Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 964564, 18 pages
http://dx.doi.org/10.1155/2015/964564

Research Article

Distributed Forest Fire Monitoring Using Wireless Sensor Networks

M. Ángeles Serna,[1] Rafael Casado,[2] Aurelio Bermúdez,[2] Nuno Pereira,[1] and Stefano Tennina[3]

[1]*CISTER/INESC TEC, ISEP, Polytechnic Institute of Porto, 4249-015 Porto, Portugal*
[2]*Computing Systems Department, University of Castilla-La Mancha, 02071 Albacete, Spain*
[3]*WEST Aquila s.r.l., University of L'Aquila, 67100 L'Aquila, Italy*

Correspondence should be addressed to Aurelio Bermúdez; aurelio.bermudez@uclm.es

Received 27 November 2014; Accepted 15 April 2015

Academic Editor: Andrei Gurtov

Disaster management is one of the most relevant application fields of wireless sensor networks. In this application, the role of the sensor network usually consists of obtaining a representation or a model of a physical phenomenon spreading through the affected area. In this work we focus on forest firefighting operations, proposing three fully distributed ways for approximating the actual shape of the fire. In the simplest approach, a circular burnt area is assumed around each node that has detected the fire and the union of these circles gives the overall fire's shape. However, as this approach makes an intensive use of the wireless sensor network resources, we have proposed to incorporate two in-network aggregation techniques, which do not require considering the complete set of fire detections. The first technique models the fire by means of a complex shape composed of multiple convex hulls representing different burning areas, while the second technique uses a set of arbitrary polygons. Performance evaluation of realistic fire models on computer simulations reveals that the method based on arbitrary polygons obtains an improvement of 20% in terms of accuracy of the fire shape approximation, reducing the overhead in-network resources to 10% in the best case.

1. Introduction

Forest fires are a common occurrence in several countries all around the world because of the general increase of hot and dry climate conditions and the presence of large forests. In most European countries such as Cyprus, France, Greece, Italy, Portugal, Spain, and Turkey, as well as parts of Africa, Australia, and USA, every summer numerous fires destroy thousands of acres of forests and pose great risks to life and infrastructure during all times of the year. In the United States, there are typically between 60,000 and 80,000 wildfires that occur each year, burning 3 million to 10 million acres of land [1]. According to the Joint Research Centre (JRC), in just one year a total of 323,896 hectares of land has been destroyed in 52,795 fires in France, Greece, Italy, Portugal, and Spain [2].

In general, forest fires have a lasting impact on social, environmental, and financial aspects. Socially, catastrophic fires can have an enormous impact with losses of human lives and destruction of properties, thus creating persisting effects on a collective and individual level [3, 4]. Forest fires, especially mega fires, can cause psychopathological disturbances to survivors [5] as well as to firefighters [6]. Environmentally, in [7] novel techniques in environmental pollution analysis clearly demonstrate how air quality can be affected in the short term, while in [8] it is argued that previously burned areas have an increased probability to be burnt again, thus intensifying the catastrophes. On a global scale, forest fires can potentially increase the total carbon footprint [9]. It is therefore imperative that society and local authorities are equipped with necessary systems to act proactively and reactively against forest fires. Strategies of fire prevention, detection, and suppression have varied over the years, and international experts encourage further development of technology and research [10].

In this perspective, wireless sensor networks (WSNs) or geosensor networks (GSNs) are being frequently used in

disaster management scenarios for promptly detecting and continuously monitoring environmental phenomena, such as toxic plumes, earthquakes, and oil spills [11, 12]. In the context of forest fire management, several recent works proposed the use of WSNs for firefighting [13, 14]. In particular, EIDOS (Equipment Destined for Orientation and Safety) [15, 16] is our novel WSN-based support system proposed for reducing hazardous situations for people working in forest firefighting operations, employing various sensors, aerial vehicles, and mobile devices to set up an information system and assist them when they are not aware of the evolution of the fire in the surroundings. In this case, the goal of the sensor network consists of obtaining a map of the fire that is displayed on the firefighters' handheld devices.

In EIDOS, each network node builds and maintains its own approximation of the forest fire shape, by using the information it gathers from the network, as the fire spreads. To do this, when a node detects the proximity of a fire, it broadcasts a fire detection notification into the network such that all nodes are able to update their local approximation of the fire shape. Therefore, the algorithm to update and maintain this approximation of the fire shape is a central piece of the EIDOS system. In this paper, we will compare innovative and efficient solutions, in the sense that they consume a small amount of resources (memory, computation, and communication) for approximating complex fire shapes.

We have developed three different fire approximation models, with different characteristics. In the first model [17], nodes represent the shape of the fire by considering the full set of positions of burnt nodes (i.e., nodes reached by the fire so far and which have sent previously a fire notification event), assuming a circular burning area around each one of such positions. The resulting fire shape will be the union of all of those circular shapes. This approach obtains good approximations, but it has the drawback of requiring the storage and forwarding of every fire event received.

To address the bottlenecks detected in the first model, in our second proposal the forest fire is approximated by means of a shape composed of several convex hulls (representing distinct burning areas) that eventually merge themselves as they grow and overlap over the time as the fire spread [18]. In this case, instead of relaying all the received fire events, each node forwards only the "new" events, that is, the events originating by the nodes whose position is not already included in any of the actual hulls. In [19], we proved that this approach greatly helps in saving network resources, since meaningless events coming from nodes that do not contribute in improving the current fire shape are discarded.

Finally, along the lines of the latter solution and with the goal of further improving the accuracy of the approximation of the fire shape in realistic scenarios, in this paper we propose a third model, which removes the assumption of convex hull-based shapes and makes use of irregular polygons of arbitrary shape. As we will demonstrate through computer simulations, this method outperforms the classical ones based on convex shapes, while keeping low the impact on the network overhead.

The rest of this paper is organized as follows. Next section provides some background on the related works in WSNs-based phenomena monitoring. Section 3 describes the architecture and functionality of the EIDOS system. After that, Section 4 formally introduces the fire approximation techniques referred to in this work, along with the new proposed one, while Section 5 presents a detailed performance assessment of them. Finally, concluding remarks and future work directions are given in Section 6.

2. Related Work

Lots of proposals addressing the problem of mapping and tracking the contour or boundary of a physical phenomenon by using WSNs can be found in the literature. A comprehensive survey can be found in [20].

In [21], one of the earliest works on contour mapping, each network node gathers the measurements sensed by its direct neighbors and uses this information to determine whether it is close to the edge of two inhomogeneous fields. The work proposes three approaches for edge detection. The first approach is a statistical approach, while the other two are based on a high pass filter and a classifier, respectively.

The procedure proposed in [22] requires a hierarchical communication structure. Sensor nodes must be able to not only detect local physical properties, such as temperature, but rather measure the properties within a certain distance. This approach is based on the fact that sensor nodes can determine the distance and direction from the border of the observed phenomenon.

In [23], the authors present an algorithm for boundary approximation in locally linked sensor networks that communicate with a remote monitoring station. They use Delaunay triangulations and Voronoi diagrams to generate a sensor communication network and to define boundary segments between sensors, respectively. The proposed algorithm identifies boundaries based on differences between neighboring sensor readings and not absolute sensor values.

In [24] a cross layer approach for obtaining the contour of the phenomenon is proposed. It incorporates data fusion techniques which use the sensing noise (often negligible), the data quantification error, and the data communication noise. Furthermore, instead of making a hard binary decision, the probability for a sensor node being a contour node is calculated at the local fusion center.

Authors of [25] describe a scheme for estimating the boundary of a large-scale phenomenon by aggregating readings along a predefined hierarchical structure within the network.

The contour mapping engine (CME) was proposed in [26] in order to build a dynamic contour map by using in-network data processing techniques. In this approach, the network must be divided into clusters. Instead of sending to the sink all the sensor readings, the cluster head at each cluster builds some contour segments and reports them to the sink.

The mechanism presented in [27] also relies on a root node for obtaining the phenomenon boundary. Nevertheless, it is particularly interesting, since it incorporates a strategy to minimize overall data communication. In this proposal, sensors exchange information only when the process under

study does not proceed as expected. However, it involves programming network nodes with a model of the phenomenon behavior (referred to as *tiny model*). Further proposals focused on reducing the amount of data to send to the sink node during the tracking of a continuous object can be found in [28–31].

There are many proposals for representing in a compact way the spatial shape of a phenomenon starting from the set of localizations where its presence has been detected. In [32], authors analyze the use of lines and Bezier curves for approximating a set of data points provided by a WSN. In [33] a set of polygons are used for representing the contour of the phenomenon, being the number of vertices that are employed a user-specified parameter. Some complex analytical frameworks, such as *Voronoi diagrams* [23], *kernel linear regression* [34], and *Gaussian kernel estimation* [35] have been also proposed for modeling sensor data.

Recently, an algorithm for phenomena tracking has been proposed in [36]. This proposal relies on a complex deformable curve model [37] to maintain an updated representation of the phenomenon. The key idea is that each sensor node is able to detect incremental changes in the phenomenon boundary in its proximity, only by exchanging messages with its neighbors. These changes are then reported to a base station, in charge (again) of aggregating all the information.

Although these mechanisms are partly distributed (most of them rely on some clustering technique [38]), to our knowledge, in all of them the participation of a base station is required at some point of the process. As the EIDOS system considers the existence of a base station optional, these proposals are not suitable for us.

The work presented in [39] is an exception. In this case, authors propose a completely distributed contour tracking algorithm for the sensor network to maintain contours (or boundaries) of a binary object incrementally as they deform, while guaranteeing that the maintained contours capture the global topological features of the object boundary.

3. EIDOS: A System for Forest Fire Management

Disaster management systems face rapidly changing situations by relying on the capability of a (complex) distributed sensor network, deployed over a wide area, to capture and report real time data from a large number of heterogeneous information sources. The final goal of the full system is to provide support for decisions making [11]. Extensive examples of the use of WSNs in such scenarios can be found in [40–44].

In this work, we focus on forest firefighting operations and in particular on the EIDOS system [15, 16], which was proposed as a disaster management system to help firefighters to increase their efficacy, while minimizing their risks. The system is based on a large and dense WSN, composed by nodes randomly dropped by aircraft in the surroundings of the area affected by a forest fire, and is able to provide the firefighters with critical information, contributing to enhancing their safety. More in detail, the data collected by the WSN is processed by the network nodes in a fully distributed and collaborative way, with the goal of tracking the position and shape of the active fire boundaries. Finally, this information is always made available to the firefighters, who are equipped with mobile handheld devices.

3.1. EIDOS Architecture. EIDOS considers three types of devices, as sketched in Figure 1. First, the nodes of the WSN (commonly referred to as "motes") are basically small computing and storage platforms with wireless radios. They are equipped with sensors able to monitor environmental parameters, such as temperature, pressure, and humidity. Optionally, (some of) these nodes are also equipped with GPS (Global Positioning System) receivers.

Besides the motes, the firefighters are directly involved in fire extinction activities and carry wireless handheld mobile devices, such as smartphones or tablets, which allow them to communicate with the WSN. The purpose of these devices is to process information in order, for example, to display the fire map by means of a graphical interface. Of course, the connection gateway between the Bluetooth or Wi-Fi technology commonly supported by generic handheld devices and the ZigBee/IEEE 802.15.4 technology usually employed by the WSN should be explicitly addressed. However, this issue is out of the scope of this paper, assuming that they are able to wirelessly interact. Finally, the system incorporates several (possibly unmanned) aerial vehicles, which perform the task of deploying the network over the area of interest.

Unlike other similar systems proposing the use of a predeployed sensor network for forest fire detection and monitoring [13, 14, 45–54], we do not explicitly rely on a base station (usually called "*sink*," or "*gateway*") which gathers the environmental data from the sensor nodes. In other words, the EIDOS system is designed to work correctly, even if there is not connectivity with the sink. For this reason, the base station has not been considered in Figure 1.

Next, we detail the behavior of network nodes after their deployment.

