



Automated Textural Image Analysis of Seabed Backscatter Mosaics: A Comparison of Four Methods

*R.D. Müller and S. Eagles

School of Geosciences and University of Sydney Institute of Marine Sciences,
The University of Sydney, Sydney, NSW 2006, Australia

*Corresponding author, E-mail: dietmar@geosci.usyd.edu.au

P. Hogarth

GeoAcoustics Limited, Shuttleworth Close, Gapton Hall Industrial Estate,
Great Yarmouth, Norfolk, England, NR31 ONQ, UK

M. Hughes

School of Geosciences and University of Sydney Institute of Marine Sciences,
The University of Sydney, Sydney, NSW 2006, Australia

Müller, R.D., Eagles, S., Hogarth, P., and Hughes, M., 2007, Automated textural image analysis of seabed backscatter mosaics: A comparison of four methods, in Todd, B.J., and Greene, H.G., eds., Mapping the Seafloor for Habitat Characterization: Geological Association of Canada, Special Paper 47, p. ??-??.

Abstract

Four seabed backscatter image classification methods that differ substantially in their approach are compared, using a dataset from Sydney Harbour. The first method is based on GeoAcoustic's commercial GeoTexture™ software. GeoTexture relies on the user to extract features from the image to train the system to recognize textures based on a statistical method, before classifying the entire image. GeoTexture achieved classification accuracies for the sand, gravel and mud classes of 47%, 67% and 77%, respectively. The second method characterizes textures in the wavelet-domain utilizing Hidden Markov Models (HMM) and uses a Bayesian method for image segmentation. In this approach, sub-images, 512 x 512 pixels large, are segmented into three classes of sediment. The best supervised classification success for this method was relatively poor with 100% for gravel, but only 29% for sand and 36% for mud. The third and fourth methods use a neural network-based approach where two different techniques are compared for feature extraction: (I) a space-domain method based on grey-level run-length features, spatial grey-level dependence matrices and grey-level difference vectors in four directions, and (II) the grey-level co-occurrence iteration algorithm (GLCIA) method, which is far superior in terms of computational speed, but relies on a smaller number of feature vectors. Method I provides accuracies of 92% and 89% for the gravel class, compared with accuracies for the sand, gravel and mud classes of 88%, 77% and 78%, respectively, for method II.

Both the GeoTexture and neural network-based approaches were found to be superior to the HMM approach, which is not mature enough for application to seabed images, at this time.

Résumé

Quatre méthodes de classification d'images de rétrodiffusion du fond marin qui diffèrent substantiellement dans leur approche sont comparées en utilisant un ensemble de données du port de Sydney. La première méthode repose sur l'utilisation du logiciel commercial GeoTexture^{MD} de GeoAcoustic. GeoTexture exige de l'utilisateur qu'il extrait les caractéristiques de l'image dans le but d'entraîner le système à reconnaître les textures à partir d'une méthode statistique, avant de classifier l'image entière. GeoTexture a permis d'obtenir des résultats de concordance de 47%, 67% et 77% dans la classification de sable, de gravier et de boue, respectivement. La deuxième méthode permet de caractériser les textures du domaine des ondelettes par la segmentation de l'image à l'aide des modèles de Markov cachés et de la méthode bayésienne. Dans cette méthode, les sous-images, de 512 x 512 pixels, sont segmentées en trois classes de sédiments. Le meilleur degré de réussite de classification selon cette méthode a été relativement faible avec des résultats de 100% pour le gravier, mais de seulement 29% pour le sable et de 36% pour la boue. La troisième et la quatrième méthodes font appel à l'approche du réseau de neurones artificiel dans laquelle deux techniques différentes sont comparées pour extraire les caractéristiques: (I) la méthode du domaine spatial fondée sur les caractéristiques d'exécution des niveaux de gris, les matrices de dépendance spatiale des niveaux de gris et les vecteurs de la différence du niveau de gris dans quatre directions, et (II) l'algorithme d'itération de cooccurrences des niveaux de gris, qui est de loin supérieur en termes de vitesse computationnelle, mais qui repose sur un plus petit nombre de vecteurs de caractéristiques. La méthode I donne des résultats de concordance de 92% et de 89% pour la classe du gravier, comparé à des résultats de concordance de 88%, de 77% et de 78%, pour les classes de sable, de gravier et de boue, respectivement, à l'aide de la méthode II.

La méthode GeoTexture et la méthode du réseau de neurones artificiel se sont montrées toutes deux supérieures à la méthode des modèles de Markov cachés, qui n'est pas suffisamment au point à ce jour pour être appliquée aux images du fond marin images.

INTRODUCTION

The sustainable management of marine resources has highlighted the need for an efficient and accurate method, to map and study the seabed over wide areas and at all water depths. Automatic processing and interpretation of seafloor data is driven by the need to understand the marine environment scientifically, commercially, and for marine planning and management purposes. Knowledge of seafloor physical properties is imperative for hazard assessment of cable routes, placement of oil platforms, pipelines and wellheads, and for laying fibre-optic cables. Further, it can help in understanding the processes of environmental change, knowledge and understanding of submarine morphology, sedimentary processes and seabed instability, and the nature of marine habitats.

Automated texture classification algorithms can be utilized to provide a quantitative measure of texture contrast, which assists in the interpretation of seafloor composition. The texture of the seafloor is the primary focus of many image-processing applications, necessitating prior understanding of texture properties. Programs vary widely in their approach to textural analysis, computational efficiency and in classification accuracy.

The objective of this paper is to compare four seabed texture classification methods in terms of execution time and classification accuracy, and to present ground-truth maps of a selected shallow-water survey area in Sydney Harbour.

Texture is presently an intuitive notion, and a standard definition of it is yet to be formulated. Texture can be defined by a significant variation in intensity levels between nearby pixels and a homogeneous property at some spatial scale larger than the resolution of the image, *i.e.*, textures are regarded as functions of the spatial variation in pixel intensities (Clarke, 1998; Fan and Xia, 2003).

Textures can be categorized into statistical and structural textures (Fan and Xia, 2003). The former views texture as a sample of a 2-D stochastic process, described by its statistical parameters, whereas the latter is usually composed of a primitive pattern repeated throughout the texture. The surface of the seafloor, encompassing various sediment types, can be described by a series of textures, and texture variations. Changes in surficial sediment composition and fine-scale surface roughness can be related to textural image variations.

Digital texture analysis aims to differentiate classes with similar textural characteristics. Digital textural image classification allows enhancement of the visibility of objects, to differentiate between objects that are not necessarily seen by a human interpreter, and it is able to process a large quantity of data. This requires the classification of a pattern using its textural characteristics. The difficulty lies in the extraction of textural features that give the greatest information pertaining to each texture. To date, most methods of classification have been grouped as statistical methods or structural methods (Fan and Xia, 2003). The former method does not assume any order, but measures the variations of local or global texture, whereas structural methods assume an underlying order and attempt to model textures mathematically. Classification involves the identification of the type of a given uniform region, whereas segmentation attempts to produce a classification map of the input image where each homogeneous textured region is identified by means of localized texture boundaries (Fan and Xia, 2003).

Commonly, first-order statistics are used to describe the acoustic energy reflected back from the seafloor and its representation as a grey-scale image. However, second-order statistics are necessary to quantify the spatial relationships of grey levels in the image (Pace and Dyer, 1979; Reed and Hussong, 1989). More recent methods have seen the application of artificial neural net-

work classifiers to several traditional statistical texture features to classify seafloor types (Müller *et al.*, 1997; Stewart *et al.*, 1994). Structural methods endeavour to characterize textures in terms of their primitives (basic elements of texture), and the placement rule governing this arrangement within the image (Blondel and Murton, 1997). To date, these methods have been only rarely applied to seafloor classification. However, an increasing number of texture classification methods are being developed with a multichannel approach, driven by advances in wavelet theory and applications (Clarke and Hamilton, 2002; Henke-Reed and Cheng, 1993; Stewart *et al.*, 1994). Tree-structured wavelet transform algorithms have been developed by Chang and Kuo (1993), and more recent studies by Laine and Fan (1993) display texture features that have been measured directly through the use of the standard wavelet and the wavelet packet energy (Tang and Stewart, 2000).

Data Acquisition

The data for this study were acquired with the GeoSwath 250 kHz system, a shallow-water, wide-swath bathymetric system from GeoAcoustics Limited. GeoSwath uses a phase measurement system that calculates the angle of the backscatter return from differential time measurements. It has the ability to produce raw and processed data at the same time because most of the data processing tasks can be undertaken in parallel with data acquisition. The survey system was installed on the DSTO vessel 440, supplied by the Australian

Defence and Science Technology Organisation (DSTO). The data were collected during a survey of Sydney Harbour over an eight-day period beginning on the 5th March 2003, in water depths ranging from a few metres to almost 50 m (Figure 1). The survey provided extensive backscatter data representing seafloor composition from several regions within Sydney Harbour. The speed of the vessel was limited to between 4 and 6 kts to provide a compromise between the survey cost and accuracy. Throughout the survey, numerous sound velocity profiles (SVP) were taken. To improve accuracy and maintain a high number of points per bin, the maximum swath width can be restricted. For this survey, the system ping rate was set to allow slant ranges between 40 and 60 m per side.

The data were processed and mosaiced using GeoAcoustics Limited's GeoSwath software. The data were binned at either 1 m or 0.5 m resolution depending on the tradeoff between complete coverage, file size and image resolution for a given area. To initialize processing, it is necessary to accurately locate the depth of the seabed near nadir. The location of the depth to the seabed is performed in GeoTexture by one of two methods, the maximum F method, or the threshold method. The maximum F method locates the position at which the sum of the squares of the data beyond the position, divided by the sum of the squares before the position, is a maximum. This method is good for detecting first events and is successful, provided the seabed is the first event. The method is compromised if there is weak back-scattering associated with, for exam-

