OCEAN 94 OSATES - Brest - septembre 1994

Neural Meshing of a Geographical Space in Regard to Oceanographic Data Location

O. Sarzeaud *
CETIIS, BP 33000
F-13791 Aix-en-Provence CEDEX

Y. Stéphan, F. Le Corre and L. Kerléguer
Centre Militaire d'Océanographie,
EPSHOM, BP 426
F-29275 Brest CEDEX

Abstract - **This paper presents a method to mesh a geographical area in regard to oceanographic data locations. The meshing is achieved using Kohonen's networks. The idea is to adapt the position of the nodes (neurons) by successively considering the location of each data and moving the closest node and its neighbours toward this location. We have successfully tested this method on an hydrological database of the North-East Atlantic Ocean. We have constrained the meshing in regard to some practical considerations (minimal size of the meshes, discrete positionning of the nodes, ability to create, remove or move nodes during the adaptation, ...). Kohonen's networks appear as an efficient and a robust meshing tool.**

Résumé - **Cet article présente une méthode de maillage d'un espace géographique en tenant compte de la position des données océanographiques. Le maillage est réalisé par réseaux de Kohonen. L'idée est d'adapter la position des noeuds (neurones) en présentant consécutivement la position de chaque donnée et en rapprochant le neurone le plus proche ainsi que ses voisins. Nous avons appliqué cette méthode avec succès en utilisant une base de données en Atlantique Nord-Est. Certaines contraintes ont été ajoutées pour prendre en compte différents aspects pratiques (taille minimale de mailles, adaptation discrète des positions, possibilité de création ou suppression de noeuds pendant l'adaptation, ...). Les réseaux de Kohonen se révèlent être un outil efficace et robuste de maillage.**

1. INTRODUCTION

Data processing is an essential task in Ocean Engineering and Oceanography. To infer and to study an oceanic area, it is important to have a great number of observations (mainly temperature and salinity measurements). The problem stems from the fact that in most cases, these observations are not issued from a regular sampling in space and time. Therefore, it is important to take into account the irregular distribution of the data in order to optimize further processing (temporal or spatial interpolation, statistical analysis, typology definition, data inversion,...).

Mapping the distribution of the data is the problem addressed in this paper. In fact, it is posed in terms of meshing a geographical space in regard to the data location (latitude, longitude, depth and date). We show that this problem can be solved by Kohonen Maps or Volumes. The basic principle of Kohonen's networks [1] is a mapping of an input space (data space) into an internal representation (state space) with the conservation of topological constraints. The networks are constituted of nodes (neurons) which are linked by the definition of a neighbourhood for each of them. The idea is to adapt the position of the nodes by successively considering the location of each data and moving the closest node and its neighbours toward this location. After several presentations of the whole database, the grid is expected to be denser where data are more numerous and looser where not.

The aim of this paper is twofold. First, we show that the basic algorithms of Kohonen's networks can be modified without alteration of the properties. In particular, many constraints can be imposed. Second, we propose a new tool for meshing.

Section 2 describes the general aspect of the problem. Section 3 gives some backgrounds about Kohonen's networks and their properties and describes the meshing algorithm. Some results of neural meshes are given and discussed in section 4.

2. THE PROBLEM

Centre Militaire d'Océanographie gathers, processes and files oceanographic data for the French Navy. The origin of these data is various. Some are issued from «routine» measurements, others are issued from intensive goal-oriented cruises in a given area. As a consequence, the data are unequally distributed in both time and space. Two main goals may be achieved with oceanographic data.

2.1 Typology definition and classification

An efficient way to have a synoptic view and knowledge of the environment in a oceanic area is to determine a series of plausible profiles presenting different types of hydrological situations. The general purpose is to give a representative idea of what temperature (or sound speed) environment is expected to be at a given time in a given place. The types are

* now at ECTIA, 1 rue de la Noë, BP 92119, 44321 Nantes cedex 3

computed using available historical data and classification techniques [2].

2.2 Climatology

Another interesting application is to perform a so-called climatology on a database. This requires a statistical smoothing and interpolation of the data set.