3.2. Node Deployment and Localization. As stated above, in the EIDOS system the WSN used to build the map of the fire is deployed by dropping motes from the air. This involves several issues, such as atmospheric conditions, aircraft speed and direction, and coverage strategy. As a consequence, the resulting network topology is highly irregular and unknown *a priori*. Therefore, the first key task of each node in the network consists of determining its own geographical position. This information can be either directly obtained from an onboard GPS receiver or estimated by running a distributed localization algorithm.

A description of the existing localization techniques for WSNs is out of the scope of this work, and can be found in [55–57]. Most of them are based on the existence of some special nodes (called *beacons* or *anchors*) with known coordinates, which periodically broadcast their position in order to help other nodes (called *blinds*) to localize themselves. Differently from the classical *range-based* solutions (e.g., [58]), in

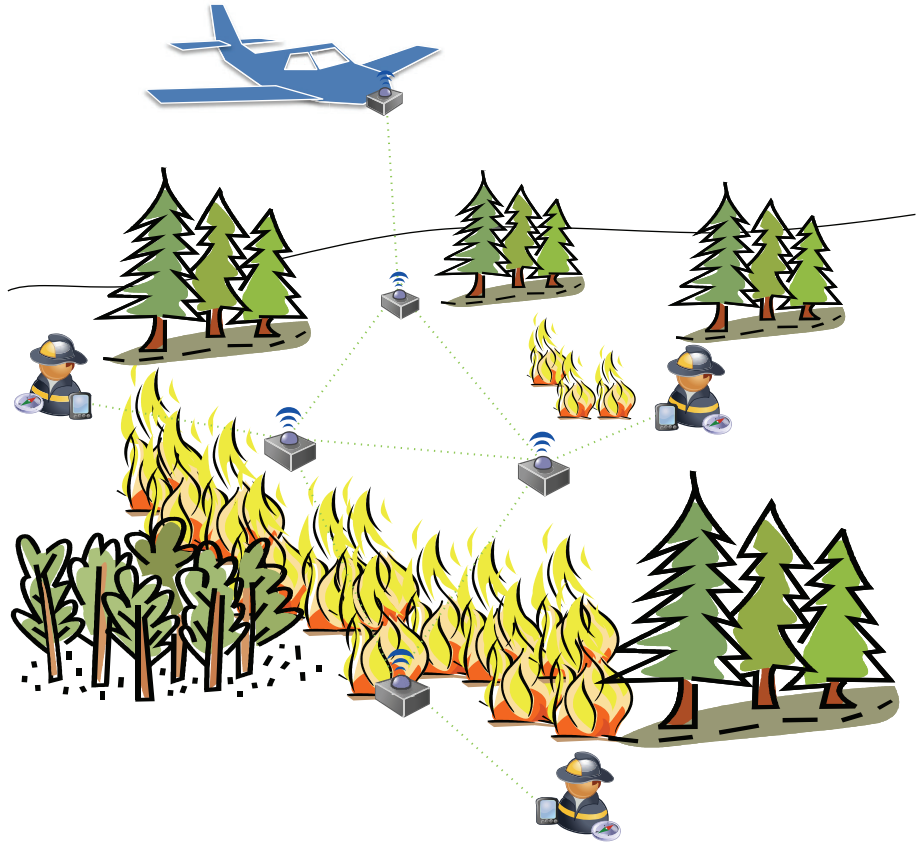


Figure 1: EIDOS architecture.

EIDOS we have opted for a *range-free* localization technique, in which blind nodes only use connectivity information to estimate their location [59].

In particular, starting from the information each node receives, a fully distributed and iterative process is executed, in which the node's location estimate is progressively refined as a rectangular area. More in detail, the localization process is started by the beacon nodes, which broadcast their position. Then, each time a node receives a localization estimate, it extends the received area by using a common radio coverage range. After that, it updates its current estimate by intersecting it with extended received area. Then, the new estimate is transmitted again, in order to help other nodes to refine their estimates. All details of this algorithm are in [59] and here we aim to recall that it has been demonstrated that with a very small percentage of anchors equipped with GPS receivers, that is, as low as 2% of the total number of nodes, the position error of the blind nodes falls below the radio range (e.g., <50 meters).

3.3. Fire Detection and Dissemination. During normal operation, each node n detecting an approaching fire front triggers a process for broadcasting its position p to the entire network. We assume that each node is able to periodically monitor the local temperature, detecting the arrival of the fire front when the sensed value overcomes a predefined threshold t_{detect}. The strategy to establish the sampling rate is out of the scope of this work. Additionally, nodes burn at a certain temperature t_{burn}, such that $t_{\text{burn}} > t_{\text{detect}}$. In our simulation experiments, we assume that, after reaching t_{detect}, nodes are able to transmit their position before burning. Otherwise, from the point of view of the mechanisms described in the following section, these nodes simply do not exist.

In order to minimize the consumption of network resources and prolong network lifetime, WSN nodes neither maintain any hierarchy nor have preliminary information about the network topology. With these restrictions, for disseminating fire detection events, EIDOS implements a variation of ABBA (Area-based Beaconless Algorithm) [60], an efficient broadcasting mechanism.

In particular, our dissemination technique is detailed in [61]: it assumes circular coverage areas and is based on the perimeter covered by the copies of the same message. Basically, a node n cancels the forwarding of a message m_p when the successive copies of m_p $(m'_p, m''_p, \ldots)$ completely cover the perimeter of n. Note that it is necessary that each node maintains a queue of messages waiting to be forwarded, along with the perimeter not yet covered by the copies of these messages. Furthermore, to enable the updating of the covered perimeter at the receiver node, messages have to explicitly include the transmitter's position.

Starting from the fire detection events it receives, each node builds and maintains a local approximation of the whole forest fire. The choice of the most appropriate fire

representation model is of paramount importance since it will have a huge impact on both (i) the accuracy of the obtained approximations and (ii) the amount of resources required (including computing power, memory, and wireless bandwidth, for each node). Section 3 introduces the formal models considered in this work.

4. Forest Fire Approximations

This section details the different models for approximating forest fires with WSNs. For each fire approximation, we will present some definitions that formally describe it. They provide a theoretical framework for the implementation of the algorithm executed by every network node in order to obtain the fire model and update it as the fire spreads. Due to space constraints, implementation details are skipped.

4.1. Circle-Based Model. The first approach consists of representing the fire by means of a set of circles generated around the position of each node detecting the fire [17]. To achieve it, each network node stores all the fire positions received from its neighbors and forwards them (unless they are discarded by the dissemination process policy [61]). In this way, nodes approximate the forest fire assuming that it is flared up and currently burning in the surroundings of the collected positions. Next, we formally define the shape considered for the approximation.

Definition 1 (point). A point $p \in \mathbb{R}^2$ is a position of the 2D plane with coordinates (p_x, p_y).

Definition 2 (distance between points). Given two points $a, b \in \mathbb{R}^2$, the Euclidean distance between them is provided by the function distance as follows: $\mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$, denoted by Distance$(a, b)$ and defined as

$$\text{Distance}\,(a,b) = \sqrt{\left(a_x - b_x\right)^2 + \left(a_y - b_y\right)^2}. \tag{1}$$

Definition 3 (circle). Given a point $p \in \mathbb{R}^2$ and a value $r \in \mathbb{R}$, then $c_p^r \in \mathbb{R}^2 \times \mathbb{R}$ denotes the area delimited by a circle centered at p with radius r.

Definition 4 (fire spread function). Let $\mathcal{F} = (\wp(\mathbb{R}^2) \times \mathbb{R}^2, \mathbb{R})$ be the set of functions from $\wp(\mathbb{R}^2) \times \mathbb{R}^2$ to $\mathbb{R}$. Given a set of points $P \in \wp(\mathbb{R}^2)$ and a point $p \in P$, then a function FireSpread $\wp(\mathbb{R}^2) \times \mathbb{R}^2 \rightarrow \mathbb{R}$, denoted by FireSpread$(P, p)$ or $F(P, p)$ (or simply $F \in \mathcal{F}$), will provide a radius for a circle representing the fire spread at point p.

Definition 5 (circle-based shape). Given a set of points $P \in \wp(\mathbb{R}^2)$ and a fire spread function $F \in \mathcal{F}$, then the function GetShape $\wp(\mathbb{R}^2) \times \mathcal{F} \rightarrow \wp(\mathbb{R}^2 \times \mathbb{R})$, denoted by GetShape$(P, F)$ or S_P^F, obtains a shape verifying that $\forall p \in P$, $c_p^{F(P,p)} \in S_P^F$.

Figure 2 shows an example of a circle-based shape.

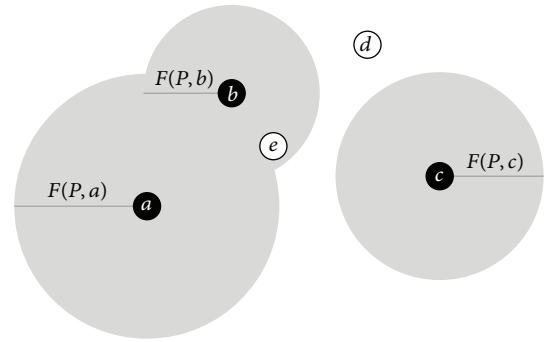


Figure 2: A circle-based shape $S^F_{\{a,b,c\}}$, represented by the shadowed area. Additional points may be analyzed to determine if they are inside the shape. In this example, Inside$(S^F_{\{a,b,c\}}, d)$ = false, while Inside$(S^F_{\{a,b,c\}}, e)$ = true (since Distance$(b, e) \leq F(\{a, b, c\}, b)$).

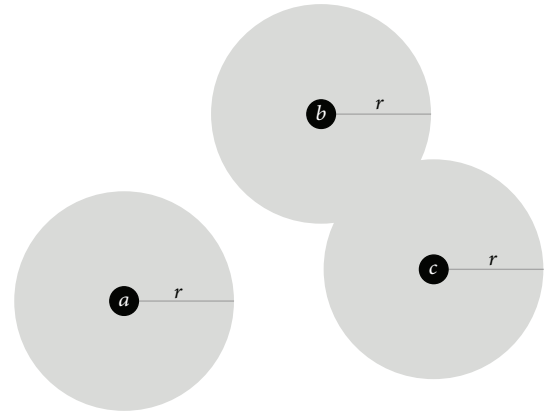


Figure 3: A homogeneous shape $S^r_{\{a,b,c\}}$. The fire spread function applied is $F(\{a, b, c\}, p) = r$.

Definition 6 (belonging function). Given a circle-based shape S_P^F and a point $a \in \mathbb{R}^2$, the function Inside $\wp(\mathbb{R}^2 \times \mathbb{R}) \times \mathbb{R}^2 \rightarrow \{\text{true}, \text{false}\}$, denoted by Inside$(S_P^F, a)$, is defined by

$$\text{Inside}\left(S_P^F, a\right) = \begin{cases} \text{true}, & \text{if } \exists c_b^r \in S_P^F \mid \text{Distance}\,(a,b) \leq r \\ \text{false}, & \text{in other case.} \end{cases} \tag{2}$$

Given a shape and an arbitrary location, this function provides a way for deciding whether that location is burning or not. Examples of application of this function are shown in Figure 2.

4.1.1. Fire Spread Functions. Different criteria can be used for defining the fire spread function. The simplest criterion consists of considering a constant radius r for every circle in the shape. The fire spread function is $F(P, p) = r$, and the resulting shape may be denoted as S_P^r. Figure 3 shows an example. We refer to this model as *homogeneous shapes*.

