Using Self-Organizing Maps Approach to Pipeline Localization

Amornrit Puttipipatkajorn, Bruno Jouvencel, Tomas Salgado-Jimenez

Using Self-Organizing Maps approach to pipeline localization

Puttipipatkajorn Amornrit

LIRMM –University of Montpellier II

161 rue Ada, 34392, Montpellier, Cedex 5, France
E-mail : putti@lirmm.fr

Bruno Jouvencel

LIRMM –University of Montpellier II

161 rue Ada, 34392, Montpellier, Cedex 5, France
E-mail : jouvencel@lirmm.fr

Salgado-Jimenez Tomas

LIRMM –University of Montpellier II

161 rue Ada, 34392, Montpellier, Cedex 5, France
E-mail : salgado@lirmm.fr

Abstract-**The aim of this paper is to detect and follow the pipeline in sonar imagery. This work is performed in two steps. The first is to split an image (first experiment) or an transformed line image of pipeline image (second experiment) into regions of uniform texture using the Gray Level Co-occurrence Matrix Method (GLCM). The second addresses the unsupervised learning method based on the Artificial Neural Networks (Self-Organizing Map or SOM) used for determining the comparative model of pipeline from the image. To increase the performance of SOM, we propose a penalty function based on data histogram visualization for detecting the position of pipeline. After a brief review of both techniques (GLCM and SOM), we will present our methods and some results from several experiments on the real world data set.**

I. INTRODUCTION

The Side Scan Sonar appears to be, like human interpretation, an advanced means to analyze the seabed. It enables us to visualize echo, shadow and sea bottom reverberation. Data from echo reverberation comes from the reflection of wave off of an object. The shadow area which appears corresponds to the absence of acoustic reverberation behind it, whilst the rest of what is visualized, to sea bottom reverberation. The only available type of sonar image is the grey level of pixels corresponding to the acoustic reflectance. Many studies have been carried out on the performance of the various families of computational methods, for instance, the 2-dimensions of FFT, the Gray Level Co-occurrence Matrix (GLCM), Gray Level Run Length and etc. In addition, a comparative study from several methods show that the GLCM is an excellent statistical tool for extracting second-order texture information from images. For our study we did not use GLCM for directly detecting the position of pipeline, as the data from Side Scan Sonar is only one dimensional space. Data therefore had to be transformed into two dimensional space using the transformed line method, (described later in this paper).

The co-occurrence matrix is used to estimate the joint probability density function of gray-level pairs in an image. The matrix is in general symmetric and, when normalized, element values are bounded by [0,1], and the sum of all element value equal to 1. Features extracted from this matrix are called second-order statistical features, for instance, energy, entropy, inverse difference moment, and etc.

The next section of this paper concerns clustering algorithms based on the Self-Organizing Map (SOM) [1]. This method is applied in various fields such as data mining [2], image segmentation [6] and also pattern recognition. The SOM is a neural network algorithm based on unsupervised learning. It is an efficient tool for visualizing multidimensional numerical data. It shows high dimensional data in the form of low dimensional 1D or 2D grid. Several methods used to visualize clustering have been written about. The most widely used of these is distance matrix technique[2][3], in particular the unified distance matrix or U-matrix. Another is the data histogram method.

The aim of this paper is to detect the position of pipeline using of the comparative model of pipeline image, derived directly from the data histogram method. To find the pipeline position, the penalty function is formed and the object of interest found while its penalty value is nearest to zero.

II. ARCHITECTURE OF SEABED RECOGNITION SYSTEM FOR DETECTING A PIPELINE

The basic seabed recognition and detection system is composed principally of the training process and testing one (Fig. 1). The aim of the first is to evaluate labeled patterns or pipeline image in order to obtain a models (i.e. a comparative model). During this phase the labeled patterns are trained by the SOM network until the network fold. At this stage the comparative model is apparent. In the second of these processes a testing phase evaluates the model of the arbitrary pipeline images using the SOM network of the aforementioned process. During this testing process, the comparative model and tested image model are compared using the penalty function to estimate the position of pipeline.

