CISTER
Research Center in
Real-Time & Embedded
Computing Systems

Technical Report

Feature Extraction in Densely Sensed Environments

Maryam Vahabi
Vikram Gupta
Michele Albano
Eduardo Tovar

CISTER-TR-140513
Version:
Date:

Feature Extraction in Densely Sensed Environments

Maryam Vahabi, Vikram Gupta, Michele Albano, Eduardo Tovar

CISTER Research Unit

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8340509

E-mail:

http://www.cister.isep.ipp.pt

Abstract

With the reduction in size and cost of sensor nodes, dense sensor networks are becoming more popular in a wide range of applications. Many such applications with dense deployments are geared towards finding various patterns or features such as peaks, boundaries and shapes in the spread of sensed physical quantities over an area. However, collecting all the data from individual sensor nodes can be impractical both in terms of timing requirements and the overall resource consumption. Hence, it is imperative to devise distributed information processing techniques that can help in identifying such features with ahigh accuracy and within certain time constraints. In this paper, we exploit the prioritized channel-access mechanism of dominance-based Medium Access Control (MAC) protocols to efficiently obtain extrema of the sensed quantities. We show how by the use of simple transforms that sensor nodes employ on local data it is also possible to efficiently extract certain features such as local extrema and boundaries of events. Using these transformations, we show through extensive evaluations that our proposed technique is fast and efficient at retrieving only sensor data point with the most constructive information, independent of the number of sensor nodes in the network.

Feature Extraction in Densely Sensed Environments

Maryam Vahabi*, Vikram Gupta*†, Michele Albano*and Eduardo Tovar*
*CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal
†Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA
Email: {mamvi, vigup, mialb, emt}@isep.ipp.pt

***Abstract*—With the reduction in size and cost of sensor nodes, dense sensor networks are becoming more popular in a wide-range of applications. Many such applications with dense deployments are geared towards finding various *patterns* or *features* such as peaks, boundaries and shapes in the spread of sensed physical quantities over an area. However, collecting all the data from individual sensor nodes can be impractical both in terms of timing requirements and the overall resource consumption. Hence, it is imperative to devise distributed information processing techniques that can help in identifying such features with a high accuracy and within certain time constraints.**

In this paper, we exploit the prioritized channel-access mechanism of dominance-based Medium Access Control (MAC) protocols to efficiently obtain extrema of the sensed quantities. We show how by the use of simple transforms that sensor nodes employ on local data it is also possible to efficiently extract certain features such as local extrema and boundaries of events.

Using these transformations, we show through extensive evaluations that our proposed technique is fast and efficient at retrieving only sensor data point with the most constructive information, independent of the number of sensor nodes in the network.

I. Introduction

Sensor network applications are designed to monitor various physical quantities like temperature, pressure and acceleration. This monitoring process is required to accurately alert the occurrence of certain *features* such as peaks, boundaries and shapes in the distribution of the physical quantities that are being measured. Based on application requirements, it might be important to extract these *features* by performing in-network processing.

While *feature extraction* may not be an issue for a small density network (for example tens of sensor nodes), it is still a challenging problem for a high density network. In applications where measurement constraints are required to have a high spatial granularity, covering even a small area (say, one-square meter), may require hundreds to thousands of sensor nodes. Some examples of densely deployed sensing applications, where *features* of physical quantity need to be monitored frequently, are sleep monitoring for health [1], smart-surfaces for space and aviation [2] and the food industry [3]. The advances in Micro-Electro-Mechanical Systems (MEMS) [4] and large-scale integration, makes it possible to deploy large number of tiny sensors for such emerging applications. Hence, it is necessary to develop algorithms that tackle the problem of *feature extraction* in high density networks.

Figure 1 shows a distribution of a physical quantity (known as a *field*) with three *active regions*. An *active region* is a physical area populated by sensor nodes that sense some *activity* (values of interest).

For example, in Figure 1 we are interested in finding an estimate of the boundaries of *active regions*. A naive approach

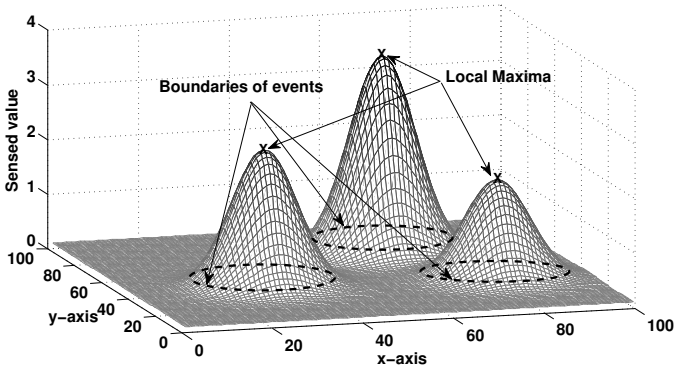


Fig. 1. An example of a 2-D physical quantity *field* with 10,000 sensor nodes. Each data point corresponds to a value sensed by an individual node.

to obtain this information is to collect readings from all sensor nodes and process them centrally. This is inefficient since typical channel access techniques do not scale with increase in number of sensor nodes [5]. It is then advantageous to devise techniques that perform *feature extraction* irrespective of network density.

Computing even simple aggregate quantities such as extrema (minimum or maximum) is not trivial for a dense network as it may require collecting data, in the worst case, from all nodes [6] (even if some sort of spatial sub-sampling is employed [7]). Dominance-based, or binary-countdown, MAC protocols help in finding the minimum value in constant time [8]–[10]. Furthermore, finding peaks and their boundaries in a distributed network, where each data point is measured by individual sensor nodes, is computationally expensive, time consuming and typically does not linearly scalable with an increase in network size.

In this work, we first establish that finding the local extrema is a challenge even after the global maximum is known. Once the global maximum is identified in constant time, we propose a few transforms that nodes employ on local data, which helps in identifying other peaks (local maxima) and their boundaries in the spread of the physical quantity being measured. Our proposed transforms, referred to as *augmenting functions*, allow the identification of local extrema in constant time. Moreover, instead of collecting all data as in the naive case, these *augmenting functions* result in fewer number of measurements being collected.

The main contributions of this paper are:

- Scalable distributed information processing approach that exploits dominance-based protocols to identify certain *features* of a *field* over a two-dimensional (2-D) deploy-

ment of dense sensor network.
- Estimation of the shape and the location of *active regions* in a *field* with a communication cost that is dependent only on the properties of the *field*, and independent of the number of sensor nodes.
- Extensive evaluations on example scenarios that show our proposed technique is fast and efficient at retrieving only the sensor data with the most constructive information.