Alternatively, each network node receiving a new fire point $p \in P$ can determine the radius r for a new circle c_p^r

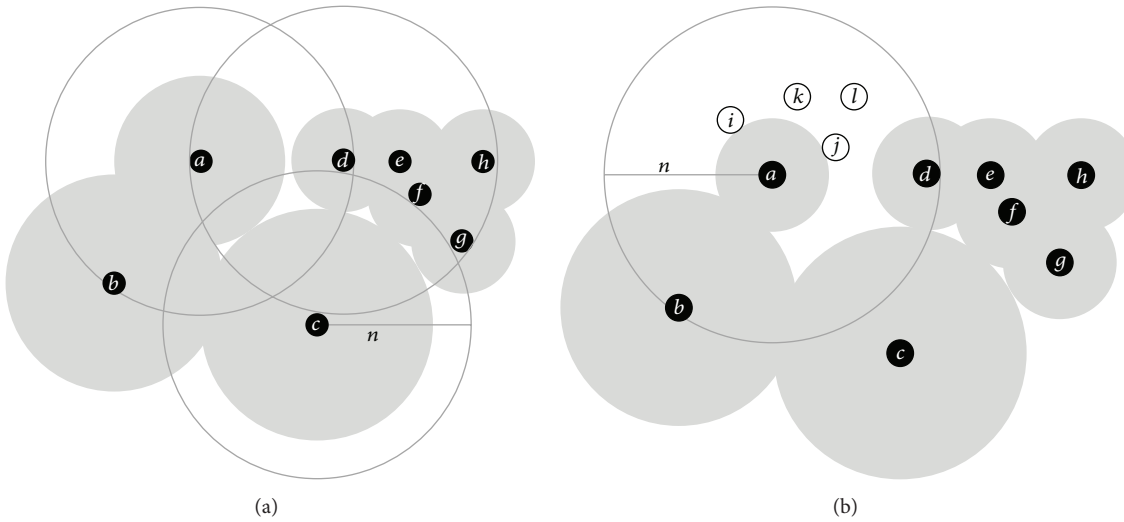


Figure 4: Examples of heterogeneous shapes based on fire density (a) and node density (b). Shapes are represented by shadowed areas. Black points represent nodes detecting fire. White points represent the rest of network nodes. In (a), the biggest circle corresponds to point c due to GetNeighborhood$(P, c_c^n) = \{c\}$. One of the smallest circles corresponds to point d, due to GetNeighborhood$(P, c_d^n) = \{a, d, e, f, g, h\}$.

for the shape as function of the fire density in a particular neighboring area. This density is computed starting from the amount of fire points currently included in the shape and formally defined in the following.

Definition 7 (neighborhood). Given a set of points $P \in \wp(\mathbb{R}^2)$ and a circular area $c_a^n \in \mathbb{R}^2 \times \mathbb{R}$, the subset of those points that are located inside this area is defined by the function GetNeighborhood $\wp(\mathbb{R}^2) \times (\mathbb{R}^2 \times \mathbb{R}) \rightarrow \wp(\mathbb{R}^2)$, denoted by GetNeighborhood(P, c_a^n), which verifies the following:

(1) GetNeighborhood$(P, c_a^n) \subseteq P$.

(2) $\forall p \in P \mid \text{Distance}(p, a) \leq n$, then $p \in$ GetNeighborhood(P, c_a^n).

This is a noninjective function. Consequently, it is not possible to recover the original set P starting from GetNeighborhood(P, c_a^n). Additionally, it is surjective, so that a randomly selected collection of points may represent a neighborhood.

Definition 8 (area density). Given a set of points $P \in \wp(\mathbb{R}^2)$ and a circular area $c_a^n \in \mathbb{R}^2 \times \mathbb{R}$, the function Density $\wp(\mathbb{R}^2) \times (\mathbb{R}^2 \times \mathbb{R}) \rightarrow \mathbb{R}$, is defined as

$$\text{Density}\left(P, c_a^n\right) = \frac{\left|\text{GetNeighborhood}\left(P, c_a^n\right)\right|}{\pi n^2}. \quad (3)$$

Starting from the previous definitions, the fire spread function may provide big circles covering less dense areas and smaller circles as point density increases. Figure 4 shows some examples.

These *heterogeneous shapes* based on fire density may be computed by each destination node (or handheld device carried by a firefighter) as fire points are received. Given a prefixed radius n for the neighborhood, according to [17], two options for the fire spread function are an inverse linear behavior $F(P, p) = A - B(\text{Density}(P, c_p^n))$ and a logarithmic behavior $F(P, p) = A + B\ln(\text{Density}(P, c_p^n))$.

A straightforward improvement of this approach consists of computing the area density by considering all the deployed network nodes, even those nodes which have not reported fire, yet. In this way, the previous definitions applied to fire density can be directly translated to node density.

We can see the benefits of this improvement by comparing Figures 4(a) and 4(b). In the new approximation (Figure 4(b)), the size of the shaded circular area centered on point a has been considerably reduced, since GetNeighborhood$(P, c_a^n) = \{a, b, d, i, j, k, l\}$. However, given that each network node does not store information about the entire topology (it is only able to know about its direct neighbors), this approach involves that density values are determined by the nodes detecting the fire, instead of being computed by the nodes receiving the corresponding notification (as before). As a consequence, the dissemination mechanism should support the propagation of this information through the network.

4.2. Hull-Based Model. In this subsection we describe a second proposal for modeling forest fires, based on a shape composed of a collection of convex hulls (from now on, only "hulls") [18, 62]. The advantage of this approach is that each node only considers those fire positions received which would imply a variation in the local approximation, ignoring the rest of fire events. Consequently, the amount of data stored and disseminated through the network is significantly reduced.

Definition 9 (relative position among points). Given three points a, b, and $c \in \mathbb{R}^2$, the relative position (*clockwise*, *counter clockwise*, or *in line*) among them is provided by

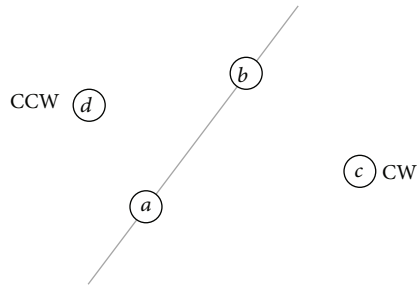


Figure 5: Relative position of three points. Given the spatial distribution of points a, b, c, and d, then $\text{Order}(a,b,c) = \text{CW}$ and $\text{Order}(a,b,d) = \text{CCW}$. In the same way, $\text{Order}(a,c,b) = \text{CCW}$, $\text{Order}(a,d,b) = \text{CW}$, and $\text{Order}(c,d,a) = \text{CCW}$. Similarly, $\text{Order}(a,b,a) = \text{Order}(a,b,b) = \text{LINE}$.

the function $\text{Order}\ \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \to \{\text{CW}, \text{CCW}, \text{LINE}\}$, denoted by $\text{Order}(a,b,c)$ and defined by

$$\text{Order}(a,b,c) = \begin{Bmatrix} \text{CW}, & \text{if } \det(a,b,c) < 0 \\ \text{CCW}, & \text{if } \det(a,b,c) > 0 \\ \text{LINE}, & \text{if } \det(a,b,c) = 0 \end{Bmatrix}, \quad \text{where } \det(a,b,c) = \begin{vmatrix} 1 & a_x & a_y \\ 1 & b_x & b_y \\ 1 & c_x & c_y \end{vmatrix}. \tag{4}$$

An example of application of this function is shown in Figure 5.

Definition 10 (hull function). Given a set of points $P \in (\mathbb{R}^2)$, the function GetHull $\wp(\mathbb{R}^2) \to \wp(\mathbb{R}^2)$, denoted by $\text{GetHull}(P)$ or H_P, verifies the following:

(1) $H_P \subseteq P$.

(2) $\forall a \in H_P, \exists b \in H_P \mid \forall p \in P, \text{Order}(a,b,p) \in \{\text{CW}, \text{LINE}\}$.

(3) $\forall a \in H_P, \exists b \in H_P \mid \forall c \in H_P, \text{Order}(a,b,c) = \text{CW}$.

An example of application of this function is shown in Figure 6(a).

GetHull is a noninjective function. Consequently, it is not possible to recover the original set P starting from $\text{GetHull}(P)$. Additionally, it is nonsurjective, so that a randomly selected collection of points does not necessarily represent a hull. For this reason, we introduce the next definition.

Definition 11 (hull). A set of points $H \in \wp(\mathbb{R}^2)$ is a hull if it verifies that $\text{GetHull}(H) = H$.

Definition 12 (enclosed set). Given a set of points $P \in \wp(\mathbb{R}^2)$ and a hull $H \in \wp(\mathbb{R}^2)$, the function Enclosed $\wp(\mathbb{R}^2) \times \wp(\mathbb{R}^2) \to \wp(\mathbb{R}^2)$, denoted by $\text{Enclosed}(P,H)$, verifies the following:

(1) $\text{Enclosed}(P,H) \subseteq P$.

(2) $\forall p \in P \mid \forall a \in H, \exists b \in H \mid \text{Order}(a,b,p) = \text{CW}$, then $p \in \text{Enclosed}(P,H)$.

Enclosed is a noninjective function since although it is possible to recover the original set H starting from $\text{Enclosed}(P,H)$, it is not possible to recover the original set P. Additionally, it is a surjective function. Therefore, given any random set of points, there is a hull enclosing it. An example of application of this function is shown in Figure 6(b).

Definition 13 (hull-based shape). Given a set of points $P \in \wp(\mathbb{R}^2)$ and a value $d \in \mathbb{R}$, the function GetShape $\wp(\mathbb{R}^2) \times \mathbb{R} \to \wp(\wp(\mathbb{R}^2))$, denoted by $\text{GetShape}(P,d)$ or S_P^d, verifies the following:

(1) $\forall H \in S_P^d, H \subseteq P \wedge H$ is a hull.

(2) $\forall H \in S_P^d, \forall Q_1, Q_2 \mid Q_1 \cup Q_2 = \text{Enclosed}(P,H) \wedge Q_1 \cap Q_2 = \varnothing, \exists a \in Q_1 \mid \exists b \in Q_2$ that verifies $\text{Distance}(a,b) < d$.

(3) $\forall H_1, H_2 \in S_P^d, H_1 \cap H_2 = \text{Enclosed}(P,H_1) \cap \text{Enclosed}(P,H_2)$.

(4) $\forall H_1, H_2 \in S_P^d, \forall a,b \in P, a \in H_1 \wedge a \notin H_2 \wedge b \notin H_1 \wedge b \in H_2 \Rightarrow \text{Distance}(a,b) > d$.

(5) $\forall p \in P, \exists H \in S_P^d \mid p \in \text{Enclosed}(P,H)$.

GetShape is a noninjective function. Consequently, it is not possible to recover the original set P starting from S_P^d. Additionally, it is nonsurjective, so that a randomly selected collection of points does not necessarily represent a shape. An example of application of this function is shown in Figure 7.

Given a hull-based shape representing a fire, the next function may be applied to determine whether an arbitrary location is burning or not.

Definition 14 (belonging function). Given a shape S_P^d and a point $p \in \mathbb{R}^2$, the function Inside $\wp(\wp(\mathbb{R}^2)) \times \mathbb{R} \times \mathbb{R}^2 \to \{\text{true}, \text{false}\}$, denoted by $\text{Inside}(S_P^d, p)$, verifies the following:

$$\text{Inside}\left(S_P^d, p\right) = \begin{cases} \text{true}, & \text{if } S_P^d = S_{P \cup \{p\}}^d \\ \text{false}, & \text{in other cases.} \end{cases} \tag{5}$$

4.3. Polygon-Based Model. The main contribution of the current work is a fire representation model based on arbitrary polygons, which is described in this subsection.

We first formally define the concept of a polygon-based shape, that is, a contour composed of several closed chains of vertices connected by segments. After that, we will introduce the criterion used to determine if a specified position of the plane is covered (or not) by a given shape.

