INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

SEABED CLASSIFICATION FROM ACOUSTIC ECHOSOUNDER RETURNS

by

DAVID ARTHUR CAUGHEY
B.A.Sc. University of Waterloo, 1988
M.A.Sc. University of Victoria, 1991

A Dissertation Submitted in Partial Fulfillment of the
Requirements of the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

We accept this dissertation as conforming to the required standard

---

Dr. R.I. Kirlin, Supervisor, Dept. of Electrical Engineering

Dr.W.S. Lu, Departmental Member, Dept. of Electrical Engineering

---

Dr. P. Agathoklis, Departmental Member, Dept. of Electrical Engineering

---

Dr. G. Spence, Outside Member, Dept. of Earth and Ocean Science

---

Dr. J. Preston, External Examiner, Esquimalt Defence Research Detachment

UNIVERSITY OF VICTORIA

Abstract

Efforts to extract information regarding the surficial composition of the ocean bottom have increased in the last decade as increases in the availability of computing power have corresponded with advances in signal processing techniques. The ability to extract information from acoustic echosounders is especially desirable due to the relatively low cost and ease of deployment of such systems. Products already exist for the acquisition and logging of echosounder returns.

An acoustic return is comprised of the incoherent backscatter from individual scatterers within the annulus of insonification that occurs when a spherically-spreading transmit pulse intersects with the ocean floor. The return is a convolution of the source ping, and the impulse response modeled by the backscatter profile. Most echosounders generate an envelope of the received signal. The bottom impulse response undergoes a dilation linear with depth due to simple geometry which can be corrected with time-scale normalization. Under certain circumstances it may be necessary to deconvolve the source ping from the envelope of the return prior to time-scale normalization. It is shown that this can be done by modelling the envelope generation function with a finite sum discrete convolution and the Hilbert transform of the source signal. A second-order Volterra kernel can be derived using a standard predictor network with constrained optimization.

Other factors which contribute to the quality of the return include off-vertical transducer angles which in fact improve the classification by eliminating the nulls that occur in the bottom impulse response due to transducer beam pattern. Spatial averaging can have the effect of beam widening if the transducer angle varies.

Simple feature extraction algorithms are shown to be moderately effective in providing separability. The computational cost of combining the resulting feature sets can be reduced if the individual feature sets are scaled appropriately, reduced and then combined, prior to a reduction to the final dimensionality. The resulting feature space axes contain contributions from both the principal axes of the individual feature sets, as well as cross-algorithmic terms.

Blind clustering of the data is provided through a two-step modification of the *K*-means algorithm. The first step generalizes it to use arbitrary classification metrics, and the second embeds this generalized kernel within a second kernel which modifies the covariance. The resulting *K*-stats kernel is very robust when successively applied to a growing number of clusters.

Dr. R.L. Kirlin, Supervisor, Dept. of Electrical Engineering

Dr.W.S. Lu, Departmental Member, Dept. of Electrical Engineering

Dr. P. Agathoklis, Departmental Member, Dept. of Electrical Engineering

Dr. G. Spence, Outside Member, Dept. of Earth and Ocean Science

Dr. J. Preston, External Examiner, Esquimalt Defence Research Detachment

Contents

List of Tables

List of Figures

Acknowledgments

I thank Dr. R. Lynn Kirlin for his guidance, support, and encouragement to finish. I could always count on him to get excited about my ideas but then recall some obscure communications paper that already dealt with the subject.

I thank Quester Tangent Corp.; in particular D. Rob Inkster for conceiving this project and co-applying for the G.R.E.A.T. scholarship and Brad Prager with whom I had many stimulating discussions regarding data processing.

I thank Dr. Larry Mayer for his support during the initial phase of the project, and for providing the Saint John Harbour data set.

I thank the Science Council of British Columbia for funding me, through the Graduate Research and Engineering Technology (G.R.E.A.T.) scholarship. This excellent program promotes collaborative research efforts with provincially-based companies.

I thank the Natural Science and Engineering Research Council for funding me via post-graduate scholarships.

I thank the staff in the office of the Dean of Graduate Students who were always friendly and helpful.

I thank Dr. Roland Poeckert of the Esquimalt Defense Research Detachment (formerly DREP) for allowing me to use his data set, and Mr. Michel Goguen of Public Works and Government Services Canada for providing me with the Caraquet ISAH-S data set.

Finally, I thank my wife, Stella, for her support and unyielding belief that I would finish, and my children for putting things into perspective.

For David and Kenneth

1 Introduction

The ability to extract information about the ocean bottom has proved elusive since it was first actively pursued in the mid-seventies. Initial attempts by oceanographers were carried out as a modelling of a physical process. However, over time, the task has been re-interpreted as that of a signal processing exercise. Recent advances have resulted from the application of the maturing signal processing techniques coupled with rapidly increasing available processing power.

Currently there are only a few groups working on the problem. Most notable are Larry Mayer's Ocean Mapping group at the University of New Brunswick [1],[2],[3], and the prolific ESMAC program [4],[5],[6],[7],[8],[9],[10]. As well, Douglas Lambert has done some work at the Naval Research Laboratory [11],[12].

The author's own publications on seabed classification include [13],[14],[15],[16], with others submitted for publication.

1.1 Motivation

Information about the ocean bottom has always been important to mankind — for the first several thousand years, the immediate concern was simply shallow water bathymetry and its implications towards navigational safety. But as exploration progressed in the later centuries, systematic surveys began to be conducted, and a demand for new technologies arose.

One of the first methods of surficial bottom classification was to fill a cavity in the bottom of the sounding lead with tallow [17]. This simple approach evolved into a sophis-

ticated variety of coring tubes, snappers and dredges, some of which have present-day incarnations. However, the basic invasive approach remained the same, and still required that the survey vessel come to a full stop prior to taking a sample.

A fundamental change in survey technology occurred with the development of the acoustic echo-sounder in the 1920's. This technique was revolutionary in that it was non-invasive and permitted large-scale surveys to occur.

It was only natural that eventually attempts would be made to apply such a powerful survey tool to the task of bottom classification. In the 1970's electronic hardware became sufficiently inexpensive that the idea of processing the data to extract bottom information characteristics became feasible. The next leap in seabed classification occurred in the mid-1980's when groups started applying recently developed signal processing techniques to data using new low-cost computing facilities.

The current motivation for seabed classification is both military and commercial. The successful use of mines in the Persian Gulf caused alarm among navies, and spawned immediate research efforts into the ability to detect anomalies in the ocean floor, as well as characterizing high-risk areas.

Commercial motivations have been driven largely by shrinking resources. The Department of Public Works and Government Services Canada (and its world-wide counterparts) have found that they can no longer afford inefficient dredging operations, and therefore require more precise information about the ocean bottom.

In the private sector, decreasing world-wide fish stocks have required that increasingly sophisticated technology be employed to locate the schools. Seabed classification

allows the ship captains to better predict good fishing areas through habitat identification.

As well, as world-wide trade increases, new markets are being found for existing shellfish. Little is known about the available habitats and populations of the various species which are believed to represent a billion dollar resource for British Columbia alone.

1.2 Constraints

The project which instigated this thesis was originally conceived by Quester Tangent Corp. of Sidney, BC, as a value-added enhancement to their line of hydrographic survey integration products. As such, it was necessary that any product resulting from the implementation of the research be able to work with the existing installed base of echosounders.

This condition imposed a couple of constraints: first, the data acquisition equipment could not be specifically tailored to best satisfy the processing needs. And second, algorithms had to be robust enough to work for a wide variety of sampling rates, pulse widths, carrier frequencies, beam widths, etc.

Additionally, since the research was conducted almost exclusively by the author, without any project funding, no experimental facilities could be created. In fact, all of the data used in this research was graciously donated by various groups with related interests.

1.3 Support

Support for this thesis came from the Science Council of British Columbia through a Graduate Research and Engineering Technology (G.R.E.A.T.) scholarship. This excellent program promotes collaborative research efforts with provincially-based companies.

Additional financial support came from the Natural Science and Engineering

Research Council through a post-graduate scholarship, and from Dr. R.Lynn Kirlin in the form of a research assistantship, also from NSERC funds.

Technical support was provided by Quester Tangent Corp. and the Ocean Mapping Group at the University of New Brunswick.

Data sets were provided by the Esquimalt Defense Research Detachment (formerly DREP), Public Works and Government Services Canada, and the Ocean Mapping Group.

1.4 Thesis Outline

This thesis is divided into seven chapters.

Chapter 1 (this chapter) provides an introduction to the thesis, including motivation for the work.

Chapter 2 covers the technical background relating to acoustic echosounder returns. In particular, the various phenomena which contribute to (and distort) the received signal are itemized.

Chapter 3 provides a detailed literature survey of bottom classification throughout the last two decades, and reviews how the approaches have changed with advances in computing power and signal processing techniques.

Chapter 4 is the first of two technical chapters dealing with the processing of the acoustic returns. Chapter 4 covers the preprocessing that is required before classical pattern recognition techniques can be applied. Topics discussed include event detection, spatial averaging, stacking, and compensation for various artefacts such as transducer angle, envelope deconvolution, time-scale normalization, spherical spreading and water column attenuation.

Chapter 5 is the second of two technical chapters dealing with the processing of the acoustic returns. Chapter 5 covers the application of pattern recognition techniques such as feature extraction, algorithmic scaling, feature space reduction, classification, and clustering.

Chapter 6 describes some of the results of this thesis. First, the various implementations of the research are mentioned, followed by analysis of three distinct data sets, provided by Public Works and Government Services Canada, the Ocean Mapping Group at UNB, and the Esquimalt Defense Research Detachment (formerly, DREP). The data sets are discussed in context of the topics dealt with in Chapters 4 and 5.

Finally, Chapter 7 presents the conclusions of this research and makes recommendations for further work.

Various appendices are provided, including reference pages for the Pattern Recognition Toolbox for MATLAB.

1.5 Contributions to the Field

The research covered by this thesis contains several novel components, including the following:

- It represents the first systematic approach to processing seabed classification of echo-sounder returns.
- It corrects errors in earlier proposed backscatter models.
- It proposes the use of bathymetric and backscatter angle noise to better simulate real-world echosounder returns.
- It was the first to identify the need for time-scale normalization to correct for depth effects.
- It includes the novel blind deconvolution of impulse functions from envelopes.

- It identifies the effects of spatial averaging and transducer angle on classification.
- It uses a computationally efficient method of feature space reduction.
- It proposes the use of certainty measures as a quantification of the quality of classification.
- It provides a list of the characteristics of successful clustering algorithms and stresses the importance of the membership/statistics duality.
- It modifies the K-means clustering algorithm to support arbitrary statistics and to remove dependency on initial conditions through iterative estimation of the number of clusters.
- It gave rise to the Pattern Recognition Toolbox for MATLAB, a suite of m-files and accompanying documentation that filled a gap in the available tools.
- It was implemented as the very successful QTC VIEW Seabed Classification System.

1.6 Typographic conventions

The following deviations from the default text are use in this thesis.

Object	Typeface	Example
Scalar variables	italic	L_{ik}
Vectors and matrices	bold, italic	$\boldsymbol{C}_i^{-1}$
MATLAB Code	sans-serif	[m,c]

Table 1.1: Typographic conventions

1.7 Trademarks and Copyrights

The following names are used throughout this thesis.

- "QTC", "QTC VIEW", and "ISAH-S" are trademarks of Quester Tangent Corp.
- "MATLAB" is a trademark of The MathWorks, Inc.
- "SPARC" and "SunOS" are trademarks of Sun Microsystems Inc.
- The "Pattern Recognition Toolbox for MATLAB" is copyright by Ahlea Systems Corp.

2 Technical Background

This chapter deals with the problem definition, and reviews some fundamental concepts that are discussed in the reviewed literature.

This thesis only considers seabed classification via acoustic remote sensing. Other methods such as box core sampling, optical or electro-magnetic analysis are beyond the scope of this thesis. In this approach, a mounted transducer on a boat or towed fish emits an acoustic signal. The signal then bounces off the seabed and returns to the transducer where it is recorded and analyzed. Fig 2.1 provides an illustration of the process.

The novel contributions of the research covered in this chapter include a correction to the bottom impulse response proposed by Tan and Mayer [2], the use of time-scale normalization to correct for depth effects [13], and the identification that source signal convolution can degrade results in shallow water [16].

2.1 Seabed Classification Overview

2.1.1 Interpretation

A certain amount of ambiguity surrounds the term "bottom classification". In some of the earlier literature it referred to specific hardware used for analysis of analog signals, whereas more recently the term has become synonymous with the algorithms used to make qualitative assessments about the ocean sediment.

Nonetheless, there are still two prevalent and distinct interpretations of the term "bottom classification" (or more recently, "seabed classification") as it pertains to the qualitative assessment of the ocean bottom sediments.

2.1.1.1 Sub-bottom profiling

The first interpretation is that of a quantitative analysis of the stratified layers of sediments which comprise the bottom. The layers, demarked by different physio-acoustic properties, are analyzed by methods akin to seismic analysis. This approach obviously requires significant sub-bottom penetration of acoustic energy, and hence typically uses low-frequency (3 kHz - 12 kHz) signals. The outputs of this interpretation are quantitative parameters, such as an impedance profile.

The techniques and algorithms for sub-bottom profiling can also be used to extract surficial physical characteristics from which one can infer a *qualitative* classification (see the next section). In particular, the use of increasingly higher frequency echo-sounders will increasingly restrict the returned energy to the water-sediment interface.

Indeed, early attempts at bottom classification all made use of sub-bottom profiling systems, whereas the trend has been for increasingly higher frequency transducers to be used with increasingly advanced signal processing techniques for analysis of surficial characteristics.

Sub-bottom profiling is still an active field, and the name "bottom classification" has persisted as a synonym.

2.1.1.2 Surficial classification

The second interpretation of "bottom classification" is that of the *surficial* composition of the ocean bottom. Although some sub-bottom penetration of acoustic energy is unavoidable (and often useful), the aim is to restrict the returned information to the composition at the water-sediment interface. This is usually ensured by using high-frequency (40 kHz -

200 kHz) signals. The output of this interpretation of bottom classification is a number of qualitatively defined *classes* or *types* (e.g., "gravel", "sand").

The data sets used in surficial classification are typically generated by either side-scan sonar or high-frequency echo-sounders. Side-scan images can be analyzed[1] with the understanding that backscatter strength and image texture provide important information about the bottom type. The advantage of this method is that side-scan sonar easily provides 100% coverage of the ocean floor. The disadvantage is that the only information available for analysis is backscatter profile for a very narrow slice of the bottom, and counter-intuitive images have been identified which contain bright, yet insubstantial, sand, as well as dark, yet effectively impenetrable, mud.

The other source of data is from echo-sounders. In this method, the time-domain acoustic return is analyzed with the understanding that the shape of the return is a result (primarily) of the roughness of the ocean bottom. The advantage to this method is that an echo-sounder is relatively inexpensive, and is easily deployed on any vessel. The disadvantage is that the coverage depends on the survey grid, and assumes homogeneity of bottom types within the acoustic footprint.

In this thesis we focus almost entirely on surficial seabed classification from echo-sounder returns, although the appropriate historical context requires a review of early classification attempts based on sub-bottom systems and techniques.

---

1. A wide variety of image processing techniques are available.

2.2 Echo-sounder Returns

The fundamental operation of an echo-sounder is to transmit an acoustic signal and then record the amplitude of the demodulated pressure wave returned from the bottom. Usually the transmission and reception are done through a single transducer. Between transmission and reception, however, the acoustic signal is modified by several phenomena — some of which exhibit angular and/or depth dependence.

The phenomena which most concern seabed classification are listed in Table 2.1 below. Other phenomena such as salinity/pressure/temperature gradients, multipath reflections, noise, etc., are generally second order effects.

Phenomena	Dependence	
	Angular	Depth
Beam pattern	✓	
Water column attenuation		✓
Spherical spreading		✓
Water column reflectors		
Bottom impulse response	✓	✓
Transducer time-varying gain		✓
Source signal		
Envelope Generation		

Table 2.1: Angular and depth dependence of acoustic phenomena

The relative weights of some of these factors (in particular, water column attenuation, spherical spreading, beam pattern, and backscatter profile) are shown later in Fig 2.3.

2.2.1 Beam pattern

The echosounder transducer is designed to have a particular beam pattern. Generally, the beam width (defined by the span between 3 dB points) will range from 8 to 43 degrees. The specifications for a particular echosounder are generally available from the manufacturer, but can be modelled [2] as

$$B(\phi) = \left[\cos^2(\frac{\phi}{2})\frac{J_1(ka\sin\phi)}{ka\sin\phi}\right]^2 \tag{2.1}$$

where B is the intensity of the beam, a is the radius of a circular transducer, $k = \omega/c$ is the wavenumber in water, ϕ is the grazing angle, and $J_1(\bullet)$ is the first Bessel function.

It is assumed that the beam pattern of the transducer is identical when both transmitting and receiving.

While the beam pattern undoubtedly affects the echosounder return, it is not necessary to correct for this effect as the angular dependence can be absorbed into the backscatter intensity profile of each sediment type. Compensation would only be required if attempting to classify the returns of one echosounder using statistics gathered with a different echosounder.

2.2.2 Water column attenuation

Sound is naturally absorbed as it passes through a medium. The amount of attenuation per metre is given by

$$A(l) = e^{-\alpha l} \tag{2.2}$$

where l is the path length in the medium and the constant α is a function of water temperature, salinity, and pressure, as well as the frequency of the sound wave. For seawater and

a frequency of 210 kHz, the attenuation is between 80 dB to 90 db per kilometre, giving α an approximate value of 0.02 m^{-1}. Thus at water depths of approximately 25 m, the incident wave has been attenuated by 2.2 dB, whereas at a commercial fisheries depths of 600 m, the attenuation is 52 dB.

Note that there is identical attenuation for both the return and outgoing paths.

2.2.3 Spherical spreading

As an acoustic wavefront propagates through the water, it expands spherically[1]. Therefore, the intensity incident on an elemental area is inversely proportional to the square of the radius, this relationship being denoted by $L(l)$

$$L(l) = \frac{1}{l^2} \tag{2.3}$$

where it is assumed that the source acoustic intensity I_0 is measured at one unit distant from the transducer.

Similarly, a single scatterer (i.e., an elemental area) will act as a point source, and therefore energy returned from it will experience the identical radius-squared attenuation, resulting in a two-way path fourth-power inverse proportionality. However, the received signal is derived from the entire area of insonification, which is proportional to the square of the depth. As a result, the total energy in the return is inversely proportional to the *square* of the depth.

---

1. Assuming no distortions due to gradients in the speed of sound due to pressure, temperature, etc.

2.2.4 Water column reflectors

Temperature inversions, currents, suspended sediments, and biology can all produce acoustic returns unrelated to the actual bottom. For the most part, these contributions to the acoustic return can be eliminated in that they are bathymetrically (and hence, temporarily) separated from the bottom.

However, some bottom-dwelling biology[1] as well as some suspended sediments may occur near enough to the water-sediment interface as to interfere with a portion of the acoustic return. The effect may vary from simply complicating the bottom picking (i.e., event detection) to actually being embedded in the return.

Apart from careful bottom picking, these effects may be difficult to correct.

2.2.5 Bottom impulse response

The bottom impulse response received by the transducer arises from three sources: namely, reflection, backscatter, and volume reverberation. That is,

$$I_{rcvd} = I_{reflect} + I_{scatter} + I_{reverb} \tag{2.4}$$

however, in models derived by Jackson and Nesbitt [18], Mourard and Jackson [20], Jackson and Briggs [19], and Tan and Mayer [2], the component due to reflection is negligible — leaving only the backscatter and volume reverberation sources illustrated in Fig 2.1.

2.2.5.1 Backscatter

We can model the backscatter component of the signal as follows: assume we have an impulse source of energy I_0, with an angular dependence determined by the transducer

1. Generally fauna, but may include flora in shallow water.

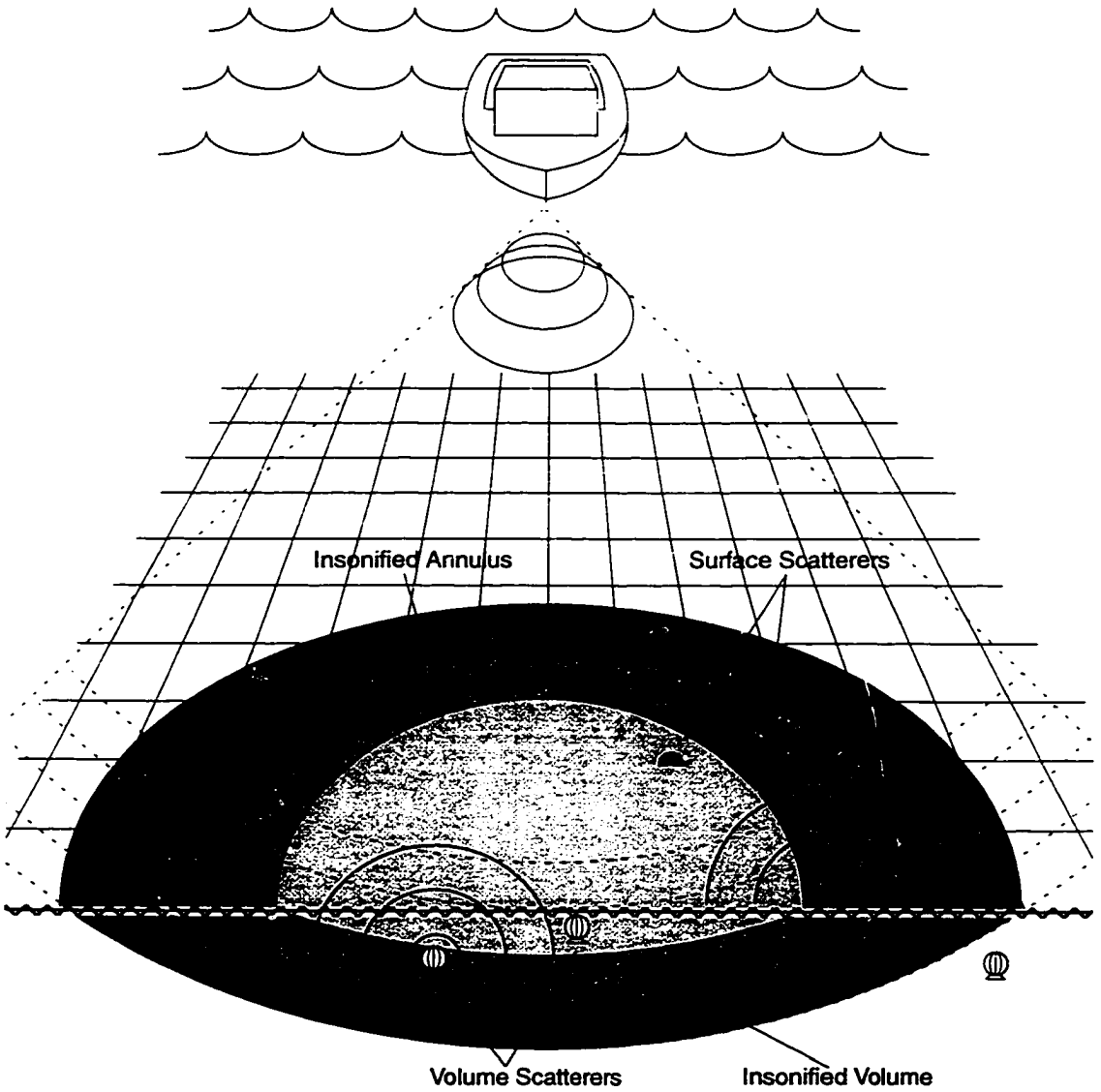


Figure 2.1: Sources of backscatter energy

beam pattern $B(\phi)$. The signal undergoes attenuation due to absorption in the water column $A(l)$, and spherical spreading $L(l)$. Thus, the energy incident at an elemental area da is given by

$$dI_{incident}(t) = I_0 A(l) L(l) B(\phi) da \tag{2.5}$$

Furthermore we note that a portion of the energy is backscattered according to $f(\phi)$

— a function of both bottom type (incorporating roughness and reflection coefficient) and frequency[1]. And, as mentioned in 2.2.2 and 2.2.3, the scattered energy undergoes the same attenuations as the source energy over the return path and is subject to identical transmit and receive beam patterns. Therefore, the received elemental energy from an elemental area on the annulus of insonification is given by

$$dI_{rcvd} = I_0 A^2(l) L^2(l) B^2(\phi) f(\phi) da \tag{2.6}$$

The total energy backscattered at a given time t in response to a source signal $s(t)$ (scaled by a factor I_0) is the result of integrating the elemental responses of Eq. (2.6) over the area of insonification

$$I_{rcvd}(t) = I_0 A^2(l) L^2(l) \iint B^2(\phi) f(\phi) s(t-\tau) da \tag{2.7}$$

The integrating variable τ is directly related to the elemental area da, defined as

$$da = r \cdot d\theta \cdot dr \tag{2.8}$$

where r is the radius of the annulus and θ is the azimuth angle.

Assuming a circular insonification footprint, we can simplify Eq. (2.7) as follows:

$$I_{rcvd}(t) = 2\pi I_0 A^2(l) L^2(l) \int B^2(\phi) f(\phi) s(t-\tau) r dr \tag{2.9}$$

And since we are interested in the response as a function of time, it is required to define the integral in terms of elapsed time from the first return. This can be done by noting that the time before an echo is received is given by

$$t = 2l/c \tag{2.10}$$

---

1. Backscatter models of varying complexity have been advanced by Jackson *et al* [18],[19],[20], [21],[22], Tan and Mayer [2], McDaniel and Gorman [23], and Poulinquen and Lurton [24].

where l is the distance from the transducer to the scatterer and c represents an appropriate value for the speed of sound in water.

Furthermore, if we define elapsed time as

$$\tau = t - t_h = \frac{2}{c}(l - h) \tag{2.11}$$

or conversely,

$$l = \frac{\tau c}{2} + h \tag{2.12}$$

and use the equality $l^2 = r^2 + h^2$ (as illustrated in Fig 2.2 below), we can arrive at an expression for the radius in terms of elapsed time

$$r^2 = \frac{c^2\tau^2}{4} + c\tau h \tag{2.13}$$

and therefore

$$2rdr = \left(\frac{c^2\tau}{2} + ch\right)d\tau \tag{2.14}$$

This allows us to rewrite Eq. (2.9) as follows

$$I_{rcvd}(t) = \pi I_0 A^2(l) L^2(l) \int B^2(\phi) f(\phi) s(t-\tau)\left(\frac{c^2\tau}{2} + ch\right)d\tau \tag{2.15}$$

And finally, since it follows from the geometry of Fig 2.2 that

$$l \cdot \cos\phi = h \tag{2.16}$$

we can express the incident angle in terms of depth and elapsed time

$$\phi = \operatorname{acos}\left(\frac{1}{\frac{\tau c}{2h} + 1}\right) \tag{2.17}$$

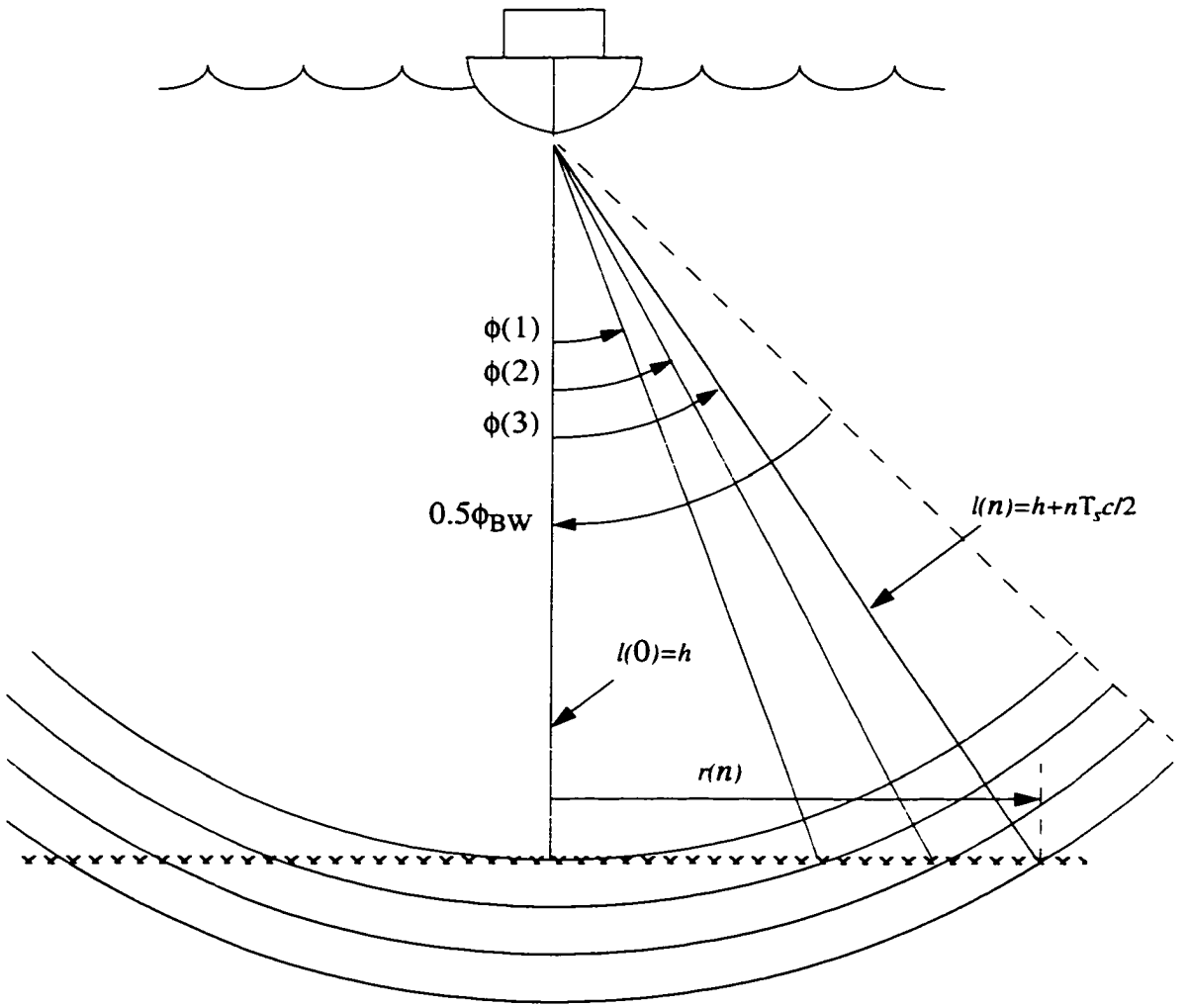


Figure 2.2: Time structure of backscattered echoes

which allows us to rewrite Eq. (2.15) strictly in terms of those variables.

Note that the fundamental difference between Eq. (2.15) and Tan and Mayer's model is that Eq. (2.15) implements a convolution between the impulse response and the source signal, whereas Tan and Mayer implement a correlation. This is made clear by gathering terms such that

$$\begin{aligned} G(l) &= \pi I_0 A^2(l) L^2(l) \\ h(\tau) &= B^2(\phi) f(\phi) \left(\frac{c^2 \tau}{2} + ch \right) \end{aligned} \tag{2.18}$$

yielding the final expression for the bottom response

$$I_{rcvd}(t) = G(\frac{tc}{2} + h) \int h(\tau) s(t - \tau) d\tau \tag{2.19}$$

Since we are using sampled data, we can only resolve times down to integer multiples of T_s, the inverse of the sampling rate f_s

$$m = \left\lfloor \frac{t}{T_s} \right\rfloor \tag{2.20}$$

Furthermore, at the water-sediment interface at depth $l = h$, the first observed return occurs at

$$t_h = 2\frac{h}{c} \cong m_h T_s \tag{2.21}$$

From this reference point, we can define an offset (expressed in the number of samples), given by

$$n = m - m_h \cong \frac{2}{cT_s}(l - h) \tag{2.22}$$

Thus we can express the incident angle ϕ in terms of depth h and sample n

$$\phi = \operatorname{acos}\left(\frac{1}{\frac{nT_s c}{2h} + 1} \right) \tag{2.23}$$

It is fundamentally important to recognize that Eq. (2.22) and Eq. (2.23) imply a dilation of the return that is linear with depth. Solving either equation for n, as shown

below, clearly illustrates that the offset n at which the wavefront is incident to the bottom with angle ϕ is linearly dependent on the depth h.

$$n = \frac{2h}{T_s c}\left(\frac{1-\cos\phi}{\cos\phi}\right) \tag{2.24}$$

Finally, two things should be noted about Eq. (2.19). First, the gain factor $G(l)$ of Eq. (2.19) is in theory dependent on the time-varying length of the acoustic path. However, in reality the losses due to spherical spreading and water column attenuation change very little over the duration of the return when compared with the angularly dependent backscatter profile and beam pattern. This is illustrated below in Fig 2.3, in which each of the factors discussed above are normalized to a value of unity (the plots assume 15 m depth, 8° beam-width, and a backscatter profile following Jackson and Briggs [19]). Therefore, the gain factor can be simplified such that it is dependent solely on the depth, $G(h)$, as opposed to the time-varying length of the acoustic path.

The second item to note about Eq. (2.19) is that it does not take noise into account. In addition to the obvious additive water column noise, small variations in the bathymetry due to ripples, large scatterers, etc. will change the distribution of the acoustic path lengths. Similarly, a noise component is introduced to the backscatter angles used to calculate the backscatter profile by small scale features and scatterer orientation. These components are not additive, and must be considered part of the bottom's impulse response, $h(t)$. In fact, no model can generate realistic data without considering these noise sources. To illustrate, Fig 2.4 shows an actual echosounder return along with the output of Eq. (2.19) with and without angular noise, using the backscatter strength model developed

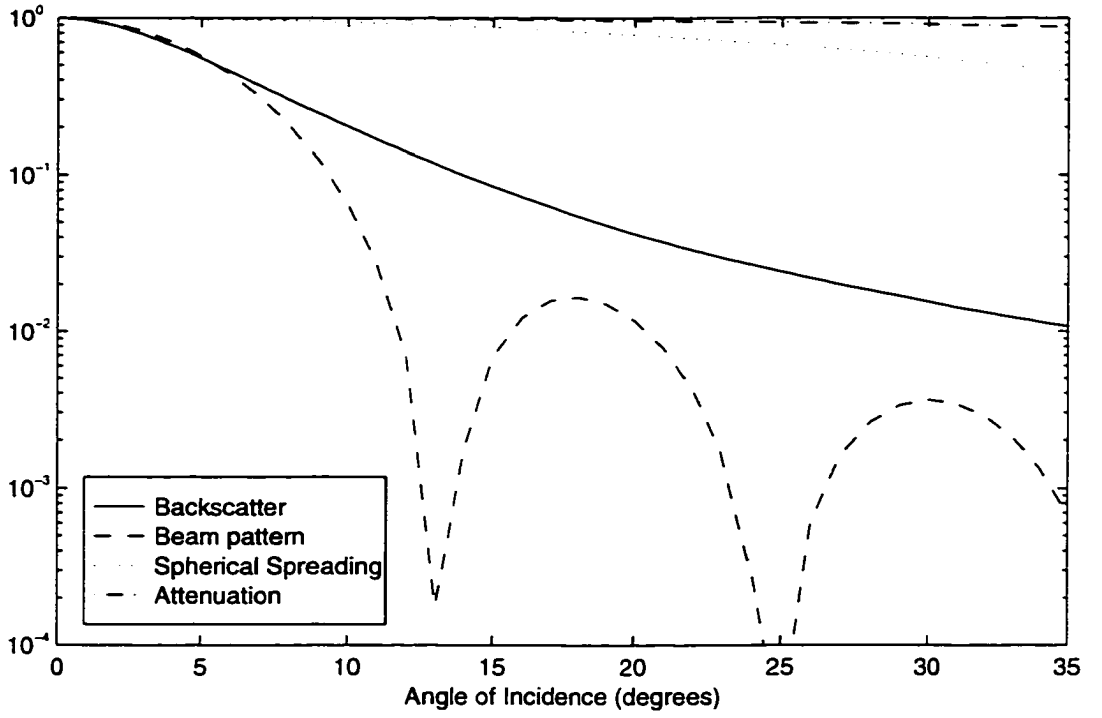


Figure 2.3: Normalized gain factors

by Jackson and Briggs [19]. Note the source signal is the same length (100 μs) as the sampling period (10 kHz), so the effects of the convolution may be considered negligible.

Without the angular noise, the return is simply too abrupt. The main difference between the noisy model output and the real return is the amplitude and width of the beam pattern side lobe (which occurs around sample 275). This difference is most likely due to the simplistic beam pattern model described in Eq. (2.1).

2.2.5.2 Volume reverberation

Despite the high-frequency carriers used by many echo-sounders, there will always be some bottom penetration. The response due to this penetration can be modelled with a volumetric integral in a derivation analogous to that of the backscatter component, and is gen-

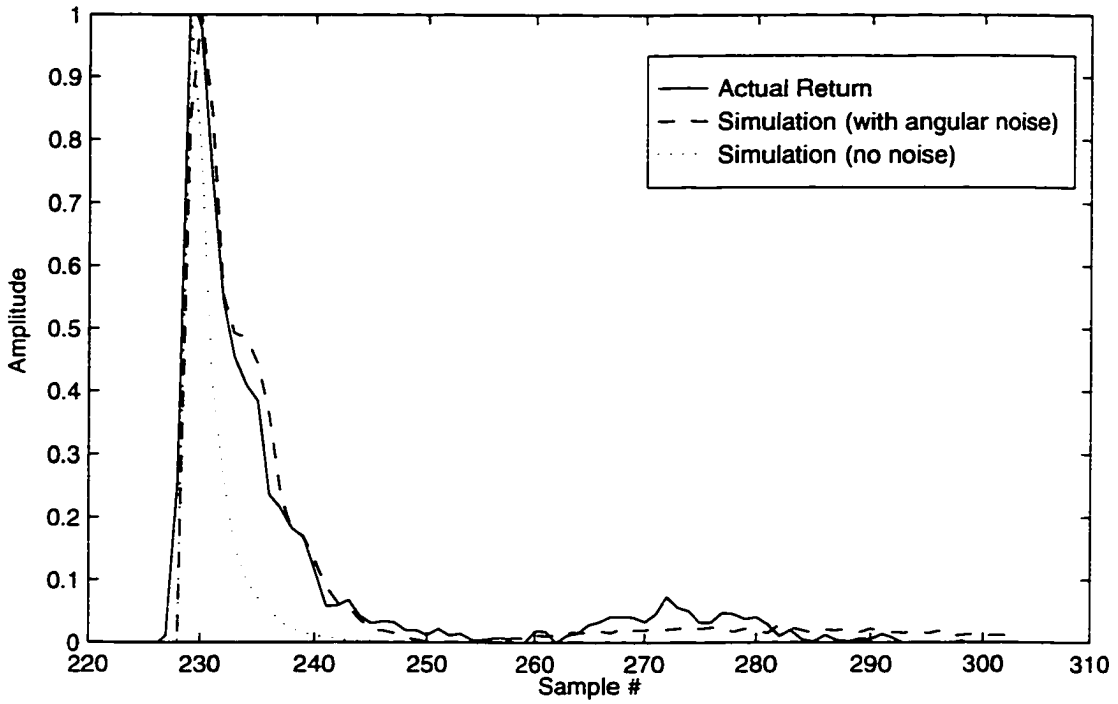


Figure 2.4: Actual and simulated echosounder returns

erally referred to as volume reverberation (not to be confused with resonance or multiple echoes).

The interesting point however, is that the different models proposed by Jackson, Poulinquen, and Tan (and associates') suggest that the volume reverberation for some common bottom types can last on the order of twice as long as the backscatter return. In other words, the volume reverberation can contribute significantly to the tail of the return.

This is in direct contrast with the Roxann™ system put forth by Chivers, Emerson and Burns [25], which asserts that the unwanted volume reverberation occurs primarily at the start of the first return, and therefore can be eliminated by not including (i.e., gating) the initial samples in their "E1" integration. The basis for this assertion is an empirical model derivation for which the authors include the following caveat:

> [The model's] theoretical poverty is matched by an experimental one. It thus becomes extremely hard to estimate the time after the start of the first normal reflection by which the sub-bottom reverberation will have decayed to a negligible level.

Furthermore, their fundamental assertion that the volume reverberation is unwanted can be argued — it was found in one case that the contributions due to volume reverberation were in fact beneficial in distinguishing certain specific bottom conditions (e.g., a thin-veneer of mud over gravel versus pure mud).

2.2.6 Source signal

It is important to realize however, that Eq. (2.19) describes the *impulse response* of the ocean bottom. Typically the source ping $s(t)$ of a bathymetric echosounder is a sinusoidal burst at a carrier frequency in the range of 40 kHz to 208 kHz. The pulse width varies with the echosounder model and can be between 60 μs and 2000 μs.

In a proposed system [26] a chirp is used instead of a ping.

As a result of using non-instantaneous excitation, the complete bottom return is defined as the convolution of the source ping and the impulse response of the ocean bottom.

$$I_{ping}(n) = I_0 I_{rcvd}(n) * s(n) \tag{2.25}$$

This convolution results in a smearing of the signal. The subsequent problem of deconvolution is discussed in greater detail in Chapter 4, but it will be noted here that due to the linear dilation of the bottom impulse response with depth, as described by Eq. (2.24), the distortion due to convolution becomes negligible as the bottom impulse becomes much longer than the source ping. The corollary is that convolution is a problem

in shallow water.

2.2.7 Transducer time-varying gain

Because of the depth-squared attenuation resulting from spherical spreading, most echosounders employ some form of time-varying gain (TVG) to ensure that the received signal can be rendered on a strip chart (on older systems), or is within the dynamic range of the digitizer.

Unfortunately, many older systems were only designed for strip chart output, with digital data acquisition being accomplished only through later invasive patches to the electronics of the echosounder. As a result, absolute numerical precision was not considered important, allowing inexpensive but non-deterministic time-varying gain algorithms to be used[1].

Care must be taken to ensure that classification methods that operate on data collected with such older (yet still common) equipment must be independent of absolute amplitude, since it may be impossible to compensate for the TVG.

2.2.8 Envelope generation

As mentioned above, many echosounders used strip charts as their sole form of output. And as a typical printing stylus cannot exhibit both positive and negative displacements, typically envelopes have been generated (in a variety of ways) for the received signals.

Unfortunately, the fact that the digitized signal is an envelope somewhat compli-

1. For example, the Navitronics swath system continuously increases the gain until the received signal exceeds a pre-set threshold (indicating the leading edge of the first return). The gain is then held constant until the return decays below a second threshold, whereupon the gain resumes increasing for the purpose of detecting the second echo. Navitronics echosounders are used extensively by the Department of Public Works and Government Services Canada.

cates processing of the data. In Chapter 4, we will employ the Hilbert transform to generate numerically tractable envelopes for the purpose of deconvolution of impulse functions from echosounder envelopes.

2.3 Summary

Seabed classification is interpreted as the qualitative assessment of the surficial composition of the ocean bottom. In this thesis, it is done using acoustic echosounder returns.

A recorded echosounder waveform consists primarily of incoherent backscatter from the ocean floor. The distribution of energy versus time is determined by the backscatter profile of the bottom (i.e., the bottom impulse response) which is windowed by transducer beam pattern. The bottom impulse response dilates with depth and may be distorted by convolution with a non-impulse source signal. In most echosounders, it is the envelope of the received signal that is recorded.

The requirement of time-scale normalization to correct for depth effects is shown by illustrating the dependence of time offset on the depth as well as incident angle (see Eq. (2.24)).

As well, an error in the model proposed by Tan and Mayer is corrected, yielding a simple backscatter model given by Eq. (2.19) and Eq. (2.20). It is also discussed that any backscatter model used to simulate source pings should contain noise contributions in both bathymetry and scattering angles.

Finally, it is shown that the convolution evident in the backscatter model may result in signal degradation in situations where the transmit pulse length is significant compared to the bottom impulse response.

3 Literature Survey

3.1 Historical Development

While systematic bathymetric surveys have been conducted for four hundred years, little changed until the introduction of the echo-sounder in the late 1920's. While individuals may have approached the problem of automated echo-sounder bottom classification as early as the sixties, most of the early literature extends back only to the mid- and late-seventies.

The papers are grouped into a number of sections according to the underlying methodology.

3.1.1 The coherence era

Early papers tended to focus on analysis of the raw data and of ping-to-ping coherence measures. This was the case up to the mid-eighties.

3.1.1.1 Milligan *et al*, (1978)

The first important reference is "Statistical grouping of acoustic reflection profiles" by Milligan, LeBlanc, and Middleton [27].

The basic premise is that one can analyze a set of data, and extract the ensemble statistics, without resorting to *a priori* models. This represented a "black-box" approach which was well ahead of its time.

A covariance matrix is generated for the ensemble, which is then transformed via eigenanalysis to a number of orthogonal vectors. These vectors represent the *principal components* of the data set. A reduced number of principal components can be used to generate a minimal-dimension reconstruction of a received signal. The weights associated

with the most significant n principal components (in this case, $n = 2$) comprise a reduced vector. Furthermore it is acknowledged that principal components may not "have a readily obvious physical interpretation". However, they do go on to identify the first principal component as the coherent reflection.