This paper is organized as follows. We give an outline of the other works related to our approach in Section II. The architecture and the system description based on an aggregate quantity function are provided in Section III. In Section IV, we describe various *augmenting functions* that transform the sensor value such that local extrema or boundaries in various directions become the global extrema. The evaluation of our work is then presented in Section V. Finally, we conclude this paper in Section VI.

II. Background and Related Work

Detection of events in sensor networks is a major application domain and a very broad topic. There are different approaches to address the problem of boundary detection in dense sensor networks. To the best of our knowledge, this is the first boundary detection technique that utilizes dominance-based MAC protocols. To this end, we provide a detailed explanation of this MAC paradigm followed by an outline of the related work.

A. Dominance-based Approach

This work is inspired from dominance-based or binary-countdown protocols [11] implemented for wired networks in the widely used CAN bus [9] as well as its wireless version, called WiDom [10]. The major reason for using a dominance-based MAC protocol is its scalability and constant time-complexity even for very dense networks.

In dominance-based MAC protocols, each node is associated with a priority value that is used to resolve the medium contention. All nodes "simultaneously" start a conflict resolution phase (*tournament*) by transmitting synchronously their priority values bit-by-bit. Since the medium acts as a logical AND operator, the node with the smallest priority value (winner) gains access to the medium .

By using sensed physical quantities (like temperature or acceleration) as the message priority, various aggregate quantities can be obtained in dense networks. For instance, the minimum of sensed value (MIN) can be found in only one *tournament* of the protocol (that is in constant time) [12].

In fact, simultaneous transmissions are a key to the time-efficiency of dominance-based MAC protocols. With carrier-sense or time-division based MAC protocols, computing the MIN value depends on the number of sensor nodes. Studies show that the computation of MIN value with standard IEEE 802.15.4 MAC linearly increases with the growth of network density (e.g. $80\ ms$ for a 40-node network size) [6], [13]. Instead, the WiDom implementation [6] guarantees a constant time ($10\ ms$) for calculating the MIN value regardless of network density.

The process of finding the MIN has been leveraged in the past to find an approximate interpolation and other aggregate quantities [5], [6]. Interpolated values are computed through an iterative process by integration of local information available in each sensor node (its own location information and measured value plus the location information and the measured value of the winner received after each *tournament*). Each sensor node computes the error between its measurement and the corresponding computed interpolation value. The error is then used as the priority value in the conflict resolution phase. After a number of iterations (defined by user), an approximate interpolation of the physical quantity is obtained. However, instead of getting the complete interpolated image, in this work, we aim at detecting certain *features* of the physical quantity *field* by selecting a set of sensor nodes that hold the most constructive information.

B. Boundary Detection

The problem of boundary detection and determining the extent of an event in sensor networks has been investigated in [7], [14]–[16]. Chintalapudi and Govindan presented localized edge detection techniques based on statistics, image processing, and classification [14]. Nowak and Mitra described a method for hierarchical boundary estimation using recursive dyadic partitioning [7]. They developed an inverse proportionality relation between energy consumption and the mean-square error in boundary detection and showed that their method is near-optimal with respect to this fundamental trade-off.

Other schemes represent the boundary of an event or the signal landscape of a sensor network compactly using in-network aggregation [17]–[19]. Gandhi et al. studied the problem of monitoring the events of sensor networks using sparse sampling [18]. However, their algorithm requires the prior knowledge of the event geometry (e.g. circle, ellipse, or rectangle) for computational efficiency.

There are also contour-based methods for deciding the type of an event [20], [21]. They consider energy-efficient techniques to construct and incrementally update a number of 2-D contour maps in a sensor network. Another field of research involves detecting holes and topological features in a sensor network are presented in [22], [23]. In these approaches, local connectivity graphs are used to infer static features of an event and require the involvement of all the nodes in the network. By contrast, our approach is quite different from the above mentioned techniques, as we exploit an underlying dominance-based MAC protocol for very sparse spatial sampling through a strategic selection of sensors.

III. System Model

We consider a sensor network where each sensor node has a unique *id*, i, and measures a particular physical quantity, s_i, using a sensing unit. Each sensor node knows its 2-D coordinates (x, y) in the plane of deployment. We assume that the *feature extraction* mechanism can either be carried out periodically as a part of a *sense-process-actuate* control-loop, or sporadically initiated by an external controller, like a data sink or a master node.

The collection of all the sensor values across the total sensed area is referred to as a *field*, as shown in Figure 1. Each data point in the *field* corresponds to a true (or non-faulty)

sensor reading value, sensed by an individual sensor node at its physical location. The spatial granularity and the size of the *field* is directly correlated to the distribution of the nodes and their spread. We also define *active region* as a physical area populated by sensor nodes that sense some *activity*. The overarching goal of our technique is to find location, boundaries and shape of an *active region*, which we referred to as *features*, in the physical environment. For illustrating our approach and its evaluation, we generated various sample *fields* by a summation of 2-D Gaussian functions (explained in Appendix A).

Function $\mathcal{M}(v_i)$ represents the process of finding the MIN over values v_i published by independent sensor nodes in a broadcast medium where $i \in \{1, 2, ...N\}$ and N is the number of sensor nodes in a broadcast domain. We exploit the property of simultaneous transmissions in dominance-based protocols to devise this function. v_i is the scaled value that each sensor node computes based on its measured value s_i and the global maximum s_{max} measured in the *field*. Each application of $\mathcal{M}(v_i)$ is referred to as a *round*. After each *round*, all the other sensor nodes know the *id* and the location of the sensor node with the MIN value. This sensor node is known as the winner of the *round*. We use $\hat{i}$ to denote the *id* of the winner. Hence, the function $\mathcal{M}$ can be formally represented as:

$$\{\hat{v}, \hat{i}, \hat{x}, \hat{y}\} = \mathcal{M}(v_i) \qquad \forall i \in \{1, 2, ...N\}$$

where $\hat{x}$ and $\hat{y}$ are the x and y coordinates of the sensor node with the global minimum value $\hat{v}$.

The choice of v_i values used by an i^{th} sensor node in the application of function $\mathcal{M}$ depends on the requirements of the application. It should be noted that a sensor node can only use its local information (such as *id*, sensor value and physical location) and other global data available from previous iterations of $\mathcal{M}$. For our goal of identifying various features, we augment (or transform) these input values such that the global minimum returned by $\mathcal{M}$ corresponds to one of the local minima or an edge of an *active region*.

