A Scheme for Efficient Tracking of Dynamic Event Region in Wireless Sensor Networks*

Yang YANG Satoshi FUJITA†
Graduate School of Engineering, Hiroshima University‡

Abstract

In this paper, we propose a distributed scheme to recognize the shape of a dynamic event region in wireless sensor networks; i.e., a scheme to track the shape of a part of a given field concerned with a dynamic event such as gas leakage and a fire. The basic idea of our proposed scheme is to identify several critical points in a given event region, and to periodically check the criticalness of such points. The performance of the scheme is experimentally evaluated by simulation. The result of simulations indicates that the proposed event tracking scheme correctly recognizes the move of an event region with sufficiently small number of transmitted messages compared with a centralized tracking scheme.

1 Introduction

According to the recent advancement of microelectronics and wireless communication technologies, **wireless sensor networks** (WSNs) have attracted considerable attentions in the fields of network computing and distributed processing [1, 6, 12]. The primary task of a WSN is to continuously monitor the surrounding environment (e.g., the temperature and the density of NO_x in the atmosphere) in order to notify the change of the status to an appropriate data aggregation point such as meteorological observator, in either an event driven or a query-based fashion.

A typical WSN is composed of a large number of tiny devices called **sensor nodes** (or nodes), each of which is capable of conducting a simple arithmetic computation, wireless communication with nearby nodes, and sensing the status of the surrounding environment. In a "multihop" version of WSNs [9, 11], which is the target of the current paper, each (sensor) node plays the role of message routers in addition to the role of a sensing device. Note that in such systems, all nodes should collaborate with each other, in order to report their status to an aggregation point in an efficient and timely manner. A lot of pervasive applications proposed in the literature are based on an automatic deployments of sensor nodes providing a continuous and/or periodic snapshot of an environment, such as habitat monitoring [2], target tracking [13], aquatic observations [4], and surveillance [17].

There are several key issues in designing an available environment monitoring system (EMS) over WSNs. The first issue is how to convey a faithful representation of the signal field to an aggregation point while keeping the amount of utilized resources sufficiently low. In general, an "event" to be monitored may spread over a wide region in the given field. For example, consider the detection of gas leakage in a given field; i.e., let us consider a WSN which is expected to report "which region in the field has a gas concentration exceeding a certain threshold." Once gas leaks out, the system must continuously monitor all points in the field to observe the spread of the gas. In a centralized approach, each node reports its status to the aggregation point as soon as it detects the change of the density of gas concentration. However, such a naive scheme does not scale, since it consumes a large amount of network resources for the communication with the aggregation point, such as CPU time, communication bandwidth, and electric power. This indicates that in order to realize a highly scalable EMS based on WSNs, we have to develop a distributed data aggregation scheme in which nodes covered by an event region autonomously organize themselves and conduct an appropriate local computation to determine an extent of the region before sending a report to the aggregation point. The second issue we have to consider is about the trade-off between time-criticalness and the accuracy of the monitored information. In fact, certain event such as gas leakage and a fire is highly time-critical, while it is generally sufficient to know an approximated direction of the event region. On the other hand, there exist other type of events which are not time critical but requires a precise location information, e.g., plants growth [16] and habitat changes [8]. In such applications, we need a precise location of target events in order to reflect the acquired data to (political) decisions.

Motivated by those considerations, a variety of data aggregation methods for WSNs have been proposed in

*A part of this research was supported by Kayamori Foundation of Informational Science Advancement.

†Corresponding author

‡Department of Information Engineering, Graduate School of Engineering, Hiroshima University, Kagamiyama 1-4-1, Higashi-Hiroshima, 739-8527, JAPAN

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6796
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6796

the literature, to acquire statistical attributes of sensed data, such as min, max, and average [5] as well as robust statistics such as median and quantiles [14]. However, most of those techniques have merely focused on numerical statistics, and did not pay attention to the "geometric shape" of the event region, although such information is generally effective to intuitively grasp an overview of the monitored event. In this paper, we consider a problem of recognizing the shape of a dynamic event region in WSNs. The proposed scheme is an extension of a shape recognition scheme for static event region, which has been proposed in our previous paper [18]. The basic idea of the current scheme is to identify several critical points in a given even region, and to repeatedly check the criticalness of such points (a formal definition of critical points is described later). By flooding an inquiry message from each critical point, nodes on the boundary of the region can certify that whether the point is actually a critical one or not. Thus, if the region moves in the field according to the spread of the event, a node on the boundary can locally check the fact of move, and can notify it to the (former) critical point with necessary information, such as the direction and the distance of the move. After receiving such notifications from the boundary, a (former) critical point hands over the role of critical point to an appropriate node in the field, to realize an efficient tracking of a dynamic event region.

