf nodes send their sensed reading values to their parent nodes. Thereafter, every non-leaf node derives an aggregated

value based on (aggregated) sensed reading values received from its child nodes and its own sensed reading value. As shown in Figure 1(a), some leaf nodes n_1, n_2, n_3 first send their reading values of 2, 4 and 5, respectively, to their parent node n_4. Then, n_4 calculates the sum of their values and its own sensed reading values of 3, i.e., 14, and propagates it to its parent node n_5. Eventually, the root derives the final sum among all sensor nodes in the region and reports it to the base station.

3.2 Itinerary-based data aggregation

The infrastructure-based approaches relies on an infrastructure to perform in-network data aggregation, incurring two rounds of messages for both query dissemination and data collection. However, in presence of sensor node failure, queries and aggregated sensed readings would be lost making these approaches not very robust and reliable. Some additional research works Manjhi et al. (2005) were proposed to improve the robustness and reliability of routing trees by replicating aggregated values and sending them through different paths towards the root. However, it incurs extra data communication cost. To save the quantity of messages, we have recently developed itinerary-based data aggregation Xu et al. (2006).

The basic idea of itinerary-based data aggregation is to navigate a query among sensor nodes in a queried region as illustrated in Figure 1(b). In every step, a query message that carries both a query specification and an immediate query result is strategically sent from one sensor node to another along a designed space filling path called *itinerary*. The width of an itinerary is bounded by a maximum radio transmission range. Sensor nodes participating in forwarding a query message are called Q-nodes. After it receives a query message, a Q-node asks its neighboring nodes for their sensed readings. Then, the Q-node incorporates all received sensed readings and its own reading into the immediate query result. Thereafter, it forwards the query message with a new intermediate query result to a succeeding Q-node. Here, the succeeding Q-node is chosen by the current Q-node. If a Q-node fails, its preceding Q-node can detect it and re-propagates the query message to another sensor node as a replacement Q-node. As such, the itinerary can be resumed from that new Q-node. The evaluation of a query completes when a specified region is completely traversed. Finally, a query result is returned to the base station.



(a) Parallel Itinerary (b) Hybrid Itinerary

Fig. 2. Parallel and hybrid itinerary

On the other hand, the length of an itinerary directly affects the query processing time. A single itinerary takes a very long processing time, especially in a large query region. Thus, as opposed to single itinerary as shown in Figure 1(b), parallel itinerary has been developed to improve query processing time. As depicted in Figure 2(a), an itinerary is split into four threads scanning four rows in a region. Their immediate query results are then aggregated at the end of the rows. However, wireless signal from two adjacent threads may lead to signal interference, message loss and finally data retransmission. As a result, longer time and more energy are consumed. To address this issue, a hybrid itinerary has been derived accordingly. Here, a query region is divided into several sections that contain multiple rows. Inside each section, a single itinerary scans all the rows. For instance, as in Figure 2(b), a query region is partitioned into two sections, each covering two rows. Within each section, a sequential itinerary is formed. Now, because of wider separation, the impact of signal interference is minimized while a higher degree of parallelism is achieved, compared with single itinerary.

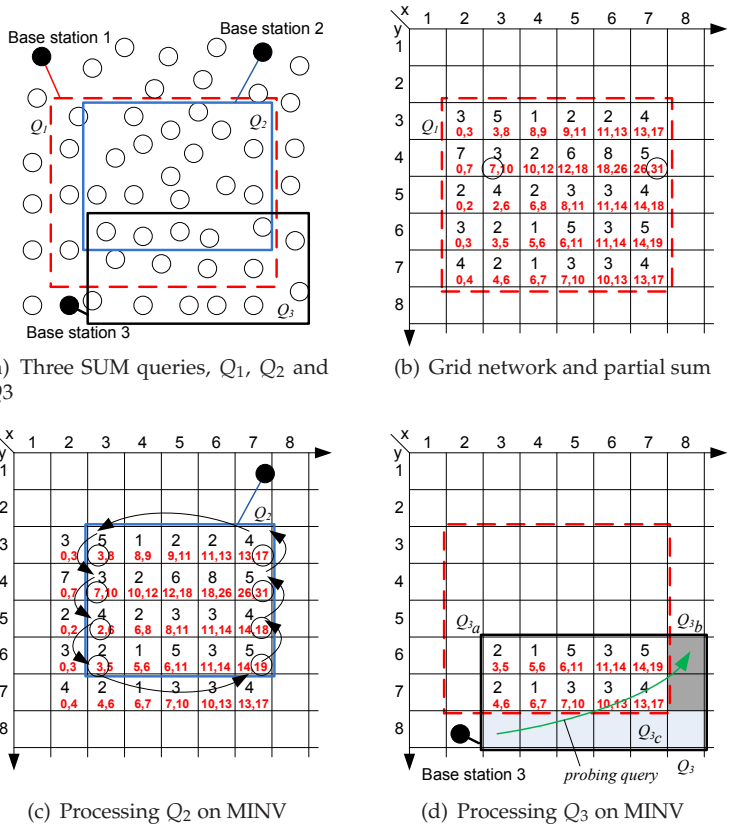
Through simulation, our developed itinerary-based approach is demonstrated outperforming infrastructure-based approaches Xu et al. (2006). Besides, the idea of itinerary-based in-network query processing has also been adopted for other types of queries and applications such as tracking nearest neighbor objects Wu et al. (2007).