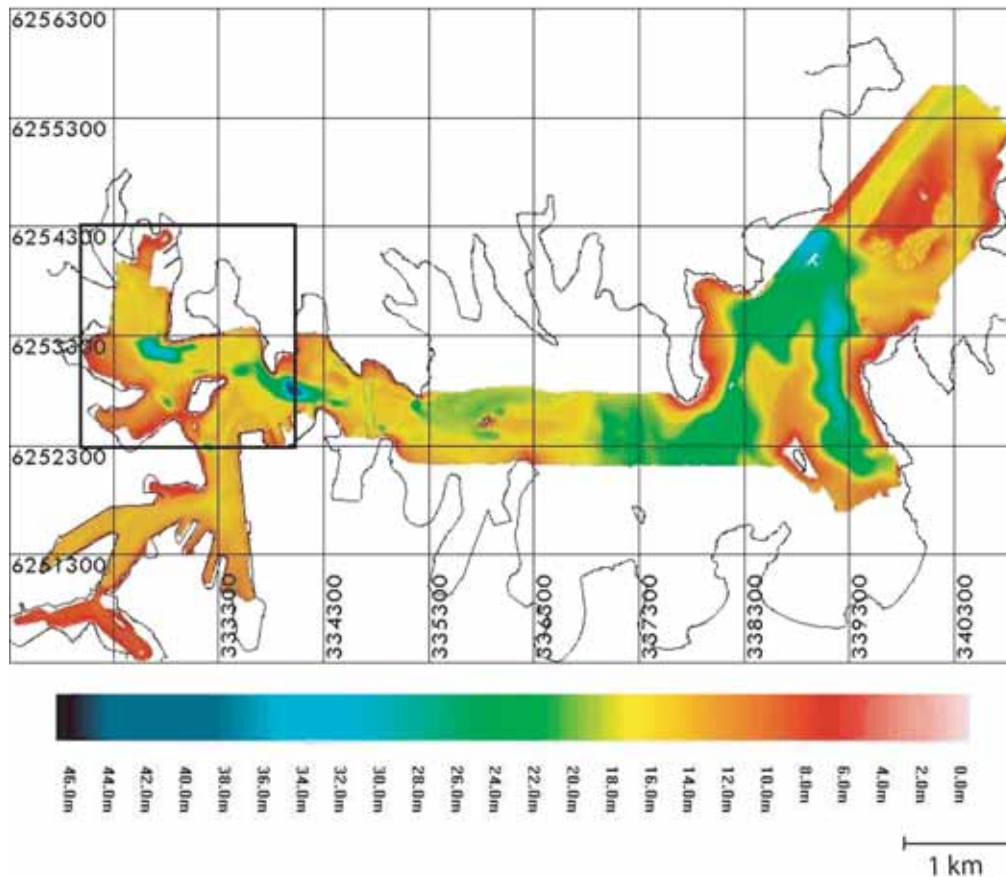


Figure 1. Colour-coded water depths in multibeam survey area, Sydney Harbour. Box indicates Balls and Blues region (see Figure 2 for details).

ple, sediment suspension in the water column. The threshold method locates the seabed by locating the first arrival to exceed the value chosen by the user. In this study, the maximum F method was chosen for all regions because there was little sediment suspension. GeoSwath jointly processes raw bathymetry data and backscatter data. This feature allows GeoTexture to account for slope and morphology of the seabed when calculating grazing angles and absolute backscatter values.

To reduce the noise in the images, the bathymetry data were smoothed using sequential filters. The data from the “Balls and Blues” area in Sydney Harbour, which will focus on for texture classification, had two filters applied. The first filter, acting on the bathymetry near the sensor, was set to 1 pixel width (0.5 m), with the second filter, acting at far range, set to 15 pixel widths (7.5 m).

Backscatter signals are attenuated differently depending upon the distance they travel. This across-track attenuation can cause errors later if digital interpretation requires the use of grey-level statistics. Thus, at all points during the processing stage, the data must be normalized to account for both the transmitting and receiving transducer beamplot and mounting plate angle, operating depth of the system and seabed grazing angle. As sequential swaths are to be joined, the amplitude must be normalized across the entire dataset. This provides good swath-to-swath correlation, ensuring that the variation in amplitude contrast with distance on the sidescan sonar swath is near negligible. This matching is particularly important for texture mapping systems using amplitude as a primary feature.

To produce a normalization function, the GeoTexture system must be trained to automatically calibrate the system for a combined beamplot function. This is achieved by feeding a beam function, which is a function of the sonar system, into GeoTexture. This requires the extraction of a beam function from a swath of relatively uniform seabed characteristics. Then, this function needs to be extrapolated to cater for all inclination angles. Two beam functions were used for the dataset, one for the Balls and Blues region, and another for the remaining mosaics.

By normalizing the pixel values across sidescan sonar records, effects from the sonar beam function and vehicle roll are suppressed, minimizing changes not due to seabed materials. After trace normalization, slant-range corrections were performed to re-map the pixels from their apparent position to the true one. To mosaic the backscatter data, slant-range corrections are essential for accurate geometry and to remove the water column. These corrections are required to account for errors in the shape and power of the returned signal from the seafloor, as a result of spreading loss and absorption (Clarke and Hamilton, 2002; Hamilton, 2001). After applying all the corrections, the data are mosaiced and saved in both “.mof” format (an internal format for GeoTexture to allow interactive texture mapping), and “.bmp” format as input for other programs.

Seafloor Samples

For supervised texture classification methods, a “ground-truth” dataset in the form of seabed samples is required. Data obtained from these samples are used to train and test texture classification methods. The total number of samples is one of the greatest restrict-

ing factors determining the accuracy of classification. Ground-truth samples are able to provide an indication of the characteristics and physical properties of the seafloor at certain locations. The most common source of data collection utilizes a grab-sampler that collects loose sedimentary material from the uppermost portion of the seabed. Samples for this study were obtained through the auSEABED database, which integrates data obtained from a multitude of seafloor surveys over time into a single database. The output contains many types of information, including physical properties and composition of the seabed texture, which can be projected into a desired map reference frame. Fifty samples, recently acquired in Sydney Harbour by the DSTO, were appended to this database for this study. The types of sediments of interest for this study were limited to mud, sand and gravel. The criterion for the grouping of sediment into these three classes was based simply on a percentage of more than 50% for any one sediment type.

Acoustic Response to Sediment Type

Backscatter intensity is influenced by a range of factors including seafloor bathymetry, seafloor roughness, the varying acoustic properties of the seafloor material and the angle of incidence of the sound wave front. The acoustic backscatter of the seafloor can allow distinctions between sediment with varying grain sizes, a principle determinant of backscatter intensity. In a mosaiced backscatter image, the strength of the backscatter return is visualized by the brightness of a particular image pixel. The acoustic reflectance of the substrate affects this reflectance, with darker pixels indicating lower reflectance and brighter (whiter) pixels representing higher reflectance. Porosity and roughness of the sediments are factors affecting signal absorption. Generally higher rates of absorption are evident in softer, fine textured soils (*i.e.*, muds and silts) resulting in low reflection. Coarser substrates, such as well-drained sandy soils, tend to scatter the sonar beam resulting in high reflectance.

Sediment Sample Extraction and Manipulation

The sediment sample locations were projected into the same reference frame as the seafloor backscatter mosaic, the Universal Transverse Mercator Projection (Zone 56; Figures 1 and 2). Once samples are positioned on the mosaic, sidescan sonar data are extracted from the location on the mosaic corresponding to ground-truth samples. The backscatter images, at these discrete locations, correspond to the average brightness or spectral reflectivity for that location. The size of the sub-area extracted is of fundamental importance for textural analysis. Areas of size 25 x 25 pixels (12.5 m²) were chosen for textural image characterization at the sediment sample locations. This sub-image size corresponds to the best compromise in terms of only containing one class of sediment, but being large enough to reflect relevant image textures.

The available data points were split into 3 categories to provide training, testing and validation of ground-truth samples. Supervised classification methods require prior information of seabed materials to perform classification. To provide a consistent approach to classification, a common dataset was used across the different texture classification methods to train, test and validate sediment classes.

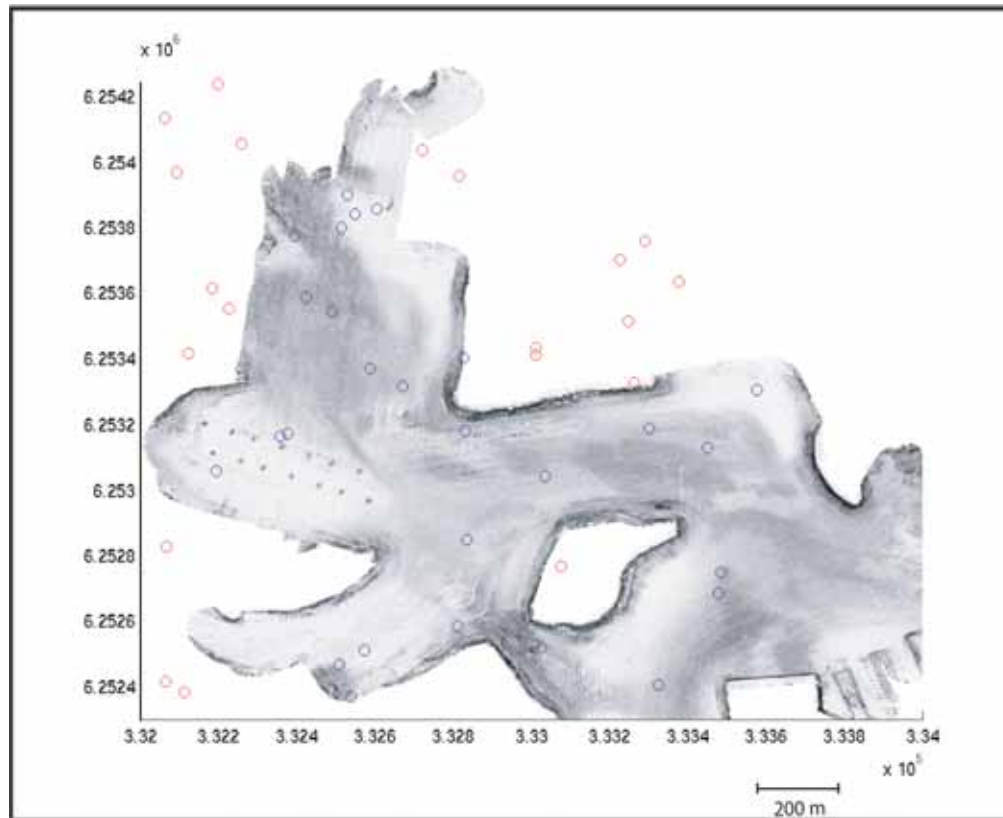


Figure 2. Mosaic of the Balls and Blues region and sediment locations overlain. Red circles = rejected sediment samples; blue circles = sediment samples used for classification.

Methods for Textural Analysis and Texture Classification

Texture Classification using GeoTexture

GeoAcoustic Limited's GeoTexture, provides a texture mapping system designed to classify seabed material. The texture mapping system is strongly reliant on user input. The GeoTexture system must be trained to recognize image textures. To quantify the description of a particular texture, 'feature' values must be extracted at the position on the image, correlating to a specific texture. The definition of this 'feature' vector is essentially what defines the texture mapping system and differentiates it from other mapping methods. GeoTexture aims to extract the maximum possible information from the strength of the backscatter signal in a sidescan record (Tamsett, 2003). GeoTexture's features are amplitude dependent, and so rely strongly on normalization of backscatter values for accurate training.

Texture Classification

The system classifies a pixel by extracting features from the image, centred on the pixel. A field is created, in feature space, to describe the features for every characterized texture. The system then computes, for the pixel being classified, the 'separations' between the centre of the field and the position in feature space. The program

will map the pixel to the characterized texture, which yields the minimum 'separation'. GeoTexture requires the user to choose between two different modes of classification.

In the first mode, for 'normal classification', a feature vector is extracted from a weighted window centrally disposed on the pixel of an image. This classification for a pixel is made on the basis of the feature vector for the pixel. In the second, for 'enhanced classification', feature vectors are extracted across a weighted window, not just at its centre. To define the textures, which are used to classify each pixel, the user must first characterize the textures and correlate the image texture with the actual seabed material. In mosaic mode, rectangles are drawn around various areas in the image to which the user ascribes specific image textures. If appropriate, more than one texture may be mapped to a single ground material. The size of the rectangle affects the classification procedure because it defines the features that represent the specified texture. As the width of the rectangle increases, the resolution of detail in the classified images decreases. However, the rectangle must still be large enough for a texture to be properly represented within it.

