How to identify global trends from local decisions? Event Region Detection on Mobile Networks

Andreas Loukas*, Marco Zuniga*, Ioannis Protonotarios*, Jie Gao†
Delft University of Technology*, Stony Brook University†
{a.loukas@, m.a.zunigazamalloa@, i.protonotarios@student.}tudelft.nl*, jgao@cs.sunysb.edu†

***Abstract*—The decentralized detection of event regions is a fundamental building block for monitoring and reasoning about spatial phenomena. However, so far the problem has been studied almost exclusively for static networks. This study proposes a theoretical framework with which we can analyze event detection algorithms suitable for large-scale mobile networks. Our analysis builds on the following insight: *the inherent trends of spatial events are well captured by the spectral domain of the network graph*. Using this framework, we propose novel local algorithms that are location-free; that work with mobile nodes and dynamic events; that operate on 3D topologies; and that are simple to implement. We are not aware of event detection algorithms possessing all these traits. Simulations based on complex oil spill traces showcase the resilience and robustness of our methods. Additionally, we demonstrate their validity for practical scenarios by evaluating them on a 105 node testbed.**

I. Introduction

The real-time monitoring of spatial phenomena, like oil spills, pollution clouds, or chemical spills, is one of the key applications fostering the development of sensor and robotic networks [1]. Recent studies have demonstrated that sensor networks can distributedly perform high-level inference tasks, such as identifying "*how many distinct events are currently occurring*" or recognizing if "*two regions are adjacent* [2, 3]. To reason about spatial events however, a network must first identify and track the events of interest – see Figure 1. Indeed, the event region detection problem has attracted significant interest [4–11]. Nevertheless, the overwhelming majority of these works focus on the static case. When the network is mobile and the events dynamic, region detection becomes a challenging problem. Each node needs to distinguish the underlying signal trends (spatial events) from noise by using local information and, preferably, without knowing its location.

This study focuses on the design and evaluation of local event region detectors for mobile networks. Our basic insight is that, the decomposition of event signals in the spectral domain (the domain spanned by the eigenfunctions of the Laplacian) captures fundamental signal trends. In fact, the identification of spatial regions can be achieved by filtering the event's signal in the spectral domain. Based on this observation, we propose a spectral framework for event detection. This is a Fourier-like framework which can be used to model a number of algorithms, such as those using basic differentials and differentials of diffusions (*e.g.*, LoG and DoG methods). By decoupling the operation of the algorithm from the underlying topology, our framework allows a systematic comparison of event region detectors (*e.g.*, with respect to their resilience and resolution), while also providing a deep insight into the effect of fundamental topological properties (*e.g.*, spectral gap).

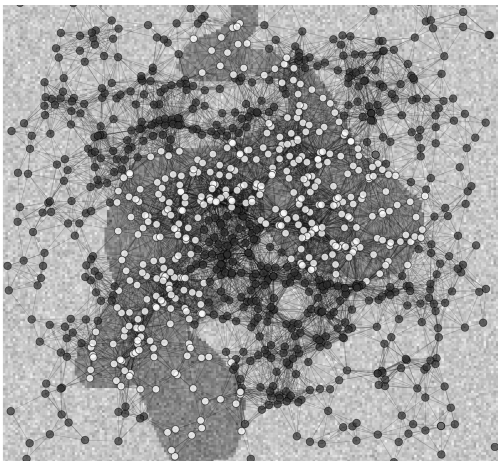


Fig. 1. A snapshot of our event region detector. The white nodes identify the presence of an oil spill (dark gray shape) out of a noisy environment (light gray background). The datasets are forecasts of the Deepwater Horizon oil spill [12]. A video showcasing the robustness of our method to node mobility and event dynamics is posted at http://youtu.be/oDyg4gO7Flo.

Our analysis has important consequences on the design of algorithms for event region detection. We propose decentralized and asynchronous algorithms that do not require location coordinates, can track fast changing signals, are little affected by mobility, can handle 3D topologies, and are simple to implement. To the best of our knowledge these are the first algorithms that possess *all* these traits.

A. Related Work

Given the rich literature in the area of event region detection, we focus our discussion on the most relevant and recent results in the field. Distributed boundary detection has two main research thrusts: the identification of network boundaries [13, 14] and of events and their boundaries [4–11]. While in the first the aim is to solely identify the nodes at the edge of the network, in the case of events, an algorithm must first detect the event region *within* the network. The two approaches can be seen as complementary. Once event regions are detected, network boundary algorithms can be used to enhance the identification of boundaries. The focus of this work is on event detection.

978-1-4799-3360-0/14$31.00 © 2014 IEEE

Even though region detection has been studied comprehensively on static networks, previous approaches are incompatible with mobility. This is due to three common assumptions:

Computation. Some studies assume that the network has a hierarchical structure [4] or that it can offload its data to a central server, where the event detection is performed [5]. Hierarchical approaches suffer from scalability problems and are not amenable to frequent changes in the network topology. Clearly, a local model of computation is preferable.

Location. The majority of previous algorithms require the geographical coordinates of nodes [6, 10, 11, 15]. Obtaining/maintaining location information is possible but incurs a high and recurring overhead, and introduces imprecision. Hence, we advocate a location-free approach, where algorithms exploit network connectivity to infer spatial correlation, such as [5, 7–9] (these location-free algorithms however suffer from either the *computation* or the *stationarity* constraint). If localization services are available, our algorithms additionally localize events.

Stationarity. We have found only a handful of location-free approaches that are local [7–9]. Nevertheless, they assume that the events are either stationary [7, 8] or change slowly [9]. The stationarity assumption is a problem not only because real events are time varying, but also because, when the position of nodes changes over time, nodes observe different events at different times and need to adapt promptly. Our analysis and simulations show that our algorithms can track fast changing signals and that they degrade gracefully with mobility.

B. Contributions.

Our work advances the area of event region detection by providing a novel perspective – the use of spectral graph theory to analyze and develop simple distributed algorithms. Our main research contributions are: (*i*) We propose a spectral framework to reason about event region detection (Section III). Our framework captures the fundamental properties of detectors, as well as the influence of the network topology. (*ii*) To benchmark our algorithms with respect to state-of-the-art Laplacian-based detectors, we modify two established algorithms used in image processing to work in wireless networks – the Laplacian of a Gaussian (LoG) and Difference of Gaussians (DoG) [16] (Section IV-A). The resulting detectors are location-free, scalable, suitable for 3D topologies, and simple to implement; but they have low resolution and can not operate in mobile networks with dynamic events. (*iii*) To overcome the limitations of LoG and DoG, we introduce two novel algorithms based on information potentials [17], the Laplacian of a Potential (LoP) and the Difference of Potentials (DoP) (Section IV-B). Our analysis and simulations, based on complex oil spill traces, validate the superior resolution of DoP, as well as its robustness to network and event dynamics (Section V). (*iv*) To validate the simplicity of our algorithms, we implement DoP on the Contiki OS and evaluate it on a testbed with 105 nodes (Section V).

