Unsupervised segmentation using a self-organizing map and a noise model estimation in sonar imagery

K.C. Yao, M. Mignotte, Christophe Collet, P. Galerne, Gilles Burel

Pattern Recognition, Vol 33, No 9, pp. 1575-1584, April 2000

UNSUPERVISED SEGMENTATION USING A SELF-ORGANIZING MAP AND A NOISE MODEL ESTIMATION IN SONAR IMAGERY

K.C. Yao[‡], M. Mignotte[‡], C. Collet[‡], P. Galerne[‡], G. Burel[•]

[‡]Laboratoire GTS (Groupe de Traitement du Signal), Ecole Navale
BP 600 - 29 240 Brest France

[•]Université de Bretagne Occidentale, LEST-UMR CNRS 6616
6 av. Le Gorgeu, BP 809 - 29 285 Brest France

April 28, 1999

Abstract

This work deals with unsupervised sonar image segmentation. We present a new estimation and segmentation procedure on images provided by a high resolution sonar. The sonar image is segmented into two kinds of regions : *shadow* (corresponding to a lack of acoustic reverberation behind each object lying on the seabed) and *reverberation* (due to the reflection of acoustic wave on the seabed and on the objects). The unsupervised contextual method we propose is defined as a two-step process. Firstly, the Iterative Conditional Estimation (**ICE**) is used for the estimation step in order to estimate the noise model parameters and to accurately obtain the proportion of each class in the Maximum Likelihood (**ML**) sense. Then, the learning of a Kohonen Self-Organizing Map (**SOM**) is performed directly on the input image to approximate the discriminating functions *i.e.* the contextual distribution function of the grey levels. Secondly, the previously estimated proportion , the contextual information and the Kohonen SOM, after learning, are then used in the *segmentation step* in order to classify each pixel on the input image. This technique has been successfully applied to real sonar images, and is compatible with an automatic processing of massive amounts of data.

Key Words : *Kohonen Self-Organizing Map, segmentation, parameter estimation, sonar imagery, Markov Random Field.*

1 INTRODUCTION

In high resolution sonar imagery[1], three kinds of regions can be visualized : *echo, shadow and sea-bottom reverberation*. The *echo* information is caused by the reflection of the acoustic wave from the object while the *shadow* zone corresponds to a lack of acoustic reverberation behind this object. The remaining information is called the *sea-bottom* reverberation area. On the pictures provided by a classification sonar, the echo features are generally less discriminant than the shadow shape for the classification sonar of object lying on the seafloor. For this reason, detection and classification of an object located on the seafloor (as wrecks, rocks, man-made objects, and so on...) are generally based on the extraction and the identification of its associated cast shadow [5]. Thus, before any classification step, one must segment the sonar image between *shadow* areas and *reverberation* areas. In fact, the *sea-bottom reverberation* and the *echo* are considered as a single class.

Unfortunately, sonar images contain speckle noise [7] which affects any simple segmentation scheme such as a **ML** segmentation. In this simple case, each pixel is classified only from its associated grey level intensity.

In order to face speckle noise and to obtain an accurate segmentation map, a solution consists in taking into account the contextual information, *i.e.* class of the neighborhood pixels. This can be done using Markov Random Field (**MRF**) models [2] and this is why Markovian assumption has been proposed in sonar imagery [4]. In this global bayesian method, pixels are classified using the whole information contained

[1]The authors thank GESMA (Groupe d'Etude Sous Marine de l'Atlantique, Brest France) for having provided real sonar images and REGION BRETAGNE for partial financial support of this work.

in the observed image simultaneously. Nevertheless, simple spatial **MRF** model have a limited ability to describe properties on large scale, and may be not sufficient to ensure the regularization process of the set of labels when the sonar image contains high speckle noise. Such a model can be improved by using a larger spatial neighborhood for each pixel [16], or a causal scale and spatial neighborhood [10] but this rapidily increases the complexity of the segmentation algorithms and the parameter estimation procedure required to make this segmentation *unsupervised*. Besides, the segmentation and the estimation procedure with such *a priori* model requires a lot of computing time. Moreover, the use of such a global method does not allow to take into account the noise correlation on the sonar image [14].

An alternate approach adopted here, uses a local method, *i.e.* takes into account the grey levels of the neighborhood pixels. In this scheme, each pixel is classified from information contained in its neighborhood. This method allowing to take into account the noise correlation is divided in two main steps : the model parameter estimation [9] and the segmentation algorithm which is fed with the previously estimated parameters.

In this paper, we adopt for the parameter *estimation step* an iterative method called Iterative Conditional Estimation (**ICE**) [13] in order to estimate, in the ML sense, the noise model parameters and specially the proportion of each class (shadow and reverberation). Followed by the training of a competitive neural network as a Kohonen **SOM** [8] in order to approximate the discriminating function (*i.e.* the contextual distribution function of the grey level). For the segmentation step, we develop a contextual segmentation algorithm exploiting efficiently the previously estimated parameters, the input sonar image, and the topology of the resulting kohonen **SOM**.

This paper is organized as follows. In section 2, we detail the parameter estimation step based on the ICE procedure in section 2.1, and the training step of the SOM in subsection 2.2. Section 3 presents the segmentation step. Experimental results both on real scenes and synthetic sonar images are presented in subsection 3.3, where we compare the results obtained with the proposed scheme, a ML segmentation and a classical monoscale Markovian segmentation. Then a conclusion is drawn in section 4.