After classifying the image, GeoTexture produces an output matrix to represent the texture discrimination in the image. The matrix shows the probabilities of classifying each of the texture classes as belonging to its own individual class. A second matrix displays how separate each texture class is from another, thus producing a quantitative measure of texture contrast.

The user must define certain parameters to initialize classification. First, a pixel on the image will be classified as ‘unrecognized’ when its feature vector is more separate from all of the textures in the collection of characteristics than the decision threshold parameter. A second parameter that needs to be set is the feature vector definition, which defines the mode for texture characterization and classification. For textures that are mainly dependent on image signal level, the parameter should be set to ‘Aleph’. Alternatively the parameter can be set to ‘Bet’, which is strongly textural and useful for those sediments whose texture definition is dependent on aspects of texture as well as signal level. The ‘Bet’ function is recommended for sidescan sonar. Third, the classification mode must be set. GeoTexture requires the user to choose between ‘normal’ classification and ‘enhanced’ classification mode to govern the method for feature extraction.

Texture Classification using Wavelets

Introduction

Two approaches to the wavelet method of classification were explored in this study, including unsupervised classification and supervised classification. The use of Hidden Markov Models (HMM) and wavelets results in texture measures differing in nature relative to conventional texture classification methods (Clausi, 2001). Here, a new HMM, called HMT-3S is used to provide statistical texture characterization in the wavelet-domain (Fan and Xia, 2003; Song and Fan, 2002, 2003).

Wavelet Theory

Wavelets are functions that divide data into different frequency components, and then analyse each component with a resolution matched to its scale. Wavelet analysis utilizes approximating functions that are contained neatly in finite domains, with the ability to analyse signals that contain discontinuities and sharp spikes effectively. The 2-D Discrete Wavelet Transform (DWT) represents an image in terms of a set of wavelet functions (Song and Fan, 2002, 2003).

Hidden Markov Models

A Markov model is a probabilistic process over a finite set of states. This method is designed to find the probability of successive states, which may depend on the prior history. It is only the outcome, not the state, visible to an external observer and therefore states are “hidden”, hence the name Hidden Markov Model. A wavelet-domain hidden Markov tree (HMT) is a tree-structured probabilistic graph capturing statistical properties of the wavelet transforms of images. The HMT-3S is developed by grouping the three DWT sub-bands into one quad-tree structure (Fan and Xia, 2003). It has the benefits of the HMT in capturing joint statistics, combined with the ability to exploit the cross correlation across DWT sub-bands. Image features can be characterized by the statistical model HMT-3S, which approaches classification by capturing dependencies across DWT sub-bands. For natural textures, in particular structural textures, HMT-3S provides accurate classification as these textures display regular or periodic spatial structures or patterns resulting in statistical dependencies across the DWT sub-bands.

Texture Classification

The wavelet-domain HMT-3S can characterize image features in the wavelet-domain, providing the foundation to perform both supervised and unsupervised classification. Multiscale context models can be used to obtain contextual information. The supervised approach incorporates knowledge of image features using a new joint multi-context and multiscale (JMCMS) approach to Bayesian segmentation. The JMCMS is used to capture contextual information, by integrating multiple context models of distinct advantages (Fan and Xia, 2003). Multiscale Bayesian approaches to image segmentation have proven efficient in applying contextual behaviour, on a coarser scale, to guide decisions on a finer scale. This approach has the advantage of capturing statistics at all scales.

The unsupervised method uses HMT-3S and JMCMS for textural analysis and classification. In addition, K-means clustering is used to identify the training samples for unknown textures based on the likelihood disparity of HMT-3S (Song and Fan, YEAR?). The unsupervised method is based on the coarsest scale, each node containing the statistical information from all other descendants at finer scales, providing a robust computation of the model likelihood.

This approach requires certain parameters to be defined before proceeding with classification. First, the “weight” parameter needs to be defined. Due to limitations in computer precision and memory, the “weight” parameter requires initialization to prevent overflow occurring when calculating the results; this parameter must be set between 50 and 300. Second, the number of training runs must be defined to indicate the number of iterations of the algorithm. The user can specify any number between 6 and 30. The program also requires the user to input the number of classes that lie in the region. This forces the program to determine, up to a limit, the number of textures that should be classified from the region. The current version of software used for this study can define any number of classes up to a limit of 6.

Seabed Classification based on Space-domain Texture Analysis and Neural Networks

Method 1: Traditional Space Domain Textural Analysis

Three space-domain textural features are extracted to classify backscatter images, namely grey-level run-length features, spatial grey-level dependence matrix, and grey-level difference vectors. Grey-level run-length is defined as a set of consecutive, collinear pixels having the same grey-level value, with the number of points in the run matching the run length. A run-length matrix can be calculated for any given direction. The Spatial Grey-Level Dependence Matrix (SGLDM) is commonly known as ‘grey-level co-occurrence’ matrix (GLCM; Haralick *et al.*, 1973). Each spatial grey-level dependence matrix determines the number of times two pixels of respective grey levels occurs in a particular direction at a specific distance. A SGLDM is a square matrix whose size is equivalent to the number of grey levels present in the input image. The grey-level difference vector (GLDR) records the frequency with which a given grey-level difference occurs between two adjacent pixels. GLDR is a one-column array, the length corresponding to the number of grey levels in an image. The grey-level run-length

features, spatial grey-level dependence matrix, and grey-level difference vectors are calculated in four directions (vertical, horizontal and along the 2 diagonals), to account for both directionality and anisotropy. By averaging these 4 features, the effect of the orientation that any feature may have in a given sub-area is removed. A full list of the features extracted is listed in Müller *et al.* (1997). This textural image analysis approach is used to identify three different types of sedimentary image textures associated with mud, sand and gravel (representative images in Figure 3).

Method 2: Grey-level Co-occurrence Iteration Algorithm

Due to a desire to increase computational speeds for determining texture features, the grey-level co-occurrence iteration algorithm (GLCITR; Clausi and Zhao, 2003) is applied. The program utilizes texture features generated from Haralick *et al.*'s (1973) grey-level co-occurrence matrix (GLCM). GLCITR features do not differ greatly from the features created with the traditional space-domain method. Instead, the differences are in the efficiency of data storage utilized with GLCITR, using linked lists and hash tables to store data entries. Hash tables are data structures allowing rapid access between nodes. Sorted linked lists can be implemented, ensuring that only non-zero co-occurrence probabilities are stored. The features created by GLCITR can be generated in any direction specified by the user. This characteristic allows the anisotropy of textures to be considered, allowing the process of perceptual grouping of textural patterns to be guided. As seafloor textures are not necessar-

ily isotropic (*e.g.*, ripple fields), textural anisotropy was explored, meaning texture attributes may vary depending on azimuth. The GLCITR approach to classification allows the option to take this characteristic into account.

Texture Classification

Artificial neural networks (ANN) have been increasingly applied to remote sensing data as they have shown potential in processing pattern recognition problems (Zhang *et al.*, 2003). Artificial neural networks techniques are effective at solving non-linear problems, and accurately approximating the non-linear relationship between observations and target parameters. After calculating the statistics, in either one of the aforementioned methods, the features created are used as inputs to train the network to recognize image textures. The Matlab® Neural Network Toolbox is used to generate the ANN used for this method. Neural networks are routine oriented devices that can be trained to recognize known textures in other areas of the image. 'Feed-forward' neural network architectures, as used in this case, consist of a set of processing units and neurons arranged in layers. The behaviour of the neural network is defined by the way the neurons are connected and by the strength of these connections, or weights, to the neurons in adjacent layers. The input layer consists of one neuron for each discriminating feature vector chosen from the statistics defined above. The number of hidden layers can be chosen with a specified number of nodes in each layer, defining the network architecture. The output layer contains one neuron for each desired output, being the three sediments to be defined, plus the no data class.

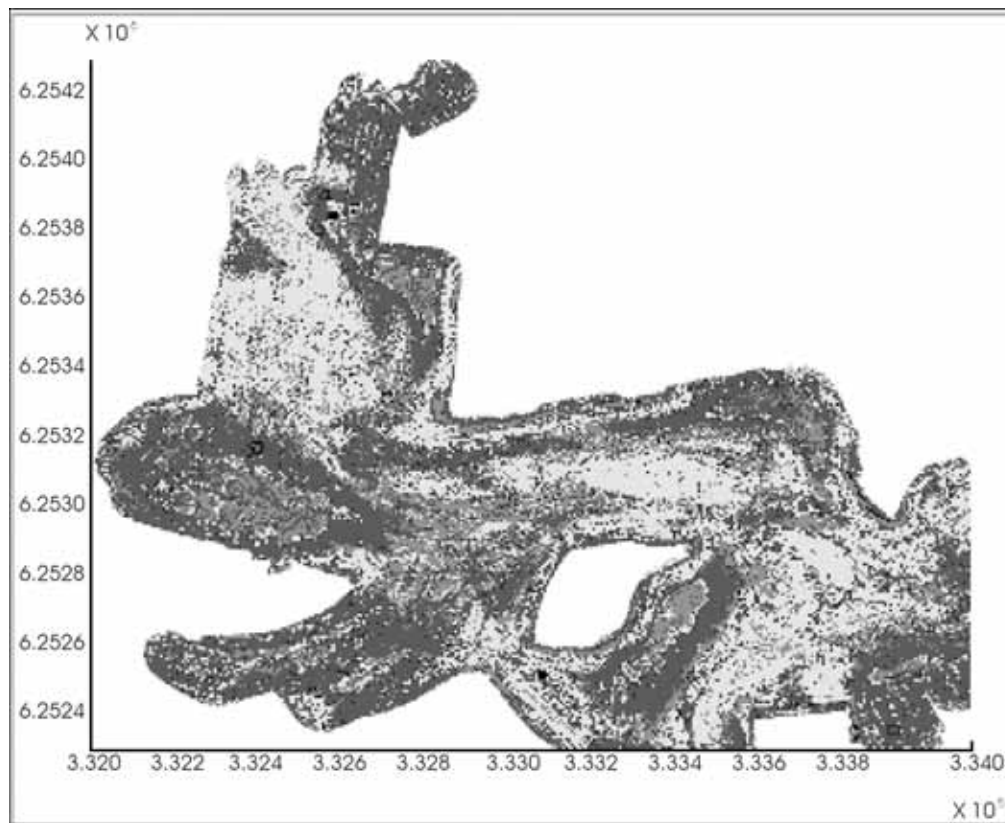


Figure 3. Supervised Classified Image ('normal' mode) with GeoTexture – Balls and Blues region. Pale grey = gravel; mid-grey = sand; dark grey = mud.

The network is trained, using a training dataset of ground-truth samples, until it can approximate a function or associate input vectors with specific output vectors, known as classification. Training the network with back-propagation is a popular method, by which the weights and biases of each node are iteratively adjusted, until the error is minimized to a degree whereby the network approximates the function to a desired level.