The function $\mathcal{M}$ can be applied to the value computed by ϕ_i, which is a function of the sensor values and their location. The codomain of the function ϕ_i denotes a set of values it can take. We assume that the cardinality of the codomain of ϕ_i is large enough that the probability of computing the same ϕ_i by two sensor nodes is negligible and thus an unique sensor node $\hat{i}$ is chosen. The time-complexity of $\mathcal{M}$ is proportional to the number of bits used to encode ϕ_i hence, it is proportional to the logarithm of the cardinality of the codomain. However, as all sensor nodes transmit simultaneously in a dominance-based MAC protocol, the time required for the application of $\mathcal{M}$ over a network is independent of the number of sensor nodes. Table I summarizes the notations and symbols that we use in the following sections where we define a set of functions to extract different *features* of the *field*.

TABLE I
SUMMARY OF THE SYMBOLS AND NOTATIONS

Symbols	Description
s_i	sensor value measured by sensor node i
s_{max}	maximum value collected by sensor nodes
$\nu_i = \frac{1+s_i}{s_{max}}$	scaled sensor value with respect to the maximum value over all sensor nodes
A_β, A_γ, A_δ	*augmenting functions* used for different *feature extraction* techniques
$\mathcal{M}(v_i)$	finding the MIN over all v_i values
$\mathcal{M}(-v_i)$	finding the MAX over all v_i values
$\phi(\nu_i, x_i, y_i)$	function computed by sensor nodes
π_s	termination condition

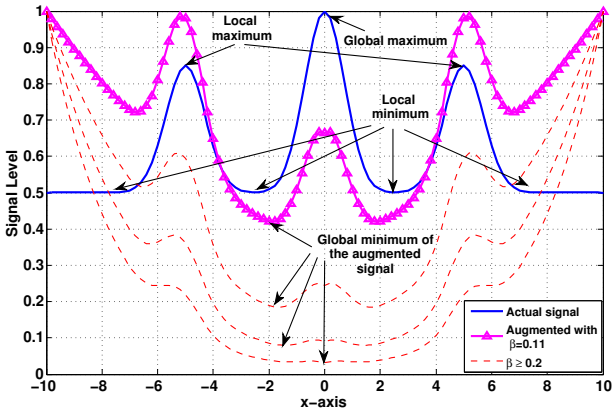


Fig. 2. An example of *augmented function* $\mathcal{A}_\beta$ of the normalized input signal with various β in range of [0.11, 0.5) in 1-D domain.

IV. Feature Extraction Using Augmenting Functions

We now describe in detail a set of *augmenting functions* that can extract an approximate but faithful representation of various *features* in the *field* by applying simple transforms on the property of sensor values. We show that this process can be done with limited number of broadcast messages.

A. $\mathcal{A}_\beta$ Distance Augmenting Function

As described earlier, a global maximum value in a sensor *field* can be easily found with function $\mathcal{M}$. However, finding the spread or boundaries of this peak (local maximum) is not trivial. If we modify the MAC protocol such that the sensor node with the global maximum does not participate in the next *round*, there is still a high chance that one of the adjacent sensor nodes to the peak will become the next local maximum. On the other hand, we might have to predefine a neighborhood range around the peak that should be excluded in the next cycle to make sure that another local maximum be the new global maximum. In this case, choosing the size and shape of the neighborhood is also a challenge.

We demonstrate the process of finding an adjacent local minimum with a simple 1-D Gaussian *field* example, shown in Figure 2. Using this example we show that, by utilizing the *augmenting function* over the input signal, the adjacent local minimum becomes the global minimum.

In this example, the *field* consists of three peaks and the highest peak (the global maximum) lies in the center. Finding the spread of this global maximum is not trivial as the global

minimum point can be one of those sensor nodes near the borders of the *field*. It is important to suitably modify the process of identifying the extrema such that a local minimum adjacent to a peak (*adjacent valley*) can be found. For this purpose, we observe that an *adjacent valley* should have a low value and its distance from the peak should also be small. Hence, each value in the *field* is transformed (multiplied) with the distance from the peak. With this transformation, the points located farther from the peak are associated with higher values (compared with its sensed value) and only points with lowest sensed value and smallest distance from the peak can become the global minimum in the augmented *field*. It is possible that this global minimum is a point in an *adjacent valley*. The boundary of this peak is found with just two *rounds* of executing $\mathcal{M}$ function: (i) finding the global maximum in the original *field*, and (ii) identifying the global minimum in the augmented *field*.

For 2-D Gaussian *fields*, the 1-D approach described above can be directly applied. Different *active regions* of the *field* are found by excluding the sensor nodes lying inside the identified *active region* from participating in the next *rounds*. The process of finding *active regions* is shown in Algorithm 1. Initially, function $\mathcal{M}$ is used to find the global maximum, then a circular area around the identified peak is filtered out. The radius of this filtering circle is set as the distance of an *adjacent valley* from the peak.

Each sensor node uses a function ϕ_β as an input to $\mathcal{M}$. The function ϕ_β takes into account two properties of the *field*; sensed value s, and sensor's proximity to the peak d and is defined as follows:

$$\phi_\beta = \nu \times \mathcal{A}_\beta(d) \tag{1}$$

ν is the scaled value and is defined as:

$$\nu = \frac{1+s}{s_{max}} \tag{2}$$

and s_{max} is the value of the global maximum. $\mathcal{A}_\beta(d)$ is the *augmenting function*, which is formulated as follows:

$$\mathcal{A}_\beta(d) = e^{\beta(\frac{d}{d_{max}})^2} \tag{3}$$

where d_{max} is the diameter of the monitored area, and it is used to normalize the distance from the peak, and β is a parameter to control the impact of distance on the augmented *field* with respect to the scaled value ν. To ensure that the winning sensor node is located at the adjacent local minimum of the *field*, the priority function ϕ_β is computed so that the distance is exponentially penalized.

Finally, by finding the global minimum over ϕ_β values at all the sensor nodes, we can find a point that lies in the *adjacent valley* with high probability. The distance between the *adjacent valley* point and the peak determines the filtering radius, R_f (line 10 in Algorithm 1). Sensor nodes that are located within the filtering circle refrain from participating in the next iterations. Repeating this procedure helps in finding all the peaks in the region. The algorithm stops when the next peak is less than a certain user defined threshold π_s, which is a fraction of the global maximum. By finding the peaks and their spread, this approach helps in identifying the location and the number of circular *active regions* in a *field*.

