Cooperative Boundary Detection in a Geosensor Network

Monika Sester

Institut für Kartographie und Geoinformatik, Leibniz Universität Hannover, Appelstraße 9a, 30167 Hannover, Germany, monika.sester@ikg.uni-hannover.de

Keywords: Geosensor networks, decentralized algorithm

Abstract

Geosensor networks can be applied to detect spatio-temporal phenomena in potentially large areas. The fundamental idea is that sensors are distributed in the environment. They can locally cooperate in order to detect higher level phenomena that a single sensor would not be able to detect. Such functionality is based on the principle, that the individual sensors do have a (often limited) computing power and communication range. In order to infer higher level information they have to locally cooperate. There are several examples for the application of geosensor networks, e.g. environment, traffic, military. The advantage of using such a geosensor network as opposed to a centralized system is its scalability, its reliability and fault tolerance.

The challenge is to devise algorithms that are able to operate locally and still achieve a common global solution. In the paper, an approach for identifying the boundary of a spatial phenomenon is presented. It starts from the assumption that sensors are dynamic and are able to move. The sensors try to sample and identify the boundary of the phenomenon by cooperatively covering the area. This is achieved using a Neural Network approach, a Self Organizing Map (SOM).

Introduction and Overview

Geosensor networks are composed of individual sensors with measuring, positioning and communication capabilities. Through cooperation of neighboring sensors the whole network is able to perform actions that go beyond an individual sensor’s capabilities. The advantages of geosensor networks lie in their scalability and also in their fault tolerance, as the role of individual sensors is not crucial - due to the high redundancy. These properties lead to a large number of applications of geosensor networks such as environmental monitoring or military applications.

From a computational and geoinformatics point of view, the challenge is to devise algorithms that are able to work locally and still achieve a common global solution. There are many spatial algorithms that operate in a centralized manner, presuming access to all the information; however, in the case where a local processing unit only has a limited view of the surrounding information, existing algorithms have to be adapted or new ones have to be devised to achieve a decentralized processing.

In the paper, an approach for identifying the boundary of a spatial phenomenon in a decentralized fashion using a geosensor network will be presented. The problem is e.g. that an oil spill has to be detected, or an area, which moves relative to its local environment, like a hill slide. The boundary information has to be derived through cooperative local processing of individual sensors that are able to measure the phenomenon. The boundary is characterized by a change in the value of the phenomenon, e.g. a binary value of presence or absence of the phenomenon. The approach starts with the assumption of having moving sensors. These sensors try to identify the boundary of the phenomenon by cooperatively covering the area. Each sensor is able to communicate to its neighboring sensors and to check if the measured phenomenon has a different value. If yes, the boundary of the phenomenon has to be in between the two sensors. In order to detect the true boundary, and at the same time be able to better delineate its possibly complex form, the area in the vicinity of the boundary has to be populated with more sensors. This is achieved using the approach of a Self Organizing Map (SOM), which goes back to Kohonen (1982). A SOM, a kind of Neural Network, is able to approximate a phenomenon by arranging a given distribution of neurons in a way that areas with high variability are being higher resolved than areas where not much happens. This is achieved by an iterative approximation.

The paper is organized as follows. First, related work in the domain of geosensor networks is presented; then the background of SOM is briefly described. Following this, the new approach for approximating and delineating a phenomenon boundary in a decentralized approach is presented. Examples show the validity and the potential of the method. Finally, an outlook on future work is given.

2 Related work

A general overview of wireless sensor networks is given in (Akyildiz, Su, Sankarasubramaniam & Cayirci 2002). Geosensor networks for the observation and monitoring of environmental phenomena are a recent trend in GIScience. Traditional geodetic networks consist of a fixed set of dedicated sensors with a given configuration and measurement regime. The processing of the data is usually done in a centralized fashion. The advent of geosensor networks brings about the chance to move from a centralized approach to an approach using distributed sensors with computation and communication capabilities (Stefanidis & Nittel 2004). The basic advantages of the decentralized approach is its scalability and fault tolerance, as the whole system does not have to rely on a single sensor, which can easily be replaced by a neighboring one, due to the high redundancy. While geosensor networks can also be operated in a centralized way - which is often done in today's approaches - the true challenge and potential lies in the decentralized processing. This way of data processing also takes into account the "first law of Geography" by Tobler (Tobler 1970), which states, that everything is related to everything else, but near things are more related than distant things. In this way, the immediate surroundings of a sensor is assumed to have the highest influence on the reaction of a sensor, and therefore can be evaluated locally and does not necessarily have to be transmitted to a central server for processing.