The concept of analyzing features having no obvious physical interpretation did not reappear for another several years. Furthermore, feature reduction via principal component analysis was not re-examined until Caughey *et al* [13], although numerical evaluation of feature suitability was done in Kavli *et al* [9]. However, in Milligan's work, the principal components were derived directly from the raw (unrectified) data, instead of from arbitrary feature vectors.

Also in Milligan *et al* is a discussion of clustering, and a multi-stage approach to classification – one pass to acquire the base statistics and perform clustering, and subsequent passes to classify new or existing data. Furthermore, they attempted blind clustering (i.e.,with a lack of *a priori* information about the number or statistics of the final classes). Again, with the exception of Caughey, all other work has assumed the availability of ground-truthed data sets.

The major shortfall of this work is that it occurred in the seventies, prior to modern computing facilities. Much of the description of clustering revolves around the premise that two Megabytes of memory exceeds the storage capacity of all but the largest computers. No description of the classification process is given, but the description of the feature space transformation used in the clustering routine suggests that it is a statistical likelihood approach similar to that of a Bayesian classifier.

3.1.1.2 Dunsiger, Cochrane and Vetter, (1981)

In their 1981 article "Seabed characterization from broad-band acoustic echosounding with scattering models" [28], Dunsiger *et al* note that "certain degrees of roughness may characterize particular sediment types".

Despite references to Milligan *et al* and the further development of some of the concepts, many of the concepts in the original article are ignored.

Dunsiger *et al* elaborate on the concept of coherence, and use the reflection coherence function $\gamma_R(f)$, the scattering coherence function $\gamma_S(f)$, and their non-linear combination

$$f) = \gamma_R(f) - \gamma_R(f)\gamma_S(f) + \gamma_S(\tag{3.1}$$

as features. The argument is that hard ocean bottoms generate coherent reflections whereas rough bottoms return incoherent scattering contributions.

Their method measures the ping-to-ping coherence of Huntec data, with the assumption that the distribution of individual scatterers changes as the footprint moves, resulting in little or no coherence. On the other hand, the reflection energy is only affected by gross morphological variations. Hence, there is a measure of hardness and roughness. However, they acknowledge that at "lower frequencies significant transmitted energy may be returned... from coherent subbottom acoustic horizons and from random subbottom impedance inhomogeneities. For certain sediment types, these subbottom energy contributions may be comparable to the incoherent scattering contributions from the water-sediment interface."

It should be remembered that this paper deals with *characterization*, and hence no

attempt at describing a classification algorithm is given.

3.1.1.3 Pace and Ceen, (1982)

In their article entitled "Seabed classification using the backscattering normally incident broadband acoustic pulses" [29], Pace and Ceen continue with the concept of coherence measures with a self-described "modest" experiment in which a survey vessel drifted in calm, sheltered seas over known bottom types. Furthermore, the transducer was kept no more than two metres from the bottom (i.e., very close).

They define the return signal as consisting of a coherent part, and an incoherent part. One contribution in their work is the use of rectified (i.e., the envelopes of) return signals.

They also assert that "if the received signal has a 'tail' and is thereby of significantly longer duration than the transmitted waveform, such a tail may be attributed to the incoherent component". However, in Pace and Ceen, since the depth is so shallow, the incoherent tail is on the same order as the transmitted waveform, and can be considered an approximation of the convolution of the incident waveform and the probability distribution of the interface heights[1]. Note that the towfish vertical position varies from one to two metres, but this not taken into account.

Another deviation from the normal operating configuration is the use of parametric arrays, in which a single transducer simultaneously radiates at two frequencies. The destructive interference yields a "self-demodulating" Gaussian modified carrier.

One of the most important contributions of this paper is the introduction of

1. Source signal convolution is discussed in Section 2.2.6

"shape" of the return signal envelopes as an important defining characteristic (i.e., feature), and the use of quantiles (specifically, quartiles) to characterize shape. The authors derive a cumulative distribution of time widths at 20% of peak amplitude of the return signal, and then assert that use of the first or fourth quartiles of this distribution will provide a feature having 1-D linear separability. No numerical results are given however.

This article examines bottom classification from a systems perspective, with the object of designing electronics to accomplish the task.

3.1.1.4 Cochrane and Dunsiger, (1983)

In the article "Remote acoustic classification of marine sediments with the application to offshore Newfoundland" [30], they continue their earlier work ([28]), which involved characterization, with a focus on a robust classification. In particular, a precise classification scheme is defined as a sequence of well-defined processing steps. One of the "practical considerations" for this automated classification is "ready adaptability to new formations". It is not clear that they achieve this goal, but its importance in a successful system was formally recognized.

As with their earlier paper, the primary feature is the coherence function $\gamma(f)$, estimated at two frequencies, and the magnitude of a new windowed sub-bottom coherence function $Q_{41}(f)$, evaluated at a single frequency.

Another couple of firsts for this article is the use of scatter plots for examining the separability with two features, as well as the use of classification in three-space. However, classification involves partitioning three-space into rectangular non-overlapping subspaces, as opposed to using a probabilistic approach. Because all classification sub-spaces

are finite, this method allows for identification of outliers.

3.1.1.5 Orlowski, (1984)

In the article "Application of multiple echoes energy measurements" [31], reflection equations are derived from the model of the superposition of a coherent and incoherent return.

The system is intended for deeper water (>50 m), and involves the integration of the primary and secondary echoes. These concepts are described in better detail in Chivers *et al* [25].

There are quite a number of conditions regarding the limits of operability. Of note is the description of some of the statistics as being Gaussian.

3.1.2 The recent era

Whereas the articles described in the previous section all dealt to some extent with coherence functions, there was a trend in the mid-eighties towards using spectral and other information. This coincides in part with the migration of "advanced" signal processing techniques into the general scientific and engineering community, and with the availability of inexpensive workstation-based computing power.

References from 1993 onward are considered part of the "current" research efforts, and are not considered in this historical review.

3.1.2.1 Reut, Pace and Heaton, (1985)

One early example of the new approach is described in a letter to Nature regarding "Computer classification of sea beds by sonar" [32]. The specification of "computer classifica-

tion" is notable since most earlier articles, including Pace and Ceen [29], had envisaged special hardware doing much of the acquisition, conditioning, and processing of the data.

In this brief letter, the cepstrum is introduced as a novel approach to analysis of sonar data. Two values are regressed from the cepstrum, and used as features. However, examining the scatter plot showing the separability of the classes, it appears that the two features are extremely highly correlated and that without arbitrary decision boundaries, many of the classes are effectively inseparable (especially, assuming classes having Gaussian distributions).

3.1.2.2 Jackson and Nesbitt, (1988)

Also occurring in mid- to late-eighties was the introduction of backscatter models (presented in simplified form in Section 2.2.5.1). In their paper entitled "Bottom classification using backscattering at vertical incidence" [18], the authors present examples in which multi-variate backscatter model *parameters*, as determined through advanced curve-fitting optimization algorithms, are used as features. This represents the first use of inverse modelling in the field, but is perhaps not practical due to the computation requirements of inverse modelling.

Jackson and others proposed successively more advanced backscatter models over the next several years.

3.1.2.3 Chivers, Emerson and Burns, (1990)

One of the more recent important articles is "New acoustic processing for underway surveying" [25], in which the Roxann™ system is described.

One of the unique aspects of this system is that it is claimed that the system derived not from oceanographers' theoretical models, but from empirical observations of how experienced fishermen used their echo-sounder displays. The method is also fundamentally similar to Orlowski [31], which was included as a reference.

The logic proceeds as follows: a bottom type can be identified by its hardness and its roughness. Rather than trying to characterize the parameters of a mathematical model of a physical process, it uses two fundamental axioms. The first is that rough bottoms leads to long 'tails' in the first returns, and that the roughness can be quantified by integrating the energy in the tail. The second is that hard bottoms are much more likely to have strong second echoes (i.e., the initial bottom return is subsequently reflected off the ocean surface and then off the bottom again), and that hardness can be quantified by integrating the energy in the second return. The two integrations are referred to as "E1" and "E2" respectively.

The Roxann system has become quite widespread, with varying reports of success. One problem is that no conditioning is performed to account for depth dependent signal dilation as described in Section 2.2.5. The situation could be handled, however, by judiciously selecting the gating parameters used to generate the E1 parameter; however it is unclear if this is done, or if it is done correctly. Chivers *et al* describe the first gating parameter as being used to filter out unwanted volume reverberation effects, which as described in Section 2.2.5.2, operates contrary to the body of literature. As this system is an empirically-derived system, there is not much doubt that the gating parameters are important – it is the rationalization which is suspect.

Some inarguable problems with the system are that feature space (defined by the Cartesian E1-E2 plane) is divided into rectangular boundaries. This allows for easy classification, but is statistically less accurate than quadratic boundaries between Gaussian clusters.

Furthermore, the system absolutely requires the presence of the second echo. In certain circumstances this second echo may not be observable, or depending on the depth and the recording system, may not be recorded.

A final problem is that much of the processing is done in analog hardware. This makes the system unable to quickly respond to new advances.

3.1.2.4 Poulinquen and Lurton, (1992)

In their conference paper "Sea-bed identification using echo sounder signals" [24], Poulinquen and Lurton describe a method of classification based on a backscatter model. A series of curves are derived for several different bottom types.

Input pings are then processed, and compared to the family of curves, and are identified with the closest match.

Although there is no feature extraction in this method, this paper is important in that its contribution is the recognition of the importance of signal conditioning. Along with picking and packaging, the signal is normalized with respect to amplitude. This rejection of the importance of absolute amplitude for the sake of shape comparison is contrary to the entire calibrative philosophy of most earlier work. In addition to normalizing, key contributions are that the signal is redefined as a running integral and averaged. While averaging was suggested in the coherence era, the concept of conditioning prior to averaging was

original, as was the processing sequence which contained classification as a minor aspect.

The rationale behind the cumulative summing is that the noise component of the signal will also be summed, and due to the uncorrelated nature of noise, the SNR will increase with time.

Little was said of the acquisition environment, so it is not certain why depth normalization was not described. One explanation is that a constant altitude towfish was used.

3.1.2.5 LeBlanc *et al*, (1992)

The article "Marine sediment classification using the chirp sonar" [26] describes the use of a chirp sonar for sediment classification. Even though it is concerned with sub-bottom profiling, it raises a number of interesting issues.

The first is the use of a fully digital system, including the support for the associated digital signal processing.

Additionally, this paper is one of the first to discuss the deconvolution of the source signal from the return signal (see Section 4.3.2). The authors note that one advantage of a compressed chirp signal is that in order to achieve a similar SNR, "a conventional pulse sonar would have to operate at a peak power of 100 times larger than the chirp pulse". This allows the authors to operate the sonar such that no ringing occurs in the transmitter, permitting accurate deconvolution. However, to accomplish this, the system is carefully calibrated, and uses separate transmitters and receivers.

The method of classification is fundamentally one of inverse modelling. The authors use non-linear curve-fitting techniques to arrive at density, porosity, compressibility, and grain size estimates. One can then use these parameters to match against those of

pre-defined bottom types.

3.2 Current Research

This section covers references dating from 1993 to the present. Contrary to the chronologically-ordered historical review, these references are organized by approach.

Most of the methods are concerned with feature extraction, although some relate to classification.

Some references discuss multiple methods and may be listed more than once.

3.2.1 Parametric modelling

This approach involves the determination of model parameters based on the received signal, and subsequent classification based on the parameters' values. Key to this concept is a model. To illustrate, spectral energy coefficients are not model parameters.

Identification can be explicit, through optimization techniques, or implicit (e.g., a least squares fit identifies a curve arising from a particular set of coefficients).

Implicit identification contains an element of vector quantization. For example, Poulinquen and Lurton [24] match signals to seven curves, each of which was generated by a model having a dozen or so parameters. Matching against one curve infers selection of a particular set of coefficients. Furthermore, only the seven specified distinct sets can occur.

3.2.1.1 Kavli, Carlin and Madsen, (1993)

In their paper "Seabed classification using artificial neural networks and other non-parametric methods"[8], the authors compare a variety of methods to a Bayesian classifier.

One of the methods tried used a polynomial regression model. This method differs from neural networks in that it is very fast and always finds a global minimum of the error function. The performance was not as good as the Bayesian classifier, being particularly poor on a test set independent of the training set.

3.2.2 Scale- and frequency-based methods

This refers primarily to wavelet and spectral methods, but can also apply to related methods (e.g., cepstral).

3.2.2.1 Milvang *et al*, (1993)

The paper "Feature extraction from backscatter sonar data" [10] proposes a number of feature extraction methods for a multi-beam echo-sounder.

One of the approaches is based on a normalized log-power spectrum proposed by Pace and Gao [33]. It was found that the Mahalanobis[1] distance increased if the data set was median filtered before processing. This method provided good separation.

3.2.2.2 Caughey *et al*, (1994)

In a contractor's report entitled "Bottom Classification using Single-Frequency Echo Sounders" [13] prepared for the Defense Research Establishment Pacific, the author uses a wavelet packet tree energy decomposition as a set of features. Principal component analysis is used to identify and extract the information-bearing coefficients.

1. The Mahalanobis distance, given by $(x\text{-}\mu)^T C^{-1}(x\text{-}\mu)$, is a measure of statistical distance. More specifically, the Mahalanobis distance corresponds to the square of the number of standard deviations between a point and the mean of a Gaussian distribution (defined by μ and C). If one makes the often dubious assumption of two clusters having identical class statistics, the Mahalanobis distance can be used to characterize the separation between the clusters. Note that the Mahalanobis distance equals Euclidean (or Cartesian) distance only if the covariance equals a scaled identity matrix (see Section 5.3.2.1 for more information)

Additionally, Fourier spectral energy coefficients are used in conjunction with the wavelet packet tree energy coefficients.

3.2.3 Neural networks

Neural networks represent highly non-linear filters with a number of attractive features. First, they can divide a feature space into arbitrarily complex partitions. Second, they can perform dynamic modelling of systems of unknown order. Third, they can, in theory, adapt to new patterns.

However, there are a number of disadvantages. Training a neural net (adapting the "synaptic" weights to produce the desired output given a particular input) is extremely difficult for all but the simplest patterns. While most networks can be shown to asymptotically converge, it is impossible to determine the length of time required, or if the results will be correct. Unsupervised learning does not really exist — networks can be trained in an unsupervised mode, but unless there is a hardwired structure to the network such as in a Kohonen self-organizing map (KSOM)[1], there is little to guarantee that the network will learn anything *useful*. Second, the ability to adapt is somewhat limited. As with all adaptive filters, there is a trade off between the error and the rate of adaptability.

Neural networks are good non-linear adaptive filters, but expecting much more will often lead to disappointment.

---

1. Essentially a vector quantization (VQ) codebook generator.

3.2.3.1 McCleave, Owens and Ingles, (1992)

In the article "Analyzing depth sounder signals with artificial neural networks" [34], the authors use the opportunity to learn about neural networks by training a special ANN chip to discriminate between the trivial case of "hard rock" and "fluff".

3.2.3.2 Kavli, Carlin and Madsen, (1993)

In their paper "Seabed classification using artificial neural networks and other non-parametric methods" [8], the authors compare a variety of methods to a Bayesian classifier.

To the disappointment of the authors, although the multi-layered perceptron gave higher classification rates on the training data, it was less consistent and less accurate on new and independent data — a persistent problem with neural networks.

The authors also tried radial basis function (RBF) networks, which use radial transfer functions (in this case, Gaussian functions) in the hidden layer. This method did not perform as well the MLP (multi-layer perceptron[1]).

3.2.3.3 Alexandrou and Pantzartzis, (1993)

The article "A methodology for acoustic seafloor classification" [35] discusses a purely simulated environment in which the reverberation probability density functions are parameterized (kurtosis estimates), and then classified by either a multi-layer perceptron (MLP), or a Kohonen self-organizing map. The simulated return is via a point scatterer model proposed by Ol'vsheski [36]. According to the authors, the perceptron achieved classification rates of 92%, and the self-organizing map correctly identified four unique bottom types.

---

1. The most common form of neural network. See Lippmann [46] for an excellent (if somewhat dated) introduction to common neural network architectures.

The use of Kohonen maps for cluster identification is an area of good potential. Note that the authors didn't expect the neural nets to identify features, but instead used calculated features as network inputs. However, if the features are normally distributed, then a Bayesian (i.e., classical) classifier can be shown to be optimal – the best a neural network can do is (expensively) imitate the Bayesian classifier.

The major reservation is that the simulated data is not similar to real-world echo-sounder data, which would have a greater angular span, and would comprise sometimes indistinct clusters.

3.2.3.4 Zerr, Maillard and Gueriot, (1994)

This paper, "Sea-floor classification by neural hybrid system" [37] is intended for side-scan sonar, but there are some interesting and potentially applicable concepts.

In particular, the authors used a sequence of simple neural networks, of which the first is a Kohonen self-organizing map to spatially distribute features, followed by the supervised MLP (multi-layer perceptron) for classification. They claim to run the MLP in constant learning mode, but this will eventually stagnate.

3.2.4 Shape analysis

Most of the methods discussed in this section contain some element of shape analysis.

3.2.4.1 Mayer, Clarke and Wells, (1993)

There is a brief mention of the use of higher order statistics in their paper, "A multi-faceted acoustic ground-truthing experiment in the Bay of Fundy" [1].

3.2.4.2 Milvang *et al*, (1993)

The paper "Feature extraction from backscatter sonar data" [10] proposes a number of feature extraction methods for a multi-beam echo-sounder.

Among them is the analysis of backscatter strength profiles through use of amplitude quantiles. This is surprisingly expensive, given that the input data needs to be sorted. Nonetheless, they chose to examine the suitability of deciles, and found extremely good separability.

Also examined is the modelling of the backscatter probability density function. Examination of the data has suggested that the gamma family of PDF's is more applicable than Gaussian. Several moments are generated, showing generally good separability.

3.2.4.3 Caughey *et al*, (1994)

In a contractor's report prepared for the Defense Research Establishment Pacific [13], the author uses an amplitude histogram in addition to other methods. The ten-bin histogram provides decent separation. Possibly not as good as quantiles, but certainly a lot cheaper.

3.2.4.4 Tan and Mayer, (in prep.)

In their paper "Seafloor classification from the data of acoustic reflections" [3], the authors condition the input signal as described by Poulinquen and Lurton [24], and then empirically identify three offsets at which separability is deemed optimal. Classification on the small training set seems good.

3.2.5 Classification methods

Independent of feature extraction is the process of classification. For any known distribution, the Bayesian classifier can be shown to be optimal. However, not all data distributions are known, which merits examination of other methods.

Kohonen self-organizing maps are sometimes used as a non-parametric approach. This is somewhat misleading, in that there is an inherent parameterization in a KSOM, built in to the description of the network. This type of classification is examined by Alexandrou and Pantzartzis [35], and Zerr, Maillard and Gueriot [37], described elsewhere in this chapter.

3.2.5.1 Kavli, Carlin and Madsen, (1993)

In their paper "Seabed classification using artificial neural networks and other non-parametric methods" [8], the authors compare a variety of methods to a Bayesian classifier.

One of the methods is the ASMOD (Adaptive Spline Modelling of Observation Data) scheme. While this would appear to be a modelling approach, it uses precalculated features, and then uses a grid partition of the feature space, making a set of hypercubes that fill up the space.

Results obtained with this method were respectable, but not as good as with a Bayesian classifier.

3.2.5.2 Tan and Mayer, (in prep.)

In their paper "Seafloor classification from the data of acoustic reflections" [3], the authors use fuzzy-set theory to perform the classification. One advantage to this approach is that one generates a confidence simultaneously with the classification.

This method shows promise of being able to conveniently handle mixtures.

3.2.6 Sub-bottom profiling

While the sub-bottom profilers are distinct from surficial classifiers, the following two systems merit some discussion due to their high profile. Few references containing detailed analyses are available.

3.2.6.1 Lambert, 1993

Lambert's packaging of the Honeywell ELAC system can operate as a surficial classifier when used with high-frequency transducers [12].

It operates on the concept that the return is divided into ten adjustable width time windows which correspond to depth increments in the sediment. It then applies algorithms based on multi-layer acoustic theory to compute acoustic impedance for each of the depth increments.

When used with high-frequency transducers, the depth lines can be interpreted as quantifiers of the signal shape, and could be used for surficial classification.

3.2.6.2 Caulfield

The Caulfield Engineering™ technique uses carefully calibrated transducers to enable an inversion of the field equations to derive a complete picture of the acoustic impedance of the sub-bottom. Again, if a high frequency transducer is used, then only the surficial impedance at that frequency will be measured.

3.3 Summary

Seabed classification started out in the late 1970's as an extension to sub-bottom profiling. Initial attempts were based on coherence measures of adjacent traces. Starting in the mid-1980's, advanced processing techniques began to be used which treated the waveforms as generic signals. Current research consists primarily of work being done by the author, the Ocean Mapping group at UNB, and the Norwegian ESMAC project.

4 Initial Processing

This chapter is the first of two chapters which discuss the means by which an acoustic return is analyzed and classified.

First, the general procedure is given by which acoustic returns are successively processed, converting raw signals into classifications.

Second, the first two steps of the procedure — event detection and pre-processing — are discussed in greater detail. This includes related issues such as picking poor data, deconvolution, time-scale normalization, amplitude normalization and spatial averaging. Novel contributions include the deconvolution of impulse functions from envelopes, and the use of time-scale normalization to correct for depth.

The subsequent analysis of the massaged acoustic returns is covered in Chapter 5.

4.1 General procedures

Run-time processing is the result of the sequential application of several discrete processes, using prior information and statistics gathered for a specific geographic locale. If these parameters are not known then analysis processing must be performed.

Each of the steps identified in these procedures are described in greater detail in subsequent sections.

4.1.1 Run-time

The sequence illustrated in Fig 4.1 is followed when the reduction mapping[1] and cluster[2] statistics are known. This chapter discusses only the first two steps after the signal has

1. The method of decreasing the dimensionality of the problem.
2. A statistically distinct group of traces, usually representative of a particular bottom type.

been acquired; namely, event detection and pre-processing. The subsequent steps are discussed in the next chapter.

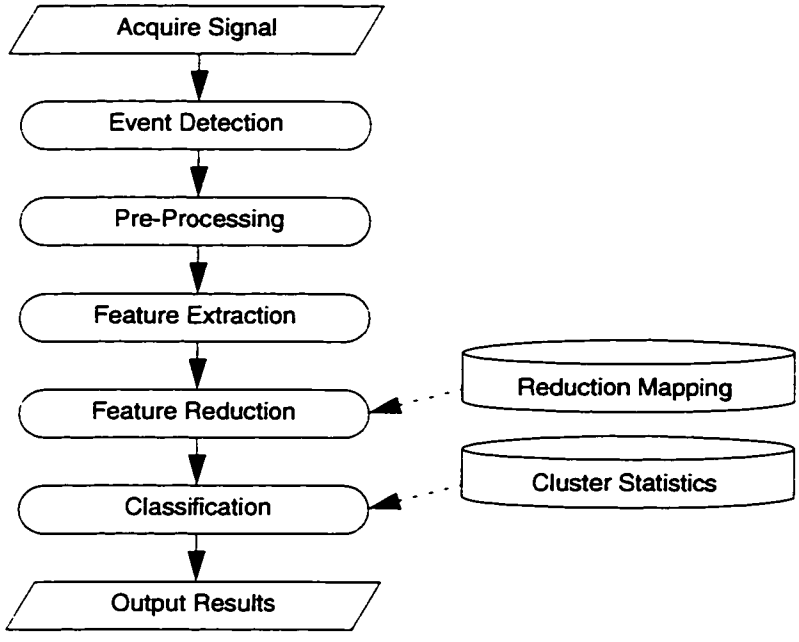


Figure 4.1: Run-time processing sequence

Note that in the context of bottom classification, event detection and bottom picking are synonymous.

4.1.2 Analysis

In situations in which there is a new data set, or when it is desired to regenerate the feature reduction mapping or cluster statistics, the run-time processing steps are modified to include principal component analysis (to determine the optimal feature reduction mapping) and clustering via either supervised or unsupervised (blind) algorithms.

However, this chapter discusses only the first two steps after the signal has been acquired; namely, event detection and pre-processing. The subsequent steps are discussed in the next chapter.

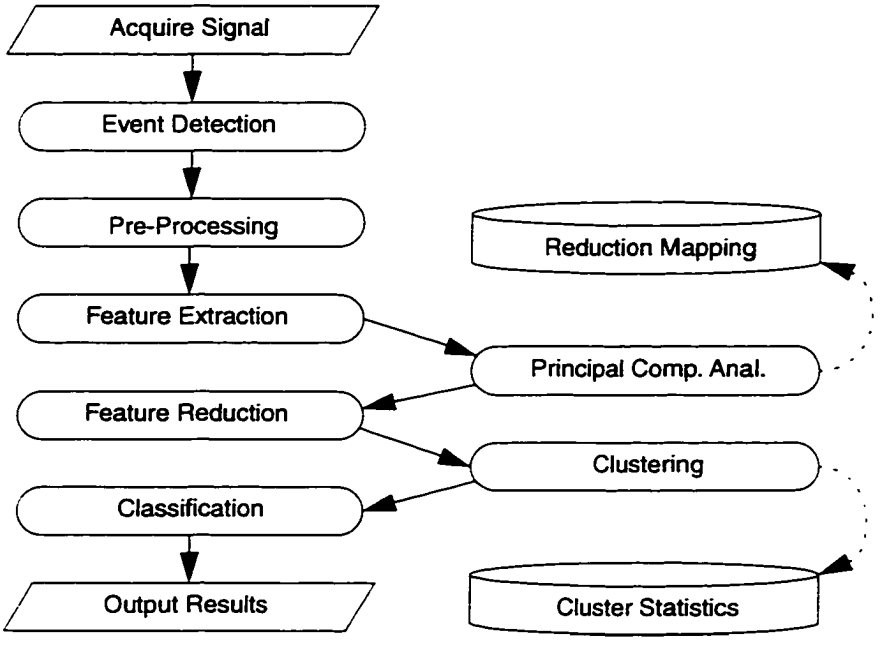


Figure 4.2: Analysis processing sequence

4.2 Event Detection

The determination of the time of arrival of the reflection from the bottom is known as bottom picking. There are a number of ways that bottom picking can be performed — there are threshold methods, correlations, energy windows, etc. — each of which has its own strengths under certain circumstances.

Furthermore, picking can be performed on the raw data or on spatially averaged data (see Section 4.3.5). However, averaging requires aligning the signals to reduce jitter, which becomes a trivial task if a reference (i.e., the bottom) has already been identified. Thus, except under very noisy circumstances, there is no justification for picking on averaged echosounder returns. No such circumstances were identified in the data sets analyzed in Chapter 6.

4.2.1 Quality of picking

There are two main factors which affect the quality of a bottom pick. First, there are distortions to the signal arising from the following:

- water column reflectors (e.g., suspended sediment, biology)
- noise
- bathymetric variations (i.e., slope or outcrops)
- heterogeneous bottoms with cobbles and boulders

Second, there is ambiguity as to the definition of "bottom". That is, from the perspective of navigation, the time to first return may be the necessary definition, whereas for habitat surveys the mean depth within the acoustic footprint may be more relevant. In the context of classification, depth is required in order to compensate for certain artefacts, most notably dilation (see Sections 2.2.5.1 and 4.3.3), suggesting that the most applicable definition is that value which pertains to most of the scatterers within the footprint.

However, since large objects resting on the bottom will generate early returns[1], important classification information may in fact exist prior to the bulk return. Unfortunately, some water column reflectors such as bottom fish and suspended sediment (which do not impart information) may also contribute to the signal in a similar manner.

4.2.2 Picking

Bottom picking algorithms of varying complexity can be applied to the signal. The advantage to picking on the spatially averaged signal as opposed to the raw signal is that it increases the SNR, thereby reducing the chance of mis-picking due to water column reflectors (such as bottom fish) and the random nature of incoherent backscatter.

1. At a sampling rate of 10 kHz, one can detect bathymetry variations of ~7.5 cm.

It was found that threshold detectors are simple, yet effective. The algorithm presented here assumes a return has a structure as identified in Fig 4.3 below.

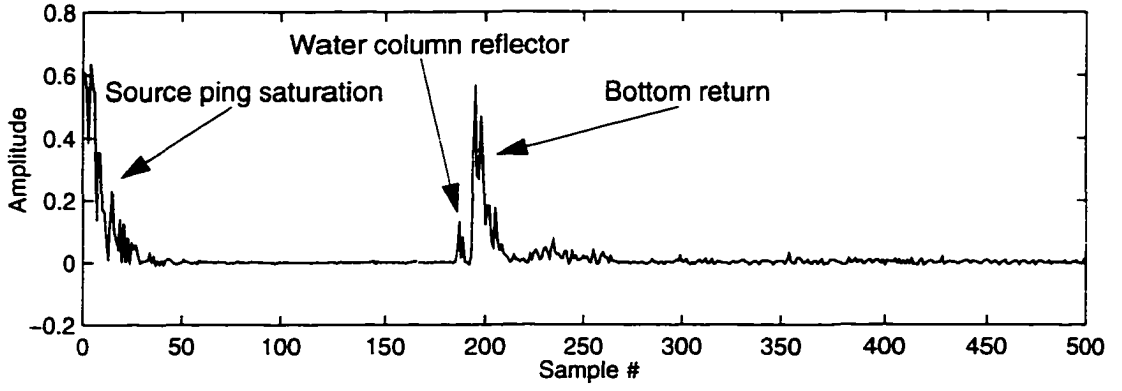


Figure 4.3: Typical bottom return

To pick the bottom correctly, it is necessary to provide a gating function to eliminate the transducer saturation resulting from the transmit pulse. This can be defined as the time at the signal has dipped below 10% of its maximum value (at sample 12 in the case of the signal illustrated in Fig 4.3). Of the remaining signal, we identify the maximum value (a value of 0.57 occurring at sample 195).

The bottom lies between the gating start and the index of the maximum value.

The basic procedure then involves finding the index of the first sample exceeding 0.75 of the maximum, and then searching backwards for the first sample below 0.25 of the maximum on a positive slope (i.e., the first sample of the bottom return is presumably greater than the previous sample).

The double threshold prevents many of the common mis-picks due to notches in highly scattered (broad) returns, where the maximum value may not necessarily be associated with the leading edge. Choosing a significantly high (e.g., 0.75) value ensures that the upper limit for the pick occurs somewhere on the leading edge, avoiding the vast majority

of water column reflectors. The second threshold picks the last near-baseline point as the bottom.

The basic procedure is modified slightly to permit adjustment of the thresholds. For example, in very shallow water the return may occur during the latter parts of the transmit pulse saturation of the transducer. Therefore, there may not be any points less than the lower threshold of 0.25 of the maximum post-gate value. This requires that the lower threshold be adjusted upwards. If no such adjusted threshold results in a bottom pick, the upper threshold of 0.75 is adjusted upwards.

This algorithm assumes that time-varying gain is being used to ensure that the bottom return is of similar amplitude as the saturation. Therefore, returns less than 0.25 are identified as bad traces, rather than picked.

Note also that this algorithm depends on the absolute amplitude of the signal. Therefore, a bias on the received signal can reduce the relative significance of the return component. Other aspects of picking will also benefit from ensuring that any bias has been removed. Bias can be reduced in a number of ways, the most simple of which is to simply subtract the minimum value. However, in addition to providing incomplete bias elimination, this method is affected by wild points[1]. A more precise method is to subtract the *mode* of the signal. An easy method of calculating the mode in MATLAB is shown below in Example 4.1.

This approach sorts the signal in ascending order, which allows the longest run

---

1. Wild points can occur naturally, or can be introduced by poor data acquisition. For example, the UNB data sets collected with early ISAH equipment padded each record with two zeroes. Thus, despite a considerable bias, the minimum point is always zero.

```
x_sort = sort( x );
change_locs = find(diff(x)>0);
longest = maxi(diff(change_locs));
mode = x_sort(change_locs(longest)+1);
```

Example 4.1: Calculating the mode of a signal

(i.e., the mode) to be identified.

4.2.2.1 Picking bad data

The subsequent process of spatial averaging assumes that all records contain legitimate acoustic returns and that adjacent traces ought to bear some similarity. Unfortunately, when legacy systems are used, the quality of the data can be questionable.

For example, the Saint John Harbour data set (as will be discussed in Section 6.2.2) was collected with older echosounding equipment[1] using an early version of the logging software. As a result, many of the records are bad, and some are obviously out of order. Additional discrepancies in GPS times suggest the data logging was not robust.

For such data sets, it may be advisable to eliminate bad records. This can be done by median filtering the vector of bottom pick results to identify any "outliers". However, rather than replace the outliers by the median value, the corresponding records are simply deleted from the data set.

1. The Navitronics echosounder in question was mounted in a vessel which sank once, while tied up at the pier. Despite being "taken apart and laid out on the dock to dry", it was felt that "the equipment never worked the same since."

4.3 Data Pre-processing

The received signal is distorted by several processes unrelated to the actual bottom type. Therefore it is necessary to compensate for these phenomena in order to accurately compare signals.

As described in Chapter 2, the main systemic sources of distortion are:

- non-vertical angle of the transmitted beam on the bottom.
- convolution with the source signal,
- linear dilation due to depth, and,
- attenuation due to spherical spreading, absorption in the water column, and possibly transducer time-varying gain,

Each of these effects are discussed below:

4.3.1 Transducer angle

Distortions introduced by non-vertical orientation of the transmitted beam (due either to a bottom slope or a boat angle) are not easily compensated. In addition, doing so would require extensive calibration of the transducer.

Fortunately, bottom classification is actually improved if the acoustic beam is slightly off-vertical, since the resultant acoustic footprint will span a broader range of incident angles. Indeed, using transducer models and backscatter models proposed by Jackson *et al* [18],[19],[20], it is not possible to simulate traces of the same length as most of the observed signals.

In fact, if a transducer beam pattern exhibits strong decay (as in Fig 4.10), then it is impossible to generate a usable return signal (assuming 12-16 bit digitization) for samples corresponding to incident angles outside of the main footprint or the first side-lobe.

The effect of the off-vertical beam angle on the acoustic footprint is shown from below in Fig 4.4. The concentric circles indicate the annuli of insonification for successive time samples. Note that the area of greatest amplitude (darkest) spans more annuli of insonification when slightly off-vertical.

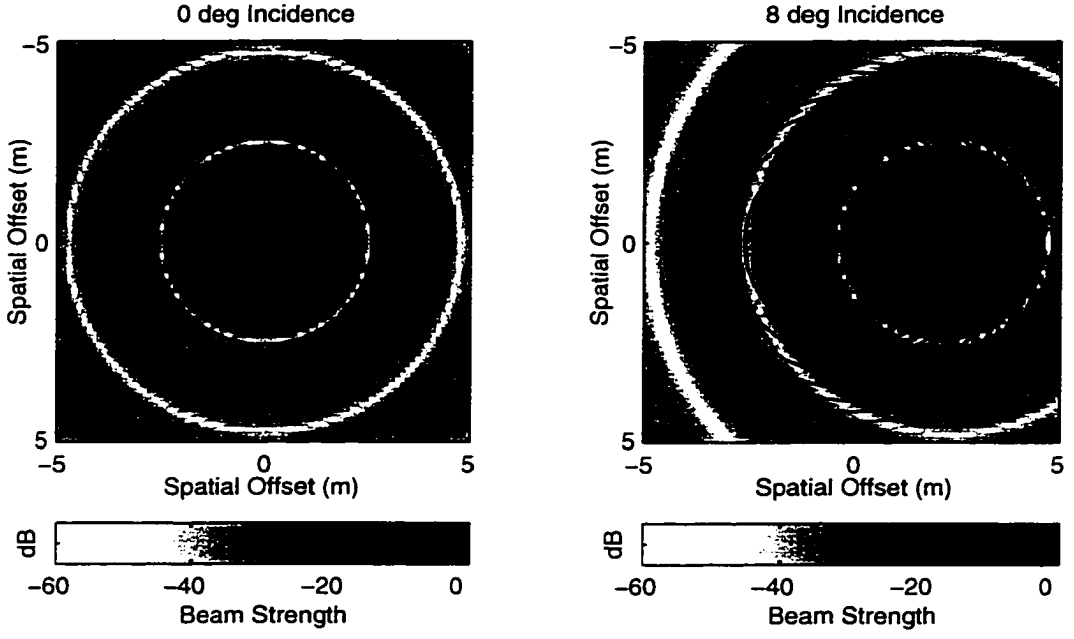


Figure 4.4: Spatial insonification pattern and isotemporal contours for transducer with 8° beam width at varying degrees of incidence.

Furthermore, the associated lack of symmetry in annulus coverage will result in no annulus consisting entirely of one incident angle corresponding to a null in the beam pattern. Therefore, there should not be any sharp nulls in the cumulative response.

Indeed, this is verified in Fig 4.5, in which the distribution of incident angles throughout an annulus of insonification is plotted as a function of time offset, for several off-vertical transducer angles φ_v. In these plots, each grayscale column corresponds to the density histogram of the incident beam strength around each annulus, and the dashed line

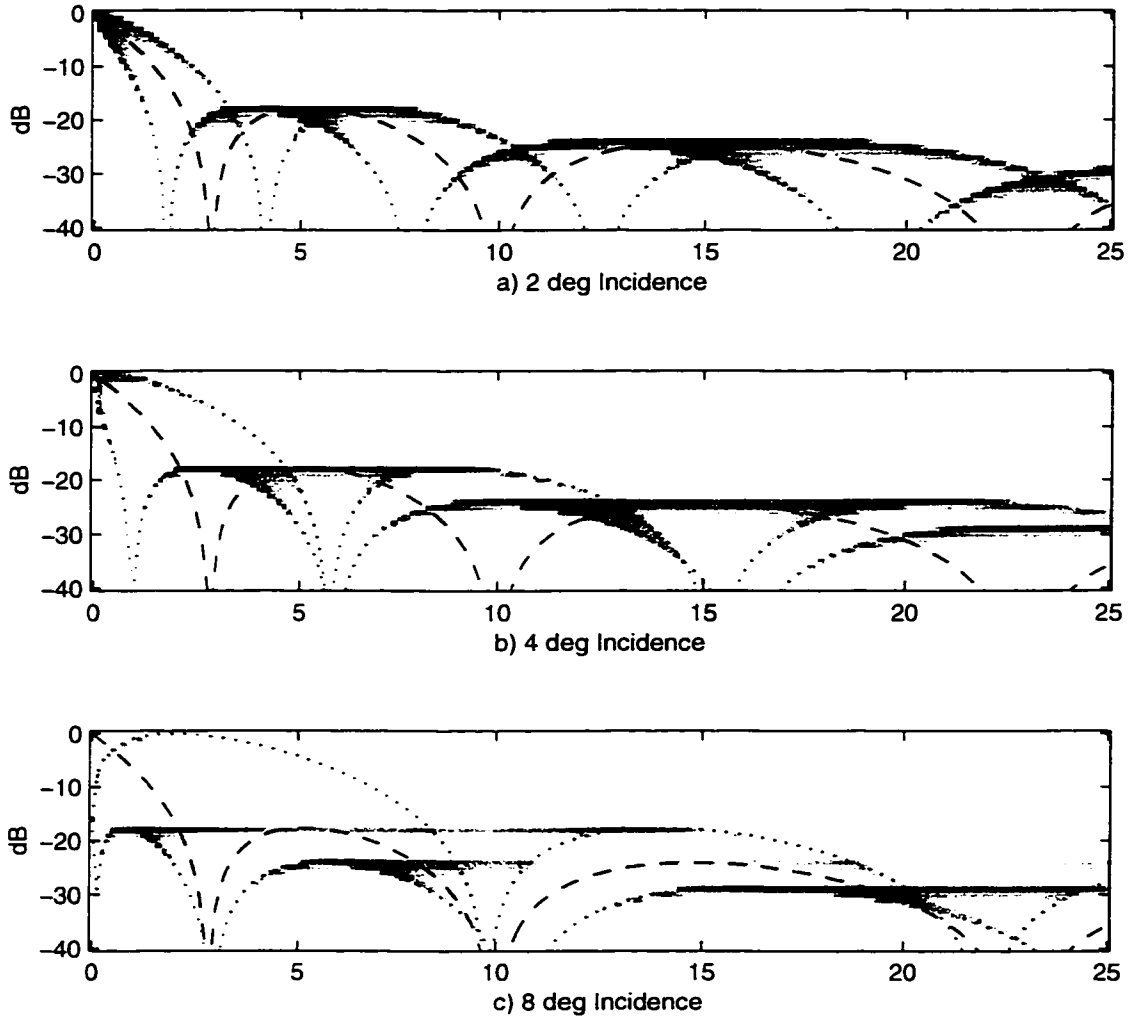


Figure 4.5: Insonification beam strength densities as a function of annulus number (or time offset) for varying transducer angles. The grayscale columns in each plot represent the density histogram, and the dashed line is the theoretical response of a zero-degree incident beam.

corresponds to the theoretical density for a perfectly vertical beam. The dotted lines (comprising the boundaries of the histograms) show the beam patterns generated by sampling the strengths along lines from the centre of insonification towards (and away from) the center of the acoustic footprint. The resulting patterns are referred to in the following paragraphs as positive and negative (respectively) axial beam patterns.

The axial beam patterns define the strongest and weakest insonification possible

for a given annulus. As well, the lobes of the axial beam patterns identify the potential maximum insonification for all annuli within a specific range. That is, the smallest annulus which contains the lobe *n* maximum value is determined by the negative axial beam pattern. The largest annulus to incorporate the maximum value of the same lobe is determined by the positive axial beam pattern. All intervening annuli likewise intersect the lobe maximum at some point (see Fig 4.4). This gives rise to the horizontal banding between lobes shown in Fig 4.5. Annuli associated with incident angles not included in these ranges will have a maximum insonification equal to that of the positive axial beam pattern.

4.3.2 Envelope deconvolution

One of the sources of signal distortions arises from the fact that the received signal does not represent an impulse response. Instead, the received signal is the convolution of the ideal bottom response and the input signal. Moreover, it is the *envelope* of the received signal which is recorded on most systems[1].

The problems arise later when trying to normalize the signal with respect to depth (Section 4.3.3), since the bottom impulse response experiences linear dilation with depth, while the source signal remains constant. Thus, before time-scale normalization can occur, the bottom impulse response should be deconvolved from the digitized envelope.

This section represents one of the novel contributions of this thesis, and is divided into subsections dealing with definitions, a direct solution, and an adaptive solution.

---

1. The rationale for which is the historical use of strip charts to record bathymetry.

4.3.2.1 Definitions

For a discretized received signal $x(t) = s(t)*h(t)$, where $s(t)$ is the source and $h(t)$ is the ideal (impulse) response of the bottom, we have the Fourier pair $X(f) = S(f) \cdot H(f)$. This allows standard deconvolution to be accomplished as follows:

$$s(t) = \mathcal{F}^{-1}\{X(f) \cdot H^{-1}(f)\} \tag{4.1}$$

provided that the impulse response $H(f)$ is known, and notwithstanding the numerical difficulties of inverting $H(f)$.

However, in the case of the envelope $v(t)$ of the received signal (which is the data that is actually acquired in most echosounders), we have

$$v(t) = |x_a(t)| = \sqrt{x_a(t) \cdot x_a^*(t)} \tag{4.2}$$

where the analytic signal $x_a(t)$ is the pre-envelope [38] of $x(t)$, defined as

$$x_a(t) = x(t) + j\tilde{x}(t) \tag{4.3}$$

with $\tilde{x}(t) = \mathcal{H}\{x(t)\}$ being the Hilbert transform of $x(t)$.