In both cases, it is important to build statistically relevant subsets regarding two criteria. First, the data of a subset must be close in respect to their geographical locations and dates (depending on the spatial and temporal correlation of the oceanographic phenomena). Second, the cardinal number of each subset has to be roughly the same (in order to have the same relevance). This problem can be solved by an adaptive meshing method.

3. NEURAL MESHING

In this section, we present a method to produce meshes adapted to the density of some data. The method relies upon the behaviour of Kohonen's networks. We first give a description of Kohonen's networks and present their construction. We then propose a modified algorithm for meshing problems.

3.1 Kohonen's networks

Kohonen's networks are artificial neural networks. As such, they are completely defined by:

- their components,
- their structure,
- the behaviour of their components.

3.1.1 Components

Kohonen's networks are made of two types of components:

- the neurons (or cells, or units),
- a function of decision of the output.

In order to discuss about Kohonen's networks, we need to define some descriptive variables:

- m: number of inputs,
- n: number of neurons and outputs,
- x_i: input number i,
- y_j: output number j,
- w_{ij}: weights between input i and neuron j.

3.1.2 Structure

The structure of Kohonen's networks is shown on figure 1. Each input is sent to all the neurons. Each neuron sends its output to the decision function which computes the output of the network. The network has as many outputs as neurons.

We emphasize on the distinction between the model and

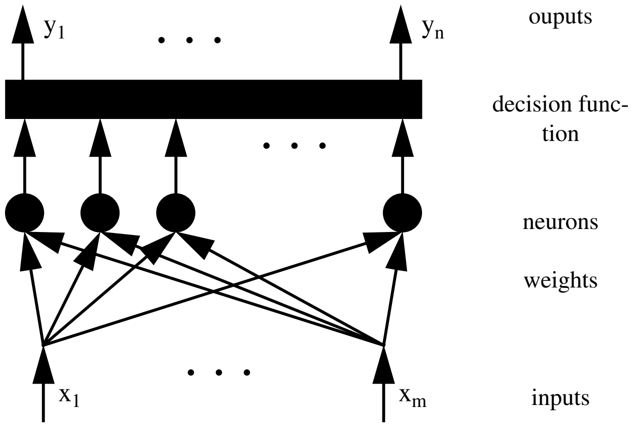

Fig. 1: Structure of Kohonen's networks.

the learning rules. As shown on figure 1, we do not include connections between neurons in the model. This differs from the original definition of Kohonen's network. As we will see in 3.2.1, the connections are introduced as part of the learning scheme [4].

3.1.3 Behaviour of the components

Each neuron j in the network computes the distance between the input vector and its weight vector:

$$d_j = \sum_i (w_{ij} - x_i)^2 \tag{1}$$

The function of decision of the output then computes the output of the network;

$$y_j = \begin{cases} 1 & \text{if} \quad d_j = \min\{d_k\} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

It is clear that the network delivers the address of the neuron which weight vector is the closest to the input vector. This neuron is called the cluster.

3.2 Construction of Kohonen's networks

3.2.1 Basic algorithm

Kohonen has associated with his network a rule that finds some values of weights, with certain properties. Kohonen's rule supposes that a neighbourhood between neurons is defined: $\aleph(j)$ is the set of neurons that are called neighbours of j. According to this neighbourhood, the network may be called a line, a map or a volume (figure 2). The aim of the rule is to reinforce the instant behaviour of the network, that is: make sure that the cluster at time t will still be the same if the same input is presented at time t+1.

With this aim, the following error function is defined:

$$E = \frac{1}{2}\sum_j \left[\left(\sum_i (w_{ij} - x_i)^2 \right) \left(\lambda y_j + \mu \sum_{k \in \aleph(j)} y_k \right) \right] \tag{3}$$

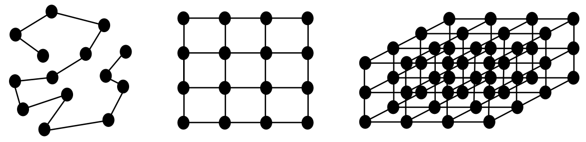

Fig . 2: Relation between the neighbourhood and the dimension of the network : (a) is a line, (b) is a map, (c) is a volume.

where μ and λ are positive real values such as $\mu \le \lambda$.