4. In-network data caching

Data caching is widely used in distributed computer systems to shorten remote data access latency. In sensor networks, data caching has one more important benefit that is saving communication energy cost. Many existing research works focused on strategies of replicating frequently accessed sensed readings in some sensor nodes closer to base stations Ganesan et al. (2003); Liu et al. (2004); Ratnasamy et al. (2002); Sadagopan et al. (2003); Shakkottai (2004); Zhang et al. (2007). In presence of multiple base stations, a research problem of finding sensor nodes for caching sensed readings is formulated as determining a Steiner tree in a sensor network Prabh & Abdelzaher (2005). In a graph, a Steiner tree is a subgraph connecting all specified vertices and providing the smallest sum of edge distances Invanov & Tuzhilin (1994). By caching data in some sensor nodes as internal vertices (that connect more than one edge) in a Steiner tree, the communication costs between those sensor nodes providing sensed readings and base stations are guaranteed to be minimized.

On the other hand, existing data caching schemes do not support data aggregation. Accordingly, we have devised a new data caching scheme called *materialized in-network view* (MINV) to support SUM, AVERAGE, COUNT, VARIANCE aggregate functions Lee et al. (2007). Specifically, MINV maintains partially computed aggregated readings in some queried sensor nodes. Then, subsequent queries, which are issued by different base stations and which cover queried sensor nodes, can be fully or partially answered by cached results.

Figure 3(a) shows a motivating example of MINV. In the figure, a SUM query Q_1 adds up the sensed readings of all sensor nodes in a query region at time t_1. At a later times t_2 and t_3, two other SUM queries, Q_2 and Q_3, respectively, are issued to summarize readings from sensor nodes in two other queried regions overlapping Q_1's. Without cache, all queries are processed independently. Ideally, if Q_1's answer can be maintained and made accessible, Q_2 and Q_3 can be answered by some cached data to save the energy costs of an entire sensor network.



(a) Three SUM queries, Q_1, Q_2 and $Q3$

(b) Grid network and partial sum

(c) Processing Q_2 on MINV

(d) Processing Q_3 on MINV

Fig. 3. Materialized in-network view

On the other hand, two major issues are faced in the development of MINV. The first and most critical issue is the presentation and placement of queried results. This directly affects the usability of cached data for any subsequent query. Another issue is about how a query can be processed if its answer is partially or fully available from the cache.

In MINV, we consider a sensed area structured into a grid as shown in Figure 3(b), as opposed to building any ad hoc routing structure that favors queries issued by some base stations at query time. Within every grid cell denoted by $cell(x,y)$, sensor nodes form a cluster and one of the sensor nodes is elected as a cluster head. Upon receiving a query, the cluster head collects sensed readings from all cluster members. Based on this setting, we can treat a sensor network as a grid of cluster heads. To answer aggregation queries, we assume parallel itinerary-based data aggregation as discussed in the previous section. Here, cluster heads serve as Q-nodes, forwarding queries and computing intermediate results. Additional to query processing, cluster heads cache every intermediate query result it receives and that it send. For grid cell $cell(x,y)$, we denote the received intermediate query result as $init(x,y)$ and the sent intermediate query result as $final(x,y)$. As shown in Figure 3(b), intermediate results derived and maintained for a SUM query (called *partial sum*) are accumulated and cached

in cluster heads within queried regions. In the figure, cluster head at $cell(3,4)$ maintains an initial partial sum (i.e., $init(3,4)$) and a final partial sum (i.e., $final(3,4)$) as 7 and 10, respectively, while its local reading is 3. Based on cached partial sums, the sum of sensed readings in all cell between $cell(x,y)$ and $cell(x',y)$ in the same row y can be determined as $final(x',y) - init(x,y)$. As in the figure, the sum of sensed readings of sensor nodes from $cell(3,4)$ through $cell(7,4)$ can be calculated as $31 - 7 = 24$.

To answer another SUM query Q_2 whose region is fully covered by Q_1's, Q_2 can simply traverse the border of its query region to collect cached partial sums. In Figure 3(c), Q_2 sums up $init(3,3)$, $init(4,3)$, $init(5,3)$ and $init(6,3)$, i.e., $3 + 7 + 2 + 3 = 15$, from the left side of its query region. Thereafter, it calculates the sum of $final(6,7)$, $final(5,3)$, $final(4,3)$ and $final(3,7)$, i.e., $19 + 18 + 31 + 17 = 85$ from the right side of the region, and subtracts 15 from it. Now the final sum is 70. Notice that only cluster heads on the border of a query region are accessed for cached partial sums and participate in query passing. By using the cache, messages between cluster heads and their members are saved. Besides, some internal grid cells inside a given query region are not accessed at all, further reducing energy costs.