The performance of the scheme is experimentally evaluated by simulation. The result of simulations indicates that: 1) the number of messages transmitted during a shape recognition significantly reduces by applying the scheme compared with a naive centralized scheme; 2) the cost of the scheme is further improved by applying a pruning of redundant critical points; and 3) the proposed event tracking scheme correctly recognizes the move of event region with sufficiently small number of transmitted messages compared with a centralized scheme.

The remainder of this paper is organized as follows. Section 2 outlines related works. A formal model of WSN is given in Section 3. Section 4 describes a scheme for recognizing the shape of a static region, and Section 5 extends it to the recognition of the shape of dynamic event. Results of simulations are shown in Section 6. Finally, Section 7 concludes the paper with future problems.

2 Related Works

Previous works for event shape recognition in WSN can be classified into two categories by their main techniques; i.e., boundary detection schemes and fault-tolerant schemes.

In **boundary detection** schemes, the shape of an event region is recognized by providing a description of the boundary. Chintalapudi et al. [3] proposed a classifier-based edge detection mechanism to generate a linear polynomial representing the boundary of an event, where each node detects an edge of the region by sampling the status of its nearby nodes. Nowak et al. [10] introduced a quadtree structure to efficiently collect detected information to a central node; i.e., it recursively partitions a given space into four subspaces, and an edge detected by a node corresponding to a leaf in the quadtree is collected to the root of the tree, in such a way that the shape of the boundary is recognized by the node corresponding to the root. Unfortunately however, such boundary detection schemes have a serious drawback such that nodes detecting a boundary should form a continuous loop to correctly recognize the shape of the overall region; i.e., it does not allow a missing of nodes on the boundary which significantly loses the robustness of the scheme.

Fault-tolerant schemes focus on bare analog signals received from sensors rather than a binary representation obtained through interpretations. Assuming that event measurements are spatially correlated, those schemes try to distinguish fault sensor measurements; i.e., disambiguate events by exchanging signals among nearby sensors. Krishnamachari et al. [7] proposed a scheme based on a distributed Bayesian method to detect and to correct such faults. However, such techniques cannot be applied to general WSNs since it requests each node to know its precise geographical location through expensive GPS or RF-based beacons. In addition, it requests each node to autonomously identify interesting output signals, in order to recognize the shape of event region merely through a collaboration among sensors.

3 Model

Let V be a set of sensor nodes. Each node in V is capable of sensing the environment via attached sensor devices, communicating with nearby nodes via wireless communication device, and conducting simple computation with a tiny CPU and a small memory. In the following, we assume that each node in V is located on a two-dimensional plane and is associated with a point in a two-dimensional coordinate space. Let $p(u)$ denote the point associated with node u. Note that $p(u)$ is a variable used only in the explanation of algorithms, and we do not allow each node u to refer to its precise location $p(u)$.

For each $u \in V$, let $N(u)$ denote the set of neighbors of u which is defined by the Euclidean distance of the corresponding points; i.e., for any $u, v \in V$, $v \neq u$, $v \in N(u)$ iff $\|p(u) - p(v)\| \leq 1$, where $\|p - q\|$ represents the Euclidean distance between points p and q[1]. In the proposed scheme, we assume that each node u knows its set of neighbors $N(u)$, and that a message transmitted by u is always received by all nodes in $N(u)$ in a single step. In addition, u detects an "event" occurred in the environment if the event region covers point $p(u)$, where

[1]We assume that the transmission radius of wireless communication device is a unit distance, for simplicity.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6796
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6796

event region is a finite region in the two-dimensional space concerned with the event (note that term "finite" means that it has a finite "boundary").

Let $E \stackrel{\text{def}}{=} \{(u, v) \in V \times V : v \in N(u)\}$ be a symmetric relation defined by the set of neighbors. A pair of V and E naturally defines an undirected graph G, where in the following, we assume that G is connected, without loss of generality. Let σ_1, σ_2, and σ_3 be three external nodes called sinks (note that $\sigma_i \notin V$ for each i). Those sinks are used for data aggregation and node localization (detailed procedure for such operations will be described later). Throughout of this paper, we assume that each sink σ_i has at least one neighbor contained in V, and knows its precise location $p(\sigma_i)$. In addition, sink σ_1 is connected with a host via a wired link. Users can interact with the WSN through the host; i.e., by issuing queries to the host and by receiving necessary information through the host. The main task of the host is to conduct a shape recognition (i.e., an approximation of the shape of a given event region) through information received from nodes in the WSN. The other operations are executed by individual sensor nodes in the WSN, in a distributed manner.