Each process in the diagram below consists of three elements: pre-processing, features extraction and neural network. The role of the pre-processing module is to remove noise and normalize the pattern. In the next section of this paper the following will be described, The line transformation, the Gray Level Co-occurrence Matrix (GLCM) and the Self-Organizing Map (SOM).
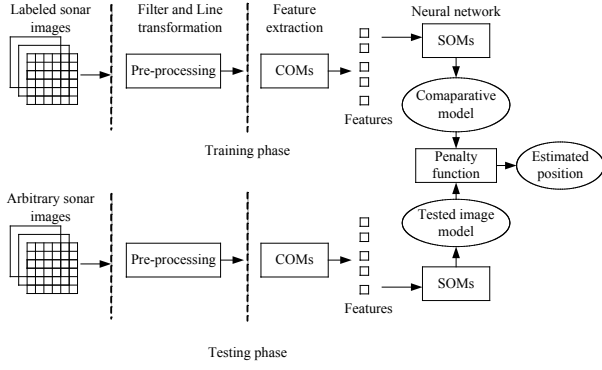
Fig. 1. Architecture of seabed recognition and detection system

III. THE GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM)

III.1. Co-occurrence Matrix

The co-occurrence matrix, P, represents the repeated occurrence of pairs of pixels (i,j) going from gray level i to gray level j through distance d along direction δ. Let $I_x=\{1,2,...,N_x\}$ and $I_y=\{1,2,...,N_y\}$ be the **X** and **Y** spatial domains, where $I_x \times I_y$ is the set of resolution of square image, and the digital image I contain a finite number of gray-level value $g \in G\{1,2,...,N_g\}$ for every pixels, formally $I:I_x \times I_y \rightarrow G$. Let the distance d is the distance between two-pixel positions (x_1,y_1) and (x_2,y_2), which indicated by angular angle δ. The matrix, P, is $N_g \times N_g$ square matrix, where N_g-1 is gray value in the image.
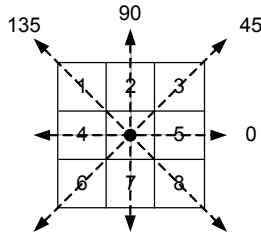

Fig. 2.

The co-occurrence matrix allows us to derive four matrix for each given distance: $P(0,d)$, $P(45,d)$, $P(90,d)$ and $P(135,d)$, as indicated in Fig. 2.

III.2. Statistical texture description functions

Prior to calculating the statistical texture descriptors, the matrices are normalized. They estimate the joint probability densities of the co-occurrence gray level.

$$P(i,j)=P(i,j)/N \text{ , where } N=\sum_i\sum_j P(i,j) \tag{1}$$

From the normalized co-occurrence matrix, a set of textural features is extracted. In our experiments, the most relevant features used are listed below.

$$\text{Energy: F1} = \sum_i\sum_j P(i,j)^2 \tag{2}$$

$$\text{Entropy: F2} = -\sum_i\sum_j P(i,j)\log P(i,j) \tag{3}$$

$$\text{Maximum probability: F3} = \max\{P(i,j)\} \tag{4}$$

$$\text{Inverse difference moment: F4} = \sum_i\sum_j \frac{P(i,j)}{1+(i-j)^2} \tag{5}$$

$$\text{Contrast: F5} = \sum_i\sum_j (i-j)^2 P(i,j) \tag{6}$$

$$\text{Homogeneity: F6} = \sum_i\sum_j \frac{P(i,j)}{1+|i-j|} \tag{7}$$

In this case, we obtain texture feature vectors, $F=\{F_1,F_2,...,F_6\}$. Each element contains information of image texture calculating from statistical description functions above.