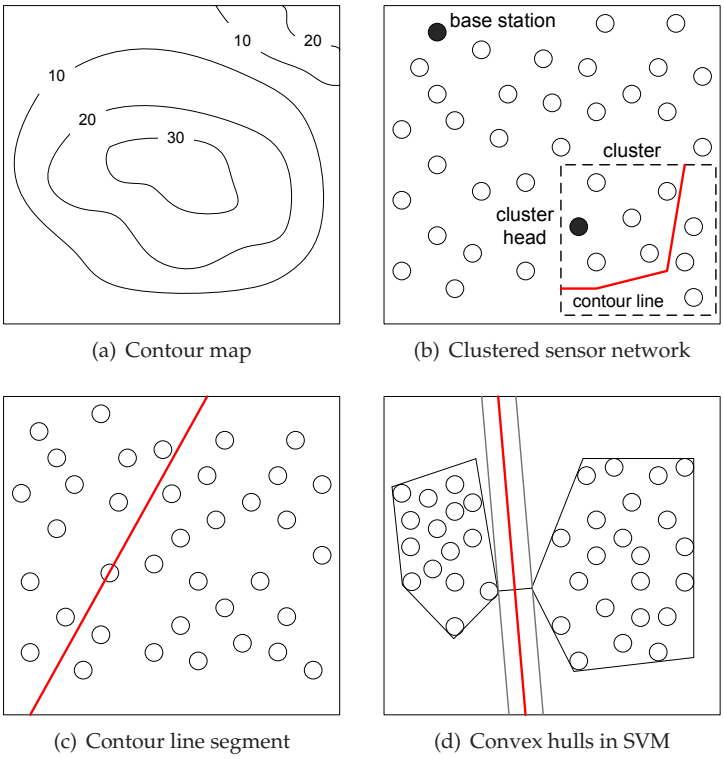
Some queries may have their query regions partially covered by previous queries. In these cases, those queries need to be decomposed into subqueries, which each subquery covers one disjointed subregion. The final query result is then computed by aggregating those subquery results. For instance, Q_3's region is partially covered by Q_1's. Thus, it is partitioned into three subqueries Q_{3a}, Q_{3b} and Q_{3c} as illustrated in Figure 3(d). While Q_{3a} is totally answered by the cached partial sums, Q_{3b} and Q_{3c} are performed as separate SUM queries. The answer of Q_3 is then obtained by adding the sums from these subqueries.

Thus far, cache information has been implicitly assumed to be available to every base stations in the above discussion. In fact, it is not energy efficient to make cache information available everywhere. In MINV, we consider that the cache information is only maintained with initial and final intermediate results in queried grid cells. In this setting, cache discovery is an issue to consider. To determine whether a cache is available for a query, we introduced a probing stage in every query evaluation as illustrated in Figure 3(d). The main idea of this probing stage is described as follows. When a query reaches the (nearest) corner of a query region, it traverses to the diagonally opposite corner and checks if available cache is present in the traversed cells on a diagonal line. If no cache is discovered, it means two possible implications: (i) no cache is available inside the query region, or (ii) a cache if exists has a small overlapped area with the query region, so that it is considered to be not useful to the query. If no cache is used, the query is executed directly from the farthest corner. Otherwise, the query is transformed into subqueries accessing the cache and deriving aggregated reading values in remaining divided areas. Notice that this additional probing stage introduces a little extra communication cost, compared to evaluating queries directly, which usually derives query results at the farthest corners of query regions and sends the results from there back to base stations. Besides, for some cases like entire query regions fully covered by a cache (e.g., Q_2 as discussed above), probe stages can be omitted.

5. In-network contour map computation

As discussed in the previous two sections, data aggregation was used to compute a single aggregated value representing the measurements for an entire sensed area or a query region. For a large sensed area, certain measurements recorded by sensor nodes, e.g., temperature,

wind speed, etc., should continuously change over the area. Data aggregation cannot effectively represent such spatially varied measurements. Thus, some other presentations, e.g., histogram, contour map, etc., should be used instead. Among those, contour maps are often used to present the approximate spatial distributions of measurements. On a contour map as illustrated in Figure 4(a), an area is divided into regions by some curves called *contour lines* and every contour line is labeled with one value. Thus, on a contour map, all measurements on a contour line labeled with v are equal to v, whereas measurements at some points not on any contour lines can be determined through interpolation according to their straight-line distances to adjacent contour lines.



(a) Contour map (b) Clustered sensor network

(c) Contour line segment (d) Convex hulls in SVM

Fig. 4. Contour map computation

Very recently, the research of contour map computation in sensor networks has started to receive attention Liu & Li (2007); Meng et al. (n.d.); Xue et al. (2006). An earlier work Xue et al. (2006) was proposed to construct a contour map as a grid, in which each grid cell carries an aggregated single value. This grid presentation can facilitate recognition and matching spatial patterns of measurements with respect to some predefined patterns for event detection and phenomenon tracking. However, the grid presentation cannot provide very precise contour maps and it may incur a large communication cost to convey individual grid cell values, especially when grids of very fine granularity are used.