By noting that $\mathcal{H}\{a(t)*b(t)\} = a(t)*\mathcal{H}\{b(t)\}$ [38], Eq. (4.2) can be expanded into a product of convolutions in the time domain.

$$\begin{aligned} v^2(t) &= x^2(t) + \tilde{x}^2(t) \\ &= (s(t)*h(t)) \cdot (s(t)*h(t)) + (\tilde{s}(t)*h(t)) \cdot (\tilde{s}(t)*h(t)) \\ &= \sum_\tau s(\tau)h(t-\tau) \sum_\tau s(\tau)h(t-\tau) + \sum_\tau \tilde{s}(\tau)h(t-\tau) \sum_\tau \tilde{s}(\tau)h(t-\tau) \end{aligned} \tag{4.4}$$

or a convolution of products in the frequency domain

$$\begin{aligned}
\mathcal{F}^{-1}\{V^2(f)\} &= \mathcal{F}^{-1}\{X^2(t) + \tilde{X}^2(t)\} \\
&= \mathcal{F}^{-1}\{[S(f)\cdot H(f)]^*[S(f)\cdot H(f)] + [\tilde{S}(f)\cdot H(f)]^*[\tilde{S}(f)\cdot H(f)]\} \\
&= \mathcal{F}^{-1}\left\{\sum_{\zeta} S(\zeta)H(\zeta)S(f-\zeta)H(f-\zeta) + \sum_{\zeta}\tilde{S}(\zeta)H(\zeta)\tilde{S}(f-\zeta)H(f-\zeta)\right\}
\end{aligned} \tag{4.5}$$

Although deconvolution is often carried out in the frequency domain. the time domain description of $v^2(t)$ has an advantage over the frequency domain description in that $s(t)$ is known to be finite in duration ($s(t) = 0, \forall t > N_s$) for echosounder pings, whereas $S(f)$ is likely to have infinite extent. This allows an estimate $\hat{v}^2(t)$ to be defined using a finite-length convolution ($N \geq N_s$) by rewriting Eq. (4.4) as follows:

$$\begin{aligned}
\hat{v}^2(t) &= \sum_{\tau}^{N} s(\tau)h(t-\tau)\sum_{\tau}^{N} s(\tau)h(t-\tau) + \sum_{\tau}^{N}\tilde{s}(\tau)h(t-\tau)\sum_{\tau}^{N}\tilde{s}(\tau)h(t-\tau) \\
&= (\boldsymbol{s}^T\boldsymbol{h}(t))(\boldsymbol{s}^T\boldsymbol{h}(t)) + (\tilde{\boldsymbol{s}}^T\boldsymbol{h}(t))(\tilde{\boldsymbol{s}}^T\boldsymbol{h}(t)) \\
&= \boldsymbol{h}^T(t)\boldsymbol{s}\boldsymbol{s}^T\boldsymbol{h}(t) + \boldsymbol{h}^T(t)\tilde{\boldsymbol{s}}\tilde{\boldsymbol{s}}^T\boldsymbol{h}(t) \\
&= \boldsymbol{h}^T(t)(\boldsymbol{B} + \tilde{\boldsymbol{B}})\boldsymbol{h}(t)
\end{aligned} \tag{4.6}$$

where the length-N sliding window $\boldsymbol{h}(t)$ and static window $\boldsymbol{s}$ are shown below:

$$\begin{aligned}
\boldsymbol{h}(t) &= \left[h(t)\ h(t-1)\ \ldots\ h(t-N+1)\right]^t \\
\boldsymbol{s} &= \left[s(0)\ s(1)\ \ldots\ s(N-1)\right]^T
\end{aligned} \tag{4.7}$$

and the outer self-products of $\boldsymbol{s}$ and $\tilde{\boldsymbol{s}}$ are given by $\boldsymbol{B}$ and $\tilde{\boldsymbol{B}}$, respectively.

To illustrate, consider a hypothetical situation using the signals shown in Fig 4.6 below. A bottom having an impulse response designated by $h(t)$ is insonified by a signal $s(t)$ consisting of a pulse-modulated sinusoid. The response received by the hydrophone is the convolution of the two signals. The envelope[1] of the signal is then generated.

Fig 4.7 shows the Hilbert envelope of the signal, as defined in Eq. (4.2), as well as

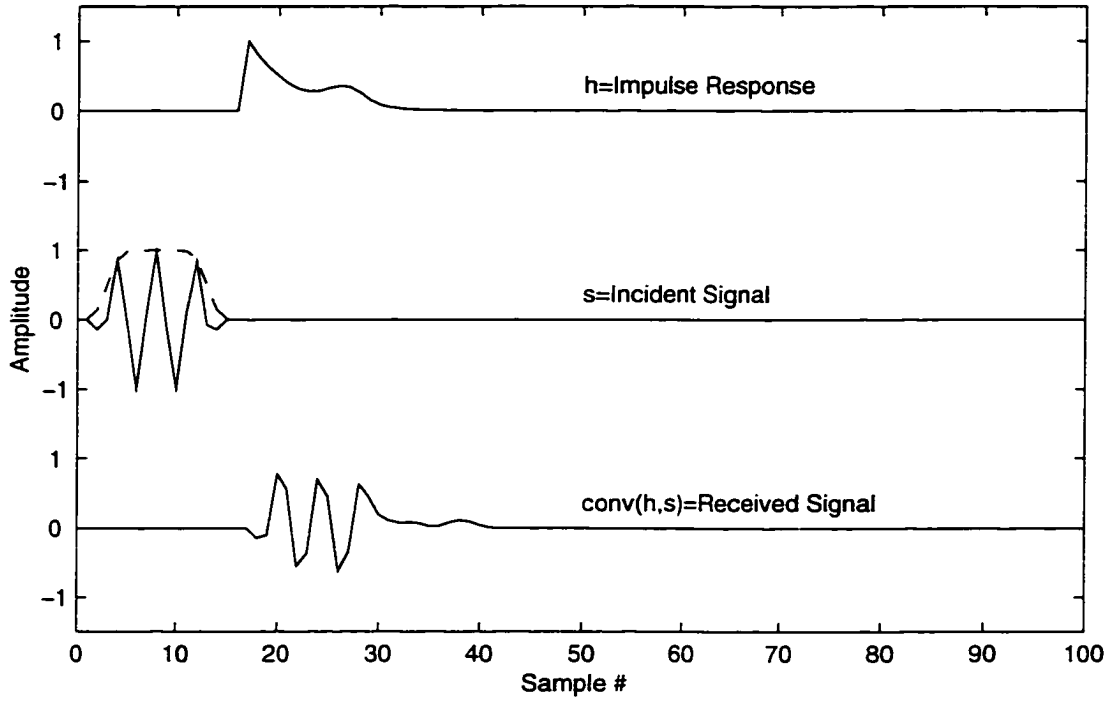


Figure 4.6: Signals used in envelope filtering example

the envelope as estimated by the filter (N=16) described by Eq. (4.6). Both versions of the envelope are overlaid with the measured bottom response.

Note that the finite-length filter estimation of the envelope does not exhibit the non-causality that exists with an FFT implementation of the Hilbert transform.

4.3.2.2 Direct solution

We define a standard predictor network with the desired signal $d(t)$ defined as the square of the measured envelope (i.e., $v^2(t)$), and an estimate $\hat{d}(t)$ defined by Eq. (4.6). The error equation of this network is then as follows:

---

1. Although echosounders typically employ simple peak detection circuitry for envelope generation, the more analytically tractable Hilbert transform [38],[39] is used here.

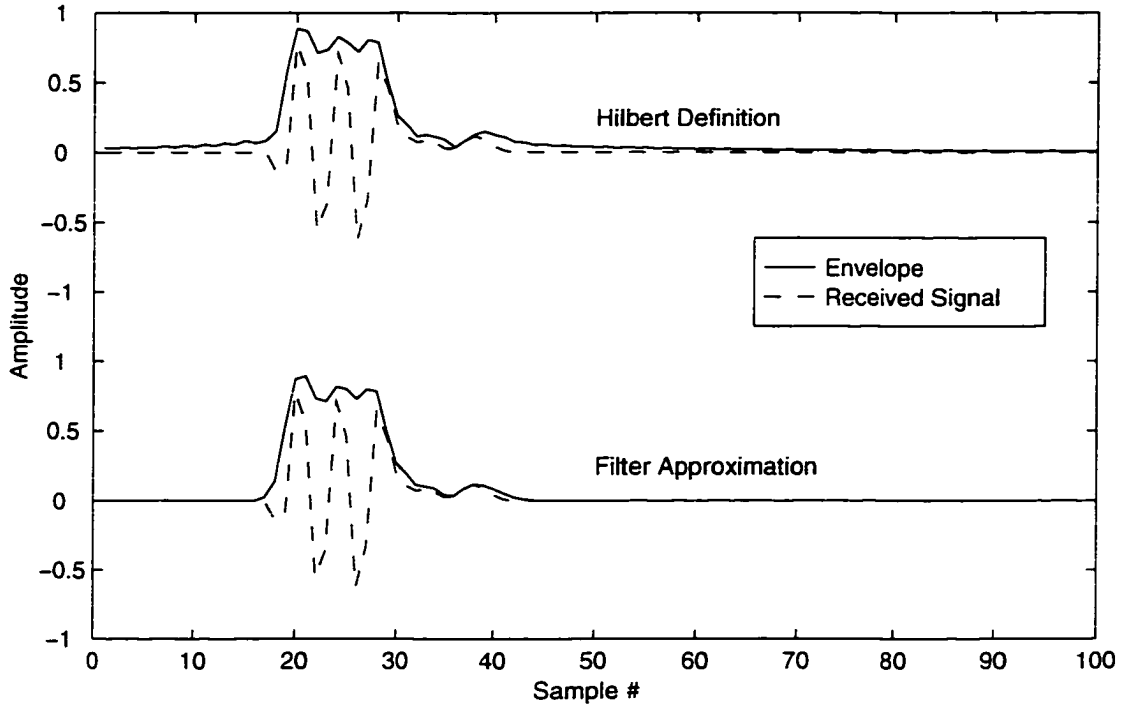


Figure 4.7: Convolution envelope functions

$$\begin{aligned} e(t) &= d(t) - \hat{d}(t) \\ &= v^2(t) - \boldsymbol{h}^T(t)(\boldsymbol{B} + \tilde{\boldsymbol{B}})\boldsymbol{h}(t) \end{aligned} \tag{4.8}$$

A filter described by Eq. (4.8) differs from standard filtering applications in that it requires that a large number of filter weights (representing an estimate of the true impulse response of the bottom) be determined from the application of a short but deterministic input sequence. Furthermore, the filter weights used in $\boldsymbol{h}(t_1)$ to estimate $\hat{d}(t_1)$ are not the same set as those used to estimate $\hat{d}(t_2)$. Remember, $\boldsymbol{h}(t)$ is a sliding window.

One approach to the determination of the impulse response is through least mean-square estimation of the complete impulse response $\hat{\boldsymbol{h}}_{lms}$. To do this, it is necessary to describe an objective function[1], for example

$$f_{lms}(\hat{\boldsymbol{h}}_{lms}) = \boldsymbol{e}^T\boldsymbol{e} \tag{4.9}$$

There exist many methods for optimizing this equation. The approach taken here is to define $\hat{\boldsymbol{h}}_{lms}$ as containing N_o variables, and the filter as having N_f taps. Therefore, the error vector $\boldsymbol{e}$ contains $N_o + N_f - 1$ elements. Then the solution is found using MATLAB's constrained least-squares optimization routine. The constraint $\hat{\boldsymbol{h}}_{lms} > 0$ represents the fact that the impulse response is an estimate of the bottom's backscatter intensity profile, and therefore only positive values are meaningful.

If more is known about the bottom *a priori*, then other constraints can also be considered.

In addition to the constraint, the error surface contains local minima requiring a mechanism for dislodging stuck solutions. Again, many methods exist, most notably simulated annealing. However, an extremely simple approach is to take advantage of the assumption that the correct solution is a backscatter intensity profile and hence is usually not subject to discontinuities, the coefficients are periodically smoothed using an order-2 MA filter.

Fig 4.8 below illustrates the results of this approach, using the signals illustrated in Fig 4.3 with $N_o = 40$ and $N_f = 16$, and an initial estimate of $\hat{\boldsymbol{h}}_{lms}(0) = [1 \ldots 1]^T$.

4.3.2.3 Adaptive solution

An alternative way of approaching the problem is to recognize that Eq. (4.6) represents a second-order Volterra kernel, and then follow Mathews [55] in treating collections of

---

1. By itself, $\boldsymbol{e}$ is not suitable for optimization since it can attain both positive and negative values. By taking its inner product, we ensure that the values are strictly positive, and we reduce the optimization target to a scalar value. Note that Eq. (4.9) is not the only objective function. For example, a positive definite matrix $\boldsymbol{W}$ can be used to create a scalar $\boldsymbol{e}^T\boldsymbol{W}\boldsymbol{e}$ which is strictly positive, yet weights the contributions of the different errors.

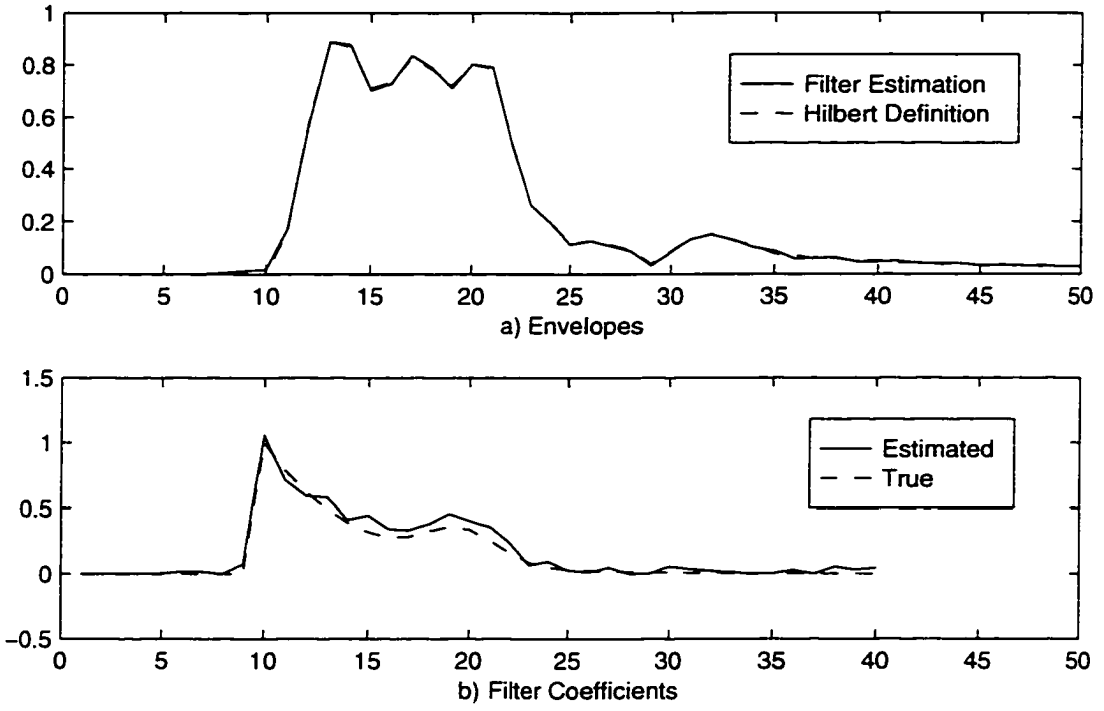


Figure 4.8: LMSE solution after 2000 iterations

order-p terms as a single vector. If we expand Eq. (4.6), and collect the constants and the variables, we can arrive at a less elegant representation of the Hilbert transform approximation of an envelope, as follows:

$$\begin{aligned}\hat{v}^2(t) &= \sum_{\tau}^{N} s(\tau)h(t-\tau)\sum_{\tau}^{N} s(\tau)h(t-\tau) + \sum_{\tau}^{N} \tilde{s}(\tau)h(t-\tau)\sum_{\tau}^{N} \tilde{s}(\tau)h(t-\tau) \\ &= vec(\boldsymbol{G}(t))^T vec(\boldsymbol{B} + \tilde{\boldsymbol{B}})\end{aligned} \tag{4.10}$$

where the $vec(\bullet)$ operator reshapes a matrix into a vector. The matrix $\boldsymbol{G}(t)$ is the outer self-product of the length-N sliding window $\boldsymbol{h}(t)$ as shown below:

$$\boldsymbol{G}(t) = \boldsymbol{h}(t)\boldsymbol{h}^T(t) \tag{4.11}$$

and the constants $\boldsymbol{B}$ and $\tilde{\boldsymbol{B}}$ are defined similarly from the static window s and its Hilbert transform $\tilde{s}$, that is

$$\boldsymbol{B} = \left[s(0)\ s(1)\ \ldots\ s(N-1)\right]^T\left[s(0)\ s(1)\ \ldots\ s(N-1)\right] \tag{4.12}$$

$$\tilde{\boldsymbol{B}} = \left[\tilde{s}(0)\ \tilde{s}(1)\ \ldots\ \tilde{s}(N-1)\right]^T\left[\tilde{s}(0)\ \tilde{s}(1)\ \ldots\ \tilde{s}(N-1)\right] \tag{4.13}$$

The important conceptual difference between Eq. (4.6) and Eq. (4.10) is that the latter views the second-order products of $\boldsymbol{G}(t)$ as individual unrelated elements in a first-order vector. This has the impact of making the objective equation a quadratic, as opposed to fourth-order as mentioned above. Furthermore, note that each of the outer product matrices is symmetric and therefore the $vec(\bullet)$ operator only needs to retain the $N(N+1)/2$ unique elements.

Using a stochastic gradient approach, a variable is modified such that the estimate of the solution moves slightly in the direction of the steepest gradient (with respect to the variable in question) of the error surface (i.e., the objective function). That is,

$$x_{i,n+1} = x_{i,n} - \frac{\mu}{2}\frac{\partial}{\partial x_{i,n}} f(\boldsymbol{x}_n) \tag{4.14}$$

where μ is an adaptation constant which governs the speed of optimization and the error of the solution.

Using the objective function given in Eq. (4.9), the derivative of the objective function with respect to a single element in $\boldsymbol{G}(t)$ is given by

$$\begin{aligned}
\frac{\partial}{\partial vec(\boldsymbol{G})^T} e^2 &= \frac{\partial}{\partial vec(\boldsymbol{G})^T}(v^2 - vec(\boldsymbol{G})^T vec(\boldsymbol{B}+\tilde{\boldsymbol{B}}))(v^2 - vec(\boldsymbol{G})^T vec(\boldsymbol{B}+\tilde{\boldsymbol{B}})) \\
&= -2v^2 vec(\boldsymbol{B}+\tilde{\boldsymbol{B}}) + 2vec(\boldsymbol{B}+\tilde{\boldsymbol{B}})vec(\boldsymbol{B}+\tilde{\boldsymbol{B}})^T vec(\boldsymbol{G}) \\
&= -2vec(\boldsymbol{B}+\tilde{\boldsymbol{B}})(v^2 - vec(\boldsymbol{B}+\tilde{\boldsymbol{B}})^T vec(\boldsymbol{G})) \\
&= -2vec(\boldsymbol{B}+\tilde{\boldsymbol{B}})e
\end{aligned} \tag{4.15}$$

Therefore, the variable $g(\tau_1, \tau_2;n) = h(\tau_1;n)h(\tau_2;n)$ corresponding to a particular element of $\boldsymbol{G}(t)$ at iteration n can be updated by changing its value as follows:

$$\begin{aligned} g(\tau_1, \tau_2;n+1) &= g(\tau_1, \tau_2;n) + \mu e(t;n) vec(\boldsymbol{B} + \tilde{\boldsymbol{B}}) \\ &= g(\tau_1, \tau_2;n) + \theta(t, \tau_1, \tau_2;n) \end{aligned} \tag{4.16}$$

However, in this application there is a non-linear relationship between the $N(N+1)/2$ unique equations and the N unknowns. The best that can be done is to adapt the vector $\boldsymbol{h}(t)$ by a vector $\boldsymbol{a}$. Then, by definition of Eq. (4.11) we have

$$\begin{aligned} \boldsymbol{G}(t;n+1) &= (\boldsymbol{h}(t;n) + \boldsymbol{a}(n)) \cdot (\boldsymbol{h}(t;n) + \boldsymbol{a}(n))^T \\ &= \boldsymbol{G}(t;n) + \boldsymbol{h}(t;n)\boldsymbol{a}^T(n) + \boldsymbol{a}(n)\boldsymbol{h}^T(t;n) + \boldsymbol{a}(n)\boldsymbol{a}^T(n) \end{aligned} \tag{4.17}$$

and therefore

$$g(\tau_1, \tau_2;n+1) = g(\tau_1, \tau_2;n) + h(\tau_1;n)a(\tau_2;n) + a(\tau_1;n)h(\tau_2;n) + a(\tau_1;n)a(\tau_2;n) \tag{4.18}$$

The difference after iteration n between the definitions of $g(\tau_1, \tau_2)$ given in Eq. (4.16) and Eq. (4.18) allows us to define an error vector

$$\boldsymbol{k}(\boldsymbol{a}) = \begin{bmatrix} a^2(t) + 2a(t)h(t) - \theta(t, t) \\ a(t)a(t-1) + a(t)h(t-1) + h(t)a(t-1) - \theta(t, t-1) \\ \vdots \\ a(t)a(t-N+1) + a(t)h(t-N+1) + h(t)a(t-N+1) - \theta(t, t-N+1) \\ a^2(t-1) + 2a(t-1)h(t-1) - \theta(t-1, t-1) \\ \vdots \\ a^2(t-N+1) + 2a(t-N+1)h(t-N+1) - \theta(t-N+1, t-N+1) \end{bmatrix} \tag{4.19}$$

as well as another objective function

$$f(\boldsymbol{a}) = \boldsymbol{k}^T(\boldsymbol{a})\boldsymbol{k}(\boldsymbol{a}) \tag{4.20}$$

and the problem becomes one of finding a suitable $\boldsymbol{a}$ such that the objective function is minimized.

If we can make the assumption that $|\boldsymbol{a}| \ll |\boldsymbol{h}(t)|$ on an element-by-element basis, then Eq. (4.19) simplifies to

$$\begin{aligned} \boldsymbol{k}(\boldsymbol{a}) &= \begin{bmatrix} 2a(t)h(t) - \theta(t, t) \\ a(t)h(t-1) + h(t)a(t-1) - \theta(t, t-1) \\ \vdots \\ a(t)h(t-N+1) + h(t)a(t-N+1) - \theta(t, t-N+1) \\ 2a(t-1)h(t-1) - \theta(t-1, t-1) \\ \vdots \\ 2a(t-N+1)h(t-N+1) - \theta(t-N+1, t-N+1) \end{bmatrix} \\ &= \boldsymbol{Q}\boldsymbol{a} - \boldsymbol{p} \end{aligned} \tag{4.21}$$

where

$$\boldsymbol{Q} = \begin{bmatrix} 2h(t) & & & & 0 \\ h(t-1) & h(t) & 0 & & \\ h(t-2) & 0 & h(t) & & \\ & & & & 0 \\ & 0 & & h(t-N+1) & h(t-N+2) \\ & & & 0 & 2h(t-N+1) \end{bmatrix} \qquad \boldsymbol{p} = \begin{bmatrix} \theta(t, t) \\ \theta(t, t-1) \\ \vdots \\ \theta(t-N+2, t-N+1) \\ \theta(t-N+1, t-N+1) \end{bmatrix} \tag{4.22}$$

The objective function can be expanded to

$$\begin{aligned} f(\boldsymbol{a}) &= (\boldsymbol{Q}\boldsymbol{a} - \boldsymbol{p})^T(\boldsymbol{Q}\boldsymbol{a} - \boldsymbol{p}) \\ &= \boldsymbol{a}^T\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{a} - 2\boldsymbol{a}^T\boldsymbol{Q}^T\boldsymbol{p} + \boldsymbol{p}^T\boldsymbol{p} \end{aligned} \tag{4.23}$$

and is minimized when the derivative of Eq. (4.23) with respect to $\boldsymbol{a}$ equals zero. This yields the solution

$$\begin{aligned} 2\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{a} - 2\boldsymbol{Q}^T\boldsymbol{p} &= 0 \\ \boldsymbol{a} &= (\boldsymbol{Q}^T\boldsymbol{Q})^{-1}\boldsymbol{Q}^T\boldsymbol{p} \end{aligned} \tag{4.24}$$

where $(\boldsymbol{Q}^T\boldsymbol{Q})^{-1}\boldsymbol{Q}^T$ is the Penrose pseudo-inverse of $\boldsymbol{Q}$[1].

For those cases in which we cannot assume that the squared terms are negligible, it is necessary to iterate on $\boldsymbol{a}$, using the Taylor series expansion

$$\boldsymbol{a}(n+1) = \boldsymbol{a}(n) + \frac{\partial}{\partial \boldsymbol{a}} f(n) + \frac{\partial^2}{\partial \boldsymbol{a}\partial \boldsymbol{a}} f(n) + \ldots \tag{4.25}$$

where the expansion usually stops after the first- or second-order terms. Because the objective function is formulated as a polynomial expression, the calculation of the Hessian matrix is straightforward, albeit verbose, suggesting that second-order optimization could be used.

Although the estimate of the bottom impulse response is not unique[2], it is better for use with classification since it does not contain irrelevant information about the source signal, nor does it reduce the effectiveness of the use of time-scale normalization to eliminate depth artefacts (discussed in Section 4.3.3 below).

4.3.3 Time-scale normalization

As mentioned in Section 2.2.5, due to the geometry of a spherical wave intersecting a plane, the bottom impulse response dilates linearly with depth. Reiterating Eq. (2.24) we have

$$n = \frac{2h}{\tau_s c}\left(\frac{1-\cos\phi}{\cos\phi}\right) \tag{4.26}$$

---

1. Note that Eq. (4.24) can be arrived at directly by solving $\boldsymbol{k}(\boldsymbol{a}) = \boldsymbol{Q}\boldsymbol{a} - \boldsymbol{p} = 0$.
2. Envelopes generated via the absolute amplitude of the analytic signal are invariant to phase.

where n is the offset in samples from the first (vertically incident) return, h is the depth, c is the speed of sound, τ_s is the sampling period, and φ is the angle of incidence.

It is possible to compare signals acquired at different depths by first normalizing their time-scales to that of a reference depth h_0. This is done by resampling the data to ensure that a given incident angle ϕ always corresponds to specific offset n. This yields the equality

$$\frac{2h_0}{\tau_s c}\frac{(1-\cos\phi)}{\cos\phi} = n = \frac{2h}{\tau_{norm} c}\frac{(1-\cos\phi)}{\cos\phi} \tag{4.27}$$

which provides the necessary resampling period

$$\tau_{norm} = \frac{h}{h_0}\tau_s \tag{4.28}$$

Another way of looking at time-scale normalization is that it ensures that the data is always sampled at specific sequence of non-uniformly spaced angles-of-interest

$$\phi_{samp} = \left[\phi_0\ \phi_1\ \cdots\ \phi_{n-1}\right] \tag{4.29}$$

Obviously, for echo-sounder data acquired from a constant depth (as with a fixed altitude tow fish), the effect of dilation may not be evident, and time-scale normalization may not be required.

4.3.3.1 Implementing time-scale normalization

Time-scale normalization is implemented using first-order piece-wise linear interpolation.

Since this is one of the more expensive steps in the processing of a return, it is advantageous to interpolate as small a portion of the signal as is necessary. If it is known

that the subsequent feature extraction is going to operate on a window of width N commencing N_{offset} samples before the bottom — after a time-scale resampling given by h/h_0 — then only a window of width Nh_0/h commencing $N_{offset}h_0/h$ samples before the bottom need be interpolated.

This technique also has the benefit of implicitly performing trace alignment — if the N time-scale normalized samples are re-written into a buffer, every record in the buffer will be aligned around sample N_{offset} (the picked bottom).

4.3.4 Amplitude normalization

Chapter 2 discussed a variety of phenomena which affect the amplitude of the acoustic return. The following sections discuss the principal sources of amplitude distortion from the perspective of the required compensation.

4.3.4.1 Water column attenuation

As a sound wave propagates through a medium it loses energy. This attenuation can be modelled as a gain factor (re-iterating Eq. (2.2))

$$A(l) = e^{-\alpha l} \tag{4.30}$$

where l is the path length in the medium and the constant α is a function of water temperature, salinity, and pressure, as well as the frequency of the wave. For seawater and a frequency of 210 kHz, the attenuation is between 80 dB to 90 dB per kilometre, giving α value of approximately 0.02 m^{-1}. Thus at water depths of approximately 50 m, the incident intensity $A(50)$ is down 4.3 dB.

Of course, the sound wave is a function of acoustic path length, which in turn is a function of time. Thus, to accurately compensate for water column attenuation, one has to represent Eq. (4.30) as a function of angle of incidence, as follows:

$$A(\varphi) = e^{-\alpha h/\cos\varphi} \tag{4.31}$$

with the attenuation (in dB) relative to vertical incidence given by

$$A_0(\varphi) = 4.343\alpha h\left(\frac{1}{\cos\varphi} - 1\right) \tag{4.32}$$

Thus, if one assumes a cut-off angle of 30°, the difference in attenuation between the beginning and the end of the return is 0.67 dB at 50 m.

It is acceptable to assume that in most cases the water column attenuation is constant over the duration of the return, and needs only be modeled as a function of depth, $A(h)$.

4.3.4.2 Spherical spreading

The spherical spreading gain as a function of acoustic path length l is given by (re-iterating Eq. (2.3))

$$L(l) = \frac{1}{l^2} \tag{4.33}$$

and is relative to the acoustic intensity measured at one metre from the source.

As with water column attenuation, accurate compensation requires that the spherical spreading be represented as a time-varying function across the duration of the return. However, the spreading loss (in dB) relative to vertical incidence is given by

$$L_0(\varphi) = -20\log_{10}\cos\varphi \tag{4.34}$$

Thus, if one assumes a cut-off angle of 30°, the difference in spreading loss between the beginning and the end of the return is 1.25 dB at any depth.

It is acceptable to assume that in most cases the spreading loss is constant over the duration of the return, and needs only be modeled as a function of depth, $L(h)$.

4.3.4.3 Time-varying gain

In order to compensate for the loss of signal energy arising from spherical spreading (and to a lesser extent, water column attenuation), many echosounders employ some form of automatic gain control. Since Eq. (4.33) is a function of spherical radius (or depth), and depth is linearly related to time, the compensation is usually implemented as a time-varying gain. Note that the spherical spreading loss described by Eq. (4.33) must be applied in both directions of the acoustic path. This would imply that the appropriate correction is a function of distance (or time) raised to the fourth power. However, as each return is in fact the 2-D integration of a signal over the acoustic footprint, the total attenuation is in only a function of the *square* of the distance (or time).

Some echosounders (e.g., the ODEC Bathy-1000) employ such a straightforward time-varying gain. Other echosounders (e.g., the DFS-6000) complicate matters by freezing the gain when the received signal exceeds a certain amplitude threshold. The gain then resumes increasing when the received signal drops below another threshold. The gain is frozen again for the secondary echo associated with the four-way reflection path. The idea behind such an irregular gain schedule is that the gain is held constant throughout the duration of the return. Unfortunately, with incoherent backscatter and various noise sources, one cannot analytically determine when the received signal will cross any of the

thresholds. As a result, one cannot use *absolute* amplitudes to make meaningful comparisons of any two signals acquired with the same echosounder.

For this reason, it may be necessary to normalize the amplitude of the signal, thereby ensuring that all signals are guaranteed to attain some known maximum. Alternatively, one could normalize according to the energy in the return, but as the energy is a function of a time-varying gain that is starting and stopping at non-deterministic points, this may be difficult. Furthermore, estimating the energy in the return requires a definite duration to the signal.

Another advantage to the simple maximum amplitude normalization method is that it provides inter-echosounder robustness. That is, between multiple echo-sounders the gains may be calibrated slightly differently (or the thresholds may be set differently), such that the amplitudes have some systemic variation. Normalization allows for comparison of signals against class statistics gathered by different echo-sounders.

A final rationale for amplitude normalization relates strictly to the initial implementation on a DSP56001 processor. The DSP56001 is a 24-bit integer processor. Therefore, scaling is a very important issue. Each processing algorithm must be carefully designed such that no operation or algorithm can generate a result which exceeds a value of ± 1.0 (in fixed-point representation). Therefore, the data must often be pre-scaled downwards. However, since integer arithmetic is being used, pre-scaling invariably results in a truncation of bits. To minimize the amount of this loss, it is necessary to ensure the data starts with the maximum possible amplitude.

4.3.5 Spatial averaging

For a typical sounder having an 8° beam-width, the Fresnel footprint has a radius of 1m at a depth of 15 m. Therefore, at a boat speed of 4 knots (2 m/s) a completely new area is insonified each second. With a ping rate on the order of 5-20 Hz, there will be considerable footprint overlap between successive pings.

This suggests that spatial averaging is possible. The advantage of spatial averaging is that the SNR is improved, the effective beam width is increased, and (optionally) the number of traces to process can be reduced.

4.3.5.1 Footprint overlap

The number of pings to use in an average is dependent on the amount of overlap, and the desired spatial resolution. Obviously, boundaries between regions will be less distinct as the amount of spatial averaging increases, suggesting that using fewer traces is better. On the other hand, noise contributions are minimized as the number of traces used in the average increases.

The number of traces to use also depends on the depth — at greater depths there is a greater per ping overlap. At the same time, however, it may be necessary to decrease the ping rate to compensate for the longer acoustic path. As a general rule, the time between pings should be larger than twice the return time, so as to reasonably prevent the possibility of the next ping occurring before the current ping is complete. For if there were to be returned backscatter energy at a time corresponding to twice the depth, there would have to be energy being scattered from 60° off vertical, which is extremely unlikely given that the energy in the return is a modulated by the beam width and the backscatter profile, both

of which are strongly decaying functions of angle[1]. Furthermore, we can ignore any contributions outside of the window provided by the beam pattern from volume reverberation since the subsurface penetration is not very large (on the order of centimetres) at the relatively high frequencies being used (>200 kHz).

As a result, if one enforces the condition that

$$f_{ping} < \frac{c}{4h} \tag{4.35}$$

i.e., that the time between pings is greater than twice that of the first return at depth h, it follows that the percentage overlap $o(s)$ between footprints can be expressed as a function of the footprint separation

$$o(s) = \frac{1}{\pi}\left(2\operatorname{acos}\frac{s}{2} - \sin 2\operatorname{acos}\frac{s}{2}\right) \tag{4.36}$$

where the separation s between successive pings can be expressed as the ratio of the lateral distance traversed each ping (i.e., with a vessel speed of v m/s) to the radius of the acoustic footprint.

$$s = \frac{t_{ping} v}{h \tan\phi} \tag{4.37}$$

The overlaps at various depths are shown below in Fig 4.9:

Note that except at very shallow depths, the overlap can be maintained at greater than 90% by using an appropriate ping rate.

---

1. From the various Jackson papers [18],[20],[19], the backscatter strength at a 30° grazing angle is about -25 dB, and the beam strength modelled via Eq. (2.1) is about -30 dB or more (depending on the transducer), corresponding to a signal strength which is down about 20 bits from vertical. Traces are typically digitized using 12 or 16 bits.

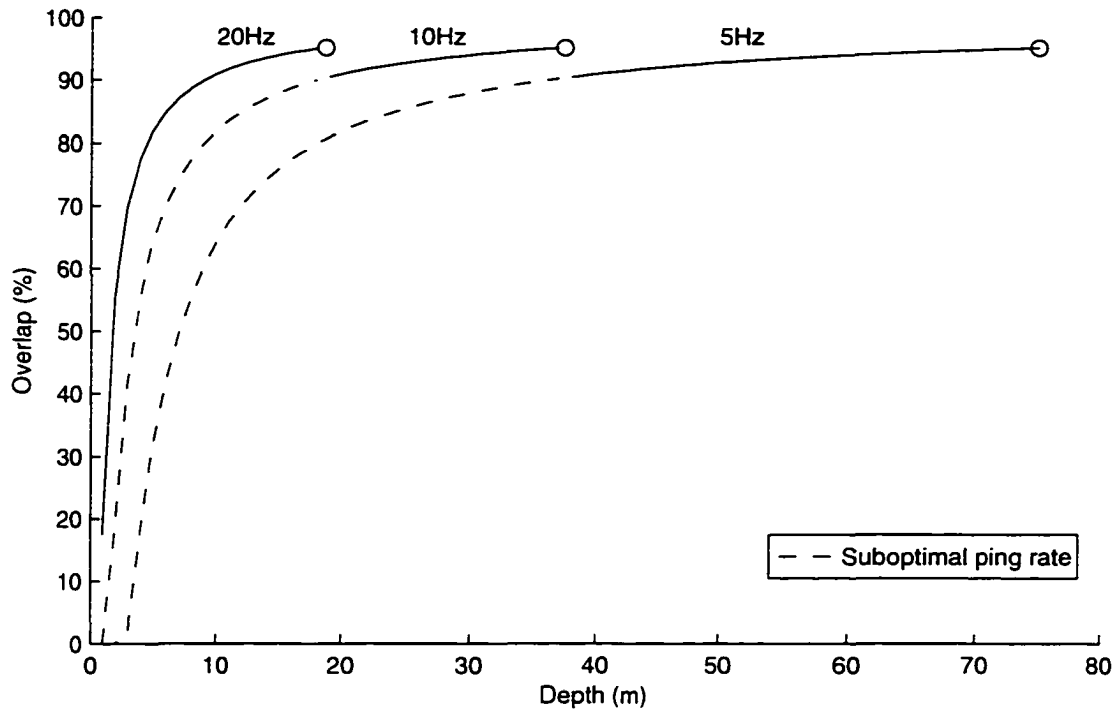


Figure 4.9: Per ping footprint overlap for different ping rates at 2 m/s boat speed

4.3.5.2 Beam widening

One of the advantages of spatial averaging is the effect of beam widening. This occurs if the incident angle changes from ping to ping, possibly due to the transducer undergoing pitch, yaw and roll. While it is difficult to characterize the widening for averaged traces comprising on the order of 10 pings, it is possible to examine the theoretical expectation. This can be defined in one dimension (e.g., if the transducer experiences only cross-track roll of angle φ) as follows:

$$E\{b(\phi, \varphi)\} = \int p(\varphi) b(\phi, \varphi) d\varphi \tag{4.38}$$

where the beam strength function $b(\phi,\varphi)$ is defined in Eq. (2.1), with the modification that ϕ defines the incident angle of the acoustic wavefront, which is independent of the orientation of the transducer (given by φ). In two dimensions, the incident angle and the trans-

ducer angle would have to be described using normal vectors, and the expectation would become a double integral.

Fig 4.10 below illustrates the expectation for a normal distribution of φ, with d = 15 m, c = 1500 m/s, f_s = 10 kHz, k = 8.8 cm^{-1}, a = 2.75 cm, and $\sigma_\varphi = 2°$.

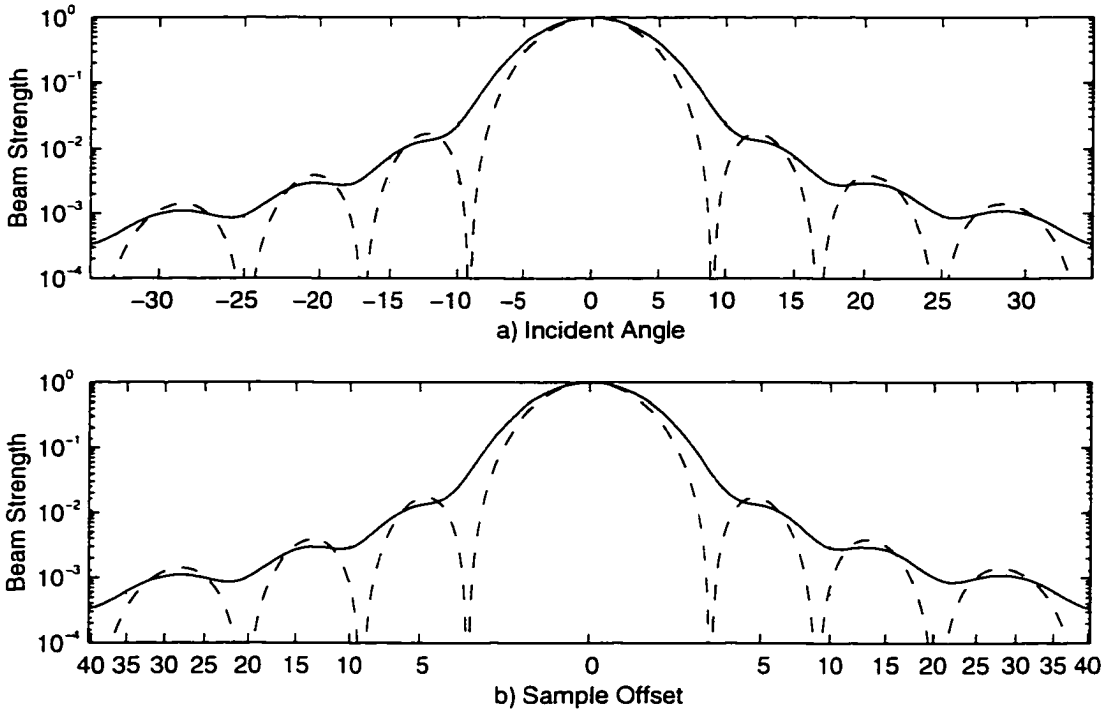


Figure 4.10: Result of normal distribution of off-vertical transducer angles on the expected beam pattern (solid line) of transducer with 8° beam width (dashed line).

The identical plots are given using different ordinate axes. This emphasizes that the non-linear relation between sample number and incident angle. It also emphasizes the benefit of beam widening, since the small off-vertical angles are otherwise very sparsely sampled.

Note that other random processes, including bathymetric variation of scatterers in the insonified annulus and variation in the scatterings angles will also contribute to the widening of the signal.

4.3.5.3 Averaging techniques

There are two methods by which an average can be generated. Perhaps the easiest method is to simply calculate the mean of N aligned signals. This generates an output every N pings. If an output is desired for every ping, then a moving average must be generated. A moving average can be generated in one of two ways: one can either create a buffer for N traces, and replace the N-th last trace with the current trace before calculating the mean at each time sample. The other method is to calculate an auto-regressive sum and scaling factor,

$$\begin{aligned} \mu_i &= x_i + \alpha\mu_{i-1} \\ s_i &= 1 + \alpha s_{i-1} \\ \bar{x}_i &= \frac{\mu_i}{s_i} \end{aligned} \tag{4.39}$$

where the scaling operation can be wrapped into the accumulation such that

$$\begin{aligned} \bar{x}_i = \mu_i &= \frac{1}{s_i}(x_i + \alpha s_{i-1}\mu_{i-1}) \\ s_i &= 1 + \alpha s_{i-1} \end{aligned} \tag{4.40}$$

An auto-regressive estimate has a couple of advantages over a moving average. First, it requires no buffering of past traces; second, the estimate can be made responsive to significant changes in the bottom type by allowing α to vary. This approach accommodates the traditional compromise between responsiveness and spatial resolution.

One can decrease, or even eliminate, the contribution of past samples to the current estimate of the mean by reducing the forgetting factor α_i. A suitable choice for α_i is the normalized cross-correlation of the estimate of the mean and the current trace, i.e.,

$$\alpha_i = \frac{\mu_{i-1} \cdot x_i}{|\mu_{i-1}||x_i|} \tag{4.41}$$

It may be desirable to further reduce the time constant of the estimate when there are marked bathymetric variations from one sample to the next. Such circumstances can occur when post-processing discontinuous data sets (in which there may be sudden positional jumps between traces), or when encountering pathological traces. For these cases, Eq. (4.41) can be modified as follows:

$$\alpha_i = \frac{h_\alpha}{h_\alpha + |\Delta h_i|} \cdot \frac{\mu_{i-1} \cdot x_i}{|\mu_{i-1}||x_i|} \tag{4.42}$$

where $|\Delta h_i|$ is the absolute value of the bathymetric change, and h_α is a constant appropriate to the expected bathymetric change. In data sets in which normal bottom gradients do not produce a depth change of more than a couple of samples per trace, assigning h_α a value of, say, 10 is adequate to ensure that the estimation process starts fresh with each distinct change in the bathymetry.

4.3.6 Trace alignment

Of course, if traces are to be averaged, jitter must be considered. While the ping-to-ping bathymetric change may be negligible, especially with a large footprint overlap, the accumulated change is still large enough to require that the traces be aligned prior to averaging.

If picking is done prior to averaging (see Section 4.2), the task of aligning becomes trivial — simply align each sample by their pick values. If time-scale normalization is implemented as discussed in Section 4.3.3.1, then this is done implicitly.

If averaging is done prior to picking, then more sophisticated trim statics are required.

4.4 Summary

A general procedure is given for analyzing the echosounder returns. The run-time sequence is illustrated in Fig 4.1. When first processing a data set, the modified sequence shown in Fig 4.1 is used, the only difference being the addition of a step for principal component analysis to determine the feature reduction matrices, and clustering to estimate the number of classes and their statistics.

The determination of the time of arrival of the reflection from the bottom is known as bottom picking. A threshold-based method is presented in which the bottom is defined as the last sample less than a threshold prior to the first sample above a second threshold. The thresholds can adapt in noise or when the return overlaps the transmit pulse saturation of the transducer.

Once the bottom return has been isolated, it is corrected for a variety of effects and distortions, the most significant of which are transducer angle, convolution with the source signal, linear dilation due to depth, and attenuation.

A novel deconvolution technique is presented which models the envelope generation function with a finite sum discrete convolution and the Hilbert transform of the source signal, as given in Eq. (4.6) through Eq. (4.13). A second-order Volterra kernel can be derived using a standard predictor network with constrained optimization.

Time-scale normalization is used to correct for depth effects, and is accomplished through a linear piece-wise interpolation of the echosounder return with a resampling period defined by Eq. (4.28).

Finally, spatial averaging can increase the effective beam width.

5 Analysis

This chapter is the second of two chapters which discuss the means by which an acoustic return is analyzed and classified. This chapter continues the processing steps outlined in Section 4.1 for signals that have been bottom picked and pre-processed.

The chapter is divided into four major sections. The first deals with feature extraction, and in particular, some simple algorithms which are used in later examinations of the data, or in related research.

The second section discusses means by which the feature sets resulting from the application of the extraction algorithms can be combined and then reduced in dimensionality. A method of scaling different feature sets is proposed, as is a computationally efficient method of combining the scaled feature sets.

The third section briefly summarizes Bayesian classification and the metrics used for determining class assignments. The use of certainty or fuzzy membership is proposed as an alternative to hard-limited decision boundaries.

The last section discusses clustering from the perspective of seabed classification. In particular, existing methods of unsupervised, or blind, clustering are presented and then modified to accommodate the existence of classes with arbitrary normal distributions.