IV. SELF-ORGANIZING MAP (SOM)

SOM belongs to the category of unsupervised learning neural networks. The SOM has only two layers of neurons, an input layer and a competitive layer. Each node in the input layer is connected to every node in the competitive layer. The nodes in the competitive layer may also be connected to each other by various modes of connection, such as squared neighboring connection.

The model of SOM used in our work is a two-dimensional array of k nodes. Each neuron k is represented by an n-dimensional vector $m_k=[m_{k1},...,m_{kn}]$, where n is the dimension of the input space. On each training step, a data sample x is randomly selected and the best-matching unit (BMU or m_c) is found on the map unit:

$$\|x-m_c\|=\min_k\{\|x-m_k\|\} \tag{8}$$

Then, the vector m_c and its neighbours on the grid are updated by bringing them closer to the sample vector:

$$m_k=m_k+\alpha(t)h_{ck}(t)(x-m_k) \tag{9}$$

where t denotes time, $\alpha(t)$ is learning rate and $h_{ck}(t)$ is a neighborhood kernel centered on the winner unit c :

$$h_{ck}(t) = \exp\left(-\frac{\|r_c - r_k\|^2}{2\sigma^2(t)}\right) \qquad (10)$$

and

$$\alpha(t) = \frac{\alpha_0}{1+100t/T} \qquad (11)$$

where $\|r_c - r_k\|$ is distance between map units of neurons c and k on the SOM grid. In equation (11), α_0 denotes initial learning rate and T is the total iterative time. Both learning rate function $\alpha(t)$ and neighborhood kernel radius decrease monotonically with time.

During the iterative training, the SOM adapt to input data set in such a way that the model vectors which belong to units close to each other on the map unit, are also close to each other in the data space.

IV.1. Data histogram method

The aim of the data histogram method is to display the number of hits in each map unit. This means that each unit of map belongs to a number of the best-matching units of any given vectors. For instance, if we have 20 input vectors and 2×2 map unit, the first possesses 6 of the best-matching units and the remaining ones 4, 2 and 8 of them respectively. The matrix of the best matching unit in the map unit is shown in fig. 3. The normalization of this matrix, called the model of the network, will be employed in the section which follows.
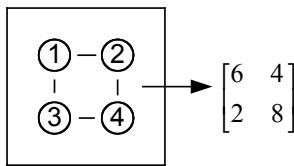

Fig. 3. Matrix of the best-matching unit in the map unit

IV.2. Penalty function

In this part of our paper we deal with the penalty function based on the histogram visualization method. For our work the SOM was not directly used to classify data, but used for evaluating a comparative model. This means that the SOM network generally contains a number of categories of given input in one model. These categories could perhaps be clustered together using the well-known method such as the U-matrix. In this section, we are only interested single group of data, i.e. pipeline data. Our model, which contains only data of this type, will be used for evaluating the penalty function shown as the equation below.

$$E^k(i,j) = \sum_{i=1}^{n}\sum_{j=1}^{m} I_w(i,j) \otimes \left\{I^k(i,j) - I_w(i,j)\right\} \qquad (12)$$

Where, I_w is the matrix model or the comparative model of specific data, I^k is the one of tested sample, k denotes the index of sliding-window and $\otimes$ is product of matrix in term by term. The best-matching sample found while the value of E is nearest to zero.

V. FIRST EXPERIMENT

The first experiment aims to determine the pipeline position by using complete pipeline image, and the GLCM is directly applied to the image without line transformation method which will be used in the second experiment.

V.1. Methods

For the first experiment the training phase has a set of labeled images of object of interest, called training window. Each contains a single object of interest and its environment with a standard size and orientation, the structure of training window shown in Fig. 4. During the training phase, the training window is firstly trained by SOM network in order to obtain a comparative model of the object, this model is represented by the matrix of probability density I_w in (12).
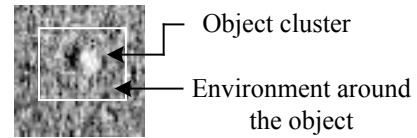

Fig. 4. The structure of training window

By means of the trained SOM model, the matrix of probability density of sliding-window, I^k, or testing window is calculated. During this phase, the trained window, I_w, is compared to tested window, I^k, sliding through the image from left to right and top to bottom. To find the object location, the penalty value E^k for every k is calculated. The object of interest is located on the tested window k which has the penalty value nearest to zero.