Feature Extraction

The first method for textural analysis is carried out with Matlab®, creating feature vectors for sub-images representing different textures. To account for directionality and anisotropy, grey-level run-length matrices, spatial grey-level dependence matrices and grey-level difference vectors are calculated in four directions: vertical, horizontal and along the two diagonals. These four statistics are averaged for each feature to remove the effect the orientation of any feature might have in a given sub-area. The total of 52 features calculated consist of the mean and standard deviation of grey-level intensity in each window, and the four directional mean and variances of statistics calculated from the grey-level run-length matrices, spatial grey-level dependence matrices and grey-level difference vectors. The second method (Zhang *et al.*, 2003) is executed in a Unix environment with the software implemented in the C-programming language. In this approach, nine statistics are used, including the maximum probability, entropy, dissimilarity, uniformity, contrast, inverse difference, inverse difference moment, correlation and mean (Clausi and Zhao, 2003).

Network Architecture

In back-propagation networks each input is weighted with an appropriate value. These weighted inputs are added to a bias, summed and passed to the transfer function. Any differentiable function may be used to generate the output from each neuron. Three transfer functions are commonly used in back-propagation networks: log-sigmoid, tan-sigmoid, and a linear function. Given a net input, ranging from negative to positive infinity, the log-sigmoid transfer function generates outputs in the range 0 to 1, and the tan-sigmoid transfer function generates outputs ranging from -1 to +1. For the purpose here, a network with tan-sigmoid transfer functions in the two hidden layers and a log-sigmoid transfer function for the output layer is chosen. A log-sigmoid output transfer function is necessary because the output is given in terms of zeros and ones.

After completing the design of the network, it is applied to a dataset in three steps: 1) training based on known input, 2) performance testing using both training data, as well as test data unknown to the neural network, but whose class-labels are known to the user, and 3) classification of unknown input. All modelling was carried out using Matlab®. Various algorithms are available for training a neural net. A common problem is that the global error minimum may never be reached due to trapping in a local minimum. To avoid this and to reduce training times, it is desirable to employ an adaptive learning rate to keep the learning step as large as possible while retaining stable learning. This can be achieved by making the learning rate responsive to the complexity of the local error surface by utilizing Levenberg-Marquardt optimization (Hagan and Manhaj, 1994), which represents a blend of the gradient descent and Gauss Newton methods depending on the rate of success during training.

To further optimize learning, a performance function can be applied. The default performance function for feed-forward networks is the mean square error. The function minimizes the sum of the average squared error between the network outputs and the target outputs.

RESULTS

GeoTexture Classification

The GeoTexture classification program provides a supervised approach to texture segmentation and the output is strongly dependent on the user's input, in terms of tying backscatter textures to seafloor sediment types. At present, GeoTexture does not support unsupervised texture mapping. GeoTexture can provide georeferenced information for all areas on the image. The geographic coordinates of the available seafloor sample data were transformed into Eastings and Northings to provide data in the same reference frame that the UTM-projected mosaic is displayed with. The size of the user defined image squares was set at a constant width of 25 x 25 and centred on a total of 66 sample sites, divided into 22 sand, 22 gravel and 22 mud samples. To map a pixel on an image to a characterized texture, a feature vector is extracted for a weighted window on the image centrally disposed on the pixel (*i.e.*, 'normal' classification). For 'enhanced' classification, feature vectors are extracted across the weighted window, not just at its centre. Both 'enhanced' and 'normal' classification modes were utilized in this section to compare classification accuracies.

The manual, interactive definition of the 25 x 25 pixel large mosaic sub-areas centred on samples is a labour intensive approach, and GeoTexture provides no automated approach for using sample sites for supervised classification. Using a Matlab® script, the classification accuracies for each classification mode were automatically generated by comparing the classified images with the entire available seafloor sample dataset in the survey area (75 gravel, 103 sand and 67 mud samples over all the classified swaths). Supervised training, using 'normal' classification, resulted in higher classification success for all sediment classes than "enhanced" mode. The Balls and Blues region displayed the highest classification accuracies; sand classified at 42%, gravel at 58% and mud at 80% accuracy. The mud class exhibited the highest classification accuracies and 3 regions obtained accuracies at 80% or above. Sand was classified with the least accuracy in the Darling West region where none of the 6 sandy areas in the region were identified correctly. These classification results are listed in Table 1 and illustrated in Figures 3 and 4. The results gained by averaging all accuracies for the input regions, obtained with normal classification, have sand classified at 29%, gravel at 56% and mud at 70%. Using enhanced classification, the classification success for the sand, mud and gravel classes, are 29%, 63% and 63%, respectively.

GeoTexture also outputs a discrimination, or confusion, matrix. This matrix is able to inform the user of the likelihood of two image textures being confused during image classification. The discrimination matrix for the 66 characterized textures was examined to inspect the separation between the different texture classes. When a particular entry, corresponding to a sub-sample, contains a value of zero in the discrimination matrix, the probability of the system being "confused" when classifying is zero. Any

Table 1. Initial statistics to calculate training and testing accuracies

Parameter	Training Accuracy (%)				Testing Accuracy (%)			
	Gravel	Sand	Mud	No data	Gravel	Sand	Mud	No data
Net 1	88	59	92	60	88	77	66	100
Net 2	92	62	92	100	83	41	66	100
Net 3	77	77	96	100	77	88	60	100
Net 4	100	100	92	100	88	65	88	100
Net 5	88	74	51	100	77	61	54	100

non-zero entries are representative of the similarity in textural properties between 2 sub-images. To attempt to improve classification, 10 sub-samples displaying non-zero entries were removed. These 10 non-zero entries were a result of the correlation of the sub-sample to other sub-samples in different texture classes. These samples could work to confuse GeoTexture when classifying. Due to time constraints, only 3 regions were mapped with this reduced dataset. Normal classification was used with a decision threshold of 5 to minimize run-time costs, and classification accuracies improved somewhat (Table 2). Sand was still poorly classified in relation to mud and gravel, although the new dataset was able to raise the average accuracy of sand to 47%. The average classification over the 3 regions for gravel and mud were 67% and 77%, respectively.

Table 2. Classification accuracies for GeoTexture-improved

Region	Classification mode	Sand accuracy (%)	Gravel accuracy (%)	Mud accuracy (%)
Darling - 50 cm	Normal	35	76	89
New Balls - 50 cm	Normal	52	58	75
New Steele - 50 cm	Normal	55	n/a	68

Results from Mapping using Wavelets

Unsupervised Approach

The unsupervised method requires no previous knowledge to perform the classification. Because there is no prior knowledge, the model training is directly performed on the entire image obtaining a global statistical characterization and the software used for this approach (Song and Fan, 2002) is designed to read 256 x 256 sized images in the .raw format. To convert images to this format, the mosaic areas were automatically segmented into smaller images using a Matlab® script, and then converted into the .raw format. This conversion was done using Photoshop® Version 7.0 to batch-convert these image formats. This conversion had negligible effects

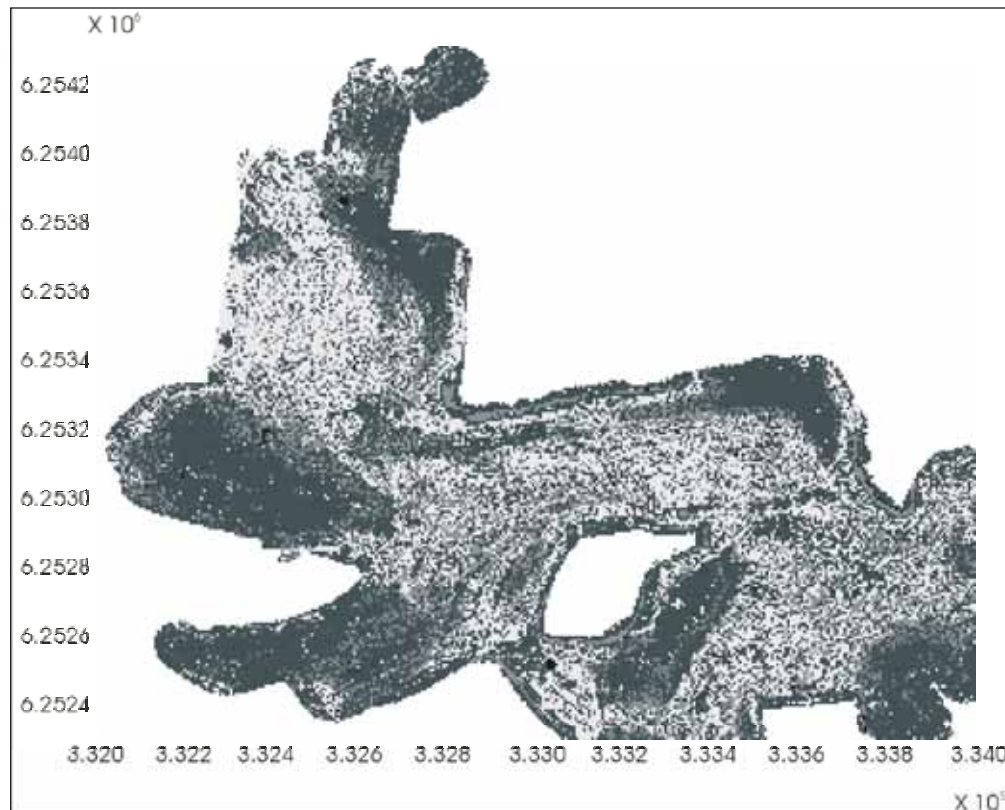


Figure 4. Supervised Classified Image ('enhanced' mode) with GeoTexture – Balls and Blues region. Pale grey = gravel; mid-grey = sand; dark grey = mud.

on program run-time. The program was initially tested on the Balls and Blues region mosaic, binned at 1 m resolution. The number of training runs and weights were experimented with to investigate the effect of these variables on program run-time. The run-time costs, when performed on a SUN Solaris 8 Blade 1000 computer, are listed in Table 3. The results show an increase in program run-times when a greater number of training runs is specified, as expected, because more iterations require more computational time. Raising the weight parameter resulted in run-times escalating, although this was not a major consideration when initializing program parameters, as the weight function is not important for the algorithm, only for the implementation by the computer.

Table 3. Run-time results when specialized parameters are varied

Varying parameter	Image size (square)	Weight function	Training runs	Number of classes	Time (sec)
Training runs	256	50	6	3	36
	256	50	15	3	41
	256	50	30	3	50
Weight function	256	50	6	3	38
	256	150	6	3	58
	256	300	6	3	58

A third parameter that must be set is the ‘number of classes’ parameter, representing the different texture classes that lie in the region. For unsupervised classification, the number of classes is assumed to be known before classification. This parameter forces the program to determine, up to a limit, the number of textures that should be classified from the region. The program was run whilst varying the ‘number of classes’ parameter (Table 4).