II. Preliminaries

This section describes how the Laplacian identifies event regions. Algorithms that use the Laplacian reason about spatial events based on the network connectivity. As such, they are location-free.

Signal curvature. As exemplified in Figure 1, the primary objective of an event region detector is to annotate each node as being part of an event region or not. The problem input is a real-valued signal x defined on the n nodes of a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Sharp signal transitions in x are used to identify region boundaries. These sharp signal transitions occur on nodes over which the magnitude of the first derivative of x is maximized with respect to their neighborhood. This process is equivalent to finding "inflection point nodes" where the second derivative of x is zero. Inflection points can be computed locally using the combinatorial Laplacian[1]

$$(Lx)(u) = \sum_{v \sim u} \left(x(v) - x(u)\right),$$

where $v \sim u$ denotes node adjacency and, as is common in matrix notation, $x(u)$ is the value of x on node u. Vector Lx is a measure of curvature which is always positive in concave regions, negative in convex regions, and zero in inflection points. After the Laplacian is computed, nodes can locally decide whether they belong to an event region or not by following a simple algorithm,

---

if $(Lx)(u) > 0$ **then**
  if $\exists v \sim u$, $(Lx)(v) < 0$ **then** *interior boundary*.
  else *within event region*.
  end if
else if $(Lx)(u) < 0$ **then**
  if $\exists v \sim u$, $(Lx)(v) > 0$ **then** *exterior boundary*.
  else *outside event region*.
  end if
else if $\forall v \sim u$, $(Lx)(u) = (Lx)(v)$ **then** *inconclusive*.
else *boundary*.
end if

Due to convention, we refer to regions with positive (negative) Laplacian values as *events* (*non-events*). Region *boundaries* are inflection points that separate a region from its surroundings. Notice that, due to the discretization of the underlying space, Lx might go from positive to negative values without crossing zero. The above algorithm thus also accounts as boundaries the nodes within/outside an event region with at least one neighbor outside/within the region.

It is important to remark that L is a local operator that is robust to incomplete information. Even though the computation of $(Lx)(u)$ assumes the values of all neighbors, if – *e.g.*, due to message loss – some values are missing, the partially computed quantity is still a good estimator of $(Lx)(u)$. What's more, in contrast to threshold-based approaches [7], the Laplacian requires no preconceived knowledge of the signal range.

Diffusion kernels. The Laplacian has two main pitfalls. First, it is very sensitive to noise; it cannot differentiate between

[1]Formally, the Laplacian is the discrete and negative version of the Laplace-Beltrami operator and it is equal to the negative trace of the Hessian matrix, which contains the partial second derivatives of x.

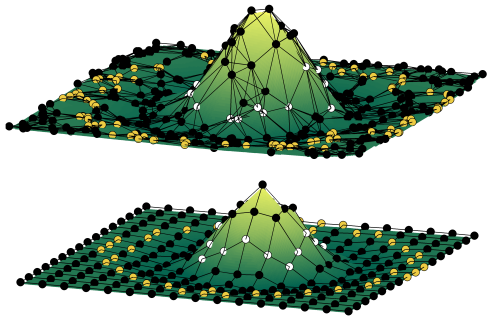


Fig. 2. The impulse response of a $\mathbf{LoG}_{10}$ kernel for a grid topology (bottom) versus a random geometric topology (top). The boundaries of the smallest and largest detectable regions (shown with white and orange markers, respectively) are complex and depend on the topology.

sharp signal transitions and those attributed to noise. Second, it suffers from a "flat-region ambiguity problem". If the signal stays constant over some region, it is impossible to decide locally whether the region is part of an event or not.

Diffusion is the prevalent approach to overcoming these problems. Before the computation of curvature the signal is diffused with a kernel $\mathbf{K}_\circ$, where $\circ$ is a parameter that controls the kernel's width. Diffusion kernels disseminate the information in the signal x over multiple hops to reveal underlying curvature trends. They smooth out sudden signal transitions and make flat regions curved.

Most diffusion-based algorithms can detect event regions over a range of scales. The choice of parameter $\circ$ controls the size of detectable regions. In general, the more a signal is diffused, the larger the size of regions that can be detected. If an event is larger than the largest detectable region, multiple events are detected. Conversely, events smaller than the smallest detectable region are not recognized. Figure 2 depicts the size of detectable events for an example LoG kernel (described in Section IV) when presented with an *impulse signal* $\delta_u(v) = 1$ if $u = v$ and $\delta_u(v) = 0$ otherwise. The smallest region is the induced subgraph that has as boundary the inflection points of the impulse response (white inner nodes). The largest region is the induced subgraph that has as boundary the out-most nodes with non-zero response (orange outer nodes). The exact influence of the network topology on the event signal x and the kernel $\mathbf{K}$ is encoded in the spectrum of L and is non-trivial. The analysis of these relations is one of the main contributions of our work and is described next.

III. Spectral Response

Studying event region detection algorithms based on their impulse response – see Figure 2 – only provides insight on how an algorithm operates in a specific topology. This section generalizes the approach by proposing a theoretical framework that decouples event detectors from the topology. By studying the spectral response of a method, we are able to quantify basic detection properties, such as its resilience to noise and its detection resolution[2].

[2]A high resolution facilitates the detection of fine-grained details at the event's boundary.

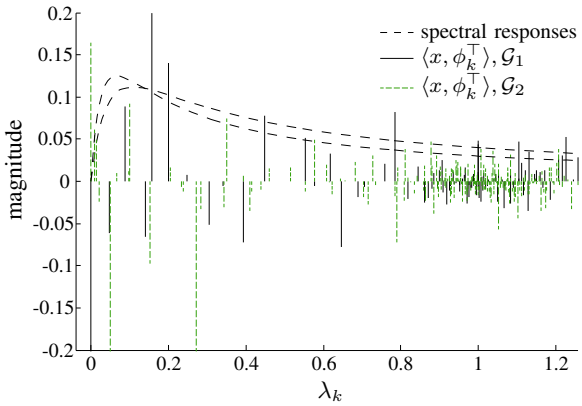


Fig. 3. Signal components (vertical bars) of two random geometric graphs, with $n_1 = 100$ and $n_2 = 200$, for two algorithmic responses (dashed lines). The topology affects the decomposition of the signal (height of bars) and the point of the algorithm's response each component corresponds to (horizontal position of bars).