4 Recognizing Static Event

In [18], we proposed a shape recognition scheme for "static" event region. In this section, we improve the efficiency of the previous scheme after providing a brief review of the scheme. As will be described later, the improved scheme is used as a basic procedure in an "event tracking" scheme for recognizing the shape of a dynamic event region. The original scheme consists of the following five parts: 1) preprocessing, 2) event detection, 3) distance field construction, 4) identification of critical points, and 5) event region approximation. An outline of the scheme is described below (we omit the first and the fifth step due to the space limitation. Interested reader should consult our previous paper [18]).

4.1 Distance Field Construction

In the following, we assume that each node u stores tuple $\langle d_1(u), d_2(u), d_3(u)\rangle$ to its local memory, where $d_i(u)$ denotes the minimum hop count from u to sink σ_i. (A concrete way to calculate those values is given in [18]). Such tuple is used to approximate the location of each node in the coordinate space, by assuming that the Euclidean distance is well approximated by the hop count. In addition, the i^{th} element of the tuple is used to navigate message transmissions towards sink σ_i through a shortest path; i.e., each node may simply forward a message towards a descent direction of the i^{th} element.

Suppose that node u detects an event with event region R. After letting $r(u) :=$ true, node u notifies the fact to sink σ_1 through a shortest path as described above,

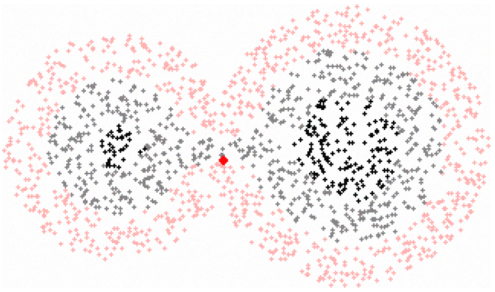

Figure 1. Critical points in an event region. A saddle point is represented as a big red dot, and nodes with larger h-values than the saddle point are denoted by dark colors.

where $r(u)$ is a local variable representing whether u is covered by an event region. Upon receiving a notification, sink σ_1 sends a Reply message along the above forwarding path in the reversed direction. Let u^* be the final receiver of the Reply message. Note that $r(u^*) =$ true must hold since it is the originator of a notification message. In the scheme, even if σ_1 receives several notification messages from different originators, it sends exactly one Reply message to those notifications, assuming that R consists of a single connected region.

Let $h(u)$ denote the **height** of node u with respect to event region R, which is defined as follows:

$$h(u) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } r(u) = \text{false, and} \\ \min_{v \in N(u)}\{h(v)\} + 1 & \text{otherwise.} \end{cases}$$

$h(u)$ represents the minimum hop count from the boundary of R to node u (we say that a node u is at the boundary if $h(u) = 1$). As for a concrete procedure to calculate the height of nodes, the reader should consult our previous paper [18].

4.2 Identification of Critical Points

The basic idea of the scheme is to recognize the shape of an event region through identifying a collection of **critical points** in the distance field. More concretely, we adopt local maxima and saddle points as the critical points characterizing the shape of the event region. An identification of local maxima is easily realized by checking whether the h-value of a node takes the largest value among its neighbors. However, an identification of saddle points is not as easy as an identification of local maxima, since a saddle point should be simultaneously adjacent with a node with higher h-value and a node with lower h-value; i.e., each node cannot decide whether it is a saddle point or not, by merely observing h-values in its neighborhood. Figure 1 illustrates a saddle point in an event region.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6796
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6796

In order to overcome such difficulty of calculating saddle points, we apply a sweep method proposed in [15]. Let U ($\subseteq V$) be a set of identified local maxima, and $G(U)$ be a subgraph of G induced by U. For each connected component in $V(U)$, we select an arbitrary node in the component as an initiator of sweep process. Let U' be the set of initiators. At first, each initiator $u \in U'$ broadcasts a Sweep message containing the name and the height of node u, to all neighbors in $N(u)$. This message is propagated to all nodes covered by the same event region R, by forwarding the message towards a descending direction of the h-value. Note that each node who received a Sweep message knows the name and the height of the corresponding local maximum. If v receives two sweep messages originating from different local maxima, v recognizes itself as a candidate of saddle point, and executes the following operations, in order to certify whether it is an actual saddle point: 1) stop the forwarding of received Sweep to the next node; and 2) broadcast a message containing $h(v)$ to $N(v)$. For any candidate saddle point v, if it does not receive a broadcast message containing an h-value not smaller than $h(v)$ from its neighbor, it can recognize itself as an actual saddle point.