Decentralized algorithms for geosensor networks have been investigated by several researchers and for different applications. Laube, Duckham & Wolle (2008) describe an algorithm to detect a moving point pattern, namely a so-called flock pattern. A flock is described as a group of objects that moves in a certain distance over a certain time. In a similar spirit, Laube & Duckham (2009) present a method for the detection of clusters in a decentralized way. Depending on the communication range, clusters of a certain size (radius) can be detected.

Walkowski (2008) presents an approach for the optimal arrangement of geosensor nodes in order to correctly describe an underlying temporally varying phenomenon, like a toxic cloud. He assumes to have sensors that are able to move; however, the determination of the locations of lacking information has to be determined in a centralized fashion. Zou & Chakrabarty (2004) describe an approach to optimally cover an area with a given set of sensors. This approach tries to find an adequate spacing among the sensors; the influence of the environment can be modeled with attractive or repulsive forces.

Density based spatial data approximation can be used in different applications. In Sester (2008) it has been used for the typification of buildings in generalization.

3. Fundamentals of Kohonen Feature Maps, Self Organizing Maps (SOMs)

Given an input space E of dimension m with stimuli x, and a map space A of dimension d with neurons. The dimension d is usually 1 or 2. Each neuron in map space is described by a tuple U = (w, p), i.e. a weight w from E and a position p from A. From the position p the neighbors of the neuron can be determined. The weight w indicates, to which stimulus the neuron reacts most. Often, the neuron positions are initially given on a regular equidistant grid, because usually no information about the underlying distribution of the phenomenon is available.

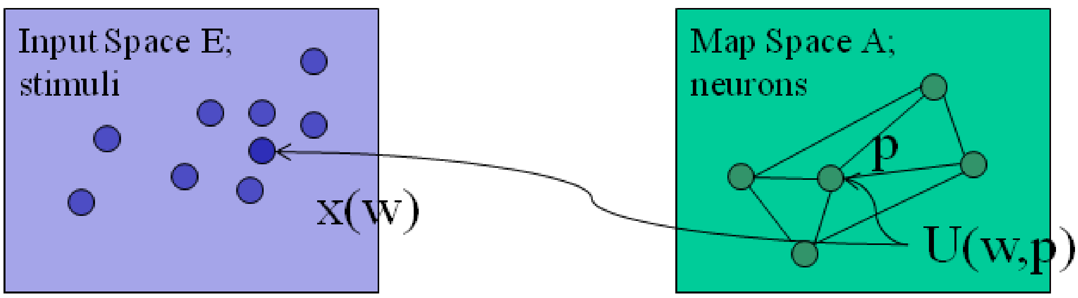

Figure 1: Schematic description of role of stimuli and neurons.

The algorithm is an iterative process and interplay of stimuli, which fire and thus attract neurons that react to them. It can be described as follows:

- Fire: Random selection and firing of stimulus v.
- Response: determination of neuron Uc with a weight that is most similar to the stimulus' weight. Similarity is determined based on spatial proximity. It can be

calculated with the Euclidean distance between stimulus and all other neuron weights; the neuron with the smallest distance is selected.

$$|v - w_c| \leq |v - w_r| \quad \forall\, r \in A$$

- Subsequently, the weights of the neuron and its neighbors are adapted in a way that they get more similar to the stimulus, i.e. move towards the stimulus.

  $$w_r^{new} = w_r^{old} + \mu h\left(v - w_r^{old}\right)$$

  This adaptation depends on the parameters learning rate η and influence range h (see below).

- These learning steps are executed iteratively, until a termination criterion is reached.

Figure 2 visualizes this situation using a one-dimensional situation: a stimulus (dark) fires and thus attracts the neighboring neurons. The middle neuron, whose weight (position) is closest to the stimulus, is adapted stronger than its neighboring neurons. In this way, there is a movement of the neurons in the vicinity of the stimulus towards it - however with different intensity, i.e. the closest one performs the largest moves and the more distant ones move less. The stimulus thus has the effect that the neuron chain is iteratively moving towards the stimulus.

Figure 2: One-dimensional situation: stimulus (dark color) triggers a change of weights (position) of the neuron chain (in light color) and their neighbors; result of adaptation (right).