A. Spectral response

The spectral response expresses the output of an algorithm given a spectral input. It is a Fourier-like formalism, meant for algorithms which operate on signals defined on graphs. We start by providing formal definitions, and then, continue to analyze the impact of the topology. The section concludes with an illustrating example.

Consider the n-dimensional vector space spanned by the eigenvalues λ and eigenfunctions ψ of a generalized Laplacian matrix [18]. According to Parseval's identity and because the eigenfunctions are an orthonormal basis of $\mathbb{R}^n$, any signal x in the same space can be decomposed into exactly n components, each corresponding to an eigenfunction. The *signal decomposition* of x is then defined as

$$x = \sum_{k=1}^{n} \left\langle x, \psi_k^\top \right\rangle \psi_k, \tag{1}$$

where $\langle\,\rangle$ expresses the inner product.

Definition 1 (spectral response)**.** *We say that an algorithm with kernel* $\mathbf{K}$ *has* spectral response $r : \mathbb{R} \to \mathbb{R}$ *if its output can be defined in the spectral form*

$$\mathbf{K}x = \sum_{k=1}^{n} r(\lambda_k) \left\langle x, \psi_k^\top \right\rangle \psi_k. \tag{2}$$

Let us examine this definition and provide initial intuition on how an algorithm operates in the spectral domain. Kernel $\mathbf{K}$ essentially alters a signal x by attenuating or amplifying its signal components $\left\langle x, \psi_k^\top \right\rangle \psi_k$. Each *signal component*, which is a vector in $\mathbb{R}^n$, captures the projection of x on ψ_k with magnitude $\left\langle x, \psi_k^\top \right\rangle$. The spectral response therefore decides how relevant each signal component of x is.

The effect of topology. The network topology affects the algorithmic output in two ways. *First, topology impacts the signal decomposition*. The basis functions ψ_k over which the signal is decomposed are particular to the graph topology. In other words, different graphs decompose a signal in different ways.

This decomposition will be studied further in Section III-B. *Second, topology impacts the signal response*. The eigenvalues λ_k map signal components to the spectral response's domain. How much each signal component $\langle x, \psi_k^\top \rangle$ is affected depends on the corresponding eigenvalue λ_k. Intuitively, in graphs with a small spectral gap (sparse graphs), the left side of r plays a more significant role.

The above two observations are illustrated in Figure 3 for two sample random geometric graphs, $\mathcal{G}_1$ (in black) and $\mathcal{G}_2$ (in green). The impact on *signal decomposition* is captured by the green and black vertical bars, corresponding to the n_1 and n_2 signal components, respectively. The horizontal position of the bars depends on their corresponding eigenvalues, and their height on $\langle x, \psi_k^\top \rangle$. In this example, we also show two sample spectral responses (dashed lines). The impact of these particular responses is captured by the higher weight given to signal components that correspond to eigenvalues close to 0.08 and 0.1, respectively. The magnitude of the k-th component in Formula 2 is then computed by multiplying the k-th bar with the value of the response function that lies on top of if, *i.e.*, that has the same eigenvalue.

B. Event detection responses

A complex event region can be described by the union of simpler regions of different sizes. Large regions compose the event's main body and small regions the details of its shape (or perhaps noise). From this point of view, an event region detector is an algorithm that distinguishes the simple regions of a complex event and filters them according to their scale (*i.e.*, favors large over small regions).

In the following, we show that the spectral form inherently captures such a decomposition. We explain that each eigenfunction partitions a graph into a different number of subgraphs (nodal domains). This partitioning induces scale: the higher the order of the eigenfunction, the smaller the size of the regions. The magnitude of the k-th signal component then quantifies the overlap between the signal and the subgraphs partitioned by the k-th eigenfunction.

The regions of eigenfunctions. A weak *nodal domain* $\mathcal{D}_{\psi_k} \subseteq \mathcal{V}$ is the maximally induced subgraph over which ψ_k does not change sign [18]. We argue that these nodal domains can be interpreted as "the regions of eigenfunctions". That is because, amongst all possible regions x, the regions x^* that maximize the magnitude of the k-th signal component are the ones that precisely overlap one of the nodal domains of ψ_k. Formally,

$$\text{if } x^* = \underset{x \in \mathbb{R}^n,\, \|x\| \leq \beta}{\arg\max} \; | \langle x, \psi_k^\top \rangle | \text{ then } \mathrm{E}[|\mathcal{D}_{\psi_k}|] = \mathrm{E}[|\mathcal{D}_{x^*}|],$$

where $\mathrm{E}[|\mathcal{D}_{\psi_k}|] = n/\mathfrak{W}(\psi_k)$ is the expected size of a nodal domain, $\mathfrak{W}(\psi_k)$ is the number of nodal domains of ψ_k, and β is a positive real number that makes sure that x^* cannot grow infinitely large.

We exploit this property to associate a minimum size to each signal component. According to the discrete nodal domain theorem [19], $\mathfrak{W}(\psi_k) \leq k$. Therefore,

$$\mathrm{E}[|\mathcal{D}_{x^*}|] = \mathrm{E}[|\mathcal{D}_{\psi_k}|] \geq \frac{n}{k}, \tag{3}$$

which means that, in expectation, the size of the *eigenfunction regions* (and of x^*) is bounded and decreases with k.

Region variance. It is important to show that all the nodal domains of an eigenfunction have similar sizes, otherwise they would not capture scale. For the random walk matrix P, which we are mainly interested in, the evidence that $\mathrm{Var}[|\mathcal{D}_{\psi_k}|]$ is small is a consequence of graph partitioning [20]. Recall that, according to the *Ncut* problem, partitioning a graph into k subgraphs is equivalent to finding the subgraphs $\mathcal{G}_1, \ldots, \mathcal{G}_k$ that minimize

$$\mathit{Ncut}(\mathcal{G}_1, \ldots, \mathcal{G}_k) = \sum_{i=1}^{k} \frac{|\{uv \in \mathcal{E} : u \in \mathcal{V}_i \text{ and } v \in \bar{\mathcal{V}}_i\}|}{|\mathcal{V}_i|},$$

with $\bar{\mathcal{V}}_i$ being the complement of $\mathcal{V}_i$. Since each cut is normalized by the size of the partition, the objective function penalizes uneven partitions and $\mathrm{Var}[|\mathcal{V}_i|]$ is small. In their seminal work, Shi and Malik showed that, for the relaxed *Ncut* problem, $\mathcal{G}_i = \mathcal{D}_{\psi_k}^{(i)}$, where i iterates over the partitions and the nodal domains [21]. Therefore, $\mathrm{Var}[|\mathcal{D}_{\psi_k}|]$ is also small.