Motivated by the importance of contour map in sensor networks, we have developed a Contour Map Engine (CME) to compute contour map in sensor networks Xu et al. (2008). More precisely, CME computes contour lines, which can be represented by the coefficients of certain curve/line equations, and thus are small to transmit. In a sensor network, every small area is assumed to be monitored by a cluster of sensor nodes as shown in Figure 4(b). Periodically, a cluster head collects sensed readings from all sensor nodes. Based on their spatial locations and reported sensed readings, the cluster head determines a contour line segment for the area and sends it to a base station. Finally, the base station connects all received contour line segments and constructs a contour map.

Logically, a contour line with respect to a given v_c divides a given area into subareas on its two sides as in Figure 4(c). On one side, all sensor nodes provides reading values not greater than v_c, whereas all other sensor nodes on another side have their readings not smaller than v_c. Here, some sensor nodes reporting their sensed readings of v_c may be distributed around the contour line. Further, given the reading values and locations of individual sensor nodes, partitioning an area by a contour line segment is somewhat equivalent to a binary classification problem. In light of this, the design of CME uses support vector machine (SVM) Christianini & Shawe-Taylor (2000), a commonly used data mining technique, to determines contour line segments. In a cluster of sensor nodes N', each sensor node n ($\in N'$) provides its location x_n and its classified value y_n, which can be either -1 or $+1$, according to its own sensed reading v_n and the contour line value v_c. Here, $y_n = \begin{cases} +1 & v_n \geq v_c \\ -1 & v_n < v_c \end{cases}$. Next, we define the classification boundary (i.e., the contour line segment) as a hyperplane by a pair of coefficients (w, b) such that $w^T x + b = 0$. Based on this, we can estimate an expected $\hat{y}$ for any location x, which may not have any sensor node as

$$\hat{y} = sgn(w^T x + b) = \begin{cases} +1 & w^T x + b \geq 0 \\ -1 & w^T x + b < 0 \end{cases}$$

Now, the classification boundary in SVM is derived to maximize the margin between the convex hull of the two sets, such that classification error for unknown locations can be minimized as depicted in Figure 4(d). The distance between any location x and the classification boundary is $\frac{|w^T x|}{||w||}$. The optimal classification boundary is derived by maximizing the margin, which can be written with Largrange multipliers α_n below:

$$\max_{\alpha} W(\alpha) = \sum_{n \in N'} \alpha_n - \frac{1}{2} \sum_{n \in N'} \sum_{m \in N'} \alpha_n \alpha_m y_n y_m x_n^T x_m$$

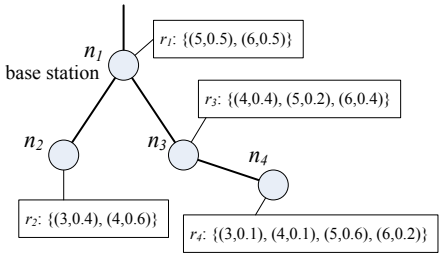
subject to $\alpha_n > 0$ and $\sum_{n \in N'} \alpha_n y_n = 0$. Finally, $\max_{\alpha} W(\alpha)$ can be solved by traditional quadratic optimization.

Thus far, our discussion has assumed a single linear contour line segment formed. To handle non-linear classification, our CME utilizes space transformation to divide sensor nodes in a sub-cluster, according to some sample training data. Then, contour line segments are derived from individual sub-clusters. Interested readers can be refer the details in Xu et al. (2008). Some other recent works (e.g., Zhou et al. (2009)) have been presented in the literature to improve the precision of contour line segments by using more sophnicated techniques.

6. In-network probabilistic data aggregation

Sensor reading values are inherently noisy and somewhat uncertain, because of possible inaccurate sensing, environmental noise, hardware defeats, etc., Thus, data uncertainty is another important issue in sensor data analysis. In the literature, uncertain data management has been extensively studied and various models are developed to provide the semantics of underlying data and queries Faradjian et al. (2002); Prabhakar & Cheng (2009). However, existing works adopts centralized approaches Faradjian et al. (2002); Prabhakar & Cheng (2009) that, however, is energy inefficient as already discussed. In-network uncertain data aggregation appears to be new research direction.

Very recently, we have started to investigate a variety of in-network data aggregation techniques for some common aggregation queries. In the following, we discuss one of our recent works on probabilistic minimum value query (PMVQ) Ye, Lee, Lee, Liu & Chen (to appear). A probability minimum value query searches for possible minimum sensed reading value(s).