Table 4. Run-time results when specialized parameters are varied

Varying parameter	Image size (square)	Weight function	Training runs	Number of classes	Time (sec)
Number of classes	256	50	6	1	32
	256	50	6	3	36
	256	50	6	4	56
Input image size	256	50	6	6	71
	256	50	6	3	36
	4000	50	6	3	1684

The type of input image also seemed to affect computation costs. When the input image had significantly less texture classes present than that specified with the ‘number of classes’ parameter, the time for classification increased but, when the image obviously displayed a range of textures, and the class was chosen appropriately, the program run-time was notably slower. This produced a range of classification times when using the same parameters on different images. The times averaged anywhere between 28 and 170 sec for classification of a 256 x 256 image. The difference in the classification times are listed in Table 5.

For some sub-images, larger weight values provided sharper texture boundaries, however, the weight contributed significantly to run-time costs. To compromise between run-time cost and classification accuracy, a training number of 6 and weight of 50 were cho-

Table 5. Run-time results when specialized parameters are varied

Input image	Time (sec)
	170
	28

sen to keep run-time costs at a minimum. To try and distinguish 3 textures and the no-data class, the program was run with the ‘number of classes’ parameter set at 4.

Results

A single class parameter could not be found to provide a consistent output result based on visual comparison of the classified image and the processed backscatter mosaic where the sample sites were overlain. This is a simple consequence of not all 256 x 256 pixel sub-areas having the same number of classes represented. Both 3 and 4 classes were used to texture segment the sub-images. The program parameters, defined for the entire mosaic, were initialized with 82% of the sub-images mapped to 3 classes, with the remaining 18% mapped to 4 classes. The weight was predominately set to 50 with only 4% of the sub-images requiring a weight of 150. The number of iterations remained consistent at 6. After the sub-images were classified and segmented, they were converted back from .raw into .bmp format using Photoshop® Version 7.0. With the use of a Matlab® script, the images were rejoined to represent the original mosaic displayed in Figure 5. The classification success for this method was evaluated by visual interpretation of the classified image where the sediment samples were overlain (Figure 6). The highest classification accuracy was obtained for gravel types (100%). Sand was classified with the least accuracy out of all texture classes at just 32%.

Supervised Approach

The supervised method of Bayesian classification is based on sub-images in .raw format with a size of up to 512 x 512 pixels. To choose areas that most accurately represent a given type of sediment, sub-areas with the greatest number of sediment samples of the same type in an area of 512 x 512 pixels were chosen. The program was trained with 3 selected images representing homogeneous sand, mud



Figure 5. Unsupervised classification with Wavelets – joined mosaic.

and gravel. The time for the training of the supervised Bayesian segmentation, with each sub-image being trained with a weight of 50 (default) and number of training runs set at 50 (default) was 164 sec on average (Table 6). This form of textural analysis requires the training computation to be performed only once, as the training results (model parameters) for each texture can be saved into data file and applied to any future segmentation of related mosaics.

Using a Matlab® script, the mosaic is then automatically segmented over the region into 512 x 512 pixel sub-images. To initial-

ize the program, the weight, number of training, and number of classes for the image must be specified. The program's run-times using a constant weight of 50 and 6 training iterations, but varying other parameters, are listed in Table 7.

The output sub-images, after classification and segmentation, are converted back to .bmp format and rejoined into a mosaic. The results are displayed in Figure 7. Using the same method to determine the classification accuracy as that used in the unsupervised training, it was found that supervised classification had adverse

Table 6. Run-time results when generic parameters are varied

Varying parameter	Image size (square)	Window size (or segmented image size)	Number of classes	Creating number of sub-samples	Time (sec)
Number of classes	4000	25	3	243	286
	4000	51	3	243	315
	4000	101	3	243	322
Input image size	256	25	3	100	180
	512	25	3	419	520
	4000	25	3	25,600	32,658

Table 7. Run-time results when generic parameters are varied

Varying parameter	Input image size (square)	Number of classes	Time (sec)
Input image size	512	3	54
	2000	3	336
	4000	3	586
Number of classes	512	3	44
	512	3	54
	512	3	132

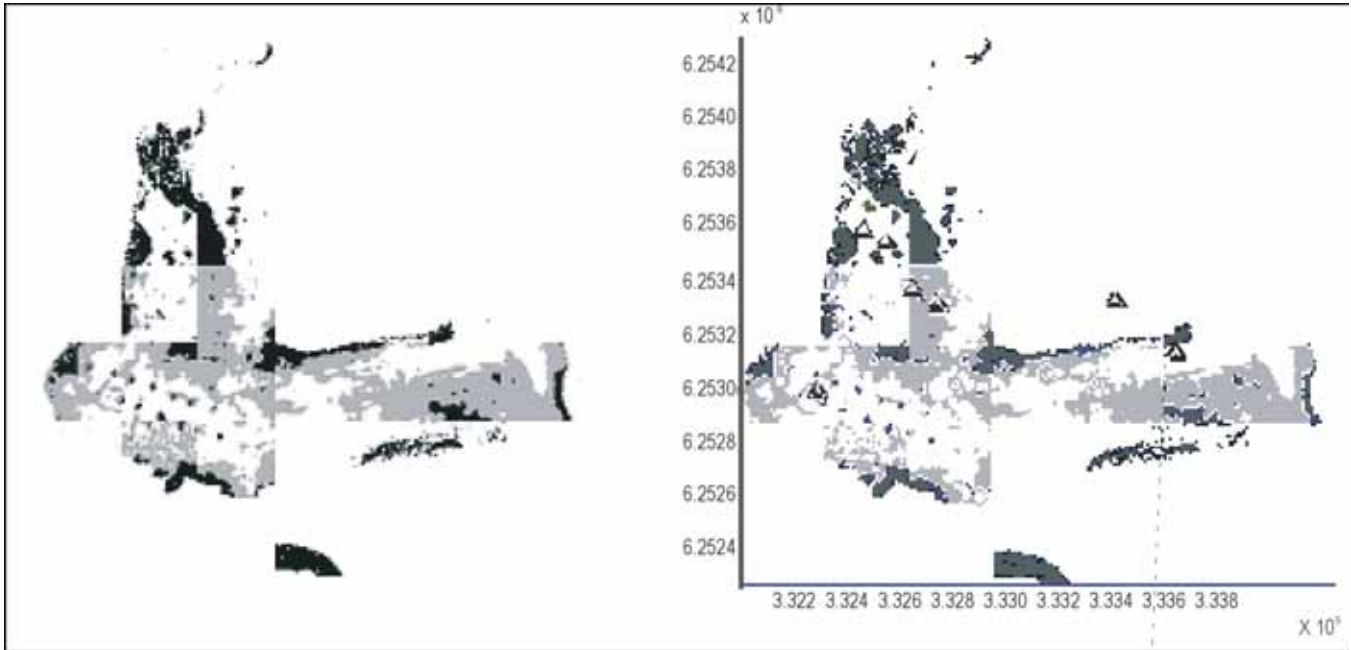


Figure 6. Unsupervised classification containing only three sediment classes. The image on the right is used for interpretation with the sediments overlaid.

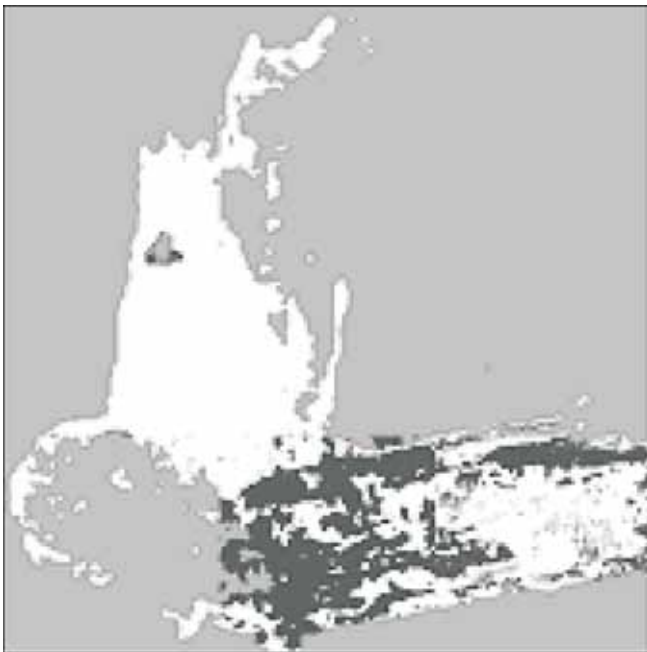


Figure 7. Rejoined image after supervised classification with wavelets.

effects on classification accuracies. Using this method, the classification success was 100% for gravel, 29% for sand and 36% for mud. This illustrates that image textures associated with sediment types sampled at a point-location on the seafloor have a substantial variance, and that this method does not allow for sufficient training to capture the natural variance of image textures associated with individual sediment types.

Neural Network-based Classification – Space Domain Feature Extraction Approach

Feature Extraction Initialization

A total of 92 samples from the 3 different texture classes, along with the “no data” examples from the image, were initially used for training, with 67 samples extracted for testing (22 samples from each texture class with one sample representing the “no data” class). The extraction of these sub-samples involves the mosaic being cropped into sub-images centred on a sample site. The time taken to perform the segmentation of the sub-areas varies depending on the size of the segmented image. The run-times, when executed on a SUN Solaris 8 Blade1000 computer are summarized in Table 6.

The sub-images are used as inputs to a Matlab® script which performs the textural analysis. One feature vector is created for each sub-image comprising 52 statistics calculated from the grey-level, run-length features, spatial grey-level dependence matrix, and grey-level difference vectors. The run-time costs for the extraction of these features, with varying window size, are listed in Table 6. The time to create feature vectors increases with increasing window size. A window size of 25 x 25 pixels required 286 sec to perform textural analysis compared to 322 sec when creating the feature vectors over a 101 x 101 pixel sized sub-sample.

Areas of 25 x 25 pixels were chosen because smaller areas are not large enough to contain textures characteristic for different seafloor classes, while larger areas may contain seabed of more than one sample class. After individual feature vectors are created, they are grouped according to sediment type and saved into classes.

It has been shown previously (Müller *et al.*, 1997) that many features do not necessarily provide discriminatory characteristics between classes. Such features must be discarded before training the neural network. To find the most texturally distinguishing statistics for each class and for each method, a Matlab® script was applied to plot the textural statistics against sample number to compare the discriminatory ability of each statistic. Sixteen of the 52 statistics were found to exhibit the largest variance between classes, thus offering the greatest level of discrimination. The selected statistics are listed in Appendix A.

Network Initialization

The network was initially configured using Levenberg-Marquardt optimization for training. The gradient descent method, that has an added momentum for the reduction of the mean square error, was utilized for program learning. To optimize learning, the number of epochs was set to 65, the goal parameter set to 0.05, and ‘tansig’ ‘tansig’ ‘logsig’ functions were used as an initial combination of network transfer functions. Execution time, associated with training and testing a neural network, requires approximately 50 sec. This figure varies with the number of training samples, number of epochs, the weights and biases of the network.

The neural network requires definition of the structure of the layers and number of nodes, which will differ for the two methods. The numbers of hidden layers and corresponding nodes in each layer have a considerable impact on the performance of the network. The number of distinguishing input vectors defines the number of nodes in the first layer, and the number of target classes defines the nodes in the output layer. The first layer was set to 16 to represent the most discriminatory statistics chosen previously. The number of nodes in the output layer was defined as 4 corresponding to the 3 sediment classes and no-data class. Using the number of nodes in the hidden layer set to approximate 2 to 2.5 times the number of nodes in the output layer, the hidden layer was initially set to contain 10 nodes.