2 ESTIMATION STEP

2.1 Iterative Conditional Estimation

2.1.1 Introduction

We consider a couple of random fields $Z = (X, Y)$ with $Y = \{Y_s, s \in S\}$ the field of observations located on a lattice S of N sites s, and $X = \{X_s, s \in S\}$ the label field. Each Y_s takes its value in $\Lambda_{obs} = \{0, ..., 255\}$ and each X_s in $\{e_0 = shadow, e_1 = reverberation\}$. The distribution of (X, Y) is defined firstly by $P_X(x)$, the distribution of X assumed to be stationary and Gibbsian (*i.e.* markovian) in this estimation step, and secondly by the site-wise likelihoods $P_{Y_s/X_s}(y_s/x_s)$. In this work, these likelihoods depend on the class label x_s. The observation Y is called the *incomplete data* whereas Z stands for the *complete data*.

In this step, we estimate the parameter vector Φ_y which defines $P_{Y/X}(y/x)$ by using the iterative method of estimation called Iterative Conditional Estimation (ICE) [13]. This method requires to find an estimator, namely $\Phi_y(X, Y)$ for completely observed data. When X is unobservable, the iterative ICE procedure defines $\Phi_y^{[k+1]}$ as conditional expectations of $\overset{\wedge}{\Phi}_y$ given $Y = y$, computed according to the current value $\Phi_y^{[k]}$. This is the best approximation of Φ_y in terms of the mean square error [13]. By denoting E_k, the conditional expectation using $\Phi_y^{[k]}$, this iterative procedure is defined as follows :

- Initialize the noise model parameters to $\Phi_y^{[0]}$.
- $\Phi_y^{[k+1]}$ is computed from $\Phi_y^{[k]}$ and $Y = y$ by :

$$\Phi_y^{[k+1]} = E_k[\overset{\wedge}{\Phi}_y \ / Y = y] \tag{1}$$

The computation of this expectation is impossible in practice, but we can approach equation (1) thanks to the law of large numbers by :

$$\Phi_y^{[k+1]} = \frac{1}{n}[\hat{\Phi}_y\,(x_{(1)}, y) + ... + \hat{\Phi}_y\,(x_{(n)}, y)] \tag{2}$$

where $x_{(i)}$, with $i = 1, ..., n$ are realizations of X according to the posterior distribution $P_{X/Y,\Phi_y(x/y,\Phi_y^{[k]})}$. Finally, we can use the **ICE** procedure for our application because we get :

- An estimator $\hat{\Phi}_y\,(X, Y)$ of the complete data : we use a Maximum Likelihood (ML) estimator for the noise model parameter estimation (see subsection 2.1.2.)
- An initial value $\Phi_y^{[0]}$ not too far from the optimal parameters (see subsection 2.1.3).
- A way of simulating realizations of X according to the *posterior* distribution $P_{Y/X}(y/x)$ by using the Gibbs sampler [6]. For the *prior* model, we adopt an 8-connexity spatial neighborhood (see Figure 1) in which β_1, β_2, β_3, β_4 represent the *a priori* potential associated to the horizontal, vertical, right and left diagonal binary cliques respectively. In our application, we want to favour homogeneous regions. Then, we define potential functions associated to the two-site cliques of the form :

$$\beta_{s,t} = \mu[1 - \delta(x_s, x_t)] \tag{3}$$

where $\beta_{s,t} = \beta_1$, β_2, β_3 or β_4 according to the type of neighboring pair $\langle s, t\rangle$, μ is a predetermined positive constant and $\delta(.)$ is the Kronecker function.

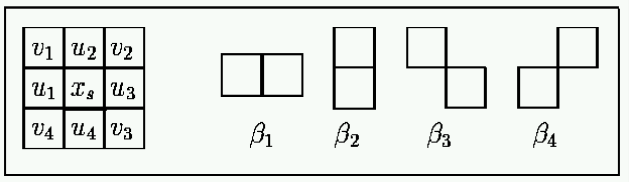

Figure 1: 2nd order neighborhood and two-site associated cliques

2.1.2 Estimation of the noise model parameters for the complete data

The Gaussian law $\mathcal{N}(\mu, \sigma^2)$, is an appropriate degradation model to describe the luminance y within *shadow* regions (essentially due to electronical noise) [15]. The most natural choice of the estimator $\hat{\Phi}_y\,(x = e_0, y)$ is the empirical mean and the empirical variance. If N_0 pixels are located in the shadow areas, we have :

$$\hat{\mu}_{ML} = \frac{1}{N_0} \sum_{s \in S: x_s = e_0} y_s \tag{4}$$

$$\sigma^2_{ML} = \frac{1}{N_0 - 1} \sum_{s \in S: x_s = e_0} (y_s - \hat{\mu}_{ML})^2 \tag{5}$$

In order to take into account the speckle noise phenomenon [7] in the reverberation areas, we model the conditional density function of the reverberation class by a shifted Rayleigh law $\mathcal{R}(\min, \alpha^2)$ [15] :

$$P(y_s/x_s = e_1) = \frac{y_s - \min}{\alpha^2} \exp\left[-\frac{(y_s - \min)^2}{\alpha^2}\right] \tag{6}$$

The maximum value of the log-likelihood function is used to determine a Maximum Likelihood estimator of the complete data. If $\hat{y}_{\min}$is the minimum grey level in the reverberation areas and N_1 the number of pixels located within this region, we obtain for $\hat{\Phi}_y\,(x = e_1, y)$ the following results [15] :

$$\sigma^2_{ML} = \frac{1}{2N_1} \sum_{s \in S:\ x_s = e_1} (y_s - \widehat{\min}_{ML})^2 \tag{7}$$

$$\widehat{\min}_{ML} \simeq \hat{y}_{\min} - 1 \tag{8}$$

In the two cases, the proportion π_k of the k^{th} class is given by empirical frequency :

$$\overset{\wedge}{\pi}_k = \frac{N_k}{N_0 + N_1} \qquad \text{with } k \in \{0, 1\} \tag{9}$$

