Acta Scientific COMPUTER SCIENCES

Volume 5 Issue 9 September 2023

Research Article

Positioning Surface Vessels for Underwater Communication using Self Organizing Maps

Cris Koutsougeras[1], Mohammad Y Saadeh[2] and Nikolaos I Xiros[3]*

[1]Department of Computer Science, Southeastern Louisiana University, Hammond, Louisiana, USA

2Department of Industrial and Engineering Technology, Southeastern Louisiana University, Hammond, Louisiana, USA

[3]Bollinger School of Naval Architecture and Marine Engineering Department of Electrical and Computer Engineering University of New Orleans, USA

***Corresponding Author:** Nikolaos I Xiros, Bollinger School of Naval Architecture and Marine Engineering Department of Electrical and Computer Engineering University of New Orleans, USA.

Received: August 08, 2023

Published: August 17, 2023

Abstract

This work concerns communications for underwater (submerged) vessels. The stage consists of a set of underwater exploration vessels which are exploring cooperatively and need to communicate. The range of underwater signals is rather short, so a set of surface vessels (boats) is used to facilitate the communications. The surface vessels must position themselves properly so as to minimize the distance which signals must travel. Each surface vessel serves one group of the submerged vessels so this distance metric has to be optimized among all such groups (one for each surface vessel). As the underwater vessels move around, the surface vessels must also continually reposition accordingly. The issue at hand is how should the movement of the surface vessels be coordinated so as to optimize communications. In addition, the grouping (which of the submerged vessels is served by which surface boat) is not static; as the submerged vessels move around, this grouping may change dynamically. This complicates the problem and a good method is needed to handle the problem of controlling the topological distribution of the set of water surface boats in response to the independent movements of the submerged vessels.

This work is an application of unsupervised intelligent methods for the purpose of controlling/planning the surface boats positions in a coordinated way without centralized decision making. The self organizing maps are adopted to make the topological distribution of the surface vessels simulate the density distribution of the submerged vessels. The results are analyzed in the light of specific measures which relate to communication efficiency and clarity.

Keywords: Underwater Exploration; Self Organizing Maps; Unsupervised Intelligent Methods

Introduction

Underwater communications and networking are recognized as critical tools for underwater exploration, subsea resource extraction, scientific data gathering, resource exploration, coastal or ocean environmental surveillance etc [1]. A range of technologies is being developed specifically for underwater communications [1-3]. In underwater works or explorations where multiple submerged vessels are involved, there is a need for communication which suffers from the range of such communications. Electromagnetic signals as well as sound signals deteriorate quite fast with the distance they have to travel, so the range of underwater signals is poor. The underwater communication must also be linked to over-the-air communications to distant stations such as over satellite links. To remedy this, vessels on the water surface may be used to facilitate the communication. A small number of surface vessels (compared to the number of submerged vessels) facilitate the communication needs of submerged vessels. The submerged vessels may communicate with other submerged ones which are in relative proximity but not with distant ones if the exploration area is large, and in any case, underwater signals must be collected by surface vessels which then re-transmit over the air. Subsequently, this generates the need for the surface vessels to position themselves so as to optimize the communication distances from the underwater ones. As the underwater vessels move, the surface ones must (continually) reposition to maintain some optimal topological distribution that optimizes the communication distances.

Thus, in this work the environment stage consists of a set of underwater exploration vessels (hereafter referred to as "explorers") which are exploring cooperatively and need to communicate, and a small number of surface vessels (hereafter referred to as "followers") which are following the "swarm" of the underwater vessels and are used for relaying communications. The issue at hand is how to control/determine the movement of the followers so as to maintain a distribution which optimize communications as the explorers move around. It is also assumed that each of the followers is to become the communications coordinator for a subset of the explorers and it will then become dedicated to facilitating the communication needs for those explorers. Essentially, this imposes a clustering of the explorers into subsets of topological vicinities; there is one cluster of explorers for each follower vessel F_i and the region which they cover becomes the exclusive service region of F_i. Given that underwater communications suffer with distance, the logical definition of the service cluster of follower F_i (let's designate it as C_i) is the set of the explorers which are closest to F_i than to any of the other followers. So a particular explorer E_j is in C_i if the distance d (E_j, F_i) is minimal over the set of followers F_m, m=1.. k.