Classification Results

The use of the initial network structure produced accurate training results for all texture classes. The classes could be consistently trained to high accuracies by implementing the ‘tansig’ and ‘logsig’ transfer functions, ‘learngdm’ learning function, and the ‘msreg’ performance function. Altering these functions had either detrimental or negligible effects on training accuracy. The ‘hardlim’ function caused the average accuracy for training to drop to 30% from the average accuracy obtained from the ‘logsig’ function. These alternate parameters, listed with associated accuracies in Table 8, were not implemented in the network.

The number of nodes in the hidden layer, which is one of the most important parameters

Table 8. Accuracies for various network functions and nodal structure

Varying parameter	Parameter	Training Accuracy (%)				Testing Accuracy (%)			
		Gravel	Sand	Mud	No data	Gravel	Sand	Mud	No data
Network functions	'purelin'	52	78	65	10	31	33	18	100
	'hardlim'	100	0	15	100	40	0	10	0
	'poslin'	100	0	0	100	50	0	0	0
	'logsig'	76	91	40	100	13	81	0	100
Nodes in hidden layer	6 nodes	52	100	32	80	44	0	12	77
	8 nodes	100	71	47	100	77	0	22	0
	10 nodes	66	93	0	100	55	22	0	0
	12 nodes	83	50	99	100	44	28	20	100

when defining network architecture, was altered to see the effects on classification accuracy (Table 8). Twelve nodes were chosen for subsequent network training, as the accuracies obtained when using this network structure surpassed network structures from all different architectural layouts.

However, low testing accuracies necessitated implementation of an early stopping technique to prevent the network from over-fitting the data. Early stopping is a technique that makes use of a validation set, in conjunction with both training and testing sets, to monitor the error on itself during the training of the network. To further improve the testing accuracy, the number of training samples was increased to 108, with 55 samples used for the testing and validation datasets respectively. The network has a larger likelihood of texture classification with a greater number of samples used.

The training and testing results, obtained with use of the early stopping technique and generation of more training samples, are listed in Table 9. The implementation of early stopping was affirmed as all successive runs were stopped by validation suggesting that the network was previously being over trained. The testing accuracies were found to increase with the introduction of early stopping. The use of a larger number of training samples was found to increase the classification accuracy, but was still found to be inadequate for texture classification.

Therefore, the selected feature vectors were reviewed and an additional seven statistics (Appendix A) were chosen that were found to contribute additional textural information for each class. The network was re-run with 23 nodes comprising the first layer, 12 nodes in the hidden layer, and four output nodes. The statistics cho-

Table 9. Training and testing accuracies obtained from varying parameters

Varying parameter	Parameter	Training Accuracy (%)				Testing Accuracy (%)			
		Gravel	Sand	Mud	No data	Gravel	Sand	Mud	No data
Introduction of early stopping	Net 1	0	94	40	100	0	100	33	100
	Net 2	85	70	88	100	77	58	20	100
	Net 3	77	85	55	100	66	77	11	100
Generation of more training samples	Net 1	92	44	74	100	88	55	57	100
	Net 2	0	74	96	100	0	68	70	100
	Net 3	85	59	92	100	96	48	45	100

sen were found to have beneficial impacts on the results, with the output network able to classify unknown data with higher accuracy. Of the five networks run with these parameters, Network 4 has the best overall test success (Table 1). When tested on a sample of unknown data (the validation dataset), Network 4 obtained accuracies of 89%, 92% and 84% for the gravel, sand and mud classes respectively.

Applying the Trained Network

After a satisfactory network has been generated, it is used classify the entire image. This method requires the input image to be segmented into smaller images, to obtain a feature vector for the sub-image, before being simulated with the neural network. This function is performed with a Matlab® script, which outputs the class most likely to be represented by the sub-image. The classification is only determined for the pixel centrally located on the sub-image, not for every pixel in the image. In a 4000 x 4000 image a total of 25,600 sub-images were processed (Figure 8). The overall time to classify this input image utilizing Network 4 is 32,658 sec.

Neural Network-based Classification – GLCITR Feature Extraction Approach

Feature Extraction Initialization

Textural analysis using this method requires an input image in the VIP format (Tong and Clausi, 2002). This conversion can be achieved utilizing the program XV, which transforms the input image into an 8-bit mode, grey-scale VIP image. All mosaics were converted in this manner with negligible run-time costs. The feature extraction initialization parameters and their associated run-time costs when executed on SUN Solaris 8 Blade 1000 computer are listed in Table 10.

The time for program execution was found to increase when a greater number of statistics were calculated. The increase in computation time was not linear as individual statistics require different numbers of calculations resulting in varied execution times. The time difference for creating feature vectors over the entire image ranged from 148 sec, when calculating just one statistic, to the maximum 925 sec required for 9 statistics (Table 10). Altering the dis-

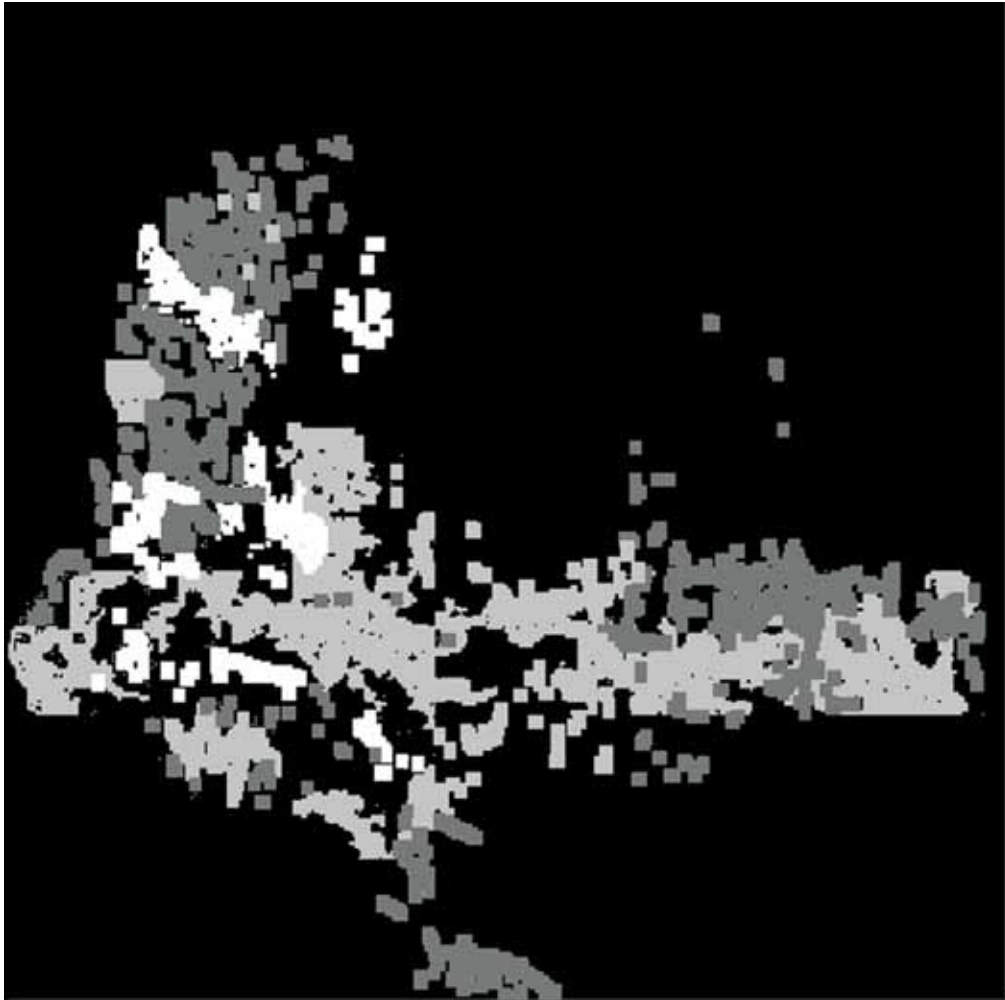


Figure 8. Classification with space-domain feature extraction utilizing neural networks: gravel = white; sand = light grey; mud = dark grey.

Table 10. Run-time results when selected parameters are varied

Varying parameter	Image size (square)	Displacement (degrees)	Window size (or segmented image size)	Number of statistics	Creating number of sub-samples	Time (sec)
Number of statistics	4000	45	25	1	15,808,576	148
	4000	45	25	4	15,808,576	529
	4000	45	25	9	15,808,576	925
Input image size	4000	0	25	9	15,808,576	924
	4000	190	25	9	15,808,576	924
	4000	Rotational invariance	25	9	15,808,576	934

placement in either the x or y direction had negligible effects on program run-time. Altering the window size had severe implications for program run-times. The time to create feature vectors increased substantially with increasing window size. A window size of 25 x 25 pixels required 925 sec to perform textural analysis compared to 2865 sec when creating the feature vectors over a 101 x 101 sized sub-sample. The input image size further influenced program run-times. Increasing the image size had corresponding increases in execution time. The results are listed in Table 11.

To maintain consistency with method 1, a window-size of 25 x 25 was used to initialize the program. Despite the increase in computation times, the maximum of 9 statistics was generated to provide a greater description of the image, increasing classification accuracies. The algorithm was applied in four directions (x, y, and diagonals).

The difference between the east–west oriented statistics, averaged with the difference between the north–south oriented statistics, was appended to the existing rotational invariant features using Matlab®. The average of the statistics of the four orientations provides a rotation invariant approach to classification. This increases the number of statistics generated to a total of 18. The program calculates statistics for a pixel centred on the window for every orientation specified. The window is run across the entire image, creating feature vectors for every pixel in the image, excluding the border. These feature vectors are saved to file in the binary format.

Features were extracted for the 92 training and 66 testing datasets, as before. These features are grouped according to sediment type and saved into texture classes. To find the most texturally distinguishing statistics for each class and for each method, the results for the statistics were plotted against sample number. The mean and logarithm of the data are taken to spread the data to a degree to allow comparison between classes. Sixteen statistics, from the total of 18, offered the greatest contribution to the level of discrimination between textures (see Appendix B).

Network Initialization

The network was initially configured using the same parameters as for our initial trials in Method I. To test the effects of introducing directionality into the feature vector set, training and testing results were compared for both rotationally invariant and variant approaches.

Classification Results

The results obtained from training the network with just eight statistics was slightly lower in comparison to the results obtained when utilizing the directionality information. The 8-7-4 network structure displayed low and inconsistent test accuracies (Table 12). The results obtained when using 16 statistics were slightly higher. Altering the transfer, learning and performance function parameters was found to be ineffectual (Table 13). Using ‘purelin’ as a transfer function produced comparable results, although it did not improve the accuracy of the network overall. The classes could be consistently trained with reasonable accuracies by implementing the ‘tansig’ and ‘logsig’ transfer functions, ‘learnqdm’ learning function, and ‘msreg’ performance function.

