วิศวกรรมสาร มข. ปีที่ 36 ฉบับที่ 2 (173-183) เมษายน – มิถุนายน 2552 บทความวิจัย
KKU Engineering Journal Vol.36 No.2 (173-183) April – June 2009

Pipeline Localization Using Unsupervised Neural Network Technique

Amornrit Puttipipatkajorn*

ABSTRACT

The main purpose of this paper is to identify and detect the position of pipeline on side scan sonar image. This work is performed in two steps. The first one is to split an image into regions of uniform texture using the Gray Level Run Length Matrix (GLRLM) which is widely used in texture segmentation application. The last one addresses the unsupervised learning method based on the Artificial Neural Networks (Self-Organizing Map or SOM). The result of SOM network based on data histogram visualization is determined as the comparative model of object of interest. To increase the performance of our method, we propose a penalty function used for estimating the position of pipeline. After a brief review of both techniques (GLCM and SOM) we present our method and some results from several experiments on the real world data set.

Keywords: Unsupervised neural network, Self-Organizing Map, Side scans sonar images, Gray Level Run Length Matrix

* Lecturer, Department of Computer Engineering, Faculty of Engineering at Kamphaengsaen, Kasetsart University, Kamphaengsaen Campus, Nakhon Pathom, 73140 E-mail: fengarp@ku.ac.th

Introduction

Besides the human interpretation, the high-resolution side scan sonar seems to be the advanced tool for analyzing the seafloor. Three kinds of regions can be visualized: echo, shadow and sea bottom reverberation. The echo information is caused by the reflection of acoustic wave from the object while the shadow area corresponds to a lack of acoustic reverberation behind the object and the remaining is the sea-bottom reverberation area. The only available type of sonar image is the gray level of the pixels corresponding to the acoustic reflectance. Unfortunately, we cannot recognize and classify the objects base on single feature; consequently several methods are proposed in order to obtain more in the aspect of second-order information. Many studies have done about the performance of various families of computational methods for texture feature extraction, for instances, the 2-dimensions of FFT, the Gray Level Co-occurrence Matrix (GLCM) (Ross *et al.*, 1995), Gray Level Run Length Matrix (GLRLM) (Chu *et al.*, 1990) and etc. In addition, a comparative study from several methods shows that the GLRL is an excellent statistical tool for extracting second-order texture information from image.

The next section of this paper concerns with clustering algorithms based on the Self-Organizing Map (SOM) (Kohonen, 2001). This method is frequently employed in various applications such as data mining (Juha *et al.*, 2000), image segmentation (Kenneth *et al.*, 1994) and also seabed recognition system (Yao *et al.*, 2000). The SOM is a neural network algorithm based on unsupervised learning. It is an efficient tool for visualizing the multi-dimensional numerical data. It represents high-dimensional data into low-dimensional grid in 1D or 2D. Several methods to visualize clustering based on the SOM can be found in the literature. The most widely used methods for visualizing the cluster structure of the SOM are distance matrix technique (Arthur *et al.*, 2001), especially the unified distance matrix or U-matrix. Unlike U-matrix, data histogram visualization method is to show the number of the best matching unit in each map unit. This information can be utilized in clustering the SOM by using zero-hit units to indicate boundary of cluster (Juha *et al.*, 1999). Our experiments aim to determine a comparative model of object of interest based on the data histogram visualization method. To find the pipeline location, the penalty function is formed and the pipeline is found while its penalty value approach to zero.

Seabed Recognition and Detection System

The basic seabed recognition and detection system is composed principally of the training and testing process (see Figure 1). The aim of the training process is to evaluate labeled patterns or pipeline image in order to obtain a model (i.e. the 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 formed. In the second of these processes, a testing phase evaluates the model of the arbitrary pipeline images using the SOM network of the previous process. During this step, the comparative model represented by training-window I_w and arbitrary image symbolized by sliding window I^k are compared using the penalty function to estimate the position of pipeline.

The diagram below shows the procedure of the seabed recognition system which 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. The following will be described in the next section: the Gray Level Run Length Matrix (GLRLM) and the Self-Organizing Map.