IV. Local Algorithms for Event Region Detection

As previously discussed, the Laplacian needs to be preceded by a diffusion kernel to overcome its sensitivity to noise and its ambiguity of flat regions. In this section, we present two types of diffusion methods. The first, which we call *heat-based*, are derived from image processing techniques and are used as benchmarks. These algorithms exhibit very good resilience to noise, but suffer from two significant drawbacks: they require node synchronization, which limits their use in dynamic settings, and have low detection resolution. The second type use information potentials as a diffusion primitive [17]. *Potential-based methods* are asynchronous, have high resolution and are not significantly affected by noise.

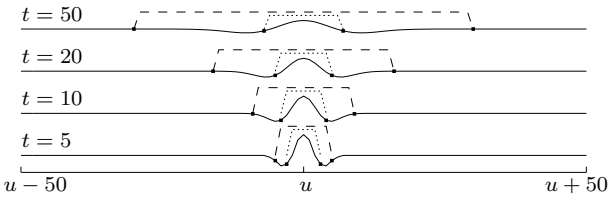
A. Heat-based methods

Heat-based methods use heat diffusion to eliminate sharp signal transitions, such as noise. We start by discussing the fundamentals of heat diffusion and then present two heat-based algorithms for event region detection, the Laplacian of a Gaussian (LoG) and the Difference of Gaussians (DoG). Since both algorithms are graph-counterparts of well-known algorithms for image processing [16], our presentation focuses mainly on the effect of the topology and on their decentralized implementation.

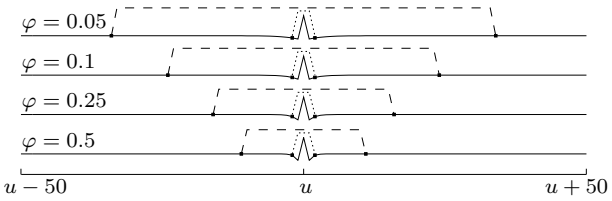
Heat-diffusion kernel. The diffusion of a signal x after t time-steps is given by the heat kernel $\mathbf{H}_t := e^{-t\mathcal{L}}$, where $\mathcal{L}$ is Chung's normalized Laplacian [22]. Equivalently,

$$\mathbf{H}_t x = \sum_{k=1}^{n} e^{-t\lambda_k} \langle x, \phi_k \rangle \phi_k, \tag{4}$$

where t is a user defined parameter and, λ and ϕ are the eigenvalues and eigenfunctions of $\mathcal{L}$. Parameter t controls how much a signal is diffused. Heat diffusion eliminates the high-order components of a signal and imposes a restriction on the



(a) LoG



(b) LoP

Fig. 4. LoG and LoP impulse responses on a 100 node path-graph. The t and φ parameters determine the smallest (dotted lines) and largest (dashed lines) event regions the two algorithms detect.

minimum size of detectable event regions. In short, as t grows, "small regions" caused by the event itself or by noise will be filtered out.

Contrary to image processing applications, which are usually solved in a single computer, we need an efficient and distributed computation of $\mathbf{H}_t x$ for large-scale wireless networks. Fortunately, heat diffusion is approximated by a simple decentralized algorithm in which nodes iteratively average their neighbor's values for t synchronous computation rounds.

$$\left(\mathbf{H}_t x\right)(u) \leftarrow \sum_{v \sim u} \frac{\left(\mathbf{H}_{t-1} x\right)(v)}{d_u} \text{ and } \left(\mathbf{H}_0 x\right)(u) = x(u). \quad (5)$$

Above d_u is the degree of u. Expressed in spectral form,

$$\tilde{\mathbf{H}}_t x = \sum_{k=1}^{n} (1-\lambda)^t \left\langle x, \psi_k^\top \right\rangle \psi_k, \quad (6)$$

where ψ_k is the k-th eigenfunction of the well-known random walk matrix P. The difference with Formula 4 matters little because $\psi_k = D^{-1/2}\phi_k$ and the change from $e^{-\lambda_k}$ to $(1-\lambda)^t$ is due to the discretization of the euclidean space. Thus, from now on when we refer to $\mathbf{H}_t$ we mean $\tilde{\mathbf{H}}_t$.

Laplacian of a Gaussian. We now present our first detection method. LoG entails computing the Laplacian of a signal that has been convolved with a heat kernel. LoG is very robust to noise. By first smoothing a signal appropriately, only transitions that correspond to significant trends are identified. Furthermore, flat regions become either convex or concave, which eliminates the *detection ambiguity* problem. For any signal x,

$$\mathbf{LoG}_t x := L\mathbf{H}_t x. \quad (7)$$

For a given t, this algorithm terminates in $t+1$ rounds and after the network exchanges $2\left|\mathcal{E}\right|(t+1)$ messages, each containing two numerical fields, the node id and the current average, that is $msg_t(u) = [u_{id}, (\mathbf{H}_t x)(u)]$.

The kernel of LoG is given by

$$\mathbf{LoG}_t(u,v) = \sum_{k=1}^{n} (1-\lambda_k)^t L\psi_k(u)\psi_k(v). \quad (8)$$

Figure 2 shows the impulse response for two sample topologies. In a *regular grid* (which is the case for images), LoG approximates a Mexican hat. The smallest and largest event regions LoG detects are discs with radius $\sqrt{t}+1$ and $t+2$, respectively. The effect of t on the size of detectable regions is more clearly seen in a *path graph*, such as the one depicted in Figure 4a. In *non-regular topologies*, the kernel exhibits more complex geometry. We can have a better understanding of how LoG works in different graphs though its spectral response. But to do that, we have to first express LoG in a spectral form. From definition,

$$\begin{aligned} \mathcal{L}\phi_k &= \lambda_k \phi_k \\ D^{-1/2} L D^{-1/2} \phi_k &= \lambda_k \phi_k \\ L D^{-1/2}(D^{1/2}\psi_k) &= \lambda_k D^{1/2}(D^{1/2}\psi_k) \\ L\psi_k &= \lambda_k D \psi_k. \end{aligned}$$