Upon recognizing itself as a critical point, node u transmits an Event message towards sink σ_1 in order to notify the following information to the host: 1) tuple $\langle d_1(u), d_2(u), d_3(u)\rangle$ indicating an approximated location of node u; 2) the height $h(u)$ of node u, and 3) if u is a saddle point, it designates two corresponding local maxima.

4.3 Pruning Unexpected Local Maxima

Although the above scheme certainly identifies several critical points in a fully distributed manner, as shown in [18], it often (mis)identifies several (non-critical) nodes as local maxima if the density of nodes is low. Such a redundancy increases the cost of shape recognition (in particular, it increases the number of unnecessary Event messages). In this subsection, we improve our previous scheme by pruning redundant local maxima. The procedure is executed after an identification of (candidates of) local maxima. Since the cost of such additional operation is much lower than the cost of redundant Event messages, as will be shown in the next section, the resultant scheme certainly reduces the overall communication cost.

In the original scheme, the distance field is constructed in such a way that the height $h(u)$ of node u is determined as i if $r(u) = \mathsf{true}$ and $\min_{v\in N(u)}\{h(v)\} = i - 1$ [18]. However, it does not guarantee that node v always has a neighbor u such that $h(u) = h(v)+1$ when v is not local maximum; i.e., v may (mis)identify itself as a local maximum when each neighbor w of v has the same (or lower) height with v, and when it has the same height with v, it is adjacent with a node with height $h(v) + 1$. Such a (mis)identification could be partially resolved in the following manner: For each node u, if it identifies itself as a local maxima, it transmits a LocalMax message including $h(u)$ to $N(u)$. Upon receiving a LocalMax message, node v transmits a NG message to u, if there exists any $w \in N(v)$ such that $h(w) > h(u)$. If node u receives an NG message, it excludes itself from a set of local maxima.

5 Event Tracking Scheme

5.1 Overview

In this section, we propose a new scheme to track the move of an event region. We assume that the shape of a given region is convex, and does not change during the computation (the support of expansion is a future work). In the proposed scheme, an identified local maximum periodically verifies its maximality, and if it detects that it is no longer a local maximum, it hands over the role of local maximum to an appropriate node in the field. More concretely, it periodically transmits a short message towards nodes at the boundary of the event region, and if a node receiving the message detects the change of the boundary, it notifies the fact to the (former) local maximum.

Key points of such a verification-based scheme are as follows: 1) how to disseminate a verification message towards the boundary of a region and how to collect notification from them; 2) how to detect the change of the boundary in a local manner; and 3) how to estimate the direction and the distance to the new local maximum from the collected notification messages. In the following explanation, we use symbol R to denote the region before change, and R' to denote the region after the change.

Recall that $h(v)$ denotes the height of node v in R. In the following, a new message Height(i, j) is introduced to notify that the height of a node is i in R and is changed to j in R', where j takes one of the following three values: 1) $j = 0$ if the node is no longer a node covered by R'; 2) $j = 1$ if it is a boundary node in R'; and 3) $j = \infty$ if it is inside of R' but is no longer a boundary node in R'.

5.2 Update of the Boundary

Recall that U denotes a set of local maxima which transmit Event messages to sink σ_1, and U' ($\subseteq U$) denotes a set of nodes who initiated a sweep process to identify saddle points. In the following, for simplicity, we assume $U = U'$; i.e., U is an independent set of G.

Propagation: After transmitting an Event message, each $u \in U'$ waits for a certain time τ, and then starts a verification of event region R. More concretely, it transmits an Update message to $N(u)$, which is propagated to nodes covered by R, in a descent direction of h-values; i.e., in a similar way to the sweep message while it is not

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6796
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6796

blocked at saddle points. If it receives an Update message from a neighbor with a positive h-value for the first time, node v transmits a copy of the message to $N(v)$ with Boolean variable $r'(v)$ which indicates whether v is covered by the new event region R'. Note that by collecting all Update messages received from neighbors, v can identify itself as a boundary node or not; i.e., if it receives an Update message with $r'(w) = \mathsf{false}$ from neighbor w and if $r'(v) = \mathsf{true}$, then it identifies itself as a boundary node in R'.

Note that during the propagation of Update messages, each node v knows a neighboring node with higher h-value than v in R. In the proposed scheme, we regard such node as a **parent** of v in the distance field, where a saddle point has several parent nodes.