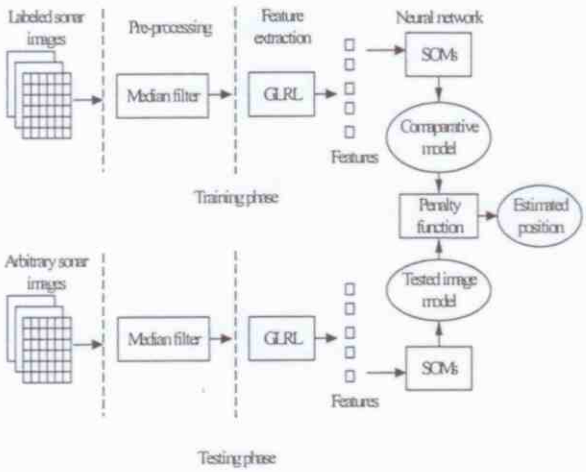


Figure 1 *Architecture of seabed recognition and detection system*

Feature extraction

A good texture feature extraction method should be able to identify the major groups of seabed patterns based on their prominent features, and so give the best information for texture classification. The Gray Level Run Length Matrix (GLRLM) is based on the estimation of the statistical distribution of observed combinations of intensities at specified positions relative to each other in the image. According to the number of intensity points (pixels) in each combination, statistical features are classified into first-order, second-order and high-order statistics.

Gray Level Run Length Matrix (Galloway, 1975)

The Gray Level Run Length Matrix is a two-dimensional matrix in which each element $p(i, j|\theta)$ gives the total number of occurrences of runs of length j at gray level i, in a given direction θ. The number of gray levels in the image is often reduced by re-quantization prior to the accumulation of the Gray Level Run Length Matrix. Even visually, quantization into 16 gray levels is often sufficient for discrimination or segmentation of textures. Using few levels is equivalent to viewing the image on a coarse scale, where as more levels give an image with more detail. However, the performance of a given GLRL-based feature, as well as the ranking of the features, may depend on the number of gray levels used. The GLRLM allows us to derive four matrices for each given direction as indicated in Figure 2.

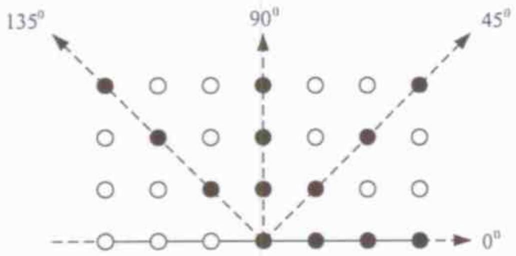


Figure 2 *Geometry for measurement of GLRLM for 4 angles*

Statistical texture features

A number of scalar texture features can be computed from the Gray Level Run Length Matrix which will be used in our experiments (see the equations as follows): Let $p(i, j|\theta)$ is the coordinate $(i\ ,\ j)$ of element of the run length matrix for a direction θ, G is the number of gray levels, R is the longest run and n is the number of pixels in the image.

Short Run Emphasis (SRE):

$$F_1 = \sum_{i=1}^{G}\sum_{j=1}^{R}\frac{p(i,j\mid\theta)}{j^2}\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (1)$$

Long Run Emphasis (LRE):

$$F_2 = \sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta)\cdot j^2\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (2)$$

Gray Level Non-uniformity (GLN):

$$F_3 = \sum_{i=1}^{G}\left(\sum_{j=1}^{R}p(i,j\mid\theta)\right)^2\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (3)$$

Run Length Non-uniformity (RLN):

$$F_4 = \sum_{j=1}^{R}\left(\sum_{i=1}^{G}p(i,j\mid\theta)\right)^2\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (4)$$

Run Percentage (RP):

$$F_5 = \frac{1}{n}\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta). \qquad (5)$$

Low Gray-Level Run Emphasis (LGRE):

$$F_6 = \sum_{i=1}^{G}\sum_{j=1}^{R}\frac{p(i,j\mid\theta)}{i^2}\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (6)$$

Low Gray-Level Run Emphasis (HGRE):

$$F_7 = \sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta)\cdot i^2\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (7)$$

Short Run Low Gray-Level Emphasis (SRLGE):

$$F_8 = \sum_{i=1}^{G}\sum_{j=1}^{R}\frac{p(i,j\mid\theta)}{i^2\cdot j^2}\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (8)$$

Short Run High Gray-Level Emphasis (SRHGE):

$$F_9 = \sum_{i=1}^{G}\sum_{j=1}^{R}\frac{p(i,j\mid\theta)\cdot i^2}{j^2}\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (9)$$