Substituting this in Formula 8, we get LoG's response

$$r_{LoG_t}(\lambda_k) = \lambda_k (1-\lambda_k)^t, \quad (9)$$

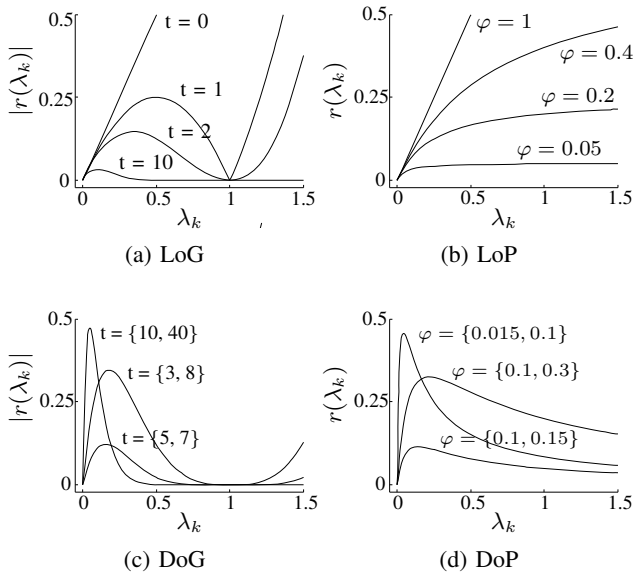
where the eigenfunctions of the spectral form are normalized by the degree matrix D. The spectral response is shown in Figure 5a for representative values. Observe that, irrespectively of parametrization, LoG removes the signal's DC-offset (*i.e.*, the first signal component). This is a common characteristic of all detectors and is essential for the identification of curvature. Furthermore, as t increases, signal components with large eigenvalues – and therefore small event regions – are progressively attenuated up to a point. The sudden increase after $\lambda_k = 1$ is insignificant as rarely $\lambda_k > 1.2$ and only occurs for small t. In Section IV-C we study LoG's spectral response further and compare it with the other algorithms presented in this paper.

Difference of Gaussians. In LoG, the size's range of detectable events is controlled by t, which is a single parameter. We now examine a method where the size of the smallest and largest events are determined independently.

Instead of using the Laplacian directly, we estimate it by subtracting two heat kernels of different widths

$$\mathbf{DoG}_{\{t_l,t_h\}} x := (\mathbf{H}_{t_l} - \mathbf{H}_{t_h})x, \quad (10)$$

with parameters $0 \leq t_l < t_h$. Intuitively, DoG operates as a bandpass filter. The parameter t_l controls how much noise is removed from the signal, while t_h eliminates very slow changing signal components. The information that lies in between the two diffused signals is preserved. When the difference between t_l and t_h becomes small, DoG approximates LoG. Moreover, DoG has similar message complexity to LoG. Since the heat kernels can be computed jointly, the algorithm



(a) LoG (b) LoP

(c) DoG (d) DoP

Fig. 5. Spectral responses for representative parameters. Since heat-based methods can become negative for $\lambda_k > 1$ (*e.g.*, $\mathbf{LoG}_1, \mathbf{DoG}_{\{3,8\}}$), we facilitate presentation by showing their absolute spectral response.

terminates in t_h rounds and after the exchange of $2\,|\mathcal{E}|\,t_h$ messages, with messages of type $msg_t(u) = [u_{id}, (\mathbf{H}_{t_h} x)(u)]$.

The spectral response of DoG is

$$r_{DoG_{\{t_l,t_h\}}}(\lambda_k) = (1-\lambda_k)^{t_l} - (1-\lambda_k)^{t_h} \tag{11}$$

and is shown in Figure 5c for representative parameter values.

The main benefits of DoG are its flexibility and high resilience to noise. As we see in its spectral response, DoG not only controls the smaller and larger sizes of detectable events independently (flexibility), but is also more aggressive in attenuating signal components (resilience). For this reason DoG generally outperforms LoG. On the flip side, choosing two parameters is more challenging than one. If the parameters are incorrectly chosen, DoG may perform worse than LoG.

B. Potential-based methods

LoG and DoG are efficient algorithms but they depend on synchronous rounds and do not tolerate dynamics (they assume *stationarity*). This is because the heat kernel is not an attractor (*i.e.*, not a steady state) of difference equation (5), but a transient state. Asynchrony and dynamics introduce disturbances that alter the system's trajectory through the state space. The disturbed and original trajectories are not guaranteed to share transient states. The synchronous constraint is not unique to LoG and DoG, but it is shared by all heat-based methods, which renders them of little use in most dynamic wireless networks. To overcome this major drawback, we propose a novel set of asynchronous kernels based on information potentials [17]. Since the potential kernel is an attractor and not a transient state, potential-based methods are not significantly affected by dynamics (*e.g.*, due to mobility or due to the change of the monitored signal) and are asynchronous.

Potential kernel. The potential kernel $\mathbf{P}_\varphi$ stems from the generalization of the decentralized algorithm for computing the heat diffusion kernel $\mathbf{H}_t$. The potential of a signal x is the steady state $\mathbf{P}_\varphi x = \lim_{t\to\infty} y_t$ of the iterative algorithm

$$y_{t+1}(u) \leftarrow (1-\varphi) \sum_{v\sim u} \frac{y_t(v)}{d_u} + \varphi x(u), \tag{12}$$

where φ is a user-defined parameter. For $\varphi = 0$ and $y_0 = x$, the above is exactly the heat kernel. When $\varphi \in (0, 1]$, the system reaches the steady state

$$\mathbf{P}_\varphi x = \sum_{k=1}^{n} \left(\frac{1-\varphi}{\varphi} \lambda_k + 1 \right)^{-1} \langle x, \psi_k \rangle \, \psi_k \tag{13}$$

independently of y_0. As shown by Loukas *et al.* [17], the algorithm convergences ϵ-close to the steady state in $\varphi^{-1} \log(c/\epsilon)$ rounds, where c is a constant that depends on the topology and is small. Furthermore, contrary to heat-based methods, information potentials have been shown to converge under very challenging conditions, such as (*i*) asynchronous operation, (*ii*) partial information (only using 25% of neighborhood information at each computation round), and (*iii*) node mobility. These characteristics make potential-based methods well suited for event detection in mobile large scale networks.

Laplacian of Potential. The LoP method computes the Laplacian after the signal has been smoothed by a potential kernel

$$\mathbf{LoP}_\varphi x := L\mathbf{P}_\varphi x.$$

Its spectral response

$$r_{LoP_\varphi}(\lambda_k) = \frac{\lambda_k}{\frac{1-\varphi}{\varphi}\lambda_k + 1} \tag{14}$$

is shown in Figure 5b.

We can gain an initial intuition on how potential-based methods compare to heat-based methods by comparing the impulse response of LoP and LoG. Figure 4b depicts the responses for a simple path graph of 100 nodes. Unlike LoG, in Figure 4a the central lobe (convex region close to u) of LoP's impulse response is narrow and always consists of u and it's neighbors. As a consequence, LoP tends to favor abrupt signal transitions and is more sensitive to noise. Beyond the central lobe, the response fades outward with a rate that depends on φ. As φ decreases, the rate also decreases and the response widens. In the continuous domain, the radius of the largest disc LoP can detect (depicted in dashed lines) is approximately $14\sqrt{(1-\varphi)/\varphi}$.