V.2. Experimental Results

In training step, the training window I_w selected from the testing image I represents the pipeline image model (see fig. 5).
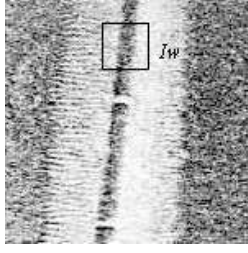
Fig. 5. The pipeline sonar image I of 200×200 pixels and training window I_w of 30×30 pixels

A size of 4×4 units of SOM grid was selected for the best result from a number of experiments. This experiment used the testing image of 50×50 pixels and the training window of 15×15 pixels. The result is shown in figure as follows.
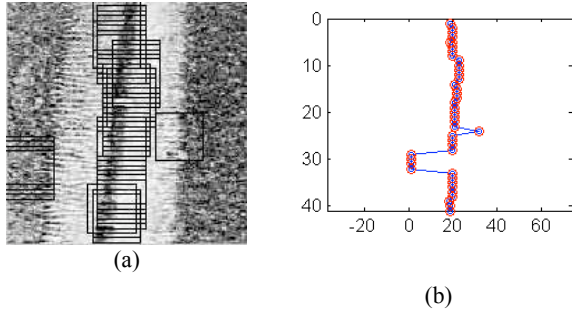

(a)

(b)

Fig. 6. (a) The pipeline found by SOM model, (b) some missing windows corresponding to position of window.

To eliminate the missing windows we calculate total mean (μ) and total standard deviation (σ) of position of each window in vertical line and the positions of missing window are replaced by the total mean, if their positions with respect to total mean are bigger than the total standard deviation.
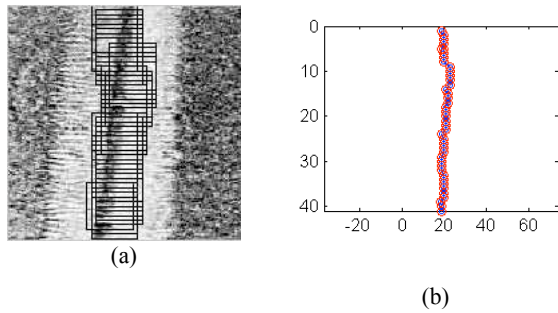

(a)

(b)

Fig. 7. Eliminating the missing windows using total mean and total standard deviation of window position.
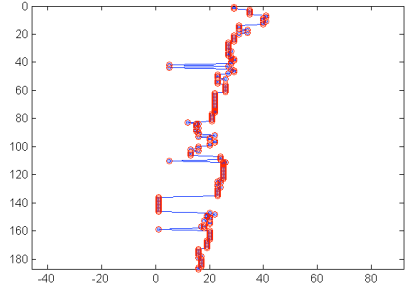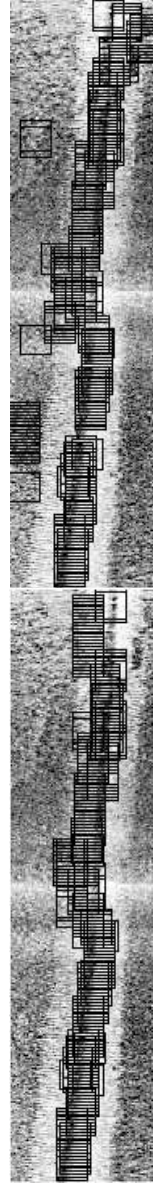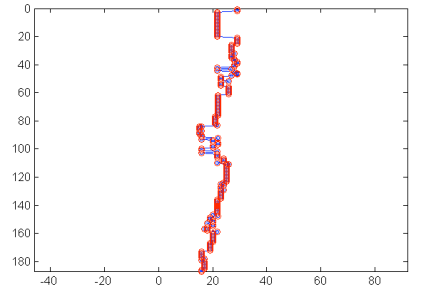