2.1.3 Initialisation

The initial parameter values have a significant impact on the rapidity of the convergence of the ICE procedure and the quality of the final estimates. In our application, we use the following method : the initial parameters of the noise model $\Phi_y^{[0]}$ are determined by applying a small non overlapping sliding window over the image and calculating the sample mean, variance and minimum grey level estimates. Each estimation calculated over the sliding window gives a "sample" $\mathbf{x}_i$, a three component vector. These samples $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$ are then clustered into two classes $\{e_0, e_1\}$ using the *K-means clustering procedure* [1]. This algorithm uses a similarity measure based on the Euclidean distance between samples. A criterion is based on the minimization of the related cost-function defined by :

$$J = \sum_{i=1}^{K} \sum_{x_l \in C_i} |\mathbf{x}_l - c_i|^2 \tag{10}$$

where the second sum is over all samples in the i^{th} cluster and c_i is the *center* of this cluster. It is easy to show that for a given set of samples and class assignments, J is minimized by choosing c_i to be the *sample mean* of the i^{th} cluster. Moreover, when c_i is a sample mean, J is minimized by assigning $\mathbf{x}_j$ to the class of the cluster with the nearest mean. A number of other criteria are given in [1]. The complete algorithm is outlined below :

1- Choose K initial clusters $c_1^{[1]}, ..., c_K^{[1]}$. These could be arbitrarily chosen, but are usually defined by :

$$c_i^{[1]} = \mathbf{x}_i \qquad (1 \leq i \leq K) \tag{11}$$

2 - At the k^{th} step, assign the sample $\mathbf{x}_l$, $(1 \leq l \leq M)$ to cluster i if

$$\left\| \mathbf{x}_l - c_i^{[k]} \right\| < \left\| \mathbf{x}_l - c_j^{[k]} \right\| \qquad (\forall j \neq i) \tag{12}$$

In fact, we reassign every sample to the cluster with the nearest mean. In the case of equality, we assign $\mathbf{x}_l$ arbitrary to i or j .

3 - Let $c_i^{[k]}$ denote the i^{th} cluster after Step 2. Determine new clusters by :

$$c_i^{[k+1]} = \frac{1}{N_i} \sum_{x_l \in c_i^{[k]}} \mathbf{x}_l \tag{13}$$

where N_i represents the number of samples in $c_i^{[k]}$. Thus, the new cluster position is the mean of the samples in the previous one.

4 - Repeat until convergence is achieved, say $c_i^{[k+1]} = c_i^{[k]} \quad \forall i$

Although it is possible to find pathological cases where convergence never occurs [1], the algorithm does converge in all tested examples. The rapidity of convergence depends on the number K, the choice of initial cluster centers and the order in which the samples are considered. In our application $K = 2$. Figure 2.a represents a sonar image and the result of the K-means clustering algorithm is reported in Figure 2.b.

On one hand, a small size window increases the accuracy of the segmentation and then the precision of the distribution mixture estimation. On the other hand, it decreases the number of pixels with which $\mathbf{x}_l$'s are computed and may increase the misclassification error. In our application, good results are obtained with a 6×6 pixels window. The **ML** estimation is then used over the K-means segmentation in order to find $\Phi_y^{[0]}$.

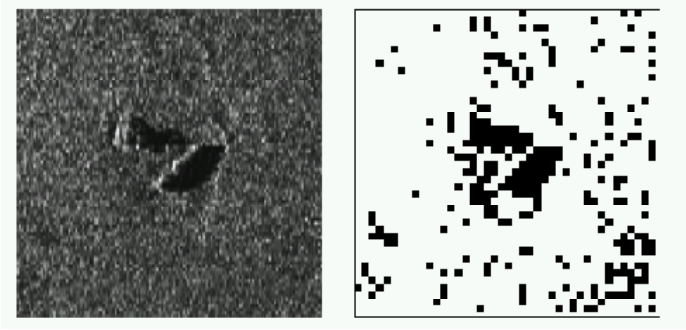

Figure 2: K-means clustering procedure. a) sonar picture involving an object and a rock shadows b) Segmentation according to the Maximum Likelihood criterion with parameter estimation given by the K-means algorithm

2.1.4 Parameter estimation procedure for the incomplete data

We can use the following algorithm to estimate the noise model parameters. Let us recall that this method takes into account the diversity of the laws in the distribution mixture estimation.

- Parameter initialization :

The K-means algorithm is used. Let us denote $\Phi_y^{[0]}$, the obtained result.

- ICE procedure :

$\Phi_y^{[k+1]}$ is computed from $\Phi_y^{[k]}$ in the following way :

$\triangleright$ Using the Gibbs sampler, n realizations $x_{(1)}, ..., x_{(n)}$ are simulated according to the posterior distribution with parameter vector $\Phi_y^{[k]}$, and with :

$$P_{Y_s/X_s}(y_s/x_s = e_0) \qquad \text{a Gaussian law for shadow area}$$

$$P_{Y_s/X_s}(y_s/x_s = e_1) \qquad \text{a shifted Rayleigh law for reverberation area}$$

$\triangleright$ For each $x_{(i)}$ with $i = 1, ..., n$, the parameter vector Φ_y is estimated with the **ML** estimator on each class:

$\triangleright$ $\Phi_y^{[k+1]}$ is obtained from $\Phi_y(x_{(i)}, y)$ with $(1 \leq i \leq n)$ by using equation (2).