(a) An example sensor network of four sensor nodes

Value v	$Pr[v_{min} = v]$
3	0.46
4	0.54
5	0.00
6	0.00

(b) Minimum value probability

Fig. 5. Example sensor network and minimum value probability

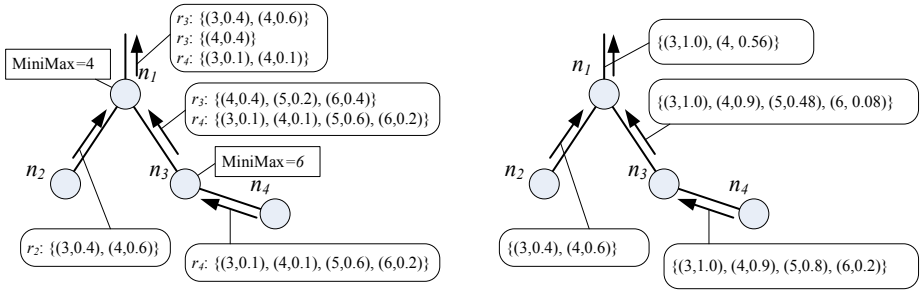
Figure 5(a) shows an example sensor network of four sensor nodes. Each sensor node n_i maintains a probabilistic sensed reading r_i, i.e., a set of possible values $\{v_{i,1}, \cdots v_{i,|r_i|}\}$. Each value $v_{i,k}$ is associated with a non-zero probability $p_{i,k}$ being a real sensed reading value. The sum of all $p_{i,k}$ ($1 \leq k \leq |r_i|$) equals 1. The sensed reading r_i of each example sensor node n_i is shown next to the node. For n_1, the actual sensed reading value may be either 5 with a probability of 0.5 or 6 with the same probability. Since every sensed reading has different possible values, it is apparently not trivial to say that 3, which is the smallest possible value among all, is the minimum since it may not actually exist. On the other hand, 4 can be the true minimum when 3 is not real. As such, more than one value can be the minimum value, simultaneously. Thus, the minimum value probability for v being the minimum v_{min} among all possible sensed reading values, denoted by $Pr[v_{min} = v]$, is introduced and defined as below:

$$Pr[v_{min} = v] = \prod_{n_i \in N} Pr[r_i \geq v] - \prod_{n_i \in N} Pr[r_i > v]$$

In our example, $Pr[v_{min} = 3]$ is equal to $(1 \cdot 1 \cdot 1 \cdot 1) - (1 \cdot 0.6 \cdot 1 \cdot 0.9) = 0.46$, $Pr[v_{min} = 4]$ is equal to $(1 \cdot 0.6 \cdot 1 \cdot 0.9) - (1 \cdot 0 \cdot 0.6 \cdot 0.8) = 0.54$, and both $Pr[v_{min} = 5]$ and $Pr[v_{min} = 6]$ are 0, as listed in Figure 5(b). Hence, the minimum value query result include 3 and 4 and their minimum value probabilities are greater than 0.

To evaluate PMVQ in sensor networks, we have devised two algorithms, namely, *Minimum Value Screening (MVS) algorithm* and *Minimum Value Aggregation (MVA) algorithm*. Both of the algorithms evaluate PMVQs in sensor networks organized as routing trees. We describe them in the following.

MVS Algorithm. Suppose that there are two probabilistic sensed readings r_i and r_j from two sensor nodes n_i and n_j, where $r_i = \{v_{i,1}, \cdots v_{i,|r_i|}\}$ and $r_j = \{v_{j,1}, \cdots v_{j,|r_i|}\}$. A value v_j ($\in r_j$) is certainly not the minimum if r_i has all its values smaller than it, i.e., $\forall_{v_i \in r_i} v_i < v_j$. Then, v_j can be safely discarded. Based on this idea, we introduced a notion called MiniMax. Among sensed readings from a subset of sensor nodes N', a MiniMax denoted by $\mathsf{MiniMax}(N')$ represents the largest possible value, formally, $\mathsf{MiniMax}(N') = \min_{n_i \in N'} \left\{ \max_{v_i \in r_i} \{v_i\} \right\}$.



(a) MVS algorithm (b) MVA algorithm

Fig. 6. MVS and MVA algorithms

This MiniMax notion is used to screen out those values that should not be minimum values. We use Figure 6(a) to illustrate how MiniMax is determined and used by MVS algorithm to eliminate some values and their probabilities from being propagated in a routing tree. First, n_4 sends its sending reading values to n_3, which in turn deduces $\mathsf{MiniMax}(\{n_3, n_4\})$, i.e., 6. Thus, n_3 propagates all its and n_4's sensed reading values to n_1. On the other hand, n_2 submits its sensed reading values to n_1. Now, n_1, i.e., the base station, determines $\mathsf{MiniMax}(\{n_1, n_2, n_3, n_4\})$, which equals 4. Thus, only n_2's $\{(3, 0.4), (4, 0.6)\}$, n_3's $\{(4, 0.4\}$ and n_4's $\{(3, 0.1), (4, 0.4)\}$ are further propagated to the connected terminal. Later, it determines the final result values according to their minimum value probabilities.

MVA Algorithm. MVA algorithm computes $Pr[v_{min} = v]$ for each candidate value v incrementally during data propagation since computation of $Pr[v_{min} = v]$ is decomposable. Recall that $Pr[v_{min} = v]$ is computed based on two terms, i.e., $\prod_{n_i \in N} Pr[r_i \geq v]$ and $\prod_{n_i \in N} Pr[r_i > v]$. These two terms can be factorized when N is divided into x disjointed subsets, i.e., $N_1, N_2, \cdots N_x$ as follows:

$$\prod_{n_i \in N} Pr[r_i \geq v] = \prod_{i \in [1,x]} \prod_{n_i \in N_i} Pr[r_i \geq v], \qquad \prod_{n_i \in N} Pr[r_i > v] = \prod_{i \in [1,x]} \prod_{n_i \in N_i} Pr[r_i > v]$$

Based on this, in any subtree covering some sensor nodes N_i, the root can calculate $\prod_{n_i \in N_i} Pr[r_i \geq v]$ and $\prod_{n_i \in N_i} Pr[r_i > v]$ for every value v. Then, only the value and these

two terms are sent to its parent instead of all individual sensed reading values as needed by MVS algorithm.