5.1 Feature Extraction

Feature extraction is the process whereby an input pattern is quantified, usually as a vector of characteristic values, or, features. The specific extraction algorithms are often chosen to take advantage of prior knowledge of the discriminating characteristics of the data, such as

noticeable differences in frequency content. But often the data is not well enough understood to prescribe specific methods. Furthermore, key information may be revealed in domains which are not necessarily obvious.

For this reason, when first examining a data set, it is often advantageous to use multiple general-purpose feature extraction algorithms.

5.1.1 Combining algorithms

Typically, the results of the multiple algorithms are simply concatenated, and those features containing redundant information, as well as those containing no information are pruned in the subsequent step of feature reduction mapping (see Section 5.2).

However, some care should be taken in selecting algorithms. Pre-processing of the data may have a large impact on which features are accentuated. Furthermore, certain algorithms simply may not generate useful discriminating information. That is, they strongly reflect *something*, but whatever it is is not indicative of bottom type. One of the constant questions that must be asked is whether a given algorithm is sensitive to depth artefacts which could not be (or were not) pre-processed.

Comparison of algorithms is discussed in greater detail in Section 5.1.3.

5.1.2 Basic algorithms

The following sections describe some of the more common algorithms, with a justification of why the algorithms were considered suitable candidates. Table 5.1 below itemizes some of the approaches, and whether they were applied to the data sets of Chapter 6.

The performance of the algorithms on real data is presented in Chapter 6, Results.

Algorithm	Applied?
Amplitude Histogram	✓
Quantiles	✓
Fourier Frequency Spectrogram	✓
Wavelet Packet Tree Decomposition	
Cumulants	✓
Integrated Curve Fitting	

Table 5.1: Feature extraction algorithms

5.1.2.1 Amplitude histogram

One of the common statements about the effect of bottom type on echosounder returns is that "the shape changes". Therefore, it seems appropriate to apply a number of algorithms which analyze the shape of a signal. One of the more common such algorithms is the amplitude histogram.

In this algorithm, a histogram is made of the amplitudes of the signal. Generally, the bins can be pre-assigned, given that the signal is known to be quantized to a fixed number of bits. However, doing so requires that the signal be carefully calibrated because an amplitude histogram with fixed bins is not invariant to scaling of the data. If the signal cannot be calibrated due to, say, the use of a threshold-based time-varying gain schedule, or the use of multiple, differing transducers, it may be necessary to normalize the amplitude of the signal (see Section 4.3.4). Alternatively, one could adapt the bin sizes to accommodate different data, but this is, in effect, normalization.

5.1.2.2 Quantiles

Quantiles are another method of quickly characterizing the shape of a signal. Quantiles differ from amplitude histograms, in that a histogram divides the *domain* of the amplitudes

into N equal bins, whereas quantiles divide the *range* of the amplitudes into N equal bins, such that each bin has equal probability.

That is, whereas a histogram returns the number of samples falling within each of a number of equally-sized bins, quantiles return the positions of variably-sized bins such that each bin contains an equal number of samples.

Milvang *et al* [10] found that quantiles provided reasonably good separation.

5.1.2.3 Fourier spectrogram

One of the most common engineering methods of analyzing signals is in the frequency domain. For pattern recognition applications, the spectrogram has the advantage of being insensitive to translations in the time domain.

The simple spectrogram has been found to provide very good results.

5.1.2.4 Wavelet packet tree decomposition

Wavelets are a method of performing time-scale analysis on non-stationary signals, in a manner analogous to the time-frequency analysis of the short term Fourier transform (spectrogram) or other transforms which approximate a TFD (e.g., using a bilinear time-frequency representation and the Wigner-Ville, cone, Choi-Williams, or RID kernels [40]). While analogies can be made between scale and frequency, wavelets have a number of advantages. First, there exist some extremely efficient implementations of the wavelet transform. Second there are a broad range of generating functions (or, wavelet *prototypes*) that can be tuned to a specific application. And finally, the wavelet transform's tessellation of the time-scale plane is non-constant, in contrast to the STFT's tessellation of the time-

frequency plane. In particular, the time and scale resolutions vary at different scales. Note however that increases in time resolution are at the expense of frequency resolution and vice-versa (i.e., the tiles in the time-scale plane have a constant area).

The wavelet packet transform is a more general transform in which arbitrary frequency resolution[1] can be achieved in either the time or frequency at any scale. This can be achieved by using a filter bank implementation, and propagating either the low- or high-frequency output to the next level, rather than just the low-frequency output.

A wavelet packet tree is a highly-redundant filter tree implementation in which both the high- and low-frequency outputs are both propagated and recorded [41]. Actually, the scalar value of each filters' output energy is recorded, generating $M = 2(2^m - 1)$ coefficients for an m-deep tree.

This technique was found to be extremely successful at characterizing seismic data, as well as some echosounder returns. Unfortunately, it did not prove appropriate to most echosounder returns. The reason for this change is that some of the early echosounder returns were for a Navitronics system with a beam width of 45° and a digitization rate of 50 kHz. Thus, at 20 m, the return was on the order of 500 samples. Most of the later data was collected using echosounders that digitized at 10 kHz, and had beam widths as narrow as 8°, producing a return with a maximum length of tens of samples.

Unfortunately, with only tens of samples, only a few iterations of dyadic sampling are required to completely obscure the sample. And it was found that the higher scales (corresponding to lower frequencies) generally contained the best discriminating informa-

1. To within limits defined by the sampling interval and record length.

tion.

5.1.2.5 Cumulants

Larry Mayer's Ocean Mapping Group [1] reported good separation using 2nd- and 4th-order moments. With the data sets available to us, we were not able to consistently attain these results.

5.1.2.6 Other methods

Other methods that have been promoted recently include various forms of curve fitting [3],[24] of the integrated[1] signal. The logic behind the use of integration is that the SNR ratio increases throughout the integration. Unfortunately, integration is very sensitive to picking jitter. Furthermore, depth-related artefacts (specifically, time-scale dilation and the difficulty in deconvolving the source ping) cause the signal to change character with depth, most notably resulting in a sharper leading edge.

Despite this, Tan and Mayer [3] use a method which focuses attention on changes specifically in the leading edge of the return. The seabed classification system implemented by Quester Tangent Corp. (under the name QTC VIEW, see Section 6.1.2) includes a variation of the method which samples the integration at a much larger number of points farther down the curve.

These sort of methods were not applied to the data.

---

1. Actually, a cumulative summation.

5.1.3 Comparing algorithms

Feature extraction algorithms can be characterized by two main attributes, cost and effectiveness. Cost encompasses speed and memory requirements, both of which can drive up the price of a system. Cost can be readily estimated.

The best measure of effectiveness is the separability that is achieved using the algorithm. Unfortunately, there are a number of issues which make the comparison of separability difficult. First, there is synergy. That is, two algorithms may be by themselves unremarkable, but may contain mutual information. Thus when combined, the two algorithms may achieve excellent separation.

Secondly, assessment of separability resulting from individual algorithms requires accurate ground-truthing, as the separation between classes requires knowledge of the composition of the classes. Otherwise, assessment must be performed using the results of a clustering algorithm applied to the feature space of each algorithm. Thus, each algorithm may be compared using different *definitions* of the classes. Furthermore, clustering is sensitive to the distribution of data in feature space, which is the very feature that is being assessed!

Finally, it is questionable whether the individual comparison of feature extraction algorithms is appropriate. The divergence [42] between two classes cannot decrease with the addition of more dimensions. Furthermore, the capability of dichotomization of feature space increases with dimensionality [42]. Notwithstanding the process of feature reduction, this would suggest that adding more features via additional algorithms improves classification. Therefore, it would seem logical to use many algorithms to pro-

duce one single "optimal" order-*M* distribution for clustering. Algorithms can then be systematically excluded (by zeroing their features), and the effects or separation can be compared versus the "optimum".

This, of course, assumes that the feature extraction algorithms are not fundamentally unsuitable, thereby actually degrading classification by their addition. For example, some feature extraction algorithms can be shown to essentially reflect the depth which may turn out to be a dominant feature in a field of subtle differences. It is advisable to look at the results of each feature extraction algorithm to determine whether it is suitable. Visual inspection in two or three dimensions is adequate.

5.1.3.1 Use of the Mahalanobis distance as a separability measure

Some authors[1] use the Mahalanobis distance as their measure of quality. The advantage of the Mahalanobis distance is that the probability of misclassification can be derived directly from the measure (assuming equal *a priori* class probabilities). Unfortunately, the Mahalanobis distance is a statistical measure between a single point and cluster, or between two clusters having identical covariances. However, the clusters generated from real data often have covariances which are not even similar, let alone identical. Therefore it is more appropriate to use a different measure — such as divergence — rather than rely on the erroneous assumption that two covariance matrices may be generalized by taking their arithmetic mean.

The Mahalanobis distance was not used in this work as a separability measure.

---

1. E.g., Kavli *et al* [9].

5.2 Feature Reduction

There are two problems with clustering full-dimensioned feature vectors; the first is that the processing time is prohibitive, and the second is that the clustering is confounded by the fact that the majority of dimensions contain noise.

Therefore, it is desirable to reduce the number of dimensions in the feature vectors. Unfortunately, there is not an optimal way of doing so — the choice of axes to be used in the reduced space can depend on different criteria. For example, one may choose to operate within a reduced feature space in which the sum of the divergences between clusters is maximized. Alternatively, one may want to maximize the minimum spacing between clusters, at the expense of other separations. And of course, other criteria exist.

One significant complication associated with seabed classification is that the entire process is being performed without prior knowledge of what constitutes correct results. Therefore, classes have not yet been identified or described (see Section 5.4, Clustering) by the time feature reduction takes place. This fact prevents use of many optimization methods of feature reduction.

Therefore, in this section we propose the use of the principal components of the entire data set as the basis for feature reduction.

5.2.1 Principal component analysis

Since the class statistics are not known, most optimization methods of feature reduction are not applicable. Instead, we propose a simple approach which involves using the principal axes of the whole data set — that is, treat the entire data set as a single aggregate density and choose the major axes of this density's covariance ellipsoid.

The logic behind this approach is that there is often an approximately linear distribution of the means of the (as yet unidentified) classes. Therefore, a plausible choice for exhibiting good class separation is the axes along which the class means are positioned.

To illustrate the plausibility of such an approach, the aggregate variance along axis-i of two hypothetical Gaussian densities (each described by an axial mean, m_i, and variance, σ_i^2) is given by

$$\sigma_{agg,i}^2 = \frac{N_1\sigma_{1,i}^2 + N_2\sigma_{2,i}^2}{N_1 + N_2} + \frac{N_1 N_2}{(N_1 + N_2)^2}(m_{1,i} - m_{2,i})^2 \tag{5.1}$$

For densities having roughly equal memberships (i.e., $N_1 = N_2$), this simplifies to

$$\sigma_{agg,i}^2 = \frac{\sigma_{1,i}^2 + \sigma_{2,i}^2}{2} + \frac{(m_{1,i} - m_{2,i})^2}{4} \tag{5.2}$$

Furthermore, if each class is approximately the same size, (i.e., $\sigma_{1,i}^2 = \sigma_{2,i}^2$), the expression for the variance along axis-i of the aggregate density reduces to

$$\sigma_{agg,i}^2 = \sigma_i^2 + \frac{(m_{1,i} - m_{2,i})^2}{4} \tag{5.3}$$

So for data sets in which the covariances of each class are well-conditioned, the variances of each principal axis of the aggregate density will only differ by the separation of the means of the individual classes.

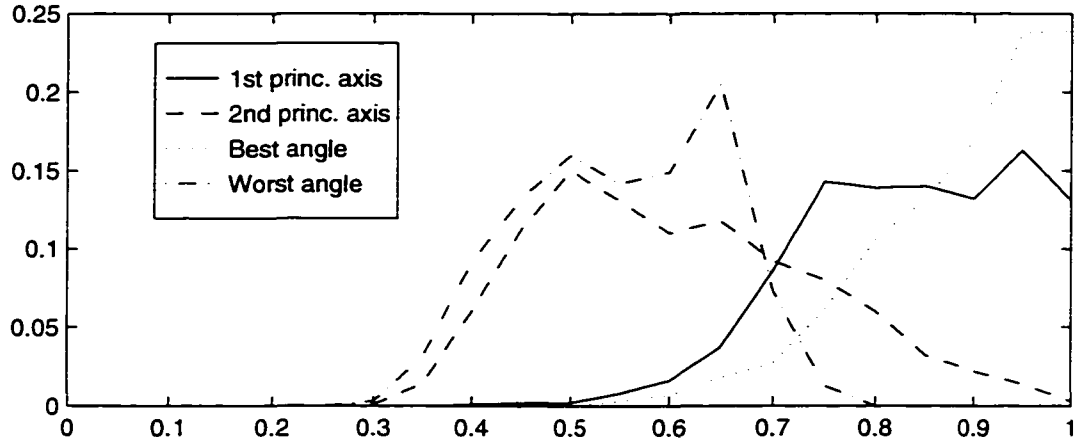
In other words, all other things being assumed equal, choosing the principal axes along which the aggregate density's variance is greatest is likely to yield the greatest difference between the means, and hence good separability. Principal component analysis provides identifies the dimensions having the largest variance.

Note that the derivation presented above assumes that information-bearing features have a larger variance over the entire data set than do features which contain only noise — i.e., there must be a reasonable signal-to-noise ratio. This can be ensured by choosing appropriate feature extraction routines, and by paying attention to algorithmic scaling (as will be discussed in Section 5.2.1.2).

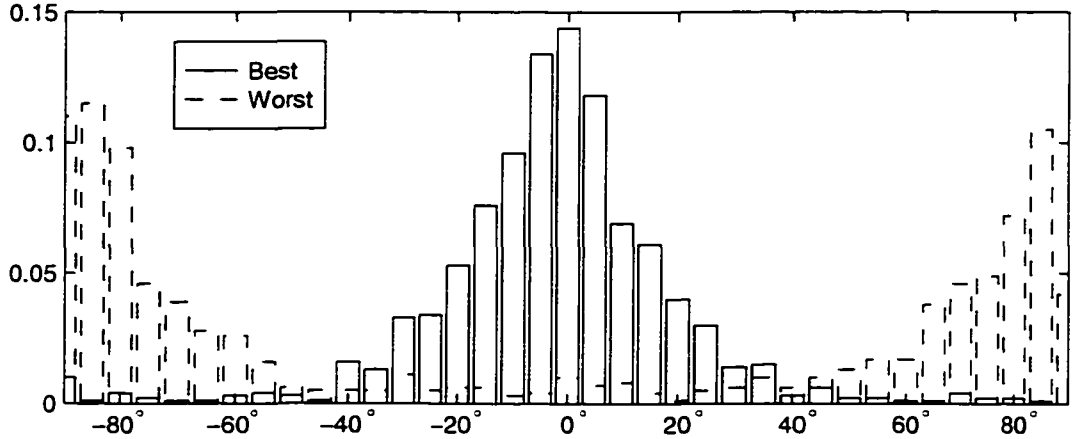
Furthermore, the justification for this method of feature reduction — i.e., the densities for each class are roughly spherical and differ only by co-linear offset — is generally not true. Nonetheless, the fundamental concept that separation of the means contributes to larger aggregate variance is valid.

The method can be further verified by a Monte Carlo simulation in which the correct classification rates of 1000 three-class 300-sample 2-D normal data sets were calculated for all possible projection angles to a resolution of 1°. The classification rates were recorded for the first and second principal components of the aggregate data set, as were the angles associated with the best and the worst classification rates. Fig 5.1a below shows the distribution of the correct classification rates as a density. Fig 5.1b shows the histogram of the best and worst classification angles relative to that of the first principal component.

As can be seen, choosing principal components based on aggregate variance alone did not necessarily produce the best results, but did represent a reasonable choice — the density of the optimum angle was centred around the first principal component, while the density of the worst classification angle was centred around the second principal component (i.e., orthogonal to the first principal component).



a) Density of classification rates for simulated data sets



b) Density of best and worst angles for simulated data sets

Figure 5.1: Monte Carlo simulation of PCA vs. optimum angle selection. Figure a shows the density of 1-D classification rates for the different projections, e.g., 0.13 of simulations were 95-100% classifiable using just the first principal component. This number increased to 0.24 of simulations when using the "best" angle. Figure b shows the location of the "best" and "worst" angles relative to the first and second principal components, e.g., for 0.14 of simulations the best angle was within ±2° of the first principal component.

5.2.1.1 Identifying principal components

Principal components can be found in two ways. The first is through the eigenanalysis of the covariance matrix of the data set. The second is through singular value decomposition of the raw data. Both will yield similar results for normal data[1].

1. The square of the singular values divided by the vector length equals the eigenvalues.

The number of dimensions of the reduced feature space depends on the data. Obviously, reducing the data to just one dimension is always possible, but the resultant classifier will probably have high rates of misclassifications. On the other hand, if a large number of dimensions are used then the less significant axes will contain mainly contributions from noise, and will actually degrade system classification rates.

Examining the sorted eigenvalues will give an idea of the suitable number of dimensions. By the time the eigenvalues have decayed a couple of orders of magnitude, their corresponding axes are unlikely to provide much information.

5.2.1.2 Algorithmic scaling

As mentioned above, use of eigenanalysis of the covariance matrix assumes that the axes with the most variance contain the most information about all classes together. For a single feature extraction algorithm this is a valid assumption, as each feature is weighted consistently within the context of the algorithm. For example, in a frequency spectrogram, one is relatively certain that if the feature representing frequency f_a has twice the magnitude of the feature representing frequency f_b, then it is likely that there is indeed twice as much energy at that frequency, or if the variance of f_a is twice that of f_b, then there is indeed twice the variability.

However, when using multiple feature extraction algorithms, the comparisons are not so intuitively obvious. For example, if one uses an amplitude histogram in addition to a frequency spectrogram, knowledge that the variance of the feature representing f_a is twice that of the feature representing histogram bin b_n provides us with little or no basis for decision making. Some measure of relative variability is required.

Furthermore, when the feature vectors are concatenated and then analyzed for variability, lack of attention to algorithmic scaling can result in one algorithm completely dominating the principal component analysis. For example, a histogram defined as a density has its bins scaled by $1/N$, which decreases its variance by $1/N^2$ — but does not change the amount of information the bins contain.

There are several methods by which the individual algorithms can be scaled, including normalizing by the product of the eigenvalues, normalizing by the sum of the eigenvalues, or simply normalizing by the maximum eigenvalue. However, it will be shown that normalizing by the sum of the eigenvalues has a number of advantages over the other methods.

First we consider normalizing the products of the F eigenvalues of an algorithm's output. The resulting scaling factor is calculated as follows:

$$s^2 = \prod_{j=1}^{F} \lambda_j^{(1/F)} \tag{5.4}$$

However, a serious problem of this approach is its sensitivity to the length of the feature vector. For example, if a Fourier spectrogram is used and it is found that all the energy is in a specific range of frequencies, the scaling factor defined above changes significantly depending on whether the full spectrogram is retained or only a subset of the coefficients (even though the discarded frequencies are known to carry no information). This can be illustrated by taking the log of Eq. (5.4), i.e.,

$$\log s^2 = \frac{1}{F} \sum_{j=1}^{F} \log \lambda_j \tag{5.5}$$

Thus, if the eigenvalues decay exponentially (linearly in a log scale) as is common, the scale factor will change significantly with each apparently insignificant eigenvalue.

In comparison, normalizing the sum of the F eigenvalues of the algorithm's generated features gives the following scale factor:

$$s^2 = \sum_{j=1}^{F} \lambda_j \tag{5.6}$$

This method does not suffer from numerical or representational complexity, and it achieves an asymptotic value for any algorithm with a only a few dominant eigenvalues (i.e., almost all real algorithms).

Furthermore, Eq. (5.6) has an advantage over the simpler method of normalizing by the maximum eigenvalue, i.e.,

$$s^2 = \max \lambda_j \tag{5.7}$$

in that it discriminates against algorithms which have a very low SNR. That is, if a feature extraction algorithm generates features which are effectively noise, then the eigenvalues will tend to be more uniform, and Eq. (5.6) will generate a scale factor which is less than that of Eq. (5.7) by up to a factor of the square root of the number of eigenvalues, F.

Note that none of these normalizing terms factor in the amount of real information in the feature vectors —they simply ensure that the principal component analysis is not dominated by a single algorithm, nor is it affected by scaling of data or of algorithms.

5.2.2 Defining the reduction matrix

The complete implementation of feature reduction requires the creation of a new feature vector $\boldsymbol{x}_{full}$, consisting of the concatenation of the N individual features sets $\boldsymbol{x}_i$. each having a dimensionality of F_i. The feature sets are normalized according to Eq. (5.6).

$$\boldsymbol{x}_{full} = \frac{1}{\sqrt{N}} \begin{bmatrix} \boldsymbol{x}_1/s_1 \\ \boldsymbol{x}_2/s_2 \\ \vdots \\ \boldsymbol{x}_N/s_N \end{bmatrix} \tag{5.8}$$

The covariance matrix of this new feature vector is then recalculated (at great expense), as are the eigenvalues and eigenvectors. The factor of $1/\sqrt{N}$ ensures that the sum of the eigenvalues of the new covariance matrix is maintained at unity.

It has been suggested that the covariance matrices of the individual feature sets can be combined to create a block-diagonal covariance matrix, thereby saving the memory- and computationally-intense step of recalculating the full covariance matrix. That is,

$$\boldsymbol{C}_{block} = \begin{bmatrix} \boldsymbol{C}_1/s_1^2 & & 0 \\ & \boldsymbol{C}_2/s_2^2 & \\ 0 & & \boldsymbol{C}_N/s_N^2 \end{bmatrix} \tag{5.9}$$

However, the eigenvectors of this matrix are also block diagonal,

$$V_{block} = \begin{bmatrix} V_1 & & 0 \\ & V_2 & \\ 0 & & V_N \end{bmatrix} \qquad A_{block} = \begin{bmatrix} A_1/s_1^2 & & 0 \\ & A_2/s_2^2 & \\ 0 & & A_N/s_N^2 \end{bmatrix} \tag{5.10}$$

resulting in each principal component (i.e., an eigenvector from V_{block} corresponding to a large eigenvalue in A_{block}) containing contributions from at most one algorithm. Therefore, if the dimension of reduced space is less than the number of algorithms, then some algorithms simply cannot be represented at all. Furthermore, if all the algorithms have a dominant eigenvalue, then each algorithm will be represented typically only once (by the algorithms' major principal axes).

In comparison, the covariance matrix resulting from the concatenated feature vector x_{full} contains components representing the individual covariance matrices as well as the desired cross-algorithmic covariance terms.

$$C_{full} = \begin{bmatrix} \frac{C_1}{s_1^2} & \frac{C_{12}}{s_1 s_2} & \cdots & \frac{C_{1N}}{s_1 s_N} \\ \frac{C_{12}}{s_1 s_2} & \frac{C_2}{s_2^2} & & \\ \vdots & & & \vdots \\ \frac{C_{1N}}{s_1 s_N} & & \cdots & \frac{C_N}{s_N^2} \end{bmatrix} \tag{5.11}$$

The resultant eigenvectors (and ultimately, principal components) now no longer segregate the contributions from individual feature sets.

Thus, it becomes a question of whether the use of x_{full} to create C_{full} is worth the

extra computation. For some data sets the principal vectors derived from both C_{block} and C_{full} may be quite different, yet result in similar distributions in reduced feature space. This illustrates the redundancy in the data provided by different algorithms. To illustrate, consider the following results from a real data set[1]. Two feature sets were used — the first

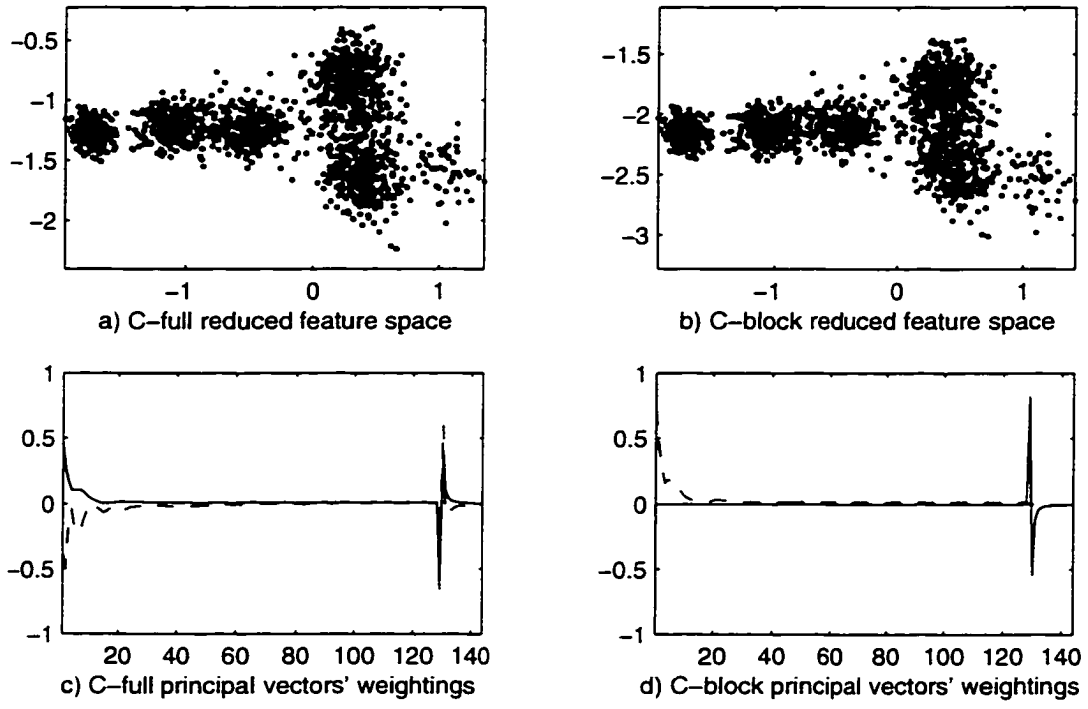


Figure 5.2: Comparison of full and block-diagonal reductions. Although different algorithms are used to generate principal vectors which used different weightings, the resultant feature space is very similar. This illustrates the redundancy in the data.

128 values of a 256-point Fourier spectrogram, and a 16-bin histogram. While the reduction of the full covariance matrix has slightly better separation than that of the block-diagonal covariance matrix, the results are reasonably similar.

1. This is the EDRD data set analyzed in Section 6.2.3. The purpose of its presentation in this section is to illustrate the concepts being discussed which are used throughout Chapter 6.

5.2.2.1 Order-*MN* reduction

Although some data sets do exist in which similar results are achieved using either the full or block-diagonal covariance matrices, this is not always the case. In this thesis, an approach was developed which can satisfy both criteria of supporting cross-covariances and greatly reducing the computational complexity.

First, an $M \times F_i$ reduction matrix was defined for each of the N feature sets, using M eigenvectors corresponding to the largest eigenvalues of the individual feature sets. That is,

$$\boldsymbol{R}_i^{(M)} = \begin{bmatrix} \boldsymbol{v}_{i1}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{v}_{iM}^{\mathrm{T}} \end{bmatrix} \tag{5.12}$$

A dimension-*MN* concatenated feature vector was then defined, consisting of the dimension-M reductions of each of the N feature vectors, each originally length $F_i \times 1$ (c.f., Eq. (5.8)).

$$\boldsymbol{x}^{(MN)} = \frac{1}{\sqrt{N}} \begin{bmatrix} \dfrac{\boldsymbol{R}_1^{(M)} \boldsymbol{x}_1}{s_1} \\ \dfrac{\boldsymbol{R}_2^{(M)} \boldsymbol{x}_2}{s_2} \\ \vdots \\ \dfrac{\boldsymbol{R}_N^{(M)} \boldsymbol{x}_N}{s_N} \end{bmatrix} \tag{5.13}$$

This order-*MN* concatenated data vector is then processed as normal (i.e., the covariance matrix is estimated, followed by eigenanalysis). For the example data illustrated in Fig 5.2, use of this approach has the effect of reducing the concatenated vector from 144

to just four dimensions. In terms of floating-point operations (based on unoptimized MATLAB code) the full vector required 89 MFLOPS, the block-diagonal method required 23 MFLOPS (but did not support cross-algorithmic terms), and the order-*MN* concatenation required only 0.1 MFLOPS to process 1500 length-144 feature vectors down to a final two dimensions.

This performance increase can be explained by noting that the covariance and eigenanalysis are both $O(n^2)$ algorithms, so that reducing from 144 to four dimensions should in theory provide three orders of magnitude improvement. However, there is overhead in the way the code was written, so the improvement is only two orders of magnitude.

The covariance matrix of the order-*MN* concatenation vector can be examined by expanding Eq. (5.13) for the case of two feature sets.

$$x^{(MN)} = \frac{1}{\sqrt{N}} \begin{bmatrix} v_{11}^{\mathrm{T}} x_1 / s_1 \\ v_{12}^{\mathrm{T}} x_1 / s_1 \\ v_{21}^{\mathrm{T}} x_2 / s_2 \\ v_{22}^{\mathrm{T}} x_2 / s_2 \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} v_{11}^{\mathrm{T}} & \mathbf{0} \\ v_{12}^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & v_{21}^{\mathrm{T}} \\ \mathbf{0} & v_{22}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} x_1 / s_1 \\ x_2 / s_2 \end{bmatrix} \tag{5.14}$$

and expressing the covariance as follows:

$$C^{(MN)} = \frac{1}{N}\begin{bmatrix} \frac{\lambda_{11}}{s_1^2} & 0 & v_{11}^{\mathrm{T}}\frac{C_{12}}{s_1 s_2}v_{21} & v_{11}^{\mathrm{T}}\frac{C_{12}}{s_1 s_2}v_{22} \\ 0 & \frac{\lambda_{12}}{s_1^2} & v_{12}^{\mathrm{T}}\frac{C_{12}}{s_1 s_2}v_{21} & v_{12}^{\mathrm{T}}\frac{C_{12}}{s_1 s_2}v_{22} \\ v_{21}^{\mathrm{T}}\frac{C_{12}}{s_1 s_2}v_{11} & v_{21}^{\mathrm{T}}\frac{C_{12}}{s_1 s_2}v_{12} & \frac{\lambda_{21}}{s_2^2} & 0 \\ v_{22}^{\mathrm{T}}\frac{C_{12}}{s_1 s_2}v_{11} & v_{22}^{\mathrm{T}}\frac{C_{12}}{s_1 s_2}v_{12} & 0 & \frac{\lambda_{22}}{s_2^2} \end{bmatrix} \tag{5.15}$$

Note that the submatrices associated with the individual algorithms have already been orthogonalized, yet there exists a submatrix filled with cross-algorithmic terms.

If one extracts an order-2 reduction matrix from $C^{(MN)}$, then we can expand it as follows:

$$\begin{aligned} x^{(2)} &= R_{(MN)}^{(2)} x^{(MN)} \\ &= \frac{1}{\sqrt{N}}\begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \end{bmatrix}\begin{bmatrix} v_{11}^{\mathrm{T}} & \mathbf{0} \\ v_{12}^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & v_{21}^{\mathrm{T}} \\ \mathbf{0} & v_{22}^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} x_1/s_1 \\ x_2/s_2 \end{bmatrix} \end{aligned} \tag{5.16}$$

which allows the reduction to be expressed as the multiplication of the full concatenated vector with an equivalent full reduction matrix.

$$x^{(2)} = \frac{1}{\sqrt{N}}\begin{bmatrix} r_{11}v_{11}^{\mathrm{T}} + r_{12}v_{12}^{\mathrm{T}} & r_{13}v_{21}^{\mathrm{T}} + r_{14}v_{22}^{\mathrm{T}} \\ r_{21}v_{11}^{\mathrm{T}} + r_{22}v_{12}^{\mathrm{T}} & r_{23}v_{21}^{\mathrm{T}} + r_{23}v_{22}^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} x_1/s_1 \\ x_2/s_2 \end{bmatrix} \tag{5.17}$$

Thus, unless the nature of the feature sets is such that all cross-algorithmic covariance terms are zero, each principal axis in the final order-2 feature reduction will clearly have components from two eigenvectors from each of the two feature covariance matrices.

The order-2 reduced feature space and reduction vectors are shown below in Fig 5.3. Note the excellent agreement in the distribution of the vectors throughout reduced

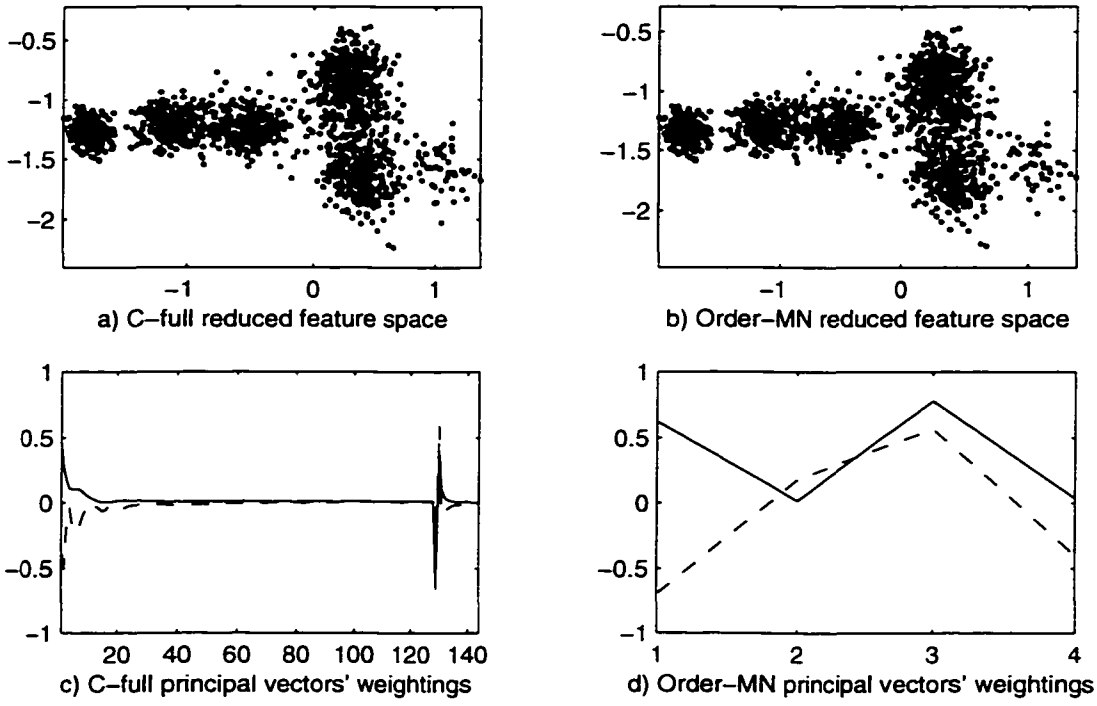


Figure 5.3: Comparison of full and order-MN concatenation reductions. The weightings shown in Figure d are those that map the intermediate 4-D reduction down to the illustrated 2-D feature space.

feature space. This agreement can be understood by examining the reduction matrices of the full vector and the order-*MN* concatenation method.

Here, the terms in the equivalent full reduction matrix of Eq. (5.17) are expanded and plotted in Fig 5.4a, and the difference between it and the order-*MN* concatenation are shown in Fig 5.4b.

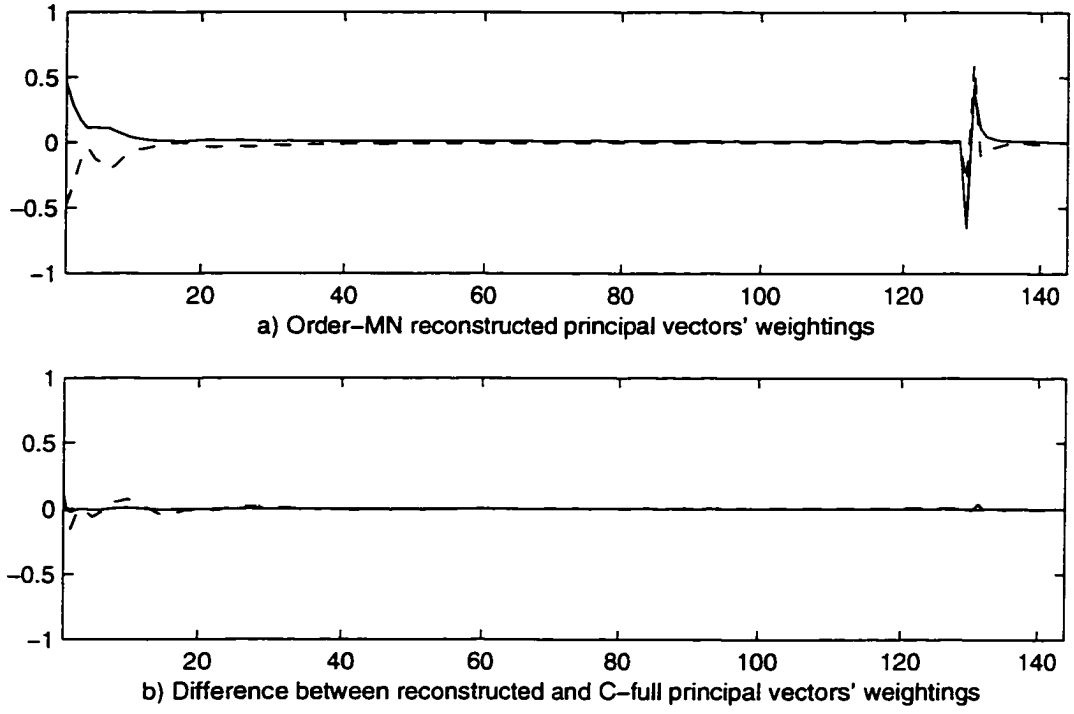


Figure 5.4: Equivalent full reduction matrix of order-MN method. The order-MN vectors are used along with the intermediate reduction vectors (not shown) to generate an equivalent one-step reduction mapping, which is very similar to the full aggregation principal vector weightings.

5.3 Classification

This section provides a brief overview of Bayesian classification for the purpose of introducing the terminology and symbols used later in this section. The interested reader is referred to Tou and Gonzales [42] for more information on game-theory based classification.

5.3.1 Bayesian classifier rule

Classification throughout this research is implemented by assigning a pattern to the class for which the conditional likelihood is greatest, i.e., assign pattern $\boldsymbol{x}$ to class ω_i iff

$$p(\omega_i)p(\boldsymbol{x}|\omega_i) > p(\omega_j)p(\boldsymbol{x}|\omega_j) \qquad \forall j \neq i \tag{5.18}$$

where the class probability can be assigned a priori, or estimated from the available data.

For the assumption of data with multi-variate Gaussian densities, the conditional probability that pattern occurred as a result of specific class is given by

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{(2\pi)^{M/2}|\boldsymbol{C}_i|^{1/2}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T\boldsymbol{C}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right] \tag{5.19}$$

where M reflects the dimensionality (length) of the pattern $\boldsymbol{x}$, and $\boldsymbol{m}_i$ and $\boldsymbol{C}_i$ are the estimated mean and covariance matrix describing the Gaussian class ω_i.

Use of Eq. (5.19) means that the calculation of the Bayesian classifier rule shown in Eq. (5.18) can be greatly simplified by taking the logarithm of both sides, yielding an expression of the form

$$\log p(\omega_i)p(\boldsymbol{x}|\omega_i) = \log p(\omega_i) - \frac{M}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{C}_i| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T\boldsymbol{C}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i) \tag{5.20}$$

From this, we can define the squared-Bayesian "distance" by dropping the constant expression of $\log 2\pi$ and grouping the other terms together as follows:

$$d_i(\boldsymbol{x}) = \frac{1}{2}h_i(\boldsymbol{x}) - Q_i \tag{5.21}$$

where the pattern-independent terms are

$$Q_i = \log p(\omega_i) - \frac{1}{2}\log|\boldsymbol{C}_i| \tag{5.22}$$

and the squared-Mahalanobis distance (i.e., the squared statistical distance of a point from the mean) [42] is given by

$$h_i(\boldsymbol{x}) = (\boldsymbol{x}-\boldsymbol{m}_i)^T\boldsymbol{C}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i) \tag{5.23}$$

The decision rule can then be rewritten as follows: assign $\boldsymbol{x}$ to the class with the

smallest Bayesian distance, i.e.

$$d_i(\boldsymbol{x}) < d_j(\boldsymbol{x}) \qquad \forall j \neq i \tag{5.24}$$

Applying this rule to the data used earlier in Section 5.2.2, the classification of feature space shown in Fig 5.5 occurs[1] (note that the class statistics are estimated using methods discussed in Section 5.4, and the boundaries between classes are the quadratic solution to equating the log likelihoods as expressed by Eq. (5.20) or Eq. (5.21) for each pair of classes).

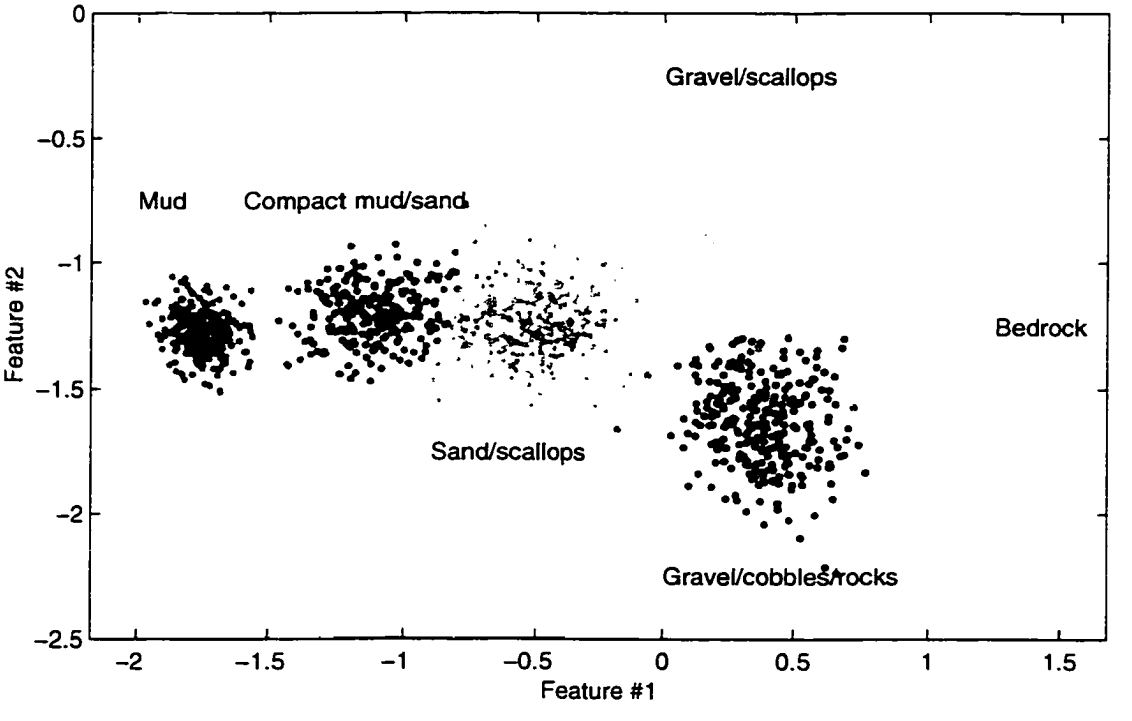


Figure 5.5: Classification of feature space. The labels attached to the classes were determined by visual inspection of ROV video.

For interest's sake, the classification results in feature space can also be mapped back against bathymetry or positional data for the source data. This is shown below in

1. The seabed types associated with the classes were identified by Dr. Poeckert of EDRD. For a more detailed analysis of this data set, see Section 6.2.3.

Fig 5.6. Fig 5.6b shows the proportional composition of the bottom, which was found to

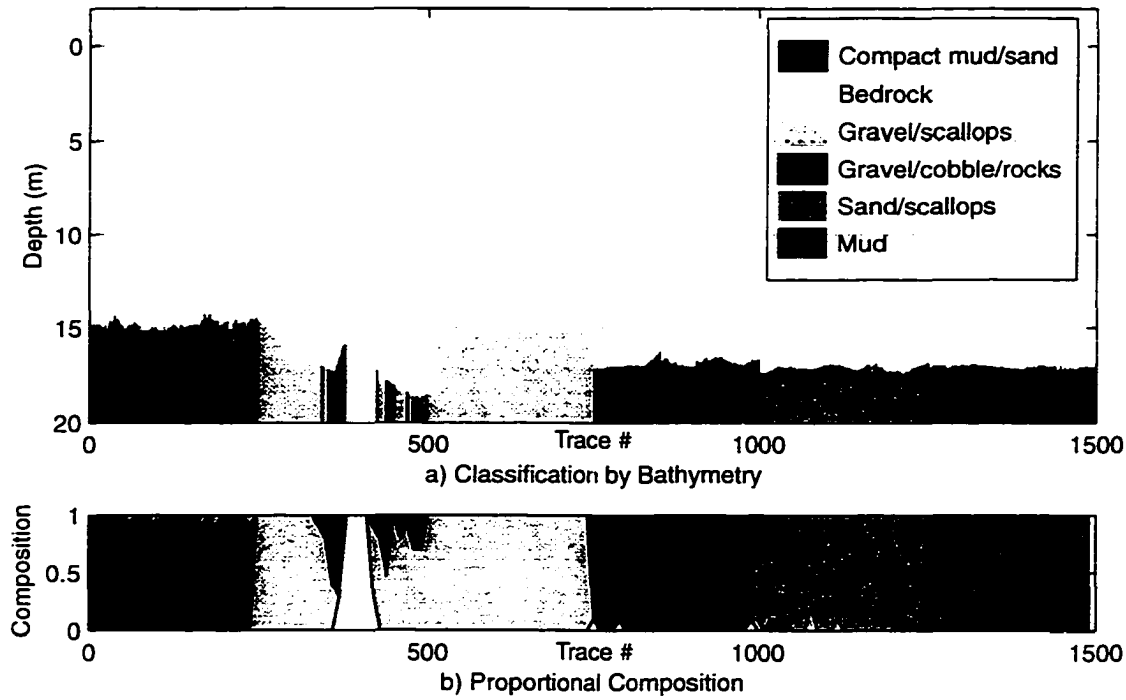


Figure 5.6: Classification versus bathymetry. There is very consistent classification in each of the six regions (with the exception of the rock outcrop around trace 400), despite the heterogeneous nature of the bottom types in some regions.

be a useful way to present information about the mixtures of seabed types. Each vertical strip of a proportional composition plot consists of a series of coloured segments (one for each bottom type) such that the segment lengths are proportional to the composition (i.e., classifications) within a section of the bottom (for example, a 50-trace window, which gives composition measurements with a maximum resolution of 2%). Proportional composition plots overcome many aliasing problems[1].