The main challenges of the operation are as follows. First, it is desired that the clustering C_m, m=1.. k be determined dynamically. As the explorers move about their business, the overall area which needs to be covered changes and the distribution densities of explorers also change. In other words, if the distribution of explorers changes substantially, a clustering that worked before may no longer be appropriate and a new clustering may be needed. If there are k followers, then as the distribution of the explorers changes, there is a need to dynamically re-compute a new clustering of the explorers in exactly k new clusters. Second, each follower should position itself so that its distance from each one of the explorers which it serves is as short as possible. An appropriate metric to capture this requirement within a particular cluster C_i is to minimize the square distance sum

$\sum_j \boldsymbol{d}(\boldsymbol{E}_j, \boldsymbol{F}_i)$ where $E_j \in C_i$ ....(1)

Another suitable optimization measure can be considered which may account not only for the actual positions of the explorer vessels but also for their respective frequency of communication activity (not the frequencies used for the channel waves). That option however, will be explored in future work and it is not taken up here.

Further, as the explorers move freely, the corresponding follower should be moving too so as to keep itself close to the position which optimizes its composite distance (Eq.1) from the served explorers. A scenario which complicates matters is when one of the explorers (say, E_X), which is served by a follower (say, F_A), while moving about its business, it enters a region served by another one of the followers (let's call this one F_B). It would make sense then that F_A should stop chasing E_X and turn to the rest of the vessels it serves. Essentially, E_X would effectively be passed to F_B's flock and now be part of F_B's service cluster. As a result of this, both F_A and F_B may need to move appropriately in order to minimize metric (1) within their new respective flocks (service clusters). So the followers essentially divide the space into territories of service and thereby the collection of explorers into service groups with each of the followers in charge of a geographic territory and thereby the group of explorers in that territory. This division of space (and responsibilities) cannot be static; it has to be dynamic; as the explorers move freely the division of the space into areas of responsibility should also change dynamically as needed.

So here lies the specific challenge: Given a distribution of the set of explorers, determine the specific topological distribution of the set of followers which minimizes the total sum of expressions (1) for all the followers. This has to be achieved continually by the followers while the explorers move. This is not easy to compute using some sort of gradient descent method. An efficient method is needed to determine the movements of the followers that is not computationally expensive. However, another view of the problem from the perspective of the distributions of the vessels, suggests that what is needed is a method by which the distribution of a set of followers approximates the distribution density of the larger number of explorers. In other words, we need the followers to map the distribution of the explorers and do so dynamically.

This is reminiscent of the Self Organizing Maps (SOMs, by Kohonen [4-6] and we took up the application of the method appropriately modified for the purposes of this setting. There are other methods drawn from problems of similar nature such as sensors deployment [8,9]. but we found that SOMs perform very well in spite of their elegant simplicity. In the following we review the method and provide the result of our effort to adapt it to this problem.

The principle of self-organizing maps

In a typical scenario in the operation of self organizing maps (SOM), we have an arbitrary distribution of a collection of n vectors (static samples) and we wish to approximate their distribution density using a small set of m markers, i.e. the distribution density of the m markers should be like the distribution density of the n vectors. This is achieved by an iterative process which starts with a uniform or random initial distribution of the m markers and then iterating over each sample vector and moving the marker which is the closest to it by a small step towards the sample as illustrated below.

$M_{new} = M + \alpha(S-M)$ --------- (2)

where $0< \alpha <1$ until equilibrium.

Adapting soms for the current problem

To adapt the original algorithm for the purposes of this application, the motion computed by the original algorithm must be projected on the surface plane since the followers are constrained to only move on it. Further, we assume that the followers can sense the relative position of the explorers in its geographical territory; the direction of the signal can be reasonably sensed (or the source triangulated) and its strength provides some measure of distance). So in this case the algorithm is modified as follows