In the course of the iterations, the moving possibilities of the neurons are reduced, with the aim of an increasing settlement to a final state. The corresponding control parameters are the learning rate η(t) and the influence range function h(*dist*, σ(t)), which describes the weighting of the winner neuron and its neighbors as a function of the neighborhood range *dist* and the number of iteration epochs t.

As influence range function h(*dist*, σ) any function can be chosen that has its maximum at *dist* = 0 and converges to zero with large values of dist. In the simplest case this can be a piecewise constant function which yields the value 1 for $x \leq \sigma$ and 0 else. Another appropriate choice is the Gaussian function:

$$h(dist, \sigma) = e^{-dist^2 / \sigma^2}$$

whereby the radius σ varies depending on training duration: in order to develop the coarse structure of the map, it should be high in the beginning; later however, it has to be reduced in order to develop the local fine structure. Thus, a function σ(t) should be chosen, which decreases with the number of iterations (learning steps t). The influence range of the function is therefore high in the beginning in order to allow for a coarse distribution of the neurons. With growing number of iterations the influence range reduces and thus has the effect that the changes are only local in the end and thus converge to a stable situation.

4. Approach for Boundary detection and delineation

4.1 Assumptions and prerequisites

The proposed approach uses some general terms and starts with some general assumptions which are described in the following. The phenomenon is described as a spatial function, which takes the binary values 1 and 0, indicating the presence and absence of the phenomenon, respectively. This restriction to binary values can be extended easily to other values, as will be shown in the outlook.

A set of sensors which are able to measure the phenomenon in the described way is distributed into the environment. The sensors are able to move. They measure the phenomenon at different locations and can communicate to their neighbors in a certain range.

The geosensor network is modeled as a graph G = (V,E), where V is the set of vertices, in our case sensor nodes, and the edges E denote the communication between the nodes. Each sensor node has a given position (x,y) at a certain time. The graph of the sensor network is set up by the communication possibilities between the neighboring nodes. One approach is to assume that a communication is possible to all the nearest neighbors; in this case the graph can be constructed using a Delaunay triangulation.

The boundary of a phenomenon is the location, where the presence of the phenomenon changes from true to false (i.e. 1 to 0) or vice versa.

4.2 Boundaries in a well distributed and dense network

A boundary between two neighboring nodes can be detected by checking, if the status of phenomenon presence changes. As there is no further information as to where this change takes place on the edge between two neighboring nodes, we assume that it is in the middle between them. The boundary then can be found by connecting all boundary nodes until the start node is reached again. This is visualized in Figure 3. The areal phenomenon together with the distribution of the sensors and their measurement is shown on the left, and the detected boundary using the strategy described is given in the middle; on the right the overlay of the detected boundary and the original phenomenon is given.

It is obvious, that it is possible to detect the boundary using this approach. However, it depends on the distribution and density of the sensors. If they are both dense and adequately distributed over the whole phenomenon and especially along the boundary, then a more precise detection is possible. The accuracy will, however, always be limited to half the distance between two nodes.

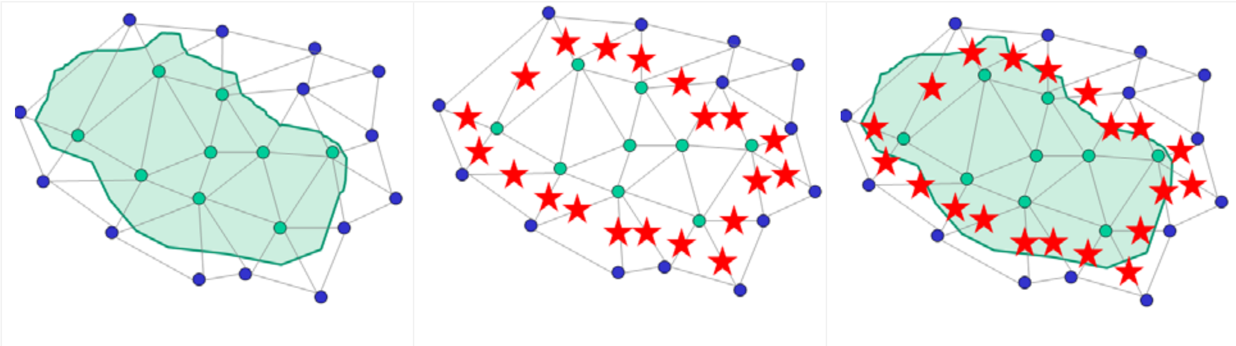

Figure 3: Detection of the boundary of a phenomenon: areal phenomenon with initial sensor distribution (left), boundary edges and approximate boundary points (stars) (middle), overlaid over phenomoneon (right).