If the sequence $\Phi_y^{[k]}$ becomes steady, the ICE procedure is ended and one proceeds the segmentation using the estimated parameters. We can use all these estimated parameters in order to get a complete unsupervised Markovian segmentation (see subsection 3.2) or only use the proportion of each class in the Kohonen SOM-based unsupervised segmentation described in subsection 2.2.

We calibrate the weight of the *"stochastic"* aspect of the ICE by choosing n. When n increases, the *"stochastic"* aspect of the algorithm decreases. The choice of a small value for n ($n = 1$ in our application) can increase its efficiency [3].

Figure 3 represents the mixture of distributions of the sonar image reported in Figure 2.a. The obtained results are given in table 1.

The quality of the estimations is difficult to appreciate in absence of real values. We can roughly perform such evaluation by comparing the image histogram with the probability density mixture corresponding to the estimated parameters. Figure 3.a shows the resulting mixture solution in graphical form. The two dashed curves in the figures represent the individual components $P_{Y/X_i}(y/e_m)$ with $0 \leq m \leq K$. The histogram

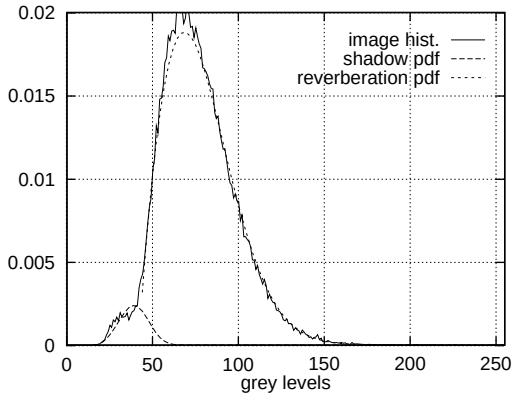

Figure 3: Image histogram of the picture reported in Figure 2 with the estimated Gaussian and Rayleigh laws.

is quite close to the mixture densities based on the estimated parameters, and a segmentation with these estimates can be done as shown in the following section.

Initialisation **of K-means** procedure			
$\Phi^{[final]}_{y(shadow)}$	$0.04_{(\pi)}$	$36_{(\mu)}$	$55_{(\sigma^2)}$
$\Phi^{[final]}_{y(sea-bottom)}$	$0.96_{(\pi)}$	$39_{(\min)}$	$1061_{(\alpha^2)}$

ICE procedure			
$\Phi^{[final]}_{y(shadow)}$	$0.03_{(\pi)}$	$32_{(\mu)}$	$17_{(\sigma^2)}$
$\Phi^{[final]}_{y(sea-bottom)}$	$0.97_{(\pi)}$	$39_{(\min)}$	$1591_{(\alpha^2)}$

Table 1 : Estimated parameters on the pictures reported in Fig.2a. π stands for the proportion of the two classes within the sonar image. μ and σ^2 are the Gaussian parameters (*shadow area*). min and α are the Rayleigh law parameters (*reverberation area*). $\Phi^{[0]}_y$ represents the initial parameter estimates and the final estimates are denoted $\hat{\Phi}_y$.

2.2 Self-Organizing Map

2.2.1 Introduction

Researchs on neurobiology have shown that centers of diverses activities as thought, speech, vision, hearing, lie in specific areas of the cortex and these areas are ordered to preserve the topological relations between informations while performing a dimensionality reduction of the representation space. Such organization led Kohonen to develop the Self-Organizing Map (**SOM**) algorithm [8]. This kind of competitive neural network is composed of one or two dimensional array of processing elements or neurons in the input space. All these neurons receive the same inputs from external world. Learning is accomplished by iterative application of unlabeled input data. As training process, the neurons evolve in the input space in order to approximate the distribution function of the input vectors. After this step, large-dimensional input vectors are, in a sense, projected down on the one or two-dimensional map in a way that maintains the natural order of the input data. This dimensional reduction could allow us to visualize and to use easily, on a one or two-dimensional array, important relationships among the data that might go unnoticed in a high-dimensional space.

The model of **SOM** used in our application is a one-dimensional array of n nodes. To each neuron N_i , a weight vector $w_i = (w_{i1}, w_{i2}, ..., w_{ip})^t \in \mathbb{R}^p$ is associated. During learning procedure, an input vector $\mathbf{x} \in$

$\mathbb{R}^p$ randomly selected among vectors of the training set, is connected to all neurons in parallel. The input $\mathbf{x}$ is compared with all the neurons in the Euclidian distance sense *via* variable scalar weight w_{ij}. At the k^{th} step, we assign the vector $\mathbf{x}$ to the winning or leader neuron N_l if :

$$\left\|\mathbf{x} - \mathbf{w}_l^{[k]}\right\| = \min_i \left\|\mathbf{x} - \mathbf{w}_i^{[k]}\right\| \tag{14}$$

All the neurons within a certain neighborhood around the *leader* participate in the weight-update process. Considering random initial values for $\mathbf{w}_i^{[0]}$ $(0 \leq i \leq n)$, this learning process can be described by the following iterative procedure :

$$\mathbf{w}_i^{[k+1]} = \mathbf{w}_i^{[k]} + \mathcal{H}_{li}^{[k]}(\mathbf{x}^{[k]} - \mathbf{w}_i^{[k]}) \tag{15}$$