Collection: After receiving Update messages from all neighbors, node v at either the boundary of R or the boundary of an intersection of R and R' transmits a Height message to $N(v)$, which will be forwarded to the initiator of the Update message along a tree used for the propagation in the reverse direction (note that each node who received an Update message has known its parent in the delivery tree). More concretely, 1) node v transmits message $\mathsf{Height}(h(v), 0)$ if $h(v) \geq 1$ and it becomes an outside node of R', 2) transmits $\mathsf{Height}(h(v), 1)$ if $h(v) \geq 1$ and it becomes a boundary node in R', and 3) transmits $\mathsf{Height}(1, \infty)$ if $h(v) = 1$ and it becomes an interior node in R' besides the boundary.

In order to reduce the communication cost, during such transmissions, several redundant messages are discarded at intermediate nodes, according to the following rules: 1) for any i and j, $\mathsf{Height}(i, j)$ is transmitted at most once, i.e., latter ones are simply discarded; 2) if a node receives both $\mathsf{Height}(i, 0)$ and $\mathsf{Height}(i', 1)$, then the former one is discarded; and 3) if it receives both $\mathsf{Height}(i, 1)$ and $\mathsf{Height}(i', 1)$, $i < i'$, then the former one is discarded.

5.3 Target Tracking

After receiving Height message from all neighbors, local maximum u of event region R conducts the following operations:

Case 1: If it receives $\mathsf{Height}(1, 1)$ from all nodes in $N(u)$, u recognizes $R' = R$, and notifies the fact to the host. It then starts the next verification process after waiting for τ time.

Case 2: If it receives $\mathsf{Height}(i, 1)$ and $\mathsf{Height}(1, \infty)$ from different nodes in $N(u)$, u recognizes that the local maximum corresponding to u moves to a node u' in event region R', and that node u' exists in the direction of a node who sent $\mathsf{Height}(1, \infty)$ to u. Thus, it tries to hand over the role of local maximum to u' according to the procedure described below.

Case 3: Otherwise, node u gives up to find its successor in R, and asks the host to start a new shape recognition process.

Detailed procedure for Case 2 is described as follows. Let i^* be the maximum value of i contained in messages $\mathsf{Height}(i, 1)$ which are received by u from its neighbors. In the proposed event tracking scheme, we use $i^* - 1$ as the hop count from u to u', where u' is the successor of u in R'. The direction of u' from u is determined by constructing a distance field from terminals of an "arc of nodes" who transmitted a $\mathsf{Height}(1, \infty)$ message towards node u. More concretely, a node who transmitted $\mathsf{Height}(1, \infty)$ is regarded as a terminal of the arc, if it receives $\mathsf{Height}(1, 1)$ message from its neighbor in the collection phase. If there are several nodes satisfying the above condition in its neighbor, a node with a largest ID is selected as the terminal node. Since we are assuming that R is convex, there exist at most two terminals on the arc (if it can identify no terminal, node u gives up to find its successor, and asks the host to start the next shape recognition process as in Case 3). Let w_1 and w_2 be two terminals of the arc. After transmitting their $\mathsf{Height}(1, \infty)$ messages, those nodes initiate a construction of a distance field by transmitting a short message, which will be forwarded by nodes who have already forwarded a $\mathsf{Height}(1, \infty)$ message, towards node u.

The direction of successor u' from u is identified by nodes which have the same hop count to two terminal nodes. Thus, by forwarding a hand-over message along a path consisting of such nodes for $i^* - 1$ hops, node u can successfully send a hand-over message to a candidate node of local maximum in the new event region R'. After receiving a hand-over message, node u' constructs a distance field originating from it, and starts a verification of the maximality in a similar way to the above procedure.

6 Simulation

6.1 Overview

We evaluate the performance of the proposed scheme by simulation. In the simulation, we assume that any message transmitted by node u is correctly received by all nodes in $N(u)$. In addition, to simplify the exposition, we assume that the host can estimate the correct location of each critical point from the tuple received from it (several figures estimated by the host will be shown below). A square region of size 20×20 is given as the field of events, where a unit distance is defined to be equal to the transmission radius of each node. Coordinate points of four corners of the region are $(0, 0)$, $(0, 20)$, $(20, 20)$, and $(20, 0)$ in a clockwise direction starting from the left bottom. Given such field, we select n random points in the field, and associate them to individual sensor nodes (note that each point is represented by a pair of reals and we assume that any two points are distinct, without loss of generality). Parameter n is appropriately determined in the simulation in such a way that the resultant graph is connected (i.e., too small n disconnects the underlying sensor network).