1. Most laser printers have insufficient resolution to print the 1500 traces of Fig 5.6a with a large enough half-tone screen to permit clear visual differentiation between grayscales. Similarly, most displays are simply incapable of displaying 1500 segments, resulting in some segments overwriting others, according to the specified colour order. To illustrate, consider traces in the range of about 1100. The Fig 5.6b indicates a 90%/10% mixture in places, but Fig 5.6a appears homogeneous.

5.3.2 Euclidean, Mahalanobis, and Bayesian metrics

It is important to note the difference between various distance metrics. The square of the Euclidean distance between two points can be calculated as sum of the squares of the differences between two points

$$d(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{m})^T(\boldsymbol{x} - \boldsymbol{m}) \tag{5.25}$$

The Mahalanobis distance differs fundamentally from the Euclidean distance in that space is transformed according to the shape of the statistical density, so that the distance represents the number of *standard deviations* away from the mean.

Thus, the Mahalanobis distance

$$h(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{m})^T\boldsymbol{C}^{-1}(\boldsymbol{x} - \boldsymbol{m}) \tag{5.26}$$

equals the Euclidean distance only when the covariance equals the identity matrix $\boldsymbol{I}$.

The Bayesian distance differs from the Mahalanobis distance in that it incorporates an offset (see Eq. (5.21)) representing pattern-independent information regarding the size of the distribution, as well as the *a priori* probability of the class. The former accommodates the requirement of the definition of a probability *density* that the integration over the plane equals unity[1]. The second merely offsets the distance according to the relative frequency that one class occurs compared to another.

Note that due to the offsets, the Bayesian "distance" can in fact be negative. Unlike the Euclidean and Mahalanobis distances, the Bayesian distance in only useful in a comparative sense. It should therefore more appropriately be termed a *metric*.

---

1. Thus, a density is not constrained to values between zero and unity, as is a probability.

5.3.2.1 Metric transformation

By applying a similarity transform, one can express the covariance matrix C, such that

$$C = VAV^T \tag{5.27}$$

where V is the matrix of eigenvectors and A is the diagonal matrix of eigenvalues. Similarly, the inverse of the covariance matrix can be written as

$$\begin{aligned} C^{-1} &= VA^{-1}V^T \\ &= (VA^{-1/2})(VA^{-1/2})^T \end{aligned} \tag{5.28}$$

This allows the Mahalanobis distance to be re-written as follows:

$$\begin{aligned} h(x) &= (x-m)^T C^{-1}(x-m) \\ &= (x-m)^T (VA^{-1/2})(VA^{-1/2})^T (x-m) \\ &= (x^*-m^*)^T (x^*-m^*) \end{aligned} \tag{5.29}$$

where the points have been linearly transformed [42] such that

$$\begin{aligned} x^* &= A^{-1/2}Vx \\ m^* &= A^{-1/2}Vm \end{aligned} \tag{5.30}$$

The advantage of this transformation is that it allows the Mahalanobis distance to be calculated as if it were the squared Euclidean distance, thereby avoiding any potential inversion problems associated with poorly conditioned covariance matrices.

Furthermore, MATLAB's efficiency in dealing with vector operations allow entire data sets to be quickly linearly transformed, squared, and summed as is required for distance calculation.

5.3.3 Certainty

A Bayesian classifier generates a single output for a given feature vector. However, the apparent unanimity of the decision can contribute to a false sense of confidence in the result, whereas in reality the classifier has simply identified the class from which it is most likely that the feature vector originated.

It is therefore useful to consider the *certainty* of classification. Here we define the *relative likelihood* of classification as being the likelihood that the class is assigned to class ω_i, as a weighted sum of the likelihoods of all the classes.

$$l_i(\boldsymbol{x}) = \frac{p(\omega_i)p(\boldsymbol{x}|\omega_i)}{\sum p(\omega_j)p(\boldsymbol{x}|\omega_j)} \tag{5.31}$$

The relative likelihood of the assigned class has a theoretical minimum value of $1/K$, representing a single intersection point of the boundaries separating K classes. But since there is essentially a zero probability that an intersection point involves more than three boundaries, the realistic minimum is 1/3. Even so, the vast majority of points with low relative likelihoods occur near the boundary between just two classes.

We then define the certainty of the classification as the maximum relative likelihood of the vector, scaled between zero and unity as follows:

$$c(\boldsymbol{x}) = \begin{cases} 2(\max\ l_i(\boldsymbol{x}) - 0.5) & \max\ l_i(\boldsymbol{x}) \geq 0.5 \\ 0 & \max\ l_i(\boldsymbol{x}) < 0.5 \end{cases} \tag{5.32}$$

The certainty can be examined in feature space, as well as relative to position or bathymetry. If feature space illustrated earlier in Fig 5.5 is mapped out in terms of certainty, Fig 5.7 results.

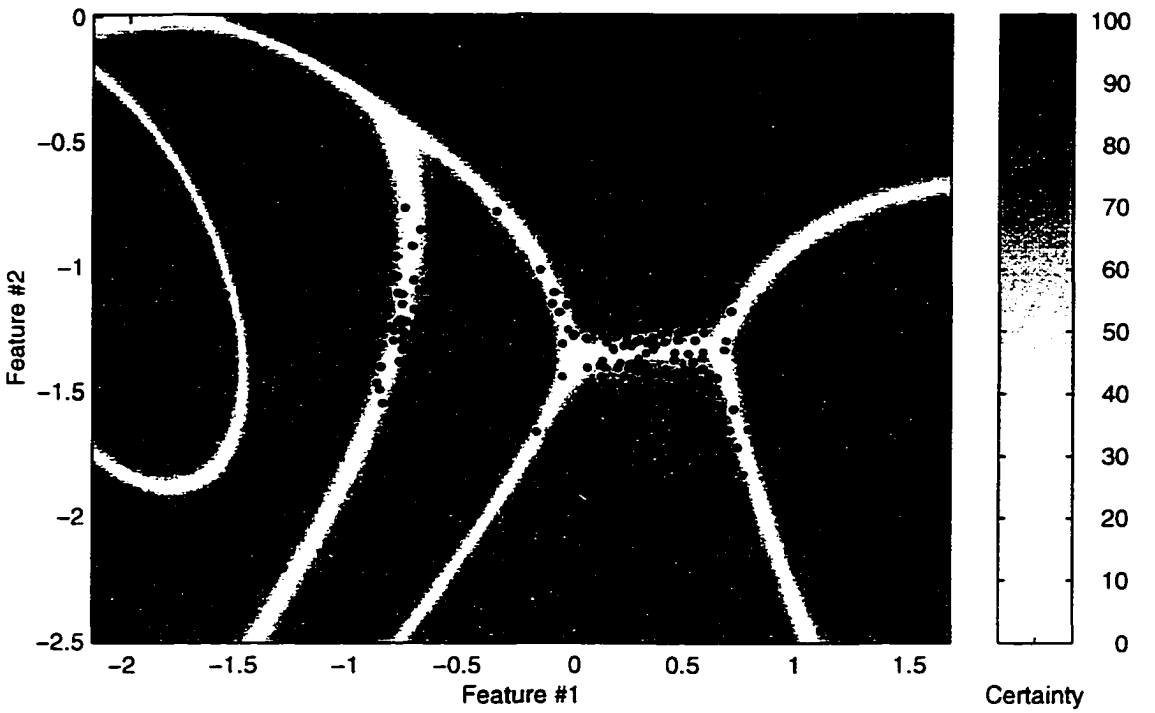


Figure 5.7: Certainty map of feature space. The only areas of uncertainty (bright) occur near the decision boundaries.

Feature vectors which lie in the brighter areas are less certain. Note, as expected, the areas of high uncertainty correspond exactly with the classification boundaries (c.f. Fig 5.5), and that only two relevant three-way intersections points occur, involving only a handful of points (only 0.3% of 1500 vectors have a certainty less than 0.5).

If certainty is mapped against the bathymetry profile, Fig 5.8 results. Here it becomes evident that areas of uncertainty occur in areas in which the composition is mixed (c.f. Fig 5.6).

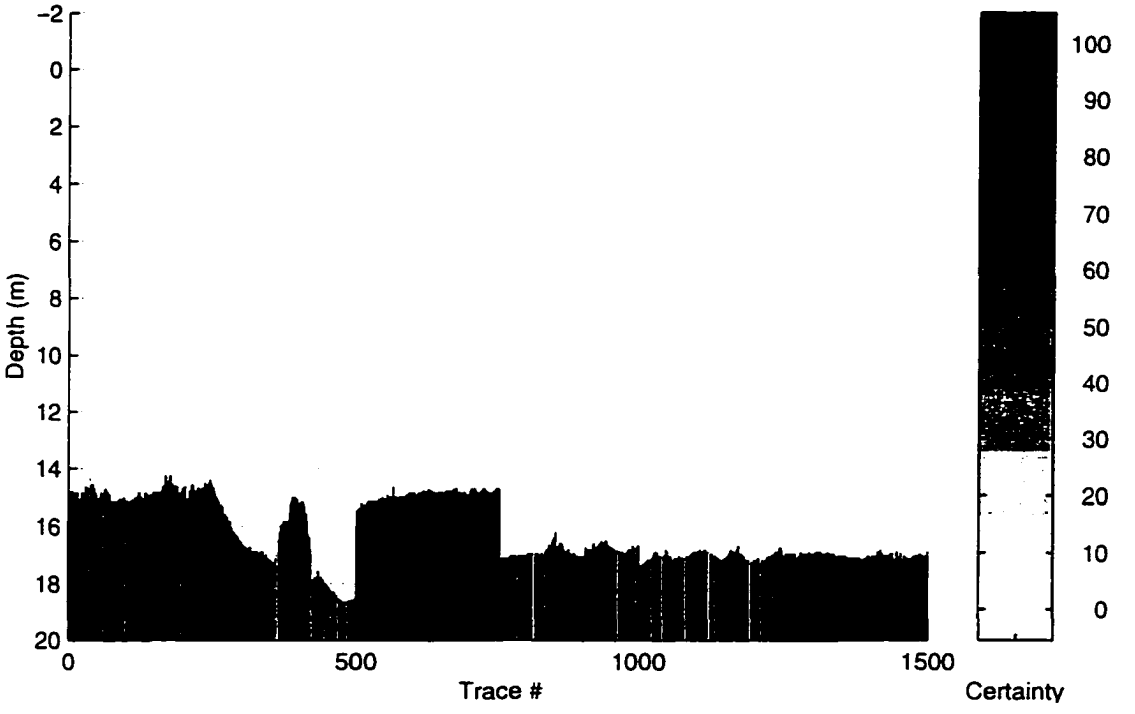


Figure 5.8: Certainty versus bathymetry profile. Areas of high uncertainty (bright) occur surrounding the bedrock outcrop around trace 400, and in areas identified by ROV video as containing heterogeneous mixtures.

5.4 Clustering

Clustering is the process of characterizing the statistical regimes in a data set. Sometimes just the statistics of the regimes must be estimated, and other times even the number of regimes must be identified.

This section is structured as follows: first, an overview of the approaches to clustering is provided. Then, the characteristics of practical algorithms are discussed. Finally, the ubiquitous K-means algorithm is modified, and then extended to form a proposed K-stats kernel, which can be used as the basis of a non-parametric clustering algorithm.

5.4.1 Supervised vs. unsupervised

There are two fundamental approaches to clustering. In the first, the user has a data set about which quite a lot of information is known. In particular, the correct classifications of the feature vectors is known. In this case, the user is able to supervise the clustering. Examples of supervised clustering are vector quantization [43], Kohonen self-organizing maps [44],[45],[46], and some optimization-based algorithms [47],[48].

However, in many applications, including seabed classification, there is often little or no prior knowledge about the classes to be studied (i.e., the number of classes, their statistics, and their *a priori* probabilities). For example, when surveying a new area, it may be inappropriate to assume that "hard rock" has certain acoustic properties. It may even be inappropriate to assume that the data set contains any examples of "hard rock".

Furthermore, variations in data acquisition equipment and procedures may mean that class statistics gathered from earlier data sets may be inapplicable to subsequent surveys.

Therefore, given the uncertainty regarding sample populations and variations in the data due to local morphology and data acquisition methodology, it is often prudent to make as few assumptions as possible about the existing classes, and to iteratively hypothesize some clusters on a training data set which is assumed to be representative. Such an approach progresses unsupervised and without the knowledge that the clusters are correct — relying only on the statistical validity of the hypothesized clusters. This approach is often called "blind" clustering.

Note that the costs associated with extensive sampling via core sampling, divers, or

ROV's make supervised clustering of seabed classification data impractical in all but a few cases, and certainly not an option for small-scale operators, such as commercial fishermen.

5.4.2 Algorithmic characteristics

There are a number of blind clustering algorithms [49],[50],[42], each with strengths and weaknesses. However, clustering algorithms suitable to real-time pattern recognition applications should all have a number of characteristics.

- The algorithm should terminate after a finite number of iterations.
- The algorithm should converge to a single solution — i.e., not converge to a cyclic solution.
- It should be obvious when the estimates of the cluster statistics have stabilized.
- The algorithm should not contain positive feedback.
- The algorithm should be robust (i.e., generate consistent solutions independent of starting points and specific compositions of the training data set).
- The algorithm should result in clusters with a meaningful interpretation in feature space.

The implications of these characteristics and the conditions or criteria that support these them are discussed in the following sections.

5.4.2.1 Memberships/statistics duality

One way to provide some of the characteristics of suitable algorithms is to require that there exist a duality between class memberships and class statistics — in other words, for fixed estimation and classification methods, a classification performed using estimated class statistics should produce the same membership list as was used to first estimate the statistics.

This duality has a number of ramifications. First of all, it results in algorithms that are finite — i.e., for any finite number of samples and classes, there are a finite number of permutations of the membership list, implying a finite number of possible class statistics.

Secondly, it becomes obvious when a solution has stabilized, since the membership list — and hence the cluster statistics — will not change from one iteration to the next.

Finally, duality may provide some insensitivity to initial conditions, because as soon as a particular state has been achieved — regardless of the states occupied earlier — the algorithm will progress deterministically. However, basic algorithm design is much more likely to affect sensitivity than the duality.

The duality criterion effectively eliminates those algorithms which are memory-based, i.e., in which the current estimates of the cluster statistics are based on prior estimates modified by contributions which are functions of the current and/or past memberships.

5.4.2.2 Robustness

Another attractive characteristic is that the algorithm be robust, i.e., relatively insensitive to initial conditions or to the particular composition of the training set used, provided that the training set is a suitable cross-section of the data.

5.4.2.3 Minimization

The solution should, as an exercise in optimization, embody some condition which gets minimized. For example, in a scenario involving normally-distributed classes, there are an

infinite number of solutions (parameterized by a scale factor in the most trivial case) which will yield a particular set of quadratic decision boundaries, and hence, membership list. It can be argued that the "correct" solution is one for which some other measure, such as the χ^2 statistics [51] for each class, is minimized. The technique advanced by Koontz and Fukunaga [47],[48] minimizes a metric of inter-class distances.

Although the correct boundaries between classes are generally not known, adherence to this condition prevents "ballooning" cluster definitions, in which one or more clusters exhibit positive feedback and engulfs all other classes.

Note that ensuring minimization also guarantees a single solution (as opposed to a cyclic solution).

5.4.2.4 Data immutability

Finally, it is desirable to have an algorithm operate such that the results at any stage have some sort of interpretation in feature space. This requires that the data itself be immutable.

Algorithms based on mutual gravitational attraction [52] usually involve adjusting the positions of each sample in feature space. It is possible for the resultant "globular clusters" to be significantly offset from the apparent source data. Furthermore, mapping the "globular cluster" memberships back to original data set may result in highly irregular boundaries.

5.4.3 A practical non-parametric clustering algorithm

In the subsequent subsections we present a practical non-parametric clustering algorithm which was used throughout the course of this research. The algorithm has at its core a generalization of the standard *K*-means algorithm.

5.4.3.1 *K*-means and generalized *K*-means kernels

One of the most commonly used blind clustering algorithms is the *K*-means algorithm [42], in which the estimates of the means of *K* clusters are iteratively refined. The inputs are the number of classes, and the initial estimates of the means. If the estimates are not provided, typically they are chosen to coincide with randomly selected members of the data set. The basic process of the *K*-means algorithm is illustrated in Fig 5.9.

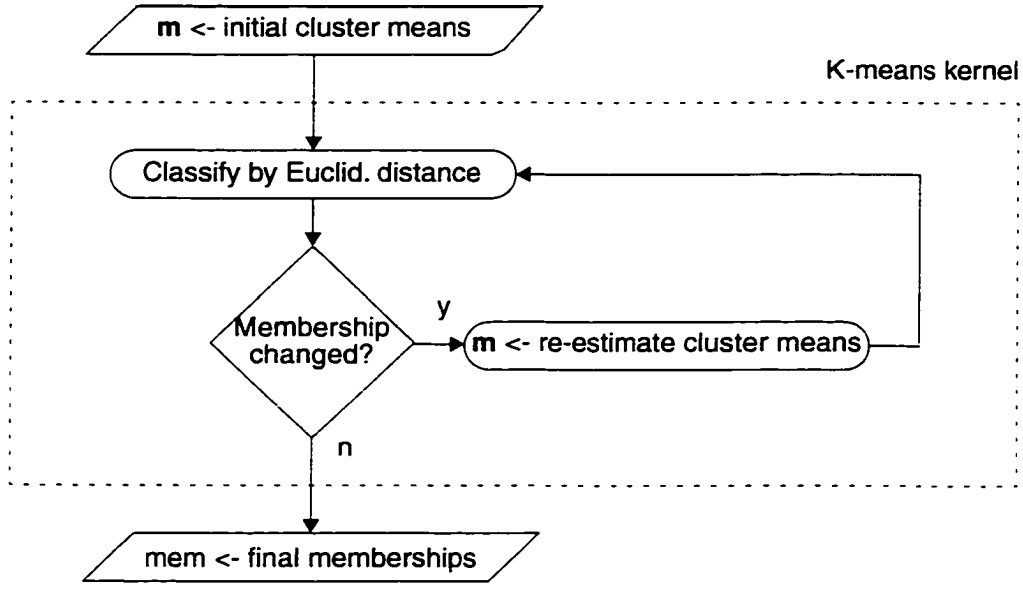


Figure 5.9: K-means kernel

Although the *K*-means algorithm is ultimately flawed, and unsuitable for all but a few data sets, it does have a number of characteristics which make it an attractive basis for re-engineering a better algorithm. In particular, *K*-means obeys the duality criterion (implying finite completion and detectable solution), and is a minimizing[1] algorithm (preventing positive feedback or cyclic solutions).

1. Mean Euclidean distance.

One of the fundamental problems with the K-means algorithm is that it uses Euclidean distance as the classification metric. This implies that the members of all clusters are assumed to have identity covariance matrices, which in turn has the effect that it generally results in inter-class boundaries which are incapable of partitioning feature space in agreement with the data. In fact, it is a trivial exercise to define two plausible and completely separable normally-distributed classes which the K-means algorithm simply cannot cluster correctly. Since the boundary between two classes with identical covariances is the straight line equidistant between the two means, reasonably correct clustering will occur only if this mid-point corresponds to an area which lies a sufficiently large number of standard deviations away from both classes' means. The sparsity of data points in this region ensures that slight variations in the boundary location do not significantly affect the estimates of the means.

Therefore, the first change that can be made to the K-means kernel is to generalize it to support other metrics than just the Euclidean distance. To do this we need to provide information other than just the initial estimates of the means of the K clusters, and then classify based on some other metric than Euclidean distance. For the case of Gaussian distributions, the generalized K-means (gK-means) kernel requires K initial estimates of the means and the covariances, and uses Bayesian distance as the classification metric. The gK-means algorithm is illustrated in Fig 5.10.

Note that the gK-means kernel does not make any assumptions about the covariance matrices, instead it simply requires that estimates be provided.

While the gK-means algorithm solves the biggest problem with the K-means algo-

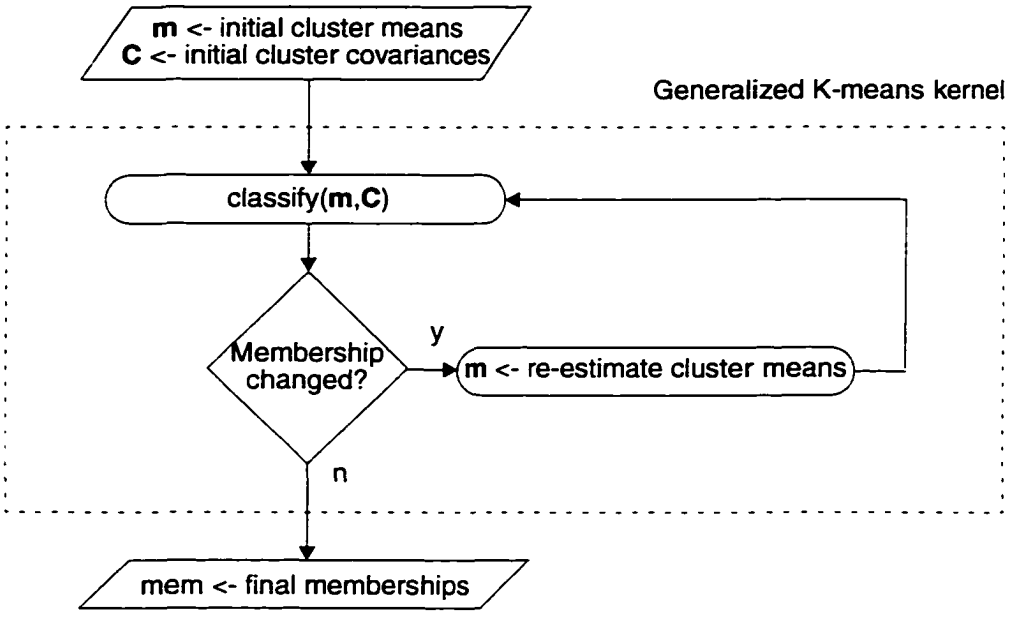


Figure 5.10: Generalized K-means kernel

rithm (i.e., the invalid assumption of identity covariance matrices), it still suffers from the second biggest problem — i.e., extreme sensitivity to initial conditions — that is, choosing different initial estimates may result in markedly different (and frequently, implausible) final memberships. This will be addressed later.

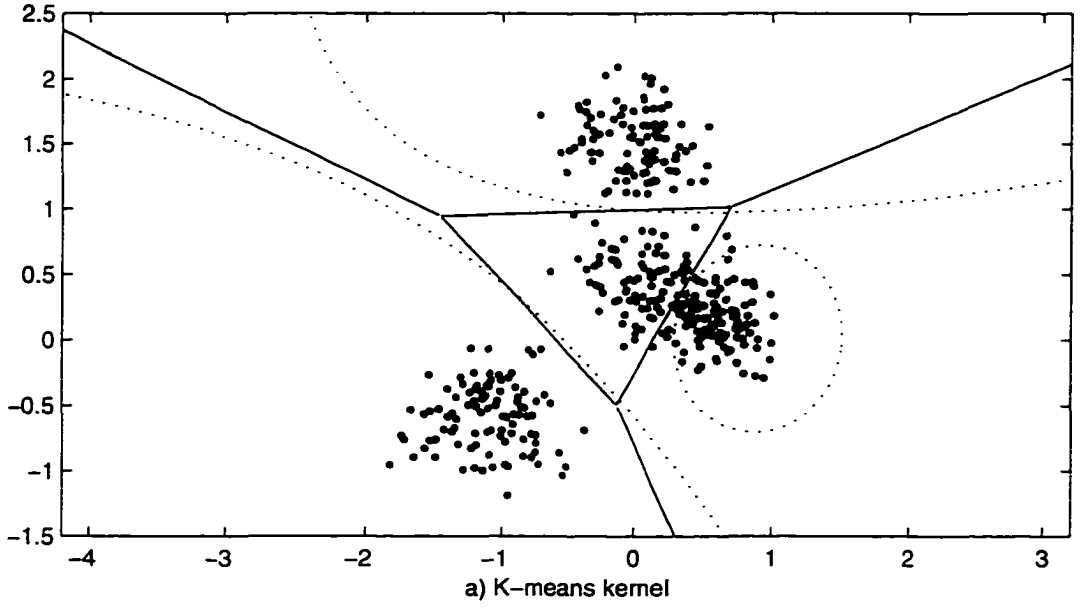
5.4.3.2 *K*-stats kernel

As the name implies, the g*K*-means algorithm still only modifies the estimates of the cluster means. To modify the covariance as well, there are two choices: one can either modify the g*K*-means kernel further to simultaneously re-estimate the means and covariances, or one can embed g*K*-means algorithm within a second kernel which modifies the estimates of the covariances.

However, if one modifies the kernel to simultaneously re-estimate both the cluster means and covariances after each classification iteration, it is not difficult to find examples

of data which show that the kernel can get stuck at locally stable but incorrect solutions.

To illustrate, consider Fig 5.11 in which a simulated set of classes has been clus-



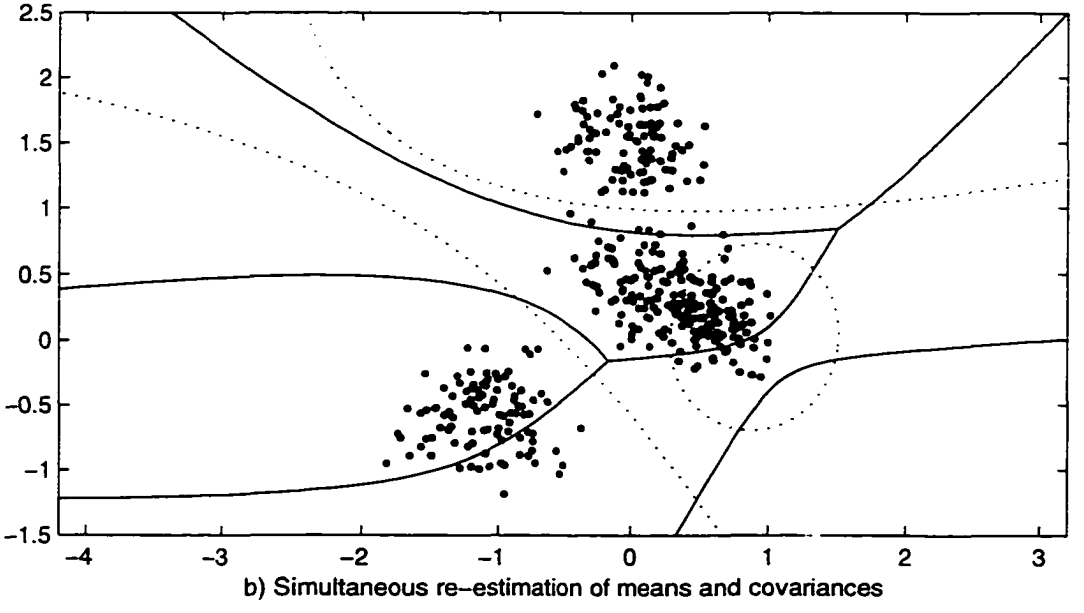


Figure 5.11: Classification boundaries illustrating stuck solutions. The dotted lines show the true decision boundaries, whereas the solid lines represent the result of the clustering algorithms.

tered using the K-means algorithm and the gK-means kernel with the modification that the covariance is simultaneously re-estimated as postulated above. For this particular data set, using the first four vectors as the estimates of the means causes the simultaneous re-estimation method to result in the incorrect solution shown in Fig 5.11b. The classes are otherwise very suited to the K-means algorithm, which achieves 92.5% correct classification with the generated clusters. Note that with better initial estimates of the means, the simultaneous update does in fact generate very good clusters, with a 94.5% classification rate. Nonetheless, the fact that the algorithm can and does get stuck renders it useless without some sort of annealing.

Separating the processes of mean re-estimation from covariances re-estimation does exactly this — the intermittent covariance update can kick the algorithm out of a local minimum. As the estimates improve, the covariances change less and less, resulting in a weaker "kick", as occurs in a simulated annealing cooling schedule [53],[54].

Thus, we can define the K-stats kernel (shown below in Fig 5.12).

The basic procedure is to identify stable cluster means via the gK-means kernel, and then re-estimate the covariances. Since a change in covariances may change the classification results, it is necessary to once again stabilize the cluster means via the gK-means kernel (using the updated estimates of the covariance). When the means are again stable, the covariances again may need to be re-estimated, and so on, until no change in the membership results from re-estimation of either the means or covariances. The excellent performance of the K-stats kernel (solid boundaries), relative to the true clusters (dotted boundaries), is shown below for the same simulated data illustrated earlier in Fig 5.11.

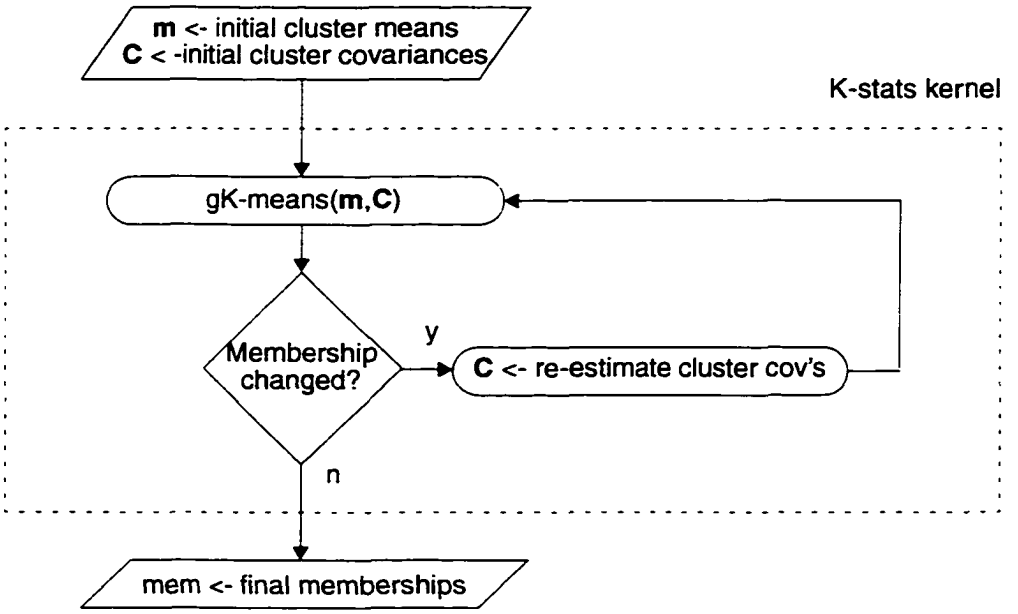


Figure 5.12: K-stats kernel

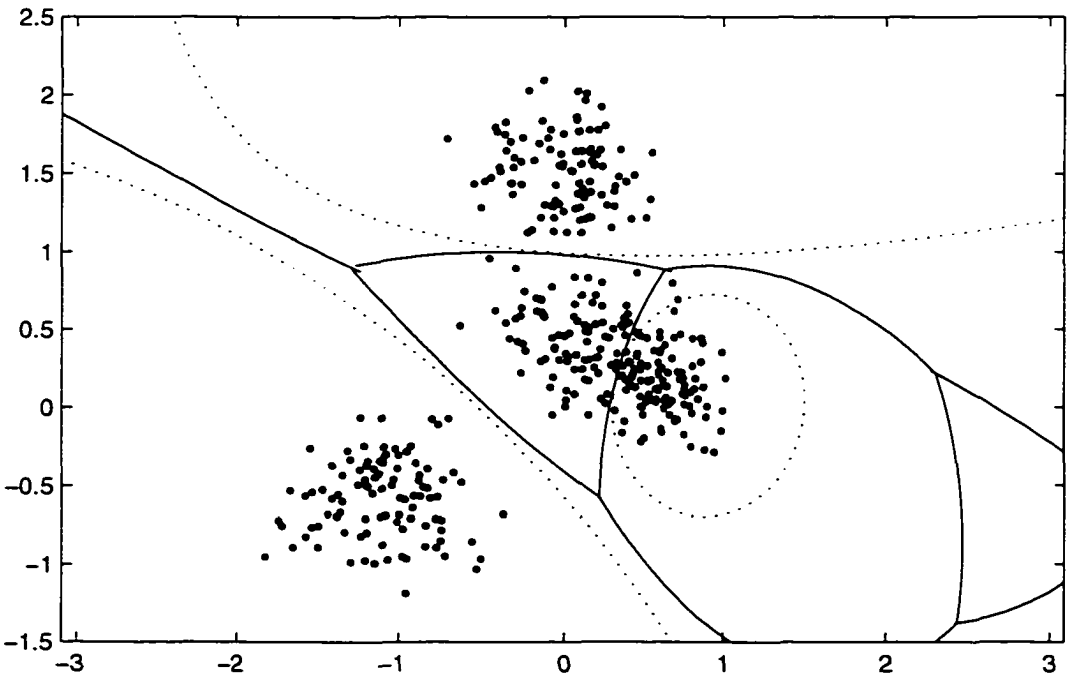


Figure 5.13: Classification boundaries derived using K-stats kernel. The dotted lines show the true decision boundaries, whereas the solid lines represent the result of the clustering algorithm. Note the excellent correspondence in regions containing data.

The K-stats kernel now provides a basis for completely characterizing K normally-

distributed clusters. The only parts that are missing are the assurance that the initial estimates of the means and covariances are plausible, and that the number of classes is plausible.

Ensuring that the number of classes is plausible naturally generates appropriate estimates, as will be shown below.

For a given stable configuration, one has to determine whether the correct number of classes have been identified. One way to do this is to determine whether the hypothesized clusters support the assumption of Gaussian data. The normalcy of a cluster can be ascertained with a χ^2 test [51]. If a single class encompasses more than one cluster it is very likely that the aggregate distribution is not normal[1]. Therefore, one can identify the clusters that require splitting as those having high χ^2 statistics.

Fundamental to this logic is that the current number of classes underestimates the correct number of classes. Overestimation will result in some classes being incorrectly identified as two or more separate classes — each with a high χ^2 statistic. In such cases, merging of classes — rather than splitting — is called for.

With regard to seabed classification, it is found that underestimating the number of classes, and then successively splitting individual classes usually provided sufficiently plausible results that merging was not required (Section 6.2.3). To prevent overestimation of the number of classes, only one class at a time is split. Another reason for not splitting all candidates simultaneously is that the χ^2 statistics of the various clusters are not independent — that is, one particularly bad hypothesis of a class may detrimentally affect the

1. Unless the encompassed clusters all overlap significantly, in which case they may not represent *distinct* classes.

accuracy of another class. And when the former is split, the overall plausibility may suddenly improve to the point where no more modification is necessary.

As to which class is the best candidate for splitting, the obvious answer is the class with the highest χ^2 statistic. Unfortunately, this may not be the case. The χ^2 statistic is itself a random variable, so that it is possible that a correct classification has a naturally high χ^2 statistic. Furthermore, there may exist two slightly overlapping classes which have a pathologically low χ^2 statistic. Therefore, it may be prudent to consider some heuristics, such as scoring the classes by some function such as the product of the χ^2 statistic and the number of points in the class. This approach will weight aggregate classes more heavily, with the tacit assumption that most classes have similar numbers of members.

As for ensuring the plausibility of the initial estimates; one can always start the algorithm with the ultimate in underestimation — i.e., with only one class. This then ensures the quality of the subsequent estimates of the class statistics is dependent on the mechanism in place for splitting classes. One way that this can be done is to take a class — whose covariance can be visualized as an ellipsoid — and replace it with two new classes centred around two new means offset from the old mean along the major axis by a distance equal to the square root of the principal eigenvalue. The principal eigenvalue of each of the new covariances is then scaled down by a like amount (i.e., the square root of the principal eigenvalue).

Other techniques can be used, but this has the advantage of splitting along a likely axis. The resultant means and covariances are reasonable first estimates of the class statistics, and can therefore be used in the *K*-stats kernel with confidence.

The complete *K*-stats unsupervised clustering algorithm is shown below.

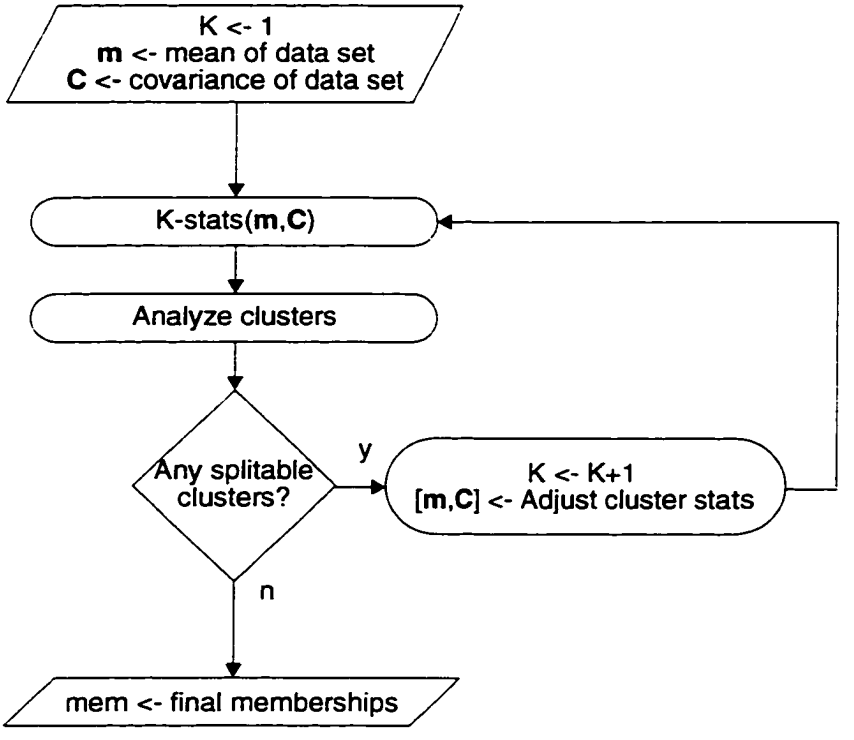


Figure 5.14: K-stats clustering algorithm

5.5 Summary

A bottom return is processed by extracting numerical features, reducing the dimensionality of the features, and then applying a distance metric to determine class assignment.

A variety of simple feature extraction algorithms are discussed. The major axes of the ellipsoid describing the aggregate data set are identified using principal component analysis. This approach is a reasonable starting point for feature space reduction when the statistics of the classes represented in the data set are not known. Feature sets can be concatenated provided they are scaled appropriately using Eq. (5.5). Furthermore, feature sets can first be reduced and then concatenated resulting in significant reductions in computa-

tional requirements while still permitting cross-algorithmic covariance terms.

Classification is performed by determining the Bayesian distance of a feature vector to each class mean, and then assigning the vector to the class for which the distance is minimized. Certainty, or the weighted sum of the relative conditional likelihoods of the pattern arising from each class, can be used to measure the quality of a classification.

Blind clustering is appropriate to seabed classification since high cost and/or risk make extensive ground-truthing impractical. Clustering algorithms should have a number of characteristics if they are to be practical. The *K*-means algorithm is fundamentally flawed, but is a good starting point for modification to the *K*-stats kernel, which is very robust when successively applied to a growing number of clusters.

6 Results

This chapter outlines some of the results of the application of the research into seabed classification.

The chapter is divided into two major sections. The first discusses the two forms in which the research has been implemented thus far. Both of the commercial implementations represent important contributions to the field. First, the "Pattern Recognition Toolbox for MATLAB" — the set of tools developed to support this research — has filled a significant gap for researchers and educators, finding application in marine acoustics, pharmacology and toxicology, seismology, and structural (fracture) analysis; as well as being used in university or college level courses in nine different countries. Second, a seabed classification system (a combination of hardware and software) was developed by porting the developed algorithms to a DSP56001 processor for real-time analysis of echosounder data. The product is proving to be very successful in an extremely lucrative market.

In the second section of this chapter the algorithms resulting from this research are applied to three distinct real-world data sets which were collected from both the east and west coasts of Canada. The results at each stage are discussed in context with the issues raised in Chapters 4 and 5.

6.1 Implementation Overview

6.1.1 Pattern Recognition Toolbox for MATLAB

Early on in the research it became evident that there were no tools available specifically designed for conducting pattern recognition research. As a result, tools would have to be developed, or else an existing development environment would have to be tailored for pat-

tern recognition research. However, by recognizing that any time spent developing tools occurs in lieu of time spent on actual research, the latter approach was taken, with MATLAB being chosen as the base development environment because of its extreme ease of use and extensibility.

The lack of pre-existing suites of functions (or "m-files") pertaining to classical pattern recognition was seen as a major gap in technology, and hence, an opportunity. As a result, all routines developed during the course of this research were carefully crafted with the specific intent that they could be part of a proper MATLAB "toolbox", consisting of routines that are consistent, well-documented, and flexible.

The toolbox emerged that emerged was turned into a product and is being currently marketed by Ahlea Systems Corp. as the "Pattern Recognition Toolbox for MATLAB". The analysis of data in this thesis makes extensive use of the toolbox. Documentation associated with the toolbox function calls are included as an appendix.

The author was solicited by The MathWorks (makers of MATLAB) to include the Pattern Recognition Toolbox in their "Connections" directory of third-party developers. In addition, the toolbox has been featured in The Mathworks' newsletter, "News and Notes".

Currently, international sales represent around 90% of the market.

6.1.1.1 Data representation

As mentioned above, the developed m-files were always intended to be part of a formal MATLAB toolbox. As such, they were designed for consistency and modularity. The following sections outline the conventions that are used in the toolbox (and consequently, throughout this thesis).

6.1.1.2 Notation

In this chapter, we use the following variables consistently

- M the dimensionality of a vector
- N the number of vectors in a data set
- K the number of classes

In examples of MATLAB code, the symbols M, N, and K represent the variables listed above, and m and c (subscripted if necessary to prevent confusion with M) will represent class statistics (either single, or multi-class lists).

6.1.1.3 Feature vectors and raw data

Individual records are always stored as $M\times1$ column vectors. Records may be combined into data sets by forming $M\times N$ matrices.

6.1.1.4 Class statistics

Class statistics are represented in one of two ways, depending on whether a single class or multi-class list is being represented.

In the case of a single class, the mean is represented as a $M\times1$ vector, and the covariance is represented by an $M\times M$ matrix.

In the case of multiple classes, the means of all the classes are compacted into a single matrix, with each *row* representing the transposed mean vector for a particular class. The covariance matrices too are compacted into a single matrix, with each row corresponding to a single class. However, this requires that the covariance matrix first be reshaped into a row vector.

The dimension of a multi-class list of means is $K \times M$ and the dimension of a multi-class list of covariances is $K \times M^2$.

The toolbox provides a number of routines for operating on multi-class lists of statistics.

6.1.1.5 Memberships

Memberships (i.e., the results of classification) are assumed to be $1 \times N$ vectors, since the process of classification really represents the ultimate in feature space reduction (i.e., $M = 1$).

Functions which accept memberships as arguments assume that the memberships only contain values between 1 and K.

6.1.1.6 Miscellaneous data

The general rule of thumb is that if there is one "thing" per record, then each "thing" is stored as a column vector. For example, class certainties are represented as a probability density consisting of K elements (the relative likelihood that a sample came from each class). In this case, the certainties for all records would be stored as a $K \times N$ matrix.

6.1.2 Seabed classification system

In addition to the development environment toolbox, a successful commercial implementation of the research algorithms is being manufactured and marketed by Quester Tangent Corporation of Sidney, B.C.

Implementation began in mid-1994 as part of an OEM[1] deal for the ODEC's

1. Original Equipment Manufacturing. In this case, providing a component to another manufacturer of hydrographic survey equipment.

Bathy-1000 digital echosounder, but continued through 1995 with the development of the stand-alone QTC VIEW Seabed Classification System. The choice of the DSP56001 card was made by the initial OEM partners. Otherwise, the extremely small memory space, the relatively slow clock speeds, the awkward I/O interface, the complete lack of development tools, and the 24-bit integer architecture would have been considered significant liabilities. The next generation of the QTC VIEW is currently being developed, and will be based on a much more standard Intel 486 architecture.