Long Run Low Gray-Level Emphasis (LRLGE):

$$F_{10} = \sum_{i=1}^{G}\sum_{j=1}^{R}\frac{p(i,j\mid\theta)\cdot j^2}{i^2}\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (10)$$

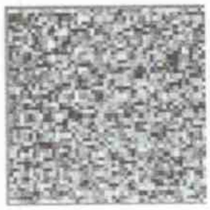
Long Run High Gray-Level Emphasis (LRHGE):

$$F_{11} = \sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta)\cdot i^2\cdot j^2\bigg/\sum_{i=1}^{G}\sum_{j=1}^{R}p(i,j\mid\theta) \qquad (11)$$

A set of statistical texture features derived from Gray-level Run Length Matrix is used as a texture feature vector $F = \{F_1, F_2, \ldots, F_{11}\}$. In this case, each element of vector contains information of image texture calculating from the functions above. The figure 3 shows the images are extracted from real sonar image used in experiments with 11 statistical features.

(a) Original image

(b) SRE image



(c) LRE image

(d) GLN image

(e) RLN image

(f) RP image

(g) LGRE image (h) HGRE image (i) SRLGE image

(j) SRHGE image (k) LRLGE image (l) LRHGE image

Figure 3 *Images of statistical texture features*

Self-Organizing Map (SOM)

The Self-Organizing Map or SOM (Kohonen, 2001) is used for pattern recognition and classification task. It belongs to the category of unsupervised learning neural networks. The SOM have only two layers of neurons, an input layer and a competitive layer (see Figure 4). 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 in the aspect of various models of connection, such as squared neighboring connection.

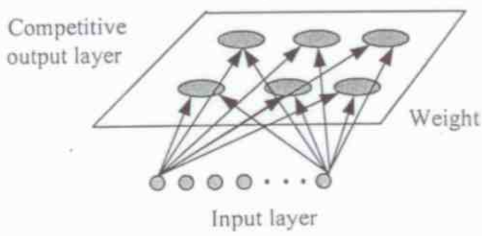


Figure 4 *Architecture of Self-organizing map*

The model of SOM used in our experiment is a two-dimensional array of *k* nodes. Each neuron *k* is represented by an *n*-dimensional vector m_k = [$m_{k1}, \ldots, m_{kn}$] 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:

$$\|x - m_c\| = \min_k \{\|x - m_k\|\} \tag{12}$$

And then, the vector m_c and its neighbors on the grid are updated by closing to the sample vector:

$$m_k = m_k + \alpha(t) h_{ck}(t)(x - m_k) \tag{13}$$

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

$$h_{ck}(t) = \exp\left(-\frac{\|r_c - r_k\|^2}{2\sigma^2(t)}\right) \tag{14}$$

and

$$\alpha(t) = \frac{\alpha_0}{1 + 100t/T} \tag{15}$$

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.

Methods

Training process

The first task of training process is the selection of a pipeline image as an image model. We call it as the training-window. The training-window is composed of a vertical single pipeline and a region of background (see Figure 5). In our experiment, a size of 30×30 pixels of training-window is used. Before the beginning of training process, the prominent features of training-window will be extracted by GLRL method and the result will be prepared for training.

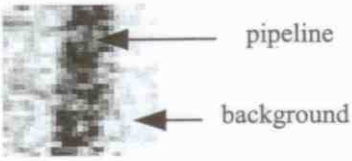
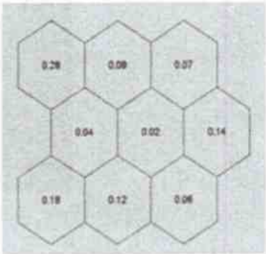


Figure 5 *The structure of training window*

During the training phase, the training-window is trained by SOM network to find the best matching unit. After the end of process, we obtain the SOM network which presents amount of the best matching units occurring in the training process. At this time, we call it as the comparative model. Basically the comparative model, in case of two-dimensional map units, is the matrix of probability density, denoted I_w. The following figure shows the comparative model of 3×3 map unit of SOM network.



$$\begin{bmatrix} 0.28 & 0.09 & 0.07 \\ 0.04 & 0.02 & 0.14 \\ 0.18 & 0.12 & 0.06 \end{bmatrix}$$

Figure 6 The comparative model of 3×3 map unit

Testing process

In the testing process, the comparative model I_w will be used to compared with the matrix model of sliding-window I^* of arbitrary image of pipeline. In our definition, the sliding-window means any considered region of arbitrary image of pipeline, while the matrix model of sliding-window can be determined by using the same procedure as the matrix model of comparative model. In the figure 7, we found that the sliding-window moves gradually on the arbitrary image from left to right (horizontal move) and top to bottom (vertical move) in order to determine the position of pipeline.