- **Initialization:** Obtain the projections of the explorers on the surface horizontal plane. Within the (horizontal) area which envelopes these projections, the followers are initially placed on a uniform grid.
- **Iteration:** The followers "compete" for "territory" or "service area" and consequently for a collection of explorers to serve. It is due to this competition that they continuously relocate in small steps each time until the competition reaches some state of equilibrium. Thus, an iteration is performed where the iterable is the set of explorers and the action is a relocation of followers by a small step at a time. The iteration continues until no significant change is made in the positioning of the followers. If the positions of the explorers were static then the iteration would eventually come to an end but if it is not (and usually it would not be static) then the iteration will be continuing as long as explorers move in order to properly reposition the followers. For each explorer E, determine which one of the followers is closest. This follower (let's call it F) is the "winner" of the current competition for service territory and will make a small movement towards the explorer E. This motion will be a small portion of the vector starting at the current F and ending at E, and projected on the horizontal plane. Assuming a coordinate system (x, y, z) with coordinates (x, y) on the horizontal plane and z the vertical one, let F = (F_x, F_y, F_z) and E=(E_x, E_y, E_z). Then according to the original SOM algorithm the movement of F should be ΔF=α(E-F) but in our case it must be projected to the horizontal plane (because E is the position of a surface vessel). Now let U be the unit vector on the horizontal plane (U = (U_x, U_y,0) where U_x and U_y are the x and y axes unit components respectively). Let us denote the projection of a vector A in the direction of another vector B as A‖B, which means that A‖B is a vector in the direction of B and its measure is (A.B)/|B| where A.B is the scalar product of A and B. Then the actual motion of F is ΔF = α (E-F) ‖U (projection of the vector difference E-F on U) where 0 < α < 1. Or, equivalently, ΔF = α(E‖U-F‖U) which means that this operation can be applied on the projections of all the vessels on a horizontal plane. If the origin of the coordinate system is placed on the water surface then Fz = 0 for all the followers, so F‖U = F, and the projection of E on the horizontal plane is obviously P = (Ex, Ey, 0) where Ez = 0. Then the motion of the winner F is ΔF = α (P-F). In short, the motion computation can be made on the projections on a horizontal plane to keep it simple.

This algorithm will run continuously to adjust the positions of each follower as the underwater ones move around. This method has been applied successfully in problems where a static density distribution needs to be approximated by a finite set of markers [7]. In the problem at hand, the density distribution to be approximated (with markers representing the finite set of followers) is that of the explorers and it is not static. We contend however, that the changes due to the movement of the explorers are not very rapid and it is possible to run a vast number of iterations of the algorithm within a short time window in which the distribution can be considered static. As the distribution of the explorers changes, the algorithm is expected to place the followers (markers) correctly and force changes of their positions which will correctly track the explorers. In short, the algorithm can settle quickly to a new distribution of the followers after a change in the positions of the explorers and the followers are able to assume such positions given that they are much faster than the explorers.

With the above simple SOM algorithm there are some issues with some markers being greedy depending on the initial placement but they have been studied and good solutions exist. For example in a variation of the algorithm, not only the winner vessel makes a move, but its immediate neighbors do too albeit with a much smaller step (much smaller α parameter). In the version used here, all the followers move according to the formula which modifies Eq. (2) as: $M_{new} = M + \alpha(d_{SM})(S-M)$ where the parameter α is a function of the distance from the "winner" vessel. So now in more detail, for every explorer E considered (each in turn and iteratively), let the closest follower be denoted as W (this is the "winner" of the competition for service territory). For each one of the (other) followers designated as F, a similar movement is computed according to the formula:

$$F = F \quad + \alpha\,(d\,)\,(F - E) \parallel U \tag{3}$$

or equivalently:

$$F = F + \alpha(d\,)\,(F \quad \parallel U - E \parallel U) \tag{4}$$

where now α is a scalar function of the distance d_{FW} of F from W, and in particular it is a sigmoid function which tapers off as the distance from the winner increases

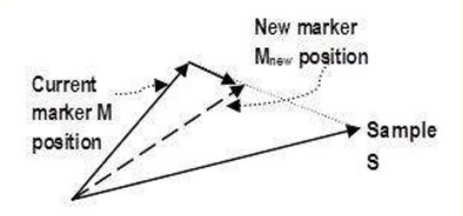

Figure 1: Marker movement towards the sample.

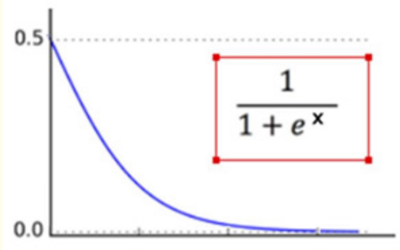

Figure 2: Decay function to adjust the followers' movement step.

So α () is defined as:

$\alpha(d_{FW}) = \beta/(1+e^{-d_{FW}})$ ------(5)

β = $0.99^{(i-1)}$------(6)

β is a correction factor that is set to diminish gradually to stabilize the search process as it converges (i.e. followers begin to establish their territories), and i is the iteration number.

It should also be noted that Eq. (4) means that all the computation of the movement of the followers, only needs to be performed on the horizontal plane of the water surface using the projections of the explorers on it.

Simulation Results

We performed simulations and obtained traces of the motion of the followers. For a given random and static topology of the explorers simulations were run with various starting topologies of the followers. The value of the following metric was computed and plotted during the simulation

$M = \sum\sum \boldsymbol{d}(\boldsymbol{E}_j, \boldsymbol{F}_i)$ where $E_j \in C_i$, and i are the follower indexes---(7)

It should be noted that each distance d in Eq. (7) has a depth component (vertical projection) and a horizontal projection component. Since the motion of followers is constrained to the surface, the depth component is not affected by the follower motions, so Eq. (7) is equivalently optimized if we use the horizontal projections in it.