The market includes the world-wide equivalents of the Coast Guard, the Department of Public Works and Government Services Canada (responsible for maintenance of Canada's navigable waterways), the Department of Fisheries and Oceans, as well as private sector customers, predominantly commercial fishing fleets. There are approximately 2000 fishing vessels in the Alaskan fishing fleet alone. Given that the product sells for approximately US$17000 (at time of printing), even a relatively small percentage of the world-wide market represents sizeable sales.

6.2 Data Sets

6.2.1 Caraquet ISAH-S data set

Mr. Michel Goguen, the district hydrographer for the Atlantic Region Hydrographic Group of the Architectural and Engineering Services Marine department of Public Works and Government Services Canada (PWGSC), was kind enough to provide us with the Caraquet ISAH-S data set. This data set illustrates the difficulty in relying on automated methods when processing data with subtle features. Furthermore, it illustrates the importance of spatial averaging.

The data set covers a PWGSC dredging dump site. Dredged material from one area is taken by barge to a specific site and then dumped. The material — usually mud or silt — then sinks to the bottom where it covers the existing seabed. If the existing seabed already consists of mud or silt, there may be very little to distinguish between the native and foreign material.

This data set was collected by PWGSC and Quester Tangent Corp. to test the sensitivity of the QTC VIEW. The data set consists of 41 cleanly navigated tracks for a total of 15597 traces of 1650 samples each. The sampling frequency is 25 kHz, and the estimated speed of sound in water is 1490 ms^{-1}. The beam width and pulse length are not known, nor is the carrier frequency.

This data set was quite easy to process for a number of reasons. First, it was stored using an easily accessible file format, which made extraction of data trivial using either Perl or MATLAB. Second, there were no significant data logging problems (which had plagued some of the other data sets). Although there were some GPS gaps, the echosounder data appears to have been recorded correctly.

Fig 6.1 below provides a sample of the data. The most obvious characteristic is that the bottom response occurs near the end of the record — sometimes resulting in a truncation of the return. This meant that processing had to be performed on shorter records.

6.2.1.1 Picking

Bottom picking (see Section 4.2) was implemented as discussed in Section 4.2.2, and resulted in only two traces not being pickable. However, as these unpickable traces were not included is the final geographic selection, they were simply ignored.

Fig 6.2 shows the results of the picking plotted against trace number, and against

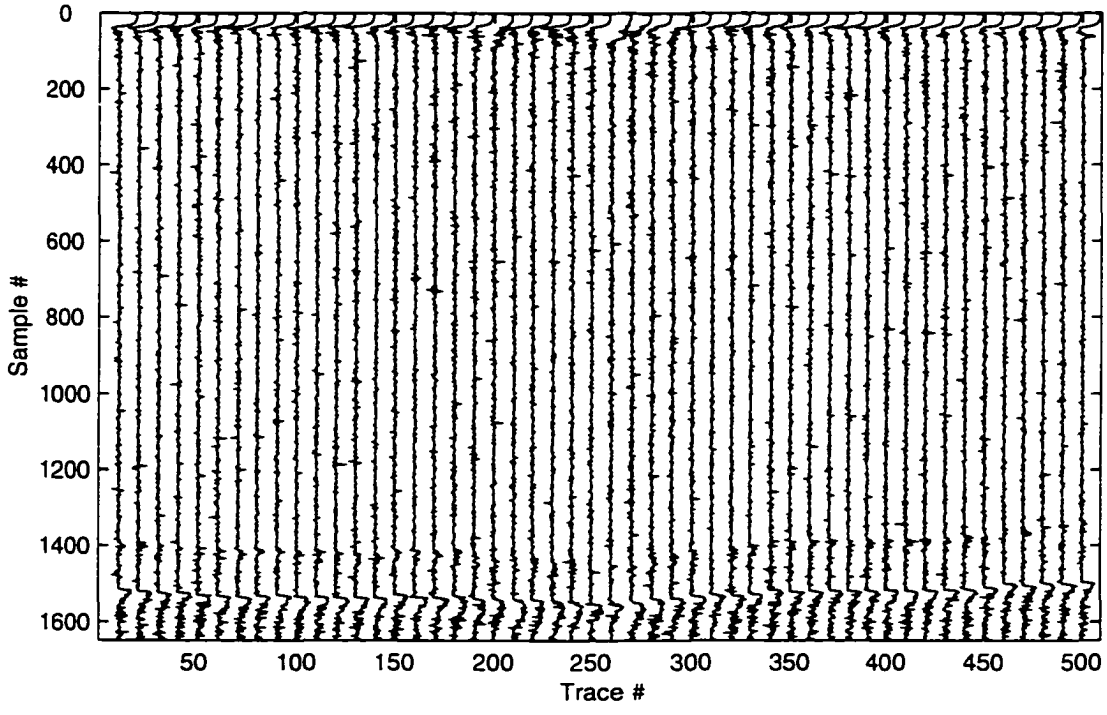


Figure 6.1: Caraquet raw data. Little data is collected after detection of the return (around sample 1525). The faint return prior to the main echo is believed to be due to cross-talk from another acoustic channel.

position information and interpolated contours.

The data set was then flattened. Given the relative uniformity of the bathymetry, time-scale normalization was not performed. The flattened data set is shown below in Fig 6.3.

The data set was also averaged as described in Section 4.3.5.3, with the exception that a constant value of 0.9 was used for the forgetting factor, α. The rationale for using a constant factor is that the normalized zero-lag cross-correlations between adjacent flattened traces (c.f., Eq. (4.41)) have an average value of 0.908 with a standard deviation of 0.016.

The averaged data set is shown in Fig 6.4. As mentioned earlier, one of the key

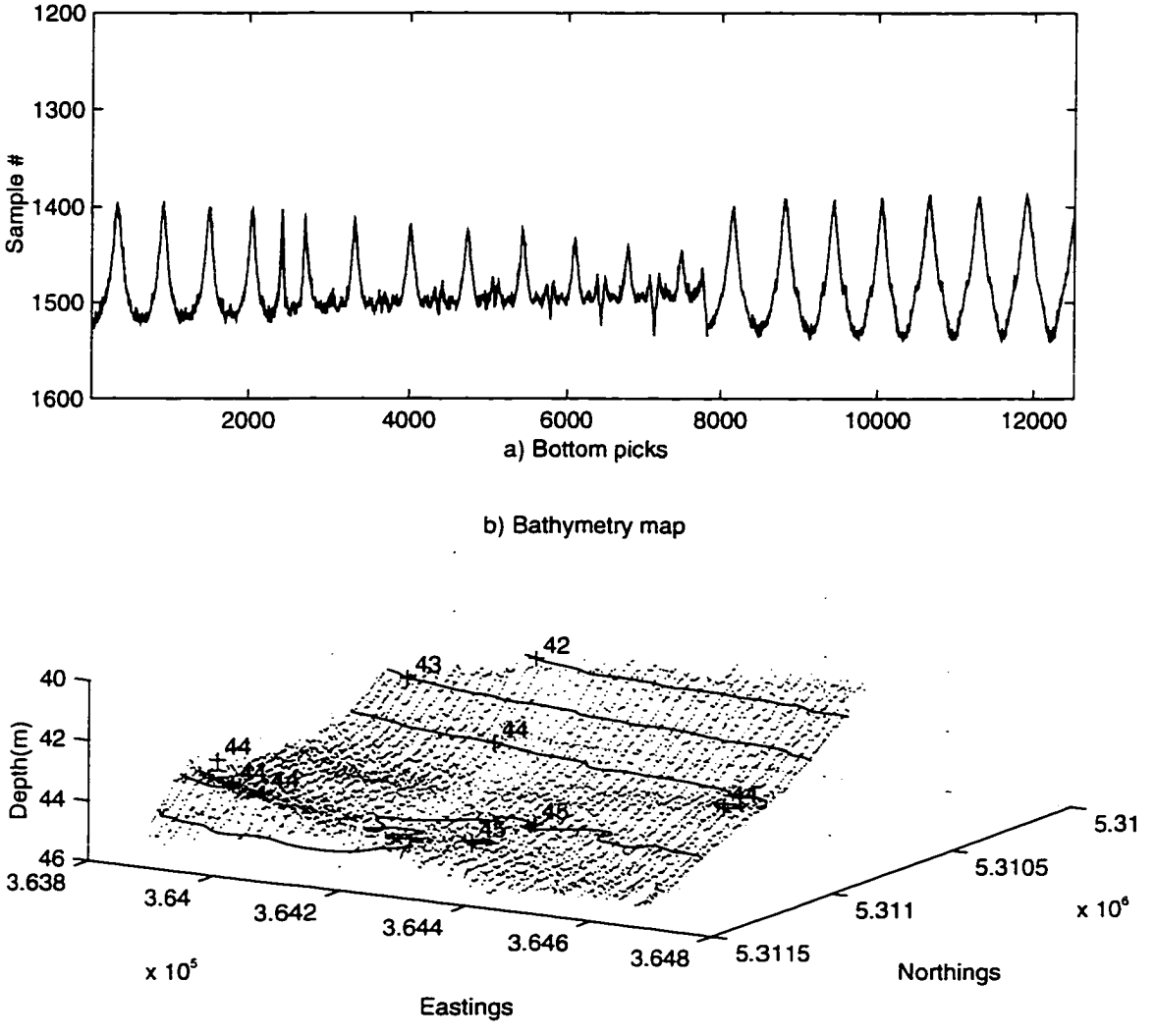


Figure 6.2: Picking results of Caraquet data set, plotted against trace number as well as draped over positional and bathymetric information.

characteristics of the data set is that the differences are subtle.

6.2.1.2 Feature extraction and reduction

The four simple feature extraction algorithms discussed in Section 5.1 were applied to the data. The resulting individual feature vector sets were each reduced to three dimensions (two, in the case of cumulants)[1]. These reduced data sets were examined individually, as

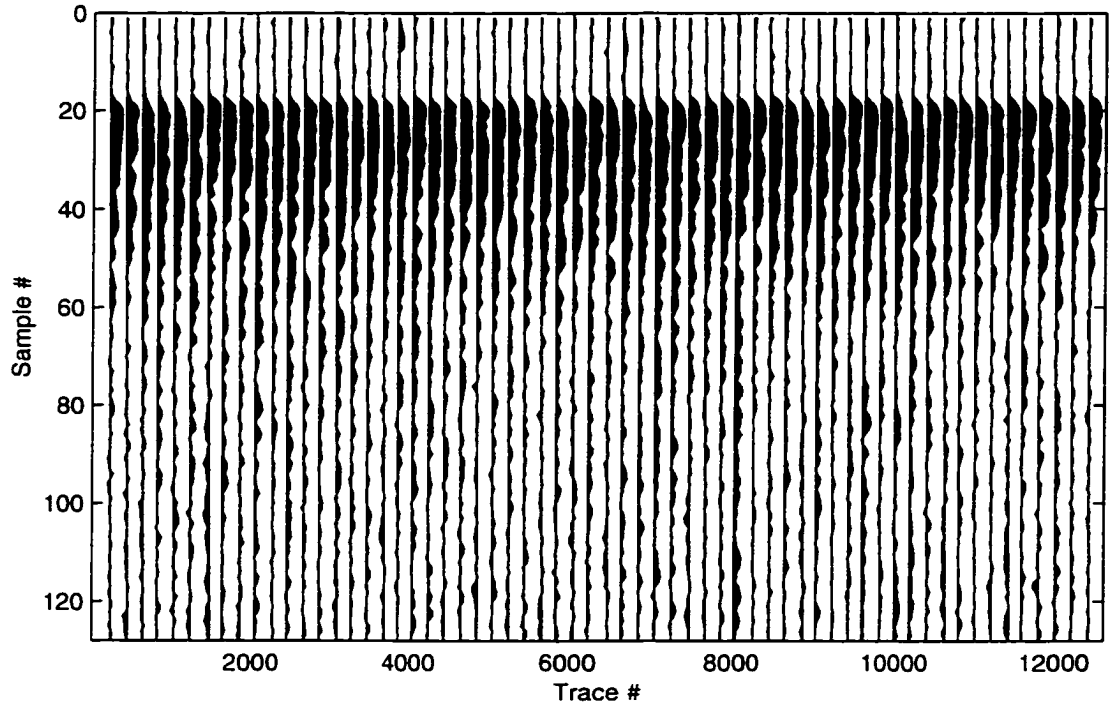


Figure 6.3: Raw flattened Caraquet data set. The sample numbers have been renumbered to correspond to the offset into the buffer of flattened records.

was the order-*MN* reduction of the aggregation of the individual reduced data sets (as per Section 5.2.1.2).

For illustrative purposes, both the raw and the averaged flattened data sets were analyzed. The results are shown in the figures below — Fig 6.5 shows the reductions to two dimensions of the individual feature vector sets for the raw data. All sets represent effectively normal distributions, making any clustering into more than one class arbitrary. In fact, only the histogram shows any spatial consistency — all other feature vector sets generate a random interspersing of classes when the clustering results are plotted using positional information. The aggregate feature vector shows no trends either — what little

1. Representing nominal values of $M = 3$, and $N = 4$. However, because the cumulant feature extraction algorithm produces only two features, the value of the "product" MN is in reality only 11.

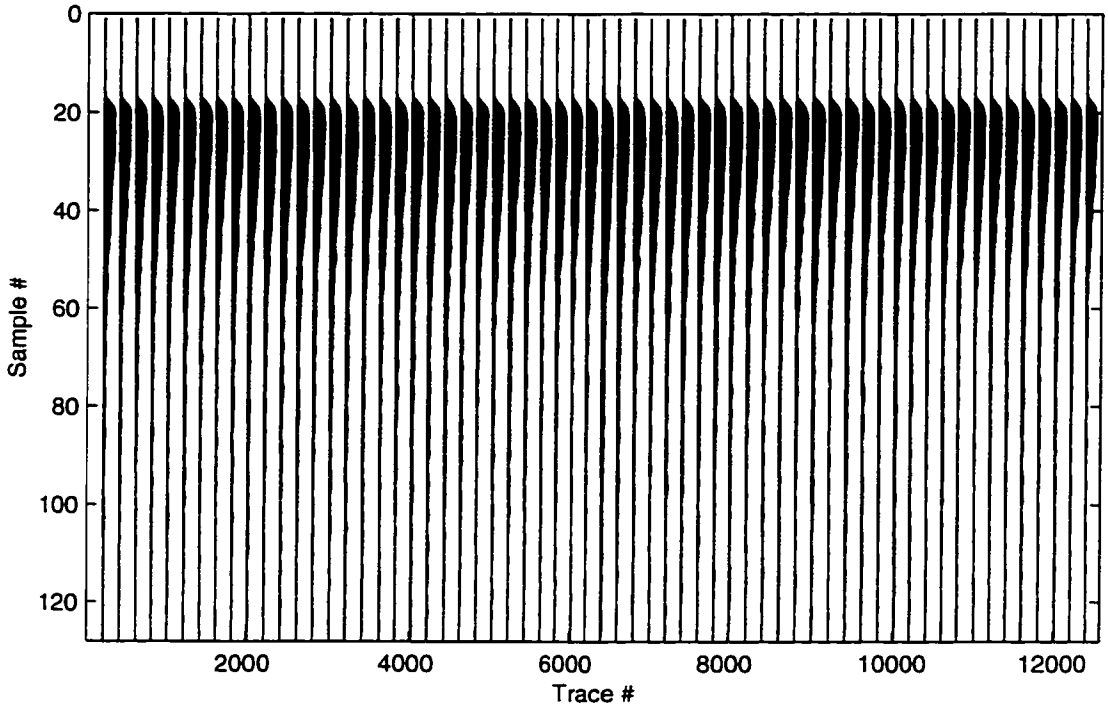


Figure 6.4: Averaged flattened Caraquet data set

information existed in the histogram apparently is diluted by the noise contributions of the other feature sets.

When the data set is averaged (as discussed in Section 6.2.1.1), distributions appear within the individual feature spaces, suggesting information about different bottom types is contained within the data. Fig 6.6 shows the feature space reductions for each of the individual feature vector sets.

. Again, the histogram appears to provide the most discriminating information. A cross-fusible stereo pair of the reduction to three dimensions is given in Fig 6.7[1].

The ramifications of using the order-*MN* reduction versus the full aggregate feature

1. For instructions on viewing cross fusible pairs, see Appendix C. These images were generated in MATLAB using a function written by the author which currently is currently available from The MathWorks ftp site.

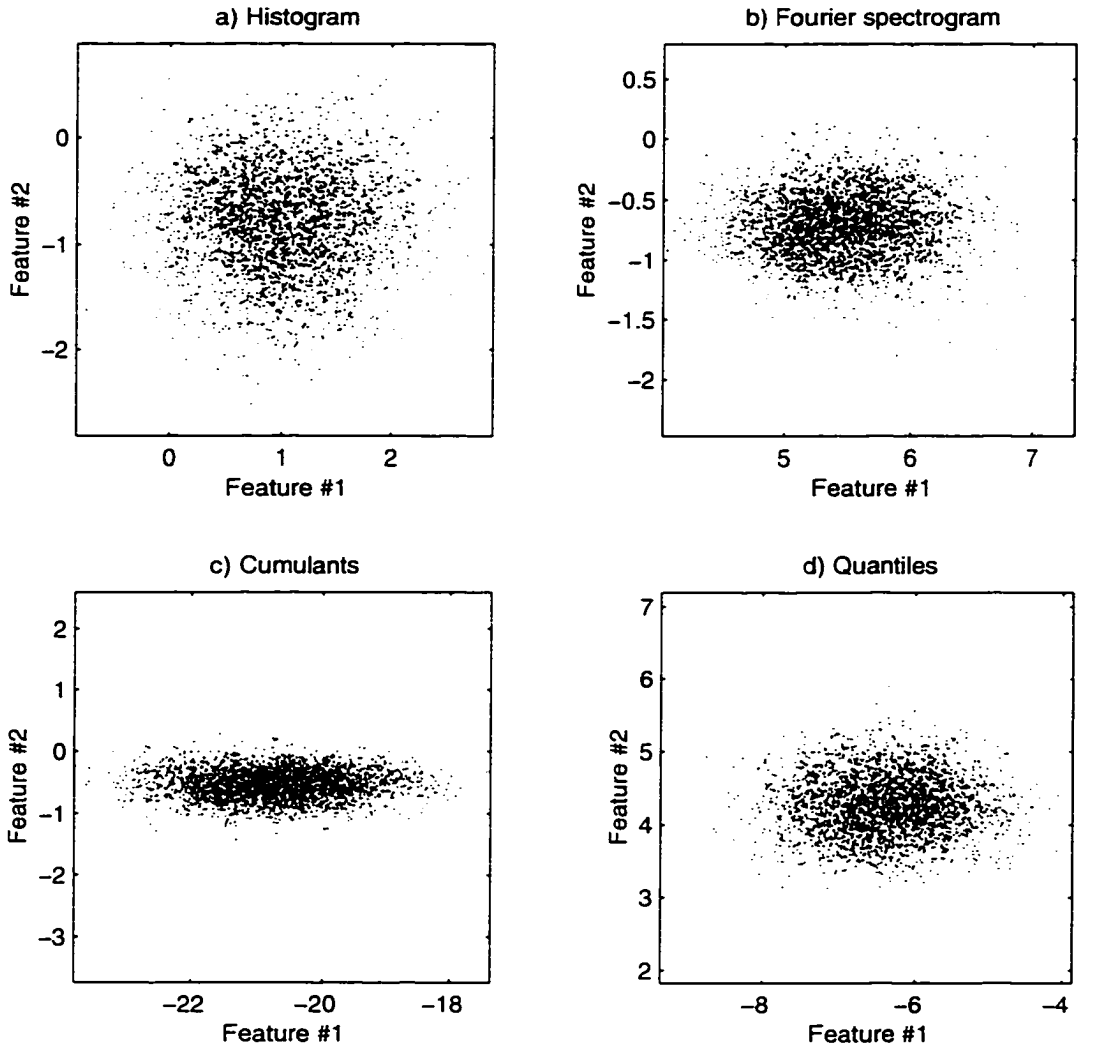


Figure 6.5: Reduced feature spaces of Caraquet raw data. The axes in each figure are the first two principal components of the feature sets, and are composed of the weighted sums of the individual features which give the maximum variance. Despite this, no separation between classes is discernible in any of the reductions.

set are illustrated by examining the sorted eigenvalues of the two feature sets. The first five values of each are tabulated in Table 6.1.

Note that for the full aggregate feature set, the remaining 93 dimensions contain the remaining 16.1% of the variance, whereas for the order-MN reduction the remaining 6 features comprise the residual variance.

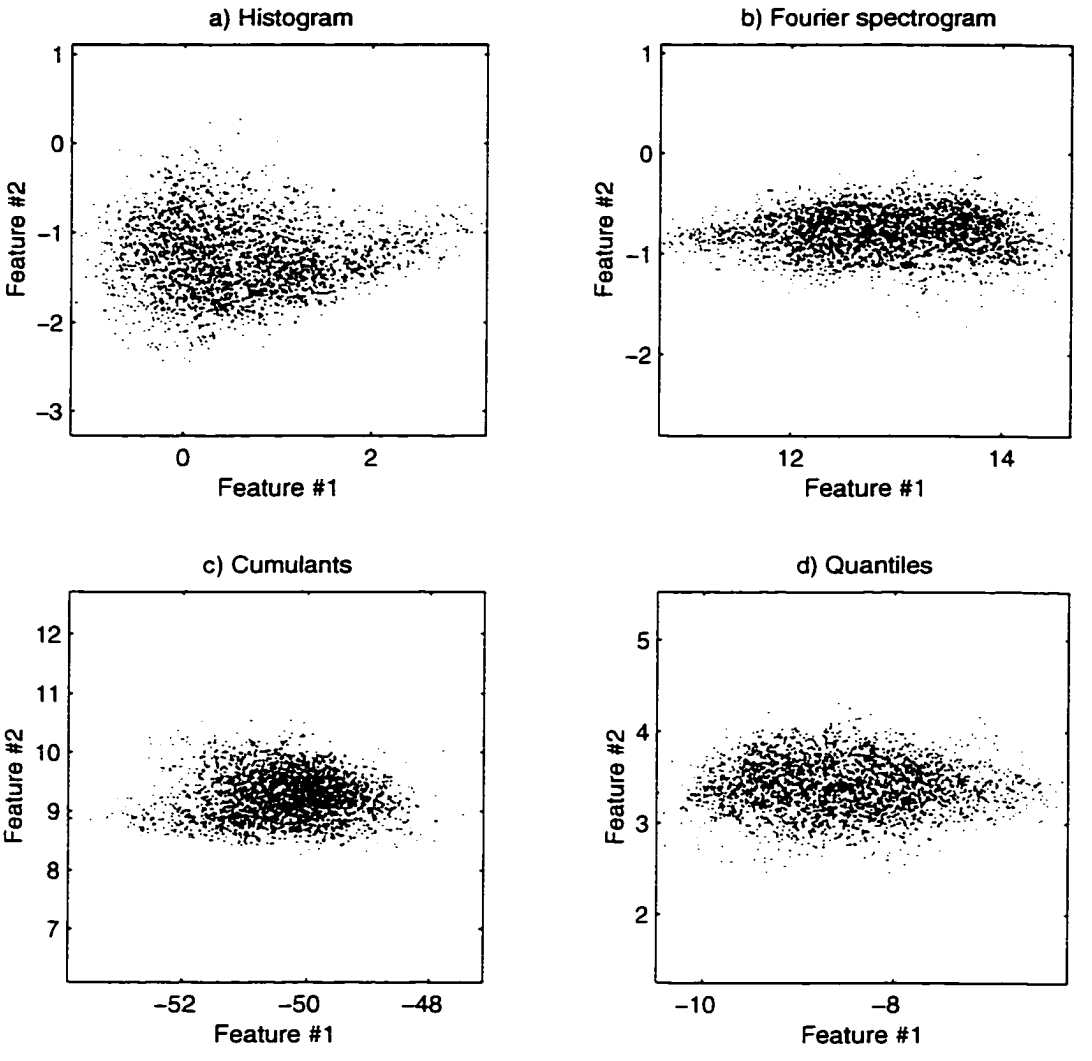


Figure 6.6: Reduced feature spaces of Caraquet averaged data. The presence of more than one class is discernible in each reduction.

The sum of the eigenvalues of the full aggregate feature set is unity by design, whereas the sum of the eigenvalues of the order-*MN* aggregation is 0.8430. The difference represents the average amount of energy per feature set discarded by the order-*MN* reduction. However, since the final eigenvalues are almost the same — meaning that the principal axes in the two feature spaces are similarly proportioned and hence implying similar distributions — it can be argued that the discarded dimensions do not contribute significantly to the final reduction.

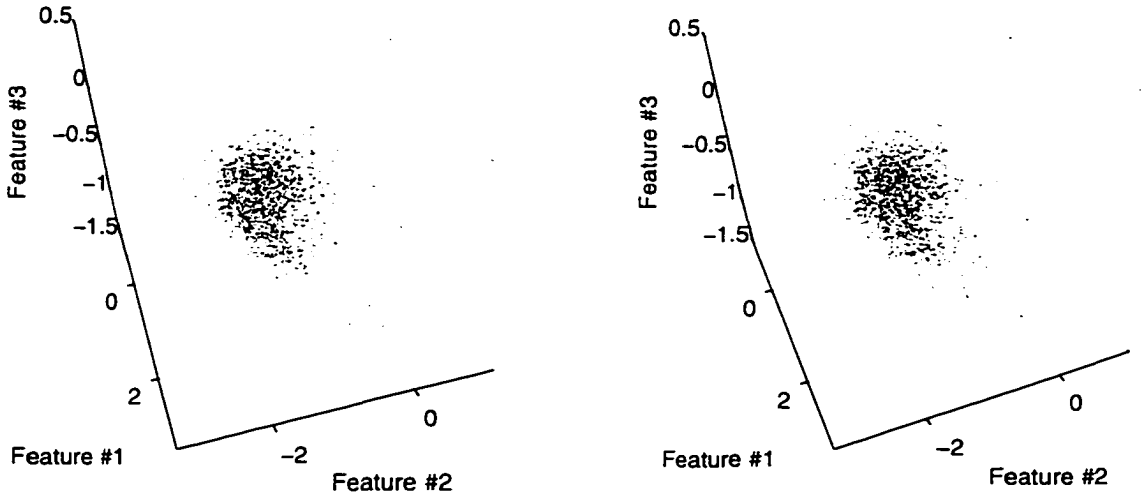


Figure 6.7: Cross-fusible stereo pair of Caraquet reduced histogram feature space showing two separable classes.

Feature Set	λ_1	λ_2	λ_3	λ_4	λ_5	% of whole
Full aggregation	0.5667	0.1475	0.0748	0.0322	0.0176	83.9
Order-MN aggregation	0.5660	0.1409	0.0712	0.0301	0.0131	97.4

Table 6.1: Sorted eigenvalues of Caraquet data

6.2.1.3 Clustering

Four reduced data sets were clustered. The reduction of the aggregate data is compared against the reduction of the amplitude histogram (which has the most apparent *prima facie* separability for a single feature set), for both two and three dimensions. The values of two and three dimensions are chosen for two reasons. First, they represent the lion's share of the information: the most significant order-*MN* reduction eigenvalues contribute 67.1%, 16.7%, 8.5%, 3.6%, and 1.6% of the total variance. Thus, contributions from dimensions higher than three are reduced more than one order of magnitude from the largest eigenvalue. Secondly, two and three dimensions are reasonably easily visualized in feature space. Obviously, two dimensions are more attractive due to potential cost savings in deal-

ing with fewer dimensions; however, with this data set the third dimension represents an increment in the retained variance of 10%, the loss of which may be detrimental to the separation of the classes.

Fig 6.8 shows a quick comparison of how the identified clusters mapped against positional space, with the assumption that only two classes are present. As can be seen, three of the four maps are very similar, with the largest difference resulting from the use of only two dimensions of the order-*MN* feature set. Therefore, it seems appropriate to use at least three dimensions. The question of which map (i.e., feature set) represents the best results is difficult to ascertain since the correct answer is not actually known. Nonetheless, we can try to quantify the quality of the results. First, we can do so both in feature space, by calculating the mean certainty of the classifications, and in the spatial domain by calculating the mean run-length of the classifications. The first measure scores highly for classes which have their boundaries (i.e., regions of high uncertainty) in regions of feature space with low densities. The second penalizes clusters which result in gradual fading from one class to another.

Both measurements are affected by the *a priori* class probabilities. For example, Fig 6.8a has a large difference in the class populations, compared with Figs 6.8b-d. This results in considerably fewer points surrounding the decision boundary in feature space[1]. And since certainty is minimized at the decision boundary, a reduction in the number of boundary points increases the mean certainty. As well, a large difference between class populations increases the probability that a large number of consecutive points will be

1. This correspondence between geographical adjacency and feature space proximity is discussed later in this section, and illustrated in Fig 6.12.

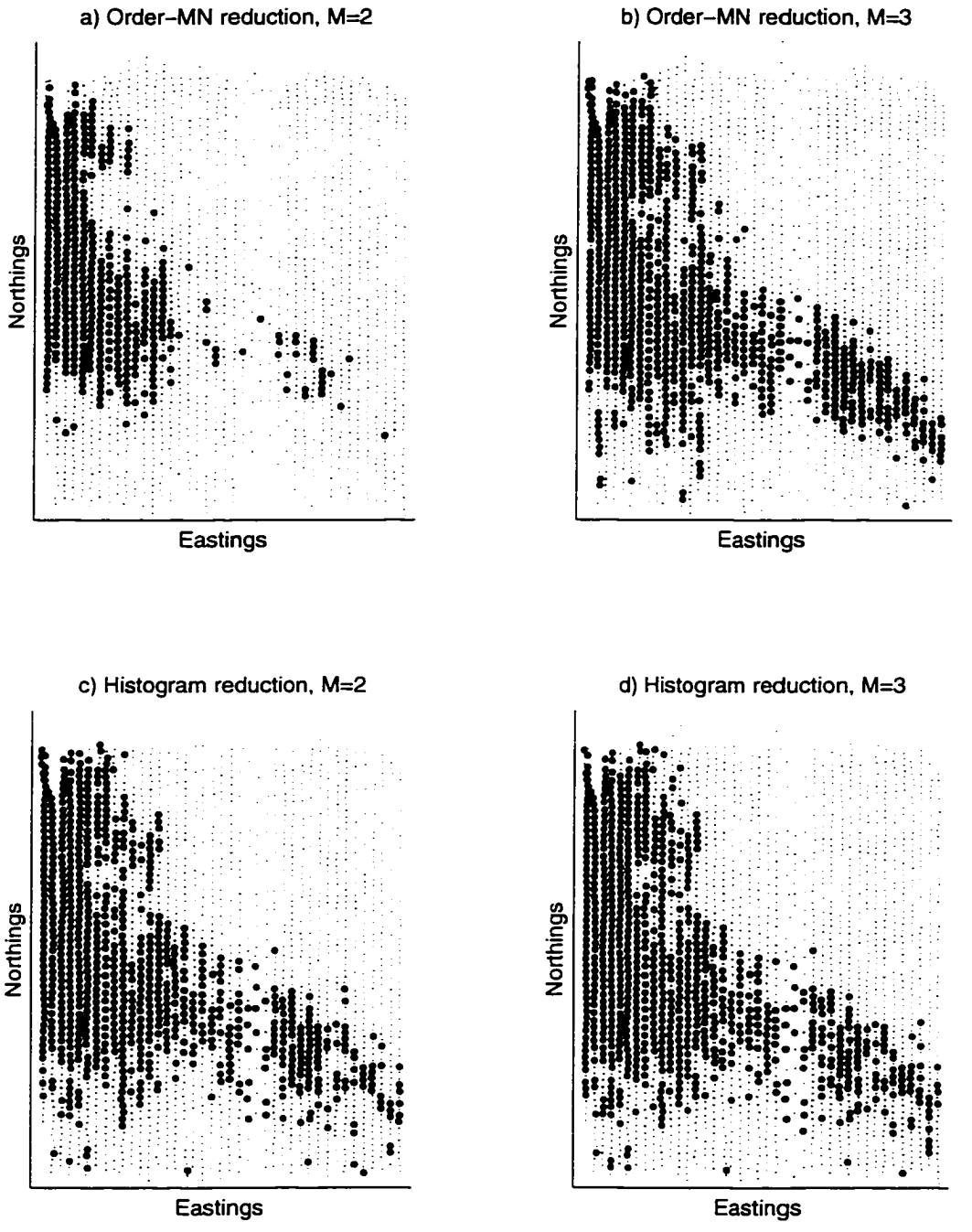


Figure 6.8: Caraquet clustering comparisons. In each map, two unspecified classes are shown, one by a small dot, and one by a large dot. The sparsely-sampled vertical strip to the right of centre represents a temporary change in the ping rate.

classified as a single class[1]. Thus, large differences in class populations can lead to misleadingly high estimates of mean certainty and mean run-length. Therefore it is useful

when comparing maps with different *a priori* class probabilities to compensate by scaling by the ratio of the probabilities. Table 6.3 below shows the number of vectors assigned to each class for each of the clustering results displayed in Fig 6.8.

Class	Fig 6.8a	Fig 6.8b	Fig 6.8c	Fig 6.8d
ω_1	2169	4178	4105	4287
ω_2	10348	8339	8412	8230
Ratio	0.2096	.5010	0.4880	0.5209

Table 6.2: Caraquet class populations

The mean certainties and mean run lengths are tabulated below, for both raw and the normalized measures. Both measures would suggest that the three-dimensional order-*MN* aggregation shown in Fig 6.8b generates slightly better clusters than the other reduced feature sets.

Method	Fig 6.8a	Fig 6.8b	Fig 6.8c	Fig 6.8d
raw	0.9016	0.8486	0.8492	0.8469
normalized	0.1890	**0.4252**	0.4144	0.4411

Table 6.3: Caraquet mean certainties

Method	Fig 6.8a	Fig 6.8b	Fig 6.8c	Fig 6.8d
raw	38.5	18.1	13.9	13.6
normalized	8.1	**9.1**	6.8	7.1

Table 6.4: Caraquet mean run-lengths

The only other question that remains is how many clusters exist. The basic premise has been that this data set contains two classes based on the nature the survey area (a dump

1. Consider the limit case in which $N_1 = 1$ and $N_2 = N\text{-}1$. Here, the mean run-length has minimum possible value of $N/3$, despite the fact that this degenerate classification is probably incorrect.

site), but since the correct results are not known, this premise may be false.

One stopping criteria for the clustering algorithm that has been empirically found to be effective involves minimizing the sum of the weighted χ^2 statistics of normalcy hypothesis tests [51]. That is, the clustering starts out with the assumption of one Gaussian class whose normalcy can be estimated with a χ^2 statistic. When the single class is split and stabilized, the two resulting classes each have their own χ^2 statistic. These are weighted by the number of vectors in each class (i.e., by the estimate of the *a priori* class probabilities) and added. So if the χ^2 statistic of the individual clusters is less than that of the aggregate cluster, then the weighted sum will decrease. Thus, the sum of the weighted χ^2 statistics will attain a minimum value when the natural number of clusters has been identified. Further splitting will result in non-Gaussian distributions, which cause higher χ^2 statistics. The results are tabulated for the first four iterations of the clustering algorithm:

	Iteration #1		Iteration #2		Iteration #3		Iteration #4	
Class	$p(\omega_i)$	χ_i^2	$p(\omega_i)$	χ_i^2	$p(\omega_i)$	χ_i^2	$p(\omega_i)$	χ_i^2
ω_1	1.0	5.61	0.295	2.47	0.072	0.73	0.036	1.07
ω_2			0.705	1.65	0.638	1.15	0.562	1.76
ω_3					0.300	1.37	0.084	1.15
ω_4							0.318	4.31
$\sum p(\omega_i)\chi_i^2$	5.61		1.89		1.18		2.49	

Table 6.5: Caraquet weighted-χ^2 sums for first four iterations of clustering algorithm

Thus Table 6.5 would suggest that there are in fact three clusters present. When these results are plotted against position, Fig 6.9 arises. The contours are generated from a

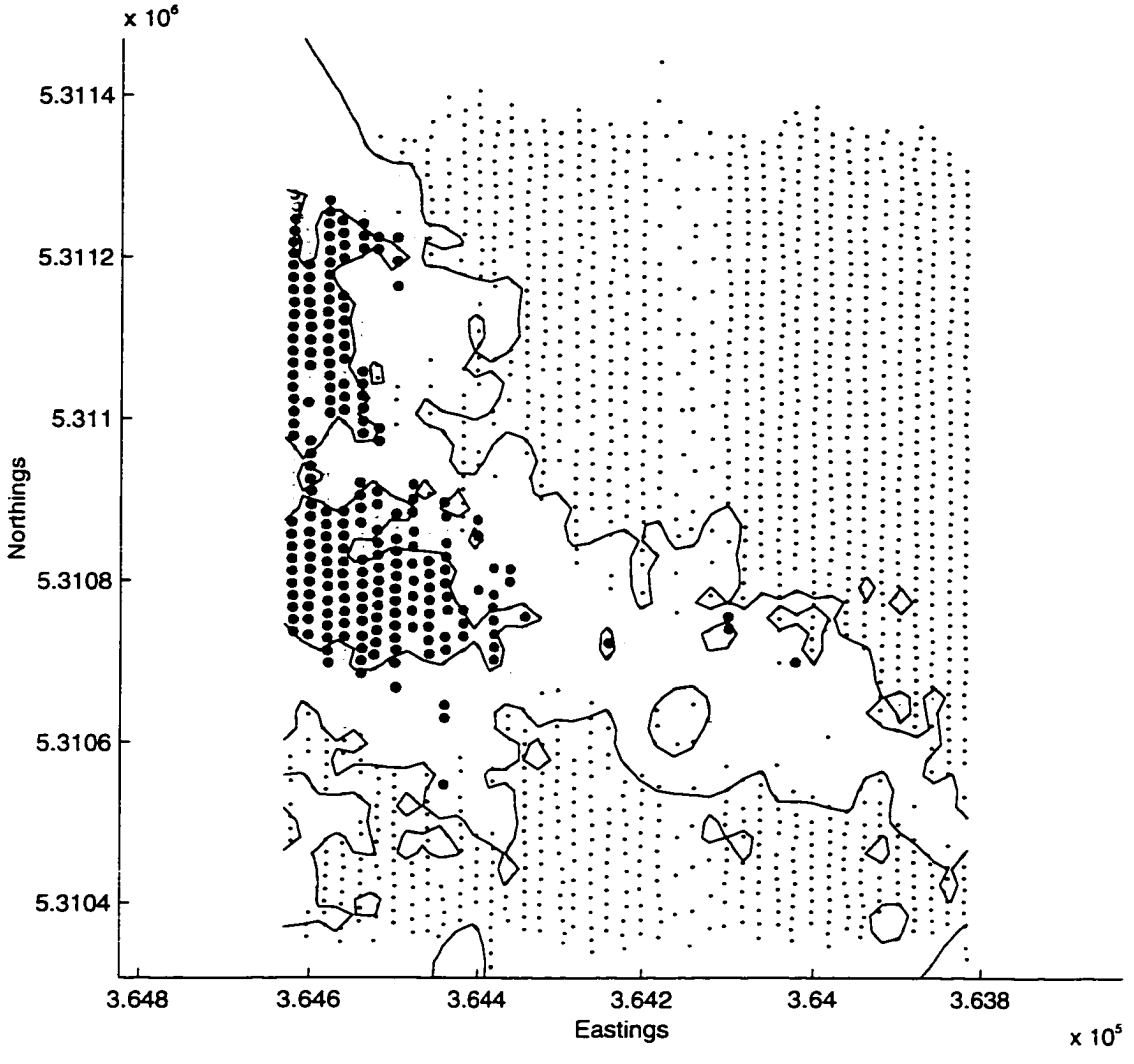


Figure 6.9: Caraquet track plots with interpolated contours. Three unspecified classes are shown, by small, medium, and large dots. The lines are approximations of the inter-class borders, derived from coarse gridding and interpolation within the survey area.

numeric 2-D interpolation[1] of the classification data. Fig 6.10 shows the same interpolation draped over interpolated bathymetry. Note the view direction is from the upper left corner of the track plots in Fig 6.8. Feature space itself is shown in Fig 6.11 as a scatter

1. Normally, it is inappropriate to perform a numeric interpolation on classifications, since a numeric combination of two or more bottom types is a meaningless concept, despite their being represented by integer indices. However, since there are only two boundaries in this case, we can take liberties, provided that the class indices are ordered correctly so as to prevent double contours.

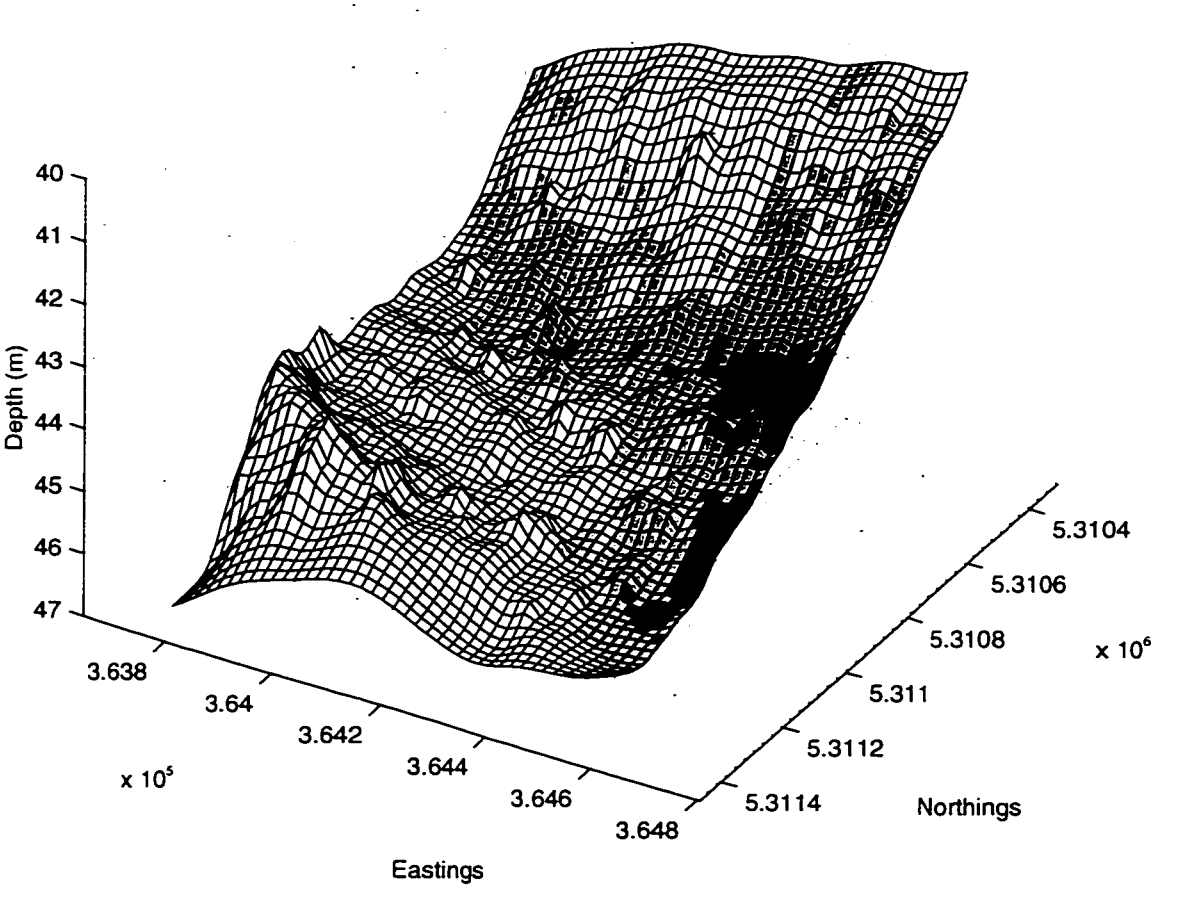


Figure 6.10: Caraquet classification draped over interpolated bathymetry. The view angle is rotated approximately 180° from the previous figures. Coarse gridding and interpolation are used to fill in the survey area.

plot, and as constant probability ellipsoids (2 standard deviations) with their 2-D projections.