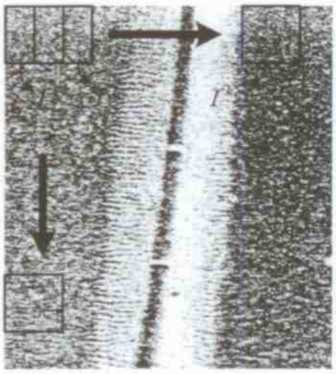


Figure 7 *Sliding-window on real pipeline image*

To find the position of pipeline on the arbitrary image, we propose the penalty function shown in the equation 16. It will be used for computing the similarity of the comparative model and the matrix model of current sliding-window. This function was developed from the histogram visualization of SOM network which it cannot identify the position of object directly. The penalty function was created from the difference of two matrices (matrix of sliding-window and matrix of comparative model). The computational result of function indicated the position according with the position of the best-matching window which gives the value of E is the nearest zero.

$$E^k(i,j)=\sum_{i=1}^{n}\sum_{j=1}^{m} I_w(i,j)\otimes\{I^k(i,j)-I_w(i,j)\} \tag{16}$$

Where I_w represents the comparative model of object of interest I^k is the matrix model of sliding-window, k denotes the index of current sliding-window and the symbol $\otimes$ is product of matrix in term by term. The figure 8 presents the example of the best matching window found on any position which gives the penalty value approaches to zero for one horizontal movement of sliding-window from left to right.

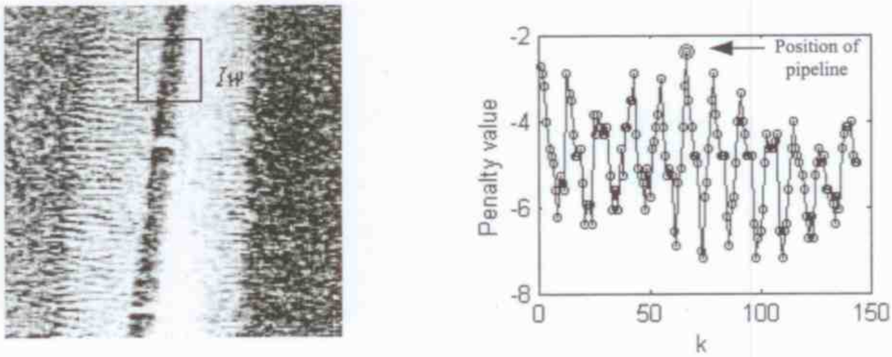
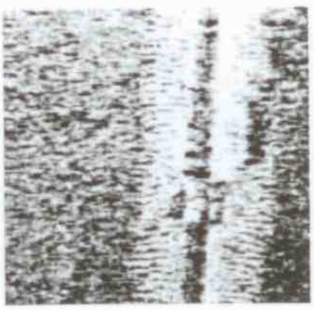


Figure 8 *The penalty value indicates the position of pipeline*

Experimental Results

We experimented on the different arbitrary images of pipeline. They contain the gray scale of 150×150 pixels. For the comparative model, it was generated from the pipeline image of 30×30 pixels on the 3×3 units of SOM network (the best result from the experiments).



(a)

(b)

Figure 9 *(a) the image of 150 × 150 pixels (b) the image of 30 × 30 pixels*

The results will be presented in 2 parts: the figure (a) and (b) show the position of the best matching window and the figure (c) and (d) present the approximated line of pipeline using the Least Median Square Regression technique based on the results from figure (a) and (b).