Algorithm 1: *Distance augmenting function $\mathcal{A}_\beta$, executed on each sensor node n_i*

```
1  begin
2      Silent ← 0;
3      s_max ← M(−φ_s) ; // find global MAX
4      π_s ← a fraction of s_max ; // termination
       condition setting
5      while s_next−peak > π_s do
6          if Silent ≠ 1 then // not filtered out
7              s_next−peak ← M(−φ_s) ; // find the new
               peak
8              Compute φ based on Equation 1;
9              < s_adj−valley, d_adj−valley > ← M(φ_β);
10             R_f ← d_adj−valley ;
11             d_i ← distance between next-peak and node n_i;
12             if d_i < R_f then
13                 Silent ← 1;
```

B. $\mathcal{A}_\gamma$ Vector Augmenting Function

Our second *augmenting function* is used for cases where we are interested in finding a boundary around all *active regions*. This approach can be used for a range of applications such as crowd monitoring for smart cities or sleep monitoring for health care. In this approach, if we assign larger values of ϕ to sensor nodes that lie on the boundary of an *active region*, then the result of applying the $\mathcal{M}$ function over the negation of ϕ value corresponds to the boundary of *active regions*. This is implemented by augmenting the sensor values with a function that grows in a particular direction.

The *vector augmenting function* $\mathcal{A}_\gamma$, is designed to work with binary *fields*, where the input signal is not smoothly distributed, and two neighboring sensor nodes may have very close or very different measurements. By applying $\mathcal{A}_\gamma$, each sensor node multiplies its measurement with a vector $\vec{u}$. The rationale behind using a direction is to find sensor nodes that sense a high value and are located as far as possible in the direction given by $\vec{u}$, which corresponds to the edge of an *active region* in that direction. To compute the function ϕ, sensor nodes transform their locations with a direction as:

$$\phi_\gamma = v_p \times \mathcal{A}_\gamma(x, y, \theta) \tag{4}$$

where v_p is a *participation value* which is either 0 or 1 depending on the sensed value being below or above a threshold. Sensor nodes with $v_p = 1$ are part of the *active region*. The *augmenting function* is defined as:

$$\mathcal{A}_\gamma(x, y, \theta) = e^{\gamma(x.\cos(\theta)+y.\sin(\theta))} \tag{5}$$

x and y are the coordinates of the sensor location and θ is the direction given by vector $\vec{u}$.

By using $\mathcal{A}_\gamma$, the value is "projected" in a direction given by the vector $\vec{u}$. The sensor node that has the largest value has a high probability of being located on the border of an *active region* in the direction of $\vec{u}$.

The working of this approach is outlined in Algorithm 2. The algorithm explores the region by choosing random directions defined in a set $\{\bar{v_\theta}\}$. We assume that the seed for

Algorithm 2: *Vector augmenting function $\mathcal{A}_\gamma$, executed on each sensor node n_i*

```
begin
    s_max ← M(−φ_s) ; // find global MAX
    π_s ← a fraction of s_max ; // termination condition setting
    {f_θ} ← ∅ ; // filtered direction set
    if s < π_s then
        v_p ← 1;
    else
        v_p ← 0;
    {v̄_θ} ← a set of θ directions ;
    foreach θ ∈ {v̄_θ} do
        if θ ∉ {f_θ} then
            Compute φ_γ based on Equation 4;
            < s_edge, (x, y)_edge, θ_edge > ← M(−φ_γ);
            if (x, y)_edge was found with other direction, θ_x then
                {f_θ} ← {f_θ} ∪ θ_x θ + 2ε;
```

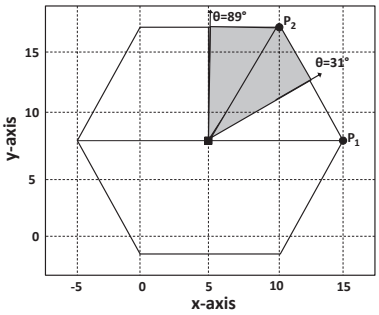


Fig. 3. An example of boundary detection for a hexagonal-shape event.

generating this pseudo random directions is generated by the initiator of the boundary detection process (thus, all the sensor nodes will use the same $\{\bar{v_\theta}\}$ in each iteration). Repeating this procedure in different directions makes it possible to find the boundary of an *active region* by computing the convex hull of the collected locations.

If two angles lead to a same point, it means that any angle over the arc confined by these two angles would result in that point. Thus, this arc can be filtered out from further investigation. Figure 3 shows an example where two angles of $\theta_1 = 31\,^\circ$ and $\theta_2 = 89\,^\circ$ lead to find the same location P_2. In this case, there is no need to examine more directions in the region of $31\,^\circ \leq \theta \leq 89\,^\circ$. We also use *marginal extension* ϵ to limit further the redundant directions in the arc denoted by $\theta_i \pm \epsilon$. We discuss the impact of ϵ in SectionV.

More iterations of the algorithm leads to more accurate boundaries, but at the cost of more resource consumption. We show in Section V that with above filtering strategy, we are able to reduce the number of dominance *rounds*, while still building a good description of the *active region*. The worst case for our algorithm happens when the event boundary looks like a perfect circle where new directions will always give new points (considering a very dense deployment of sensors and a marginal extension of $\epsilon = 1\,^\circ$, in this case, up to 359 individual readings will be collected). However, the results show that the number of readings is usually much less in practice.

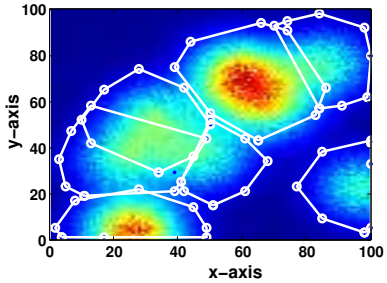


Fig. 4. An example of boundary detection around the peaks with non-uniform distribution.

C. $\mathcal{A}_\delta$ Joint Augmenting Function

As described earlier, the *distance augmenting function* $\mathcal{A}_\beta$ identifies circular *active regions*. For complex *fields*, $\mathcal{A}_\beta$ may need several circular *active regions* to cover a non-circular shape. On the other hand, *vector augmenting function* $\mathcal{A}_\gamma$ only provides a convex hull of all the *active regions*. So, a *field* with several isolated *active regions* is identified as one large shape, which may not provide enough insights regarding the structure of the *active regions*.

To find the boundary of an *active region* with non-circular distribution, we devised a new *augmenting function* $\mathcal{A}_\delta$, that is a composition of $\mathcal{A}_\beta$ and $\mathcal{A}_\gamma$. By applying $\mathcal{A}_\delta$ sensor nodes that identified with $\mathcal{M}$ function should have the following properties: (i) lie close to the peak, and (ii) locate on the edge of the local boundary in a given direction. The value used by each sensor node is then defined as follows:

$$\phi_\delta = \mathcal{A}_\delta \times \frac{\mathcal{A}_\beta(d)}{\mathcal{A}_\gamma(x, y, \theta)} \tag{6}$$

where $\mathcal{A}_\delta$ is given by:

$$\mathcal{A}_\delta = \nu^{-\delta}$$

$\mathcal{A}_\delta$ is an inverse polynomial of degree δ of the scaled sensed value ν, and δ is a parameter to guarantee that the low values lying far away from a peak bring a stronger contribution to the ϕ values. As a consequence, for a given value of θ, the point that has the minimum value of ϕ_δ is more likely to lie on the boundary in the θ direction. We sweep the area around a peak with different values of θ. In our evaluation, for θ, we used equal intervals of $\pi/4$ ($\theta \in \{0, \pi/4, ...2\pi\}$)[1]. Thus, after finding a new peak, the locations of the nearest *adjacent valleys* in eight directions are found and the convex hull of all these readings represents the area around that *active region*. This enables us to find complex geometric shapes according to the shape of *active regions* instead of only circles, as shown with an example in Figure 4.