Further, due to the fact that $Pr[v_{min} = v]$ should be zero whenever $\prod_{n_i \in N_i} Pr[r_i \geq v] = \prod_{n_i \in N_i} Pr[r_i > v]$ for any non-empty N_i, it is safe to omit value v from being propagated. In addition, for integer sensed reading values, $\prod_{n_i \in N_i} Pr[v_{min} > v]$ should be equal to $\prod_{n_i \in N_i} Pr[v_{min} \geq v + 1]$. Therefore, either $\prod_{n_i \in N_i} Pr[v_{min} > v]$ or $\prod_{n_i \in N_i} Pr[v_{min} \geq v + 1]$ can be sent to a parent node and the omitted probabilities can be deduced by the parent node.

Figure 6(b) illustrates MVA algorithm. First, n_4 sends each of its value v and $Pr[v_{min} \geq v]$, i.e., $(3, 1.0)$, $(4, 0.9)$, $(5, 0.8)$, $(6, 0.2)$ to n_3. Similarly, n_2 sends $(3, 1.0)$ and $(4, 0.6)$ to n_1. Then, n_3 calculates $Pr[v_{min} \geq v]$ for all its know values, i.e., 3, 4, 5 and 6. Next, n_3 forwards $(3, 1.0)$, $(4, 0.9)$, $(5, 0.48)$ and $(6, 0.8)$ to n_1. Further, n_1 computes $Pr[v_{min} = v]$ as n_3. However, $Pr[v_{min} = 5]$ and $Pr[v_{min} = 6]$ are both 0, so 5 and 6 are filtered out. At last, n_1's $Pr[v_{min} = 3]$ and $Pr[v_{min} = 4]$ are determined and they are equal to zero; and both 3 and 4 are the query result.

Compared with MVS algorithm, MVA algorithm considerably saves communication costs and battery energy. Through detailed cost analysis and simulation experiments as in Ye, Lee, Lee, Liu & Chen (to appear), MVA algorithm provides costs linear to the number of sensor nodes, while MVS incurs significantly large communication costs with respect to the increased number of sensor nodes.

In addition to probabilistic minimum query, we have also investigated other probabilistic queries in sensor networks, e.g., probabilistic minimum node query (PMNQ) Ye, Lee, Lee, Liu & Chen (to appear) that searches for sensor nodes that provide probabilistic minimum values and probabilistic top-k value query that search for k smallest (or largest) values Ye, Lee, Lee & Liu (to appear).

7. Summary and future directions

Wireless sensor networks are important tools for many fields and applications. Meanwhile, in sensor networks, data acquisition that collects data from individual sensor nodes for analysis is one of the essential activities. However, because of scarce sensor node battery energy, energy efficiency becomes a critical issue for the length of sensor network operational life. Over those years, many research works have studied various in-network query processing as one of the remedies to precious precious sensor node energy. By in-network query processing, queries are disseminated and processed by sensor nodes and a small volume of (derived) data is collected and transmitted rather than raw sensed readings over costly wireless communication. Subject to the supported types of queries and potential optimizations, a variety of in-network query processing techniques have been investigated and reported in the literature.

This chapter is devoted to review representative works in in-network data aggregation, data caching, contour map computation and probabilistic data aggregation. With respect to those areas, we also discussed our recent research results, namely, itinerary-based data aggregation, materialized in-network view, contour mapping engine and probabilistic minimum value search. Itinerary-based data aggregation navigates a query among sensor nodes in a queried region for an aggregated value. Compared with infrastructure-based approaches, it incurs fewer rounds of messages and can easily deal with sensor node failure in the

course of query processing. To boost the performance of multi-queries issued from different base stations, materialized in-network views provide partial results for previous queries to subsequent aggregation queries. It is different from existing works that cache sensed readings independently and that cannot directly support data aggregation. Contour mapping engine adopts data mining techniques to determine contour line segments in sensor networks, whereas some other works relies on centralized processing or provide less accurate contour maps. Last but not least, probabilistic minimum value search is the initial research result on uncertain sensed data aggregation. As sensed reading values are mostly imprecise, handling and querying probabilistic sensor data is currently an important on-going research direction.

In addition, recent research studies have shown uneven energy consumption of sensor nodes that sensor nodes in some hotspot regions have more energy consumed than others Perillo et al. (2005). Such hotspot problems are currently studied from the networking side. Besides, heterogeneous sensor nodes are going to be very common in sensor networks. Thus, we anticipate that future in-network query processing techniques should be able to handle uneven energy consumption and to make use of super sensor nodes, while many existing works mainly presume homogeneous sensor nodes and consider even energy consumption.