While the optimum value of Eq. (7) over all possible topologies is not known, the simulations show a consistent decrease of the metric over the simulation runs. The tabulated results as well as some typical plots and screenshots of randomized starting points for the same sample data set are provided below. In this example, 40 data points (that represent submerged vessels) and 5 surface vessels (that represent the followers) were randomly placed on a grid that is 200 x 200 as shown in the figure. The followers were initially placed randomly at different locations then the competition algorithm was deployed. The initial placement of the follower vessels was chosen at four different locations to demonstrate the effectiveness of the algorithm. Figure 3 represents the case when the followers were placed randomly on the grid, Figure 4-6 represent the cases with the followers starting as a cluster in one corner of the grid. The blue asterisks represent the initial cluster of followers and the red circles represent their final destinations.

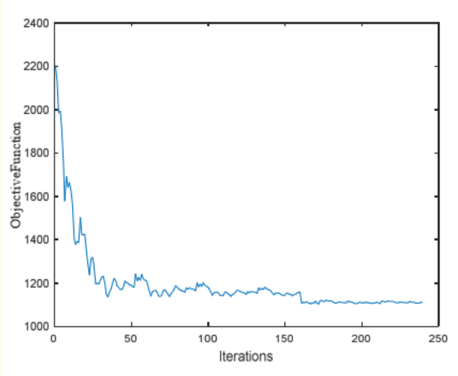

Figure 3.1: Minimization of the objective function.

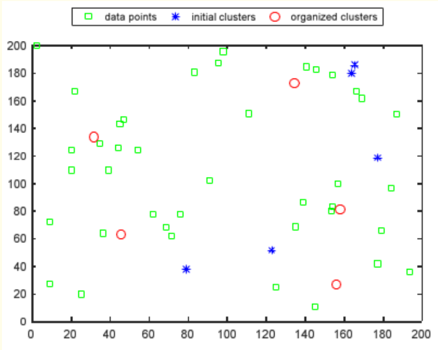

Figure 3.2: Resulting distribution map.

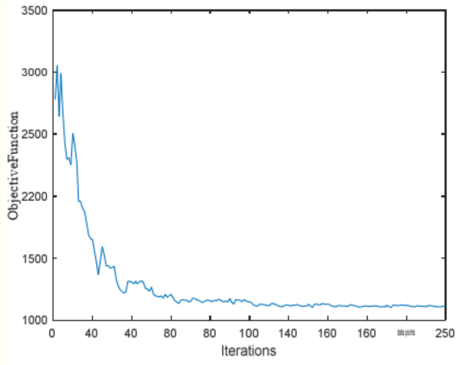

Figure 4.1: Minimization of the objective function.

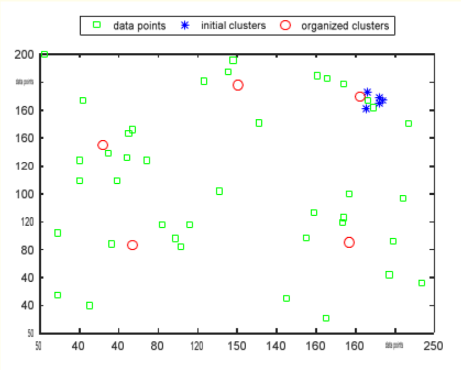

Figure 4.2: Resulting distribution map.

Case 1: random distribution on the grid (M = 1109)

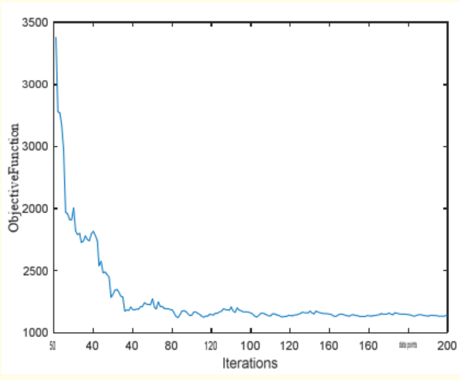

Figure 5.1: Minimization of the objective function.