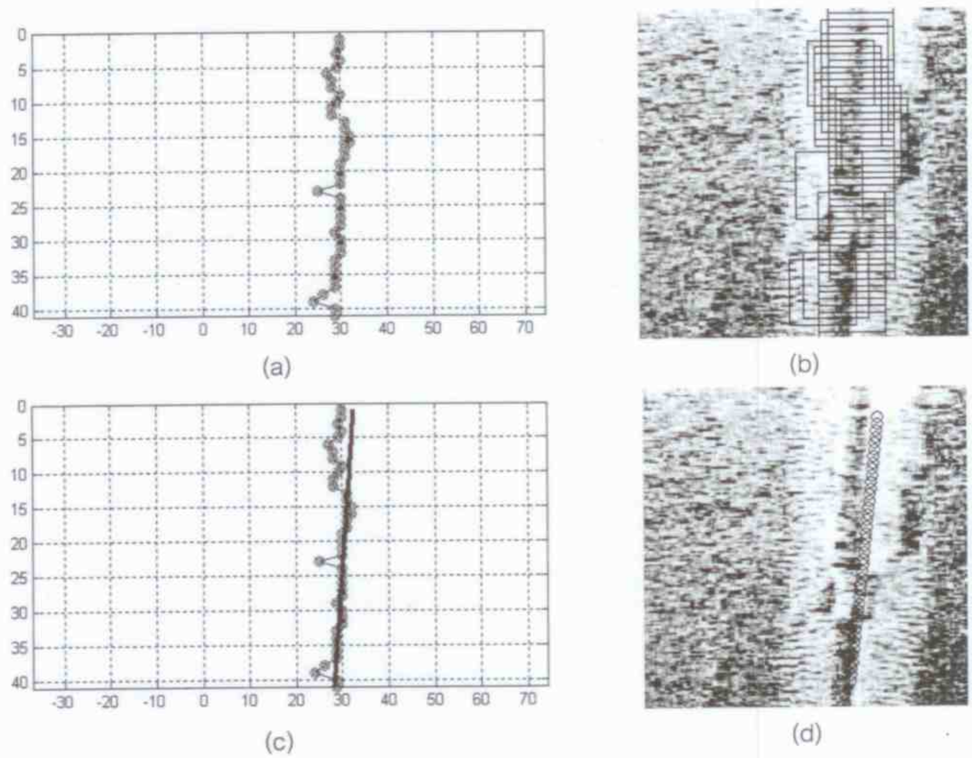


Figure 10 *The results of experiment no.1.*

For the first test, we found that our method can detect the pipeline correctly, even if the bottom-part of pipeline covered by the undesired object. For the figure below presented the relationship between the actual positions and the estimated positions along the horizontal direction in term of the Root Mean Square Error.

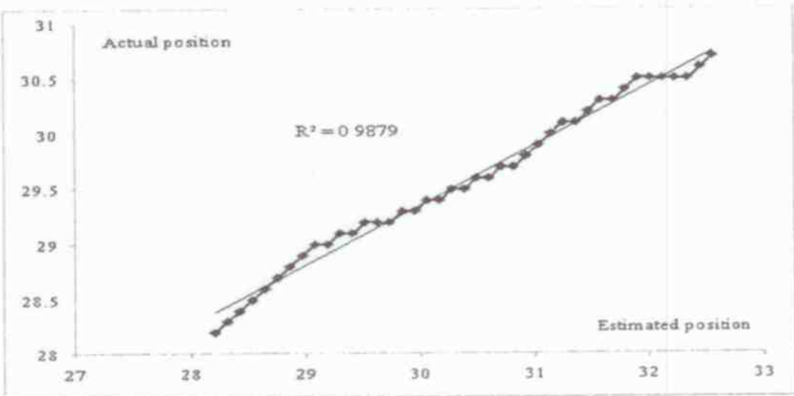


Figure 11 *The Root Mean Square Error of experiment no.1.*

For the second experiment we found that the tested image was not clear due to the bad environment of seafloor shown by white color on the background. It caused the oscillating of estimated positions around the pipeline (see figure 12). Thus the Root Mean Square Error value was less than the value of first experiment (see figure 13).