8. References

Biswas, R., Thrun, S. & Guibas, L. J. (2004). A Probabilistic Approach to Inference with Limited Information in Sensor Networks, *Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN), Berkeley, CA, Apr 26-27*, pp. 269–276.

Bojkovic, Z. & Bakmaz, B. (2008). A Survey on Wireless Sensor Networks Deployment, *WSEAS Transactions on Communications* 7(12).

Christianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press.

Doherty, L. & Pister, K. S. J. (2004). Scattered Data Selection for Dense Sensor Networks, *Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN), Berkeley, CA, Apr 26-27*, pp. 369–378.

Faradjian, A., Gehrke, J. & Bonnet, P. (2002). GADT: A Probability Space ADT for Representing and Querying the Physical World, *Proceedings of the 18th IEEE International Conference on Data Engineering (ICDE), San Jose, CA, Feb 26 - Mar 1*, pp. 201–211.

Faulkner, M., Olson, M., Chandy, R., Krause, J., Chandy, K. M. & Krause, A. (2011). The Next Big One: Detecting Earthquakes and Other Rare Events from Community-Based Sensors, *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN), Chicago, IL, Apr 12-14*, pp. 13–24.

Ganesan, D., Estrin, D. & Heidemann, J. S. (2003). Dimensions: Why Do We Need a New Data Handling Architecture for Sensor Networks?, *Computer Communication Review* 33(1): 143–148.

Group, T. W. (n.d.). TinyOS, http://www.tinyos.net/.

Hellerstein, J. M., Hong, W., Madden, S. & Stanek, K. (2003). Beyond Average: Toward Sophisticated Sensing with Queries, *Proceedings of Information Processing in Sensor Networks, Second International Workshop (IPSN), Palo Alto, CA, Apr 22-23*, pp. 63–79.

Huang, Z., Wang, L., Yi, K. & Liu, Y. (2011). Sampling Based Algorithms for Quantile Computation in Sensor Networks, *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), Athens, Greece, Jun 12-16*, pp. 745–756.

Invanov, A. O. & Tuzhilin, A. A. (1994). *Minimal Networks: The Steiner Problem and Its Generalizations*, CRC Press.

Lee, K. C. K., Zheng, B., Lee, W.-C. & Winter, J. (2007). Materialized In-Network View for Spatial Aggregation Queries in Wireless Sensor Network, *ISPRS Journal of Photogrammetry and Remore Sensing* 62: 382–402.

Li, Z., Zhu, Y., Zhu, H. & Li, M. (2011). Compressive Sensing Approach to Urban Traffic Sensing, *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS), Minneapolis, MN, Jun 20-24*, pp. 889–898.

Liu, X., Huang, Q. & Zhang, Y. (2004). Combs, Needles, Haystacks: Balancing Push and Pull for Discovery in Large-Scale Sensor Networks, *Proceedings of the 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys), Baltimore, MD, Nov 3-5*, pp. 122–133.

Liu, Y. & Li, M. (2007). Iso-Map: Energy-Efficient Contour Mapping in Wireless Sensor Networks, *Proceedings of the 27th IEEE International Conference on Distributed Computing Systems (ICDCS), Toronto, Ontario, Canada, Jun 25-29*, p. 36.

Madden, S. & Franklin, M. J. (2002). Fjording the Stream: An Architecture for Queries Over Streaming Sensor Data, *Proceedings of the 18th IEEE International Conference on Data Engineering, San Jose, CA, Feb 26 - Mar 1*, pp. 555–566.

Madden, S., Franklin, M. J., Hellerstein, J. M. & Hong, W. (2002). TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks, *Proceedings of The 5th USENIX Symposium on Operating System Design and Implementation (OSDI), Boston, MA, Dec 9-11*.

Manjhi, A., Nath, S. & Gibbons, P. B. (2005). Tributaries and Deltas: Efficient and Robust Aggregation in Sensor Network Streams, *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), Baltimore, MD, Jun 14-16*, pp. 287–298.

Meng, X., Nandagopal, T., Li, L. & Lu, S. (n.d.). Contour Maps: Monitoring and Diagnosis in Sensor Networks, 50(15): 2920–2838.

Musaloiu-Elefteri, R., Liang, C.-J. M. & Terzis, A. (2008). Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks, *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN), St. Louis, MO, Apr 22-24*, pp. 421–432.

Perillo, M. A., Cheng, Z. & Heinzelman, W. B. (2005). An Analysis of Strategies for Mitigating the Sensor Network Hot Spot Problem, *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous), San Diego, Jul 17-21*, pp. 474–478.

Prabh, S. & Abdelzaher, T. F. (2005). Energy-Conserving Data Cache Placement in Sensor Networks, *ACM Transactions on Sensor Networks* 1(2): 178–203.

Prabhakar, S. & Cheng, R. (2009). Data Uncertainty Management in Sensor Networks, *Encyclopedia of Database Systems*, pp. 647–651.

Ratnasamy, S., Karp, B., Yin, L., Yu, F., Estrin, D., Govindan, R. & Shenker, S. (2002). GHT: a Geographic Hash Table for Data-Centric Storage, *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Atlanta, GA, Sept 28*, pp. 78–87.