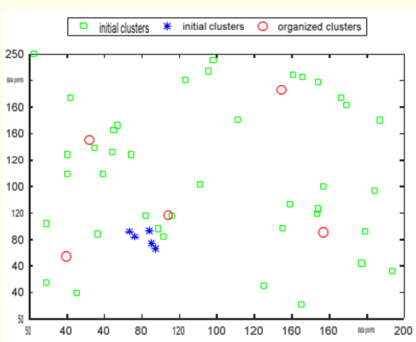

Figure 5.2: Resulting distribution map.

Case 2: clustered at the top-right corner (M = 1111.3)

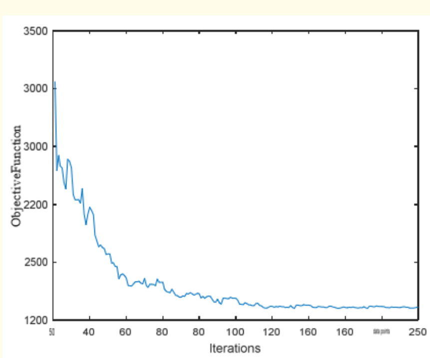

Figure 6.1: Minimization of the objective function.

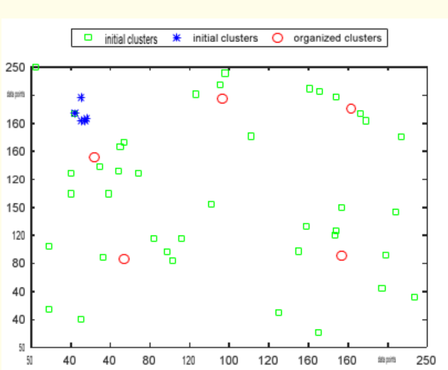

Figure 6.2: Resulting distribution map.

Case 3: clustered at the bottom-left corner (M = 1137. 1)

Case 4: clustered at the top-left corner (M = 1109.2)

In all the sample cases demonstrated in Figures 3-6, the algorithm was successful at identifying self-organizing maps while consistently decreasing the value of the metric (7).

It is noteworthy to observe that the final destination of the followers was only slightly different each time it converged into final destinations. Each distribution offers one solution for a multi-dimensional network. SOMs are not closed form solution methods; rather, they reach the solution after successive iterations that build on one another. Different initial placements of the followers will affect the projections of these followers as explained in Eq. 4. Movement of the followers is achieved in an incremental and sequential manner. Thus, some of the followers can become territorial early on during the iterative process and stabilize in unique locations. This early convergence shifts the mobility to the remaining followers to establish a stable and balanced distribution of followers. This may explain the slight variations in the followers' final destinations.

Conclusions

This work deals with the problem of how surface boats may follow submerged exploration vessels so as to best facilitate communications. Self-Organizing Maps are used to provide an elegant and very efficient solution to this problem. This problem boils down to mapping a density distribution with a finite set of markers. This is a better approach to what otherwise can be accomplished with the k-means method [10], but without the computational troubles associated with k-means, determination of centroids etc. The simulations show that the SOMs method works well and minimizes an overall distance metric which reflects the negative effect of distance on underwater communications.

The simulated examples show how the clusters were randomly placed and clustered at the bottom corner of the grid, at the right top, and the left top of the grid. In each one of these cases, the final

Bibliography

1. "Editorial Underwater Acoustic Communications: Where We Stand and What Is Next?" *IEEE Journal of Oceanic Engineering* 44.1(2019): 1-6.
2. N Tang., *et al*. "Research on development and application of underwater acoustic communication system". *Journal of Physics Conference Series* 1617(2020): 012036.
3. M Jahanbakht., *et al*. "Internet of underwater things and big marine data analytics -- a comprehensive survey". *arXiv* (2012): 06712.
4. T Kohonen. "Self-organized formation of topologically correct feature maps". *Biological Cybernetics* 43(1982): 59-69.

5. T Kohonen. "The self-organizing map". *Proceedings IEEE* 78(1990): 1464-1480.
6. F Murtagh and M Hernández-Pajares. "The Kohonen self-organizing map method: an assessment". *Journal of Classification* 12(1995): 165–190.
7. C Koutsougeras., *et al*. "Event-driven sensor deployment using selforganizing maps". *International Journal of Sensor Network* 3.3 (2008): 142-151.
8. G Wang., *et al*. "Sensor relocation in mobile networks". In Proc. IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., Miami, FL, USA 4(2005): 2302-2312.
9. G Wang., *et al*. "Movement-assisted sensor deployment". *IEEE Transactions on Mobile Computing* 5.6(2006): 640-652.
10. Arthur D and Vassilvitskii S. "k-means++: the advantages of careful seeding". In Proc. of The Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA (2007): 1027-1035.