The lateral interactions among topographically close elements are modeled by the application of *a neighborhood function* or a smoothing Kernel defined over the winning neuron [8]. This Kernel can be written in terms of the Gaussian function :

$$\mathcal{H}_{li}^{[k]} = \alpha^{[k]} \exp\left(-\frac{d^2(l,i)}{2(\sigma^{[k]})^2}\right) \tag{16}$$

where $d(l,i) = \|l - i\|$ is the distance between the node l and i in the array, $\alpha^{[k]}(t)$ is the learning-rate factor and $\sigma^{[k]}$ defines the width of the Kernel at the iteration k . For the convergence, it is necessary that $\mathcal{H}_{li}^{[k]} \to 0$ when $k \to T$, where T is the total number of steps of the process [8]. Therefore, for the first step, $\alpha^{[k]}$ should start with a value that is close to unity, thereafter decreasing monotonically [8]. To achieve this task, we use :

$$\alpha^{[k]} = \alpha^{[0]}(1 - \frac{k}{T}) \tag{17}$$

Moreover, as learning proceeds, the size of the neighborhood should be diminished until it encompasses only a single unit. So, we applied for the width of the Kernel the monotonically decreasing function:

$$\sigma^{[k]} = \sigma^{[0]} \left(\frac{\sigma^{[T-1]}}{\sigma^{[0]}}\right)^{k/(T-1)} \tag{18}$$

The ordering of the map occurs during the first steps, while the remaining steps are only needed for the fine adjustement of the weight values.

2.2.2 Iterative learning step

The learning process is performed directly on the real image to be segmented. An input vector is filled with the grey levels of the pixels contained in a 3×3 pixels window sliding over the image (cf. figure 4). Therefore, each neuron has nine weights allowing to locate it in the input space. At each step, the location of the window in the image is randomly chosen and the weights are modified according to (15). Experiments have shown that this trainning strategy provides as good results as an ordered image scanning process while spending less processing time.

σ has a significant impact on the quality of the convergence. We have to start with a fairly large value to globally order the map. The initial value σ_0 of σ can be half the length of the network. During learning, σ has to decrease monotonically until it reaches a small value. Experiments have shown that $\sigma_{T-1} = 0.1$ is a good choice and provides the minimum quantization error defined by :

$$E_{quant} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{w}_l\| \tag{19}$$

where the summation is over all the N windows of the image and $\mathbf{w}_l$ is the weight vector associated to the leader neuron of the input vector $\mathbf{x}_i$ after learning step.

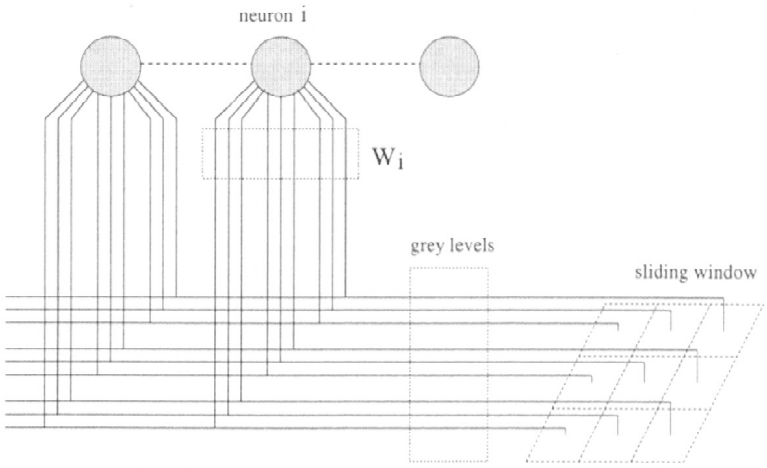

Figure 4: Model of the SOM used for the segmentation. An 3×3 sliding window is used to feed the SOM.

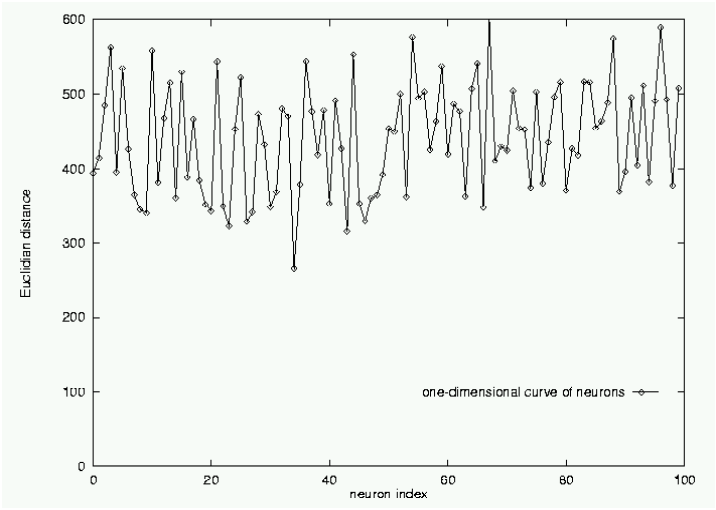

Figure 5: The distance graph between the 100 neurons of the SOM before learning.