V. Evaluation

We evaluated our proposed approaches in various simulated scenarios by considering two metrics:

[1]Smaller intervals will result in better accuracy of boundary at the cost of higher number of broadcast messages.

Number of *Rounds*: A sensor node gets channel-access permit to broadcast a message in one *round* of dominance-based protocol if its value is the global minimum. Hence, the number of *rounds* corresponds to the number of message transmissions.

Accuracy: Accuracy represents the fraction of sensor nodes, that declared themselves to be located in the *active region*(*s*) N_{det} to the number of sensor nodes that truly lie in the *active region*(*s*) N_{true}:

$$\text{Accuracy} = \frac{N_{det}}{N_{true}} \times 100. \tag{7}$$

In cases where the detected area is larger than the actual *active region*, ($N_{det} > N_{true}$), accuracy is more than 100%, which signifies the overestimation of the *active region*.

We now present the results of the *augmenting functions* in six example scenarios.

A. Identifying the Active Regions

We considered a network of size 100×100 sensor nodes in a shared medium, either a bus or a "single" wireless broadcast domain. First, we show the performance of *distance augmenting function* to identify circular *active regions*. For evaluating our approach, we generated various scenarios with several *active regions*. The *active regions* may or may not overlap resulting in complex *fields*. The details of the scenarios are provided in Figure 14 in Appendix A.

In each *round* of execution, after finding the global maximum, all the sensor nodes compute local values of ϕ_β according to the Equation 1. Figure 5 illustrates the number of circular *active regions* with different termination rules for scenario *sc1*. Increasing the threshold level helps the algorithm to converge faster in smaller number of *rounds*, but at the cost of reduced accuracy.

Figure 6 shows number of *rounds* and the corresponding obtained accuracy for different threshold values. In some scenarios, increasing the termination threshold reduces the number of *rounds* as well as the overestimated area. This happens for scenarios *sc1*, *sc3* and *sc6*, where the *active regions* can overlap. However, for scenarios *sc2*, *sc4* and *sc5*, where either few *active regions* exist or the *active regions* are far apart, the number of *rounds* and accuracy remains almost the same for all termination conditions. The existence of isolated peaks in the *field* results in identifying all *active regions* by a fixed number of circular sections. The high value of overestimation in *sc4* and *sc5* is due to the steepness of the peak in the *field*. This steepness causes an overestimation in the filtering radius which in turn leads to much larger (squared) overestimation of the circle's area.

The choice of the exponential coefficient β in Equation 3 impacts the relative weight of distance from the peak with respect to the value of the *field* at a given point. Choosing an optimal value of β is not possible without the prior knowledge of the *field*, but the order of β can be chosen based on the size of the *field* such that a proper trade-off between the sensor value at a given sensor node and its distance from the peak is maintained. Particularly, β should be chosen such that the impact of distance can be pronounced compared to the distance normalization (d_{max} in Equation 3). For an area of 100×100,

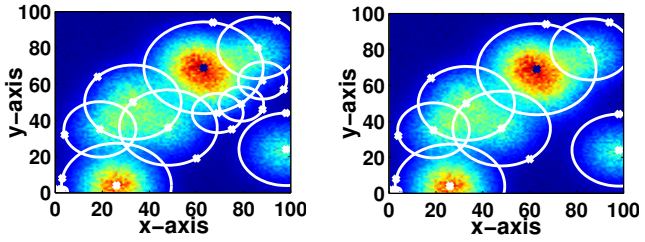


Fig. 5. The effect of termination threshold, π_s on the detection of *active regions*. The algorithm terminates when a new detected peak is (a) 20% and (b) 30% of the global maximum value.

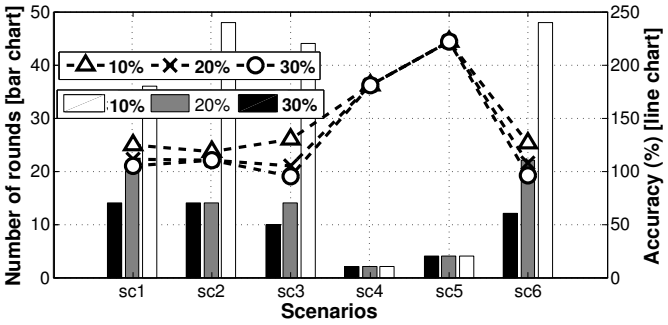


Fig. 6. The number of *rounds* and the accuracy of *active region* estimation in different scenarios for various $\pi_s = \{10\%, 20\%, 30\%\}$ of the global maximum.

we found that suitable values of β are in the order of 10. Specific results on the accuracy for three scenarios *sc2*, *sc4* and *sc5* are shown in Figure 7. For scenarios with isolated peaks, higher values of β improve the coverage estimation accuracy, but for scenarios where the spread of peaks can overlap, the effect of changing β is less pronounced, as in the case of scenario *sc2*.

B. Convex-hull around Active Regions

For evaluating the convex-hull approach described in Section IV-B, we convert the *field* to a binary *field* by thresholding, such that a sensor node's value is 1 if the sensed value is greater than 10% of the maximum value, and 0 otherwise. The details of the scenarios are also provided in Figure 15 in Appendix A. We set the marginal extension angle to $\epsilon = 1^\circ$ and $\gamma = 1$. We compared our technique with a TDMA approach, where a fixed number of randomly chosen sensor nodes send their measurements. For the random approach, the number of readings was set to 150.

Figure 8 shows the accuracy of our second approach in terms of average percentage of coverage area by running the simulation over 100 iterations. As shown, our technique covers more than 97% of the area in all the scenarios through transmitting 26 to 33 broadcast messages compared with 36% to 72% coverage by 150 randomly chosen packets. Hence, we acquire a more accurate boundary estimation with 77% less broadcast messages.