4.3.1. Polygon-Based Shapes

Definition 15 (vertex). Let $\mathbb{V}$ be the set of vertices. Given a vertex $p \in \mathbb{V}$, the function Position $\mathbb{V} \to \mathbb{R}^2$, denoted by

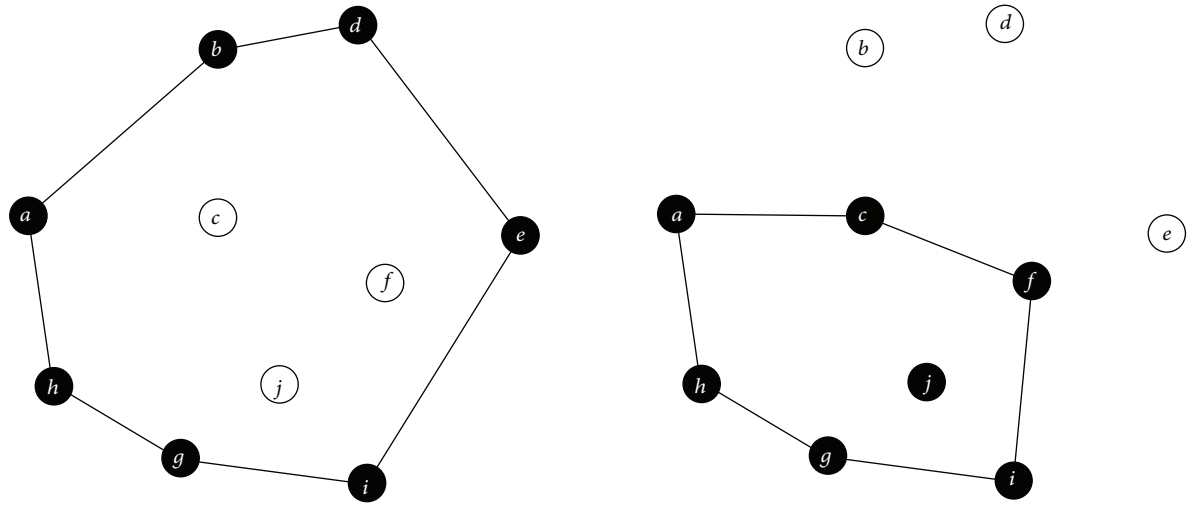
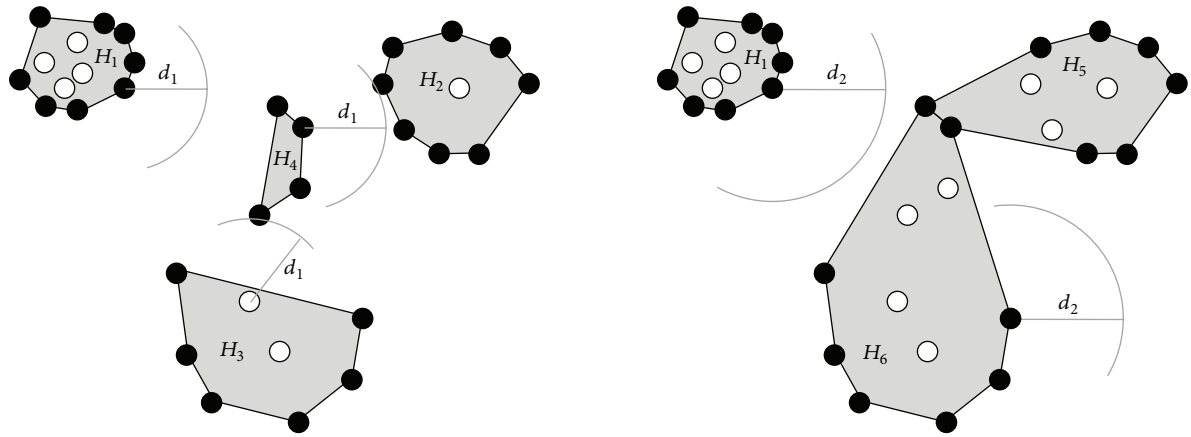


FIGURE 6: (a) Hull obtained from a set of points. Given $P = \{a, b, c, d, e, f, g, h, i, j\}$, then $H_P = \{a, b, d, e, g, h, i\}$ (black points). (b) Set of points enclosed by a hull. Given $P = \{a, b, c, d, e, f, g, h, i, j\}$ and a hull $H = \{a, c, f, g, h, i\}$, then Enclosed$(P, H) = \{a, c, f, g, h, i, j\}$ (black points).



(a) GetShape$(P, d_1) = \{H_1, H_2, H_3, H_4\}$

(b) GetShape$(P, d_2) = \{H_1, H_5, H_6\}$

FIGURE 7: Set of points enclosed by a hull-based shape. Given a spatial distribution for P, the amount of hulls provided by GetShape depends on the applied threshold distance. (a) and (b) show results for two different values, d_1 and d_2, assuming that $d_1 < d_2$. Hulls are represented by linked black points. Points not belonging to any hull are represented by white points. Independently of the value of the threshold distance, all the points of P are enclosed into some hull.

Position$(p) = P$, provides the 2D position of the vertex. This is a noninjective function since multiple overlapped vertices $p_1, p_2 \dots p_n$ may be located at the same point, that is, Position(p_1) = Position$(p_2) = \cdots =$ Position$(p_n) = P$. In this case, all vertices are referred to as *clones*.

For the sake of clarity, we will use circles labeled with capital letters for representing points and we will represent vertices by means of misplaced boxes labeled with small letters with numerical subscripts used to distinguish among clones (see, e.g., Figure 8).

Definition 16 (sequence functions). Let $\mathcal{F} = (\wp(\mathbb{V}) \times \mathbb{V}, \mathbb{V})$ be the set of functions from $\wp(\mathbb{V}) \times \mathbb{V}$ to $\mathbb{V}$. Given a set of vertices $V \subset \mathbb{V}$ and a vertex $v \in \mathbb{V}$, the function next $\wp(\mathbb{V}) \times \mathbb{V} \to \mathbb{V}$, denoted by next$(V, v)$, or simply $v^{\rightarrow}$, provides the next vertex to v in V, that is, $\forall a \in V$, $a^{\rightarrow} \in V$. Similarly, the inverse function prev $\wp(\mathbb{V}) \times \mathbb{V} \to \mathbb{V}$, denoted by prev$(V, v)$ or $v^{\leftarrow}$, provides the previous vertex to v in V, satisfying that $\forall a, b \in V \mid a^{\rightarrow} = b$ implies that $b^{\leftarrow} = a$.

Definition 17 (polygon-based shape). Given a set of vertices $V \in \wp(\mathbb{V})$ and a function next$(V, v) \in \mathcal{F}$, a shape

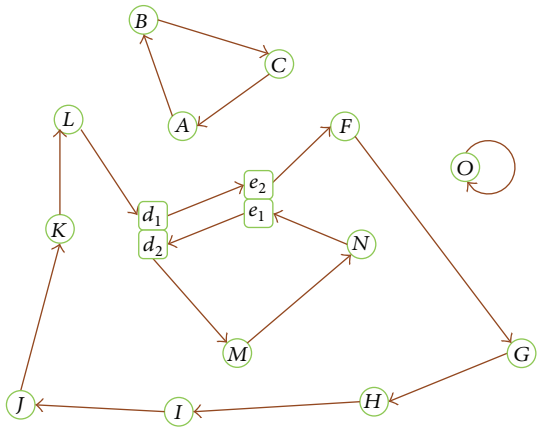


FIGURE 8: Example of a polygon-based shape $S = \{[a\,b\,c][d_1\,e_2\,f\,g\,h\,i\,j\,k\,l][d_2\,m\,n\,e_1][o]\}$. Boxes help us to distinguish among several cloned vertices (placed into the same location). For clarity, segment oo (with null length) has been drawn as a curved vector starting and ending at the same point.

$S \in \mathbb{S} = (\wp(\mathbb{V}) \times \mathcal{F})$, denoted by $S_{(V,\text{next})}$, or S, is a set of vertices maintaining a relationship of sequence among them.

Definition 18 (segment). Given two vertices $a, b \in \mathbb{V} \mid \text{Position}(a) = A$ and $\text{Position}(b) = B$, the relation $a^{\rightarrow} = b$ will be denoted by a segment ab and graphically represented by a vector from point A to point B. Consequently, a shape will be represented as a directed graph (as shown in the example of Figure 8).

Definition 19 (chain). Given a shape $S_{(V,\text{next})} \in \mathbb{S}$, it may be partitioned in k disjoint ordered subsets $C_1, C_2, \ldots C_k$ called chains, verifying the following:

(1) $\forall v \in S, \exists C \subset S \mid v \in C$.

(2) $\forall C_i, C_j \subset S, C_i \cap C_j = \varnothing$.

(3) $\bigcup_{i=1}^{k} C_i = S$.

(4) $\forall C \subset S$ composed of n vertices, denoted by $[v_0\ v_1\ \cdots\ v_{n-1}]$, it verifies that $\forall i, 0 \leq i < n, v_i^{\rightarrow} = v_{((i+1) \bmod n)}$.

Graphically, each chain of the shape is represented as a subgraph composed of a cyclic sequence of n consecutive segments. Note that a chain allows n different notations. Also, when appropriate, irrelevant subchains in a chain are abbreviated by "$\cdots$."

4.3.2. Coverage Issues

Definition 20 (point of a segment). Given a point $P \in \mathbb{R}^2$ and a segment $ab \in S$, we say that P belongs to ab or $P \in ab$, if it is verified that

(1) $(P_y - A_y)(P_y - B_y) \leq 0$,

(2) $A_x + ((B_x - A_x)/(B_y - A_y))(P_y - A_y) = P_x$.

Definition 21 (horizontal semiline). Given a point $P \in \mathbb{R}^2$, a horizontal semiline $y = P_y$, or $P^{\multimap}$, is defined $\forall x > P_x$.

Definition 22 (segment crossing a semiline). Given a shape $S \in \mathbb{S}$, a point $P \in \mathbb{R}^2$ defining the semiline $P^{\multimap}$, and a segment $ab \in S$, we say that ab crosses $P^{\multimap}$ or $ab \pitchfork P^{\multimap}$, if the following conditions are satisfied:

(1) $A_y \neq B_y$.

(2) $A_x + ((B_x - A_x)/(B_y - A_y))(P_y - A_y) > P_x$.

(3) $(P_y - A_y)(P_y - B_y) \leq 0$.

(4) $P_y \neq \min(A_y, B_y)$.

Definition 23 (set of segments crossing a semiline). Given a shape $S \in \mathbb{S}$ and a point $P \in \mathbb{R}^2$ defining the semiline $P^{\multimap}$, the subset of segments of S crossing $P^{\multimap}$, denoted by $X^P \subset S$, verifies that

(1) $\forall ab \in S \mid ab \pitchfork P^{\multimap}, ab \in X^P$,

(2) $\forall ab \in X^P, ab \pitchfork P^{\multimap}$.

Definition 24 (belonging function). Given a shape $S_{(V,\text{next})} \in \mathbb{S}$ and a point $P \in \mathbb{R}^2$, the function Inside $\mathbb{S} \times \mathbb{R}^2 \rightarrow \{\text{true}, \text{false}\}$, denoted by $\text{Inside}(S, P)$, is defined by

$$\text{Inside}(S, P) = \begin{cases} \text{true}, & \text{if } (\exists a \in S \mid \text{pos}(a) = P) \vee (\exists ab \in S \mid P \in ab) \vee \left(\left|X^P\right| \text{ is odd}\right) \\ \text{false}, & \text{in other cases.} \end{cases} \tag{6}$$

$|X^P|$ indicates the *cardinality* of $|X^P|$, that is, the amount of segments crossing $P^{\multimap}$. Figure 9 shows an example of the behavior of this function.

A WSN deployed over a forest area with the purpose of monitoring the evolution of a wildfire will produce a set of 2D points indicating the presence of fire in the specific locations of certain network nodes. Although the information collected is discrete, the fire spreads continuously over the area. For this reason, the shapes should be "interpolated" starting from the collection of gathered points but by establishing a minimum distance threshold among these points to allow the space "in the middle" to be considered to be actually burning or not. This is formally stated in the next definition.