Fig. 8.a. Before eliminating the missing windows

Fig. 8.b. After eliminating the missing windows

VI. SECOND EXPERIMENT

The second experiment aims to determine the position of pipeline by using only one line of image. We have to therefore transform data from one dimensional space to two dimensional space so that we can use the GLCM technique. This technique will be presented below.

VI.1. Line Transformation

In general the sonar image is composed of numerous lines of returning signal reflected off of seabed. As already said, we cannot directly apply the GLCM as data from it is only one dimensional space. This problem can be solved by using the line transformation method, which permits this data to be transformed into two dimensional space, or from a line to an image of a line (i.e. the transformed line image). This method is set out step by step below.

Firstly the dimension of the image is given, the number of lines are equivalent to $Max(I)+3$ and the columns are equal to $N*3$. Then each point is put on the image, for instance, if I(n) is of 0, it will be put in the second line of the image, and I(n) = 1 must be in the third line of the image, etc (see fig. 9).
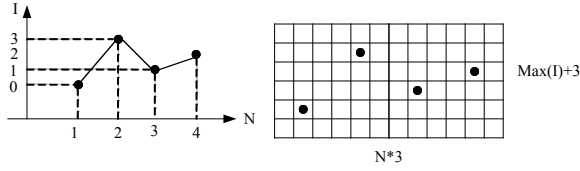

Fig. 9. The data points present the amplitude of the signal vs. time.

Each column is separated by a distance of 3 pixels so that we can put the 8 pixels around each point by the same value (Fig. 10).
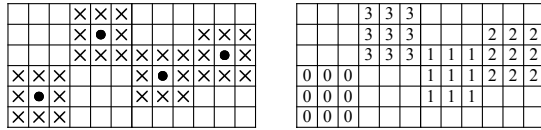

Fig. 10.

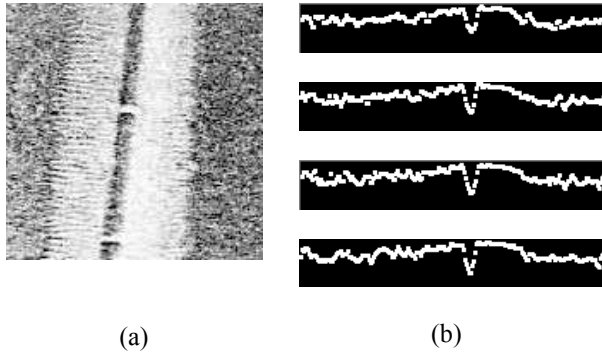Finally the transformed line images are shown in figure below.

(a)   (b)

Fig. 11. Some transformed line images (right) extracted from the pipeline image (left)

VI.2. Methods

In the training phase we have a set of transformed line images of pipeline image, defined in standard size (Fig. 12). During the training-phase, the training-window is firstly trained by SOM network in order to obtain a comparative model of pipeline as the first experiment.

Fig. 12. The structure of training-window

During the testing phase, the matrix model of sliding-window, I^k, is calculated from the arbitrary transformed line images by means of the trained SOM model, discussed in the previous section. After the matrix model of I^k, is compared with the one of I_w. To find the object location, the penalty value E^k for every sliding-window from left to right is calculated. Finally the pipeline is located on the sliding-window k which has the penalty value the nearest zero.