As seen by its spectral response in Figure 5b, LoP is susceptible to the presence of small regions, which occur around $\lambda_k = 1$. This helps in capturing fine-grained boundaries (resolution) but renders the method less resilient to noise. In signals with little noise, LoP outperforms LoG and exhibits a similar performance to DoG and DoP (explained next).

Difference of Potentials. Our last detection method exploits the idea of subtracting two potential kernels of different width

$$\mathbf{DoP}_{\{\varphi_l,\varphi_h\}} x := \left(\mathbf{P}_{\varphi_h} - \mathbf{P}_{\varphi_l}\right) x, \tag{15}$$

where $0 < \varphi_l < \varphi_h \leq 1$ are user defined parameters. A network gets ϵ-close to the steady state after the exchange of $2\varphi^{-1}\log(c/\epsilon)|\mathcal{E}|$ messages of format $msg_t(u) = [u_{id}, (\mathbf{P}_{\varphi_l}x)(u), (\mathbf{P}_{\varphi_h}x)(u)]$.

Unlike LoP, DoP is more resilient to noise. Decreasing φ_l widens its central lobe and, as a consequence, the smallest event region that it can detect.

DoP's spectral response is

$$r_{DoP_{\{\varphi_l,\varphi_h\}}}(\lambda_k) = \frac{\lambda_k(\varphi_h - \varphi_l)}{(\varphi_l(1-\lambda_k)+1)(\varphi_h(1-\lambda_k)+1)}. \quad (16)$$

Similar to DoG, DoP provides flexibility for parametrization, as well as resilience to noise. However unlike the first, DoP also has good resolution. That is because its spectral response does not completely eliminate small events, but only decreases their importance. In Section V we will see that, in most scenarios, DoP outperforms all other algorithms.

C. Comparison

First, it is important to mention that all four detectors eliminate the flat-region ambiguity problem. By removing the signal component that corresponds to λ_1, detectors eliminate a signal's DC-offset. The sign of the resulting signal reveals whether a region is convex or concave. It is also important to note that, our detectors work in 3D as well as in 2D topologies. As reported by Chung *et al.* [22], Laplacian and heat-based algorithms can be used for graphs embedded in any dimension.

We next compare the resilience to noise and resolution of the different methods. To have a better understanding of this comparison it is important to recall that: (*i*) eigenvalues are rarely higher than 1.2, and (*ii*) eigenvalues around 1.0 capture high-order events (small regions).

Resilience. This is the ability of a detector to attenuate small events, which can be caused by noise. As we saw in Section III, the expected size of event regions described by the k-th signal component is larger than n/k. Therefore, noise will be mainly decomposed into high order components. A method is thus resilient to noise if its spectral response decreases sufficiently fast after its peak response $\left(\lambda_{peak} = \arg\max_{\lambda_k} r(\lambda_k)\right)$. In fact, we can order detectors with respect to their resilience by looking at the magnitude of $\partial r/\partial\lambda_k$ for $\lambda_k > \lambda_{peak}$.

$$\text{DoG} > \text{DoP} > \text{LoG} > \text{LoP}. \qquad \text{(resilience)}$$

The derivation is left out as it easily follows from Formulas (9), (11), (14), and (16). Summarizing, due to their sharp responses, DoG and DoP are better at removing noise.

Resolution. The resolution of a detector depends on its ability to capture details, which are encoded in high order signal components (but usually not as high as noise). Since in most graphs high order signal components are concentrated around $\lambda_k = 1$, we quantify the resolution of a detector through the ratio of its response at $\lambda_k = 1$, normalized to its peak response, $\lim_{\lambda\to 1} r(\lambda)/r(\lambda_{peak})$. The resulting ordering is

$$\text{LoP} > \text{DoP} > \text{DoG} \approx \text{LoG}. \qquad \text{(resolution)}$$

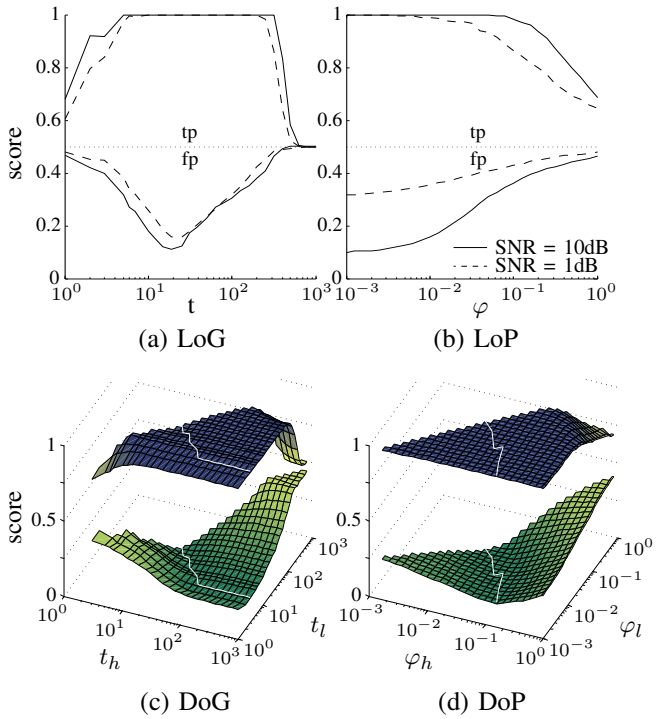


Fig. 6. Exhaustive comparison of event detectors for a simple grid topology and a star-shaped event region. Each graph depicts the true positive rate (tp – top curves/surfaces) versus the false positive rate (fp – bottom curves/surfaces). The white line in the bottom two figures annotates the best performing parameter combinations.

Notice that a method with high resilience is not necessarily characterized by low resolution. DoP attains both by having a sharp response near λ_{peak} that flattens out for higher λ_k, but still decreasing monotonically. Heat-based methods on the other hand suffer from low resolution; their response is always zero at $\lambda_k = 1$. When no noise exists, LoP consistently outperforms the three other algorithms.

V. Evaluation

Our evaluation consists of three parts. First, we compare the performance of the proposed algorithms with respect to their resilience, resolution, and sensitivity to parametrization. To achieve a systematic comparison, this part ignores the effect of irregular topologies. We then focus on signal and network dynamics and evaluate their effect on detection. Last, we perform a testbed experiment which demonstrates the feasibility and simplicity of our approach.

Metrics. We quantify detection accuracy in terms of the true positive and false positive scores (tp, fp). The true positive score measures the proportion of event points (nodes) which are correctly identified. The false positive score expresses the proportion of non-event points which are incorrectly identified as part of an event region. Ideally, tp = 1 and fp = 0.