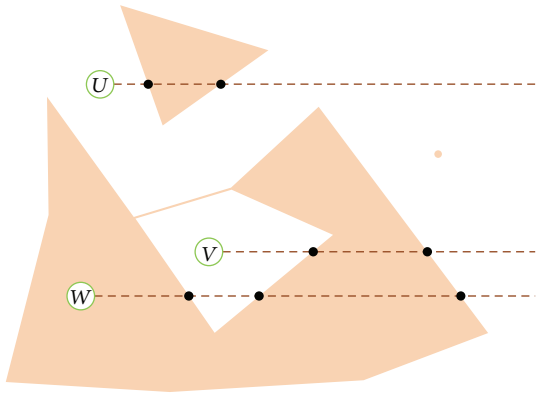


FIGURE 9: For the shape S of Figure 8, Inside(S, U) = false, Inside(S, V) = false, and Inside(S, W) = true.

Definition 25 (shape covering a set of points with a distance threshold). Given a set of points $Q \in \wp(\mathbb{V})$, a shape $S \in \mathbb{S}$ covers Q with a threshold d if it verifies that

(1) $\forall a \in S$, Position$(a) \in Q$,

(2) $\forall ab \in S$, Distance (Position(a), Position$(b)) \leq d$,

(3) $\forall A \in Q$, Inside(S, A) = true,

(4) $\forall A, B \in Q \mid$ Distance$(A, B) \leq d$, then $\forall P \in \mathbb{R}^2$, Inside$(\{[a, b]\}, P) \Rightarrow$ Inside(S, P),

(5) $\forall A, B, C \in Q \mid$ Distance$(A, B) \leq d$, Distance$(B, C) \leq d$, and Distance$(C, A) \leq d$, then $\forall P \in \mathbb{R}^2$, Inside$(\{[a, b, c]\}, P) \Rightarrow$ Inside(S, P).

5. Performance Evaluation

In this section, we will analyze the quality of the approximation produced by the proposed fire models. After describing the simulation environment and the evaluation methodology used, we present a preliminary study aimed at choosing the optimal value for the parameters associated with each model. Finally, we provide the results corresponding to the comparative evaluation.

5.1. Simulation Environment. In the context of the EIDOS system, we have developed a simulation environment [16], in which we can deploy a WSN, spread a forest fire, place firefighters, and see the evolution of the fire fronts that they are faced with. As shown in Figure 10, this tool is composed of several independent and interconnected modules, which share information by means of a global MySQL database [63].

In short, first we use Farsite [64] to simulate a fire over a particular forest area, under realistic conditions, that is, by using real geographical, environmental, and vegetation data. Then, a WSN simulator (developed in Python/TOSSIM [65]) executes the EIDOS application in each network node, having as inputs the evolution of the temperatures generated by Farsite.

Besides the WSN simulation, a graphical user interface (area display), developed with Adobe Flash [66], shows the evolution of the fire and allows the user to place and move firefighters across the scenario (Figure 12(a)). The evaluation environment also incorporates a handheld device simulator, developed with Adobe Air and interacting with the other components by means of Flash Remoting and Flash Media Server technology. This tool shows the fire approximation performed by the WSN in the surroundings of the position of the firefighter (Figure 12(b)).

Regarding the radio propagation, we assume the use of omnidirectional antennas and the same transmission power for all network nodes. In order to reproduce a realistic scenario, the WSN simulator incorporates a noise and interference model and the well-known *Friis free-space* signal propagation model [67]. We have modeled the radio of the Iris motes [68], applying a transmission power of 3 dBm and a minimum reception power of −90 dBm. Under these conditions, we obtain an approximate radio range of 50 meters. The simulated protocol for media access control is the basic CSMA [65].

5.2. Evaluation Methodology. At the beginning of each simulation run, the nodes are randomly distributed in a square area of 2500 × 2500 meters. We have considered network sizes varying from 2000 to 15000 nodes, corresponding to connectivity degrees (average number of direct neighbors per node) from 3.02 to 23.6. During the simulation, a forest fire with three separate ignition points and changing wind conditions spreads in the deployment area two hours after the beginning of the simulation, and four hours later it has reached approximately half of the simulation area (Figure 11).

Sensor nodes behave as detailed in Section 3.3. For localization purposes, in this paper we assume that all nodes know their location with negligible error. However, this is not a limitation, since the objective is to make a fair comparison of the different solutions proposed, benchmarking them against the baseline "circular shape" model. Each time a node detects a fire in its proximity (by a sudden rise in the sensed temperature), it broadcasts its position. Our simulation model also takes into account that in a short period of time, the node affected by the fire is burnt, and consequently it becomes not operational any longer. Note that, although sensor nodes may cease to be operative as the fire spreads, network connectivity is supposed to be never lost. In realistic situations, this assumption holds true thanks to the redundancy level in the number of deployed nodes or, in extreme cases, to the addition of new nodes dropped by the aircraft. This means that any new fire detection event always reaches every (survivor) network nodes; thus, they are able to estimate the same fire shape, in a fully distributed way.

In order to increase the representativeness of the obtained results, 10 independent simulation runs have been performed for each setup and the statistics have been averaged.

The Farsite simulator has been assumed as provider of ground-truth fire spreading images over the time and the images obtained by each approximation method have been compared against those ones. In particular, Farsite outputs a set of raster files. Each raster is a 2D grid of cells representing the whole simulation area (for this work, we have set cells of 10×10 meters size each). The raster is a Time of Arrival (TOA)

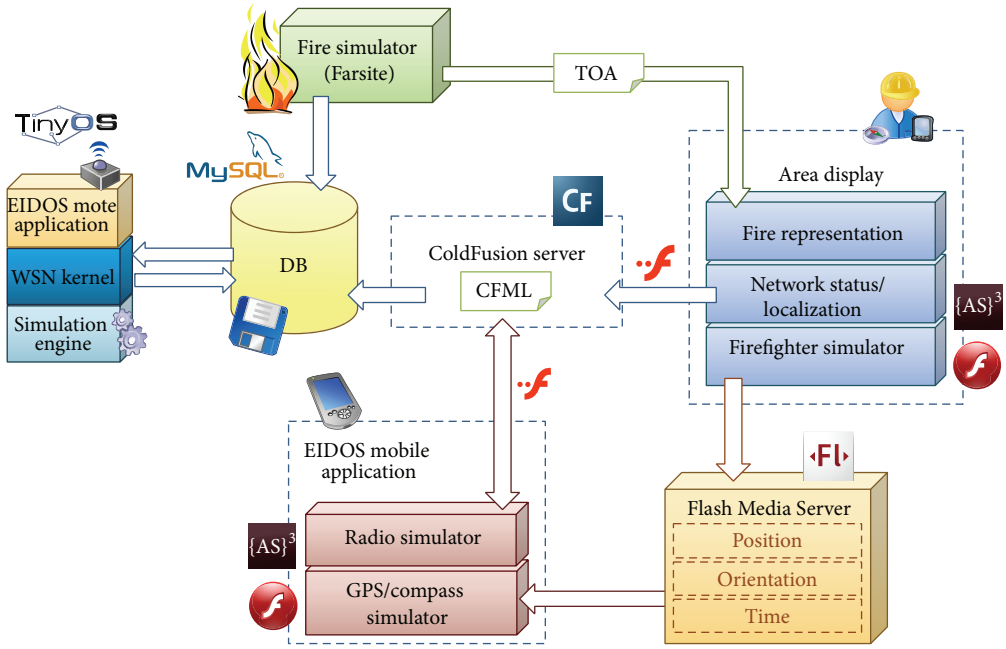


Figure 10: Architecture of the EIDOS simulation environment.

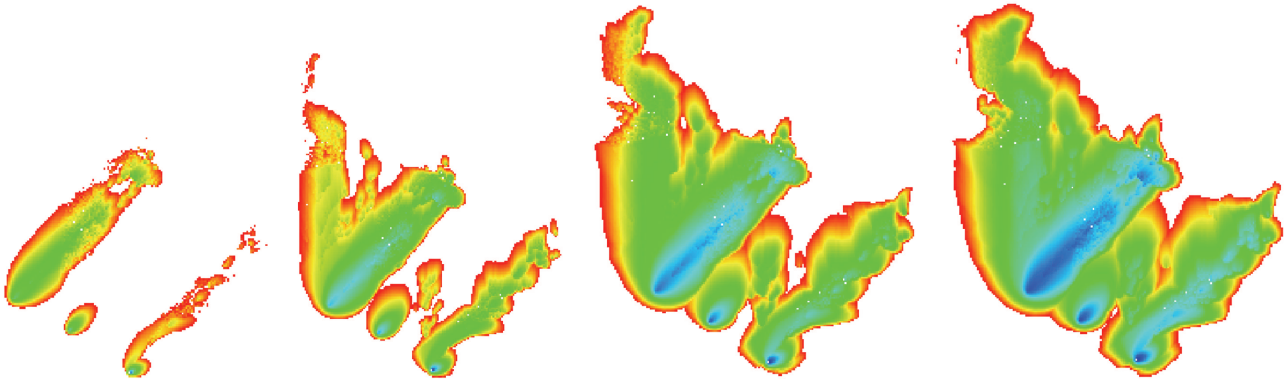
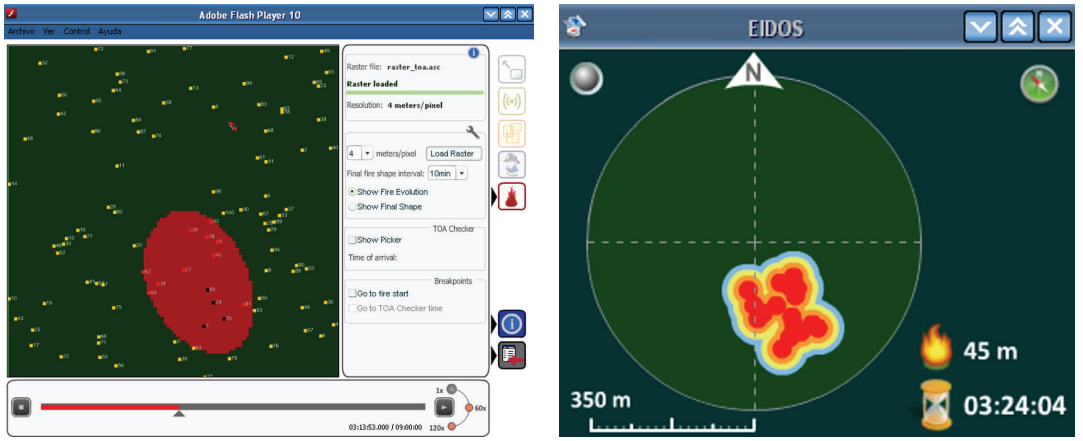
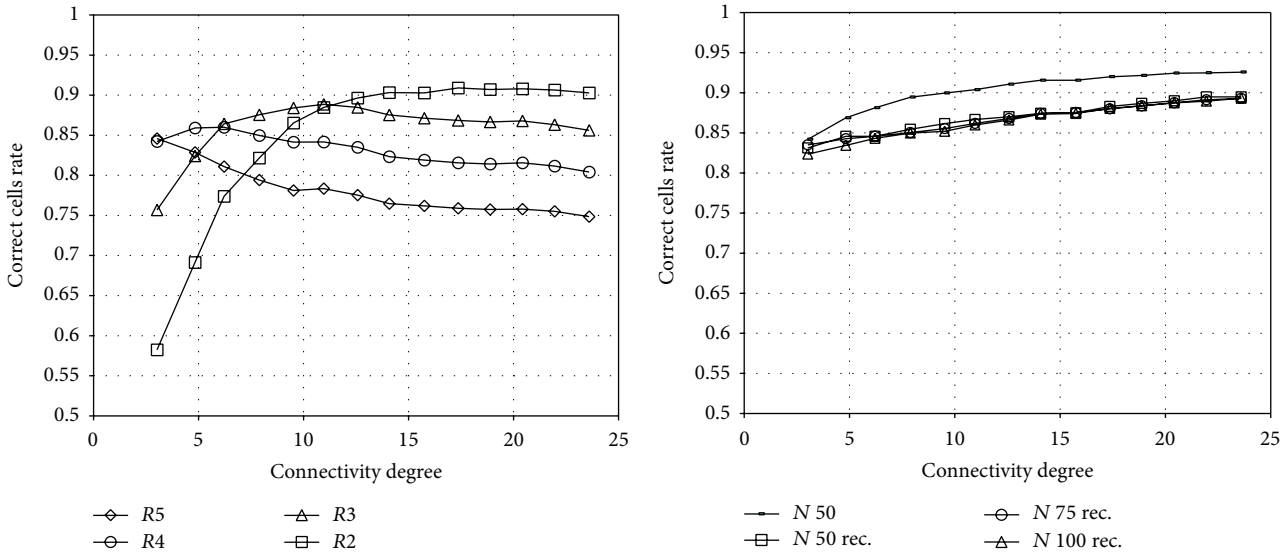


Figure 11: Aspect of the original fire.