In order to improve the accuracy of the delineation of the boundary, in the following section an approach is presented which is able to better approximate the boundary by allowing the sensors to move into the direction of the boundary. The movement of the sensors will be controlled by the SOM.

4.3 Iterative Adaptation of Boundary using SOM

As a priori the location of the boundary is not known, the idea of the approach is to detect it by a cooperative effort of locally communicating sensors. The SOM offers the basis for this. Its basic ability is to sample a phenomenon by allocating more neurons to the sensitive areas indicated by a change in the presence of the phenomenon.

In order to adapt the SOM to the given problem, the following prerequisites have to be given. First of all, it has to be determined, what are neurons and what are stimuli. Then the attraction range and the movement behavior of the neurons has to be described. In our case, all sensors act as neuron and as stimulus at the same time. This means, that a sensor A fires and attracts sensors N_A in its local neighborhood (see Figure 4). The type of attraction depends on the sensor measurement (which will be termed type of sensor t_A for brief) of the neighboring sensor:

- If the neighbor is of a different type (t_A != t_B), then it will move towards the firing stimulus.
- If the neighbor is of the same type, then it will perform the same movements as the firing stimulus.

This behavior is motivated by the underlying assumption that the boundary is in between two sensors of different type. Thus, if two neighboring nodes of different type are detected, then they can approximate the boundary more exactly by moving towards each other. Therefore, not only neuron A moves into the direction of the closest neighbor of different type B (together with its neighbors of same type), but also neighbor B will walk towards neuron A - also together with its neighbors of same type. In this way, both neurons try to approximate the boundary by moving towards it.

A special case has to be taken into account: if a neuron fires and does not find any neighbor which is of different type, then there is no stimulus to walk into any direction. In order to reach a status, where each sensor has at least one neighbor of a different type, there is the possibility of a random movement of this neuron into an arbitrary direction and with an arbitrary distance, in order to randomly explore the possible boundary.

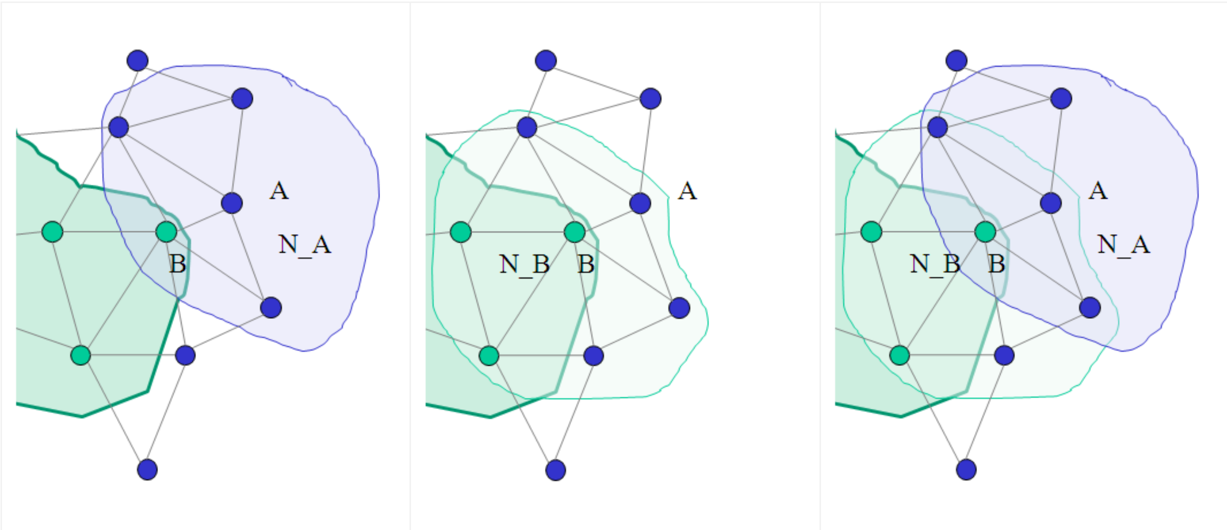

Figure 4: Neighborhood of firing neuron A: NA; closest neighboring sensor of different type B, with its neighbors NB.