The contribution of each weight to the error function is:

$$\frac{\partial E}{\partial w_{ij}} = (w_{ij} - x_i)\left(\lambda y_j + \mu \sum_{k \in \aleph(j)} y_k\right) \qquad (4)$$

Therefore, the modification of each weight must be done as follows:

$$\Delta w_{ij} = \begin{cases} -\alpha(w_{ij} - x_i) & \text{if j is the cluster} \\ -\beta(w_{ij} - x_i) & \text{if j is a cluster neighbour} \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

where α and β are proportional to λ and μ.

Practically, the algorithm to determine the values of the weights works as follows:

1/ Initialize the weights to some small random values
2/ While α and β are non zero:
2.1/ Choose an input vector.
2.2/ Compute the output of the network.
2.3/ Modify the weights according to (5).
Decrease α and β.

This algorithm computes the values of the weights so that the network realizes a partitioning of the input space that respects the topological constraints imposed by the neighbourhood. That is: each neuron corresponds to a portion of the input space, and neurons which are neighbours correspond to adjacent portions of the input space.

One important property of the Kohonen's networks is their ability to self-organize with conservation of the density of the input. Without any modification of the algorithm, we hence can expect them to produce a mesh adapted to the density of the data location, which is a major motivation of this work.

3.2.2 Customizing the basic algorithm

There are five possibilities to modify the above algorithm:

- **Pre-processing** : the initialization of the weights and the method to choose the inputs can be modified;
- **Network behaviour:** additional terms can be used in (1) in order to increase or decrease some distances with respect to some criteria;
- **Adaptation rule**: additional terms can be used in the cost function (3), which results in modification of the adaptation rule (5);
- **Constraints**: additional rules corresponding to some constraints can complete (5);
- **Current and post-processing** : during or after a global construction of the network, the weights and the network structure can be analysed and if necessary modified.

3.3 Using Kohonen's networks and algorithm for meshing

The idea of applying Kohonen's networks and algorithm to meshing problems was first proposed for Electrical Engineering in [3]. We propose an extended approach in Oceanography and Ocean Engineering.

3.3.1 Basic ideas

The idea of using Kohonen's networks for meshing purpose appears quite natural. The neurons, plotted in the input space using their weights as coordinates, represent the nodes of the meshing, while the neighbourhood between the neurons defines the meshes. This implies that the dimension of the neighbourhood be the same as the input space one. To mesh a 2D area (resp. a 3D area), the neighbourhood must define a map (resp. a volume). Therefore, the meshing is achieved by adapting the position of some pre-defined nodes, and not by creating such nodes.

To strictly adhere to the meshing definition, Kohonen's networks have to achieve a total partitioning of the area. This is done by constraining the peripheral nodes of the network to be located on the boundary of the area to mesh. Another point concerns the overlapping of meshes. There is no proof of no overlapping of some meshes (due to inversions in the topological order of the neurons). This effect can be taken into account by constraining the meshes to a minimal size (it automatically prevents from nodes inversions).

3.3.2 Modifications of the basic algorithm

Pre-processing. The initialization of the weights is not randomly done. The weights are computed so that the original meshing is a regular grid, the peripheral nodes being located on the boundary.

Constraints. Some constraints complete the adaptation rule (5). First, peripheral nodes can only move along the boundary on which they were initialized. Second, all nodes have discrete movements, i.e. they can only move with respect to a certain resolution. Third, a minimal size of mesh is imposed, which prevents from degenerated meshes. Finally, colinearity (in two dimensions) or coplanarity (in three dimensions) may be imposed to nodes positions.

Current and post-processing. Some treatments can be proposed to refine the mesh. First, a regularization algorithm can be used to squeeze broken lines in the mesh. Second, manual modifications can be done: movements, suppression, or additions of nodes. All this processing can be done during the meshing phase or after the mesh is achieved.

3.4 Meshing algorithm

The basic algorithm mixed with the modifications presented above result in the meshing algorithm.

1/ Initialization of the weights. The peripheral nodes position and resolution constraints must already be taken into account.

2/ Random choice of one among all inputs; computation of the output of the network with the help of (1) and (2).

3/ For each concerned neuron, modification of the weights following (5) plus the rules expressing the constraints.