As a final note, the effects of spatial averaging (see Section 4.3.5) can be illustrated by examining the position in feature space of temporally adjacent samples as the survey vessel crosses class boundaries in the spatial domain. The theory is that geographically adjacent locations with similar physical characteristics ought to generate similar acoustic

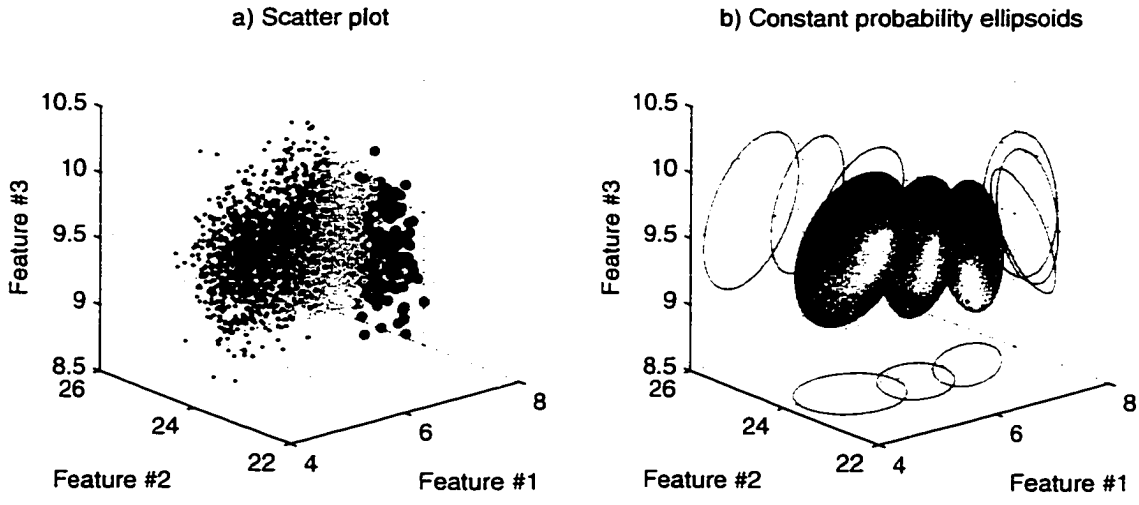


Figure 6.11: Three class partitioning of Caraquet feature space. The individual points are shown as a scatter plot, as well as the identified classes 2-sigma constant probability ellipsoids (and 2-D projected ellipses).

returns, which in turn generate similar feature vectors, which correspond to proximal locations in feature space. This is illustrated in Fig 6.12, in which 100 temporally adjacent returns are examined. The returns are taken from the sixth track from the left of Fig 6.9. The traces commence near the bottom and progress northwards through three distinct bottom types. The locations in feature space are plotted both for the raw and the spatially averaged data. The averaged data shows a strong proximity despite translational jumps corresponding to classification changes. However, the raw data exhibits no such tendency. In fact, one way to characterize the data is to measure the mean Euclidean distance in feature space between temporally adjacent points. Although the two data sets span the same amount of feature space, the mean separation is over four times larger for the raw data (0.8816 versus 0.2034).

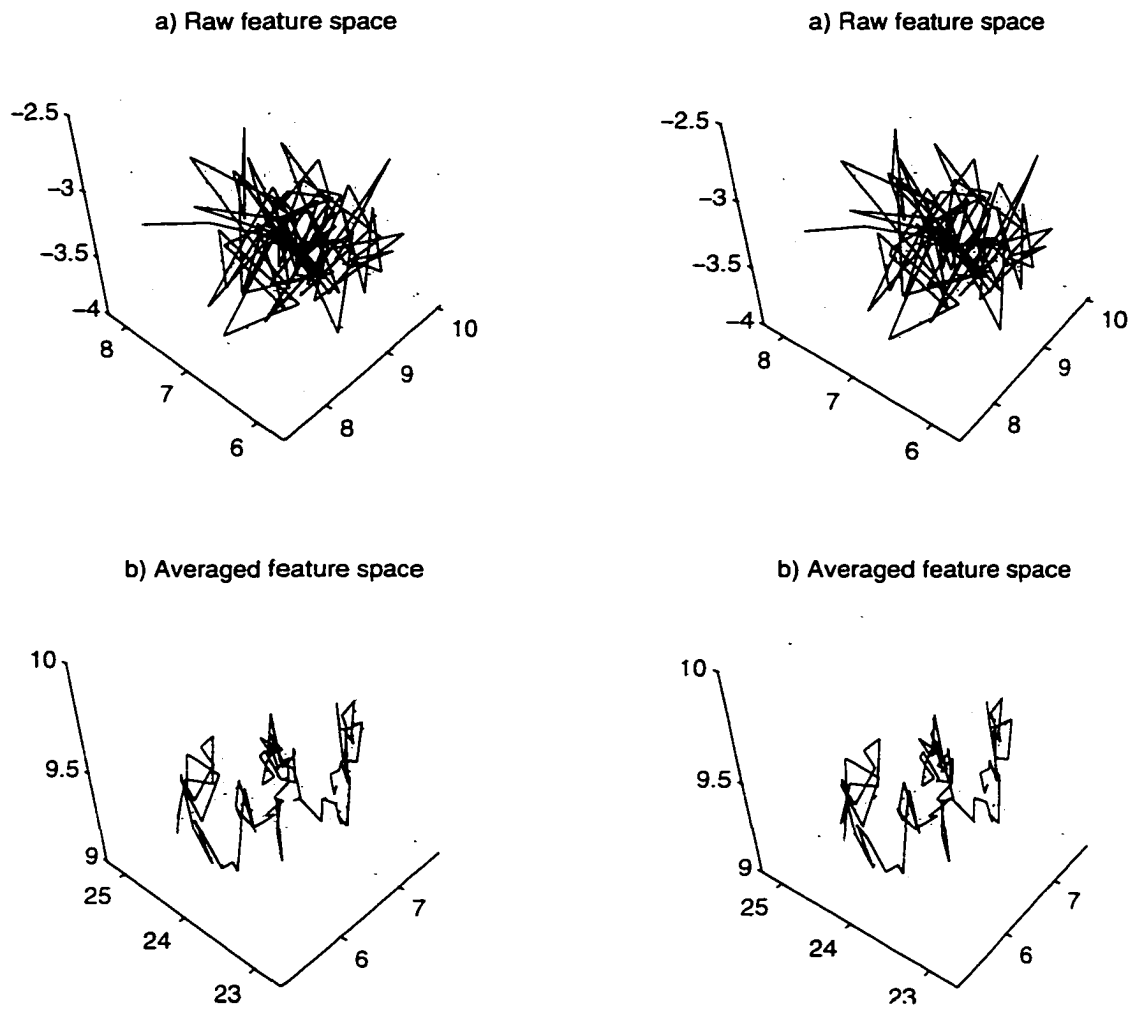


Figure 6.12: Cross-fusible stereo pair of adjacency in Caraquet feature space. Figure a shows the random distribution in feature space of geographically adjacent points for the raw data set. Figure b illustrates the tight coupling revealed by spatial averaging.

6.2.2 UNB (Saint John Harbour) data set

Dr. Larry Mayer's Ocean Mapping Group at the University of New Brunswick (UNB) has kindly provided us with a data set collected as part of the HYGRO'93 [1] project. The author was present as part of a Quester Tangent sponsored team for part of the data collection which occurred over several days in August 1993.

The data set covers an area of Saint John Harbour in New Brunswick. Track lines for a subset of the data available are illustrated in Fig 6.13. Records corresponding to the 33 survey lines labeled 1750100 through 1750132 were analyzed.

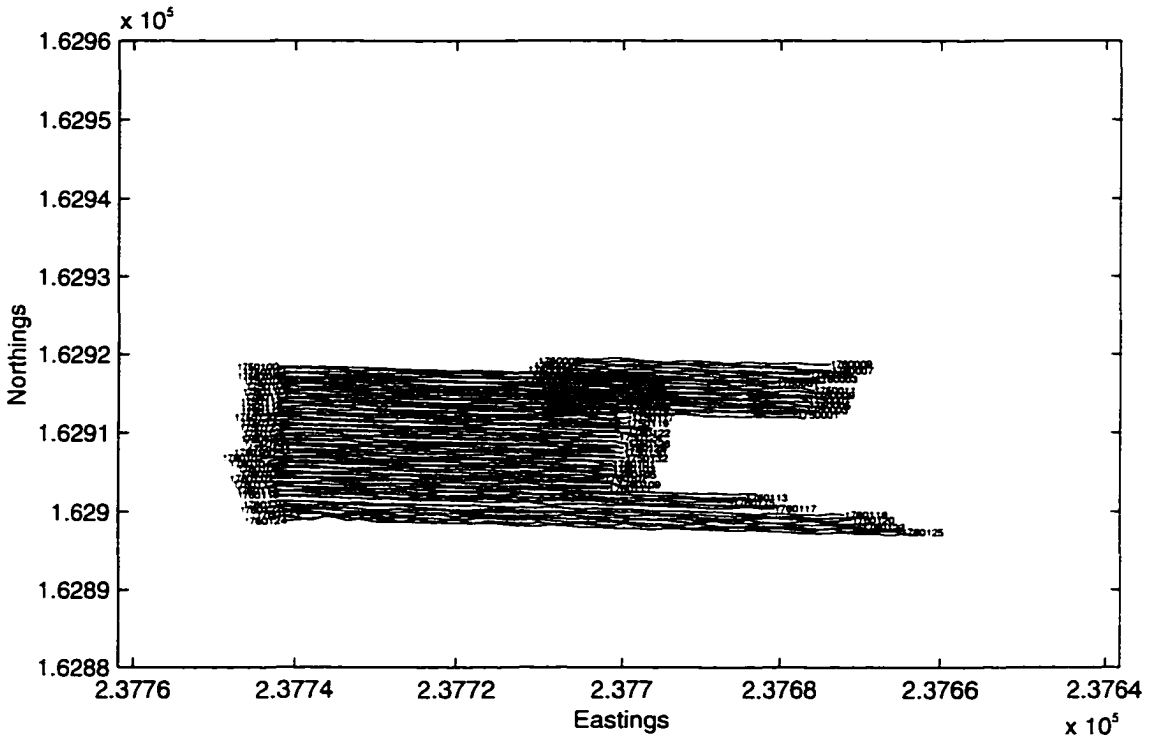


Figure 6.13: Coverage of UNB data sets. Survey lines numbered 1750100 through 1750132 were analyzed.

This data set was collected by the Ocean Mapping Group with assistance from Public Works and Government Services Canada and Quester Tangent Corp. Almost 54 thousand returns are included, each 1497 samples long. The sampling frequency is 10 kHz, and the estimated speed of sound in water is approximated at 1500 ms^{-1}. The beam width and pulse length are not known.

This data set was fairly awkward to process. First, it was stored using an early version of the QTC file format, which made extraction laborious. The position information was dumped from a CARIS system in a time base that was unrelated to that of the ISAH-S data. Manually calibrated reference tables had to be consulted to calculate the offset time for each track, and then every time stamp in each data file had to be compared to the start and end times of each track to identify whether the return was part of a survey line.

Numerous intermediary files had to be created to facilitate access of returns.

Of the initial subset of 53723 returns, 50266 were part of survey lines. Of these, only 49751 were valid records. Data logging problems caused about 1% of the records to be suspect.

Fig 6.14 below provides a sample of the data. One of the characteristics of this data set is that the individual survey lines cover a wide range of depths (5.3 to 38.5 m).

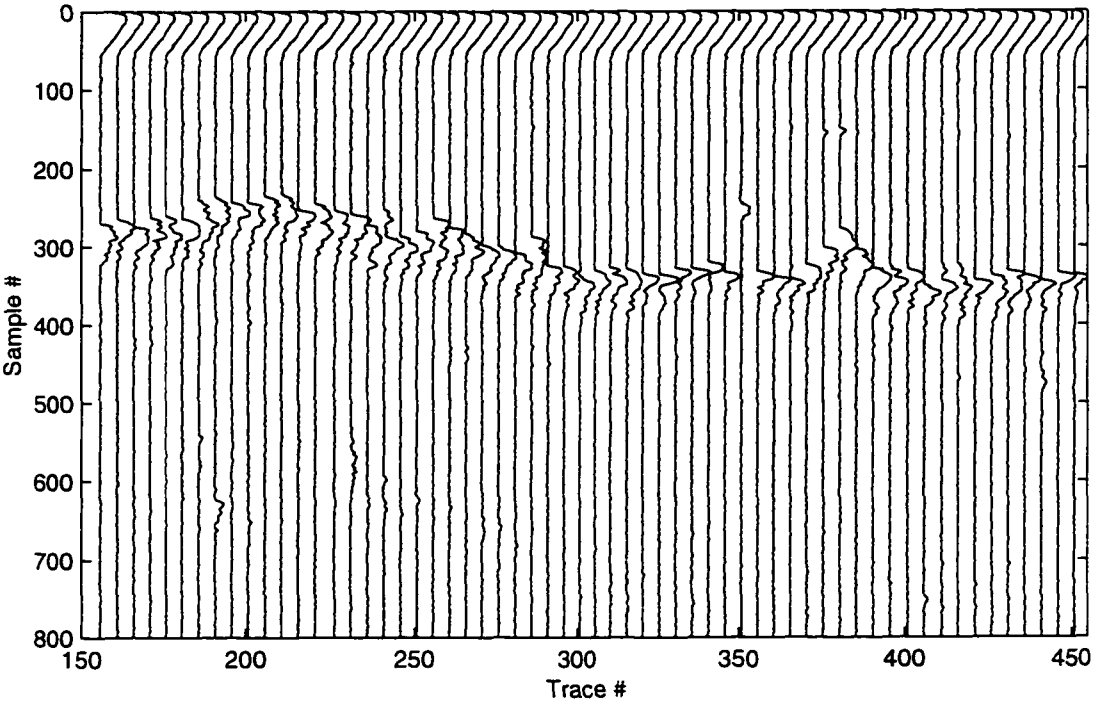


Figure 6.14: UNB raw data. Note the presence of invalid traces due to data logging problems.

6.2.2.1 Picking

As can be seen in Fig 6.14, some of the traces (e.g., number 350) are invalid. Some are blank (i.e., no apparent bottom response) while others are legitimate acoustic returns which are incompatible with the surrounding bottom[1]. Therefore, autoregressive estimates

of the mean are influenced substantially by invalid returns unless the forgetting factor is permitted to adapt, as suggested in Section 4.3.5.3. However, for this data set the SNR is quite high — enabling very accurate picks even without spatial averaging (as discussed in Section 4.2).

Even so, the pick results contain wild points which correspond to invalid or out-of-context traces. These traces were identified by median filtering, and removed from the data set (as per Section 4.2.2.1). Since the bathymetry usually changes slowly, the median filtering was done on a high-pass filtered copy of the pick values. The effect of this is shown in Fig 6.15 in which the raw and median filtered picks correspond to the end of survey line 1750100, all of line 1750101, and the first half of line 1750102.

The resulting bathymetry map is shown along with classification results in Fig 6.21.

The data set was then normalized to a depth of 20 m and flattened such that the bottom occurred 8 samples into a 128-sample window. The flattened data set for survey line 1750123 is shown in Fig 6.16.

6.2.2.2 Feature extraction and reduction

As with the Caraquet data set, four simple feature extraction algorithms discussed in Section 5.1 were applied to both the raw and spatially averaged data. The resulting feature vector sets were each reduced to three dimensions (two, in the case of cumulants). These reduced data sets were examined individually, as was the order-*MN* reduction of the aggre-

---

1. E.g., the response and secondary echo indicate a depth of 15 m in the midst of a survey line consistently at 11 m.

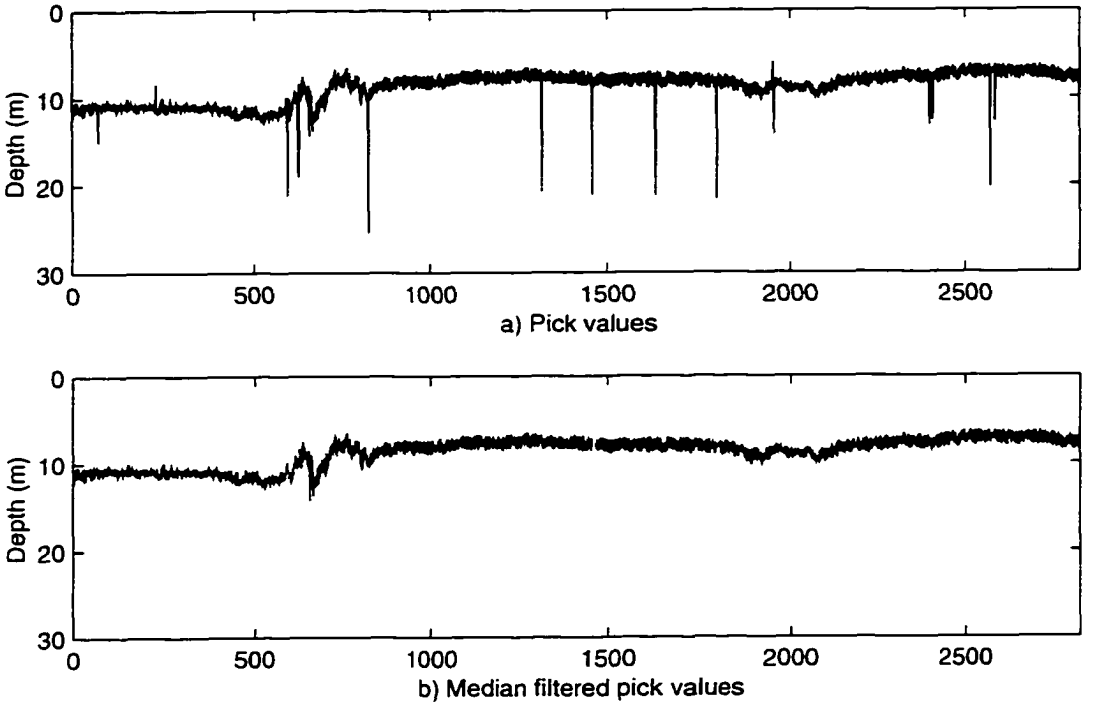


Figure 6.15: Picking results of UNB data set. Clearly visible wild points were identified and eliminated from the data set by median filtering.

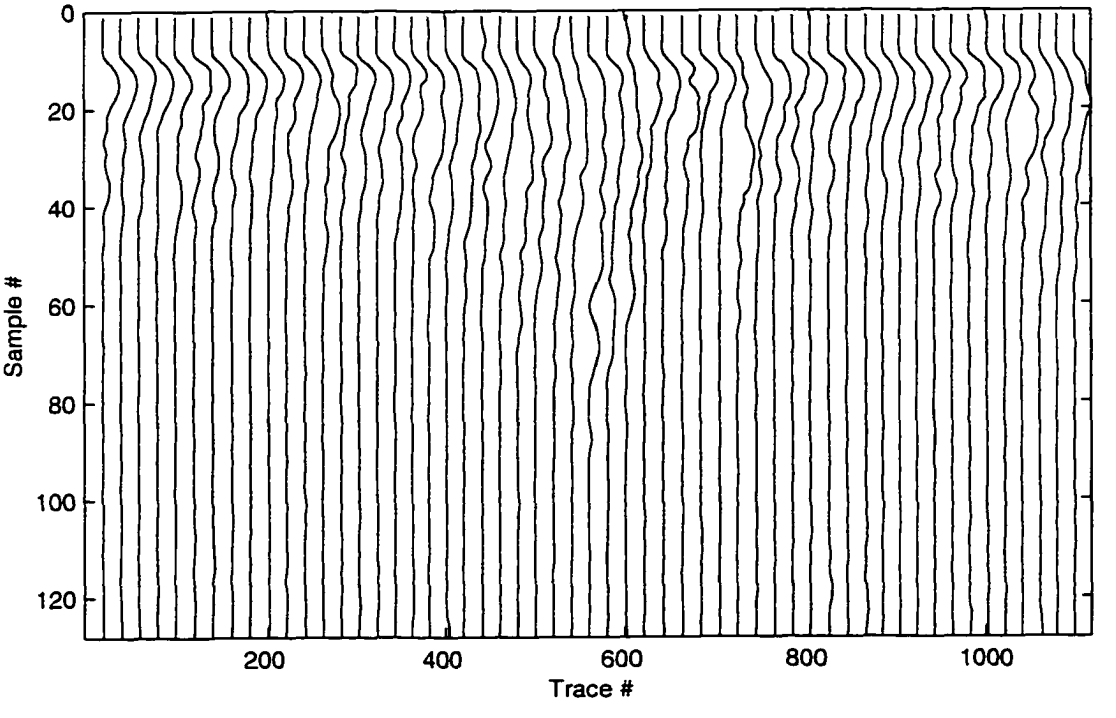


Figure 6.16: Flattened raw UNB data. The sample numbers have been renumbered to correspond to the offset into the buffer of flattened records.

gation of the individual reduced data sets. The individual feature for two dimensions sets are shown in Figs 6.17 and 6.18. Note that none of the feature sets of Fig 6.17 show significant differentiation in the data, whereas all of those of Fig 6.18 do.

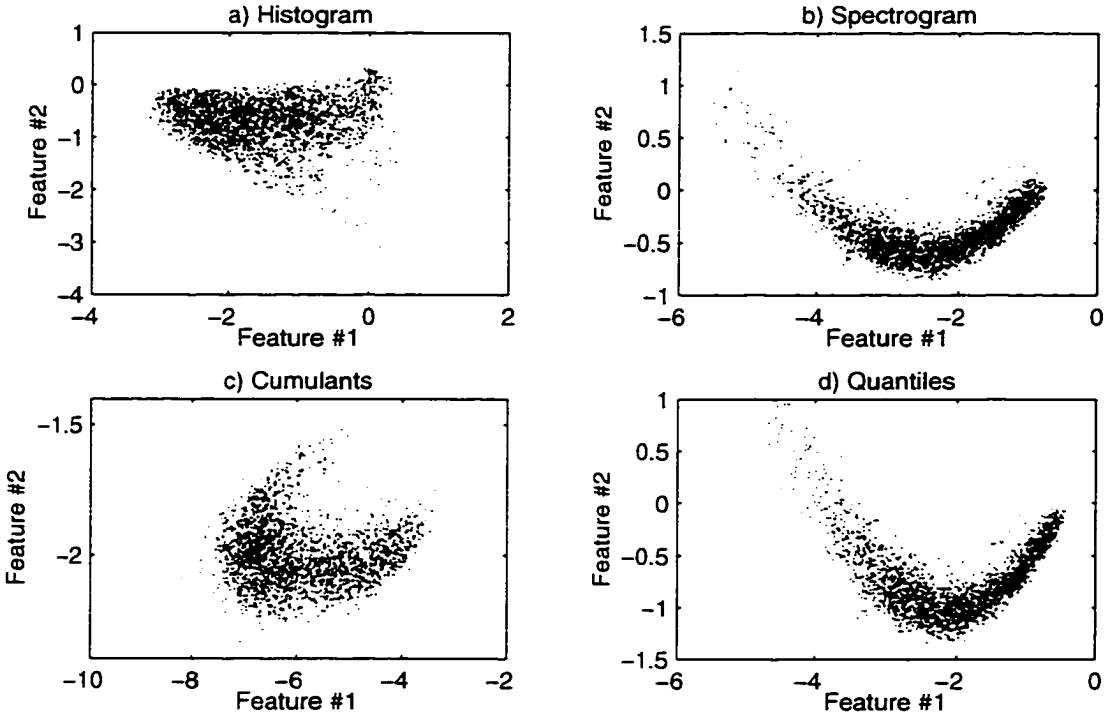


Figure 6.17: Feature sets arising from raw UNB data.The axes in each figure are the first two principal components of the feature sets, i.e., weighted sums of the individual features. While clearly non-Gaussian, no classes appear separable.

The sorted eigenvalues of the order-*MN* aggregations are tabulated in Table 6.6.

Reduced feature space was defined as having two dimensions, since 91.0% and 95% of the variance are contained in the first two axes. The reduced order-*MN* aggregations are shown in Fig 6.19. The obvious thing to note is that while maintaining the same basic sausage shape, the spatially averaged data set exhibits some clumping of data. The extremely high densities arise from traces that are extremely highly correlated from one

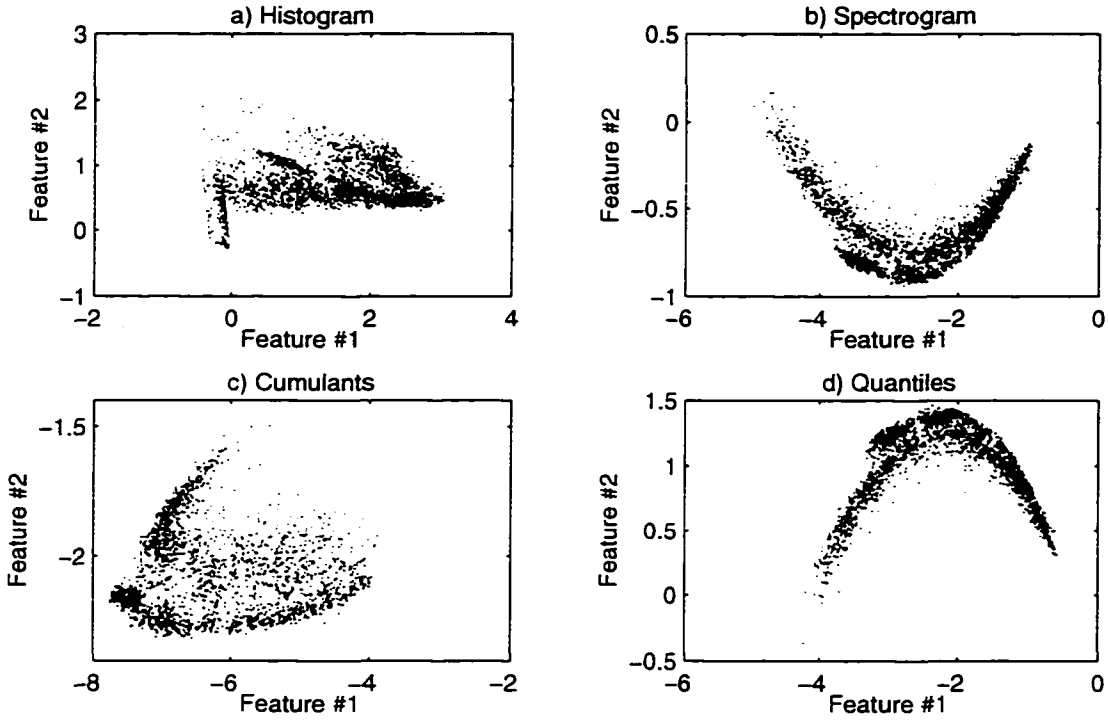


Figure 6.18: Feature sets arising from averaged UNB data. Spatial averaging has provided concentrations upon which the clustering algorithm can consistently base clusters.

Feature Set	λ_1	λ_2	λ_3	λ_4	λ_5	% of whole
Raw	0.8055	0.1268	0.0348	0.0073	0.0041	99.7
Averaged	0.7270	0.1376	0.0583	0.0116	0.0096	99.3

Table 6.6: Sorted eigenvalues of UNB data

ping to the next. In some regions, the normalized correlation coefficient has a mean value of 0.95, resulting in large number of past pings being included in the estimate of the mean.

6.2.2.3 Clustering

Fig 6.19a represents a difficult interpretation for both humans and clustering algorithms since there is no visible clumping of data points[1]. Any division is somewhat arbitrary — one can chop the sausage-shaped distribution into three, four, five, or more roughly-equal chunks, with obviously significant ramifications on the boundaries in the spatial domain.

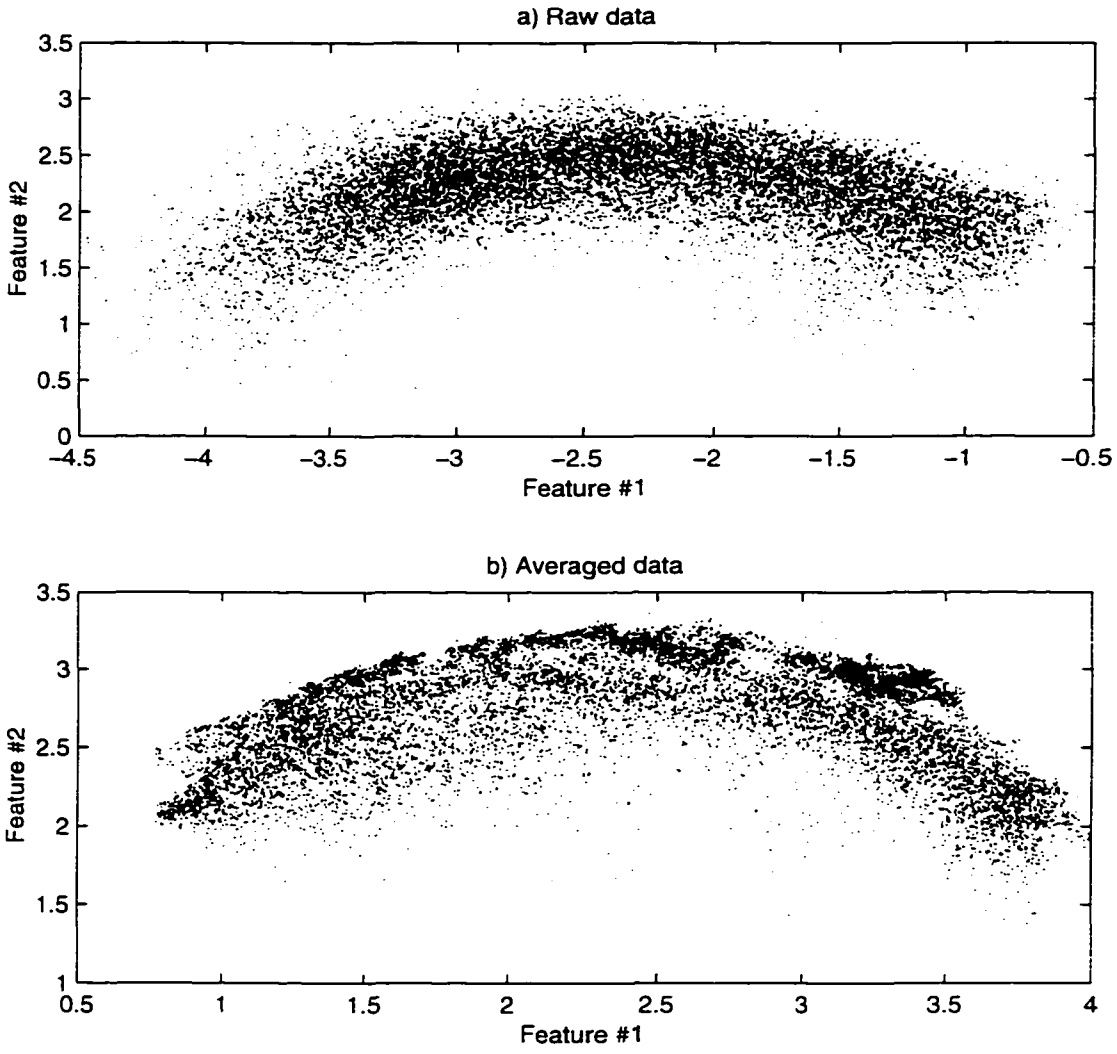


Figure 6.19: Reduction of UNB feature sets. Figure a illustrates the arbitrary nature of dividing a continuous distribution of points into a finite number of clusters. Figure b illustrates the effects of spatial averaging.

The clumps in Fig 6.19b, while not indicative of the statistics of the entire cluster, do provide easy targets for the clustering algorithm. One should be aware that the clumps will decrease the normalcy of the clusters. This should be taken into account in assigning

1. Non-clustered distributions of feature vectors can be caused in four ways: first, the survey site can in fact cover a continuum of bottom types. Second, the bottom types can be dominated by feature sets which provide poor separability, or the feature extraction algorithms themselves are ineffective. Third, the features can be affected by artefacts, such as depth, which may have a continuous distribution. Finally, the feature vectors can be noisy.

the stopping criteria for the clustering algorithm, which assumes Gaussian clusters.

Analysis of the histogram feature set by the weighted-sum of χ^2 statistics indicates six classes. The clustering is performed and the membership is used to partition feature space as shown in Fig 6.20a. The resulting track plot is shown in Fig 6.20b for grossly undersampled data (only every 25^{th} classification is shown).

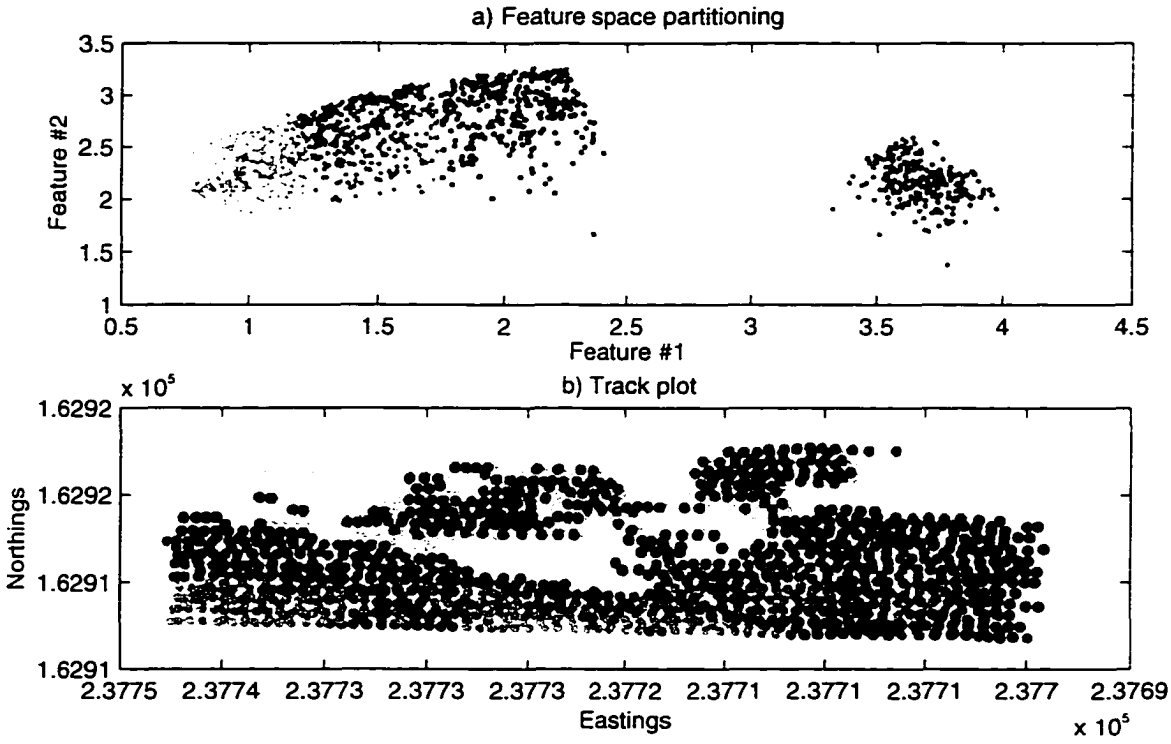


Figure 6.20: Results of classification of UNB data. Figure a illustrates the classifications and decision boundaries in feature space. Figure b shows the corresponding classifications in geographic space.

Although the results are not shown, clustering was also performed on the raw data. The quantifications of the classification quality are given below.

Finally, Fig 6.21 shows the classifications draped over bathymetry. Note that there is an obvious relation between classification and the bathymetric structure. In terms of

Method	Certainties	Run lengths
Raw	0.8048	4.21
Averaged	0.8570	46.1

Table 6.7: UNB classification quantifications

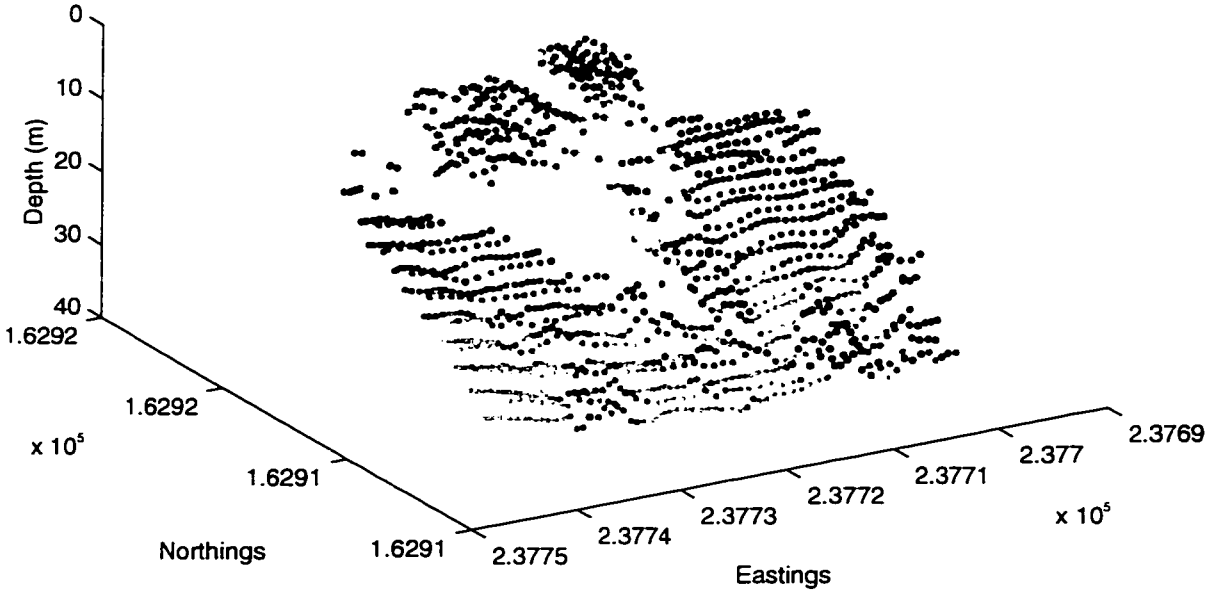


Figure 6.21: UNB classifications draped over bathymetry, illustrating both the correlations between depth and classification, as well as the exceptions.

blind clustering this is both good and bad — the concern is always that the classification is being driven by artefacts (particularly, depth). On the other hand, if the bathymetric structure is representative of differences in bottom type, then one expects to see a correlation.

6.2.3 EDRD data set

Dr. Roland Poeckert of the Esquimalt Defence Research Detachment of the Defence Research Establishment Atlantic (formerly DREP) was kind enough to arrange permission to use some data he had collected. Due to the sensitive nature of the data, no information can be provided regarding location. A more detailed analysis of the data is available [14].

This data set was gathered by a transducer mounted on a towed fish which maintained a nominal altitude (from the ocean bottom) of 15 m. The sampling rate was 10 kHz, the speed of sound was assumed to be 1500 ms^{-1}, the beam width was 8°, the carrier frequency was 208 kHz, and the pulse width was 100 μs. The data set had already been spatially averaged prior to being made available to use. The averaging was done by calculating the mean of adjacent blocks of five samples. The ping rate was 6 Hz.

Several Gigabytes of data were collected, but the subset analyzed here consists of 6 discontiguous segments of 250 returns which were identified by Dr. Poeckert as being representative of different bottom types. These assessments were made by reviewing ROV video tape recordings. Fig 6.22 shows the log amplitude of the EDRD data set.

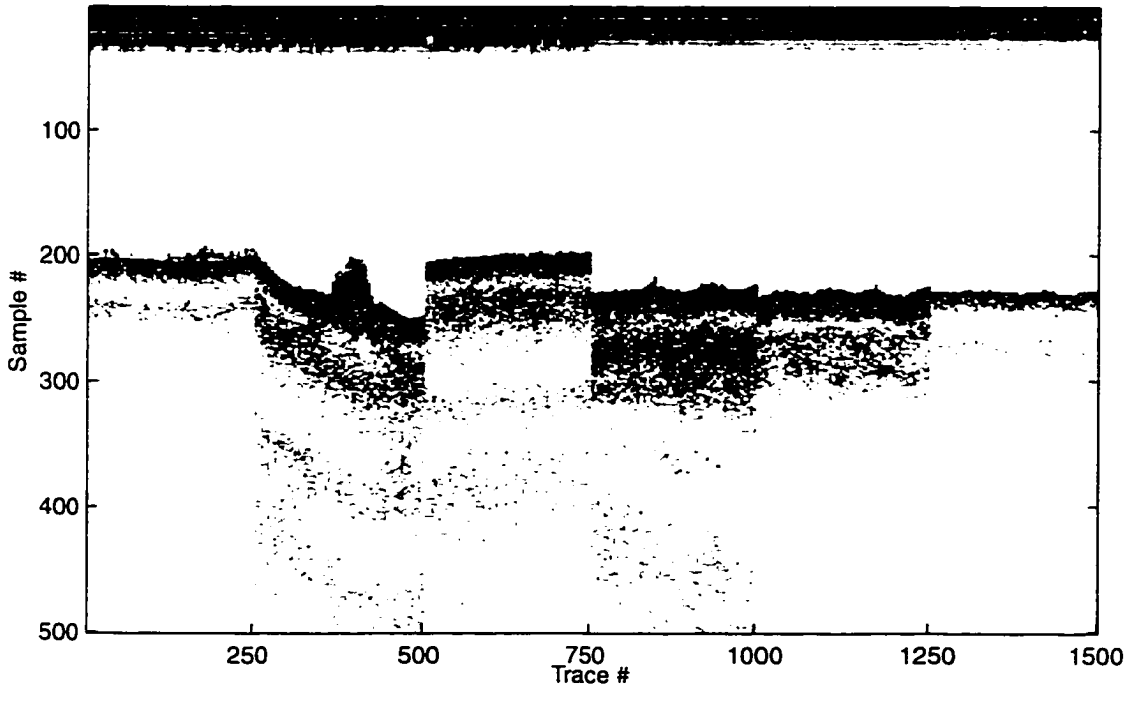


Figure 6.22: EDRD data set, using a log-amplitude gray-scale plot. The six distinct sections correspond to six archetypes identified from ROV video.

This data set has been featured elsewhere in this thesis, most notably in Section 5.2.1.2 (Algorithmic scaling), Section 5.3 (Classification), and Section 5.3.3 (Certainty). For this reason, this analysis of the data set in this section will be brief.

6.2.3.1 Picking

The data set was straightforward to pick. There were the occasional returns which contained indications of biology, but nothing very unusual.

6.2.3.2 Feature extraction and reduction

The same procedure was followed for the EDRD as was followed for the Caraquet and UNB data sets. The reduced feature spaces resulting from the individual algorithms is shown below in Fig 6.23.

Note the excellent clustering potential in all feature sets except cumulants.

The sorted eigenvalues of the order-*MN* aggregations are tabulated in Table 6.8.

Feature Set	λ_1	λ_2	λ_3	λ_4	λ_5	% of whole
EDRD	0.7758	0.0748	0.0525	0.0159	0.0082	99.0

Table 6.8: Sorted eigenvalues of EDRD data

6.2.3.3 Clustering

Given that there is not much difference between the variance contained in the second and third principal axes, it is of interest to consider clustering in three dimensions. Fig 6.24 below shows the 3-sigma ellipsoids and their 2-D projections.

Although there is no positional information for this data set, classifications versus bathymetry are given in Fig 5.6. These classifications agree with the original selection of

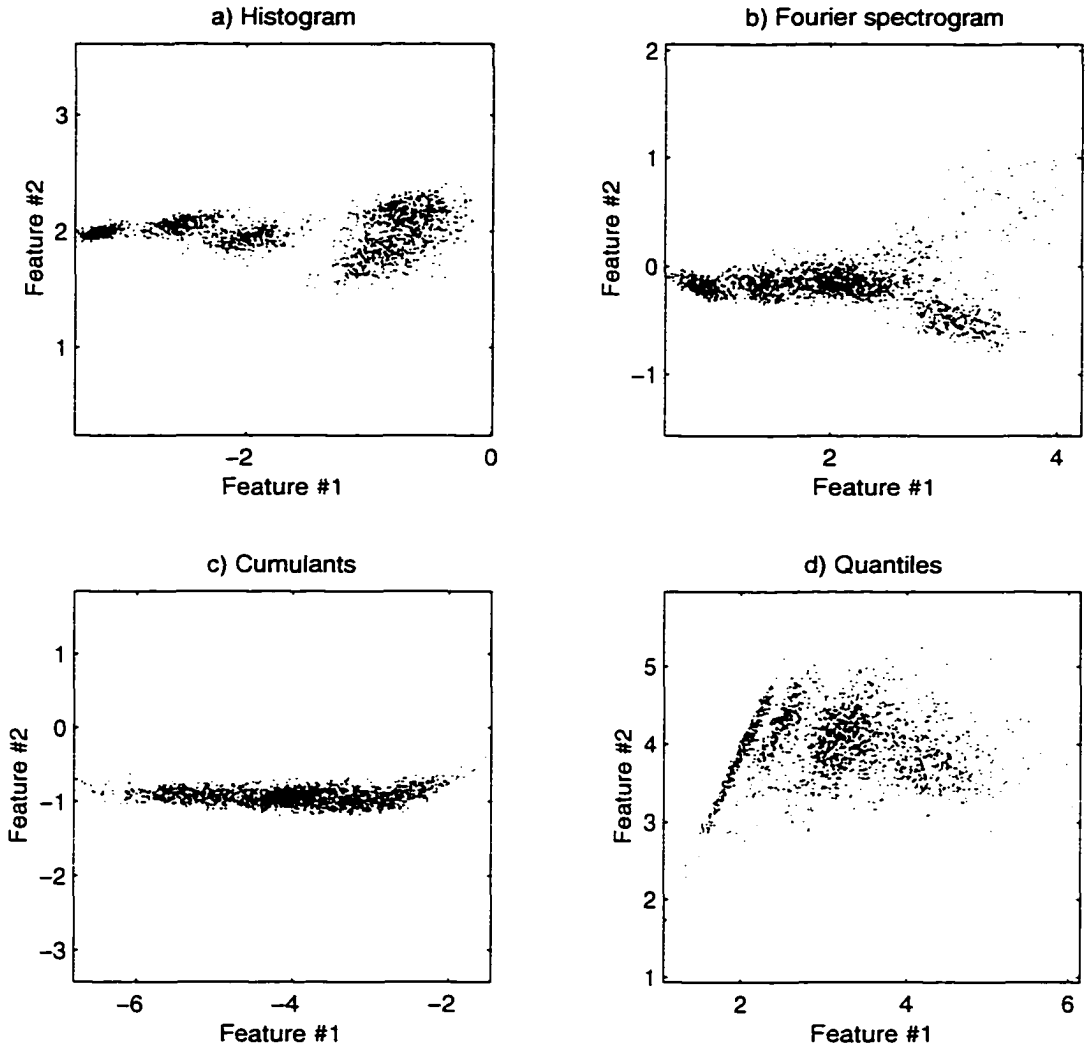


Figure 6.23: Reduced feature spaces of EDRD data. Note the excellent separability in all feature spaces except that of cumulants.

the data — i.e., Dr. Poeckert had selected six bottom types he wished to identify and then selected data from six regions which were found (using an ROV and either video or still photos) to be typical of each of the bottom types.