Increasing the marginal extension angle, ϵ, still provides a satisfactory coverage area estimation, while reducing the number of messages. We tested the performance of our proposed

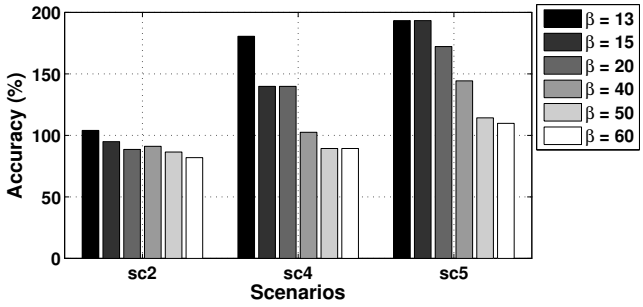


Fig. 7. The effect of β on the accuracy.

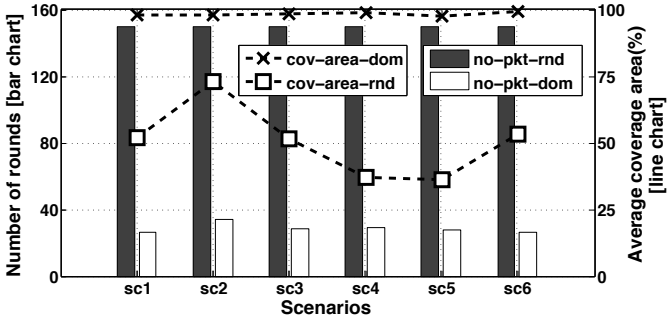


Fig. 8. The number of *rounds* and the average of coverage area estimation for $\mathcal{A}_\gamma$ and random approach in different scenarios.

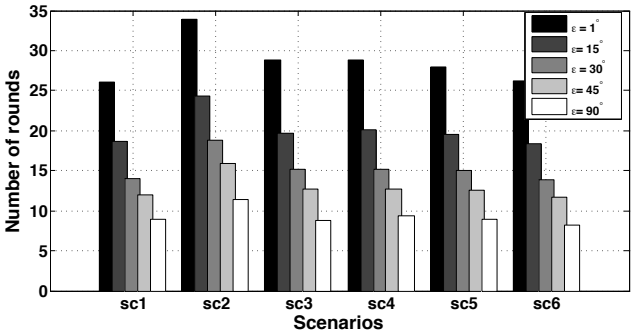


Fig. 9. The number of *rounds* in $\mathcal{A}_\gamma$ with different marginal extension angle, ϵ.

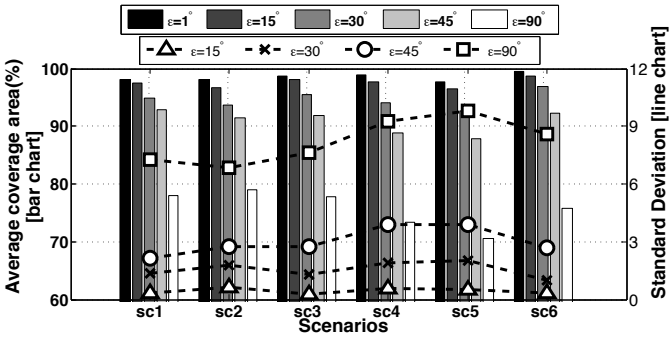


Fig. 10. The average of estimated coverage area with different marginal extension angle, ϵ and its standard deviation.

algorithm with different marginal extension angles in all the mentioned scenarios. As shown in Figure 9, by increasing ϵ, the number of dominance *rounds* decreases, since by enlarging the angle, more search space is filtered out and consequently less packets are needed to be broadcast.

However, increasing ϵ might diminish the performance of the algorithm in terms of estimated coverage area. Figure 10 shows the performance degradation by increasing the ϵ up to $90°$. Under the latter setting, the proposed algorithm is able to cover around 70% of the *active region*. In addition to the shown value of ϵ in the graph, we also ran the simulation for $\epsilon = \{5°, 10°\}$ and took the standard deviation of the average coverage area for $\epsilon = (1° : \epsilon)$. Since the results for $\epsilon = \{5°, 10°\}$ were very close to the case where $\epsilon = 1°$, the coverage area computed by these values are not shown in the figure. As can be seen in Figure 10, $\epsilon = 15°$ has the smallest standard deviation (smaller than 0.65), which suggests that the coverage area computed by $\epsilon = 15°$ leads to the same coverage area as given by $\epsilon = 1°$, while at the same time using $\epsilon = 15°$, requires much smaller number of *rounds* compared with $\epsilon = 1°$ setting, as shown in Figure 9.

C. Non-circular Active Regions

As discussed in Section IV-C, $\mathcal{A}_\delta$ helps in finding the boundary of a non-circular *active region*, instead of identifying circular *active regions* from a complex shape.

Figure 11 shows the comparison of the performance of *distance augmenting function* with the *joint augmenting function*. For scenarios where a number of *active regions* lie very close to each other, the number of rounds required by $\mathcal{A}_\beta$ is 20% more than that for $\mathcal{A}_\delta$. For more sparse events, the number of rounds required by $\mathcal{A}_\beta$ is 50% less compared with $\mathcal{A}_\delta$. This happens due to the higher number of iterations that are needed to be performed by $\mathcal{A}_\delta$ to find the boundary of more than one complex shape. It should be noted that $\mathcal{A}_\delta$ is able to find the boundary of a non-circular *active region* and detects it as one region instead of a group of several close *active regions*.

Comparison of *vector augmenting function* and joint augmentation is shown in Figure 12. It is evident that the combined approach helps in demarcating different *active regions* while $\mathcal{A}_\gamma$ detects the overall outer boundary.

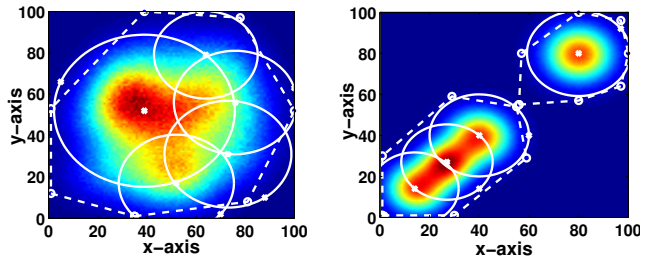
D. The Impact of Network Density

Finally, we compared our techniques under various network densities. As our approaches only depend on collecting the global extrema of various *augmenting functions*, the results indicate that increasing the network density has almost no impact on the number of rounds. Figure 13 shows the number of *rounds* required in each technique with respect to the network size. The slight variation in the number of *rounds* is due to the effect of termination condition in $\mathcal{A}_\beta$ and the randomness in the choice of θ in $\mathcal{A}_\gamma$.