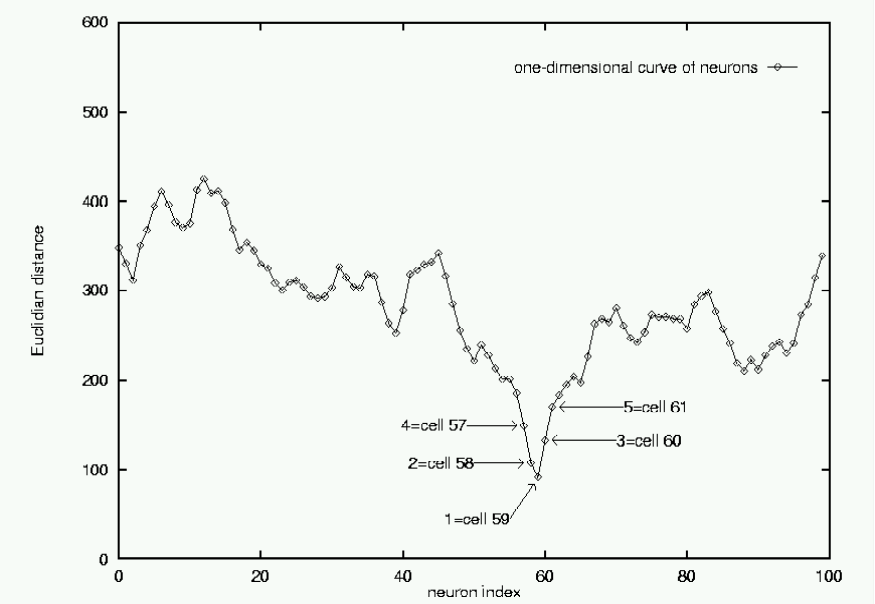

Figure 6: The distance graph between neurons obtained after learning

3 Segmentation step

3.1 SOM Segmentation

The classification task consists in running the sliding window over the image. For each location of the window, the corresponding input vector $\mathbf{x}$ is compared with all the neurons using (14). The *winning neuron*, the one which leads to the smallest distance, gives the class of the pixel located in the center of the window. However, before any classification task, we have to *calibrate* the map in order to associate the label *shadow* or *reverberation* to each neuron.

Assuming that the input vector $\mathbf{x}_0 = (0, ..., 0)^t$ should represent a window setting on a perfect *shadow* area, it is very useful to define the distance graph representing the Euclidean distance in the 9-dimensional space between the point $\mathbf{x}_0$ and all the neurons. Such a graph is given in figure 5, before and after learning for a hundred-neuron network.

Both these figures show that the maximal distance between two successive cells is widely smaller after learning than before. We can deduce that, after learning, neurons that are topologically close in the array are close in the input space too. As a matter of fact, neurons that are physical neighbors should respond to a similar input vectors. The calibration of the map uses this topological property and the portion π_0 estimated in subsection 2.1 of the pixels labelled as *shadow* in the image. This process can be summarized as follows:

1 - Initially, we affect the class *reverberation* to all neurons.

2 - We seek the most evident prototype of the shadow class. This neuron is the winning unit according to equation (14) when inputting the vectors $\mathbf{x}_0$. Then, we affect it to the *shadow* class.

3 - We affect the *shadow* class to pixels for which the leader neuron belongs to the *shadow* class. We can deduce the intermediate *shadow* portion π_{int} provided by the resulting image.

4 - If π_{int} is smaller than π_0, we have to select an additional prototype of a shadow class among neurons of the *reverberation* class. According to the topological preserving properties of the map, this additional neuron should be a direct neighbor of an already *shadow* labelled neuron. Among both the possible neighbors, we take the one which has the smallest Euclidean distance with the point $\mathbf{x}_0$. Go to 3.

5 - If π_{int} is larger than π_0, we stop the process.

Experiments have shown that E_{quant} is a monotonically decreasing function of the number of steps and reaches an asymptotic value for large value of T. One hundred times the number of network units sems to be a reasonable compromise solution between speed and quality of learning. In our application, 100 neurons have been chosen for the network.

3.2 Markovian segmentation

The segmentation of sonar images in two classes can be viewed as a statistical labelling problem according to a global Bayesian formulation in which the posterior distribution $P_{X/Y}(x/y) \propto \exp[-U(x,y)]$ has to be maximized [2]. In our case, the corresponding *posterior* energy $U(x,y)$ to be minimized is :

$$U(x,y) = \underbrace{\sum_{s \in S} \Psi_s(x_s, y_s)}_{U_1(x,y)} + \underbrace{\sum_{\langle s,t \rangle} \beta_{s,t} \left[1 - \delta(x_s, x_t)\right]}_{U_2(x)} \tag{20}$$

where U_1 denotes the adequacy between observations and labels $\left(\Psi_s(x_s, y_s) = \ln\left[P_{X_s/Y_s}(x_s/y_s)\right]\right)$ and U_2 expresses the energy of *a priori* model. In order to minimize this energy function, we use a deterministic relaxation technique called **ICM** algorithm [2, 4, 16, 11].

3.3 Results on real scenes

We compare the segmentation performance of the proposed **SOM**-based algorithm described in subsection 3.1 with a **ML** segmentation and a classical markovian segmentation using a deterministic relaxation technique such as the **ICM** [2]. All the segmentation results exploit the parameter estimation step presented in section 2.1. This estimation step is used to estimate both the noise model parameters for the ML segmentation, the markovian segmentation and the proportion of the *shadow* class for the **SOM** segmentation.