The pseudo-code of the approach is as follows:

```
0) until convergence

1) random selection of Neuron A, of type t_A

2) neighbors of A: n_A

3) select neighbor from n_A, which is of different type of A and
   closest to A: Neuron B and its neighbors n_B

4) if there are no neighbors of different type -> goto case 1 (8)

5) move A and its neighbors of same type from n_A towards B

6) move B and its neighbors of same type from n_B towards A

7) continue with 0

8) case 1: move A and its neighbors in arbitrary direction in
   order to randomly explore the boundary

9) continue with 0
```

The movements of the neurons and their neighbors is illustrated in the Figure 5: neighbors of the same type of A are moving into the direction of B (left); neighbors of same type of B are moving towards A (middle); on the right both movements are shown.

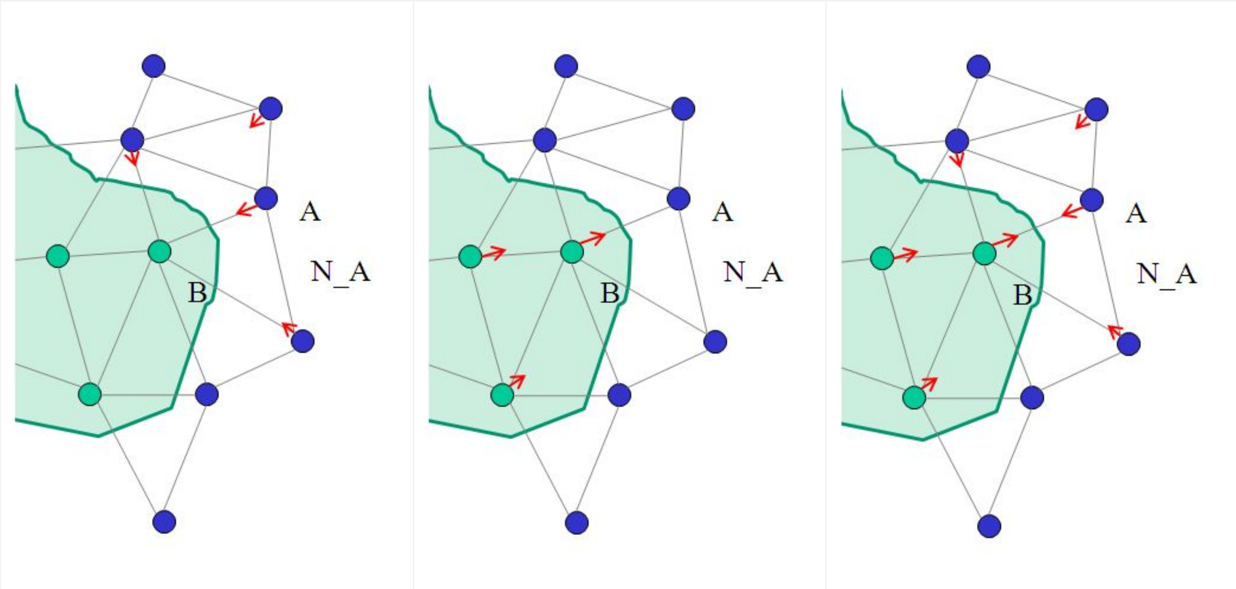

Figure 5: Movement of neurons A and B and their neighbors of same type NA (left) and NB (middle); right: both movements.

After the convergence of the system (i.e. no more movements of the neurons) the boundary detection is finished. The boundary neurons are labeled and communicate their position to the central node. At this stage it is possible to use the interpolation from 4.2 to determine the position of the boundary in between two nodes. This is shown in Figure 6.