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6796
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6796

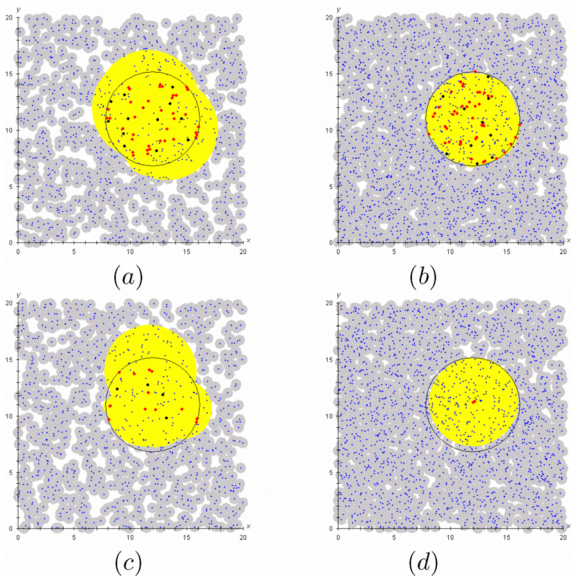

Figure 2. The shape of event region R_1 estimated by the proposed scheme.

In this section, we consider a simple circle with radius 4.2 centered at point $(12, 11)$ as an event region, and call it R_1. In the following figures, the boundary of an event region is represented by a black solid line and each node is represented by a gray circle of radius 0.5. Thus, the reader can easily check the connectivity of the resultant graph G since two nodes are connected by an edge in G if the corresponding circles have a non-empty intersection.

6.2 Accuracy of Shape Recognition

Yellow region shown in Figure 2 are outputs of the proposed scheme for event region R_1. The red dots in Figure 2 indicate local maxima calculated by the scheme, and black dots indicate saddle points each of which is identified by two sets of connected local maxima through sweep process. Figure 2 (a) and (c) are outputs for $n = 900$, and Figure 2 (b) and (d) are outputs for $n = 1800$. In the former case, each node has six neighbors in average, and in the latter case, each node has 13 neighbors in average. The difference between upper figures and lower figures is whether the pruning of redundant local maxima is conducted or not. It is immediate from the figures that the accuracy of approximation increases as increasing the density of the underlying WSN, and the pruning operation certainly reduces the number of redundant local maxima while keeping the accuracy of the shape recognition.

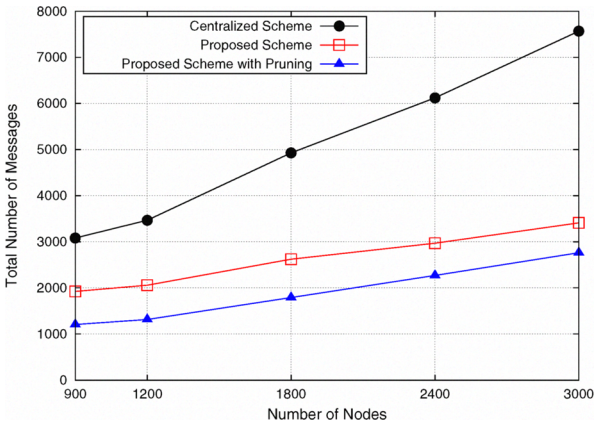

Figure 3. A comparison of the total number of messages transmitted during shape recognition procedure of centralized scheme, proposed scheme and the improvement.

6.3 Efficiency of Shape Recognition

Next, we evaluate the efficiency of our shape recognition scheme in terms of the number of transmitted messages. Each data described below is an average over 10 experiments (each instance is randomly generated by selecting n random points in the given field, while the location of sinks is fixed). Recall that in our scheme, a shape recognition is initiated by a node detecting an event by transmitting a notification which will be forwarded to sink σ_1, and terminates when σ_1 receives all Event messages concerned with the event. In the following, we assume that sink σ_1 is placed at point $(0, 20)$.