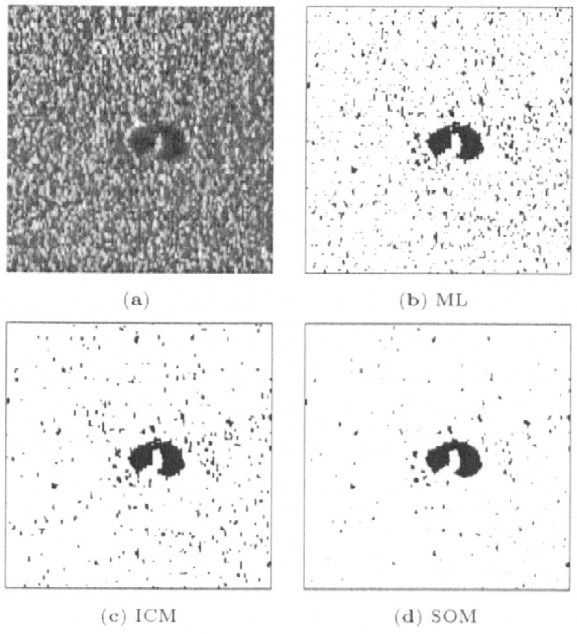

Figure 7: (a) - A real sonar image involving a sandy sea floor with the cast shadow of a tyre. Two-class segmentation results obtained with this image using: (b) - **ML** segmentation, (c) - Markovian segmentation with a deterministic relaxation technique as **ICM**, (d) - The **SOM** based segmentation result. (see Table 2 for the estimated parameters). The **SOM** segmentation exhibits a good robustness againts the speckle noise (which induces false small shadow areas compared to the others approaches)

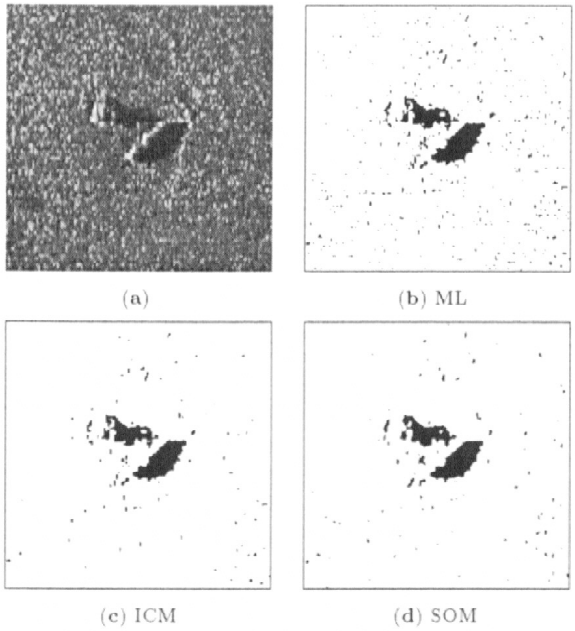

Figure 8: (a) - A real sonar image involving an object and a rock shadows. Two-class segmentation results obtained with (a). (b) - **ML** segmentation, (c) - Markovian segmentation with **ICM** technique, (d) - **SOM**-based segmentation method. (see Table 3 for the estimated parameters). The **ML** and the **ICM** do not permit to totally eliminate the speckle noise effect (creating *shadow* mislabelled isolated pixels).

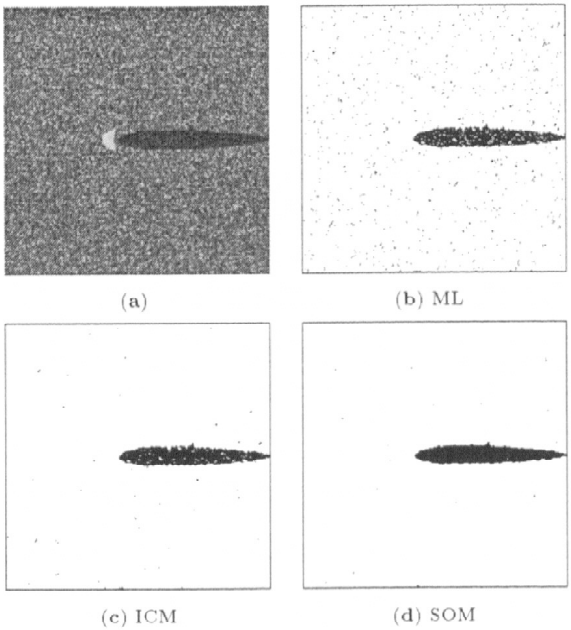

Figure 9: (a) - A synthetic sonar image of a sphere lying on the seabed. Segmentation results obtained with: (b) - **ML** segmentation, (c) - Markovian **ICM** technique, (d) - **SOM**-based segmentation method.

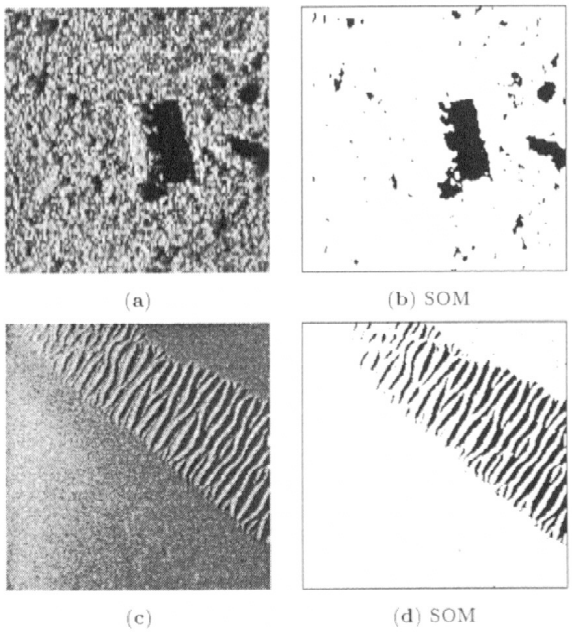

Figure 10: Real sonar images of a cylindrical object (a) and of ridges of sand (c). Their corresponding SOM-based segmentation results are depicted respectively in (b) and (d):