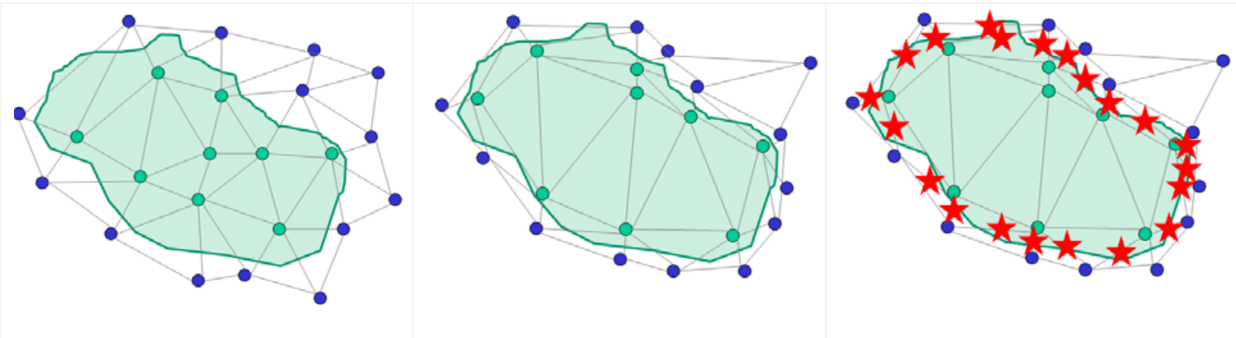
Figure 6: Detection of the boundary of a phenomenon: areal phenomenon and initial sensor distribution (left), sensor distribution after convergence of SOM (middle), approximate boundary points (stars) (right).

5 Examples and Results

In the following, some examples with a first implementation of this concept are shown. Given are a spatial phenomenon and a set of n sensors that is distributed randomly over the hypothesized area of the phenomenon. All the examples are calculated with the same parameters:

- the number of neurons (sensors) is n =40
- the neighborhood is determined using a Delaunay Triangulation
- a one-hop neighborhood is used, which is additionally limited at a distance threshold of max. 400 m, i.e. neighboring sensors beyond this value are not considered
- the movement of the sensors is increasingly reduced with growing number of iterations (function h)
- the number of iterations is 4 ∗ n, i.e. each sensor can fire in average four times

The first example shows a standard situation. Figure 7 presents the initial situation on the left, the movement of the sensors in the middle, and the position of the sensors after the adaptation on the right. The color of the sensor points indicate their position: in yellow are sensors that are located on the boundary; in blue are sensors inside the phenomenon, in red are sensors outside the boundary of the phenomenon. Initially some sensors lie outside and some inside the phenomenon. The partially large random movement of some sensors has the positive effect that also in areas where no sensor was located initially the boundary can be delineated. Not all the sensors were able to capture the boundary. Some “stranded” in the middle of the phenomenon and were not able to recover by being dragged by another neighboring sensor. However, the majority of the sensors is distributed along the boundary. A drawback is that some of the sensors cluster at certain places on the boundary - instead of being evenly distributed along the boundary.

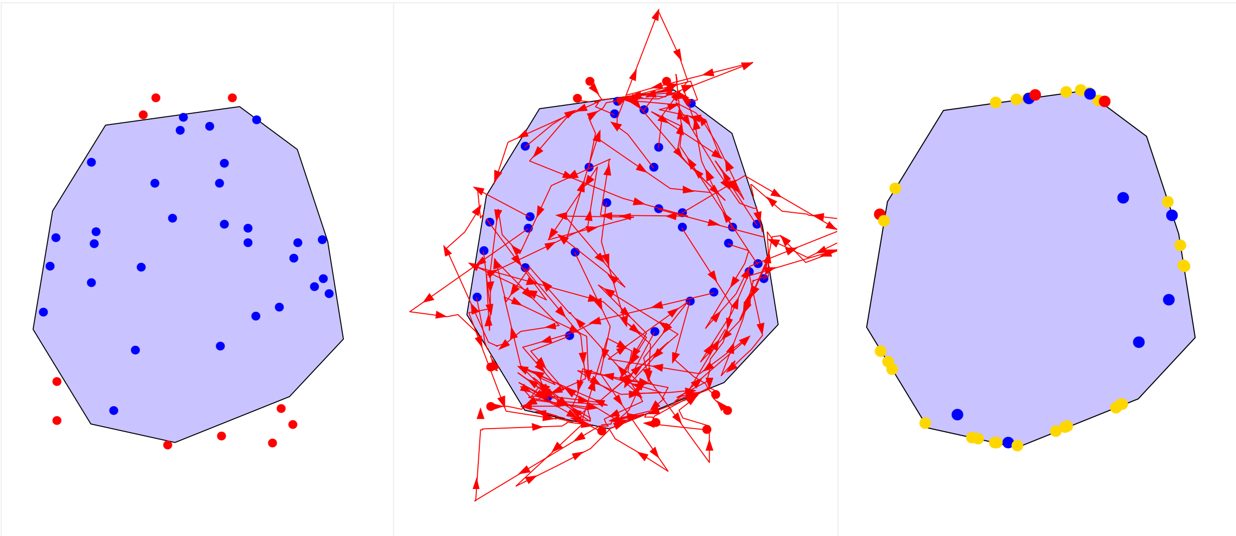

Figure 7: Detection of the boundary of a phenomenon: areal phenomenon and initial sensor distribution (left), movement of neurons (middle) approximate boundary points in yellow (right).