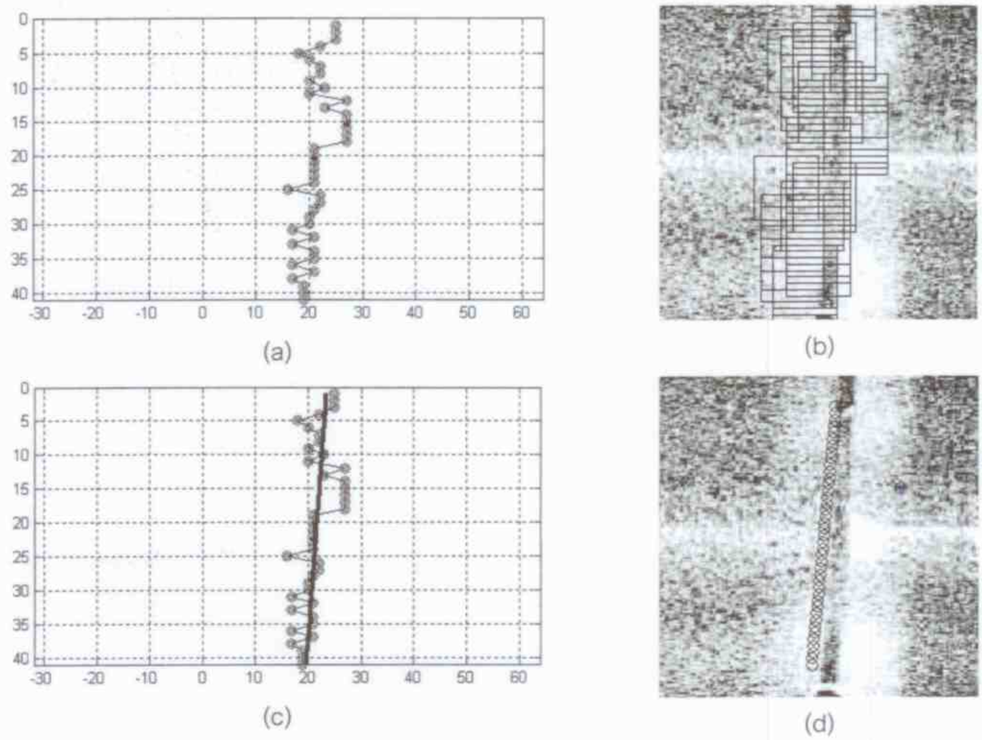


Figure 12 *The results of experiment no.2.*

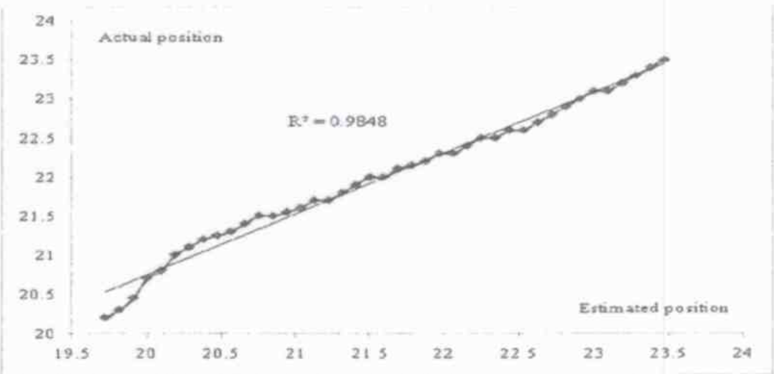


Figure 13 *The Root Mean Square Error of experiment no.2.*

For the third experiment, we found that the estimated positions were at the right location, because the pipeline image of tested image was clear comparing with the others (see figure 14) and we obtained the good relationship of the actual position and the estimated position comparing with the first two experiments which indicated by the Root Mean Square Error in figure 15.