VI. Conclusion and Future Work

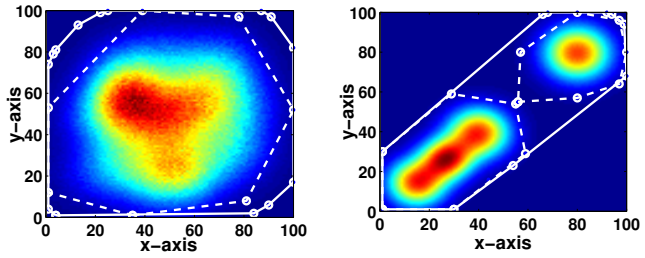
In this paper, we presented a set of techniques that identify various *features* in the distribution of sensed physical quantities over a dense deployment of sensor nodes, confined in a single broadcast coverage area.

With such simple-yet-effective modifications we can obtain the location and shapes of *active regions* with a number of messages proportional to the properties of the *field*.



(a) (b)

Fig. 11. $\mathcal{A}_\beta$ versus $\mathcal{A}_\delta$, dash lines show the boundaries provided by $\mathcal{A}_\delta$ and solid lines refer to *active regions* detected by $\mathcal{A}_\beta$; (a) $\beta = 30$, $\gamma = 0.3$; (b) $\beta = 30$, $\gamma = 0.2$; with $\beta = 10$ for $\mathcal{A}_\beta$.



(a) (b)

Fig. 12. $\mathcal{A}_\gamma$ versus $\mathcal{A}_\delta$, dash lines show the boundaries provided by $\mathcal{A}_\delta$ and solid lines refer to that given by $\mathcal{A}_\gamma$ (a) $\beta = 30$, $\gamma = 0.3$; (b) $\beta = 30$, $\gamma = 0.2$; with $\epsilon = 15°$ for $\mathcal{A}_\gamma$.

For the future, our primary objective is to extend the devised approaches for a larger dense network, where sensor nodes do not necessarily lie in a single broadcast area. This will require dividing the area into several single broadcast areas and then carefully combining the information from each broadcast area to identify *active regions* within certain time constraint. We are also studying *feature extraction* in 3-D deployment of sensor network. The 3-D monitoring raises nontrivial issues in abstracting the environment. In this context, efficient data gathering and further enhancements on *augmenting functions* are required.

ACKNOWLEDGMENTS

The authors would like to thank Raghuraman Rangarajan for his valuable comments and suggestions. This work was partially supported by the North Portugal Regional Operational Program (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by National Funds through FCT (Portuguese Foundation for Science and Technology), within project ref. NORTE-07-0124-FEDER-000063 (BEST-CASE, New Frontiers); by National Funds through FCT and by ERDF (European Regional Development Fund) through COMPETE (Operational Programme 'Thematic Factors of Competitiveness'), within projects FCOMP-01-0124-FEDER-037281 (CISTER), FCOMP-01-0124-FEDER-020312 (SMARTSKIN) and FCOMP-01-0124-FEDER-028990 (PATTERN); by FCT and the EU ARTEMIS JU under grant nr. 621353 (DEWI); also by FCT and ESF (European Social Fund) through POPH (Portuguese Human Potential Operational Program) under PhD grant SFRH/BD/67096/2009.

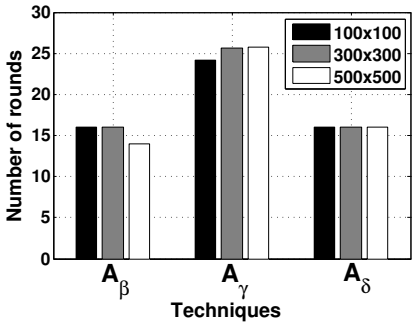


Fig. 13. The impact of density on the performance of each technique for scenario *sc2*.

REFERENCES

[1] J. Liu, W. Xu, M.-C. Huang, N. Alshurafa, M. Sarrafzadeh, N. Raut, and B. Yadegar, "A dense pressure sensitive bedsheet design for unobtrusive sleep posture monitoring," in *IEEE PerCom*, 2013.

[2] J. A. Paradiso, J. Lifton, and M. Broxton, "Sensate media-multimodal electronic skins as dense sensor networks," *BT Technology Journal*, vol. 22, no. 4, 2004.

[3] M. Connolly and F. O'Reilly, "Sensor networks and the food industry," in *Workshop on REALWSN*, 2005.

[4] J. W. Gardner, V. K. Varadan, and O. O. Awadelkarim, *Microsensors, MEMS, and smart devices*. Wiley Online Library, 2001, vol. 1.

[5] A. Ehyaei, E. Tovar, N. Pereira, and B. Andersson, "Scalable data acquisition for densely instrumented cyber-physical systems," in *IEEE ICCPS*, 2011.

[6] N. Pereira, R. Gomes, B. Andersson, and E. Tovar, "Efficient aggregate computations in large-scale dense wsn," in *IEEE RTAS*, 2009.

[7] R. Nowak and U. Mitra, "Boundary estimation in sensor networks: Theory and methods," in *ACM/IEEE IPSN*, 2003.

[8] A. K. Mok and S. Ward, "Distributed broadcast channel access," *Computer Networks*, vol. 3, 1979.

[9] "Can specification version 2.0," *Bosch Gmbh, Stuttgart, Germany*, 1991.

[10] N. Pereira, B. Andersson, and E. Tovar, "Widom: A dominance protocol for wireless medium access," *IEEE Trans. on Industrial Informatics*, vol. 3, no. 2, 2007.

[11] A. K. Mok and S. A. Ward, "Distributed broadcast channel access," *Computer Networks*, vol. 3, no. 5, 1979.

[12] B. Andersson, N. Pereira, W. Elmenreich, E. Tovar, F. Pacheco, and N. Cruz, "A scalable and efficient approach for obtaining measurements in can-based control systems," *IEEE Trans. on Industrial Informatics*, vol. 4, no. 2, 2008.

[13] N. Pereira and B. Andersson, "Widom vs ieee 802.15.4 for computing min in a single broadcast domain," IPP Hurray, Tech. Rep., 2008.

[14] K. K. Chintalapudi and R. Govindan, "Localized edge detection in sensor fields," *Ad Hoc Networks*, vol. 1, no. 2, 2003.

[15] Y. Wang, J. Gao, and J. S. Mitchell, "Boundary recognition in sensor networks by topological methods," in *ACM MobiCom*, 2006.

[16] M. Singh, A. Bakshi, and V. K. Prasanna, "Constructing topographic maps in networked sensor systems," in *IEEE ASWAN*, 2004.

[17] C. Buragohain, S. Gandhi, J. Hershberger, and S. Suri, "Contour approximation in sensor networks," in *DCOSS*. Springer, 2006.

[18] S. Gandhi, S. Suri, and E. Welzl, "Catching elephants with mice: sparse sampling for monitoring sensor networks," *ACM Trans. on Sensor Networks*, vol. 6, no. 1, 2009.