Testing of the network did not produce accurate classification results. The texture classes were most often classified with less than 50% accuracy. In attempts to increase testing accuracies, ‘early stopping’ was implemented. Following the previous method, the number of training samples was increased resulting in a corresponding decrease in testing and validation samples (Table 14). The addition of early stopping did not significantly improve the network’s ability to distinguish different textural classes. However, the addition of more training features resulted in the training dataset containing 108 samples, with the testing and validation sets containing 55 samples each. The addition of more data samples corresponded to an increase and stabilization in classification accuracies (Table 14).

Table 11. Run-time results when generic parameters are varied

Varying parameter	Image size (square)	Displacement (degrees)	Window size (or segmented image size)	Number of statistics	Creating number of sub-samples	Time (sec)
Window size	4000	Rotational invariance	25	3	15,808,576	925
	4000	Rotational invariance	51	3	15,808,576	1472
	4000	Rotational invariance	101	3	15,808,576	2865
Input Image size	256	Rotational invariance	25	3	53,824	17
	512	Rotational invariance	25	3	238,144	22
	4000	Rotational invariance	25	3	15,808,576	925

Table 12. Training and testing accuracies when network architecture is altered

Architecture	Training accuracy (%)				Testing accuracy (%)			
	Gravel	Sand	Mud	No data	Gravel	Sand	Mud	No data
8-7-4	0	20	80	100	0	4	88	0
8-7-4	100	0	11	0	79	0	22	0
16-7-4	40	37	62	3	0	44	77	0
16-7-4	81	29	25	3	77	16	9	100

The improvement in accuracies was only slight, and the removal of four statistics that contributed little to discriminating between the four classes, resulted in the use of just 12 statistics. This alteration of the feature vectors resulted in a major improvement of training accuracies for all classes. The network structure used consisted of 12 input nodes, a hidden layer containing ten nodes and four nodes comprising the output layer. The training and testing results are summarized in Table 15. Of the five networks run with these parameters, Network 3 has the best overall test success, with an average of 87%. Network 3 was used to classify the validation data, obtaining accuracies of 77%, 88% and 78% for the gravel, sand and mud classes respectively. Program initialization for this method created feature vectors over the entire image and stored results to file. In a 4000 x 4000 image, a total of 15,808,576 features were calculated (Figure 9). The overall time to classify an input image, utilizing Network 3 is 1850 sec.

DISCUSSION

Review of GeoTexture

Supervised classification required textural areas to be defined over windows of a specified size at defined locations. Maintaining consistencies with textural analysis proved difficult, as human perception and judgment was required to sustain a constant window size for characterization. In the absence of an automated method to position the window, the location of the textural areas proved labour intensive. The factors found to have the greatest effects on program execution time included whether ‘enhanced’ or ‘normal’ mode was specified for classification, the number of samples used for textural analysis and the input image size. Run-times were found to increase dramatically when ‘enhanced’ mode was specified. Linear increases were observed with increases in the number of samples characterized, and more time was required with larger input images. Supervised classification utilized 66 samples for program training. When simulated with ‘normal’ classification, average classification times exceeded 300 sec. This value further increased when enhanced classification was implemented requiring 7 times the run-time costs for supervised classification set in ‘normal’ mode.

The supervised classification approach produced reasonable accuracies, but is associated with a labour intensive approach for textural analysis. The same set of characterized textures was used for all input mosaics providing consistencies with run-times, and stabilizing accuracies. Despite the correlation between the number of characterized textures and the time for program execution, a greater

Table 13. Training and testing accuracies when network functions are altered

Function	Training accuracy (%)				Testing accuracy (%)			
	Gravel	Sand	Mud	No data	Gravel	Sand	Mud	No data
'purelin'	92	85	94	3	32	41	21	100
'hardlim'	59	33	92	44	33	0	44	100
'poslin'	89	0	62	48	55	0	36	0
'logsig'	90	100	85	82	31	27	31	100

number of samples were found beneficial to classification accuracy. Although more samples can aid classification, they can also work to confuse the program, especially when there is a large degree of variability within a specific class, relative to the differences between patterns in separate classes. This suggests that the quality of the samples is more beneficial to program training than the number of samples. The analysis of the discrimination matrix provided valuable insight to the separation of the characterized textures and promoted the removal of those textures working to confuse the system.

Review of the Wavelet/Hidden Markov Chain Method

Both unsupervised and supervised algorithms were tested in this study and required different training initializations. Images of identical size have to be used for training and image segmentation, but training requires selection of sub-images that are isotropic, *i.e.*, contain only one sediment type. Given the relatively large size of the training images (512 x 512 pixels = 65.536 m²) it is difficult to find images that unambiguously contain only one texture for training purposes. If one reduces the size of the training images to overcome this problem, then the size of images used for segmentation has to be reduced accordingly, and the problem is turned around, as during

Table 14. Accuracies obtained when parameters are varied

Varying parameter	Parameter	Training Accuracy (%)				Testing Accuracy (%)			
		Gravel	Sand	Mud	No data	Gravel	Sand	Mud	No data
Introduction of early stopping	Net 1	18	92	51	34	31	49	42	0
	Net 2	62	44	59	74	53	40	0	100
	Net 3	25	77	0	44	75	3	0	100
Generation of more training samples	Net 1	81	7	18	0	77	11	22	0
	Net 2	37	85	77	30	55	32	35	100
	Net 3	70	48	55	3	41	33	52	100

Table 15. Improved set of statistics to calculate accuracies

Parameter	Training accuracy (%)				Testing accuracy (%)			
	Gravel	Sand	Mud	No data	Gravel	Sand	Mud	No data
Net 1	78	59	82	60	38	77	66	100
Net 2	92	62	72	100	83	41	66	0
Net 3	77	77	96	100	77	88	84	100
Net 4	100	100	42	100	88	65	13	100
Net 5	88	74	51	64	77	61	54	100

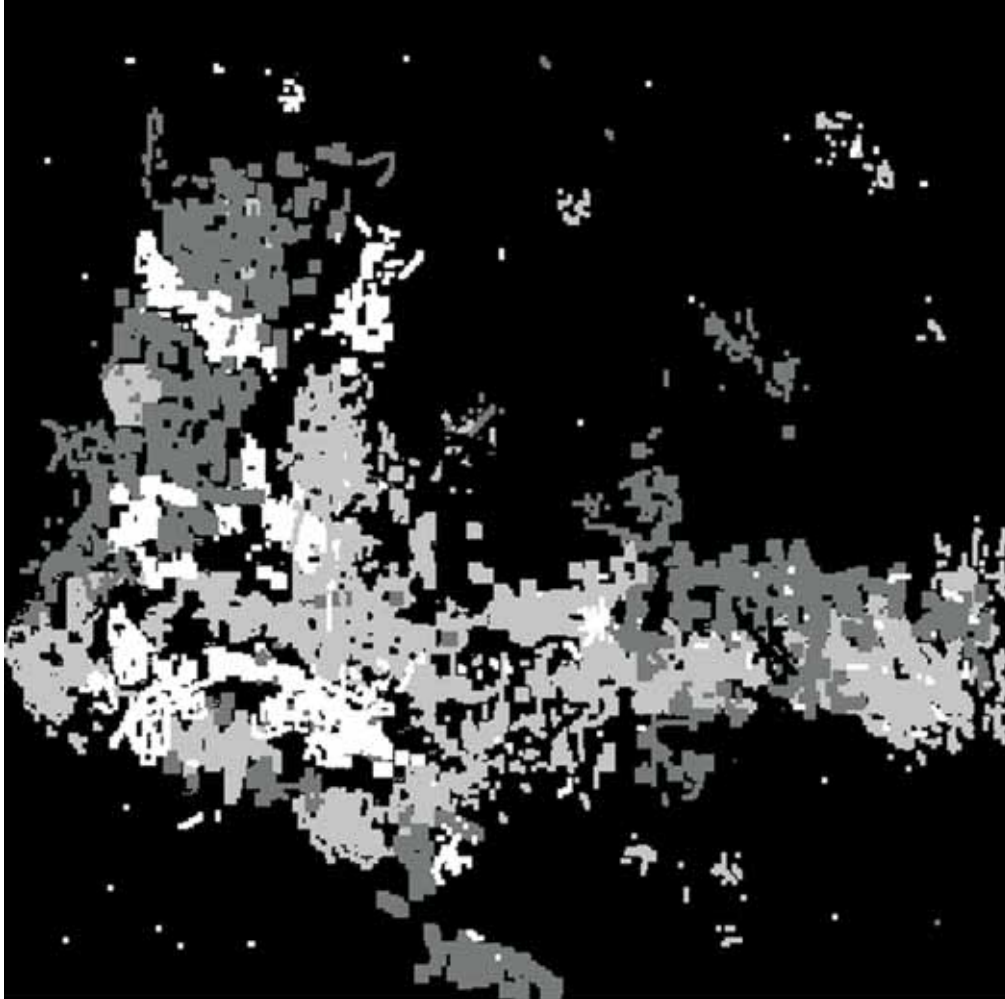


Figure 9. Classification with GLCITR method for textural analysis utilizing neural networks: gravel = white; sand = light grey; mud = dark grey.

classification every sub-image is segmented into the same number of classes (in this case 3 sediment types and “no-data” areas). The smaller the sub-images for segmentation, the larger the likelihood becomes that one forces a fixed number of classes onto a sub-image that actually only contains a subset of these classes. Therefore, there is a trade-off between optimal training image and segmentation image size; training would work better with smaller (texturally isotropic) images, and segmentation would work better with larger images representing the full heterogeneity in terms of all classes being represented.

Review of Neural Network Approach

The spatial-domain method for textural analysis requires the extraction of sub-regions and the calculation of the feature vectors associated with these regions. These processes can be automated providing a relatively effortless characterization process. Although requiring minimal labour costs, the problems associated with this method of textural analysis are related to the run-time costs. The relatively subjective nature governing the choice of feature vector statistics can result in very different trained networks based on the same dataset. All networks benefited from the introduction of ‘early stop-

ping’ to avoid over-training and from a carefully chosen set of statistics used for characterization. The GLCIA iterative algorithm for textural analysis (Clausi and Zhao, 2003) proved to be complex in nature relative to the traditional space domain method, mainly because the total number of feature vectors is much smaller, as the method is not designed to make use of grey-level run-length or grey-level difference vectors. The use of anisotropy in feature vectors had beneficial results on the training of the network. The choice of the most discriminatory statistics, coupled with the network architecture, provided the most significant impacts on the success of the network (see Appendix C).

Comparison between Methods of Texture Classification

The objective of this study was to compare different seafloor classification methods in terms of both classification accuracy and data-preprocessing and program run times. The latter is an important consideration before adopting any given method for close to real-time seabed classification, for example at sea. A standard set of data points was used to maintain consistency in validating the classification success of each respective method.