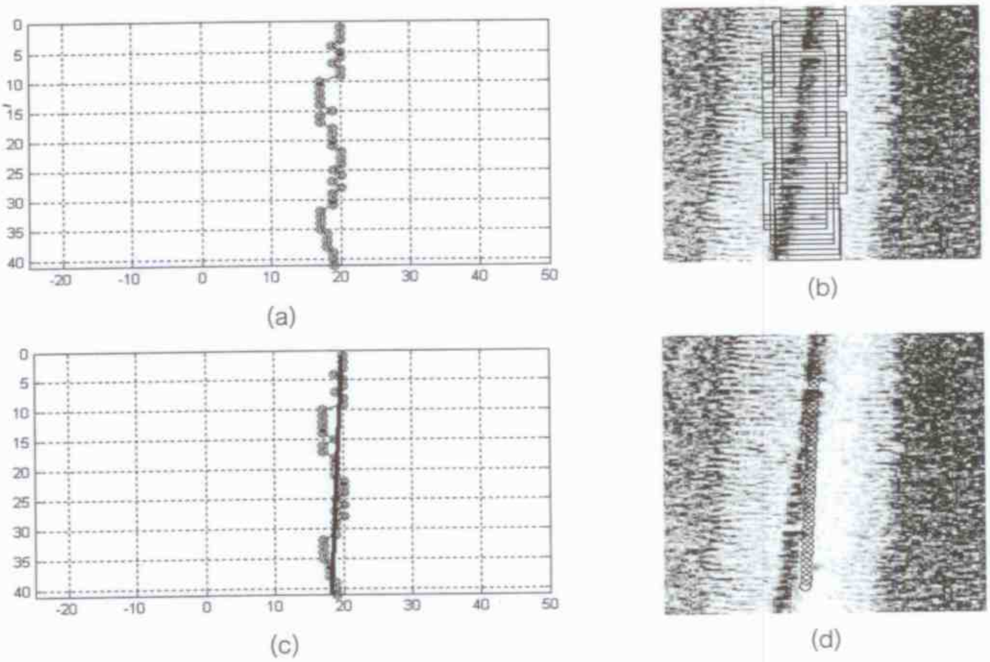


Figure 14 *The results of experiment no.3.*

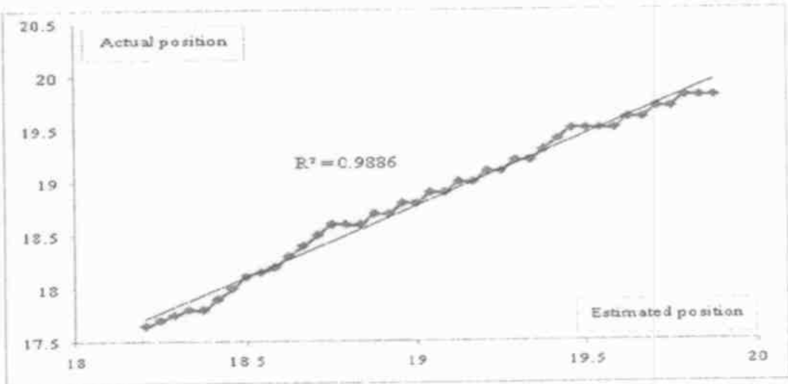


Figure 15 *The Root Mean Square Error of experiment no.3.*

Conclusions

This paper proposes a technique for approaching to pipeline localization using a comparative model based on the SOM. The experiments show that the SOM performs well in finding the position of pipeline in real world sonar image. The advantage of this technique is simple and robust. However this method has the limitation of computational time due to Gray Level Run Length Matrix calculation. In addition, this technique encountered the rotation problem of pipeline, because size of sliding-window is fixed by its size and direction. For future work, we will attempt to improve this technique for determining the position of pipeline in different directions and also penalty function to identify more precisely the best-matching window.

References

Ross F. Walker, Paul Jackway and I.D Longstaff. (1995). Improving Co-occurrence Matrix Feature Discrimination. Proceedings of DICTA'95, The 3rd Conference on Digital Image Computing: Techniques and Applications, 6-8th December: 643-648.

A. Chu, C.M. Sehgal and J.F. Greenleaf. (1990). Use of gray value distribution of run lengths for texture analysis. Pattern Recognition Letters Vol. 11: 415-420, 1990.

T. Kohonen. (2001). Self-Organizing Maps. 3rd edition. Springer-Verlag Berlin.

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

W. Kenneth Stewart, Min Jiang and Martin Marra. (1994). A Neural Network Approach to Classification of Side scan Sonar Imagery from a Mid-ocean Ridge Area. IEEE journal of oceanic engineering VOL. 19, NO. 2.

K.C. Yao, M. Mignotte, C. Collet, P. Galerne and G. Burel. (2000). Unsupervised segmentation Using a Self-Organizing Map and noise model estimation in sonar imagery. Pattern Recognition 33: 1575-1584.

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

Juha Vesanto. (1999). SOM-Based Data Visualization Methods. Helsinki University of Technology, Finland.

RM.M Galloway. (1975). Texture Analysis Using Gray Level Run Lengths. Computer and Image Processing Vol. 4: 172-179.