4/ current processing if necessary.

5/ if the adaptation gains are non zero, return to step 2; else end of the adaptation.

6/ post-processing if necessary.

4. RESULTS

4.1 Examples of neural meshes

To test the neural meshing method, we have considered a database of oceanographic data in the North-East Atlantic Ocean. The data set is composed by approximately 7,000 profiles from which we have extracted the geographical location as well as the depth of each measurement. Our goal is to achieve a meshing of the global area in regard of the data locations applying the algorithm described previously.

Figure 3a presents the geographical distribution of the measurements. Figure 3b gives the result of a neural mesh with a 26x26 map (625 meshes) with elementary constraints (minimal size of meshes of 5’, resolution of 5’). The number of iterations is 50,000 (corresponding to about 7 presentations of the whole database).

Figure 4 shows a three-dimensional mesh with a 6x6x6

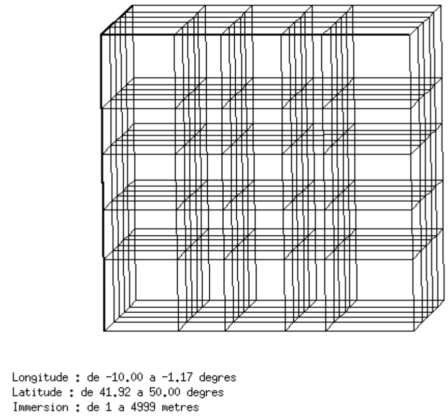

Fig. 4: Three-dimensional mesh with coplanarity constraints.

Kohonen volume (125 meshes). The inputs are part of the database above the abyssal plain. Coplanarity constraints were added to enforce the meshes to be parallelepipeds.

4.2 Discussion

The quality of the meshes is good. As expected, nodes are denser where data are more numerous. Note for example the size of the meshes in the South of the area as well as in the North-West (figure 3a and 3b) where no data are available (earth and continental shelf).

Because of the small number of nodes, the three-dimensional mesh shown in figure 4 is less spectacular in terms of the fitting of the nodes distribution to the data distribution. Nevertheless, the uniform vertical distribution of the measurements is well represented by the mesh.

Many numerical experiments have been achieved to check the efficiency of the algorithm and the robustness of Kohonen’s networks to alterations and constraints. In particular, we have added or removed nodes during adaptation.

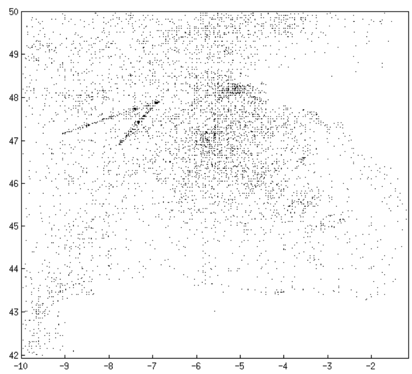

Fig. 3a: Distribution of the measurements.

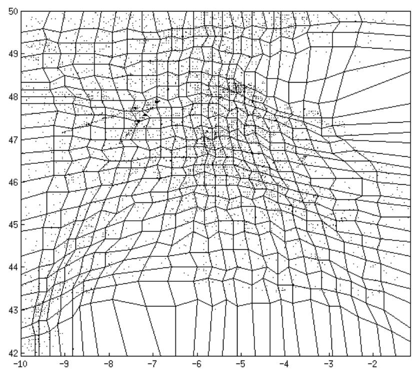

Fig. 3b: Neural meshing of the geographical space (625 meshes).

We have fusionned adjacent meshes. In all cases, the network behaviour was not altered. An interesting point concerns also the basic form of the maps. As shown in figure A.1, it is possible to define maps or volume of any shape (e.g. triangle). Hence, the network can be designed to fit one given meshing problem in an optimal way.

The major drawback of the neural meshing may be that the computational time is quite heavy (the meshing in figure 3b requires about 3 minutes 45 seconds on a SparcStation 20).

The neural algorithm as described previously can be used in many scientific domains. Let us quote finite-element meshing (for which it was originally proposed), bathymetry modelling, data interpolation in case of non-uniform distributions, box-inverse methods (typically relying on meshes shown in figure 4),... Neural meshing may also be a useful tool in cruise design (to optimize the strategy of measurements).