(a) Forest area display (b) Firefighter mobile application

Figure 12: User interfaces developed in the context of the EIDOS simulation environment.



(a) Homogeneous shapes

(b) Heterogeneous shapes

Figure 13: Quality of the circle-based approximation versus network degree (time = 5 hours).

raster; that is, each cell has a timestamp value TOA(cell) indicating the instant when the fire has reached that cell. This allows us to analyze how the fire spreads along time, since, for a given simulation time t, the fire has reached a cell if TOA(cell) $\leq t$.

In order to measure the accuracy of a given model, an unbiased criterion consists of comparing the TOA raster produced by Farsite with respect to the equivalent TOA raster obtained by each model. Thus, for a given time t, the amount of cells correctly estimated by each model is given by the sum of the recognized burning cells minus the amount of missed burning cells and minus the amount of cells wrongly estimated as burning. Sometimes, instead of using the absolute number of correct cells, we will use the correct cells rate, by normalizing that value over the total amount of burning cells (according to the Farsite output).

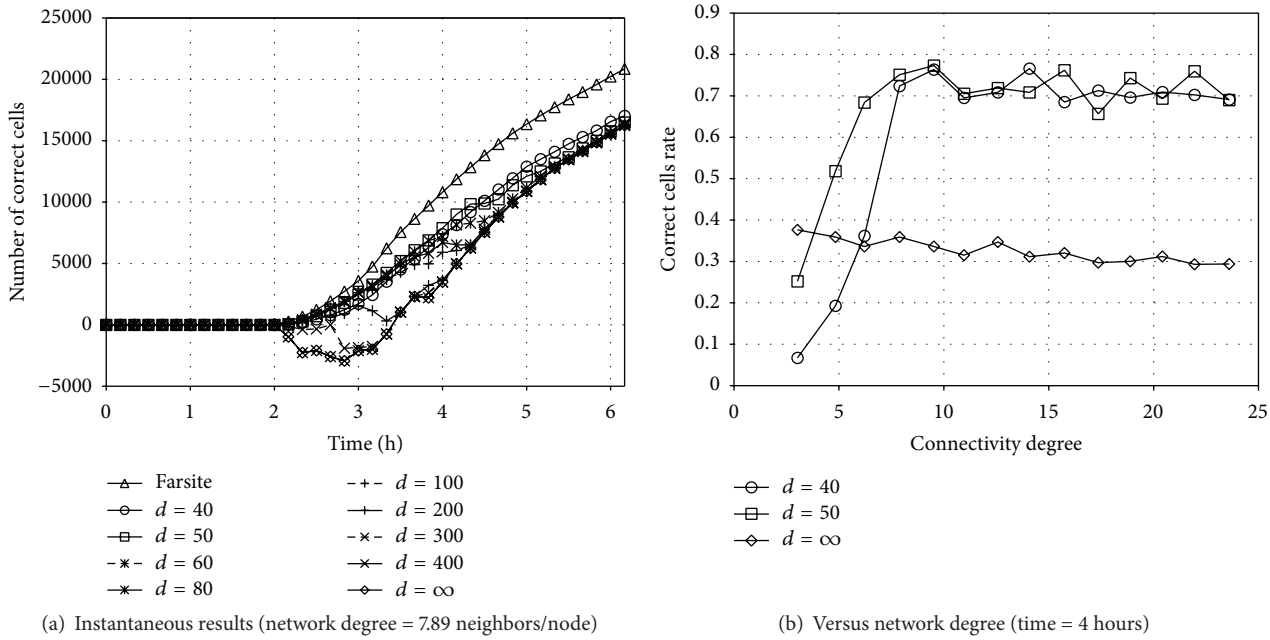
5.3. Tuning the Fire Approximation Models. In this section, we will evaluate the performance of the three distinct models to approximate the shape of the fire of Figure 11, while Section 5.4 will focus on an overall comparison.

5.3.1. Circle-Based Model. Figure 13(a) shows the quality of the circle-based model, by considering the use of homogeneous shapes, with different radius values (i.e., the parameter r in the model), expressed in units of raster cells. We can see that (as intuitively expected) bigger circles are suitable for lower network degrees and vice versa. From these results we have selected the best radius to be applied in function of the network degree; and we have approximated them by a logarithmic fire spread function $F(P, p) = 6.1419 - 1.283\ln(\text{Density}(P, c_p^n))$. More details about that may be found in [17].

Figure 13(b) shows the quality of heterogeneous shapes composed of circles with radius obtained applying the previously computed fire spread function. In this case, numerical values in the legend represent the radius of this area (i.e., the parameter n in the model), expressed in meters. Each node only knows the amount of neighbours located under its radio coverage (because of alternative communication protocols that broadcast node positions are not considered); therefore, the "N 50" series in this plot is the only one that corresponds to the use of heterogeneous shapes based on node density. On the other hand, as each node stores and relays all the received fire positions, they are able to compute heterogeneous shapes based on fire density even on areas bigger than the coverage area. For this reason, we have considered fire density areas with radius equal to 50, 75, and 100 meters.

Overall, we may conclude that the best results for heterogeneous shapes are obtained when they are based on node density, even if fire density was considered for bigger areas.

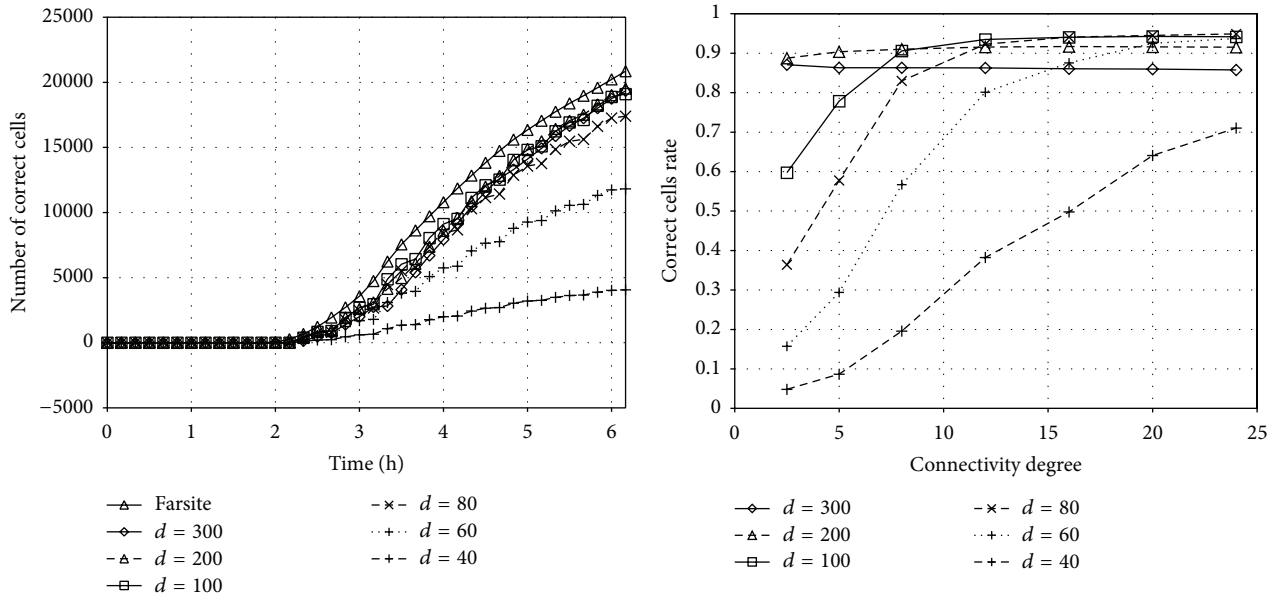
5.3.2. Hull-Based Model. Figure 14(a) shows the accuracy of the approximation obtained by the hull-based model, in function of the distance of fusion considered (i.e., the parameter d in the model), expressed in meters. The "Farsite" series shows the real amount of burning cells (verifying that $\text{TOA}_{\text{farsite}}[\text{cell}] \leq t$), representing an ideal upper bound for any fire approximation. The rest of series show the quality achieved by the multiple hull-based approximation. We can appreciate that the "$d = \infty$" series (that implies a shape composed of only one hull) offers a poor approximation to the fire, representing the lower bound for this technique. Additionally, the "$d = 400$" series performs similarly to the "$d = \infty$" series, making this analysis for higher values of d unnecessary. On the other hand, as d decreases, the accuracy of the representation increases. Additionally, we observe that the "$d = 40$" series exhibits a slightly worse behavior than



(a) Instantaneous results (network degree = 7.89 neighbors/node)

(b) Versus network degree (time = 4 hours)

Figure 14: Quality of the hull-based approximation.



(a) Instantaneous results (network degree = 7.89 neighbors/node)

(b) Versus network degree (time = 5 hours)

Figure 15: Quality of the polygon-based approximation.

the "$d = 50$" series during the first part of the simulation. For this reason, we do not continue analyzing smaller values for d.

Figure 14(b) shows the influence of network degree on the accuracy of the obtained approximation. The "$d = \infty$" series shows that an approximation based on a single convex hull is not negatively affected by low network densities (on the contrary, it even slightly improves). The reason is that errors produced by not covered burning areas are implicitly corrected as the hull grows. However, this approach is able to correctly estimate only 32% of a forest fire. On the other hand, beyond a certain density threshold, approximations based on multiple hulls are very accurate, correctly estimating up to 72% of a forest fire. Also, they (especially the "$d = 50$" series) are not significantly affected by low densities until the network is practically unconnected.

In conclusion, from both charts of Figure 14 we can state that a value for $d = 50$ meters is fair, assuming that