Figure 3 compares the result on three schemes, i.e., the first one is a centralized scheme, the second one is our original scheme, and the third one is our improved (i.e., pruning) scheme. As expected, the second scheme certainly reduces the number of messages of the first scheme, and the third scheme further improves the second scheme. A detailed analysis of the simulation result indicates that the number of several messages does not change by such improvement; for example, in both of the second and the third schemes, as increasing n from 900 to 3000, 1) the number of notifications similarly increases from 203.7 to 842.4; 2) the number of Reply messages takes a constant value 22.6; 3) the number of Inside and Sweep messages increases from 127.7 to 416.6; and 4) the number of Height messages similarly increases from 193.8 to 472.3. However, it certainly reduces the number of Event messages transmitted in the second scheme; e.g., it reduces from 1246.7 to 443.6 when $n = 900$, and it reduces from 1244.5 to 173.6 when $n = 3000$. Although LocalMax and NG causes an additional cost such as 85.5 for $n = 900$ and 423.4 for $n = 3000$ (recall that

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6796
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6796

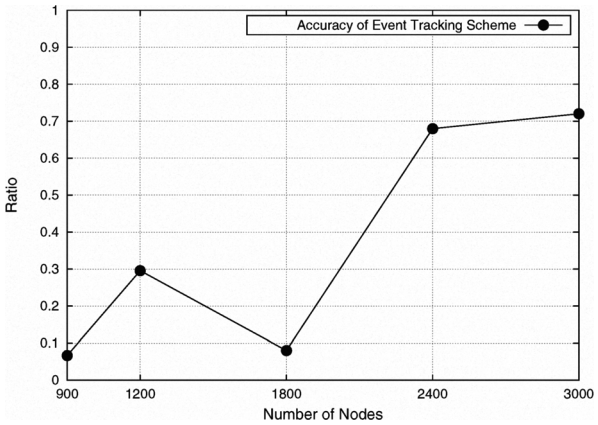

Figure 4. Accuracy of our event tracking scheme.

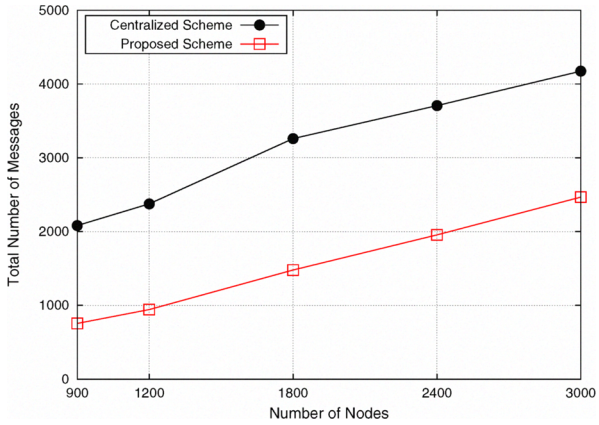

Figure 5. A comparison of the total number of messages transmitted during the event tracking procedure of proposed scheme and centralized scheme.

those messages are newly introduced to the third scheme, and are proportional to the number of neighbors of each node), a significant reduction of Event makes the total cost of the third scheme to be lower than the second scheme for every n.

6.4 Event Tracking Scheme

In this subsection, we evaluate the performance of the proposed event tracking scheme by assuming that our improved scheme is adopted as the underlying shape recognition scheme. We consider a scenario in which event region is initially given as R_1, and after completing a shape recognition of R_1, the center of the region "moves" from $(12, 11)$ to $(10, 10)$. In the following, we refer to the region after the move as R_1', and represent the height of node v in region R_1' as $h'(v)$.

Let u be a local maximum of R_1 identified by our shape recognition scheme, and u^* be the successor of u calculated by our event tracking scheme (note that the scheme may not find such u^* if the density of the network is low). Now let us define the **accuracy** of selecting u^* as a successor of u as: 1) $h'(u^*)/\max_v\{h'(v)\}$ if such u^* is identified by the scheme, and 2) 0 otherwise. Figure 4 shows how the accuracy of our proposed scheme varies by increasing the number of nodes in the network, where each value is an average over 10 random instances, as before. Although it takes a small value for small n, which is mainly due to a fail of identification of successor u^* (recall that the accuracy takes value zero for such case), the accuracy gradually approaches to 0.72 for sufficiently large n; i.e., it correctly identifies the direction of the move, and at the same time, the distance to the new local maximum is estimated almost correctly.

Finally, we evaluate the efficiency of the proposed event tracking scheme in terms of the number of transmitted messages. Recall that in our scheme, an event tracking is initiated by each local maximum u by transmitting an Update message to its neighbors, and terminates when the verification of maximality of successor u' completes. As a competitor, we consider a naive scheme which conducts the following steps in a centralized manner: 1) sink σ_1 collects location information from "all" nodes who detect the change of the coverage by the event region (i.e., node v can recognize itself as a member of $R_1 - R_1'$ if it observes a change of $r(v)$ from true to false, and a member of $R_1' - R_1$ if it observes a change of $r(v)$ from false to true), and 2) the host estimates the current shape by those differential information and the previous shape R_1. Figure 5 shows the result. As shown in the figure, the proposed scheme significantly improves the performance of the naive centralized scheme; e.g., it reduces the number of messages to 36% for $n = 900$, and to 60% for $n = 3000$, while the cost of the boundary updating procedure is almost the same as the verification of the maximality of u', which takes 295.3 for $n = 900$ and 1017.9 for $n = 3000$.