4.5 Future developments

We are now trying to introduce new criteria to guide the meshing evolution. These criteria concern the spatial and inter-annual variability of oceanographic data. The first one will be treated during the pre-treatment phase, by modifying the way of choosing a data. The second one will be treated by transforming the network behaviour: an additional term will be added in the distance computation in (1).

5. CONCLUSION

We have presented in this paper a robust algorithm for meshing problems. It relies on the behaviour of Kohonen networks. It is shown that the basic form of Kohonen networks can be widely modified without any loss of properties so that they can afford valuable techniques in data processing. The example of the meshing of a geographical space is presented. Neural meshing has turned out to be an efficient algorithm which has been included in a general oceanographic data processing scheme.

REFERENCES

[1] Kohonen, T. (1989) *Self-Organization and Associative Memory*, 3rd ed., Springer-Verlag.

[2] Stéphan, Y. and Frachon, B. (1993) «A Modular Neural Architecture for Bathythermograms Classification», *Proc. of OCEANS'93*, Vol 1, pp I.433-I.437.

[3] Sarzeaud, O., Stéphan, Y. and Touzet, C. (1991) «Finite-element meshing using Kohonen's self-organizing maps», *Proc. of ICANN'91*, Vol 2, pp 1313-1317.

[4] Sarzeaud, O. (1993) «Contribution à l'étude et au développement de systèmes connectionnistes modulaires», thèse de l'université de Montpellier II.

APPENDIX

The goal of this appendix is to give some illustrative examples of self-organized feature maps. Each point represents one couple input. Each neuron is represented in the input space using its two weights. The location of a neuron is then given by a bold plot. All inputs have been normalized.

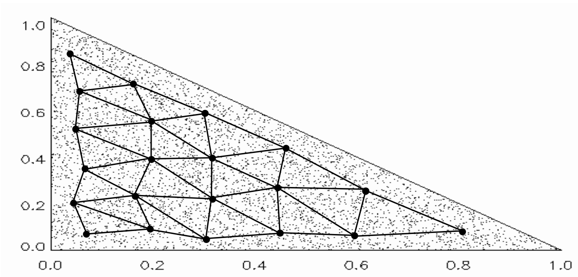

Fig. A1: triangular maps self-organized in a triangular space. The original form has been truncated to be triangular with a triangular neighbourhood. It is remarkable that the basic properties of the maps are not altered under such constraints.

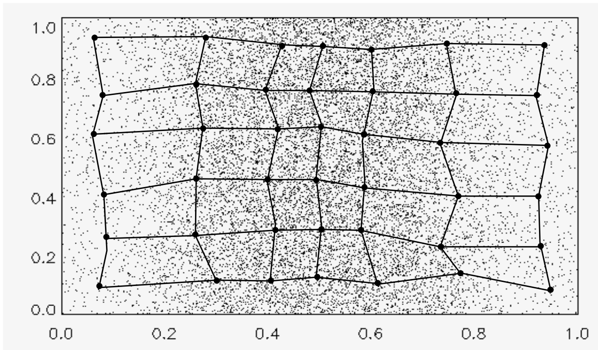

Fig. A2: example of self-organization in a X-normal distribution. Inputs are regularly distributed on the Y-axis and normally distributed on the X-axis around 0.5. Note that the neural density is higher in the middle of the space.

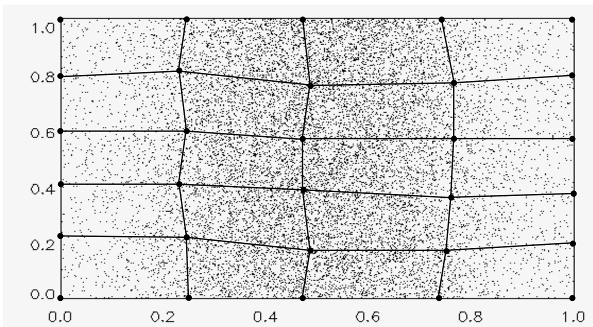

Fig. A3: self-organization with boundary constraints. It is a simple example of neural mesh.