Fig 5.5 shows the clustering in feature space. The excellent separation between clusters is a function of selecting only "typical" representatives of each class. For example, if 250 vectors were chosen for each of the six classes represented in the UNB data set,

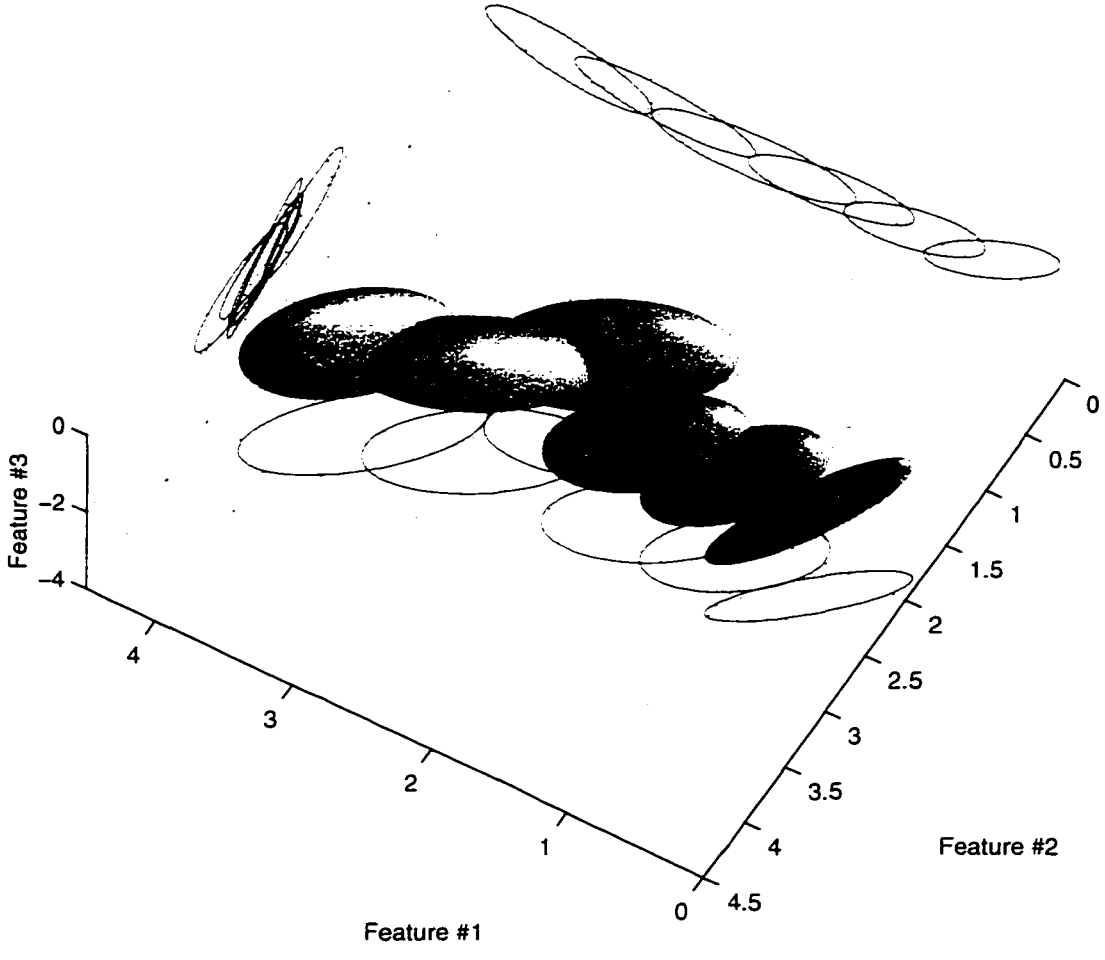


Figure 6.24: Feature space ellipsoids of EDRD data. The three-sigma constant probability ellipsoids are also projected onto 2-D. Separability in 2-D is greatest using the first two principal axes.

with the proviso that the vectors were taken only from the centre of consistent regions (see Fig 6.20), then the feature space associated with these vectors would show very distinct clusters. It is a question of selecting data to support a hypothesis (which is an unverified hypothesis in the case of the UNB data).

The certainty of the data set is mapped out both in feature space and versus bathymetry — see Figs 5.7 and 5.8. Note that in feature space, the excellent separation of the clusters results in the regions of low certainty corresponding to inter-class boundaries

occurring in areas of relatively low density.

6.3 Summary

The research associated with this thesis has had two implementations thus far: the first is the packaging of the development tools into the Pattern Recognition Toolbox for MATLAB currently being marketed by Ahlea Systems Corp., and the second is the complete seabed classification system (QTC VIEW) developed by Quester Tangent Corporation as a result of the research collaboration.

The processing techniques discussed in Chapters 4 and 5 are applied to three real world data sets provided by the Department of Public Works and Government Services Canada, the Ocean Mapping Group at UNB, and the Esquimalt Defense Research Detachment (formerly DREP), and the results are discussed.

7 Summary and Recommendations

7.1 Summary of Thesis

7.1.1 Technical background

Seabed classification is interpreted as the qualitative assessment of the surficial composition of the ocean bottom. In this thesis, seabed classification is performed using acoustic echosounder returns.

The signal and noise contributions of a recorded echosounder waveform are presented. In particular, attention is paid towards the linear dilation of the bottom impulse response with increasing depth, the further distortion introduced by convolution with a non-impulse source signal, and possible loss of information arising from envelope generation.

The requirement of time-scale normalization to correct for depth effects is shown by illustrating the dependence of the return on depth as well as incident angle.

As well, a simple backscatter model is given which corrects an error in the model proposed by Tan and Mayer. Also, it is discussed that any backscatter model used to simulate source pings should contain noise contributions in both bathymetry and scattering angles.

Finally, it is shown that the convolution evident in the backscatter model may result in signal degradation in situations where the transmit pulse length is significant compared to the bottom impulse response.

7.1.2 Initial processing

A general procedure is given for analyzing the echosounder returns. When first processing a data set, the basic run-time sequence is modified by the addition of a step for principal

component analysis to determine the feature reduction matrices, and clustering to estimate the number of classes and their statistics.

The determination of the time of arrival of the reflection from the bottom is known as bottom picking. A threshold-based method is presented in which the bottom is defined as the last sample less than a threshold prior to the first sample above a second threshold. The thresholds can adapt in noise or when the return overlaps the transmit pulse saturation of the transducer.

Once the bottom return has been isolated, it is corrected for a variety of effects and distortions, the most significant of which are transducer angle, convolution with the source signal, linear dilation due to depth, and attenuation.

A novel deconvolution technique is presented which models the envelope generation function with a finite sum discrete convolution and the Hilbert transform of the source signal. A second-order Volterra kernel can be derived using a standard predictor network with constrained optimization.

Time-scale normalization is used to correct for depth effects, and is accomplished through a linear piece-wise interpolation of the echosounder return.

Finally, spatial averaging is shown to increase the effective beam width.

7.1.3 Analysis

A bottom return is processed by extracting numerical features, reducing the dimensionality of the features, and then applying a distance metric to determine class assignment.

A variety of simple feature extraction algorithms are discussed. The major axes of the ellipsoid describing the aggregate data set are identified using principal component

analysis. This approach is a reasonable starting point for feature space reduction when the statistics of the classes represented in the data set are not known. Feature sets can be concatenated provided they are scaled appropriately. Furthermore, feature sets can first be reduced and then concatenated resulting in significant reductions in computational requirements while still permitting cross-algorithmic covariance terms.

Classification is performed by determining the Bayesian distance of a feature vector to each class mean, and then assigning the vector to the class for which the distance is minimized. Certainty, or the weighted sum of the relative conditional likelihoods of the pattern arising from each class, can be used to measure the quality of a classification.

Blind clustering is appropriate to seabed classification since high cost and/or risk make extensive ground-truthing impractical in all but a very few cases. Clustering algorithms should have a number of characteristics if they are to be practical. The *K*-means algorithm is fundamentally flawed, but is a good starting point for modification to the *K*-stats kernel, which is very robust when successively applied to a growing number of clusters.

7.1.4 Results

The research associated with this thesis has had two implementations thus far: the first is the packaging of the development tools into the Pattern Recognition Toolbox for MATLAB currently being marketed by Ahlea Systems Corp., and the second is the complete seabed classification system (QTC VIEW) developed by Quester Tangent Corporation as a result of the research collaboration.

The processing techniques discussed in Chapters 4 and 5 are applied to three real

world data sets provided by the Department of Public Works and Government Services Canada, the Ocean Mapping Group at UNB, and the Esquimalt Defense Research Detachment (formerly DREP), and the results are discussed.

7.2 Recommendations for Further Research

Time-scale normalization is a first-order approach to correcting depth effects resulting from signal dilation due to the geometry of a spherically-spreading wavefront intersecting the ocean bottom. However, the piecewise linear interpolation changes some of the characteristics of the signal, such as the noise spectra. Therefore, the change in an echosounder return as a function of depth should be thoroughly examined along with ways of compensating for the changes without introducing more distortion.

The feature extraction algorithms used in this thesis are very simple and provide varying amounts of separation. However, to improve results, additional feature extraction algorithms should be evaluated.

Feature reduction methods are currently based on the assumption that no information is known about the represented classes. However, after clustering, a hypothesis has been formed. Iteration should be considered with the class statistics provided by the clustering algorithm being used to determine a feature space reduction matrix with improved separation and clustering ability.

Bayesian classification assumes hard-limiting class boundaries which do not support the existence of overlapping classes. As a result simple statistics estimation can use skewed samples of the data and result in biased estimates. With the assumption of normally distributed data, correction of the cluster statistics should be possible.

Bibliography

[1] Mayer, L., Clarke, J.E.H., and Wells, D. A multi-faceted acoustic ground-truthing experiment in the Bay of Fundy. *Acoustic Classification and Seabed Mapping*, pages 203-219. Institute of Acoustics, Bath, U.K., 1993.

[2] Tan, Z., and Mayer, L. The reverberation models of sea bottom (draft). 1994.

[3] Tan, Z. *Seafloor classification from the data of acoustic reflections.* Technical Report, Ocean Mapping Group, UNB, 1994.

[4] Kristensen, J.H., and Pohner, F. ESMAC, a nordic research programme for environmental mapping and characterization of the seafloor. *Acoustic Classification and Seabed Mapping*, pages 131-139. Institute of Acoustics, Bath, U.K., 1993.

[5] Shippey, G., Vikgren, K., Elhammer, A., and Finndin, R. Design of a marine geographic information system for seabed mapping and classification. *Acoustic Classification and Seabed Mapping*, pages 195-202. Institute of Acoustics, Bath, U.K., 1993.

[6] Eggen, T.H. Acoustic sediment classification experiment by means of multibeam echosounders. 1993.

[7] Huseby, R.B., Milvang, O., Solberg, A., and Weisteen, K. Seabed classification from backscatter sonar data using statistical methods. *Acoustic Classification and Seabed Mapping*, pages 415-420. Institute of Acoustics, Bath, U.K., 1993.

[8] Kavli, T., Carlin, M., and Madsen, R. Seabed classification using artificial neural networks and other nonparametric methods. *Acoustic Classification and Seabed Mapping*, pages 141-148. Institute of Acoustics, Bath, U.K., 1993.

[9] Kavli, T., Carlin, M., and Weyer, E. Real-time seabed classification using multi-frequency echo sounders. *Proceedings of Oceanology International 94*, pages 1-9. Oceanology International, Brighton, U.K., 1994.

[10] Milvang, O., Ragnar, B.H., Weisteen, K., and Solberg, A. Feature extraction from backscatter sonar data. *Acoustic Classification and Seabed Mapping*, pages 157-164. Institute of Acoustics, Bath, U.K., 1993.

[11] Lambert, D.N. A new computerized single frequency seafloor classification system. *Unknown*, pages 99-105. NSTL, 1990.

[12] Lambert, D.N., Cranford, J.C., and Walter, D.J. Development of a high resolution acoustic seafloor classification survey system. *Proceedings of the Institute of Acoustics.* 15(2), 1993.

[13] Caughey, D.A., Prager, B., and Klymak, J. *Bottom Classification using Single-Frequency Echo Sounders.* Contractor's Report 94-56, Defense Research Establishment Pacific, 1994.

[14] Caughey, D.A., Prager, B., and Inkster, D.R. *Echo Sounder Seabed Classification Study.* Contractor's Report 95-22, Defense Research Establishment Pacific, 1995.

[15] Prager, B.T., Caughey, D.A., and Poeckert, R.H.,. Bottom Classification: Operational Results from QTC VIEW. *Proceedings of Oceans'95.* IEEE, San Diego, CA, 1995.

[16] Caughey, D.A., Kirlin, R.L. Blind Deconvolution of Echosounder Envelopes. *ICASSP'96.* IEEE, Atlanta, GA, 1996.

[17] Dietrich, G. *General Oceanography.* John Wiley & Sons, 1963.

[18] Jackson, D.R., and Nesbitt, E. Bottom classification using backscattering at vertical incidence. *Proceedings of Spring 1988 ASA Meeting.* Acoustical Society of America, Seattle, WA, 1988.

[19] Jackson, D.R., and Briggs, K.B. High-frequency bottom backscattering: Roughness versus sediment volume scattering. *Journal of the Acoustical Society of America.* 92(2):962-977, August, 1992.

[20] Mourard, P.D., and Jackson, D.R. High frequency sonar equation models for bottom backscatter and forward loss. *Proceedings of Oceans'89,* pages 1168-1175. IEEE, New York, NY, 1989.

[21] Jackson, D.R., Winebrenner, D.P., and Ishimaru, A., *Journal of the Acoustical Society of America.* 79:1410-1422, 1986.

[22] Jackson, D.R., Baird, A.M., Crisp, J.J., and Thomson, P.A.G., *Journal of the Acoustical Society of America.* 80:1188-1199, 1986.

[23] McDaniel, S.T., and Gorman, A.D., *Journal of the Acoustical Society of America.* 73:1476-1486, 1983.

[24] Poulinquen, P., and Lurton, X. Sea-bed identification using echo sounder signals. *European Conference on Underwater Acoustics,* pages 535-538. Commission of the European Communities, Luxembourg, 1992.

[25] Chivers, R.C., Emerson, N., and Burns, D.R. New Acoustic Processing for Underway Surveying. *The Hydrographic Journal.* (56):9-17, April, 1990.

[26] LeBlanc, L.R., Mayer, L., Rufino, M., Schock, S.G., and King, J. Marine sediment classification using the chirp sonar. *Journal of the Acoustical Society of America.* 91(1):107-115, January, 1992.

[27] Milligan, S.D., LeBlanc, L.R., and Middleton, F.H. Statistical grouping of acoustic reflection profiles. *Journal of the Acoustical Society of America.* 64(3):795-807, September, 1978.

[28] Dunsiger, A.D., Cochrane, N.A., and Vetter, W.J. Seabed characterization from broad-band acoustic echosounding with scattering models. *IEEE Transactions of Oceanic Engineering.* OE-6(3):94-106, July, 1981.

[29] Pace, N.G., and Ceen, R.V. Seabed classification using the backscattering of normally incident broadband acoustic pulses. *The Hydrographic Journal.* (26):9-16, October, 1982.

[30] Cochrane, N.A., and Dunsiger, A.D. Remote acoustic classification of marine sediments with application to offshore Newfoundland. *Canadian Journal of Earth Sciences.* 201195-1211, 1983.

[31] Orlowski, A. Application of multiple echoes energy measurements for evaluation of sea bottom type. *Oceanologia.* (19):61-78, 1984.

[32] Reut, Z., Pace, N.G., and Heaton, M.J.P. Computer classification of sea beds by sonar. *Nature.* 314(4):426-428, April, 1985.

[33] Pace, N.G., and Gao, H. Swathe seabed classification. *IEEE Journal of oceanic engineering.* 13(2):83-90, April, 1988.

[34] McCleave, B.W., Owens, J.K., and Ingles, F.M. Analyzing depth sounder signals with artificial neural networks. *Sea Technology.* 39-42, March, 1992.

[35] Alexandrou, D., and Pantzartzis, D. A methodology for acoustic seafloor classification. *unknown*, pages ??-??. unknown, 1993.

[36] Olvsheski. Unknown Title. *Unknown.* Unknown(Unknown):Unknown, Unknown, 1992.

[37] Zerr, B., Maillard, E., and Geuriot, D. Sea-floor classification by neural hybrid system. *Proceedings of Oceans'94*, pages 239-243. IEEE, Brest, 1994.

[38] Dugundji, J. Envelopes and Pre-Envelopes of Real Waveforms. *IRE Transactions on Information Theory.* 53-57, March, 1958.

[39] Rice, S.O. Envelopes of Narrow-Band Signals. *Proceedings of the IEEE.* 70(7):692-699, July, 1982.

[40] Boashash, B. Time-Frequency Signal Analysis. Haykin, S. (editor), *Advances in Spectrum Analysis and Array Processing*, pages 418-517. Prentice-Hall, 1991.

[41] Learned, R.E., Karl, W.C., and Willsky, A.S. Wavelet Packet Based Transient Signal Classification. *Proceeding of the International Symposium on Time-Frequency and Time-Scale Analysis*, pages 109-112. IEEE, Victoria, B.C., 1992.

[42] Tou, J.T., and Gonzales, R.C. *Pattern Recognition Principles.* Addison-Wesley, New York, 1974.

[43] Gersho, A., and Gray, R.M. *Vector Quanization and Signal Compression.* Kluwer Academic Publishers, Boston, 1992.

[44] Kohonen, T. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 1984.

[45] Kohonen, T., Masisara, K., and Saramaki, T. Phonotopic Maps -- Insightful Representation of Phonological Features for Speech Representation. *Proceedings of 7th Inter. Corf. on Pattern Recognition.* IEEE, Montreal, 1984.

[46] Lippmann, R.P. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine.* 4-20, April, 1987.

[47] Koontz, W.L.G., and Fukunaga, K. A Nonparametric Valley-Seeking Technique of Cluster Analysis. *IEE Transactions on Computers.* C-21(2):171-178, February, 1972.

[48] Koontz, W.L.G., and Fukunaga, K. Asymptotic Analysis of a Nonparametric Clustering Technique. *IEEE Transactions on Computers.* C-21(9):967-974, September, 1972.

[49] Gitman, I. An Algorithm for Nonsupervised Pattern Classification. *IEEE Transactions on Systems, Man, and Cybernetics.* SMC-3(1):66-74, January, 1973.

[50] Ball, G.H., and Hall, D.J. *ISODATA, A Novel Method of Data Analysis and Pattern Classification.* Technical Report AD699616, Stanford Research Institute, 1965.

[51] Bendat, J.S., and Piersol, A.G. *Random Data: Analysis and Measurement Procedures.* John Wiley & Sons, New York, 1986.

[52] Watanabe, S. *Pattern Recognition: Human and Mechanical.* Wiley-Interscience, New York, 1985.

[53] Velis, D.R., and Ulrych, T.J. Simulated Annealing Wavelet Estimation via Fourth-Order Cumulant Matching. 1995.

[54] Haykin, S. *Neural Networks: A Comprehensive Foundation.* Macmillan College Publishing, New York, 1994.

[55] Mathews, J.V., Adaptive Polynomial Filters, IEEE SP Magazine, pp.10-26, July 1991

A Aggregate Variance Estimation

Given a class whose mean and (biased) variance have been estimated as follows:

$$m_i = \frac{1}{N_i}\sum_j^{N_i} x_{i,j} \qquad \sigma_i^2 = \frac{1}{N_i}\sum_j^{N_i}(x_{i,j} - m_i)^2 \tag{A.1}$$

The estimates of the mean and variance of the aggregation of two such classes can be written as follows. First, define an aggregate data vector

$$\boldsymbol{x}_{agg} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \tag{A.2}$$

Then define the mean of this vector

$$\begin{aligned} m_{agg} &= \frac{1}{N_{agg}}\sum_j^{N_{agg}} x_{agg,j} \\ &= \frac{1}{N_1 + N_2}\left(\sum_j^{N_1} x_{1,j} + \sum_j^{N_2} x_{2,j}\right) \\ &= \frac{N_1 m_1 + N_2 m_2}{N_1 + N_2} \end{aligned} \tag{A.3}$$

The expression of the aggregate variance can then be expressed in terms of σ_1^2 and σ_2^2, and m_1 and m_2.

$$\begin{aligned} \sigma_{agg}^2 &= \frac{1}{N_{agg}}\sum_j^{N_{agg}} (x_{agg,j} - m_{agg})^2 \\ &= \frac{1}{N_1 + N_2}\left(\sum_j^{N_1}(x_{1,j} - m_{agg})^2 + \sum_j^{N_2}(x_{2,j} - m_{agg})^2\right) \end{aligned} \tag{A.4}$$

where each summation of Eq. (A.4) can be expressed as follows:

$$\begin{aligned}\sum_{j}^{N_i}(x_{i,j}-m_{agg})^2 &= \sum_{j}^{N_i}\left(x_{i,j}-\frac{N_1m_1+N_2m_2}{N_1+N_2}\right)^2 \\ &= \sum_{j}^{N_i}x_{i,j}^2-2N_im_i\frac{N_1m_1+N_2m_2}{N_1+N_2}+N_i\frac{(N_1m_1+N_2m_2)^2}{(N_1+N_2)^2} \\ &= N_i\sigma_i^2-2N_im_i\frac{N_1m_1+N_2m_2}{N_1+N_2}+N_i\frac{(N_1m_1+N_2m_2)^2}{(N_1+N_2)^2}+N_im_i^2\end{aligned} \tag{A.5}$$

Reincorporating Eq. (A.5) into the expression of the aggregate variance results in the following

$$\begin{aligned}\sigma_{agg}^2 &= \frac{1}{N_1+N_2}\left(N_1\sigma_1^2-2N_1m_1\frac{N_1m_1+N_2m_2}{N_1+N_2}+N_1\frac{(N_1m_1+N_2m_2)^2}{(N_1+N_2)^2}+N_1m_1^2\right) \\ &+ \frac{1}{N_1+N_2}\left(N_2\sigma_2^2-2N_2m_2\frac{N_1m_1+N_2m_2}{N_1+N_2}+N_2\frac{(N_1m_1+N_2m_2)^2}{(N_1+N_2)^2}+N_2m_2^2\right) \\ &= \frac{1}{N_1+N_2}\left(N_1\sigma_1^2+N_2\sigma_2^2-2(N_1m_1+N_2m_2)\frac{N_1m_1+N_2m_2}{N_1+N_2}\right. \\ &\qquad \left.+(N_1+N_2)\frac{(N_1m_1+N_2m_2)^2}{(N_1+N_2)^2}+(N_1m_1^2+N_2m_2^2)\right) \\ &= \frac{N_1\sigma_1^2+N_2\sigma_2^2}{N_1+N_2}+\frac{(N_1+N_2)(N_1m_1^2+N_2m_2^2)-(N_1m_1+N_2m_2)^2}{(N_1+N_2)^2}\end{aligned} \tag{A.6}$$

When the expressions involving the means are expanded and grouped, the following final expression is produced

$$\sigma_{agg}^2 = \frac{N_1\sigma_1^2+N_2\sigma_2^2}{N_1+N_2}+\frac{N_1N_2}{(N_1+N_2)^2}(m_1-m_2)^2 \tag{A.7}$$

B Incremental Statistical Estimates

Given a class whose mean and variance have been estimated as follows:

$$m^{(N)} = \frac{1}{N}\sum_{j}^{N} x_j \qquad \sigma^{2(N)} = \frac{1}{N-1}\sum_{j}^{N}(x_j - m^{(N)})^2 \tag{B.1}$$

The estimates of the mean and variance of the class given one more sample can be expressed as follows

$$\begin{aligned} m^{(N+1)} &= \frac{1}{N+1}\left(\sum_{j}^{N} x_j + x_{N+1}\right) \\ &= \frac{N}{N+1}m^{(N)} + \frac{1}{N+1}x_{N+1} \end{aligned} \tag{B.2}$$

and

$$\begin{aligned} \sigma^{2(N+1)} &= \frac{1}{N}\sum_{j}^{N+1}(x_j - m^{(N+1)})^2 \\ &= \frac{1}{N}\sum_{j}^{N}\left(x_j - \frac{N}{N+1}m^{(N)} - \frac{x_{N+1}}{N+1}\right)^2 + \frac{1}{N}\left(x_{N+1} - \frac{N}{N+1}m^{(N)} - \frac{x_{N+1}}{N+1}\right)^2 \\ &= \frac{1}{N}\sum_{j}^{N}\left(x_j - m^{(N)} + \frac{m^{(N)} - x_{N+1}}{N+1}\right)^2 + \frac{N}{(N+1)^2}(x_{N+1} - m^{(N)})^2 \\ &= \frac{1}{N}\sum_{j}^{N}\left((x_j - m^{(N)})^2 + 2(x_j - m^{(N)})\frac{m^{(N)} - x_{N+1}}{N+1} + \left(\frac{m^{(N)} - x_{N+1}}{N+1}\right)^2\right) \\ &\quad + \frac{N}{(N+1)^2}(x_{N+1} - m^{(N)})^2 \\ &= \frac{N-1}{N}\sigma^{2(N)} + \left(\frac{m^{(N)} - x_{N+1}}{N+1}\right)^2 + \frac{N}{(N+1)^2}(x_{N+1} - m^{(N)})^2 \\ &= \frac{N-1}{N}\sigma^{2(N)} + \frac{(m^{(N)} - x_{N+1})^2}{N+1} \end{aligned} \tag{B.3}$$

C Stereo Viewing Instructions

The following excerpt is taken from the sidebar on page 29 of the August 1989 Computer magazine[1].

> The stereo pairs in this article are presented for cross-fused viewing with the left eye's view on the right and the right eye's view on the left. The pairs are most easily viewed using special goggles or mirrors, but most people can view them directly with a little patience and practice. The pairs are best viewed at a distance between 16 and 32 inches. Closer viewing is possible, but more stressful because it requires you to cross your eyes more.
>
> Cross your eyes until you see four objects, then slowly reduce the amount of crossing until the middle two images fuse into one. The merged image is usually blurred at first, but you can bring it into focus by slowly relaxing your eyes' focus to infinity while keeping the two middle images merged.
>
> Some people find that holding an index finger down along the bridge of the nose or moving it between the eyes and the page help achieve the proper merging and focusing. It also helps if you frame the images with black paper. If one image appears higher on the page than the other, rotate the page until they merge. It takes some practice, but once the images are merged and in focus, you can easily maintain the viewing.

---

1. Helman, J., and Hesselink, L. Representation and Display of Vector Field Topology in Fluid Flow Data Sets, *Computer*, 22(8):27-36, August 1989

D Pattern Recognition Toolbox Reference Pages

The following information is used with permission of Ahlea Systems Corp.

boundary

Purpose

Calculate the decision boundary between 2-D gaussian classes.

Synopsis

b = boundary(m_1, c_1, m_2, c_2, pw_1, pw_2, T)

Description

Two Gaussian densities have a quadratic boundary which is a parabola, an ellipse, or a hyperbola (or in degenerate cases, one or two straight lines).

This function calculates the decision boundary between two such Gaussian densities described by their means m_1, m_2 and covariances c_1, c_2. The curves are calculated at T points representing the solution to the quadratic equation generated by equating the logarithm of the likelihood functions for the two densities, that is:

$$\log p(\omega_i)p(\boldsymbol{x}|\omega_i) = \log p(\omega_j)p(\boldsymbol{x}|\omega_j)$$

where the likelihood function is given by

$$p(\omega_i)p(\boldsymbol{x}|\omega_i) = \frac{p(\omega_i)}{\sqrt{2\pi}^M |C_i|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\mu_i)C_i^{-1}(\boldsymbol{x}-\mu_i)}$$

This is *not* intended to be a user-callable routine. It is used exclusively by plotboundary.

See Also

plotboundary

certain

Purpose

Calculate the scaled certainties of classifications.

Synopsis

e = certain(X, M, C)

Description

Calculate the certainties of classification for each point in X given the multi-class statistics M and C.

A row vector will be returned where each element corresponds to the certainty for the appropriate classification.

The certainty is a number $0 < e <= 1$, where the probabilities have been scaled to the minimum of 0.5 or the lowest classified certainty.

See Also

fuzzclassify, likelihood

chi2

Purpose

Calculate the chi-squared statistics.

Synopsis

X2 = chi2(Xl [,'raw'])
X2 = chi2(X, K, deltax [,'raw'])

Description

This function implements a χ^2 test, as per "Random Data" by Bendat and Piersol.

The default values for K = 12 and deltax = 0.4.

If 'raw' is specified then the data is assumed to be normal, with zero mean and unit variance. Otherwise the data is transformed so that it will be. This allows a value of s = 1 (as per Bendat and Piersol).

See Also

normalize

class_add

Purpose

Add class statistics to multi-class lists of statistics.

Synopsis

$[M_{new}, C_{new}]$ = class_add(M_{old}, C_{old}, m_{add}, c_{add} [,index])

Description

This function inserts single-class stats (or multi-class lists) defined by mean m_{add} and covariance c_{add} to an existing list (M_{old},C_{old}).

The default action is to append the new class(es), but if index is specified then the class(es) is(are) inserted.

class_delete

Purpose

Delete class statistics from multi-class lists of statistics.

Synopsis

$[M_{new}, C_{new}]$ = class_delete(M_{old}, C_{old}, index)

Description

This function deletes the class(es) defined by index from an existing list (M_{old}, C_{old}).

class_extract

Purpose

Extract class statistics from multi-class lists of statistics.

Synopsis

$[M_{new}, C_{new}]$ = class_extract(M_{old}, C_{old}, index)

Description

This function extracts a class defined by index from an existing list (M_{old},C_{old}).

If index is a vector, multi-class lists are returned, otherwise single-class statistics are returned.

class_join

Purpose

Join members of multi-class lists of statistics.

Synopsis

$[M_{new}, C_{new}]$ = class_join(M_{old}, C_{old}, indices)

Description

This function joins statistical distributions from two parts. The source distribution described by the multi-class statistics M_{old} and C_{old}, with the joinable class specified by indices.

class_map

Purpose

Generate a map relating two membership lists.

Synopsis

[map, imap] = class_map($member_1$, $member_2$)

Description

This function defines the mapping between two membership lists.

Thus, if a clustering algorithm has re-assigned the class numbers, this function will find the closest fit.

The two return vectors can be used to map multi-class lists of statistics as following:

```
m2 <- m1(imap,:);
c2 <- c1(imap,:)
member2 <- map(member1)
```

or conversely,

```
m1 <- m2(map,:);
c1 <- c2(map,:)
member1 <- imap(member2)
```

Notes

It is assumed that the two membership lists are reasonably close to begin with.

See Also

score

class_multi

Purpose

Force class statistics into multi-class lists of statistics.

Synopsis

[M_{new}, C_{new}] = class_multi(M_{old}, C_{old})

Description

This function eliminates ambiguity over the dimensionality and number of classes that can exist when examining only either the means or covariances alone. It does this by converting single class statistics defined by (M_{old},C_{old}) into their multi-class lists representations.

If (M_{old},C_{old}) are already multi-class lists, then nothing is done.

class_rand

Purpose

Generate random multi-class lists of statistics.

Synopsis

[M_{new},C_{new}] = class_rand(K, M, Q)

Description

This function generates a multi-class list of K M-dimensional classes. The maximum value for any eigenvalue is given by Q. M must be less than 3.

class_split

Purpose

Divide member of multi-class lists of statistics.

Synopsis

[M_{new},C_{new}] = class_split(M_{old}, C_{old} [,index])

Description

This function splits a statistical distribution into two parts. The source distribution described by M_{old} and C_{old} can either use a single class statistics, or use entries from a multi-class list with the splitable class specified by index.

classify

Purpose

Associate feature vectors with Gaussian-distributed classes.

Synopsis

member = classify(X, M, C [,PW])

Description

This function performs Gaussian classification on the data X described by their multi-class statistics M and C. Results are returned as a vector of memberships. One can pass in an option array of the class probabilities.

Classification is performed by calculating the Bayesian distances and then selecting the class having the minimum distance (i.e., highest likelihood).

See Also

dbayes, likelihood

cluster

Purpose

Perform non-parametric clustering on Gaussian data.

Synopsis

[M, C] = cluster(X [, <options>])

Description

This function tries to cluster the points in X (defined as columns vectors). It returns the multi-class statistics M and C of the discovered clusters.

Adding the option 'graphics' will cause pretty plots to be displayed. (This is meaningful only for 2-D or 3-D data!)

Options are:

'graphics'		turns on graphics
'max'	<int>	expected max # of classes
'amax'	<int>	absolute max # of classes
'thresh'	<value>	used in class division
'init'	m,c	initial means and covs
'member'	<vector>	initial membership

coveig

Purpose

Perform eigenanalysis on the covariance matrix of a data set.

Synopsis

[v, d, cx] = coveig(X)

Description

This function performs eigenanalysis on a matrix of feature vectors X.

The eigenvectors v and eigenvalues d are returned sorted by descending eigenvalue. The covariance matrix is returned in cx.

dbayes

Purpose

Calculate distance of a points to Gaussian clusters.

Synopsis

d = dbayes(x, M, C [,PW])

Description

This function returns the squared Bayesian distances for each point in X to clusters described by the multi-class statistics M and C. Similarity transforms are used to normalize units of comparison.

The likelihood that a vector belongs to class *i* is the product of the probability of class *i* and the conditional probability density of the vector. This can be expressed as a function of the Bayesian distance d as follows:

$$
\begin{aligned}
p(\omega_i)p(\boldsymbol{x}|\omega_i) &= \frac{p(\omega_i)}{\sqrt{2\pi}^M |\boldsymbol{C}_i|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\mu_i)\boldsymbol{C}_i^{-1}(\boldsymbol{x}-\mu_i)} \\
&= \frac{1}{\sqrt{2\pi}^M} e^{-\frac{1}{2}(\boldsymbol{x}-\mu_i)\boldsymbol{C}_i^{-1}(\boldsymbol{x}-\mu_i) - \frac{1}{2}\log|\boldsymbol{C}_i| + \log p(\omega_i)} \\
&= \frac{1}{\sqrt{2\pi}^M} e^{-d}
\end{aligned}
$$

where μ_i and C_i are extracted for each class from the multi-class statistics M and C.

From this number, one can calculate the distance of a point to a class as a function of standard deviations by

$$\sigma_i = d - \frac{1}{2}\log|\boldsymbol{C}_i| + \log p(\omega_i)$$

however it should be noted that classifications are based likelihoods and not strictly the number of standard deviations.

If pw (the *a priori* probability of each of the classes) is not specified, the values are assume uniform.

See Also

mahalanobis, likelihood, pdf

divergence

Purpose

Calculate the divergence between Gaussian classes.

Synopsis

J = divergence(M, C)
J = divergence(m_1, c_1, m_2, c_2)

Description

This function returns the divergence of two distributions described by their multi-class statistics M and C, or their single-class statistics (m_1,c_1) and (m_2,c_2).

entropy

Purpose

Calculate entropy of a process (assumed Gaussian).

Synopsis

H = entropy(X, M, C, delta)

Description

This function calculates the entropy of X supposing that it came from a theoretical Gaussian distribution having a p.d.f. described by the multi-class statistics M and C.

Note that entropy uses probabilities, not probability densities. Unfortunately, all we know is the p.d.f. Therefore, this routine uses the mapping

```
H(x) = A*(Hpdf(x) - log2(A)*sum(pdf(x)));
```

where

```
Hpdf(x) = -sum(pdf(x).*log2(pdf(x)));
```

The scaling factor A is used to 'discretize' the continuous p.d.f. as required for the entropy calculation. The parameter delta is used to determine the sampling of the p.d.f. (actually, the integration range). As delta approaches zero, so will the calculated probabilities, and hence, so will the entropy.

estimate

Purpose

Estimate the statistics of data sets.

Synopsis

[M, C] = estimate(X, member)

Description

This function performs estimates the means and covariances of the column vectors of X for the classes defined by member.

The results of class *i* are returned as rows within the M and C matrices. The covariance has been reshape'd to a row vector.

If the data set X contains no representatives of a class, (i.e., K > i, but find(member==i)==[]), then estimate returns NaN (not-a-number) as the estimates of statistics of the missing classes.

If the data set X contains fewer representatives of a class than there are dimensions, then the covariance matrix will be of insufficient rank, and estimate will print out a warning.

fillclassify

Purpose

Grid and colour the current figure according to classification.

Synopsis

fillclassify(M, C [,N])
[G,X] = fillclassify(M, C [,N])

Description

This routine classifies every point in the current axis. It does this by gridding (see meshgrid) the feature space according to the current axis limits. The number of grid points along the *x*-axis is given by N (default value is 80), and the *y*-axis is interpolated at a proportionate number of locations (depending on the aspect ratio of the axis).

Unfortunately, the axes limits must already be defined.

If no output argument is specified, then classified interpolation matrix is plotted using image, and a gray scale colour map.

Notes

This function uses the undocumented (and potentially obseletable) axis attribute RenderLimits.

See Also

meshgrid, griddata, classify, plotclasses

fuzzclassify

Purpose

Calculate the certainties of classifications.

Synopsis

e = fuzzclassify(X, M, C [,PW])

Description

Calculate the certainties of classification for each point in X given the stats M and C.

The certainties of the classifications are simply the relative likelihoods. That is the likelihoods for each class are calculated, and the relative likelihood is given by

$$e_i(x) = \frac{l_i(x)}{\sum_j l_j(x)}$$

The relative likelihood will vary from zero to unity.

See Also

certain, likelihood

kmeans

Purpose

K-means clustering algorithm.

Synopsis

[member,M] = kmeans(X, K)
[member,M] = kmeans(X, Z)

Description

This function implements the K-means clustering algorithm. X is the *MxN* data matrix, where *M* is the number of dimensions and *N* is the number of vectors. X may contain complex data.

If the second argument is a scalar, then it is interpreted as K — the number of desired clusters.

The second argument can also be interpreted as Z — the initial estimate of the cluster centres, and the number of clusters is inferred from the size of Z. In this case Z is a *KxM* matrix.

A row vector associating each sample with one of the K classes is returned in member, and the final estimates of the means are return in M. If a cluster in Z is not represented by at least one vectior in X, then the original estimate of the mean for that cluster is returned.

See Also

cluster

likelihood

Purpose

Calculate likelihoods for Gaussian distributions.

Synopsis

p = likelihood(X, M, C, PW)

Description

This function returns the likelihood $p(\omega_i)p(x|\omega_i)$, where

$$p(x|\omega_i) = \frac{1}{\sqrt{2\pi}^M |C_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)C_i^{-1}(x-\mu_i)}$$

and where μ_i and C_i are extracted for each class from the multi-class statistics M and C. The result is returned in a KxN matrix p, where K is the number of classes and N is the number of samples.

Note, the class probabilities must be specified.

See Also

pdf, dbayes

mahalanobis

Purpose

Calculate the Mahalanobis distances.

Synopsis

d = mahalanobis(X, m, c)
d = mahalanobis(X)

Description

The function is used to calculate the distances from every point in X to each of the classes described by the means and covariances. If the means and covariances are

not specified they are estimated from the data. The Mahalanobis distance is given as

$$h_{ij} = (x_j - \mu_i)C_i^{-1}(x_j - \mu_i)$$

A $K \times N$ matrix is returned.

Using an identity covariance matrix will generate Euclidean distances.

See Also

dbayes

maxi

Purpose

Index of the largest component.

Synopsis

I = maxi(X)

Description

This function returns the index of the maximum element of X, as defined by the built-in max command.

See Also

max, mini

mini

Purpose

Index of the smallest component.

Synopsis

I = mini(X)

Description

This function returns the index of the minimum element of X, as defined by the built-in min command.

See Also

min, maxi

normalize

Purpose

Normalize data set to zero-mean, unit-variance.

Synopsis

```
Y = normalize( X )
[Y, m, c] = normalize( X )
```

Description

This function normalizes an M-dimensional distribution such that it is zero-mean, unit variance.

It does this by making use of a similarity transform, such that

$$\mu = E\{x\}$$
$$C = E\{(x-\mu)(x-\mu)^{\mathrm{H}}\}$$
$$y = \Lambda^{-1/2} V^{\mathrm{H}}(x-\mu)$$

where Λ and V are the eigenvalues and eigenvectors, respectively, of the estimated covariance c.

pdf

Purpose

Calculate values of p.d.f. for Gaussian distributions.

Synopsis

```
p = pdf( X, M, C )
```

Description

This function returns the probability density function

$$p(x|\omega_i) = \frac{1}{\sqrt{2\pi}^{M}|C_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)C_i^{-1}(x-\mu_i)}$$

where μ_i and C_i are extracted for each class from the multi-class statistics M and C. The result is returned in a KxN matrix p, where K is the number of classes and N is the number of samples.

Note, class probabilities are not considered.

See Also

likelihood, dbayes

plotboundary

Purpose

Calculate and plot the boundaries between Gaussian classes.

Synopsis

plotboundary(M, C [,PW] [,T] [,LS_1,LS_2])

Description

This function plots the boundaries between the classes defined by their multi-class statistics M and C.

T is an optional number (default 200) of ticks used to generate the curves. A higher value of T yields better intersections, but takes longer. LS_1 and LS_2 are the primary and secondary boundary line styles (either both or none must be specified, although one or both may be an empty string).

PW are the class probabilities.

See Also

plotellipse, plotclasses, plotoutliers

plotclasses

Purpose

Pretty plot of multi-class data.

Synopsis

h = plotclasses(X, M, C [,member] [,member2] [,LS_1,LS_2])
h = plotclasses(X, M, C [,member] [,member2], 'simple')

Description

This function generates pretty plots of the data in either 2-D or 3-D feature space, with the data X "classified" according to the membership list member. If member is not specified then Bayesian classification is performed.

If X is 2-D, then the decision boundaries are plotted along with the data, unless 'simple' is specified.

If X is 3-D, then the 1-std ellipsoids are also plotted, unless 'simple' is specified.

LS_1 and LS_2 refer to the optional line styles used by plotboundary (called only if 'simple' is not specified.)

If member2 is specified then it is used to determine the class probabilities required to accurately determine 2-D decision boundaries. The default is to use uniform probabilities.

See Also

fillclassify, plotellipse, plotboundary, plotoutliers

plotellipse

Purpose

Draw ellipse(s)/ellipsoid(s) described by means and covariances.

Synopsis

h = plotellipse(M, C [,linestyle] [,T])
[x, y] = plotellipse(M, C [,linestyle] [,T])
[x, y, z] = plotellipse(M, C [,linestyle] [,T])

Description

This function draws a 2-D ellipse or 3-D ellipsoid centred at M with covariance C, using an optional linestyle.

M and C can describe either a single ellipse/ellipsoid or designate multi-class lists, in which case linestyle applies to all drawn ellipses/ellipsoids.

If a single output argument is specified then the handles for each line element drawn are returned in h. Otherwise the x, y (and possibly, z) coordinates of the last ellipse/ellipsoid are returned.

See Also

plotboundary, plotclasses, plotoutliers

plotoutliers

Purpose

Plot outliers boundaries.

Synopsis

h = plotoutliers(M, C [,member] [,D] [,linestyle])

Description

This function plots the boundaries which define the outliers (given by the D standard deviation) of the multi-class statistics M and C.

If member is specified then it is used to determine the class probabilities required to accurately determine 2-D decision boundaries. The default is to use uniform probabilities.

See Also

plotellipse, plotboundary, plotclasses

randnorm

Purpose

Generate some Gaussian data.

Synopsis

[X, member] = randnorm(N, M, C)

Description

This function generates normally distributed points such the data have statistics defined by M and C.

If M and C are multi-class descriptors, then N points are generated for each class. N may also be a vector of sizes.

The data are returned in X, with the corresponding membership in member.

reduce

Purpose

Reduce dimensionality of data set.

Synopsis

$X_{reduced}$ = reduce(X, v, d, N, [balance])

Description

This function reduces the dimensionality of a vector space X to N using the eigenvalues d and vectors v.

This routine assumes values and vectors have been sorted in descending order.

If balance=1 (default), then the eigenvectors are scaled by the inverse square root of the eigenvalues, yielding an $X_{reduced}$ which has a spherical distribution with unit variance.

See Also

coveig

score

Purpose

Compare calculated versus known classifications.

Synopsis

sc = score(member, $member_{true}$)

Description

This function compares the member results of the classify function against the known $member_{true}$ list.