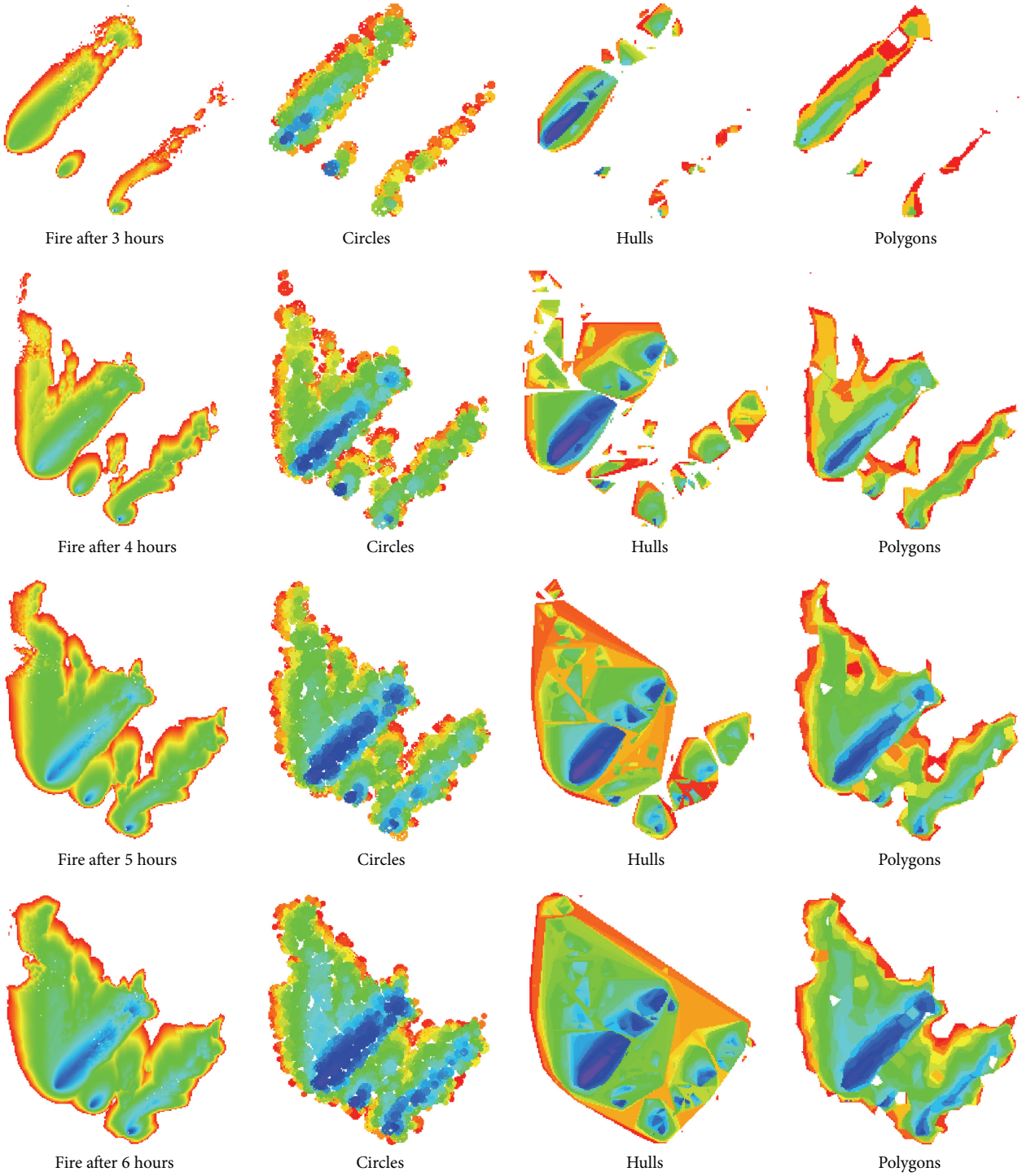
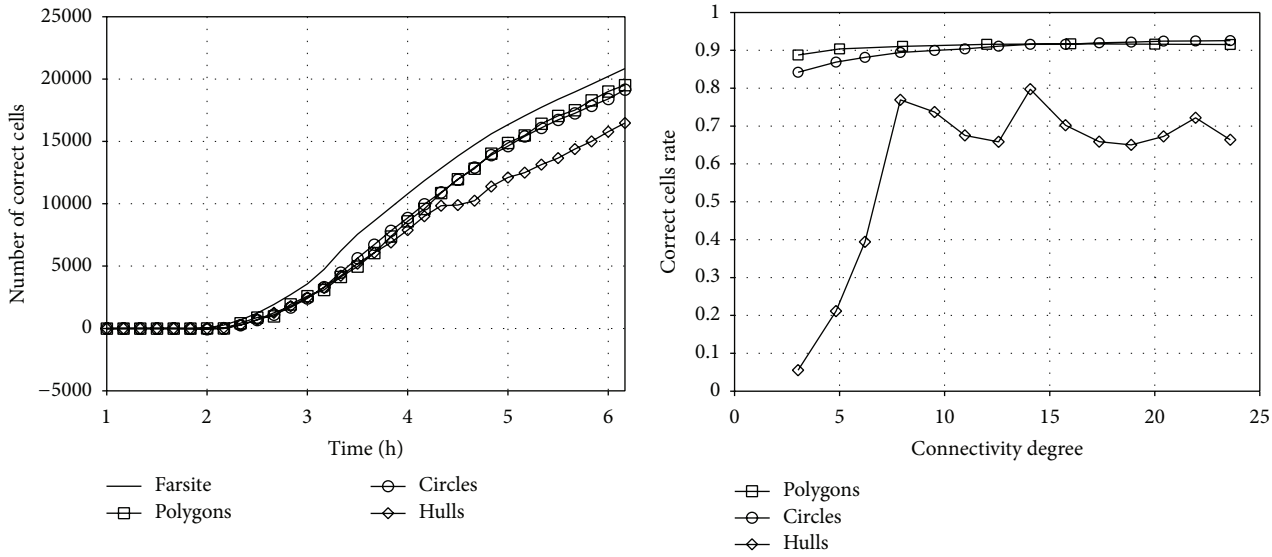


FIGURE 16: Aspect of the original fire (left column) and the corresponding approximations at different simulation times.



(a) Instantaneous results (network degree = 7.89 neighbors/node)

(b) Versus network degree (time = 5 hours)

Figure 17: Quality of the approximations.

the network connectivity degree is not too low. A more detailed analysis including memory requirements may be found in [18].

5.3.3. Polygon-Based Model. Figure 15 presents the results for the polygon-based model, by considering different values for the distance parameter (d), expressed in meters. In Figure 15(a) we can observe that, for low distances (40, 60, and 80 meters), the quality of the obtained approximation is poor. The reason is that the polygons in the shape do not merge, leading to lots of burning regions missed by the approximation. However, highest threshold distances are not the best option, since the improvement in the accuracy tends to become marginal.

In Figure 15(b) we can see that, in general, higher distances are less affected by network density (they are more stable). For sparse networks, the quality of the approximation grows up with distance. The reason is that low values of the distance lead to poor quality levels since several burning cells are not identified. On the other hand, as network degree increases, lower distance thresholds perform better. This is due to the fact that greater distance values lead to loss of resolution in the fire shape approximation; that is, fire "holes" (still not burning cells) are considered to be already burning. Therefore, a reasonable election for the following comparative analysis may be a threshold distance $d = 200$ meters, since it maintains a good quality for a wide range of densities.

5.4. Comparative Analysis. After tuning the optimal parameter values for each fire model, this subsection presents a comparative analysis. Before presenting the quantitative results obtained, Figure 16 shows the fire evolution presented in Figure 11 and the corresponding outputs provided by the analyzed proposals. At the beginning, the circle-based model overestimates the burning areas (reporting fire where there is not), whereas the other models underestimate them. As time evolves, we can appreciate that convex hulls grow and their fusion produce significant overestimations.

Figure 17 corroborates the previous appreciations by numerically showing the accuracy of the approximations provided by the analyzed fire models. In Figure 17(a) we can observe that the circle-based and polygon-based approximations provide the best results along the entire simulation run time, while, at the end of the simulation, when the fire has spread all over the area, the polygon-based approximation outperforms the circle-based one. In Figure 17(b) the influence of network degree on the quality of the approximation is shown. In this plot the same conclusion as before is confirmed. Circle-based and polygon-based approximations obtain accuracy levels close or above 90%, for all network densities, whereas the hull-based one only obtains about 70%. For sparser networks, the circle-based approximation underperforms the polygon-based one.

Finally, Figure 18 compares the amounts of node memory required by each model as the fire spreads. The plot shows that the circle-based approximation presents the highest memory requirements, since it needs all the information listened from the medium. On the other hand, the in-network aggregation proposals reduce these requirements to approximately 10%. Once the circle-based method has been discarded, we can see that hulls require less memory than polygons at the end of the simulation, when the shape representing the fire is composed of only a few large hulls.

6. Conclusions and Future Works

In this work, we have focused on the EIDOS platform, aimed at reducing human risks in forest firefighting operations. This system is based on a large WSN deployed from the air in the

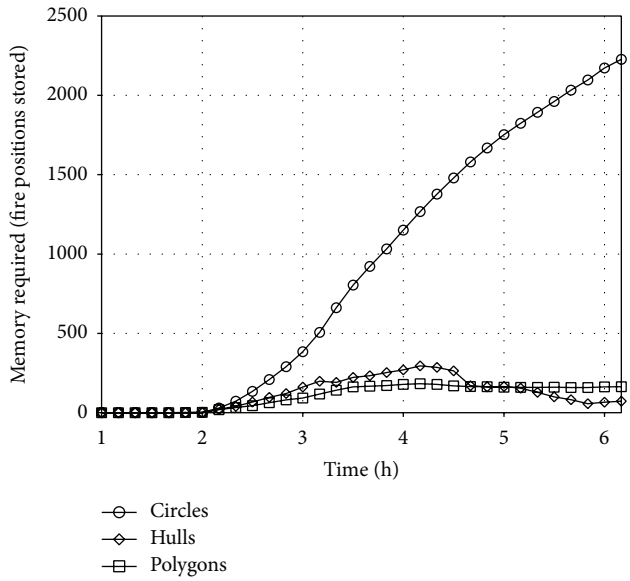


Figure 18: Instantaneous memory requirements (network size = 5000 nodes).

surroundings of the area affected by the fire. A communication layer allows the dissemination of fire detection events to the whole network. Starting from the information it listens to, each WSN node maintains an updated approximation of the current fire's shape. We have introduced three mathematical models for representing the fire, based on using circles, convex hulls, and arbitrary polygons. After tuning them, we have comparatively analyzed the quality of the outputs provided by these approaches. For this, we have compared them with the output provided by the Farsite fire simulation tool. Results clearly demonstrate that the proposed approach using the polygon-based method outperforms the other approaches. More in detail, while the circle-based and polygon-based models obtain accurate approximations of the fire and are close to each other, particularly for higher network densities, the circle-based model exhibits the highest memory storage and communications overhead requirements.

As future works, we plan to extend the fire models in order to handle 3D shapes. We also are going to evaluate the impact of localization and communication errors on the accuracy of the final approximations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partly supported by the Spanish MINECO and the European Commission (FEDER funds) under the Project TIN2012-38341-C04-04. This work was partially supported by National Funds through FCT (Portuguese Foundation for Science and Technology) and by ERDF (European Regional Development Fund) through COMPETE (Operational Programme "Thematic Factors of Competitiveness") within Project FCOMP-01-0124-FEDER-028990 (PATTERN) and by FCT and the EU ARTEMIS JU funding, within Project ARTEMIS/0004/2013, JU Grant no. 621353 (DEWI, http://www.dewi-project.eu).

References

[1] National Interagency Fire Center, "Total Wildland Fires and Acres (1960–2009)," http://www.nifc.gov/fireInfo/fireInfo_stats_totalFires.html.

[2] JRC Scientific and Technical Reports, *Report no 10 Forest Fires in Europe 2009*, Publications Office of the European Union, Luxembourg, 2009.

[3] G. Boustras and N. Boukas, "Forest fires' impact on tourism development: a comparative study of Greece and Cyprus," *Management of Environmental Quality*, vol. 24, no. 4, pp. 498–511, 2013.

[4] G. Boustras, N. Boukas, E. Katsaros, and A. Ziliaskopoulos, "Wildland fire preparedness in Greece and Cyprus; lessons learnt from the catastrophic fires of 2007 and beyond," in *Wildfire and Community: Facilitating Preparedness and Resilience*, D. Paton and F. Tedim, Eds., Charles C. Thomas, Springfield, Ill, USA, 2013.

[5] D. Adamis, V. Papanikolaou, R. C. Mellon, and G. Prodromitis, "P03-19—the impact of wildfires on mental health of residents in a rural area of Greece. A case control population based study," *European Psychiatry*, vol. 26, supplement 1, p. 1188, 2011, Proceedings of the 19th European Congress of Psychiatry.

[6] C. Psarros, C. G. Theleritis, S. Martinaki, and I.-D. Bergiannaki, "Traumatic reactions in firefighters after wildfires in Greece," *The Lancet*, vol. 371, no. 9609, 2008.

[7] Y. Liu, R. A. Kahn, A. Chaloulakou, and P. Koutrakis, "Analysis of the impact of the forest fires in August 2007 on air quality of Athens using multi-sensor aerosol remote sensing data, meteorology and surface observations," *Atmospheric Environment*, vol. 43, no. 21, pp. 3310–3318, 2009.

[8] F. Moreira, O. Viedma, M. Arianoutsou et al., "Landscape-wildfire interactions in southern Europe: implications for landscape management," *Journal of Environmental Management*, vol. 92, no. 10, pp. 2389–2402, 2011.