From the three different approaches to classification, texture mapping with GeoTexture was found to require the least time to perform the classification when mapped utilizing 'normal' classification mode. However, the data-preprocessing/interactive mosaic sub-image selection times were excessive for large numbers of seabed samples. The implementation of 'enhanced' classification results in a substantial increase in computation times, in our case with worse classification results. GeoTexture can classify an input image of size 4000 x 4000 within 100 to 300 sec utilizing 'normal' classification. The program requires over 2000 sec to texture classify the same image using 'enhanced' classification. GeoTexture performed favourably, but not to the same extent as the neural network approach to classification, providing an average accuracy for all texture classes of 63%.

The wavelet-based method for classification provides competitive run-time costs, but unsatisfactory results. The times required for the supervised approach to classification with wavelets is slightly less than 500 sec, providing a computationally efficient, but mostly unsuccessful, method of classification, where the average accuracy for all sediments is 55%. The unsupervised method for wavelet analysis is less efficient, requiring 1684 sec for program execution on our sample dataset. The fundamental problem with the design of Fan and Xia's (2003) method for this application is that training and classification image sizes are identical, but for training a given image must be texturally isotropic, whereas for classification images of the same size are always expected to contain all mapped classes, and neither condition is usually met.

The neural network approach to classification proves to be the most computationally expensive, but the most successful method. Utilizing 18 statistics from the iterative method for textural analysis, the run-time efficiency far surpassed the traditional space-domain method for feature extraction. The neural network approach to classification generated the highest classification accuracies from the 3 methods considered. Both methods for textural analysis provided high accuracies for training and testing, although the results from the space-domain features produced slightly higher results. The average classification accuracy amongst the texture classes was 88%, slightly higher than the average accuracy of 81% obtained from the statistics calculated from the iterative method of feature extraction.

CONCLUSIONS

The GeoTexture classification method is promising, but currently relies too much on interactive user-input. Future GeoTexture advances aimed at increasing the automation of the system may provide a more reliable and efficient method for performing texture classification. The supervised wavelet-based method adapted from Fan and Xia (2003) is currently not suitable for efficient texture classification of marine backscatter images, but further development of this relatively recent method may make it more successful. The success of the neural network-based methods provides a promising outlook for further development and application, especially the GLCIA-based texture mapping approach, which combines computationally efficient texture-mapping with the flexibility and efficiency of neural network-based training and classification.

ACKNOWLEDGMENTS

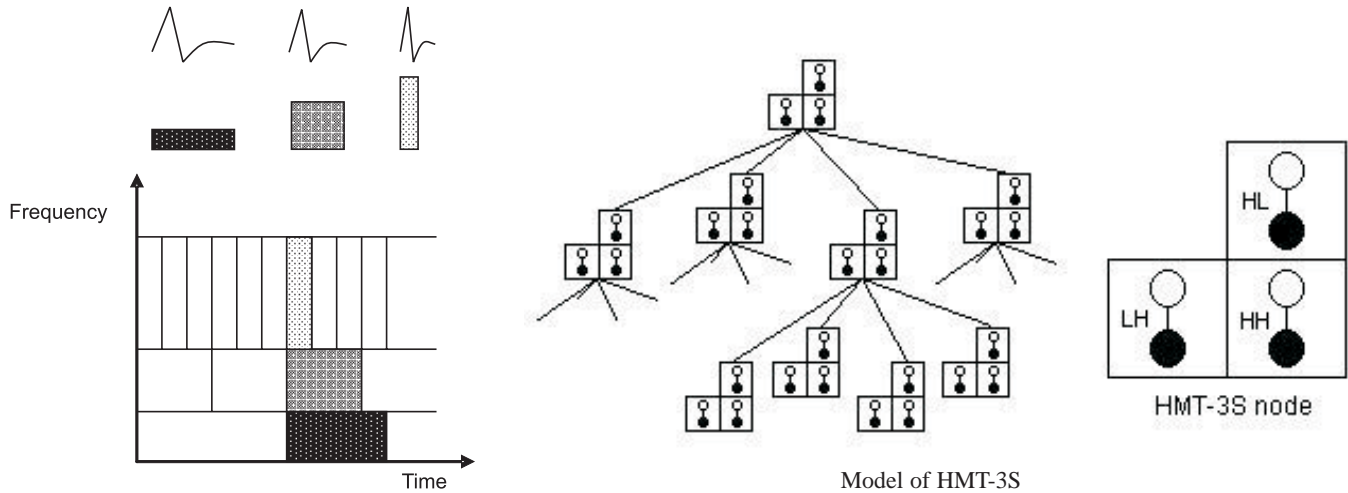
We are grateful to Xiaomu Song (Oklahoma State University) who supplied us with his binary code for both unsupervised and supervised wavelet-based image segmentation. We are also indebted to David Clausi who shared his source code for GLCIA textural analysis with us. Reviewer Yuri Rzhakov provided valuable comments on the manuscript. Adrienne Moseley provided valuable technical assistance for drafting figures and proofreading of the text.

REFERENCES

- Blondel, P., and Murton, B.J., 1997, Handbook of Seafloor Sonar Imagery: John Wiley & Sons, Chichester, 314 p.
- Chang, T., and Kuo, C.-C.J., 1993, Texture Analysis and Classification with Tree-structured Wavelet Transform: IEEE Transactions on Image Processing, v. 2, p. 429-441.
- Clarke, D.J.E.H., 1998, The effect of fine scale seabed morphology and texture on fidelity of Swath bathymetric sounding data: Sea Technology, v. 39, no. 6, p. 87-90.
- Clarke, P.A., and Hamilton, L.J., 2002, The ABCS Program for the Analysis of Echo Sounder returns for Acoustic Bottom Classification: Defence Science and Technology Organisation Report, DSTO-GD-0215 (www.dsto.defence.gov.au/index.html).
- Clausi, D.A., 2001, Comparison and fusion of co-occurrence, Gabor and MRF texture features for classification of SAR sea-ice imagery: Atmosphere-Ocean, v. 3, p. 183-194.
- Clausi, D.A., and Zhao, Y.P., 2003, Grey level co-occurrence integrated algorithm (GLCIA): a superior computational method to rapidly determine co-occurrence probability texture features: Computers & Geosciences, v. 29, no. 7, p. 837-850.
- Fan, G., and Xia, X.G., 2003, Wavelet-based texture analysis and synthesis using hidden Markov models: IEEE Transactions on Circuits and Systems, v. 50, no. 1, p. 106-120.
- Hagan, M., and Manhaj, M., 1994, Training feed forward networks with the Marquand algorithm: IEEE Transactions on Neural Networks, v. 5, p. 989-993.
- Hamilton, L.J., 2001, Acoustic Seabed Classification System: Defence, Science and Technology Organisation Report, DSTO-TN-0401 (www.dsto.defence.gov.au/index.html).
- Haralick, R.M., Shanmugam, K., and Dinstein, I., 1973, Textural feature for image classification: IEEE Transactions on Systems, Man, and Cybernetics, v. 3, p. 610-621.
- Henke-Reed, M.B., and Cheng, S.N.C., 1993, Cloth texture classification using wavelet transform: Journal of Imaging Science and Technology, v. 37, p. 610.
- Laine, A., and Fan, J., 1993, Texture classification by wavelet package signatures: IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 15, p. 1186-1191.
- Müller, R.D., Overkov, N.C., Royer, J.Y., Dutkiewicz, A., and Keene, J.B., 1997, Seabed classification of the South Tasman Rise from SIMRAD EM12 Automatic Classification of Shallow Water Seafloor Backscatter Images: Australian Journal of Earth Sciences, v. 44, p. 689-700.
- Pace, N.G., and Dyer, C.M., 1979, Machine classification of sedimentary sea-bottom: Institute of Electrical and Electronics Engineers Transaction on Geoscience Electronics, v. 17, p. 52-56.
- Reed, T.B., and Hussong, D.M., 1989, Digital image processing techniques for enhancement and classification of SeaMARC II side scan sonar imagery: Journal of Geophysical Research, v. 94, p. 7469-7490.
- Song, X., and Fan, G., 2002, A study of supervised, semi-supervised and unsupervised multi-scale Bayesian image segmentation, in Proceedings of the 45th IEEE International Midwest Symposium on Circuits and Systems, Tulsa, OK, August 2002, ?? p.
- Song, X., and Fan, G., 2003, Unsupervised Bayesian image segmentation using wavelet-domain hidden Markov models [abstract]: Proceedings of the IEEE International Conference on Image Processing, Barcelona, Spain, September 2003, ?? p.
- Stewart, W.K., Jiang, M., and Marra, M., 1994, A neural network approach to classification of sidescan sonar imagery from a midocean ridge area [abstract]: IEEE Journal of Oceanic Engineering, v. 19, no. 2, p. 214-224.
- Tamsett, 2003,
- Tang, X., and Stewart, W.K., 2000, Optical and sonar image classification: Wavelet Packet Transform vs Fourier Transform: Computer Vision and Image Understanding, v. 79, p. 25-46.
- Zhang, T.L., et al., 2003, Evaluating the performance of artificial neural network techniques for pigment retrieval from ocean color in Case I waters - art. no. 3286. Journal of Geophysical Research-Oceans, v. 108, no. C9, p. 3286-3286.

Appendix A

Wavelet theory: The 2-D Discrete Wavelet Transform (DWT) represents an image in terms of a set of wavelet functions. The wavelets HL, LH, and HH.



Appendix B

Neural Network-based classification – Space Domain Feature Extraction Approach: Discriminatory Features selected from statistics created with the traditional Space-Domain Feature Extraction method of classification (Müller *et al.*, 1997)

a) Initial 16 feature vectors which offered the greatest textural discrimination between classes:

1. Mean intensity of region
2. Variance of long-run emphasis
3. Variance of low grey-level run emphasis
4. Variance of high grey-level run emphasis
5. Mean long-run low grey-level emphasis
6. Mean long-run high grey-level emphasis
7. Variance of grey-level distribution
8. Mean spatial grey-level contrast
9. Variance of spatial grey-level correlation
10. Variance of spatial grey-level energy
11. Variance of spatial grey-level contrast
12. Mean variance of grey-level difference
13. Variance in standard deviation of grey-level difference
14. Variance in entropy of grey-level difference
15. Variance in cluster prominence of grey-level difference
16. Variance in cluster prominence of grey-level difference

b) Additional 7 feature vectors which were found to contribute additional textural information for each class:

1. Low grey-level run emphasis
2. Short-run low grey-level run emphasis
3. Run percentage
4. Variance of short-run low grey-level emphasis
5. Variance of short-run high grey-level emphasis
6. Variance of long-run low grey-level emphasis
7. Variance of long-run high grey-level emphasis

Appendix C

Neural Network-based classification – GLCITR Feature Extraction Approach: 16 Discriminatory Features selected from statistics created with the Grey-Level Co-occurrence Iterative algorithm (Clausi, 2002). Statistics 13 to 16 were later removed resulting in a final set of 12 feature vectors.

1. Mean
2. Mean including anisotropy
3. Angular second moment (energy)
4. Angular second moment including anisotropy
5. Dissimilarity
6. Dissimilarity including anisotropy
7. Homogeneity
8. Homogeneity including anisotropy
9. Inverse difference moment
10. Inverse difference moment including anisotropy
11. Entropy
12. Entropy including anisotropy
13. Contrast
14. Contrast including anisotropy
15. Correlation
16. Correlation including anisotropy