Fig. 13. Real transformed line image of pipeline

VI.3. Experimental results

In the second experiment the transformed line image of 30 lines of pipeline image is selected from the testing image to create the matrix model or the comparative model of pipeline (see Fig. 14).
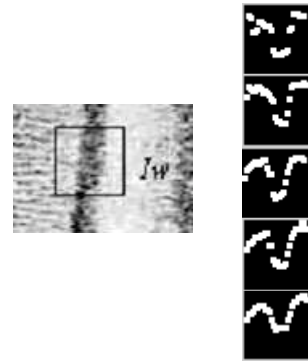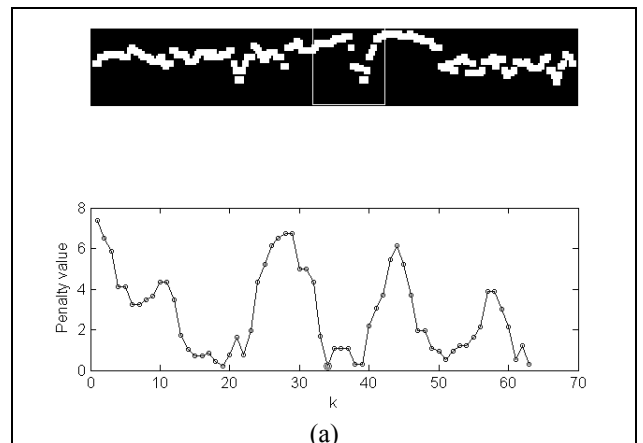

Fig. 14. The pipeline image model (left) and some transformed line images of pipeline (right)

An area of 2×2 units of SOM grid was selected for the best result from a number of experiments. The experiments tested it with five consecutive lines of sonar image. The results are shown below:
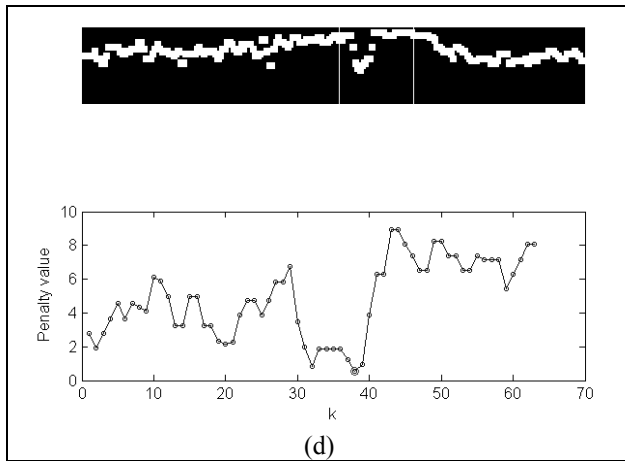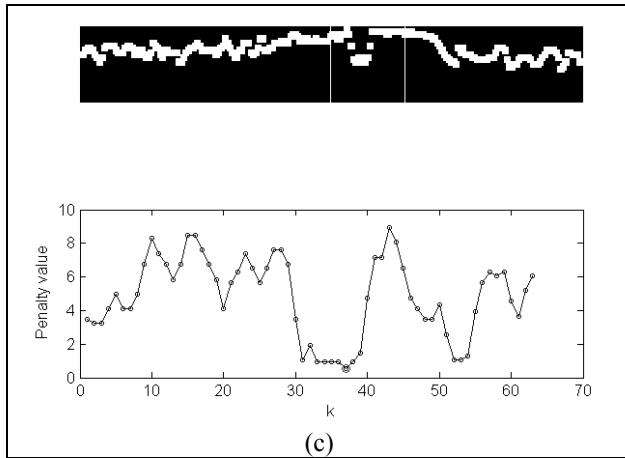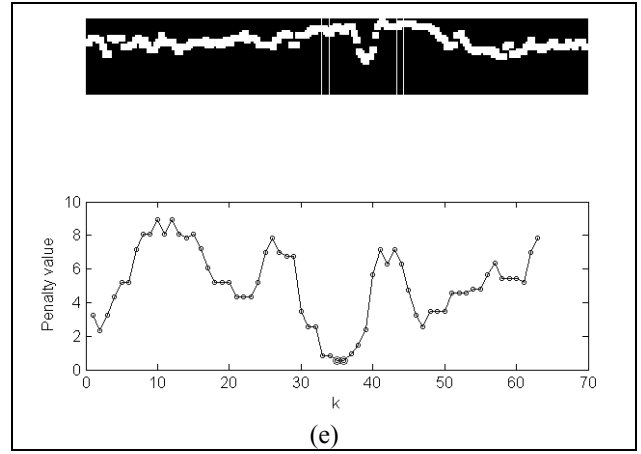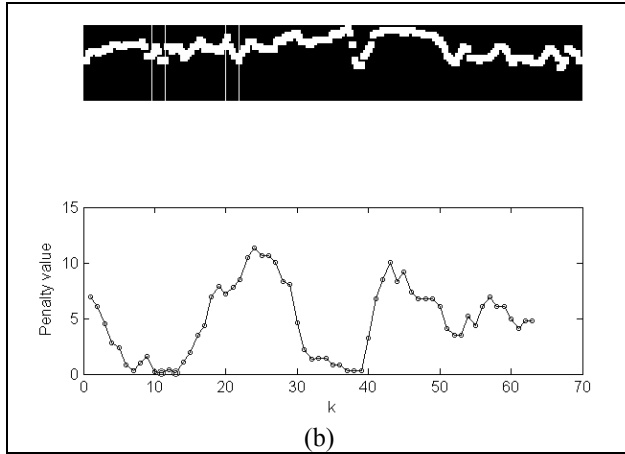