[9] H. Holeman, "Environmental problems caused by fires and firefighting agents," in *Fire Safety Science—Proceedings of the 4th International Symposium*, T. Kashiwagi, Ed., pp. 61–77, International Association for Fire Safety Science, Ottawa, Canada, 1994.

[10] Voice of America, "International Experts Study Ways to Fight Wildfires," http://www.voanews.com/content/a-13-2009-06-24-voa7-68788387/411212.html.

[11] G. Jakobson, J. Buford, and L. Lewis, "Guest editorial: situation management," *IEEE Communications Magazine*, vol. 48, no. 3, pp. 110–111, 2010.

[12] S. Nittel, "A survey of geosensor networks: advances in dynamic environmental monitoring," *Sensors*, vol. 9, no. 7, pp. 5664–5678, 2009.

[13] D. M. Doolin and N. Sitar, "Wireless sensors for wildfire monitoring," in *Smart Structures and Materials 2005: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, vol. 5765 of *Proceedings of SPIE*, pp. 477–484, San Diego, Calif, USA, March 2005.

[14] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, "FireWxNet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments," in *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys '06)*, pp. 28–41, ACM, Uppsala, Sweden, June 2006.

[15] E. M. García, A. Bermúdez, R. Casado, and F. J. Quiles, "Collaborative Data Processing for Forest Fire Fighting. In adjunct poster/demo," in *Proceedings of European Conference on Wireless Sensor Networks (EWSN '07)*, Delft, The Netherlands, 2007.

[16] E. M. García, M. Á. Serna, A. Bermúdez, and R. Casado, "Simulating a WSN-based wildfire fighting support system," in *Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA '08)*, pp. 896–902, Sydney, Australia, December 2008.

[17] M. A. Serna, A. Bermudez, and R. Casado, "Circle-based approximation to forest fires with distributed wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '13)*, pp. 4329–4334, April 2013.

[18] M. Á. Serna, A. Bermúdez, and R. Casado, "Hull-based approximation to forest fires with distributed wireless sensor networks," in *Proceedings of the IEEE 8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing: Sensing the Future (ISSNIP '13)*, pp. 265–270, April 2013.

[19] M. A. Serna, A. Bermúdez, R. Casado, and P. Kulakowski, "A convex hull-based approximation of forest fire shape with distributed wireless sensor networks," in *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '11)*, pp. 419–424, IEEE, Adelaide, Australia, December 2011.

[20] S. Srinivasan, S. Dattagupta, P. Kulkarni, and K. Ramamritham, "A survey of sensory data boundary estimation, covering and tracking techniques using collaborating sensors," *Pervasive and Mobile Computing*, vol. 8, no. 3, pp. 358–375, 2012.

[21] K. K. Chintalapudi and R. Govindan, "Localized edge detection in sensor fields," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 273–291, 2003.

[22] S. Duttagupta, K. Ramamritham, and P. Ramanathan, "Distributed boundary estimation using sensor network," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Sysetems (MASS '06)*, pp. 316–325, Vancouver, Canada, October 2006.

[23] M. I. Ham and M. A. Rodriguez, "A boundary approximation algorithm for distributed sensor networks," *International Journal of Sensor Networks*, vol. 8, no. 1, pp. 41–46, 2010.

[24] P.-K. Liao, M.-K. Change, and C.-C. J. Kuo, "A cross-layer approach to contour nodes inference with data fusion in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 2775–2779, March 2007.

[25] R. Nowak and U. Mitra, "Boundary estimation in sensor networks: theory and methods," in *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks (IPSN '03)*, pp. 80–95, Palo Alto, Calif, USA, 2003.

[26] Y. Xu, W.-C. Lee, and G. Mitchell, "CME: a contour mapping engine in wireless sensor networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS '08)*, pp. 133–140, IEEE, Beijing, China, July 2008.

[27] K. King and S. Nittel, "Efficient data collection and event boundary detection in wireless sensor networks using tiny models," in *Proceedings of the 6th International Conference on Geographic Information Science (GIScience '10)*, pp. 100–114, Springer, Zurich, Switzerland, 2010.

[28] W.-R. Chang, H.-T. Lin, and Z.-Z. Cheng, "CODA: a continuous object detection and tracking algorithm for wireless ad hoc sensor networks," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 168–174, January 2008.

[29] S.-W. Hong, S.-K. Noh, E. Lee, S. Park, and S.-H. Kim, "Energy-efficient predictive tracking for continuous objects in wireless sensor networks," in *Proceedings of the IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC '10)*, pp. 1725–1730, IEEE, Istanbul, Turkey, September 2010.

[30] S. Park, H. Park, E. Lee, and S.-H. Kim, "Reliable and flexible detection of large-scale phenomena on wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 6, pp. 933–936, 2012.

[31] C. Zhong and M. Worboys, "Continuous contour mapping in sensor networks," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 152–156, January 2008.

[32] Y. Li, S. W. Loke, and M. V. Ramakrishna, "Performance study of data stream approximation algorithms in wireless sensor networks," in *Proceedings of the 13th International Conference on Parallel and Distributed Systems (ICPADS '07)*, pp. 1–8, IEEE, Hsinchu, Taiwan, December 2007.

[33] S. Gandhi, J. Hershberger, and S. Suri, "Approximate isocontours and spatial summaries for sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 400–409, ACM, Cambridge, Mass, USA, April 2007.

[34] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 1–10, ACM, April 2004.

[35] G. Jin and S. Nittel, "Towards spatial window queries over continuous phenomena in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 4, pp. 559–571, 2008.

[36] G. Jin and S. Nittel, "Efficient tracking of 2D objects with spatiotemporal properties in wireless sensor networks," *Journal of Parallel and Distributed Databases*, vol. 29, no. 1-2, pp. 3–30, 2011.

[37] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[38] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[39] X. Zhu, R. Sarkar, J. Gao, and J. S. Mitchell, "Light-weight contour tracking in wireless sensor networks," in *Proceedings of the 27th Conference on Computer Communications (INFOCOM '08)*, pp. 1175–1183, IEEE, Phoenix, Ariz, USA, April 2008.

[40] N. A. A. Aziz and K. A. Aziz, "Managing disaster with wireless sensor networks," in *Proceedings of the 13th International Conference on Advanced Communication Technology: Smart Service Innovation through Mobile Interactivity (ICACT '11)*, pp. 202–207, IEEE, Dublin, Ireland, February 2011.

[41] R. I. D. Silva, V. D. D. Almeida, A. M. Poersch, and J. M. S. Nogueira, "Wireless sensor network for disaster management,"

in *Proceedings of the 12th IEEE/IFIP Network Operations and Management Symposium (NOMS '10)*, pp. 870–873, IEEE, Osaka, Japan, April 2010.

[42] S. George, W. Zhou, H. Chenji et al., "DistressNet: a wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Communications Magazine*, vol. 48, no. 3, pp. 128–136, 2010.

[43] S. Saha and M. Matsumoto, "A framework for disaster management system and WSN protocol for rescue operation," in *Proceedings of the IEEE Region 10 Conference (TENCON '07)*, pp. 1–4, IEEE, Taipei, Taiwan, November 2007.

[44] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. LaHusen, "Air-dropped sensor network for real-time high-fidelity volcano monitoring," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 305–318, ACM, Kraków, Poland, June 2009.

[45] A. A. Alkhatib, "A review on forest fire detection techniques," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 597368, 12 pages, 2014.

[46] T. Antoine-Santoni, J.-F. Santucci, E. de Gentili, X. Silvani, and F. Morandini, "Performance of a protected wireless sensor network in a fire. Analysis of fire spread and data transmission," *Sensors*, vol. 9, no. 8, pp. 5878–5893, 2009.

[47] Y. E. Aslan, I. Korpeoglu, and Ö. Ulusoy, "A framework for use of wireless sensor networks in forest fire detection and monitoring," *Computers, Environment and Urban Systems*, vol. 36, no. 6, pp. 614–625, 2012.

[48] K. Bouabdellah, H. Noureddine, and S. Larbi, "Using wireless sensor networks for reliable forest fires detection," *Procedia Computer Science*, vol. 19, pp. 794–801, 2013.

[49] J. Fernández-Berni, R. Carmona-Galán, J. F. Martínez-Carmona, and Á. Rodríguez-Vázquez, "Early forest fire detection by vision-enabled wireless sensor networks," *International Journal of Wildland Fire*, vol. 21, no. 8, pp. 938–949, 2012.

[50] M. Hefeeda and M. Bagheri, "Forest fire modeling and early detection using wireless sensor networks," *Ad-Hoc & Sensor Wireless Networks*, vol. 7, no. 3-4, pp. 169–224, 2009.

[51] P. S. Jadhav and V. U. Deshmukh, "Forest fire monitoring system based on ZIG-BEE wireless sensor network," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 12, pp. 187–191, 2012, http://www.ijetae.com/files/Volume2Issue12/IJETAE_1212_32.pdf.

[52] Y. Li, Z. Wang, and Y. Song, "Wireless sensor network design for wildfire monitoring," in *Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA '06)*, pp. 109–113, IEEE, Dalian, China, June 2006.

[53] J. Lloret, M. Garcia, D. Bri, and S. Sendra, "A wireless sensor network deployment for rural and forest fire detection and verification," *Sensors*, vol. 9, no. 11, pp. 8722–8747, 2009.

[54] B. Son, Y. Her, and J. Kim, "A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains," *International Journal of Computer Science and Network Security*, vol. 6, no. 9, pp. 124–130, 2006.

[55] U. Mansoor and H. M. Ammari, "Chapter 9: localization in three-dimensional wireless sensor networks," in *The Art of Wireless Sensor Networks: Volume 2: Advanced Topics and Applications*, H. M. Ammari, Ed., Signals and Communication Technology, pp. 325–363, Springer, Berlin, Germany, 2014.

[56] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[57] S. Tennina, M. Di Renzo, F. Graziosi, and F. Santucci, "Chapter 8. Distributed localization algorithms for wireless sensor networks: from design methodology to experimental validation," in *Wireless Sensor Networks*, S. Tarannum, Ed., InTech, 2011.

[58] S. Tennina, M. Di Renzo, F. Graziosi, and F. Santucci, "ESD: a novel optimisation algorithm for positioning estimation of WSNs in GPS-denied environments—from simulation to experimentation," *International Journal of Sensor Networks*, vol. 6, no. 3-4, pp. 131–156, 2009.

[59] E. M. García, A. Bermúdez, and R. Casado, "Range-free localization for air-dropped WSNs by filtering neighborhood estimation improvements," in *Proceedings of the 1st International Conference on Computer Science and Information Technology (CCSIT '11)*, pp. 325–337, Springer, Bangalore, India, 2011.

[60] F. J. Ovalle-Martínez, A. Nayak, I. Stojmenović, J. Carle, and D. Simplot-Ryl, "Area-based beaconless reliable broadcasting in sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 1-2, pp. 20–33, 2006.

[61] M. Á. Serna, E. M. García, A. Bermúdez, and R. Casado, "Information dissemination in WSNs applied to physical phenomena tracking," in *Proceedings of the 4th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '10)*, pp. 458–463, Florence, Italy, October 2010.

[62] F. P. Preparata and M. I. Shamos, *Computational Geometry*, Texts and Monographs in Computer Science, Springer, New York, NY, USA, 1985.

[63] MySQL, MySQL website, 2014, http://www.mysql.com/.

[64] Firemodels.org, "Farsite fire area simulator," 2014, http://www.firemodels.org/index.php/national-systems/farsite.

[65] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire TinyOS applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 126–137, ACM, Los Angeles, Calif, USA, November 2003.

[66] Adobe Flash, 2014, http://www.adobe.com/products/flash.html.

[67] H. T. Friis, "A note on a simple transmission formula," in *Proceedings of the I.R.E. and Waves and Electrons*, May 1946.

[68] MEMSIC, "MEMSIC Wireless Sensor Networks products," 2014, http://www.memsic.com/.