Fig.6	**ICE**	procedure	
$\Phi_{y(shadow)}^{[final]}$	$0.02_{(\pi)}$	$36_{(\mu)}$	$85_{(\sigma^2)}$
$\Phi_{y(sea-bottom)}^{[final]}$	$0.98_{(\pi)}$	$46_{(\min)}$	$1878_{(\alpha^2)}$

Table 2 : Estimated parameters on the picture reported in Fi.6

Fig.7	**ICE**	procedure	
$\Phi_{y(shadow)}^{[final]}$	$0.03_{(\pi)}$	$25_{(\mu)}$	$32_{(\sigma^2)}$
$\Phi_{y(sea-bottom)}^{[final]}$	$0.97_{(\pi)}$	$35_{(\min)}$	$1430_{(\alpha^2)}$

Table 3 : Estimated parameters on the picture reported in Fi.7

Fig.8	**ICE**	procedure	
$\Phi_{y(shadow)}^{[final]}$	$0.03_{(\pi)}$	$34_{(\mu)}$	$39_{(\sigma^2)}$
$\Phi_{y(reverberation)}^{[final]}$	$0.97_{(\pi)}$	$42_{(\min)}$	$1412_{(\alpha^2)}$

Table 4 : Estimated parameters on the picture reported in Fi.8

Figures 6, 7, 8 and 9 show the segmentation results obtained with the different methods. Example of the noise model parameters Φ_y obtained with our scheme are given in Tables 2, 3 and 4.

Experiments indicate that the SOM segmentation requires less computation than the markovian segmentation (30" for the SOM estimation-segmentation whereas roughly 100" are required for unsupervised

scale causal Markovian modelization [12] on IBM 43P-200MHz workstation). Besides, the ICM algorithm do not permit to decrease the number of false alarm (wrong detections) due to the speckle noise effect. The SOM segmentation performs better, exhibits a good robustness versus speckle noise (false alarms have been eliminated), and allows us to preserve the shadow shapes of little rocks.

Manufactured objects or rock shadows are better segmented with our method than with the others (cf. Figures 6 and 7) and their shape are close to the result we expected. The cast shadow of a manufactured object (a cylinder) reported in Figure 7 has a geometric shape (contrary to the cast shadow of the rock) that will be discriminant for the classification step.

4 Conclusion

We have described an unsupervised segmentation procedure based on a *parameter estimation step* (which offers an appropriate estimation of the noise model) and a segmentation step well adapted for sonar image segmentation problem. The estimation step takes into account the diversity of the laws in the distribution mixture of a sonar image and can be used with the Kohonen **SOM**-based segmentation in order to solve the difficult problem of unsupervised sonar image segmentation. This scheme is computationally simple and appears as an interessing alternative to existing complex hierarchical markovian methods. This method has been validated on several real sonar images demontrating the efficiency and robustness of this scheme.

References

[1] S. Banks. *Signal processing image processing and pattern recognition.* Prentice Hall, 1990.

[2] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–236, 1974.

[3] B. Braathen, P. Masson, and W. Pieczynski. Global and local methods of unsupervised bayesian segmentation of images. *machine GRAPHICS and VISION*, (1):39–52, 1993.

[4] C. Collet, P. Thourel, P. Pérez, and P. Bouthemy. Hierarchical MRF modeling for sonar picture segmentation. In *Proc. 3^{rd} IEEE International Conference on Image Processing*, volume 3, pages 979–982, Lausanne, Sept. 1996.

[5] P. Galerne, K. Yao, and G. Burel. Objects Classification using Neural Network in Sonar Imagery. In *Proc. European Symposium on Lasers and Optics in Manufacting EUROPTO*, volume SPIE 3101, pages 306–314, Munich, June 1997.

[6] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.

[7] J. W. Goodman. Some fundamental properties of speckle. *Journal of Optical Society of America*, 66(11):1145–1150, November 1976.

[8] T. Kohonen. *Self organizing maps.* Springer, 1995.

[9] P. Masson and W. Pieczynski. S.e.m algorithm and unsupervised statistical segmentation of satellite images. *IEEE Trans. On Géoscience and Remote Sensing*, (3):618–633, 1993.

[10] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy. Unsupervised Hierarchical Markovian segmentation of sonar images. In *Proc. ICIP*, volume 3, Santa Barbara, Californie, USA, October 1997.

[11] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy. Unsupervised Markovian segmentation of sonar images. In *Proc. ICASSP*, volume 4, pages 2781–2785, Munchen, May 1997.

[12] M. Mignotte, C. Collet, P. Perez, and P. Bouthemy. Sonar image segmentation using an unsupervised hierarchical mrf model. *IEEE Trans. on Image Processing*, Accepted for publication on March 6 1999 - To appear, 1998.

[13] F. Salzenstein and W. Pieczinsky. Unsupervised bayesian segmentation using hidden markovian fields. *Proc. ICASSP*, pages 2411–2414, May 1995.

[14] F. Schmitt, L. Bonnaud, and C. Collet. Contrast control for sonar pictures. *Signal and Image Processing, SPIE'96 - Technical Conference on Application of Digital Image Processing XIX*, 2847:70–82, August 1996.

[15] F. Schmitt, M. Mignotte, C. Collet, and P. Thourel. Estimation of noise parameters on sonar images. In *SPIE Statistical and Stochastic Methods for Image processing*, volume 2823, pages 1–12, Denver, 4-5 August 1996.

[16] P. Thourel, C. Collet, P. Bouthemy, and P. Pérez. Multiresolution analysis and MRF modelling applied to the segmentation of shadows in sonar pictures. In *Proc. 2^{nd} Asian Conference on Computer Vision*, volume 2, pages 81–85, Singapore, Dec. 1996.