7 Concluding Remarks

In this paper, we proposed a distributed scheme to recognize the shape of a dynamic event region by WSNs. The proposed scheme significantly reduces the number of transmitted messages by identifying critical points in the given event region, and by repeatedly applying a verification of the criticalness of such identified points. The result of simulations indicate that the proposed scheme (almost) correctly identifies the direction and the distance of a move of event region, and the number of message transmissions is sufficiently small compared with a centralized scheme which collects differential information

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6796
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6796

from the nodes.

A future problem is to extend the scheme such that: 1) the shape of the target event region is not restricted to be convex; 2) the efficiency of the scheme is further improved; and 3) the shape of the region can change during a recognition process (the current version allows the move of an event region, but does not allow the change of the shape). We are planning to implement the proposed scheme in actual WSNs consisting of hundreds of sensor nodes, and apply it to actual applications.

References

[1] P. Bonnet, J.E. Gehrke, and P. Seshadri, "Querying the Physical World," IEEE personal communications, 7(5): 10–15 (Oct. 2000).

[2] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and j. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," In Proc. 2001 ACM SIGCOMM Workshop Data Communication in Latin America and the Caribbean, pp. 20–41 (Apr. 2001).

[3] K. K. Chintalapudi and R. Govindan, "Localized Edge Detection in Sensor Fields," In Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA), pp. 59–70 (May 2003).

[4] A. Dhariwal, B. Zhang, B. Stauffer, and C. Oberg, "NAMOS: Networked Aquatic Microbial Observing System," In Proc. 2006 International Conference on Robotics and Automation (ICRA 06), (2006).

[5] M. Greenwald and S. Khanna, "Power-Conserving Computation of Order-Statistics over Sensor Networks," In Proc. the 23rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS), pp. 275–285 (2004).

[6] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A scalable and robust communication paradigm for sensor networks," In Proc. the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), (2000).

[7] B. Krishnamachari and S. Iyengar, "Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks," IEEE Trans. on Computers., 53(3): 241-250 (March 2004).

[8] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," In Proc. the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), (2002).

[9] H. O. Marcy and W. J. Kaiser, "Wireless Integrated Network Sensors: Low power systems on a chip," In Proc. the 24th European Solid State Circuits Conference, (1998).

[10] R. Nowak and U. Mitra, "Boundary Estimation in Sensor Networks: Theory and Methods," In Proc. IPSN 2003 (F. Zhao and L. Guibas, Eds.), LNCS 2634, pp. 80–95, 2003.

[11] R. Rahman, M. Alanyali, and V. Saligrama, "Distributed tracking in multihop sensor networks with communication delays," IEEE Transactions on Signal Processing, 55: 4656–4668 (2007).

[12] M. Rosencrantz, G. Gordon and S. Thrun, "Decentralized sensor fusion with distributed particle filters," In Proc. Conference on Uncertainty in Artificial Intelligence (UAI), 2003.

[13] J. Shin, L. Guibas, and F. Zhao, "A Distributed Algorithm for Managing Multi-target Identities in Wireless Ad-hoc Sensor Networks," In Proc. IPSN 2003 (F. Zhao and L. Guibas, Eds.), LNCS 2634, pp. 223–238, 2003.

[14] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, "Medians and Beyond: New aggregation techniques for sensor networks," In Proc. the 2nd International Conference on Embedded Networked Sensor Systems, pp. 239–249 (2004).

[15] P. Skraba, Q. Fang, A. Nguyen, and L. Guibas, "Sweeps over wireless sensor networks," In Proc. 5th International Conference on Information Processing in Sensor Networks (IPSN), pp. 143-151 (April 2006).

[16] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T.Dawson, P. Bouonadonna,, D. Gay, and W. Hong, "A macroscope in the redwoods," In Proc. the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys), (2005).

[17] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," In Proc. the 2nd European Workshop on Wireless Sensor Networks (EWSN), (2005).

[18] Y. Yang, S. Fujita, S. Kamei, "A shape recognition scheme for wireless sensor networks based on a distance field method," In Proc. ICA3PP 2009, (2009), to appear.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6796
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6796