Detection performance. Our first experiment considers a grid topology of 400 nodes and a star-shaped event region corrupted with additive Gaussian noise. The star event occupied approximately 25% of the area of the network and it

was positioned at its center. Notice that the choice of shape influences the comparison. While methods with high resilience perform better with simple events (such as circles), methods with high resolution favor events with complex boundaries. The star is a good benchmark because it sits in the middle of these extremes. The simulations were performed in MATLAB.

The results – see Figure 6 – confirm the higher resilience but lower resolution of heat-based methods (as per our analysis). This is particularly noticeable in subfigures 6 (a-b). While for low SNR LoG outperforms LoP, the accuracy of the second is higher when SNR=10 dB. For DoG and DoP the trend is slightly different. Subfigures 6 (c-d), which have an SNR=1 dB, show that DoP is better than DoG. This is because a noisy star event favors DoP slightly. For coarser-shaped events (not shown due to lack of space), DoG performs better than DoP. *Also observe that potential-based methods are less sensitive to parametrization.* As seen in subfigure (b), LoP's accuracy consistently improves as φ decreases. This is because the size of the smallest regions it detects is independent of φ. DoP's performance on the other hand degrades when the parametrization is suboptimal, but – unlike DoG – it does so gracefully. As long as the size of the largest identifiable region stays larger than the size of the star ($\varphi_l < 0.1$) all of the event points are recognized. False positives occur when the size of the smallest identifiable event region becomes smaller than the size of noise regions ($\varphi_h > 0.1$).

Dynamics. We next examine how our best detector (DoP) performs in more challenging scenarios. We pose two main challenges: node mobility and signal dynamics. In addition, we use complex events. The test events are digitized versions of the Deepwater Horizon oil spill forecast obtained from ten numerical models [12] and contain regions of various sizes and detail. Figure 1 shows a snapshot of a simulation run. Due to their variability, the accurate detection of oil spills demands both resolution and resilience over a large range of scales. As is the case in most real-world scenarios, we selected values according to the guidelines of Figure 6 and not using exhaustive search. In our simulations, $n = 1000$, the node degree varied between 1 and 78, the network diameter was between 14 and 16, and the SNR = 5 dB[3].

Mobility. We simulated node mobility using the random waypoint model. Figure 7 compares the online performance of DoP to that obtained when solving the problem offline for each network snapshot (offline DoP and DoG). For each algorithm, we plot the quartiles of tp and fp scores over 10 runs. *Overall, DoP outperformed DoG.* While DoG exhibited low performance due to its low resolution, DoP detected the oil spills accurately, even for high speeds. *Mobility undoubtedly introduces a transient error that depends on speed.* The phenomenon occurs because nodes do not realize that they enter or exit an event region until they re-converge ϵ-close to the new steady state. *Nevertheless, DoP's accuracy deteriorates slowly as the speed increases.* This is attributed to two main reasons. First, while convergence requires $1/\varphi_l \log(c/\epsilon)$ rounds, ϵ should not necessarily be small. It is sufficient that the sign of $(\mathbf{DoP}_\varphi)(u)$ changes. Second, mobility acts as a spatial low-pass filter that reduces noise. For nodes that remain within the same region (event or non-event), the likelihood that a node is affected by noise is smaller when the node is moving.

[3]A visual illustration of the topology and of the dynamics in question can be found in the video mentioned in the caption of Figure 1.

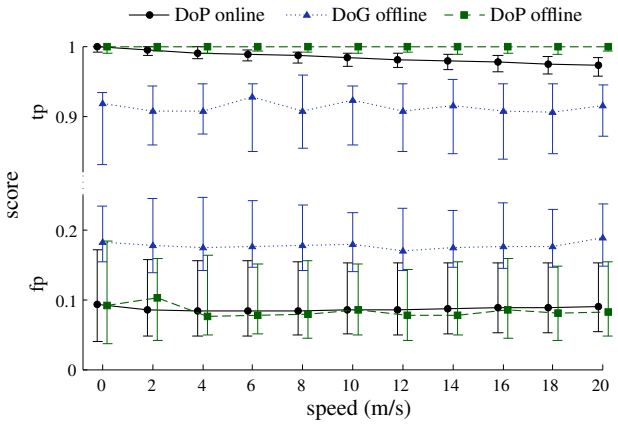


Fig. 7. Detector accuracy versus node speed in the random waypoint mobility model. Despite the mobility, DoP's online performance is similar to the offline solution. Both are significantly better than DoG. Test signals were generated from the Deepwater Horizon oil spill dataset [12].

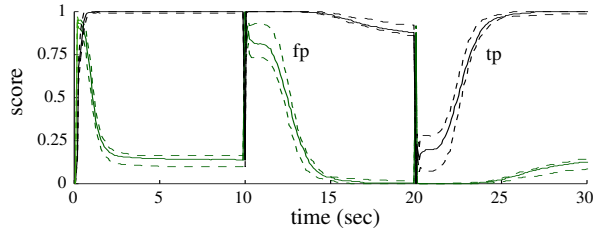


Fig. 8. Potential-based detection methods are very robust to signal dynamics. To demonstrate, we invert the signal at $t = 10$ and $t = 20$ seconds. The median (line) as well as the min and max scores (dashed lines) show how DoP quickly overcome the event region inversion.

Signal dynamics. To evaluate how potential-based method cope with signal dynamics, we inverted the event and non-event regions ($x = -x$) twice, once at $t = 10$ and once more at $t = 20$ seconds. The inversion stresses the ability of a detector to adapt as it instantaneously cancels the validity of any previous detection. Figure 8, shows the min, median, and max scores across time. *We can see that DoP quickly overcomes any dynamics induced by the signal.* Even in the worst case, the steady state is reached after about 10 seconds. Similar performance was achieved by LoP.

Empirical results. We evaluated DoP in a testbed of 105 resource constrained devices (MSP430 micro-controller, CC1101 radio), deployed on the ceiling of our building at TU Delft. DoP was implemented in Contiki OS, over a simple asynchronous MAC protocol (NullMAC). For the duration of each computation round (0.5 second), each device transmitted an average of 3.5 packets. We simulated the sampling process, by constructing a virtual signal and then assigning to each node the value underneath it – see Figure 9. We repeated the experiment for three parametrizations, two noise levels, and five different signals. DoP's median (tp, fp) scores are summarized in the following table.