(a)

(b)

(e)

Fig. 15.(a-e). Each picture shows the pipeline position indicated by the best-matching window marked in the transformed line images above.

(c)

(d)

VII. CONCLUSIONS

This paper puts forward a new approach for detecting the position of pipeline using the comparative model based on the SOM. The objective is to find the position of pipeline in real time. Though the process described has yet to be tested in a real time experiment, our work has shown that the SOM can perform well in real word sonar images.

Both of experiments are designed for a standard size of sliding-window. In addition, we have only considered the pipeline position in vertical line. The main advantage of this technique is that it is simple and robust. However it has a high computational time due to co-occurrence matrix calculation.

In our future work, we hope to improve the techniques and also the penalty function to identify more precisely the best-matching window.

REFERENCES

[1] T. Kohonen. "Self-Organizing Maps", 3rd edition. Springer-Verlag, Berlin, 2001.

[2] Juha Vesanto and Esa Alhoniemi. "Clustering of the Self-Organizing Map". IEEE Transaction on Neural Network 11 (2000) 586-600.

[3] Juha Vesanto. "SOM-Based Data Visualization Methods". Laboratory of Computer and Information Science, Helsinki University of Technology, Finland, November 1999.

[4] Arthur Flexer. "On the use of Self-Organizing Maps for clustering and visualization". Intelligent Data Analysis 5 (2001) 373-384, IOS Press.

[5] Vladan M Babovic and Gopakumar R. "Seabed Recognition using neural networks". D2K Technical Report, 0399-1, March, 1999.

[6] K.C. Yao, M. Mignotte, C. Collet, P. Galerne and G. Burel. "Unsupervised segmentation using a Self-

Organizing Map and a noise model estimation in sonar imagery". Pattern Recognition 33 (2000) 1575-1584.

[7] N.Pican, E. Trucco, M. Ross, D.M. Lane, Y. Petillot and I. Tena Ruiz. "Texture Analysis for Seabed Classification: Co-occurrence Matrices vs Self-Organizing Maps". Ocean Systems Laboratory, Department of Computing and Electrical Engineering, Heriot-Watt University, Riccarton, Edinburgh, EH14 4AS, UK.

[8] Ross F. Walker, Paul Jackway and I.D. Longstaff. "Improving Co-occurrence Matrix Feature Discrimination". Proceedings of DICTA'95, The 3rd Conference on Digital Image Computing: Techniques and Applications, 6-8th December, 1995, pp643-648.