Sadagopan, N., Krishnamachari, B. & Helmy, A. (2003). The ACQUIRE Mechanism for Efficient Querying in Sensor Networks, *IEEE International Workshop on Sensor Network Protocols and Applications (SNPA'03), held in conjunction with the IEEE International Conference on Communications (ICC), Anchorage, AL*.

Shakkottai, S. (2004). Asymptotics of Query Strategies over a Sensor Network, *Proceedings of The 23rd IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Hong Kong, China, Mar 7-11*.

Silberstein, A., Braynard, R., Ellis, C. S., Munagala, K. & Yang, J. (2006). A Sampling-Based Approach to Optimizing Top-k Queries in Sensor Networks, *Proceedings of the 22nd International Conference on Data Engineering (ICDE), Atlanta, GA, Apr 3-8*, p. 68.

Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M. & Corke, P. I. (2005). Data Collection, Storage, and Retrieval with an Underwater Sensor Network, *Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems (SenSys), San Diego, CA, Nov 2-4*, pp. 154–165.

Wu, S.-H., Chuang, K.-T., Chen, C.-M. & Chen, M.-S. (2007). DIKNN: An Itinerary-based KNN Query Processing Algorithm for Mobile Sensor Networks, *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE), Istanbul, Turkey, Apr 15-20*, pp. 456–465.

Xu, Y., , Lee, W.-C. & Mitchell, G. (2008). CME: A Contour Mapping Engine in Wireless Sensor Networks, *The 28th International Conferences on Distributed Computing Systems (ICDCS), Beijing, China, Jun 17-20*, pp. 133–140.

Xu, Y., Lee, W.-C. & Xu, J. (2007). Analysis of A Loss-Resilient Proactive Data Transmission Protocol in Wireless Sensor Networks, *Proceedings of 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM), Anchorage, AL, May 6-12*, pp. 1712–1720.

Xu, Y., Lee, W.-C., Xu, J. & Mitchell, G. (2006). Processing Window Queries in Wireless Sensor Networks, *Proceedings of the 22nd International Conference on Data Engineering (ICDE), Atlanta, GA, Apr 3-8*, p. 70.

Xue, W., Luo, Q., Chen, L. & Liu, Y. (2006). Contour Map Matching for Event Detection in Sensor Networks, *Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, IL, Jun 27-29*, pp. 145–156.

Yao, Y. & Gehrke, J. (2003). Query Processing in Sensor Networks, *Online Proceedings of The First Biennial Conference on Innovative Data Systems Research (CIDR), Asilomar, CA, Jan 5-8*.

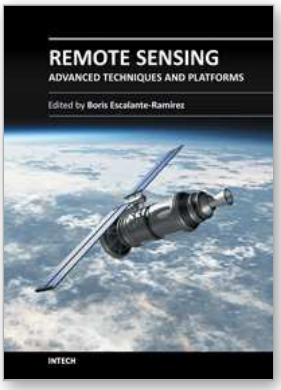
Ye, M., Lee, K. C. K., Lee, W.-C., Liu, X. & Chen, M. C. (to appear). Querying Uncertain Minimum in Wireless Sensor Networks, *IEEE Transactions on Knowledge and Data Engineering* .

Ye, M., Lee, W.-C., Lee, D. L. & Liu, X. (to appear). Distributed Processing of Probabilistic Top-k Queries in Wireless Sensor Networks, *IEEE Transactions on Knowledge and Data Engineering* .

Zhang, W., Cao, G. & Porta, T. L. (2007). Data Dissemination with Ring-Based Index for Wireless Sensor Networks, *IEEE Transactions on Mobile Computing* 6(7): 832–847.

Zhou, Y., Xiong, J., Lyu, M. R., Liu, J. & Ng, K.-W. (2009). Energy-Efficient On-Demand Active Contour Service for Sensor Networks, *Proceedings of IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS), Macau, China, Oct 12-15*, pp. 383–392.

Zurich, T. W. R. G. . E. (n.d.). The Sensor Network Museum, http://www.snm.ethz.ch/Main/HomePage.



Remote Sensing - Advanced Techniques and Platforms
Edited by Dr. Boris Escalante

ISBN 978-953-51-0652-4
Hard cover, 462 pages
Publisher InTech
Published online 13, June, 2012
Published in print edition June, 2012

This dual conception of remote sensing brought us to the idea of preparing two different books; in addition to the first book which displays recent advances in remote sensing applications, this book is devoted to new techniques for data processing, sensors and platforms. We do not intend this book to cover all aspects of remote sensing techniques and platforms, since it would be an impossible task for a single volume. Instead, we have collected a number of high-quality, original and representative contributions in those areas.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ken C.K. Lee, Mao Ye and Wang-Chien Lee (2012). Energy Efficient Data Acquistion in Wireless Sensor Network, Remote Sensing - Advanced Techniques and Platforms, Dr. Boris Escalante (Ed.), ISBN: 978-953-51-0652-4, InTech, Available from: http://www.intechopen.com/books/remote-sensing-advanced-techniques-and-platforms/data-acquistion-in-wireless-sensor-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821