The following example in Figure 8 shows a situation, where the distribution of sensors in the beginning was unequal, i.e. did not fully cover the phenomenon, namely the lower right corner. Still, however, the system is able to capture and delineate the boundary.

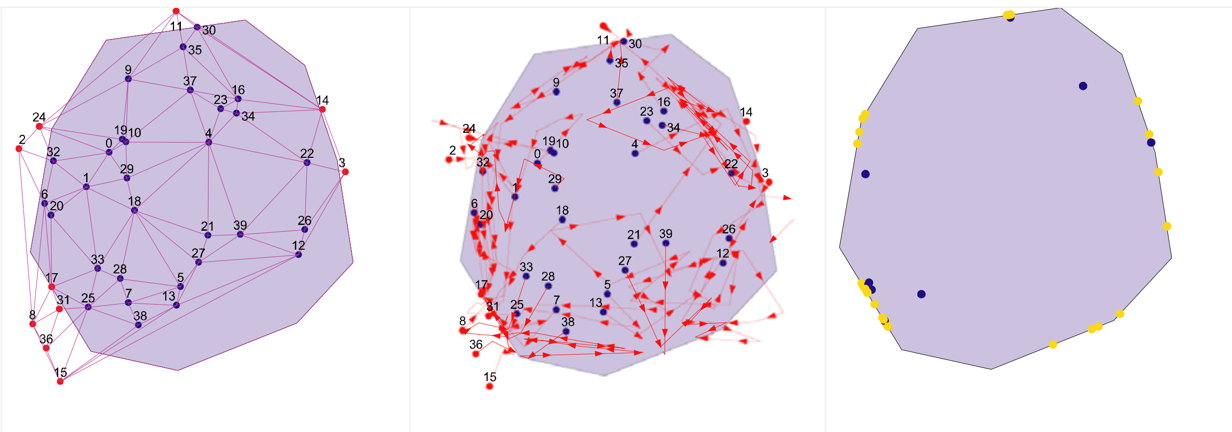

Figure 8: Detection of the boundary of a phenomenon with unequal initial distribution of sensors (lower right is not covered): areal phenomenon and initial sensor distribution (left), movement of neurons (middle) approximate boundary points in yellow (right).

The next example in Figure 9 shows a concave phenomenon. The example clearly demonstrates that also a boundary of this shape can be delineated by the sensors. Also here, the distribution of the sensors is not equal in the beginning: there are no sensors to the right and up of the phenomenon. The exploratory character of the SOM, however, leads to the coverage of the phenomenon and the delineation of its boundary.