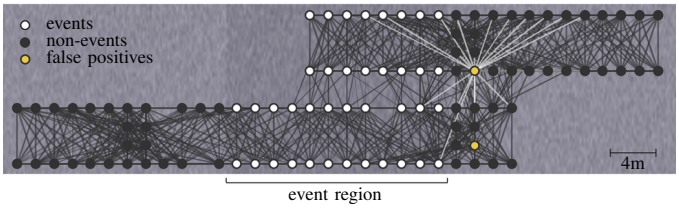


Fig. 9. Graphical depiction of the testbed's topology, annotated with the results of $\mathbf{DoP}_{\{0.1,0.2\}}$. Black/white nodes are the nodes that correctly identified the event/non-event (dark gray area in the middle). Orange nodes are false positives. In this instance, SNR=5 dB and the score is (1, 0.059).

SNR	$\mathbf{DoP}_{\{0.05,0.15\}}$	$\mathbf{DoP}_{\{0.1,0.2\}}$	$\mathbf{DoP}_{\{0.15,0.2\}}$
1 dB	(1.00, 0.14)	(0.95, 0.15)	(0.92, 0.18)
5 dB	(1.00, 0.08)	(1.00, 0.08)	(1.00, 0.09)

Observe that the above scores are similar to those achieved in simulations. *Our empirical evaluation not only showcases the ease of implementation of potential-based methods ($\approx$ 25 lines of code), but also their robustness to the vagaries of wireless communication*. Low-power wireless links are known to have high temporal and spatial variability, to be asymmetric and to have clustering effects. Furthermore, packet losses are commonplace and affect the performance of any protocol requiring one-hop information. We would like to note that most false positives occurred in the ambiguous zone near the boundary. Figure 9 depicts such an example. In the figure, both false positives were exterior boundaries with strong links toward the event region. As such, they were "pulled" to higher values by their neighbors within the event region.

VI. Conclusions

This paper proposed decentralized algorithms for detecting event regions on mobile networks. These are the first local, location-free detectors that do not assume signal stationarity. In addition, we provided novel insight on the problem of event region detection. We showed that, by examining how an event detector affects the signal decomposition in the spectral domain (*i.e.*, its spectral response), we decouple its effect from the network topology. Using this idea, we characterized the performance of our novel detectors (LoP and DoP) and compared them to established image processing algorithms (LoG and DoG). Our algorithms exhibit good resolution and resilience. They are also efficient, simple, and tolerant to mobility and signal dynamics. However, an important question is left open: "*What are suitable parameters for a given deployment?*". This paper provided general guidelines, but not an exact methodology for choosing parameters. Stricter theoretical guidelines might be possible if a better understanding of the spectrum of random (geometric) graphs is attained.

Acknowledgements. Andreas Loukas was supported by the Dutch Technology Foundation STW and the Technology Program of the Ministry of Economic Affairs, Agriculture and Innovation (D2S2 project). Jie Gao acknowledges support by NSF through grants DMS-1221339, CNS-1217823, CNS-1116881

References

[1] P. Barbosa, N. White, and N. Harris, "Wireless sensor network for localized maritime monitoring," in *Proc. IEEE AINA*, 2008.
[2] M. Duckham, D. Nussbaum, J.-R. Sack, and N. Santoro, "Efficient, decentralized computation of the topology of spatial regions," *IEEE Trans. Comput.*, vol. 60, no. 8, 2011.
[3] J. Jiang and M. Worboys, "Detecting basic topological changes in sensor networks by local aggregation," in *Proc. ACM SIGSPATIAL*, 2008.
[4] R. Nowak, U. Mitra, and R. Willett, "Estimating inhomogeneous fields using wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, 2004.
[5] M. Martalo and G. Ferrari, "Low-complexity one-dimensional edge detection in wireless sensor networks," *EURASIP J. Wireless Commun. and Netw.*, vol. 2, no. 11, 2010.
[6] K. Shih, S. Wang, H. Chen, and P. Yang, "CollECT: Collaborative event detection and tracking in wireless heterogeneous sensor networks," *ELSEVIER Computer Communications*, vol. 31, no. 14, 2008.
[7] G. Jin and S. Nittel, "Ned: An efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks," in *Proc. IEEE MDM*, 2006.
[8] A. Dogandzic and B. Zhang, "Distributed estimation and detection for sensor networks using hidden markov random field models," *IEEE Trans. Signal Process.*, vol. 54, no. 8, 2006.
[9] T.-Y. Wang and Q. Cheng, "Collaborative event-region and boundary-region detections in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 6, 2008.
[10] F. Li, J. Luo, C. Zhang, S. Xin, and Y. He, "UNFOLD: Uniform fast on-line boundary detection for dynamic 3d wireless sensor networks," in *Proc. ACM MobiHoc*, 2011.
[11] M. Ding and X. Cheng, "Robust event boundary detection in sensor networks-a mixture model based approach," in *Proc. IEEE INFOCOM*, 2009.
[12] Y. Liu, R. Weisberg, C. Hu, L. Zheng, Eos, and 2011, "Tracking the deepwater horizon oil spill: A modeling perspective," *EOS Trans. American Geophysical Union*, vol. 92, no. 6, 2011.
[13] Y. Wang, J. Gao, and J. Mitchell, "Boundary recognition in sensor networks by topological methods," in *Proc. ACM MOBICOM*, 2006.
[14] H. Zhou, H. Wu, and M. Jin, "A robust boundary detection algorithm based on connectivity only for 3d wireless sensor networks," in *Proc. IEEE INFOCOM*, 2012.
[15] Y. Yang, Y. Liu, Q. Zhang, and L. Ni, "Cooperative boundary detection for spectrum sensing using dedicated wireless sensor networks," in *Proc. IEEE INFOCOM*, 2010.
[16] M. Basu, "Gaussian-based edge-detection methods – a survey," *IEEE Trans. Syst., Man, Cybern.*, vol. 32, no. 3, 2002.
[17] A. Loukas, M. Zuniga, M. Woehrle, M. Cattani, and K. Langendoen, "Think globally, act locally: On the reshaping of information landscapes," in *Proc. ACM/IEEE IPSN*, 2013.
[18] T. Biyikoglu, J. Leydold, and P. F. Stadler, *Laplacian Eigenvectors of Graphs: Perron-Frobenius and Faber-Krahn Type Theorems*. Springer, 2007.
[19] E. B. Davies, G. M. L. Gladwell, J. Leydold, and P. F. Stadler, "Discrete nodal domain theorems," *Linear Algebra and its Applications*, vol. 336, no. 1-3, 2001.
[20] U. von Luxburg and U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, 2007.
[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000.
[22] F. Chung, A. G. Yan, and S. Yau, "Higher eigenvalues and isoperimetric inequalities on riemannian manifolds and graphs," *Communications on Analysis and Geometry*, vol. 8, no. 6, 2000.