[19] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek, "Beyond average: Toward sophisticated sensing with queries," in *ACM/IEEE IPSN*, 2003.

[20] W. Xue, Q. Luo, L. Chen, and Y. Liu, "Contour map matching for event detection in sensor networks," in *ACM SIGMOD*, 2006.

[21] M. Li and Y. Liu, "Iso-map: Energy-efficient contour mapping in wireless sensor networks," *IEEE Trans. on Knowledge and Data Engineering*, vol. 22, no. 5, 2010.

[22] S. Funke, "Topological hole detection in wireless sensor networks and its applications," in *ACM DIALM-POMC*, 2005.

[23] A. Kröller, S. P. Fekete, D. Pfisterer, and S. Fischer, "Deterministic boundary recognition and topology extraction for large sensor networks," in *ACM SODA*, 2006.

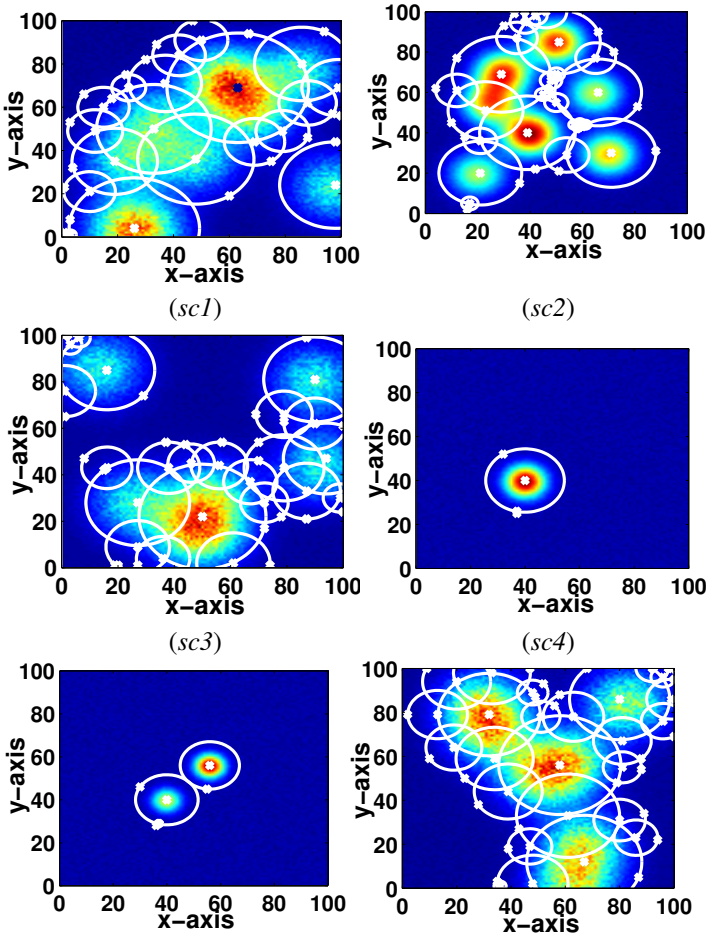


Fig. 14. Six scenarios with different *active regions*; each sensor node sets the value of $\beta = 10$ to compute its priority —see Equation (1-3). Each circle represents one filtering zone that is computed by two readings from the sensor network and excludes sensor nodes located inside the circle from participation in the future iteration(s) of the algorithm. π_s is set to 10% of the global maximum value.

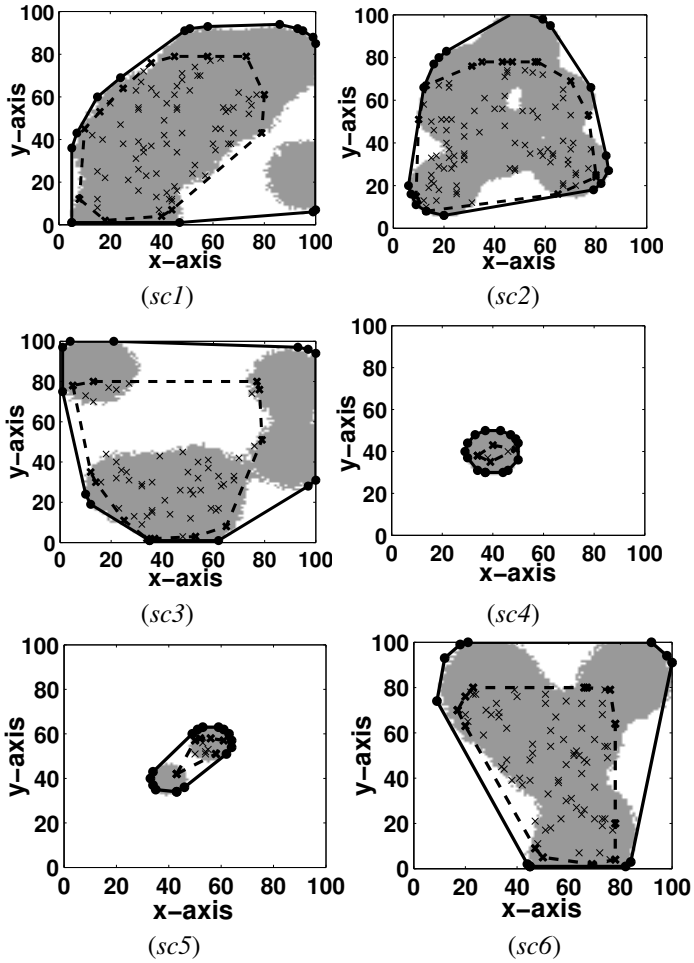


Fig. 15. Six scenarios with different *active regions*; solid lines show the boundary computed by our algorithm with $\epsilon = 1^\circ$ and 40 iterations of the algorithm, and dash lines show the boundary computed by the random algorithm with 150 random readings.

Appendix A
Example Scenarios

The *field* is assumed to be sum of the signals generated by a set of active sources (like a heater or a car exhaust). The spread from each source is assumed to be Gaussian with the following representation:

$$f(x_i) = k_i e^{-\alpha(x-x_i)^2+(y-y_i)^2} \tag{8}$$

The *field* is then represented by:

$$f(x) = \sum_{i=1}^{n} f(x_i) + \mathcal{N}(0, \sigma^2). \tag{9}$$

An example of a 2-D *field* is shown in Figure 1, where the z-axis corresponds to sensor values and the spread of the *field* is assumed to be the sum of several 2-D Gaussian signals. The scenarios for evaluation are generated by varying the parameters n and k_i in the signal model given by (8) and (9).

The scenarios which are used in the simulation experiment for evaluating *distance augmenting function* and *vector augmenting function* are given in Figure 14 and Figure 15 respectively.