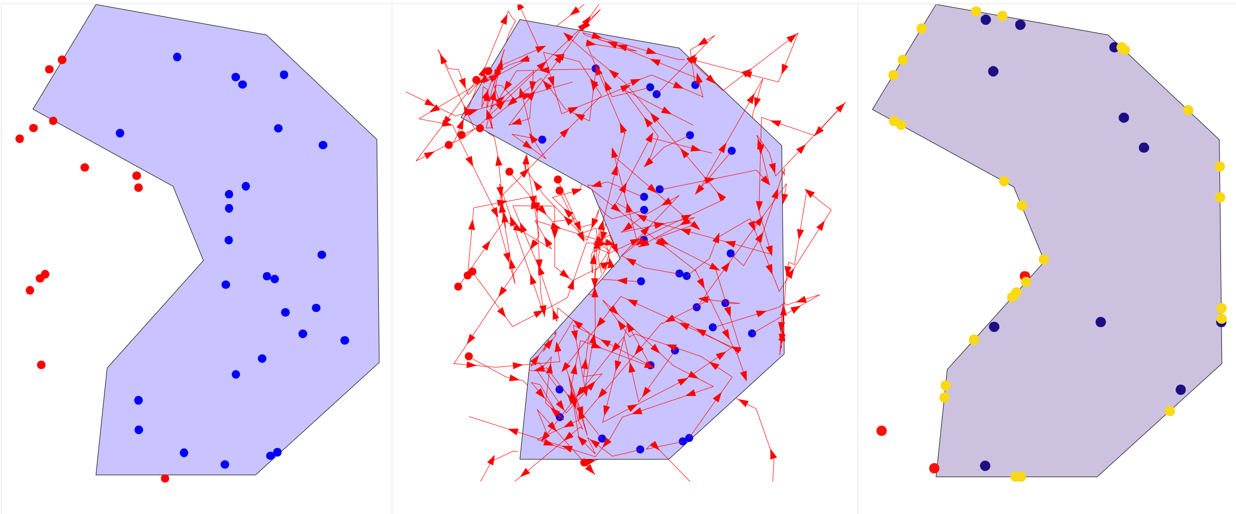

Figure 9: Detection of the boundary of a phenomenon: areal phenomenon and initial sensor distribution (left), movement of neurons (middle) and approximate boundary points in yellow (right).

5.1 Discussion and outlook on future work

The first implementation using the parameters described above show the feasibility of the approach. After applying the SOM, the boundary of the phenomena can be better delineated. There are, however, also some investigations that are needed in order to improve the results and also evaluate and analyze them thoroughly with respect to performance characteristics.

Parameters: The experiments have been conducted with a fixed set of parameters and functions. Other parameters, e.g. concerning number of iterations and learning rate and influence range function should be tested.

Dependency on communication range: In the current implementation, only the nearest neighbors have been used, corresponding to a single hop communication. Test should show if the results could be improved when a higher communication range is used.

Clustering of Sensors: The fact that sensors on opposite sides of the boundary move towards each other leads to the phenomenon that they have the tendency to cluster at boundary points. As an improvement, a concentration of sensors on one point in space has to be prohibited leading to a more even distribution along the boundary.

Evaluation of resources: An important issue in geosensor networks is the efficient management of the available resources. The different resources of the sensors have to be evaluated against each other: communication, computation and movement. Larger movement rates may lead to a higher probability of better coverage of the phenomenon, however, also to a higher consumption of resources.

Selection of initial neuron distribution: In the proposed concept, the initial distribution is a two-dimensional irregular grid. As the phenomenon to be detected is a linear structure (the boundary), another possible approach could be to use a linear chain of neurons.

Binary vs. continuous phenomenon measurements: In this approach there was the assumption that the phenomenon is binary. If it is of continuous nature, then appropriate measures for the change in phenomenon have to be found, i.e. threshold values, which define that two measurements are dissimilar. Such thresholds for gradients could be calculated in the local environment of the sensors. Values which are sufficiently different from mean values are candidates for changes in the phenomenon. Such thresholds could also be propagated through the network and thus be used collaboratively.

References

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002), 'Wireless sensor networks: a survey', Comput. Netw. 38(4), 393–422.

Kohonen, T. (1982), 'Self-organized formation of topologically correct features maps', Biological Cybernetics 42, 59–69.

Laube, P. & Duckham, M. (2009), Decentralized spatial data mining in distributed systems, in H. Miller & J. Han, eds, 'Geographic Knowledge Discovery, Second Edition', CRC, Boca Raton, FL, pp. 211–220.

Laube, P., Duckham, M. & Wolle, T. (2008), Decentralized movement pattern detection amongst mobile geosensor nodes, in T. Cova, K. Beard, M. Goodchild & A. Frank, eds, 'Lecture Notes in Computer Science 5266', Springer, Berlin, pp. 211–220.

Sester, M. (2008), Self-organizing maps for density-preserving reduction of objects in cartographic generalization, in P. Agarval & A. Skupin, eds, 'Self-Organising Maps. Applications in GI Science', John Wiley and Sons, pp. 30–40.

Stefanidis, A. & Nittel, S., eds (2004), Geosensor Networks, CRC Press.

Tobler, W. (1970), 'A computer movie simulating urban growth in the Detroit region', Economic Geography 46(2), 234–240.

Walkowski, A. C. (2008), Model based optimization of mobile geosensor networks, in L. Bernard, A. Friis-Christensen & H. Pundt, eds, 'AGILE Conf.', Lecture Notes in Geoinformation and Cartography, Springer, pp. 51–66. 13

Zou, Y. & Chakrabarty, K. (2004), 'Sensor deployment and target localization in distributed sensor networks', ACM Trans. Embed. Comput. Syst. 3(1), 61–91.